POLITECNICO DI TORINO

Laurea Magistrale in Ingegneria Gestionale

Tesi di laurea Magistrale

Studio sulle metodologie Data Driven e sugli algoritmi di AI impiegati nella predizione dello stato di povertà delle famiglie.

Caso studio nel sistema economico delle Filippine

Relatrice	Candidato
Prof. Tania CERQUITELLI	Francesco Mario PISCOPO

6 Dicembre 2023

Sommario

L'identificazione e la predizione dello stato di povertà familiare costituiscono un compito fondamentale per il progresso socio-economico, nonché una sfida globale sempre più centrale, come dimostrato dalla presenza di tale tematica al primo posto nei 17 Obiettivi dell'Agenda 2030 dello Sviluppo Sostenibile.

Dall'avvento dell' AI (Artificial Intelligence), numerose sono state le sue applicazioni in diversi campi, tra cui, inevitabilmente, anche nella lotta alla povertà. Il numero di studi effettuati negli ultimi anni in tale ambito è in forte crescita, e sta raccogliendo sempre più interesse tra gli studiosi della comunità scientifica.

Il presente studio esamina l'efficienza delle metodologie Data-Driven e degli algoritmi di Intelligenza Artificiale impiegati in letteratura nell'ambito della predizione della povertà. In particolare, nella parte centrale del lavoro, si effettua uno studio mirato alla predizione dello stato di povertà delle famiglie nelle Filippine, un contesto caratterizzato da una complessa tessitura geografica e socioculturale, mediante l'impiego di differenti tecniche di Machine Learning, tra cui modelli di classificazione e di regressione, partendo da variabili di natura differente, come informazioni circa il reddito e le abitudini di spesa delle famiglie, ed informazioni riguardo la provenienza geografica e la composizione delle famiglie stesse.

Indice

Elenco delle tabelle

Elenco delle figure

Acronimi

AI

Artificial Intelligence

ARMM

Regione Autonoma nel Mindanao Musulmano

AUC

Area Sotto la Curva

IQR

Interquartile Range (Scarto Interquartile)

KDD

Knowledge Discovery in Databases

LightGBM

Light Gradient Boosting Machine

ML

Machine Learning

MSE

Mean Squared Error (Errore Quadratico Medio)

NCR

National Capital Region

ONU

Organizzazione delle Nazioni Unite

PHP

Peso Filippino

PSA

Philippine Statistics Authority

R^2

Coefficiente di determinazione R^2

RBF

Radial Basis Function Kernel

RF

Random Forest

RMSE

Root Mean Square Error (Radice dell'Errore Quadratico Medio)

ROC

Curva Caratteristica Operativa Ricevitore

SVM

Support Vector Machine

THI

Total Household Income

Capitolo 1

Introduzione

La lotta alla povertà è una delle sfide più urgenti e complesse che, al giorno d'oggi, l'umanità è chiamata ad affrontare. Diverse organizzazioni internazionali e governi, pertanto, operano con l'obiettivo di porre fine a tale problematica. Secondo le più recenti stime effettuate dall'ente internazionale non governativo Action Aid sono circa 902 milioni le persone che, nel mondo, vivono in condizioni di estrema povertà, e dunque con meno di 1.90 $ al giorno [1]; un dato a dir poco sconcertante. Per comprendere ancora meglio l'importanza che tale tematica ricopre nello scenario globale, basti pensare che le Nazioni Unite, nello stilare l'*Agenda 2030 per lo Sviluppo Sostenibile*, nel 2015, hanno inserito come primo dei diciassette obiettivi presentati quello di *"Porre fine ad ogni forma di povertà nel mondo"* [2].

Prima di entrare nel dettaglio del seguente lavoro di tesi, è doveroso, dunque, illustrare il concetto di povertà, e comprendere il suo vero significato. La definizione iniziale proposta negli Obiettivi del Millennio dall'ONU (Organizzazione delle Nazioni Unite), nel 2000, trasformati poi negli Obiettivi dell'Agenda 2030 per lo Sviluppo Sostenibile, è quella che vede la povertà come la condizione di chi vive con meno di un 1 $ al giorno [3], ad oggi aggiornata con il valore, precedentemente citato, di 1.90 $ al giorno. Dunque, nell'originaria definizione ufficiale di povertà assoluta si tiene conto unicamente della disponibilità di denaro necessario a soddisfare i bisogni primari, di cibo, per i vestiti e per l'abitazione [3], e dunque viene fornita unicamente una lettura in chiave economica, con diretto riferimento al reddito e ai consumi della persona. Nella definizione attuale e più moderna di povertà, invece, si fa riferimento a tutta una serie di problematiche che sono strettamente legate ad essa, e che sono da essa inscindibili, come ad esempio la disuguaglianza sociale che può essere letta contemporaneamente come causa ed effetto della povertà stessa, così come la difficoltà di accesso ai servizi primari, ad esempio strutture sanitarie adeguate, strutture per l'istruzione, acqua potabile, cibo nutriente, una abitazione adeguata ed altre ancora [3].

In questo scenario, l'impiego delle metodologie Data-Driven e degli algoritmi di

AI (Artificial Intelligence) per la predizione e l'identificazione dello stato di povertà rappresenta un campo di ricerca e applicazione di enorme interesse. L'approccio Data-Driven, ovvero basato sull'analisi di ampie quantità di dati, consente di individuare con maggiore precisione i fattori che influenzano la povertà e di sviluppare strategie mirate per il suo contrasto. Tale approccio, in particolare negli ultimi anni, è stato ampiamente esplorato in letteratura, ed è stato oggetto di numerosi studi di ricerca, di cui solamente una piccola porzione è di seguito citata a titolo di esempio[4], [5], [6].

Particolarmente significativo in questo ambito è l'impiego di tecniche di ML (Machine Learning), che includono modelli di clustering, di classificazione e di regressione, tra i tanti. Questi strumenti si rivelano essenziali per analizzare e interpretare complesse relazioni tra le variabili in gioco, come reddito, abitudini di consumo, provenienza geografica, composizione delle famiglie e numerosi altri. In aggiunta a questa tipologia di dati provenienti prevalentemente da statistiche e sondaggi nazionali o internazionali, gli studi più recenti si soffermano sull'impiego di dati di natura differente, come ad esempio immagini satellitari e dati geo-spaziali [7], [8], [9], andando a testimoniare l'incredibile potenzialità del Machine Learning, e dunque la possibilità che esso offre di poter affrontare un problema cruciale come quello della povertà in maniera innovativa, sfruttando tutta una serie di fonti di dati che i modelli econometrici ampiamente utilizzati fino a prima dell'avvento dell'AI, non erano in grado di sfruttare. L'uso di algoritmi di AI, in questo contesto, apre, dunque, a nuove prospettive per la predizione dello stato di povertà, permettendo un'analisi ancor più mirata, precisa e fortemente personalizzata ed adattabile alle diverse realtà territoriali in cui il problema della povertà è predominante.

Il presente lavoro si propone, quindi, di esplorare in dettaglio come le metodologie Data-Driven e gli algoritmi di AI possano contribuire a risolvere la sfida globale della povertà. In particolare, lo studio è strutturato nel seguente modo: nel Capitolo 2 si propone una spiegazione dettagliata del processo di estrazione di conoscenza dai dati, KDD (Knowledge Discovery in Databases), evidenziando gli step necessari da eseguire e presentando una lista di algoritmi di ML adatti per gli scopi preposti. In secondo luogo, nel Capitolo 3, si riporta lo stato dell'arte in merito agli studi effettuati sulla predizione della povertà mediante l'applicazione di modelli di ML, evidenziando i risultati più interessanti fino ad ora ottenuti, la tipologia di dati utilizzati e soprattutto gli algoritmi di ML maggiormente impiegati e maggiormente efficienti. Il cuore del presente lavoro è, però, il Capitolo 4, in cui si illustra, passo dopo passo, lo studio di ricerca effettuato per predire lo stato di povertà delle famiglie nel contesto socioeconomico delle Filippine. Esso fornisce un esempio concreto dell'applicabilità e dell'efficacia di queste metodologie in uno scenario complesso e diversificato; la situazione socioeconomica e geografica delle Filippine, infatti, rende il Paese un caso studio rilevante, in cui l'impiego di metodologie Data-Driven e di algoritmi di AI può fornire spunti significativi per la comprensione

e il contrasto del fenomeno della povertà. In particolare, si propone una profonda analisi esplorativa dei dati, mirata alla comprensione del contesto di lavoro e delle variabili a disposizione, seguito da una attenta fase di preprocessing dei dati stessi, e dunque della loro preparazione per l'applicazione dei modelli di Machine Learning selezionati. Lo studio si conclude con la valutazione delle loro performance e della loro efficienza in merito alla predizione. Il lavoro di tesi termina, dunque, con il Capitolo 5, in cui si propongono riflessioni e spunti per migliorare il lavoro svolto e si discutono potenzialità e sviluppi futuri sul tema trattato.

Capitolo 2

Estrazione di conoscenza da database - KDD

Nel contesto economico e tecnologico attuale, sinteticamente contestualizzato nel capitolo 1, le basi di dati sono diventate depositi incommensurabili di informazioni. Tuttavia, queste immense riserve di dati, se non opportunamente rielaborate rischiano di essere di difficile interpretazione: ciò che è fondamentale è la capacità di estrarre conoscenza significativa da esse. Questo processo di estrazione e trasformazione dei dati in informazioni utili e comprensibili è definito come *KDD (Knowledge Discovery in Databases)*.

Il KDD non si limita alla semplice analisi dei dati, ma consiste di una serie di passaggi complessi, che vanno dalla preparazione e pulizia dei dati, alla loro analisi attraverso metodi statistici e algoritmi di apprendimento automatico, fino alla visualizzazione e interpretazione dei risultati ottenuti [10]. In effetti, il termine "Knowledge Discovery" sottolinea l'importanza di andare oltre l'analisi superficiale, cercando pattern, relazioni e anomalie che possano offrire una visione profonda e concreta dei dati in esame.

Un componente fondamentale del KDD è la fase di data mining, in cui si applicano tecniche e strumenti specifici per identificare e svelare tali pattern nei dati. Tuttavia, è cruciale comprendere che il data mining è solo una fase del processo complessivo di KDD, sebbene sia spesso utilizzato come sinonimo [11].

In questo capitolo, si esplorerà in profondità il concetto di KDD, delineando le sue fasi chiave, le sfide associate e le principali tecniche utilizzate.

Il processo KDD mira all'analisi di grandi volumi di dati per estrarre modelli e informazioni rilevanti, convertendo dati di basso livello in conoscenza di alto livello. Esso, illustrato in figura 2.1, può essere delineato attraverso una serie di fasi sequenziali:

1. **Data Exploration**: analisi preliminare della struttura del dataset, mirata alla

comprensione della distribuzione delle variabili e delle relazioni che intercorrono tra esse;

2. **Data Preprocessing**: ottimizzazione dei dati per il Data Mining, che comprende:

   (a) **Data Integration**: armonizzazione di diverse fonti di dati;

   (b) **Data Cleaning**: eliminazione di rumori e incongruenze;

   (c) **Data Selection**: selezione di dati pertinenti, dal dataset;

   (d) **Data Transformation**: operazioni di normalizzazione, aggregazione e discretizzazione per facilitare l'estrazione informativa.

3. **Data Mining**: applicazione di algoritmi specifici per la deduzione di conoscenza dai dati.

4. **Pattern Evaluation**: valutazione di modelli basata su metriche predefinite per identificare informazioni salienti.

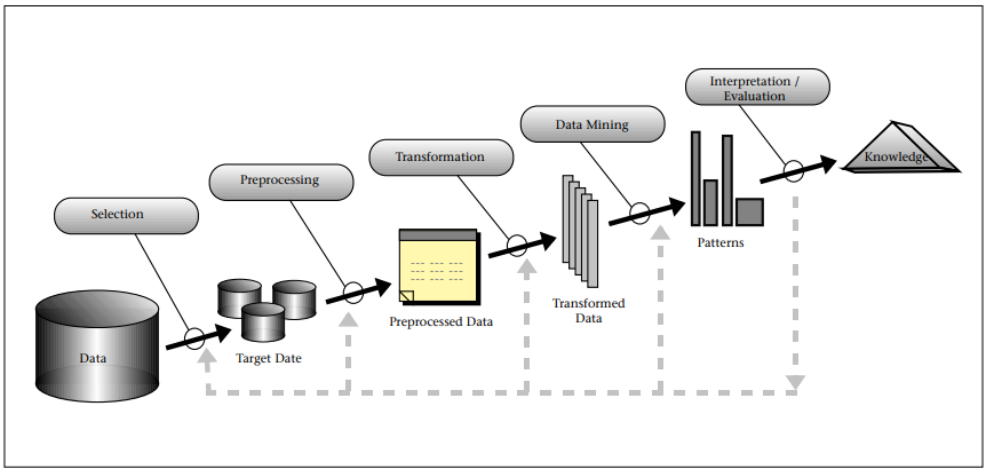

Figura 2.1: Panoramica degli step che compongono il processo di KDD [10]

L'integrazione del KDD in studi specifici rappresenta un avanzamento metodologico significativo. Tuttavia, è essenziale considerare potenziali limitazioni, tra cui problemi legati alla privacy, dipendenza dalla qualità dei dati in ingresso e costi associati alle infrastrutture e competenze richieste [12].

2.1 Data Exploration

La *Data Exploration* rappresenta una fase critica nel processo di analisi dei dati. Consiste nell'esaminare, organizzare, e visualizzare insiemi di dati per scoprire schemi, anomalie, relazioni, tendenze e ottenere intuizioni preliminari sui dati stessi [13]. Questo passaggio è essenziale prima di poter applicare algoritmi di machine learning, in quanto una comprensione adeguata dei dati può influenzare profondamente la scelta del modello e delle tecniche di analisi.

Può essere effettuato mediante tecniche differenti, più o meno matematicamente rigorose, tra cui:

- **Statistiche Descrittive**: questo approccio implica l'utilizzo di misure come media, mediana, moda, deviazione standard e quartili. Queste metriche offrono una panoramica generale della distribuzione e della centralità dei dati [14].
- **Visualizzazione dei dati**: grafici come istogrammi, box plots, scatter plots e heatmaps sono strumenti essenziali per visualizzare e comprendere la distribuzione, la correlazione e la struttura dei dati [15].
- **Analisi delle Componenti Principali (PCA)**: è una tecnica di riduzione della dimensionalità che permette di visualizzare dati ad alta dimensionalità in uno spazio bidimensionale o tridimensionale, conservando la maggior parte della varianza dei dati [16].
- **Identificazione di Anomalie**: tecniche come l'Isolation Forest o il One-Class SVM sono utilizzate per identificare outlier o anomalie nei dati, che potrebbero rappresentare errori o eventi rari [17].

La corretta esplorazione dei dati può evidenziare problemi come valori mancanti, outlier, e errori nei dati. La sua importanza non si limita solo alla fase preliminare di un'analisi, ma si estende anche durante e dopo l'applicazione di algoritmi di ML (Machine Learning), per interpretare e validare i risultati ottenuti [10].

L'esplorazione dei dati rappresenta, dunque, un pilastro fondamentale nel processo di analisi dei dati. Garantisce una comprensione profonda dei dati, permettendo agli analisti e ai data scientist di prendere decisioni informate riguardo alla scelta di tecniche e algoritmi da applicare.

2.2 Data Preprocessing

In molti contesti applicativi, i database, a causa della loro vastità e dell'origine eterogenea dei dati, sono affetti da incoerenze, rumore o dati mancanti. Una qualità scadente dei dati porta inevitabilmente all'acquisizione di informazioni non ottimali.

Pertanto, è cruciale implementare un'efficace fase di Preprocessing dei dati, in modo da ottimizzare sia la qualità dell'informazione che l'efficienza degli algoritmi di Data Mining. Studi hanno dimostrato che l'utilizzo di metodi di Preprocessing prima del Data Mining incrementa significativamente la pertinenza e la rapidità d'estrazione dei modelli [11]. L'intera procedura di Data Preprocessing può essere categorizzata in quattro fasi chiave, come illustrato nella figura 2.2:

- **Data Cleaning**: questa fase si occupa di gestire e correggere anomalie quali missing values, outlier e incoerenze. Il rumore nei dati, essendo una variazione casuale, può essere individuato attraverso metodi come il clustering. Esistono diverse metodologie per gestire i dati mancanti, che includono l'eliminazione della tupla, l'imputazione manuale, l'assegnazione di una costante globale, l'utilizzo del valore mediano o medio, o l'applicazione di metodi statistici come la regressione o gli alberi decisionali.

- **Data Integration**: in molte circostanze, è necessario combinare dati da diverse fonti. Durante questa integrazione, la ridondanza informativa rappresenta una sfida. Una variabile può essere etichettata come ridondante se è derivabile da un altro attributo o set di attributi. L'analisi di correlazione può rivelare tali ridondanze.

- **Data Reduction**: questa fase mira a comprimere l'ampiezza dei dataset conservando al contempo l'integrità informativa, facilitando così l'efficienza degli algoritmi di data mining. Questo processo, noto anche come selezione di dati e caratteristiche, può essere facilitato tramite l'analisi di correlazione. In situazioni con attributi quantitativi, metriche come il coefficiente di correlazione lineare di Pearson [18], diventano particolarmente rilevanti.

- **Data Transformation**: questa tappa implica la ristrutturazione dei dati in una forma più adatta al mining. Ciò può includere la generazione di nuovi attributi, aggregazione di dati, normalizzazione per garantire una scala uniforme e discretizzazione, dove i valori numerici sono sostituiti da intervalli. Tale approccio metodologico permette di massimizzare l'efficacia e l'efficienza dell'estrazione di informazioni dai grandi volumi di dati.

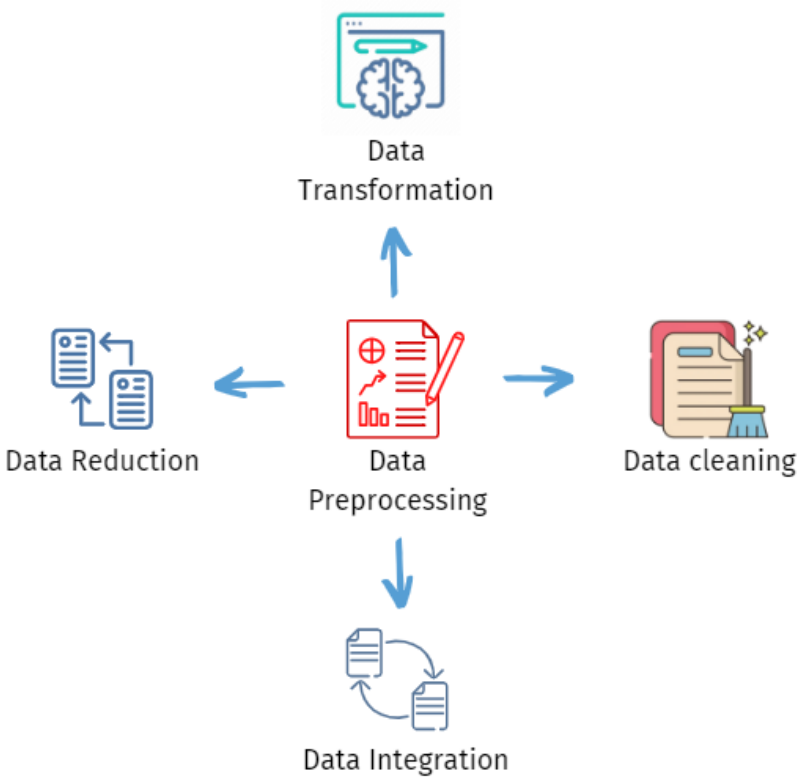

Figura 2.2: Preprocessing Steps

2.3 Data Mining

Successivamente alla fase di Preprocessing dei dati, si procede con l'attività di Data Mining, finalizzata all'identificazione di pattern significativi all'interno dei dataset. Questa fase ha come obiettivo principale quello di scoprire modelli e relazioni nascoste nei dati che possono essere utilizzati per prendere decisioni o previsioni informate [19]. Un pattern può essere descritto come una manifestazione ricorrente o anomala nei dati e funge da rappresentazione concisa e semanticamente ricca del dataset. Per essere considerato significativo, un pattern deve essere valido sui dati con un certo grado di confidenza, essere intuitivamente comprensibile da un punto di vista sintattico e semantico, ed essere inedito e avere potenziali applicazioni pratiche [20].

Il data mining è caratterizzato da due tipologie principali di modellazione:

- **Modellazione Descrittiva**: mira a identificare pattern comprensibili che rappresentano i dati, in modo da comprendere eventi passati, o presenti, legati ai dati stessi;

- **Modellazione Predittiva**: mira ad utilizzare un gruppo di variabili per predire i valori di una o più variabili, in modo da stimare eventi o risultati futuri [21].

A seconda delle specifiche esigenze, si possono applicare vari algoritmi e metodologie di Data Mining [11], [22], come sintetizzato nella tabella 2.1:

Metodo	**Tipologia**	**Descrizione**
Caratterizzazione e Discriminazione	Descrittiva	Analisi preliminare per esplorazione dei dati. Confronto di distribuzione dei valori degli attributi per record di una stessa classe (classificazione) e analisi dei valori di un attributo tra diverse classi (discriminazione).
Ricerca di Pattern Frequenti, Regole di Associazione	Descrittiva	Consiste nella rilevazione di pattern e identificazione di associazioni tra gruppi di record.
Classificazione	Predittiva	Consiste nella definizione di un modello matematico basato su dati storici per prevedere le classi di appartenenza di informazioni non ancora note.
Regressione	Predittiva	Consiste nella previsione dei valori di una variabile dipendente sulla base dei valori di altre variabili indipendenti (regressori).
Clustering	Descrittiva	Consiste nella segmentazione di un insieme eterogeneo in sottogruppi con caratteristiche analoghe.

Tabella 2.1: Algoritmi e metodologie di Data Mining

Ogni specifico problema, richiede, dunque, l'impiego di specifici modelli di ML. Nei problemi di predizione di valori di una particolare variabile, oppure nei problemi

in cui si ha la necessità di predire la classe di appartenenza di un determinato dato, gli algoritmi maggiormente pertinenti sono quelli di natura predittiva, e dunque, modelli di *Classificazione* e modelli di *Regressione*.
Nonostante la finalità ultima degli algoritmi di classificazione e di regressione possa essere simile, ovvero predire l'esito o il valore di una variabile, una differenza sostanziale tra essi è la tipologia di dati che sono in grado di ricevere in input. In particolare, gli algoritmi di *Regressione* sono particolarmente impiegati nel caso in cui ci siano variabili continue, mentre gli algoritmi di *Classificazione* sono particolarmente adatti per attributi categorici. Dunque, gli algoritmi di regressione sono impiegati quando si vuole calcolare il valore numerico di una particolare variabile dipendente, partendo da variabili indipendenti, che possono essere di natura differente a seconda della tipologia di algoritmo che viene utilizzato; gli algoritmi di classificazione, invece, sono utilizzati per assegnare una categoria (label) ad una particolare osservazione, e dunque consiste nella previsione della classe di appartenenza dell'osservazione stessa.
Le tabelle 2.2 e 2.3 riportano rispettivamente, in maniera sintetica, i principali modelli di Classificazione e Regressione con le loro caratteristiche chiave e la tipologia di dati con cui possono funzionare.

Algoritmo	**Caratteristiche**	**Tipologia di Dati**
Regressione Logistica	Modello lineare basato sulla funzione logistica	Numerici, categorici (codifica)
Alberi Decisionali	Divide l'insieme basandosi sul valore delle variabili	Numerici, categorici
Random Forest	Combinazione di alberi decisionali	Numerici, categorici
SVM	Iperpiano ottimale per dividere le classi	Numerici (standardizzati), categorici (codifica)
K-Nearest Neighbors	Classifica sulla base della classe maggioritaria dei k vicini	Numerici (normalizzati)
Reti Neurali	Ispirato alla struttura cerebrale, con neuroni artificiali	Numerici (normalizzati), categorici (codifica)
Gradient Boosting	Combinazione di alberi in sequenza per correggere errori degli alberi precedenti	Numerici, categorici

Tabella 2.2: Panoramica degli algoritmi di Classificazione

Algoritmo	**Caratteristiche**	**Tipologia di Dati**
Regressione Lineare	Relazione lineare tra variabili	Numerici, categorici (previa codifica)
Regressione Polinomiale	Estende la linearità includendo potenze delle variabili	Numerici
Regressione Ridge	Penalizzazione L2 dei coefficienti	Numerici, categorici (previa codifica)
Regressione Lasso	Penalizzazione L1 dei coefficienti	Numerici, categorici (previa codifica)
Regressione Elastic Net	Combinazione di L1 e L2	Numerici, categorici (previa codifica)
SVR (Support Vector Regression)	Iperpiano che meglio si adatta con margine	Numerici (meglio se standardizzati)
Alberi Decisionali per Regressione	Divide lo spazio delle variabili producendo un valore continuo	Numerici, categorici
Random Forest per Regressione	Combinazione di alberi decisionali, output come media delle stime	Numerici, categorici
Gradient Boosting per Regressione	Combinazione di alberi in sequenza per correggere errori	Numerici, categorici
Reti Neurali per Regressione	Adattamento delle reti neurali per produrre valori continui	Numerici (meglio se normalizzati), categorici (previa codifica)

Tabella 2.3: Panoramica degli algoritmi di Regressione

Nelle sezioni successive si propone una spiegazione dettagliata degli algoritmi impiegati nelle analisi del capitolo 4.

2.3.1 Linear Regressor

Il *Linear Regressor* è uno dei modelli di regressione maggiormente utilizzati, grazie alla sua semplicità di applicazione. Si basa sul concetto statistico di regressione lineare. Lo scopo principale del modello è predire il valore di una variabile dipendente continua sulla base di una o più variabili indipendenti [23]. Si basa sull'assunzione che esista una relazione lineare tra le variabili indipendenti e la variabile dipendente.

Sia Y la variabile dipendente e $X_1, X_2, ..., X_p$ le variabili indipendenti. Il modello di regressione lineare può essere espresso come:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \epsilon \tag{2.1}$$

dove:

- β_0 è l'intercetta;
- $\beta_1, \beta_2, ..., \beta_p$ sono i coefficienti delle variabili indipendenti;
- ϵ rappresenta l'errore casuale.

La validità della regressione lineare si basa su alcune ipotesi fondamentali [24], di seguito elencate:

1. **Linearità**: la relazione tra la variabile dipendente e le variabili indipendenti è lineare.
2. **Indipendenza**: gli errori (residui) sono indipendenti tra loro.
3. **Omotasticità**: la varianza degli errori è costante attraverso tutte le osservazioni.
4. **Normalità**: gli errori seguono una distribuzione normale.
5. **Assenza di collinearità**: le variabili indipendenti non sono altamente correlate tra loro.

La regressione lineare è intuitiva, interpretabile e computazionalmente efficiente. Tuttavia, è limitata dalla sua capacità di catturare solo relazioni lineari e dalla necessità che le sue ipotesi siano soddisfatte. In situazioni in cui esistono relazioni non lineari o le ipotesi non sono soddisfatte, modelli più complessi potrebbero essere necessari. In ogni caso, rappresenta un importante punto di partenza per analisi predittive, come si vedrà nel capitolo 4.5.1.

2.3.2 Random Forest

Il *Random Forest* è un algoritmo di apprendimento supervisionato sviluppato da Leo Breiman e Adele Cutler. È un metodo ensemble che combina le previsioni di diversi alberi decisionali per produrre un risultato finale più accurato e robusto rispetto a un singolo albero [25].

Il suo funzionamento si basa sui concetti di bagging (Bootstrap Aggregation) e di selezione casuale di features per la generazione di un insieme di alberi decisionali. In particolare, ad ogni iterazione, avvengono due principali azioni: da un lato viene selezionato un sottoinsieme randomico di dati, con sostituzione, tecnica di campionamento statistico del bootstrapping per l'appunto, per generare un albero decisionale, e dall'altro, nella creazione di ogni nodo dell'albero, viene effettuata una selezione casuale di features [25]. Queste due azioni, fanno sì che si generino alberi diversi tra loro, e che abbiano come base sottoinsiemi diversi di dati e features, andando notevolmente a ridurre il problema dell'overfitting, introducendo variabilità nei dati. Realizzato l'ensamble di alberi, a seconda del tipo di modello utilizzato, e dunque se si abbia a che fare con un problema di Regressione o di Classificazione, l'algoritmo procede all'aggregazione dei risultati in modo differente. In particolare, nel caso della Classificazione il Random Forest aggrega i voti forniti dai singoli alberi per prendere la decisione finale, mentre nel caso della Regressione, calcola la media delle previsioni [25].

Come ogni modello di ML, anche il Random Forest presenta dei pregi e dei difetti, elencati sinteticamente nella tabella 2.4.

Vantaggi	**Svantaggi**
Grazie all'utilizzo di molteplici alberi e alla randomizzazione nella selezione dei dati e delle caratteristiche, tende ad essere meno propenso all'overfitting rispetto ad un singolo albero di decisione.	A causa del numero di alberi utilizzati, l'addestramento e la previsione possono richiedere più tempo rispetto ad altri modelli.
Può gestire variabili categoriche senza la necessità di codifica e può gestire valori mancanti.	Sebbene offra una migliore accuratezza rispetto ad un singolo albero di decisione, può essere meno interpretabile.
Fornisce una stima dell'importanza di ogni caratteristica nella previsione della variabile di risposta.	

Tabella 2.4: Vantaggi e Svantaggi delle Random Forest

Dunque, il Random Forest è sicuramente una delle alternative più valide per effettuare previsioni di variabili continue e non, proprio grazie al fatto che può impiegare anche dati non elaborati, ed è resistente all'overfitting. Tuttavia, l'elevato tempo di elaborazione che generalmente caratterizza questo algoritmo può rappresentare un limite nel suo impiego.

Si illustra, di seguito, il funzionamento dei modelli **RandomForestRegressor** e **RandomForestClassifier** della libreria python Scikit-Learn.

Random Forest Regressor

Il *Random Forest Regressor* è uno stimatore che adatta una serie di alberi decisionali su differenti sotto-campioni del dataset, ed utilizza la media per aggregare i risultati dei singoli alberi, al fine di controllare l'overfitting e al fine di migliorare l'accuratezza predittiva [26].

É un modello che presenta numerosi iper-parametri, che possono essere utilizzati per migliorarne le performance, tra cui quelli principali, elencati in seguito:

- **n_estimators**: è un parametro di tipo *int*. Indica il numero di alberi da generare. Il valore di default è pari a *100* [26];
- **criterion**: è un parametro che può assumere i seguenti valori: *"squared_error", "absolute_error", "friedman_mse", "poisson"*. Indica la funzione da utilizzare per misurare la qualità dello split. Il valore di default è *"squared_error"* [26];
- **max_depth**: è un parametro di tipo *int*. Indica la massima profondità dell'albero. Il valore di default è *None* [26];
- **min_samples_split**: è un parametro di tipo *int* o *float*. Indica il numero minimo di campioni necessari per dividere un nodo interno. Il valore di default è *2* [26];
- **min_samples_leaf**: è un parametro di tipo *int* o *float*. Indica il numero minimo di campioni richiesto per essere in un nodo foglia. Il valore di default è *1* [26];
- **min_weight_fraction_leaf**: è un parametro di tipo *float*. Indica la frazione minima ponderata della somma dei pesi (di tutti i campioni in ingresso) necessaria per essere in un nodo foglia. Il valore di default è *0.0* [26];
- **max_features**: il numero di caratteristiche da considerare in fase di ricerca del miglior split. Può assumere i seguenti valori: *"sqrt", "log2", None*, oppure valori di tipo *int* o *float*. Il valore di default è *1.0* [26];

- **bootstrap**: è un parametro di tipo *boolean*. Indica se venga utilizzato o meno i campioni bootstrap nella costruzione degli alberi. Se il valore è impostato su Falso, viene utilizzato l'intero set di dati per costruire ogni albero. Il valore di default è *True* [26].

Random Forest Classifier

Il *Random Forest Classifier* è uno stimatore che adatta una serie di alberi decisionali su differenti sotto-campioni del dataset, ed utilizza l'aggregazione dei voti dei singoli alberi decisionali per ottenere il risultato finale, al fine di controllare l'overfitting e al fine di migliorare l'accuratezza predittiva [27].

Così come il RandomForestRegressor, anche il RandomForestClassifier è un modello che presenta numerosi iper-parametri, che possono essere utilizzati per migliorarne le performance, tra cui quelli principali, elencati in seguito:

- **n_estimators**: è un parametro di tipo *int*. Indica il numero di alberi da generare. Il valore di default è pari a *100* [27];
- **criterion**: è un parametro che può assumere i seguenti valori: *"gini", "entropy", "log_loss"*. Indica la funzione da utilizzare per misurare la qualità dello split. Il valore di default è *"gini"*. Tale parametro è specifico per ogni albero [27];
- **max_depth**: è un parametro di tipo *int*. Indica la massima profondità dell'albero. Il valore di default è *None* [27];
- **min_samples_split**: è un parametro di tipo *int* o *float*. Indica il numero minimo di campioni necessari per dividere un nodo interno. Il valore di default è *2* [27];
- **min_samples_leaf**: è un parametro di tipo *int* o *float*. Indica il numero minimo di campioni richiesto per essere in un nodo foglia. Il valore di default è *1* [27];
- **min_weight_fraction_leaf**: è un parametro di tipo *float*. Indica la frazione minima ponderata della somma dei pesi (di tutti i campioni in ingresso) necessaria per essere in un nodo foglia. Il valore di default è *0.0* [27];
- **max_features**: il numero di caratteristiche da considerare in fase di ricerca del miglior split. Può assumere i seguenti valori: *"sqrt", "log2", None*, oppure valori di tipo *int* o *float*. Il valore di default è *1.0* [27];
- **bootstrap**: è un parametro di tipo *boolean*. Indica se venga utilizzato o meno i campioni bootstrap nella costruzione degli alberi. Se il valore è impostato su

Falso, viene utilizzato l'intero set di dati per costruire ogni albero. Il valore di default è *True* [27].

Si può, dunque, notare come i modelli RandomForestRegressor e RandomForestClassifier siano fondamentalmente analoghi, ma utilizzati in contesti diversi: la regressione in presenza di variabili continue, e la classificazione in presenza di variabili discrete.

2.3.3 LightGBM

Per poter spiegare correttamente il funzionamento del LightGBM (Light Gradient Boosting Machine), è doveroso illustrare brevemente il funzionamento del Gradient Boosting, suo modello base.

Il *Gradient Boosting* è un algoritmo di apprendimento ensemble che combina le predizioni di diversi modelli più semplici per migliorare l'accuratezza e ridurre l'overfitting. In particolare, il Gradient Boosting costruisce un modello aggiuntivo che predice l'errore residuo del modello combinato, cercando di correggere le stime precedenti [28]. La costruzione del modello avviene sequenzialmente, rendendo il processo adattivo: ogni modello nell'ensemble cerca di correggere gli errori dei modelli precedenti.

L'idea di base dietro il Gradient Boosting è di applicare iterativamente l'algoritmo di boosting all'errore residuo calcolato rispetto alle predizioni correnti. L'algoritmo può essere riassunto nei seguenti passi:

1. Si inizia con una previsione iniziale, che potrebbe essere una costante o il risultato di un modello molto semplice.

2. Si calcolano gli errori residui tra le previsioni correnti e i valori reali.

3. Si costruisce un nuovo modello per prevedere questi errori residui.

4. Si aggiorna il modello combinato sommando le previsioni del nuovo modello alle previsioni correnti, moltiplicate per un fattore di apprendimento (noto come *learning rate*).

5. Si ripetono i passi dal 2 al 4 per un numero prestabilito di volte o fino a quando l'errore residuo non scende al di sotto di una soglia.

In figura 2.3 viene riportato uno schema visivo del meccanismo di funzionamento del Gradient Boosting.

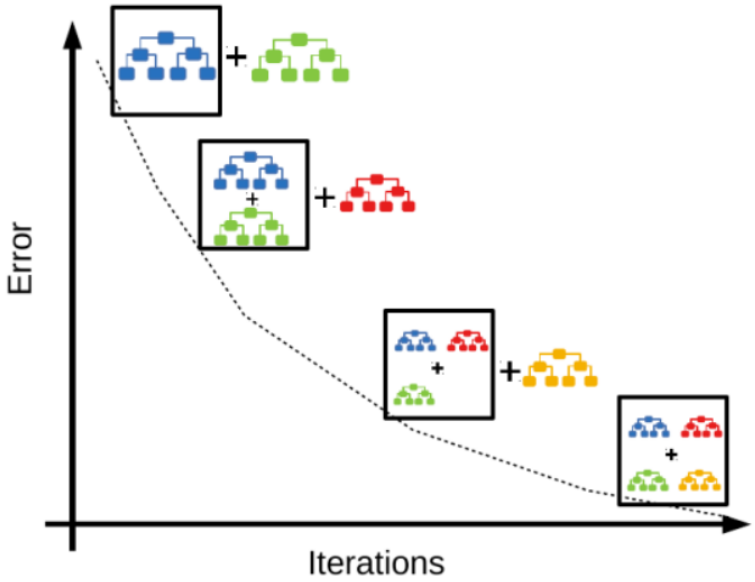

Figura 2.3: Schema esemplificativo del funzionamento del modello Gradient Boosting [29]

La chiave del Gradient Boosting è, dunque, nel modo in cui ottimizza l'errore. Utilizza la discesa del gradiente, un algoritmo di ottimizzazione, per minimizzare la funzione di perdita [30]. Pertanto, invece di costruire modelli sulle etichette reali, il Gradient Boosting costruisce modelli sugli errori rispetto alle previsioni correnti, direzionando l'ensemble verso una migliore performance a ogni iterazione.

Come ogni modello, anche il Gradient Boosting presenta dei vantaggi e degli svantaggi, sinteticamente elencati in tabella 2.5.

Vantaggi	**Svantaggi**
Può raggiungere una performance elevata in molte applicazioni.	Tende a essere più lento e richiede più risorse computazionali rispetto ad algoritmi più semplici.
Meno sensibile all'overfitting rispetto ad altri algoritmi quando si dispone di un grande volume di dati.	La performance può degradare se il numero di alberi (iterazioni) è troppo elevato e i dati sono pochi, portando a un overfitting.
Può gestire dati misti di variabili numeriche e categoriche.	Richiede una messa a punto accurata dei parametri per ottenere risultati ottimali.

Tabella 2.5: Vantaggi e svantaggi del Gradient Boosting

Detto ciò, il modello *LightGBM* è una particolare implementazione del Gradient Boosting, noto per le sue prestazioni e la sua efficienza [31]. Mentre il Gradient Boosting tradizionale costruisce un albero alla volta, il LightGBM utilizza una strategia di crescita basata su foglia, piuttosto che una strategia basata su profondità, offrendo vantaggi significativi in termini di velocità e prestazioni, come illustrato in figura 2.4.

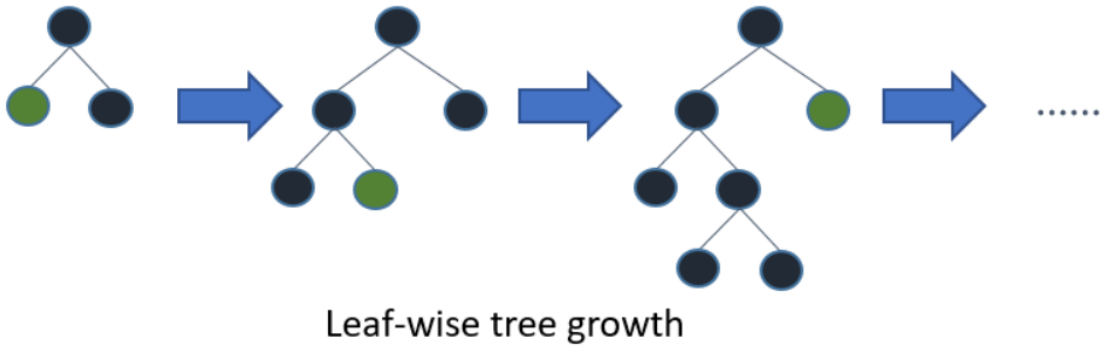

Figura 2.4: Schema esemplificativo del funzionamento del modello LightGBM [29]

Le principali caratteristiche che distinguono LightGBM da altre implementazioni di Gradient Boosting sono elencate di seguito:

- **Crescita basata su foglia**: a differenza delle tradizionali strategie di crescita basate sulla profondità, LightGBM ottimizza la crescita dell'albero preferendo la scissione di foglie con una perdita maggiore; ciò può ridurre la perdita maggiormente rispetto ad una crescita basata sulla profondità [31], e può generare alberi con foglie che non sono necessariamente bilanciate (si veda figura 2.4). Ciò può risultare in alberi più profondi, ma con una migliore precisione, a spese della capacità di generalizzazione.
- **Ottimizzazione per grandi dataset**: è particolarmente adatto a dataset di grandi dimensioni.
- **Supporto per l'apprendimento in categoria**: può gestire direttamente le variabili categoriali, senza la necessità di codifica one-hot.
- **Parallelismo e supporto GPU**: per aumentare ulteriormente l'efficienza, LightGBM supporta l'apprendimento parallelo e può essere eseguito su GPU.

Nel contesto della regressione, in particolare, LightGBM cerca di ottimizzare il MSE come funzione obiettivo predefinita, ma è anche flessibile e permette di

definire funzioni obiettivo personalizzate. Le sue caratteristiche, combinate con la capacità di gestire grandi quantità di dati, lo rendono uno strumento prezioso per applicazioni di regressione in diversi settori, come quello della predizione del reddito totale annuo, che si vedrà nel paragrafo 4.5.1, ma anche in problemi di classificazione, come si vedrà nella sezione 4.5.2.

2.3.4 Support Vector Machines

Il SVM (Support Vector Machine) è un modello di apprendimento supervisionato utilizzato per problemi di classificazione e regressione. SVM è particolarmente efficace in spazi ad alta dimensione o quando il numero di dimensioni supera il numero di campioni.

L'idea principale degli algoritmi SVM è quella di trovare un iperpiano che separi al massimo le diverse classi nei dati di addestramento, e dunque, che massimizzi il margine tra le classi di dati. Ciò avviene trovando l'iperpiano che ha il margine maggiore, definito come la distanza tra l'iperpiano e i punti dati più vicini di ciascuna classe. Una volta determinato, i nuovi dati possono essere classificati determinando da quale lato dell'iperpiano cadono [32]. Quanto detto, come si può intuire, è valido per problemi binari, oppure per dati linearmente separabili.

Nel caso di problemi multi-classe, o in presenza di dati non linearmente separabili, tuttavia, si possono utilizzare determinati accorgimenti in modo da rendere gli algoritmi SVM utilizzabili. In tal caso si parla di algoritmi *Kernelized SVM*. Essi, si basano su funzioni di kernel, le quali sono impiegate per calcolare la similarità tra due punti nel nuovo spazio delle caratteristiche trasformato in seguito all'applicazione della funzione, partendo da due punti di dati nello spazio delle caratteristiche originale [32].

L'applicazione della funzione di kernel è ciò che permette ai dati non linearmente separabili di essere trattati andandoli a mappare in uno spazio di dimensioni superiori.
Tra le funzioni di kernel maggiormente utilizzate, vi è il RBF (Radial Basis Function Kernel), il quale è la funzione di default per modelli SVM multi-classe. In esso la similarità tra due punti nello spazio delle caratteristiche trasformato è una funzione esponenzialmente decrescente della distanza tra i vettori e lo spazio di input originale [32], e si basa sulla seguente funzione matematica: $K(x, x') = e^{-\gamma||x-x'||^2}$, in cui il parametro γ è un parametro relativo alla larghezza del kernel, che determina l'estensione o la portata dell'influenza di un singolo campione di addestramento. É dunque, un parametro che va opportunamente selezionato in fase di tuning, in quanto un valore troppo elevato potrebbe condurre ad overfitting, mentre un valore troppo basso potrebbe condurre ad un modello eccessivamente semplice e non in grado di catturare la complessità dei dati. Un altro importante parametro del RBF kernel SVM è il parametro C, anche noto come *parametro di regolarizzazione*. Esso

controlla il rapporto tra l'ottimizzazione del margine e la riduzione del termine di errore di addestramento. Il valore di default è 10, e un valore troppo alto può generare overfitting [33].

In figura 2.5 è riportato il principio di funzionamento di diversi modelli di SVM Classifier.

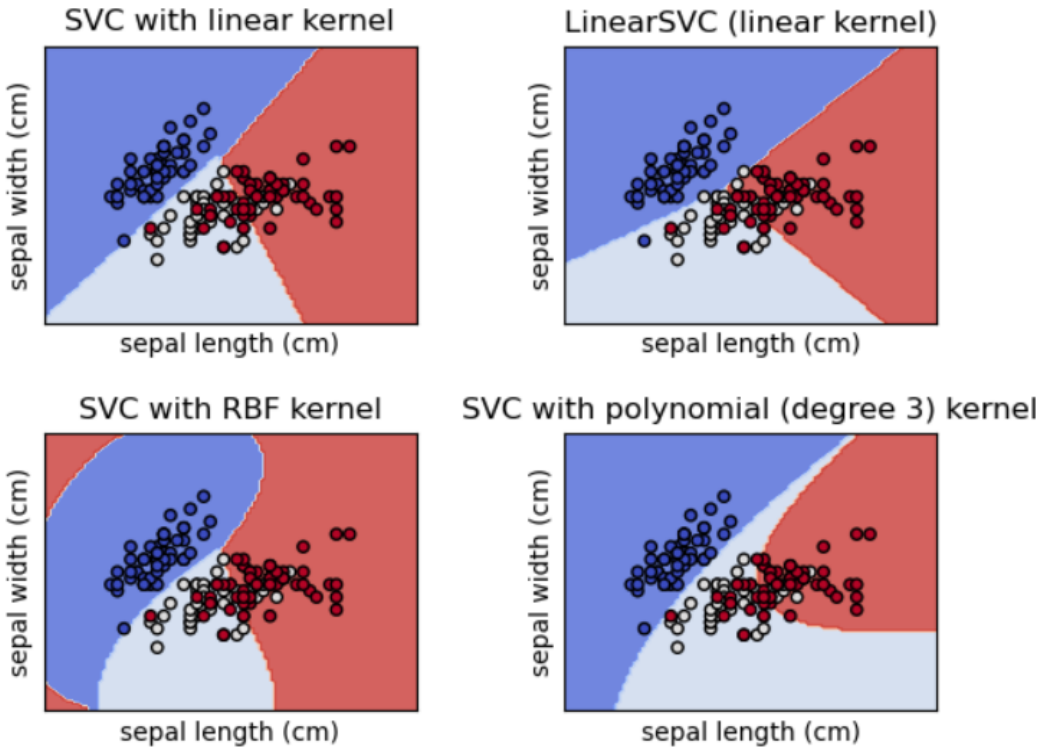

Figura 2.5: Principio di funzionamento di modelli di classificazione SVM lineari e non [34]

Come ogni modello, anche gli algoritmi SVM presentano vantaggi e svantaggi. In particolare, risultano particolarmente efficaci in spazi di dimensione elevata, sono versatili e discretamente resistenti all'overfitting. D'altro canto, però, non sono particolarmente adatti a set di dati molto grandi, rischiando di essere inefficienti in termini di tempo, a causa dell'elevata complessità computazionale che vi è alla base. Sono, infine, particolarmente sensibili alla scelta dei parametri.

2.4 Performance Evaluation

La valutazione delle prestazioni di un modello di apprendimento automatico è una fase critica nel processo di sviluppo di un sistema di Machine Learning. Un'accurata valutazione permette di comprendere l'efficacia del modello nel risolvere un determinato compito e di confrontarlo con altri modelli, consentendo anche di identificare aree di miglioramento [30]

Le metriche di valutazione delle performance variano a seconda della tipologia dei modelli impiegati. Vengono di seguito riportate le principali metriche per algoritmi di regressione, ed algoritmi di classificazione.

2.4.1 Performance Evaluation dei modelli di Regressione

La valutazione delle prestazioni di un modello di regressione è fondamentale per comprendere l'efficacia del modello nel rappresentare la relazione tra variabili indipendenti e dipendenti. La scelta delle metriche di valutazione dipende dalla natura del problema e dalla distribuzione dei dati [30].

Le metriche maggiormente utilizzate per valutare i modelli di regressione, sono:

- **MSE (Mean Squared Error (Errore Quadratico Medio))**: misura la differenza media tra i valori osservati e le previsioni del modello. È definito come:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

  dove y_i è il valore osservato e $\hat{y}_i$ è il valore predetto dal modello per l'i-esima osservazione.

  Il MSE fornisce una misura del grado di errore del modello in termini quadratici. Un valore pari a 0 indica una perfetta aderenza delle previsioni ai dati osservati [35]. Poiché esso eleva al quadrato le differenze, tende a penalizzare errori singoli maggiori, rendendolo particolarmente sensibile agli outliers.

- **RMSE (Root Mean Square Error (Radice dell'Errore Quadratico Medio))**: è la radice quadrata dell'MSE. Dà un'idea della deviazione standard degli errori del modello:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

  Rispetto al MSE, l'RMSE fornisce un'interpretazione più diretta, poiché è espresso nelle stesse unità della variabile target. Rappresenta la dispersione media degli errori del modello e, come per MSE, valori inferiori indicano una maggiore accuratezza del modello [35].

- R^2 **(Coefficiente di determinazione R^2)**: il coefficiente di determinazione, noto anche come R^2, misura la proporzione della varianza nella variabile dipendente che viene spiegata dal modello. Il suo valore varia tra 0 e 1, con 1 che indica una perfetta adattabilità dei dati al modello:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

  dove $\bar{y}$ è la media dei valori osservati.

  Il valore di R^2 oscilla tra 0 e 1, con valori vicini a 1 che indicano un'elevata capacità del modello di spiegare la variabilità nei dati. Tuttavia, un R^2 elevato non garantisce necessariamente che il modello sia appropriato; pertanto, deve essere utilizzato con cautela e in combinazione con altre metriche e analisi [35].

In aggiunta alle metriche di valutazione delle performance, bisogna considerare anche le tecniche di validazione dei modelli, le quali incidono sulle loro capacità di previsione. Le due tecniche principali sono:

- **Metodo holdout**: in esso, il dataset è diviso in due sottogruppi: un dataset di test ed uno di addestramento. Il dataset di training è utilizzato per addestrare il modello, mentre quello di test è impiegato per valutarne le prestazioni [36]. Sebbene l'approccio holdout sia computazionalmente efficiente e di facile implementazione, presenta una variazione intrinseca in base alla divisione casuale dei dati. Ciò implica che il modello possa portare a risultati non consistenti, specie se il set di test non rappresenta adeguatamente la distribuzione complessiva dei dati [37], e se è di ridotte dimensioni [36].

- **Cross-validation**: tale tencica, altrimeni nota come *k-fold cross validation*, affronta la limitazione dell'holdout mediante la segmentazione del dataset in k sottogruppi o "fold". Durante il processo di validazione, $k-1$ fold sono utilizzati per l'addestramento mentre il fold rimanente è utilizzato come set di validazione [36]. Questo procedimento è ripetuto k volte, assicurando che ogni fold sia utilizzato esattamente una volta come set di validazione. Le prestazioni del modello sono quindi aggregate, solitamente attraverso la media, dai risultati ottenuti da ogni singolo fold. Nonostante la sua maggiore robustezza nella stima delle performance rispetto all'holdout, la *Cross-Validation* richiede una maggiore capacità computazionale data la necessità di addestrare il modello k volte. Tuttavia, consente di avere una visione più completa delle prestazioni del modello su diversi subset di dati [37], anche se può fare difficoltà per dataset eccessivamente grandi [36].

2.4.2 Performance Evaluation dei modelli di Classificazione

Così come detto per i modelli di Regressione, anche per i modelli di Classificazione, la valutazione delle performance è uno step fondamentale, in quanto consente di avere un'indicazione chiara della capacità che essi hanno di generalizzare su dati non visti, aiutando così a scegliere il modello più appropriato per un'applicazione specifica.

Le metriche per la valutazione dei modelli di classificazione sono di seguito elencate:

- **Matrice di Confusione**: stabilisce una corrispondenza tra le previsioni del modello e i valori reali di classe, offrendo un quadro dettagliato delle previsioni corrette e di quelle errate realizzate dal modello. Le componenti principali di una matrice di confusione per una classificazione binaria sono:

  - *Veri Positivi (TP)*: rappresenta il numero di elementi correttamente identificati come positivi dal modello.
  - *Falsi Positivi (FP)*: rappresenta il numero di elementi errati classificati come positivi, ma che in realtà sono negativi.
  - *Veri Negativi (TN)*: rappresenta il numero di elementi correttamente identificati come negativi dal modello.
  - *Falsi Negativi (FN)*: rappresenta il numero di elementi erroneamente classificati come negativi dal modello, ma che in realtà sono positivi [36].

  Le principali metriche derivanti da questa matrice sono: Accuracy, Precisione, Recall, e F1-score [36], ivi spiegate.

- **Accuracy (Accuratezza)**: è una delle metriche più semplici ed intuitive, e per questo una delle più utilizzate. Fornisce una misura generale di quanto il modello in questione sia efficace nel classificare correttamente le istanze. Matematicamente è definita come il rapporto tra il numero di previsioni corrette ed il numero totale di previsioni effettuate [36]:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2.2}$$

  É importante, nel momento in cui si osserva il valore dell'Accuracy, tenere a mente una sua grande limitazione. Il suo valore, nel caso di classi sbilanciate, è fortemente influenzato dalla numerosità della classe maggioritaria, per cui si potrebbe ottenere un valore elevato di accuracy andando a predire unicamente la classe più frequente, ed ignorando completamente la o le classi minoritarie. Ciò conduce ad un significativo errore, in quanto tale metrica suggerisce che

il modello stia predicendo correttamente le varie istanze, ma in realtà sta ignorando completamente alcune classi. Occorre, dunque, osservare anche le altre metriche di valutazione esistenti per i modelli di Classificazione, come quelle di seguito elencate.

- **Precision (Precisione)**: è una metrica che quantifica la correttezza delle predizioni positive fatte da un modello di classificazione. In termini pratici, la precisione risponde alla domanda: "Di tutte le istanze che il modello ha classificato come positive, quante erano effettivamente positive?". Matematicamente, la precisione si definisce come rapporto tra le istanze correttamente classificate come positive e tutte le istanze classificate come positive [36], [38]:

$$\text{Precision} = \frac{\text{Veri Positivi (TP)}}{\text{Veri Positivi (TP)} + \text{Falsi Positivi (FP)}} \tag{2.3}$$

- **Recall (Richiamo) o Sensibilità**: quantifica la capacità del modello di identificare tutti i casi positivi reali all'interno del set di dati. Matematicamente, il richiamo si definisce come il rapporto tra le istanze correttamente classificate come positive e tutte le effettive istanze positive [36], [38]:

$$\text{Recall} = \frac{\text{Veri Positivi (TP)}}{\text{Veri Positivi (TP)} + \text{Falsi Negativi (FN)}} \tag{2.4}$$

- **F1-Score**: è una metrica che combina sia la precisione sia il richiamo in un unico punteggio. Esso rappresenta la media armonica tra precisione e richiamo. L'F1-Score si definisce come la media armonica tra precisione e recall e fornisce un singolo punteggio che bilancia le due metriche [36], [38]:

$$\text{F1-Score} = 2 \times \frac{\text{Precisione} \times \text{Richiamo}}{\text{Precisione} + \text{Richiamo}} \tag{2.5}$$

- **Curva ROC e AUC**:
  - la **ROC (Curva Caratteristica Operativa Ricevitore)** è un grafico che illustra la performance di un modello di classificazione binaria attraverso tutti i livelli di soglia di classificazione. L'asse delle ordinate (Y) mostra il Tasso di Veri Positivi (Sensibilità), mentre l'asse delle ascisse (X) rappresenta il Tasso di Falsi Positivi (1-Specificità). Un modello perfetto avrà una curva ROC che passerà per il punto (0,1), mentre un modello casuale avrà una curva ROC rappresentata da una linea diagonale dall'angolo inferiore sinistro all'angolo superiore destro [36];

– l'**AUC (Area Sotto la Curva)** rappresenta l'area sottesa dalla curva ROC. Questa metrica offre una misura aggregata delle performance del modello di classificazione su tutti i possibili livelli di soglia. Un AUC di 1 indica una classificazione perfetta, mentre un AUC di 0,5 suggerisce una performance equivalente al caso casuale [36].

Bisogna, infine, porre l'attenzione su un aspetto importante. Mentre ognuna delle metriche menzionate fornisce una visione preziosa sulla qualità di un modello di classificazione, è fondamentale considerare tutte le metriche in combinazione. Ad esempio, un modello potrebbe avere un'elevata precisione, ma un recall molto basso, indicando che potrebbe essere troppo conservativo nella classificazione delle istanze positive.

Capitolo 3

Stato dell'arte

Il bisogno crescente di attuare politiche di riduzione della povertà in maniera efficace ha portato alla nascita di una significativa quantità di ricerche focalizzate sulla predizione e quantificazione della povertà. Gli algoritmi di ML sono diventati, negli ultimi anni, strumenti fondamentali per analizzare e interpretare la vasta mole di dati a disposizione, al fine di trarre informazioni precise sulla distribuzione e le dinamiche della povertà in diverse regioni del mondo [7].

Nel corso di questo capitolo, si andranno ad esplorare le principali ricerche e metodologie adottate nel campo della *Poverty Prediction*, delineando gli sviluppi, le sfide e le opportunità di questo ambito in rapida evoluzione.

La trattazione di seguito proposta fa fede principalmente alle analisi del Review Paper [5], in cui sono raccolte le più recenti evidenze, metodologie ed applicazioni dei più innovativi studi effettuati sul topic della predizione della povertà, ma anche a singoli paper reputati particolarmente interessanti, tra cui [4], [6], [39] e [40].

Le prime ricerche in tal senso risalgono al 2016. In particolare, il focus principale di tali studi è stato quello di confrontare modelli di AI con i modelli econometrici principalmente in uso, al fine ci comprendere se, ed in che misura, l'Artificial Intelligence potesse essere impiegata in problemi di predizione della povertà [5]. Dopo aver dimostrato l'efficacia di tali modelli, le ricerche successive, nonché quelle più recenti in termini temporali, hanno avuto come obiettivo quello di confrontare tra loro l'efficacia di diversi modelli di Machine Learning [5].

Tra i motivi principali dell'importanza dei modelli di AI, e tra i loro punti di forza rispetto ai modelli econometrici vi è la capacità che essi hanno di gestire la multicollinearità, i livelli di accuratezza elevati, la maggiore velocità di calcolo e la grande capacità di gestire big data [5].

Gli aspetti appena elencati non sono, tuttavia, gli unici punti di forza dei modelli di Machine Learning. Essi, infatti, offrono la possibilità di effettuare, in aggiunta alla predizione della variabile di interesse, anche la fase di selezione delle features maggiormente rilevanti. É stato dimostrato come, variabili differenti possano avere

impatti diversi sulla predizione della povertà, e come la scelta delle variabili più influenti abbia un forte impatto sull'accuratezza delle previsioni dei modelli [5], [39].

In aggiunta a ciò, l'innovazione introdotta dall'impiego dei modelli di ML risiede anche nella tipologia di dati che essi sono in grado di gestire, consentendo di ampliare la lista dei fattori che possono essere sintomo, o causa, della condizione di povertà, ed offrendo l'opportunità di analizzarli. Se, infatti, i dati impiegati nei modelli econometrici erano di natura prevalentemente economica, e derivanti da survey nazionali ed internazionali e dalle analisi sul censimento della popolazione, con l'introduzione dei modelli di AI, e grazie alla loro capacità di estrazione delle variabili necessarie per le analisi, è stato possibile iniziare ad utilizzare dati di telerilevamento, i registri delle chiamate e i dati dell'e-commerce, tra i tanti [5].

In sintesi, i principali aspetti da evidenziare riguardo lo stato dell'arte degli studi sulla poverty prediction mediante algoritmi di AI sono relativi a 3 fattori:

1. la scelta di considerare la povertà come un problema multidimensionale o meno;
2. la tipologia dei dati impiegati;
3. i modelli utilizzati.

In particolare, tra i vari paper che hanno affrontato l'argomento, non tutti hanno utilizzato il medesimo approccio metodologico nel trattare il problema della povertà. Si denotato due principali categorie: una tipologia di analisi multidimensionale con un approccio non monetario, come ad esempio [6] ed una incentrata unicamente su un approccio monetario della povertà, come ad esempio [4]. É necessario fare chiarezza su cosa si intende per approccio monetario e non; per approccio monetario si intende considerare unicamente aspetti economici legati alla povertà, come ad esempio il reddito o le abitudini di consumo delle singole persone o delle famiglie, a seconda di chi sia il soggetto dell'analisi. Per approccio non monetario, invece, si intende una visione più ampia della povertà, in cui le voci economiche non sono reputate sufficienti per delineare un fenomeno così complesso, e pertanto si prendono in considerazione altri aspetti come il livello di istruzione, le condizioni di salute e l'accesso alla sanità dei soggetti coinvolti nell'analisi [5], tra i tanti.
Ciò detto, se i primi studi adottavano principalmente un approccio monetario, gli studi più recenti, anche in relazione alla maggiore importanza che i governi hanno iniziato ad attribuire agli aspetti non economici, adottano con maggior frequenza un approccio non monetario. In figura 3.1 sono riportati gli approcci selezionati per affrontare la povertà dai paper scritti sul tema negli anni dal 2016 al 2022, ed è visibile la tendenza crescente negli ultimi anni ad adottare principalmente un approccio non monetario. Gli acronimi presenti nella figura sono una forma

abbreviata dei titoli dei suddetti paper, di cui verranno elencati solo alcuni a titolo di esempio, come *'PCML': Poverty Classification Using Machine Learning: The Case of Jordan* [4] e *'MLB40': Machine Learning Approach for Bottom 40 Percent Households (B40) Poverty Classification* [39].

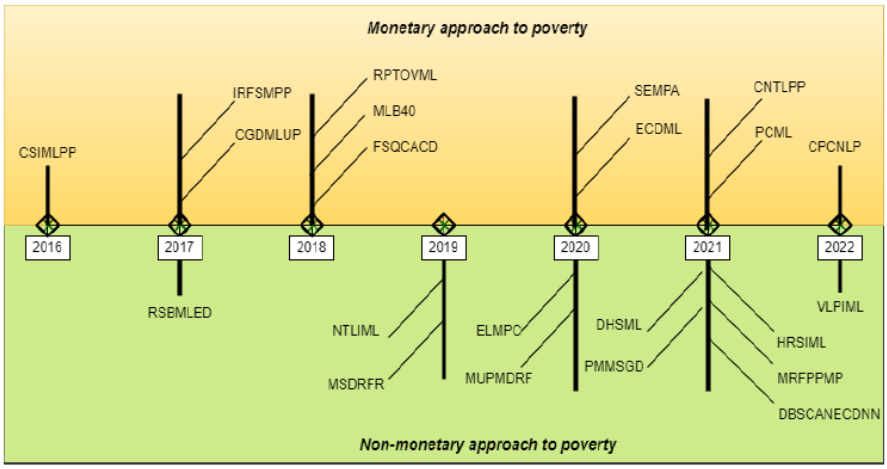

Figura 3.1: Tipologia di approccio al problema della povertà nei paper dal 2016 al 2022 [5]

Il secondo aspetto su cui porre l'attenzione, come accennato, è la tipologia di dati impiegati nelle ricerche effettuate sul tema. Anche in questo caso si possono dividere in due grandi categorie, i *Field Data* (Dati di campo), ed i *Remote Sensoring Data* (Dati di telerilevamento). La grande differenza tra queste due tipologie di dati è che i dati di campo sono raccolti generalmente mediante survey e report ufficiali governativi, e sono particolarmente onerosi da raccogliere, sia in termini temporali che economici, mentre i dati di telerilevamento sono raccolti tramite satellite, e sono generalmente immagini [5]. I dati di campo sono stati dunque divisi in 'Survey data', raccolti da organizzazioni internazionali e ricercatori indipendenti, e non possono essere utilizzati per identificare la condizione di un paese in via ufficiale, e 'Governamental data', ovvero dati governativi raccolti dai governi ed utilizzati come dati ufficiali stanti a riportare le condizioni del Paese in esame; i dati di telerilevamento, invece, sono divisi in 'Night-time data' e 'Day-time data', ovvero immagini satellitari notturne e diurne [5]. In figura 3.2 è riportata una suddivisione dettagliata dei dati impiegati.

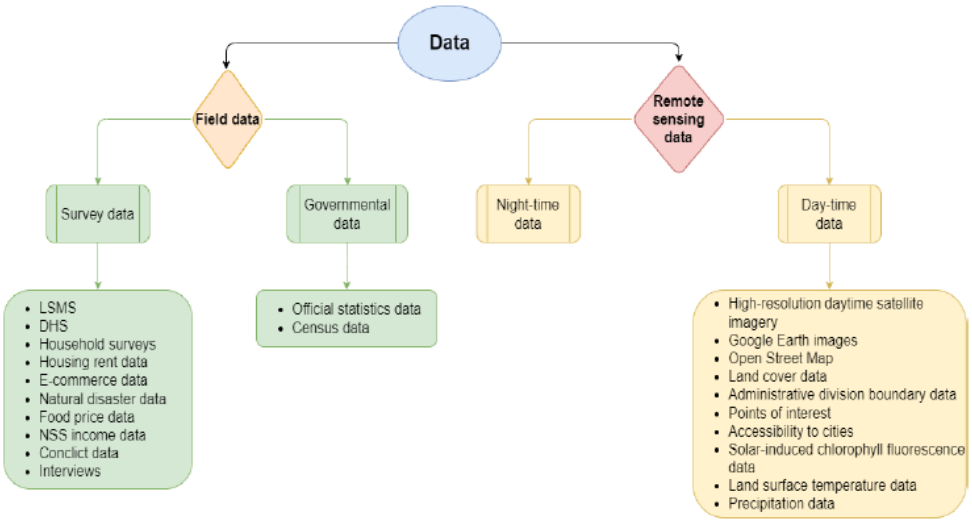

Figura 3.2: Tipologia di dati impiegati nei paper dal 2016 al 2022 [5]

In figura 3.3, invece, sono riportate le frequenze di utilizzo dei vari dati presentati. Si può notare come oltre il 35% dei paper aventi come tema la poverty prediction abbia utilizzato dati locali ufficiali per effettuare le proprie analisi, e che il 22% di esse abbia impiegato dati provenienti da indagini sulle famiglie, e dati provenienti da immagini satellitari notturne [5]. É interessante sottolineare come, nonostante i dati satellitari siano di recente impiego, hanno in realtà avuto sin da subito una grande diffusione, dimostrandosi particolarmente utili.

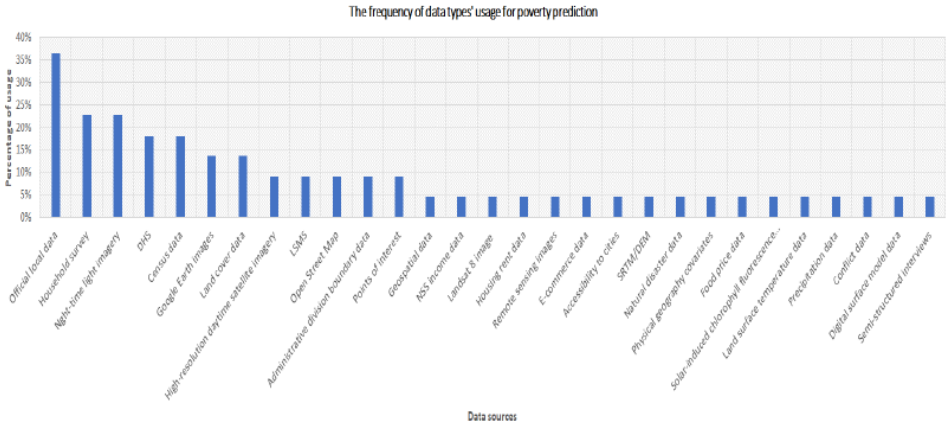

Figura 3.3: Frequenza dei dati impiegati nei paper dal 2016 al 2022 [5]

Il terzo aspetto da analizzare, nonché forse il più interessante ai fine del seguente lavoro di tesi, è quello relativo ai modelli di ML principalmente impiegati

nell'affrontare la tematica della poverty prediction.

Nei paper prodotti fino a metà del 2022 riguardo l'impiego dell'Intelligenza Artificiale al fine di predire lo stato di povertà, sono stati testati ed utilizzati numerosi modelli, quasi 60, a dimostrare il crescente interesse nell'introdurre fortemente l'AI nella lotta alla povertà. I modelli a cui si è fatto ricorso nei vari studi sono fondamentalmente appartenenti a due categorie: modelli di Machine Learning e modelli di Deep Learning.

Per quanto riguarda i primi, si fa riferimento a modelli di apprendimento automatico basati su algoritmi generalmente più semplici e meno stratificati rispetto ai secondi. Sono modelli allenati per apprendere automaticamente e non necessitano di essere riprogrammati. Per poterli far funzionare correttamente, però, è necessario che vi sia una profonda comprensione dei dati in esame per potere creare e modellare correttamente le caratteristiche che si vogliono analizzare. I modelli di Deep Learning, in maniera esemplificativa, invece, sono modelli che hanno come obiettivo quello di simulare il funzionamento del cervello umano. Sono pertanto, basati su algoritmi particolarmente complessi, e sono fortemente stratificati. Hanno bisogno di enormi quantità di dati e richiedono elevate capacità computazionali per poter funzionare. Tra i più noti modelli di Machine Learning, ci sono sicuramente gli Alberi Decisionali, il Random Forest ed il Support Vector Machine, il cui funzionamento è stato illustrato nel capitolo 2, mentre tra i principali modelli di Deep Learning troviamo le Reti Neurali Convoluzionali e Ricorrenti, particolarmente impiegate nel riconoscimento delle immagini ed in problemi di classificazione, motivo per cui hanno trovato impiego in diversi degli studi effettuati sulla poverty prediction. Questi ultimi due, a causa della loro complessità, non sono stati approfonditi ulteriormente nel presente lavoro di tesi.

In figura 3.4 sono riportati i 10 modelli di AI maggiormente utilizzati nei paper considerati.

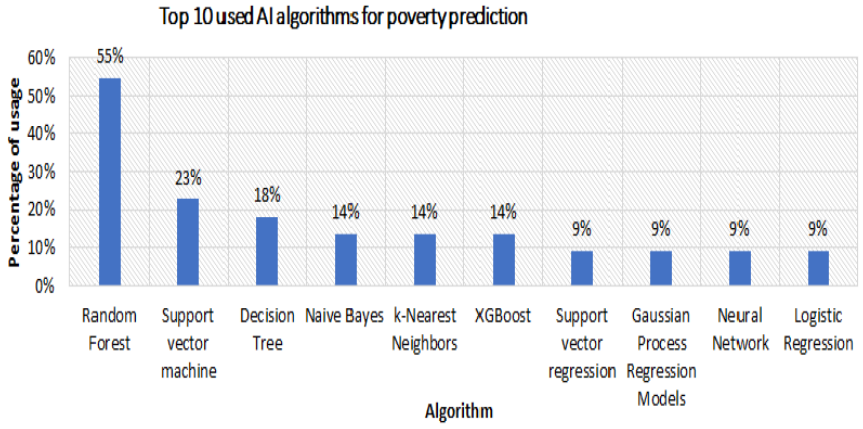

Figura 3.4: Top 10 modelli utilizzati nei paper dal 2016 al 2022 [5]

Tra i tanti, il più usato in letteratura, con una netta predominanza rispetto agli altri, con una percentuale del 55% è il Random Forest. Numerose sono le ragioni per cui esso è così largamente impiegato; infatti, è un modello particolarmente robusto e resistente a problematiche quali l'overfitting e la presenza di outlier, è in grado di gestire la multicollinearità, come illustrato nel paragrafo 2.3.2, ed è di facile applicazione in quanto è un modello fortemente automatizzato [5]. A seguire, con una percentuale di utilizzo del 23% vi sono i modelli SVM (Support Vector Machine), anch'essi descritti nel paragrafo 2.3.4, seguiti dagli Alberi Decisionali. Dalla figura 3.4, si può notare come la quasi totalità dei principali modelli utilizzati siano di Machine Learning, piuttosto che di Deep Learning, proprio per la maggiore semplicità di applicazione che hanno i primi rispetto ai secondi.

Quanto finora riportato, è dunque, una sintesi dei più interessanti lavori prodotti sul tema della poverty prediction, con particolare attenzione non tanto sulle modalità con cui si sono svolte queste analisi, in quanto ognuna di esse è basata su una strategia che fosse coerente con i dati a disposizione e personalizzata sul contesto di riferimento, quanto sugli strumenti utilizzati per realizzarle e sui fattori chiave che hanno avuto una forte influenza sulla loro riuscita.

É sulla base di una profonda analisi dei paper qui sinteticamente presentati e delle implicazioni che ognuno di essi ha avuto sul tema della poverty prediction, con l' aggiunta di un attento studio teorico del processo di KDD e dei principali modelli di ML, riportati nel capitolo 2, che trae fondamento il cuore del presente lavoro di tesi, lo studio di ricerca dettagliatamente affrontato nel successivo capitolo, il numero 4, mirato alla predizione dello stato di povertà delle famiglie filippine.

Capitolo 4

Caso studio: poverty prediction nel contesto socioeconomico delle Filippine

Il presente capitolo ha come obiettivo quello di illustrare l'analisi condotta su un Dataset denominato *Filipino Family Income and Expenditure* [41], mirata allo sviluppo di modelli predittivi in grado di identificare le famiglie filippine a rischio povertà.

In particolare, lo studio ha due focus principali:

1. Identificare quale algoritmo di ML sia maggiormente performante nel predire il reddito delle famiglie filippine, partendo dalle abitudini di consumo delle stesse ed adottando un approccio monetario della povertà;

2. Predire la fascia di reddito delle famiglie filippine, identificate prendendo come punto di riferimento la soglia di povertà del Paese, in modo da valutare le performance di diversi modelli di Classificazione, considerando sia indicatori di natura socio-economica, che demografica, adottando un approccio non monetario.

Per poter comprendere appieno il significato della suddetta analisi, è, tuttavia, necessario, delineare brevemente il contesto socio-economico e politico in cui versa il Paese.

4.1 Contesto socio-economico delle Filippine

Le Filippine, ufficialmente Repubblica delle Filippine, sono uno Stato insulare del Sud-est asiatico situato nell'oceano Pacifico; a nord è separato da Taiwan dallo stretto di Luzon, a ovest è bagnato dal mar Cinese Meridionale, a sud-ovest dal mare di Sulu verso il Borneo, a sud dal mare di Celebes che lo separa dalle altre isole dell'Indonesia e a est dal mare delle Filippine [42]. La posizione nei pressi della cintura di fuoco del Pacifico e il clima tropicale fanno delle Filippine un'area frequentemente colpita da terremoti e tifoni, di cui alcuni anche molto violenti [42].

L'arcipelago comprende 7641 isole distribuite in tre regioni geografiche principali: Luzon a nord, Visayas nel centro e Mindanao a sud. Dal 1976 la capitale è ufficialmente Manila, nella cui area metropolitana di Metro Manila risiede il governo. Con una popolazione stimata di quasi 111 milioni di persone, le Filippine sono il 13º Paese più popoloso del mondo e altri undici milioni di filippini vivono all'estero [42]. Varie etnie e culture convivono sulle isole delle Filippine, che sono considerate come una nazione di recente industrializzazione; fin dalla loro indipendenza l'economia delle Filippine è stata in continua crescita [42]. All'inizio del XXI secolo furono avviate riforme economiche che hanno portato il settore terziario a superare l'agricoltura come principale attività economica e attualmente i servizi incidono per oltre la metà del PIL del Paese [42]. Tuttavia le Filippine devono ancora affrontare molte sfide nel settore delle infrastrutture, mancando inoltre un adeguato sviluppo del settore del turismo, dell'istruzione, dell'assistenza sanitaria e dello sviluppo umano [42].

Come accennato il territorio è diviso in tre grandi gruppi insulari: Luzon, Visayas e Mindanao. In questi tre gruppi insulari sono situate le 17 regioni che compongono il Paese, di seguito elencate: Ilocos (Regione I), Valle di Cagayan (Regione II), Luzon Centrale (Regione III), Calabarzon (Regione IV-A), Mimaro (Regione IV-B), Bicol (Regione V), Visayas Occidentale (Regione VI), Visayas Centrale (Regione VII), Visayas Orientale (Regione VIII), Penisola di Zamboanga (Regione IX), Mindanao Settentrionale (Regione X), Davao (Regione XI), Soccsksargen (Regione XII), Caraga (Regione XIII), Regione Autonoma nel Mindanao Musulmano (ARMM), Cordillera (CAR), Regione Capitale Nazionale (NCR) (Metro Manila).

Le regioni non hanno un vero e proprio organismo governativo, ma sono al servizio delle province che hanno un proprio governo [42]. L'unica regione con un proprio governo è la Regione Autonoma nel Mindanao Musulmano, la quale, dal 2019 è stata soppressa e sostituita dalla Bangsamoro Autonomous Region in Muslim Mindanao, la quale ha comunque un governo autonomo pur restando parte del Paese.

La Repubblica delle Filippine è una repubblica di tipo presidenziale, con il Presidente che ricopre contemporaneamente le cariche di Capo dello Stato e Capo del Governo [42].

L'economia delle Filippine è la 42ª più grande al mondo, con un prodotto interno lordo calcolato nel 2012 pari a 250.182 milioni di dollari, valore nominale e a 419.572 milioni di dollari a parità di potere di acquisto, con un PIL pro-capite nominale di 2612 dollari e a parità di potere d'acquisto di 4380 dollari [42]. Le esportazioni primarie includono semiconduttori e prodotti elettronici, mezzi di trasporto, abbigliamento, prodotti in rame, prodotti petroliferi, olio di cocco e frutti. I principali partner commerciali sono Stati Uniti, Giappone, Cina, Singapore, Corea del Sud, Paesi Bassi, Hong Kong, Germania, Taiwan e Thailandia [42].

La valuta delle Filippine è il PHP (Peso Filippino).

Grazie alla posizione sul mare, ogni isola delle Filippine ha un forte sviluppo nel campo della pesca. I lavori legati alla natura (agricoltura, allevamento, pesca) sono molto presenti in tutte le isole; ciononostante, il Paese sta attraversando una trasformazione da un'economia basata sull'agricoltura ad un'economia basata maggiormente sui servizi; esso produce, infatti, il 57% del PIL, contro il 31% dell'industria e il 12% dell'agricoltura, nonostante una forza lavoro, in quest'ultima, pari al 32% della popolazione [42].

4.2 Caratterizzazione del Dataset

Il Dataset impiegato nell'analisi, denominato *Filipino Family Income and Expenditure* [41], è un dataset strutturato caratterizzato da una collezione di oltre 40mila records e 60 colonne, contenenti importanti informazioni in merito alle abitudini di consumo, agli introiti e ad altre variabili demografiche significative delle famiglie filippine. I dati impiegati sono il risultato dell'unione dei dati provenienti dalla "Family Income and Expenditure Survey" del 2012 e del 2015, condotta dal PSA (Philippine Statistics Authority) a cadenza triennale (dal 2020 in poi, a causa dei repentini mutamenti socioeconomici del paese, si è deciso di passare ad una analisi biennale).

Gli attributi, di tipo numerico e nominale, presenti nel Dataset, possono essere ricondotti a sei macrocategorie di riferimento:

1. **Reddito**: 4 attributi relativi al reddito familiare;
2. **Spese e consumi**: 19 attributi contenenti informazioni riguardo le spese delle famiglie: alimentari, mediche, per l'abbigliamento, per l'istruzione etc.
3. **Famiglia**: 6 attributi contenenti informazioni in merito alla tipologia di famiglia, alla numerosità, alla tipologia di occupazione dei membri, alla regione di residenza etc.
4. **Capofamiglia**: 7 attributi contenenti informazioni riguardo il sesso del capofamiglia, l'età, l'occupazione, lo stato civile etc.

5. **Abitazione**: 10 attributi contenenti informazioni in merito alla tipologia di abitazione, alle dimensioni, alla struttura etc.

6. **Beni posseduti**: 14 attributi contenenti informazioni riguardanti il numero di beni posseduti, la tipologia etc.

Gli attributi numerici legati alle spese ed agli introiti delle famiglie, sono espressi in *peso filippino*, ₱, e sono misure annuali.
In particolare, gli attributi oggetto dell'analisi, in ordine alfabetico e divisi per categoria, sono:

- **Reddito**:

    1. *Agricultural Household indicator*: attributo nominale. Indica se la famiglia sia o meno di natura agricola. Alcuni records riportano come valore di tale attributo "2"; tuttavia, essendo tale attributo di natura booleana, si può supporre che tali records siano da considerare outliers, e vanno opportunamente trattati;
    2. *Main Source of Income*: attributo nominale che indica la principale fonte di reddito della famiglia. Può assumere 3 distini valori ("Wage/Salaries", "Enterpreneurial Activities", "Other sources of Income");
    3. *Total Household Income*: reddito familiare complessivo;
    4. *Total Income from Entrepreneurial Acitivites*: reddito familiare proveniente da attività imprenditoriali.

- **Spese e consumi**:

    1. *Alcoholic Beverages Expenditure*: misura numerica contenente l'indicazione della spesa delle famiglie in bevande alcoliche;
    2. *Bread and Cereal Expenditure*: spesa in pane e cereali;
    3. *Clothing, Footwear and Other Wear Expenditure*: spesa in abbigliamento, calzature ed altri indumenti;
    4. *Communication Expenditure*: spesa per le comunicazioni (telefonia, internet, etc.);
    5. *Crop Farming and Gardening expenses*: spesa per l'agricoltura ed il giardinaggio;
    6. *Education Expenditure*: spesa per l'istruzione;
    7. *Fruit Expenditure*: spesa per frutta;
    8. *Housing and water Expenditure*: spesa per la casa e per l'acqua (corrente elettrica, tasse, mantenimento della casa, etc.);

9. *Meat Expenditure*: spesa in carne;
10. *Medical Care Expenditure*: spesa per l'assistenza medica;
11. *Miscellaneous Goods and Services Expenditure*: spesa per beni e servizi vari;
12. *Restaurant and hotels Expenditure*: spesa per ristoranti e alberghi;
13. *Special Occasions Expenditure*: spese per le occasioni speciali;
14. *Tobacco Expenditure*: spesa in tabacco ed articoli da fumo;
15. *Total Fish and marine products Expenditure*: spesa totale per Pesce e Prodotti Ittici;
16. *Total Food Expenditure*: spesa alimentare complessiva;
17. *Total Rice Expenditure*: spesa complessiva per il riso;
18. *Transportation Expenditure*: spesa per i trasporti;
19. *Vegetables Expenditure*: spesa per gli ortaggi;

- **Famiglia**:

  1. *Members with age less than 5 year old*: numero di membri della famiglia con età inferiore ai cinque anni;
  2. *Members with age 5 - 17 years old*: numero di membri della famiglia con età compresa tra i cinque ed i diciassette anni;
  3. *Region*: regione di residenza. Attributo nominale che può assumere diciassette distinti valori ("I-Ilocos Region", "II-Cagayan Valley", "III-Central Luzon", "IVA-CALABARZON", "IVB-MIMAROPA", "V-Bicol Region", "VI-Western Visayas", "VII-Central Visayas", "VIII-Eastern Visayas", "IX-Zasmboanga Peninsula", "X-Northern Mindanao", "XI-Davao Region", "XII-SOCCSKSARGEN", "Caraga", "ARMM", "CAR", "NCR"), corrispondenti alle diciassette regioni di cui si compongono le Filippine, illustrate nel paragrafo 4.1;
  4. *Total number of family members*: numero totale di componenti della famiglia;
  5. *Total number of family members employed*: numero totale di membri della famiglia occupati;
  6. *Type of Household*: tipo di famiglia. Attributo nominale che può assumere tre distinti valori ("Single Family", "Extendend Family" e "Two or More Nonrelated Persons/Members")

- **Capofamiglia**:

1. *Household Head Age*: attributo nominale indicante l'età del capofamiglia;
2. *Household Head Class of Worker*: attributo nominale indicante la classe di lavoro del capofamgglia. Può assumere otto distinti valori ("Employer in own family-operated farm or business", "Self-employed wihout any employee", "Worked without pay in own family-operated farm or business", "Worked for government/government corporation", "Worked for private establishment", "Worked for private household", "NA";
3. *Household Head Highest Grade Completed*: attributo nominale indicante il più alto grado di istruzione conseguito dal capofamiglia. Può assumere ventitre distinti valori, i quali per semplictà non verranno elencati. Tuttavia, possono essere riassunti in cinque categorie di livello d'istruzione:
   (a) Istruzione Primaria;
   (b) Istruzione Secondaria;
   (c) Istruzione Superiore;
   (d) Laurea di Primo Livello;
   (e) Laurea di Secondo Livello/Corso di Formazione Professionale;
4. *Household Head Job or Business Indicator*: attributo nominale. Indica se il capofamiglia sia impiegato/abbia un proprio business, o meno. Può assumere due valori ("With Job/Business" o "No Job/Business");
5. *Household Head Marital Status*: attributo nominale che indica lo stato civile del capofamiglia. Può assumere sei distinti valori ("Single", "Married", "Divorced/Separated", "Widowed", "Annulled", "Unknown");
6. *Household Head Occupation*: attributo nominale indicate l'occupazione del capofamiglia. Può assumere trentotto distini valori, i quali per semplicità, non verranno elencati;
7. *Household Head Sex*: attributo nominale che indica il sesso del capofamiglia. Può assumere due valori ("Female", "Male").

- **Abitazione**:

1. *Electricity*: attributo booleano indicante se l'abitazione sia o meno fornita di elettricità. Assume valore pari ad 1 in caso di esito positivo, 0 altrimenti;
2. *House Age*: indica l'età dell'abitazione. Il dato è esrpesso in anni;
3. *House Floor Area*: indicazione numerica della superficie dell'abitazione. Il dato è espresso in m^2;
4. *Imputed House Rental Value*: attributo contenente il valore economico imputato all'affitto dell'abitazione;

5. *Main Source of Water Supply*: attributo nominale indicante la fonte principale di approviggionamento idrico. Può assumere undici valori distinti ("Dug well", "Lake, river, rain and others", "Own use, faucet, community water system", "Own use, tubed/piped deep well", "Peddler", "Protected spring, river, stream, etc", "Shared, faucet, community water system", "Shared, tubed/piped deep well", "Tubed/piped shallow well", "Unprotected spring, river, stream, etc", "Others");
6. *Tenure Status*: attributo nominale indicante lo stato di possesso dell'abitazione. Può assumere otto valori distinti ("Own or owner-like possession of house and lot", "Rent-free house and lot without consent of owner", "Rent-free house and lot with consent of owner" "Rent house/room including lot", "Own house, rent-free lot without consent of owner", "Own house, rent-free lot with consent of owner", "Own house, rent lot", "Not Applicable",);
7. *Toilet Facilities*: attributo nominale contenente l'indicazione della tipologia dei servizi igienici presenti nell'abitazione. Può assumere otto distinti valori ("Water-sealed, sewer septic tank, used exclusively by household", "Water-sealed, sewer septic tank, shared with other household", "Water-sealed, other depository, used exclusively by household", "Water-sealed, other depository, shared with other household", "Open pit", "Closed pit", "Others", "None");
8. *Type of Building/House*: attributo nominale stante ad indicare la tipologia di abitazione. può assumere sei distinti valori ("Single house", "Multi-unit residential", "Institutional living quarter", "Duplex", "Commercial/industrial/agricultural building", "Other building unit (e.g. cave, boat)");
9. *Type of Roof*: attributo nominale che indica la tipologia di tetto dell'abitazione, in termini di solidità strutturale. Può assumere sei valori distinti ("Strong", "Quite Strong", "Light", "Very Light", "Salvaged", "Not Applicable");
10. *Type of Walls*: attributo nominale che indica la tipologia di pareti dell'abitazione, in termini di solidità strutturale. Può assumere sei valori distinti ("Strong material (galvanized,iron,al,tile,concrete,brick,stone,asbestos)", "Mixed but predominantly strong materials", "Mixed but predominantly salvaged materials", "Mixed but predominantly light materials", "Light material (cogon,nipa,anahaw)", "Salvaged/makeshift materials", "Not Applicable").

- **Beni posseduti**:

1. *Number of Airconditioner*: numero di condizionatori d'aria;

2. *Number of bedrooms*: numero di camere da letto;
3. *Number of Car, Jeep, Van*: numero di auto, Jeep, Van;
4. *Number of CD/VCD/DVD*: numero di CD/VCD/DVD;
5. *Number of Cellular phone*: numero di telefoni cellulare;
6. *Number of Component/Stereo set*: numero di componenti/stereo set;
7. *Number of Landline/wireless telephones*: Numero di telefoni fissi/wireless;
8. *Number of Motorcycle/Tricycle*: Numero di moto/tricicli;
9. *Number of Motorized Banca*: attributo dal significato dubbio. Potrebbe trattarsi di un errore grammaticale;
10. *Number of Personal Computer*: numero di PC;
11. *Number of Refrigerator/Freezer*: numero di refrigeratori/frigoriferi;
12. *Number of Stove with Oven/Gas Range*: numero di cucine con forno/ fornelli a gas;
13. *Number of Television*: numero di televisioni;
14. *Number of Washing Machine*: numero di lavatrici.

La lista di attributi precedentemente elencata, risulta essere particolarmente dettagliata, e consente di effettuare un gran numero di analisi circa lo stato economico delle famiglie filippine.
Tuttavia, prima di poter entrare nel vivo dell'analisi stessa, è necessario comprendere meglio quali siano le variabili di maggiore interesse per gli obiettivi preposti. In particolare, si provvede, nel seguente paragrafo, ad effettuare una profonda esplorazione dei dati in esame.

4.3 Data Exploration

Prima di entrare nel vivo della trattazione, è doveroso fare una precisazione. Il Dataset impiegato nell'analisi [41], essendo pubblicato sulla piattaforma di Data Science Kaggle, è di natura pubblica. Pertanto, diversi studi ed approfondimenti sono stati effettuati su di esso, specie quelli relativi ad analisi esplorative sulla sua composizione e sulle sue caratteristiche [43], [44], nonché analisi più approfondite, come analisi predittive [45]. Risulta dunque, alta la probabilità, in questa fase iniziale, di presentare risultati che potrebbero essere già stati ottenuti da ricerche precedenti, specialmente per quanto concerne l'esplorazione del dataset. Tuttavia, a scanso di equivoci, ogni risultato illustrato da qui in avanti, è frutto di un lavoro proprio, scevro da condizionamenti di lavori precedenti.

La fase di Data Exploration, fondamentale per il sussegursi dello studio, è stata effettuata mediante l'uso di due tool: la versione accademica del software

RapidMiner [46], una piattaforma open source dotata di un'interfaccia drag-and-drop, all'occorrenza programmabile, che consente di personalizzare i propri casi d'uso [47]; la piattaforma *Google Colaboratory* [48], servizio gratuito appartenente al pacchetto di programmi offerto da ®Google, il quale offre la possibilità di configurare notebook Jupyter, sui quali è possibile effettuare analisi dati attraverso differenti linguaggi di programmazione, tra cui Python, impiegato nel presente studio.

4.3.1 Analisi della variabile *Total Household Income*

La prima variabile che si è provveduto ad analizzare, di primaria importanza per gli scopi del presente lavoro di tesi, è la *Total Household Income*, la cui distribuzione tramite istogramma è riportata nella seguente immagine.

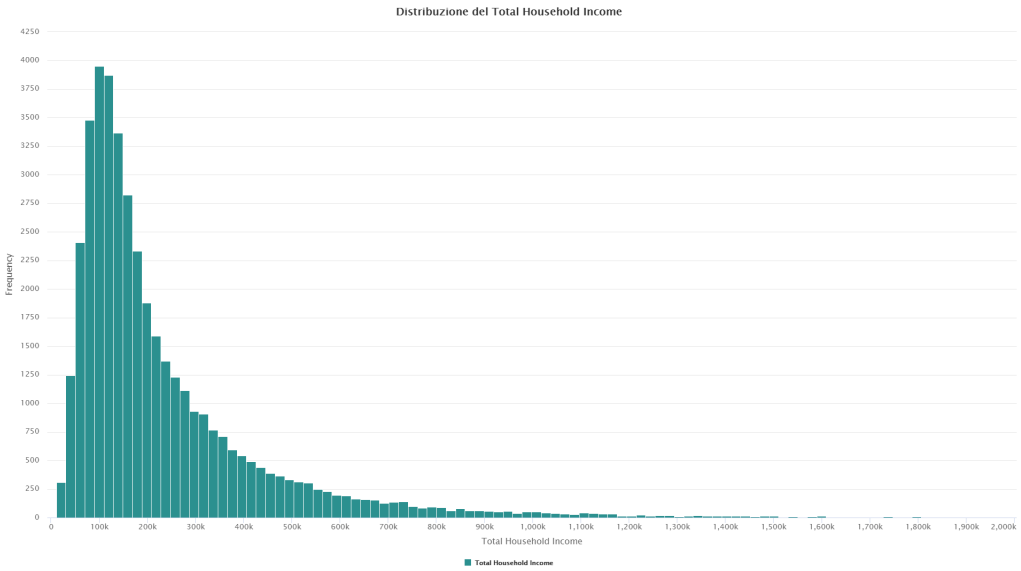

Figura 4.1: Istogramma raffigurante la distribuzione della variabile *Total Household Income* con n.bin = 100.

Il primo aspetto che si può notare, è come la maggior parte dei valori sia concentrata al di sotto della soglia dei 2M₱. Pertanto, poichè il fine ultimo dell'analisi è predire le famiglie a rischio povertà, è ragionevole suppore che i record relativi ad un THI (Total Household Income) superiore ai 2M₱ possano essere considerati degli outliers, in quanto relativi a famiglie sicuramente non a rischio povertà, e dunque, come vedremo nel paragrafo 4.4, filtrati.

In seconda istanza, si vuole comprendere la distribuzione dei records per regione, per capire se il Dataset sia bilanciato oppure se ci siano regioni sotto rappresentate.

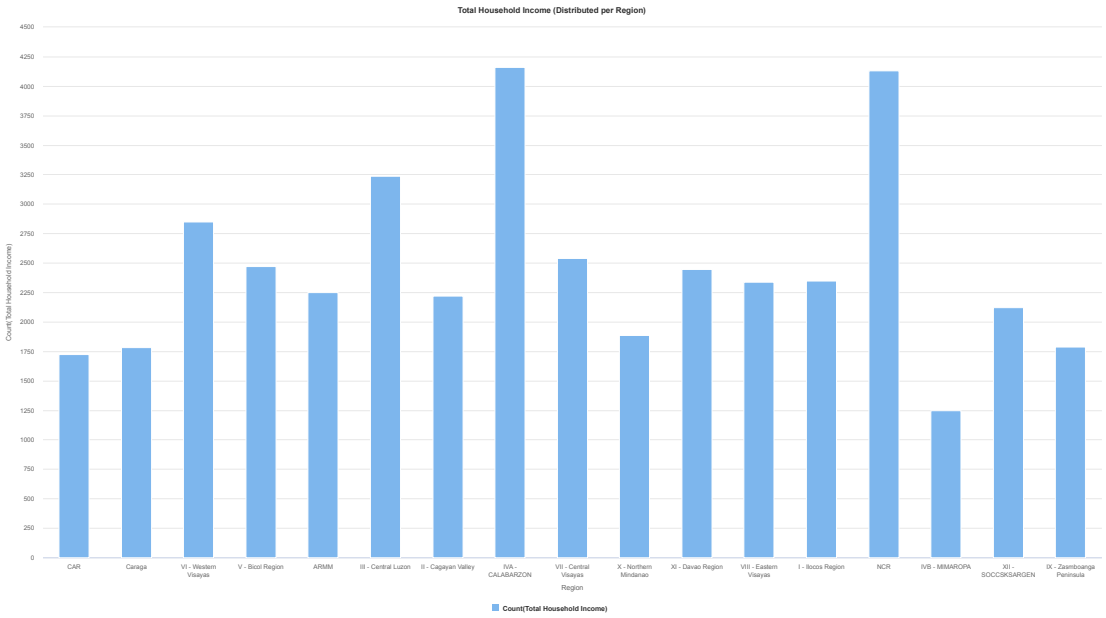

Figura 4.2: Distribuzione del numero di record per Regione.

Dal diagramma a barre di figura 4.2 si evince come ci siano due regioni particolarmente rappresentate, IV-A - Calabarzon e NCR (National Capital Region), mentre la regione del IVB - Mimaropa risulta la regione a cui appartiene il minor numero di famiglie presenti nel campione. Ciò potrebbe, tuttavia, essere ricondotto alla differenza di numero di abitanti delle regioni considerate. In particolare, Mimaropa presenta un numero di abitanti di circa 8 volte inferiore rispetto a NCR e circa 7 volte inferiore rispetto a Calabarzon, per cui è ragionevole supporre che il campione rappresentativo necessario per analizzare il comportamento delle famiglie ivi residenti sia inferiore rispetto a quello necessario per rappresentare le prime due.

A tal punto, risulta anche interessante capire come la ricchezza media, in termini di THI, sia distribuita per regione.

Dalla figura 4.3 risulta evidente come, la regione che presenta una ricchezza media decisamente superiore alle altre, con un valore di 486.861₱ annui, sia la regione NCR; la regione con il minor valore del THI medio, pari a 137.746₱ annui, invece, è quella dell' ARMM (Regione Autonoma nel Mindanao Musulmano). Ciò è perfettamente coerente con le aspettative, e con quanto descritto nel paragrafo 4.1, in quanto la regione NCR è quella contenente l'area metropolitana di Metro Manila, centro politico, economico, sociale e culturale del paese [49], mentre la regione ARMM, prima della sua soppressione nel 2019, risultava essere l'area più povera delle Filippine, a tal punto da ricevere, a dispetto della sua autonomia, finanziamenti

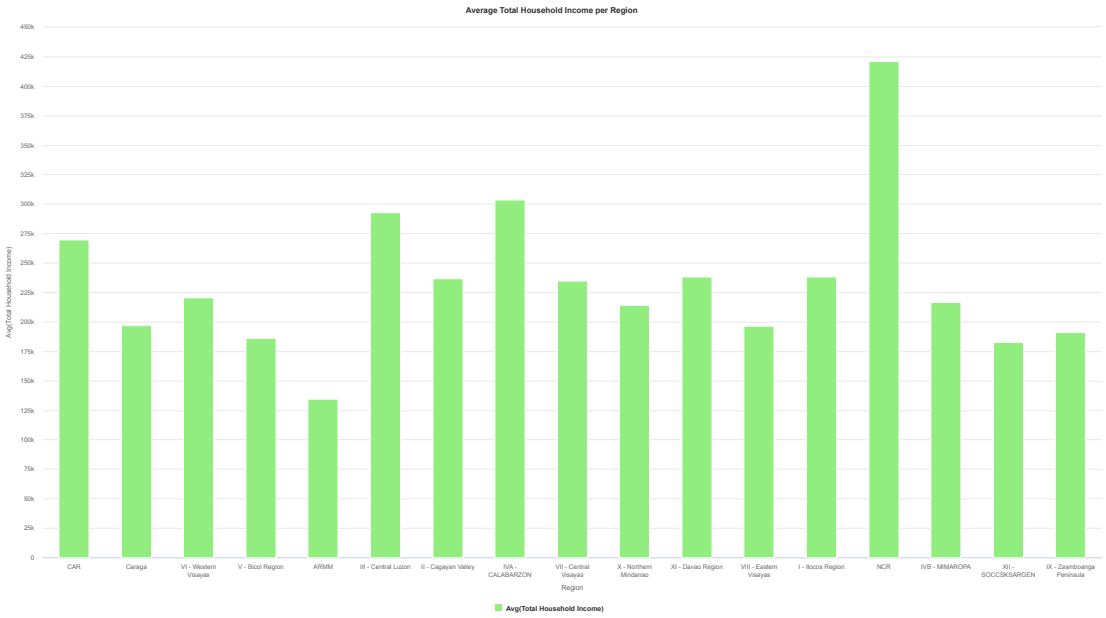

Figura 4.3: Reddito Totale Annuo Medio diviso per Regione.

per circa il 98% dal Governo centrale [50]. Le altre due regioni caratterizzate da un livello di reddito familiare annuo superiore alla media sono, la regione IV-A-Calabarzon, e la regione III-Central Luzon, le quali sono geograficamente confinanti con la regione NCR, ed appartenenti alla più grande delle tre isole che compongono le Filippine, l'isola Luzon (si veda paragrafo 4.1). Tale aspetto mette nuovamente in risalto come l'area di più ricca del paese sia quella della capitale, e come tale ricchezza si vada ad estendere fino alle sue regioni confinanti.

4.3.2 Analisi delle abitudini di Spesa delle famiglie per regione

Dopo aver analizzato gli aspetti principali legati alla distribuzione del reddito medio delle famiglie per Regione, risulta interessante capire le abitudini di spesa delle stesse nelle varie aree del Paese. Le figure dalla 4.4, alla 4.20, riportano quanto desiderato. I grafici a barre sono stati ottenuti mediante il codice Python riportato nell'Appendice A.1.

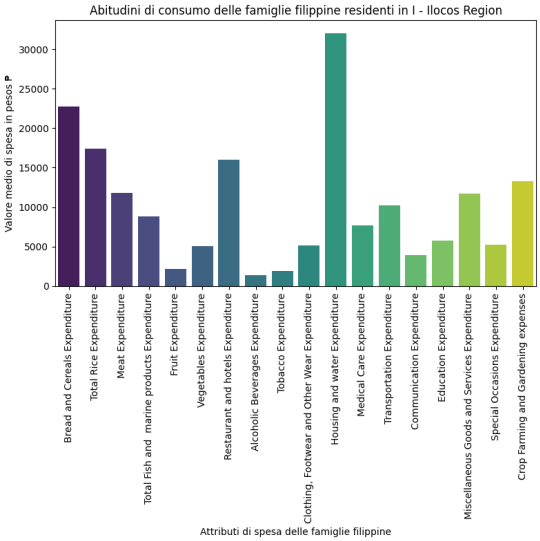

Figura 4.4: Abitudini di consumo, in media, nella Regione I-Ilocos.

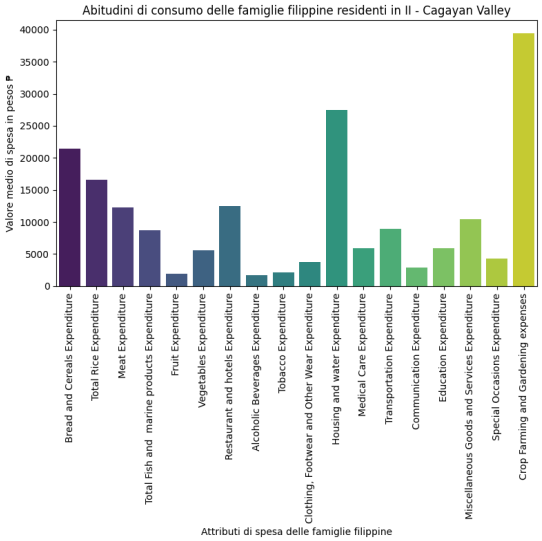

Figura 4.5: Abitudini di consumo, in media, nella Regione II-Cagayan Valley.

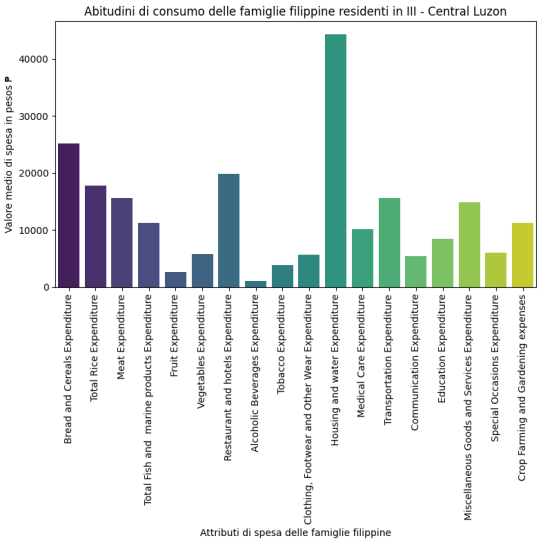

Figura 4.6: Abitudini di consumo, in media, nella Regione III-Central Luzon.

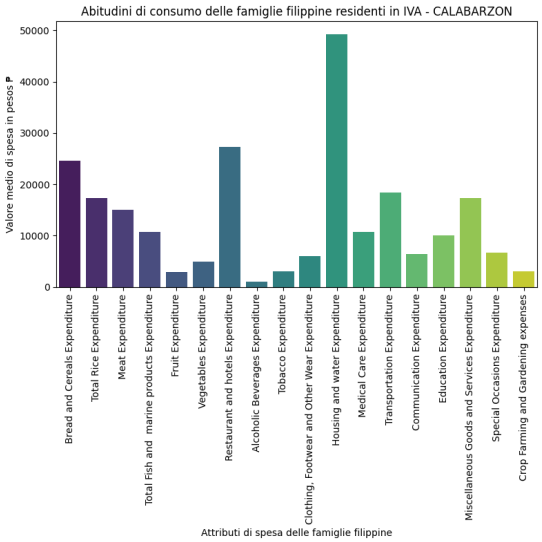

Figura 4.7: Abitudini di consumo, in media, nella Regione IV-A-Calabarzon.

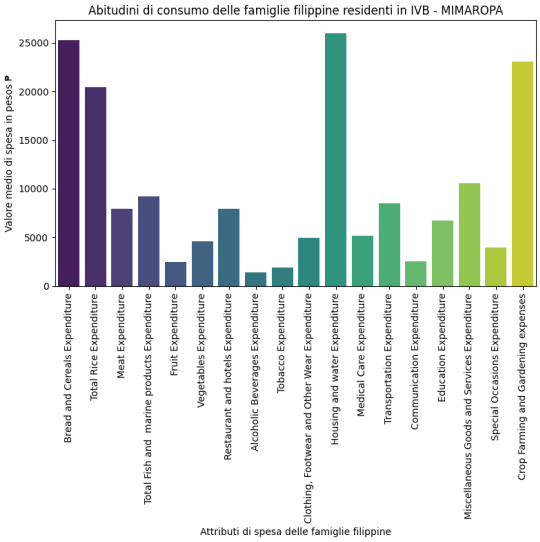

Figura 4.8: Abitudini di consumo, in media, nella Regione IV-B-Mimaropa.

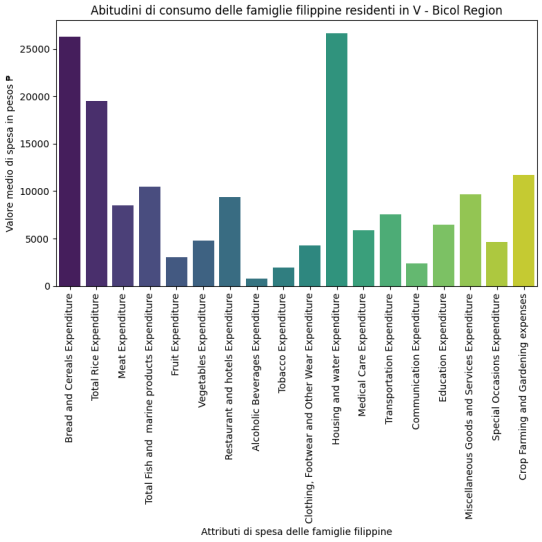

Figura 4.9: Abitudini di consumo, in media, nella Regione V-Bicol Region.

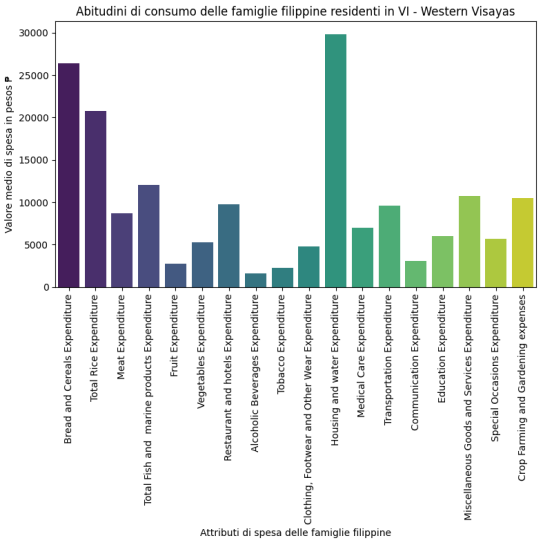

Figura 4.10: Abitudini di consumo, in media, nella Regione VI-Western Visayas.

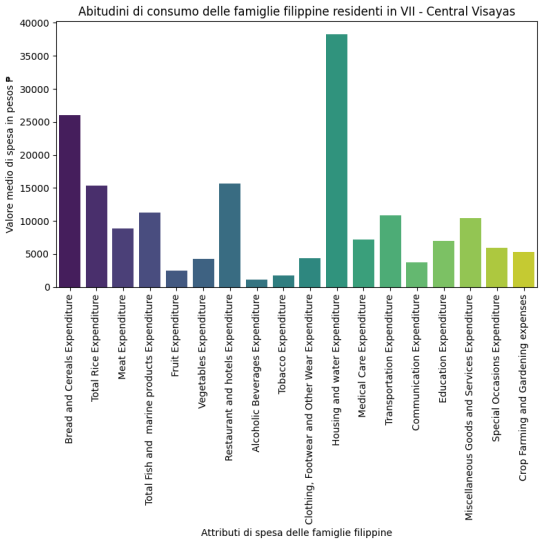

Figura 4.11: Abitudini di consumo, in media, nella Regione VII-Central-Visayas.

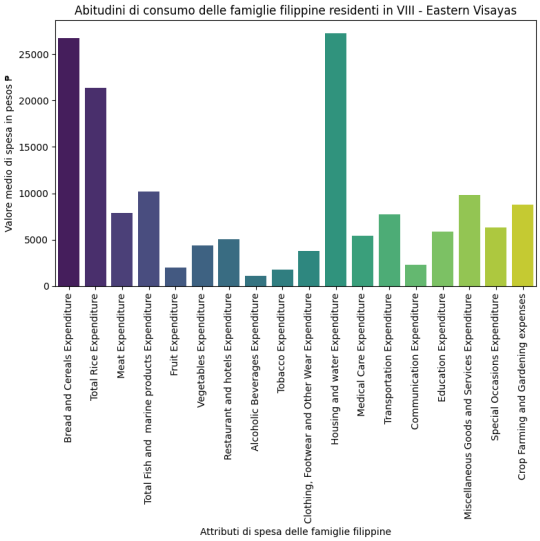

Figura 4.12: Abitudini di consumo, in media, nella Regione VIII-Eastern Visayas.

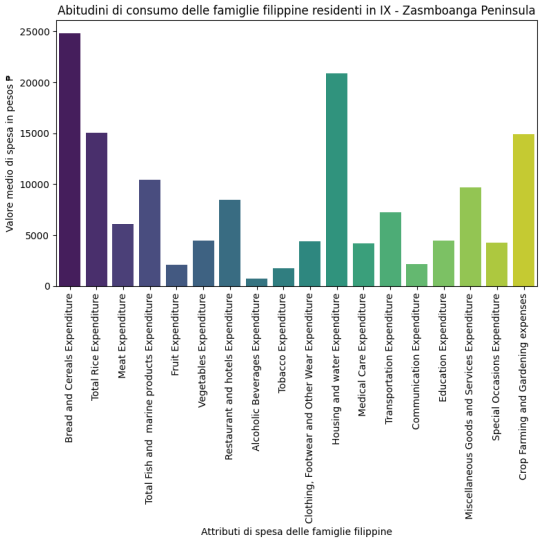

Figura 4.13: Abitudini di consumo, in media, nella Regione IX-Zasmboanga Peninsula.

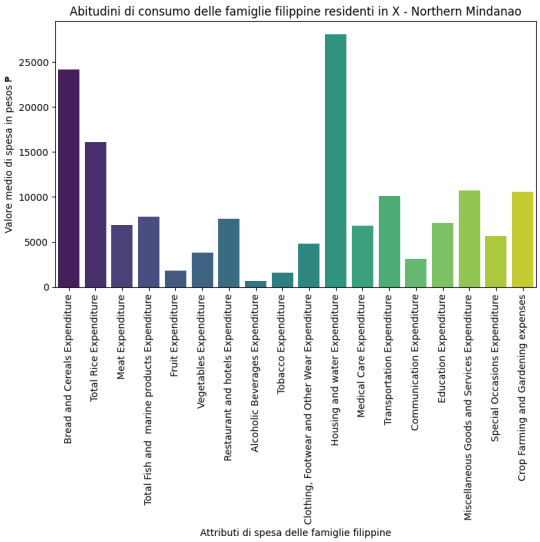

Figura 4.14: Abitudini di consumo, in media, nella Regione X-Northern Mindanao.

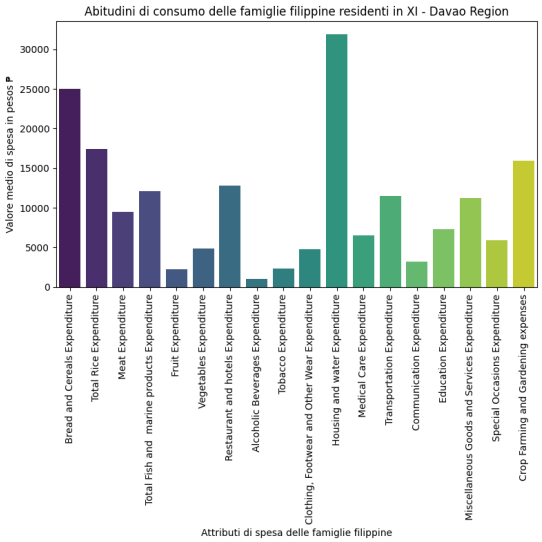

Figura 4.15: Abitudini di consumo, in media, nella Regione XI-Davao Region.

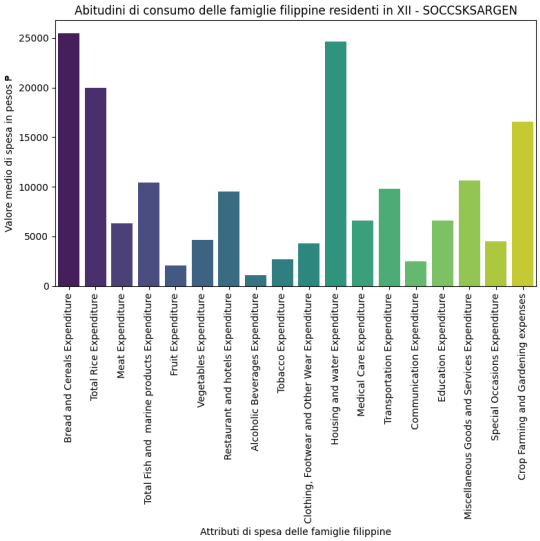

Figura 4.16: Abitudini di consumo, in media nella Regione XII-Soccsksargen.

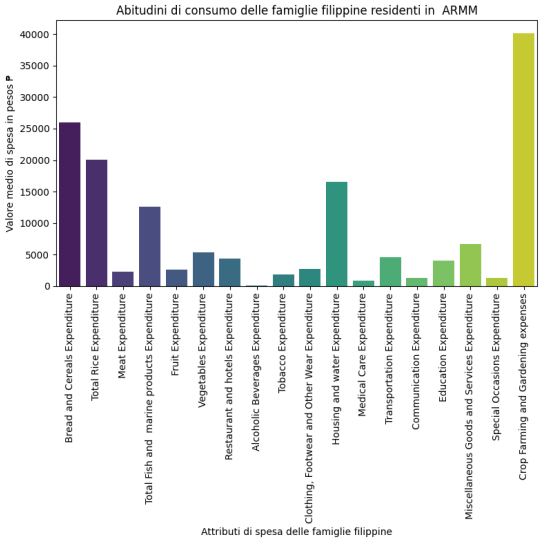

Figura 4.17: Abitudini di consumo, in media, nella Regione ARMM.

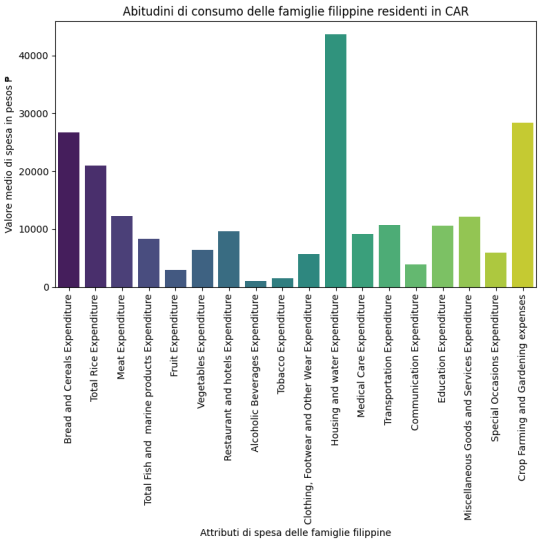

Figura 4.18: Abitudini di consumo, in media, nella Regione CAR.

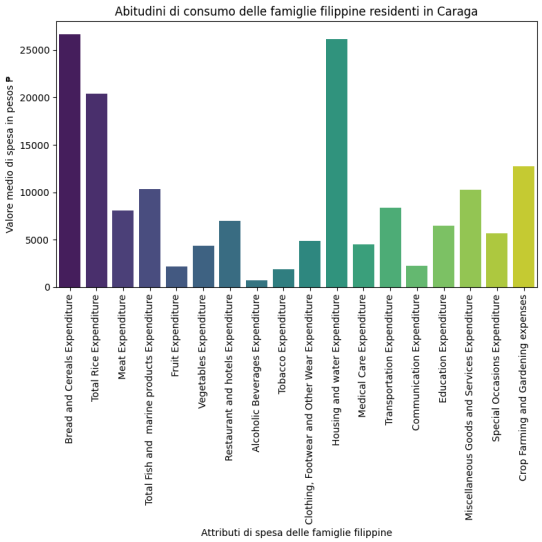

Figura 4.19: Abitudini di consumo, in media, nella Regione Caraga.

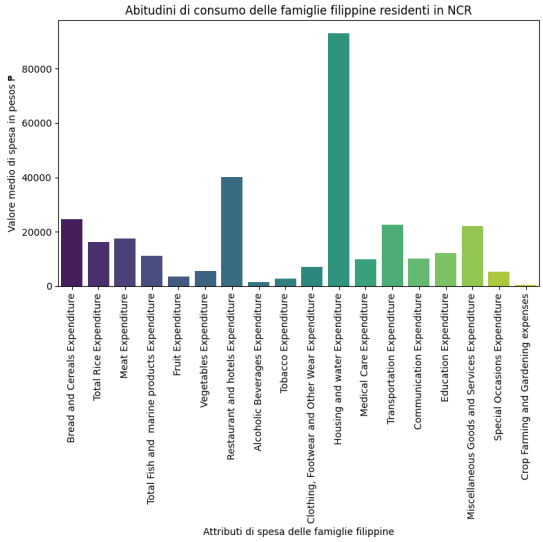

Figura 4.20: Abitudini di consumo, in media, nella Regione NCR.

Dalle immagini precedentemente mostrate, risulta evidente come le abitudini di consumo medie differiscano da regione a regione. Tuttavia, per dodici regioni su diciassette, la principale voce di spesa annua è quella relativa alle spese legate all'abitazione e all'acqua (*Housing and Water Expenditure*), seguita, per otto regioni delle dodici sopra considerate, dalle spese in pane e cereali (*Bread and Cereal Expenditure*); tale voce di spesa risulta essere, invece, quella dominante per tre delle cinque regioni in cui non dominano le spese in acqua e della casa. Per quanto riguarda le due regioni in cui la voce di spesa dominante non corrisponde né alla *Housing and Water Expenditure*, né alla *Bread and Cereal Expenditure*, ovvero le regioni II-Cagayan Valley e ARMM, la maggior porzione delle spese è quella legata all'agricoltura e al giardinaggio (*Crop Farming and Gardening expenses*).
A tal punto, si possono notare alcuni aspetti interessanti. Le regioni con minor reddito medio annuo (si veda 4.3), e dunque, in ordine di THI medio decrescente, le Regioni: Caraga, VIII-Eastern Visayas, IX-Zasmboanga Peninsula, V-Bicol Region, XII-SOCCSKSARGEN, ARMM, sono quelle per cui le principali voci di spesa sono, da un lato di carattere alimentare (*Bread and Cereal Expenditure, Total Rice Expenditure, Total Fish and marine products Expenditure*), dall'altro legate alle spese agricole (*Crop Farming and Gardening expenses*). Si vuole, pertanto, comprendere il motivo dell'importanza di quest'ultima voce di spesa, e di conseguenza si desidera analizzare le principali attività dei capofamiglia in queste regioni.

4.3.3 Analisi sul Capofamiglia

Analisi sulle principali Occupazioni svolte dai capofamiglia, per Regione

Nel presente paragrafo si procede dunque, ad analizzare le principali attività lavorative dei capofamiglia nelle regione precedentemente considerate. Le figure dalla 4.21 alla 4.26, illustrano le cinque occupazioni maggiormente svolte dai capofamiglia nelle regioni con minor THI medio. I grafici a barre sono stati ottenuti mediante il codice Python riportato nell'Appendice A.2.

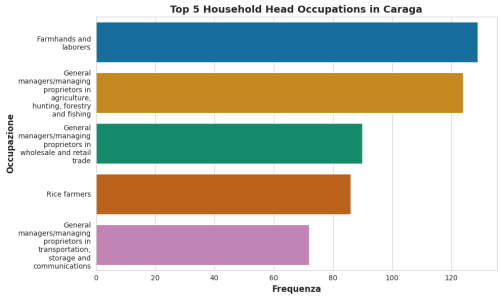

Figura 4.21: Top 5 occupazioni più frequenti dei capofamiglia nella regione di Caraga.

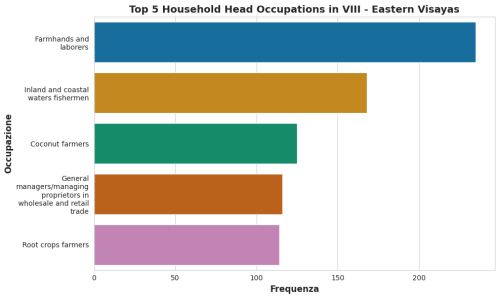

Figura 4.22: Top 5 occupazioni più frequenti dei capofamiglia nella regione VIII-Eastern Visayas

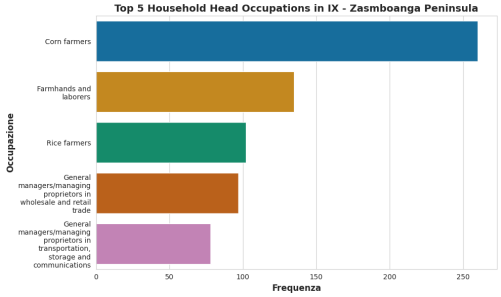

Figura 4.23: Top 5 occupazioni più frequenti dei capofamiglia in IX-Zasmboanga Peninsula.

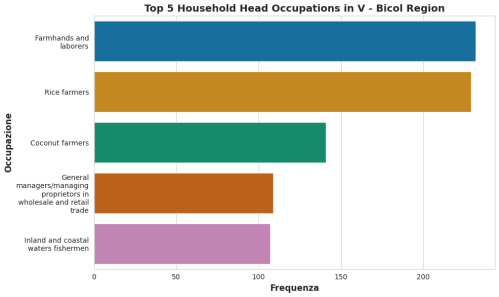

Figura 4.24: Top 5 occupazioni più frequenti dei capofamiglia nella V-Bicol Region.

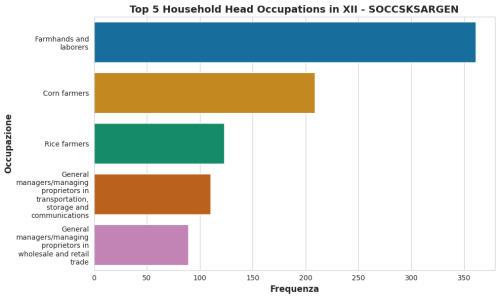

Figura 4.25: Top 5 occupazioni più frequenti dei capofamiglia nella regione XII-SOCCSKSARGEN.

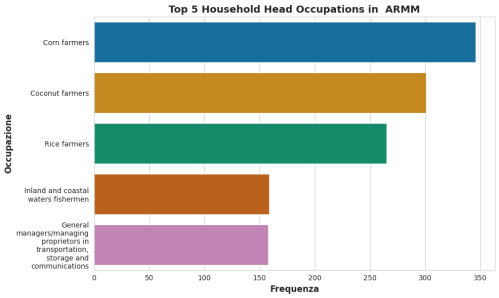

Figura 4.26: Top 5 occupazioni più frequenti dei capofamiglia nella regione ARMM

Si può immediatamente notare come, le attività lavorative maggiormente svolte dai capofamiglia nelle regioni sopracitate, siano prevalentemente legate alla natura, ed in particolare all'agricoltura e alla pesca.
A tal punto, risulta interessante capire quali siano le principali occupazioni dei capofamiglia nelle regioni con maggior THI medio annuo, per comprendere se ci siano o meno differenze circa la tipologia di attività lavorativa predominante tra le aree con maggior differenza di reddito annuo medio familgliare. Dalla figura 4.3 risulta come le regioni con maggior ricchezza media annua, con un valore di THI medio annuo superiore ai 250k ₱, siano, in ordine di THI decrescente: NCR, IVA-CALABARZON, III-Central Luzon, CAR. Nelle figure dalla 4.27 alla 4.30 vengono riportate le cinque occupazioni più frequenti per le sopracitate regioni.

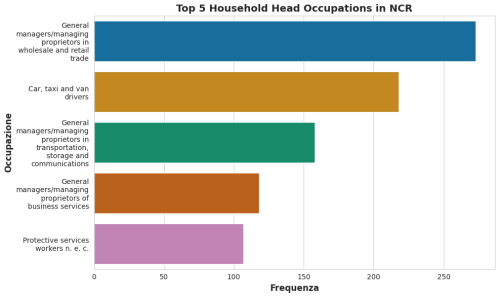

Figura 4.27: Top 5 occupazioni più frequenti dei capofamiglia nella regione NCR.

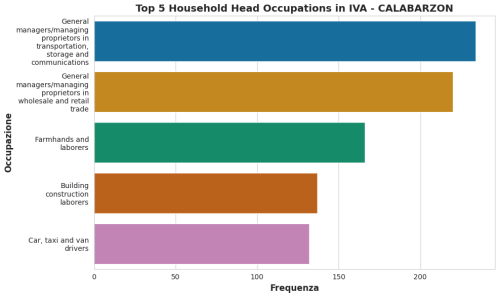

Figura 4.28: Top 5 occupazioni più frequenti dei capofamiglia nella regione IV-A-CALABARZON.

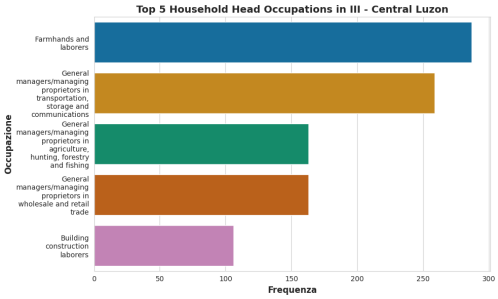

Figura 4.29: Top 5 occupazioni più frequenti dei capofamiglia nella regione III-Central Luzon.

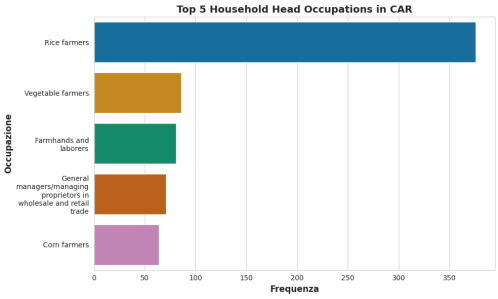

Figura 4.30: Top 5 occupazioni più frequenti dei capofamiglia nella regione CAR

Per quanto riguarda le due regioni più ricche delle Filippine, la regione NCR e la regione IV-A-Calabarzon, le principali occupazioni dei capofamiglia sono prettamente di carattere manageriale e di gestione di proprietà; per quanto riguarda la regione III-Central Luzon, le due attività principali dei capofamiglia sono quella di bracciante agricolo e quella di gestori di proprietà di trasporti, magazzinaggio e comunicazione, denotando una divisione piuttosto netta tra l'attività agricola e quella imprenditoriale. Infine, per la regione del CAR, la principale occupazione dei capofamiglia rimane quella agricola. Ciò si dimostra nuovamente in linea con il contesto socio-economico del Paese, in cui è in atto una progressiva transizione da un'economia prettamente agricola, ad una maggiormente incentrata sui servizi, ed in cui il cuore di questo sviluppo coincide, e da qui si sta pian piano diramando, con la regione della capitale, la regione NCR. In particolare, è interessante notare come, proprio nel 2015, il settore dei servizi sia responsabile del 57% del PIL, contro il 12% legato al settore agricolo, a fronte però, di una percentuale di popolazione impiegata nell'agricoltura del 32% [42]. Pertanto, nonostante una significativa fetta della popolazione filippina sia impiegata nell'industria agricola, la maggior parte della ricchezza del Paese è generata dalla ridotta porzione di lavoratori impiegati nel settore dei servizi.

Analisi sul Sesso del capofamiglia

Nel presente paragrafo si intende fare un approfondimento sul sesso del capofamiglia, per capire quale sia la percentuale di capofamiglia uomini e quale sia quelle di donne. È inoltre, interessante capire la tipologia di famiglia a cui il capofamiglia, uomo o donna che sia, fa riferimento.

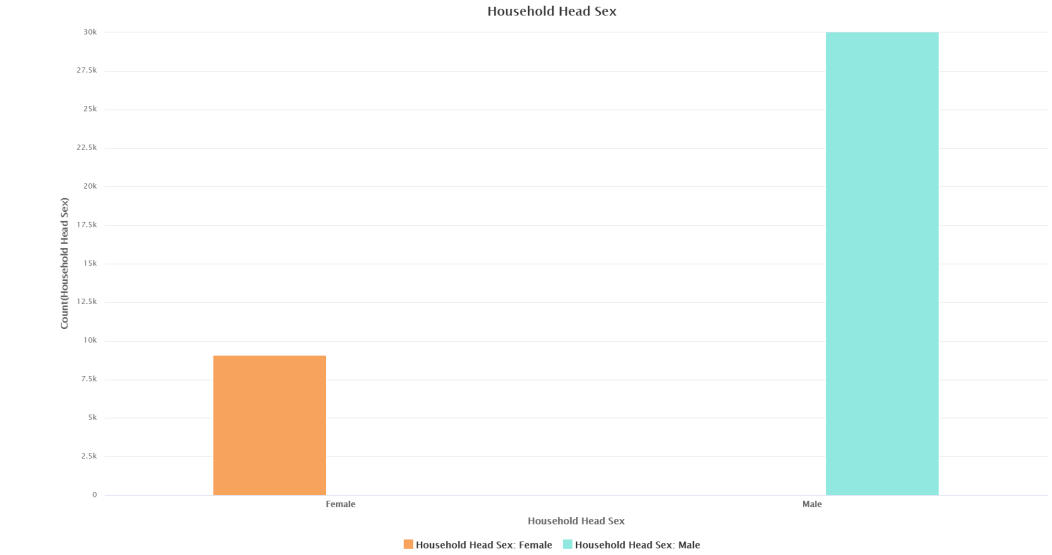

Figura 4.31: Distribuzione del sesso del capofamiglia

Dalla figura 4.31, si può notare come la percentuale di capofamiglia uomini, pari a 72,19% del totale, sia decisamente superiore rispetto a quella dei capofamiglia donne, pari a 27,81%. A tal punto, può essere interessante capire la tipologia di famiglia, di cui questi uomini e queste donne sono a capo, e capire se ci sono differenze nel sesso del capofamiglia, in base al tipo di famiglia. La figura 4.32 mostra il risultato di questa analisi.

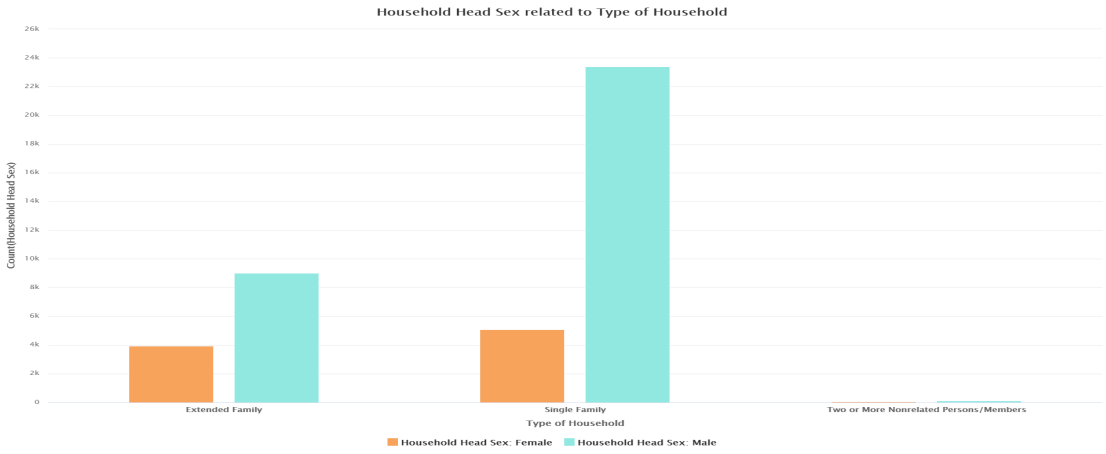

Figura 4.32: Sesso del capofamiglia in relazione alla tipologia di famiglia

Si può notare dunque, come, a prescindere dalla tipologia di famiglia considerata, e dunque se si tratti di famiglia singola, o di famiglia estesa, il genere maschile sia

quello predominante per i capofamiglia.
Andando ad analizzare, invece, il sesso del capofamiglia, in relazione allo stato civile dello stesso, si osserva come ci siano casi in cui il genere femminile sia prevalente rispetto a quello maschile, in particolare nel caso di separazione/divorzio e di separazione dovuta alla morte del partner, come si può vedere in figura 4.33

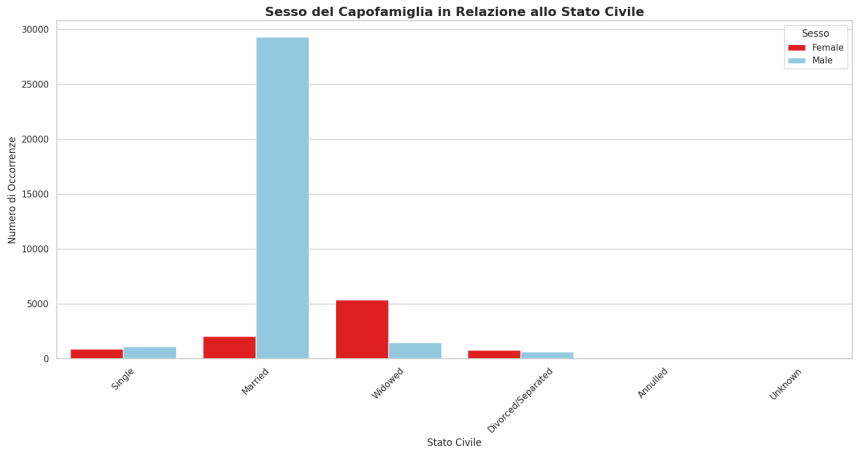

Figura 4.33: Sesso del capofamiglia in relazione allo stato civile

Analisi sull' Età del capofamiglia

Procedendo nell'analisi delle caratteristiche dei capofamiglia, nel presente paragrafo si intende esplorare l'età dei capofamiglia, al fine di individuare la fascia di età maggiormente rappresentata. In figura 4.34 si può osservare quanto descritto.

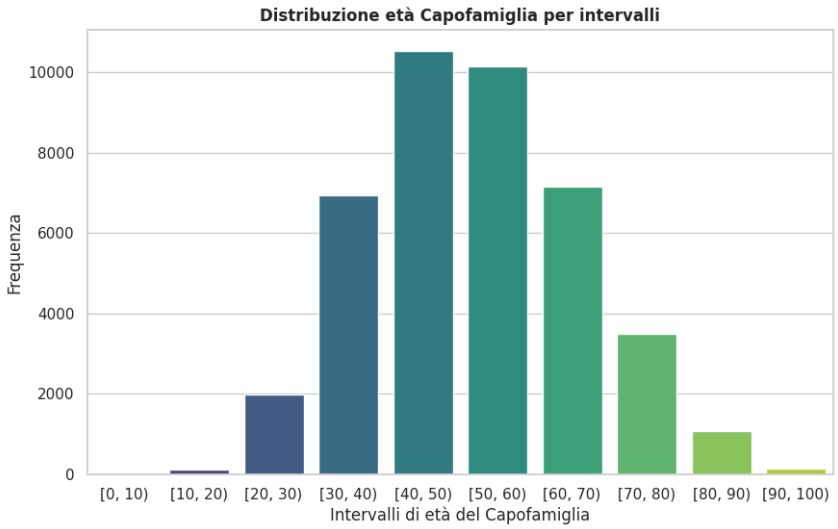

Figura 4.34: Distribuzione età dei capofamiglia, per intervalli

Si può notare, dunque, come la fascia di età maggiormente rappresentata sia quella tra i 40 ed i 50 anni. Tuttavia, differenziando questa analisi per il sesso dei capofamiglia, il risultato cambia. In particolare, se per gli uomini, la fascia di età predominante rimane quella tra i 40 ed i 50 anni, per le donne, diventa quella tra i 60 ed i 70 anni, come mostrato in figura 4.35.

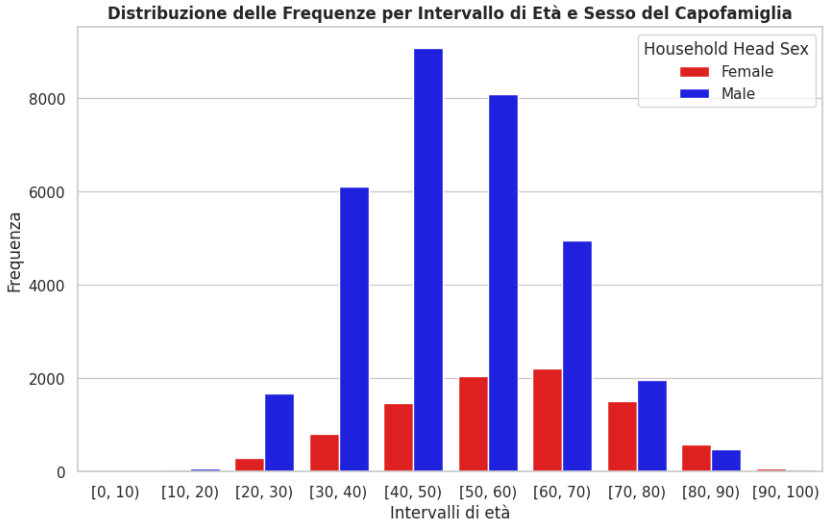

Figura 4.35: Distribuzione età dei capofamiglia per sesso

Analisi sul Livello di Istruzione del capofamiglia

Un altro punto importante per la caratterizzazione dei capofamiglia delle famiglie filippine, è quello dell'istruzione. Si vuole, innanzitutto, capire i livelli di istruzione maggiormente frequenti nei capofamiglia del paese, e da qui, poi, ampliare l'analisi. In figura 4.36 è illustrato il grado di istruzione dei capofamiglia, in ordine decrescente di frequenza.

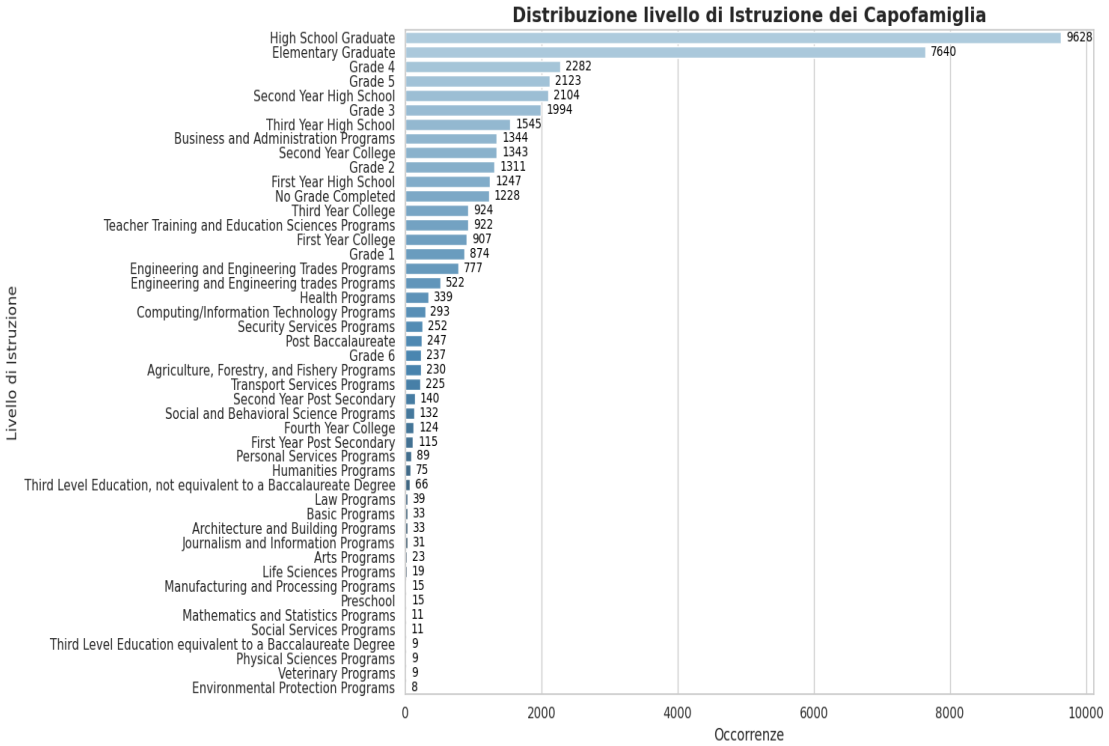

Figura 4.36: Distribuzione livello di istruzione dei capofamiglia

Dalla figura 4.36, si evince come, il livello di istruzione dei capofamiglia, sia piuttosto vario. Se da un lato, infatti, il grado di istruzione predominante è quello del diploma di scuola superiore, dall'altro, le successive tre voci sono relative ad un livello di istruzione elementare, o addirittura inferiore. Molto ridotta è, invece, la percentuale di persone che hanno completato un programma di studi equivalente ad un percorso di laurea; tra questi, tuttavia, i programmi di Business and Administration sono quelli che svettano per frequenza. La figura 4.37, invece, mostra la percentuale di uomini e di donne, per ogni livello di istruzione.

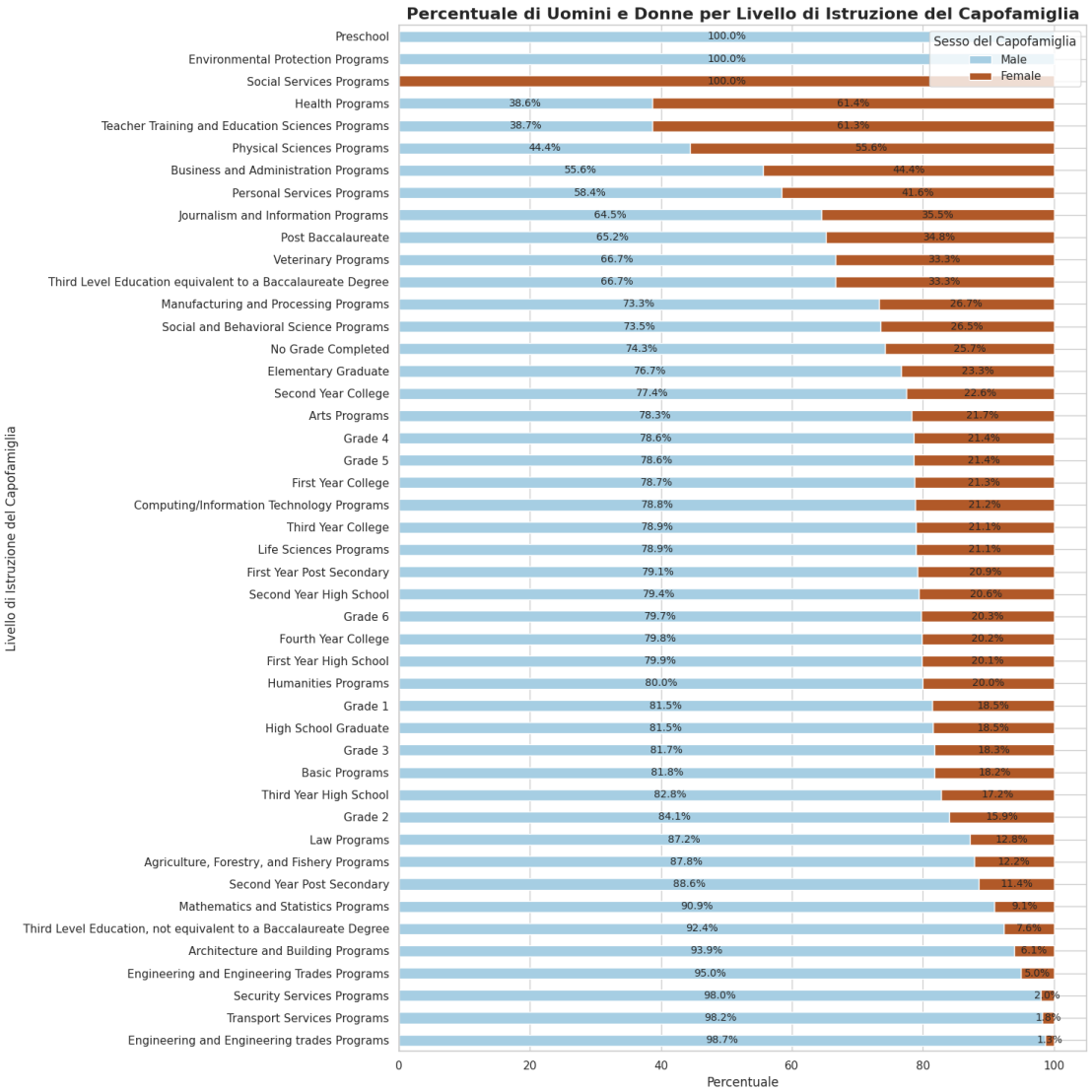

Figura 4.37: Livello di istruzione del capofamiglia per genere

A primo impatto, si può notare come ci siano numerosi gradi di istruzione a prevalenza maschile, e solamente quattro, peraltro altamente specializzanti, come ad esempio programmi di servizi sociali o in medicina, a prevalenza femminile.
La cosa più interessante, tuttavia, è che, calcolando la percentuale di donne capofamiglia con un'istruzione inferiore o uguale al diploma elementare, essa sia significativamente inferiore a quella maschile, con un valore pari al 28.4 % contro il 39.9% di quella degli uomini. Ciò suggerisce come, nonostante in valore assoluto, il numero di capofamiglia donne sia nettamente inferiore a quello degli uomini (figura 4.31), il loro livello di istruzione sia, in media, superiore a quello maschile.

A tal punto, dopo aver analizzato il livello di istruzione per genere, risulta interessante capire come, e se, varia il grado di istruzione per regione, ed in particolare se esistono delle differenze tra le regioni con maggiore e minore reddito annuo medio. Le figure dalla 4.38 alla 4.43, generate mediante il codice riportato in Appendice A.3, mostrano il livello di istruzione dei capofamiglia nelle regioni con minor reddito annuo medio.

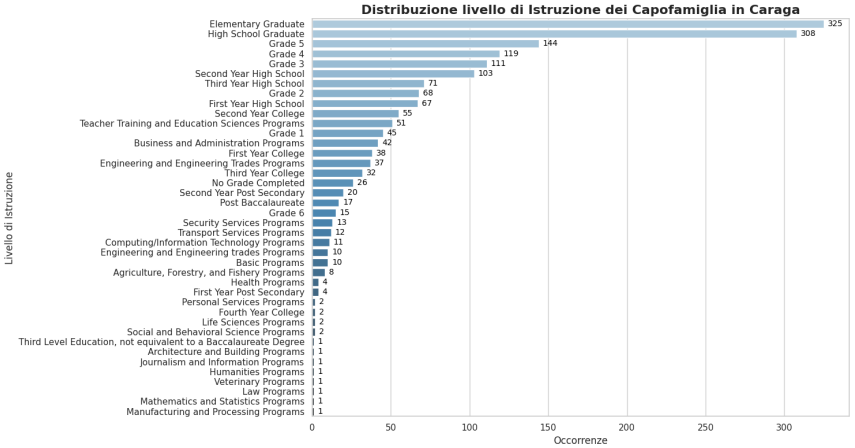

Figura 4.38: Livello di istruzione dei capofamiglia nella regione di Caraga.

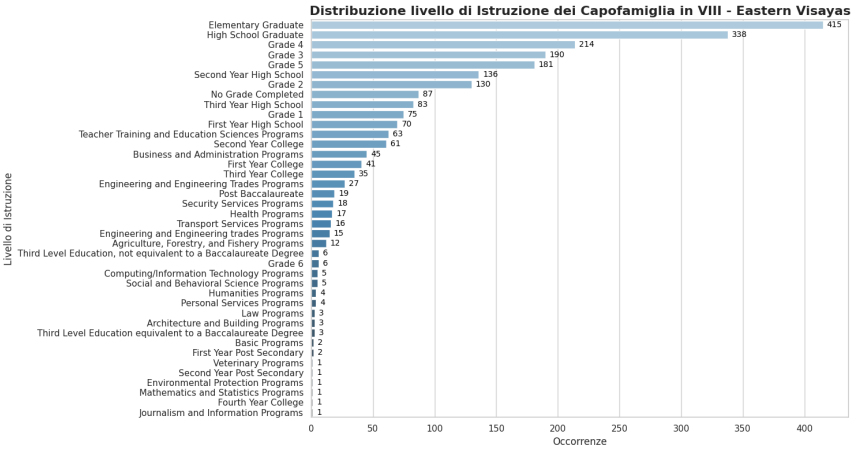

Figura 4.39: Livello di istruzione dei capofamiglia nella regione VIII-Eastern Visayas.

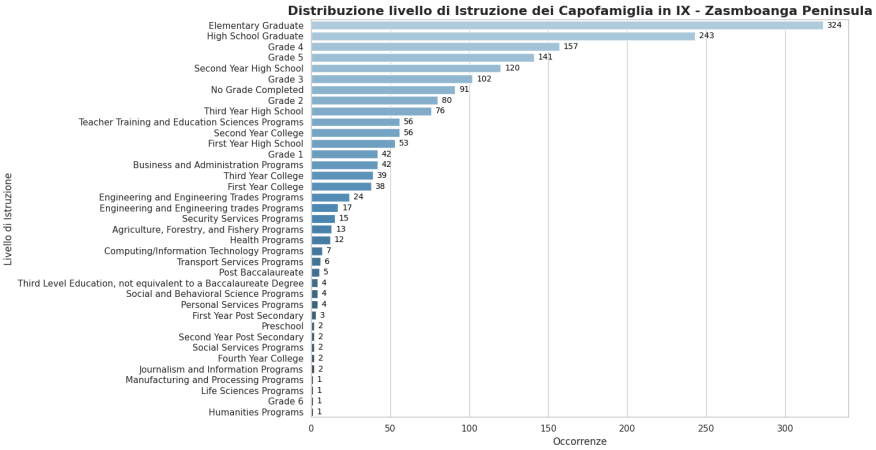

Figura 4.40: Livello di istruzione dei capofamiglia in IX-Zasmboanga Peninsula.

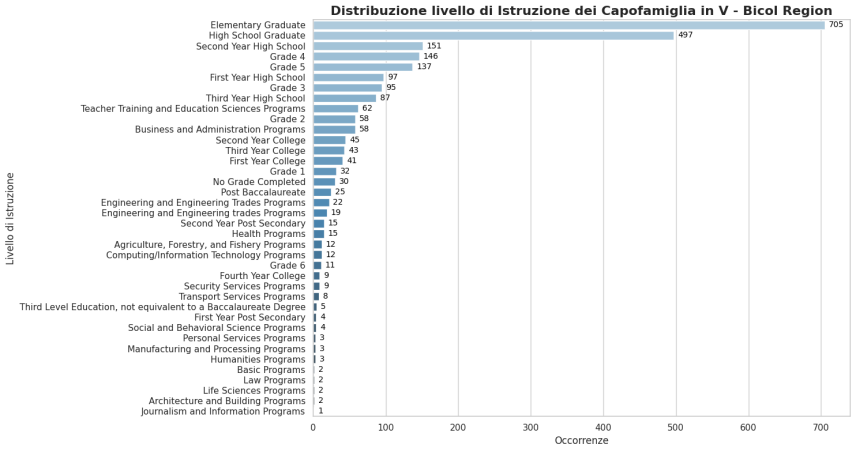

Figura 4.41: Livello di istruzione dei capofamiglia nella V-Bicol Region.

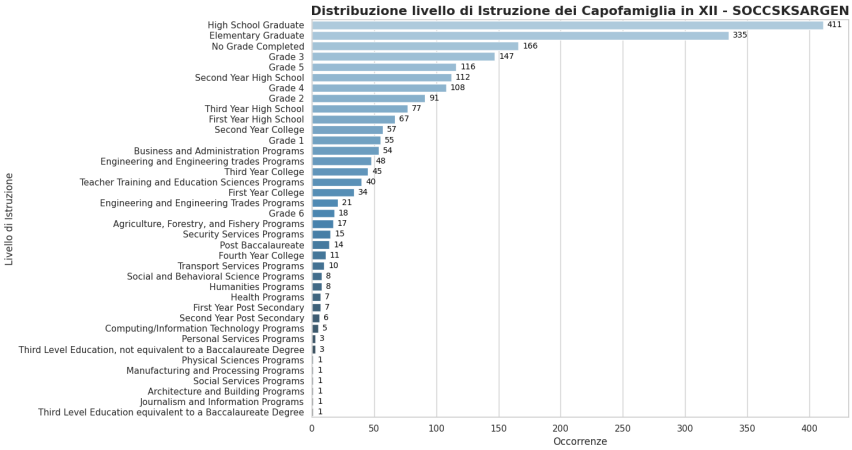

Figura 4.42: Livello di istruzione dei capofamiglia nella regione XII-Soccsksargen.

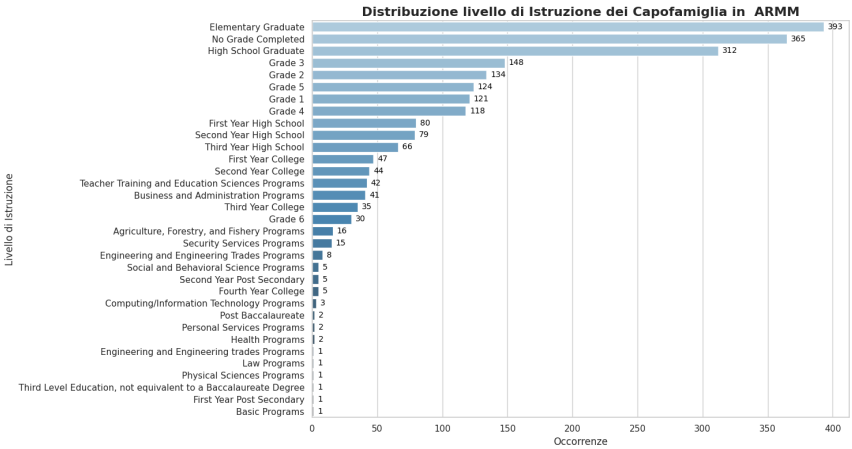

Figura 4.43: Livello di istruzione dei capofamiglia nella regione ARMM

È interessante notare come, in tutte le regioni, ad eccezione della regione ARMM e della regione XII-Soccsksargen, il grado di istruzione più frequente sia il diploma elementare, seguito dal diploma di scuola superiore. Per quanto riguarda la regione XII-Soccsksargen, invece, il livello di istruzione principale è il diploma di scuola superiore, seguito da quello elementare. Infine, per quanto concerne la regione ARMM, prevale il diploma elementare, seguito da coloro che non hanno ottenuto nessun titolo di studio. Soltanto in terza posizione si trovano coloro che hanno conseguito un diploma di scuola superiore. Ciò è nuovamente coerente con le aspettative, proprio in relazione al fatto che la regione ARMM è la più povera del paese.

Le figure dalla 4.44 alla 4.47 mostrano, invece, il livello di istruzione dei capofamiglia nelle regioni con maggior THI medio.

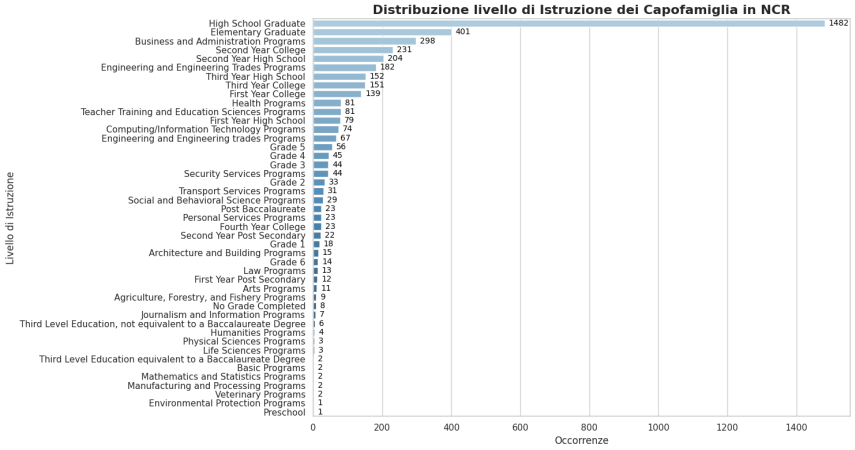

Figura 4.44: Livello di istruzione dei capofamiglia nella regione NCR.

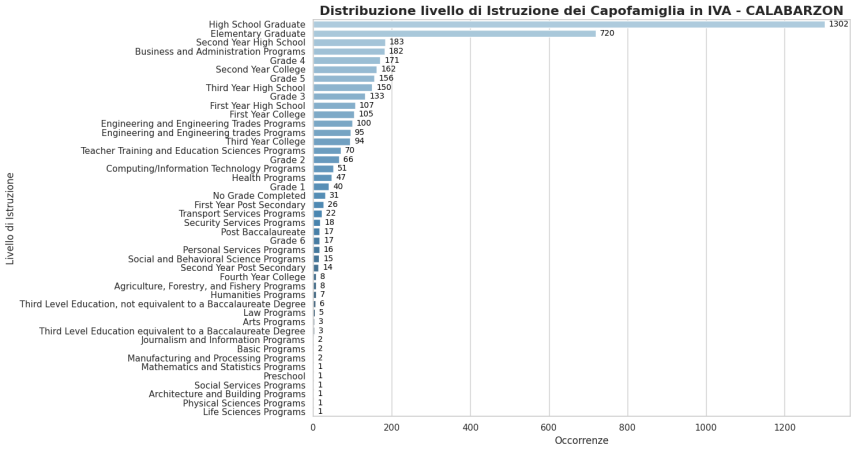

Figura 4.45: Livello di istruzione dei capofamiglia nella regione IVA-Calabarzon.

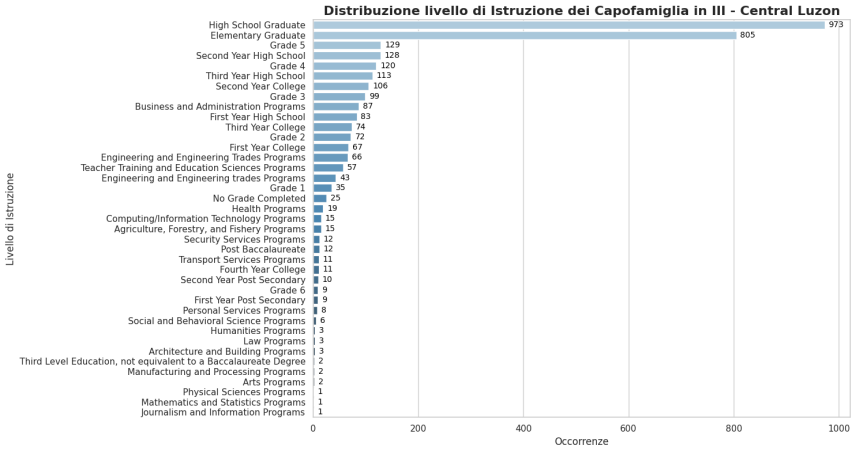

Figura 4.46: Livello di istruzione dei capofamiglia nella regione III-Central Luzon.

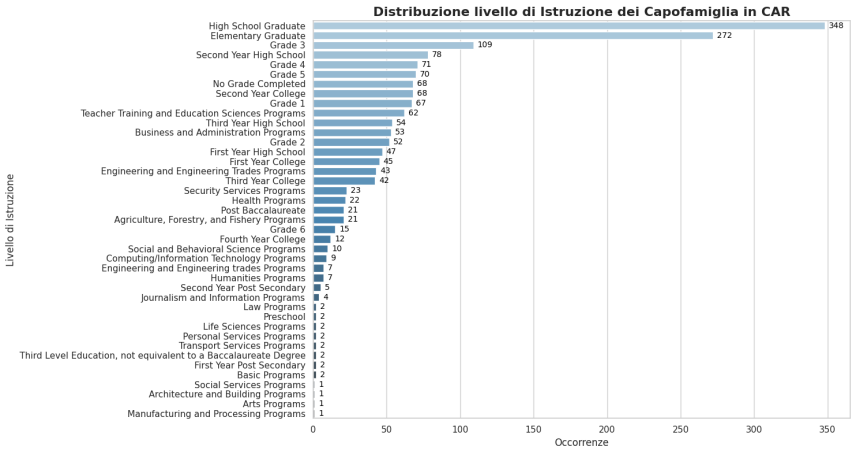

Figura 4.47: Livello di istruzione dei capofamiglia nella regione CAR

In tutte le regioni considerate, il grado di istruzione principale dei capofamiglia è il diploma di scuola superiore, seguito dal diploma elementare. In particolare, come si evince dalla figura 4.44, nella regione NCR sono presenti in numero significativo diverse persone che hanno conseguito studi di livello superiore, come programmi di Business and Administration e di Ingegneria. Nuovamente il risultato ottenuto è coerente con quanto ci si aspettava, essendo la regione NCR il centro economico, politico e culturale del paese.

A tal punto, diventa interessante capire se ci sia una correlazione tra il livello di istruzione del capofamiglia, ed il THI medio annuo. La figura 4.48, generata mediante il codice python riportato in Appendice A.4 mostra una serie di boxplot aventi l'obiettivo di illustrare tale analisi.

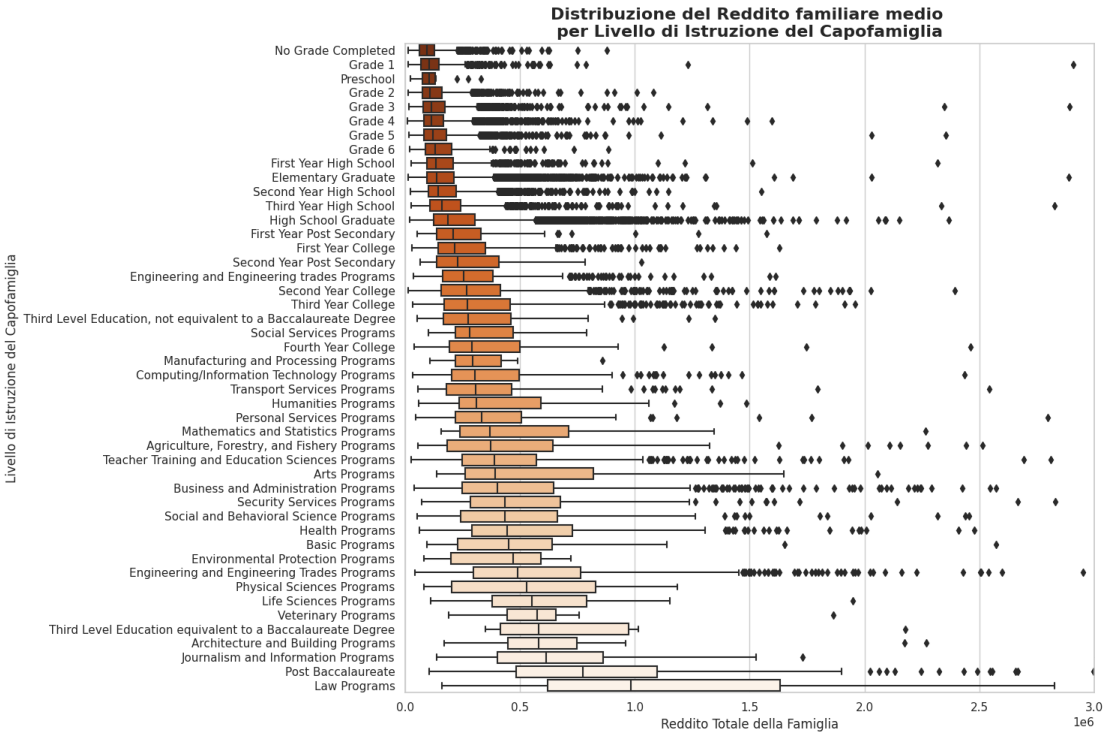

Figura 4.48: Distribuzione del THI medio annuo in relazione al livello di istruzione del capofamiglia.

È molto interessante notare come il livello medio di reddito familiare sia crescente al crescere del livello di istruzione del capofamiglia (livello medio indicato dalla linea centrale dei boxplot), e come dunque, in media, le famiglie il cui capofamiglia ha un grado di istruzione elevato, in particolare una laurea oppure un programma equivalente, siano quelle con maggior reddito medio annuo.
Tuttavia, dal grafico si evince la presenza di un discreto numero di outliers, specialmente per quanto riguarda i gradi di istruzione inferiori, stanti ad indicare situazioni in cui, nonostante il capofamiglia sia poco istruito, il reddito annuo medio familiare sia comunque più alto della media. Ciò potrebbe, però, essere imputabile a diversi fattori, ad esempio la presenza in famiglia di figure che ricoprono lavori altamente retribuiti, oppure con un grado di istruzione superiore rispetto al capofamiglia. In ogni caso, però, con i dati forniti, non è possibile individuare le ragioni di tale anomalia, e dunque non è possibile approfondire ulteriormente tale aspetto.

4.3.4 Analisi sula Composizione delle famiglie

L'obiettivo del presente paragrafo è quello di analizzare la composizione delle famiglie filippine. La figura 4.49 illustra le occorrenze del numero di membri delle

famiglie filippine. Si può notare come le famiglie filippine siano tendenzialmente molto numerose, con un numero medio di membri pari a 5.

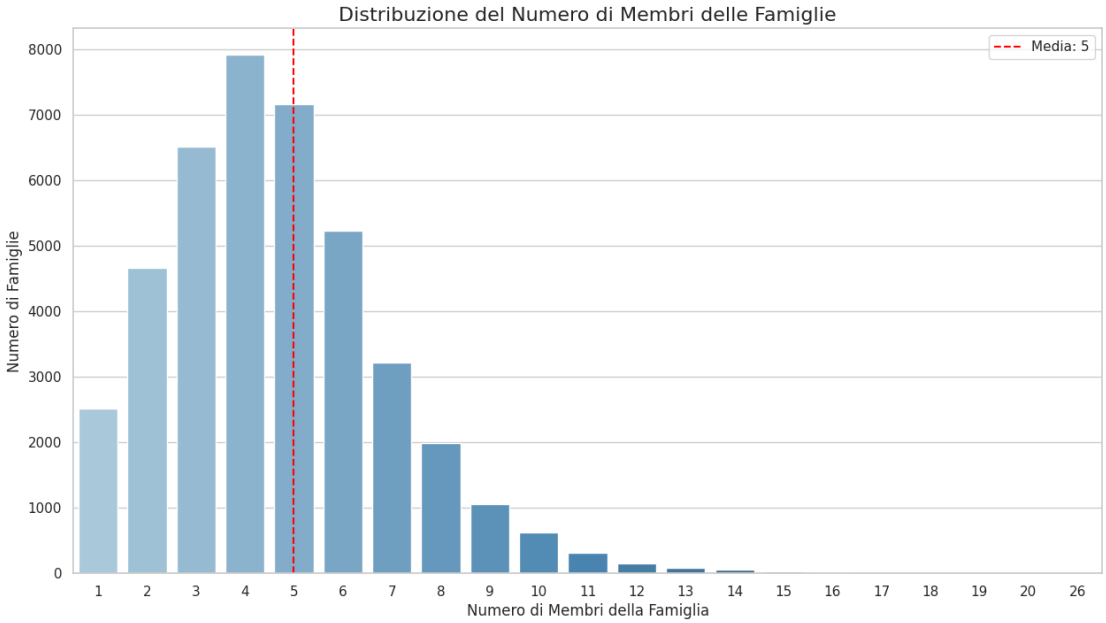

Figura 4.49: Distribuzione del numero di membri delle famiglie filippine

A tal punto, può risultare interessante vedere come varia la distribuzione del numero di membri delle famiglie, al variare della tipologia di famiglia, e dunque se si considerano le *Single Family* oppure le *Extended Family*. La figura 4.50 mostra quanto descritto.

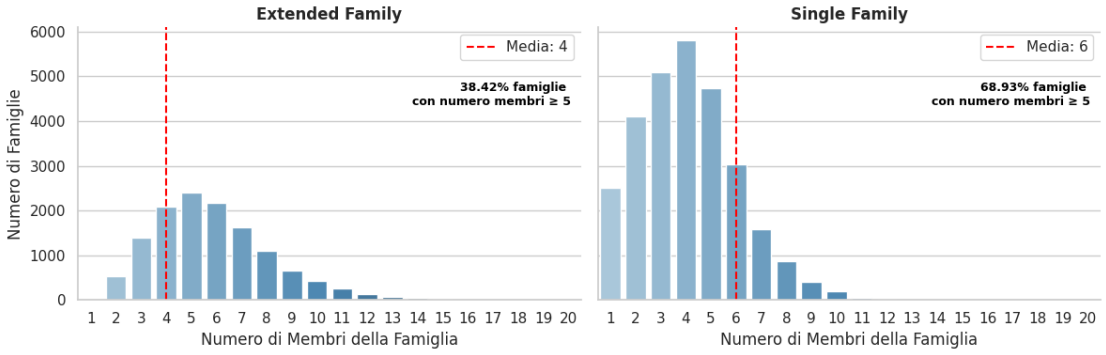

Figura 4.50: Distribuzione del numero di membri delle famiglie filippine differenziata per tipologia di famiglia

Sorprendentemente, il numero medio di membri delle *Single Family*, pari a sei, è superiore a quello delle *Extended Family*, pari a quattro. Ciò potrebbe essere dovuto alla maggiore numerosità delle Single Family, eppure, andando a calcolare la percentuale di famiglie aventi un numero di membri superiore o uguale a 5, si osserva come per le Single Family questo valore sia pari a 68.93%, mentre per le Extended Family sia 38.42%.

Avendo analizzato la numerosità delle famiglie filippine, può risultare interessante approfondire tale aspetto, cercando di capire come sia distribuita la numerosità di membri con età inferiore ai 5 anni, oppure con età compresa tra i 5 ed i 17 anni, all'interno delle famiglie stesse. In figura 4.51 viene rappresentato tale aspetto.

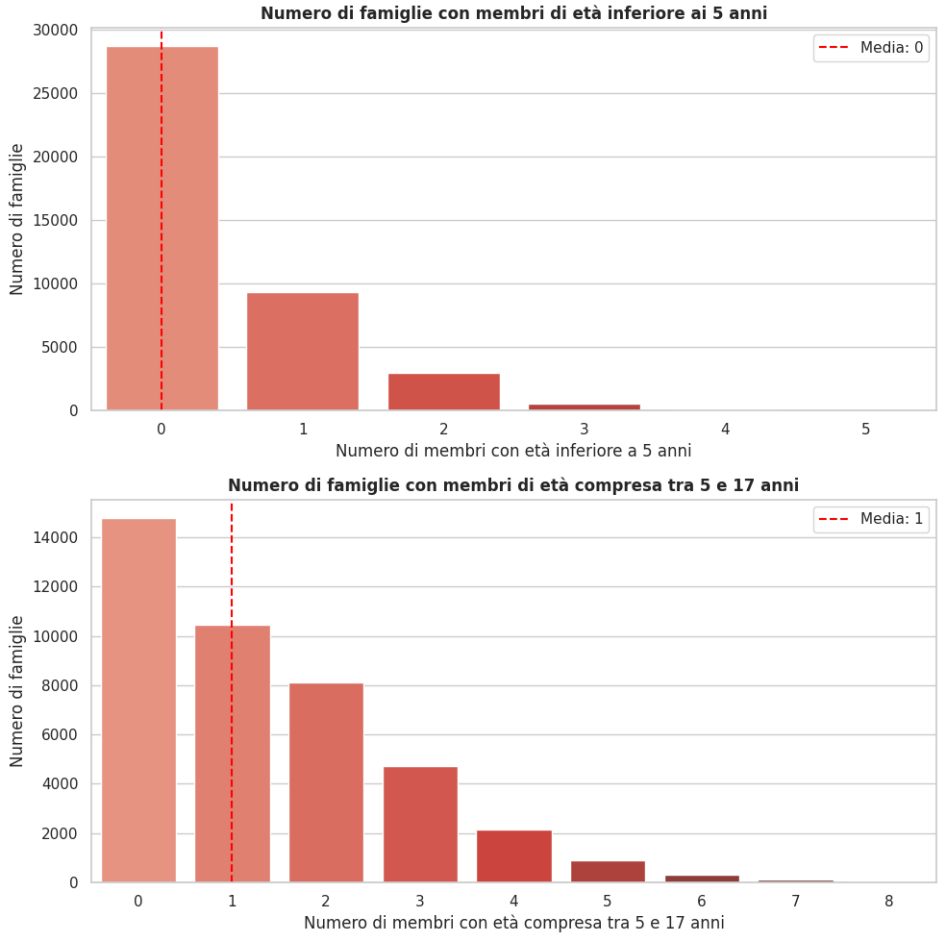

Figura 4.51: Distribuzione del numero di membri delle famiglie con età inferiore ai 18 anni

Si può osservare come, in media, la maggior parte delle famiglie filippine non presenta membri con età inferiore ai 5 anni, mentre, sempre in media, esse presentano

membri con età compresa tra i 5 ed i 17 anni, e dunque al di sotto dell'età minima per l'assunzione lavorativa nel paese.
Per completezza, si vuole osservare se tale valore varia in base alla tipologia della famiglia. Le figure 4.52 e 4.53 riportano quanto descritto.

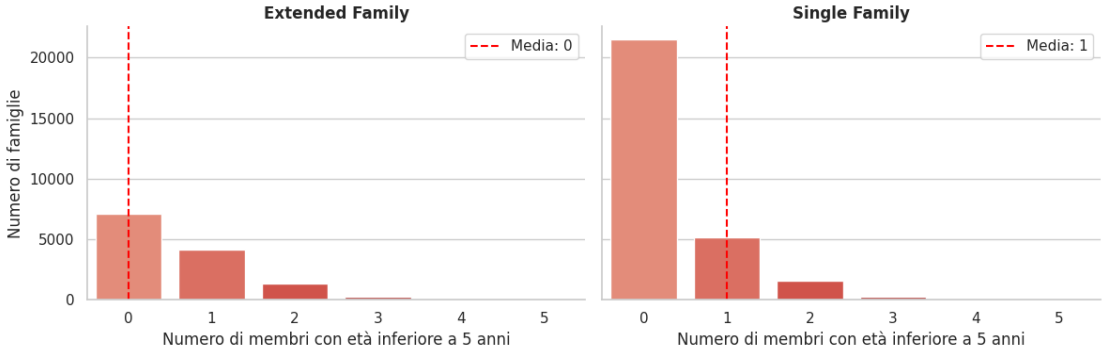

Figura 4.52: Distribuzione del numero di membri delle famiglie con età inferiore ai 5 anni, differenziata per tipologia di famiglia

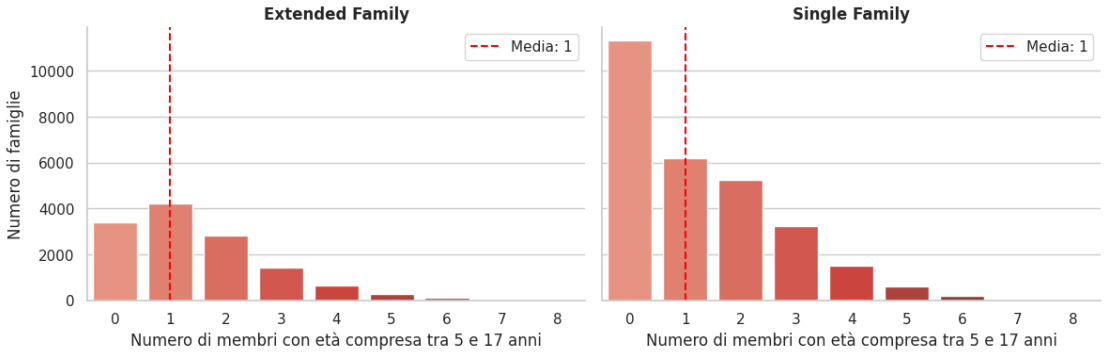

Figura 4.53: Distribuzione del numero di membri delle famiglie con età compresa tra 5 e 17 anni, differenziata per tipologia di famiglia

Dalla figura 4.52 si evince come per le *Single Family*, il numero medio di membri con età inferiore a 5 anni sia uno, così come il numero medio di membri con età compresa tra i 5 ed i 17 anni. Per quanto riguarda le *Extended Family*, invece, la situazione resta invariata, e dunque analoga a quella descritta in precedenza e facente riferimento alla figura 4.51.

È doveroso specificare che i valori medi illustrati nelle figure precedenti sono valori approssimati all'intero più vicino, in quanto, per ovvi motivi, non avrebbe senso, trattandosi di persone, considerare il valore effettivo della media, e dunque, un valore decimale.

Ciò detto, si procede con l'analisi della distribuzione del numero di membri delle famiglie occupati. In figura 4.54 è riportata la distribuzione del numero di membri delle famiglie occupati.

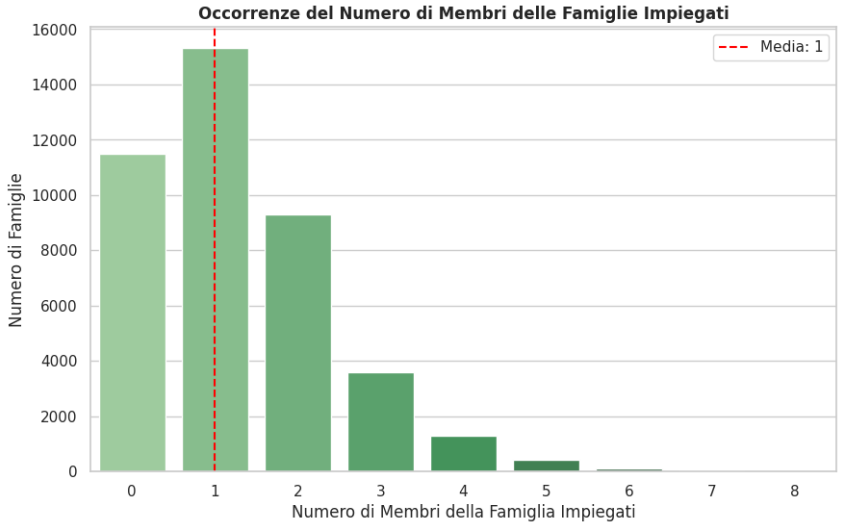

Figura 4.54: Distribuzione del numero di membri delle famiglie occupati

È immediato notare come il valor medio del numero di membri occupati nelle famiglie filippine sia uno.

Come fatto in precedenza, anche in questo caso è interessante vedere se ci sono differenze in base alla tipologia di famiglia. La figura 4.55 mostra quanto detto.

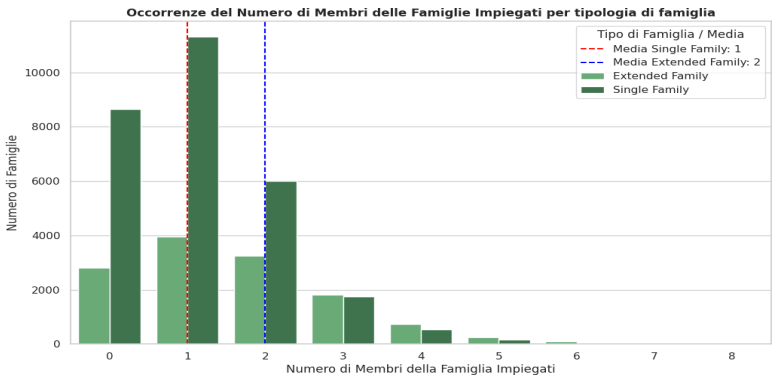

Figura 4.55: Distribuzione del numero di membri delle famiglie occupati, differenziata per tipologia di famiglia

Differenziando l'analisi per tipologia di famiglia, il valor medio di membri occupati nelle *Extended Family* aumenta, arrivando a 2. Dunque, mediamente,

in questa tipologia di famiglie, c'è almeno un altra persona che lavora oltre al capofamiglia.

Infine, si vuole fare un'analisi sul numero di membri occupati in relazione al numero di membri della famiglia, per capire se all'aumentare della numerosità della famiglia, aumentano gli occupati. La figura 4.56, generata mediante il codice python riportato in Appendice A.5 illustra la distribuzione degli occupati in relazione alla numerosità della famiglia, differenziata per numerosità e tipologia di famiglia.

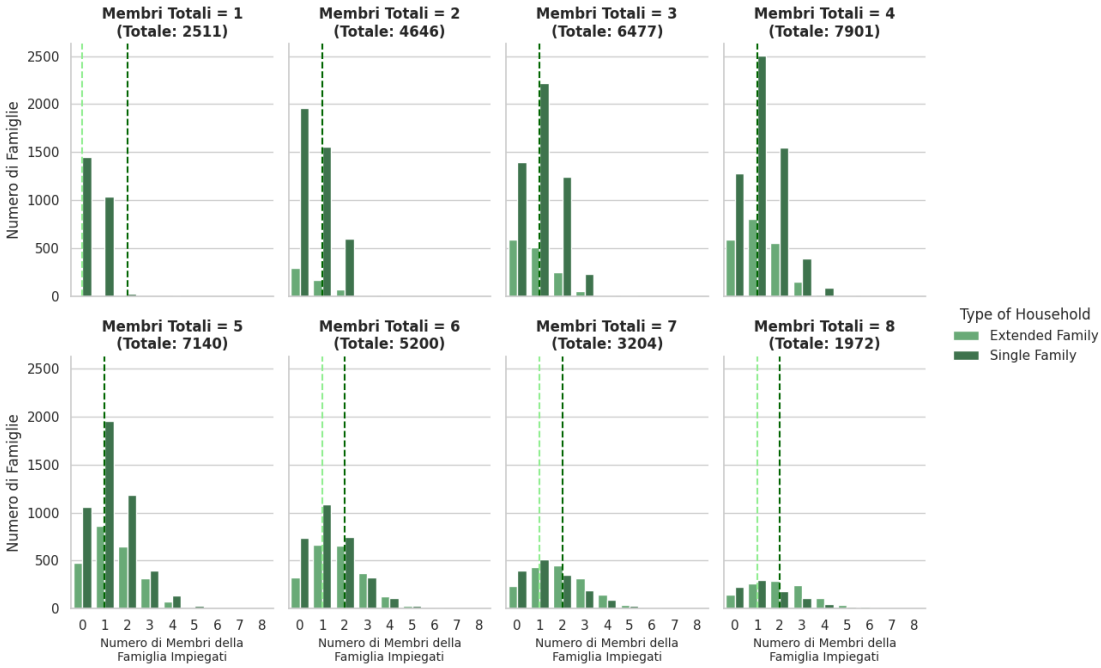

Figura 4.56: Distribuzione del numero di membri delle famiglie occupati, in relazione al numero di membri delle famiglie

Analizzando i grafici riportati in figura 4.56, risultano evidenti alcuni aspetti interessanti. Innanzitutto, osservando il grafico relativo a famiglie aventi un unico componente, si può osservare come, per famiglie che vengono etichettate come *Single Family*, il numero medio di membri occupati sia pari a due (linea media tratteggiata di color verde scuro). Ciò non è chiaramente possibile, pertanto si può considerare tale dato come un outlier, e verrà opportunamente trattato nella sezione 4.4.
È interessante, inoltre, notare come il numero di famiglie con un componente che non è occupato, è maggiore di quello delle famiglie con un singolo componente, che però è occupato. Osservando i restanti grafici, invece, risulta evidente come il numero medio di membri occupati nelle *Extended Family* (linea media tratteggiata di color verde chiaro), indipendentemente dal numero di componenti della famiglia,

sia sempre pari ad uno, il capofamiglia. Nelle *Single Family*, invece, per famiglie con numero di membri superiore a sei, le persone occupate, in media diventano due. I grafici relativi a famiglie con più di otto componenti sono stati trascurati, in quanto facenti riferimento ad un campione eccessivamente ridotto.

4.4 Data Preprocessing

La fase esplorativa, particolarmente importante per comprendere la natura dei dati con cui si lavora, è stata seguita dalla fase di *Data Preprocessing*. L'obiettivo di tale operazione è stato quello di migliorare ulteriormente la qualità dei dati in input, già di per sé significativa, al fine di aumentare la performance degli algoritmi che verranno impiegati nel corso dello studio.
Poiché si ha un duplice obiettivo di analisi, come annunciato nell'incipit del capitolo 4, la fase di preparazione dei dati è stata differenziata in base allo scopo preposto.

4.4.1 Preparazione dati per la predizione del THI mediante le variabili di spesa

La presente analisi è stata condotta in parte mediante Rapidminer [46], in parte utilizzando Google Colaboratory [48]. In figura 4.57 è riportata la pipeline della prima fase di *Data Cleaning*, effettuata con Rapidminer. Mediante l'operatore 'Read CSV' si sono importati i dati presenti nel database *Family Income and Expenditure*. Dopodiché si è proceduto con la ricerca di possibili missing values presenti e con la loro sostituzione mediante gli operatori 'Declare Missing Values' e 'Replace Missing Values'. Si è scelto di sostituire eventuali valori mancanti con il valore più frequente della variabile.

A tal punto, si è scelto di selezionare le variabili di interesse, ovvero le variabili di spesa presenti nel dataset, e nello specifico *Alcoholic Beverages Expenditure; Bread and Cereals Expenditure; Clothing, Footwear and Other Wear Expenditure; Communication Expenditure; Crop Farming and Gardening expenses; Education Expenditure; Fruit Expenditure; Housing and water Expenditure; Meat Expenditure; Medical Care Expenditure; Miscellaneous Goods and Services Expenditure; Restaurant and hotels Expenditure; Special Occasions Expenditure; Tobacco Expenditure; Total Fish and marine products Expenditure; Total Food Expenditure; Total Rice Expenditure; Transportation Expenditure; Vegetables Expenditure*, mediante l'operatore 'Select Attributes', e si è deciso di escludere tutte le altre variabili. Al fine di comprendere quali tra le sopracitate variabili sia di maggior importanza per la predizione del *Total Household Income*, si è deciso di osservare la correlazione tra tale attributo e gli attributi di spesa, generando, mediante il codice python riportato nell'Appendice A.6, la matrice di correlazione tra di esse, mediante la

correlazione di Pearson [18]. In particolare, risulta interessante capire se ci siano variabili particolarmente correlate tra loro, in modo da poterle opportunamente trattare. Si osservano, dunque, variabili con una correlazione superiore a 0.7, ed in tal caso si decide se eliminare la variabile meno correlata con il THI, oppure aggregare tra loro più variabili di spesa affini, ed impiegare la variabile aggregata nell'analisi predittiva.

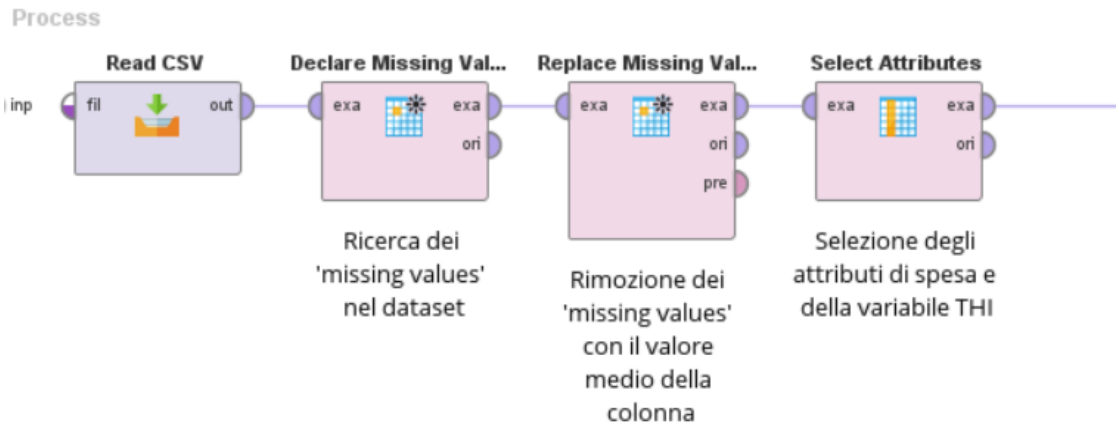

Figura 4.57: Data Cleaning

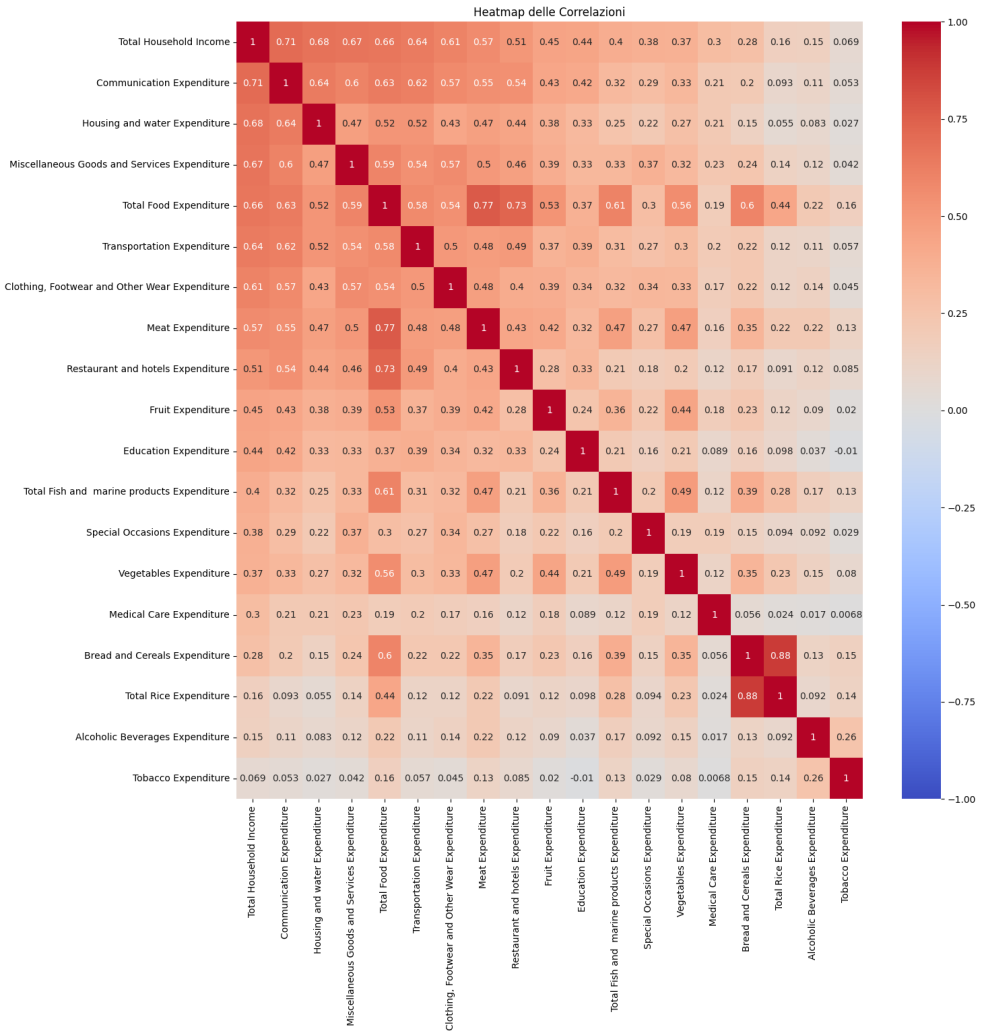

Figura 4.58: Matrice di correlazione tra le variabili di spesa ed il THI

Osservando attentamente la matrice di correlazione in figura 4.58, si può notare come alcuni attributi relativi alle spese alimentari siano altamente correlati con l'attributo *Total Food Expenditure*, come ad esempio l'attributo *Meat Expenditure*, con un valore di correlazione di 0.77. Poiché nella prima fase della nostra analisi verranno impiegati algoritmi sensibili alla multicollinearità, si decide di rimuovere l'attributo meno correlato con il *Total Household Income*, e dunque, si decide di rimuovere la colonna relativa alla *Meat Expenditure*. Analogamente, le variabili *Total Rice Expenditure* e *Bread and Cereal Expenditure* sono fortemente correlate, con un valore pari a 0.88, e pertanto, si decide di eliminare la colonna relativa alla *Total Rice Expenditure*, meno correlata con il THI.
Dall'analisi della heatmap della correlazione, risulta evidente come le variabili

Restaurant and hotels Expenditure e *Total Food Expenditure* siano altamente correlate, con un valore di correlazione pari a 0.73. Tuttavia, essendo la voce di spesa *Restaurant and hotels Expenditure* concettualmente distinta dalla spesa totale in cibo, in quanto contenente informazioni non legate unicamente ai pasti, bensì anche al pernottamento in strutture alberghiere, si decide di mantenerla nel dataset. Queste operazioni rientrano nella fase denominata *Data Reduction*.
Avendo compreso quali variabili impiegare nell'analisi predittiva del THI, si è proceduto ad osservare attentamente la loro distribuzione, partendo proprio dalla *Total Household Income*. Concentrandosi sulla figura 4.1, si è notato come la distribuzione del reddito familiare annuo sia positivamente asimmetrica, con una lunga coda a destra. Analogamente, osservando una ad una le variabili di spesa, si è notato che la distribuzione di ognuna di esse fosse del tutto simile a quella del THI, e dunque asimmetrica positiva (per non appesantire la trattazione non vengono riportati gli istogrammi delle singole variabili di spesa). Ci si è interrogati, a tal punto, su come trattare questi attributi così distribuiti, e si è deciso di applicare ad ognuna di essi una trasformazione logaritmica del tipo $ln(1 + x)$ per poter ridurre l'asimmetria positiva e rendere queste distribuzioni più simili ad una distribuzione normale. Inoltre, per poter applicare una standardizzazione *Z-Score* [51] a tali variabili, al fine di ridurre il tempo di convergenza degli algoritmi impiegati nell'analisi ed illustrati nel paragrafo 4.5, si è provveduto ad effettuare una outlier detection mediante lo *Scarto Interquartile*, IQR [52]. In particolare, si sono calcolati il primo ed il terzo quartile, $Q1$ e $Q3$, rappresentanti il 25° ed il 75° percentile, ovvero i punti al di sotto dei quali cadono il 25% ed il 75% dei dati, e si è calcolato l'indice IQR come differenza tra i due, $IQR = Q3 - Q1$. A tal punto si sono definiti i limiti, inferiore e superiore, come $LI = Q1 - 1.5 * IQR$ e $LS = Q3 + 1.5 * IQR$, e si sono scartati i valori al di fuori di essi. Questa fase, nota come *Data Transformation*, è stata effettuata grazie al codice python riportato nell'Appendice A.7.
Le figure dalla 4.59 alla 4.75 raffigurano le distribuzione delle variabili di spesa e del reddito annuo dopo l'applicazione della trasformazione logaritmica e della rimozione dei valori effettuata mediante l'IQR.

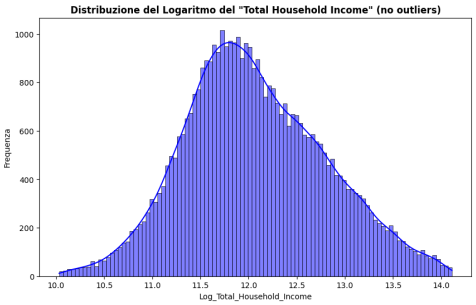

Figura 4.59: Distribuzione logaritmica del THI

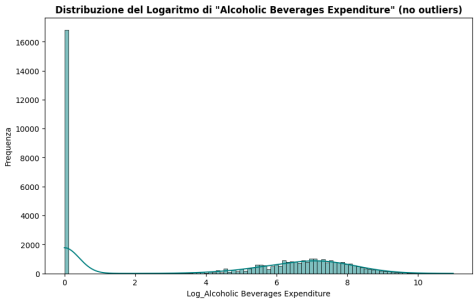

Figura 4.60: Distribuzione logaritmica delle spese in Bevande Alcoliche

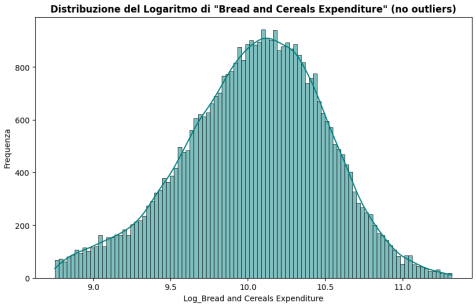

Figura 4.61: Distribuzione logaritmica delle spese in Pane e Cereali

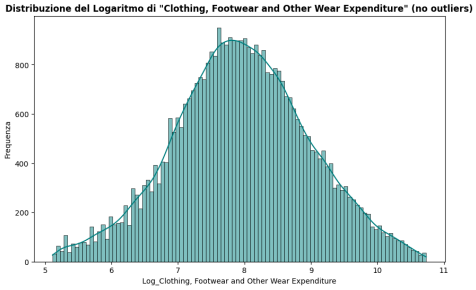

Figura 4.62: Distribuzione logaritmica delle spese in Abbigliamento, Calzature e simili

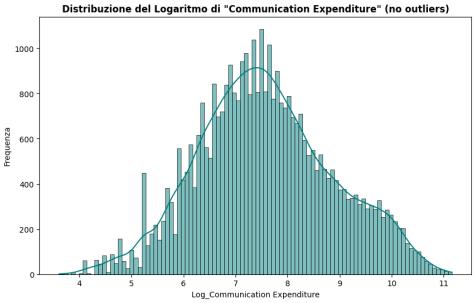

Figura 4.63: Distribuzione logaritmica delle spese in Comunicazione

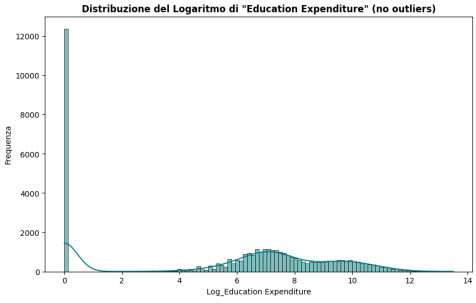

Figura 4.64: Distribuzione logaritmica delle spese in Istruzione

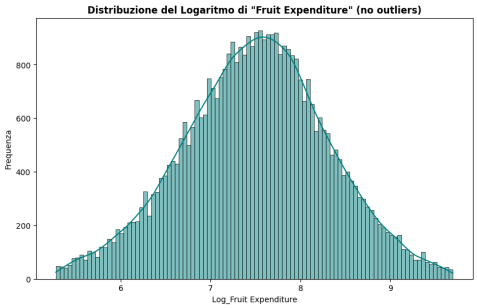

Figura 4.65: Distribuzione logaritmica delle spese in Frutta

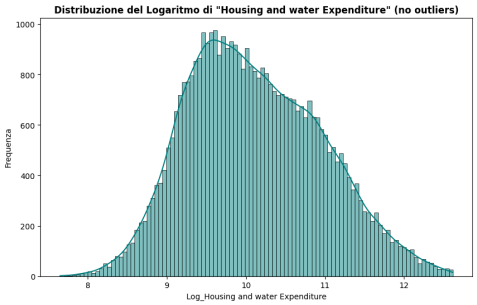

Figura 4.66: Distribuzione logaritmica delle spese per la casa ed in acqua

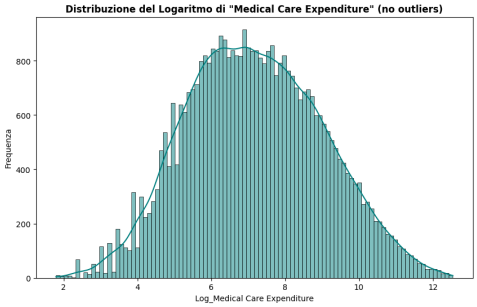

Figura 4.67: Distribuzione logaritmica delle spese in Cure Mediche

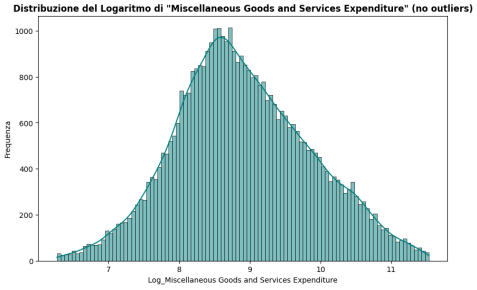

Figura 4.68: Distribuzione logaritmica delle spese in Beni e Servizi vari

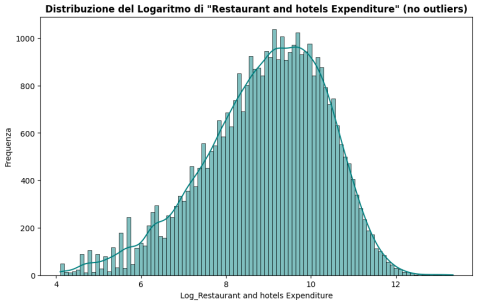

Figura 4.69: Distribuzione logaritmica delle spese in Ristorazione ed Hotel

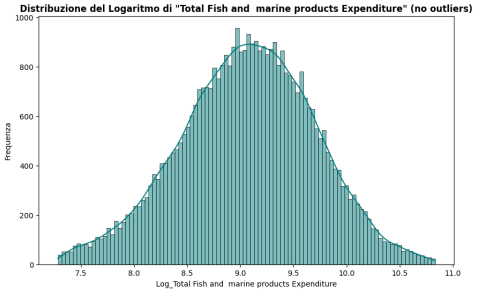

Figura 4.70: Distribuzione logaritmica della spesa totale in prodotti Ittici

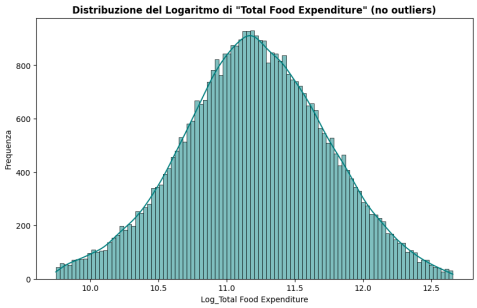

Figura 4.71: Distribuzione logaritmica della spesa totale in Cibo

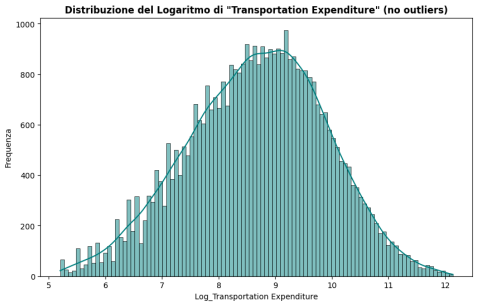

Figura 4.72: Distribuzione logaritmica delle spese in Trasporti

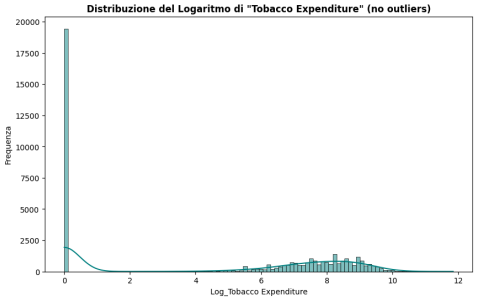

Figura 4.73: Distribuzione logaritmica delle spese in Tabacco

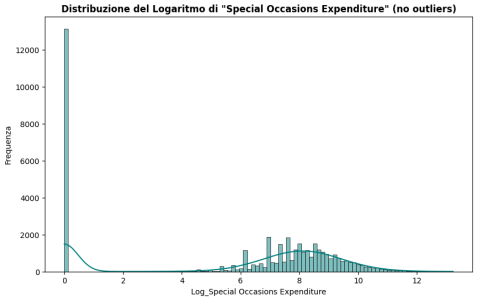

Figura 4.74: Distribuzione logaritmica delle spese per Occasioni Speciali

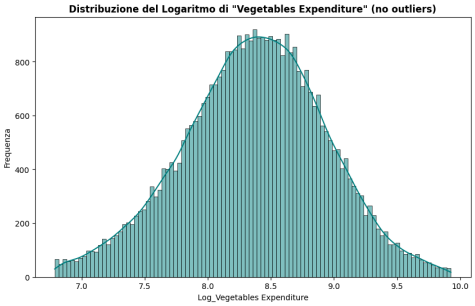

Figura 4.75: Distribuzione logaritmica delle spese in Verdure

Dai grafici precedentemente riportati si può osservare come la maggior parte delle variabili di spesa segua una distribuzione normale, con un andamento piuttosto simmetrico. Tuttavia, osservando le figure 4.60, 4.64, 4.73 e 4.74, e dunque gli attributi *Alcoholic Beverages Expenditure*, *Education Expenditure*, *Tobacco Expenditure* e *Special Occasions Expenditure*, si nota come la distribuzione sia fortemente influenzata da un gran numero di valori pari a zero. Ciò è legato al fatto che non tutte le famiglie hanno spese in tali ambiti; ma, se per le spese in alcool, tabacco ed occasioni speciali è del tutto naturale, una grande presenza di mancate spese in istruzione può denotare una condizione di povertà, in cui ci sono famiglie che probabilmente non sono in grado di investire nell'educazione dei proprio figli. Tuttavia, poiché il presente obiettivo di analisi non è quello di classificare le famiglie a rischio povertà, bensì comprendere come le variabili di spesa possano essere impiegate per predire il reddito delle famiglie filippine, si è deciso di trattare tali valori nulli come outliers e dunque rimuoverli, al fine di ridurre la numerosità dei valori delle sopracitate variabili e di rendere la loro distribuzione più simile ad una normale, in modo da poter applicare la standardizzazione Z-Score senza ulteriori problemi. Le figure dalla 4.76 alla 4.79 mostrano le distribuzioni ottenute dopo aver filtrato i dati, selezionando, per ognuno degli attributi, valori superiori ad uno. Ad eccezione della variabile *Education Expenditure* in figura 4.77, le altre variabili mostrano un andamento tendente ad una normale, specialmente per quanto riguarda le spese in bevande alcoliche.

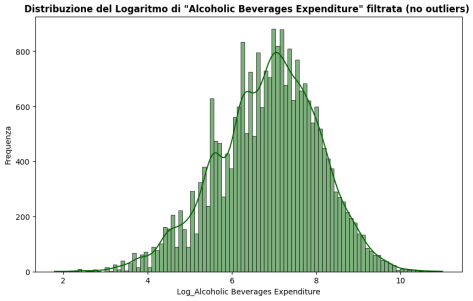

Figura 4.76: Distribuzione logaritmica delle spese in Alcolici, filtrata

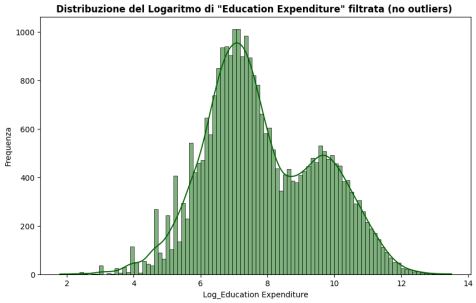

Figura 4.77: Distribuzione logaritmica delle spese in Istruzione, filtrata

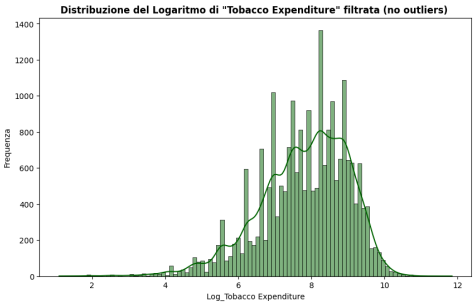

Figura 4.78: Distribuzione logaritmica delle spese in Tabacco, filtrata

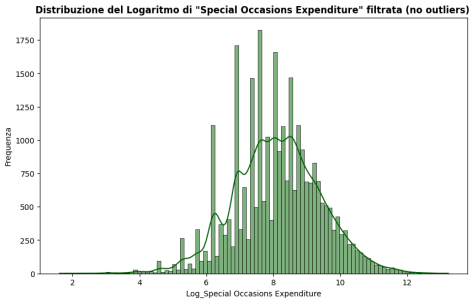

Figura 4.79: Distribuzione logaritmica delle spese per Occasioni Speciali, filtrata

A tal punto, si è proceduto a standardizzare ognuna delle precedenti variabili, mediante lo Z-Score, ovvero applicando una trasformazione del tipo $Z = \frac{X-\mu}{\sigma}$, in cui X è il generico valore della variabile che si vuole standardizzare, μ è la media della sua distribuzione e σ è la sua deviazione standard.
Per non appesantire la trattazione non vengono riportati i grafici delle distribuzioni delle variabili dopo aver applicato la standardizzazione Z-Score in quanto analoghi ai grafici precedenti, ma centrati sullo zero.

Infine, si è creato un nuovo file CSV, denominato *Preprocessed_1_Family_ _Income_and_Expenditure* in cui sono state selezionate unicamente le variabili di spesa, opportunamente rinominate sostituendo gli spazi, i due punti e le virgole con il carattere underscore, e la variabile da predire, il THI. Tutto ciò, è stato realizzato mediante il codice python riportato nell'Appendice A.8.

Dunque, i dati cosi rielaborati sono pronti per essere utilizzati dagli algoritmi di ML.

4.4.2 Preparazione dati per la predizione della fascia di reddito delle famiglie filippine

La presente analisi è stata svolta interamente mediante il notebook jupiter *Google Colaboratory*. L'obiettivo di questa fase è preparare i dati per poter utilizzare efficacemente gli algoritmi di classificazione per poter predire la fascia di reddito di appartenenza delle famiglie filippine.

In un primo momento si è provveduto a selezionare le variabili di interesse, e dunque si è deciso di escludere le variabili di spesa, impiegate nell'analisi precedente. Si è deciso, inoltre, di rimuovere altri due attributi, ritenuti non rilevanti per l'analisi in essere, ovvero *Main Source of Income* e *Imputed House Rental Value*. Partendo dai risultati ottenuti in fase di Data Exploration, si è deciso, inoltre, di rimuovere

le righe corrispondenti al valore pari a 2 dell'attributo *Agricultural Household Indicator*, in quanto reputato outlier. Sono state dunque ritenute importanti ai fini dello studio le variabili di seguito elencate:

- *Agricultural Household indicator*;
- *Crop Farming and Gardening expenses*;
- *Electricity*;
- *House Age*;
- *House Floor Area*;
- *Household Head Age*;
- *Household Head Class of Worker*;
- *Household Head Highest Grade Completed*;
- *Household Head Job or Business Indicator*;
- *Household Head Marital Status*;
- *Household Head Occupation*;
- *Household Head Sex*;
- *Main Source of Water Supply*;
- *Members with age 5 - 17 years old*;
- *Members with age less than 5 year old*;
- *Number of Airconditioner*;
- *Number of bedrooms*;
- *Number of Car, Jeep, Van*;
- *Number of CD/VCD/DVD*;
- *Number of Cellular phone*;
- *Number of Component/Stereo set*;
- *Number of Landline/wireless telephones*;
- *Number of Motorcycle/Tricycle*;

- *Number of Motorized Banca*;
- *Number of Personal Computer*;
- *Number of Refrigerator/Freezer*;
- *Number of Stove with Oven/Gas Range*;
- *Number of Television*;
- *Number of Washing Machine*;
- *Tenure Status*;
- *Toilet Facilities*;
- *Total Household Income*;
- *Total Number of Family members*;
- *Total number of family members employed*;
- *Type of Building/House*;
- *Type of Household*;
- *Type of Roof*;
- *Type of Walls.*

Prima di eseguire qualsiasi tipo di operazione sulle variabili in gioco, si è effettuata una ricerca dei missing values nelle colonne del dataset, e sono stati trovati valori mancanti nelle variabili *Household Head Class of Worker* e *Household Head Occupation*. Essi sono stati rimpiazzati con il valore più frequente della colonna, e dunque con il valore modale.

A tal punto, sono state separate le variabili categoriche da quelle numeriche, e si è lavorato separatamente su di esse. Per quanto riguarda le variabili numeriche *Household Head Age, House Floor Area* e *House Age*, sono state trasformate mediante la trasformazione logaritmica $ln(1 + x)$, in modo da rendere la loro distribuzione prossima ad una gaussiana. Successivamente, sono stati rimossi gli outliers con il metodo dello scarto interquartile, e dunque con il calcolo dell' indice IQR, come effettuato per le variabili numeriche rielaborate nel paragrafo 4.4.1. Infine, tramite la standardizzazione Z-Score, anch'essa presentata nel paragrafo 4.4.1, sono stati normalizzati gli attributi di interesse. La loro distribuzione è illustrata nella figura 4.80.

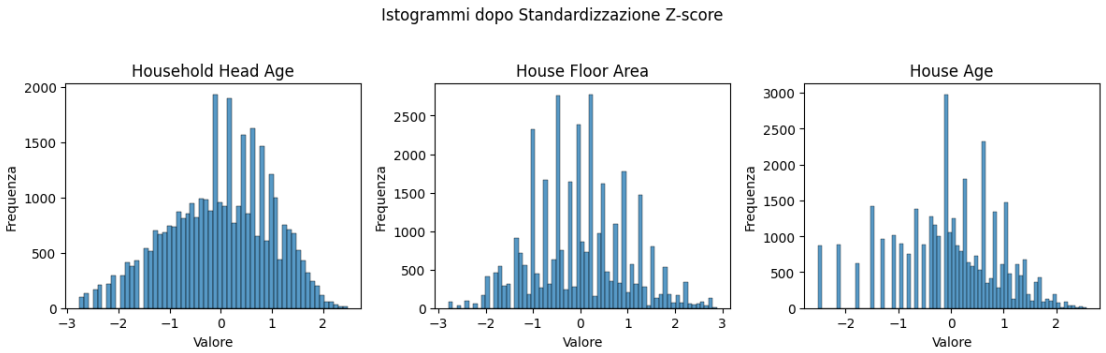

Figura 4.80: Distribuzione delle variabili *Household Head Age, House Floor Area* e *House Age* normalizzate

Si sono dunque, analizzate le variabili categoriche. Si è deciso di osservare il numero di valori unici che ognuno degli attributi categorici potesse assumere, in modo da capire quale tipologia di codifica applicare. Si è notato come gli attributi *Household Head Highest Grade Completed* e *Household Head Occupation*, potessero assumere un numero particolarmente elevato di valori distinti, ed in particolare, 46 per il primo e 378 per il secondo. Si è provveduto quindi, ad effettuare un meticoloso lavoro di selezione manuale mirata a ridurre la numerosità di tali valori, andando a generare due nuovi attributi, opportunamente inclusi nel dataset, e denominati *Household Head Highest Grade Completed (Categories)* e *Household Head Occupation (Categories)*. I 46 valori relativi al grado di istruzione del capofamiglia sono state racchiusi nelle cinque categorie di seguito elencate:

- *Elementary Education or below*;
- *Secondary Education*;
- *Technical Education and Vocational Training*;
- *Higher Education (College/University)*;
- *Masterś Degree and Programs*;

mentre i 378 valori relativi all'occupazione del capofamiglia sono stati riassunti nelle undici categorie di lavoro di seguito elencate:

- *Agriculture & Livestock Farming*;
- *Building & Construction*;
- *Commerce & Sales*;

- *Education & Training*;
- *Entertainment, Art, Music, Sport & Fashion*;
- *Health Care & Social Services*;
- *Management & Administration*;
- *Personal Services, Law Enforcement, Catering & Hospitality*;
- *Production, Handicraft & Manufacturing*;
- *Science, Engineering & ICT*;
- *Transportation & Logistics*.

Per non appesantire la trattazione, non vengono riportati i valori divisi nelle corrispondenti categorie. Tuttavia, sono visibili nel codice python riportato nella Appendice A.9.

A tal punto, si è provveduto a trasformare la variabile numerica di interesse *Total Household Income*, in una variabile categorica, opportunamente rinominata *Income Category*, mediante la funzione **pd.cut** della libreria **pandas** di python. Si sono, quindi, suddivisi i valori del reddito in tre categorie, prendendo come riferimento il valore della soglia di povertà indicato dal Philippine Statistics Authority (PSA) per il 2015, pari a 22747 PHP annui [53]. Le categorie create sono, dunque:

- *Under Poverty Treshold*: comprende tutti i valori del reddito inferiori alla soglia di povertà;
- *1-2x Poverty Treshold*: comprende tutti i valori del reddito compresi tra una e due volte la soglia di povertà;
- *Over 2x Poverty Treshold*: comprende tutti i valori del reddito superiori a due volte la soglia di povertà.

Si è poi deciso di osservare la distribuzione dei valori nelle categorie per capire se fossero o meno bilanciate. Si è notato che le tre categorie generate fossero notevolmente sbilanciate, ed in particolare la categoria più numerosa, *Over 2x Poverty Treshold*, contenesse il 97.23% dei dati, mentre la categoria *1-2x Poverty Treshold* il 2.56% e la categoria *Under Poverty Treshold* solamente lo 0.21%.

A tal punto, si è deciso di procedere percorrendo due strade distinte. Nel primo caso si è scelto di non attuare una strategia di bilanciamento del dataset, andando a generare un dataset sbilanciato, rinominato *Preprocessed_2_Family_Income_and_ _Expenditure_no_sampling*. L'obiettivo di tale operazione è quello di valutare se, e come, cambiano le performance degli algoritmi di Classificazione al variare del

bilanciamento dei dati in input. I risultati di questa analisi sono presentati nel paragrafo 4.6.3.

Nel secondo caso, invece, si è proceduto ad attuare una duplice strategia di sampling, al fine di bilanciare correttamente il dataset di input. In particolare, si è attuata una strategia di Undersampling per la categoria maggiormente rappresentata ed una strategia di Oversampling per le altre due categorie; la strategia SMOTE. È stata scelta la strategia SMOTE in quanto, rispetto all'oversampling randomico, essa consente di evitare il problema dell'overfitting [54]. Entrando ancor di più nel dettaglio si è scelto di ridurre della metà la dimensione della categoria *Over 2x Poverty Treshold*, e di portare le altre due categorie alla nuova dimensione della suddetta categoria, ottenendo 17178 record per ognuna di esse.

Per poter applicare le tecniche di sampling, tuttavia, è stato necessario codificare le variabili presenti nel dataset. Si è scelto di applicare una codifica binaria per gli attributi binari appunto, mediante il Label encoder, ed una codifica One Hot Encoding per le altre variabili categoriche, mediante le funzioni **labelencoder** e **pd.getdummies**, rispettivamente. La funzione pd.getdummies è stata preferita a quella OneHotEncoder, in quanto di più semplice applicazione. La codifica delle variabili categoriche, mirata a rappresentare con valori numerici le varie categorie, è un passaggio fondamentale per poter utilizzare tali variabili con modelli di Machine Learning che richiedono variabili numeriche in input [55]. Una delle problematiche della codifica one hot, tuttavia, è che essa crea una nuova variabile per ognuna delle categorie dell'attributo originale, il che può comportare un significativo aumento delle dimensioni del dataset, oltre che aumentare la sparsità nei dati. Dopo aver applicato tale codifica, infatti, gli attributi caratterizzanti il dataset sono diventati 108, il che potrebbe comportare una riduzione della performance degli algoritmi impiegati; ciononostante si è deciso di mantenerli.

Con la creazione delle nuove variabili, e in seguito alla generazione dei nuovi record mediante le tecniche di sampling precedentemente illustrate, si è provveduto a generare un nuovo dataset contenente gli attributi preprocessati, denominato *Preprocessed_2_Family_Income_and_Expenditure*, pronto per essere utilizzato nella successiva fase di Data Mining.

Il codice python impiegato per realizzare quanto descritto nel presente paragrafo è riportato nell'Appendice A.9.

4.5 Data Mining

Successivamente alla fase di Data Preprocessing, che, come detto nell'incipit del paragrafo 4.4, è stata fondamentale per la preparazione dei dati oggetto dell'analisi, si è proceduto con la fase di Data Mining.

Obiettivo del presente paragrafo è quello di illustrare le tecniche di Data Mining adottate per analizzare e predire il valore della variabile *Total Household Income* e la classe di appartenenza delle famiglie Filippine, per comprendere il rischio di povertà che esse corrono. Tali analisi rivestono notevole importanza nella comprensione delle dinamiche socio-economiche di una nazione in grande sviluppo come le Filippine, fornendo insights interessanti per l'attuazione di tecniche di previsione dello stato di povertà delle famiglie, che possono essere impiegate per la formulazione di politiche efficaci mirate al combattere la povertà. Le metodologie applicate nel corso del presente studio sono molteplici, ed opportunamente scelte in base all'obiettivo prefissato.

4.5.1 Predizione del THI mediante le variabili di spesa

L'analisi di Data Mining, effettuata sul dataset *Preprocessed_1_Family_ _Income_and_Expenditure* creato appositamente in fase di Preprocessing, è una analisi predittiva effettuata mediante algoritmi di Regressione. In particolare, avendo come obiettivo quello di predire il valore di una variabile continua, partendo da attributi anch'essi continui, si è deciso che i migliori strumenti da utilizzare fossero modelli di Regressione, altrimenti noti, per l'appunto, come *Continuous Value Classifiers* [22].

A seguito di una attenta ricerca, si è deciso di impiegare tre specifici modelli con caratteristiche differenti tra loro, in modo da poter effettuare un confronto completo tra metodologie diverse. Si è deciso dunque, di testare l'efficacia dei modelli: **LinearRegressor** [56], **RandomForestRegressor** [26], **LGBMRegressor** [57], ampiamente descritti nel paragrafo 2.3.

L'analisi è stata svolta interamente tramite *Google Colaboratory*. La libreria utilizzata per far funzionare i modelli di regressione è *Scikit-Learn (sklearn)* [56], ed il codice python prodotto per i fini sopracitati è quello riportato nell'Appendice A.10. La versione del codice presentata è l'ultima versione elaborata, contente tutte le modifiche atte a migliorare le performance degli algoritmi testati. In una fase primordiale del codice, infatti, non era stata implementata la porzione contente l'ottimizzazione dei parametri del RandomForestRegressor e del LGBMRegressor, effettuata mediante lo stimatore **RandomizedSearchCV**. Tale stimatore, a differenza del 'GridSearchCV', il quale testa il miglior set di parametri tra tutte le combinazioni possibili di valori indicati, effettua la selezione del miglior set di parametri sulla base di un gruppo di valori selezionato in maniera randomica. In particolare, la selezione dei parametri avviene tramite il calcolo dell'errore quadratico medio (MSE) [56]; viene selezionata la combinazione di valori che restituisce il minor MSE. Il motivo per cui si è escluso il 'GridSearchCV' è che il tempo richiesto per il completamento della sua funzione era eccessivamente elevato e ciò comportava una disconnessione del programma in esecuzione ed un mancato ottenimento dei

risultati. Dunque, in fase di fine tuning, si è deciso di selezionare una serie di valori da testare per ogni attributo, selezionando opportunamente tali valori, partendo da quelli di default indicati nelle librerie dei modelli selezionati [26], [57].

I valori ottimali trovati, sono, dunque, per i parametri del **RandomForestRegressor**: {'n_estimators': 50, 'min_samples_split': 10, 'min_samples_leaf': 1, 'max_features': 'sqrt', 'max_depth': None}, mentre per il **LighGBMRegressor**: {'subsample': 0.9, 'reg_lambda': 0.1, 'reg_alpha': 0, 'num_leaves': 60, 'n_estimators': 190, 'min_child_samples': 30, 'max_depth': 10, 'learning_rate': 0.05, 'colsample_bytree': 0.6}

Una volta individuati i parametri ottimali per ognuno dei modelli di regressione selezionati, e dunque aver costruito i modelli da utilizzare, si è provveduto ad addestrare ognuno di essi sul dataset di training creato nella fase iniziale del codice. La dimensione scelta per il dataset di test è del 20% della dimensione del dataset di partenza. L'obiettivo di tale operazione è quello di evitare l'overfitting, e consentire dunque una adeguata analisi predittiva sul dataset di test.

4.5.2 Predizione della fascia di reddito delle famiglie filippine

L'analisi di Data Mining, effettuata sul dataset *Preprocessed_2_Family_ _Income_and_Expenditure* creato appositamente in fase di Preprocessing illustrata nel paragrafo 4.4.2, è una analisi predittiva effettuata mediante algoritmi di Classificazione. In particolare, avendo come obiettivo quello di predire la fascia di reddito di appartenenza delle famiglie filippine, si è deciso che i migliori strumenti da utilizzare fossero, per l'appunto, modelli di Classificazione.

A seguito di una attenta ricerca, si è scelto di impiegare tre specifici modelli con caratteristiche differenti tra loro, in modo da poter effettuare un confronto completo tra metodologie diverse. Si è deciso dunque, di testare l'efficacia dei modelli: *Random Forest*, *Support Vector Machine (RBF)* e *LightGBM*, ampiamente descritti, rispettivamente, nei paragrafi 2.3.2, 2.3.4 e 2.3.3.

Così come fatto per il primo obiettivo di analisi, presentato nel paragrafo 4.5.1, anche per sviluppare il presente obiettivo si è operato interamente sull'ambiente di lavoro offerto da *Google Colaboratory*.

I modelli specifici selezionati sono stati dunque, il **RandomForestClassifier** [27] ed il **SVC** [34] della libreria *Scikit-Learn (sklearn)*, e il **LGBMClassifier** [58] della libreria *lightgbm*; il codice python prodotto per effettuare l'analisi è quello riportato nell'Appendice A.11, e viene di seguito spiegato.

In primis, dopo aver caricato il dataset in questione, si è definita la variabile di interesse da predire, la *Income Category*, generata in fase di preprocessing. Per evitare problematiche con il funzionamento del modello LGBMClassifier, sensibile alle etichette delle numerose variabili in gioco, si è poi provveduto a sostituire i

caratteri non supportati dal modello come virgole, spazi bianchi, parentesi graffe e due punti con il carattere underscore ('_'). Fatto ciò, si è effettuata la divisione del dataset in set di addestramento e in set di test, scegliendo come dimensione del set di test il 20% della dimensione del dataset originario. Per fare in modo che la divisione casuale del dataset sia riproducibile e che generi sempre il medesimo set di addestramento e test è stato impostato il parametro *random_state* pari a 42, dove il valore usato è stato scelto in quanto convenzionalmente usato in letteratura, ma non ha un vero e proprio significato matematico.

In seguito, si sono definiti i modelli da utilizzare, insieme ad un elenco di iper-parametri da testare per ognuno di essi, riportato di seguito:

- **RandomForestClassifier**:
  - '*n_estimators*': [100, 200, 300];
  - '*max_features*': ['sqrt', 'log2'];
  - '*max_depth*': [10, 20, 30];
  - '*min_samples_split*': [2, 5, 10];
  - '*min_samples_leaf*': [1, 2, 4];
  - '*bootstrap*':[True, False];
- **SVC**:
  - 'C': [1, 10, 50];
  - 'γ': [0.01, 0.001];
  - '*kernel*': ['rbf'];
- **LGBMClassifier**:
  - '*n_estimators*': [100, 200, 300];
  - '*learning_rate*': [0.01, 0.1, 0.5];
  - '*max_depth*': [3, 5, 7];
  - '*num_leaves*': [31, 50, 70];
  - '*min_child_samples*': [20, 30, 40];
  - '*subsample*': [0.8, 1.0];
  - '*colsample_bytree*': [0.8, 1.0];

Il significato di ognuno di questi parametri è stato approfondito nei paragrafi 2.3.2, 2.3.4, 2.3.3; pertanto, per non appesantire la trattazione, non verranno rispiegati.

Giunti a tal punto, si procede, mediante la funzione **RandomizedSearchCV**, alla ricerca dei migliori iper-parametri, tra quelli proposti, per ognuno dei modelli in questione e per lo specifico dataset che si è utilizzato. Anch'essa è parte della libreria *sklearn*. Come accennato nel paragrafo 4.5.1, la funzione RandomizedSearcCV opera una selezione randomica di un numero fissato di combinazioni di iper-parametri tra quelli definiti, e, per ognuna di esse il modello viene addestrato e valutato [59]. In particolare, la valutazione avviene mediante la cross-validation, anch'essa esaustivamente spiegata nel paragrafo 2.4. La combinazione di iper-parametri che consente di ottenere i migliori risultati in fase predittiva è scelta sulla base di metriche ben precise, che possono essere esplicitate o meno. In questo caso specifico si è scelto di non esplicitarla e di usare la metrica di default dei modelli di classificazione, l'accuratezza.

Sulla base dei modelli generati con le migliori combinazioni di iper-parametri, si è proceduto alla generazione delle previsioni delle classi dell'attributo *Income Category*, ed al calcolo delle probabilità stimate associate alle previsioni.

Infine, addestrati e testati i modelli, si è provveduto al calcolo e alla stampa delle principali metriche di valutazione selezionate per ogni modello, ovvero:

- *Accuracy*;
- *Precision*;
- *Recall*;
- *F1 Score*;

e si è raffigurata, sempre per ognuno di essi, la matrice di confusione. I risultati ottenuti sono illustrati e discussi nel paragrafo 4.6.3.

4.6 Performance Evaluation

4.6.1 Performance Evaluation dei modelli impiegati per predire THI

Una volta addestrati i tre modelli, come visto nel paragrafo 4.5.1, si procede alla valutazione delle loro performance, eseguendoli sul dataset di test, mediante il calcolo del RMSE e del R^2.

In tabella 4.1 sono riportati i migliori risultati ottenuti. Al fine di osservare le performance dei modelli di regressione, si è ripetuto l'intero procedimento numerose volte, ottenendo però, risultati pressoché simili in ogni esecuzione.

I grafici a dispersione riportati nelle figure dalla 4.81 alla 4.83 mostrano le capacità predittive di ogni modello di regressione, e dunque, quanto ognuno di essi

Modello	RMSE	R^2
Linear Regressor	0.3065	0.8454
Random Forest Regressor	0.2849	0.8674
LightGBM Regressor	0.2781	0.8745

Tabella 4.1: Valutazione delle prestazioni dei modelli di Regressione

sia in grado di predire il valore del logaritmo del THI. Sull'asse delle ordinate sono riportati i valori predetti dal modello in questione, mentre sull'asse delle ascisse sono riportati i valori reali.

Nonostante i valori del RMSE e del R^2 siano simili tra i vari modelli, il LightGBM presenta risultati migliori per entrambe le metriche di valutazione. Si può dunque affermare che per il dataset in questione, il LightGBM sia il modello che maggiormente è in grado di predire il valore del reddito familiare annuo partendo dalle variabili di spesa, fornendo dunque un esito migliore, seguito però a ruota dal Random Forest, il quale è solamente di poco inferiore.

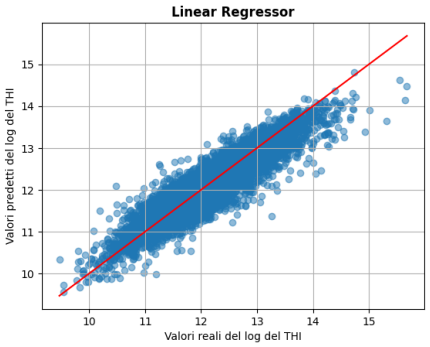

Figura 4.81: Scatterplot raffigurante i valori predetti ed i valori reali del THI mediante *Linear Regressor*

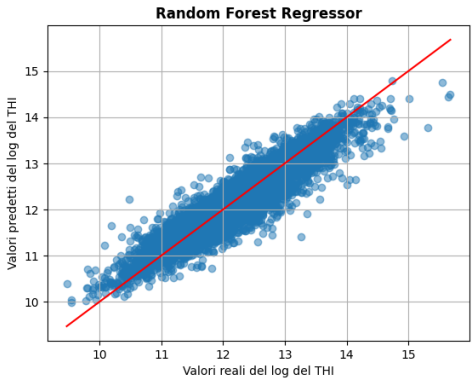

Figura 4.82: Scatterplot raffigurante i valori predetti ed i valori reali del THI mediante *Random Forest Regressor*

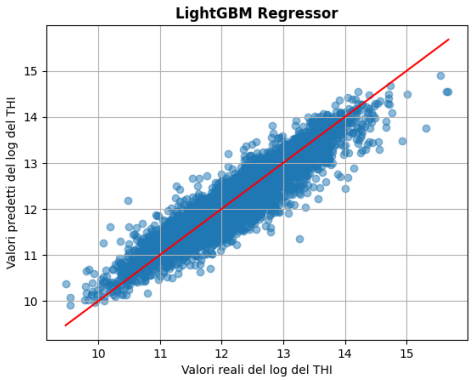

Figura 4.83: Scatterplot raffigurante i valori predetti ed i valori reali del THI mediante *LightGBM Regressor*

4.6.2 Performance Evaluation dei modelli impiegati per predire la fascia di reddito delle famiglie filippine - senza sampling dei dati

Nel presente paragrafo si illustrano i risultati ottenuti dalle analisi mostrate nel paragrafo 4.5.2, mediante l'utilizzo del dataset fortemente sbilanciato dal nome *Preprocessed_2_Family_Income_and_Expenditure_no_sampling*.

Prima di esporre i risultati, si vuole ricordare la distribuzione dei dati nelle tre categorie dell'attributo categorico *Income Category*: 97.23% nella categoria *Over 2x Poverty Treshold*, 2.56% nella categoria *1-2x Poverty Treshold* ed il rimanente 0.21% nella categoria *Under Poverty Treshold*, indicate nelle immagini seguenti rispettivamente con i valori 2, 1 e 0.

Le metriche che si è provveduto ad analizzare sono quelle principali per un modello di Classificazione, *Precision*, *Recall* e *F1 Score*, calcolate classe per classe, e l'*Accuracy*, valore cumulato, tutte ampiamente discusse nel paragrafo 2.4.2. I loro valori, divisi per modello utilizzato, sono riportati in tabella 4.2; i loro valori medi, divisi per modello, sono, invece, riportati in figura 4.3. Nelle figure 4.84 e 4.85 sono riportate le matrici di confusione ottenute dai 2 modelli in questione.

Classe	Random Forest			LightGBM		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
0	0.0000	0.0000	0.0000	0.8000	0.3636	0.5000
1	0.5714	0.1530	0.2414	0.6154	0.3060	0.4088
2	0.9772	0.9978	0.9978	0.9818	0.9958	0.9887

Tabella 4.2: Risultati delle metriche di valutazione del Random Forest e del LightGBM, per ogni classe, senza sampling dei dati in input

Modello	CV Score	Accuracy	Precision	Recall	F1 Score
Random Forest	0.9747	0.9744	0.5162	0.3836	0.4096
LightGBM	0.9758	0.9769	0.7991	0.5551	0.6325

Tabella 4.3: Risultati delle metriche di valutazione del Random Forest e del LightGBM, in media, senza sampling dei dati in input

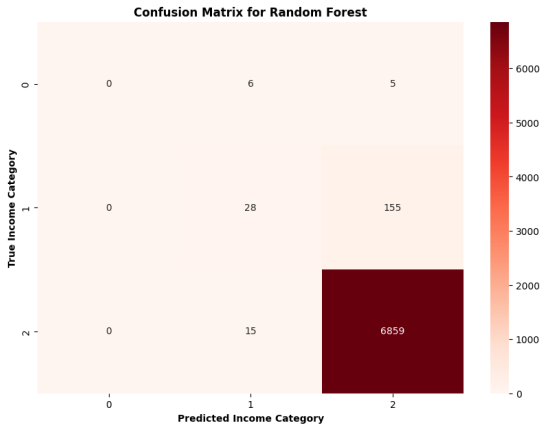

Figura 4.84: Matrice di confusione ottenuta dal *RandomForestClassifier*, senza sampling dei dai in input

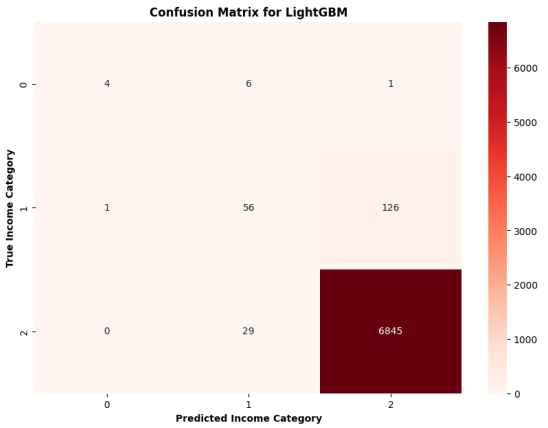

Figura 4.85: Matrice di confusione ottenuta dal *LGBMClassifier*, senza sampling dei dai in input

Dall'analisi delle tabelle 4.2 e 4.3, e dalle matrici di confusione 4.84 e 4.85, sono numerosi gli aspetti interessanti che saltano all'occhio, e sono estremamente coerenti con quanto ci si aspettava.

Il primo aspetto che risulta evidente è la grande differenza nei valori delle varie metriche tra la classe 2, la classe maggioritaria nel dataset, e le altre due classi.

Se per la prima, infatti, si riscontrano valori eccellenti, per entrambi i modelli utilizzati, di Precision, Recall e F1 Score, per le classi 0 ed 1 si ottengono valori pessimi, soprattutto per quanto riguarda la Recall. Inoltre, osservando i risultati del Random Forest Classifier, essi sono inferiori rispetto a quelli del LightGBM Classifier, già di per sé molto bassi. É interessante notare come, per il Random Forest, per quanto riguarda la classe 0, i valori delle tre metriche siano nulli, e tali valori siano accompagnati, in fase di esecuzione del codice, dal warning riportato in figura 4.86.

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning:
Precision is ill-defined and being set to 0.0 in labels with no
predicted samples.
Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

Figura 4.86: Warning relativo al calcolo delle metriche di Precision, Recall e F1 Score per la classe 0 per il modello Random Forest

Andando ad analizzare la libreria *scikit-learn* di python si è scoperto che tale warning, relativo all'impossibilità di calcolare le tre metriche richieste, è molto comune nel caso di dataset fortemente sbilanciati, come quello in questione. Diventa dunque, indispensabile, al fine di ottenere dei risultati consistenti, bilanciare correttamente i dati di input dei modelli.

Un ultimo aspetto che merita di essere analizzato, in quanto particolarmente interessante, è relativo al valor medio delle metriche calcolate. Se si osservasse unicamente l'Accuracy, infatti, si correrebbe il rischio di affermare che entrambi i modelli generati stiano predicendo le classi in questione in maniera eccellente. In realtà, però, si sta predicendo correttamente soltanto la classe maggioritaria, mentre si stanno ottenendo dei risultati pessimi per le due classi di minoranza. É dunque, fondamentale, quando si ha a che fare con dataset sbilanciati, osservare anche il valore delle altre metriche, quali la Precision e la Recall. Così facendo si andrebbe a notare il problema appena descritto, e diventerebbe immediato comprendere la necessità di un corretto bilanciamento dei dati prima di poter utilizzare algorimti di Machine Learning.

4.6.3 Performance Evaluation dei modelli impiegati per predire la fascia di reddito delle famiglie filippine - con sampling dei dati

Nel presente paragrafo si illustrano i risultati ottenuti dalle analisi mostrate nel paragrafo 4.5.2, mediante l'utilizzo del dataset dal nome *Preprocessed_2_Family_ _Income_and_Expenditure*, in cui è stata applicata una meticolosa duplice strategia di sampling.

Nell'eseguire il codice, soprattutto nelle fasi iniziali, si sono riscontrate numerose criticità, legate prevalentemente al modello SVC. A causa dell'elevato numero di records del dataset in questione, infatti, il modello SVC è risultato particolarmente inadatto. Esso, oltre a richiedere diverse ore per la ricerca dei suoi migliori iper-parametri, nonché per essere addestrato, è risultato essere la causa di continue interruzioni inaspettate nell'esecuzione del codice, come mostrato in figura 4.87.

Figura 4.87: Interruzione inaspettata dell'esecuzione del codice

A causa dei continui errori in fase di esecuzione del modello, e dopo numerosi tentativi, si è deciso di rimuovere il modello SVC dall'analisi, e di valutare unicamente le performance del **RandomForestClassifier** e del **LGBMClassifier**.

In tabella 4.4 sono riportati i risultati, separati per classe, delle metriche di valutazione utilizzate, mentre in tabella 4.5 sono riportati i risultati medi delle stesse metriche di cui sopra. Per ulteriore chiarezza, le classi presentate in tabella 4.4 indicano i seguenti range:

- *Classe 0*: indica la fascia di reddito che contiene valori al di sotto della soglia di povertà;
- *Classe 1*: indica la fascia di reddito che comprende valori tra una e due volte la soglia di povertà;
- *Classe 2*: indica la fascia di reddito che contiene valori superiori a due volte la soglia di povertà.

I risultati di seguito elencati sono quelli ottenuti addestrando i modelli con i migliori iper-parametri individuati dalla funzione **RandomizedSearchCV**, ovvero:

- **RandomForestClassifier**: 'n_estimators': 100, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_features': 'sqrt', 'max_depth': 30, 'bootstrap': True;

- **LGBMClassifier**: 'subsample': 1.0, 'num_leaves': 50, 'n_estimators': 100, 'min_child_samples': 20, 'max_depth': 7, 'learning_rate': 0.5, 'colsample_bytree': 0.8.

Classe	Random Forest			LightGBM		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
0	0.9931	0.9986	0.9958	0.9991	0.9988	0.9990
1	0.9743	0.9824	0.9783	0.9804	0.9832	0.9818
2	0.9891	0.9756	0.9823	0.9837	0.9811	0.9824

Tabella 4.4: Risultati delle metriche di valutazione del Random Forest e del LightGBM, per ogni classe

Modello	CV Score	Accuracy	Precision	Recall	F1 Score
Random Forest	0.9826	0.9855	0.9855	0.9855	0.9855
LightGBM	0.9855	0.9878	0.9877	0.9877	0.9877

Tabella 4.5: Risultati delle metriche di valutazione del Random Forest e del LightGBM

I valori ottenuti per entrambi i modelli sono particolarmente interessanti. Osservandole nel loro insieme, le varie metriche suggeriscono come entrambi i modelli in questione stiano mostrando eccellenti performance in termini di predizione. In particolare, analizzandole voce per voce singolarmente, possiamo affermare che la quasi totalità delle previsioni sia corretta (si veda l'*Accuracy*). É opportuno sottolineare, come, nel caso in questione, l'accuracy sia una metrica affidabile, in quanto il dataset utilizzato è perfettamente bilanciato, come visto nel paragrafo 4.4.2. In aggiunta, osservando i valori della *Precision* per ogni classe, risulta evidente come anche l'affidabilità nelle previsioni positive sia elevata, e dunque che entrambi i modelli, nel momento in cui hanno predetto i valori di una classe, lo hanno fatto correttamente nella quasi totalità delle volte. Proseguendo, concentrandosi sui valori del *Recall* per ogni classe, anche l'identificazione dei veri positivi è stata ottima, e quindi, nuovamente, entrambi i modelli sono stati in grado di identificare quasi tutti i veri positivi. Ne deriva, dunque, che anche l'*F1-score*, media armonica tra questi ultimi due, sia un valore significativamente positivo, stante ad indicare la bontà dei modelli generati.

Per poter ottenere un ulteriore riscontro visivo dei risultati dei modelli ottenuti, si è generata la matrice di confusione per ognuno di essi, riportata in figura 4.88 e figura 4.89.

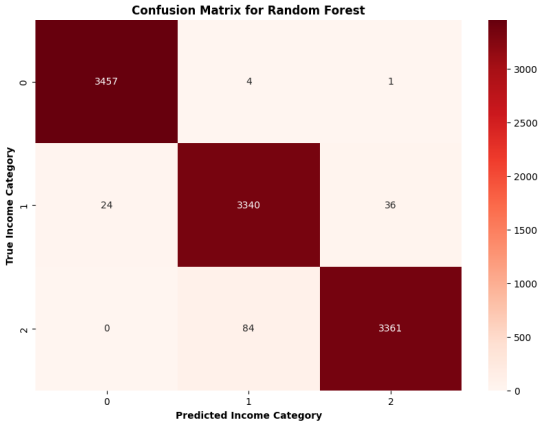

Figura 4.88: Matrice di confusione ottenuta dal *RandomForestClassifier*

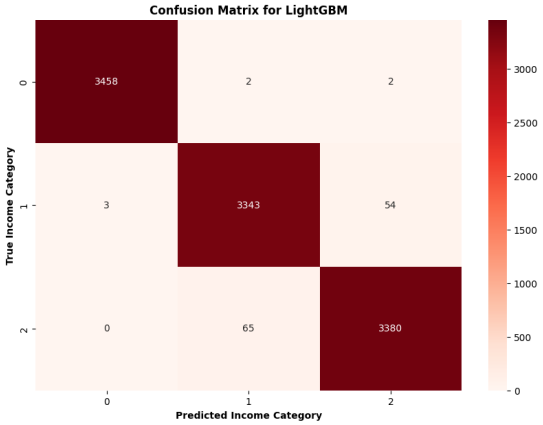

Figura 4.89: Matrice di confusione ottenuta dal *LGBMClassifier*

Le due matrici di confusione confermano quanto detto in precedenza, ed illustrano l'efficacia di entrambi i modelli nella predizione delle fasce di reddito delle famiglie filippine.

In ultima analisi, per concludere la trattazione, è doveroso sottolineare come, nonostante entrambi i modelli addestrati siano eccellenti nel predire le classi della categoria *Income Category*, il modello LGBMClassifier, cosi come dimostrato per il LGBMRegressor utilizzato nel paragrafo 4.5.1, presenta performance migliori, e si dimostra, per il problema e la tipologia di dati in esame, superiore rispetto al Random Forest.

Un ultimo aspetto interessante da evidenziare, infine, è legato al valore della metrica *Cross-Validation Score*, la quale indica una media dell'accuratezza delle diverse iterazioni della cross-validation. Entrambi i modelli registrano un valore di esso prossimo ad uno; ciò indica l'ottima capacità dei due di generalizzare bene su nuovi dati.

Capitolo 5

Conclusioni

La povertà è una delle problematiche più importanti che la società moderna si trova a dover fronteggiare. I governi di tutto il mondo, così come le principali organizzazioni internazionali, operano con l'obiettivo di ridurre ed infine eliminare questa terribile condizione in cui tante, troppe persone, sono costrette a vivere.

In questo contesto e a fronte di questa enorme sfida si inserisce perfettamente, come potente alleato, l'Intelligenza Artificiale.

Nel presente lavoro di tesi, dopo aver spiegato, passo dopo passo, come funziona ed in cosa consiste il processo di estrazione di conoscenza dai dati KDD (Knowledge Discovery in Databases), si è provveduto ad illustrare lo stato dell'arte circa gli studi effettuati nell'ambito della poverty prediction mediante le potenti tecniche di AI. In questo contesto sono state approfondite diverse tematiche, tra cui i due principali approcci impiegati in letteratura per trattare la povertà, l'approccio monetario e quello non monetario; le diverse tipologie di dati utilizzate in input dai principali tool testati, ed ultimo, ma non per importanza, i numerosi modelli di Machine Learning e Deep Learning adottati nelle varie ricerche. A seguito di ciò, sfruttando le conoscenze e le informazioni acquisite dallo studio dei lavori pre-esistenti sul tema e dai concetti teorici sull'applicazione delle metodologie Data-driven, sul funzionamento e la creazione di opportuni modelli di ML, si è effettuata una analisi di ricerca di predizione dello stato di povertà delle famiglie filippine, con dati provenienti da survey nazionali effettuate dal PSA (Philippine Statistics Authority) relativamente agli anni 2012 e 2015, mediante due obiettivi principali:

- predire il reddito annuo delle famiglie adottando un approccio alla povertà monetario partendo da attributi di natura prevalentemente economica, come reddito e spese, mediante modelli di regressione;
- predire la fascia di reddito di appartenenza delle famiglie, adottando un approccio non monetario, considerando variabili legate al capofamiglia, alla

numerosità della famiglia, all'abitazione e all'istruzione, tra le tante, mediante modelli di classificazione.

Per il raggiungimento di entrambi gli obiettivi preposti, si è effettuata una profonda analisi esplorativa del dataset, seguita da una opportuna rielaborazione dei dati, differenziata per obiettivo, e si sono andati a generare e testare i diversi modelli di regressione e classificazione selezionati. Infine, si sono confrontati i risultati ottenuti dall'applicazione dei vari modelli e sono stati individuati quelli maggiormente performanti. In particolare, sia nel primo che nel secondo caso il modello che ha fornito i migliori risultati è stato il *LightGBM*, dimostrandosi particolarmente efficace nel caso della classificazione. Ciononostante, anche il *Random Forest* si è dimostrato essere un algoritmo notevole in entrambi i casi, fornendo risultati ottimi, seppur leggermente inferiori al LightGBM, come visto nelle sezioni 4.5.1 e 4.5.2. Le figure 4.82 e 4.83 raffigurano le capacità predittive di entrambi i modelli di regressione, confrontando valori reali e valori predetti. É immediato notare come le due figure siano sostanzialmente molto simili, andando a testimoniare come entrambi forniscano risultati ampiamente validi. Discorso analogo si può fare osservando le matrici di confusione riportate nelle figure 4.88 e 4.89. Esse offrono un'idea visiva delle capacità predittive di entrambi i modelli di classificazione, sottolineando nuovamente come tutti e due siano ottimi per gli obiettivi preposti.

Un aspetto particolarmente interessante che è emerso nel corso dello studio e che trova riscontro con la teoria sul tema, è l'incredibile importanza che assume un dataset correttamente preparato. Confrontando i risultati mostrati nei paragrafi 4.6.2 e 4.6.3 si può osservare come le performance dei modelli siano strettamente legate alla qualità del dataset di input. In particolare, prendendo il medesimo modello, come può ad esempio essere il *Random Forest*, ed impiegandolo con dataset differenti, di cui uno perfettamente bilanciato nelle sue classi, ed uno fortemente sbilanciato, i risultati che si ottengono sono significativamente diversi, e peggiorano notevolmente al diminuire della qualità dei dati. É imprescindibile, dunque, il bilanciamento dei dati se si ha come obiettivo quello di generare un modello predittivo efficace.

Ciononostante, quello appena descritto non è l'unico modo per rendere ancora migliori le performance dei modelli di ML. Si potrebbe, ad esempio, approfondire ulteriormente la fase di feature selection e di feature engineering, andando ad affinare le variabili in gioco, oppure si potrebbero testare nuovi modelli di natura differente da quelli considerati, per poterne valutare le performance, o ancora combinare entrambe le azioni.

Alla luce di quanto detto, si possono trarre le seguenti conclusioni; le potenzialità dei modelli di Machine Learning in una sfida cruciale come quella contro la povertà sono evidenti. Essi, infatti, offrono la possibilità di effettuare analisi permettendo

di sfruttare tutta una serie di dati che, sia per tipologia, che per dimensione, altri strumenti utilizzati in passato, come ad esempio quelli econometrici, non erano in grado di utilizzare. I modelli di AI, hanno dunque, spalancato le porte a nuove modalità di ricerca, consentendo di impiegare dati geospaziali, i quali, a differenza dei dati di campo provenienti da survey estremamente costose da realizzare e relative necessariamente al passato, consentono di monitorare il problema della povertà in tempo reale, istante per istante potenzialmente. Inoltre, i tool di AI consentono di offrire una visione più ampia sul problema stesso della povertà, permettendo di introdurre nell'analisi variabili che apparentemente potrebbero sembrare lontane da esso, ma che in realtà si sono dimostrate parte integrante, come il livello di istruzione del capofamiglia ad esempio.

In aggiunta a ciò, un altro aspetto interessante dello studio effettuato, è legato alla possibilità di estendere quanto fatto a contesti diversi. In particolare, partendo da dataset strutturati in maniera simile a quello utilizzato, e dunque contenente informazioni sia monetarie che non circa la conduzione delle famiglie di un Paese, si possono testare gli algoritmi usati nel contesto specifico della predizione della povertà nelle Filippine, in altri contesti economici. Ciò che probabilmente andrà a variare sarà l'importanza e gli effetti che le variabili possono assumere nella predizione, ma i risultati ottenuti nel presente lavoro di tesi possono fungere da base per studi simili. Nell'affermare ciò, tuttavia, non si vuole avere la presunzione di aver generato una linea guida universale per affrontare problemi di questa natura, ma semplicemente affermare che questo studio può essere di aiuto per lavori futuri sul tema in questione.

Per quanto, dunque, il presente lavoro sia un lavoro preliminare che può essere ancora ampliato e migliorato, va comunque riconosciuto che i risultati ottenuti utilizzando i modelli di AI sono notevoli. Ciò è un'ulteriore testimonianza dell'importanza che l'utilizzo dei modelli di AI può rivestire nell'affrontare le più principali sfide globali. Se i risultati ottenuti sono così significativi per la lotta alla povertà, non si può ignorare il fatto che i medesimi risultati si potrebbero ottenere in altri ambiti, come ad esempio nel campo della medicina, in cui l'AI sta trovando sempre più impiego.

Sono quindi impossibili da negare le potenzialità delle tecnologie Data-Driven nell'affrontare problematiche centrali che l'essere umano non ha tuttora risolto. C'è una enorme necessità che ricerche in tal senso continuino e che vengano superati definitivamente i limiti delle tecnologie precedenti all'avvento dell'AI.

Nonostante quanto detto sia incoraggiante ed i risultati ottenuti forniscano un importante strumento per la lotta alla povertà, ci sono degli aspetti che è doveroso sottolineare e su cui è opportuno porre l'attenzione. Il lavoro di ricerca effettuato nel contesto delle Filippine è stato possibile grazie alla presenza dei dati raccolti e strutturati dal PSA e quindi dal governo Filippino; in assenza di tali dati, non sarebbe stato possibile fare alcuna delle analisi presentate. Non è scontato, quindi,

che dei dati così ben raccolti ed organizzati siano presenti per ogni Paese, ed in loro assenza sarebbe richiesto un enorme sforzo per poter realizzare un dataset analogo su cui effettuare uno studio simile. É dunque, possibile, che i Paesi più poveri non abbiano le potenzialità per poter effettuare costosi sondaggi ed ottenere dati come quelli qui utilizzati, nonostante questi paesi siano quelli in cui ci sarebbe maggior bisogno di effettuare studi e ricerche per combattere la povertà. Fornire loro gli strumenti per la raccolta di tali dati rappresenterebbe un enorme passo in avanti nella lotta alla povertà.

In conclusione, le sfide sul tema sono ancora numerose, e necessitano di maggiori studi e di un grande impegno collettivo. Tuttavia, i progressi che l'AI ha permesso di fare in questo ambito, in un intervallo di tempo relativamente breve da quando è stato introdotto per la prima volta, lasciano ben sperare.

Appendice A

Codice Python

A.1 Generazione Istogrammi raffiguranti le abitudini di spesa delle famiglie, in media, per Regione

Python_Code/Istogramma_spesa_famiglie_per_regione.py

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Caricamento del dataset in formato CSV
file_csv = 'Family Income and Expenditure.csv'
df = pd.read_csv(file_csv)



attributi_spesa = ['Bread and Cereals Expenditure',
'Total Rice Expenditure',
'Meat Expenditure',
'Total Fish and  marine products Expenditure',
'Fruit Expenditure',
'Vegetables Expenditure',
'Restaurant and hotels Expenditure',
'Alcoholic Beverages Expenditure',
'Tobacco Expenditure',
'Clothing, Footwear and Other Wear Expenditure',
'Housing and water Expenditure',
'Medical Care Expenditure',
'Transportation Expenditure',
'Communication Expenditure',
'Education Expenditure',
```

```
'Miscellaneous Goods and Services Expenditure',
'Special Occasions Expenditure',
'Crop Farming and Gardening expenses'] # Lista degli attributi di spesa da visualizzare
regioni = df['Region'].unique() # Valori distinti dell'attributo 'Region'

# Ciclo for per la generazione di grafici distinti per ogni regione
for regione in regioni:
    plt.figure(figsize=(8, 8))
    dati_regione = df[df['Region'] == regione]

    # Calcolo del valor medio di spesa per ogni attributo 'Expenditure'
    valori_medi = dati_regione[attributi_spesa].mean()

    # Generazione del bar column diagram. Palatte di colori in scala cromatica che vanno dal blu al giallo.
    sns.barplot(x=valori_medi.index, y=valori_medi, palette='viridis')

    plt.title(f'Abitudini di consumo delle famiglie filippine residenti in {regione}')
    plt.ylabel('Valore medio di spesa in pesos')
    plt.xlabel('Attributi di spesa delle famiglie filippine')
    plt.xticks(rotation=90)

    plt.tight_layout()
    plt.show()
```

A.2 Generazione diagrammi a barre raffiguranti le 5 occupazioni più comuni per le regioni con minor THI medio

Python_Code/Top_5_Occupation_per_region_bar_diagram.py

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import textwrap

# Caricamento del dataset in formato CSV
file_csv = "Family Income and Expenditure.csv"
df = pd.read_csv(file_csv)

# Inserimento nome della regione
```

```
regione_selezionata = input("Inserisci il nome della regione desiderata: ")

# Filtraggio dati per la regione selezionata
df_region = df[df["Region"] == regione_selezionata]

# Conteggio dei valori più comuni di "Household Head Occupation" nella regione selezionata
top_occupations = df_region["Household Head Occupation"].value_counts().head(5)

# Creazione del grafico a barre
plt.figure(figsize=(10, 6))
sns.set_style("whitegrid")
sns.barplot(x=top_occupations.values, y=top_occupations.index, palette="colorblind")

plt.xlabel("Frequenza", fontsize=12, weight= 'bold')
plt.ylabel("Occupazione", fontsize=12, weight= 'bold')
plt.title(f"Top 5 Household Head Occupations in {regione_selezionata}", fontsize=14, weight= 'bold')
plt.xticks(rotation=0)
wrapped_labels = [textwrap.fill(label, width=20) for label in top_occupations.index]
plt.gca().set_yticklabels(wrapped_labels, fontsize=10)


plt.tight_layout()
plt.show()
```

A.3 Distribuzione del livello di istruzione del Capofamiglia, per Regione

Python_Code/Household_education_level_per_region.py

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

data = pd.read_csv('Family Income and Expenditure.csv')

rename_dict = {
    "Other Programs in Education at the Third Level, First Stage, of the Type that Leads to an Award not Equivalent to a First University or Baccalaureate Degree": "Third Level Education, not equivalent to a Baccalaureate Degree",
```

```
    "Other Programs of Education at the Third Level, First Stage, of the Type that Leads to a Baccalaureate or First University/Professional Degree (HIgher Education Level, First Stage, or Collegiate Education Level)": "Third Level Education equivalent to a Baccalaureate Degree"
}
data['Household Head Highest Grade Completed'] = data['Household Head Highest Grade Completed'].replace(rename_dict)

# Elenco regioni
regions = data['Region'].unique()


sns.set_theme(style="whitegrid", rc={"figure.figsize":(15, 8)})

# Creazione grafico per ogni regione
for region in regions:
    subset = data[data['Region'] == region]

    ax = sns.countplot(
        y=subset['Household Head Highest Grade Completed'],
        order=subset['Household Head Highest Grade Completed'].value_counts().index,
        palette="Blues_d"
    )

    # Aggiunta le etichette con il numero di occorrenze a ogni barra
    for p in ax.patches:
        ax.annotate(f'{int(p.get_width())}',
                    (p.get_width(), p.get_y() + p.get_height()/2),
                    ha='left', va='center',
                    fontsize=10, color='black',
                    xytext=(5,0),
                    textcoords='offset points')


    plt.title(f"Distribuzione livello di Istruzione dei Capofamiglia in {region}", fontsize=16, weight='bold')
    plt.ylabel("Livello di Istruzione", fontsize=12)
    plt.xlabel("Occorrenze", fontsize=12)
    plt.tight_layout()

    plt.show()
```

A.4 Distribuzione del Reddito familiare medio per livello di istruzione del Capofamiglia

Python_Code/THI_per_Household_education_level.py

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt


data = pd.read_csv('Family Income and Expenditure.csv')

rename_dict = {
    "Other Programs in Education at the Third Level, First Stage, of the Type that Leads to an Award not Equivalent to a First University or Baccalaureate Degree": "Third Level Education, not equivalent to a Baccalaureate Degree",
    "Other Programs of Education at the Third Level, First Stage, of the Type that Leads to a Baccalaureate or First University/Professional Degree (HIgher Education Level, First Stage, or Collegiate Education Level)": "Third Level Education equivalent to a Baccalaureate Degree"
}

data['Household Head Highest Grade Completed'] = data['Household Head Highest Grade Completed'].replace(rename_dict)


sns.set_theme(style="whitegrid", rc={"figure.figsize":(15, 10)})

# Creazione Boxplot
sns.boxplot(
    x=data['Total Household Income'],
    y=data['Household Head Highest Grade Completed'],
    palette="Oranges_r",
    order=data.groupby('Household Head Highest Grade Completed').median()['Total Household Income'].sort_values().index
)


plt.title("Distribuzione del Reddito familiare medio \nper Livello di Istruzione del Capofamiglia", fontsize=16, weight='bold')
plt.ylabel("Livello di Istruzione del Capofamiglia", fontsize=12)
plt.xlabel("Reddito Totale della Famiglia", fontsize=12)
plt.xlim(0, 0.3e7)  # Limite sull'asse delle x
plt.tight_layout()

plt.show()
```

A.5 Numero di membri della famiglia impiegati, a fronte del numero totale di membri, diviso per tipologia di famiglia

Python_Code/Fam_members_employed_per_fam_type_and_numeber.py

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt


data = pd.read_csv('Family Income and Expenditure.csv')

# Selezione dei valori 'Single Family' e 'Extended Family'
# e con numero di membri minore di 9
filtered_data = data[
    (data['Type of Household'].isin(['Single Family', 'Extended Family'])) &
    (data['Total Number of Family members'] <= 8)
]

# Calcolo del numero totale di famiglie per ogni valore di 'Total Number of Family members'
total_families = filtered_data['Total Number of Family members'].value_counts()


sns.set_theme(style="whitegrid")

# Creazione catplot
g = sns.catplot(
    data=filtered_data,
    x='Total number of family members employed',
    hue='Type of Household',
    col='Total Number of Family members',
    kind='count',
    height=4,
    aspect=0.7,
    palette="Greens_d",
    col_wrap=4
)

# Aggiunta linee medie
for ax in g.axes.flat:
    total_members = int(ax.get_title().split('= ')[-1])
    data_for_axis = filtered_data[filtered_data['Total Number of Family members'] == total_members]
```

```
38
39      single_mean = round(data_for_axis[data_for_axis['Type of
        Household'] == 'Single Family']['Total number of family members
        employed'].mean())
40      extended_mean = round(data_for_axis[data_for_axis['Type of
        Household'] == 'Extended Family']['Total number of family members
        employed'].mean())
41
42      ax.axvline(x=single_mean, color='lightgreen', linestyle='--')
43      ax.axvline(x=extended_mean, color='darkgreen', linestyle='--')
44
45
46  g.set_axis_labels('Numero di Membri della\nFamiglia Impiegati', '
        Numero di Famiglie')
47
48
49  for ax in g.axes.flat:
50      col_title = ax.get_title().split('= ')[-1]
51      ax.set_title(f"Membri Totali = {col_title}\n(Totale: {
        total_families[int(col_title)]})", weight = 'bold')
52      ax.set_xlabel('Numero di Membri della\nFamiglia Impiegati',
        fontsize=10)
53      ax.set_xticks(ax.get_xticks())  # Assicura che i ticks siano
        mostrati
54      ax.set_xticklabels(ax.get_xticks(), rotation=0)  # Mostra la
        scala dei valori
55
56  g.tight_layout()
57
58  plt.show()
```

A.6 Matrice di Correlazione tra THI e variabili di spesa

Python_Code/Correlation_Matrix.py

```
1  import pandas as pd
2  import seaborn as sns
3  import matplotlib.pyplot as plt
4
5
6  data = pd.read_csv('Family Income and Expenditure.csv')
7
8  # Selezione delle variabili di spesa
9  expenditure_columns = [col for col in data.columns if 'Expenditure'
       in col or 'expenditures' in col]
```

```
10
11 # Selezione del THI
12 selected_data = data[['Total Household Income'] + expenditure_columns
       ]
13
14 # Calcolo del coefficiente di correlazione di Pearson
15 correlation_matrix = selected_data.corr()
16
17 # Ordinamento delle colonne in base al valore assoluto della loro
       correlazione con "Total Household Income"
18 sorted_columns = correlation_matrix['Total Household Income'].abs().
       sort_values(ascending=False).index
19 sorted_correlation_matrix = correlation_matrix.loc[sorted_columns,
       sorted_columns]
20
21 # creazione della heatmap delle correlazioni
22 plt.figure(figsize=(15,15))
23 sns.heatmap(sorted_correlation_matrix, annot=True, cmap='coolwarm',
       vmin=-1, vmax=1)
24 plt.title('Heatmap delle Correlazioni')
25 plt.show()
```

A.7 Trasformazione logaritmica e IQR outliers detection variabili di spesa e THI

Python_Code/Log_transformation_and_IQR_outliers_detection.py

```
1 import pandas as pd
2 import numpy as np
3 import seaborn as sns
4 import matplotlib.pyplot as plt
5
6
7 df = pd.read_csv('Family Income and Expenditure.csv')
8
9 # Selezione colonne relative agli attributi di spesa
10 expenditure_columns = [col for col in df.columns if 'Expenditure' in
       col or 'expenditures' in col]
11
12 # Applicazione della trasformazione logaritmica (ln(1+x)) alle
       colonne selezionate
13 for col in expenditure_columns:
14     log_col = col + '_log'
15     df[log_col] = np.log1p(df[col])
16
17     # Calcolo IQR, LI, LS
```

```
    Q1 = df[log_col].quantile(0.25)
    Q3 = df[log_col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # Rimozione outliers
    filtered_df = df[(df[log_col] >= lower_bound) & (df[log_col] <=
    upper_bound)]

    # Generazione istogrammi
    plt.figure(figsize=(10, 6))
    sns.histplot(filtered_df[log_col], kde=True, color="teal", bins
    =100)
    plt.title(f'Distribuzione del Logaritmo di {col} avendo rimosso
    gli outliers', weight ='bold')
    plt.xlabel(f'Log_{col}')
    plt.ylabel('Frequenza')
    plt.show()
```

A.8 Preprocessing obiettivo 1 e creazione file CSV

Python_Code/Preprocessing_1.py

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv('Family Income and Expenditure.csv')

# Sostituzione degli spazi, dei due punti e delle virgole con "_"
df.columns = df.columns.str.replace(' ', '_').str.replace(':', '_').
    str.replace(',', '')

# Attributi di spesa da trattare separatamente
specified_columns = ['Alcoholic_Beverages_Expenditure', '
    Education_Expenditure', 'Tobacco_Expenditure', '
    Special_Occasions_Expenditure']

# Limiti personalizzati per gli attributi selezionati separatamente
custom_bounds = {
    'Alcoholic_Beverages_Expenditure': (1, 14),
    'Tobacco_Expenditure': (1, 14),
    'Education_Expenditure': (1, 14),
```

```
    'Special_Occasions_Expenditure': (1, 14)
}

# Attributi di spesa selezionati con l'aggiunta del THI
expenditure_columns = [col for col in df.columns if ('Expenditure' in col or 'expenditures' in col or col == 'Total_Household_Income')
                       and col not in ['Total_Rice_Expenditure', 'Meat_Expenditure']]

for col in expenditure_columns:
    log_col = col + '_log'
    df[log_col] = np.log1p(df[col])

    # Calcolo IQR, LI, LS
    Q1 = df[log_col].quantile(0.25)
    Q3 = df[log_col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    if col in specified_columns:
        # Rimozione outliers basata sull'IQR e applicazione dei limiti personalizzati per le variabili trattate separatamente
        custom_lower_bound, custom_upper_bound = custom_bounds[col]
        filtered_df = df[(df[log_col] >= lower_bound) & (df[log_col] <= upper_bound) &
                         (df[log_col] >= custom_lower_bound) & (df[log_col] <= custom_upper_bound)].copy()
    else:
        # Solo rimozione outliers basata sull'IQR per le restanti variabili
        filtered_df = df[(df[log_col] >= lower_bound) & (df[log_col] <= upper_bound)].copy()

    # Standardizzazione con Z-Score
    z_col = log_col + '_zscore'
    filtered_df[z_col] = (filtered_df[log_col] - filtered_df[log_col].mean()) / filtered_df[log_col].std()

    # Generazione istogrammi per le variabili standardizzate
    plt.figure(figsize=(10, 6))
    sns.histplot(filtered_df[z_col], kde=True, color="teal", bins=100)
    plt.title(f'Distribuzione del Logaritmo Standardizzato di "{col}"', weight='bold')
    plt.xlabel(f'Z-Score_{col}')
    plt.ylabel('Frequenza')
    plt.show()

```

```
# Creazione del nuovo file CSV
columns_to_save = [col + '_log' for col in expenditure_columns]
df[columns_to_save].to_csv('
    Preprocessed_1_Family_Income_and_Expenditure.csv', index=False)
print(df[columns_to_save].head())
```

A.9 Preprocessing obiettivo 2 e creazione file CSV

Python_Code/Preprocessing_2.py

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from scipy import stats
import seaborn as sns
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from imblearn.over_sampling import SMOTE
from imblearn.under_sampling import RandomUnderSampler
from imblearn.pipeline import Pipeline


df = pd.read_csv('Family Income and Expenditure.csv')

# Rimozione variabili di spesa, le variabili non necessarie e gli
    outliers
df = df.loc[:, ~df.columns.str.contains('Expenditure|expenses', case=
    False)]
columns_to_remove = ['Main Source of Income', 'Imputed House Rental
    Value']
df.drop(columns=columns_to_remove, inplace=True)
df = df[df['Agricultural Household indicator'] != 2]

# Mappatura delle categorie di occupazione
occupation_categories = {
    'Agriculture & Livestock Farming': ['Farmhands and laborers','
    Rice farmers','Hog raising farmers','Vegetable farmers','Corn
    farmers','Coconut farmers',
                                        'Root crops farmers','Fruit
    tree farmers','Coffee and cacao farmers','Forestry laborers','
    Cattle and dairy farmers',
                                        'Other livestock farmers','
    Field legumes farmers','Cotton and fiber crops farmers','Forest
    tree planters','Chicken farmers',
```

```
25                                          'Duck raisers','Other poultry
 farmers','Other field crop farmers','Sugarcane farmers','Fish-
farm cultivators (excluding prawns)',
26                                          'Other orchard farmers','
Ornamental plant growers','Grain and spice milling machine
operators','Deep-sea fishermen',
27                                          'Agronomists and related
scientists','Other plant growers',
28                                          'Seaweeds cultivators','Other
 animal raisers','Tree nut farmers','Other aqua products
cultivators','Hunters and trappers',
29                                          'Agricultural or industrial
machinery mechanics and fitters','Farm technicians','Fishermen n.
e. c.','Fishery laborers and helpers',
30                                          'Foresters and related
scientists','Hunting and trapping laborers','Inland and coastal
waters fishermen',
31                                          'Minor forest products
gatherers','Motorized farm and forestry plant operators'],
32
33     'Building & Construction': ['Building construction laborers','
Carpenters and joiners','Masons and related concrete finishers',
34                                 'Painters and related workers','
Construction and maintenance laborers: roads, dams and similar
constructions',
35                                 'Plumbers, pipe fitters and other
related workers','Roofers',
36                                 'Structural-metal preparers, erectors
 and related workers',
37                                 'Sheet-metal workers','Riggers and
cable splicers','Stone splitters, cutters and carvers',
38                                 'Floor layers and tile setters','
Insulation workers','Builders (traditional materials)',
39                                 'Building and fire inspectors',
40                                 'Building caretakers','Building frame
 and related trades workers n. e. c.','Civil engineering
technicians',
41                                 'Draftsmen','Earth-moving and related
 plant operators','Building and related electricians',
42                                 'Well drillers and borers and related
 workers'],
43
44     'Transportation & Logistics': ['Transport conductors','Heavy
truck and lorry drivers','Car, taxi and van drivers','Motorcycle
drivers',
45                                    'Bus drivers','Ship\'s deck crews
and related workers','Lineman, line installers and cable splicers'
,
```

```
46                                  'Ships\' deck officers and pilots'
    ,'Air transport service supervisors','Lifting truck operators',
47                                  'Transport clerks','Drivers of
    animal-drawn vehicles and machinery','Postal service supervisors',
48                                  'Maritime transport service
    supervisors','Locomotive engine drivers',
49                                  'Air traffic safety technicians','
    Aircraft engine mechanics and fitters',
50                                  'Crane, hoist and related plant
    operators','Hand or pedal vehicle drivers',
51                                  'Marine craft mechanics','
    Transport and communications service supervisors n. e. c.'],
52
53  'Management & Administration': ['General managers/managing
    proprietors in transportation, storage and communications',
54                                  'General managers/managing
    proprietors in wholesale and retail trade',
55                                  'General managers/managing
    proprietors of restaurants and hotels','General managers/managing
    proprietors in manufacturing',
56                                  'General managers/managing
    proprietors in personal care, cleaning and relative services',
57                                  'General managers/managing
    proprietors in agriculture, hunting, forestry and fishing',
58                                  'General managers/managing
    proprietors in construction','Professional, technical and related
    officers',
59                                  'Justices','Legal and related
    business associate professionals','Other finance and sales
    associate professionals',
60                                  'Production and operations
    managers in agriculture, hunting, forestry and fishery',
61                                  'Production and operations
    managers in transport, storage and communications','Sales and
    marketing managers',
62                                  'Research and development
    managers','Production and operations managers in restaurant and
    hotels',
63                                  'Production and operations
    managers in manufacturing','Other office clerks',
64                                  'Production and operations
    managers in personal care, cleaning and relative services','
    Service and related workers',
65                                  'Production and operations
    managers in wholesale and retail trade','Production and operations
     managers in construction',
66                                  'Production and operations
    managers in business services','Finance and administration
    managers','Supply and distribution managers',
```

```
67                                      'Other specialized managers','
    Personnel and industrial relations managers','Personnel and human
    resource development professionals',
68                                      'Government administrators (
    including career executive service officers)','Legislative
    officials','Secretaries',
69                                      'Directors and chief executives
    of corporations','Traditional chiefs and heads of villages','
    Government tax and excise officials',
70                                      'Government social benefits
    officials','Other government associate professionals','Estate
    agents','Government licensing officials',
71                                      'Administrative secretaries and
    related associate professionals','Other business professionals','
    Other business services and trade brokers',
72                                      'General managers/managing
    proprietors n. e. c.','General managers/managing proprietors of
    business services',
73                                      'Other administrative associate
    professionals','Receptionists and information clerks','Bookkeepers
    ','Coding, proof-reading and related clerks',
74                                      'Statistical and finance clerks',
    'Stocks clerks'],
75
76     'Production, Handicraft & Manufacturing': ['Electronics
    mechanics and servicers','Welders and flamecutters', 'Motor
    vehicle mechanics and related trades workers',
77                                                  'Mechanical
    engineering technicians','Production supervisors and general
    foremen','Food preservers',
78                                                  'Basketry weavers,
    brush makers and related workers','Wood products machine operators
    ',
79                                                  'Tellers and other
    counter clerks','Incinerator, water treatment and related plant
    operators',
80                                                  'Electrical mechanics
     and fitters','Electronics fitters',
81                                                  'Weaving and knitting
     machine operators','Sewing machine operators',
82                                                  'Glass and ceramics
    kiln and related machine operators','Production clerks',
83                                                  'Tailors, dressmakers
     and hatters','Mining and quarrying laborers',
84                                                  'Assembling laborers'
    ,'Business machines mechanics and repairers','Automated assembly-
    line operators',
```

```
85                                                  'Baked goods and
    cereal and chocolate products machine operators','Bakers, pastry
    cooks and confectionery makers',
86                                                  'Blacksmiths,
    hammersmiths, and forging-press workers','Bleaching, dyeing and
    cleaning machine operators',
87                                                  'Bookbinders and
    related workers','Brewers and wine and other beverage machine
    operators',
88                                                  'Cabinet/furniture
    makers and related workers','Cement and other mineral products
    machine operators',
89                                                  'Charcoal makers and
    related workers','Chemical processing plant operators n. e. c.','
    Chemical products machine operators n. e. c.',
90                                                  'Concessionaires and
    loggers','Dairy products machine operators','Dairy products makers
    ','Data entry operators',
91                                                  'Electrical equipment
     assemblers','Electronic equipment assemblers','Fiber preparers',
92                                                  'Fiber preparing,
    spinning and winding machine operators','Food and beverage tasters
     and graders',
93                                                  'Fruit, vegetable and
     nut processing machine operators','Glass makers, cutters,
    grinders and finishers',
94                                                  'Glass, ceramics and
    related plant operators n. e. c.','Hand launderers and pressers','
    Hand packers and other manufacturing laborers',
95                                                  'Handicraft workers
    in wood and related materials','Industrial robot operators',
96                                                  'Jewelry and precious
     metal workers','Labor contractors and employment agents','Machine
     tool operators',
97                                                  'Machine-tool setters
     and setter operators','Mail carriers and sorting clerks',
98                                                  'Meat and fish
    processing machine operators','Mechanical machinery assemblers',
99                                                  'Messengers, package
    and luggage porters and deliverers','Metal drawers and extruders',
    'Metal finishing, plating and coating machine operators',
100                                                 'Metal melters,
    caster and rolling mill operators','Metal molders and coremakers',
    'Metal, rubber and plastic products assemblers',
101                                                 'Metal-wheel grinders
    , polishers and tool sharpeners','Mineral ore and stone-processing
     plant operators','Miners and quarry workers',
```

```
102                                                 'Mining-plant
    operators','Musical instrument makers and tuners','Other machine
    operators and assemblers',
103                                                 'Other sales
    supervisors','Paper pulp plant operators','Paperboard, textile and
     related products assemblers','Papermaking plant operators',
104                                                 'Pawnbrokers and
    money lenders','Petroleum and natural gas refining plant operators
    ','Pharmaceutical and toiletry products machine operators',
105                                                 'Plastic products
    machine operators','Potters and related clay and abrasive formers'
    ,'Power production plant operators',
106                                                 'Precision instrument
     makers and repairers','Rattan, bamboo and other wicker furniture
    makers','Rubber products machine operators',
107                                                 'Sewers, Embroiderers
     and related workers','Shoemakers and related workers','Shoemaking
     and related machine operators',
108                                                 'Shotfirers and
    blasters','Silk-screen, block and textile printers','Sugar
    production machine operators',
109                                                 'Sweepers and related
     laborers','Tanners',
110                                                 'Technician, skilled,
     semi-skilled workers','Telecommunication equipment installers and
     repairers',
111                                                 'Textile and leather
    products machine operators n. e. c.','Textile, leather and related
     patternmakers and cutters',
112                                                 'Tobacco preparers
    and tobacco products makers','Tool-makers and related workers',
113                                                 'Upholsterers and
    related workers','Varnishers and related painters',
114                                                 'Weavers, knitters
    and related workers','Wood and related products assemblers','Wood
    processing plant operators','Wood treaters',
115                                                 'Woodworking machine
    setters and setter-operators','Word processor and related
    operators',
116                                                 'Workers reporting
    occupations unidentifiable or inadequately defined'],
117
118  'Commerce & Sales': ['Cashiers and ticket clerks','Shop
    salespersons and demonstrators',
119                          'Trade brokers','Stall and market
    salespersons','Buyers','Travel consultants and organizers',
120                          'Freight handlers','Sales supervisors in
    retail trade','Sales supervisors in wholesale trade',
```

```
121                          'Accounting and bookkeeping clerks','Door-
    to-door and telephone salespersons','Insurance representatives',
122                          'Market and sidewalk stall vendors','
    Technical and commercial sales representatives'],
123
124     'Education & Training': ['Science and mathematics teaching
    professionals','School supervisors and principals',
125                          'Nonformal education teaching
    professionals other than technical and vocational trainors/
    instructors',
126                          'Vocational education teaching
    professionals','Teaching associate professionals','Pre-elementary
    education teaching professionals',
127                          'General elementary education teaching
    professionals','General secondary education teaching professionals
    ',
128                          'College, university and higher education
     teaching professionals','Education methods specialists',
129                          'Other teaching professionals','
    Librarians, archivists and curators','Library and filing clerks',
130                          'School principals','Technical and
    vocational instructors/trainors',
131                          'Non-ordained religious associate
    professionals','Science and mathematics elementary education
    teaching professionals'],
132
133     'Health Care & Social Services': ['Professional nurses','Medical
    doctors','Pharmacists','Dentists',
134                                      'Midwifery associate
    professionals','Professional midwives','Optometrists and opticians
    ',
135                                      'Medical assistants','Dental
    assistants','Pharmaceutical assistants',
136                                      'Veterinarians','Medical
    equipment operators','Garbage collectors','Social work
    professionals',
137                                      'Child care workers','Faith
    healers','Nursing associate professionals',
138                                      'Social work associate
    professionals','Traditional medicine practitioners',
139                                      'Nutritionists-dietitians','
    Medical technologists','Veterinary assistants',
140                                      'Other health associate
    professionals (except nursing)','Other health professionals (
    except nursing)',
141                                      'Other life science technicians
    ','Physiotherapists','Protective services workers n. e. c.',
```

```
142                                     'Safety, health and quality
    inspectors (vehicles, processes and products)','Undertakers and
    embalmers',
143                                     'Senior officials of
    humanitarian and other special-interest organizations'],
144
145  'Personal Services, Law Enforcement, Catering & Hospitality': ['
    Cooks','Waiters, waitresses and bartenders','Hairdressers, barbers
    , beauticians and related workers',
                                                                  ,
146
    Institution-based personal care workers','Home-based personal care
     workers','Debt collectors and related workers',
                                                                  ,
147
    Other personal services workers, n. e. c.','Housekeepers and
    related workers','Enlisted personnel n. e. c.',
                                                                  ,
148
    Police inspectors and detectives','Police officers','Prison guards
    ','Firefighters',
                                                                   ,
149
    Shoe cleaning and other street services elementary occupations','
    Staff officers','Street ambulant vendors',
                                                                  ,
150
    Travel guides', 'Doorkeepers, watchpersons and related workers','
    Customs and immigration inspectors',
                                                                  ,
151
    Domestic helpers and cleaners','Personal care and related workers,
     n. e. c.','Other supervisors, n. e. c.',
                                                                  ,
152
    Helpers and cleaners in offices, hotels and other establishments',
    'Officers, n. e. c.','Telephone switchboard operators',
                                                                  ,
153
    Vehicle, window and related cleaners', 'Appraisers and valuers','
    Bet bookmakers and croupiers','Other computer professionals',
                                                                  ,
154
    Butchers, fishmongers and related food preparers','Combat soldiers
    ','Commanding officers','Computer assistants'],
155
156  'Science, Engineering & ICT': ['Architects','Civil engineers','
    Electrical Engineers','Industrial engineers',
157                                     'Mechanical engineers','Chemical
    engineers','Economists','Geodetic engineers and related
    professionals',
158                                     'Accountants and auditors','Systems
     analysts and designers','Electrical engineering technicians',
159                                     'Electronics and communications
    engineering technicians','Electronics and communications engineers
    ',
```

```
160                                     'Advertising and public relations
    managers','Chemists','Life science technicians','Town planners and
     related professionals',
161                                     'Statisticians','Lawyers','Other
    social science professionals','Ship and aircraft controllers and
    technicians',
162                                     'Mining and metallurgical
    engineering technicians','Other physical science and engineering
    technicians',
163                                     'Photographers and image and sound
    recording equipment operators','Computer programmers','Statistical
    , mathematical and related associate professionals',
164                                     'Decorators and commercial
    designers','Religious professionals','Computer equipment operators
    ',
165                                     'Aircraft pilots, navigators and
    flight engineers','Other engineers and related professionals','
    Computer engineers and related professionals'],
166
167     'Entertainment, Art, Music, Sport & Fashion' : ['Radio,
    television and other announcers','Authors, journalists and other
    writers','Pressman letterpresses and related workers',
168                                                     'Companions and
    valets','Sculptors, painters and related artists','Composers,
    musicians and singers',
169                                                     'Fashion and
    other models','Athletes and related workers','Broadcasting and
    telecommunications equipment operators',
170                                                     'Choreographers
    and dancers','Compositors, typesetters and related workers','Other
     creative or performing artists',
171                                                     'Photographic and
     related workers','Photographic products machine operators','
    Street, nightclub and related musicians, singers and dancers',
172                                                     'Stenographers
    and typists']
173
174 }
175
176 occupation_to_category = {occupation: category for category,
    occupations in occupation_categories.items() for occupation in
    occupations}
177
178 df = df.copy()
179
180 # Creazione colonna 'Household Head Occupation (Categories)
181 if "Household Head Occupation" in df.columns:
182     df.loc[:, 'Household Head Occupation (Categories)'] = df['
    Household Head Occupation'].map(occupation_to_category)
```

```
183     print(df[['Household Head Occupation', 'Household Head Occupation
        (Categories)']].head())
184 else:
185     print("La colonna 'Household Head Occupation' non esiste nel
        dataframe.")
186
187 # Mappatura delle categorie di livelli di istruzione
188 grade_categories = {
189
190     'Elementary Education or below': ['No Grade Completed','Preschool
        ','Grade 1','Grade 2','Grade 3','Grade 4','Grade 5','Grade 6','
        Elementary Graduate',],
191
192     'Secondary Education': ['First Year High School','Second Year
        High School','Third Year High School','High School Graduate'],
193
194     'Technical Education and Vocational Training': ['Post
        Baccalaureate',
195                                                     'Other Programs in
        Education at the Third Level, First Stage, of the Type that Leads
        to an Award not Equivalent to a First University or Baccalaureate
        Degree',
196                                                     'Other Programs of
        Education at the Third Level, First Stage, of the Type that Leads
        to a Baccalaureate or First University/Professional Degree (HIgher
         Education Level, First Stage, or Collegiate Education Level)'],
197
198     'Higher Education (College/University)': ['First Year College','
        First Year Post Secondary','Second Year College','Second Year Post
         Secondary',
199                                                     'Third Year College','
        Fourth Year College'],
200
201     'Master\'s Degree and Programs': ['Agriculture, Forestry, and
        Fishery Programs','Architecture and Building Programs','Arts
        Programs','Basic Programs',
202                                 'Business and Administration Programs','
        Computing/Information Technology Programs','Engineering and
        Engineering Trades Programs',
203                                 'Engineering and Engineering trades Programs
        ','Environmental Protection Programs','Health Programs','
        Humanities Programs',
204                                 'Journalism and Information Programs','Law
        Programs','Life Sciences Programs','Manufacturing and Processing
        Programs',
205                                 'Mathematics and Statistics Programs','
        Personal Services Programs','Physical Sciences Programs','Security
         Services Programs',
```

```
206                         'Social Services Programs','Social and
    Behavioral Science Programs','Teacher Training and Education
    Sciences Programs',
207                         'Transport Services Programs','Veterinary
    Programs'],
208     }
209
210
211 grade_to_category = {grade: category for category, grades in
    grade_categories.items() for grade in grades}
212
213 df = df.copy()
214
215 # Creazione 'Household Head Highest Grade Completed (Categories)'
216 if "Household Head Highest Grade Completed" in df.columns:
217     df.loc[:, 'Household Head Highest Grade Completed (Categories)']
    = df['Household Head Highest Grade Completed'].map(
    grade_to_category)
218     print(df[['Household Head Highest Grade Completed', 'Household
    Head Highest Grade Completed (Categories)']].head())
219 else:
220     print("La colonna 'Household Head Highest Grade Completed' non
    esiste nel dataframe.")
221
222 # Rimozione delle colonne originali
223 df.drop(['Household Head Occupation', 'Household Head Highest Grade
    Completed'], axis=1, inplace=True)
224
225 # Ricerca variabili categoriche
226 categorical_columns = df.select_dtypes(include=['object']).columns
227
228 # Visualizzazione del numero di valori distinti per ogni variabile
    categorica
229 print("Numero di valori distinti per variabile categorica:")
230 for col in categorical_columns:
231     print(f"{col}: {df[col].nunique()}")
232
233 # Ricerca dei missing values
234 missing_values_count = df.isnull().sum()
235 print("\nConteggio dei missing values per colonna:")
236 print(missing_values_count)
237
238 # Ricerca e stampa righe con almeno un missing value
239 rows_with_missing = df[df.isnull().any(axis=1)]
240
241 if not rows_with_missing.empty:
242     print("\nRiga contenente almeno un missing value:")
243     print(rows_with_missing.iloc[0])
244 else:
```

```
245     print("\nNon ci sono righe con missing values.")
246
247 # Sostituzione missing values con il valore modale della colonna
248 for column in df.columns:
249     if df[column].isnull().any():
250         most_frequent_value = df[column].mode()[0]
251         df.loc[df[column].isnull(), column] = most_frequent_value
252
253 # Verifica che non ci siano più missing values
254 print("\nConteggio dei missing values dopo la sostituzione:")
255 print(df.isnull().sum())
256
257 def plot_histograms_seaborn(data, columns, stage):
258     fig, axes = plt.subplots(nrows=1, ncols=len(columns), figsize
    =(12, 4))
259     fig.suptitle(f'Istogrammi dopo {stage}')
260     for ax, col in zip(axes, columns):
261         sns.histplot(data[col], kde=False, ax=ax)
262         ax.set_title(col)
263         ax.set_xlabel('Valore')
264         ax.set_ylabel('Frequenza')
265     plt.tight_layout(rect=[0, 0.03, 1, 0.95])
266     plt.show()
267
268 # Trasformazione logaritmica
269 df['Household Head Age'] = np.log1p(df['Household Head Age'])
270 df['House Floor Area'] = np.log1p(df['House Floor Area'])
271 df['House Age'] = np.log1p(df['House Age'])
272
273 # Distribuzione post trasformazione logaritmica
274 plot_histograms_seaborn(df, ['Household Head Age', 'House Floor Area'
    , 'House Age'], 'Trasformazione Logaritmica')
275
276 # Rimozione outliers usando l'IQR
277 Q1 = df[['Household Head Age', 'House Floor Area', 'House Age']].
    quantile(0.25)
278 Q3 = df[['Household Head Age', 'House Floor Area', 'House Age']].
    quantile(0.75)
279 IQR = Q3 - Q1
280 df = df[~((df[['Household Head Age', 'House Floor Area', 'House Age'
    ]] < (Q1 - 1.5 * IQR)) | (df[['Household Head Age', 'House Floor
    Area', 'House Age']] > (Q3 + 1.5 * IQR))).any(axis=1)]
281
282 df = df.copy()
283
284 # Standardizzazione Z-score
285 df[['Household Head Age', 'House Floor Area', 'House Age']] = stats.
    zscore(df[['Household Head Age', 'House Floor Area', 'House Age'
    ]])
```

```
286
287 # Distribuzione post standardizzazione
288 plot_histograms_seaborn(df, ['Household Head Age', 'House Floor Area'
        , 'House Age'], 'Standardizzazione Z-score')
289
290 # Definizione soglia di povertà in PHP
291 poverty_threshold = 22747
292 double_pt = poverty_threshold * 2
293
294 # Trasformazione 'Total Household Income' in una variabile categorica
295 df['Income Category'] = pd.cut(df['Total Household Income'],
296                                bins=[0, poverty_threshold, double_pt,
        df['Total Household Income'].max()],
297                                labels=['Under Poverty Treshold', '1-2
    x Poverty Treshold', 'Over 2x Poverty Treshold'],
298                                include_lowest=True)
299
300
301 print(df[['Total Household Income', 'Income Category']].head())
302
303 # Calcolo della distribuzione delle categorie di reddito
304 income_category_distribution = df['Income Category'].value_counts()
305
306 # Calcolo della percentuale di ciascuna categoria
307 income_category_percentage = df['Income Category'].value_counts(
        normalize=True) * 100
308
309
310 print("Distribuzione delle categorie di reddito:")
311 print(income_category_distribution)
312 print("\nPercentuale di ciascuna categoria di reddito:")
313 print(income_category_percentage)
314
315 # Mappatura delle etichette
316 income_category_mapping = {'Under Poverty Treshold': 0, '1-2x Poverty
        Treshold': 1, 'Over 2x Poverty Treshold': 2}
317 df['Income Category'] = df['Income Category'].map(
        income_category_mapping)
318
319 # Calcolo del numero di esempi nella classe maggioritaria
320 majority_class_count = df['Income Category'].value_counts().max()
321
322 # Strategie di sampling
323 over_strategy = {0: majority_class_count // 2, 1:
        majority_class_count // 2}  # Over-sampling delle classi
        minoritarie
324 under_strategy = {2: majority_class_count // 2}  # Under-sampling
        della classe maggioritaria
325
```

```
326 # Creazione della pipeline di sampling
327 over = SMOTE(sampling_strategy=over_strategy)
328 under = RandomUnderSampler(sampling_strategy=under_strategy)
329 pipeline = Pipeline([('over', over), ('under', under)])
330
331 # Preparazione delle features (X) e la target (y)
332 X = df.drop(['Income Category', 'Total Household Income'], axis=1)  #
        Escludere la colonna target e 'Total Household Income'
333 y = df['Income Category']
334
335 # Codifica delle variabili categoriche in X
336 for col in X.select_dtypes(include=['object']).columns:
337     if len(X[col].unique()) == 2:
338         #LabelEncoder per variabili binarie
339         le = LabelEncoder()
340         X[col] = le.fit_transform(X[col])
341     else:
342         #OneHotEncoder per le altre variabili categoriche
343         X = pd.get_dummies(X, columns=[col], drop_first=True)
344
345 # Applicazione della pipeline di sampling
346 X_resampled, y_resampled = pipeline.fit_resample(X, y)
347
348 # Conversione di y_resampled in un DataFrame per la concatenazione al
        dataset
349 y_resampled_df = pd.DataFrame(y_resampled, columns=['Income Category'
    ])
350
351 # Creazione nuovo DataFrame con i dati bilanciati
352 balanced_df = pd.concat([pd.DataFrame(X_resampled, columns=X.columns)
    , y_resampled_df], axis=1)
353
354 # Visualizzazione della distribuzione delle categorie in 'Income
    Category' post sampling
355 print("Distribuzione delle categorie di 'Income Category' nel
    DataFrame bilanciato:")
356 print(balanced_df['Income Category'].value_counts())
357
358 balanced_df.to_csv('Preprocessed_2_Family_Income_and_Expenditure.csv'
    , index=False)
```

A.10 Predizione del THI mediante modelli di Regressione

Python_Code/THI_prediction_with_Regression.py

```
import pandas as pd
import numpy as np
import lightgbm as lgb
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from lightgbm import LGBMRegressor
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import RandomizedSearchCV
from sklearn.metrics import r2_score

df = pd.read_csv('Preprocessed_1_Family_Income_and_Expenditure.csv')

# Selezione delle variabili indipendenti
expenditure_cols = [col for col in df.columns if 'Expenditure' in col
    or 'expenditures' in col]

X = df[expenditure_cols]
y = df['Total_Household_Income_log']

# Divisione del dataset in set di addestramento e test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
    =0.2, random_state=42)

# Regressione lineare
linear_reg = LinearRegression()
linear_scores = cross_val_score(linear_reg, X_train, y_train, cv=5,
    scoring='neg_mean_squared_error')
linear_rmse_scores = np.sqrt(-linear_scores)
print("Linear Regression RMSE:", linear_rmse_scores.mean())

# Calcolo R^2 per la Regressione Lineare
linear_reg.fit(X_train, y_train)
y_pred_linear = linear_reg.predict(X_test)
r2_linear = r2_score(y_test, y_pred_linear)
print(f"R^2 per la Regressione Lineare: {r2_linear:.4f}")

# Random Forest Regressor con fine tuning
param_distrib_rf = {
    'n_estimators': np.arange(50, 100, 200),
    'max_features': ['sqrt', 'log2'],
    'max_depth': [10, 20, 30, None],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

forest_reg = RandomForestRegressor()

```

```
47 # Ricerca dei parametri ottimali del Random Forest Regressor con
       RandomizedSearchCV e addestramento del modello
48 random_search_rf = RandomizedSearchCV(forest_reg, param_distributions
       =param_distrib_rf, n_iter=15, cv=5,
49                                        scoring='neg_mean_squared_error'
       ,
50                                        return_train_score=True,
       random_state=42)
51
52 random_search_rf.fit(X_train, y_train)
53 best_forest_reg = random_search_rf.best_estimator_
54 print("Migliori parametri trovati per Random Forest Regressor:",
       random_search_rf.best_params_)
55
56 # Valutazione performance del Random Forest Regressor ottimizzato
57 forest_scores_opt = cross_val_score(best_forest_reg, X_train, y_train
       , cv=5, scoring='neg_mean_squared_error')
58 forest_rmse_scores_opt = np.sqrt(-forest_scores_opt)
59 print("Random Forest (ottimizzato):", forest_rmse_scores_opt.mean())
60
61 # Calcolo R^2 per RandomForestRegressor
62 best_forest_reg.fit(X_train, y_train)
63 y_pred_forest = best_forest_reg.predict(X_test)
64 r2_forest = r2_score(y_test, y_pred_forest)
65 print(f"R^2 per Random Forest Regressor: {r2_forest:.4f}")
66
67 # LightGBM
68 # Definizione dei parametri per LightGBM
69 param_distrib_lgbm = {
70     'n_estimators': np.arange(50, 201, 10),
71     'learning_rate': [0.001, 0.01, 0.05, 0.1, 0.5],
72     'max_depth': [-1, 10, 20, 30, 40],
73     'num_leaves': np.arange(20, 61, 10),
74     'min_child_samples': [5, 10, 20, 30],
75     'subsample': [0.5, 0.6, 0.7, 0.8, 0.9, 1.0],
76     'colsample_bytree': [0.5, 0.6, 0.7, 0.8, 0.9, 1.0],
77     'reg_alpha': [0, 0.1, 0.5, 1],
78     'reg_lambda': [0, 0.1, 0.5, 1]
79 }
80
81 lgbm_reg = lgb.LGBMRegressor(force_col_wise=True, verbose = -1)
82
83 # Ricerca dei parametri ottimali del LightGBM Regressor con
       RandomizedSearchCV e addestramento del modello
84 random_search_lgbm = RandomizedSearchCV(lgbm_reg, param_distributions
       =param_distrib_lgbm, n_iter=10, cv=5,
85                                         scoring='
       neg_mean_squared_error',
```

```
                                                return_train_score=True,
    random_state=42)

random_search_lgbm.fit(X_train, y_train)
best_lgbm_reg = random_search_lgbm.best_estimator_
print("Migliori parametri trovat per LightGBM Regressor:",
    random_search_lgbm.best_params_)

# Valutazione della performance del LightGBM ottimizzato
lgbm_scores_opt = cross_val_score(best_lgbm_reg, X_train, y_train, cv
    =5, scoring='neg_mean_squared_error')
lgbm_rmse_scores_opt = np.sqrt(-lgbm_scores_opt)
print("LightGBM (ottimizzato):", lgbm_rmse_scores_opt.mean())

# Calcolo R^2 per LightGBM Regressor
best_lgbm_reg.fit(X_train, y_train)  # Usa il modello con i parametri
     ottimizzati se li hai
y_pred_lgbm = best_lgbm_reg.predict(X_test)
r2_lgbm = r2_score(y_test, y_pred_lgbm)
print(f"R^2 per LightGBM Regressor: {r2_lgbm:.4f}")


# Creazione grafico parametri predetti vs parametri reali
def plot_predictions(y_real, y_pred, title):
    plt.scatter(y_real, y_pred, alpha=0.5)
    plt.title(title, weight='bold')
    plt.xlabel('Valori reali del log del THI')
    plt.ylabel('Valori predetti del log del THI')
    plt.plot([min(y_real), max(y_real)], [min(y_real), max(y_real)],
    color='red')  # Linea di predizione perfetta
    plt.grid(True)
    plt.show()


plot_predictions(y_test, y_pred_linear, 'Linear Regressor')
plot_predictions(y_test, y_pred_forest, 'Random Forest Regressor')
plot_predictions(y_test, y_pred_lgbm, 'LightGBM Regressor')
```

A.11 Predizione della fascia di reddito di appartenenza delle famiglie filippine con modelli di Classificazione

Python_Code/Income_bracket_prediction.py

```
import pandas as pd
import numpy as np
```

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, RandomizedSearchCV, cross_val_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from lightgbm import LGBMClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix

df = pd.read_csv('Preprocessed_2_Family_Income_and_Expenditure.csv')
X = df.drop('Income Category', axis=1)
y = df['Income Category']

# Pulizia dei nomi delle caratteristiche per far funzionare il LightGBM
X.columns = [col.replace(' ', '_').replace('{', '_').replace('}', '_').replace(':', '_').replace(',', '_') for col in X.columns]

# Divisione dati in set di addestramento e test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Definizione dei modelli selezionati
models = {
    "Random Forest": RandomForestClassifier(),
    #"SVC": SVC(probability=True),
    "LightGBM": LGBMClassifier(verbose = -1)
}

# Parametri selezionati per la Randomized Search (selezione iperparametri ottimali)
param_dist = {
    "Random Forest": {'n_estimators': [100, 200, 300], 'max_features': ['sqrt', 'log2'], 'max_depth': [10, 20, 30],
                      'min_samples_split': [2, 5, 10], 'min_samples_leaf': [1, 2, 4], 'bootstrap': [True, False],},
    #"SVC": {'C': [1, 10, 50], 'gamma': [0.01, 0.001], 'kernel': ['rbf']},
    "LightGBM": {'n_estimators': [100, 200, 300], 'learning_rate': [0.01, 0.1, 0.5], 'max_depth': [3, 5, 7],
                 'num_leaves': [31, 50, 70], 'min_child_samples': [20, 30, 40], 'subsample': [0.8, 1.0],
                 'colsample_bytree': [0.8, 1.0]}
}
results = {}

for model_name, model in models.items():
```

```
40      randomized_search = RandomizedSearchCV(model, param_dist[
    model_name], n_iter=6, cv=5, random_state=42, n_jobs=-1)
41      randomized_search.fit(X_train, y_train)
42      best_model = randomized_search.best_estimator_
43      best_score = randomized_search.best_score_  # Media dei punteggi
    di cross-validation per la miglior combinazione di iperparametri
44
45      y_pred = best_model.predict(X_test)
46
47      # Calcolo delle metriche per ogni classe (ad eccezione dell'
    accuracy che è totale)
48      accuracy = accuracy_score(y_test, y_pred)
49      precision_per_class = precision_score(y_test, y_pred, average=
    None)
50      recall_per_class = recall_score(y_test, y_pred, average=None)
51      f1_per_class = f1_score(y_test, y_pred, average=None)
52      conf_matrix = confusion_matrix(y_test, y_pred)
53
54      # Calcolo della media delle metriche
55      precision_avg = precision_score(y_test, y_pred, average='macro')
56      recall_avg = recall_score(y_test, y_pred, average='macro')
57      f1_avg = f1_score(y_test, y_pred, average='macro')
58
59      results[model_name] = {
60          'Best Parameters': randomized_search.best_params_,
61          'Cross-Validation Score': best_score,
62          'Accuracy': accuracy,
63          'Precision per Class': precision_per_class,
64          'Recall per Class': recall_per_class,
65          'F1 Score per Class': f1_per_class,
66          'Average Precision': precision_avg,
67          'Average Recall': recall_avg,
68          'Average F1 Score': f1_avg,
69          'Confusion Matrix': conf_matrix
70      }
71
72  # Stampa dei risultati
73  for model_name, metrics in results.items():
74      print(f"Results for {model_name}:")
75      for metric, value in metrics.items():
76          if isinstance(value, np.ndarray):
77              print(f"{metric}:")
78              for i, val in enumerate(value):
79                  print(f"    Classe {i}: {val}")
80          else:
81              print(f"{metric}: {value}")
82      print("\n")
83
84  def plot_confusion_matrix(conf_matrix, model_name):
```

```
85      plt.figure(figsize=(10, 7))
86      sns.heatmap(conf_matrix, annot=True, fmt='g', cmap='Reds', cbar=
        True)
87      plt.xlabel('Predicted Income Category', weight = 'bold')
88      plt.ylabel('True Income Category', weight = 'bold')
89      plt.title(f'Confusion Matrix for {model_name}', weight = 'bold')
90      plt.show()
91
92  # Matrice di confusione per ogni modello
93  for model_name, metrics in results.items():
94      conf_matrix = metrics['Confusion Matrix']
95      plot_confusion_matrix(conf_matrix, model_name)
```

Bibliografia

[1] Action Aid. *Povertà nel mondo: si vive con due dollari al giorno?* Available at `https://www.actionaid.it/informati/notizie/poverta-mondo` (2023) (cit. a p. 1).

[2] Nazioni Unite. *Obiettivo 1: Porre fine ad ogni forma di povertà nel mondo.* Available at `https://unric.org/it/obiettivo-1-porre-fine-ad-ogni-forma-di-poverta-nel-mondo/` (cit. a p. 1).

[3] Fondazione Fontana World Social Agenda. *Povertà.* Available at `https://www.worldsocialagenda.org/1.1-Poverta//` (cit. a p. 1).

[4] A Alsharkawi, M Al-Fetyani, M Dawas, H Saadeh e M Alyaman. «Poverty Classification Using Machine Learning: The Case of Jordan». In: *Sustainability* 13.3 (2021) (cit. alle pp. 2, 27–29).

[5] Aziza Usmanova, Ahmed Aziz, Dilshodjon Rakhmonov e Walid Osamy. «Utilities of Artificial Intelligence in Poverty Prediction: A Review». In: *Sustainability* 14.21 (2022), pp. 1–39 (cit. alle pp. 2, 27–32).

[6] Chris Browne, David S Matteson, Linden McBride, Leiqiu Hu, Yanyan Liu, Ying Sun, Jiaming Wen e Christopher B Barrett. «Multivariate random forest prediction of poverty and malnutrition prevalence». In: *PLoS One* 16.9 (2021) (cit. alle pp. 2, 27, 28).

[7] Neal Jean, Marshall Burke, Michael Xie, W. Matthew Davis et al. «Combining satellite imagery and machine learning to predict poverty». In: *Science* 353.6301 (2016), pp. 790–794. DOI: `10.1126/science.aaf7894` (cit. alle pp. 2, 27).

[8] Guie Li, Zhongliang Cai, Yun Qian e Fei Chen. «Identifying Urban Poverty Using High-Resolution Satellite Imagery and Machine Learning Approaches: Implications for Housing Inequality». In: *Land* 10.6 (2021) (cit. a p. 2).

[9] J. E. Steele, P. R. Sundsøy, C. Pezzulo, V. A. Alegana, T. J. Bird, J. Blumenstock et al. «Mapping poverty using mobile phone and satellite data». In: *Journal of The Royal Society Interface* 14.127 (2017) (cit. a p. 2).

[10] Usama Fayyad, Gregory Piatetsky-Shapiro e Padhraic Smyth. «From Data Mining to Knowledge Discovery in Databases». In: *AI Magazine* 17.3 (1996), pp. 37–54 (cit. alle pp. 4–6).

[11] Jiawei Han, Jian Pei e Micheline Kamber. *Data Mining: Concepts and Techniques*. Elsevier, 2011 (cit. alle pp. 4, 7, 9).

[12] *KDD Process in Data Mining*. Available at `https://www.geeksforgeeks.org/kdd-process-in-data-mining/` (cit. a p. 5).

[13] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977 (cit. a p. 6).

[14] I. H. Witten, E. Frank, M. A. Hall e C. J. Pal. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2011 (cit. a p. 6).

[15] J. Heer e B. Shneiderman. «Interactive dynamics for visual analysis». In: *Queue* 10.2 (2012), p. 30 (cit. a p. 6).

[16] I. T. Jolliffe e J. Cadima. «Principal component analysis: a review and recent developments». In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374.2065 (2016), p. 20150202 (cit. a p. 6).

[17] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola e R. C. Williamson. «Estimating the Support of a High-Dimensional Distribution». In: *Neural Computation* 13.7 (2001), pp. 1443–1471 (cit. a p. 6).

[18] *Correlazione di Pearson*. Available at `https://it.wikipedia.org/wiki/Indice_di_correlazione_di_Pearson` (cit. alle pp. 7, 69).

[19] *Data Mining*. Available at `https://www.geeksforgeeks.org/data-mining/` (cit. a p. 8).

[20] *Data Mining:cos'è, significato, tecniche ed esempi*. Available at `https://www.bigdata4innovation.it/data-science/data-mining/data-mining-cose-perche-conviene-utilizzarlo-e-quali-sono-le-attivita-tipiche/` (cit. a p. 8).

[21] Andrea Pasini, Flavio Giobergia e Elena Baralis. *Data Mining: le tipologie di analisi*. Available at `https://www.businessintelligencegroup.it/le-3-tipologie-di-analisi-dei-big-data-descrittive-predittive-e-prescrittive/` (cit. a p. 9).

[22] *Data Mining Techniques*. Available at `https://www.geeksforgeeks.org/data-mining-techniques/` (cit. alle pp. 9, 82).

[23] N.R. Draper e H. Smith. *Applied Regression Analysis*. Wiley, 1998 (cit. a p. 13).

[24] D.C. Montgomery, E.A. Peck e G.G. Vining. *Introduction to Linear Regression Analysis*. Wiley, 2012 (cit. a p. 13).

[25] Leo Breiman. «Random forests». In: *Machine learning* 45.1 (2001), pp. 5–32 (cit. a p. 14).

[26] *RandomForestRegressor - ScikitLearn*. Available at `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html` (cit. alle pp. 15, 16, 82, 83).

[27] *RandomForestClassifier - ScikitLearn*. Available at `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html` (cit. alle pp. 16, 17, 83).

[28] J. H. Friedman. «Greedy function approximation: A gradient boosting machine». In: *Annals of Statistics* 29.5 (2001), pp. 1189–1232 (cit. a p. 17).

[29] *XGBoost vs LightGBM*. Available at `https://neptune.ai/blog/xgboost-vs-lightgbm` (cit. alle pp. 18, 19).

[30] T. Hastie, R. Tibshirani e J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer New York Inc., 2009 (cit. alle pp. 18, 22).

[31] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye e Tie-Yan Liu. «LightGBM: A highly efficient gradient boosting decision tree». In: *Advances in neural information processing systems*. 2017, pp. 3146–3154 (cit. a p. 19).

[32] *Introduction to Support Vector Machines (SVM)*. Available at `https://www.geeksforgeeks.org/introduction-to-support-vector-machines-svm/` (cit. a p. 20).

[33] *Support Vector Machines - IBM*. Available at https://www.ibm.com/docs/it/spss-modeler/18.4.0?topic=node-svm-expert-options (cit. a p. 21).

[34] *Support Vector Machines - ScikitLearn*. Available at `https://scikit-learn.org/stable/modules/svm.html` (cit. alle pp. 21, 83).

[35] Tania Cerquitelli e Elena Baralis. *Regression Analysis: Fundamentals*. Available at `https://dbdmg.polito.it/dbdmg_web/wp-content/uploads/2022/11/Regression-Analysis.pdf` (cit. alle pp. 22, 23).

[36] Elena Baralis. *Classification Analysis: Fundamentals*. Available at `https://dbdmg.polito.it/dbdmg_web/wp-content/uploads/2021/10/8-DMClassification.pdf` (cit. alle pp. 23–26).

[37] Ron Kohavi. «A study of cross-validation and bootstrap for accuracy estimation and model selection». In: *IJCAI*. Vol. 14. 2. 1995, pp. 1137–1145 (cit. a p. 23).

[38] D. M. Powers. «Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation». In: *Journal of Machine Learning Technologies* 2.1 (2011), pp. 37–63 (cit. a p. 25).

[39] N. S. Sani, M. A. Rahman, A. A. Bakar, S. Sahran e H. M. Sarim. «Machine learning approach for bottom 40 percent households (B40) poverty classification». In: *International Journal of Advanced Science, Engineering and Information Technology* 8 (2018), p. 1698 (cit. alle pp. 27–29).

[40] Azuraliza Abu Bakar, Rusnita Hamdan e Nor Samsiah Sani. «Ensemble learning for multidimensional poverty classification». In: *Sains Malaysiana* 49.2 (2020), pp. 447–459 (cit. a p. 27).

[41] Francis Paul Flores. *Filipino Family Income and Expenditure*. Available at `https://www.kaggle.com/datasets/grosvenpaul/family-income-and-expenditure?datasetId=2823&sortBy=voteCount` (2017) (cit. alle pp. 33, 35, 40).

[42] *Filippine*. Available at `https://it.wikipedia.org/wiki/Filippine` (cit. alle pp. 34, 35, 51).

[43] *Notebook: Exploratory Data Analysis*. Available at `https://www.kaggle.com/code/issatingzon/exploratory-data-analysis/notebook` (2017) (cit. a p. 40).

[44] *Report: The Real Filipino Family*. Available at `https://www.kaggle.com/code/brenborbs/the-real-filipino-family/report` (2017) (cit. a p. 40).

[45] *Notebook: FamilyIncomeAndExpenditure*. Available at `https://www.kaggle.com/code/enmayordomo/familyincomeandexpenditure/notebook` (2020) (cit. a p. 40).

[46] *Rapidminer*. Available at `https://rapidminer.com/` (cit. alle pp. 41, 68).

[47] *Che cosa è il data mining?* Available at `https://www.trendmicro.com/it_it/what-is/machine-learning/data-mining.html` (2023) (cit. a p. 41).

[48] *Google Colaboratory*. Available at `https://colab.google/` (cit. alle pp. 41, 68).

[49] *Regione Capitale Nazionale*. Available at `https://it.wikipedia.org/wiki/Regione_Capitale_Nazionale` (cit. a p. 42).

[50] *Regione Autonoma nel Mindanao Musulmano*. Available at `https://it.wikipedia.org/wiki/Regione_Autonoma_nel_Mindanao_Musulmano` (cit. a p. 43).

[51] *Standardizzazione Z-Score*. Available at `https://it.wikipedia.org/wiki/Standardizzazione_(statistica)` (cit. a p. 71).

[52] *Scarto Interquartile, IQR*. Available at `https://it.wikipedia.org/wiki/Scarto_interquartile` (cit. a p. 71).

[53] Philippines Statistics Authority. *Updated 2015 and 2018 Full Year Poverty Statistics*. Available at `https://psa.gov.ph/statistics/poverty/stat-tables/released/Updated%202015%20and%202018%20Full%20Year%20Poverty%20Statistics` (cit. a p. 80).

[54] Swatsik Satpathy. *SMOTE for Imbalanced Classification with Python*. Available at `https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/` (cit. a p. 81).

[55] *One hot encoding*. https://www.geeksforgeeks.org/ml-one-hot-encoding-of-datasets-in-python/ (cit. a p. 81).

[56] *Regression - ScikitLearn*. Available at `https://dbdmg.polito.it/dbdmg_web/wp-content/uploads/2021/11/8-ScikitLearn-Regression.pdf` (cit. a p. 82).

[57] *LightGBMRegressor - ScikitLearn*. Available at `https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMRegressor.html` (cit. alle pp. 82, 83).

[58] *LGBMClassifier - ScikitLearn*. Available at `https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier.html` (cit. a p. 83).

[59] *RandomizedSearchCV - ScikitLearn*. Available at `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html` (cit. a p. 85).