# POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Informatica


Tesi di Laurea Magistrale

# Classificatori associativi per dati spaziali-temporali

Supervisori

Prof. Paolo GARZA

Dr. Luca COLOMBA

Candidato

Giuseppe MOSCARELLI

Aprile, 2022

# Sommario

Il Bike Sharing è un'iniziativa nata con lo scopo di ridurre le emissioni dei gas serra, soprattutto nelle grandi città. Infatti, l'idea su cui si basa questo sistema è quello di spingere i cittadini a condurre uno stile di vita più ecologico e sostenibile, evitando di utilizzare i propri mezzi per spostamenti lungo brevi tratti. Negli ultimi anni questo settore ha avuto una forte crescita, in quanto offre soluzioni abbastanza economiche e pratiche che si coniugano molto bene con il nuovo paradigma della "smart mobility".

Questa tesi ha come obiettivo lo sviluppo di un classificatore associativo capace di predire degli eventi caratterizzati da informazioni spaziali e temporali, in modo tale da rilevare anticipatamente delle situazioni critiche riguardanti lo stato di alcune stazioni di bike sharing.

Le regole di associazione su cui si basa tale classificatore sono state estratte da un dataset contenente dati relativi a stazioni della città di San Francisco. La scelta dell'utilizzo delle regole di associazione è stata basata sul fatto che esse forniscono un mezzo per l'estrazione di relazioni rilevanti tra i dati in modo facilmente interpretabile.

Il classificatore ottenuto potrebbe essere utilizzato in applicazioni reali in modo da rilevare e risolvere anticipatamente ed in maniera mirata delle situazioni critiche, migliorando così l'efficienza del servizio fornito ed ottimizzando la gestione delle risorse da parte delle aziende.

# Ringraziamenti

Prima di procedere con la trattazione, vorrei dedicare qualche riga a tutti coloro che mi sono stati vicini in questo percorso di crescita personale e professionale.

Un sentito grazie va al mio relatore Paolo Garza ed il mio correlatore Luca Colomba per la vostra infinita disponibilità, nonostante la distanza, e per la vostra pazienza e tempestività ad ogni mia richiesta. Grazie per avermi fornito tutto il necessario per portare a termine questo elaborato nel migliore dei modi.

Ringrazio infinitamente i miei genitori, mio fratello, mia sorella e tutti i miei familiari. Grazie per avermi sempre sostenuto, moralmente ed economicamente, senza indugi lungo tutto il mio percorso di studi. Se sono giunto a questo traguardo è soprattutto grazie a voi.

Un grazie di cuore va a tutti i miei amici che sono mi sono stati costantemente vicini nei momenti di sconforto e che hanno gioito, insieme a me, nel conseguimento graduale dei miei obiettivi personali ed accademici. Il vostro sostegno è stato per me di fondamentale importanza.

Ringrazio, infine, tutti i colleghi conosciuti durante il mio percorso universitario, con i quali ho avuto la fortuna di instaurare un bel rapporto di amicizia e di condividere la mia esperienza accademica ed i traguardi raggiunti. Senza i vostri consigli ed il vostro supporto, non ce l'avrei mai fatta.

Grazie infinite a tutti voi.

# Indice

# Elenco delle tabelle

# Elenco delle figure

# Acronimi

**KDD**

Knowledge Discovery from Data

**PrefixSpan**

**Prefix**-projected **S**equential **pa**ttern mining

**RSS**

Residual Squared Sum

**PST**

Pacific Standard Time (Nord America)

**MSE**

Mean Squared Error

# Capitolo 1

# Introduzione

## 1.1 Contesto

Il Bike Sharing è un'iniziativa nata con lo scopo principale di ridurre le emissioni dei gas serra, ma anche quello di ridurre la congestione del traffico ed apportare dei benefici per la salute dei cittadini. Infatti, l'idea su cui si basa questo sistema è quella di spingere i cittadini a condurre uno stile di vita più ecologico e sostenibile, evitando di utilizzare i propri mezzi per spostamenti lungo brevi tratti. Il maggior utilizzo delle bici, infatti, è volto a coprire gli spostamenti casa-lavoro o comunque tra punti di interesse vicini tra di loro.

Esistono diversi sistemi di bike sharing: alcuni prevedono la presenza di stazioni fisse di raccolta delle bici; altri, invece, permettono agli utenti di trovare e depositare le bici, una volta terminata la corsa, in qualsiasi punto della città.

Negli ultimi anni, il settore del bike sharing ha avuto una repentina diffusione fino a diventare, al giorno d'oggi, una soluzione largamente utilizzata in quasi tutto il mondo. Quella del bike sharing, infatti, è stata un'importante iniziativa che ha offerto soluzioni abbastanza economiche e pratiche che bene si coniugano con il nuovo paradigma della "smart mobility", ma anche con la sempre più pressante necessità di trovare delle alternative sostenibili per l'ambiente.

## 1.2 Obiettivo della tesi

L'obiettivo della tesi è quello di capire se un classificatore associativo, ovvero un classificatore basato su delle regole di associazione, possa essere adatto nel fare predizioni su degli eventi caratterizzati da informazioni spaziali e temporali. La scelta di utilizzare un classificatore associativo è stata basata sul fatto che

le regole di associazione rappresentano una tecnica di analisi dei dati facilmente interpretabile e che permette di estrarre relazioni di alta importanza tra i dati di interesse. Queste relazioni saranno quindi utili nel rilevare quelle situazioni più frequenti che saranno utilizzate infine per effettuare la predizione.

Il dataset di partenza utilizzato contiene dei dati rappresentanti delle informazioni relative a delle stazioni di bike sharing situate nella città di San Francisco e alcune città limitrofe. Tuttavia, dopo alcune analisi e considerazioni, ho ritenuto opportuno focalizzarmi soltanto sulle stazioni relative alla città di San Francisco e scartare quelle situate nelle altre città. In particolare, il dataset di interesse è un file di log contenente, per ogni stazione, il numero di bici disponibili e di slot liberi in ogni istante di tempo.

Il formato di tale dataset non è adatto all'estrazione delle regole di associazione, quindi si è proceduto ad apportare le modifiche necessarie ad ottenere un formato più appropriato. Dopo di che, è stato utilizzato un algoritmo per l'estrazione delle regole di associazione che saranno utilizzate dal classificatore associativo per effettuare la predizione.

Una volta ottenuto il classificatore associativo si è proceduto ad una fase di test in modo da valutarne le prestazioni e confrontarle con quelle di alcuni algoritmi di classificazione tradizionali. È stato così possibile valutare eventuali punti di forza e/o di debolezza del classificatore associativo ottenuto.

Questo lavoro potrebbe essere utile in applicazioni reali riguardanti il bike sharing. Per esempio, sarebbe possibile migliorare l'efficienza del servizio di ridistribuzione delle bici, andando a rilevare anticipatamente delle situazioni critiche per una determinata stazione e quindi di risolverle in anticipo ed in modo mirato. Questo permetterebbe di evitare eventuali disagi per i clienti, ma anche di gestire in maniera più efficiente e mirata le spese dovute alla ridistribuzione delle bici, evitando così all'azienda inutili uscite di denaro.

## 1.3 Organizzazione dell'elaborato

Il presente elaborato è strutturato in 5 diversi capitoli:

- **Capitolo 1:** contiene un'introduzione del lavoro svolto. Esso fornisce una descrizione generale del tipo di problema affrontato, l'obiettivo della tesi ed, infine, l'organizzazione del documento.

- **Capitolo 2:** presentazione dei componenti di base che sono stati utilizzati durante il lavoro (regole di associazione, classificatori tradizionali, ecc) ed i concetti teorici su cui essi si basano.

- **Capitolo 3:** descrizione generale della struttura dei dati utilizzati e del lavoro svolto partendo dalla manipolazione dei dati di partenza, fino all'ottenimento del classificatore associativo. Infine, si descrive il confronto effettuato tra il risultato dei test del classificatore ottenuto e quelli dei test di alcuni classificatori tradizionali.
- **Capitolo 4:** descrizione dettagliata dei dati specifici utilizzati e presentazione dei risultati ottenuti dai vari esperimenti svolti durante le diverse fasi del lavoro.
- **Capitolo 5:** enunciazione delle conclusioni con relativi accenni a possibili lavori futuri volti alla prosecuzione del presente lavoro svolto.

# Capitolo 2

# Concetti teorici di base

Nel presente capitolo verranno descritti i componenti di base utilizzati durante il lavoro ed i concetti teorici su cui essi si basano. Verrà inoltre fornita anche una panoramica generale delle tecniche e delle metodologie utilizzate per portare a compimento l'obiettivo della tesi.

## 2.1 Big Data

Il termine Big Data [1] indica genericamente una raccolta di dati la cui scala, diversità e complessità richiedono nuove architetture, tecniche, algoritmi e analisi per gestirli ed estrarre da essi valore e ed informazioni nascoste. La sfida del Big Data consiste, quindi, nel riuscire ad analizzare e mettere in relazione una grandissima mole di dati spesso anche eterogenei.

Le problematiche affrontate dal Big Data sono dunque:

- **Volume**: è necessario gestire enormi moli di dati;
- **Varietà**: i dati possono essere molto eterogenei tra di loro (strutturati, non strutturati);
- **Velocità**: è necessario gestire l'arrivo dell'enorme flusso di dati;
- **Veridicità**: i dati possono essere di bassa qualità o essere incerti;
- **Valore**: è l'aspetto più importante in quanto consiste nella trasformazione di un'informazione interessante in valore utile che potrà essere successivamente sfruttato.

Due dei possibili framework che è possibile utilizzare per la gestione dei Big Data sono MapReduce e Spark.

## 2.2 Data Mining

Il Data Mining [2] è l'insieme di metodologie e tecniche che hanno come obiettivo quello di estrarre delle informazioni nascoste e potenzialmente utili, a partire dai dati a disposizione. Le informazioni vengono estratte automaticamente, tramite l'utilizzo di alcuni algoritmi e vengono rappresentate tramite dei modelli astratti, denotati come *pattern*.

## 2.3 Knowledge Discovery from Data

Il processo di estrazione è chiamato Knowledge Discovery from Data (KDD) [2] (Figura 2.1) ed è composto da diverse fasi:

- **Selezione**: partendo dall'enorme quantità di dati a disposizione, si selezionano soltanto i dati di interesse, ovvero quelli contenenti delle informazioni rilevanti su cui si desidera andare a lavorare;
- **Preprocessing**: consiste in una serie di tecniche e procedure che hanno come obiettivo quello di migliorare la qualità dei dati iniziali. Infatti, i dati provenienti dal mondo esterno saranno molto "sporchi", ovvero di bassa qualità. Quindi, per estrarre dei pattern di buona qualità, sono necessarie alcune operazioni di pulizia, come per esempio: la rimozione degli outliers, la mitigazione degli effetti del rumore, la gestione di eventuali valori mancanti e la risoluzione di eventuali inconsistenze;
- **Trasformazione**: è una fase strettamente collegata alla fase di preprocessing ed è composta da una serie di tecniche consistenti nella trasformazione dei dati in una forma più congeniale alla procedura di mining, come per esempio l'eliminazione di una serie di features;
- **Estrazione**: è la fase che fa da fulcro al processo di data mining in quanto consiste in una serie di tecniche per l'estrazione di pattern contenenti delle informazioni utili precedentemente nascoste nei dati;
- **Interpretazione**: consiste nella fase di analisi e valutazione della qualità effettiva dei pattern estratti. Consiste quindi, nell'interpretare, con l'aiuto degli esperti di dominio, i pattern ottenuti per dare un senso a quello che è stato estratto.

Esistono diverse tecniche per l'estrazione di pattern nell'ambito del Data Mining. Esse possono essere suddivise in due categorie: **metodi descrittivi**, che estraggono modelli interpretabili descrivendo i dati; **metodi predittivi**, i quali, a partire da

un insieme noto di variabili, cercano di predire una variabile ignota (l'etichetta di classe).

Le principali tecniche di analisi, utilizzate per l'estrazione dei pattern, sono le seguenti:

- **Regole di associazione**: consiste nell'estrazione di correlazioni frequenti nascoste nei dati di interesse e che possono essere utilizzate per strategie particolari, come il marketing o decisioni sulle strategie di gestione di alcuni servizi aziendali;
- **Classificazione**: consiste nella predizione di un'etichetta di classe, ovvero nella predizione di eventi discreti, basandosi su una conoscenza pregressa. La classificazione ha quindi come obiettivo, quello di definire un *modello* (pattern astratto) che sarà usato dal classificatore per effettuare le predizioni su nuovi dati in input;
- **Clustering**: è una tecnica che ha come obiettivo quello di raggruppare i dati presenti nel dataset in diversi gruppi, detti *cluster*, in base ad una misura di somiglianza/similarità. I dati presenti all'interno dello stesso cluster saranno molto simili tra di loro e saranno diversi dai dati appartenenti ad altri cluster. A differenza della classificazione, in questo caso, non è necessaria una conoscenza pregressa;
- Altre tecniche: esistono diverse altre tecniche come le serie temporali o spaziali, riconoscimento delle eccezioni o la regressione, che consiste nella predizione di un valore continuo.

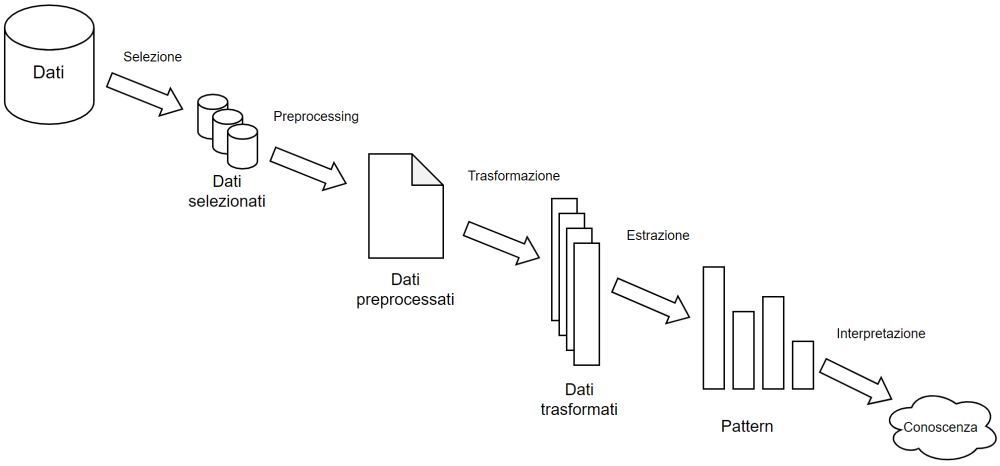



**Figura 2.1:** Processo del KDD

## 2.4 Regole di associazione

L'estrazione delle regole di associazione [3] è una delle tecniche utilizzate per l'analisi dei dati. Le regole di associazione, infatti, permettono di estrarre dei legami di co-occorrenza tra elementi all'interno dello stesso dataset. Un dataset è composto da un insieme di transazioni a loro volta composte da un insieme di item. Un esempio di dataset è mostrato nella Tabella 2.1.

| **ID Transazione** | **Items** |
|---|---|
| 1 | a,b,d |
| 2 | c,b,e |
| 3 | a,c,d |
| 4 | a,b,d,e |
| 5 | b,d,e |
| 6 | a,d |
| ... | ... |

**Tabella 2.1:** Esempio di dataset transazionale.

Solitamente gli item all'interno di una transazione non sono ordinati, come non lo sono le transazioni stesse tra di loro.

Una regola di associazione può essere rappresentata nella seguente forma:

$$A, B => C \tag{2.1}$$

dove A e B costituiscono il corpo (o antecedente) della regola, mentre C costituisce la testa (o conseguente) della regola. Il simbolo => indica una co-occorenza e non implica un nesso di causalità.

L'estrazione delle regole di associazione è una tecnica esplorativa che permette di descrivere la base dati transazionale per mezzo di pattern. I pattern estratti rappresentano un concetto più generale rispetto al particolare contenuto della base dati. Questo permette di applicare le regole di associazione ad ogni tipo di dataset. Le basi dati transazionali sono ovviamente diverse dalle normali basi dati strutturate. Esse, infatti, possono contenere record (rappresentati dalle transazioni) di lunghezza variabile e contengono item tutti dello stesso tipo. È possibile trasformare una base dati strutturata, che può essere pensata come una tabella, in una transazionale, semplicemente considerando ogni riga della tabella come una transazione ed ogni coppia attributo-valore come un item.

In definitiva, è possibile dire che le regole di associazione non cercano di effettuare delle predizioni, ma forniscono una descrizione del dataset di partenza.

### 2.4.1 Definizioni

Diamo adesso una serie di definizioni, in modo da rendere più semplice la comprensione dei concetti che saranno trattati nel seguito.

- **Itemset**: è una collezione di item;
- **K-itemset**: è la cardinalità dell'itemset;
- **SupportCount**: rappresenta il numero di volte in cui un itemset compare in una base dati transazionale;
- **Supporto**: è la frequenza relativa di un itemset. Esso è dato quindi dal rapporto tra il numero di transazioni contenenti sia il conseguente che l'antecedente ed il numero totale di transazioni che compongono la base dati transazionale. Data la regola di associazione A => B, in termini matematici il supporto può essere formulato come segue:

$$sup = \frac{\#\{A, B\}}{\#\ totale\ di\ transazioni} \tag{2.2}$$

  Quindi in pratica il supporto misura la probabilità a priori che capitino insieme antecedente e conseguente;
- **Itemset frequente**: è quell'itemset il cui supporto è maggiore o uguale ad un valore minimo di frequenza indicato con *minsup*;
- **Confidenza**: considerando la regola di associazione A => B, rappresenta la probabilità condizionata di B, dato A. espresso in altri termini, la confidenza esprima la probabilità di trovare il conseguente B, avendo trovato l'antecedente A. Essa può essere calcolata come:

$$conf = P(A|B) = \frac{sup\,(A, B)}{sup\,(A)} \tag{2.3}$$

### 2.4.2 Estrazione delle regole di associazione

Per estrarre le regole di associazione da un dataset, devono essere definiti determinati vincoli che guideranno la generazione delle regole risultanti. Le regole di associazione ottenute dovranno quindi rispettare determinati vincoli, quali:

- Supporto minimo (minsup): gli itemset che verranno mantenuti, chiamati itemset frequenti, saranno quelli con un valore di supporto maggiore o uguale al minsup:

$$supporto \geq minsup \tag{2.4}$$

- Confidenza minima (minconf): gli itemset che verranno mantenuti saranno quelli con un valore di confidenza maggiore o uguale al minconf:

$$confidenza \geq minconf \tag{2.5}$$

I risultati degli algoritmi di estrazione devono rispettare i precedenti vincoli in modo da essere corretti (non devono contenere informazioni indesiderate) e completi (contengono di sicuro tutte le informazioni desiderate).

Poiché all'interno degli itemset non conta l'ordine, essi rappresentano una combinazione, mentre le regole sono delle permutazioni, in quanto l'ordine è importante.

### Approcci per l'estrazione

Esistono diversi approcci per l'estrazione delle regole:

- **Brute Force**: è uno degli algoritmi più semplici, ma con il costo computazionale più alto. Esso consiste nel considerare tutte le possibili permutazioni e, per ognuna di esse, verificare poi se soddisfa i vincoli di supporto e confidenza stabiliti. L'elevato costo computazionale, dovuto all'ingente numero di permutazioni possibili, rende questo approccio poco fattibile. La complessità computazionale è quindi espressa da:

$$O\left(|T|\ 2^d\ w\right) \tag{2.6}$$

  dove |T| è il numero di transazioni totali, $d$ il numero di item e $w$ la lunghezza della transazione.

- Generare tutte combinazioni (che sono meno delle permutazioni) e calcolare il loro supporto. Se una combinazione ha un supporto basso, essa non genererà mai delle regole interessanti. Quindi, verranno considerate soltanto le combinazioni con un supporto adeguatamente alto e soltanto da queste verranno poi generate le permutazioni, ovvero le regole.

  Esistono diversi algoritmi che è possibile utilizzare per l'estrazione delle combinazioni frequenti, come il principio Apriori o l'FP-Growth.

  Per aumentare l'efficienza della ricerca delle combinazioni, possono essere attuate diverse strategie:

  - Ridurre il numero di transazioni: consiste nello scartare le transazioni che contengono item inutili dal punto di vista dell'analisi;
  - Ridurre il numero di candidati: consiste nello scartare le le combinazioni non frequenti. In altre parole, si effettua un'operazione di *pruning* dello spazio di ricerca;

– Ridurre il numero di confronti: consiste nell'utilizzare algoritmi di confronto efficienti.

**Effetto della soglia di supporto sull'estrazione**

La scelta di un valore appropriato per la soglia di supporto *minsup* non è scontata. Infatti, il valore scelto avrà delle conseguenze sul numero e sulla qualità delle regole estratte. In particolare:

- se il valore di *minsup* è troppo alto, possono essere persi degli itemset che includono degli item rari, ma interessanti;
- se il valore di *minsup* è tropo basso, si ha un costo computazionale abbastanza alto ed inoltre il numero di itemset frequenti che vengono estratti sarà elevato.

## 2.5 Sequenze frequenti

Spesso alle transazioni sono associate informazioni temporali che permettono di collegare tra loro gli eventi che riguardano una specifica entità. Un esempio di database sequenziale è mostrato nella Tabella 2.2.

| **Entità** | **Tempo** | **Eventi** |
|---|---|---|
| A | 10 | 2,3,5 |
| A | 12 | 6,8 |
| A | 15 | 9 |
| B | 11 | 4,6 |
| B | 23 | 5,8 |
| C | 14 | 2,1,5,7 |
| C | 19 | 5,7 |

**Tabella 2.2:** Esempio di dataset sequenziale.

Una sequenza [4] è una lista ordinata **elementi** (transazioni):

$$s =< e_1\ e_2\ e_3\ ... > \tag{2.7}$$

ogni elemento è composto da un insieme di **eventi** (item):

$$e_i = \{i_1,\ i_2,\ ...,\ i_k\} \tag{2.8}$$

Ad ogni transazione è associato uno specifico istante temporale. Nella Figura 2.2 è possibile vedere una rappresentazione di una sequenza.

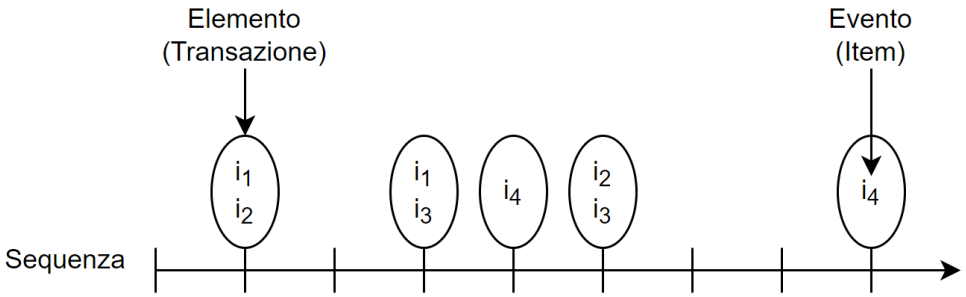



**Figura 2.2:** Esempio di una sequenza

La **lunghezza** di una sequenza indica il numero degli elementi di cui è composta quella sequenza; mentre una **k-sequence** rappresenta una sequenza composta da k elementi.

Una **sottosequenza** è una sequenza contenente degli elementi che sono dei sottoinsiemi degli elementi di una sequenza più grande. Un esempio di sottosequenze è mostrato nella Tabella 2.3

| **Sequenze** | **Sottosequenze** |
|---|---|
| < {2,4} {3,5,6} {8} > | < {2} {3,5} > |
| < {2,4} {2,4} {2,5} > | < {2} {4} > |
| < {2,4} {5,7} {8,10} > | < {4} {5,7} {10} > |

**Tabella 2.3:** Esempio di sottosequenze.

Il **supporto** di una sottosequenza rappresenta la frazione di frequenze che contengono quella determinata sottosequenza. Ovvero il rapporto tra il numero di sequenze che contengono la sottosequenza in esame ed il numero totale di sequenze presenti nel database.

Un **pattern sequenziale** è una sottosequenza frequente, ovvero una sottosequenza che ha un supporto maggiore o uguale al valore di *minsup*.

### 2.5.1 Estrazione dei pattern sequenziali

L'estrazione dei pattern sequenziali [5] è un importante problema nel Data Mining che trova ampia applicazione. La maggiore complessità dell'estrazione è dovuta alla necessità di esaminare un numero esponenziale di sottosequenze contenute in una sequenza. Molti degli algoritmi utilizzati per il mining delle sottosequenze si basano sulla metodologia del principio Apriori, la quale permette di ridurre il numero di combinazioni possibili. Tuttavia, il principio Apriori riscontra dei

problemi nel caso di database sequenziali molto grandi o nel caso in cui il numero di pattern sequenziali da estrarre è elevato. Per questo motivo sono stati sviluppati degli algoritmi alternativi che cercano di ovviare ai limiti precedentemente descritti, migliorando così le performance del processo di estrazione. Uno di questi algoritmi è il PrefixSpan, il quale è stato impiegato in questa tesi per la fase di estrazione dei pattern.

### 2.5.2 PrefixSpan

Come detto, l'algoritmo utilizzato per la parte di estrazione dei pattern sequenziali è stato il PrefixSpan, descritto in [5]. Il PrefixSpan è un metodo che estrae l'insieme completo dei pattern, riducendo però notevolmente il costo per la produzione delle sottosequenze candidate. L'idea principale su cui esso si basa è quella in cui, invece di proiettare l'intero database di sequenze, considerando tutte le possibili occorrenze di sottosequenze frequenti, la proiezione viene basata solo su prefissi frequenti. Infatti, ogni sottosequenza frequente può sempre essere generata partendo un prefisso frequente. In questo modo si riesce a ridurre sostanzialmente la dimensione dei dataset previsti e a superare di gran lunga le performance degli algoritmi basati sul principio Apriori.

## 2.6 Classificazione

La Classificazione [6] è una tecnica predittiva del Data Mining che consiste nella predizione di eventi discreti, basandosi su una conoscenza pregressa. Data una collezione di record (training set), ogni record è composto da un insieme di attributi di cui uno (etichetta di classe) esprime la classe di appartenenza del record. La Classificazione ha come obiettivo quello di costruire un modello capace di esprimere il valore dell'attributo di classe in funzione dei valori degli altri attributi. Il modello ottenuto sarà poi usato dal classificatore per effettuare le predizioni su nuovi dati in input, privi di etichetta di classe. Uno schema del processo di classificazione è mostrato nella Figura 2.3.

Un task di classificazione richiede la suddivisione del dataset di partenza in due diverse partizioni:

- **Training set**: è la partizione che sarà utilizzata per effettuare il training del modello;
- **Test set**: è una partizione utilizzata per valutare le prestazioni del modello ottenuto.

I parametri che indicano la qualità di una certa tecnica di classificazione sono i seguenti:

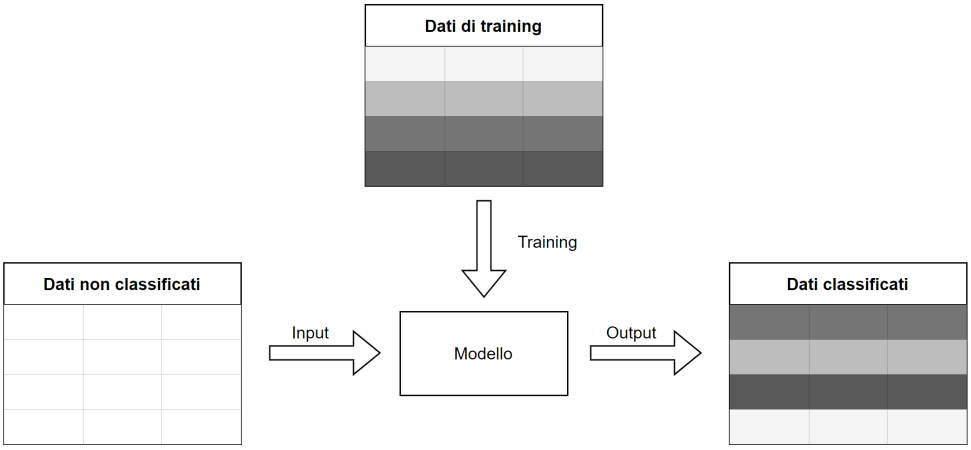



**Figura 2.3:** Processo di classificazione. I dati di training sono utilizzati per allenare il modello. Una volta ottenuto il modello, esso assegnerà i dati in input, privi di etichetta, ad una specifica classe, basandosi sul valore degli attributi.

- Accuratezza: indica la qualità della predizione;
- Efficienza: indica quanto tempo impiega una tecnica nella costruzione del modello;
- Scalabilità: indica quanto buona sia una tecnica ad adattarsi all'aumento delle dimensioni del database;
- Robustezza: indica quanto buona sia una tecnica in caso di presenza di rumore;
- Interpretabilità; indica quanto facile sia capire il funzionamento del modello.

In questo lavoro sono stati utilizzati diversi classificatori tradizionali come: l'Albero di Decisione, Regressione Logistica, Naive Bayes e il Random Forest. Inoltre, è stato implementato un classificatore associativo utilizzando i pattern estratti dal dataset. Poiché l'albero di decisione è stato il classificatore, tra quelli tradizionali, ad aver raggiunto le migliori performance, ho deciso di utilizzarlo come termine di paragone per il classificatore associativo da me implementato.

Nei successivi due paragrafi si fornisce una descrizione teorica di questi ultimi due classificatori.

### 2.6.1 Albero di decisione

Un albero di decisione è un modello predittivo, usato sia per la classificazione che per la regressione, che si basa sulla segmentazione del spazio di dati in diverse partizioni, il più possibile omogenee.

Ogni nodo interno dell'albero rappresenta un attributo del dataset; un arco verso un nodo figlio rappresenta un possibile valore per quell'attributo; una foglia rappresenta, infine, il valore predetto per l'etichetta di classe, a partire dai valori degli altri attributi. In altre parole, l'assegnazione dell'etichetta di classe viene eseguita scorrendo i nodi dell'albero, in base ai valori delle diversi attributi per un certo record che si vuole classificare, seguendo un percorso che parte dal nodo radice, fino ad arrivare ad un nodo foglia contenente il valore da assegnare. Ogni nodo interno dell'albero rappresenta quindi una divisione dello spazio di dati in due o più partizioni, ovvero in regioni distinte e non sovrapposte.

Tutti i record che, seguendo il percorso, finiscono in quella determinata regione saranno predetti con un certo valore che corrisponderà alla media dell'etichetta di classe, nel caso di regressione, oppure alla classe più presente tra i dati di training presenti in quella regione, nel caso di classificazione.

L'attributo da utilizzare per effettuare ad ogni passo la partizione dello spazio di dati, viene scelto in modo da minimizzare alcuni errori e delle funzioni di loss. Nel caso di regressione, la funzione da minimizzare è l'RSS (Residual Squared Sum), mentre nel caso di classificazione possono essere utilizzate le seguenti misure:

- **Gini Index**: rappresenta una misura di purezza. È un numero compreso tra 0 ed 1. Un basso valore indica che un nodo contiene principalmente record appartenenti ad una sola classe;
- **Entropia**: è un'alternativa al Gini index che indica la quantità di disordine;
- **Classification Error Rate**: indica la frazione dei record di training che non appartengono alla classe più presente in una determinata partizione. Questa misura non è sufficientemente adatta alla costruzione degli alberi di classificazione in quanto essa tende, per come è stata definita, a portare ad overfitting.

**Vantaggi**

+ È di facile interpretazione;
+ Non è un modello costoso da costruire e converge velocemente;
+ La classificazione è abbastanza rapida;
+ Se il problema è facile, raggiunge una buona accuratezza

**Svantaggi**

− Va in crisi in caso di valori mancanti;
− Va in crisi se il problema non è linearmente separabile.

### 2.6.2 Classificatore associativo

Un classificatore associativo [6] è un algoritmo di classificazione in grado di generare un modello, basandosi su delle regole di associazione estratte dai dati. Una regola di associazione ha la seguente forma:

$$(Condizione) \rightarrow y \tag{2.9}$$

dove la condizione (un itemset) rappresenta il corpo della regola, mentre la testa è rappresentata dall'etichetta di classe.

Una regola **copre** un certo record $x$, quando gli attributi del record $x$ che vogliamo classificare soddisfano la condizione della regola. Quindi se un record verifica la condizione di una regola, il classificatore associativo classificherà quel record con l'etichetta indicata dalla testa della regola. Se nessuna regola viene verificata il classificatore assegnerà una classe di default.

Alla luce di quanto detto, possono essere distinte due fasi su cui la classificazione associativa si basa:

1. Estrazione dei pattern: è la fase più pesante, in quanto, tramite un determinato algoritmo, vengono estratti tutti i pattern frequenti sensati. Dopo di che, essi dovranno essere ordinati in base a supporto, confidenza o correlazione ed infine verranno selezionati soltanto quelli più importanti.

2. Verifica dei pattern estratti: questa è la fase in cui viene effettuata la classificazione vera e propria (come descritto in precedenza).

**Vantaggi**

+ È di facile interpretazione;

+ La qualità è solitamente più alta di quella degli alberi di decisione;

+ La classificazione è molto efficiente (più veloce rispetto a quella degli alberi);

+ È robusto rispetto alla presenza di valori mancanti, in quanto in quel caso la regola non viene verificata;

+ Scala molto bene con l'aumentare della dimensione del training set.

**Svantaggi**

− Non scala bene con l'aumentare del numero di attributi;

− Il supporto delle regole non è sempre un buon indicatore.

### 2.6.3 Metriche di Valutazione

Le metriche di valutazione sono utilizzate per stimare quanto il modello sarà accurato nell'effettuare la predizione. Tale valutazione si basa sui conteggi dei record di test predetti correttamente e/o erroneamente dal modello. Quindi le varie metriche differiscono l'una dall'altra in base all'uso che fanno di questi conteggi.

#### Matrice di confusione

Una matrice di confusione (Figura 2.5) è una tabella che riporta il numero di istanze predette correttamente ed erroneamente in base al loro valore reale. La matrice ha per righe le classi predette e per colonne le classi reali. Nelle diverse celle si hanno dei valori che indicano:

- **TP (True Positive)**: numero di istanze positive, predette come positive;
- **FP (False Positive)**: numero di istanze negative, predette come positive;
- **TN (True Negative)**: numero di istanze negative, predette come negative;
- **FN (False Negative)**: numero di istanze positive, predette come negative;

Sulla diagonale principale della matrice si ha il numero di record correttamente predetti, mentre, nel resto della matrice si ha il numero di record classificati erroneamente.

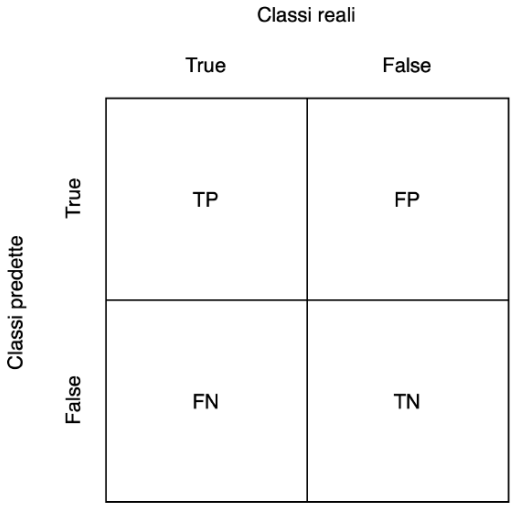



**Figura 2.4:** Matrice di confusione

### Accuratezza

L'**accuratezza** è una metrica che rappresenta la percentuale totale di record correttamente classificati. Infatti essa è il rapporto tra il numero di record correttamente classificati ed il numero di predizioni totali. In termini matematici:

$$Accuratezza = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.10}$$

L'accuratezza potrebbe essere una misura ingannevole nel caso di classi sbilanciate. Per esempio, se consideriamo 98 elementi appartenenti alla classe "blu", 2 elementi appartenenti alla classe "red" ed un classificatore D(x)= blu per qualsiasi x, quindi che classifica sempre come blu, questo ci dà il 98% di accuratezza, ma il 100% di elementi "red" sono classificati erroneamente.

### Precisione

La **precisione** è una metrica che rappresenta la percentuale dei record che sono stati classificati correttamente come positivi, rispetto a tutti quelli che sono stati classificati come positivi. La formula è la seguente:

$$Precisione = \frac{TP}{TP + FP} \ . \tag{2.11}$$

### Richiamo

Il **richiamo** è una metrica che rappresenta la percentuale di record che sono stati classificati correttamente come positivi rispetto a tutti quei record che erano realmente positivi. In altre parole rappresenta la percentuale di positivi che il classificatore è riuscito a scoprire. La formula è la seguente:

$$Richiamo = \frac{TP}{TP + FN} \ . \tag{2.12}$$

### F1 Score

L'**F1 Score** è una metrica utilizzata soprattutto nel caso di classi sbilanciate e consiste nella media armonica tra precisione e richiamo. Il range per l'F1 score è [0,1] ed indica quanto corretto (ovvero quante istanze siano state classificate correttamente) e quanto robusto sia un classificatore. Matematicamente può essere espressa come:

$$F1Score = 2 * \frac{1}{\frac{1}{precisione} + \frac{1}{richiamo}} \ . \tag{2.13}$$

### Curva ROC

La **curva ROC** (Receiver Operating Characteristics) è una delle più importanti metriche di valutazione delle performance di un modello di classificazione. ROC è una curva di probabilità, mentre l'AUC (Area Under Curve) rappresenta il grado di separabilità. Quest'ultima specifica quanto capace sia il modello a riconoscere e distinguere fra le classi. La curva ROC è un grafico del False Positive Rate (asse x) contro il True Positive rate (asse y), per un numero di diversi valori di soglie candidate tra 0 ed 1. Di solito, si fa variare la soglia della probabilità per cui un record appartenga alla classe positiva (quindi per calcolare il primo valore si setta a 0%, per l'ultimo valore al 100%). Il True Positive Rate consiste nel numero di record correttamente classificati come positivi rispetto al numero totale di record realmente positivi (in pratica coincide con il richiamo). Mentre il False Positive Rate è il rapporto tra il numero di falsi positivi ed il numero di record realmente negativi. Le formule sono riportate sotto:

$$True\ Positive\ Rate = \frac{TP}{TP + FN} \tag{2.14}$$

$$False\ Positive\ Rate = \frac{FP}{FP + TN}\ . \tag{2.15}$$

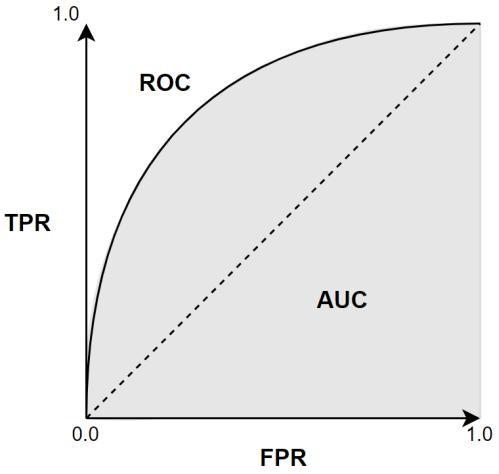



**Figura 2.5:** Curva ROC

# Capitolo 3

# Analisi dei dati e classificatore associativo per dati spaziali-temporali

In questo capitolo verrà fornita una descrizione della struttura generale dei dati utilizzati ed una trattazione del lavoro svolto. Partendo dalla manipolazione dei dati di partenza, si arriverà fino all'ottenimento del classificatore associativo, le cui performance di test saranno infine confrontate con quelle di un classificatore tradizionale come l'albero di decisione.

Tutti i risultati ottenuti dalle varie analisi ed esperimenti saranno trattati nel capitolo successivo.

## 3.1 Panoramica del lavoro svolto

Il primo passo effettuato è stato quello di analizzare i dati offerti dal dataset di partenza [7], in modo da rilevare e ripulire eventuali incongruenze. Dopo questa prima parte di pulizia dei dati, si è proseguito con la manipolazione del dataset ottenuto in modo da ricavare un formato adeguato per l'estrazione dei pattern sequenziali, tenendo conto delle informazioni spaziali e temporali delle diverse stazioni.

Dalla prima fase di estrazione dei pattern si è evidenziato uno scarso legame tra gli eventi riguardanti le stazioni ed il loro vicinato. Infatti, i pattern estratti non sono stati di elevata qualità. Per questo motivo si è pensato di effettuare un'ulteriore fase di analisi nella quale è stato eseguito uno studio sulla correlazione dei dati appartenenti al dataset ripulito.

Successivamente, si è passati alla fase di implementazione del classificatore associativo: utilizzando i pattern ottenuti dalla precedente fase di estrazione, è stato effettuato il test del suddetto classificatore.

Il dataset ripulito è stato inoltre utilizzato per ottenere un nuovo dataset contenente le informazioni spaziali e temporali delle varie stazioni, sul quale sono stati poi applicati diversi classificatori tradizionali. Il classificatore che ha ottenuto le migliori performance, e su cui quindi mi sono focalizzato maggiormente, è stato l'albero di decisione.

L'ultima fase della tesi riguarda quindi il confronto delle performance dei due classificatori ottenuti.

## 3.2 Descrizione del dataset

Per lo svolgimento del lavoro sono stati utilizzati due dataset contenenti dei dati riguardanti delle stazioni di bike sharing situate nelle città di San Francisco ed alcune città limitrofe. In particolare i due dataset sono i seguenti:

- **status.csv**: dataset contenente il numero di bici e slot disponibili per una data stazione e per ogni minuto. La struttura dei record appartenenti a questo dataset è la seguente:

$$station\_id,\ bikes\_available,\ docks\_available,\ time \tag{3.1}$$

  dove:

  - station_id: indica l'identificatore univoco della stazione;
  - bikes_available: indica il numero di biciclette disponibili;
  - docks_available: indica il numero di slot liberi;
  - time: indica la data e l'istante temporale della lettura.

- **station.csv**: dataset contenente dati riguardanti una stazione in cui gli utenti possono prelevare o depositare le biciclette. La struttura dei record appartenenti a questo dataset è la seguente:

$$id,\ name,\ lat,\ long,\ dock\_count,\ city,\ instalation\_date \tag{3.2}$$

  dove:

  - id: identificativo univoco della stazione;
  - name: nome della stazione;
  - lat: latitudine;

- long: longitudine;
- dock_count: numero totale di slot di cui la stazione dispone;
- city: città in cui si trova la stazione;
- installation_date: data in cui la stazione è stata installata.

Il formato utilizzato per rappresentare le informazioni temporali è il PST (Pacific Standard Time), ovvero il formato americano:

$$YYYY - DD - MM\ HH : MM : SS \tag{3.3}$$

## 3.3 Analisi dei dati

Questa fase di analisi è stata svolta in modo da scovare ed eventualmente correggere o filtrare alcune possibili incongruenze contenute nei dati di partenza, dovute ad errori o malfunzionamenti del sistema di rilevazione.

### 3.3.1 Valutazione delle oscillazioni del numero totale di slot

Una delle prime analisi che è stata condotta è stata quella riguardante la valutazione di eventuali oscillazioni tra il numero di slot totali di ciascuna stazione (contenuta nel dataset *station*) e la somma tra il numero di bici disponibili e di slot liberi (contenuti nei record del dataset *status*).

### 3.3.2 Valutazione della regolarità di raccolta dei dati

In questa fase l'obiettivo è stato quello di ottenere un stima approssimativa della regolarità con cui sono stati raccolti i dati a disposizione nel dataset *status*.

Per ogni stazione, ho ricavato il primo timestamp, l'ultimo timestamp ed il numero di letture in quella finestra di tempo. Dopo di che ho calcolato la frequenza media di rilevamento tramite il rapporto tra numero totale di rilevamenti ed il numero totale di minuti trascorsi tra il primo e l'ultimo rilevamento. Un esempio del dataframe ottenuto è mostrato in Figura 3.1.

### 3.3.3 Statistiche sulle situazioni critiche

Infine, è stata condotta una valutazione delle statistiche riguardanti le situazioni in cui una stazione si trovasse in una situazione critica. Per situazione critica in questo caso si intende una situazione in cui la stazione sia completamente vuota, completamente piena, quasi vuota o quasi piena. Negli ultimi due casi è stata definita una soglia appropriata per determinare il numero di bici o slot disponibili in modo da ricadere in quella specifica situazione critica.

| | station_id | start | end | count | difference_in_minutes | frequency |
|---|---|---|---|---|---|---|
| **0** | 2 | 2013-08-29 12:06:01 | 2015-08-31 23:59:02 | 1046898 | 1054793.0 | 0.992515 |
| **1** | 3 | 2013-08-29 12:06:01 | 2015-08-31 23:59:02 | 1047113 | 1054793.0 | 0.992719 |
| **2** | 4 | 2013-08-29 12:06:01 | 2015-08-31 23:59:02 | 1047100 | 1054793.0 | 0.992707 |
| **3** | 5 | 2013-08-29 12:06:01 | 2015-08-31 23:59:02 | 1047142 | 1054793.0 | 0.992746 |
| **4** | 6 | 2013-08-29 12:06:01 | 2015-08-31 23:59:02 | 1047142 | 1054793.0 | 0.992746 |
| **...** | ... | ... | ... | ... | ... | ... |

**Figura 3.1:** Esempio di dataframe ottenuto nella valutazione della regolarità di raccolta dei dati

Ho prima calcolato le statistiche separatamente per ogni stazione, creando un dataframe che contenesse le seguenti informazioni:

- numero di volte in cui la stazione è completamente piena;
- numero di volte in cui la stazione è completamente vuota;
- percentuale di volte in cui la stazione è completamente piena rispetto al numero totale di letture per quella stazione;
- percentuale di volte in cui la stazione è completamente vuota rispetto al numero totale di letture per quella stazione;
- percentuale di criticità totali: rapporto tra la somma del numero di situazioni in cui la stazione è piena ed il numero di situazioni in cui la stazione è vuota, diviso il numero totale di letture del dataframe;
- numero di volte in cui il la quantità "bikes_available" è uguale a 1 e quante volte è uguale a 2;
- numero di volte in cui il la quantità "docks_available" è uguale a 1 e quante volte è uguale a 2;

Un esempio del dataframe ottenuto è mostrato in figura 3.2.
Successivamente ho poi analizzato le statistiche sulle situazioni critiche considerando l'intero dataset.

## 3.4 Pulizia dei dati

Una volta effettuate tutte le analisi necessarie, ho proceduto con il filtraggio di tutti quei record che contenevano delle incongruenze. In particolare, ho deciso di

| station_id | full | %full | 1_bike_available | 2_bikes_available | empty | %empty | 1_dock_available | 2_docks_available | %total_critical_situation |
|---|---|---|---|---|---|---|---|---|---|
| **2** | 1340 | 0.001280 | 1793 | 4608 | 466 | 0.000445 | 1516 | 2309 | 0.011861 |
| **3** | 1056 | 0.001008 | 1205 | 3510 | 4817 | 0.004600 | 13223 | 34812 | 0.011861 |
| **4** | 10618 | 0.010140 | 29688 | 57923 | 6614 | 0.006316 | 12838 | 37107 | 0.011861 |
| **5** | 1321 | 0.001262 | 6934 | 12093 | 0 | 0.000000 | 0 | 112 | 0.011861 |
| **6** | 3564 | 0.003404 | 15438 | 14245 | 3633 | 0.003469 | 13852 | 24770 | 0.011861 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |

**Figura 3.2:** Esempio di dataframe ottenuto dal calcolo delle statistiche sulle situazioni critiche.

eliminare tutti quei record che contenessero un valore assoluto dell'oscillazione, definita in precedenza, maggiore di 5. Ho deciso di effettuare il filtraggio piuttosto che una correzione dei record in esame, in quanto il loro numero è risultato essere davvero esiguo rispetto al numero di record totali.

Il dataset risultante è stato salvato e sarà quello utilizzato sia per l'estrazione dei pattern sequenziali frequenti, sia per la generazione di un nuovo dataset in formato tabellare su cui poi allenare i classificatori tradizionali.

## 3.5 Estrazione dei pattern

Per la parte riguardante l'estrazione dei pattern frequenti, è stato utilizzato come punto di partenza il codice sviluppato in un precedente lavoro [8], apportando le modifiche e le correzioni del caso. Il suddetto lavoro consiste nell'estrazione dei pattern frequenti, tenendo conto del concetto di spazio e di tempo. I pattern ottenuti terranno conto dello stato, in diversi istanti temporali, di una certa stazione di riferimento e delle stazioni ad essa vicine.

Il classificatore spaziale-temporale che vogliamo ottenere deve basarsi su questo tipo di sequenze. Quindi, le regole che bisogna ottenere devono tenere conto di situazioni critiche della stazione di riferimento e, contemporaneamente, anche delle eventuali situazioni critiche, in determinati istanti di tempo successivi, riguardanti le stazioni che distano un certo numero di delta spaziali dalla stazione di riferimento.

### 3.5.1 Dettagli implementativi

Come detto nei precedenti paragrafi, sono stati considerati diversi stati in cui le stazioni sarebbero potute trovarsi:

- **Piena**: situazione in cui non sono presenti slot liberi;

- **Vuota**: situazione in cui non sono presenti biciclette disponibili;
- **Quasi Piena**: situazione in cui sono presenti 1 o 2 slot liberi;
- **Quasi Vuota**: situazione in cui sono presenti 1 o 2 biciclette disponibili;
- **Normale**: quando la stazione non si trova in nessuno degli stati precedenti.

In questa fase di estrazione sono stati utilizzati diversi parametri:

- *intervallo temporale*: la dimensione dello slot temporale considerato;
- *intervallo spaziale*: la dimensione dello slot spaziale considerato;
- *numero di delta temporali*: numero di diversi slot temporali considerati;
- *numero di delta spaziali*: numero di diversi slot spaziali considerati.
- *supporto*: supporto considerato per fare il training dell'algoritmo PrefixSpan.

Inoltre, sono state implementate diverse versioni del codice riguardante l'estrazione dei pattern. Ovvero:

- una prima variante in cui si è tenuto conto di tutti gli eventi critici, anche se essi si ripetono consecutivamente. Per esempio, se una stazione diventa Piena e lo rimane per un certo intervallo di tempo, ogni istante di tempo considerato che ricade in quell'intervallo, il sistema continua a segnalare quella stazione come Piena e quindi ne tiene conto anche nell'estrazione dei pattern;
- una seconda variante in cui si è tenuto conto solo dell'inizio della criticità: quindi si è tenuto conto solo della variazione dello stato, ovvero solo il primo istante temporale in cui si è avuto il passaggio da uno stato all'altro;
- una variante in cui vengono estratti i pattern separatamente per diverse fasce orarie considerate.

### 3.5.2 Struttura dei pattern

Ogni pattern ottenuto da questa fase di estrazione ha come riferimento una stazione e descrive lo stato di questa stazione di riferimento e quello del suo vicinato. La struttura dei pattern è la seguente:

$$[[[['Stato\_T0\_\Delta s'], ['Stato\_T1\_\Delta s'], ['Stato\_T2\_\Delta s']]], ['conf - supCount']] \tag{3.4}$$

Una regola avrà quindi al suo interno un numero di elementi pari al *numero di delta temporali* considerato come parametro. L'ultimo di questi elementi sarà considerato come la testa (o conseguente) della regola, mentre tutti quelli precedenti

costituiranno il corpo (o antecedente) della regola.

Ogni elemento appartenente ad un certo delta temporale contiene internamente una stringa consistente in una lista di eventi critici separati da una virgola:

$$['Stato\_\Delta T\_\Delta s, Stato\_\Delta T\_\Delta s, Stato\_\Delta T\_\Delta s, ...'] \tag{3.5}$$

dove:

- Stato: rappresenta lo stato della stazione;
- $\Delta$T: è fisso per tutta la stringa e rappresenta il delta temporale considerato;
- $\Delta$s: può variare e rappresenta il delta spaziale. Se esso è 0, indica la stazione di riferimento, mentre se è maggiore di 0, indica che si tratta di una stazione vicina a quella di riferimento, ovvero indica una stazione che dista $\Delta$s intervalli spaziali da quella di riferimento.

Infine, l'ultimo elemento della regola 3.4 contiene delle informazioni relative alla confidenza (*conf*) e al supportCount (*supCount*) della regola. Queste informazioni saranno poi utili nella successiva fase di filtraggio dei pattern.

### 3.5.3 Trasformazione del dataset

Come già detto in precedenza, è stato necessario trasformare il dataset in un formato adatto all'estrazione dei pattern frequenti.

Come primo passo, sono stati determinati, per tutti i record del dataset, gli stati delle stazioni in base al numero di biciclette o slot disponibili. Dopo di che si è tenuto conto soltanto degli stati d'interesse.

Il passo successivo è stato quello di raggruppare i record in intervalli di tempo la cui lunghezza è stata specificata tramite il parametro *"intervallo temporale"*. In questo modo, per ogni intervallo, si è ottenuto una lista di tutte quelle stazioni, con il rispettivo stato, che avessero un timestamp che ricadesse all'interno di quell'intervallo temporale.

Dopo di che, sono state create le finestre temporali andando a considerare gli intervalli ottenuti ed il parametro *"numero di delta temporali"*, che indica la grandezza della finestra temporale da considerare.

A questo punto è stata introdotta l'informazione spaziale. Utilizzando la formula di Harvesine, è stata calcolata la distanza tra tutte le stazioni. Dopo di che, utilizzando i parametri *"intervallo spaziale"* e *"numero di delta spaziali"*, è stato determinato in modo appropriato il vicinato da considerare per ogni stazione.

Il formato così ottenuto è stato quello utilizzato per l'estrazione dei pattern frequenti.

### 3.5.4 Elaborazione e salvataggio dei pattern

Dopo l'esecuzione dell'algoritmo del PrefixSpan, è stato applicato un filtro in modo da mantenere quelle sequenze con almeno due finestre temporali e che contenessero almeno un elemento appartenente ad uno slot temporale 0 (T0) e ad uno slot spaziale 0. In altre parole, deve essere presente almeno un evento critico riguardante la stazione di riferimento all'istante corrente. Il passo successivo è stato quello di calcolare per ogni sequenza la confidenza di avere l'ultimo elemento della sequenza data la parte che la precede (l'antecedente). Tale valore è stato ottenuto tramite il rapporto del supportCount dell'antecedente della sequenza, diviso il supportCount dell'intera sequenza.

Infine, i pattern sono stati ordinati e salvati sia secondo confidenza decrescente, sia secondo supportCount decrescente.

## 3.6 Filtraggio dei pattern

Una volta estratti i pattern, essi sono stati filtrati e selezionati in modo intelligente. In questo modo è stato possibile utilizzarli per la creazione di un modello che permetta di effettuare delle predizioni con delle performance ragionevoli.

Il filtro implementato, come prima cosa, va a leggere il file testuale in cui si trovano i pattern di interesse. Dopo di che, esso mantiene soltanto tutti quei pattern che contengono nel conseguente un elemento con stato "Quasi Piena", riferito solamente alla stazione di riferimento (ovvero con delta spaziale uguale a 0).

Inoltre, sono stati tenuti in considerazione diversi parametri, che mi hanno permesso di filtrare ulteriormente i pattern:

- soglia di confidenza(*conf_threshold*): è stata la soglia più utilizzata in quanto mi ha permesso di mantenere soltanto i pattern con una confidenza maggiore della soglia specificata;
- soglia di supporto (*sup_threshold*): permette di mantenere soltanto i pattern con un supportCount più alto del valore specificato;
- un flag (*filter_normal*): un booleano che permette di mantenere soltanto i pattern che contengono esclusivamente lo stato "QuasiPiena".

## 3.7 Analisi della correlazione dei dati

Dalle precedenti fasi di estrazione e filtraggio, si sono ottenuti dei pattern abbastanza deludenti in termini di confidenza ed informazioni contenute. Per questo motivo,

si è pensato di effettuare un'ulteriore fase di analisi nella quale è stato eseguito uno studio sulla correlazione dei dati appartenenti al dataset ripulito. In questo modo è stato possibile vedere se ci fossero delle correlazioni tra diverse stazioni in certi istanti di tempo.

Considerando il fatto che nei dati sono contenute stazioni appartenenti a città diverse, ho deciso di trattare separatamente le diverse città.

È stata considerata una finestra temporale con $n$ istanti di tempo e degli intervalli temporali di una certa dimensione usati per il campionamento. La dimensione $n$ delle finestra e la grandezza degli intervalli temporali sono stati utilizzati come parametri in modo da eseguire degli esperimenti al variare di essi. La dimensione dell'intervallo è stata utilizzata come frequenza di campionamento per creare delle finestre su cui poi è stata calcolata la media del numero di bici disponibili.

Per ogni città, è stata quindi ottenuta una tabella che ha come righe i timestamp e come colonne il numero medio di bici disponibili al tempo corrente e agli $n$ istanti di tempo precedenti, per ogni stazione appartenente alla determinata città presa in considerazione. In altre parole, per ogni stazione si avrà un numero $n$ di colonne corrispondente alla dimensione $n$ della finestra temporale.

Da questa tabella è stata poi calcolata la matrice di correlazione dalla quale è stato possibile controllare l'eventuale presenza di stazioni molto correlate positivamente o negativamente ed, in quel caso, capire eventualmente come fossero posizionate tra di loro tali stazioni, ovvero capire se esse fossero vicine o meno. Inoltre, tale matrice di correlazione ha permesso di comprendere anche l'informazione relativa agli specifici istanti temporali in cui ci fossero eventualmente state delle correlazioni elevate. In altre parole, questo studio ha permesso di analizzare la presenza o meno di legami spaziali e temporali tra eventuali situazioni altamente correlate.

## 3.8 Applicazione dei classificatori tradizionali

Il passo successivo è stato quello di costruire dei dataset opportuni, in modo da allenare diversi classificatori tradizionali, valutarne le prestazioni e confrontarle con quelle del classificatore associativo.

### 3.8.1 Creazione dei dataset

Partendo dal dataset ripulito, ho creato quindi dei dataset contenenti un record per ogni timestamp in cui è contenuto, per ogni stazione, il numero di bici e di

posti disponibili nell'istante corrente e in $n$ istanti temporali precedenti. Anche in questa fase, infatti, sono stati considerati due parametri:

- Dimensione dell'intervallo temporale;
- Dimensione della finestra $n$: rappresenta il numero di istanti temporali da considerare.

La dimensione dell'intervallo è stata utilizzata per effettuare il finestramento in intervalli di tempo. Per ogni intervallo è stata poi calcolata la media del numero di bici e slot disponibili.

Poiché l'obiettivo era quello di predire lo stato di ciascuna stazione, ho ottenuto un dataset diverso per ciascuna stazione. In ogni dataset è stato tenuto conto, nella colonna "status", dello stato della particolare stazione di riferimento che si voleva predire, ovvero dello stato della stazione di riferimento all'istante di tempo successivo all'istante corrente T0.

Poiché i pattern ottenuti dalla precedente fase di estrazione, che hanno dato delle confidenze leggermente più alte, sono stati quelli estratti tramite la configurazione che teneva conto della variazione di stato per la configurazione Normale-QuasiPiena, gli stati che sono andato a considerare sono stati "QuasiPiena", indicato con l'etichetta di classe "QP" e comprendente entrambe le situazioni critiche di stazione quasi piena e piena, e lo stato "Normale", indicato con l'etichetta di classe "N".

I dataset così ottenuti sono stati divisi, in base al tempo, in training e test set, utilizzando una proporzione del 70-30. I diversi training set sono stati poi utilizzati per allenare i diversi classificatori. In questo modo, per ogni tipo di classificatore ho ottenuto un modello diverso per ogni stazione.

### 3.8.2 Training e test dei classificatori

Dopo aver ottenuto i dataset, ho proseguito con l'effettuare il training di 5 diversi classificatori, utilizzando i parametri di default. In particolare, i classificatori considerati sono stati:

- Albero di decisione
- Support Vector Machine
- Logistic Regression
- Naive Bayes
- Random Forest

Dopo di che, ho testato i modelli ottenuti sui rispettivi test set. Per valutare le prestazioni sono state considerate varie metriche come accuratezza, richiamo, precisione ed f1-score. Poiché per ogni classificatore si ha un modello per ogni stazione, ho proceduto a calcolare la media delle prestazioni dei vari modelli sulle diverse stazioni, in modo da facilitare il confronto fra i diversi classificatori. I risultati generali sono stati quindi ottenuti effettuando una micro-media. Essa consiste nel sommare tutti i rispettivi valori delle singole matrici di confusione ottenute per ogni stazione, ottenendo così un'unica matrice di confusione globale sulla quale calcolare infine le metriche di valutazione. Non è stato possibile calcolare la media classica in quando alcuni modelli su determinate stazioni predicevano tutti i record come appartenenti alla classe negativa (N), ottenendo così un'accuratezza nulla.

## 3.9 Implementazione del classificatore associativo

Questa fase è il cuore di tutto il presente lavoro. In questo step, infatti, è stato implementato il classificatore associativo vero e proprio, il quale effettua le predizioni basandosi sui pattern estratti nelle diverse fasi di estrazione. Verranno quindi descritti in maniera più dettagliata i diversi passi svolti.

### 3.9.1 Manipolazione del dataset

Partendo dal dataset contenente i dati puliti, è stato ottenuto un dataset iniziale contenente soltanto lo status delle 35 stazioni della città di San Francisco. Dopo di che, esso è stato splittato in training e test set coerentemente con lo splitting che è stato eseguito nel caso nella classificazione per mezzo di modelli tradizionali, ricavando il timestamp che faceva da divisorio tra training e test set.

### 3.9.2 Estrazione dei pattern dal training set

I pattern sono stati estratti partendo del training set e seguendo diverse configurazioni scelte. È stata utilizzata la versione dell'estrazione che considera come eventi di interesse soltanto i passaggi di stato. Gli stati considerati sono stati:

- QuasiPiena: situazione critica che include anche il caso "Piena";
- Normale: stato che include tutte le restanti situazioni.

Una volta estratti, i pattern sono stati salvati in ordine di confidenza decrescente.

### 3.9.3 Filtraggio dei pattern ottenuti

Una volta estratti i pattern, ho proceduto con il filtraggio in modo da scartare tutti quelli che nel conseguente non contenevano soltanto la stazione di riferimento e lo stato "QuasiPiena". Tuttavia, il numero ancora troppo elevato di pattern avrebbe portato a dei risultati con una precisione molto bassa ed un richiamo molto alto. Questo è dovuto al fatto che, essendoci molte regole, verrebbe sicuramente trovato un pattern che sarebbe verificato. In questo modo tutti i record del test set sarebbero predetti con "QP".

Per ovviare a questo problema, ho provato a seguire due strade diverse che mi hanno permesso di ridurre il numero di pattern:

1. Filtrare le regole utilizzando una soglia di confidenza e di supporto;
2. Mantenere solo le poche regole che hanno solo lo stato "QuasiPiena" anche nell'antecedente (ossia non considerare lo stato "Normal").

Ho quindi inserito all'interno del filtro due nuove soglie per confidenza e supporto, in modo da scartare tutti quei pattern che non superassero tali soglie, ed anche un flag che permettesse, all'occorrenza, di scartare tutti i pattern che contenessero lo stato "Normale" nel corpo.

### 3.9.4 Test dei pattern

Questa è la fase più importante per quanto riguarda la classificazione associativa. Lo pseudocodice del classificatore associativo è fornito nell'Algoritmo 1.

Una volta ottenuti i vari file contenenti i pattern estratti e filtrati, ho proceduto all'implementazione del codice per la classificazione delle entry del test set (lo stesso test set utilizzato nel caso della classificazione dei modelli tradizionali), utilizzando questa volta proprio i pattern ottenuti.

Per ogni stazione appartenente alla città di San Francisco, ho prelevato il rispettivo test set (riga 5 dell'Algoritmo 1) e per ognuno di essi ho preceduto poi con il test dei pattern (righe 9 - 28).

In particolare, per ogni record, sono andato a verificare se esistesse almeno un numero di pattern verificati uguale ad una certa soglia prefissata (riga 22).

Per la verifica che un certo pattern fosse verificato, ho controllato se tutti gli item dell'antecedente, trovassero una corrispondenza nella entry corrente del test set (riga 17).

I singoli item del pattern indicano un certo stato per la stazione di riferimento e per alcuni suoi vicini, in determinati intervalli di tempo, Allo stesso tempo, i record del test set contengono, per ogni istante temporale, lo stato della stazione di

riferimento e quello di tutti i suoi vicini nell'istante corrente e nei 4 slot temporali precedenti. Quindi la verifica è consistita in un controllo di consistenza tra le informazioni contenute nel pattern e quelle contenute nel record corrente del test set.

Qualora il classificatore trovi un numero di pattern verificati almeno uguale al numero minimo desiderato, classificherà quel determinato record come "QuasiPiena" (QP) (righe 22 - 25), in caso contrario come "Normale" (N) (riga 12).

Una volta ottenuta una predizione per tutto il test set della stazione corrente, ho calcolato la matrice di confusione e le rispettive metriche ("richiamo", "precisione" ed "F1-score") in modo da valutare la bontà del classificatore (righe 29 - 31). In questo modo, ho ottenuto una matrice di confusione con le rispettive metriche, per ognuna delle stazioni.

Per avere un'idea generale del modello, ho proceduto, quindi, con il calcolo della micro-media (righe 33 - 35), ovvero ho sommato tutte le singole matrici di confusione, ottenendo una singola matrice di confusione risultante, dalla quale ho calcolato nuovamente le metriche generali.

## 3.10 Confronto tra il classificatore associativo e l'albero di decisione

Nella parte sperimentale sono stati effettuati diversi test del classificatore associativo e dell'albero di decisione, utilizzando diverse versioni degli algoritmi e diverse configurazioni, variando i molteplici parametri a disposizione. Una volta ottenuti i vari risultati dei test, si è proceduto ad effettuare un confronto tra i migliori risultati ottenuti dai due classificatori, in modo da confrontarne le performance generali.

**Algoritmo 1** Implementazione del classificatore associativo.

```
1: procedure ASSOCIATIVE_CLASSIFIER(num_tot_pattern)
2:     ▷ num_pattern is the minimum number of patterns that must match in order that the record is classified as positive.
3:     ▷ Execution
4:     for station in SanFranciscoStations do
5:         read test_df
6:         ▷ Initialization
7:         y_test ← test_df["status"]
8:         y_pred ← [ ]
9:         for record in test_df do
10:            read pattern_file
11:            num_pat_matched ← 0
12:            prediction ← "N"
13:            for pattern in pattern_file do
14:                procedure EXTRACT_INFO(pattern)
15:                    Extract usefull information from pattern's items
16:                end procedure
17:                if pattern matches then
18:                    num_pat_matched ← num_pat_matched + 1
19:                else
20:                    continue to the next pattern
21:                end if
22:                if num_pat_matched == num_tot_pat then
23:                    prediction ← "QP"
24:                    break
25:                end if
26:            end for
27:            y_pred ←+ prediction
28:        end for
29:        procedure COMPUTE_METRICS(y_test, y_pred)
30:            compute correlation matrix and metrics for a single station
31:        end procedure
32:    end for
33:    procedure COMPUTE_OVERALL_METRICS
34:        compute overall correlation matrix and metrics for all stations
35:    end procedure
36: end procedure
```

# Capitolo 4

# Valutazione Sperimentale

In questo capitolo saranno forniti i risultati dettagliati ottenuti tramite le diverse fasi descritte nel capitolo precedente. Partendo dai risultati ottenuti delle analisi condotta sui dati di partenza, si arriverà fino ai risultati dagli esperimenti effettuati tramite i test dell'albero di decisione e del classificatore associativo.

## 4.1 Descrizione del dataset

Il dataset di partenza [7] contiene dei dati riguardanti 70 diverse stazioni di bike sharing site nella città di San Francisco ed in alcune città confinanti come Mountain View, Palo Alto, Redwood City e San Jose.

I dati coprono una finestra temporale di due anni, che va dall'Agosto 2013 all'Agosto 2015. Tuttavia, alcune stazioni presentano delle letture che ricoprono un arco temporale più breve in quanto sono state installate in un secondo momento.

I dati riguardanti tutte le letture, relative al numero di biciclette disponibili e di slot vuoti, sono contenuti nel dataset "status.csv"; i dati contenenti tutte le informazioni sulle stazioni, quali coordinate, città di appartenenza, numero di slot totali e data dell'installazione, sono invece contenuti nel file "station.csv".

## 4.2 Analisi dei dati

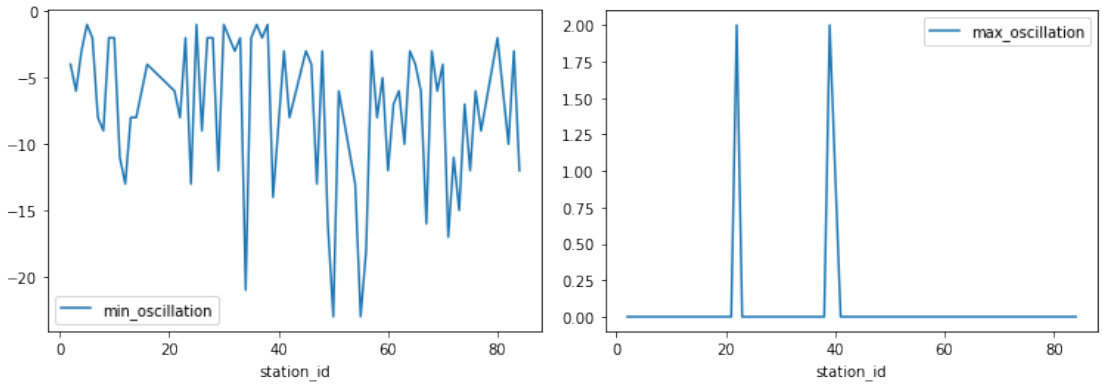
Questa fase è stata condotta in modo tale da rilevare e, se necessario, correggere eventuali inconsistenze o rumori presenti nel dataset originale, dovuti a malfunzionamenti del sistema di rilevazione.

### 4.2.1 Valutazione oscillazioni del numero totale di slot

Una volta ottenuta l'informazione dell'oscillazione, data dalla differenza tra il numero di slot totali disponibili per una determinata stazione e la somma tra il numero di bici disponibili e di slot vuoti, ho ricavato il range di variazione dell'informazione ottenuta. Ho quindi ottenuto:

- Oscillazione minima: -23
- Oscillazione massima: 2

Questo indica che, considerano tutte le stazioni, nei casi estremi abbiamo 23 posti in meno e 2 in più nella somma di bici disponibili e posti liberi rispetto al totale di slot indicato dal file "station.csv". Nella figura 4.1 sono presenti dei grafici che mostrano il comportamento dell'oscillazione massima e minima per le diverse stazioni.




**(a)** Oscillazione minima **(b)** Oscillazione masima

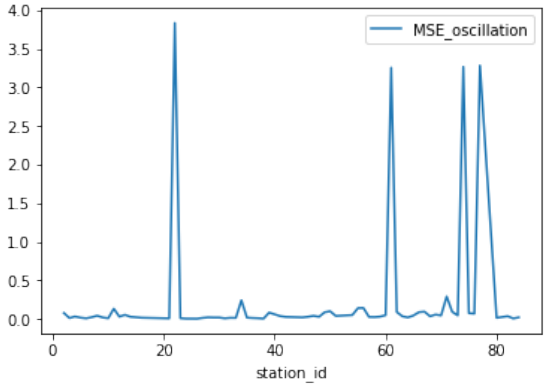
**Figura 4.1:** L'immagine (a) contiene il plot dell'oscillazione minima per ogni stazione. L'immagine (b) contiene, invece, il plot dell'oscillazione massima per ogni stazione.

Come è possibile notare, la situazione peggiore si ha considerando l'oscillazione minima. Questo significa che è più frequente per le diverse stazioni avere una somma di bici e posti disponibili minore del numero totale di posti effettivi.

Per valutare quali stazioni presentavano in media un'oscillazione più ampia ho calcolato il Mean Squared Error (MSE) per ogni stazione. Nella figura 4.2a è mostrato il plot dei risultati.

Dopo di che ho ordinando le stazioni per MSE decrescente, in modo da ottenere le stazioni con un elevato valore di Mean Squared Error. Come si può notare dalla

figura 4.2b, le stazioni con MSE più alto sono quelle con id 22, 61, 74 e 77.




(a)

| station_id | MSE_oscillation |
|---|---|
| 22 | 3.834335 |
| 77 | 3.282540 |
| 74 | 3.265925 |
| 61 | 3.254769 |
| 71 | 0.290300 |
| ... | ... |

(b)

**Figura 4.2:** L'immagine (a) rappresenta il plot dell' MSE dell'oscillazione al variare delle stazioni. L'immagine (b) contiene le stazioni con un valore elevato di Mean Squared Error.

Il passo successivo è stato quello di valutare, per le sole stazioni con picchi di MSE, il numero di situazioni in cui l'oscillazione ha avuto un valore assoluto >5, rispetto al totale di rilevamenti, in modo da capirne la percentuale. Come è possibile notare dai risultati di questo studio, rappresentati nella Tabella 4.1, il numero di situazioni critiche di questo tipo è irrilevante rispetto al numero totale di rilevamenti. Le percentuali di criticità ottenute sono infatti molto basse.

| id_stazione | # criticità | # tot rilevamenti | % criticità |
|---|---|---|---|
| 22 | 1 | 1047141 | 9.549812e-07 |
| 61 | 242 | 1047141 | 2.311055e-04 |
| 74 | 420 | 1047140 | 4.010925e-04 |
| 77 | 10 | 1047139 | 9.549831e-06 |

**Tabella 4.1:** La Tabella contiene i valori legati allo studio della percentuale di casi critici in cui l'oscillazione ha avuto un valore assoluto maggiore di 5.

A questo punto, per valutare se i picchi delle oscillazioni rappresentavano una situazione temporanea o meno, ho plottato il valore dell'oscillazione con il passare del tempo per le precedenti stazioni di interesse. I grafici risultanti, mostrati in Figura 4.3, mostrano che effettivamente i picchi dell'oscillazione sono generalmente costanti. Questo indica, quindi, che generalmente le situazioni in cui si sono avuti degli errori in lettura, da parte del sistema, sono rimaste costanti per un certo

intervallo di tempo.

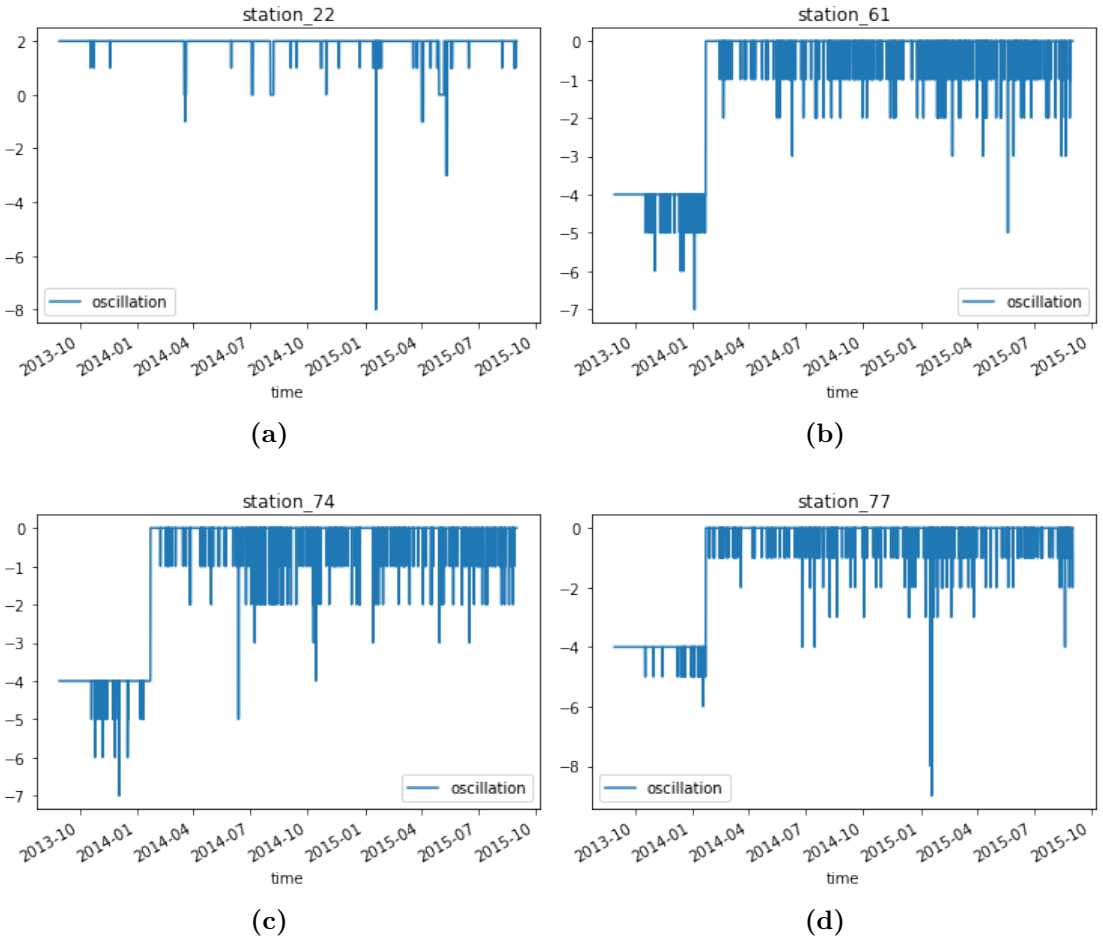



**Figura 4.3:** Le immagini (a), (b), (c) e (d) rappresentano il valore dell'oscillazione allo scorrere del tempo, rispettivamente per le stazioni 22, 61, 74 e 77.

Salvando in un dataframe (Figura 4.4) tutte le situazioni in cui si presentasse un'oscillazione in valore assoluto maggiore di 5, ho ottenuto un numero totale di righe pari a 6524. Questo numero è comunque molto piccolo considerando il fatto che il dataset di partenza contiene quasi 72 milioni di record. Inoltre, anche dalla Figura 4.4 è possibile notare che le situazioni con letture che presentano picchi dell'oscillazione in valore assoluto maggiore di 5, non sono soltanto letture temporanee, ma rappresentano degli istanti temporali consecutivi in cui il valore dell'oscillazione è costante. Questa è un'ulteriore conferma sul fatto che molto probabilmente queste letture errate siano state dovute ad un malfunzionamento

temporaneo.

Dato l'esiguo numero di record contenenti una situazione critica, ho deciso di filtrarli ottenendo così un dataset ripulito, che è stato poi utilizzato per tutte le fasi di lavoro successive.

| | station_id | time | oscillation |
|---|---|---|---|
| **1771793** | 3 | 2015-01-19 17:31:02 | -6 |
| **1771794** | 3 | 2015-01-19 17:32:03 | -6 |
| **1771795** | 3 | 2015-01-19 17:33:02 | -6 |
| **1771796** | 3 | 2015-01-19 17:34:02 | -6 |
| **1771797** | 3 | 2015-01-19 17:35:02 | -6 |
| **...** | ... | ... | ... |
| **71662241** | 84 | 2015-01-19 17:56:03 | -8 |
| **71662242** | 84 | 2015-01-19 17:57:02 | -8 |
| **71662243** | 84 | 2015-01-19 17:58:03 | -8 |
| **71662244** | 84 | 2015-01-19 17:59:02 | -8 |
| **71662245** | 84 | 2015-01-19 18:00:02 | -8 |

6524 rows × 3 columns

**Figura 4.4:** Esempio di dataframe ottenuto considerando tutte le situazioni in cui si presentasse un'oscillazione in valore assoluto maggiore di 5.

### 4.2.2 Valutazione della regolarità di raccolta dei dati

In questa fase l'obiettivo è stato quello di ottenere un stima approssimativa della regolarità della lettura dei dati a disposizione nel dataset *"status.csv"*.

Come si può notare dal dataset ottenuto per questo scopo (Figura 4.5), osservando la colonna *"frequency"* (*count/difference_in_minutes*), possiamo affermare che la frequenza media dei rilevamenti sia di circa un minuto per tutte le stazioni. Il rapporto non è 1 in quanto, dando un'occhiata veloce al dataset, effettivamente sembra che sia saltata qualche lettura con un salto di 2 minuti tra due letture successive.

Per avere una visione globale ho poi plottato la frequenza di rilevamento per tutte le stazioni. Dal grafico ottenuto, mostrato in Figura 4.6, è possibile notare che la frequenza media di tutte le stazioni si aggira intorno allo 0.99. In ultima analisi, possiamo quindi affermare che in media la frequenza di rilevamento sia di 1 minuto per tutte le stazioni.

| | station_id | start | end | count | difference_in_minutes | frequency |
|---|---|---|---|---|---|---|
| **0** | 2 | 2013-08-29 12:06:01 | 2015-08-31 23:59:02 | 1046898 | 1054793.0 | 0.992515 |
| **1** | 3 | 2013-08-29 12:06:01 | 2015-08-31 23:59:02 | 1047113 | 1054793.0 | 0.992719 |
| **2** | 4 | 2013-08-29 12:06:01 | 2015-08-31 23:59:02 | 1047100 | 1054793.0 | 0.992707 |
| **3** | 5 | 2013-08-29 12:06:01 | 2015-08-31 23:59:02 | 1047142 | 1054793.0 | 0.992746 |
| **4** | 6 | 2013-08-29 12:06:01 | 2015-08-31 23:59:02 | 1047142 | 1054793.0 | 0.992746 |
| **...** | ... | ... | ... | ... | ... | ... |

**Figura 4.5:** Esempio di dataframe ottenuto in modo da calcolare la regolarità con cui sono stati raccolti i dati.

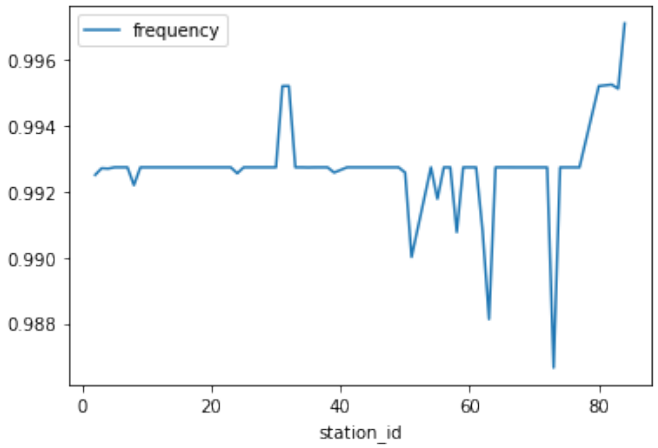



**Figura 4.6:** Grafico che mostra la frequenza di raccolta dei dati per tutte le stazioni.

### 4.2.3 Statistiche sulle situazioni critiche

L'ultima fase di analisi del dataset è consistita in una valutazione delle statistiche riguardanti le situazioni in cui una stazione si trovasse in una situazione critica. Per situazione critica, in questo caso, si intende una situazione in cui la stazione sia completamente vuota, completamente piena, quasi vuota o quasi piena. Negli ultimi due casi è stata definita una soglia appropriata per determinare il numero di bici o slot disponibili affiché si ricada in quella specifica situazione critica. In questo lavoro questa soglia è stata settata a 2. Quindi, se una stazione, ad un certo istante, contiene un numero di bici disponibili pari a 2 o 1, sarà considerata "QuasiVuota"; mentre se contiene un numero di slot disponibili pari a 2 o 1, sarà considerata "QuasiPiena".

L'analisi è stata condotta sul dataset ripulito dalle letture inconsistenti, contenente un numero totale di record pari a 71977910. I risultati ottenuti da questo studio, mostrati nella Tabella 4.2 e nella Figura 4.7, mostrano come lo stato critico "Piena" sia in generale più raro rispetto allo stato "Vuota". Mentre la combinazione di stati critici più frequente in assoluto è quella "QuasiVuota/Vuota".

| ***Stato/i critico/i*** | ***# criticità*** | ***% criticità*** |
|---|---|---|
| **Piena** | 326118 | 0.45% |
| **Vuota** | 527645 | 0.73% |
| **QuasiPiena** | 2027537 | 2.82% |
| **QuasiVuota** | 2783260 | 3.87% |
| **QuasiPiena/Piena** | 2353655 | 3.27% |
| **QuasiVuota/Vuota** | 3310905 | 4.60% |
| **Piena/Vuota** | 853763 | 1.19% |
| **Totale** | 5664560 | 7.87% |

**Tabella 4.2:** La Tabella contiene il numero di record in cui le stazioni si sono trovate nello o negli stati critici considerati e la percentuale di questo numero di record rispetto al totale dei record contenuti nel dataset ripulito.

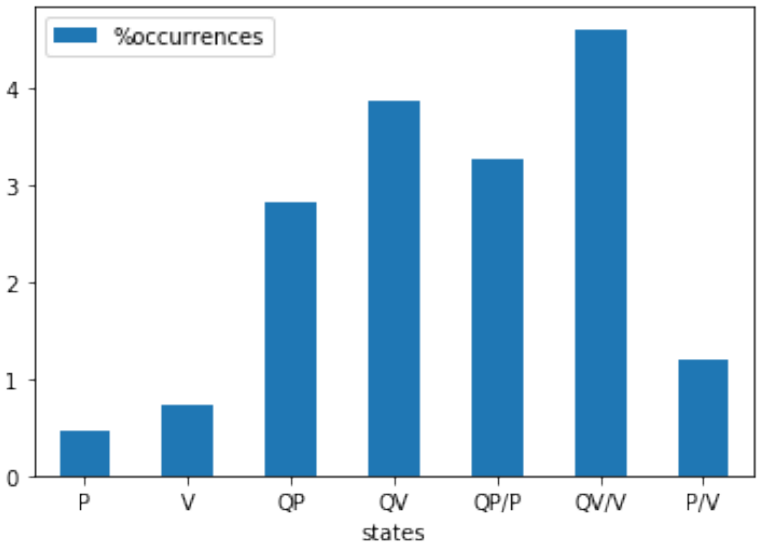



**Figura 4.7:** Grafico che mostra la percentuale di occorrenze per i diversi stati critici considerati.

## 4.3 Estrazione dei pattern

In questa fase sono stati eseguiti molteplici esperimenti, variando tutti i parametri a disposizione, in modo da provare a trovare una configurazione ottimale che permettesse di estrarre dei pattern di una qualità sufficientemente elevata, ovvero dei pattern con confidenze e supporti abbastanza alti. Questi pattern, infatti, serviranno poi al classificatore associativo nella fase di classificazione.

In particolare i parametri utilizzati per le diverse configurazioni sono stati i seguenti:

- Stato o stati considerati;
- intervallo temporale: la dimensione dello slot temporale considerato;
- intervallo spaziale: la dimensione dello slot spaziale considerato;
- numero di delta temporali: numero di diversi slot temporali considerati;
- numero di delta spaziali: numero di diversi slot spaziali considerati;
- supporto: supporto considerato per fare il training dell'algoritmo PrefixSpan.

Inoltre, sono state implementate anche diverse versioni dell'algoritmo di estrazione che mi hanno permesso di ottenere ulteriori configurazioni. Tali varianti dell'algoritmo sono:

- TimeSlots: una versione che mi ha permesso di dividere l'arco della giornata in diverse fasce orarie ed estrarre i pattern separatamente per ognuna di esse;
- StateChange: una versione che mi ha permesso di considerare come evento di interesse, nella fase di estrazione dei pattern, soltanto il passaggio da uno stato all'altro, piuttosto che tutte le occorrenze di stati critici.

Analizziamo adesso i risultati estratti per mezzo di diverse configurazioni, ottenute tramite varie combinazioni di valori dei parametri e versioni dell'algoritmo considerate.

### 4.3.1 Piena_Vuota

Partiamo dal considerare le diverse estrazioni, eseguite considerando come eventi di interesse le situazioni in cui le stazioni si trovassero nello stato "Piena" e lo stato "Vuota".

In questa fase è stata considerata la versione del codice di estrazione che tiene conto di tutti gli eventi critici che accadono e non il cambiamento di stato.

### Full_Empty_30min_1000meters_0support(3-3)

In questa fase ho iniziato ad estrarre i pattern considerando la combinazione di stati Piena_Vuota, con un intervallo temporale di 30 minuti, un intervallo spaziale di 1 km, soglia di supporto pari a 0, 3 intervalli temporali e 3 intervalli spaziali .

Nel grafico presente in Figura 4.8 è mostrato il numero di pattern estratti rispetto ai diversi valori di confidenza.

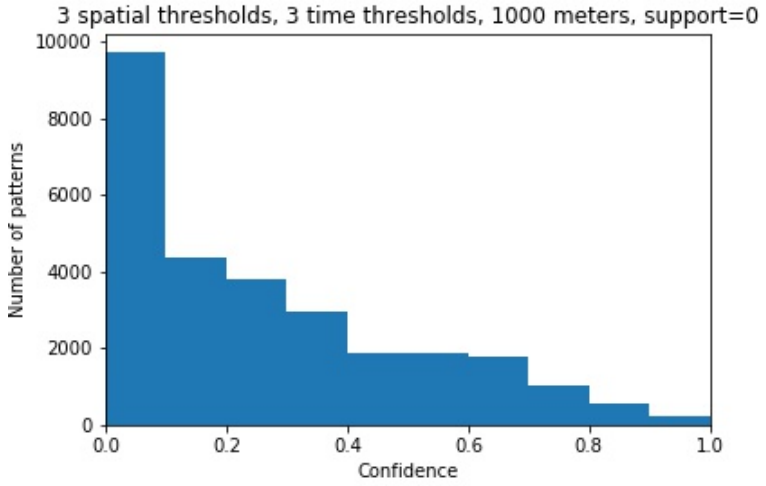
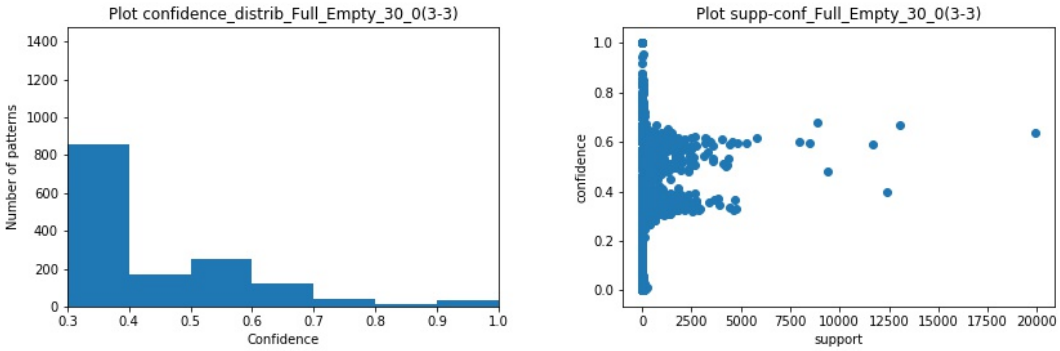



**Figura 4.8:** Istogramma che mostra il numero di pattern ottenuti dall'estrazione al varare della confidenza. L'intervallo temporale è stato settato a 30 minuti.

Dopo aver applicato il filtro che mantiene soltanto tutte le regole che nella testa considerano soltanto la stazione di riferimento, ho plottato nuovamente il numero di pattern rispetto ai valori di confidenza ed anche un scatterplot che mostra come sono distribuiti i pattern in base alla confidenza e al supporto (supporCount). Dal grafico in Figura 4.9b possiamo notare che le confidenze ottenute sono abbastanza basse. Inoltre, abbiamo due picchi a circa 0.35 e 0.6. Mentre dal grafico in Figura 4.9a, che indica la distribuzione delle regole rispetto alla confidenza, possiamo notare che si sono ottenute molte regole a confidenza bassa.

Come si può vedere dai grafici, in questo primo esperimento, considerando un intervallo di 30 minuti, ho ottenuto molti pattern con una confidenza bassa e pochi con una confidenza alta. Inoltre, i pattern con confidenza alta hanno un supportCount molto basso. Quindi, ho proseguito con altre prove diminuendo l'intervallo temporale a 15 minuti e provando ad aumentare il numero di delta temporali, in modo da valutare se, in questo modo, sarebbe stato possibile accorgersi che una stazione stava cambiando e quindi di reagire in tempo.




**(a)** **(b)**

**Figura 4.9:** L'istogramma (a) rappresenta il numero dei pattern ottenuti dopo il filtraggio; lo scatterplot (b) mostra la distribuzione dei pattern filtrati in base alla confidenza ed il supporto.

### Full_Empty_15min_1000meters_0support(3-3)

In questa fase ho estratto i pattern considerando la combinazione di stati Piena_Vuota, con un intervallo temporale di 15 minuti, un intervallo spaziale di 1 km, soglia di supporto pari a 0, 3 intervalli temporali e 3 intervalli spaziali.

Nel grafico presente in Figura 4.10 è mostrato il numero di pattern estratti rispetto ai diversi valori di confidenza.

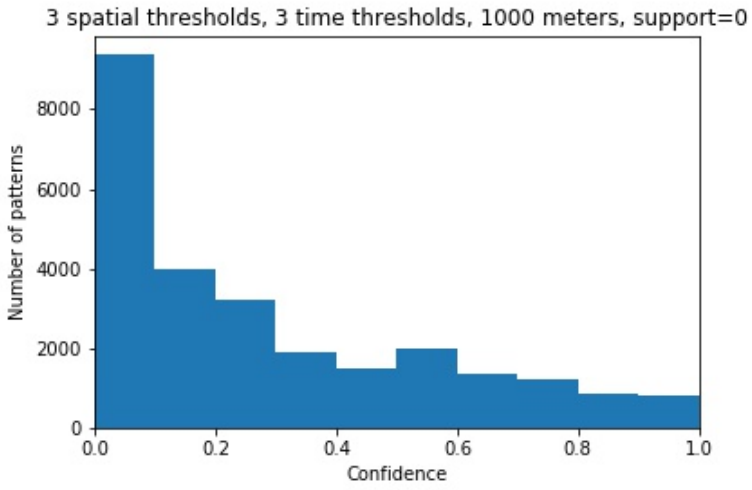
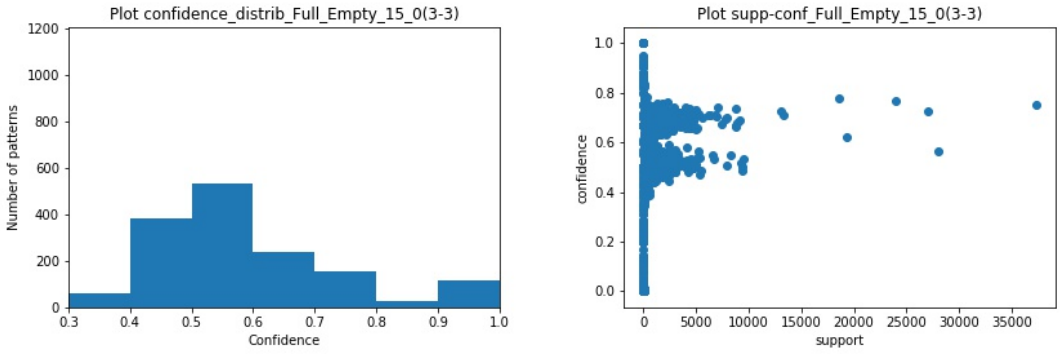



**Figura 4.10:** Istogramma che mostra il numero di pattern ottenuti dall'estrazione al varare della confidenza. L'intervallo temporale è stato settato a 15 minuti.

Dopo aver applicato il filtro, ho plottato nuovamente il numero di pattern rispetto ai valori di confidenza ed anche un scatterplot che mostra come sono distribuiti i pattern in base alla confidenza e al supporto (supporCount).

Considerando il grafico in Figura 4.11b e confrontandolo con quello ottenuto nel caso dei 30 minuti (Figura 4.9b), notiamo che, in questo caso, la confidenza è un po' più alta. Infatti, i due picchi sono uno sullo 0.5 e l'altro sullo 0.7, mentre dall'altra parte erano un po' più in basso. Questo indica il fatto che diminuendo l'intervallo temporale, si è ottenuta qualche regola a confidenza più alta, come è possibile notare anche dal grafico della distribuzione delle regole rispetto alla confidenza presente in Figura 4.11a.




**(a)** **(b)**

**Figura 4.11:** L'istogramma (a) rappresenta il numero dei pattern ottenuti dopo il filtraggio; lo scatterplot (b) mostra la distribuzione dei pattern filtrati in base alla confidenza ed il supporto.

Per capire quante volte si verifica il passaggio da Vuota a Piena e viceversa, riporto di seguito i pattern estratti che rappresentano questa situazione:

$$[[[['Vuota\_T0\_0'], ['Piena\_T1\_0']]], ['0.0012656 - 63']] \tag{4.1}$$

$$[[[['Piena\_T0\_0'], ['Vuota\_T1\_0']]], ['0.0023673 - 74']] \tag{4.2}$$

Come possiamo vedere, le confidenze sono abbastanza basse. Questo vuol dire che si passa da uno stato all'altro molto di rado in 15 minuti. Tuttavia, il passaggio da Piena a Vuota ha una confidenza di circa il doppio rispetto al passaggio da Vuota a Piena. Questo significa, sostanzialmente, che è più probabile che una stazione sia vuota piuttosto che Piena, come d'altronde già constatato nell'analisi iniziale del dataset.

Se invece andiamo ad analizzare il pattern che indica che se una stazione è Vuota in un certo istante di tempo, allora sarà ancora Vuota nell'istante di tempo successivo:

$$[[[['Vuota\_T0\_0'], ['Vuota\_T1\_0']]], ['0.7486941 - 37267']] \tag{4.3}$$

possiamo vedere che ha una confidenza quasi del 75%. Quindi capita molto spesso.

Mentre, considerando la stessa situazione per lo stato Piena:

$$[[[['Piena\_T0\_0'], ['Piena\_T1\_0']]], ['0.7683792 - 24018']] \tag{4.4}$$

anche in questo caso abbiamo una confidenza abbastanza alta. Questo indica che il 75/76% delle volte una stazione mantiene il suo stato, quindi un classificatore "stupido", che predice con lo stato precedente, nel 76% di casi farebbe una predizione corretta.

### Full_Empty_15min_1000meters_0support(4-3)

In questa fase ho estratto i pattern considerando la combinazione di stati Piena_Vuota, con un intervallo temporale di 15 minuti, un intervallo spaziale di 1 km, soglia di supporto pari a 0, 4 intervalli temporali e 3 intervalli spaziali.

Nel grafico presente in Figura 4.12 è mostrato il numero di pattern estratti rispetto ai diversi valori di confidenza.

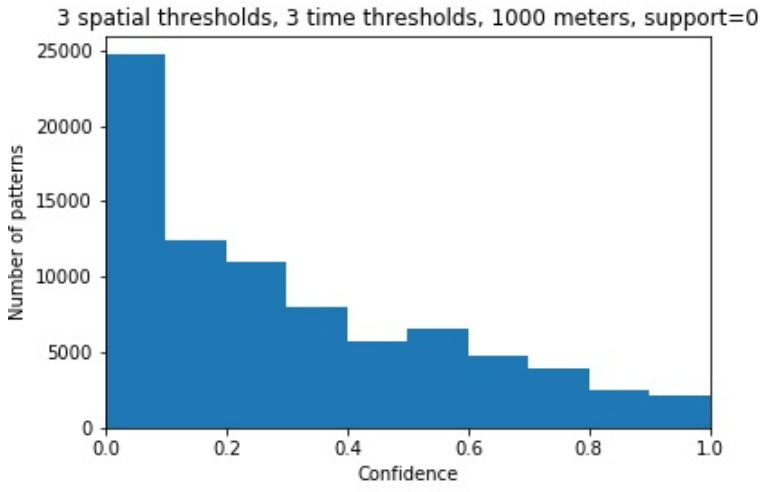
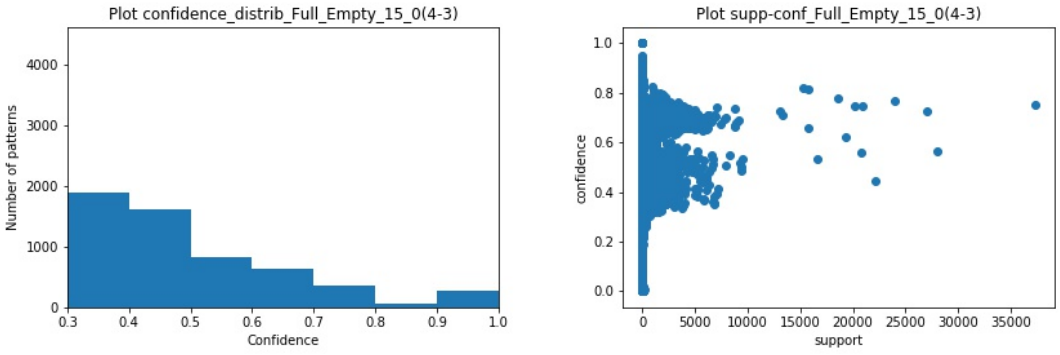



**Figura 4.12:** Istogramma che mostra il numero di pattern ottenuti dall'estrazione al varare della confidenza. L'intervallo temporale è stato settato a 15 minuti.

Dopo aver applicato il filtro, ho plottato nuovamente i soliti due grafici che mostrano la distribuzione dei pattern rispetto ai valori di confidenza e lo scatterplot dei pattern in base ai valori di confidenza e supporto (supporCount). I suddetti grafici sono presenti in Figura 4.13.




**(a)** **(b)**

**Figura 4.13:** L'istogramma (a) rappresenta il numero dei pattern ottenuti dopo il filtraggio; lo scatterplot (b) mostra la distribuzione dei pattern filtrati in base alla confidenza ed il supporto.

In questo caso si sono ottenute quindi delle regole un po' più lunghe che prima invece erano assenti, tuttavia, esse non sono caratterizzate da un supporto molto alto. Questo significa che conta molto di più l'informazione sulla granularità dell'intervallo temporale considerato piuttosto che il numero di delta temporali precedenti da considerare.

### 4.3.2 Vuota_QuasiVuota

In questa sezione verranno prese in considerazione le estrazioni eseguite considerando come eventi di interesse le situazioni in cui le stazioni si trovassero nello stato "Vuota" e lo stato "QuasiVuota".

Anche in questa fase è stata considerata la versione del codice di estrazione che tiene conto di tutti gli eventi critici che accadono e non il cambiamento di stato.

#### Empty_almostEmpty_30min_1000meters_0support(3-3)

In questa fase ho estratto i pattern considerando un intervallo temporale di 30 minuti, un intervallo spaziale di 1 km, soglia di supporto pari a 0, 3 intervalli temporali e 3 intervalli spaziali.

La distribuzione dei pattern estratti rispetto ai valori di confidenza è mostrata nel grafico in Figura 4.14.

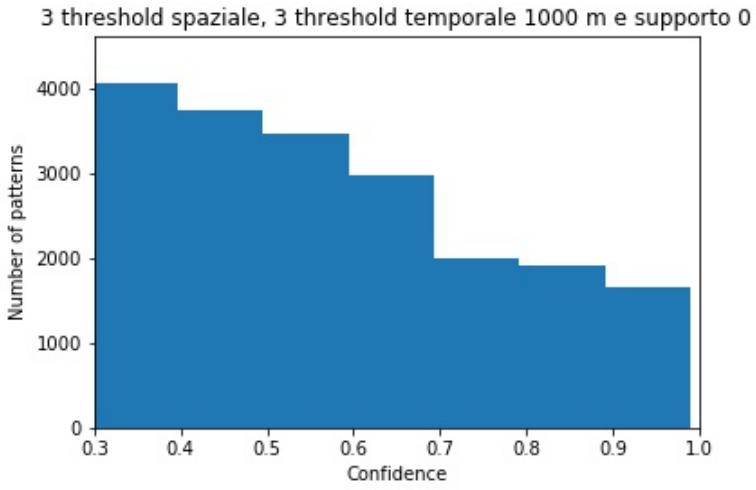
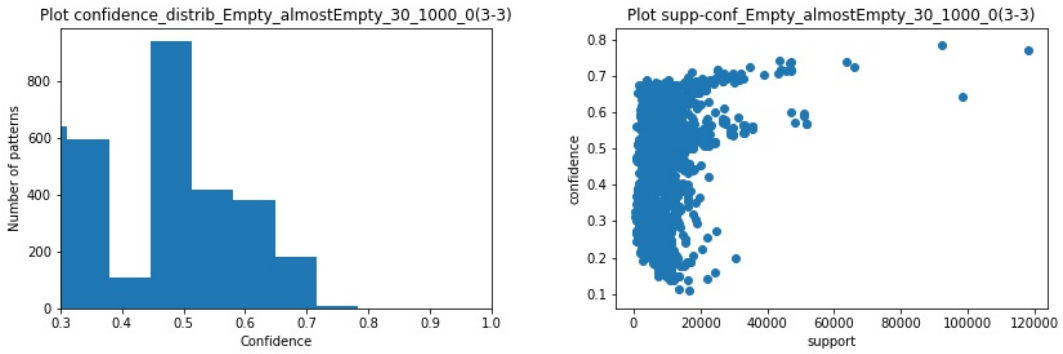



**Figura 4.14:** Istogramma che mostra il numero di pattern ottenuti dall'estrazione al varare della confidenza. L'intervallo temporale è stato settato a 30 minuti.

I grafici presenti in Figura 4.15, mostrano la situazione dei pattern dopo aver applicato il filtro, in modo da tenere in considerazione soltanto i pattern di interesse.




**(a)** **(b)**

**Figura 4.15:** L'istogramma (a) rappresenta il numero dei pattern ottenuti dopo il filtraggio; lo scatterplot (b) mostra la distribuzione dei pattern filtrati in base alla confidenza ed il supporto.

Ho valutato, inoltre, quante volte si presentano dei casi in cui una stazione si trova nello stato "Vuota" o "QuasiVuota", prelevando tali regole dall'output del PrefixSpan:

$$([['Vuota\_T0\_0']], 53311) \tag{4.5}$$

$$([['QuasiVuota\_T0\_0']], 152819) \tag{4.6}$$

Come possiamo vedere, lo stato "QuasiVuota" si verifica circa il triplo delle volte in cui si presenta lo stato "Vuota".

Inoltre, le regole a confidenza più alta sono quelle che indicano una situazione stazionaria in cui una stazione che si trova nello stato "QuasiVuota", riamane tale anche negli istanti successivi. Questo, infatti, è confermato dalle due regole riportate come esempio:

$$[[[['QuasiVuota\_T0\_0'], ['QuasiVuota\_T1\_0'], ['QuasiVuota\_T2\_0']]], ['0.78 - 92363']] \tag{4.7}$$

$$[[[['QuasiVuota\_T0\_0'], ['QuasiVuota\_T1\_0']]], ['0.77 - 117916']] \tag{4.8}$$

### Empty_almostEmpty_15min_1000meters_0support(3-3)

In questa fase ho estratto i pattern considerando la combinazione di stati Vuota_QuasiVuota, con un intervallo temporale di 15 minuti, un intervallo spaziale di 1 km, soglia di supporto pari a 0, 3 intervalli temporali e 3 intervalli spaziali.

La distribuzione dei pattern estratti rispetto ai valori di confidenza è mostrata nel grafico in Figura 4.16.

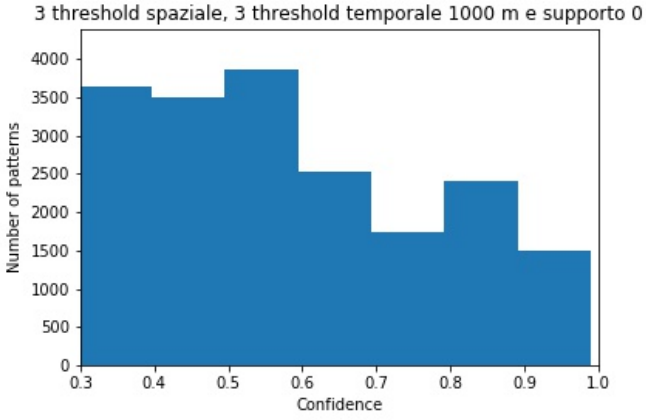
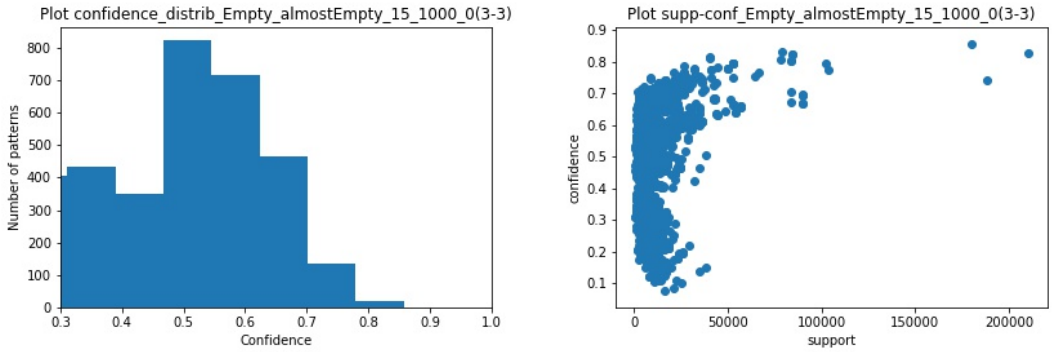



**Figura 4.16:** Istogramma che mostra il numero di pattern ottenuti dall'estrazione al varare della confidenza. L'intervallo temporale è stato settato a 15 minuti.

I grafici presenti in Figura 4.17, mostrano, invece, la situazione dei pattern dopo aver applicato il filtro, in modo da tenere in considerazione soltanto i pattern di interesse.




**(a)** **(b)**

**Figura 4.17:** L'istogramma (a) rappresenta il numero dei pattern ottenuti dopo il filtraggio; lo scatterplot (b) mostra la distribuzione dei pattern filtrati in base alla confidenza ed il supporto.

Come è possibile notare, dopo il filtraggio si sono persi molti pattern a confidenza più alta.

Analizzando i pattern ottenuti, anche in questo caso quelli a confidenza più alta indicano una certa stazionarietà. Infatti:

$$[[[['QuasiVuota\_T0\_0'], ['QuasiVuota\_T1\_0'], ['QuasiVuota\_T2\_0']]], ['0.86 - 180033']] \tag{4.9}$$

$$[[[['QuasiVuota\_T0\_0, QuasiVuota\_T0\_1'], ['QuasiVuota\_T1\_0, QuasiVuota\_T1\_1'], ['QuasiVuota\_T2\_0']]], ['0.83 - 79059']] \tag{4.10}$$

Concentrandomi invece sul passaggio da "QuasiVuota" a "Vuota" ho ricavato alcuni pattern come:

$$[[[['QuasiVuota\_T0\_0'], ['Vuota\_T1\_2, Vuota\_T1\_2, Vuota\_T1\_0'], ['Vuota\_T2\_0']]], ['0.7157 - 5421']] \tag{4.11}$$

$$[[[['QuasiVuota\_T0\_0'], ['Vuota\_T1\_1, Vuota\_T1\_1, Vuota\_T1\_0'], ['Vuota\_T2\_0']]], ['0.7146 - 4028']] \tag{4.12}$$

Come possiamo vedere, anche in questo caso è indicata una certa stabilità. Infatti, se all'istante T0 la stazione di riferimento era QuasiVuota e poi successivamente si era svuotata, allora rimane Vuota.

### 4.3.3 Piena_QuasiPiena

In questa sezione verranno esaminati i pattern estratti considerando come eventi di interesse le situazioni in cui le stazioni si trovassero nello stato "Piena" e lo stato "QuasiPiena".

Anche in questa fase, come nelle configurazioni precedenti, è stata considerata la versione del codice di estrazione che tiene conto di tutti gli eventi critici che accadono e non il cambiamento di stato.

#### Full_almostFull_30min_1000meters_0support(3-3)

In questa fase ho estratto i pattern considerando un intervallo temporale di 30 minuti, un intervallo spaziale di 1 km, soglia di supporto pari a 0, 3 intervalli temporali e 3 intervalli spaziali.

La distribuzione dei pattern estratti rispetto ai valori di confidenza è mostrata nel grafico in Figura 4.18.

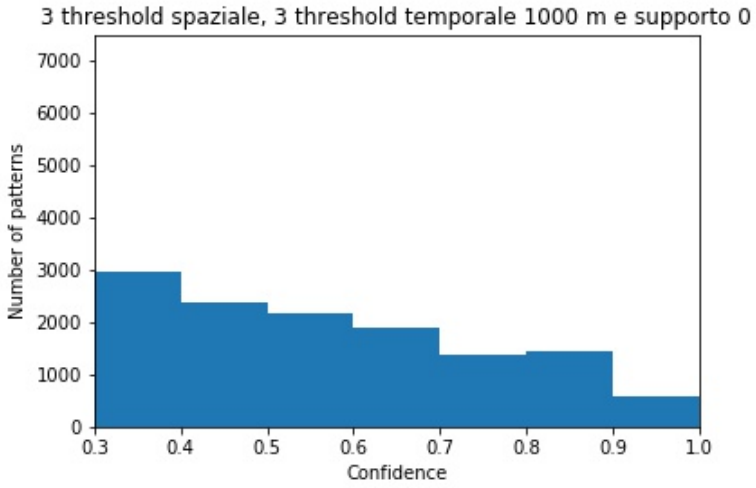
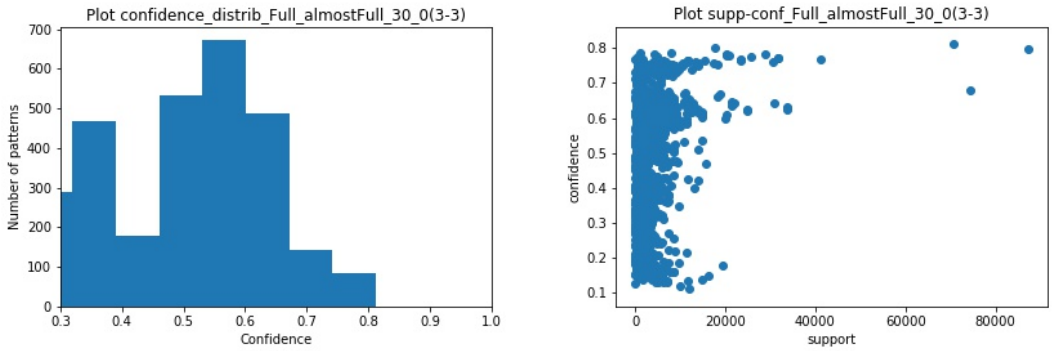



**Figura 4.18:** Istogramma che mostra il numero di pattern ottenuti dall'estrazione al varare della confidenza. L'intervallo temporale è stato settato a 30 minuti.

I grafici presenti in Figura 4.19, mostrano, invece, la situazione dei pattern dopo aver applicato il filtro, in modo da tenere in considerazione soltanto i pattern di interesse.




**(a)** **(b)**

**Figura 4.19:** L'istogramma (a) rappresenta il numero dei pattern ottenuti dopo il filtraggio; lo scatterplot (b) mostra la distribuzione dei pattern filtrati in base alla confidenza ed il supporto.

Dopo il filtraggio si sono perse alcune regole a confidenza più alta. Quelle rimaste, invece, hanno una confidenza in media abbastanza bassa.

Ho valutato, inoltre, quante volte si presentano dei casi in cui una stazione si trova nello stato "Piena" o "QuasiPiena", prelevando tali regole dall'output del PrefixSpan:

$$['Piena\_T0\_0;\ 33304' \\ 'QuasiPiena\_T0\_0;\ 109021',\ ...] \tag{4.13}$$

Come possiamo vedere, anche per il caso "Piena" e "QuasiPiena", lo stato "QuasiPiena" si verifica circa il triplo delle volte rispetto a quelle in cui si presenta lo stato "Piena". Inoltre, questi numeri, confrontati con quelli ottenuti nel caso di Vuota_QuasiVuota, confermano le statistiche ottenute nella prima fase di analisi del dataset. Infatti, i casi in cui le stazioni si trovano negli stati "Vuota" o "QuasiVuota", sono maggiori rispetto ad i casi in cui le stazioni si trovano nello stato "Piena" o "QuasiPiena".

Inoltre, per studiare anche in questo caso la stazionarietà dello stato delle stazioni, ho prelevato il pattern che indica la probabilità che una stazione si trovi nello stato "QuasiPiena" in un certo istante di tempo e rimanga tale anche nell'intervallo di tempo successivo. Il pattern prelevato è il seguente:

$$[[[['QuasiPiena\_T0\_0'], ['QuasiPiena\_T1\_0']]], ['0.79920 - 87130']] \tag{4.14}$$

Come possiamo vedere, quasi nell'80% dei casi una stazione che si trova nello stato "QuasiPiena", rimane in tale stato anche nell'istante di tempo successivo. Questo conferma ancora una volta che i dati siano molto stazionari.

### Full_almostFull_15min_1000meters_0support(3-3)

In questa fase ho estratto i pattern considerando la combinazione di stati Piena_QuasiPiena, con un intervallo temporale di 15 minuti, un intervallo spaziale di 1 km, soglia di supporto pari a 0, 3 intervalli temporali e 3 intervalli spaziali.

La distribuzione dei pattern estratti rispetto ai valori di confidenza è mostrata nel grafico in Figura 4.20.

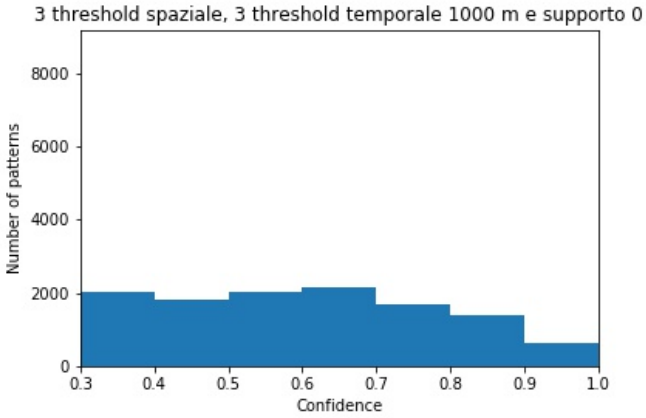



**Figura 4.20:** Istogramma che mostra il numero di pattern ottenuti dall'estrazione al varare della confidenza. L'intervallo temporale è stato settato a 15 minuti.

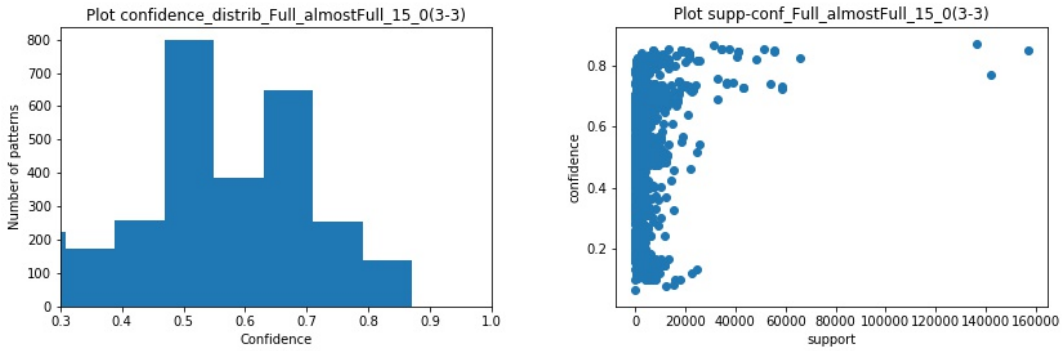
I grafici presenti in Figura 4.21, mostrano, invece, la situazione dei pattern dopo aver applicato il filtro, in modo da tenere in considerazione soltanto i pattern di interesse.

Anche in questo caso dopo il filtraggio si sono perse alcune regole a confidenza alta. Infatti, la confidenza più alta è stata di circa l'85%. Inoltre, come possiamo vedere dallo scatterplot in Figura 4.21b, ci sono pochi pattern con supporto elevato.

Prendendo in esempio alcuni pattern che contengono nel conseguente lo stato "Piena", tra quelli a confidenza più alta si hanno:

$$[[[['Piena\_T0\_0, Piena\_T0\_3, Piena\_T0\_3'], ['Piena\_T1\_0'], ['Piena\_T2\_0']]], ['0.7491 - 230']] \tag{4.15}$$

$$[[[['Piena\_T0\_0'], ['Piena\_T1\_0'], ['Piena\_T2\_0']]], ['0.7344 - 24217']] \tag{4.16}$$




**(a)** **(b)**

**Figura 4.21:** L'istogramma (a) rappresenta il numero dei pattern ottenuti dopo il filtraggio; lo scatterplot (b) mostra la distribuzione dei pattern filtrati in base alla confidenza ed il supporto.

Come è possibile notare, i pattern non contengono nessuna informazione sullo stato "QuasiPiena". Questo vuol dire che lo stato "QuasiPiena" non è molto utile nel determinare se lo stato della stazione di riferimento sarà "Piena".

### 4.3.4 Tutti gli stati

A questo punto si è provato ad incrociare le informazioni sui vari stati, considerando la versione del codice di estrazione che tiene conto di tutti gli stati critici. Ho eseguito quindi le estrazioni considerando gli stati: Piena, QuasiPiena, Vuota e QuasiVuota. Tuttavia, in questo caso, settando un supporto pari a 0, l'algoritmo non è riuscito a convergere. Partendo, quindi, da un supporto pari al 15%, ho provato a ridurlo il più possibile in modo da estrarre più pattern possibili. Le configurazioni per cui sono riuscito ad ottenere dei pattern sono state le seguenti:

- 15min_1000m_01sup(3-3) (10% di supporto)
- 15min_1000m_01sup(4-3) (10% di supporto)
- 15min_1000m_005sup(3-3) (5% di supporto)
- 15min_1000m_005sup(4-3) (5% di supporto)
- 30min_1000m_01sup(3-3) (10% di supporto)
- 30min_1000m_015sup(3-3) (15% di supporto)
- 30min_1000m_005sup(3-3) (5% di supporto)

Di seguito si esaminerà la configurazione più interessante.

### AllStates_15min_1000meters_005support(4-3)

In questa fase di estrazione ho considerato tutti gli stati, un intervallo temporale di 15 minuti, un intervallo spaziale di 1 km, una soglia di supporto del 5%, 4 intervalli temporali e 3 intervalli spaziali.

La Figura 4.22 mostra la distribuzione dei pattern ottenuti al variare della confidenza.

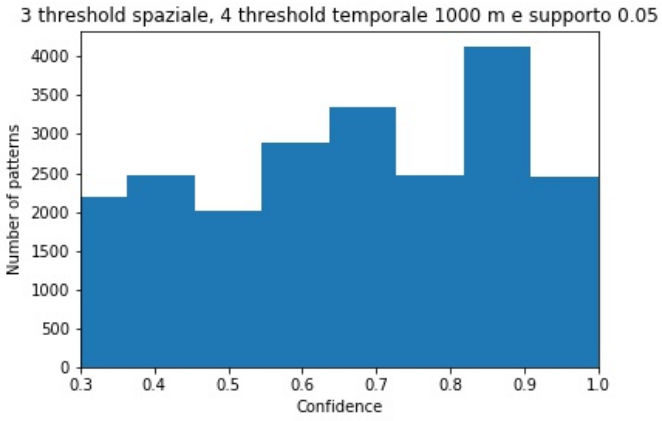
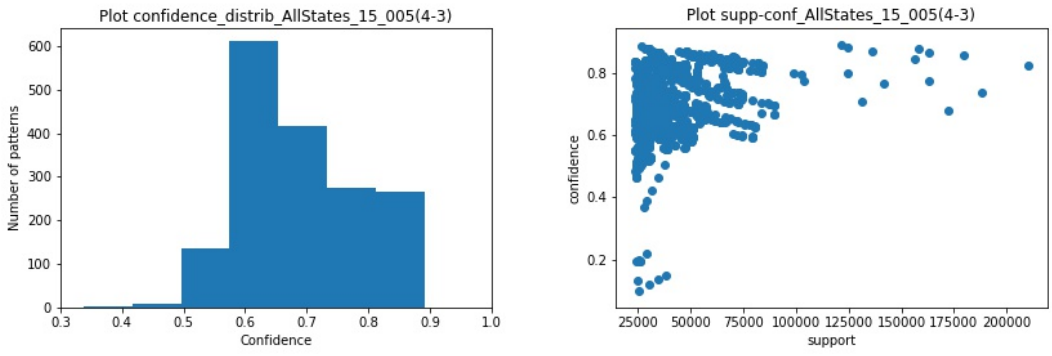



**Figura 4.22:** Istogramma che mostra il numero di pattern ottenuti dall'estrazione al varare della confidenza. L'intervallo temporale è stato settato a 15 minuti.

Dopo il filtraggio, invece, la distribuzione dei risultati ottenuti è mostrata in Figura 4.23.




**(a)** **(b)**

**Figura 4.23:** L'istogramma (a) rappresenta il numero dei pattern ottenuti dopo il filtraggio; lo scatterplot (b) mostra la distribuzione dei pattern filtrati in base alla confidenza ed il supporto.

Quindi abbiamo avuto qualche pattern a confidenza alta, ma la maggior parte è incentrata sul 60%. Inoltre, i supporti ottenuti sono stati in generale abbastanza elevati.

Tuttavia, analizzando i pattern estratti, ho notato che essi non presentavano quasi nessuna occorrenza degli stati "Piena" o "Vuota". Questo sicuramente è stato dovuto al fatto che le situazioni in cui le stazioni fossero piene o vuote erano molte meno rispetto a quelle in cui le stazioni fossero quasi piene o quasi vuote. Questo aspetto è stato confermato anche dal risultato dell'output del PrefixSpan:

$$\begin{aligned}&[\prime Vuota\_T0\_0;\ 75483\prime,\\ &\prime QuasiPiena\_T0\_0;\ 184907\prime,\\ &\prime QuasiVuota\_T0\_0;\ 254443\prime,\\ &\ \ \prime Piena\_T0\_0;\ 47933\prime,\ ...]\end{aligned} \tag{4.17}$$

Come possiamo notare, infatti, il supportCount degli stati "Piena" e "Vuota" è di un ordine di grandezza inferiore rispetto agli stati "QuasiPiena" e "QuasiVuota".

Inoltre, non si registrano nemmeno passaggi di stato da "QuasiPiena" a "Quasi-Vuota" e viceversa. Questo è comprensibile in quanto, considerando un intervallo temporale di 15 minuti, è difficile che una stazione passi da uno stato all'altro.

In ultima analisi, quindi, la presente configurazione, dove si è tenuto conto di tutti gli stati, non mi ha permesso di estrarre dei pattern significativi ed utili al fine della predizione.

### 4.3.5 TimeSlots

Un altro esperimento che è stato condotto, in modo da provare ad ottenere dei pattern migliori, è stato quello di dividere il dataset in fasce orarie diverse e quindi estrarre i pattern per ogni fascia oraria.

Il motivo per cui è stato deciso di effettuare questo esperimento è stato quello di controllare se fosse possibile estrarre dei pattern più interessanti ai fini della classificazione, per alcune determinate fasce della giornata tra quelle considerate. Le fasce orarie considerate sono state:

- 0 - 6
- 6 - 10
- 10 - 14
- 14 - 17

- 17 - 20
- 20 - 24

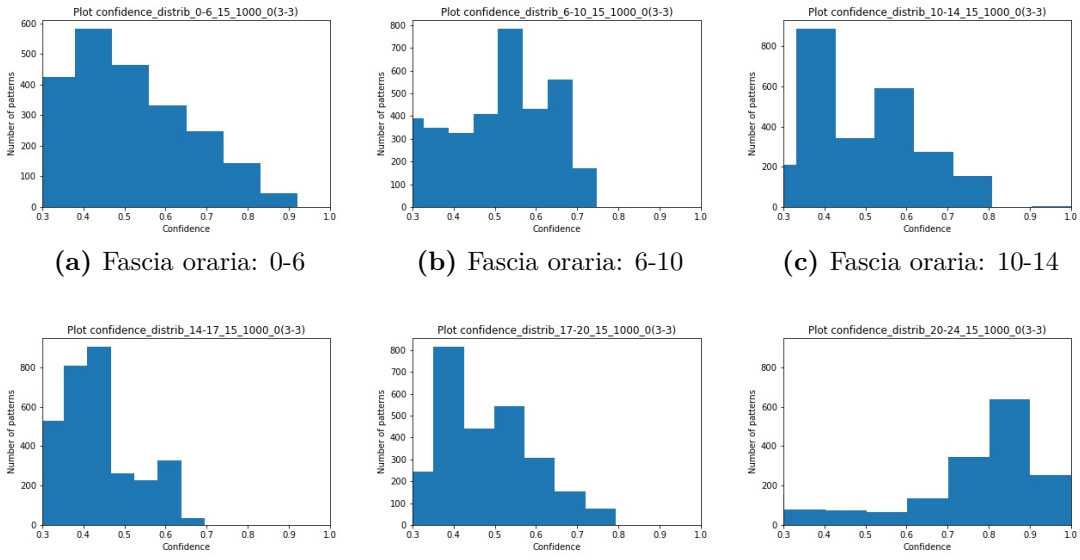
Verranno di seguito esaminati i pattern estratti considerando come eventi di interesse le situazioni con stati "Piena_QuasiPiena" e "Vuota_QuasiVuota" e come intervalli temporali 15 e 30 minuti.

Anche in quest'ultima fase è stata considerata la versione del codice di estrazione che tiene conto di tutti gli eventi critici che accadono.

Di seguito verranno analizzati i risultati ottenuti dall'estrazione settando l'intervallo temporale a 15 minuti.

### Time_Slots_Empty_almostEmpty_15min_1000meters_0support(3-3)

Eseguendo l'estrazione dei pattern separatamente per le diverse fasce orarie, si sono ottenuti alcuni cambiamenti. Nella Figura 4.24 è mostrata la distribuzione dei pattern rispetto alla confidenza per le diverse fasce orarie, una volta aver effettuato il solito filtraggio.




**(a)** Fascia oraria: 0-6 **(b)** Fascia oraria: 6-10 **(c)** Fascia oraria: 10-14

**(d)** Fascia oraria: 14-17 **(e)** Fascia oraria: 17-20 **(f)** Fascia oraria: 20-24

**Figura 4.24:** La figura contiene l'insieme degli istogrammi che mostrano la distribuzione dei pattern in base alla confidenza, per le diverse fasce orarie considerate.

In particolare, possiamo osservare che in alcune fasce, come la fascia 0-6 e la fascia 20-24, si sono ottenuti alcuni pattern con confidenze più alte. Tuttavia, questi pattern indicano una situazione stazionaria, come è possibile osservare dai seguenti pattern di esempio estratti dalla fascia **0-6**:

$$[[[['QuasiVuota\_T0\_0'], ['QuasiVuota\_T1\_0'], ['QuasiVuota\_T2\_0']]], ['0.9215 - 41898']] \tag{4.18}$$

$$[[[['QuasiVuota\_T0\_0, QuasiVuota\_T0\_1'], ['QuasiVuota\_T1\_0, QuasiVuota\_T1\_1'], ['QuasiVuota\_T2\_0']]], ['0.9078 - 19341']] \tag{4.19}$$

Mentre in altre fasce, come la fascia **14-17**, la situazione è meno stazionaria, ma si hanno dei pattern a confidenza leggermente più bassa:

$$[[[['QuasiVuota\_T0\_0, Vuota\_T0\_3'], ['Vuota\_T1\_1, Vuota\_T1\_0'], ['Vuota\_T2\_0']]], ['0.6690 - 1146']] \tag{4.20}$$

$$[[[['QuasiVuota\_T0\_0, QuasiVuota\_T0\_2'], ['Vuota\_T1\_1, Vuota\_T1\_0'], ['Vuota\_T2\_0']]], ['0.6506 - 3220']] \tag{4.21}$$

**Time_Slots_Full_almostFull_15min_1000meters_0support(3-3)**

In questo caso si sono ottenute delle regole con confidenze leggermente più alte rispetto al caso "Vuota_QuasiVuota". Nella figura 4.25 è mostrata la distribuzione dei pattern rispetto alla confidenza per le diverse fasce orarie.

Anche in questo caso si è confermato il trend di stazionarietà in alcune fasce con pattern a confidenza più alta, ed una certa dinamicità in fasce con pattern a confidenza generalmente più bassa.

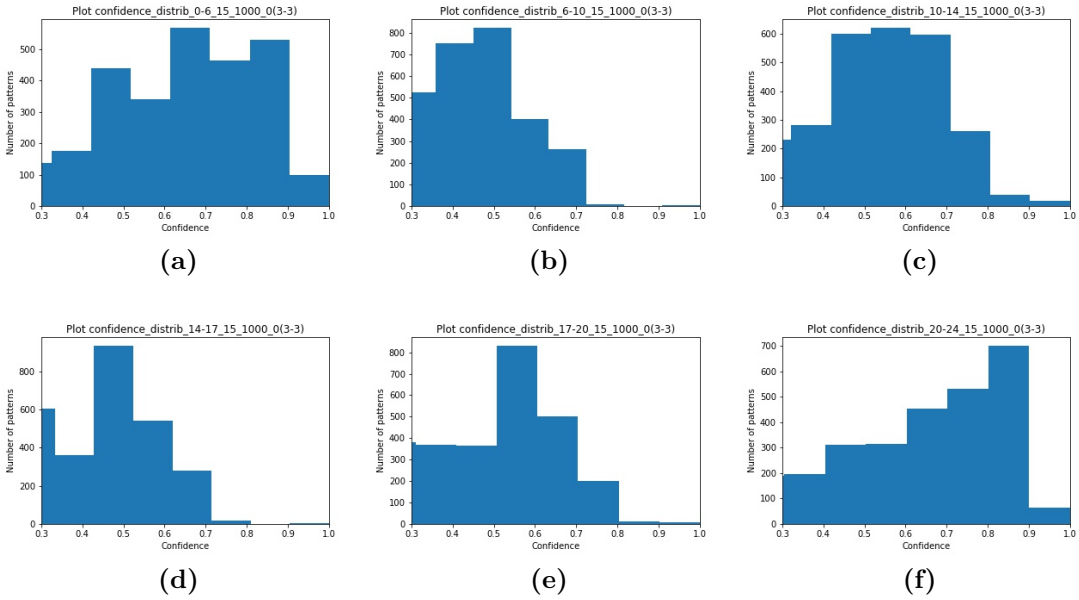



**Figura 4.25:** La figura contiene l'insieme degli istogrammi che mostrano la distribuzione dei pattern in base alla confidenza, per le diverse fasce orarie considerate.

### 4.3.6 StateChange

Vista la stazionarietà delle situazioni critiche mostrate dai pattern estratti, ho provato ad ottenere la probabilità che ha una stazione, che si trova in un certo stato, di rimanere, negli istanti di tempo successivi, nello stato attuale, in uno stato immediatamente adiacente, oppure uno stato normale.

Inizialmente è stato tenuto conto dei passaggi di stato solo da una situazione critica ad un'altra. In particolare, sono stati considerati come eventi di interesse, quei momenti in cui una stazione sia passata allo stato "Vuota" o "QuasiVuota", per la configurazione Vuota_QuasiVuota, oppure quei momenti in cui una stazione sia passata allo stato "Piena" o "QuasiPiena", per la configurazione "Piena_QuasiPiena".

Oltre a queste due configurazioni, se ne sono poi aggiunte altre due che tengono conto anche del passaggio della stazione allo stato "Normale". In precedenza, non era stato possibile aggiungere il nuovo stato "Normale" in quanto non si teneva conto soltanto del cambiamento di stato, ma di tutti gli istanti di tempo in cui una stazione si trovava in un certo stato. In quel caso, infatti, se avessi inserito lo stato Normale, sarebbe aumentato a dismisura il numero di item, facendo esplodere il

problema. Questo, ovviamente, sarebbe dovuto al fatto che il numero di situazioni in cui una stazione si trova in uno stato normale è sicuramente molto elevato. Utilizzando, invece, la variante dell'algoritmo di estrazione che tiene conto soltanto del passaggio da uno stato all'altro, sono riuscito a considerare lo stato "Normale" nella generazione dei pattern.

Come eventi di interesse sono stati quindi considerati i seguenti passaggi di stato:

- QuasiVuota -> Normale
- Normale -> QuasiVuota
- QuasiVuota -> Vuota
- Vuota -> QuasiVuota
- Normale -> Vuota
- Vuota -> Normale

Ovviamente la stessa cosa è stata fatta per il caso "Piena_QuasiPiena".

### StateChange_Empty_almostEmpty

In questa prima fase sono stati considerati gli stati "Vuota" e "QuasiVuota". Le configurazioni che sono state provate ed hanno dato dei risultati in output sono le seguenti:

- StateChange_Empty_almostEmpty_15min_1000m_sup0 (3-3)
- StateChange_Empty_almostEmpty_15min_1000m_sup0 (4-3)
- StateChange_Empty_almostEmpty_15min_1000m_sup005 (5-3)
- StateChange_Empty_almostEmpty_15min_500m_sup0 (3-3)
- StateChange_Empty_almostEmpty_15min_500m_sup002 (4-3)
- StateChange_Empty_almostEmpty_15min_500m_sup002 (5-3)
- StateChange_Empty_almostEmpty_30min_1000m_sup0 (3-3)
- StateChange_Empty_almostEmpty_30min_1000m_sup0 (4-3)
- StateChange_Empty_almostEmpty_30min_1000m_sup005 (5-3)

dove il primo numero tra le parentesi rappresenta il numero di istanti temporali considerati, mentre il secondo rappresenta il numero di delta spaziali considerati.

Come è possibile vedere, spesso per far convergere l'algoritmo ho dovuto aumentare il supporto minimo. Le regole estratte con questa configurazione, tuttavia, non sono state molto utili, in quanto hanno avuto una confidenza generalmente bassa, soprattutto le regole ottenute dopo il filtraggio di interesse.

### StateChange_Full_almostFull

In questo caso sono stati presi in considerazione gli stati "Piena" e "QuasiPiena". Le configurazioni che sono state provate ed hanno dato dei risultati in output sono le seguenti:

- StateChange_Full_almostFull_15min_1000m_sup0 (3-3)
- StateChange_Full_almostFull_15min_1000m_sup0 (4-3)
- StateChange_Full_almostFull_15min_1000m_sup0 (5-3)
- StateChange_Full_almostFull_15min_500m_sup0 (3-3)
- StateChange_Full_almostFull_15min_500m_sup001 (4-3)
- StateChange_Full_almostFull_15min_500m_sup001 (5-3)
- StateChange_Full_almostFull_30min_1000m_sup0 (3-3)
- StateChange_Full_almostFull_30min_1000m_sup002 (4-3)
- StateChange_Full_almostFull_30min_1000m_sup002 (5-3)

In questo caso, dopo il filtraggio, si sono avute alcune regole a confidenza più alta, ma con un supporto un po' più basso. Inoltre, in diverse configurazioni utilizzate, le regole a confidenza più alta contengono informazioni riguardanti soltanto la stazione di riferimento, come è possibile vedere dalle seguenti due regole di esempio prelevate dalla configurazione "StateChange_Full_almostFull_15 min_500m_sup0(5-3)":

$$[[[['Piena\_T0\_0'], ['QuasiPiena\_T2\_0'], ['QuasiPiena\_T3\_0, Piena\_T3\_0'], ['QuasiPiena\_T4\_0']]], ['0.791331 - 3743']] \tag{4.22}$$

$$[[[['Piena\_T0\_0'], ['QuasiPiena\_T1\_0'], ['QuasiPiena\_T3\_0, Piena\_T3\_0'], ['QuasiPiena\_T4\_0']]], ['0.781724 - 3653']] \tag{4.23}$$

### StateChange_Normal_Empty_almostEmpty

In questo caso, invece, oltre ai cambiamenti di stato agli stati "Vuota" e "Quasi-Vuota", è stato preso in considerazione anche il cambiamento di stato allo stato "Normale". Le configurazioni che sono state provate ed hanno dato dei risultati in output sono le seguenti:

- StateChange_Normal_Empty_almostEmpty_15min_1000m_sup002 (3-3)
- StateChange_Normal_Empty_almostEmpty_15min_1000m_sup01 (4-3)
- StateChange_Normal_Empty_almostEmpty_15min_1000m_sup01 (5-3)
- StateChange_Normal_Empty_almostEmpty_15min_500m_sup002 (3-3)
- StateChange_Normal_Empty_almostEmpty_15min_500m_sup002 (4-3)
- StateChange_Normal_Empty_almostEmpty_15min_500m_sup005 (5-3)
- StateChange_Normal_Empty_almostEmpty_30min_1000m_sup005 (3-3)
- StateChange_Normal_Empty_almostEmpty_30min_1000m_sup01 (4-3)
- StateChange_Normal_Empty_almostEmpty_30min_1000m_sup01 (5-3)

In questo caso, dopo il filtraggio, si sono avuti alcuni pattern con confidenza più alta ed anche un supporto molto più alto. Il supporto alto è dovuto al fatto che adesso abbiamo incluso il passaggio allo stato “Normale” che sarà molto presente come evento. Tuttavia, andando a selezionare le regole che nel conseguente avessero lo stato “QuasiVuota” oppure “Vuota” si è scesi molto come valore di confidenza ed, inoltre, le regole non contenevano delle informazioni molto utili.

### StateChange_Normal_Full_almostFull

Anche in questo caso, oltre agli stati "Piena" e "QuasiPiena", si è tenuto conto del passaggio di stato allo stato "Normale". Le configurazioni che sono state provate ed hanno dato dei risultati in output sono le seguenti:

- StateChange_Normal_Full_almostFull_15min_1000m_sup002 (3-3)
- StateChange_Normal_Full_almostFull_15min_1000m_sup002 (4-3)
- StateChange_Normal_Full_almostFull_15min_1000m_sup002 (5-3)
- StateChange_Normal_Full_almostFull_15min_500m_sup002 (3-3)
- StateChange_Normal_Full_almostFull_15min_500m_sup002 (4-3)

- StateChange_Normal_Full_almostFull_15min_500m_sup002 (5-3)
- StateChange_Normal_Full_almostFull_30min_1000m_sup002 (3-3)
- StateChange_Normal_Full_almostFull_30min_1000m_sup002 (4-3)
- StateChange_Normal_Full_almostFull_30min_1000m_sup005 (5-3)

Anche in questo caso, l'introduzione del passaggio di stato a "Normale" come evento, non ha portato a dei pattern di qualità migliore. Infatti, dopo aver applicato il filtraggio, i pattern a confidenza più alta, che contenessero un conseguente con uno stato diverso da "Normale", non contenevano nessuna informazione riguardo il passaggio di stato a "Normale". Questo può essere verificato dai seguenti pattern estratti dalla configurazione "StateChange_Normal_Full_almostFull_15min_1000m_sup002 (5-3)":

$$[[[['Piena\_T0\_0'], ['QuasiPiena\_T2\_0, Piena\_T2\_0'], ['Piena\_T3\_0'], ['QuasiPiena\_T4\_0']]], ['0.74336 - 3531']] \tag{4.24}$$

$$[[[['Piena\_T0\_0'], ['QuasiPiena\_T1\_0, Piena\_T1\_0'], ['Piena\_T3\_0'], ['QuasiPiena\_T4\_0']]], ['0.74300 - 3507']] \tag{4.25}$$

### StateChange_Normal_almostEmpty

Visti i risultati non molto soddisfacenti ottenuti dalle estrazioni precedenti, ho provato ad effettuare delle estrazioni considerando soltanto il passaggio da uno stato normale ad uno critico e viceversa, dove lo stato critico comprendesse sia lo stato "QuasiPiena", sia lo stato "Piena" o, equivalentemente per il caso dello stato "Vuota", sia lo stato "QuasiVuota" che lo stato "Vuota". Sono stati svolti, quindi, il presente esperimento e quello successivo in cui lo stato "QuasiVuota" include anche lo stato "Vuota" e lo stato "QuasiPiena" include anche lo stato "Piena".

Le configurazioni che sono state provate per il caso corrente ed hanno dato dei risultati in output sono le seguenti:

- StateChange_Normal_almostEmpty_15min_1000m_sup01 (3-3)
- StateChange_Normal_almostEmpty_15min_1000m_sup01 (4-3)
- StateChange_Normal_almostEmpty_15min_1000m_sup01 (5-3)
- StateChange_Normal_almostEmpty_15min_500m_sup001 (3-3)

- StateChange_Normal_almostEmpty_15min_500m_sup002 (4-3)
- StateChange_Normal_almostEmpty_15min_500m_sup002 (5-3)
- StateChange_Normal_almostEmpty_30min_1000m_sup002 (3-3)
- StateChange_Normal_almostEmpty_30min_1000m_sup002 (4-3)
- StateChange_Normal_almostEmpty_30min_1000m_sup005 (5-3)

In questo caso non è stato possibile effettuare l'estrazione impostando la soglia di supporto a 0. Quindi, molte regole sono state eliminate in fase di estrazione. Inoltre, le regole ottenute non sono state molto significative.

### StateChange_Normal_almostFull

Come detto nel precedente paragrafo, in questo caso l'estrazione è stata svolta considerando lo stato "Normale" e lo stato "QuasiPiena", comprendente anche lo stato "Piena".

Le configurazioni che sono state provate ed hanno dato dei risultati in output sono le seguenti:

- StateChange_Normal_almostFull_15min_1000m_sup0 (3-3)
- StateChange_Normal_almostFull_15min_1000m_sup002 (4-3)
- StateChange_Normal_almostFull_15min_1000m_sup002 (5-3)
- StateChange_Normal_almostFull_15min_500m_sup0 (3-3)
- StateChange_Normal_almostFull_15min_500m_sup0 (4-3)
- StateChange_Normal_almostFull_15min_500m_sup0 (5-3)
- StateChange_Normal_almostFull_30min_1000m_sup0 (3-3)
- StateChange_Normal_almostFull_30min_1000m_sup002 (4-3)
- StateChange_Normal_almostFull_30min_1000m_sup005 (5-3)

Tra le regole estratte, in questo caso, quelle più significative sono state quelle estratte dalla configurazione ottenuta usando un intervallo temporale di 30 minuti e un intervallo spaziale di 1000 metri. Esse, infatti, tengono in considerazione entrambi gli stati e in diverse regole sono contenute informazioni anche sul vicinato.

## 4.4 Analisi della correlazione dei dati

Dalle precedenti fasi di estrazione e filtraggio, si sono ottenuti dei pattern abbastanza deludenti in termini di confidenza ed informazioni contenute. Per questo motivo, si è pensato di effettuare un'ulteriore fase di analisi nella quale è stato eseguito uno studio sulla correlazione dei dati appartenenti al dataset ripulito.

Le città presenti nel dataset sono state: Mountain View, Palo Alto, Redwood City, San Francisco e San Jose.

Per ogni città, è stata quindi ottenuta una tabella che ha come righe i timestamp e come colonne il numero medio di bici disponibili al tempo corrente e a 4 istanti di tempo precedenti, per ogni stazione appartenente alla determinata città presa in considerazione.

Sono stati effettuati diversi esperimenti, variando il parametro che indica la dimensione dell'intervallo temporale da considerare. In particolare, sono state ottenute le matrici di correlazione per le diverse città, utilizzando le seguenti configurazioni:

- Intervallo di 15 minuti, 5 istanti temporali
- Intervallo di 30 minuti, 5 istanti temporali
- Intervallo di 60 minuti, 5 istanti temporali

Ad una prima occhiata sembrano esserci forti correlazioni tra lo stato all'istante corrente (T0) e quello agli istanti precedenti di una stessa stazione, mentre tra i dati di stazioni diverse ci sono correlazioni molto scarse. Infatti, come si può vedere anche dalla Figura 4.26, che riporta come esempio la matrice di correlazione delle stazioni appartenenti alla città di Mointain View, considerando stazioni diverse abbiamo delle correlazioni che in valore assoluto non superano lo 0.45.

Inoltre, le stazioni che presentano delle correlazioni più elevate in valore assoluto, sono sempre vicine tra di loro. Questo esclude l'ipotesi dell'esistenza di un legame tra stazioni mediamente lontane che si svuotano con bici che vanno verso altre stazioni.

Per verificare ulteriormente questa ipotesi, si è deciso di dividere l'arco della giornata in diverse fasce orarie ed effettuare l'analisi della correlazione singolarmente per le diverse fasce orarie. Le fasce orarie considerate sono state le stesse di quelle usate nell'estrazione dei pattern, ovvero:

- **0 - 6**
- **6 - 10**
- **10 - 14**

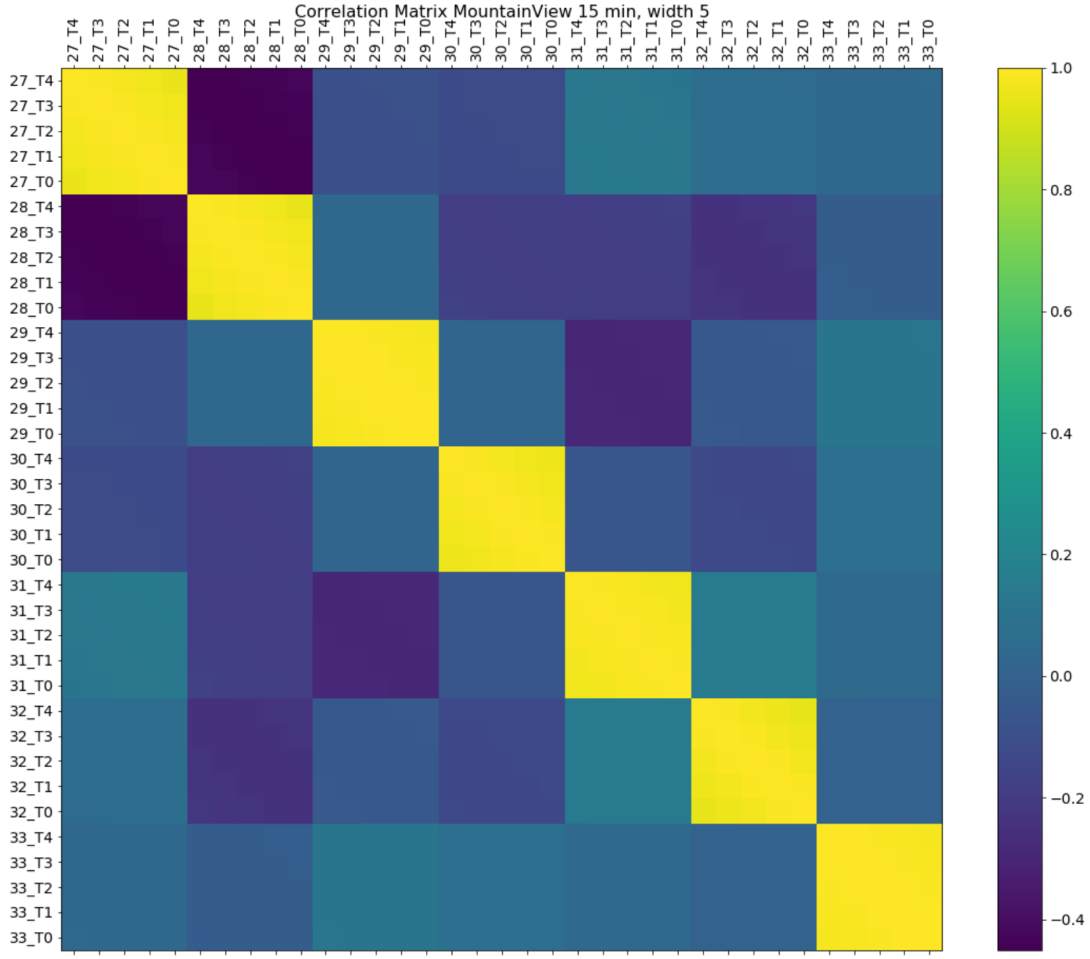



**Figura 4.26:** Matrice di correlazione relativa alla città di Mountain View, con un intervallo di 15 minuti e 5 istanti di tempo.

- **14 - 17**
- **17 - 20**
- **20 - 24**

Prendendo in esame la città di San Francisco, visto che è la città che contiene più stazioni, effettivamente mi sono accorto che la situazione cambia da una fascia all'altra, soprattutto considerando un intervallo di 30 minuti; mentre con gli altri intervalli la situazione sembra essere più stabile anche al variare delle fasce orarie.

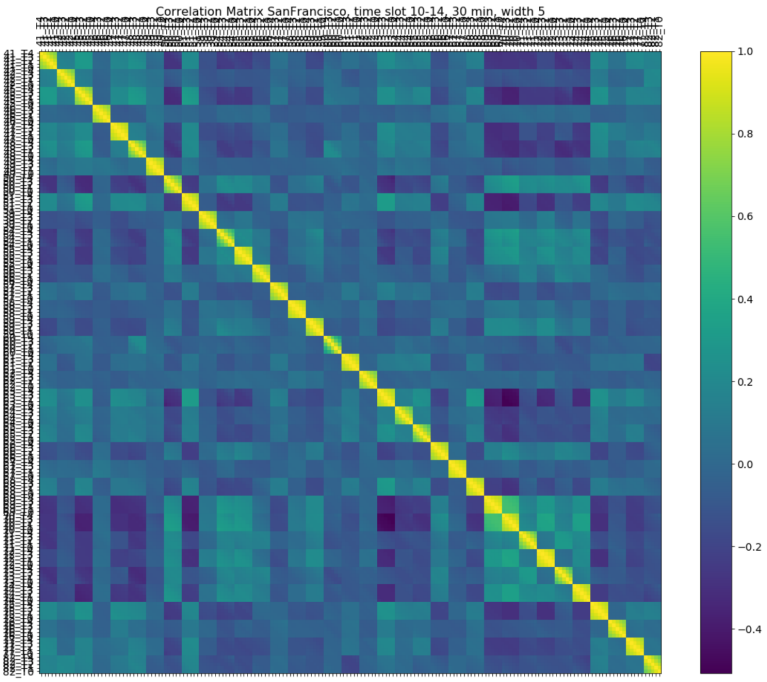
Nelle fasce **6-10** e **10-14** ho notato delle correlazioni più alte tra stazioni diverse e più basse tra i diversi istanti temporali della stessa stazione, mentre nelle fasce

successive, le correlazioni tra stazioni diverse tendono a diminuire gradualmente, mentre le correlazioni tra i diversi istanti temporali della stessa stazione aumentano. Nella Figura 4.27 sono riportate come esempio due matrici di correlazione corrispondenti alle fasce orarie 10-14 e 20-24. Dalla figura, infatti, possiamo notare come, nel caso della fascia 10-14 (Fig. 4.27a), si abbiano dei colori generalmente più chiari, mentre nella diagonale possiamo notare la presenza di chiazze verdi, che indicano una correlazione più bassa. Nella fascia 20-24 (Fig. 4.27b), invece, abbiamo generalmente dei colori più scuri, molto vicini allo 0, mentre la diagonale è molto più gialla, indicando una correlazione più alta tra i diversi istante temporali della stesse stazioni.

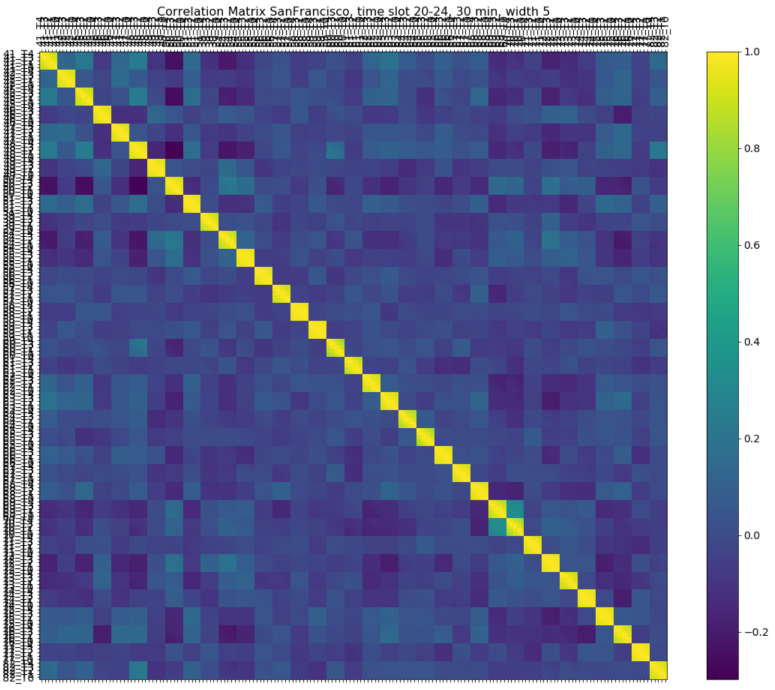
Un pattern presente in quasi tutte le fasce orarie è rappresentato dalla presenza di una zona più verde in basso a destra, come è possibile notare chiaramente nella figura 4.27a. Prendendo in esame proprio la fascia 10-14, possiamo notare che i valori di correlazione più alti si aggirano intorno allo 0.50, tra la stazione 69 e 70. Esse sono risultate essere molto vicine tra di loro.

In ultima analisi, posso dire che i valori di correlazione sono risultati generalmente molto bassi, soprattutto tra stazioni lontane. Per di più, per le altre città presenti nel dataset, le correlazioni sono state mediamente ancora più basse rispetto alla città di San Francisco, pur considerando la suddivisione in fasce orarie. Per questo motivo, unitamente al fatto che la città di San Francisco è risultata essere quella contenente il maggior numero di stazioni, nelle successive fasi del lavoro ho scartato tutte le altre città concentrandomi soltanto su quest'ultima.

Inoltre, il fatto che i valori di correlazione più alti si sono avuti fra gli istanti diversi della stessa stazione, giustifica il fatto che molti pattern estratti con confidenze alte contenessero le informazioni riguardanti la sola stazione di riferimento e non quelle facenti parti del vicinato.




**(a)** Fascia oraria: 10-14




**(b)** Fascia oraria: 20-24

**Figura 4.27:** Matrici di correlazione relative alla città di San Francisco, con un intervallo di 30 minuti, 5 istanti di tempo e fasce orarie 10-14 (a) e 20-24 (b).

## 4.5 Applicazione dei classificatori tradizionali

Dopo aver ottenuto i dataset, come descritto nel paragrafo 3.8 del capitolo precedente, ho proseguito con l'effettuare il training di 5 diversi classificatori, utilizzando i parametri di default e, dopo di che, ho proceduto con la fase di test, in modo da confrontare le performance dei diversi classificatori. Come già detto in precedenza, i classificatori utilizzati sono stati:

- Albero di decisione
- Support Vector Machine
- Logistic Regression
- Naive Bayes
- Random Forest

I risultati generali sono stati ottenuti effettuando una micro-media, consistente nel sommare tutti i rispettivi valori delle singole matrici di confusione ottenute per ogni stazione, ottenendo così un'unica matrice di confusione globale ed infine calcolare le metriche di valutazione su quest'ultima.

I risultati medi ottenuti da questa prima fase di test, considerando un intervallo di 30 minuti e 5 istanti temporali, sono mostrati nella Tabella 4.3.

| | avg_accuracy | avg_recall | avg_precision |
|---|---|---|---|
| **DecisionTree** | 0.968716 | 0.535608 | 0.559921 |
| **SVC** | 0.969961 | 0.202770 | 0.793037 |
| **LogisticRegression** | 0.974696 | 0.564590 | 0.667915 |
| **NaiveBayes** | 0.867180 | 0.750643 | 0.176197 |
| **RandomForest** | 0.976811 | 0.406726 | 0.865866 |

**Tabella 4.3:** Risultati ottenuti dal test dei modelli allenati con i parametri di default e considerando un intervallo temporale di 30 minuti e 5 istanti temporali.

Come è possibile notare dai risultati, l'accuratezza ottenuta è stata molto alta rispetto al richiamo e la precisione. Questo è dovuto al fatto che nei casi di dataset sbilanciati come il nostro, l'accuratezza è una metrica ingannevole. Per questo motivo, ho preferito non considerarla più e concentrarmi sulle altre metriche, soprattutto sulla precisione. Tuttavia, per completezza, ho deciso di considerare anche l'F1-score come ulteriore metrica.

### 4.5.1 Tuning degli iperparametri

Come passo successivo, ho proceduto ad una fase di tuning degli iperparametri dei vari modelli, effettuando una Grid Search. In questo modo ho ottenuto i modelli allenati tramite la miglior configurazione degli iperparametri, in modo da ottimizzare diverse metriche.

Ho deciso di non considerare il Suppor Vector Machine (SVC) in quanto si è mostrato molto più lento degli altri durante l'esecuzione della Grid Search.

Considerando i modelli ottenuti dalla Grid Serach ottimizzando il richiamo, i risultati ottenuti dalla fase di test sono contenuti nella Tabella 4.4.

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **DecisionTree** | 0.929179 | 0.423879 | 0.582176 |
| **LogisticRegression** | 0.954204 | 0.322330 | 0.481880 |
| **NaiveBayes** | 0.776261 | 0.138967 | 0.235732 |
| **RandomForest** | 0.882196 | 0.311211 | 0.460110 |

**Tabella 4.4:** Risultati ottenuti dal test dei modelli che hanno ottimizzato il **richiamo** in fase di fine tuning e considerando un intervallo temporale di 30 minuti e 5 istanti temporali

La Tabella 4.5 mostra, invece, i risultati ottenuti considerando i modelli che ottimizzano la precisione.

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **DecisionTree** | 0.724332 | 0.773774 | 0.748237 |
| **LogisticRegression** | 0.521860 | 0.735946 | 0.610683 |
| **NaiveBayes** | 0.735608 | 0.174352 | 0.281891 |
| **RandomForest** | 0.269931 | 0.651468 | 0.381705 |

**Tabella 4.5:** Risultati ottenuti dal test dei modelli che hanno ottimizzato la **precisione** in fase di fine tuning e considerando un intervallo temporale di 30 minuti e 5 istanti temporali

Considerando infine i modelli che ottimizzano la l'f1-score, si sono ottenuti i risultati mostrati in Tabella 4.6.

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **DecisionTree** | 0.768942 | 0.709954 | 0.738272 |
| **LogisticRegression** | 0.642038 | 0.552849 | 0.594115 |
| **NaiveBayes** | 0.750841 | 0.174948 | 0.283776 |
| **RandomForest** | 0.776855 | 0.436989 | 0.559342 |

**Tabella 4.6:** Risultati ottenuti dal test dei modelli che hanno ottimizzato l' **F1-score** in fase di fine tuning e considerando un intervallo temporale di 30 minuti e 5 istanti temporali

Come è possibile vedere dalle tabelle contenenti i risultati dei vari test, l'algoritmo di classificazione che ha avuto delle performance generalmente migliori, soprattutto rispetto la precisione, è stato l'albero di decisione. Per questo motivo è stato l'algoritmo utilizzato come termine di paragone per il classificatore associativo da me implementato.

## 4.6 Confronto Albero di Decisione - Classificatore Associativo

In questa sezione si presentano i migliori risultati ottenuti dai diversi esperimenti eseguiti utilizzando gli alberi di decisioni e diverse versioni del classificatore associativo. Tali esperimenti verranno presentati in maniera dettagliata nei successivi paragrafi 4.7 e 4.8.

Nella Figura 4.28a e nella Figura 4.28b sono contenute le configurazioni che hanno portato ai migliori risultati di precisione rispettivamente per il classificatore associativo e l'albero di decisione.

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| QP_conf40%_0-6_18match | 0.0122 | 0.9362 | 0.0240 |
| QP_conf40%_6-10_15match | 0.0083 | 0.6087 | 0.0165 |
| QP_conf40%_10-14_25match | 0.0126 | 0.9091 | 0.0249 |
| QP_conf40%_14-17_25match | 0.0049 | 1.0000 | 0.0098 |
| QP_conf40%_17-20_25match | 0.0066 | 1.0000 | 0.0132 |
| QP_conf40%_20-24_15match | 0.0168 | 0.8857 | 0.0331 |

**(a)** Classificatore associativo

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| 0-6(90%) | 0.7729 | 0.9376 | 0.8473 |
| 6-10(60%) | 0.3600 | 0.5703 | 0.4414 |
| 10-14(60%) | 0.5398 | 0.6572 | 0.5927 |
| 14-17(60%) | 0.3165 | 0.5498 | 0.4018 |
| 17-20(70%) | 0.2576 | 0.5846 | 0.3576 |
| 20-24(85%) | 0.7152 | 0.8698 | 0.7850 |

**(b)** Albero di decisione

**Figura 4.28:** Migliori risultati di precisione ottenuti dal test del classificatore associativo (a) e dell'albero di decisione(b) con le rispettive configurazioni utilizzate nelle varie fasce orarie.

Nella Figura 4.28a sono stati presentati i risultati ottenuti nella classificazione associativa, utilizzando le diverse configurazioni ottenute considerando: come stato delle stazioni solamente lo stato "QuasiPiena" (QP); una soglia di confidenza del 40%, in modo da mantenere solo i pattern più rilevanti; una determinata fascia oraria della giornata; il numero minimo di pattern da verificare in modo da effettuare la classificazione.

Nella Figura 4.28b sono stati invece presentati i risultati ottenuti dalla classificazione dell'albero di decisione, utilizzando le diverse configurazioni ottenute considerando: una determinata fascia oraria della giornata; la soglia di probabilità che un record appartenga alla classe positiva, in modo da effettuare la classificazione.

Come possiamo vedere, in quasi tutte le fasce orarie, il classificatore associativo ha raggiunto delle precisioni più alte se confrontate con quelle raggiunte, nelle rispettive fasce orarie, dall'albero di decisione. In alcuni casi è stata raggiunta addirittura una precisione del **100%**. Tuttavia, il richiamo ottenuto dal classificatore associativo è nettamente più basso rispetto quello dell'albero di decisione.

Di seguito sono invece considerate le prestazioni generali dei due modelli, ottenute mediando i risultati appartenenti alle diverse fasce orarie e fissando alcuni parametri di default come: il numero di pattern da verificare, nel caso del classificatore associativo; la soglia di probabilità che un record fosse classificato come appartenente alla classe positiva ("QuasiPiena"), nel caso dell'albero di decisione.

Per valutare le prestazioni generali dei due modelli, sono state utilizzate due configurazioni di default per il classificatore associativo e una per l'albero di decisione. In particolare:

- **Classificatore associativo (1)**: considerando soltanto lo stato "QuasiPiena", una soglia di confidenza dei pattern del 40% e diversi numeri di pattern da

verificare di default;

- **Classificatore associativo (2)**: considerando sia lo stato "Normale" che lo stato "QuasiPiena", una soglia di confidenza dei pattern del 50% e diversi numeri di pattern da verificare di default;
- **Albero di decisione**: il modello che ha massimizzato la precisione nella fase di Grid Search e diversi valori di default per la soglia di precisione.

In Figura 4.29 sono presentati i risultati ottenuti dai tre diversi modelli considerati con le rispettive configurazioni di default.

| | oa_recall | oa_precision | oa_f1_score |
|---|---|---|---|
| **QP_conf40_1match** | 0.76559 | 0.76435 | 0.76497 |
| **QP_conf40_15match** | 0.03366 | 0.77871 | 0.06453 |
| **QP_conf40_18match** | 0.01615 | 0.79556 | 0.03166 |
| **QP_conf40_20match** | 0.00830 | 0.77966 | 0.01643 |
| **QP_conf40_25match** | 0.00307 | 0.94444 | 0.00612 |
| **QP_conf40_30match** | 0.00126 | 0.87500 | 0.00252 |

**(a)** Classificatore associativo (1)

| | oa_recall | oa_precision | oa_f1_score |
|---|---|---|---|
| **0.50** | 0.69097 | 0.74119 | 0.71520 |
| **0.60** | 0.62636 | 0.77530 | 0.69292 |
| **0.70** | 0.53920 | 0.80290 | 0.64514 |
| **0.80** | 0.45421 | 0.82769 | 0.58654 |
| **0.85** | 0.41803 | 0.84467 | 0.55927 |
| **0.90** | 0.37499 | 0.84061 | 0.51863 |

**(b)** Albero di decisione

| | oa_recall | oa_precision | oa_f1_score |
|---|---|---|---|
| **N/QP_conf50_1match** | 0.70513 | 0.76965 | 0.73598 |
| **N/QP_conf50_15match** | 0.30948 | 0.81011 | 0.44787 |
| **N/QP_conf50_18match** | 0.25526 | 0.81130 | 0.38834 |
| **N/QP_conf50_20match** | 0.18975 | 0.79001 | 0.30600 |
| **N/QP_conf50_25match** | 0.13011 | 0.75656 | 0.22204 |
| **N/QP_conf50_30match** | 0.10421 | 0.76949 | 0.18356 |

**(c)** Classificatore associativo (2)

**Figura 4.29:** Prestazioni generali dei tre modelli considerati. La sigla "oa" sta per "overall" e sta ad indicare i risultati generali, ricavati mediando i risultati ottenuti nelle diverse fasce orarie, fissando il numero minimo di match (nel caso del classificatore associativo) o la soglia di probabilità (nel caso dell'albero di decisione).

Come possiamo notare, i risultati di precisione ottenuti dal classificatore associativo, considerando solamente lo stato "QuasiPiena" ed una soglia di confidenza al 40% (Figura 4.29a), superano la maggior parte dei risultati di precisione ottenuti dall'albero (Figura 4.29b), raggiungendo una precisione massima del **94.44%**.

Inoltre, i risultati ottenuti dal classificatore associativo, considerando entrambi gli stati ed una confidenza del 50% (Figura 4.29c), riescono a migliorare ulteriormente non solo alcuni risultati di precisione ottenuti dall'altro classificatore associativo, ma ottengono anche un richiamo notevolmente più alto, ma pur sempre più basso da quello raggiunto dall'albero di decisione.

In conclusione possiamo dire che, se si vogliono ottenere dei risultati con precisione generalmente abbastanza alte e non si è interessati ad avere un richiamo altissimo, i classificatori ottenuti rappresentano una valida alternativa agli algoritmi di classificazione tradizionali.

## 4.7 Esperimenti con l'Albero di Decisione

Gli esperimenti riguardanti l'albero di decisione e il classificatore associativo sono stati svolti in parallelo, in modo da valutare passo passo l'impatto dei vari esperimenti sulle prestazioni dei due classificatori.

In questa sezione, si partirà con il mostrare i risultati ottenuti dai vari esperimenti effettuati utilizzando l'albero di decisione come algoritmo di classificazione.

### 4.7.1 Introduzione soglia di probabilità

L'obiettivo è quello di fare, per gli alberi di decisione, qualcosa di simile a quello che è stato fatto, come vedremo nella sezione dedicata agli esperimenti effettuati sul classificatore associativo, con il test dei pattern. In quella fase sono andato ad aumentare il numero minimo di regole che dovevano essere verificate affinché il record di test fosse stato classificato come "QuasiPiena". In questo modo è stato possibile far salire la precisione a discapito del richiamo.

Per ottenere dei risultati analoghi, ho quindi introdotto nella fase di test dell'albero una soglia di probabilità: un record di test è stato classificato come "QuasiPiena" soltanto qualora l'albero di decisione indicasse una probabilità di appartenenza alla classe positiva ("QuasiPiena", appunto) uguale o superiore alla soglia di probabilità stabilita.

Sono quindi partito con una soglia del 50% ed ho proseguito testando valori di soglia via via più alti. In questo modo ho potuto verificare se, anche utilizzando un albero di decisione, si potesse riuscire ad ottenere un valore di precisione che si avvicinasse al valore abbastanza alto ottenuto dalla fase di test dei pattern, utilizzando un certo numero minimo di pattern da verificare.

Considerando i modelli ottenuti dalla Grid Search, in modo da ottimizzare la precisione, si sono ottenuti dei risultati mostrati nella Tabella 4.7.

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **0.50** | 0.72433 | 0.77377 | 0.74824 |
| **0.60** | 0.65124 | 0.80410 | 0.71964 |
| **0.70** | 0.60544 | 0.82294 | 0.69763 |
| **0.80** | 0.48981 | 0.84650 | 0.62055 |
| **0.85** | 0.38061 | 0.85625 | 0.52698 |
| **0.90** | 0.05490 | 0.92040 | 0.10361 |

**Tabella 4.7:** Risultati ottenuti dal test degli alberi di decisione allenati con i parametri ottenuti dal fine tuning ottimizzando la precisione e considerando un intervallo temporale di 30 minuti e 5 istanti temporali

Come è possibile vedere, aumentando il valore della soglia di probabilità, aumenta anche la precisione a discapito del richiamo. Il valore più alto ottenuto per la precisione è stato del 92% che è comunque più basso del 93,4% ottenuto tramite i pattern. Osservando i risultati si può notare che la precisione media è più o meno in linea con la soglia di probabilità che è stata usata nei vari casi. In particolare, è quasi sempre di poco più alta della soglia di probabilità.

### 4.7.2 Cammini di decisione con soglia di probabilità del 90%

Partendo dalla precedente fase di test, in cui ho settato una soglia minima di probabilità, un'analisi che è sembrata utile fare, è stata quella di capire quali siano stati i cammini dell'albero di decisione che hanno portato ad avere una precisione del 92%, settando la soglia di probabilità al 90%.

Ho quindi ricavato i cammini che hanno portato ad un nodo in cui la probabilità di appartenere alla classe "QuasiPiena" fosse del 90%. In altri termini, ho ricavato e stampato la sequenza di regole che hanno determinato la predizione di quei record classificati come "QuasiPiena". In questo modo, è stato possibile capire quali siano stati gli attributi utilizzati dagli alberi nell'effettuare la predizione.

Poiché gli alberi ottenuti dalla Grid Search hanno avuto una profondità massima di 2-4, i percorsi ottenuti comprendono da 2 a 4 attributi. Inoltre, tutti i record di ciascuna stazione che vengono classificati come "QuasiPiena" seguono lo stesso percorso, quindi gli attributi utilizzati saranno gli stessi per tutti i record.

Un riassunto dei cammini di decisione estratti è il seguente:

——————— STAZIONE 45 ———————

Regole usate per predire i record:
decision node : (docks_av_45_T0 = 1.1) <= 2.21666
decision node : (bikes_av_48_T3 = 0.36666) <= 4.45000
decision node : (bikes_av_65_T2 = 6.6) > 2.74999

——————— STAZIONE 59 ———————

Regole usate per predire i record:
decision node : (docks_av_59_T0 = 2.0) <= 2.65000
decision node : (bikes_av_54_T0 = 14.0) > 13.10000

——————— STAZIONE 63 ———————

Regole usate per predire i record:
decision node : (docks_av_63_T0 = 3.0) > 2.81666
decision node : (docks_av_63_T0 = 3.0) <= 3.71666
decision node : (bikes_av_71_T2 = 17.0) > 16.40000

——————— STAZIONE 65 ———————

Regole usate per predire il record 0:
decision node : (docks_av_65_T0 = 0.53333) <= 2.95000
decision node : (bikes_av_65_T0 = 14.46666 > 12.9499
decision node : (bikes_av_70_T1 = 11.83333) > 8.61666
decision node : (docks_av_65_T0 = 0.53333) <= 1.14999

——————— STAZIONE 72 ———————

Regole usate per predire i record:
decision node : (docks_av_72_T0 = 1.0) <= 2.91666
decision node : (docks_av_72_T0 = 1.0) <= 2.01666
decision node : (bikes_av_72_T0 = 22.0) > 21.98333

Come è possibile vedere, gli alberi utilizzano nelle decisioni anche le informazioni relative al numero di bici disponibili, mentre nel classificatore associativo viene considerato soltanto il numero di slot disponibili.

### 4.7.3 Rimozione informazioni sul numero di bici disponibili

Avendo notato che nei cammini di decisione, ottenuti nella fase precedente, venisse utilizzata l'informazione sul numero di bici disponibili, che invece non è presente nel classificatore associativo, ho deciso di provare a rimuovere queste informazioni in modo da avere un confronto alla pari tra i due classificatori. Tuttavia, il numero di bici disponibili era spesso vicino al numero di slot totali, quindi non è chiaro

perché il classificatore abbia scelto di utilizzare quell'informazione piuttosto che il numero di slot vuoti per determinare se una stazione fosse nello stato "QuasiPiena". Rimuovendo le informazioni sulle biciclette disponibili, quindi, mi non mi aspettavo grandi variazioni delle performance dell'albero.

Per effettuare questa prova, ho quindi aggiunto un flag che mi permettesse di rimuovere le colonne contenenti le informazioni sul numero di biciclette disponibili sia dal training set che dal test set e quindi di effettuare nuovamente la Grid Search, usando il nuovo training set, e, dopo di che, di testare i modelli ottenuti sul nuovo test set.

I risultati del test sono stati riportati in Figura 4.30a, mentre in Figura 4.30b sono riportati i risultati che erano stati ottenuti in precedenza, utilizzando le informazioni sul numero di bici disponibili.

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **0.50** | 0.73314 | 0.77152 | 0.75184 |
| **0.60** | 0.65806 | 0.79916 | 0.72178 |
| **0.70** | 0.60791 | 0.81685 | 0.69706 |
| **0.80** | 0.47587 | 0.84182 | 0.60803 |
| **0.85** | 0.36756 | 0.84898 | 0.51301 |
| **0.90** | 0.04768 | 0.90943 | 0.09060 |

**(a)** Senza informazioni sul numero di bici

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **0.50** | 0.72433 | 0.77377 | 0.74824 |
| **0.60** | 0.65124 | 0.80410 | 0.71964 |
| **0.70** | 0.60544 | 0.82294 | 0.69763 |
| **0.80** | 0.48981 | 0.84650 | 0.62055 |
| **0.85** | 0.38061 | 0.85625 | 0.52698 |
| **0.90** | 0.05490 | 0.92040 | 0.10361 |

**(b)** Con informazioni sul numero di bici

**Figura 4.30:** Risultati dei test dell'albero per le diverse soglie di probabilità utilizzando il dataset senza le informazioni sul numero di bici disponibili (a) o quello contenente tale informazione (b).

Come mi aspettavo, i risultati ottenuti mostrano un lieve peggioramento della precisione media per tutte le soglie di probabilità. Infatti, come è possibile vedere c'è una differenza di qualche punto percentuale tra le rispettive soglie di probabilità. Per questo motivo ho deciso di continuare a tenere in considerazione i risultati già ottenuti comprendendo anche il numero di bici disponibili.

### 4.7.4 Suddivisione in fasce orarie

L'ultimo esperimento che è stato condotto utilizzando gli alberi di decisione, è stato quello di suddividere il dataset in fasce orarie e quindi allenare degli alberi di decisione separatamente per ogni singola fascia oraria. In questo modo è stato possibile valutare sia come variavano le performance dell'albero da una fascia oraria

all'altra, sia le performance generali di questo approccio, mediando sui risultati ottenuti sulle diverse fasce.

Una volta eseguita nuovamente la Grid Search, per ogni fascia oraria, ho proseguito con il test dei modelli ottenuti massimizzando la precisione. I risultati di tali test sono mostrati di seguito.

### FASCIA ORARIA 0 - 6

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **0.50** | 0.935546 | 0.907432 | 0.921275 |
| **0.60** | 0.924758 | 0.909907 | 0.917273 |
| **0.70** | 0.911757 | 0.912009 | 0.911883 |
| **0.80** | 0.825726 | 0.927883 | 0.873829 |
| **0.85** | 0.798340 | 0.934888 | 0.861235 |
| **0.90** | 0.772891 | 0.937584 | 0.847309 |

**Tabella 4.8:** Risultati ottenuti dal test degli alberi di decisione ottenuti dal fine tuning ottimizzando la precisione, considerando un intervallo temporale di 30 minuti, 5 istanti temporali e la fascia oraria **0-6**.

In questo caso, come è possibile vedere dalla Tabella 4.8, si sono raggiunti dei risultati con precisioni abbastanza alte.

### FASCIA ORARIA 6 - 10

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **0.50** | 0.432062 | 0.563326 | 0.489039 |
| **0.60** | 0.359952 | 0.570349 | 0.441359 |
| **0.70** | 0.193087 | 0.553846 | 0.286346 |
| **0.80** | 0.113230 | 0.542857 | 0.187377 |
| **0.85** | 0.075685 | 0.510040 | 0.131811 |
| **0.90** | 0.075685 | 0.510040 | 0.131811 |

**Tabella 4.9:** Risultati del test degli alberi di decisione che hanno ottimizzato la precisione, considerando gli stessi parametri precedenti e la fascia oraria **6-10**.

In questo caso, come è possibile notare dalla Tabella 4.9, si sono avuti dei peggioramenti delle performance rispetto alla fascia precedente. Probabilmente questo è dovuto al fatto che in questa fascia c'è molto movimento, quindi il classificatore non riesce bene ad effettuare la predizione.

### FASCIA ORARIA 10 - 14

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **0.50** | 0.595328 | 0.640625 | 0.617147 |
| **0.60** | 0.539773 | 0.657187 | 0.592721 |
| **0.70** | 0.190025 | 0.639066 | 0.292944 |
| **0.80** | 0.077652 | 0.556561 | 0.136288 |
| **0.85** | 0.075758 | 0.563380 | 0.133556 |
| **0.90** | 0.046717 | 0.483660 | 0.085204 |

**Tabella 4.10:** 10-14 **10-14**.

I risultati mostrati in Tabella 4.10 sono stati leggermente migliori rispetto a quelli della fascia precedente, ma comunque non ottimi.

### FASCIA ORARIA 14 - 17

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **0.50** | 0.373887 | 0.529412 | 0.438261 |
| **0.60** | 0.316518 | 0.549828 | 0.401758 |
| **0.70** | 0.250247 | 0.537155 | 0.341430 |
| **0.80** | 0.161227 | 0.475219 | 0.240768 |
| **0.85** | 0.090999 | 0.377049 | 0.146614 |
| **0.90** | 0.090010 | 0.374486 | 0.145136 |

**Tabella 4.11:** Risultati del test degli alberi di decisione che hanno ottimizzato la precisione, considerando gli stessi parametri precedenti e la fascia oraria **14-17**.

Per questa fascia sono stati ottenuti dei risultati abbastanza deludenti. Inoltre, l'aumento della soglia non ha generalmente portato ad un aumento della precisione, mentre ha determinato un calo del richiamo.

### FASCIA ORARIA 17 - 20

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **0.50** | 0.539483 | 0.540680 | 0.540081 |
| **0.60** | 0.268635 | 0.577778 | 0.366751 |
| **0.70** | 0.257565 | 0.584590 | 0.357582 |
| **0.80** | 0.128413 | 0.545455 | 0.207885 |
| **0.85** | 0.067897 | 0.513966 | 0.119948 |
| **0.90** | 0.061255 | 0.506098 | 0.109282 |

**Tabella 4.12:** Risultati ottenuti dal test degli alberi di decisione allenati con i parametri ottenuti dal fine tuning ottimizzando la precisione, considerando un intervallo temporale di 30 minuti e 5 istanti temporali e la fascia oraria **17-20**.

Come è possibile vedere dalla Tabella 4.12, per questa fascia sono stati ottenuti dei risultati simili a quelli della fascia oraria precedente.

### FASCIA ORARIA 20 - 24

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **0.50** | 0.814674 | 0.842135 | 0.828177 |
| **0.60** | 0.791304 | 0.852459 | 0.820744 |
| **0.70** | 0.789674 | 0.852199 | 0.819746 |
| **0.80** | 0.760326 | 0.857230 | 0.805876 |
| **0.85** | 0.715217 | 0.869795 | 0.784969 |
| **0.90** | 0.536413 | 0.854545 | 0.659098 |

**Tabella 4.13:** Risultati ottenuti dal test degli alberi di decisione allenati con i parametri ottenuti dal fine tuning ottimizzando la precisione, considerando un intervallo temporale di 30 minuti e 5 istanti temporali e la fascia oraria **20-24**.

In quest'ultima fascia oraria invece sono stati ottenuti dei risultati abbastanza buoni.

### Risultati generali

Calcolando le matrici di confusione generali per i diversi valori di soglia, ovvero sommando le matrici per le diverse fasce orarie fissando i diversi valori di soglia, ho ottenuto le performance medie per ciascuna soglia di probabilità. I risultati ottenuti sono mostrati nella Tabella 4.14.

| | overall_recall | overall_precision | overall_f1_score |
|---|---|---|---|
| **0.50** | 0.690968 | 0.741192 | 0.715200 |
| **0.60** | 0.626365 | 0.775296 | 0.692918 |
| **0.70** | 0.539204 | 0.802902 | 0.645147 |
| **0.80** | 0.454209 | 0.827688 | 0.586542 |
| **0.85** | 0.418028 | 0.844667 | 0.559271 |
| **0.90** | 0.374989 | 0.840615 | 0.518625 |

**Tabella 4.14:** Risultati generali ottenuti dalle matrici di confusione globali consistenti nella la somma di tutte le matrici di confusione ottenute nelle diverse fasce orarie, per ciascun valore della soglia di probabilità.

Mentre le combinazioni fascia oraria – soglia di probabilità che hanno ottenuto i migliori risultati di precisione sono mostrate nella Tabella 4.15.

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **0-6(90%)** | 0.7729 | 0.9376 | 0.8473 |
| **6-10(60%)** | 0.3600 | 0.5703 | 0.4414 |
| **10-14(60%)** | 0.5398 | 0.6572 | 0.5927 |
| **14-17(60%)** | 0.3165 | 0.5498 | 0.4018 |
| **17-20(70%)** | 0.2576 | 0.5846 | 0.3576 |
| **20-24(85%)** | 0.7152 | 0.8698 | 0.7850 |

**Tabella 4.15:** Risultati migliori ottenuti dalle diverse soglie di probabilità per ciascuna fascia oraria.

## 4.8 Esperimenti con il Classificatore Associativo

Una volta ottenuto il dataset, aver estratto i pattern ed averli filtrati, come spiegato nel capitolo precedente, ho preceduto con la fase di test del classificatore associativo, considerando i pattern ottenuti.

In questo paragrafo saranno analizzati i risultati ottenuti dai vari esperimenti eseguiti utilizzando diverse configurazioni.

### 4.8.1 Esperimento con intervallo temporale di 30 minuti

Una volta estratti i pattern, settando un intervallo temporale uguale a 30 minuti ed averli filtrati in modo da scartare tutti i pattern che nel conseguente non contenevano soltanto la stazione di riferimento e lo stato “QuasiPiena”, ho proceduto con la fase di test. Tuttavia, avendo ottenuto un numero ancora molto elevato di pattern (1614), i risultati hanno mostrato una precisione molto bassa ed un richiamo molto alto. Questo è stato dovuto al fatto che, essendoci molti pattern, ne venisse sicuramente trovato uno che verificasse il record di test. In questo modo tutti i record del test set venivano predetti con “QP”, ottenendo un richiamo del 100% ed una precisione bassissima.

Per ovviare a questo problema ho provato a seguire due strade diverse che mi hanno permesso di ridurre il numero di pattern:

1. Filtrare le regole utilizzando una soglia di confidenza;
2. Mantenere solo le poche regole che contengono solo "QuasiPiena" anche nell’antecedente (ossia non considerare lo stato "Normale").

Ho quindi rieseguito il filtraggio, utilizzando le seguenti configurazioni:

- Considerando gli stati "Normale" e "QuasiPiena" ed una soglia di confidenza del 40%, ottenendo 404 pattern;
- Considerando gli stati "Normale" e "QuasiPiena" ed una soglia di confidenza del 50%, ottenendo 136 pattern;
- Considerando gli stati "Normale" e "QuasiPiena" ed una soglia di confidenza del 55%, ottenendo 11 pattern;
- Considerando soltanto lo stato "QuasiPiena" ed una soglia di confidenza dello 0%, ottenendo 132 pattern;
- Considerando soltanto lo stato "QuasiPiena" ed una soglia di confidenza del 40%, ottenendo 48 pattern;

- Considerando soltanto lo stato "QuasiPiena" ed una soglia di confidenza del 50%, ottenendo 19 pattern.

La prima configurazione ha dato in output ancora troppi pattern, presentando quindi gli stessi problemi legati alla bassa precisione. Per questo motivo ho proceduto con il test delle altre configurazioni riportate in tabella 4.16 insieme ai rispettivi risultati ottenuti.

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **N/QP_conf55%_1match** | 0.6870 | 0.7680 | 0.7253 |
| **N/QP_conf50%_1match** | 0.7715 | 0.7708 | 0.7712 |
| **N/QP_conf50%_2match** | 0.7715 | 0.7708 | 0.7712 |
| **N/QP_conf50%_5match** | 0.7715 | 0.7708 | 0.7712 |
| **N/QP_conf50%_20match** | 0.6180 | 0.7723 | 0.6866 |
| **QP_conf0%_1match** | 0.7866 | 0.6810 | 0.7300 |
| **QP_conf50%_1match** | 0.7715 | 0.7708 | 0.7712 |
| **QP_conf40%_1match** | 0.7715 | 0.7708 | 0.7712 |
| **QP_conf40%_2match** | 0.7289 | 0.7843 | 0.7556 |
| **QP_conf40%_5match** | 0.4812 | 0.8279 | 0.6087 |
| **QP_conf40%_10match** | 0.2885 | 0.8509 | 0.4309 |
| **QP_conf40%_20match** | 0.1332 | 0.9059 | 0.2323 |
| **QP_conf40%_30match** | 0.0439 | 0.9250 | 0.0839 |
| **QP_conf40%_35match** | 0.0321 | 0.9339 | 0.0622 |
| **QP_conf40%_40match** | 0.0139 | 0.9156 | 0.0275 |
| **QP_conf40%_45match** | 0.0002 | 0.6667 | 0.0004 |

**Tabella 4.16:** Risultati ottenuti dalle diverse configurazioni, tenendo in considerazione i pattern estratti con un intervallo temporale di 30 minuti, un intervallo spaziale di 1 km, 3 delta spaziali e 3 delta temporali.

.

Come è possibile vedere dalla Tabella 4.16, sono partito utilizzando una soglia di confidenza del 55% e del 50%, settando il numero di pattern ad 1. Tra le due, quella che ha dato dei risultati migliori è stata quella del 50%. Per questo motivo ho

effettuato ulteriori test con questa soglia, incrementando il numero di pattern prima a 2, poi a 5 ed infine a 20. Tuttavia, non ho notato nessun sostanziale miglioramento della precisione. Questo può essere spiegato dal fatto che molti pattern ottenuti includano altri pattern successivi, magari con una confidenza minore. Infatti, in questo caso, se viene verificato quello che include gli altri, verranno verificati anche tutti quelli inclusi.

A questo punto, ho quindi deciso quindi di effettuare il test considerando soltanto lo stato "QuasiPiena" e provando a settare la soglia di confidenza a 0%, 40% e 50% ed il numero di pattern ad 1. Le configurazioni con soglia a 40% e 50% hanno ottenuto le stesse prestazioni. Tuttavia, ho deciso di proseguire i test usando la soglia del 40%, poiché è stata quella che ha ottenuto più pattern dalla fase di estrazione e, quindi, mi ha permesso di effettuare dei test, settando il numero di pattern ad un valore più alto.

In definitiva, il valore di precisione più alto è stato raggiunto tramite la configurazione che prevede soltanto lo stato **"QuasiPiena"**, una soglia di confidenza al **40%** ed un numero minimo di pattern da verificare uguale a **35**. Questa configurazione ha infatti ottenuto una precisione del **93,4%**.

Partendo dai risultati ottenuti, bisogna trarre alcune considerazioni importanti.

Aumentando molto il numero di pattern che devono essere verificati nella fase di test, affinché un record sia classificato come “QuasiPiena”, andiamo a perdere uno dei punti forti delle regole di associazione, che è quello dell’interpretabilità. Infatti, in questo caso si avrebbero molte regole che spiegano il perché di una certa decisione nella classificazione.

Inoltre, considerando la matrice di confusione generale per il caso in cui ho ottenuto il 93,4% di precisione, presente in Figura 4.31, possiamo vedere che si hanno 325 casi in cui viene individuata una situazione che è veramente critica, 23 casi in cui il sistema predice una situazione come critica, ma in realtà non lo è, e 9785 casi critici persi, ovvero situazioni in cui la situazione era critica, ma il sistema l’ha predetta come normale.

```
AVERAGE VALUES FOR PATTERN TEST:
Confusion matrix:
[[275992 23]
 [9785  325]]
tot_tp=325, tot_fn=9785, tot_fp=23, tot_tn=275992
accuracy=0.96572; recall=0.03214; precision=0.93390; f1_score=0.06215
```

**Figura 4.31:** Matrice di confusione relativa alla configurazione che ha dato il miglior risultato di precisione ed ottenuta considerando soltanto lo stato "QuasiPiena", settando la soglia di confidenza al 40% e un numero di match pari a 35.

Da tale matrice di confusione, possiamo anche vedere come il richiamo ottenuto sia stato molto basso. Tuttavia, nel contesto del bike sharing, avere una precisione elevata permette di rilevare un numero di casi critici con una sicurezza abbastanza alta. Quindi, per esempio, permetterebbe di mandare l'addetto alla ridistribuzione delle bici, in modo da risolvere delle criticità, e in 325 casi su 348 non si farebbe fare il giro a vuoto. Allo stesso tempo, il fatto di avere 9785 falsi negativi implica che al cliente potrebbe essere indicata una stazione con uno stato normale, anche se in realtà essa si trova in uno stato critico. Tuttavia, resta il fatto che non è detto che tutte le stazioni del vicinato siano in una situazione critica; quindi, si potrebbe verificare una situazione in cui per il sistema una situazione sia critica, considerando il vicinato, in quanto magari c'è una stazione vicina nello stato critico, ma poi effettivamente il cliente scelga una delle stazioni vicine che in realtà non erano nello stato critico. Quindi, in questo caso, il numero elevato di falsi negativi non impatterebbe in maniera negativa sul cliente.

### 4.8.2 Esperimento con intervallo temporale di 60 minuti

In questo esperimento ho provato ad estrarre i pattern utilizzando un intervallo temporale di 60 minuti. Dopo di che ho proceduto con filtraggio utilizzando con le seguenti configurazioni:

- Considerando gli stati "Normale" e "QuasiPiena" ed una soglia di confidenza di 0%, ottenendo 1614 pattern;
- Considerando gli stati "Normale" e "QuasiPiena" ed una soglia di confidenza di 50%, ottenendo 119 pattern;
- Considerando gli stati "Normale" e "QuasiPiena" ed una soglia di confidenza di 55%, ottenendo 1 pattern;
- Considerando soltanto lo stato "QuasiPiena" ed una soglia di confidenza di 0%, ottenendo 132 pattern;
- Considerando soltanto lo stato "QuasiPiena" ed una soglia di confidenza di 40%, ottenendo 48 pattern;
- Considerando soltanto lo stato "QuasiPiena" ed una soglia di confidenza di 50%, ottenendo 18 pattern.

Successivamente, tenendo anche conto delle osservazioni fatte nel caso dell'esperimento con 30 minuti, ho proceduto con i test dei pattern filtrati considerando le configurazioni riportate in Tabella 4.17 insieme ai rispettivi risultati ottenuti.

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **N/QP_conf50%_2match** | 0.6827 | 0.6813 | 0.6820 |
| **N/QP_conf50%_5match** | 0.6827 | 0.6813 | 0.6820 |
| **N/QP_conf50%_10match** | 0.6827 | 0.6813 | 0.6820 |
| **QP_conf40%_5match** | 0.3968 | 0.7734 | 0.5245 |
| **QP_conf40%_10match** | 0.2455 | 0.8077 | 0.3766 |
| **QP_conf40%_20match** | 0.1151 | 0.8571 | 0.2029 |
| **QP_conf40%_30match** | 0.0386 | 0.8680 | 0.0739 |
| **QP_conf40%_35match** | 0.0291 | 0.8958 | 0.0564 |
| **QP_conf40%_40match** | 0.0117 | 0.8966 | 0.0232 |
| **QP_conf40%_45match** | 0.0000 | 0.0000 | 0.0000 |

**Tabella 4.17:** Risultati ottenuti dalle diverse configurazioni, tenendo in considerazione i pattern estratti con un intervallo temporale di 60 minuti, un intervallo spaziale di 1 km, 3 delta spaziali e 3 delta temporali.

.

Come si può notare dagli esperimenti presenti in Tabella 4.17, ho iniziato considerando entrambi gli stati "Normale" e "QuasiPiena". Tuttavia, ho ottenuto dei valori bassi. Aumentando leggermente il numero minimo di pattern (partendo da 2, prima a 5 e poi a 10) che dovevano essere verificati affinché il record del test set sia classificata come "QuasiPiena", non ho notato nessun miglioramento nella precisione.

Ho quindi fatto una prova considerando soltanto lo stato "QuasiPiena" ed ho ottenuto una precisione già superiore, considerando soltanto 5 pattern necessari ed una soglia di confidenza settata a 40%. Ho provato quindi ad aumentare il numero di pattern per questa nuova configurazione, ottenendo un miglioramento della precisione fino all'**89,66%**, utilizzando un numero di pattern uguale a **40**. Come si può vedere, spingendosi oltre e provando ad aumentare ulteriormente il numero di pattern necessari, si raggiungono dei casi limite. Infatti, i classificatori non riescono a trovare un numero sufficiente di pattern verificati, classificando tutte i record del test set come "Normali". Questo spiega il fatto per cui si ottiene una precisione nulla. In definitiva, i risultati ottenuti considerando un intervallo di 60 minuti sono stati meno soddisfacenti rispetto a quelli ottenuti considerando un intervallo di 30 minuti.

### 4.8.3 Aumento del richiamo

In questa esperimento ho provato a far aumentare il richiamo, facendo quindi diminuire il numero di falsi positivi, pagando però un decremento della precisione. Per raggiungere questo obiettivo, ho effettuato le predizioni utilizzando due classificatori in cascata: prima di tutto ho applicato il classificatore che fa uso delle regole di associazione; successivamente tutte i record classificati come negativi (stato "Normale") sono stati dati in pasto ad un albero di decisione in modo da eventualmente correggere la predizione dallo stato "Normale" a quello critico. Tuttavia, poiché i record che verrebbero scartati sarebbero soltanto quelli classificati come positivi dal classificatore associativo (325), non mi aspettavo di ottenere delle performance molto lontane da quelle ottenute dall'albero di precisione.

Gli alberi che sono stati utilizzati nel test sono stati quelli ottenuti dalla Grid Search ottimizzando l'F1 Score, mentre la configurazione scelta per il classificatore associativo è stata quella che aveva dato la migliore precisione in assoluto, pari al 93,4%, ovvero quella con soglia di confidenza al 40%, numero minimo di pattern da verificare uguale a 35 e considerando soltanto lo stato "QuasiPiena".

Nella Tabella 4.18 si mettono a confronto i nuovi risultati ottenuti con quelli ottenuti in precedenza dal classificatore associativo e dall'albero di decisione.

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **CascadeClassifiers** | 0.76894 | 0.70989 | 0.73823 |
| **AssociativeClassifier** | 0.03214 | 0.93390 | 0.06215 |
| **DecisionTree** | 0.76894 | 0.70995 | 0.73827 |

**Tabella 4.18:** Risultati ottenuti dai classificatori in cascata, dal classificatore associativo con la migliore configurazione e dell'albero di decisione ottenuto ottimizzando l'F1-Score.

.

Come è possibile notare, il richiamo è passato dal 3,21% al **76,89%**, mentre si è ottenuta una riduzione della precisione dal 93,4% al **70,99%**. Come ci si aspettava, i precedenti risultati si avvicinano di molto a quelli che si erano ottenuti tramite l'utilizzo dell'albero di decisione.

In definitiva si può affermare che le predizioni di situazioni critiche ottenute con le regole, sono predizioni molto affidabili (93,4% di precisione). Se invece si vuole aumentare anche un po' il richiamo, si possono utilizzare i classificatori classici che però, nella versione standard, hanno una precisione più bassa.

### 4.8.4 Aumento dell'interpretabilità

Come detto in precedenza, utilizzando un numero alto di pattern che devono essere verificati nella fase di test, affinché un record di test venga classificato come "QuasiPiena", andiamo a perdere uno dei punti forti della classificazione associativa, che è quello dell'interpretabilità.

Poiché fra il numero di regole che sono verificate in fase di test ce ne sono molte che sono un sotto insieme delle altre, si potrebbe pensare di eliminarle in modo da migliorare l'interpretabilità. Di seguito è riportato un esempio di una regola che ne include un'altra:

$$[\prime QuasiPiena\_T0\_0, QuasiPiena\_T0\_3, QuasiPiena\_T0\_3\prime] \rightarrow [\prime QuasiPiena\_T1\_0\prime] \tag{4.26}$$

$$[\prime QuasiPiena\_T0\_0, QuasiPiena\_T0\_3\prime] \rightarrow [\prime QuasiPiena\_T1\_0\prime] \tag{4.27}$$

Come è possibile vedere, la regola 4.27 è inclusa nella 4.26, in quanto se viene verificata la 4.26, che indica che ci sono almeno due vicini all'istante T0 che distano tre delta spaziali dalla stazione di riferimento e che si trovano in una situazione critica, sicuramente sarà verificata anche la 4.27 che indica, invece, che c'è almeno una stazione vicina all'istante T0 che dista tre delta spaziali e che si trova in una situazione critica. L'interpretabilità sarebbe utile non solo per capire se la predizione ottenuta abbia senso o meno, ma anche per effettuare delle eventuali pianificazioni future (come, per esempio, aumentare o diminuire il numero di stazioni in una certa area) in modo da ridurre il numero di situazioni critiche.

Partendo, quindi, dalla configurazione che mi ha permesso di ottenere il migliore risultato di precisione (soglia di precisione del 40%, numero di pattern da verificare uguale a 35 e considerando solo lo stato "QuasiPiena") ho ottenuto i pattern verificati in un paio di situazioni in cui la predizione è stata "QuasiPiena" e quelli in un paio di situazione in cui la predizione è stata "Normale".

I risultati ottenuti hanno mostrato la presenza di qualche ripetizione, tuttavia, non è facile ed immediato ridurre il numero di pattern verificati.

### 4.8.5 Suddivisione in fasce orarie

In questo esperimento ho eseguito nuovamente l'estrazione dei pattern, considerando separatamente le solite fasce orarie, in modo da andare a valutare sia come variavano le performance del classificatore associativo da una fascia oraria all'altra, sia le performance generali del classificatore, mediando sui risultati ottenuti sulle varie fasce. Dopo l'estrazione, ho proceduto con il filtraggio dei pattern di interesse, ottenendo i seguenti numeri di pattern per ciascuna fascia oraria:

- "0-6": 22 pattern totali
- "6-10": 20 pattern totali
- "10-14": 41 pattern totali
- "14-17": 32 pattern totali
- "17-20": 45 pattern totali
- "20-24": 21 pattern totali

Questi pattern sono poi stati quelli utilizzati dal classificatore per effettuare la fase di test.

Come prima prova ho settato il numero minimo di pattern a 20, in modo da poter effettuare un test su tutte le diverse fasce orarie, ottenendo i risultati mostrati nella Tabella 4.19.

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **QP_conf40%_0-6_20match** | 0.0000 | 0.0000 | 0.0000 |
| **QP_conf40%_6-10_20match** | 0.0000 | 0.0000 | 0.0000 |
| **QP_conf40%_10-14_20match** | 0.0391 | 0.8158 | 0.0747 |
| **QP_conf40%_14-17_20match** | 0.0069 | 0.8750 | 0.0137 |
| **QP_conf40%_17-20_20match** | 0.0148 | 0.7143 | 0.0289 |
| **QP_conf40%_20-24_20match** | 0.0016 | 0.7500 | 0.0033 |

**Tabella 4.19:** Risultati ottenuti dal classificatore associativo nelle diverse fasce considerando un numero minimo di match uguale a 20, una soglia di confidenza al 40% e come stato soltanto "QuasiPiena".

Come è possibile notare per alcune fasce in cui il numero di pattern totali è molto vicino a 20, si sono ottenuti dei risultati nulli. Questo è dovuto al fatto che il classificatore non è riuscito a classificare nessun record come positivo.

Dopo di che, visto che alcune fasce mi permettevano di utilizzare un numero più alto di pattern, ho proceduto con l'effettuare i test per ogni singola fascia in modo da ottenere una precisione più alta possibile.

### FASCIA ORARIA 0 - 6

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **QP_conf40%_0-6_1match** | 0.9530 | 0.9359 | 0.9444 |
| **QP_conf40%_0-6_15match** | 0.0340 | 0.9248 | 0.0656 |
| **QP_conf40%_0-6_18match** | 0.0122 | 0.9362 | 0.0240 |
| **QP_conf40%_0-6_20match** | 0.0000 | 0.0000 | 0.0000 |
| **QP_conf40%_0-6_25match** | 0.0000 | 0.0000 | 0.0000 |

**Tabella 4.20:** Risultati ottenuti dal classificatore associativo nella fascia oraria **0-6** al variare del numero minimo di match e considerando una soglia di confidenza al 40% e come stato soltanto "QuasiPiena".

Come è possibile vedere dalla Tabella 4.20, in questa fascia ho ottenuto una precisione massima del **93,62%**, settando il numero minimo di pattern a **18**. Se invece guardiamo la prima riga, si può notare che con un solo pattern si raggiunge un richiamo del **95.30%**, che è molto più alto del solito, ma anche una precisione abbastanza alta.

### FASCIA ORARIA 6 - 10

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **QP_conf40%_6-10_1match** | 0.5965 | 0.5864 | 0.5914 |
| **QP_conf40%_6-10_15match** | 0.0083 | 0.6087 | 0.0165 |
| **QP_conf40%_6-10_18match** | 0.0000 | 0.0000 | 0.0000 |
| **QP_conf40%_6-10_20match** | 0.0000 | 0.0000 | 0.0000 |
| **QP_conf40%_6-10_25match** | 0.0000 | 0.0000 | 0.0000 |

**Tabella 4.21:** Risultati ottenuti dal classificatore associativo nella fascia oraria **6-10** al variare del numero minimo di match e considerando una soglia di confidenza al 40% e come stato soltanto "QuasiPiena".

Come illustrato dalla Tabella 4.21, in questa fascia oraria non si sono raggiunti risultati soddisfacenti. Probabilmente questo è dovuto al fatto che nel periodo della giornata considerato c'è molto movimento, quindi il classificatore non riesce bene

ad effettuare la predizione. Infatti, poiché questa fascia ricopre un momento della giornata che coincide con l'apertura dei negozi, uffici, università ecc, probabilmente non ci sarà una certa regolarità nelle situazioni delle stazioni, oppure semplicemente i pattern non sono in grado di effettuare le predizioni.

### FASCIA ORARIA 10 - 14

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **QP_conf40%_10-14_1match** | 0.7001 | 0.6846 | 0.6923 |
| **QP_conf40%_10-14_15match** | 0.0928 | 0.7313 | 0.1647 |
| **QP_conf40%_10-14_20match** | 0.0391 | 0.8158 | 0.0747 |
| **QP_conf40%_10-14_25match** | 0.0126 | 0.9091 | 0.0249 |
| **QP_conf40%_10-14_30match** | 0.0051 | 0.8000 | 0.0100 |

**Tabella 4.22:** Risultati ottenuti dal classificatore associativo nella fascia oraria **10-14** al variare del numero minimo di match e considerando una soglia di confidenza al 40% e come stato soltanto "QuasiPiena".

In questo caso i risultati sono stati leggermente migliori, ma comunque più bassi della fascia 0-6.

### FASCIA ORARIA 14 - 17

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **QP_conf40%_14-17_1match** | 0.6330 | 0.6089 | 0.6208 |
| **QP_conf40%_14-17_15match** | 0.0158 | 0.6957 | 0.0309 |
| **QP_conf40%_14-17_20match** | 0.0069 | 0.8750 | 0.0137 |
| **QP_conf40%_14-17_25match** | 0.0049 | 1.0000 | 0.0098 |
| **QP_conf40%_14-17_30match** | 0.0000 | 0.0000 | 0.0000 |

**Tabella 4.23:** Risultati ottenuti dal classificatore associativo nella fascia oraria **14-17** al variare del numero minimo di match e considerando una soglia di confidenza al 40% e come stato soltanto "QuasiPiena".

In questa fascia oraria si è raggiunta una precisione del 100%, con un numero di pattern uguale a 25. Tuttavia, questo deriva dal fatto che per alcune stazioni si è raggiunta una precisione del 100%, mentre per tutte le altre una precisione nulla. La matrice di confusione generale ottenuta nel caso precedente è mostrata in Figura 4.33.

```
AVERAGE VALUES FOR PATTERN TEST:
Confusion matrix:
[[40604 0]
[1006 5]]
tot_tp=5, tot_fn=1006, tot_fp=0, tot_tn=40604
accuracy=0.97582; recall=0.00494; precision=1.0; f1_score=0.00984
```

**Figura 4.32:** Matrice di confusione globale relativa alla configurazione che ha dato il 100% di precisione, ottenuta considerando soltanto lo stato "QuasiPiena", settando la soglia di confidenza al 40% e un numero di match pari a 25.

Come è possibile vedere dalla Figura 4.33, sommando tutti i risultati delle diverse stazioni, ci sono stati solo 5 veri positivi totali e 0 falsi positivi totali. Questo ha determinato una precisione del 100%. Molte stazioni hanno invece ottenuto una precisione dello 0% dovuto al fatto che il classificatore ha classificato tutti i record di test come negativi, ovvero appartenenti alla classe “Normale”.

### FASCIA ORARIA 17 - 20

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **QP_conf40%_17-20_1match** | 0.5601 | 0.5879 | 0.5737 |
| **QP_conf40%_17-20_15match** | 0.0310 | 0.6562 | 0.0592 |
| **QP_conf40%_17-20_20match** | 0.0148 | 0.7143 | 0.0289 |
| **QP_conf40%_17-20_25match** | 0.0066 | 1.0000 | 0.0132 |
| **QP_conf40%_17-20_30match** | 0.0044 | 1.0000 | 0.0088 |
| **QP_conf40%_17-20_35match** | 0.0015 | 1.0000 | 0.0029 |

**Tabella 4.24:** Risultati ottenuti dal classificatore associativo nella fascia oraria **17-20** al variare del numero minimo di match e considerando una soglia di confidenza al 40% e come stato soltanto "QuasiPiena".

Anche in questa fascia oraria si è raggiunta una precisione del 100% con diversi numeri di pattern. Il motivo è lo stesso di quello spiegato nella fascia precedente. Per completezza vengono riportate le matrici di confusione generali relative al caso di 25 match (Figura 4.33), 30 match (Figura 4.34) e 35 match (Figura 4.35).

```
AVERAGE VALUES FOR PATTERN TEST:
Confusion matrix:
[[40260 0]
[1346 9]]
tot_tp=9, tot_fn=1346, tot_fp=0, tot_tn=40260
accuracy=0.96765; recall=0.00664; precision=1.0; f1_score=0.01319
```

**Figura 4.33:** Matrice di confusione globale relativa alla configurazione che ha dato il 100% di precisione, ottenuta considerando soltanto lo stato "QuasiPiena", settando la soglia di confidenza al 40% e un numero di match pari a 25.

```
AVERAGE VALUES FOR PATTERN TEST:
Confusion matrix:
[[40260 0]
[1349 6]]
tot_tp=6, tot_fn=1349, tot_fp=0, tot_tn=40260
accuracy=0.96758; recall=0.00442; precision=1.0; f1_score=0.00881
```

**Figura 4.34:** Matrice di confusione globale relativa alla configurazione che ha dato il 100% di precisione, ottenuta considerando soltanto lo stato "QuasiPiena", settando la soglia di confidenza al 40% e un numero di match pari a 30.

```
AVERAGE VALUES FOR PATTERN TEST:
Confusion matrix:
[[40260 0]
[1353 2]]
tot_tp=2, tot_fn=1353, tot_fp=0, tot_tn=40260
accuracy=0.96748; recall=0.00147; precision=1.0; f1_score=0.00294
```

**Figura 4.35:** Matrice di confusione globale relativa alla configurazione che ha dato il 100% di precisione, ottenuta considerando soltanto lo stato "QuasiPiena", settando la soglia di confidenza al 40% e un numero di match pari a 35.

### FASCIA ORARIA 20 - 24

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **QP_conf40%_20-24_1match** | 0.8321 | 0.8744 | 0.8527 |
| **QP_conf40%_20-24_15match** | 0.0168 | 0.8857 | 0.0331 |
| **QP_conf40%_20-24_18match** | 0.0016 | 0.7500 | 0.0033 |
| **QP_conf40%_20-24_20match** | 0.0016 | 0.7500 | 0.0033 |
| **QP_conf40%_20-24_25match** | 0.0000 | 0.0000 | 0.0000 |

**Tabella 4.25:** Risultati ottenuti dal classificatore associativo nella fascia oraria **20-24** al variare del numero minimo di match e considerando una soglia di confidenza al 40% e come stato soltanto "QuasiPiena".

In quest'ultima fascia invece, il risultato più alto è stato quello raggiunto utilizzando **15** pattern, raggiungendo una precisione del **88.57%**.

### Risultati generali

Come fatto nel caso dell'albero di precisione, ho ottenuto le matrici di confusione globali sommando per ciascun numero minimo di pattern considerato le varie matrici di confusione ottenute nelle diverse fasce orarie. In questo modo ho ottenuto le performance medie per ciascun numero minimo di pattern. I risultati ottenuti, sono mostrati nella Tabella 4.26.

| | overall_recall | overall_precision | overall_f1_score |
|---|---|---|---|
| **QP_conf40_1match** | 0.76559 | 0.76435 | 0.76497 |
| **QP_conf40_15match** | 0.03366 | 0.77871 | 0.06453 |
| **QP_conf40_18match** | 0.01615 | 0.79556 | 0.03166 |
| **QP_conf40_20match** | 0.00830 | 0.77966 | 0.01643 |
| **QP_conf40_25match** | 0.00307 | 0.94444 | 0.00612 |
| **QP_conf40_30match** | 0.00126 | 0.87500 | 0.00252 |

**Tabella 4.26:** Risultati generali ottenuti dal classificatore associativo al variare del numero minimo di match e considerando una soglia di confidenza al 40% e come stato soltanto "QuasiPiena".

La precisione più alta si è avuta nel caso di 25 pattern, con un risultato del **94.44%**. Tuttavia, nei casi in cui si è utilizzato un numero elevato di pattern, il contributo di alcune fasce sarà totalmente nullo, in quanto il numero totale di pattern estratti è inferiore.

Mentre i risultati in assoluto migliori nelle varie fasce sono stati ottenuti dalle configurazioni mostrate in Tabella 4.27.

| | overall_recall | overall_precision | overall_f1_score |
|---|---|---|---|
| **QP_conf40%_0-6_18match** | 0.0122 | 0.9362 | 0.0240 |
| **QP_conf40%_6-10_15match** | 0.0083 | 0.6087 | 0.0165 |
| **QP_conf40%_10-14_25match** | 0.0126 | 0.9091 | 0.0249 |
| **QP_conf40%_14-17_25match** | 0.0049 | 1.0000 | 0.0098 |
| **QP_conf40%_17-20_25match** | 0.0066 | 1.0000 | 0.0132 |
| **QP_conf40%_20-24_15match** | 0.0168 | 0.8857 | 0.0331 |

**Tabella 4.27:** Risultati migliori ottenuti dal classificatore associativo dai diversi numeri minimi di match e considerando una soglia di confidenza al 40% e come stato soltanto "QuasiPiena".

### 4.8.6 Classificatore "stupido"

Un altro esperimento che è stato condotto, per quanto riguarda il classificatore associativo, è consistito nel provare a far classificare tutti i record con lo stato che avevano all'istante precedente e vedere che performance si riesce ad ottenere con questo tipo di classificatore più "stupido".

Questo test ha permesso di valutare se la fascia in cui si sono avuti i migliori risultati, ovvero quella "0-6", sia talmente piatta e stazionaria che non serve avere qualcosa di intelligente, ma basta un classificatore abbastanza stupido che predica semplicemente con lo stato precedente. Per fare questo test, basta semplicemente utilizzare come *"y_predict"* la *"y_true"* slittata in avanti di 1. I risultati ottenuti sono mostrati nella Tabella 4.28.

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **time_slot_0-6** | 0.8896 | 0.8891 | 0.8893 |
| **time_slot_6-10** | 0.4899 | 0.4913 | 0.4906 |
| **time_slot_10-14** | 0.6138 | 0.6142 | 0.6140 |
| **time_slot_14-17** | 0.4812 | 0.4807 | 0.4810 |
| **time_slot_17-20** | 0.5203 | 0.5203 | 0.5203 |
| **time_slot_20-24** | 0.7848 | 0.7852 | 0.7850 |

**Tabella 4.28:** Risultati ottenuti dal classificatore stupido nelle diverse fasce orarie.

Come ci si aspettava, per le fasce 0-6 e 20-24 questo tipo di classificatore riesce ad ottenere delle buone performance, anche se non buone come quelle ottenute tramite il classificatore associativo. Questo conferma il fatto che in quelle fasce la situazione è abbastanza stazionaria. Tuttavia, le prestazioni del classificatore associativo da me ottenuto sono state migliori di quelli ottenuti in questo esperimento.

### 4.8.7 Considerazione dello stato "Normale"

L'idea di questo test è nata dal fatto che per l'albero ci fossero dei cammini con una confidenza più alta di circa il 70%, mentre nelle regole estratte e poi successivamente filtrate queste confidenze non erano presenti. Questo è stato dovuto al fatto che molte regole a confidenza più alta sono state filtrate dopo essere state estratte. Per esempio, era stato preso in considerazione un filtro che tenesse conto soltanto del cambiamento di stato a "QuasiPiena", quindi avevo filtrato tutte le regole che contenessero nel corpo lo stato "Normale". Questo filtro, che ha eliminato diverse regole con una confidenza più alta, era stato implementato in quanto, considerando l'intero dataset ed anche lo stato "Normale", si ottenevano troppe regole, spingendo il classificatore a predire sempre come "QuasiPiena". Tuttavia, adesso che il dataset è stato diviso per fasce e quindi il numero di pattern è più basso, potrebbe aver senso provare a considerare anche i pattern contenenti anche il passaggio di stato a "Normale".

Applicando quindi il nuovo filtro, che mantiene anche i pattern contenenti lo stato "Normale" e con una soglia di confidenza al 40%, si sono ottenuti i seguenti numeri di pattern per ciascuna fascia oraria:

- "0-6": 287 pattern totali
- "6-10": 159 pattern totali

- "10-14": 306 pattern totali
- "14-17": 287 pattern totali
- "17-20": 323 pattern totali
- "20-24": 301 pattern totali

Il tempo necessario per il testing dei pattern è aumentato considerevolmente a causa dell'ancora troppo elevato numero di pattern. Quindi ho proceduto a ricavare una soglia di confidenza più alta, in modo da ridurre il numero di pattern e provare ad aumentare anche la precisione. Nella Figura 4.36 sono mostrati i plot, per le diverse fasce orarie, del numero di pattern ottenuti al variare della soglia di confidenza.

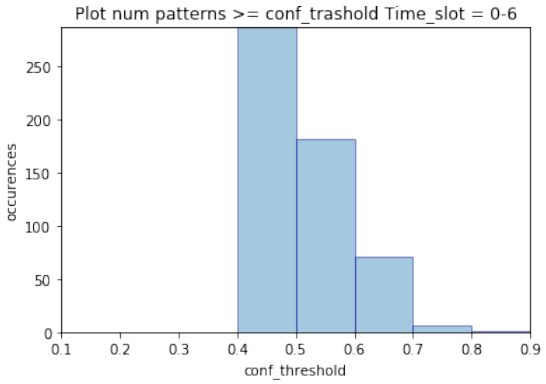
Come è possibile vedere in 3 fasce (4.36 b, d ed e) non ci sono pattern con una confidenza più alta del 60%. Ho quindi effettuato i test considerando come valori per la soglia 40%, 50% e 55% e come numero minimo di pattern da verificare 1. In questo modo mi sono potuto fare un'idea dell'impatto del valore della soglia di confidenza sui risultati ottenuti.

I risultati del test con una soglia di confidenza al **40%** ed un numero di pattern settato ad **1** sono mostrati nella Tabella 4.29.

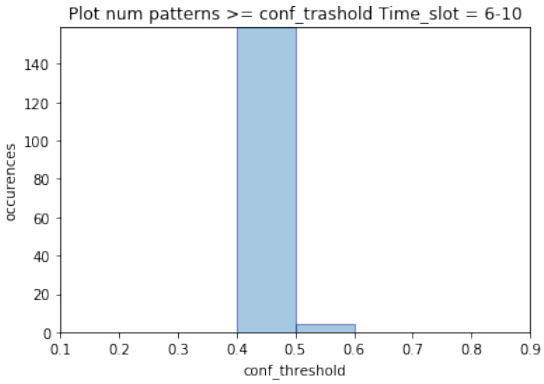
| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **N/QP_conf40%_0-6_1match** | 1.0000 | 0.0466 | 0.0891 |
| **N/QP_conf40%_6-10_1match** | 0.5965 | 0.5864 | 0.5914 |
| **N/QP_conf40%_10-14_1match** | 0.7140 | 0.2421 | 0.3616 |
| **N/QP_conf40%_14-17_1match** | 0.6330 | 0.6089 | 0.6208 |
| **N/QP_conf40%_17-20_1match** | 0.5601 | 0.5879 | 0.5737 |
| **N/QP_conf40%_20-24_1match** | 0.8603 | 0.3717 | 0.5191 |

**Tabella 4.29:** Risultati ottenuti dal classificatore associativo nelle diverse fasce orarie, considerando una soglia di confidenza al 40%, un numero minimo di pattern da verificare pari ad 1 e come stati "Normale" e "QuasiPiena".

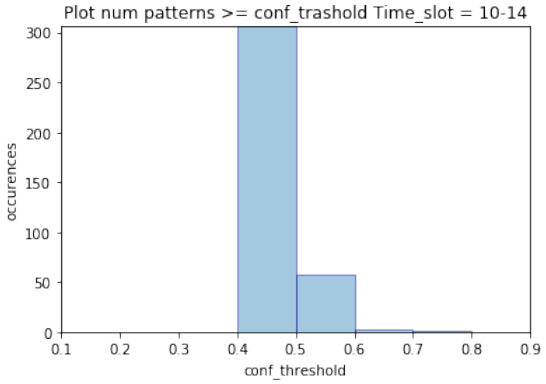
I risultati bassi di precisione e più alti di richiamo sono dovuti al fatto che molto spesso con un numero minimo di pattern da verificare pari ad 1, il classificatore riesce a trovare sempre almeno un pattern che sia verificato. Quindi in molti casi, per molte stazioni capita che nessun record venga predetto come "Normale". Questo può essere visto da un esempio preso dalla matrice di confusione totale per la fascia 0-6, presente in Figura 4.37.



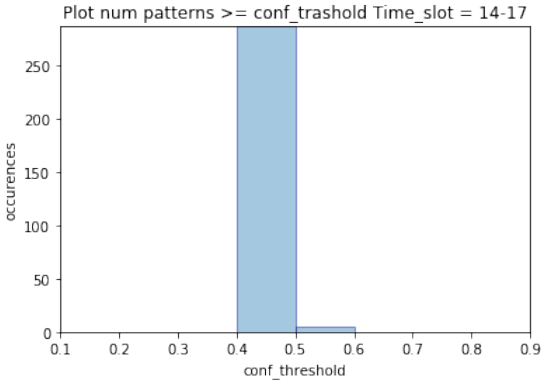

**(a)** fascia oraria 0-6



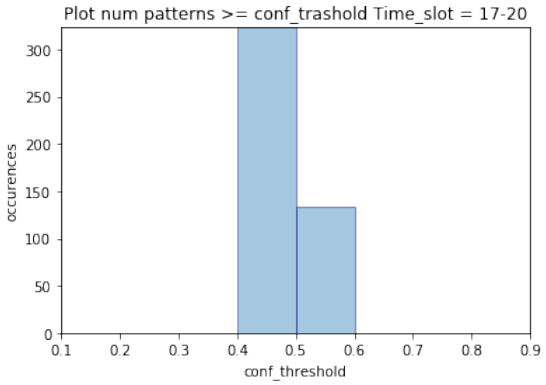

**(b)** fascia oraria 6-10



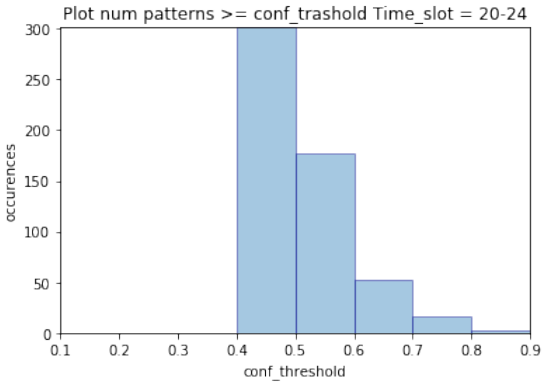

**(c)** fascia oraria 10-14




**(d)** fascia oraria 14-17




**(e)** fascia oraria 17-20




**(f)** fascia oraria 20-24

**Figura 4.36:** L'immagine contiene i plot relativi al numero di pattern con i diversi valori di confidenza per ognuna delle fasce orarie considerate.

```
AVERAGE VALUES FOR PATTERN TEST:
Confusion matrix:
[[0 73945]
[0 3615]]
tot_tp=3615, tot_fn=0, tot_fp=73945, tot_tn=0
accuracy=0.04660; recall=1.0; precision=0.04660; f1_score=0.08906
```

**Figura 4.37:** Matrice di confusione globale relativa alla configurazione ottenuta settando la soglia di confidenza al 40% e un numero di match pari a 1 e considerando gli stati "Normale" e "QuasiPiena".

Per questo motivo si è proceduto ad aumentare la soglia di confidenza.

Filtrando i risultati, usando una soglia del 50%, i numeri di pattern ottenuti per le diverse fasce sono stati i seguenti:

- "0-6": 182 pattern totali
- "6-10": 4 pattern totali
- "10-14": 57 pattern totali
- "14-17": 5 pattern totali
- "17-20": 133 pattern totali
- "20-24": 177 pattern totali

I risultati del test con una soglia di confidenza al **50%** ed un numero di pattern settato ad **1** sono stati quelli contenuti in Tabella 4.30.

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **N/QP_conf50%_0-6_1match** | 0.9546 | 0.9335 | 0.9439 |
| **N/QP_conf50%_6-10_1match** | 0.1710 | 0.6308 | 0.2691 |
| **N/QP_conf50%_10-14_1match** | 0.7001 | 0.6846 | 0.6923 |
| **N/QP_conf50%_14-17_1match** | 0.6320 | 0.6092 | 0.6204 |
| **N/QP_conf50%_17-20_1match** | 0.5601 | 0.5879 | 0.5737 |
| **N/QP_conf50%_20-24_1match** | 0.8533 | 0.7689 | 0.8089 |

**Tabella 4.30:** Risultati ottenuti dal classificatore associativo nelle diverse fasce orarie, considerando una soglia di confidenza al 50%, un numero minimo di pattern da verificare pari ad 1 e come stati "Normale" e "QuasiPiena".

Filtrando i risultati, usando una soglia del 55%, i numeri di pattern ottenuti per le diverse fasce sono stati, invece, i seguenti:

- "0-6": 113 pattern totali
- "6-10": 0 pattern totali
- "10-14": 11 pattern totali
- "14-17": 0 pattern totali
- "17-20": 21 pattern totali
- "20-24": 88 pattern totali

I risultati del test con una soglia di confidenza pari al 55% ed un numero di pattern settato ad 1 sono mostrati in Tabella 4.31.

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| **N/QP_conf55%_0-6_1match** | 0.9546 | 0.9335 | 0.9439 |
| **N/QP_conf55%_6-10_1match** | 0.0000 | 0.0000 | 0.0000 |
| **N/QP_conf55%_10-14_1match** | 0.6395 | 0.6886 | 0.6632 |
| **N/QP_conf55%_14-17_1match** | 0.0000 | 0.0000 | 0.0000 |
| **N/QP_conf55%_17-20_1match** | 0.5601 | 0.5879 | 0.5737 |
| **N/QP_conf55%_20-24_1match** | 0.8321 | 0.8744 | 0.8527 |

**Tabella 4.31:** Risultati ottenuti dal classificatore associativo nelle diverse fasce orarie, considerando una soglia di confidenza al 55%, un numero minimo di pattern da verificare pari ad 1 e come stati "Normale" e "QuasiPiena".

In definitiva, considerando i risultati ottenuti per le diverse soglie di confidenza, le configurazioni che sembrano aver dato dei risultati generalmente migliori sono state quelle con una soglia del 50%. Quindi, ho deciso di procedere effettuando i test utilizzando questa soglia di confidenza per le regole e variare il numero minimo di pattern.

I risultati globali, ottenuti mediando su tutte le fasce per ciascun numero di pattern, sono stati quelli riportati Figura 4.38a, mentre in Figura 4.38b abbiamo quelli che si erano ottenuti considerando soltanto lo stato “QuasiPiena”.

Come si può notare, si è avuto un leggero miglioramento nella precisione per quanto riguarda i numeri di pattern 1, 15, 18 e 20. Inoltre, anche il richiamo è stato notevolmente migliorato.

| | oa_recall | oa_precision | oa_f1_score |
|---|---|---|---|
| N/QP_conf50_1match | 0.70513 | 0.76965 | 0.73598 |
| N/QP_conf50_15match | 0.30948 | 0.81011 | 0.44787 |
| N/QP_conf50_18match | 0.25526 | 0.81130 | 0.38834 |
| N/QP_conf50_20match | 0.18975 | 0.79001 | 0.30600 |
| N/QP_conf50_25match | 0.13011 | 0.75656 | 0.22204 |
| N/QP_conf50_30match | 0.10421 | 0.76949 | 0.18356 |

**(a)** Stati considerati: "Normale" e "QuasiPiena"

| | oa_recall | oa_precision | oa_f1_score |
|---|---|---|---|
| QP_conf40_1match | 0.76559 | 0.76435 | 0.76497 |
| QP_conf40_15match | 0.03366 | 0.77871 | 0.06453 |
| QP_conf40_18match | 0.01615 | 0.79556 | 0.03166 |
| QP_conf40_20match | 0.00830 | 0.77966 | 0.01643 |
| QP_conf40_25match | 0.00307 | 0.94444 | 0.00612 |
| QP_conf40_30match | 0.00126 | 0.87500 | 0.00252 |

**(b)** Stato considerato: "QuasiPiena"

**Figura 4.38:** L'immagine (a) contiene i risultati dell'esperimento effettuato considerando entrambi gli stati "Normale" e "QuasiPiena" ed una soglia di confidenza al 50%. L'immagine (b) contiene i risultati dell'esperimento effettuato considerando solamente lo stato "QuasiPiena" ed una soglia di confidenza al 40%. La sigla "oa" è l'acronimo di "overall", per indicare che i risultati mostrati sono la media di quelli ottenuti su tutte le diverse fasce orarie.

### 4.8.8 Inserimento di una soglia di supporto nel filtraggio dei pattern

Un ulteriore esperimento che è stato condotto, in modo da aumentare ulteriormente la precisione a discapito del richiamo, è stato quello di applicare un filtro che scarti i pattern con richiami più bassi. Infatti, i pattern che si presentano meno volte nei dati di training potrebbero essere statisticamente poco stabili nel tempo e quindi potrebbero portare il classificatore a fare delle predizioni errate (anche se usando più regole di match in parte si cerca già di ovviare a questo problema). Ho proceduto quindi a filtrare i pattern, mantenendo solo quelli con un valore di confidenza maggiore del 50% ed un valore di supporto maggiore o uguale a 100 e a 200, in modo da vedere fosse possibile aumentare ulteriormente la confidenza.

I numeri di pattern ottenuti per ogni fascia oraria, considerando una soglia di supporto settata a 100, sono stati i seguenti:

- "0-6": 146 pattern totali
- "6-10": 3 pattern totali
- "10-14": 45 pattern totali
- "14-17": 5 pattern totali
- "17-20": 128 pattern totali
- "20-24": 99 pattern totali

I risultati globali, ottenuti mediando su tutte le fasce per ciascun numero di pattern, sono stati quelli riportati Figura 4.39a, mentre in Figura 4.39b abbiamo quelli che si erano ottenuti considerando una soglia di supporto settata a 0.

| | oa_recall | oa_precision | oa_f1_score |
|---|---|---|---|
| N/QP_conf50_sup100_1match | 0.70080 | 0.77184 | 0.73461 |
| N/QP_conf50_sup100_15match | 0.29387 | 0.81344 | 0.43176 |
| N/QP_conf50_sup100_18match | 0.14960 | 0.76441 | 0.25023 |
| N/QP_conf50_sup100_20match | 0.13652 | 0.75878 | 0.23141 |
| N/QP_conf50_sup100_25match | 0.09384 | 0.74713 | 0.16674 |
| N/QP_conf50_sup100_30match | 0.04656 | 0.66839 | 0.08706 |

**(a)** Soglia di supporto: 100

| | oa_recall | oa_precision | oa_f1_score |
|---|---|---|---|
| N/QP_conf50_sup0_1match | 0.70513 | 0.76965 | 0.73598 |
| N/QP_conf50_sup0_15match | 0.30948 | 0.81011 | 0.44787 |
| N/QP_conf50_sup0_18match | 0.25526 | 0.81130 | 0.38834 |
| N/QP_conf50_sup0_20match | 0.18975 | 0.79001 | 0.30600 |
| N/QP_conf50_sup0_25match | 0.13011 | 0.75656 | 0.22204 |
| N/QP_conf50_sup0_30match | 0.10421 | 0.76949 | 0.18356 |

**(b)** Soglia di supporto: 0

**Figura 4.39:** L'immagine (a) contiene i risultati dell'esperimento effettuato considerando una soglia di supporto settata a 100. L'immagine (b) contiene i risultati dell'esperimento effettuato considerando una soglia di supporto settata a 0. La sigla "oa" è l'acronimo di "overall", per indicare che i risultati mostrati sono la media di quelli ottenuti su tutte le diverse fasce orarie.

Come possiamo vedere, nella maggior parte di casi abbiamo avuto dei valori di richiamo e precisione più bassi.

Eseguendo il filtraggio con un valore di soglia di confidenza al 50% e un valore di soglia di supporto a 200, il numero di pattern ottenuti è stato il seguente:

- "0-6": 81 pattern totali
- "6-10": 3 pattern totali
- "10-14": 27 pattern totali
- "14-17": 3 pattern totali
- "17-20": 104 pattern totali
- "20-24": 48 pattern totali

I risultati del test sono stati mostrati in Tabella 4.32.

| | overall_recall | overall_precision | overall_f1_score |
|---|---|---|---|
| N/QP_conf50_sup200_1match | 0.70008 | 0.77212 | 0.73434 |
| N/QP_conf50_sup200_15match | 0.13408 | 0.74226 | 0.22713 |
| N/QP_conf50_sup200_18match | 0.05206 | 0.63827 | 0.09627 |
| N/QP_conf50_sup200_20match | 0.04728 | 0.62981 | 0.08796 |
| N/QP_conf50_sup200_25match | 0.03140 | 0.60839 | 0.05972 |
| N/QP_conf50_sup200_30match | 0.02093 | 0.62366 | 0.04050 |

**Tabella 4.32:** Risultati ottenuti dal classificatore associativo per diversi numeri minimi di match, considerando una soglia di confidenza al 50%, una soglia di supporto pari a 200 e come stati "Normale" e "QuasiPiena".

Come è possibile notare, in questo caso invece si è avuto un degrado generale delle performance.

### 4.8.9 Considerazione dell'effettivo cambio di stato

In quest'ultimo esperimento ho fatto in modo che il classificatore associativo consideri l'effettivo cambio di stato delle stazioni in fase di classificazione. In altre parole, nella fase di classificazione sono andato a valutare non soltanto gli stati delle stazioni in quel preciso istante considerato, ma anche lo stato delle stazioni negli istanti di tempo precedenti, in modo da controllare se ci sia stato effettivamente un cambio di stato.

I risultati ottenuti dal test, con una soglia di confidenza del 50%, soglia di supporto a 0 e numero di pattern 1, sono mostrati in Figura 4.40a, mentre in Figura 4.40b sono presentati i risultati precedentemente ottenuti non consideranto il cambio di stato.

Come ci si aspettava, si è avuto una diminuzione sostanziale del richiamo. Tuttavia, si è avuto un degrado delle performance riguardo la precisione nelle fasce in cui si aveva una situazione stazionaria ed un miglioramento nelle fasce in cui la situazione era più variabile. Questo potrebbe essere dovuto al fatto che, con questo approccio, si vanno a perdere i casi di stazionarietà, ma si migliora in quei casi in cui la situazione è più variabile.

Ho quindi proceduto con i test, utilizzando i soliti valori di default per il numero di pattern, in modo da calcolare le prestazioni generali del modello. I risultati ottenuti sono mostrati nella Figura 4.41a, mentre in Figura 4.41b sono mostrati i risultati ottenuti in precedenza considerano soltanto lo stato corrente.

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| N/QP_conf50%_0-6_1match | 0.0329 | 0.7169 | 0.0629 |
| N/QP_conf50%_6-10_1match | 0.0584 | 0.6242 | 0.1068 |
| N/QP_conf50%_10-14_1match | 0.0972 | 0.7368 | 0.1718 |
| N/QP_conf50%_14-17_1match | 0.0168 | 0.7083 | 0.0329 |
| N/QP_conf50%_17-20_1match | 0.1815 | 0.5899 | 0.2777 |
| N/QP_conf50%_20-24_1match | 0.1625 | 0.6500 | 0.2600 |

**(a)** Con cambio di stato

| | avg_recall | avg_precision | avg_f1_score |
|---|---|---|---|
| N/QP_conf50%_0-6_1match | 0.9546 | 0.9335 | 0.9439 |
| N/QP_conf50%_6-10_1match | 0.1710 | 0.6308 | 0.2691 |
| N/QP_conf50%_10-14_1match | 0.7001 | 0.6846 | 0.6923 |
| N/QP_conf50%_14-17_1match | 0.6320 | 0.6092 | 0.6204 |
| N/QP_conf50%_17-20_1match | 0.5601 | 0.5879 | 0.5737 |
| N/QP_conf50%_20-24_1match | 0.8533 | 0.7689 | 0.8089 |

**(b)** Senza cambio di stato

**Figura 4.40:** L'immagine (a) contiene i risultati ottenuti nelle diverse fasce orarie, considerando soltanto un match e l'effettivo cambio di stato delle stazioni tra il momento corrente e quello precedente. L'immagine (b) contiene, invece, risultati che si erano ottenuti non considerando il cambio di stato.

| | oa_recall | oa_precision | oa_f1_score |
|---|---|---|---|
| N/QP_conf50_1match | 0.08418 | 0.65108 | 0.14908 |
| N/QP_conf50_15match | 0.00370 | 0.66129 | 0.00736 |
| N/QP_conf50_18match | 0.00244 | 0.65854 | 0.00486 |
| N/QP_conf50_20match | 0.00217 | 0.66667 | 0.00433 |
| N/QP_conf50_25match | 0.00081 | 0.52941 | 0.00162 |
| N/QP_conf50_30match | 0.00063 | 0.50000 | 0.00126 |

**(a)** Con cambio di stato

| | oa_recall | oa_precision | oa_f1_score |
|---|---|---|---|
| N/QP_conf50_1match | 0.70513 | 0.76965 | 0.73598 |
| N/QP_conf50_15match | 0.30948 | 0.81011 | 0.44787 |
| N/QP_conf50_18match | 0.25526 | 0.81130 | 0.38834 |
| N/QP_conf50_20match | 0.18975 | 0.79001 | 0.30600 |
| N/QP_conf50_25match | 0.13011 | 0.75656 | 0.22204 |
| N/QP_conf50_30match | 0.10421 | 0.76949 | 0.18356 |

**(b)** Senza cambio di stato

**Figura 4.41:** L'immagine (a) contiene i risultati generali dell'esperimento effettuato considerando l'effettivo cambio di stato delle stazioni tra il momento corrente e quello precedente. L'immagine (b) contiene, invece, risultati generali che si erano ottenuti non considerando il cambio di stato. La sigla "oa" è l'acronimo di "overall", per indicare che i risultati mostrati sono la media di quelli ottenuti su tutte le diverse fasce orarie.

Come è possibile vedere si è ottenuto un sostanziale degrado delle performance generali.

# Capitolo 5

# Conclusioni

Nel presente capitolo verranno enunciate le conclusioni sui risultati ottenuti ed eventuali possibili lavori futuri, volti alla prosecuzione del lavoro effettuato.

## 5.1 Risultati ottenuti

Nel presente documento sono stati presentati diversi esperimenti con i relativi risultati ottenuti, volti a raggiungere l'obiettivo della tesi, ovvero quello di verificare se un classificatore associativo fosse in grado di effettuare delle predizioni di buona qualità, basandosi su dei pattern contenenti informazioni spaziali e temporali. Lo scopo di queste predizioni consiste nel rilevare anticipatamente e prevenire delle situazioni critiche riguardanti delle stazioni di bike sharing.

Una volta trovata la configurazione che mi ha permesso di estrarre dei pattern con una qualità sufficiente, li ho filtrati in maniera opportuna e utilizzati per effettuare vari test del classificatore associativo. Nel contempo, ho effettuato anche vari test utilizzando l'albero di decisione, in modo da avere un termine di paragone per i risultati via via ottenuti dal classificatore associativo.

Sono state implementate diverse versioni del classificatore associativo in modo da esplorare diverse strade, con l'obiettivo di raggiungere dei risultati di precisione sufficientemente elevati. Proprio per massimizzare la precisione è stata introdotta, nel classificatore associativo, una soglia indicante il numero minimo di pattern che dovessero essere verificati affinché il record di test venisse classificato come positivo. In questo modo, al crescere di questa soglia, è stato possibile incrementare la precisione a discapito del richiamo. Una cosa simile è stata fatta con l'albero di decisione, introducendo una soglia indicante la probabilità minima che un record

dovesse avere di appartenere alla classe positiva, per essere classificato effettivamente come positivo.

Uno dei primi esperimenti è consistito nel provare a variare la dimensione dell'intervallo considerato (paragrafi 4.8.1 e 4.8.2), in modo da valutare l'effetto di questo parametro sulla classificazione. I risultati ottenuti, considerando degli intervalli prima di 30 minuti e poi di 60 minuti, hanno mostrato dei peggioramenti nella precisione al crescere della dimensione dell'intervallo.

Un altro importante esperimento condotto è stato quello di introdurre la possibilità di suddividere i dati in diverse fasce orarie della giornata ed estrarre i pattern separatamente per le diverse fasce (paragrafo 4.8.5). La stessa cosa è stata ovviamente fatta nel caso dell'albero di decisione. Questo mi ha permesso di confrontare sia le prestazioni di entrambi i classificatori nelle diverse fasce orarie, sia le prestazioni generali mediando su tutte le fasce.

È stata, inoltre, implementata un'ulteriore versione del classificatore associativo che tenesse conto non solo del cambiamento di stato di una stazione verso lo stato "QuasiPiena" (che comprende anche lo stato critico "Piena"), ma anche il passaggio allo stato "Normale" (paragrafo 4.8.7). Questo ulteriore esperimento mi ha permesso di incrementare ulteriormente alcuni risultati di precisione raggiunti in precedenza.

Sono stati, infine, condotti altri esperimenti che però non hanno mostrato dei risultati altrettanto soddisfacenti, come: provare ad incrementare i bassi risultati di richiamo ottenuto (paragrafo 4.8.3); provare ad aumentare l'interpretabilità del classificatore associativo (paragrafo 4.8.4); inserire una soglia di supporto nel filtraggio dei pattern (paragrafo 4.8.8); considerare l'effettivo cambio di stato nella classificazione associativa (paragrafo 4.8.9).

In ultima analisi, dai risultati ottenuti, posso affermare che, nonostante i dati a disposizione non presentassero delle correlazioni sufficientemente elevate in modo da ottenere dei pattern di altissima qualità, il classificatore associativo implementato ha permesso di ottenere dei risultati comunque abbastanza precisi se confrontati con quelli ottenuti tramite un algoritmo di classificazione tradizionale, come l'albero di decisione. D'altra parte, l'albero di decisione ha ottenuto dei risultati migliori per quanto riguarda il richiamo. Tuttavia, nel contesto del bike sharing, avere una precisione elevata permetterebbe di rilevare un numero di casi critici con una sicurezza abbastanza alta, permettendo, per esempio, di mandare l'addetto alla ridistribuzione delle bici, in modo da risolvere delle criticità in maniera mirata, evitando giri a vuoto e risparmiando, così, tempo e denaro. In conclusione, quindi, i risultati ottenuti possono essere reputati abbastanza soddisfacenti.

## 5.2 Lavori futuri

Un possibile sviluppo futuro potrebbe essere quello di utilizzare un dataset che presenta delle correlazioni tra stazioni vicine più significative rispetto alle scarse correlazioni contenute del dataset utilizzato in questo lavoro. Questo permetterebbe di estrarre dei pattern con confidenze più alte e, di conseguenza, ottenere delle predizioni più precise.

Un'altra possibile implementazione potrebbe essere quella di utilizzare, nella classificazione associativa, l'approccio del model ensemble. Questo approccio permetterebbe di combinare, nel processo di decisione, diversi classificatori in parallelo, ognuno dei quali potrebbe utilizzare l'insieme di regole estratte considerando una diversa fascia oraria della giornata. In questo modo, la predizione finale sarebbe determinata in base ad un voto di maggioranza o, in alternativa, ad una media pesata in base alla fascia oraria ritenuta più significativa. Questo permetterebbe anche di avere delle metriche di valutazione più precise rispetto a quelle ottenute mediando sulle diverse fasce orarie.

# Bibliografia


[1] Prof. Paolo Garza. *Introduction to Big Data*. https://dbdmg.polito.it/dbdmg_web/wp-content/uploads/2021/09/01_Intro_BigData_BigData_NewStyle.pdf (cit. a p. 4).

[2] Prof. Elena Baralis. *The data Mining Process*. https://dbdmg.polito.it/dbdmg_web/wp-content/uploads/2021/10/DSTBD_7-DMProcess-IT.pdf (cit. a p. 5).

[3] Prof. Elena Baralis, Prof. Silvia Chiusano. *Association Rules Fundamentals*. https://dbdmg.polito.it/dbdmg_web/wp-content/uploads/2021/10/DSTBD_9-DMassrules.pdf (cit. a p. 7).

[4] Prof. Matteo Golfarelli. *Pattern Sequenziali*. http://bias.csr.unibo.it/golfarelli/DataMining/MaterialeDidattico/2017/12-Pattern%20sequenziali.pdf (cit. a p. 10).

[5] Jian Pei, Jiawei Han, B. Mortazavi-Asl, H. Pinto, Qiming Chen, U. Dayal e Mei-Chun Hsu. «PrefixSpan,: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth». In: *Proceedings 17th International Conference on Data Engineering*. 2001, pp. 215–224. DOI: 10.1109/ICDE.2001.914830 (cit. alle pp. 11, 12).

[6] Prof. Elena Baralis. *Classification fundamentals*. https://dbdmg.polito.it/dbdmg_web/wp-content/uploads/2021/10/DSTBD_10-DMClassification.pdf (cit. alle pp. 12, 15).

[7] Ben Hamner. *SF Bay Area Bike Share*. https://www.kaggle.com/benhamner/sf-bay-area-bike-share (cit. alle pp. 19, 33).

[8] Martina Toma. *Pattern Extraction*. https://github.com/MartinaToma/Tesi_TomaMartina_Finale (cit. a p. 23).