POLITECNICO DI TORINO

Master degree course in DATA SCIENCE AND ENGINEERING

Master Degree Thesis

Machine and Deep Learning for Disease Mapping

Supervisors
prof. Daniele Jahier Pagliari
prof. Paola Berchialla[1]

Candidates
Luca Viada

[1] Dept. of Clinical and Biological Sciences, University of Torino

Anno accademico 2023-2024

Acknowledgements

I would like to extend special thanks to Professor Daniele Jahier Pagliari for guiding and supporting me throughout this project. I am grateful to the TrustAlert project for providing me with the opportunity to develop this thesis, supplying the data and their expertise in the medical domain. In particular I would like to thank Professor Paola Berchialla for her guidance, as well as the support from Veronica Sciannameo and Alessia Visconti.

A heartfelt thank you also goes to my family for their unwavering support over the years, and to my friends who have stood by me.

Summary

The use of Machine Learning and Deep Learning in the medical field has revolutionized the management and analysis of healthcare data, with particular focus on Electronic Health Records (EHRs). In this context, the TrustAlert [1] project aims to create an integrated platform for analyzing patient medical trajectories with the goal, for instance, to identify clusters of patients with similar characteristics. This thesis, conducted within the framework of TrustAlert, and in collaboration with the Department of Clinical and Biological Sciences, Università degli Studi di Torino focuses on the analysis of data derived from structured health databases of a local health authority in Piedmont (ASL Cuneo 2), such as hospital discharge records, outpatient visits, and emergency room visits. To achieve this goal, Machine Learning and Deep Learning techniques have been used to analyze administrative healthcare data, including diagnosis and procedure codes, to identify relations and patterns between events. In particular, the core work of the thesis has focused on using a Convolutional Autoencoder (ConvAE), trained in a self-supervised way, for this task. This model can extract meaningful latent representations, and could be used as an informational basis for resource allocation and priority setting in healthcare.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, the utilization of Machine Learning (ML) and Deep Learning (DL) in the medical field has revolutionized the management of healthcare data, particularly concerning Electronic Health Records (EHRs). With the ability to efficiently analyze vast amounts of data and extract complex insights, these advanced algorithms have created new possibilities for improving healthcare quality and optimizing clinical processes, both in the technical and management domains.

For example, an application of ML and DL in electronic patient records is the prediction of disease risk and complications[3]. By analyzing data within EHRs, it's possible to identify patients at risk of developing specific conditions or adverse events. For instance, predictive models can be trained to forecast the risk of stroke, diabetes, heart failure, and other medical conditions, enabling healthcare providers to intervene early and personalize treatment plans based on each patient's specific needs. This approach not only enhances healthcare quality but can also help reduce costs associated with managing chronic illnesses through targeted preventive interventions.

Not only strictly related to EHR analysis, another application area is computer-aided diagnosis, leveraging DL to analyze medical images and automatically detect anomalies or lesions. Advances in deep neural network learning capabilities have led to significant improvements in diagnostic accuracy across various medical specialties, including radiology, dermatology, and pathology. For example, DL algorithms have been developed to diagnose skin cancer with a level of accuracy comparable to that of expert dermatologists, analyzing dermoscopic images and photographs of moles[4].

Despite the numerous advantages offered by ML and DL in analyzing electronic patient records, there are also challenges to address. Data privacy and security, for instance, are fundamental concerns as ML and DL algorithms require access to sensitive patient information. Also, the interpretation of models and transparency in decision-making processes is crucial to ensure the trust of clinicians and patients in adopting these technologies[7].

Other approaches can be the use of ML and DL algorithms to process large amounts of EHR data and produce visual representations of patients in a multidimensional space. These representations enable the identification of clusters of patients with similar or shared characteristics, which can be used for classification, prediction, or population segmentation purposes.

These approaches can be based on different types of Neural Networks, like Recurrent Neural Networks or Convolutional Autoencoders as in this thesis. An example is *Predicting Clinical Events via Recurrent Neural Networks* [8] which has been developed to transform EHR data into image-based representations, enabling the automatic identification of clusters of patients with similar diseases or comorbidities. In order to achieve this point another main role useful to reach the scope are the dimensionality reduction techniques, such as t-Distributed Stochastic Neighbor Embedding (t-SNE), to visualize EHR data in a two-dimensional space, facilitating the visual identification of patient clusters[11].

These developments represent a significant step forward in healthcare data analysis and the identification of patient subgroups with similar characteristics. Challenges remain, such as data standardization, patient privacy, and interpretation of the obtained results. Nevertheless, the use of ML and DL for EHR data analysis continues to promote personalized and evidence-based medicine, opening up new avenues for research and clinical practice.

EHR adoption in Italy

In Italy, the adoption of Electronic Health Records (EHRs) is steadily increasing within the healthcare sector, with numerous hospitals, clinics, and public and private healthcare facilities implementing computerized systems for managing healthcare information. According to the *Fondazione Ugo Bordoni*, a prominent research organization, recent data indicates that a significant number of Italian healthcare facilities have adopted at least partially EHR systems, driving towards improved efficiency and quality of healthcare delivery[12].

EHRs can offer several advantages in the Italian healthcare context. These include the potential to decrease medical errors, facilitate seamless information exchange among healthcare providers, enhance care coordination, and streamline administrative procedures. These benefits were underscored by the Ministry of Health in their publication[13]. The Ministry has then outlined a Strategic Plan for Digital Health, focused on the adoption and development of computerized systems to enhance the efficiency of the healthcare system and ensure quality healthcare delivery[16].

The full implementation of EHRs is not without challenges. AGENAS, the National Agency for Regional Health Services, underscores the need to address obstacles such as data format standardization, privacy and security of healthcare information, as well as resistance to change among healthcare providers and the significant financial investments required for acquiring and maintaining computerized systems[14].

In order to regulate the use of Electronic Health Records, specific regulations and guidelines have been introduced in our country. The Italian Data Protection Authority has issued guidelines for healthcare data management and privacy protection, in alignment with the Personal Data Protection Code and the General Data Protection Regulation (GDPR) of the European Union, to ensure the security and privacy of healthcare information stored and transmitted through computerized systems[15].

For example some hospitals here in Piedmont are using nowadays J-His [2], which is an advanced computer system utilized in hospitals across the Piemonte region to efficiently and securely manage patients' health information. Through J-His, medical staff can quickly access patients' test results and medical records, enabling more timely and effective diagnosis and treatment, access to the emergency department, recording who accesses it and for what purpose. J-His can also facilitate the consultation of radiology reports, providing immediate and organized access to critical diagnostic data.

In conclusion, focusing in this thesis, we aim to test the ability of a model taken from literature, trained on EHRs used in America, to work on some first examples of EHRs available here in Italy, adapting them and trying to observe which results can be reached as firsts steps. This work can be a starting point for future possible works enabling society improvements in this domain.

Chapter 2

Background

2.1 Machine Learning and Deep Learning

Machine learning emerges as a crucial component within the field of AI, focusing on the principle of instructing computers to learn progressively from observed data, resembling the intricate mechanisms of human learning. By acquiring knowledge through this process, computers gain the ability to generalize effectively to new inputs. What sets Machine Learning apart from traditional software is its reliance not solely on predetermined instructions crafted by developers to perform tasks. Consider, for example, the task of identifying different fruits from images: rather than manually coding the software to recognize individual fruits, machine learning models develop this capability by exposure to extensive datasets containing labeled fruit images. Over time, they enhance their proficiency in accurately categorizing each image. The key to success in machine learning is therefore often having a lot of data.

Since computers were first created, there has been a desire to make them perform and learn like humans. This goal has fascinated researchers, software developers, and engineers. Machine learning now shines as the guiding light toward achieving this ambitious aim. Its impact is felt across various fields, seamlessly becoming a part of our daily lives, from the functionalities of our smartphones and websites to the displays of train departure schedules. In the following chapters, we explore briefly the main concepts on which this thesis relies giving the basic concepts useful to understand how the work has been conducted.

2.1.1 Importance of Data

The core elements which feed ML algorithms are data, where of course quality and quantity influence the performance.

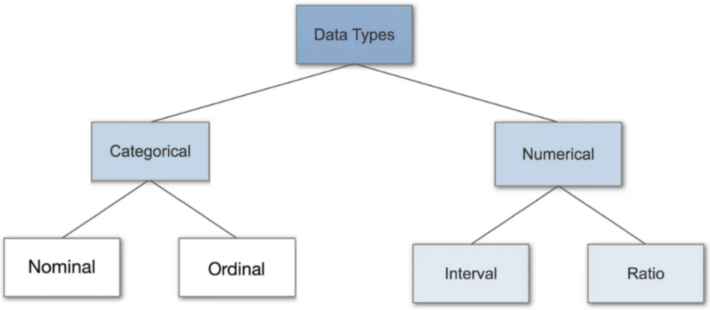


Figure 2.1: Data categories [5]

As shown in Figure 2.1 generally data can be categorized in the following macro-categories:

- *Numerical data*: this type of data, often referred to as quantitative data, encompasses measurable values such as prices, measurements, and phone numbers. It can be further classified into *Ratio Data*, where zero indicates absence and equal intervals have meaning, and *Interval Data*, with meaningful intervals but no true zero point. Unlike other data types, numerical data is not bound to any specific point in time, it comprises raw numeric values.
- *Categorical data*: this category involves data organized based on characteristics or attributes, such as gender, ethnicity, postal codes, and workplace. Unlike numerical data, categorical data is non-numerical in nature. It is commonly used to group similar data together, such as individuals sharing certain qualitative properties. Typically, the variable associated with categorical data remains fixed. Here can be further described as *Nominal Data*, without any inherent order, and *Ordinal Data*, with a meaningful order or ranking.
- *Time series data*: this type of data consists of data points indexed at specific time intervals. Due to the nature of its collection over time, time series data may exhibit no-uniformity, indicating variations in data collection intervals. However, they possess clear starting and ending points, setting them apart from numerical data.

- *Text data*: textual data comprises words, sentences, or paragraphs from which Machine Learning models extract features and insights. Since Machine Learning algorithms operate using mathematical functions, text data needs to be transformed into vectors before being utilized by these models.

2.1.2 Feature engineering

Exploring datasets presents various hurdles, given that not all data holds significance or usefulness. Success depends on utilizing data that is not only extensive but also reflective of the specific scenario being generalized. Consequently, the precision and authenticity of the training data become essential.

It has been observed and demonstrated that the success of a ML project greatly depends on identifying effective features. This necessitates the employment of *feature engineering*, a critical and intricate facet of ML projects. At its core, feature engineering entails the transformation of raw data into features that efficiently capture and represent the more relevant information of the underlying problem at hand.

In general feature engineering encompasses two fundamental processes:

- Feature selection: discerning and retaining the most pertinent features within the training dataset while discarding extraneous ones that offer little relevance to the problem under consideration.

- Feature extraction: when existing features fall short, this process involves amalgamating or manipulating existing features to craft novel ones that better align with the objectives of a specific ML task enhancing its effectiveness and predictive capabilities.

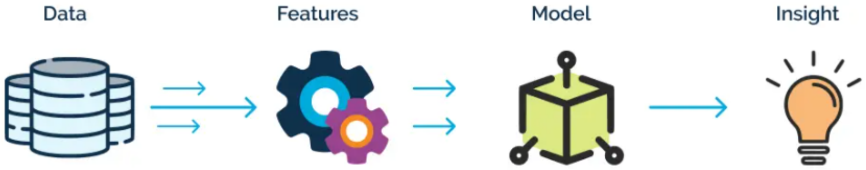


Figure 2.2: ML pipeline [6]

2.2 Sequence Models

Sequence models in machine learning (ML) and deep learning (DL) specialize in handling sequential data, where element order is crucial unlike traditional models assuming input independence. They capture dependencies and patterns within sequences. They excel in tasks like natural language processing (NLP), time series analysis, and speech recognition.

Recurrent Neural Networks (RNNs) [17] are common sequence models, iterating through updates of hidden states based on inputs to retain memory of past inputs. Variants such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) address issues like vanishing gradients and long-range dependencies. Additionally, other common architectures used in sequence modeling include transformers, which are proficient at capturing global dependencies, and temporal Convolutional Neural Networks (CNNs), effective for local feature extraction and capturing temporal patterns.

Some scenarios where sequence models prove invaluable:

- Speech Recognition: given an input audio clip (X), the task is to map it to a text transcript (Y). Here, both the input and output are sequential, as the audio clip unfolds over time (X), and the transcript comprises a sequence of words (Y) [19]. Recurrent neural networks, along with their variants, have been instrumental in advancing speech recognition technologies [20].

- Music Generation: while the input (X) can vary, the output (Y) is consistently a sequence. For instance, the input might be empty or a single integer denoting the desired music genre, while the output entails a sequence of musical notes [21].

- Sentiment Classification: here the input (X) constitutes a sequence, typically a phrase, and the goal is to predict the sentiment conveyed. This illustrates another realm where sequence models excel [22].

- DNA Sequence Analysis: where DNA is represented by nucleotides A, C, G, and T, sequence models aid in identifying patterns and labeling DNA segments, such as those corresponding to proteins [23].

- Machine Translation: given an input sentence in one language, the objective is to generate its translation in another language, a task inherently reliant on sequential processing [24].

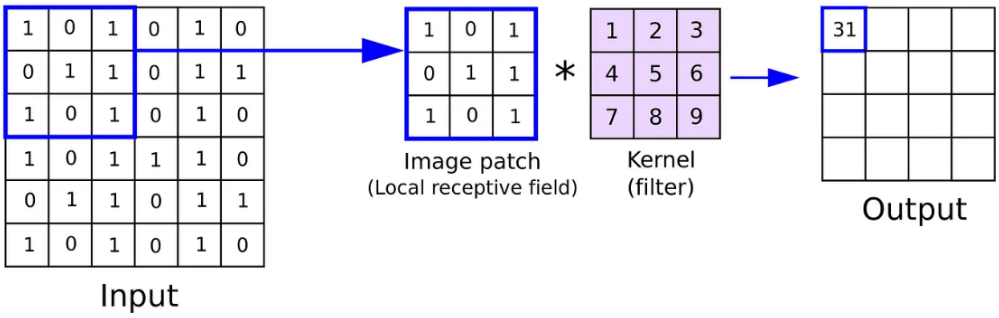
- Video Activity Recognition: here a sequence of video frames is provided, and the goal is to recognize the underlying activity depicted [25].
- Named Entity Recognition: similarly a sentence is given, and the task is to identify and label specific entities, such as people[26].

These examples highlight the versatility of sequence models across various domains, where they usually operate in Supervised Learning settings with labeled data (X, Y) [27]. It's noteworthy that sequence problems can manifest in different forms, where both input and output could be sequences, or only one of them. Additionally, lengths of X and Y may vary, adding complexity to the modeling task [28].

2.3 Convolutional Neural Network

In this section, we will focus on the knowledge and principles underlying Convolutional Neural Networks (CNNs) that form a crucial component of the chosen architecture for our task.

CNN is a powerful class of deep learning models widely used for various tasks in computer vision, natural language processing, and beyond. They are a class of Deep Learning models designed for processing structured grid-like data, such as images and videos, or like in our case to model sequential data.



1	0	1	0	1	0
0	1	1	0	1	1
1	0	1	0	1	0
1	0	1	1	1	0
0	1	1	0	1	1
1	0	1	0	1	0

1	0	1
0	1	1
1	0	1

1	2	3
4	5	6
7	8	9

31			

Figure 2.3: Kernel steps example [9]

2.3.1 Architecture

CNNs architecture include multiple types of layer, each serving a specific purpose in feature extraction and classification. The typical architecture of a CNN includes:

- **Convolutional Layers**: these layers apply convolution operations to the input data, extracting features through filters or kernels enabling the network to capture spatial hierarchies and patterns present in the input.

- **Activation Functions**: they introduce non-linearity into the network, allowing CNNs to learn complex relationships within the data. Popular activation functions used in CNNs include ReLU (Rectified Linear Unit), Sigmoid, and Tanh.

- **Pooling Layers**: it reduces the spatial dimensions of the feature maps generated by convolutional layers, aiding in feature selection and enhancing computational efficiency. Common pooling operations include max pooling and average pooling.

- **Fully Connected Layers**: they integrate the extracted features and perform classification or regression tasks. These layers connect every neuron in one layer to every neuron in the subsequent layer, enabling high-level feature learning.

In its architecture, CNNs incorporate various layers, encompassing an input layer, hidden layers, and an output layer. These layers function collaboratively to extract meaningful features from the input data.

The hidden layers play a pivotal role, incorporating one or more layers dedicated to convolutions. These Convolutional Layers essentially perform mathematical operations, such as dot products, between a convolution kernel and the input data matrix often utilizing the Frobenius inner product. In details given two matrices **A** and **B** as

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1M} \\ a_{21} & a_{22} & \dots & a_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NM} \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1M} \\ b_{21} & b_{22} & \dots & b_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ b_{N1} & b_{N2} & \dots & b_{NM} \end{bmatrix}$$

the computation of Forbenius inner product is defined as

$$\langle A, B \rangle = \sum_{i=1}^{N} \sum_{j=1}^{M} a_{ij} b_{ij}$$

2.3.2 Activation funcions

The aim of these functions is to inject non-linearity into the neuron's output, enabling an artificial Neural Network to adeptly handle intricate tasks. By assessing the relevance of the data received, the output of these functions determines the degree of influence on the subsequent neuron.

For this thesis the CNN use the Rectified Linear Unit function (ReLU) enhancing the network's capacity to learn complex patterns. In detail, the ReLU has the following structure:

$$\mathrm{ReLU}(x) = \frac{x + |x|}{2} = \begin{cases} 0 & \text{se } x < 0 \\ x & \text{se } x \geq 0 \end{cases}$$

where x is the input to a neuron. It promotes sparse activation, efficiently handles gradient propagation, and requires only simple computations. Moreover, it remains scale-invariant, enhancing network stability and robustness across diverse datasets and tasks.

Other variations of these functions that are widely used instead of Relu can be Tanh and Sigmond that can be observed in Figure 2.4.

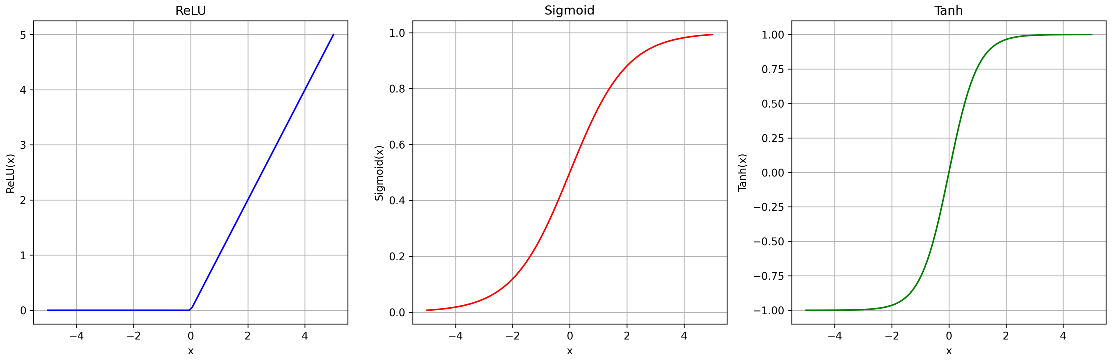


Figure 2.4: ReLU, Sigmoid, Tanh activation functions plot

2.3.3 Kernel

Another important component in CNN is the kernel 2.3, that is nothing but a filter that is used to extract the features from the data.

As the convolution kernel traverses across the input data matrix, the convolution operation generates what is known as a feature map, essentially a representation highlighting significant features within the data. These feature maps serve as inputs for subsequent layers, promoting hierarchical feature extraction and abstraction. Besides convolutional layers, CNN architectures frequently include other crucial components such as pooling layers, fully connected layers, and normalization layers. These elements contribute to the network's ability to learn hierarchical representations of data.

2.4 Word Embeddings

Another concept useful for our aims and on which the principal scope for this thesis is Word embeddings. By definition from the NLP domain, Word Embeddings are dense vector representations of words in a high-dimensional space, where each word is mapped to a numerical vector. These embeddings capture semantic and syntactic similarities between words, enabling Machines to understand and process Natural Language more effectively. The concept of word embeddings has revolutionized NLP tasks, such as language modeling, sentiment analysis, and machine translation.

Initially, these embeddings consist of random floating-point values, which later become trainable parameters. Like the weights in a dense layer, these values are adjusted during training to capture the nuances of word relationships. The dimensions of these embeddings play a crucial role in delineating the vocabulary's intricacies. With higher-dimensional embeddings, finer-grained relationships between words can be captured, making them suitable for larger datasets. Conversely, smaller dimensions are often preferred for smaller datasets.

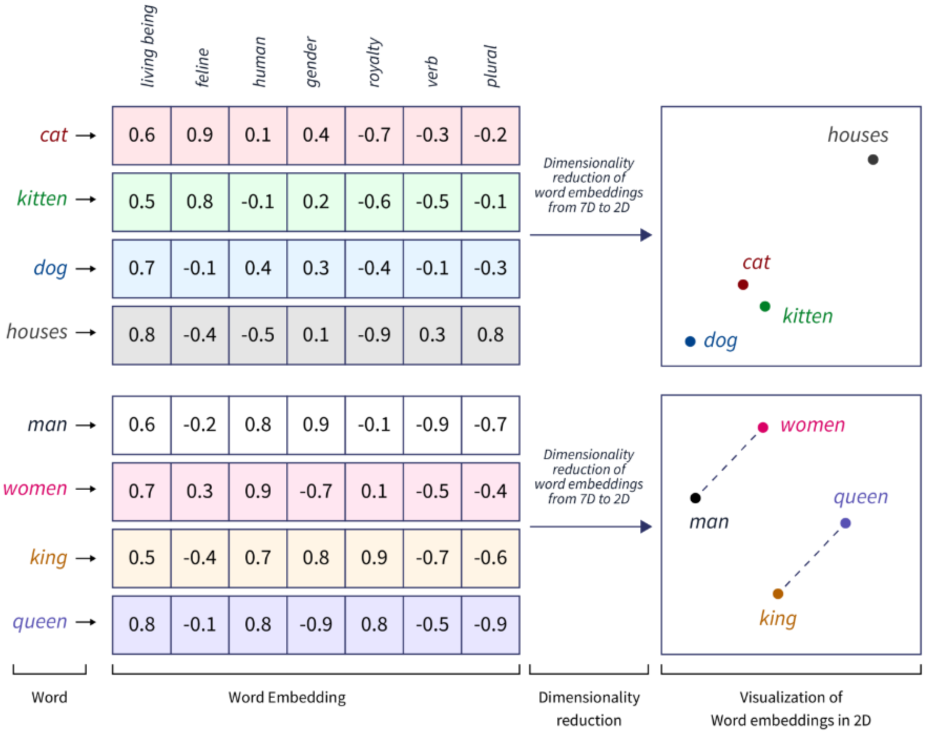
During training, each word in the vocabulary is encoded by referencing the corresponding dense vector in a lookup table. This process renders word embeddings as dynamic representations, continually refined through training iterations. Thus, they are commonly referred to as a "lookup table" in the realm of Natural Language Processing.

A graphic representation of this process can be observed in Figure 2.5.

2.5 From machine representation to human visualization

2.5.1 Dimensionality Reduction

After the conversion to machine-readable information to process the data using the powerful tools and concepts described, we need, at the end of the pipeline, to make them human-readable and understandable, while also considering dimensionality reduction, a crucial technique in Machine Learning and data analysis. This technique aims to reduce the number of features or dimensions in a dataset while preserving its essential information. This process enables more efficient computation, visualization, and interpretation of high-dimensional data. In this section, we introduce dimensionality reduction techniques specifically tailored for representing data in two dimensions, followed by clustering analysis for exploring patterns within the reduced-dimensional space.



	living being	feline	human	gender	royalty	verb	plural
cat →	0.6	0.9	0.1	0.4	-0.7	-0.3	-0.2
kitten →	0.5	0.8	-0.1	0.2	-0.6	-0.5	-0.1
dog →	0.7	-0.1	0.4	0.3	-0.4	-0.1	-0.3
houses →	0.8	-0.4	-0.5	0.1	-0.9	0.3	0.8
man →	0.6	-0.2	0.8	0.9	-0.1	-0.9	-0.7
women →	0.7	0.3	0.9	-0.7	0.1	-0.5	-0.4
king →	0.5	-0.4	0.7	0.8	0.9	-0.7	-0.6
queen →	0.8	-0.1	0.8	-0.9	0.8	-0.5	-0.9

Dimensionality reduction of word embeddings from 7D to 2D
houses
cat
kitten
dog
Dimensionality reduction of word embeddings from 7D to 2D
women
man
queen
king
Word
Word Embedding
Dimensionality reduction
Visualization of Word embeddings in 2D

Figure 2.5: Word Embedding and Dimensionality reduction pipeline example [10]

In the literature, the two algorithm considered two reach this scope are:

- Principal Component Analysis (PCA): PCA is one of the most widely used dimensionality reduction techniques. It identifies the principal components of variation in the data and projects it onto a lower-dimensional subspace while retaining as much variance as possible. PCA simplifies data representation into two dimensions through the use of the first two principal components. This simplification aids in visualizing and exploring patterns within the data.

- t-Distributed Stochastic Neighbor Embedding (t-SNE): t-SNE is a non-linear dimensionality reduction technique particularly suited for visualization purposes. It preserves local structures and clusters of data points in the original high-dimensional space while mapping them to a lower-dimensional space, typically two or three dimensions. t-SNE is effective in revealing the underlying structure and relationships between data points, making it suitable for exploratory data analysis and visualization.

2.5.2 Visualization

Once the high-dimensional data has been effectively represented in two dimensions using dimensionality reduction techniques, clustering analysis can be performed to identify inherent patterns or groups within the data.

Clustering algorithms are essential tools in data analysis, used to identify inherent structures or patterns within datasets. Among the commonly employed clustering algorithms is K-means.

The algorithm used in this thesis is K-means which is a popular clustering algorithm that partitions data into K clusters based on similarity. Here's a how it works:

- **Initialization**: begin by randomly selecting K centroids, which represent the centers of the clusters

- **Assignment**: assign each data point to the nearest centroid, forming K clusters

- **Update centroids**: Recalculate the centroids as the mean of the data points within each cluster

- **Repeat**: iterate the assignment and centroid update steps until convergence, where the centroids no longer change significantly or a specified number of iterations is reached

- **Convergence**: K-means aims to minimize the sum of squared distances between data points and their respective centroids

K-means is computationally efficient and works well for datasets with distinct, well-separated clusters. However, it requires specifying the number of clusters (K) a priori and may converge to local optima depending on the initial centroids [29].

Chapter 3

Related works

In this chapter, we will provide summaries of three papers that serve as fundamental texts to acquaint readers with the thesis domain. These summaries offer an accessible overview of the main concepts and insights discussed in each paper, providing readers with a thorough understanding of the scholarly landscape that informs our research.

3.1 "BEHRT"

Starting from the first paper titled "BeHRt: transformer for electronic Health Records"[30], it introduces a novel deep neural network model called BEHRT, a model based on BERT architecture but fine tuned for the analysis of electronic health records (EHR). The model aims to predict future health conditions of patients, leveraging deep learning techniques to learn representations from raw or minimally-processed data. BEHRT is based on a Transformer-based architecture, pre-trained using a masked language model (MLM) task, and then fine-tuned for downstream tasks such as disease prediction.

The document highlights the importance of deep learning in capturing complex relationships within EHR data and compares BEHRT's performance with other existing models such as Deepr and RETAIN. It demonstrates BEHRT's superior predictive power in predicting a wide range of diseases, outperforming other models by more than 8% in terms of absolute improvement. The model's architecture is designed to capture various modalities of EHR data, including diseases, age, segment, and position, allowing for personalized and interpretable risk prediction.

Furthermore, the document discusses the visual investigation of disease embeddings, showing how BEHRT's representations form distinguishable patterns and natural stratifications, such as gender-specific diseases. It also emphasizes the model's self-attention mechanism, which enables it to uncover complex relationships among diagnoses beyond temporal adjacency.

It concludes by highlighting BEHRT's ability to understand latent characteristics of diseases and its potential for personalized risk prediction and disease phenomapping, presenting BEHRT as a promising model for personalized risk prediction and disease analysis in electronic health records, showcasing its superior performance and interpretability compared to existing models.

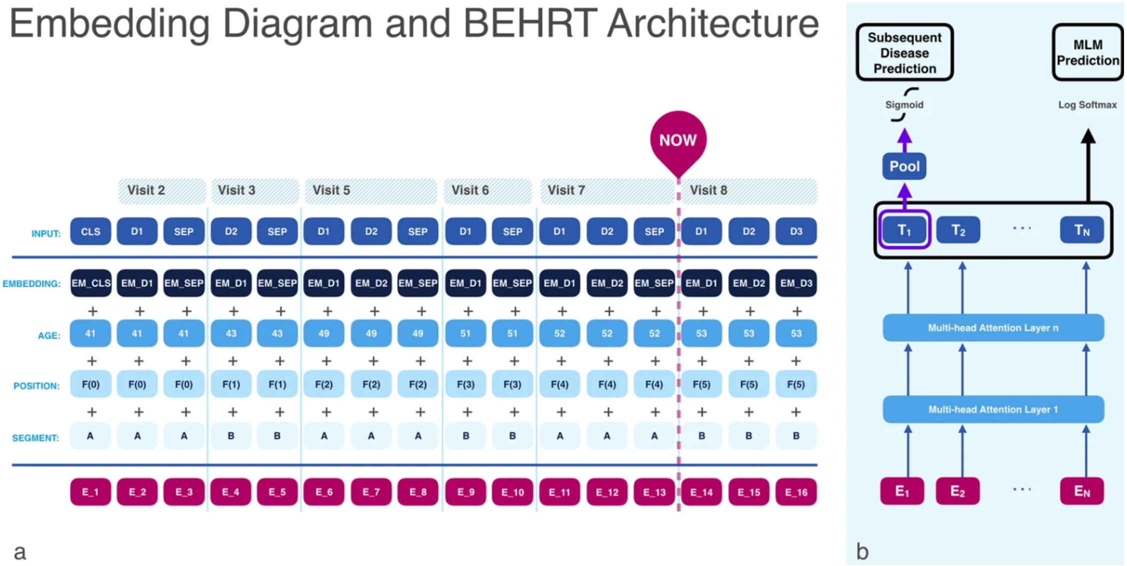

Embedding Diagram and BEHRT Architecture
NOW
Visit 2
Visit 3
Visit 5
Visit 6
Visit 7
Visit 8
INPUT:
CLS D1 SEP D2 SEP D1 D2 SEP D1 SEP D1 D2 SEP D1 D2 D3
EMBEDDING:
EM_CLS EM_D1 EM_SEP EM_D1 EM_SEP EM_D1 EM_D2 EM_SEP EM_D1 EM_SEP EM_D1 EM_D2 EM_SEP EM_D1 EM_D2 EM_D3
AGE:
41 41 41 43 43 49 49 49 51 51 52 52 52 53 53 53
POSITION:
F(0) F(0) F(0) F(1) F(1) F(2) F(2) F(2) F(3) F(3) F(4) F(4) F(4) F(5) F(5) F(5)
SEGMENT:
A A A B B A A A B B A A A B B B
E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_10 E_11 E_12 E_13 E_14 E_15 E_16
a
Subsequent Disease Prediction
MLM Prediction
Sigmoid
Log Softmax
Pool
T1 T2 ... TN
Multi-head Attention Layer n
Multi-head Attention Layer 1
E1 E2 ... EN
b

Figure 3.1: BEHRT architecture [30]

3.2 "Med-BERT"

The second paper considered in the literature, "Med-BERT: pretrained contextualized embeddings on large scale structured electronic health records for disease prediction" [31], discusses the development and evaluation of Med-BERT, a contextualized embedding model pre-trained on structured electronic health record (EHR) data, and its application in disease prediction tasks. Med-BERT is designed to capture the complex semantics of inputs and inject knowledge into new tasks, showing significant improvements in predicting diseases such as diabetes and pancreatic cancer. The model utilizes a bidirectional transformer and deep structure, and it has been shown to be extremely helpful when transferring to new tasks.

The study introduces the design of the Med-BERT embedding layers, including code embeddings, serialization embeddings, and visit embeddings, to accommodate the structured EHR modality. Unlike BERT, Med-BERT does not use specific special tokens for starting or separating sentences at the input layer due to differences in the input formats of EHR and text. The study also discusses the pretraining of Med-BERT using optimization algorithms and recommended hyperparameters.

Furthermore, the document highlights the potential of Med-BERT in improving prediction performance of baseline deep learning models on different sizes of training samples, reduce the weight of data labelling, and enabling the training of powerful deep learning predictive models with limited training sets. The study also evaluates Med-BERT in disease prediction tasks, demonstrating its effectiveness in fine-tuning on small sample sizes and its generalization across different datasets.

In conclusion, the document establishes the feasibility and usefulness of contextualized embedding of structured EHR data and demonstrates the potential of Med-BERT in improving disease prediction performance and reducing data acquisition costs. The study also outlines future research directions, including the inclusion of other sources such as time, medications, procedures, and laboratory tests as inputs for Med-BERT, and the exploration of task-specific visualizations and interpretations.

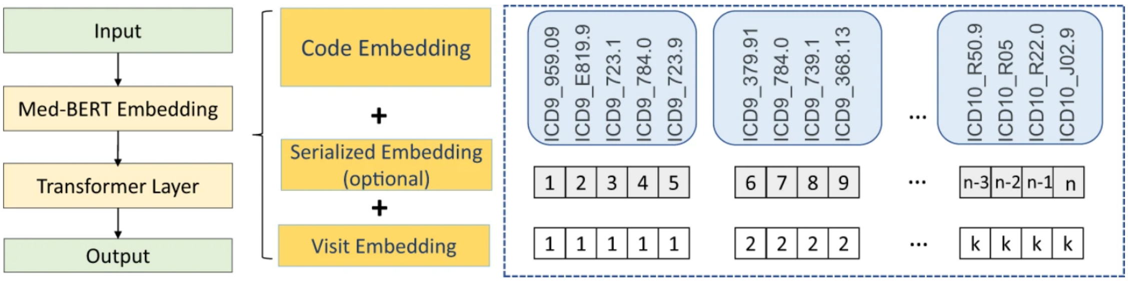


Figure 3.2: Med-BERT architecture [31]

3.3 Convolutional Autoencoder (ConvAE)

The last paper considered, "Deep representation learning of electronic health records to unlock patient stratification at scale"[32], discusses the use of deep learning to derive patient representations from electronic health records (EHRs) and enable patient stratification at scale. The authors propose an unsupervised framework based on Deep Learning, specifically a model called ConvAE, to process heterogeneous EHRs and derive patient representations for efficient patient stratification. The study used de-identified EHRs from the Mount Sinai Health System data warehouse, comprising approximately 4.2 million patients and spanning the years from 1980 to 2016. The dataset included general demographic details, clinical descriptors, diagnosis codes, medications,procedure codes, vital signs, lab tests, and preprocessed clinical notes.

The ConvAE model, which combines word embeddings, convolutional neural networks (CNN), and autoencoders, was evaluated on a diverse cohort of patients and outperformed several baselines in a clustering task to identify patients with different complex conditions. The results showed that ConvAE can generate patient representations that lead to clinically meaningful insights and enable patient stratification at scale. The study also highlighted the potential for ConvAE to identify various clinically relevant subtypes for different disorders, including type 2 diabetes, Parkinson's disease, and Alzheimer's disease.

The authors acknowledged several limitations of the study, including the potential for noise and biases in the data, the inclusion of false positives in patient cohorts, and the need for better disease subtyping algorithms. They also outlined future directions for research, including enabling multilevel clustering, replicating the study on EHRs from different healthcare institutions, and evaluating the use of disease subtypes as labels for training supervised models.

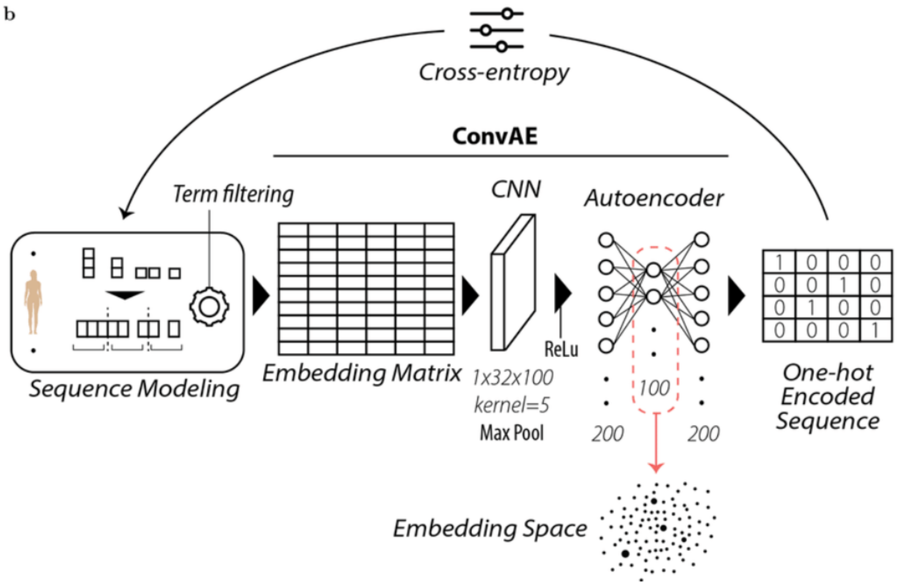


Figure 3.3: ConvAE architecture [32]

Chapter 4

Materials and methods

In this chapter each step of the work are presented, starting from the datasets details, with a detailed description of each feature available, moving then to the preprocessing steps, next to the description of the model architecture and finally to the evaluation of the results achieved.

4.1 Data Analysis

As mentioned in previous sections, the data have been made available by the TrustAlert research project. In detailed preliminary steps, we obtained different datasets describing the administrative record of ASL Cuneo 2, in particular, we had different time window slots with a four-month dimension covering the period from the first quarter of 2020 until the second one of 2023.

For privacy purposes, the data were fully anonymized and were also grouped by a key value identifying some macro geographical references within the province of Cuneo and Turin.

In Figure 4.1 we can observe the grouping keys used on the source dataset for the extractions. Focusing on each time slot it can be observed that some datasets were poor of information or unbalanced. Some extractions probably were affected by some error in the grouping conditions and also some file was corrupted in their structure after the import in the environment where the model has been tested.

Under this condition a *data exploration* and *analysis* has been conducted to find a dataset that could be capable of presenting the most unbalanced representation of the population of patients, covering also most of the possible cases of events for the medical interaction domain.

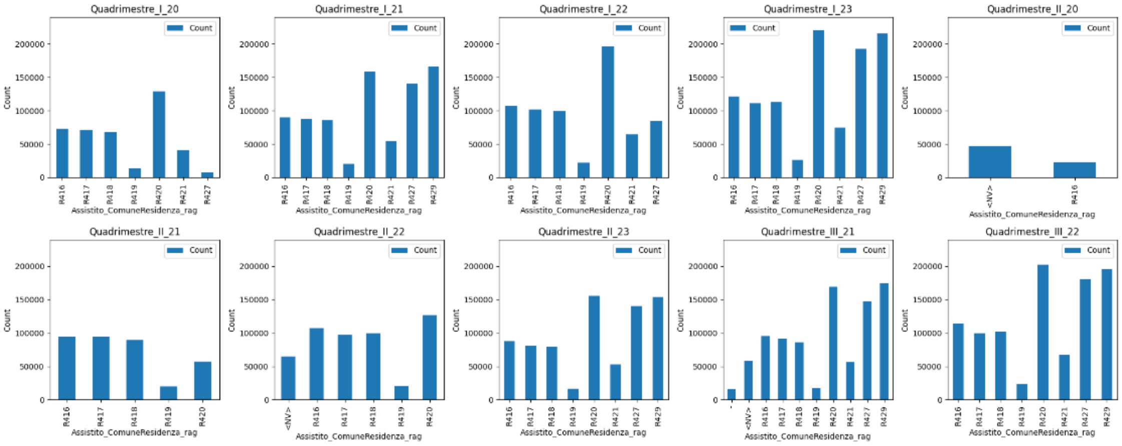


Figure 4.1: Distribution Aggregation Code Province Reference full cohort of data

Moving deeply in detail, each dataset analyzed presents the same structure in terms of columns and the population of each dataset varies from 70k patients, this was the case where some problem corrupted the extraction, to the bigger one of size over 1 million which is the one chosen for training the model.

Each row of the dataset is composed of different features characterizing one single event related to a particular patient, in details we have:

- *Assistito_ComuneResidenza_rag*: this column is the one mentioned which has been using from source dataset to aggregate the data in the macro province zone references related to the patient residence. It has an alphanumeric structure, which is a code used in the process of anonymization of data, and in the selected dataset it assumes all the possible values available in the full cohort of data. In particular the possible values of this column are:

$$\{\mathrm{R16}, \mathrm{R17}, \mathrm{R18}, \mathrm{R19}, \mathrm{R20}, \mathrm{R21}, \mathrm{R27}, \mathrm{R29}\}$$

  without null value and the distribution of each value can be observed in Figure 4.1;

- *Assistito_GruppoEta*: this feature aims to describe the age of each patent. It doesn't give the exact age of each patient but it's divided into non-overlapping 10-year ranges. For example, each interval represents a separate age group and does not overlap with others, allowing for clear and organized categorization of data based on patients' age. In the following Figure 2.1 we can observe the distribution in the cohort of data, which can be observed that the trends in the distribution of each period are comparable.

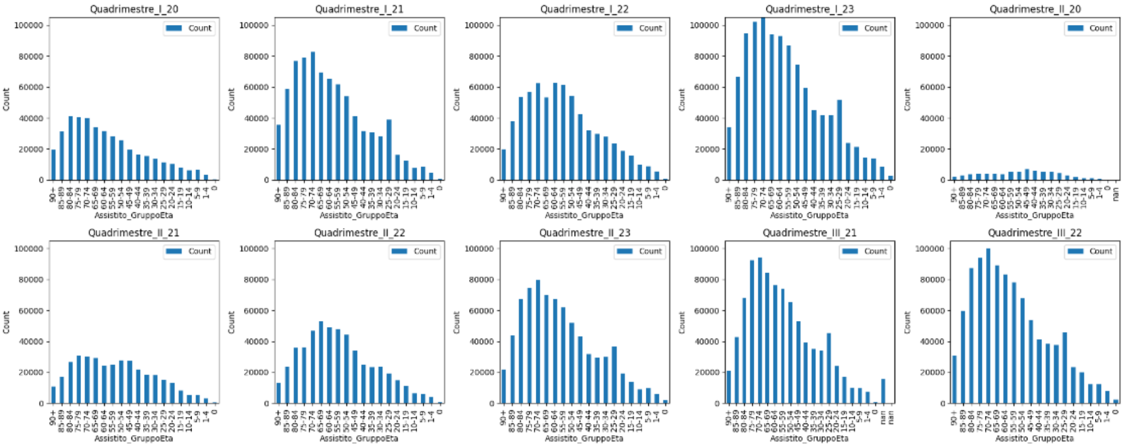


Figure 4.2: Age range distribution full cohort of data

- *Assistito_CodiceFiscale_Criptato*: here it contains securely encrypted Fiscal Code assigned to each patient. This encryption process is essential to safeguarding privacy and ensuring compliance with data protection regulations. By encrypting sensitive information such as Fiscal Code, the database ensures that personal identifiers are protected from unauthorized access or disclosure. This measure enhances data security and privacy, reassuring patients that their personal information is handled with the utmost care and confidentiality. Some example of values are

Table 4.1: Sample Fiscal Code encrypted

Indice	**Codice fiscale criptato**
1	*F*@\]5*E*”(”*G*‘)”8*F*, .*T*2*O*/\#
2	*MC*, @@ : (*L*82*U*131*J* > *V*&*K*−!0
3	*E* = *C* + 1*E*”3*U* > *W*8, (7′*NTA**E*′

- *Struttura*: in this column, we can find numeric codes each associated with the name of a hospital facility or ASL (Local Health Authority). These codes represent the location where the specific healthcare service is provided, enabling clear and organized tracking of healthcare activities in relation to the hosting facility. Here is a little sample of possible values:

Table 4.2: Sample Code Strucure

Code	Structure Name
001254	Ospedale Michele Ferrero E Pietro Ferrero - Industriali - ()
01003800	Ospedale Michele E Pietro Ferrero - (Verduno)
000398	Az. Ospedal. S. Croce E Carle - ()
500073	Casa Di Cura Citta' Di Bra - ()

- *Specialità*: this column in the dataset describes the medical specialty associated with each healthcare event for the patient. This field provides information about the specific area of expertise or focus of the healthcare professional or department involved in the event. The "Specialty" column allows for categorizing healthcare events based on the medical specialty involved, facilitating analysis and management of patient care across different medical disciplines. Here is a little sample of possible values:

Table 4.3: Sample Specialty Codes

Code	Specialty Name
76	Neurochirurgia Pediatrica
33	Neuropsichiatria Infantile
56	Medicina Fisica E Riabilitazione
78	Urologia Pediatrica
36	Ortopedia E Traumatologia

- *Data*: this field gives a time reference for healthcare events related to the patient based on the chronological sequence in which they occurred. This allows for defining a trajectory of care for each patient, providing a chronological overview of medical activities and treatments received over time. Utilizing this field enables healthcare providers to assess the progression of the patient's health, identify correlations between health–

care events and treatment responses, as well as plan future interventions in a timely and personalized manner.

- *Tipo*: The 'Type' field provides a high-level description of the event associated with a facility. This field serves as a categorical identifier for different types of healthcare events.

Table 4.4: Type column possible value

Value	**Description**
Ambulatoriale	Outpatient services provided at a medical facility
ProntoSoccorso D	Emergency room visits
ProntoSoccorso R	Emergency room visits
Farmaci D	Dispensation of medications
Farmaci P	Prescription of medications
Farmaci S	Medication administration in healthcare
Assistenza domiciliare	Healthcare services provided within patients' homes

- *Evento*: here are described 3 different columns related to the events description, the columns represent related healthcare events, and each detailed through three categories: *Evento*, *Evento_diagnosi_principale*, and *Evento_intervento_principale*. These columns are populated based on the medical procedure performed. If a procedure only requires recording the event, then only the *Evento* column is populated. If a main diagnosis related to the event is later identified, it is recorded in the *Evento_diagnosi_principale* column. If, instead, an intervention related to the event is performed, it is recorded in the *Evento_intervento_principale* column. In some cases, all three columns may be populated, providing a comprehensive overview of the healthcare event and related actions.

- *Farmaco*: the column contains drug codes and their descriptions for certain medical events. Depending on the specific medical event, this column may or may not contain values. For example, if a medical event involves the administration of medication, the column will be populated with the corresponding drug code and description. However, if the medical event does not involve medication, this column may remain empty. The purpose of this column is to provide information about the drugs administered during relevant medical events, facilitating a detailed record of healthcare procedures and treatments.

Evento	DiagnosiPrincipale	InterventoPrincipale
087[M]-Edema polm..	518.84-Insufficienza Re..	93.90-Respirazione ...
087[M]-Edema polm..	518.81-Insufficienza Re..	93.96-Altro Tipo di...
098[M]-Bronchite ..	466.19-Bronchiolite acu..	93.94-Medicamento R...
087[M]-Edema polm..	518.81-Insufficienza Re..	93.96-Altro Tipo Di...
208-Interventi Sul..	574.01-Calcolosi Della..	51.01-Aspirazione ...
544[C]-Sostituzio..	820.01-Epifisi (Separaz..	81.52-Sostituzione ...

Table 4.5: Sample columns Evento

- *Qta, Valore*: these columns derive from the administrative nature of the data and represent the quantity and cost of healthcare events or services provided. The quantity column indicates the number of events or services, while the value column represents the associated cost. These columns allow for tracking the volume and cost of healthcare services delivered, providing valuable insights into healthcare activity and expenses.

4.2 Preprocessing

Data preprocessing plays a fundamental role in the field of machine learning and deep learning, especially when dealing with complex datasets rich in information from the healthcare domain. In this chapter, we will explore the methodologies and techniques used to transform a dataset of patient-related events into a suitable representation for analysis and prediction.

The transformation of raw data into a form suitable for model training involves several steps, including handling missing values, feature normalization, feature engineering, and anomaly management. In the context of healthcare data, these steps are crucial to ensure the validity and reliability of the generated models.

This thesis is focused on defining care trajectories for each patient. As mentioned care trajectories represent the temporal sequence of healthcare events and therapeutic actions performed for a patient throughout their treatment journey.

4.2.1 Data Transformation

Consequently, to the previous section and with all the characteristics observed during the data analysis the following action has been performed on data in order to transform each row of the dataset in an event related to a specific patient, concatenating them in order to obtain a trajectory suitable with the considered model.

Starting from the structure described in 4.1 we applied these steps:

- *re-code patients*: the first step applied is the re-mapping of the unique identifier for the patient, as mentioned in the raw data-id is described by the Fiscal Code field which is encrypted and so not easy to read or directly refer to. During the analysis it has been tested the integrity that each patient is strictly related to one exclusive province zone, so here this column is the first to be removed from our dataset in order to reduce the features. For what concern the patient fiscal code, each of them has been mapped to a new key-patient value composed by the prefix *pat* concatenated by an underscore with an increasing number for each different patient.

- *cleaning events*: following we moved to pre-process the core of patients' events. In particular, here two processes have been conducted in parallall, the first one concerning the transformation of each field to make it more suitable for machine readability rather than humans. This transformation involved adapting the structure of the data to make it more powerful for the final encoding. As mentioned in the data analysis subsection, most of the relevant columns for the event encoding were preset in a *code-description* structure. So the main possible problem could be some noise in the description making two different fields different also in the case they represent the same object. Since a direct encoding was available here for each field in consideration it has been kept the code part of the structure. The second process conducted for cleaning has been also a human-driven feature selection. Some columns have been removed from the dataset, as mentioned the first one was related to the residence of each patient, but for what concern the events two columns have been removed *Struttura*, related to the Structure in which the event was performed and *Specialità*, specifying a first high-level reference to a macro-category domain. These two fields have been removed due to a relation *MxN* between them and also because we want to give an unbiased representation from the point of view of the location of the performed event in order to obtain a more general event. Also the lasts two columns *Qta* and *Valore* has been removed in accordance with their administrative domain relevance.

- *aggregating events*: after the first cleaning and the identification of relevant features, each row has been aggregated using just the domain-related field, it has been cleaned from *null* values to further reduce dimensionality and replace this concatenation with a specific code. Also here an increasing number has been assigned, referring to a specific event and of course, keeping the same code for same event definition.

- *create the trajectory*: finally, step converted has been the definition of the trajectory. From the last described step our dataset has changed its structure starting from the first 13 different fields and reducing itself to the *Code_patient*, *Code_Event* and *Data* fields. First of all, for each patient it has been organized the dataset into partitions by grouping identical *Code_patient*, ordering them chronologically based on the *Data* column, and then defining event trajectories for each *Code_patient.* The structure obtained is the following for example:

$$pat\_1234 = [12, 587, 1, 653, ..., 12]$$

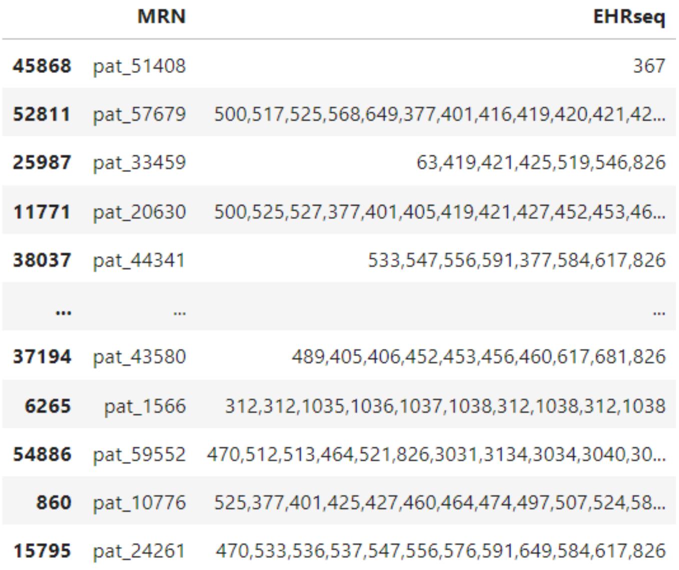
In conclusion from previous step of pre-processing described we obtained this final dataset structure shown in Figure 4.3 ready to feed the model.

4.2.2 Dictionary generation

In the previous section, we discussed the various preprocessing steps involved in transforming raw healthcare data into a suitable trajectory. Now, we turn our attention to the next stage in our pre-train pipeline: defining an encoding dictionary for learning representations by a Convolutional Neural Network (CNN).

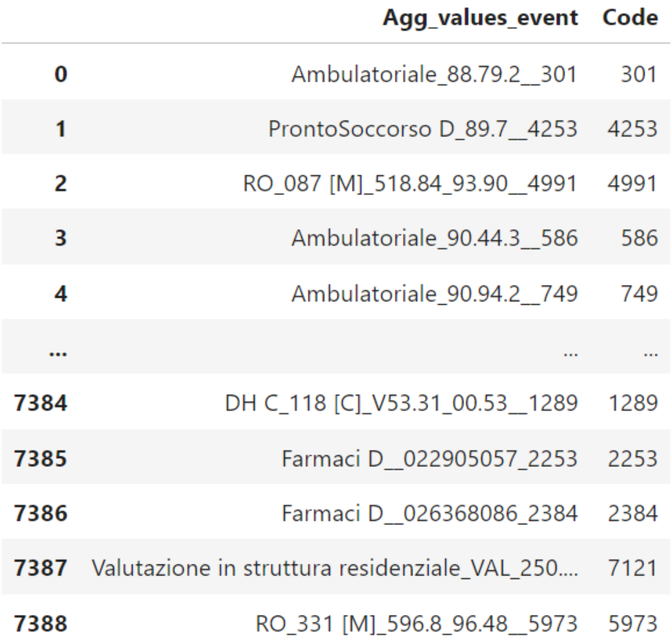
The encoding dictionary plays a pivotal role in the feature extraction process, providing a structured mapping between the raw data and their numerical representations. It serves as a bridge between the complexities of real-world healthcare events and the computational requirements of machine learning models.

The generation process of the dictionary is located between the *cleaning* and *aggregating* preprocess steps, in particular, all the possible distinct



	MRN	EHRseq
45868	pat_51408	367
52811	pat_57679	500,517,525,568,649,377,401,416,419,420,421,42...
25987	pat_33459	63,419,421,425,519,546,826
11771	pat_20630	500,525,527,377,401,405,419,421,427,452,453,46...
38037	pat_44341	533,547,556,591,377,584,617,826
...	...	...
37194	pat_43580	489,405,406,452,453,456,460,617,681,826
6265	pat_1566	312,312,1035,1036,1037,1038,312,1038,312,1038
54886	pat_59552	470,512,513,464,521,826,3031,3134,3034,3040,30...
860	pat_10776	525,377,401,425,427,460,464,474,497,507,524,58...
15795	pat_24261	470,533,536,537,547,556,576,591,649,584,617,826

Figure 4.3: Sample data post preprocessing



	Agg_values_event	Code
0	Ambulatoriale_88.79.2__301	301
1	ProntoSoccorso D_89.7__4253	4253
2	RO_087 [M]_518.84_93.90__4991	4991
3	Ambulatoriale_90.44.3__586	586
4	Ambulatoriale_90.94.2__749	749
...	...	...
7384	DH C_118 [C]_V53.31_00.53__1289	1289
7385	Farmaci D__022905057_2253	2253
7386	Farmaci D__026368086_2384	2384
7387	Valutazione in struttura residenziale_VAL_250....	7121
7388	RO_331 [M]_596.8_96.48__5973	5973

Figure 4.4: Dictionary structure sample

combinations have been sampled from the dataset in order to populate the dictionary.

In Figure 4.4 we can observe its structure, divided into two columns where the first ones contains all the *non-null* values human readable and the second one contains its encoded version machine readable.

After this process, we obtained a cohort of:

- *73.706* different patients
- *9.346* different events in the dictionary
- *15* mean events for each trajectory, from this point of view we have a lot of trajectories with a number of events greater than 200 and many more events with lower than 10 events

4.3 ConvAE Model

As mentioned in the Related Word section the ConvAE (Convolutional Autoencoder) [32] architecture is a representation learning model designed to transform patient Electronic Health Record (EHR) subsequences into low-dimensional, dense vectors. The architecture consists of three stacked modules: embeddings, Convolutional Neural Networks (CNNs), and Autoencoders (AEs). The architecture used can be observed in Figure 4.5.

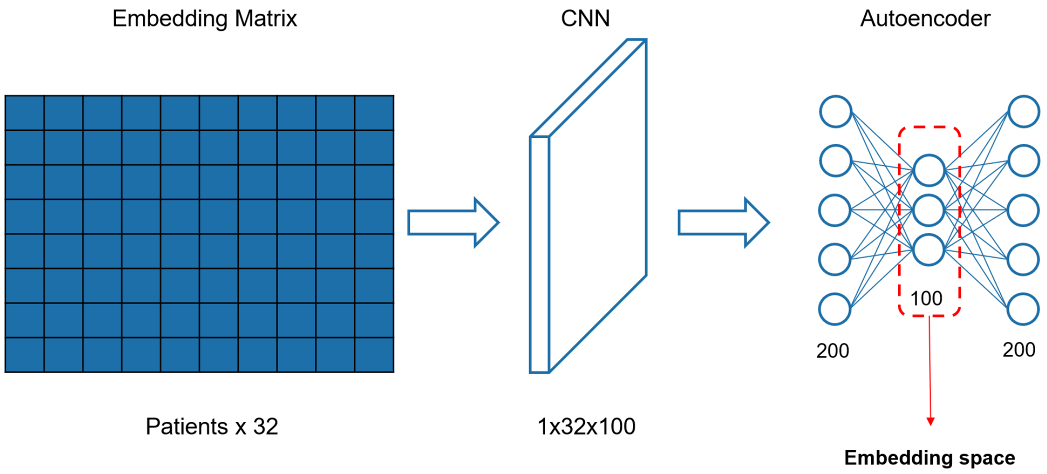


Figure 4.5: Highlight Model Architecture components

First, the architecture assigns each medical concept to an N-dimensional embedding vector to capture the semantic relationships between medical concepts. Each trajectory is standardized to a fixed length, which can be chosen by a parameter. All the parts that exceed this window are allocated as new trajectories for the same patient. This embedding matrix retains temporal information as the rows are temporally ordered according to patient visits. The CNN module then applies temporal filters to each embedding matrix, treating the matrices as RGB images with a third "depth" dimension. In detail it reshape the embedding matrix to a new dimension $\mathbb{R}^{1\times L\times N}$ then apply filters $\mathbf{k} \in \mathbb{R}^{1\times h\times N}$. In details for each filter j we have:

$$(\mathbb{R})_j = \mathrm{ReLU}\left(\sum_{i=0}^{h-1} k_i * \tilde{e}_i + b_j\right), \quad j = 1, \ldots, f$$

where we have

- $(\mathbb{R})_j$ represents the j-th element of the sequence;
- ReLU is the rectified linear activation function;
- k_i is the coefficient of the embedding e_i;
- b_j is the bias term associated with the j-th element;
- N is the dimension of the embedding;
- f is the output dimension.

This approach enables the model filters to learn independent weights for each encoding dimension, activating the most salient features in each dimension of the embedding space. The output of the CNN module is reshaped into a concatenated vector to learn different weights for each embedding dimension, highlighting representations of patient histories relevant to identify the most relevant characteristics of their semantic space.

The final module, the Autoencoder, learns the embedded representations for each patient subsequence. It reconstructs each initial input one-hot subsequence of medical terms from their encoded representations. The model is trained by minimizing the cross-entropy loss, allowing it to be trained without any supervised training samples. It extracts the hidden representation as the encoded representation of each patient subsequence, then it is transformed into a sequence of encodings that can be post-modeled to obtain unique vector patient representations.

In details the CNN module employed 50 filters with a kernel size of 5, utilizing the Rectified Linear Unit (ReLU) activation function. Meanwhile, the autoencoder consisted of four hidden layers comprising 200, 100, 200, and $|V| \times 32$ hidden nodes, respectively, with $|V|$ representing the vocabulary size. Activation functions employed include ReLU for the initial three layers and Softplus for the final layer, facilitating the generation of continuous output.

In details the Softplus function is represented as:

$$f(x) = \ln(1 + e^x)$$

For training the ConvAE is trained by minimizing the cross entropy loss:

$$CE(\text{Softmax}(O), s) = -\frac{1}{L}\sum_{j=1}^{L} \log(\text{Softmax}(O^j)_{w_j})$$

- O^j represents the output of the neural network for instance j,
- w_j represents the correct label for instance j,
- L is the total number of instances in the dataset,
- Softmax(x) applies the Softmax function to the input x,
- log(x) represents the natural logarithm of x.

4.4 Logistic Regression

Logistic regression is a widely used statistical technique for binary classification tasks. It is a type of regression analysis where the dependent variable is categorical, typically representing one of two classes (e.g., 0 or 1, true or false, yes or no). The logistic regression model predicts the probability that an observation belongs to a particular category based on one or more independent variables.

The logistic regression model employs the logistic function, also known as the sigmoid function, to map the output to a probability value between 0 and 1.

The sigmoid function is defined as:

$$f(z) = \frac{1}{1 + e^{-z}}$$

where z is a linear combination of the input features and their corresponding coefficients. The logistic regression model seeks to learn the optimal coefficients to maximize the likelihood that the predicted class matches the true class labels in the training data.

Mathematically, the logistic regression model can be represented as:

$$p(y = 1|\mathbf{x}; \mathbf{w}) = \frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}}$$

$$p(y = 0|\mathbf{x}; \mathbf{w}) = 1 - p(y = 1|\mathbf{x}; \mathbf{w})$$

where $p(y = 1|\mathbf{x}; \mathbf{w})$ represents the probability that the target variable y is equal to 1 given the input features $\mathbf{x}$ and model parameters $\mathbf{w}$. Similarly, $p(y = 0|\mathbf{x}; \mathbf{w})$ represents the probability that y is equal to 0. The model parameters $\mathbf{w}$ are learned during the training process.

Logistic regression can handle both linear and nonlinear relationships between the independent variables and the log-odds of the target variable. Despite its simplicity, logistic regression is often used as a baseline model for binary classification tasks due to its interpretability, efficiency, and effectiveness.

Chapter 5

Results

Throughout this thesis, we have delved into the dynamics of Machine Learning and Deep Learning applied to the medical field, in this concluding chapter, we will present the key findings that have emerged from our experiments, in particular:

- highlighting the performance of trained ConvAE models
- the analysis of results obtained through clustering algorithms
- the application of learned representations in a classification context

Through these detailed analyses, we will not only evaluate the effectiveness of our proposed model but also understand how these representations fit well in downstream possible tasks. We will begin by examining the details of the ConvAE training process, exploring various variants and parameters to understand their impact on model performance. Subsequently, we will focus on the analysis of results obtained through clustering algorithms, providing a detailed view of the structure and coherence of identified clusters. Finally, we will examine the results of our applied analysis, assessing the utility of learned representations in subsequent classification tasks.

Through these sections, we aim to provide a comprehensive overview of our findings, highlighting both the successes achieved and the challenges encountered during our work.

5.1 Training ConvAE Model

As a primary outcome, we evaluated the effectiveness of our data, after the preprocessing pipeline to adapt to the model chosen from the literature, in training the model. Specifically, two variants of the model were trained: the first using the same hyperparameter configuration as provided in the paper, and the second using an optimal hyperparameter configuration to maximize the model's accuracy, Table 5.1.

Model	Baseline	Optimized Parameters
ConvAE	0.765	0.854

Table 5.1: Accuracy achieved for each variant tested

Parameter	Baseline
Number of epochs	5
Batch size	128
Embedding size ℕ	100
Kernel size	5x5

Parameter	Optimized
Number of epochs	6
Batch size	256
Embedding size ℕ	100
Kernel size	6x6

Table 5.2: Accuracy achieved for each variant tested

The optimal configuration was determined through the exploration of various parameter configurations during training. In addition to these two variants, other variations were also tested with some modifications during the preprocessing phase. These included aggregation, where certain columns were removed during the generation of distinct combinations to reduce dictionary dimensionality and filtering where events with low frequencies were filtered to focus on more frequent events and reduce dimensionality and finally, as mentioned, the hyperparameter tuning where different parameter combinations of the model were tested to identify the best fit for our use case, starting from the baseline proposed in the ConvAE architecture. For what concern aggregating and filtering, these two variants led to only marginal improvements in terms of accuracy so we consider only hyperparameter tuning as a relevant result.

The aforementioned accuracy was calculated as follows using the self-reconstruction error function that is defined as the mean squared difference between the original input x and the generated output x', divided by the number of elements in the vectors:

$$E(x, x') = \frac{1}{n}\sum_{i=1}^{n}(x_i - x'_i)^2$$

Where:

- $E(x, x')$ is the self-reconstruction error function.
- x is the original input.
- x' is the output generated by the autoencoder.
- n is the number of elements in the vectors x and x'.
- x_i and x'_i are the individual elements in the vectors x and x', respectively.

5.2 Clustering results

The second part of the results that we are gonna analyze, concerns the clusterization on the obtained representation from the ConvAE model.

Starting from our 100-dimensional matrix representing each patient trajectory in the latent space, first, a t-SNE algorithm for dimensionality reduction has been computed. As mentioned in previous chapters it aims to reduce the 100-dimensional representations to bi-dimensional in order to make the representation suitable for a human readable plot. Then a k-Means Clustering algorithm was computed in different configurations in terms of number of clusters in order to observe how the data could be split. As shown in Figure 5.1 the final representation chosen is with a number of cluster parameters equal to 4. Since the clustering algorithm was executed automatically, a descriptive analysis was conducted to explore the underlying patterns within the dataset. Following the application of the clustering algorithm, a detailed analysis of each cluster was performed to evaluate the homogeneity and distinctiveness of the grouped data points.

In particular, given the schema of our dataset, a sample of relevant columns for this analysis has been selected. In particular, the columns under investigation are the ones regarding some aggregated information related to patients, like Age, Residence, Event Speciality, and Event Type.

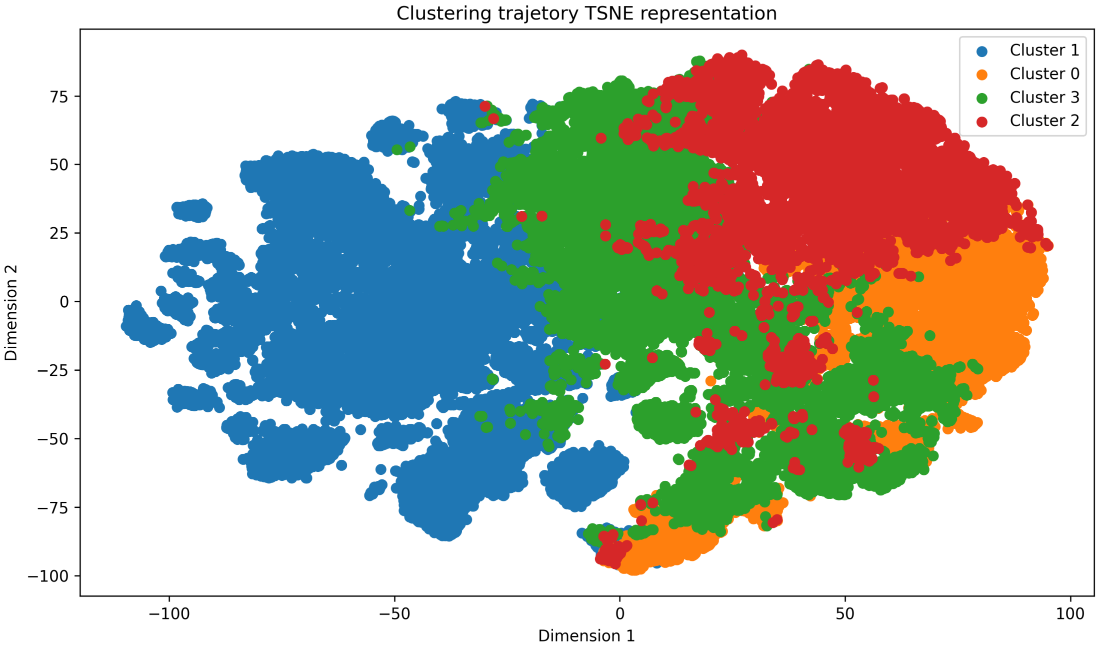


Figure 5.1: Highlight Model Architecture components

In Figure 5.2, the details that emerged from the analysis are described and overall, the results of the automatic clustering analysis are not positive. The identified clusters do not exhibit heterogeneous characteristics based on the analyzed columns. The absence of distinct patterns or trends within the clusters suggests that the clustering algorithm did not effectively partition the dataset into meaningful clusters. Further refinement of the clustering approach or exploration of alternative training methodologies in the model may be necessary to extract useful insights from the dataset under the clustering purpose.

In particular we can observe easily that for the Residence and Age distribution among the clusters we have a really homogeneous pattern among them, quite different situation seems for Specialties and Types however, it is not sufficient to distinguish clearly discriminatory characteristics among each cluster.

The reason why these clustering results are not positive lie in the fact that the model was not specifically trained for this particular purpose. Instead, its primary goal was to generate representations encompassing the entirety of each patient's trajectory, rather than optimizing for clustering performance.

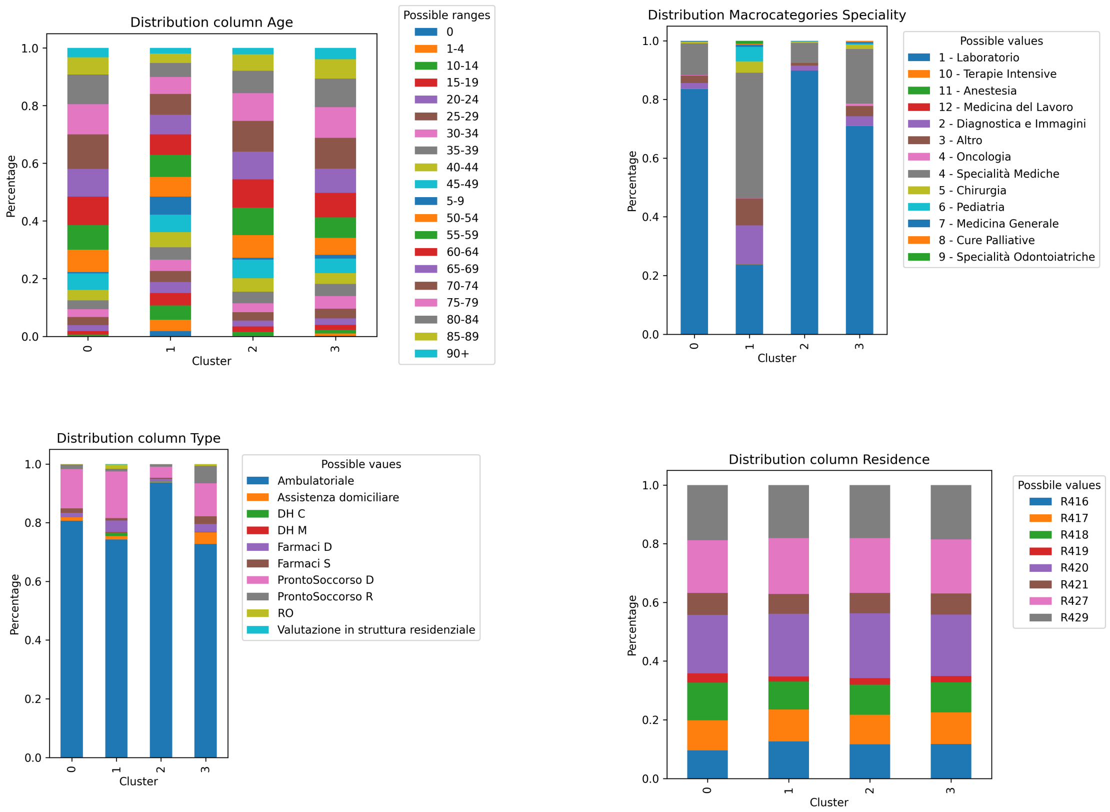


Figure 5.2: Column Age, Speciality, Type and Residence distribution

5.3 Downstream classification task

The last results that we will discuss are related to a little downstream task executed after the entire process, aiming to simulate a possible real pipeline scope. In particular, here we focused on a particular scope similar to the TrustAlert research purpose focusing on a particular category of patients affected by respiratory diseases.

After extracting the representation, the initial step involved human-guided labeling for patients suffering from respiratory diseases, with guidance from our collaborators in the TrustAlert project. Specifically, the TrustAlert project provided guidelines for identifying events indicative of respiratory pathologies. Using the generated transcoding dictionary, we identified the corresponding event codes and subsequently extracted patients along with their trajectories, labeling them as samples of patients with respiratory pathologies.

In our cohort of patients, we have identified 106 different patients with related events to this category. Starting from these then we need to find a similar subset of patients with some similar characteristics in order to obtain a negative sample useful for the classification.

In particular, as similar characteristics, we focused on two elements, *Age distribution* and *Trajectory size*. This is important for keeping the same distribution of people among ages and events that affect patients, obtaining a realistic representation set that accurately reflects the population you're modeling and avoiding inadvertently introducing bias into the model, improving generalization, and helping the model to learn robustly how to recognize the label focusing on more robust feature which can represent relevant information, instead of focusing on weakly ones like frequency of particular events. For this classification task in particular we've used a Logistic Regression algorithm.

5.3.1 Model Evaluation

The performance of a logistic regression model is typically evaluated using metrics such as accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC). These metrics provide insights into different aspects of the model's performance, such as its ability to correctly classify instances of each class, its overall predictive power, and its robustness to class imbalance. Giving more details on these metrics we have that, accuracy measures the proportion of correctly predicted instances out of all instances in the dataset. Precision measures the proportion of true positive predictions out of all positive predictions made by the model. Recall, also known as sensitivity, measures the proportion of true positive predictions out of all actual positive instances in the dataset. The F1-score is the harmonic mean of precision and recall, providing a balanced measure of the model's performance. AUC-ROC measures the trade-off between the true positive rate and the false positive rate across different probability thresholds, providing a comprehensive evaluation of the model's discrimination ability.

In details on our subset data these are the performances achieved by the Logistic Regression:

- **Model Accuracy:** This value represents the fraction of correct predictions made by the model. In this case, the accuracy value is 0.8889, which means that 88.89% of the model's predictions are correct.

- **Precision:** It represents the model's ability to not mislabel negative examples as positive. In our case, the precision for class 0 is 0.9286 and for class 1 is 0.8235. So, when the model predicts class 0, it is correct 92.86% of the time, and when it predicts class 1, it is correct 82.35% of the time.

- **Recall (Sensitivity):** It represents the model's ability to correctly identify positive examples. In our case, the recall for class 0 is 0.8966 and for class 1 is 0.875. So, 89.66% of positive examples from class 0 were correctly identified, and 87.5% of positive examples from class 1 were correctly identified.

- **F1-score:** It is a weighted average of precision and recall. It represents the balanced accuracy between precision and recall. In our case, the F1-score for class 0 is 0.9123 and for class 1 is 0.8485.

- **AUC-ROC:** This value represents the Area Under the Receiver Operating Characteristic Curve (AUC-ROC), which is a measure of the model's discriminatory ability. The higher the AUC-ROC, the better the model's ability to distinguish between positive and negative classes. In this case, the AUC-ROC is 0.9181, indicating a good degree of separation between the classes.

In Figure 5.3 we can observe a graphical representation of the decision boundary learned by the Logistic Regression algorithm.

It is clear that finding patients with these deseases would have been possible also in input space (by looking at the events in their trajectories). However, this results demonstrates the ability of our ConvAE to linearly separate patients with similar medical conditions in latent space, thus showcasing the effectiveness of the learned representations.

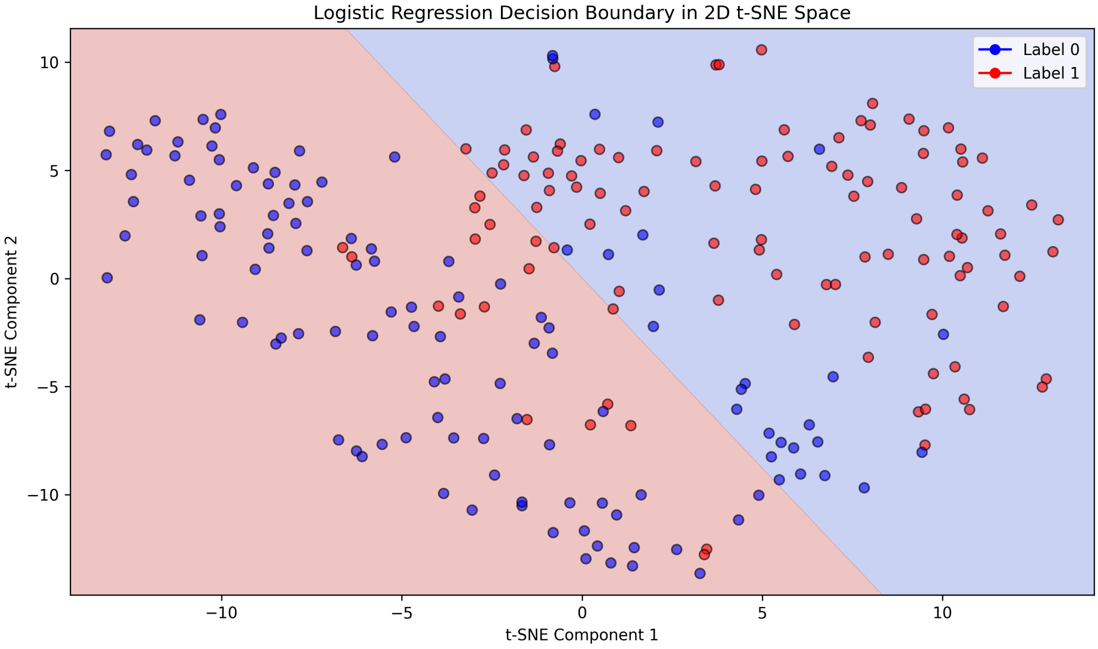


Figure 5.3: Decision Boundary categorization task

Chapter 6

Conclusions and future works

This study aimed to test and verify how a model designed to work with American EHRs could perform in a scenario like the Italian one, where this detailed type of data, albeit partially, is not yet available. Focusing on the final results discussed in the previous section, it is appropriate to concentrate on the downstream task executed, which, although focused on a small amount of available data, still demonstrated how this model could achieve adequate performance due to its architecture. Turning attention to possible future work, this result paves the way for potential utilization of this useful representation element within pipelines of varying complexity, leading to an effective representation that can be fine-tuned for multiple purposes.

A possible limitation of our approach may be related to the administrative nature of the data and their imbalanced content towards outpatient events, mainly related to blood tests. Due to the high cardinality of events associated with a single type of medical test, this can lead to a "dilution" of the information content of the datasets, resulting in a more complex learning phase. Therefore, for future work, various paths may open up depending on the different applications intended for this representative model, including experimenting with medical data filtering in the preprocessing phase to focus more on specific categories, each with relevant information content for the intended purpose.

Bibliography

[1] *TrustAlert - Research Project.* URL: https://www.trustalert.it/ (accesso alla pagina: 02 febbraio 2024).

[2] *S3K - J-his provider.* URL: https://s3k.it/jhis/ (accesso alla pagina: 10 febbraio 2024).

[3] Z. Obermeyer and E. J. Emanuel, *Predicting the Future—Big Data, Machine Learning, and Clinical Medicine*, *New England Journal of Medicine*, vol. 375, no. 13, pp. 1216–1219, 2016. https://doi.org/10.1056/nejmp1606181

[4] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, *Dermatologist-level classification of skin cancer with deep neural networks*, *Nature*, vol. 542, no. 7639, pp. 115–118, 2017. https://doi.org/10.1038/nature21056

[5] Shubham Panchal. *Data Types in Statistics.* Towards Data Science, October 2018. https://towardsdatascience.com/data-types-in-statistics-347e152e8bee.

[6] Unknown author. *What is Feature Engineering?.* SplashBI. https://splashbi.com/what-is-feature-engineering/.

[7] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, J. Dean, et al., *Scalable and accurate deep learning with electronic health records*, *npj Digital Medicine*, vol. 1, no. 1, p. 18, 2018. https://doi.org/10.1038/s41746-018-0029-1

[8] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, *Doctor AI: Predicting Clinical Events via Recurrent Neural Networks*, *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 301–318, 2016.

[9] Dhaval Rajpara. *Convolutional Neural Networks (CNN) Architectures Explained.* Medium, 2018. https://medium.com/@draj0718/convolutional-neural-networks-cnn-architectures-explained-716fb197b243.

[10] Unknown author. *TensorFlow Word Embeddings.* Scaler, n.d. https://www.scaler.com/topics/tensorflow/

tensorflow-word-embeddings/.

[11] T. A. Lasko, J. C. Denny, and M. A. Levy, *Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data*, *PLOS ONE*, vol. 8, no. 6, p. e66341, 2013. https://doi.org/10.1371/journal.pone.0066341

[12] Fondazione Ugo Bordoni, *Digitalization of the healthcare sector in Italy: the state of the art*, 2020.

[13] Ministry of Health, *The benefits of Electronic Health Records*, 2018.

[14] National Agency for Regional Health Services (AGENAS), *Challenges and opportunities in the implementation of healthcare information systems*, 2019.

[15] Italian Data Protection Authority, *Guidelines for healthcare data management and privacy protection*, 2017.

[16] Ministry of Health, *Strategic plan for digital health*, 2019.

[17] IBM Cloud, *Recurrent Neural Networks*, https://www.ibm.com/cloud/learn/recurrent-neural-networks.

[18] Coursera, *Natural Language Processing Specialization*, https://www.coursera.org/learn/nlp-sequence-models.

[19] DeepAI, *Sequence Model*, https://deepai.org/machine-learning-glossary-and-terms/sequence-model.

[20] Towards Data Science, *Recurrent Neural Networks and LSTM*, https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5.

[21] Towards Data Science, *Music Generation using Deep Learning*, https://towardsdatascience.com/music-generation-using-deep-learning-85010fb982e2.

[22] Kaggle, *Sentiment Classification using LSTM*, https://www.kaggle.com/spsayakpaul/sentiment-classification-using-lstm.

[23] Nature, *Machine Translation*, https://www.ibm.com/cloud/learn/machine-translation.

[24] Towards Data Science, *Video Activity Recognition and Its Applications*, https://towardsdatascience.com/video-activity-recognition-and-its-applications-ecda68998f54.

[25] Towards Data Science, *Named Entity Recognition*, https://towardsdatascience.com/named-entity-recognition-3fad3f53c91e.

[26] Machine Learning Mastery, *Sequence Prediction*, https://machinelearningmastery.com/sequence-prediction/.

[27] ScienceDirect, *Sequence Modeling*, `https://www.sciencedirect.com/topics/computer-science/sequence-modeling`.

[28] Analytics Vidhya, *Understanding Sequence Models in Deep Learning*, `https://www.analyticsvidhya.com/blog/2021/04/understanding-sequence-models-in-deep-learning/`.

[29] T. Cover and P. Hart, *Nearest neighbor pattern classification*, *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21-27, 1967.

[30] , @articleDBLP:journals/corr/abs-1907-09538, author = Yikuan Li and Shishir Rao and José Roberto Ayala Solares and Abdelaali Hassaïne and Dexter Canoy and Yajie Zhu and Kazem Rahimi and Gholamreza Salimi Khorshidi, title = BEHRT: Transformer for Electronic Health Records, journal = CoRR, volume = abs/1907.09538, year = 2019, url = http://arxiv.org/abs/1907.09538, eprinttype = arXiv, eprint = 1907.09538, timestamp = Sat, 23 Jan 2021 01:21:00 +0100, biburl = `https://dblp.org/rec/journals/corr/abs-1907-09538.bib`, bibsource = dblp computer science bibliography, https://dblp.org

[31] , @miscrasmy2020medbert, title=Med-BERT: pre-trained contextualized embeddings on large-scale structured electronic health records for disease prediction, author=Laila Rasmy and Yang Xiang and Ziqian Xie and Cui Tao and Degui Zhi, year=2020, eprint=2005.12833, archivePrefix=arXiv, primaryClass=cs.CL

[32] , *landi2020deep, title=Deep representation learning of electronic health records to unlock patient stratification at scale, author=Landi, Italo and Glicksberg, Benjamin S and Lee, Hyun-Chul and others, journal=npj Digital Medicine, year=2020, doi=10.1038/s41746-020-0301-z* `https://doi.org/10.1038/s41746-020-0301-z`

[33] *Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer Science & Business Media.

[34] *Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning.* MIT Press.