Machine Learning Project 2022/23

Lucia Fores[a], Marco Guarino[b], Elena Jiang.[c]
[a]fores.1836451@studenti.uniroma1.it
[b]guarino.1895383@studenti.uniroma1.it
[c]jiang.1846716@studenti.uniroma1.it

February 10, 2023

Abstract

In this project we aimed to create a ML model that helped in the prediction about patients whose health history is known.
Specifically we tried to create a model that would help in predict the possibility to incur into a cardiovascular events in the successive 6 months after the questioning of the model.
The EHR are a complex type data that represent the health history of a patient: for this reason we decided to create a model that only took account of the events that happened to the patients and than we decided to abstract more and only take account of the visits of the patients.
Our results provide a good starting point for a more deep study opening the possibility to expand and improve the models.

1 Introduction, context, and motivations

EHRs are a **relatively new type of data** whose importance is increasing day-by-day specifically with the intent of using them for models that could be helpful in make **more accurate and patient-related diagnosis in a shorter time**.
One of the problem of EHR is the necessity to deal with **heterogeneous data** that do not always follow a standard of collection and also the fact that the data can be incomplete and prone to error.
Even by having data that represents actual patients that are of some interest for the specific task of the model, the problem is that most of the time the characteristic that is studied is the one for which there are the **least data present**.

In this project, in which we tried to create a model that was able to detect the presence of a cardiovascular events in the following 6 months after the querying of the model, **we faced all the main problems related to the use of EHR**.
For the project we decided to use **Deep Learning models** in order to create a system that was able to predict the cardiovascular risk of a patient in a given space of time.

As a proposed strategy to work with EHR we proposed some Deep Learning models whose goal was to learn **the connections between the different events that occurred to a patients** (and how much the combination of all of this could have led to the occurrence of a cardiovascular disease).
Additionally we also tried to check if we could only consider the presence of some most important events in the life of a patient and not all the events that occurred to them by checking if this led to an improvement of the performances of our models.
After the proposed models we also performed a Bayesian Optimization (hyperparameter tuning) thanks to the **bayes-opt** library in order to find the best combination of hyperparameter to increase the performances of our models.

The motivation behind this project is to study (and hopefully contribute to) an emerging and still unripe field of research, hoping to find a good strategy to work with difficult data to manage and to help in the development of strategies that can be then used with different tasks always in the health field.

Even though working with Deep Learning models has represented a big challenge for use, we believe that by doing a good pre-processing and a good study of which data to use is possible to achieve satisfying results for the task.

2 Dataset description

As the first step in our project we decided to study and describe the given dataset.

2.1 Analysis description

To describe the dataset a preliminary analysis of the data has been made.
Specifically for each table all the features have been inspected.
The idea at the base of this type of analysis is to clearly describe the data sample on which the model has been created so to understand all the possible weakness and strengths of the model.
Please note that in the description of the dataset we refer to the following concepts:

- **Patients' meta-id**: tuple that indicates the id of the patients in the dataset
- **AMD Codes**: codes created by the italian medical association of diabetologists (AMD - Associazione Medici Diabetologi -); this codes are used to describe each possible observation about diabetic patients, for a complete description of each code (in italian) please refer to the *File Dati AMD* which can be found on the *Annali AMD* (2019)
- **Date**: the date in the dataset always refers to the **day of the beginning of the event**; this means that, for example, if a patient has been diagnosed with a temporal continuous disease (like diabetes) in the dataset will only be indicated the date in which the patient has been diagnosed with; also note that the dates are in the **aaaa/mm/gg** format
- **ATC Codes**: codes that refer to the ATC classification system (Anatomical Therapeutic Chemical) used for the systematic classification of drugs and controlled by the World Health Organization
- **STITCH Codes**: codes created by the STITCH research center (Sapienza Information-Based Technology InnovaTion Center for Health) that are used to encode some events of interest for patients

For each table of the dataset are given:

- The name of the table
- A general description of the data presented in the table
- The name of the feature and a specific description of each feature in the table

2.2 Dataset analysis

anagraficapazientiattivi In the table are given all the **biographical data** of the patients.
Specifically the **features** are:

- **idcentro**: one of the value of the meta-id of the patients
- **idana**: other value of the meta-id of the patients
- **sesso**: the gender of the patients
- **annodiagnosidiabete**: the year in which the patient has been diagnosed with diabetes
- **tipodiabete**: the diabetes' type of the patients
- **scolarita**: the level of education of the patient
- **statocivile**: the marital status of the patient
- **professione**: the job of the patient
- **origine**: the origin of the patient
- **annonascita**: the birth year of the patient
- **annoprimoaccesso**: the year of the first hospitalization of the patient

- **annodecesso**: the death year of the patient

Please note that in the inspection made the feature *scolarita, statocivile, professione* and *origine* were not taken in consideration since they are not useful for the task required to the model; also note that in this examination *idcentro* and *idana* were not taken in consideration since it's a general observation on the data and not patient-related.
So the **general consideration** that can be made thanks to the **graphical representation of the distribution** are the following: we can say, as depicted in Figure 1 that the dataset is **pretty much equally distributed between male patients and female patients**, also **all the patients** have been diagnosed with **the same type of diabetes**, as Figure 2 suggests.

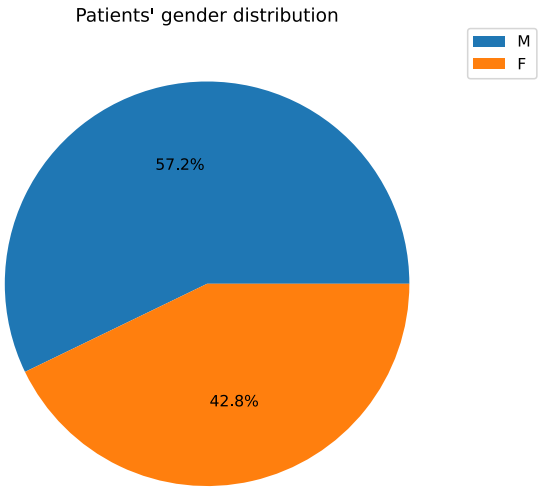

Figure 1: Gender distribution

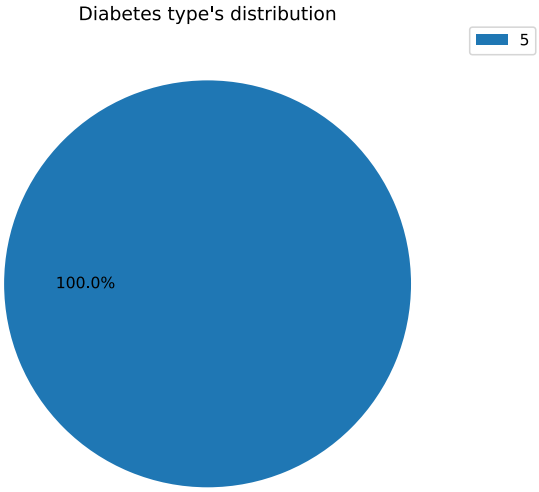

Figure 2: Diabetes' type distribution

prescrizionidiabetenonfarmaci In the table are given all the **diets** and the **blood glucose checks** prescripted to the patients.
Specifically the **features** are:

- **idcentro**: one of the value of the meta-id of the patients
- **idana**: other value of the meta-id of the patients
- **date**: date of the beginning of the event
- **codiceamd**: AMD code that encode the event
- **valore**: value assigned to the AMD code

The **events** that can be found in this table are listed in Table 1:

AMD Code	Meaning
AMD152	Self-monitoring of blood glucose
AMD086	Self-monitoring of blood glucose
AMD228	Integrated management
AMD090	Diet only
AMD096	Insulin pump

Table 1: Events listed in prescrizionidiabetenonfarmaci table

prescrizionidiabetefarmaci In the table are given all the **diabetes-related drugs** prescripted to the patients.
Specifically the **features** are:

- **idcentro**: one of the value of the meta-id of the patients

- **idana**: other value of the meta-id of the patients
- **codiceatc**: ATC code related to the drug
- **quantita**: drug's quantity prescripted to the patients
- **idpasto**: id that refers to the meal prescripted to the patients
- **descrizionefarmaco**: description of the drug prescripted to the patients (specifically the **commercial name** of the drug)

prescrizioninondiabete In the table are given all the **prescriptions of the drugs non diabetes-related** taken from the patients.
Specifically the **features** are:

- **idcentro**: one of the value of the meta-id of the patients
- **idana**: other value of the meta-id of the patients
- **data**: beginning date of the event
- **codiceamd**: AMD code that encode the event
- **valore**: value assigned to the AMD code

Note that the table is organized in a way that the **AMD codes** describe the **drug families** and the **values** are the **ATC code** related to the specific drug.
The **AMD codes** that can be found in this table are listed in Table 2

AMD Code	Meaning
AMD121	Antihypertensive drugs
AMD124	Lipid-lowering drugs
AMD131	Antiplatelet drugs

Table 2: Events listed in prescrizioninondiabete table

esamistrumentali In the table are given all the information related to the **medical examinations** performed by the patients.
Specifically the **features** are:

- **idcentro**: one of the value of the meta-id of the patients
- **idana**: other value of the meta-id of the patients
- **data**: beginning date of the event
- **codiceamd**: AMD code that encode the event
- **valore**: value assigned to the AMD code

Note that all the **values** are either **N** (**Normal**) or **P** (**Pathological**).
The **AMD Codes** that can be found in this table are listed in Table 3

AMD Code	Meaning
AMD034	Cardiovascular autonomic tests
AMD035	EMG (electromyography)
AMD040	ECG (electrocardiogram)
AMD041	Echocardiography
AMD042	Stress test
AMD043	Coronarography
AMD050	Eye examination
AMD051	Retinography
AMD052	Fluoroangiography
AMD079	Echocolordoppler lower limbs

AMD Code	Meaning
AMD080	Lower limb arteriography
AMD116	Ecostress
AMD117	Holter Pressure
AMD118	Blood pressure self-control
AMD125	Peripheral sensitivity test
AMD126	ABI (ankle / arm pressure index)
AMD132	Myocardioscintigraphy
AMD135	OCT (optical coherence tomography)
AMD137	Angio RMN lower art

Table 3: Events listed in esamistrumentali table

esamilaboratorioparametri In the table are given all the information related to the **laboratory tests** performed by the patients.
Specifically the **features** are:

- **idcentro**: one of the value of the meta-id of the patients
- **idana**: other value of the meta-id of the patients
- **data**: beginning date of the event
- **codiceamd**: AMD code that encode the event
- **valore**: value assigned to the AMD code

The **AMD Codes** that can be found in this table are listed in Table 4

AMD Code	Meaning	AMD Code	Meaning
AMD001	Height	AMD024	AER (microalb excretion rate)
AMD002	Weight	AMD026	A / C Urinary (albumin / creatinine ratio U)
AMD003	Abdominal circumference	AMD028	Creatinine clearance
AMD004	PAS (systolic)	AMD109	GT range
AMD005	PAD (diastolic)	AMD110	CPK
AMD007	Fasting blood sugar	AMD111	Microalbuminuria
AMD008	HbA1c	AMD113	Potassemia
AMD009	Creatininemia	AMD134	Azotemia
AMD010	Total cholesterol	AMD226	Creatininuria
AMD011	Triglycerides	AMD305	HbA1c
AMD012	HDL cholesterol	AMD910	Microalbuminuria
AMD013	LDL cholesterol	AMD911	A / C Urinary (albumin / creatinine ratio U)
AMD014	Uricemia		
AMD015	GOT	AMD912	Ab anti Insulin
AMD016	GPT	AMD913	Urine albumin
AMD017	Amylase	AMD914	Ab anti GAD 65
AMD018	Alkaline phosphatase	AMD915	Ab IA2 / ICA 512
AMD021	Platelets	AMD916	Ab anti GAD 65
AMD022	Hemoglobin	AMD917	Ab IA2 / ICA 512
AMD023	Protein Urine	AMD927	BMI

Table 4: Events listed in esamilaboratorioparametri table

Note that there are **some events** that are related to **two different codes** and this is due to the fact that **the same parameter can be measured with different measure units**.

esamilaboratorioparametricalcolati In this table can be found the **values** and the **correspondences between some AMD Codes and the STITCH Codes**.
Specifically the **features** are:

- **idcentro**: one of the value of the meta-id of the patients
- **idana**: other value of the meta-id of the patients
- **data**: beginning date of the event
- **codiceamd**: AMD code that encode the event
- **valore**: value assigned to the AMD code
- **codicestitch**: STITCH code that encode the event

The **AMD codes** present in this table **are all also present** in the *esamilaboratorioparametri* table. The **STITCH codes**, and the relative correspondence to the AMD code, that can be found in the table are listed in Table 5

STITCH Code	Meaning	Correspondent AMD Code
STITCH001	BMI	AMD927
STITCH002	LDL Cholesterol	AMD013
STITCH003	Non-HDL Cholesterol	
STITCH004	eGFR MDRD	
STITCH005	eGFR CKD-EPI	AMD304

Table 5: STITCH codes listed in esamilaboratorioparametricalcolati table

diagnosi In the table are listed **all the possible diagnosis** that can be made on a patient. Specifically the **features** are:

- **idcentro**: one of the value of the meta-id of the patients
- **idana**: other value of the meta-id of the patients
- **data**: beginning date of the event
- **codiceamd**: AMD code that encode the event
- **valore**: value assigned to the AMD code

The **AMD codes** that can be found in this table are listed in Table 6

AMD Code	Meaning
AMD037	Mononeuropathy
AMD038	Polyneuropathy
AMD039	Autonomic neuropathy
AMD044	Ischemic heart disease
AMD045	Heart failure
AMD046	Angina
AMD047	Myocardial infarction
AMD048	Coronary angioplasty
AMD049	Coronary bypass
AMD053	Non-proliferative retinopathy
AMD054	Proliferative retinopathy
AMD055	Laser-treated diabetic retinopathy
AMD056	Maculopathy
AMD057	Severe blindness or low vision
AMD058	Not diabetic foot
AMD059	Trophic lesion
AMD060	Previous ulcer
AMD061	Major (non-traumatic) amputation
AMD062	Minor (non-traumatic) amputation
AMD063	Osteomyelitis / Charcot osteopathy / periostitis
AMD064	Soft tissue infection
AMD065	Incipient nephropathy
AMD066	Overt nephropathy
AMD067	Chronic renal failure
AMD068	Dialysis
AMD303	Ischemic stroke
AMD307	Hemorrhagic stroke

AMD Code	Meaning
AMD069	Nephrotic diabetes syndrome
AMD070	TIA
AMD071	Stroke, unspecified
AMD072	Lower limb arteriopathy
AMD081	Lower limb angioplasty
AMD082	Peripheral by-pass
AMD083	Hypertension
AMD097	Cigarette smoke
AMD119	Hypertensive retinopathy
AMD129	Absent diabetic neuropathy
AMD130	Non diabetic retinopathy
AMD204	Diabetic retinopathy present of unspecified severity
AMD205	Diabetes blindness
AMD206	Blindness from other causes
AMD207	Gangrene
AMD208	Revascularization of intracranial vessels and neck
AMD209	Cerebral vascular disease
AMD210	Preproliferative retinopathy
AMD226	Symptomatic non-severe hypoglycemia
AMD227	Asymptomatic non-severe hypoglycemia
AMD245	Foot malformations
AMD247	Other comorbidities
AMD257	Non-diabetic nephropathy
AMD300	Advanced diabetic ophthalmopathy
AMD302	Not diabetic nephropathy

Table 6: Events listed in diagnosi table

3 Task 1 description

After the dataset analysis, from which we saw that the dataset is made by **heterogeneous data**, we started the **pre-processing of the dataset** to obtain only the patients that were truly interesting for our task (also called as **active patients**).
To better show the process of **cleaning and selecting the data**, at the end of each step, we reported the **number of patients remaining and the class distribution**.

3.1 Select events of interest

We started with the **deletion of duplicate rows** in the tables **anagraficapazientiattivi** and **diagnosi**, so that we knew, by construction, that we deleted all possible errors made by the creators of the dataset (in fact we arranged our dataset in a way that let us be sure about the fact that there weren't two, or more, identical patients and two, or more, equal diagnosis made on the same day for the same patient).
Our goal was to consider only the patients with at least one **cardiovascular event** in their health history.
The cardiovascular events considered are **macro-cardiovascular events**.
Specifically these events are:

- **AMD047**: Myocardial infraction
- **AMD048**: Coronary angioplasty
- **AMD049**: Coronary bypass
- **AMD071**: Ictus
- **AMD081**: Lower limb angioplasty
- **AMD082**: Peripheral By-pass Lower Limbs
- **AMD208**: Revascularization of intracranial and neck vessels
- **AMD303**: Ischemic stroke

For being able to choose patients with at least one of the previous diagnosis we worked on the table ***diagnosi***.
In order to get only the interest patients, we **filtered the table over the macro-events** and we found the patients with at least one cardiovascular event.
At the beginning of the work of pre-processing the number of patients in the table **anagraficapazientiattivi** was **250000** and after the filtering the number of remain patients was **50000**.
The graph of the distribution in Figure 3 shows that the **80% of patients were discarded**.
We then decided to delete some **irrelevant features** (in the **anagraficapazientiattivi** table) for the task so we deleted the columns *scolarita*, *statocivile*, *professione*, *origine* and *annoprimoaccesso*.

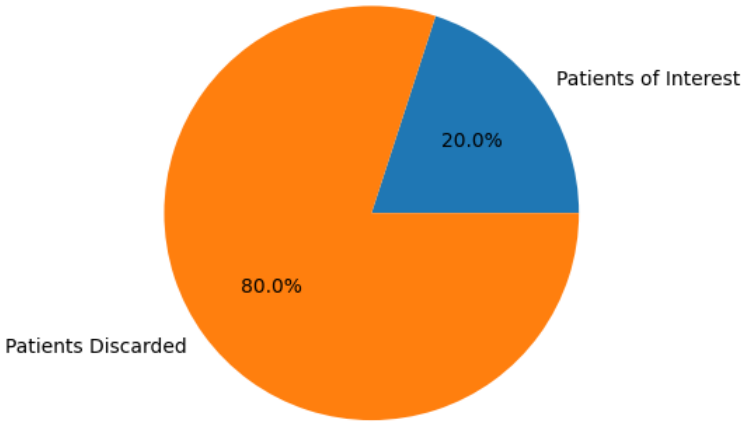

Figure 3: Select events of interest distribution

3.2 Invalid feature cleaning

After getting only the patients with at least one cardiovascular event we proceeded in the pre-processing of the dataset by **deleting all the events whose dates were not valid**.
In particular, we looked for dates of events, that regarded a particular patient, that were stored in the dataset as **happened before his birth or after his death**.
We also looked for those events that were **registered without a date** (and so present as *nan*s in the table).
We proceeded by getting rid of all the events with this characteristic, so we checked all the tables that contained patients' events.
After the inspection over each events' table we filtered again the **anagraficapazientiattivi** so that we got rid of those patients whose all events were deleted in the check.
The first table we considered was **diagnosi**: only a minority of patients were removed in this case; in fact the number of active patients, after the previous step, was **50000** and after this check the number of remain patients was **49912**.
The graph of the distribution Figure 4 shows that only a small number of patients were discarded.

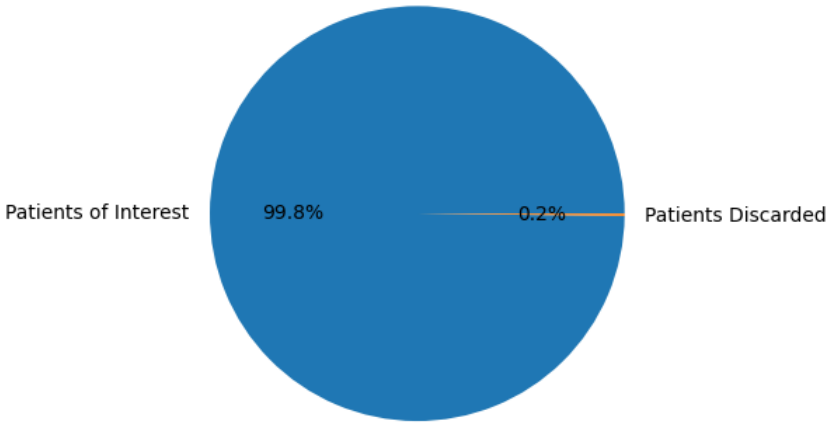

Figure 4: Invalid feature cleaning distribution on diagnosi table

3.3 Remove patients with all dates in the same month

In order to give the model a real understanding of an heath record history of a patient we considered only those patients whose **history was long enough to give useful information to the models**.
For this, we decided to delete all the patients whose **health record trajectory was shorter than or equal to one month**.
We decided to consider as a **trajectory** the **union of all the events that regarded a patient**: this means that we decided to delete those patients whose union of all the diagnosis and examinations (both self-sexaminations, instrumental and laboratory ones) was all focused in one month or less.
The tables considered were **diagnosi**, **esamilaboratorioparametri**, **esamilaboratorioparametricalcolati** and **esamistrumentali**.
Firstly for each *esami* table, we cleaned its invalid feature like we did with **diagnosi** in subsection 3.2, in this way when we finished to remove the patients with the trajectory long at most one month, we filtered just once over **anagraficapazientiattivi** to retrieve the active patients.
After inspecting all the tables we computed the intersection of all patients to be deleted and updated the table **anagraficapazientiattivi**.
So the number of active patients, after the previous step, was **49912** and after this step of the pre-processing the remaining patients are **49590**.
The graph of the distribution in Figure 5 shows again that only a small number of patients was discarded.

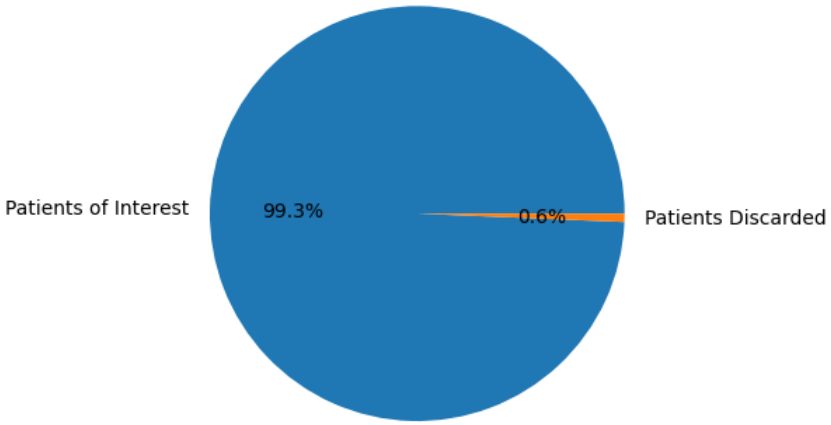

Figure 5: Active patient distribution according to examination and diagnosis table updates

We then applied the invalid feature cleanup to the tables **prescrizionidiabetenonfarmaci**, **prescrizionidiabetefarmaci** and **prescrizioninondiabete**: so we deleted the events with the invalid feature from all these tables.
Anyway the number of active patients remained the same, **49590**.

3.4 Modify the actual range of EsamiLaboratorioParametri

One of the main problem of the project was **dealing with heterogeneous and differently-collected data**.
Precisely because of the different sources from which the data were extracted, we found the necessity to **check if the values for the events respected the standard value scale**.
Precisely, we modified the actual ranges of the AMD code and STITCH code shown in Table 7 according to their *True range*.
We check for the events stored in the tables **esamilaboratorioparametri** and **esamilaboratorioparametricalcolati**.
In particular in the table **esamilaboratorioparametri** we check for the AMD codes while in the other table we analyse the STITCH codes.

Code	Meaning	True Range
AMD004	Systolic blood pressure	$40 \leq x \leq 200$
AMD005	Diastolic blood pressure	$40 \leq x \leq 130$
AMD007	Fasting blood glucose	$50 \leq x \leq 500$
AMD008	HbA1c	$5 \leq x \leq 15$
AMD009	Creatininemia	Not available
AMD111	Microalbuminuria	Not available
STITCH001	BMI	Not available
STITCH002	LDL Cholesterol	$30 \leq x \leq 300$
STITCH003	Non-HDL Cholesterol	$60 \leq x \leq 330$
STITCH004	eGFR MDRD	Not available
STITCH005	eGFR CKD-EPI	Not available

Table 7: Events listed in prescrizionidiabetenonfarmaci table

By inspecting the tables, if we found a patient whose value of one of his examinations with the AMD or STITCH code listed in the table Table 7 was **lower or higher than the true range**, we changed it: when the true value is under the true range, we changed it to the value of the minimum limit, for example, if a patient had 38 as the value for the examination with code AMD004 it became 40; similarly, if the real value is higher than the true range, we modified the real value with the maximum limit.

It is worth to notice that in the table there are some code whose *true range* is not available, so we decided to not take account of this codes in the changing.

3.5 Cohort selection and label definition

As we said before we needed to perserve only the patients whose information could be useful for the model: since we want to create a system capable of **predicting if a patient will have a cardiovascular event in the following 6 months** from when it's questioned, we decided to **delete all the patients whose health record trajectory was shorter than or equal to 6 months**.
To do so we at first **unified all the events** (meaning we unified all the tables we had in our dataset) and then proceeded into computing **the date of the first event of a patient and the date of the last event of a patient**: then we checked how many months passed between those two dates and kept only the interesting patients.
The number of active patients at the previous step was **49590** and after the elimination of the patients we are not interested in, the number of the remaining patients was **47702**.
The graph of the distribution Figure 6 shows the final distribution of patient.

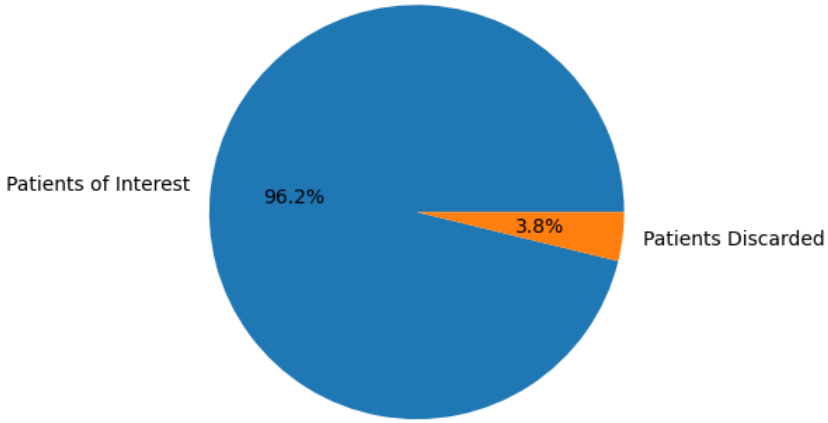

Figure 6: elimination of the patients that have a short trajectory

After getting rid of all the not interesting patients we faced the problem of the **label computing**; we divided the patients into **two classes**: the **positive ones (1)**, that were the patients that had, starting from the date of the last event that occurred to them, a cardiovascular event in the previous six months, and the **negative ones (0)** which were all the other ones.

So before all the work made on the pre-processing the number of active patients was **250000**, and after cleaning process the number of the active patients is **47702**.
The distribution of patients depicted in Figure 7, shows that more than the 80% of patients was discarded.

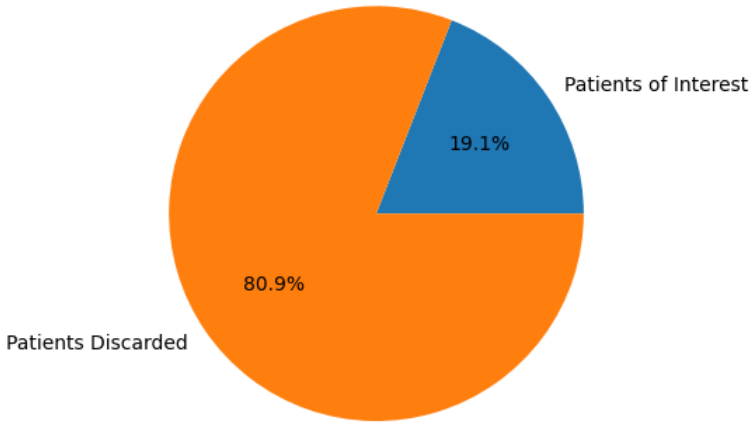

Figure 7: Final patients distribution

At the end we merged the labeled table with all events with the **anagraficapazientiattivi** in a final dataframe (that containes all the information and events for each patients) called **definitive**.
By inspecting the data we found out that there were **8980 patients with cardiovascular events** (positive class) and **38722 patients that didn't had a cardiovascular event** (negative class).
The distribution of the patients is shown in Figure 8.

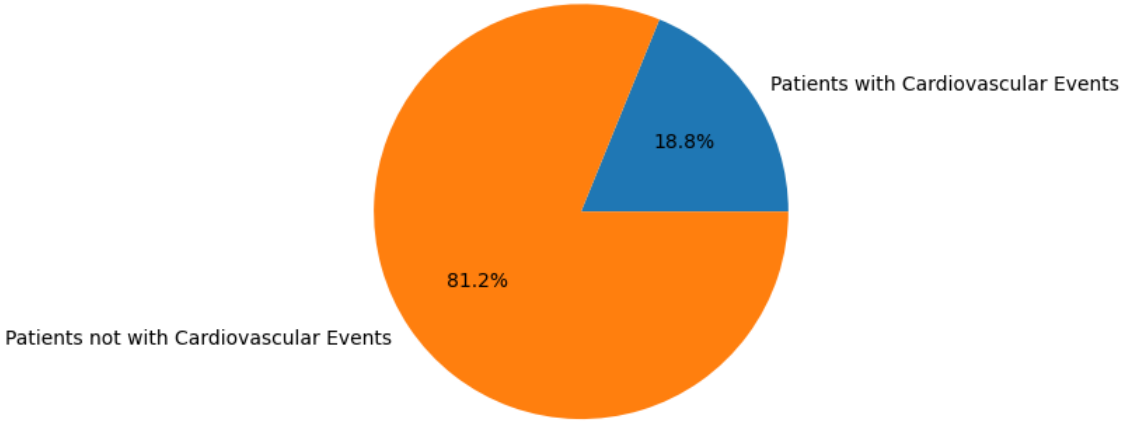

Figure 8: Patients with or not with Cardiovascular Events

We also found out that in the **definitive dataframe** there were **3273605 Cardiovascular events** and **12499589 not-Cardiovascular events** which were almost 80% of all the events.
The distribution of events is shown in Figure 9.

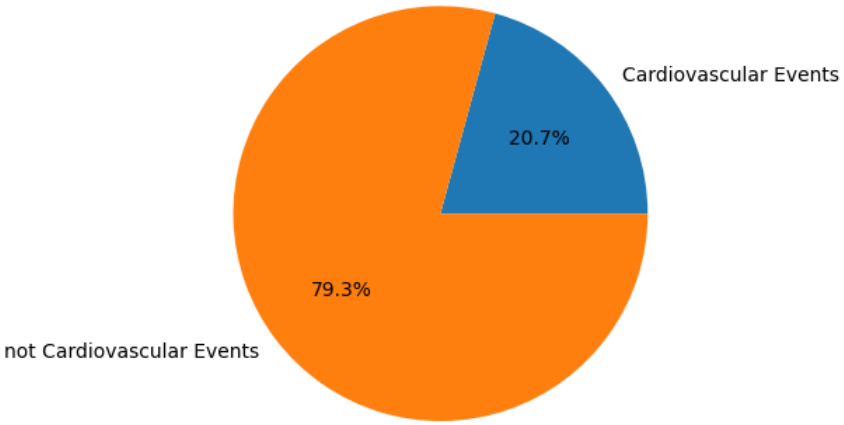

Figure 9: Cardiovascular events and other events

4 Task 2 description

In the second part of the project we managed to get a **balanced dataset** and then we **evaluated different models** on it.

4.1 Features' choice

The first action we focused on was managing the dataset in a way that let us have only the information we thought could be helpful to the models.
Specifically we thought to feed the models only with **the events that occurred in a patient's life** (and so we fuse all the different codes in a single feature called ***codici***); we then decided to drop all the features that didn't felt important for the task (thinking that only the progression of the events would be interesting but not their values since the large and heterogeneous intervals used to describe them).
So at the end we decided to drop the following features: **idpasto**, **quantita**, **codiceamd**, **descrizione-farmaco**, **valore**, **codicestitch**, **codiceatc**, **annodiagnosidiabete**, **tipodiabete**, **annodecesso**.
In this way we ended up having a comprehensive dataset (called ***definitive dataset***) that **stored all the events happened for all the patients**.
So at the end of this action we ended up by having a dataset with the features depicted in Table 8:

Feature	Meaning
idcentro	Half of patients' id
idana	Half of patients' id
sesso	Patients' gender
annonascita	Patients' birth year
codici	Codes of (all) events that could happen to patients
data	Day on which the event happened
datamax	Day on which the most recent event happened
label	Label computed for the patient (in subsection 3.5 and propagated to all the events of the patient)

Table 8: Features after the choice

It's important to notice the presence of three particular features:

- **sesso**
  We decided to keep the gender of the patients as something important for the models because we wanted to try to let them be aware about the possible gender differences in the occurrences of cardiovascular diseases (as stated in Gao, Chen, Sun, and Deng (2019))
- **datamax**
  This feature won't really be useful for the models (and will be later dropped) but we will use it during the class balancing
- **annonascita**
  This feature won't be really be useful for the models (and will be later dropped) but we will use it to create a specific feature for one of our model

The dataset was anyway really imbalanced (the number of patients that didn't had a cardiovascular events in the 6 month before their last event were much more than the one that had it) so we started our work at balancing it.

4.2 First approach at dataset balancing - A "smart" up-sampling -

The first approach we used to balance the dataset consisted in creating a **smart way to up-sample the minority class**: instead of just duplicate some rows of the dataset that represented events related to patients with positive labels we tried to create **"realistic" copies** of the patients with **"realistic" histories for their health records**; to do so we got all the rows of the dataset about positive patients and then shuffled and deleted some rows representing some events that occurred.
In this way we created some copies that weren't totally the same as the patients they came from but still preserved a **likely health record history**.

It's important to notice that all our copies have **distinct ids** (this was decided because we thought that would be helpful for the models to let them know which patients had which events and so we didn't want to compromise the true health history of the real patients).
We created as much copies as we could to not imbalance the class distribution by getting too much positive patients and in this way, at the end of this first part of our class balancing, we had with the following distributions:

- **Total number of patients**: 74642 (started from 47702)
- **Number of negative patients (label 0 - not cardiovascular events -)**: 38722
- **Number of positive patients (label 1 - got cardiovascular events -)**: 35920 (started from 8980)
- **Total number of events**: 22949247 (started from 15773156)
- **Number of negative events**: 12499566
- **Number of positive events**: 10449681 (started from 3273590)

Notice that after this part of the balancing we dropped the feature **datamax** because, even though it was crucial for the creation of the copies, that would not be an important information for the models.
Before proceeding with the class balancing we decided to add a new feature in our dataset that would have been useful in the future: we created the feature ***etaevento*** that represented the age of the patients when the specific event occurred; after its creation we dropped the feature **annonascita**.

4.3 Latter approach at dataset balancing

The latest approach to class balancing had a not-so-easy genesis as we tried different approaches before getting to the one we lastly used.

4.3.1 First attempt - SMOTE and SMOTENC -

Our first idea was to create, in an advanced way, the latest instances of the positive class that were missing to balance the class distribution.
To do so we immediately thought about using **SMOTE** so that we would create synthetic but likely new instances of the data: SMOTE requested to have only numerical features so we encoded all the values thanks to a **LabelEncoder**; anyway due to the way on which SMOTE interpolates the values of the features we ended up having some values that couldn't be decoded, so we decided to go back to our steps.
After some researches we found out that the correct way to work with heterogeneous datasets (with both numerical and categorical values) was to use **SMOTENC** that had a different way to interpolates values (the same as SMOTE for numerical values but the assignment of the **most frequent value among the neighbors** for the categorical ones): we tried to use it but due to some limitations about time and resources we were not able to ever finish the computation of the missing instances.
So we decided to proceed with an **undersampling of the majority class**.

4.3.2 Second attempt - RandomUnderSampler -

Our second idea was to make an under-sampling of the majority class so that we would balance the distributions.
At first we tried to use the **RandomUnderSampler** implemented in the **scikitlearn** library but that messed up with the ordering of the labels and the features so we decided to **manually make the under-sampling**.

4.3.3 Last attempt - Manual under-sampler -

We decided to continue with our idea to make the under-sampling but we made it directly on the dataset: to do so we decided to **randomly delete some rows that were labeled as negative**.
At the end of our work we had the following distributions:

- **Total number of patients**: 74642
- **Number of negative patients (label 0 - not cardiovascular events -)**: 38722

- **Number of positive patients (label 1 - got cardiovascular events -)**: 35920
- **Total number of events**: 20899362 (started from 22949247)
- **Number of negative events**: 10449681 (started from 12499566)
- **Number of positive events**: 10449681

Even though the dataset remained imbalanced from the number of patients point of view we decided to consider it balanced because the **number of positive and negative events (on which we made our predictions later) were the same**.
After the class balancing we then **evaluated our dataset** with some models.

4.4 Dataset Evaluation

Before evaluating our dataset we made the last preparation on the data: specifically we decided to apply a **label encoding** on every categorical feature we had (beside the feature ***data***) and we decided to **translate the dates in UNIX timestamps**.
We then evaluated our balanced dataset upon three different models:

LSTM The first model we presented was a simple Long Short Term Memory (LSTM): we used a simple architecture created thanks to the library **keras**; our model was **sequential** and consisted in a **LSTM layer** and a **Dense layer** with **ReLU** as activation function.
It's important to notice that we tried to train the neural network both with the Sigmoid function and the ReLU function and we got the same result on loss and accuracy; we also tried to train it over more epochs but we always got the same results we had by train it with only one epoch (so in an attempt to optimize time and resources we ended up by training it only for one epoch).
With our balanced dataset we got an **accuracy of 84.32%**.
Specifically the **confusion matrix** computed starting from the model's predictions is represented in Table 9:

	0	**1**
0	2090270 (TN)	0 (FP)
1	655430 (FN)	1434173 (TP)

Table 9: LSTM Confusion Matrix

The relative **F-score** for the LSTM is **0.81**.
So, as we can see from the metrics presented, the model **is really good at predict the negative class** (it in fact manage to predict correctly the negative class the **100%** of time) **and is pretty much good at predict also the positive class** (it manage to predict correctly the positive class **68.63%** of the time).

T-LSTM The second model presented is a Time Aware Long Short Term Memory (T-LSTM): for this model we used the same architecture presented for the LSTM but we fed it with a dataset that had also the feature ***etaevento*** so that it would have been aware about the specific time (related to the life of the patient) in which the event occurred (meaning that we gave to the model the information about **how old the patients were when their events occurred**).
Even with this model we saw that the train produced the same results independently from the number of epochs so we decided to train it upon a single one.
With our balanced dataset we also got an **accuracy of 84.32%**.
Specifically the **confusion matrix** computed starting from the model's predictions is represented in Table 10:

	0	**1**
0	2090270 (TN)	0 (FP)
1	655425 (FN)	1434178 (TP)

Table 10: T-LSTM Confusion Matrix

The relative **F-score** for the T-LSTM is **0.81**.
So, as we can see from the metrics presented, the T-LSTM has the **same overall performances** than LSTM.

PubMedBERT As for PubMedBERT we managed only to create a dataset that could be helpful for the model (we provided the same dataset as before but we **substituted the codes with their human-readable description**).
Unfortunately we weren't able to understand how to make the model work so we do not have results for it.

5 Task 3 description

During the third and last part of the project we tried to present a **new prediction strategy** to use with our model.

5.1 The idea behind the strategy

In EHR (Electronic Health Records) we have a lot of data that in reality refer to the same "composite event" in which the single "sub-events" took place.
Specifically in our dataset we had **visits** not actually encoded (but only "virtually" encoded, since we could know that a visit took place anytime we saw more than one event related to the same patient that took place on the same day than another).
We wanted to create a prediction strategy that **only took account of the date (and the order) of the "composite event" but not the order of the "sub-events"**.
To do so we decided to **actually encode the visits** (and so we grouped all the events of a patients that took place on the same day) and then we decided to **choose a representative** for the visit and we decided to **store the number of "sub-events"** that took place during the visit.
We decided to take a representative for the visit because we started from the concept of *class representative* of an *equivalence class* in algebra: in equivalence class the application of some functions on a representative of the class will produce the same output no matter which representative we choose (because it's independent from the representative and only dependent from the class); as the representative of the visit we choose the **mode** of all the event that took place during the visit.
We also decided to remember how many events happened during a visit because we thought that that could be a measure of the "importance" of the visit (the more events took place the more accurate the visit would have been).
So at the end of this process our dataset had the shape represented in the Table 11:

Feature	Meaning
idcentro	Half of patients' id
idana	Half of patients' id
sesso	Patients' gender
datavisita	Day on which the visit happened
rappresentantevisita	Code of the event chosen as visits' representative
numeroeventi	Number of events that happened during the visit
etavisita	Patients' age when the visit happened
label	Label computed for the patient (in task 1 and propagated to all the events of the patient)

Table 11: Features of the reshaped dataset

With this new information in our dataset, we decided to evaluate the data on a model.

5.2 Evaluation of the dataset

We decided to evaluate our dataset over a T-LSTM.
Even with the same results for LSTM and T-LSTM (as stated in subsection 4.4) we decided to continue to use a T-LSTM because we thought that the information provided by how old the patients were when the event occurred would still be helpful for the model: it is in fact known (Rodgers et al. (2019)) that the ageing of a patient is a factor (together with the gender) that must be considered in the study (and in our case for the prediction) of the possibility of the occurrence of cardiovascular diseases.
With the new shape of the dataset the T-LSTM gave us an **accuracy of 86.92%**.
Specifically the **confusion matrix** computed starting from the model's prediction is represented in Table 12:

	0	**1**
0	227719 (TN)	676 (FP)
1	60481 (FN)	178643 (TP)

Table 12: Custom Model Presented Confusion Matrix

The relative **F-score** for the model is **0.85**.
From the confusion matrix we can see that, even though the **performance on the negative class are slightly worse** (it now predict correctly the class the **99.7%** of the time while with the T-LSTM the prediction accuracy on the negative was 100%), the **performance on the positive class are a little bit better** (it now predict correctly the class the **74.7%** of the time while with the T-LSTM the prediction accuracy on the positive class was 68.63%).
We can then appreciate a slightly improvement in the performance even if it isn't actually much compared with the results obtained from the previous models.

5.3 Bayesian Optimization

We then performed a **Bayesian Optimization** thanks to the python library **bayes-opt** hoping to find the best configuration for the hyper-parameters of the model.
Unfortunately we weren't able to find the correct configuration because, probably because we performed the optimization over a few hyper-parameters and for a limited number of iterations (due to a lack of time and resources), the model that we created with the best configuration returned from the optimization performed worse than our original model.

References

Annali amd. (2019). Retrieved from https://aemmedi.it/annali-amd/

Gao, Z., Chen, Z., Sun, A., & Deng, X. (2019). Gender differences in cardiovascular disease. *Medicine in Novel Technology and Devices*, *4*, 100025. Retrieved from https://www.sciencedirect.com/science/article/pii/S2590093519300256 DOI: https://doi.org/10.1016/j.medntd.2019.100025

Rodgers, J. L., Jones, J., Bolleddu, S. I., Vanthenapalli, S., Rodgers, L. E., Shah, K., ... Panguluri, S. K. (2019, apr). Cardiovascular risks associated with gender and aging. *Journal of Cardiovascular Development and Disease*, *6*(2), 19. Retrieved from https://doi.org/10.3390%2Fjcdd6020019 DOI: 10.3390/jcdd6020019