Machine Learning Project 2022/23

Del Signore Lorenzo[a], Di Marco Andrea[b], Popa Alexandru.[c]
[a]delsignore.1605952@studenti.uniroma1.it
[b]dimarco.1835169@studenti.uniroma1.it
[c]popa.1840967@studenti.uniroma1.it

July 10, 2023

Abstract

In this project we analyzed patient medical and personal data to create a trajectory of events and try to predict whether the patient is going to have a cardiovascular disease in the next six month. We modeled a number of approaches both in the system architecture and the patents' trajectory definition, including LSTM architecture, T-LSTM, Transformer architecture and Bayesian Optimization on the Transformer architecture without Positional Encoding. We concluded that the best algorithm for this task is the Transformer model with optimized hyper-parameters since it focuses more on the events rather than the order of said events, thanks to the Attention Matrix mechanism.

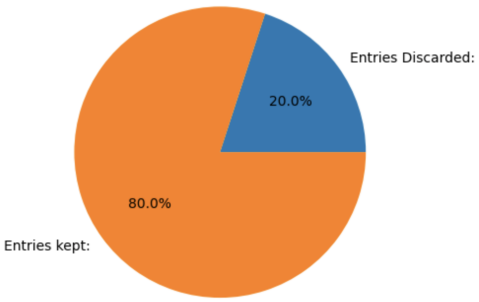
1 Data Preprocessing

1.1 Select events of interest

We started off by discarding all the patients with no macro cardiovascular event in their trajectory, since we only want to analyze patients with at least one macro cardiovascular event.

We managed to easily perform this task by using Pandas library built in commands.

After this operation the amount of discarded patients was 50000 out of 250000.



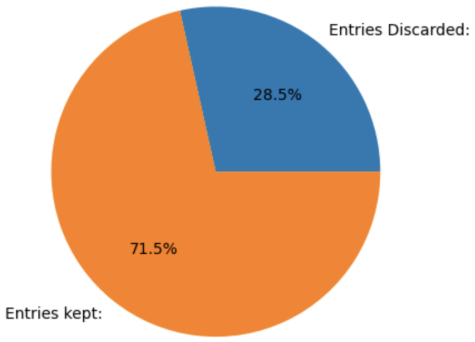
1.2 Invalid feature cleaning

We then proceeded to delete wrongfully reported events such as events that have allegedly happened before the patient's birth or after their death.

We ended up discarding 16340153 events out of 57338410.

1.3 Remove patients with all dates in the same month

Since we want to train the model only with long trajectories we shall remove the patients with all events taking place in the same month.



1.4 Values out of range

The values in the esami_laboratorio_parametri data set aren't all legal values. We had to modify their ranges in order to keep them withing correct ranges.

Sadly we could not find out the real ranges of all the codes in the data set.

We decided to change the exceeding values to the bound they were violating. If a value was higher than the upper-bound of that code we change it to the upper-bound.

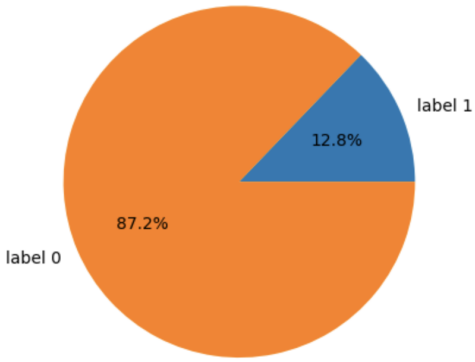
Out of 7295926 entries we only ended up changing 10522 (0.14%)

1.5 Cohort selection and label definition

After all the previous steps we kept only patients with at least two events (either macro or not) and labeled as 1 all the patients that had at least one macro cardiovascular event in the last six months of their respective trajectory. We labeled as 0 all the others.

We managed to do this by creating a data set with the latest event for each patient, we then proceeded to create another data set with dates corresponding to a six month offset from their latest event. If a patient ended up having a macro event dated after said *six month ago* date, then their label has been set as 1.

After this procedure we ended up with the following label distribution.



1.6 Merge micro-tasks

We finally merged all the results from the above procedures in two data sets:

- `final_patients`: This data set contains the list of every valid patient and their label
- `final_df`: This data set contains all the events of the patients in `final_patients` from all the data sets

2 LSTM & T-LSTM Models

2.1 Data preprocessing

For the subsequent steps to work we first had to change the value types of the `final_df` data frame.

1. **eta**: We created a column that contains the patient's age during the event.
2. **annonascita** and **annodiagnosidiabete**: we dropped these columns because tey are not relvant to our task since we already have the patient's age during every event.
3. **scolarità**: we filled the *NaN* values with zeroes
4. **statocivile**: we filled the *NaN* values with zeroes
5. **professione**: we filled the *NaN* values with zeroes
6. **origine**: we filled the *NaN* values with zeroes
7. **sesso**: we change the M with 0 and F with 1
8. **valore**: some values are strings and other values are floating numbers, the latter are left unchanged, for the strings:
    - The strings with letters in it have been encoded with the LabelEncoder from Pedregosa et al. (2011)
    - The strings with only digits in it, have been cast has floating numbers.

2.2 Delete history

We deleted the last six months of history of every patient to avoid giving the model prediction hints into the future.

At first we only performed this action on patients with label 1, we then found out that the model would recognize such patients from the others not because of the events but simply because they had shorter sequences. We solved this problem by deleting the last six months of every patient.

2.3 Naive class up-sampling

The classes of patients are not balanced, for this reason we used two different up-sampling methods, the first one is creating $m = 2$ copies of each patients with events being shaffled and having a probability $p = 0.05$ of being deleted.

The second up-sampling method is described in the section SMOTE.

2.4 PubMedBERT

For the subsequent steps to work we need to encode the AMD codes as numbers. We require this action to make the codes readable by the model.

We encoded the AMD codes using the pretrained PubMedBERT model from Gu et al. (2020), with the plain text meaning of the AMD codes like so.

The PubMedBERT model though returns an encoding of way too many features, we thus used the PCA algorithm from Pedregosa et al. (2011) to bring the features down to a manageable amount. Given Google Colab's limitations the aforementioned manageable amount turned out to be 1.

2.5 Sequences

We defined a trajectory for every patient starting with their details:

1. **sex**: As shown by Lori Mosca and Wenger (2011) gender can be a factor.
2. **scholarship**: For the reasons shown by Petrelli et al. (2022)
3. **marital status**: As shown by Chun Wai Wong (2018)
4. **profession**: As shown by Ghahramani et al. (2020)

5. **origin**: As shown by Chaturvedi (2003)

We defined two different sequences:

1. **Normal**: These sequences start with the details of the patient and then follows with every event of the patient written as such:
   (a) Patient's age during the event
   (b) PubMedBERT encoding
   (c) Value
2. **Time-Sensitive**: These sequences have the same format as the normal ones but for every period of 3 month in which the patient has no events, these sequence have a series of empty characters (three zeroes). This represents the passage of time between the events.

2.6 SMOTE

The second method used to balance the classes, other than the up-sampling, has been performed with the SMOTE algorithm.

We called the SMOTE algorithm on the data frame containing the processed sequences twice. Once for the normal sequences and once for the time-sensitive.

After this action we obtained a balanced class distribution.

2.7 LSTM & T-LSTM Models Architecture

We can now finally build and train our models. For this task we chose the LSTM model framework provided by keras (Abadi et al. (2015)) Since we are dealing with binary classification we added to the model a dense layer with the softmax function that returns the probability of the two classes.

We trained the two models with the respective sequences and the following hyper-parameters:

- **epochs**: 10
- **batch size**: 32
- **loss**: sparse categorical crossentropy
- **optimizer**: adam
- **activation**: softmax
- **test size**: 20%

More detailed test results are in section Final Results.

3 Transformer Model

3.1 Model Architecture

We wanted to build a new model that wouldn't take into account the sequence order. For this reason we built a Transformer model without positional encoding, this way the Transformer doesn't know the order of the sequence, relying only on the attention mechanism as shown by Vaswani et al. (2017).

After the feed forward network we added a linear layer with a two feature output and finally fed these features to a softmax that returns the probability of the two classes.

The hyper parameters used for this model are:

- **head size**: 256
- **num heads**: 4
- **ff dim**: 4
- **num transformer blocks**: 4
- **mlp units**: 128

- **mlp dropout**: 0.4
- **dropout**: 0.25

In depth results of the model can be found at the section Final Results.

3.2 Bayesian Optimization

We performed an additional search for the hyper parameters using Bayesian Optimization with the `bayes_opt` python library. Due to the limitations imposed by Google Colab we only performed a short search of the parameters' space by having the algorithm do five iterations. Each iteration built and trained a Transformer model for two epochs.

The proposed parameters are shown in Table 1

	head size	**heads**	**ff dim**	**transformer blocks**	**mlp units**	**mlp dropout**	**dropout**
value	413	3	3	2	1	0.0370	0.0117

Table 1: Suggested parameters.

In depth results of the model can be found at the section Final Results.

4 Final Results

In this section we will compare the performance of the four resulting models: *LSTM*, *T-LSTM*, *Transformer* and the Transformer with *Bayesian* Optimization.

The results show that when it comes to accuracy, the time-sensitive sequences do have an impact on the LSTM architecture. Given our problem to predict macro cardiovascular events in the patient's future we can take the precision metric on class 1 as the most relevant since a wrongful class 0 prediction (False Negative) can lead to disastrous result in the patient's life while a wrongful class 1 prediction (False Positive) might even be better of for the patient in the long run, leading them to assume a healthier lifestyle.

Given these considerations we can reliably say that the better model is the Transformer with no positional encoding, trained with the parameters suggested by the Bayesian Optimization.

	LSTM	T-LSTM	Transformer	Bayesian
Accuracy	30.44%	47.69%	84.31%	**85.57%**

Table 2: Test results.

	Precision		Recall		F1 Score		AUC
class	0	1	0	1	0	1	
LSTM	**81.51%**	95.06%	96.03%	**77.83%**	**88.17%**	**85.59%**	**86.93%**
T-LSTM	73.89%	74.02%	74.78%	73.11%	74.33%	73.56%	73.94%
Transformer	79.45%	92.02%	93.58%	75.37%	85.94%	82.87%	84.47%
Byesian	78.03%	**98.64%**	**99.03%**	71.63%	87.28%	82.99%	85.33%

Table 3: More detailed test results.

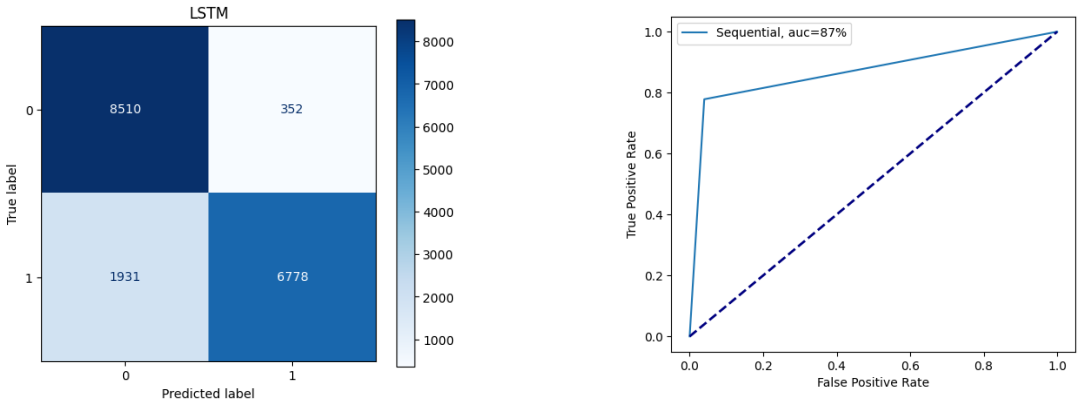


Figure 1: Vanilla-LSTM Model

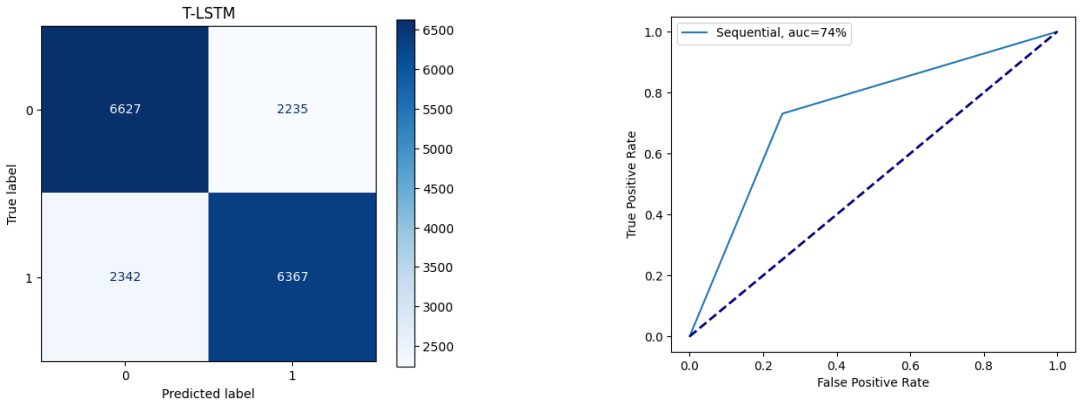


Figure 2: T-LSTM Model

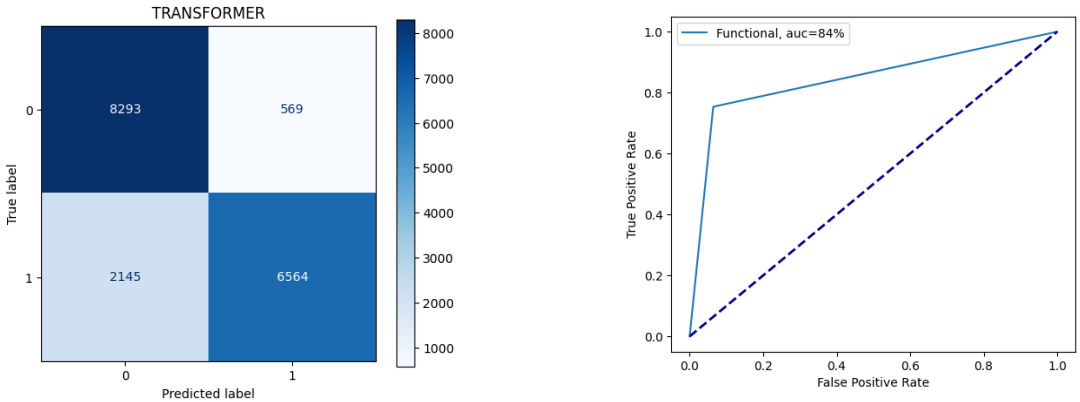


Figure 3: Transformer Model

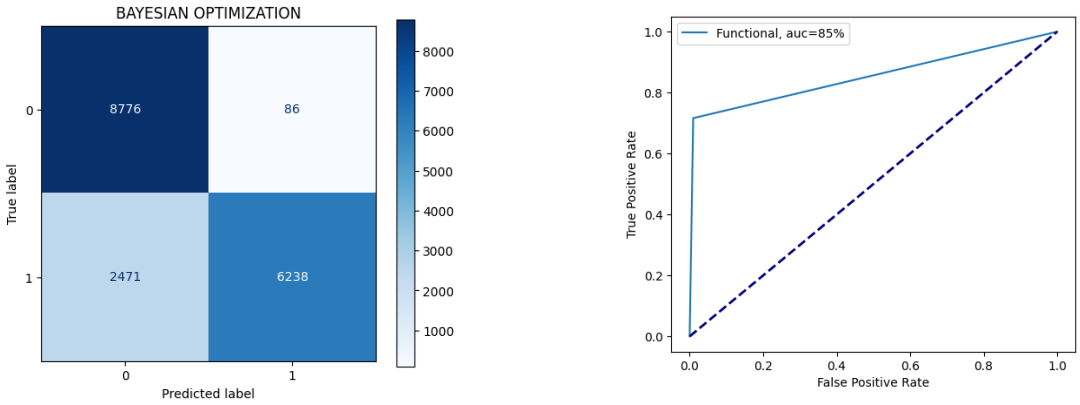


Figure 4: Transformer with Bayesian Optimization

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems.* Retrieved from https://www.tensorflow.org/ (Software available from tensorflow.org)

Chaturvedi, N. (2003). Ethnic differences in cardiovascular disease. *Heart*, *89*(6), 681–686. Retrieved from https://heart.bmj.com/content/89/6/681 DOI: 10.1136/heart.89.6.681

Chun Wai Wong, A. N. M. G. A. S. M. P. W. M. A. P. K. M. M. A. M., Chun Shing Kwok. (2018). Marital status and risk of cardiovascular diseases: a systematic review and meta-analysis.

Ghahramani, R., Aghilinejad, M., Kermani-Alghoraishi, M., Roohafza, H., Talaei, M., Sarrafzadegan, N., & Sadeghi, M. (2020, June). Occupational categories and cardiovascular diseases incidences: a cohort study in iranian population. *Journal of preventive medicine and hygiene*, *61*(2), E290—E295. Retrieved from https://europepmc.org/articles/PMC7419113 DOI: 10.15167/2421-4248/jpmh2020.61.2.1359

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., ... Poon, H. (2020). *Domain-specific language model pretraining for biomedical natural language processing.*

Lori Mosca, E. B.-C., & Wenger, N. K. (2011). Sex/gender differences in cardiovascular disease prevention.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Petrelli, A., Sebastiani, G., Di Napoli, A., Macciotta, A., Di Filippo, P., Strippoli, E., ... d'Errico, A. (2022). Education inequalities in cardiovascular and coronary heart disease in italy and the role of behavioral and biological risk factors. *Nutrition, Metabolism and Cardiovascular Diseases*, *32*(4), 918-928. Retrieved from https://www.sciencedirect.com/science/article/pii/S093947532100538X DOI: https://doi.org/10.1016/j.numecd.2021.10.022

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). *Attention is all you need.*