PAU CHAVES (U215102) & ANIOL PETIT (U214019)

FINAL PROJECT: HEART ATTACK PREDICTION

1. BACKGROUND

Heart disease remains one of the most significant global health challenges, leading to millions of deaths annually and posing a major economic burden on healthcare systems. In the United States alone, it accounts for approximately **647,000 deaths per year**, making it the leading cause of mortality. The primary risk factors include **high blood pressure, high cholesterol, smoking, diabetes, and genetic predisposition**. Other lifestyle-related elements, such as **poor diet, lack of physical activity, and chronic stress**, further increase the probability of developing heart disease.

The great issue with heart diseases is that normally one doesn't realize he is at risk before experiencing severe symptoms such as **chest pain or pressure**. For that reason, detecting individuals at high risk before they experience those symptoms is essential for **preventative intervention**, and prediction of heart attacks based on a person's health features can play a crucial role here. Because of that, the primary goal of this study is to develop a **predictive model** that can determine whether an individual is likely to have or develop heart disease or attack based on their health and lifestyle data.

2. DATASET

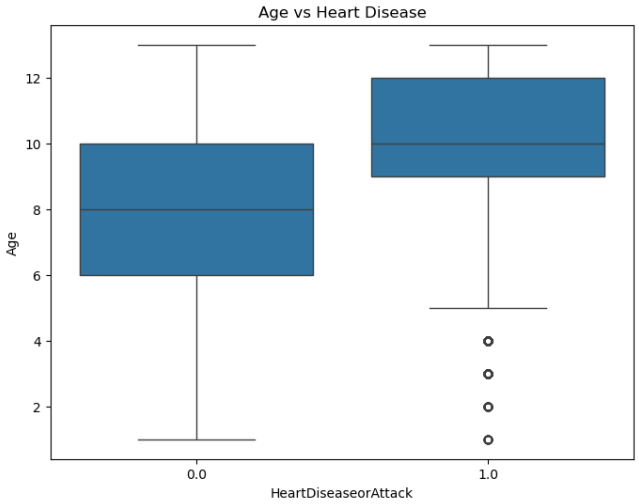
This classification problem is tackled using **machine learning techniques** applied to the **Behavioral Risk Factor Surveillance System (BRFSS) dataset from 2015**. The dataset consists of **253,680 survey responses**, including self-reported health behaviors, chronic conditions, and the use of preventive healthcare services. Note that the dataset was picked from Kaggle, and it was a cleaned version of the original dataset from where the data was collected, which originally had over one thousand features and it was quite messy.

After some data visualization, we quickly realized that the key challenge of this dataset is its **class imbalance**, with **only 23,893 individuals diagnosed with heart disease compared to 229,787 without the disease**. This imbalance must be carefully managed to avoid biased predictions that favour the majority class.

Furthermore, we could see how different features had very different scales thanks to the histograms plotted for every feature, which allowed us to see that many variables were binary, other numerical, other were categorized into bins…so it was clear that we had to standardize the data to work on the same scale, so we can avoid misrepresentative results.

As for linear relations between variables, we didn't identify many of them. After plotting the correlation matrix and some 2D plots between some features we thought to be interesting and our target feature, there wasn't any relation as strong as to be pointed out. The only thing worth mentioning is the mean age of subjects that suffered heart

attacks, which is quite high, indicating that subjects that are more likely to suffer heart attacks may be older (despite some minor existing cases that are younger, of course).



3. METHODS

a) Data Preprocessing

Before jumping to defining and training any models, it is fundamental to clean and process our data according to the necessities discovered during the data visualization phase. In our case, we pointed out as potential issues the heavy class imbalance in our target variable from one side, and from the other side the quite different scales between different features. These are two aspects that must be handled in order to avoid any bias or inconclusive results.

Regarding the imbalance, we applied two different techniques and then trained our models with both sets. The first approach was to use SMOTE to balance classes. This procedure basically picks all instances of the minority class, and from its k-nearest neighbours it generates synthetic records of this same minority class, balancing our target variable. This approach, though, has two principal setbacks: in the first place, in a dataset like ours with many records, adding records to the minority class results in a huge dataset, making the whole training process much more expensive and time consuming. In second place, generating synthetic samples introduces some bias, as the way of generating them makes it quite easy for machine learning algorithms to establish patterns between records and achieving great levels of accuracy which may not be fully representative.

The second approach was to balance classes, instead of adding samples to the minority class, removing them form the majority class. We removed samples randomly with probability 0.9, as we discovered that the ratio was approximately of 10 samples of majority class for each sample of minority class. This solution gives us a much more economical and viable in terms of computing time approach, however, reducing the dataset size so drastically can decrease performance of our model significantly. For simplicity, from this moment

on,we'll refer as **dataset 1** to the dataset balanced with SMOTE, and **dataset 2** to the dataset balanced removing records from the minority class.

As for standardization, we just applied standard scaling in both cases, ensuring that the mean of the data was 0 and the standard deviation was 1, having all data in the same scale, which as stated before is crucial for models' performance.

We finally split our data onto training and testing set, giving the testing set 30% of the total data.

b) Algorithms

For our prediction task, we decided to train 5 different machine learning models and compare their performance: Logistic Regression, k Nearest Neighbours, Decision Tree, Random Forest and Neural Network.

We started by training each of these models with all data and with arbitrary hyperparameters to compare the performance of each of them in this first training round. After that, we chose the best performing ones (considering both accuracy and efficiency) to be fine tuned and try to improve them.

The first training round with dataset 1 revealed that Logistic Regression had a much worse accuracy than the rest of the models and that Neural Networks took extremely long to execute, so these two were not tuned. The models trained with dataset 2 gave all similar results, so all of them were tuned

c) Fine-Tuning and Feature Selection

The first step for tuning our models was to find the optimal hyperparameters that maximize their performance. When using dataset 1, we did that using *RandomizedCV* and only a subset of the data, as otherwise, given the huge size of our dataset, the process would be completely intractable and would take too long to run. We must consider though that this approach can result in hyperparameters that are not the optimal ones completely, as we are not using all the information we should to be sure the process works as we expect it to. However, when using dataset 2 we used *GridSesarchCV* with all the available data, as the size of this dataset makes it possible and viable, and leading to results we can be more confident to be accurate.

After that, we did feature selection. We completed this process with 2 different techniques: Mutual Information (Filtering method) and Recursive Feature Elimination (Wrapper Method). These two processes found two different subsets of features ordered by their importance in our prediction task, which allowed us to retrain our models with only these subsets and see how they behave: if they increased their accuracy, the running time went down, etc.

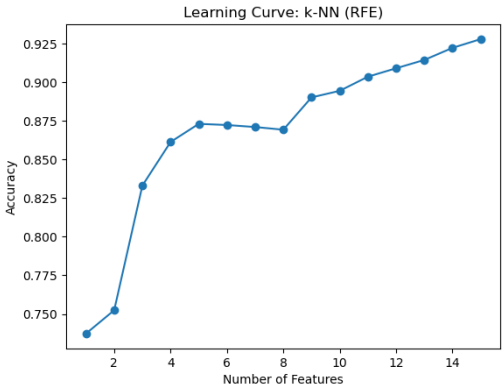
4. RESULTS

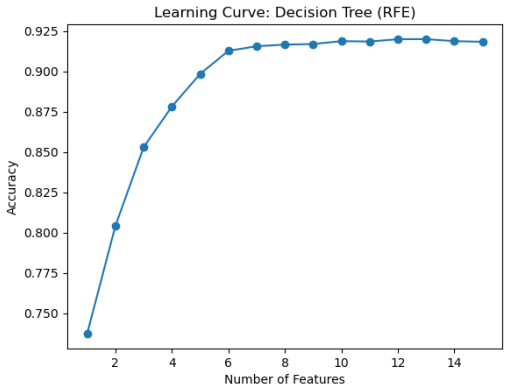
The following table shows the results obtained when testing the models trained with dataset 1 on our testing set:

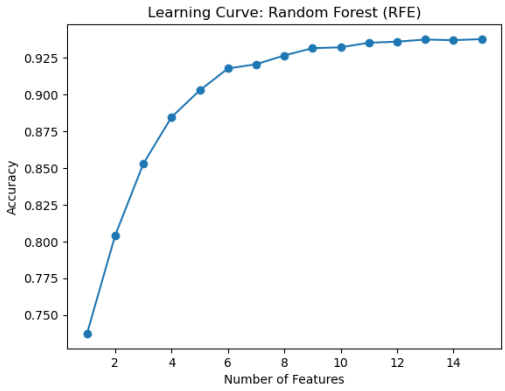
PAU CHAVES (U215102) & ANIOL PETIT (U214019)

Model	1st Training		2nd Training (Hyperparameters)		3rd Training (RFE features)			4th Training (MI features)		
	Accuracy	Run Time	Accuracy	Run Time	Accuracy	Run Time	Features	Accuracy	Run Time	Features
Logistic Regression	78.41%	0.50s	None		None			None		
k-NN	89.40%	137.50s	93.39%	458.33s	92.78%	705.65s	15	92.64%	854.18s	15
Decision tree	91.40%	34.86s	91.86%	5.48s	92.01%	3.18s	13	91.94%	4.87s	11
Random Forest	94.66%	74.75s	90.75%	46.35s	93.76%	47.05s	15	94.22%	44.31s	14
Neural Network	90.72%	306.41s	None		None			None		
kNN + Decision Tree	None	None	92.85%	437.6241s	None			None		

The following plots show the learning curves for the models that were tuned using dataset1 (note that we display only the ones for RFE features as they're very similar to the ones of the MI features, so 1 plot per model is enough to understand the behavior):



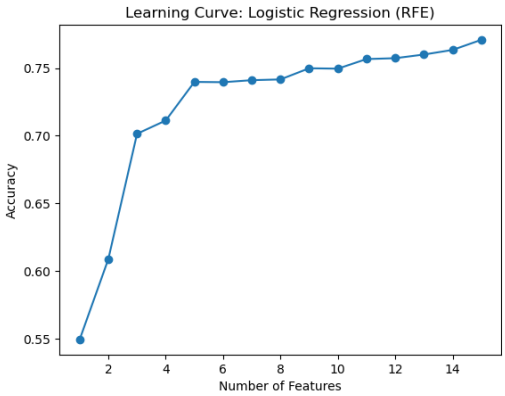


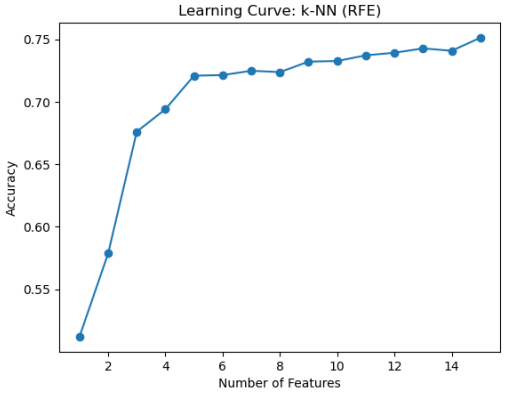


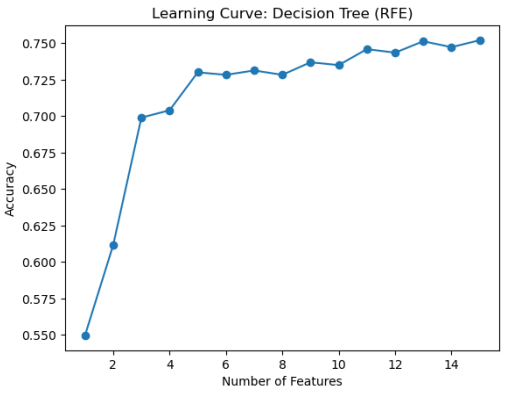
The following table and plots represent the same than the ones above, but this time for the models that were trained using dataset 2.

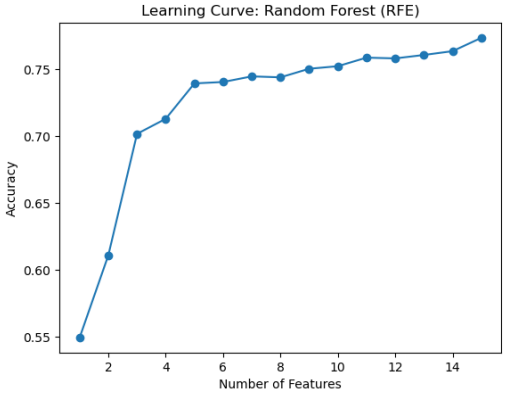
Model	1st Training		2nd Training (Hyperparameters)		3rd Training (RFE features)			4th Training (MI features)		
	Accuracy	Run Time	Accuracy	Run Time	Accuracy	Run Time	Features	Accuracy	Run Time	Features
Logistic Regression	77.03%	0.17s	77.07%	0.15	77.09%	0.03s	15	77.13%	0.03	13
k-NN	72.77%	0.77s	74.98%	3.15s	75.12%	14.48s	15	75.89%	25.05s	12
Decision tree	67.60%	0.21s	75.13%	0.15	75.25%	0.20s	15	75.37%	0.17s	15

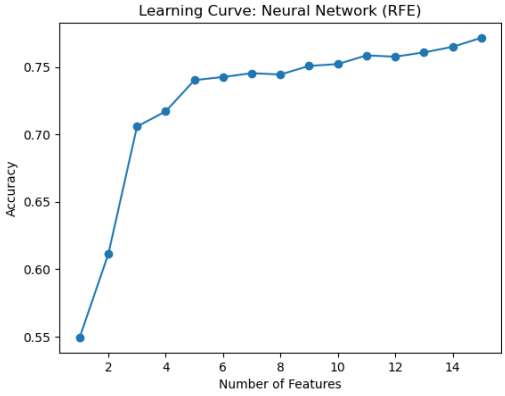
Random Forest	75.80%	3.67s	77.12%	2.77s	77.22%	2.51s	15	77.19%	4.58s	12
Neural Network	75.93%	37.73s	77.14%	7.81s	77.17%	14.48s	15	77.21%	10.27s	14



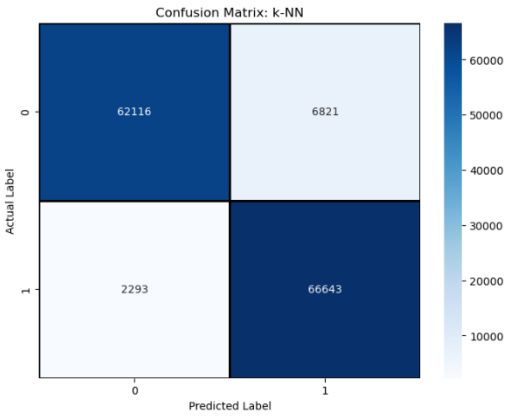


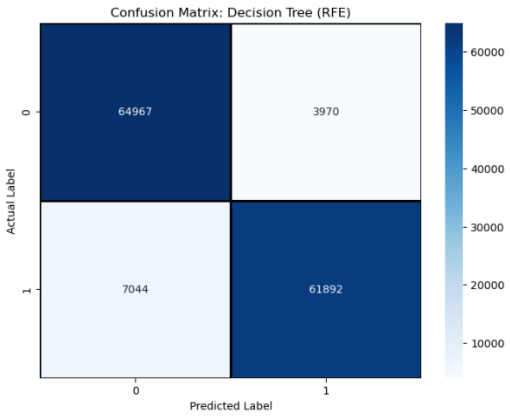


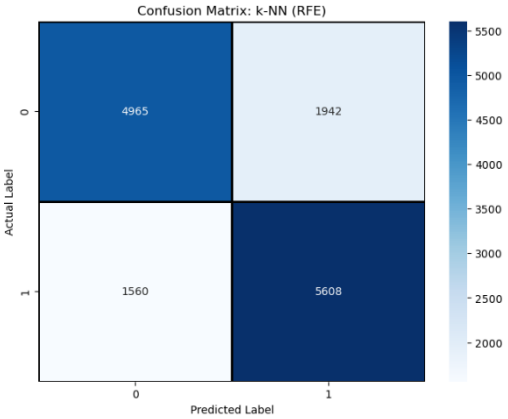




Finally, we'll show some examples of confusion matrices. There's a confusion matrix for each training round and model, so we'll just display a few of them to get an idea of the structure and be able to discuss them:







5. DISCUSSION

- The accuracy levels of models trained with dataset 1 and dataset 2 are a lot different.
    - The models trained with dataset 1 achieved almost all of them more than 90% accuracy at some point of the training, having the peak accuracy at 94.66% with the first Random Forest training. We kind of expected this behaviour, as we already mentioned that generating synthetic samples may cause it to be easier to classify them. However, we have also experienced much larger running times, caused by the huge size of the dataset.
    - The models trained with dataset 2 have much less accuracy, all of them below 80%, with the peak accuracy at 77.22% with Random Forest using RFE features. This was also expected, and is probably a more realistic result, or at least not as biased as the previous ones. Furthermore, these results are achieved in a much more efficient way thanks to the reduced dataset size.
- In all models, those trained with dataset 1 and those with dataset 2, the optimal hyperparameters quite improved their performance in almost all cases. The only exception was Random Forest for dataset 1 didn't improve using hyperparameters, probably because as we said the process using *RandomizedCV* with a subset of the data is not the most accurate process and is likely to make errors. That means that effectively we found in most cases the optimal hyperparameters for each model, which is a key step when fine-tuning a machine learning model to enhance its accuracy. Reducing the number of features didn't increase or decrease the accuracy significantly, just small fluctuations, which may indicate that some features are redundant or meaningless for our task. This is also important as it allows us to enhance efficiency by dropping some features and simplify our models.
- The learning curves show that the top 4 features already can reach quite reasonable levels of accuracy, suggesting that these 4 might be the top predictors for having a heart disease or attack. The rest of the features help to reach slightly higher accuracy values, which means that despite not being the best predictors, they are still important for the task. However, we observe clearly that in most cases the accuracy stabilizes at around 10-15 features, and the plot does not suggest that it would increase by adding more features. That is a clear indicator that there are some features (few ones) that are redundant and don't add much information to predict heart disease or attacks.
- Inspecting both the RFE and MI features, the top predictors for our task seem to be: High Blood Pressure, High Cholesterol, Age, BMI and General Health. Looking at it with perspective, this makes sense regarding the idea we have of causes for heart attacks, and we also detected a subtle pattern between age and heart attacks in data inspection, which confirms our suspicion.

- Finally, we haven't seen any pattern in the mistakes made by our models, as we obtained cases where class 0 was much more misclassified than class 1, cases where it worked the other way round, and cases where both classes were equally misclassified. Hence, we can't extract any important insight about that.
- Overall, we've seen that our models perform reasonably good on our task, being Random Forest the best one in terms of accuracy in any case, which is also a model that performs reasonably good in terms of running time complexity, so it is a very good choice for this task. We found that we can have a quite good insight of whether an individual is likely to have a heart disease or attack with just a few features, but if we want to get a more confident insight we should be using some more data. Finally, we must consider that we have to be careful handling class imbalance with SMOTE, because it might introduce some bias in our prediction, as we've seen with much higher accuracy levels; so it might be a more interesting choice dropping records of the majority class to avoid introducing bias, at expense of a probable lower accuracy because the model will dispose of less samples to be trained and learn from the data.

Note: For more plots, figures and information on the dataset information and visualization, training process and the results check the notebooks uploaded. They are posted with the outputs already, keep them because the running time, especially for notebook 1, is extremely large for some cases as finding the optimal hyperparameters or plotting learning curves. Notebook heart_attack_prediction.ipynb uses dataset 1 and notebook heart_attack_prediction_2.ipynb uses dataset 2.

6. APPENDIX

Below we describe, in much more detail, what each feature in the dataset indicates. We referred to what the author mentioned in the datacard:

- HighBP: Indicates if the person has been told by a health professional that they have High Blood Pressure.
- HighChol: Indicates if the person has been told by a health professional that they have High Blood Cholesterol.
- CholCheck: Cholesterol Check, if the person has their cholesterol levels checked within the last 5 years.
- BMI: Body Mass Index, calculated by dividing the person's weight (in kilogram) by the square of their height (in meters).
- Smoker: Indicates if the person has smoked at least 100 cigarettes.
- Stroke: Indicates if the person has a history of stroke.
- Diabetes: Indicates if the person has a history of diabetes, or currently in pre-diabetes, or suffers from either type of diabetes.

- PhysActivity: Indicates if the person has some form of physical activity in their day-to-day routine.
- Fruits: Indicates if the person consumes 1 or more fruit(s) daily.
- Veggies: Indicates if the person consumes 1 or more vegetable(s) daily.
- HvyAlcoholConsump: Indicates if the person has more than 14 drinks per week.
- AnyHealthcare: Indicates if the person has any form of health insurance.
- NoDocbcCost: Indicates if the person wanted to visit a doctor within the past 1 year but couldn't, due to cost.
- GenHlth: Indicates the person's response to how well is their general health, ranging from 1 (excellent) to 5 (poor).
- Menthlth: Indicates the number of days, within the past 30 days that the person had bad mental health.
- PhysHlth: Indicates the number of days, within the past 30 days that the person had bad physical health.
- DiffWalk: Indicates if the person has difficulty while walking or climbing stairs.
- Sex: Indicates the gender of the person, where 0 is female and 1 is male.
- Age: Indicates the age class of the person, where 1 is 18 years to 24 years up till 13 which is 80 years or older, each interval between has a 5-year increment.
- Education: Indicates the highest year of school completed, with 0 being never attended or kindergarten only and 6 being, having attended 4 years of college or more.
  Income: Indicates the total household income, ranging from 1 (at least $10,000) to 6 ($75,000+)