CCT COLEGE DUBLIN

HIGHER DIPLOMA IN SCIENCE IN DATA ANALYTICS FOR BUSINESS

**EDUARDO SALDIVAR HERNANDEZ, ID 2023077**

**LAÍS VIEIRA MARQUES AMENO – ID 2023101**

**LÍNIKA VICENTE FERREIRA DE ALMEIDA, ID 2023045**

**ASSIST INSURANCE COMPANY WITH PERSONALIZED HEALTHY PLANS BASED ON HEART DISEASE PARAMETERS**

**STRATEGIC THINKING – CA 2**

**PROFESSOR: Dr. James Garza**

Dublin, Ireland

2023

**Table of Contents**

[1 INTRODUCTION 4](#_Toc150961932)

[2 DATA UNDERSTANDING 5](#_Toc150961933)

[**2.1** **Collect initial data** 6](#_Toc150961934)

[**2.2** **Exploratory Data Analyse (EDA)** 6](#_Toc150961935)

[**2.2.1 Correlation** 10](#_Toc150961936)

[3. DATA PREPARATION 10](#_Toc150961937)

[**3.1Cleaning** 11](#_Toc150961938)

[**3.2Transform data type** 12](#_Toc150961939)

[**3.3Normality Test** 12](#_Toc150961940)

[**3.4Logarithmical transformation to normalize the data** 12](#_Toc150961941)

[4. MODELLING 13](#_Toc150961942)

[**4.1 Modelling - First Semester** 13](#_Toc150961943)

[**4.2 Modelling after changes - Second Semester** 17](#_Toc150961944)

[4.2.1 SelectKBest VS PCA 17](#_Toc150961945)

[4.2.2 Modeling 19](#_Toc150961946)

[4.2.3 Under sampling 20](#_Toc150961947)

[4.2.4 Linear discriminant analysis (LDA) 21](#_Toc150961948)

[4.2.5 SHAP 23](#_Toc150961949)

[5. EVALUATION 27](#_Toc150961950)

[6. DEPLOYMENT 29](#_Toc150961951)

[REFERENCE 31](#_Toc150961952)

FIGURE LIST

[Figure 1- Sumary information](#_Toc150102059) 7

[Figure 2- Missing values 8](#_Toc150102060)

[Figure 3- Hearth Stroke Cases](#_Toc150102061) 9

[Figure 4 - Hearth Stroke per Gender 9](#_Toc150102062)

[Figure 5 - Heath stroke for smokers………………………………………………………………..](#_Toc150102063) 9

[Figure 6 - Hearth Stroke per Gender](#_Toc150102064) 9

[Figure 7- Heath stroke compared with hypertension………………………………………….….](#_Toc150102065)10

[Figure 8 - Heap Map of correlation………………………………………………………………](#_Toc150102066) 10

[Figure 9 - Histogram for the continuous data](#_Toc150102067) 11

[Figure 10 - Number of components to explain variance](#_Toc150102068) 14

[Figure 11 - Data after apply Smote……………………………………………………………...](#_Toc150102069) 15

[Figure 12- Confusion Matrix for SVM algorithm](#_Toc150102070) 16

[Figure 13 - Top 10 important featutes](#_Toc150102071) 18

[Figure 14 - Classification models](#_Toc150102072) 19

[Figure 15 - Balance classes……………………………………………………………………...](#_Toc150102073) 20

[Figure 16 – Classification report](#_Toc150102074) 21

[Figure 17 - Confusion matrix](#_Toc150102075) 22

[Figure 18 - ROC Curves for LDA………………………………………………………………](#_Toc150102076) 22

[Figure 19 – Shap Value](#_Toc150102074) 24

[Figure 20 - Shap Value 1](#_Toc150102075) 25

[Figure 21 - Main features and shap magnitude …………………………………………………](#_Toc150102076) 25

[Figure 22 - Shap value impact on model output………………………………………………](#_Toc150102076) 26

# INTRODUCTION

The CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology is being used to approach a specific goal and business scenario. The goal is to assist an insurance company in providing personalized healthy plans to its customers based on their lifestyle choices, hobbies, and daily routine.

By providing such personalized plans, the insurance company can conserve resources and focus on the core objective of reducing the costs of their customers. The insurance provider can create customized policies for each customer by mining and analyzing customer data to find patterns and trends in consumer behavior. This individualized strategy can aid the insurance company in retaining clients and boosting client satisfaction, creating a win-win scenario for both the client and the business.

The objective of this project is improving health outcomes and reduce health care coast for the company and prevent any eventual dispends with patients that have heart stroke conditions. By doing this the company will be benefit by differentiating from them competitors.

To build the model, we are taking in the consideration most of the features the people whose had heart disease.

The dataset is composed of the parameters of their life style, hearth stroke, level of education, gender, and some health parameters.

The requirements for this project are:

* Information about people who had heart stroke.
* Cleaning the data to perform a Machine Learning. Model
* Prediction of the customer, if this person is a candidate to have a heart disease.

Considering the purpose of this project is improve the project from the first semester, we did some changes as is possible to see in the table. In this semester we reorganized all the steps to improve the model and let everything more clearly. We changed the plots including all titles, clean the code and let it more organized, the code was more commented. In the data preparation we applied normalization and Log transformation.

# DATA UNDERSTANDING

The second phase of the CRISP-DM Methodology is Data Understanding, that can be defined as “the knowledge that you have about the data, the needs that the data will satisfy, its content and location” (Ladley, 2016).

In this phase, it involves the preparation and understanding of our dataset. This is a very important step because it will help us prevent any unexpected problem or situation with all of our data and its values.

We selected the Hearth Stroke dataset for this project, which includes clinical data on individuals with and without this illness. Given that the firm we serve is a health insurer that offers insurance policies, this dataset makes sense for our business study.

Considering that the company is a health insurance company and wants to reduce customer expenses by creating a customized health plan that identifies which possible illnesses the customer may have. We will develop a project where the company can previously identify this before closing a contract with the client.

We will take as a starting point the analysis of a predisposition for this type of serious illness such as hearth stroke. And our main goal is to improve health outcomes and reduce healthcare costs for the company, and avoid possible costs with patients that might have this disease.

For this purpose we are going to apply three different classification models: KNN, SVM, Random Forest. This coice was made due:

KNN - Due to its ability to cluster similar data together, is able to recognize patterns in data pertaining to risk factors that result in strokes. It can demonstrate how clients with risk indicators are similar to one another and assist in locating comparable clients, allowing insurance plans to be customized.

SVM - divides clients into distinct risk classes in a linear fashion according to risk variables. Additionally, it clearly delineates the limit between clients at high and low risk.

Random Forest - is capable of deciphering intricate patterns and recognizing clients with peculiar traits that might pose an uncommon risk. able to identify the traits that are most important in predicting the likelihood of a stroke and aids in determining the overall risk connected to various client profiles.

### **Collect initial data**

The dataset was collected in the kaggle website. This data provides information about people who had or not had a heart stroke and some information about risk factors that may contribute to this disease. Using the information at hand, we attempt to comprehend how various clinical characteristics and risk factors affect the incidence of heart stroke. Clinical characteristics and risk factors include diabetes prevalence, gender, age, cholesterol, glucose, BMI, and smoking frequency.

Upon downloading the dataset, we meticulously examined the accompanying documentation to grasp the data's structure, format, and any inherent biases or limitations that could impact our analysis.

### **Exploratory Data Analyse (EDA)**

The Hearth Stroke dataset have 16 columns and the meaning of each column in the data is as below:

1. Gender: The gender of the patient

2. Age: The age of the patient

3. Education: The studies of the patient

4. currentSmoker: The patient is a current smoker? (No = 0, Yes = 1)

5. cigsPerDay: Quantity of cigarretes that the patient smokes each day

6. BPMeds: The patient use blood pressure medication? (No = 0, Yes = 1)

7. prevalentStroke: The patient had a stroke recently? (No = 0, Yes = 1)

8. prevalentHyp: The patient had hypertension problems recently? (No = 0, Yes = 1)

9. diabetes: The patient have diabetes? (No = 0, Yes = 1)

10. totChol: Total of cholesterol of the patient

11. sysBP: Systolic blood pressure of the patient

12. diaBP: Diastolic blood pressure of the patient

13. BMI: Body mass index of the patient

14. heartRate: Heart rate of the patient

15. glucose: Glucose of patient

16. Heart\_stroke: The patient will have a heart\_stroke? (No = 0, Yes = 1)

We began working with the pandas library for reading and storing data in a variable called 'DF.' Pandas is a fast, powerful, flexible, and easy-to-use open-source data analysis and manipulation tool, built on top of the Python programming language (Pandas, 2018).

In the EDA we are going to do some assumptions, do a comprehension of the dataset and check missing values. We performed the 'head' command to get an overview of the collected data and observe the first rows. Using the head() method we will be able to see our first 5 rows of our dataset. This method is useful when you want to take a look of the dataset and have a quick view to check If we have some unusual values.

Using the info() python method, we are able to see some details of our features, like if we have some Non-null and the type of data that we have. Here it’s important to check specially the type of data, sometimes it’s possible that a column which is numerical give us a Dtype = object. This means that there is something wrong in our data and we need to clean some values.

In the Figure 1 is we can see, we have normal features like the gender, age and education of the person. The most important features in this dataset are the ones that are more dedicated to the health and actions that the person have, for example, currentSmoker, cigsPerDay, diabetes, glucose.

In a previous analysis is possible to notice in data.info this dataset is distributed in 4238 entries and 16 columns. And there are data with different type, being float64(8), int64(4), object(4).

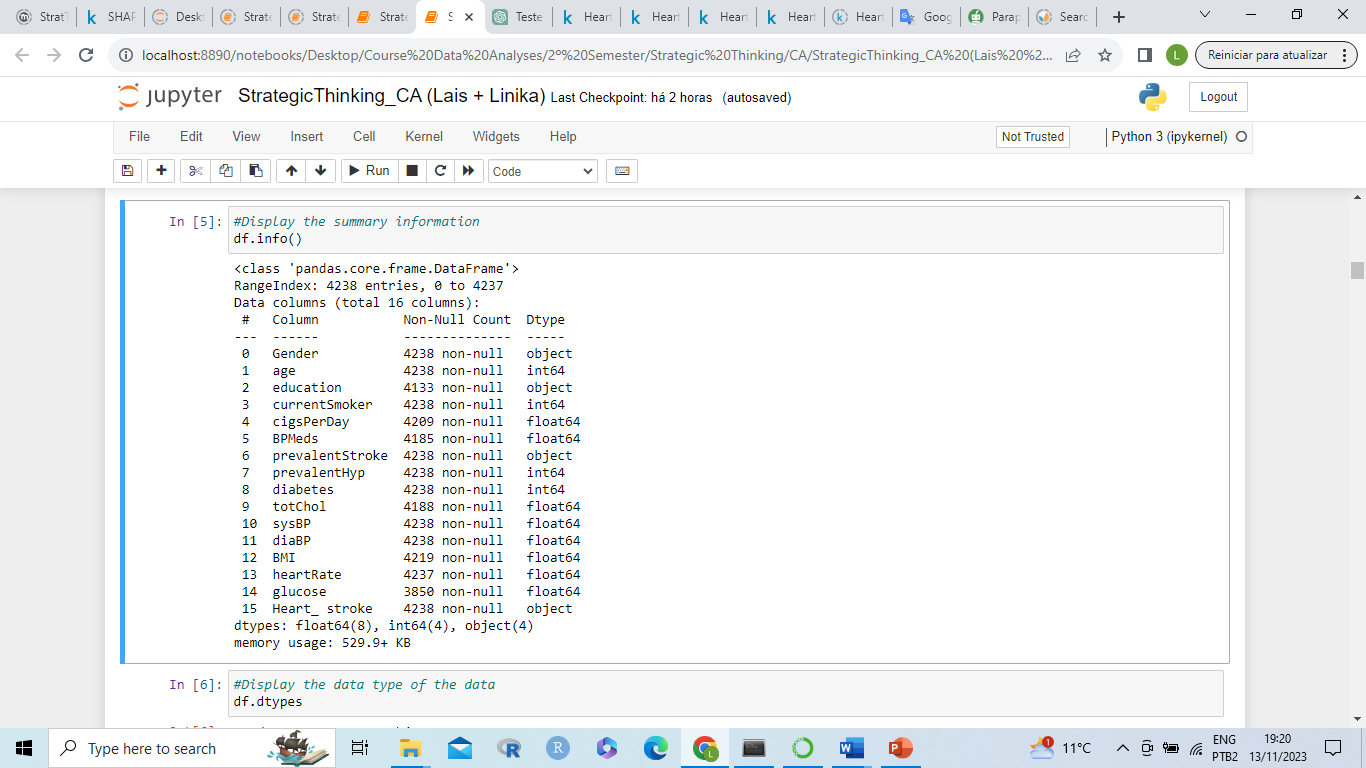


Figure 1 – Sumary information

For identify the number of data per column was applied the nunique() method. And for identify the number of null values per column was applied isnull().sum.

In df.type we observed there are float data (cigsPerDay,BPMeds, totChol, sysBP, diaBP, BMI, heartRate, glucose), int64 data (age, currentSmoker, prevalentHyp, diabetes), object data (Gender, education, prevalentStroke, Heart\_ stroke).

In isnull().sum() Figure 2 shows the columns education, cigsPerDay, BPMeds, totChol, BMI, heartRate, glucose presents considerable number of missing values in this dataset.

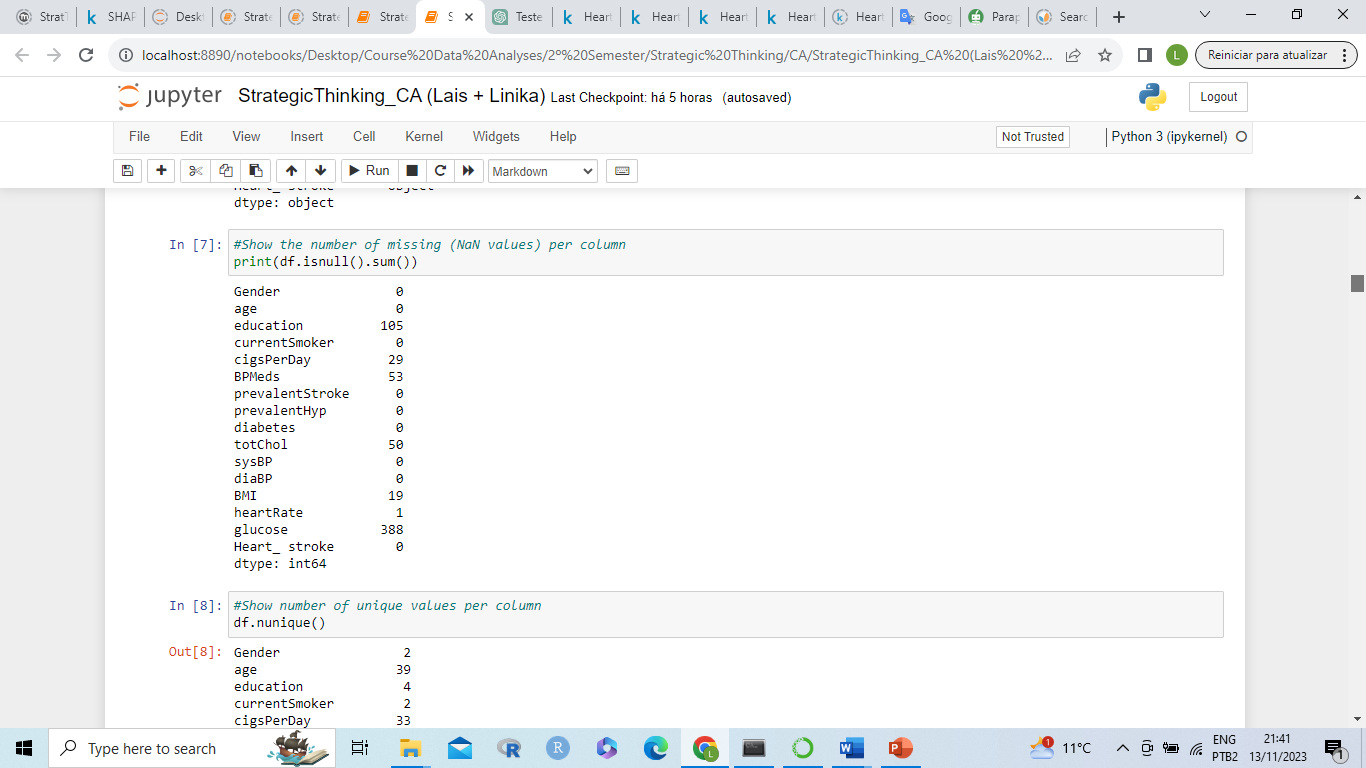


Figure 2. Missing values

After making sure that there were no other unusual values in our dataset, we decided that is time to make a clean in our columns. The column ‘education’ was dropped once it do not make any difference for this analysis.

According to our analysis so far, it seems our dataset contains 540 null values in total. We will proceed to determine how many null values are by column. The NA are distributed in the columns cigsPerDay (continuous data), BPMeds (continuous data), totChol (continuous data), BMI (continuous data), heartRate (object), glucose (continuous data).

In the Figure 3 show the number of patients with hearth stroke. The number of patients that did not have a heart stroke is 84.8% it is way greater compared with patients that had a hearth stroke 15.2%.

In this plot is also possible to observe our column is imbalance, considering there is a relevant difference between the number of cases and it might bring some inconsistence when apply the models. We are going to add some synthetic data in our training to balance our target variable.

When compared the number of cases per gender in Figure 4 there are more Male cases of hearth stroke being 343 cases versus 301 cases for women.

|  |  |
| --- | --- |
|  |  |
| Figure 3. Hearth Stroke Cases | Figure 4 – Hearth Stroke per Gender |

In this Figure 5 shows the number of heart stroke for No smoker and Smoker, there is slightly different between them which people that had the habit of smoke were more susceptible to have hearth stroke being 333 cases of heart stroke versos 311 case for no smokers. In Figure 6 patients which did not have diabetes had a hight level of heart stroke, and almost all of the patients that have diabetes are more likeable to get a heart stroke.

|  |  |
| --- | --- |
|  |  |
| Figure 5 – Heath stroke for smokers | Figure 6 – Hearth Stroke per Gender |

In the Figure 7 the number of patients without hypertension are greater, but it seems that having hypertension might contribute for heart stroke.

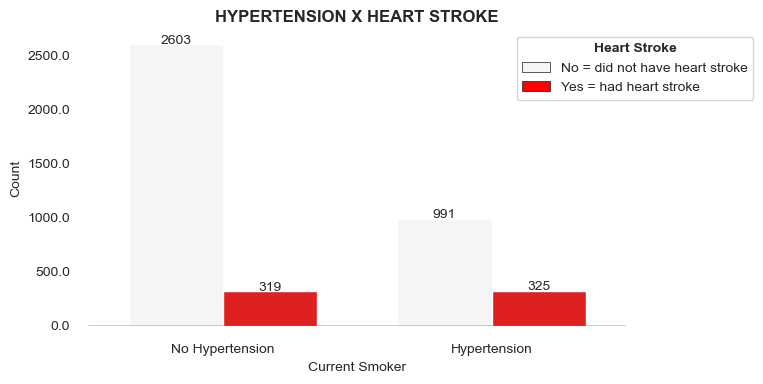


Figure 7 – Heath stroke compared with hypertension

### **2.2.1 Correlation**

Were applied Person Correlation for the continuous data to understand the strength and direction of the linear relationship between pairs of continuous variables. We can observe diaBP and sysBP are highter correlated.

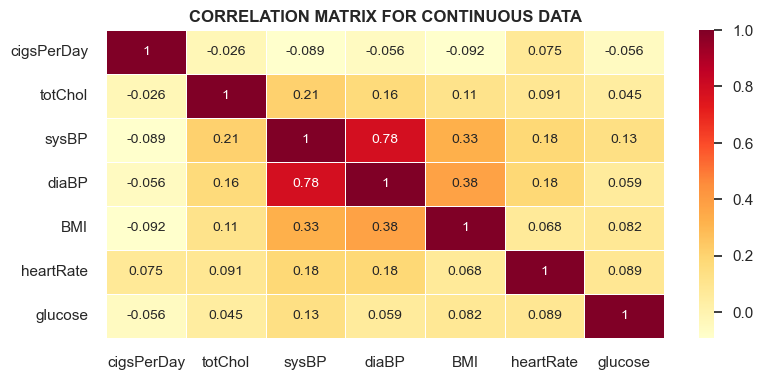


Figure 8 – Heap Map of correlation

# 3. DATA PREPARATION

Data preparation is a crucial step in any data analysis or machine learning task. It entails cleaning and transforming raw data into a format that can be easily analyzed and utilized to train a model. The purpose of data preparation is to guarantee that the data is consistent, accurate, and devoid of errors or missing information (Pro, project 2022).

## **Cleaning**

For decide what is the best approach to fill missing values, was initially plotted a histogram (Figure 9) for the continuous data to see how the data are distributed.

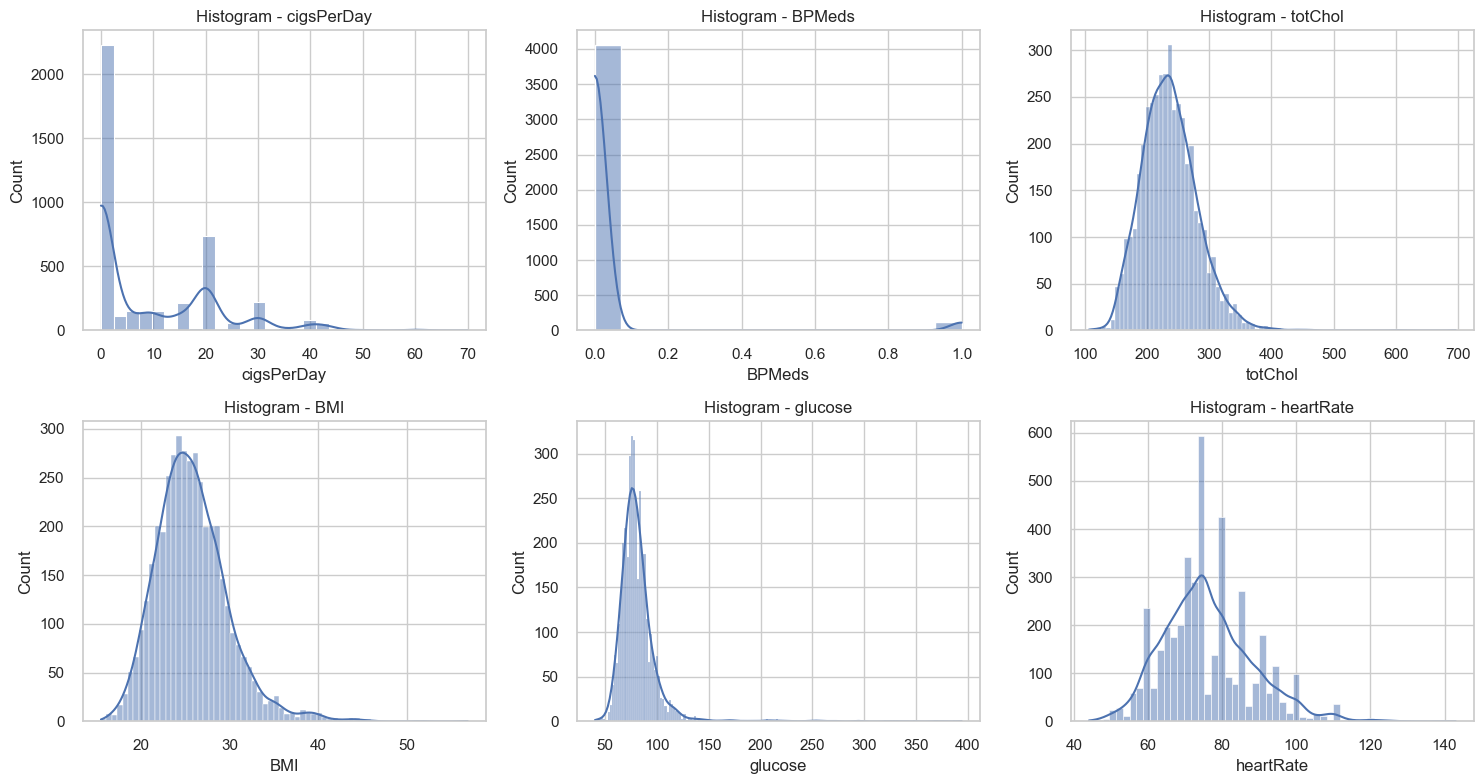


Figure 9 – Histogram for the continuous data

In the plot cigsPerDay there are some outliers. We consider that is possible that some people smokes 40 cigarettes or more by day, so we don’t see why we should think that this is some finger mistake. We will not remove them.

In totChol column have outliers as well. We find it possible for a person to have that much of cholesterol so we will not delete them. We can see this column seems has a normal distribution. Most of the patients have between 200 and 300 of cholesterol. BMI and glucose also appear to have a normal distribution.

The normal distribution is a continuous probability distribution that is symmetrical around its mean, most of the observations cluster around the central peak, and the probabilities for values further away from the mean taper off equally in both directions. Extreme values in both tails of the distribution are similarly unlikely. While the normal distribution is symmetrical, not all symmetrical distributions are normal. This visualization shows three columns have histogram exhibiting a bell-shaped and other two with a skewness shape.

For data with bell-shaped the method to fill NA gonna be Mean. And for the columns with skewness shaped we are going to fill null values with median. For the categorical columns, the missing values are going to fill with the mode of each column once it is usual for categorical data.

## **Transform data type**

In this dataset there is categorical data, binary data and continuous data. Before perform the models, the categorical data will be transformed in binary data. It avoid misinterpretations, unwanted weightings and it is more compatibility with the model we are going to perform KNN and SVM. First the data was separated conform the datatype (for normalization the continuous data), than applied hot ecoding to organize all the dataset in a numerical format before perform the models

* k-NN: calculate the separation between two points, so this binary encoding makes distance calculations more efficient.
* SVM: the hyperplane separation plane is found. The binary representation makes it easier to define distinct decision margins. Random Forest: is made up of decision trees where in each, the binary representation is more appropriate for decision making.

## **Normality Test**

In the continuous data were applied Shapiro-Wilk Test with level of significance 0.05. This test was chose once it normalize the data and work properly for moderate size of dataset.

The null hypothesis (H0) of the Shapiro-Wilk test states that the data follow a normal distribution. Therefore, if the test's value-p result is less than a certain significance level 0.05, we reject the null hypothesis and come to the conclusion that the data do not follow a normal distribution.

After apply Shapiro-Wilker Test the data demonstrated do not have normality considering the p value is < 0.05. For the purpose to give more stability in our modelling we are going to try normalising the data with Logarithmical transformation.

## **Logarithmical transformation to normalize the data**

Log transformations were applied, and it was found that the data was not normal. Since these transformations did not achieve the desired normal distribution, we plan to utilize MinMaxScaler for scaling the data. This approach add the data into a range between 0 and 1, it allow to adjust the data but not necessary normalise them. This step is taken to prepare the data for algorithms that are sensitive to feature scales.

# MODELLING

## **Modelling - First Semester**

Two alternative approaches are being used for this modelling; one uses PCA and other does not use PCA at all. Reducing the amount of features and making the model more predictable are the objectives of PCA in this application.

For this purpose, for which the dataset represents a classification problem, the algorithms Logistic Regression, SVM and KNN were used due to their functionalities for this approach.

Logistic Regression was chosen for the reason it predicts binary outcomes and in this dataset the heart stroke that represent the dependent variable is a binary data. This model is also very interpretable because it generates simple coefficients. Logistic Regression also works with limited datasets; because we are working with a disease dataset, it can be difficult to acquire data. This model's simplicity makes it simple to learn and apply on datasets with few samples.

SVM algorithm is to find a hyperplane in an N-dimensional space, where N is the number of features that distinctly classifies the data points. For this study was applied SVM, considering it is a machine-learning method useful for classification. The method was applied to predict heart stroke considering the parameters of heart stroke dataset.

KNN K-Nearest Neighbor was used to categorize and predict outcomes for heart stroke patients. By completing these procedures, a more reliable model for predicting cardiac events was created. This algorithm merely maintains the data and uses similarity to classify a new data point.

In the first method, the PCA was applied to reduce the amount of features. Then were tested tree different models for classification problem: Logistic Linear Regression, SVM, and KNN.

Considering in this dataset the number of people with heart stroke were imbalance, Smote algorithm was used for stabilise the model increasing the representation of the minority class in the dataset, which can improve the performance of machine learning models. And for standardization the MinMax method was used to scale the dataset.

In the second method, we utilized the models of logistic linear regression, SVM, and KNN instead of PCA to decrease the features. SMOTE technique will be used to balance the data, and MinMax and StandardScaler will be used to scale it.

Different challenges have been applied to improve the accuracy of the models as applied different models for classifications problems, Smote technique for balance the data, different method of standarization, and cross validation technique.

As a result of the first method with use o PCA, was observed 2 components explain more than 80% of the data, however we decided to use 6 components for improve the variance of the model and also because Logistic Regression require at least 6 features for work properly.

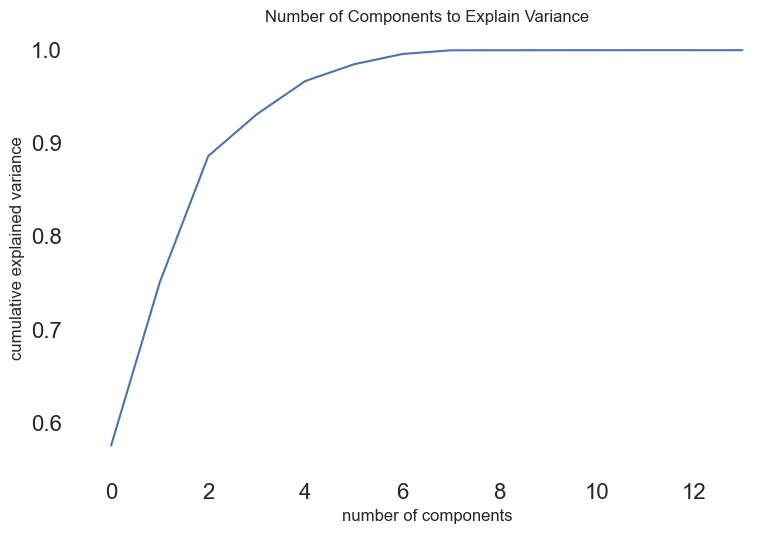


Figure 10 - Number of components to explain variance

Smote was applied to balance the dataset, once the number of people with Heart stroke was smaller than people with no heart stroke. This way we ensure that the model will be trained equally on both classes, improving generalization and prediction performance.

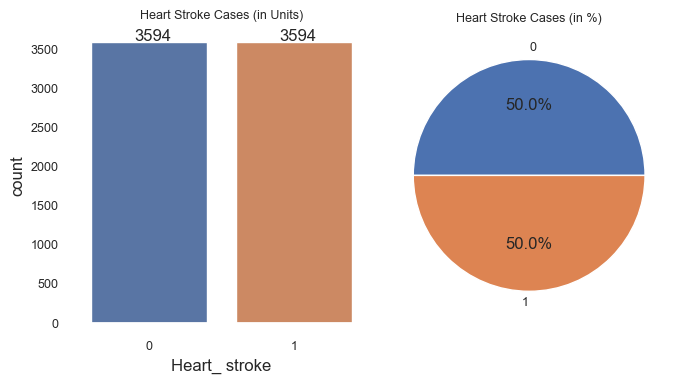


Figure 11 - Data after apply Smote

The data were scaled using the Method MinMax and proceed with Splitting the data in training 70% and test 30%. This function splits arrays or matrices into random subsets for train and test data that minimize the potential for bias in your evaluation and validation process and also avoid model overfitting.

In the table below it is possible to visualize the accuracy presented by each model applied to this analysis after use of PCA.

|  |  |
| --- | --- |
| Models | Accuracy |
| Logistic Regression | 65.27% |
| SVM | 68% |
| KNN | 79.88% |

After apply PCA to reduce the features followed by Logistic Regression the model showed accuracy of 65.27%. For SVM the accuracy was 68%. And for KNN the accuracy was 79.88%.

It is possible to see in the confusion matrix that applying the SVM algorithm model performs admirably in the classification of the two classes 0 and 1. The requirement for both classes is relatively good, the model predicted 0.70 for class 0.0 and 0.66 for class 1.0, being the accuracy 68%.

precision recall f1-score support

0.0 0.70 0.63 0.66 1090

1.0 0.66 0.73 0.69 1067

accuracy 0.68 2157

macro avg 0.68 0.68 0.68 2157

weighted avg 0.68 0.68 0.68 2157

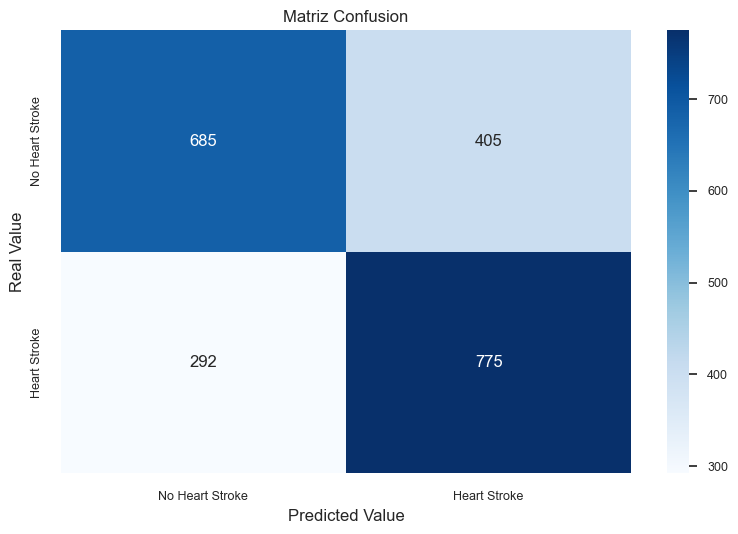


Figure 12. Confusion Matrix for SVM algorithm

**WITHOUT PCA**

In the data preparation step of this dataset, was observed that some columns in this data show a normal distribution, being for this reason used also Standard Scale method. Other reason is for some models as SVM and KNN this method can improve the performance and accuracy by ensuring that all features contribute equally to the learning process.

In the table below it is possible to visualize the accuracy presented by each model applied to this analysis without use PCA. And considering different type of method for scale data, MinMax and Standart Scale.

|  |  |  |
| --- | --- | --- |
|  | MinMax | Standart scale |
| Logistic Regression | 63.97% | 63.97% |
| SVM | 65.50% | 93% |
| KNN | 61.29% | 61.89% |

Applying MinMax scale the Logistic Regression model kept the same performance with both methodology for scale data, however SVM model increase the accuracy from 65.50% to 93% after application of Standart scale and grid search. KNN did not show much variation.

## **Modelling after changes - Second Semester**

# SelectKBest VS PCA

The SelectKBest is a feature selection technique used in machine learning tasks. It assesses the relevance of features to the target variable and selects the best features using a statistical test, in our case, the ANOVA f\_classif, as we are dealing with a classification problem.

On the other hand, PCA is a dimensionality reduction technique that transforms the original features into a new set of features called principal components. Each principal component is a linear combination of the original features.

In this project, we initially used PCA to reduce dimensionality and capture most of the variance in the original dataset to enhance our model's performance.

At this stage of the project, we opted for a new approach using SelectKBest to specifically choose the most informative features from the dataset.

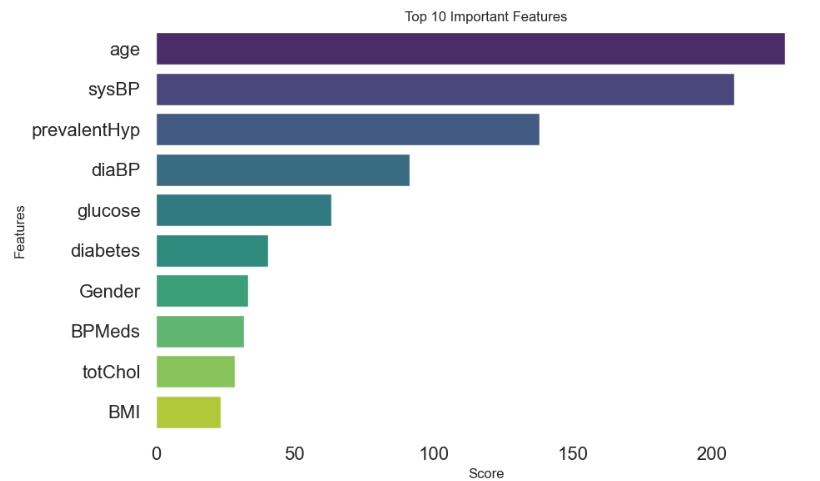


Figure 13 – Top 10 important featutes

So, we obtained the 10 most informative features from the dataset and proceeded to define the independent features based on K=10, as shown in the figure above.

The columns age and sysBP have the highest scores; however, the decision on which features to keep depends on various factors such as the model's objective, interpretability of features, and the possibility of overfitting.

We could keep the five most important features by choosing the ones with the top five scores, but we are considering the context of the business problem. Probably, features with lower scores may be crucial for the interpretability or relevance of the model since we are dealing with a dataset related to heart attack. Considering this context, we will keep all 10 features.

In comparison with the PCA we performed in the initial project, SelectKBest did not show much difference in our final result, although the columns exhibited significant relevance.

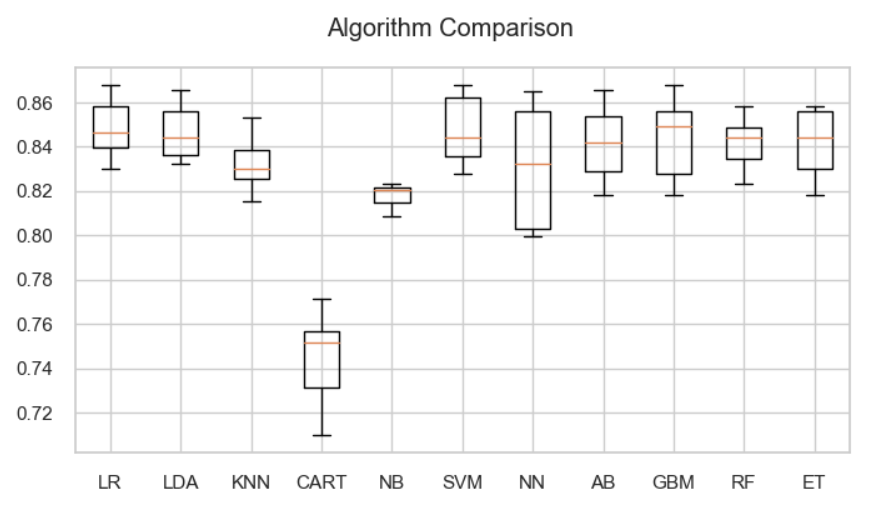
We conclude that both methods can reduce the dimensionality of features with different approaches. PCA creates a new dataset while preserving variance with the principal component, and SelectKBest selects only the best features based on a statistical calculation. Both methods are highly effective when dealing with a substantial number of columns.

# Modeling

We defined the independent columns with the variable "X" based on SelectKBest, and our dependent column with the variable "y" respectively: age, sysBP, prevalentHyp, diaBP, glucose, diabetes, Gender, BPMeds, totChol, BMI, and the target: Heart\_Stroke.

The next step was to split the data into training and testing sets, with 70% for training and 30% for testing.

We used a series of classification models in this first part to evaluate which model would achieve the best accuracy.



14 – Classification models

As we can see in the figure above, SVM achieved the best accuracy of 0.8479. Therefore, we will use SVM to further evaluate on the test set. In the confusion matrix, we can assess the model's performance more comprehensively between the classes.

The model appears to perform well in predicting class 0 (negative cases) but poorly in predicting class 1 (positive cases). The low recall for class 1 indicates that the model is missing a significant number of actual positive instances. The F1-score for class 1 is also very low, suggesting that the model struggles to balance precision and recall for the positive class.

# Under sampling

We are working with a dataset where our goal is to classify whether it is a heart attack or not, respectively 1 and 0, with the majority class being 0.

In the initial project, we used the oversampling process by adopting SMOTE to create synthetic data in the minority class. We are aware that this process needs to be done with caution, as it can add noise to the dataset or lead to overfitting when adjusting the model. Now, we have opted to adopt undersampling, which aims to equalize the two classes by reducing the class with more samples, class 0, in our project.

We can say that undersampling is a technique used to balance the classes in the training set for better model learning. However, during this process, this technique may randomly exclude data from the majority class that could be important for the final machine learning outcome, possibly causing bias.

We proceed by creating a new training set balanced between classes 0 and 1.



15 – Balance classes

Once the classes are balanced in the training set, we ran the classification models again to identify which one would have the best fit. LDA had the best fit at 65%. We observed a drop in accuracy across all models after balancing the training set, and one possible cause for this performance decrease is that by balancing the training set, we may have removed important instances from the majority class, resulting in the loss of valuable information for model training. Nevertheless, we will proceed with LDA and adopt some metrics to check if we can improve the model fit.

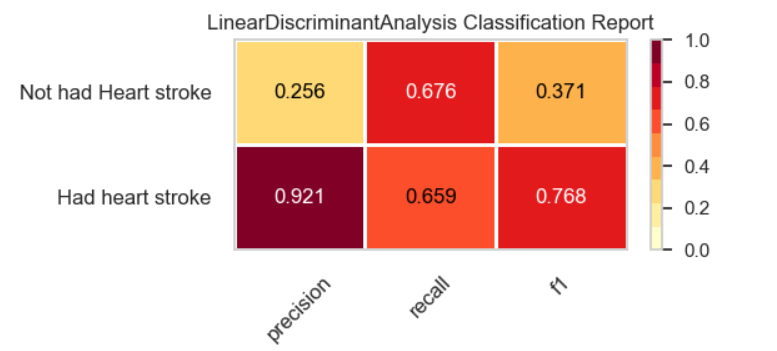
# Linear discriminant analysis (LDA)

LDA makes predictions by estimating the probability that a new set of inputs belongs to each class. The class with the highest probability is the output class, and a prediction is made.

The model uses Bayes' Theorem to estimate the probabilities. Briefly, Bayes' Theorem can be used to estimate the probability of the output class (k) given the input (x) using the probability of each class and the probability of the data belonging to each class (Brownlee, 2016).

We tuned the LDA model through grid search, which involves optimizing several key parameters. The choice of an appropriate solver is crucial for efficient computation, and the application of a shrinkage method is employed to enhance model performance. Determining the optimal number of components ensures effective dimensionality reduction, and setting the tolerance for convergence is essential for controlling the precision of the model during the optimization process.

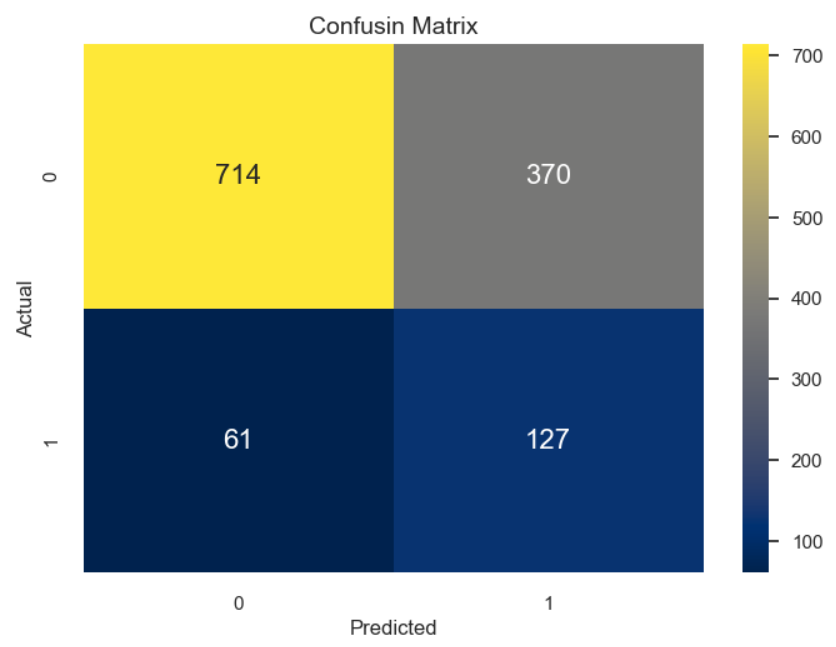
After running the grid search, we obtained the following parameters: shrinkage='auto', solver='lsqr,' and adjusted the model according to the obtained parameters. To evaluate the model, we created a classification report, and the results are as follows:



16 – Classification report

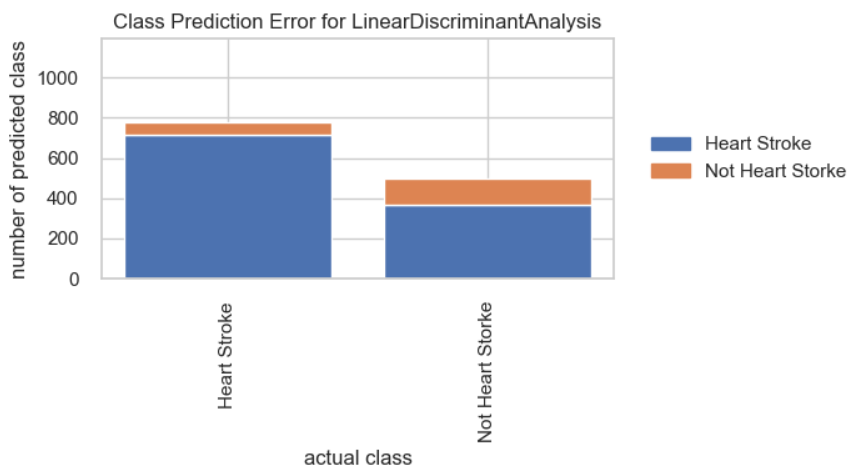
The precision for Class 1 is relatively low, indicating that many positive predictions for Class 1 are incorrect. The F1-score is moderate, influenced by the imbalance in the classes. However, the recall metrics for both classes are similar, indicating that the model is not favoring one specific class over the other.

Following that, we created the confusion matrix to describe the model's performance on the test dataset by comparing the model's predictions with the actual labels.



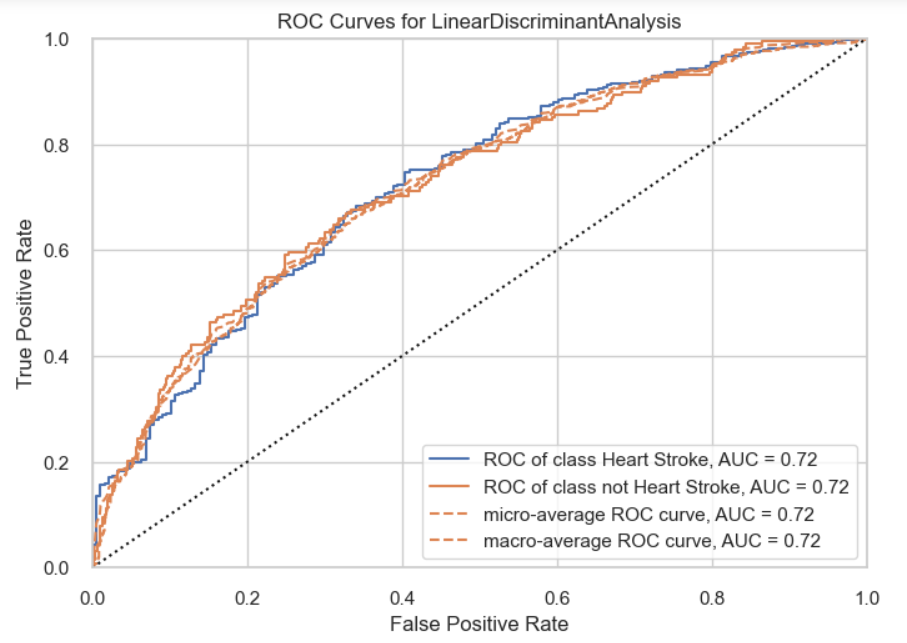
17 – Confusion matrix

True Positive (TP): 127 (Predicted as 1, and the actual label is also 1). True Negative (TN): 714 (Predicted as 0, and the actual label is also 0). False Positive (FP): 370 (Predicted as 1, but the actual label is 0). False Negative (FN): 61 (Predicted as 0, but the actual label is 1).



18 – Class prediction error

We still observe the model better adjusting to the majority class "Not Heart Stroke." We continued evaluating the model's performance across classes using ROC AUC to comprehend the false positive rate in relation to the true positive rate at different cutoff points.



18 – ROC Curves for LDA

A ROC AUC between 0.5 to 0.7 indicates weak to moderate performance. The model is discerning between classes, but there is room for improvement. Therefore, a value of 0.72 suggests that the model has a moderate ability to discriminate between classes but can still be enhanced.

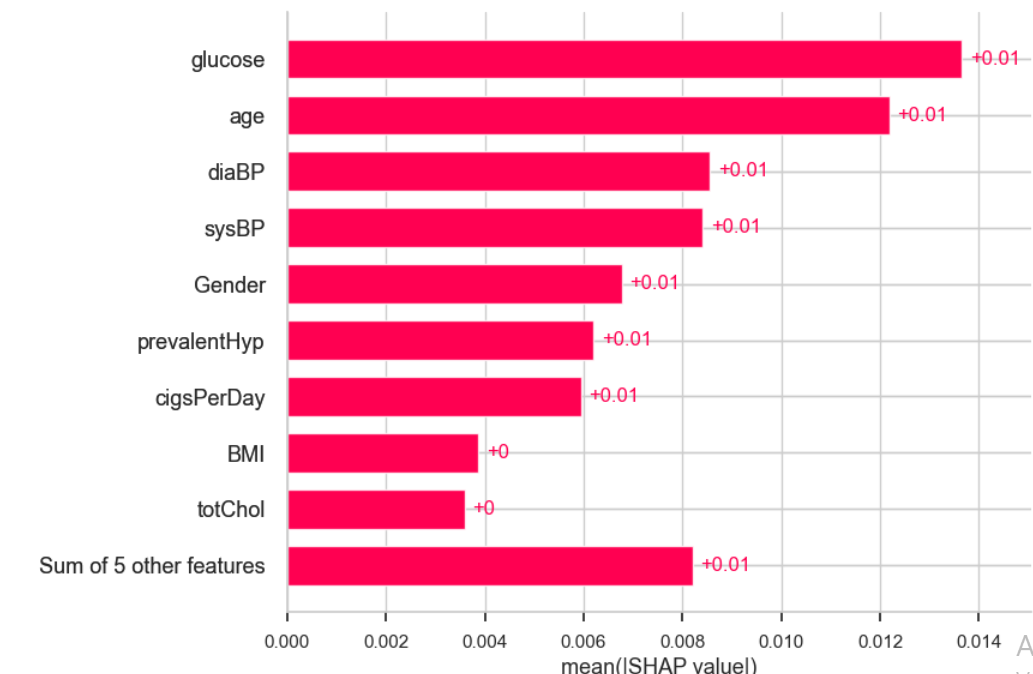
# SHAP

In order to use SHAP, we need a tree-based model so we decide to apply the random forest classification. A random forest is a supervised algorithm that uses an ensemble learning method consisting of a multitude of decision trees, the output of which is the consensus of the best answer to the problem. A random forest is nothing more than a series of decision trees with their findings combined into a single final result. They are so powerful because of their capability to reduce overfitting without massively increasing error due to bias.

SHAP (SHapley Additive exPlanations) values are a way to explain the output of any machine learning model. It uses a game theoretic approach that measures each player's contribution to the final outcome. In machine learning, each feature is assigned an importance value representing its contribution to the model's output.

SHAP values show how each feature affects each final prediction, the significance of each feature compared to others, and the model's reliance on the interaction between features.

The reason we decide to apply SHAP system here is mainly to see and compare if the best features that feature selection gives us is the same as what SHAP throws at us. The SHAP most important figures are the following ones:



19 – Shap Value

Here the features are ordered from the highest to the lowest effect on the prediction. It takes in account the absolute SHAP value, so it does not matter if the feature affects the prediction in a positive or negative way. We can see that the features order changed, for example or top 1 feature now is glucose while top 1 feature from feature selection is age. In machine learning, each feature is assigned an importance value representing its contribution to the model's output. SHAP values show how each feature affects each final prediction, the significance of each feature compared to others, and the model's reliance on the interaction between features.

Another function of SHAP that helped us during this project is the function to see the interpretability of the features in each values of our predictions. Value 1 might be positive as value 20, but the impact of the features has a different impact. Here I will show the interpretability of the features on one value.

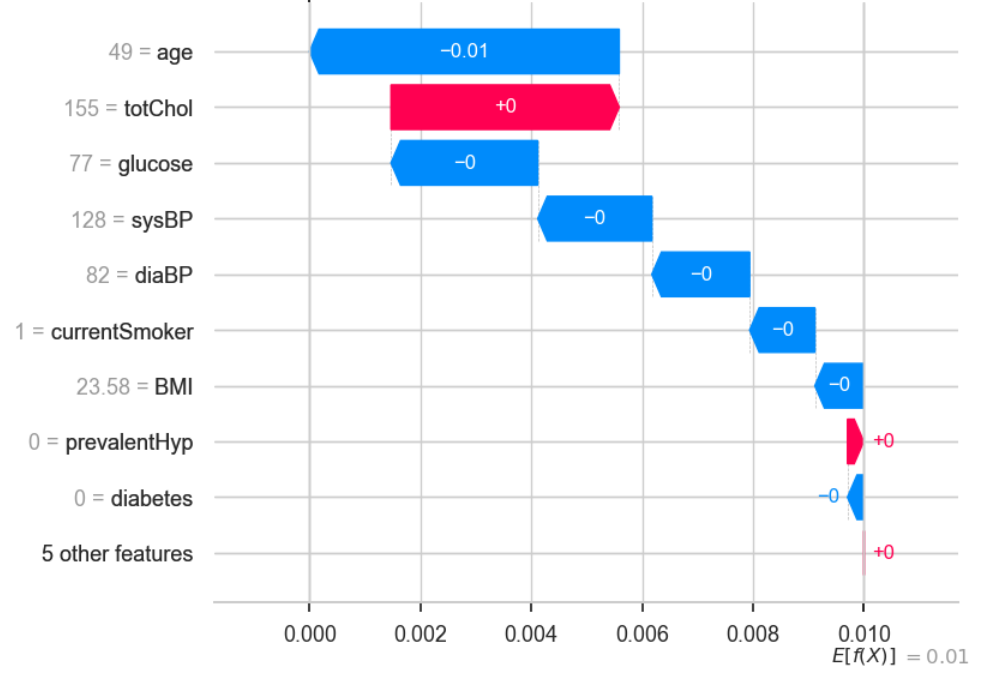
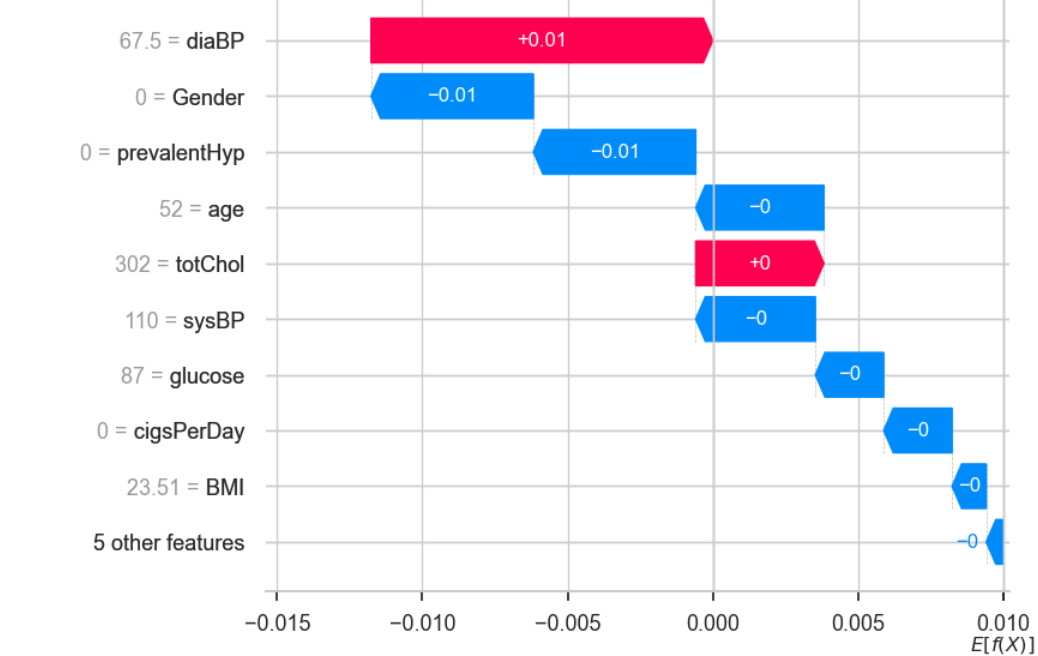


Figure 20 – Shap Value 1

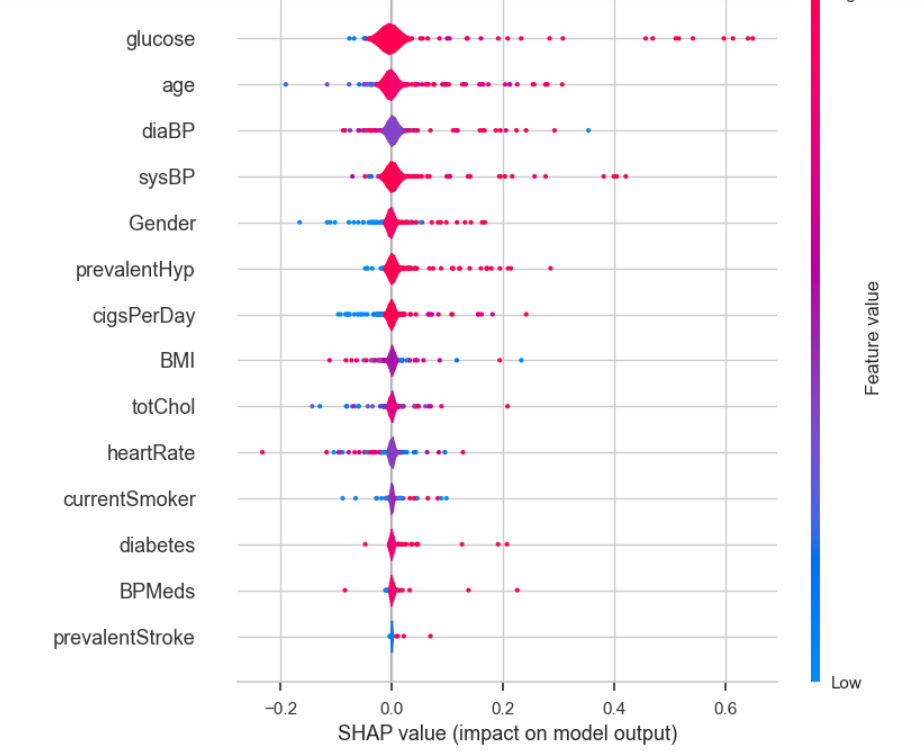
This plot shows us what are the main features affecting the prediction of a single observation, and the magnitude of the SHAP value for each feature. What we can see here is that age and totChol are the features with more impact in this scenario. If we remember, glucose should be the feature with the most of the importance in the model but we here in this example that age impacts more in the prediction. Let’s look for another value and compare.



21 – Main features and shap magnitude

In this scenario, we see different features generating different impacts on the prediction.

Other useful function that made us decide to use SHAP into our project was the summary plot. The summary plot combines feature importance with feature effects. Each point on the summary plot is a Shapley value for a feature and an instance. The position on the y-axis is determined by the feature and on the x-axis by the Shapley value. The color represents the value of the feature from low to high.



22 – Shap value impact on model output

On the bee swarm the features are also ordered by their effect on prediction, but we can also see how higher and lower values of the feature will affect the result.

All the little dots on the plot represent a single observation. The horizontal axis represents the SHAP value, while the color of the point shows us if that observation has a higher or a lower value, when compared to other observations. If you look at the feature “glucose ', you will see that it is mostly high with a positive SHAP value. It means higher the feature glucose is tending to a positive effect in the output. SHAP importance is measured at row level. It represents how a feature influences the prediction of a single row relative to the other features in that row and to the average outcome in the dataset.

Apart from machine learning interpretability and explanation, SHAP value can be used for:

**Model debugging**: by examining the SHAP values, we can identify any biases or outliers in the data that may be causing the model to make mistakes.

**Feature importance**: Identifying and removing low-impact features can create a more optimized model.

**Anchoring explanations:** We can use SHAP values to explain individual predictions by highlighting the essential features that caused that prediction. It can help users understand and trust a model's decisions.

**Model summaries:** It can provide a global summary of a model in the form of a SHAP value summary plot: It gives an overview of the most important features across the entire dataset.

**Detecting biases:** The SHAP value analysis helps identify if certain features disproportionately affect particular groups. It enables the detection and reduction of discrimination in the model.

**Fairness auditing:** It can be used to assess a model's fairness and ethical implications.

**Regulatory approval:** SHAP values can help gain regulatory approval by explaining the model's decisions.

# EVALUATION

Importing data analysis and machine learning libraries. The dataset "heart\_disease.csv" was utilized, the patient's age, gender, and other factors pertaining to their health are taken into consideration during this study. The study's objective is to create a model that can forecast whether or not someone would experience heart disease.

The aim variable is "heart\_ stroke," with 0 indicating no heart problem and 1 indicating a heart problem, Following the dataset import, the code verifies the unique values in the dataset and categorizes the columns with fewer than 10 unique values as categorical and the remaining columns as continuous.

The code then removes the "education" column as it is irrelevant for predicting heart disease,

it also checks the number of records and features in the dataset, the number of values, and the number of missing values. The missing values in the numerical columns are replaced with the median.

The next step is to split the dataset into training and testing data using the train\_test\_split function from sklearn.model\_selection. The continuous columns in the dataset are then scaled using the StandardScaler function from sklearn.preprocessing. It then creates several machine learning models, including Logistic Regression, Linear Discriminant Analysis, MLPClassifier, GradientBoostingClassifier, Decision Tree, Random Forest, RandomForestClassifier, ExtraTreesClassifier, AdaBoost, KNeighbors, SVM and Gaussian Naive Bayes. These models are trained on the training dataset and evaluated on the testing dataset using the cross\_val\_score function from sklearn.model\_selection.

Finally, the code creates a confusion matrix and displays it using the ConfusionMatrixDisplay function from sklearn.metrics. The classification report to take metrics such as precision, recall, and F1-score for each class and the ROCAUC to provide insights into areas for improvement of the classes related to heart stroke.

In order to use SHAP, we needed to use a tree-based machine learning algorithm. We proceed to use Random Forest Classifier. The max depth of the result was 21, but in the end we reduced it to 3. It’s important to mention that the accuracy of our model was 84%. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables, as in the case of regression, and categorical variables, as in the case of classification. It performs better for classification and regression tasks. In this tutorial, we will understand the working of random forest and implement random forest on a classification task.

**SHAP examples and visualizations help when it comes to understand the impact each feature has in our values. It’s interesting when we compare each value and their features to understand each case and see why the prediction is correct or not.**

# ****DEPLOYMENT****

In conclusion, to predicting heart disease uses machine learning models to predict the likelihood of heart disease based on patient data. The code imports the necessary libraries, selects the relevant columns, removes irrelevant columns, and scales the continuous columns.

With this code was able to train several machine learning models and evaluates their performance using metrics such as cross-validation, confusion matrix, end ROCAUC. There are numerous ways to deploy the code, such via developing a web application or a mobile application.

Healthcare professionals can utilize the trained machine learning models to forecast the possibility of heart disease in their patients and create suitable treatment regimens. In order to prevent heart attacks and other heart-related ailments, it is essential to predict heart disease. By examining patient data, machine learning algorithms have demonstrated their efficacy in forecasting the likelihood of cardiac disease. In order to identify high-risk patients and create treatment programmers that cater to the individual needs of each patient, this predictive ability is crucial.

The process of creating a heart disease prediction model involves several critical steps. Firstly, relevant columns must be selected from the dataset, demographic details, medical history, and risk factors for heart disease are all included in these columns. In order to prevent the machine learning models from favoring variables that don't help forecast heart disease, irrelevant columns must also be eliminated.

The next step is to scale continuous columns, like blood pressure and cholesterol levels, to prevent the machine learning models from favoring characteristics with greater values.

Scaling these features ensures that the models can effectively analyze patient data and provide accurate predictions, after preprocessing the dataset, machine learning models are trained on the dataset. Algorithms created especially for predicting heart disease are used to train these models.

Support vector machines, decision trees, and LDA are the most often used models. These models are tested using confusion matrix and ROCAUC approaches after being trained on the dataset.

A crucial stage in developing a prediction model for heart disease is evaluating machine learning models.

The predictive capability of the models is evaluated via cross-validation. The dataset is divided into many pieces using this method, and the model is trained on one chunk before being tested on another. By contrasting the actual values with the anticipated values, the ROC AUC is used to assess the model's performance.

The heart disease prediction model can be deployed using several methods, such as creating a web application or a mobile application.

Being able to explain why a model made a particular prediction helps debug potential biases, identify data issues, and justify the model's decisions. We have explored SHAP values and how we can use them to provide interpretability for machine learning models. While having an accurate model is essential, companies need to go beyond accuracy and focus on interpretability and transparency to gain the trust of users and regulators.

Healthcare professionals can input patient data into these applications and use the user-friendly interface to get estimates about the possibility of heart disease. To give healthcare practitioners real-time forecasts, the model can also be included into electronic health record systems.

In conclusion, the early identification and prevention of heart disease depend on the development of a heart disease prediction model employing machine learning algorithms. The model gives medical professionals an efficient tool for assessing the risk of heart disease in their patients and creating suitable treatment regimens. Data preprocessing, model training, and evaluation are three crucial elements in the process of developing a prediction model for heart disease. These procedures are crucial for making sure the model is reliable and efficient. A web application or a mobile application can be developed to deploy the heart disease prediction model and give healthcare practitioners a user-friendly interface for estimating the likelihood of heart disease.

# REFERENCE

Brownlee, J. (2016). *Linear Discriminant Analysis for Machine Learning*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/linear-discriminant-analysis-for-machine-learning/>.

Brownlee, J. (2020). *Random Oversampling and Undersampling for Imbalanced Classification*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/>.

Burke, Sarah. “Prepare Data for Analysis with Python | Jupyter Notebook Template Download.” Cypress North, 25 Nov. 2019, cypressnorth.com/data-analysis/prepare-google-analytics-data-for-analysis-with-python/. Accessed 7 May 2023.

https://www.facebook.com/jason.brownlee.39. “How to Calculate Precision, Recall, F1, and More for Deep Learning Models.” Machine Learning Mastery, 28 Mar. 2019, machinelearningmastery.com/how-to-calculate-precision-recall-f1-and-more-for-deep-learning-models/.

Cross Validated. (n.d.). *How to distinguish the continuous and categorical variable based on the number of unique values*. Available at: <https://stats.stackexchange.com/questions/462891/how-to-distinguish-the-continuous-and-categorical-variable-based-on-the-number-o>.

D, K. (2023). *Optimizing Performance: SelectKBest for Efficient Feature Selection in Machine Learning*. [online] Medium. Available at: <https://medium.com/@Kavya2099/optimizing-performance-selectkbest-for-efficient-feature-selection-in-machine-learning-3b635905ed48> [Accessed 11 Nov. 2023].

Frost, J. (2018). *Normal Distribution in Statistics*. [online] Statistics By Jim. Available at: <https://statisticsbyjim.com/basics/normal-distribution/>.

IBM. “Quick Start: Build and Deploy a Machine Learning Model in a Jupyter Notebook.” Www.ibm.com, 21 Nov. 2021, www.ibm.com/docs/en/cloud-paks/cp-data/4.0?topic=gsbdtm-build-deploy-machine-learning-model-in-jupyter-notebook. Accessed 7 May 2023.

Nadakinamani, Rajkumar Gangappa, et al. “Clinical Data Analysis for Prediction of Cardiovascular Disease Using Machine Learning Techniques.” Computational Intelligence and Neuroscience, vol. 2022, no. PMC8767405, 11 Jan. 2022, pp. 1–13, [https://doi.org/10.1155/2022/2973324. Accessed 7 May 2023](https://doi.org/10.1155/2022/2973324.%20Accessed%207%20May%202023).

Phillips, B. (2013). StatsMiniBlog: *Continuous vs. Categorical*. [online] ADC Online Blog. Available at: <https://blogs.bmj.com/adc/2013/05/13/statsminiblog-continuous-vs-categorical/#:~:text=Categorical%20variables%2C%20aka%20discrete%20variables>.

pythonexamples.org. (n.d.). *How to Get Columns of Numeric Datatype from DataFrame*? - Python Examples. [online] Available at: <https://pythonexamples.org/pandas-dataframe-select-columns-of-numeric-datatype/#:~:text=To%20select%20columns%20that%20are>

scikit-learn. (n.d.). *1.13. Feature selection*. [online] Available at: <https://scikit-learn.org/stable/modules/feature_selection.html#univariate-feature-selection>.

scikit-learn. (n.d.). *sklearn.preprocessing.LabelEncoder*. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html#:~:text=LabelEncoder%5Bsource%5D>

Singh, Poornima, et al. “Effective Heart Disease Prediction System Using Data Mining Techniques.” International Journal of Nanomedicine, vol. Volume 13, no. PMC5863635, Mar. 2018, pp. 121–124, [https://doi.org/10.2147/ijn.s124998. Accessed 7 May 2023](https://doi.org/10.2147/ijn.s124998.%20Accessed%207%20May%202023).

Shubhankar Rawat. “Heart Disease Prediction.” Medium, Towards Data Science, 10 Aug. 2019, towardsdatascience.com/heart-disease-prediction-73468d630cfc. Accessed 7 May 2023.

www.kaggle.com. (n.d.). Heart Disease Dataset. [online] Available at: <https://www.kaggle.com/datasets/mirzahasnine/heart-disease-dataset>. Accessed 08 April 2023.

www.scikit-yb.org. (n.d.). *ROCAUC — Yellowbrick v1.5 documentation*. [online] Available at: <https://www.scikit-yb.org/en/latest/api/classifier/rocauc.html>.

pythonexamples.org. (n.d.). *How to Get Columns of Numeric Datatype from DataFrame*? - Python Examples. [online] Available at: <https://pythonexamples.org/pandas-dataframe-select-columns-of-numeric-datatype/#:~:text=To%20select%20columns%20that%20are>

scikit-learn. (n.d.). *sklearn.preprocessing.LabelEncoder*. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html#:~:text=LabelEncoder%5Bsource%5D>

Brownlee, J. (2020). *Random Oversampling and Undersampling for Imbalanced Classification*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/>