THE SMITH PARASITE

MACHINE LEARNING 2022/2023

DSAA_AA_202223_GROUP60

BEATRIZ CARMO - 20220685
JOÃO MALHO - 20220696
LIZAVETA BARYSIONAK - 20220667
MARTA ANTUNES - 20221094
TOMÁS CÔRTE-REAL – 20221639

Table of Contents

1. INTRODUCTION

Every day the number of new diseases is increasing and sometimes is difficult, by traditional medicine techniques, to predict if patients have a certain disease, once diseases can give the same symptoms and patients can also have symptoms without being contaminated, just due to their life habits.

This study was done to develop, through application of machine learning techniques, a predictive model to identify which listed patients are more often to carry a certain parasite named "Smith Parasite". Through a simple observation of a dataset with "n" number of patients" regarding health and habits profile of each one, we teach our model to learn the standardized behaviour of patient symptoms once they are a parasite carrier, and after that once it receives a similar data, predict with high accuracy which patients are often to have the parasite.

This type of processes can be implemented in medicine to support medical attention and patient screening, which we believe to have a big positive impact in daily basis of medicine routine.

Note this report has lists, images, tables and graphs associated to it in annexes pages, each one is marked with a distinct reference that will correspond to the same reference in annexes.

2. EXPLORATION

2.1. Problem Definition

A new disease has recently been discovered by Dr. Smith, in England. We have been brought in to investigate. The disease has already affected more than 5000 people, with no apparent connection between them.

The most common symptoms include fever and tiredness, but some infected people are asymptomatic. Regardless, this virus is being associated with post-disease conditions such as loss of speech, confusion, chest pain and shortness of breath.

The conditions of the transmission of the disease are still unknown and there are no certainties of what leads a patient to suffer or not from it. Nonetheless, some groups of people seem more prone to be infected by the parasite than others.

In this study, our goal is to build a predictive model that answers the question, "Who are the people more likely to suffer from the Smith Parasite?" With that goal, we can access a small quantity of sociodemographic, health, and behavioural information obtained from the patients.

2.2. Algorithm Definition

This process regards statistical, data mining and machine learning algorithms, which will be presented in detail in each process pipeline step, although the main algorithm used to get the result prediction was Random Forest, which is the most precise model for this type of binary problems, and is also so far, the preferred in most business analysis over predictive problems as this one.

3. PRE-PROCESSING

3.1. Data Gathering

Data: Before beginning this analysis, it was important to make sure if the available data was valid with no incoherent values, was necessary to check patient profile data which was been applied unsupervised and supervised machine learning methodologies.
The training set was used to build the machine learning models. In this set, we had the ground truth associated to each patient, if the patient has the disease (Disease = 1) or not (Disease = 0). And the test set was used to check how well models' performance is on unseen data, this set does not have access to the ground truth.

Features information - **(Table 1) in annexes page 1*
Features description – **(List 1) in annexes page 1 and 2*

To facilitate study the provided data has been merged in two distinct datasets, train dataset and test dataset

3.2. Feature Engineering

In this stage features were analysed and altered data to get a valid and consistent dataset in order to be possible to work with it, analysing the possible existence of missing values, capital letters, existence of outliers, spearman correlation between features avoiding redundant features and encoding, considering that these changes must be done in Train Data such as in Test Data.
The necessary data interventions were, feature ***"Education"*** had 13 missing values, those were replaced by mode ("University Complete (3 or more years)"), feature ***"Region"*** had city of London written as "LONDON" which have been replaced by "London". Once our data is not normalized (*image 1 and image 2*) was checked by spearman correlation the inexistence of redundant features (*image 3*).
The outlier removal process required a detailed analysis, due to the uniqueness of metrics variables. Analysis was relied mostly on pair plots interpretation and Interquartile Range (IQR) analysis *(image 4 and 5)* leading to removal of patients that had Birth Years before 1900 (12 elements, 1,5% of total data) and patients High Cholesterol Levels above 500 (2 elements, 0,25% of total remaining data), was checked that this removal didn't compromise the balance of data (*image 6*)
It also required the Boolean features like "Exercise" and "Smoking Habit" to be changed to binary features, "Yes" to 1 and "No" to 0.
Last approach needed was transformation of categorical features "Region", "Education", "Drinking Habit", "Fruit Habit", "Water Habit", "Check-up" and "Diabetes" to numerical features through Dummy Encoding with Pandas function *get_dummies*.

3.3. Feature Selection

Was used feature importance of decision tree classifier to measure which set of variables were relevant to build and test our model with. The dataset was divided in two datasets, X and y, and fitted in Decision Tree supervised learning method regarding the following parameters:

- **X** - X will always regard full train data **excluding** the result feature which is the Disease column in this case
- **y** - y will always regard only the result features, which is Disease column in this case
- **Entropy** - Entropy in statistical mechanics is a measure of the number of ways a system can be arranged, often taken to be a measure of 'disorder' (the higher the entropy, the higher the disorder).
- **max_depth = 10** - The maximum depth of the tree, is the maximum of a binary tree, is the number of nodes along the longest path from the root node down to the farthest leaf node.

After fitting our data, run the function above to get the more notable features, and was defined, as normal approach, the maintain features with relevance higher than 2% reducing our data to 11 relevant features (*image 7*).

- **Feature Importance** - Calculated as the decrease in node impurity weighted by the probability of reaching that node. The node probability can be calculated by the number of samples that reach the node, divided by the total number of samples, the higher the value the more important feature.

After preparing the data we check the correlation with the variables before encoding, to see If we have problems of multicollinearity and to find out how much the variables are associated with each other, which does not happen. (*Image 8*)

4. MODELLING

This prediction is based on Random Forest (*image 9*) which is a classification algorithm consisting of many decisions trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

Random Forest uses a technique called ensemble techniques named bagging or bootstrap aggregation, this technique consists in training a particular dataset (named as train data) through several models, per each model it is provided a random sample of train data, for model 1 sample 1, for model 2 sample 2 and so on, each model will provide a prediction, this test division by several models and by several samples (which use a Row Sampling with Replacement) is called bootstrap after all models provide their results, in this binary test, the final result is created based on the average of the results of all models, if the majority of the models result is 1 then the final prediction will be 1, this vote is called aggregation.

In Random Forest this ensemble technique is replaced by decision trees, it gives to each decision tree model a random sample of rows and features and predicts by vote the final prediction (*image 10*)

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. Decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules. The deeper the tree, the more complex the decision rules and the fitter the model. Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. A leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor is called root node. (*Image 9*)

Once this process work with **Decision Trees,** it must be considered 2 properties:

- **Low Bias** - If a decision tree is created to its complete depth, then what will happen is that will be trained for our train set, so train error will be extremely low
- **High Variance** - When we get our new test data, once the number of results is high and not well balanced, they are prone to give a larger amount of errors

When create a decision tree model to its complete depth it tends to lead into **overfitting,** to avoid this has set a max depth of ten.

5. ASSESSMENT

This study was evaluated with supervised and unsupervised models although the model that reported the best score was Random Forest. Note that all models have hyperparameter settings tunned.

5.1. Logistic Regression:

Logistic regression is a powerful supervised ML algorithm used for binary classification problems (when target is categorical). Logistic regression uses a logistic function, defined in (*image 11*), to model a binary output variable. The primary difference between linear regression and logistic regression is that logistic regression's range is bounded between 0 and 1. In addition, as opposed to linear regression, logistic regression does not require a linear relationship between inputs and output variables. (*Image 11*)

Parameters used:

- **Solver** - 'liblinear' because data is a small data and liblinear solver is the indicated one for small datasets

- **random_state = 0** - in logistic regression it does not make any difference which is the random state because score does not change, although 0 was the target sample to analyse.

This prediction model had a f1-score of 0,78 which is not a good prediction score for the subject problem with 22 false positives elements and 13 false negatives elements in confusion matrix (*result 1*).

5.2 – Decision Tree:

Explained in modelling topic of this report and supported by (*image 9*)
Parameters used:

- **criterion = 'entropy'** - criterion is the function to measure the quality of a split.
- **max_depth = 10** - The maximum depth of the tree, is the maximum of a binary tree, is the number of nodes along the longest path from the root node down to the farthest leaf node.
- **min_samples_split = 2** - The minimum number of samples required to split an internal node.
- **max_features = 'auto'** - The number of features to consider when looking for the best split.
- **class_weight = 'balanced'** - Weights associated with classes
- **min_samples_lead = 1** - The minimum number of samples required to be at a leaf node.
- **random_state = 0** – target sample to analyse

This prediction model had a f1-score of 0,96 which is a good prediction score, although this subject problem allows accuracies at 1, the decision tree result presents 5 false positives elements and 1 false negative element in confusion matrix (*result 2*)

5.3 – Random Forest Model

Explained in modelling topic of this report and supported by (*image 5*)
Parameters used:

- **criterion = 'entropy'**
- **max_depth = 10**
- **min_samples_split = 2**
- **max_features = 'auto'**
- **bootstrap = False** - The whole dataset is used to build each tree. While tuning the hyperparameters of our model to our dataset, was noted that setting bootstrap=False results in a most performing model

- **oob_score = False** - Only available if bootstrap=True, once bootstrap is False then oob_score is also False
- **warm_start = False** - Will not reuse the solution of the previous call to fit and add more estimators to the ensemble
- **class_weight = 'balanced'**
- **min_samples_lead = 1** - The minimum number of samples required to be at a leaf node.
- **random_state = 0** – target sample to analyse

This prediction model had a f1-score of 1 which is a perfect prediction score once this subject problem allows accuracies at 1, the random forest result presents 0 false positives elements and 0 false negative element in confusion matrix (*result 3*)

5.4 – K Nearest Neighbors Model (KNN)

The **k-nearest neighbors' algorithm**, also known as **KNN or k-NN**, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. Working off the assumption that similar points can be found near one another.
To implement tests over this model, data needed to be normalized in order to turn this type of following models more performing, normalization was made via Minmax scaler changing the input features to a scale between 0 and 1.

Parameters used:

- **n_neighbors = 5** - Number of neighbors to use by default, this are the closest neighbors to compare.
- **leaf_size = 1** - Leaf size passed to BallTree or KDTree. This can affect the speed of the construction and query, as well as the memory required to store the tree. The optimal value depends on the nature of the problem. The leaf size controls the minimum number of points in each node.
- **p = 1** - **Manhattan Distance** Captures the distance between two points by aggregating the pairwise absolute difference between each variable, has better performance than the Euclidean distance method.
- '**weights = 'distance**' - In this case, closer neighbors of a query point will have a greater influence than neighbors which are further away.
- **algorithm = 'auto'** - Will attempt to decide the most appropriate algorithm based between ball tree, kd tree and brute force.

This prediction model had a f1-score of 0,94 which is not the best prediction score, the KNN results presents 3 false positives elements and 6 false negative elements in confusion matrix (*result 4*)

5.5 – Neural Network

A **neural network** is a method in artificial intelligence that teaches computers to process data in a way that is inspired by the human brain. It is a type of machine learning process, called deep learning, which uses interconnected nodes or neurons in a layered structure that resembles the human brain. Neural networks are a set of algorithms, modelled loosely after the human brain, which are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labelling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text, or time series, must be translated.

Parameters used:

- **activation = 'tanh' -** The hyperbolic tan function, returns f(x) = tanh(x)
- **solver** – 'adam' - Refers to a stochastic gradient-based optimizer, works pretty well on relatively large datasets (with thousands of training samples or more), this data set is not big but was noted that adam perform very well in it.
- **learning_rate_init = 0.0505** - The initial learning rate used. It controls the step-size in updating the weights. Used due solver = 'adam'
- **learning_rate = 'constant'** - Learning rate schedule for weight updates
- **random_state = 0** – target sample to analyse

This prediction model had a f1-score of 0,97 which is not the best prediction score, the neural network results present 1 false positives element and 4 false negative elements in confusion matrix (*result 5*).

5.6 – Receiver Operating Characteristic

ROC or Receiver Operating Characteristic plot is used to visualise the performance of a binary classifier. It gives us the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) at different classification thresholds.
In classification, there are many different evaluation metrics. The most popular is accuracy, which measures how often the model is correct. This is a great metric because it is easy to understand and getting the most correct guesses is often desired.
Another common metric is AUC, area under the receiver operating characteristic (ROC) curve. The Receiver operating characteristic curve plots the true positive (TP) rate versus the false positive (FP) rate at different classification thresholds. The thresholds are different probability cut-offs that separate the two classes in binary classification. It uses probability to tell us how well a model separates the classes. (*Image 12*)

6 – CONCLUSIONS

About the problem it was concluded that with similar datasets and similar problems by random forest model can predict with 1 accuracy if patients are infected or not, and by this case study is possible to confirm that random forest is the best model for binary predictions.
Which can be a huge step in medicinal environments, also that for binary problems random forest is one of the most accurate models and in fact most business decisions are analysed with it.

Future research propositions model as this one can be adjusted and implemented in high difficult situations in medicine, as analysing cancer possibilities as our professor already began developing and investigating, new virus, new bacteria, new diseases, and can be for example, in further steps, combined in software such as the Robin Robot (bibliography) to make a first direct tracking of patients helping doctor performance and time of response by deep learning approaches.
Also study and work over more advanced model, in deep learning, that are possibly more performing and precise, also work with big data in order to implement more Data Mining practices, apply more dense Machine Learning Model and even organize result by clusters, measuring groups of patients, if study applied in medicine.

7 – REFERENCES

1. *Building A Logistic Regression in Python, Step by Step | by Susan Li | Towards Data Science*. (n.d.). Retrieved December 11, 2022, from https://medium.com/towards-data-science/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8

2. *Course: 202223 - Aprendizagem Automática - S1*. (n.d.). Retrieved December 11, 2022, from https://elearning.novaims.unl.pt/course/view.php?id=2351

3. *Robin The Robot Comforts Kids In Hospitals, Can Help With Covid-19*. (n.d.). Retrieved December 11, 2022, from https://www.forbes.com/sites/jeffkart/2020/06/17/robin-the-robot-comforts-kids-in-hospitals-can-help-with-covid-19/?sh=445e9f6574cc

4. *scikit-learn: machine learning in Python — scikit-learn 1.2.0 documentation*. (n.d.). Retrieved December 11, 2022, from https://scikit-learn.org/stable/

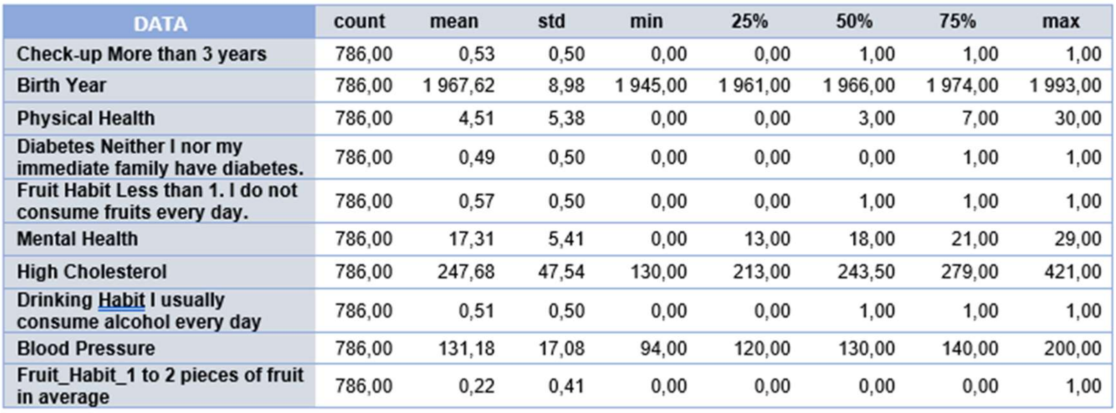
5. *Understanding Random Forest. How the Algorithm Works and Why it Is… | by Tony Yiu | Towards Data Science*. (n.d.). Retrieved December 11, 2022, from https://towardsdatascience.com/understanding-random-forest-58381e0602d2

6. *What are Neural Networks? | IBM*. (n.d.). Retrieved December 11, 2022, from https://www.ibm.com/cloud/learn/neural-networks

7. *Data Mining: The Textbook | by Charu C. Aggarwal*

8 – ANNEXES

Table 1 - Features information



DATA	count	mean	std	min	25%	50%	75%	max
Check-up More than 3 years	786,00	0,53	0,50	0,00	0,00	1,00	1,00	1,00
Birth Year	786,00	1 967,62	8,98	1 945,00	1 961,00	1 966,00	1 974,00	1 993,00
Physical Health	786,00	4,51	5,38	0,00	0,00	3,00	7,00	30,00
Diabetes Neither I nor my immediate family have diabetes.	786,00	0,49	0,50	0,00	0,00	0,00	1,00	1,00
Fruit Habit Less than 1. I do not consume fruits every day.	786,00	0,57	0,50	0,00	0,00	1,00	1,00	1,00
Mental Health	786,00	17,31	5,41	0,00	13,00	18,00	21,00	29,00
High Cholesterol	786,00	247,68	47,54	130,00	213,00	243,50	279,00	421,00
Drinking Habit I usually consume alcohol every day	786,00	0,51	0,50	0,00	0,00	1,00	1,00	1,00
Blood Pressure	786,00	131,18	17,08	94,00	120,00	130,00	140,00	200,00
Fruit_Habit_1 to 2 pieces of fruit in average	786,00	0,22	0,41	0,00	0,00	0,00	0,00	1,00

List 1 – Features description

Sociodemographic Data:

- **PatientID** - The unique identifier of the patient
- **Birth_Year** - Patient Year of Birth
- **Name** - Name of the patient
- **Region** - Patient Living Region
- **Education** - Answer to the question: What is the highest grade or year of school you have?
- **Disease** - The dependent variable. If the patient has the disease (Disease = 1) or not (Disease = 0)

Health Related Data:

- **Height** - Patient"s height
- **Weight** - Patient"s weight
- **Checkup** - Answer to the question: How long has it been since you last visited a doctor for a routine Checkup? [A routine Checkup is a general physical exam, not an exam for a specific injury, illness, or condition.]
- **Diabetes** - Answer to the question: (Ever told) you or your direct relatives have diabetes?
- **High_Cholesterol** - Cholesterol value
- **Blood_Pressure** - Blood Pressure in rest value
- **Mental Health** - Answer to the question: During the past 30 days, for about how many days did poor physical or mental health keep you from doing your usual activities, such as self-care, work, or recreation?
- **Physical Health** - Answer to the question: Thinking about your physical health, which includes physical illness and injury, for how many days during the past 30 days was your physical health not good to the point where it was difficult to walk?

Habits Related Data:

- **Smoking_Habit** - Answer to the question: Do you smoke more than 10 cigars daily?

- **Drinking_Habit** - Answer to the question: What is your behavior concerning alcohol consumption?
- **Exercise** - Answer to the question: Do you exercise (more than 30 minutes) 3 times per week or more?
- **Fruit_Habit** - Answer to the question: How many portions of fruits do you consume per day?
- **Water_Habit** - Answer to the question: How much water do you drink per day?

Image 1 – Numerical Features Distribution by Histograms

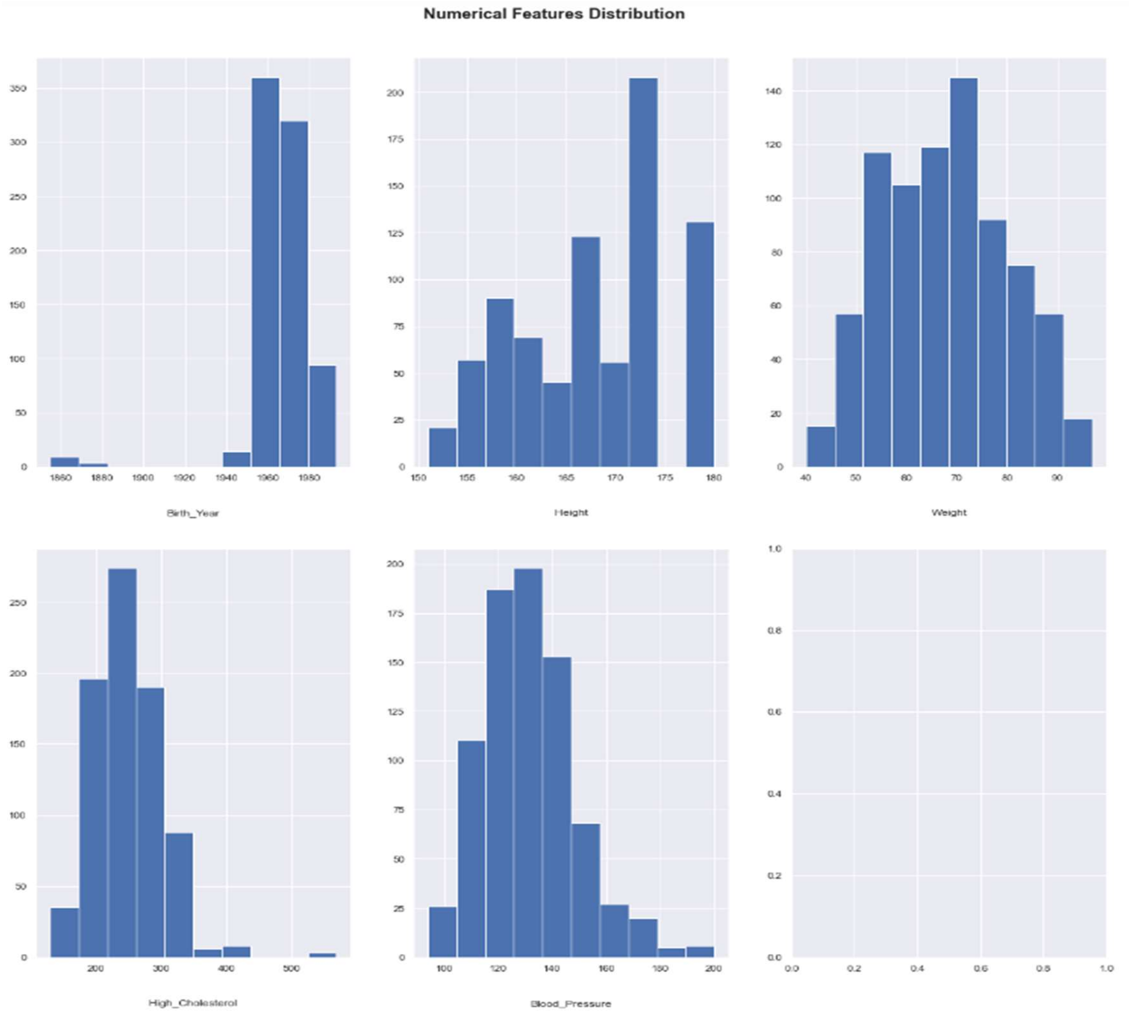


Image 2 – Numerical Features Distribution by Boxplot

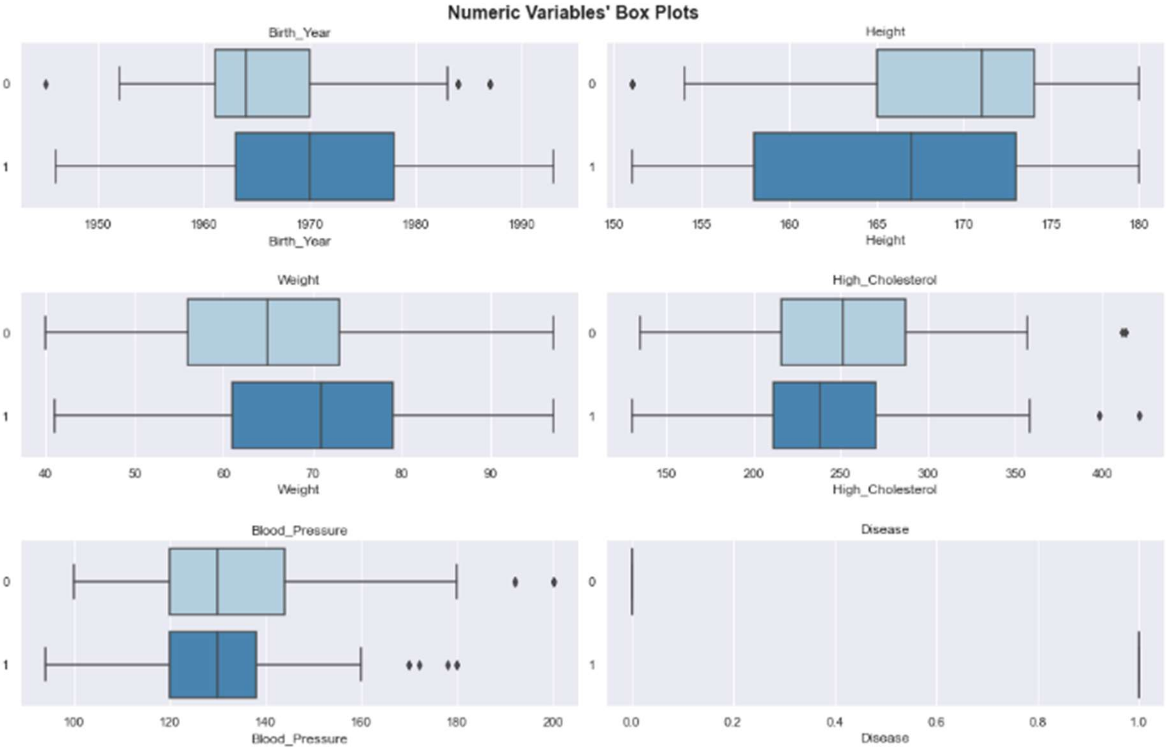


Image 3 – Numerical Features correlation by Spearman

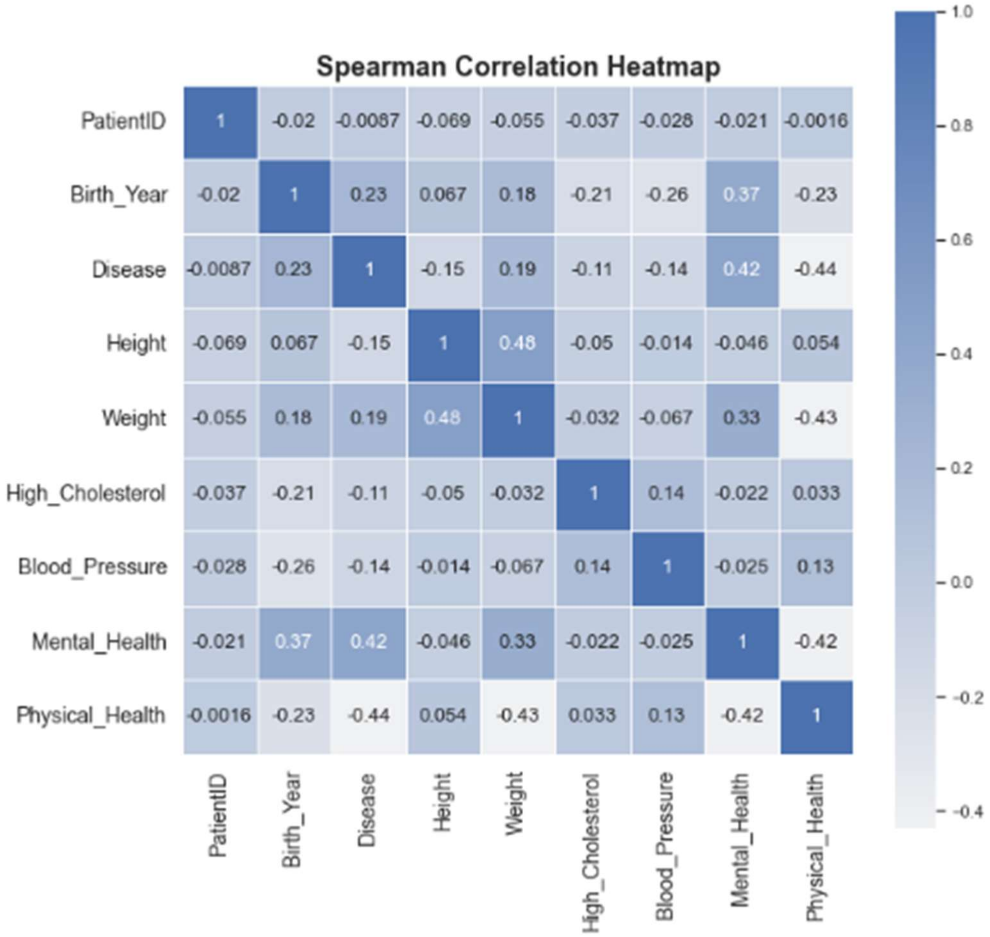


Image 4 – Pair Plot Outliers view

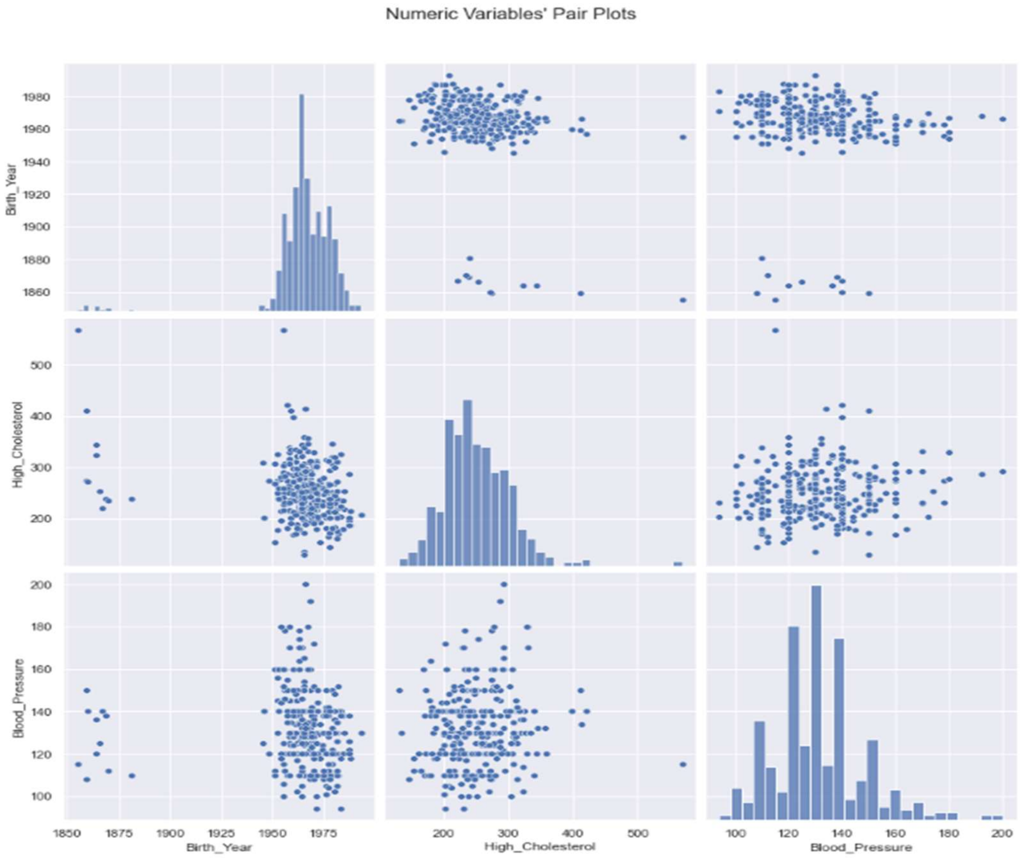


Image 5 – Box Plot Outliers view

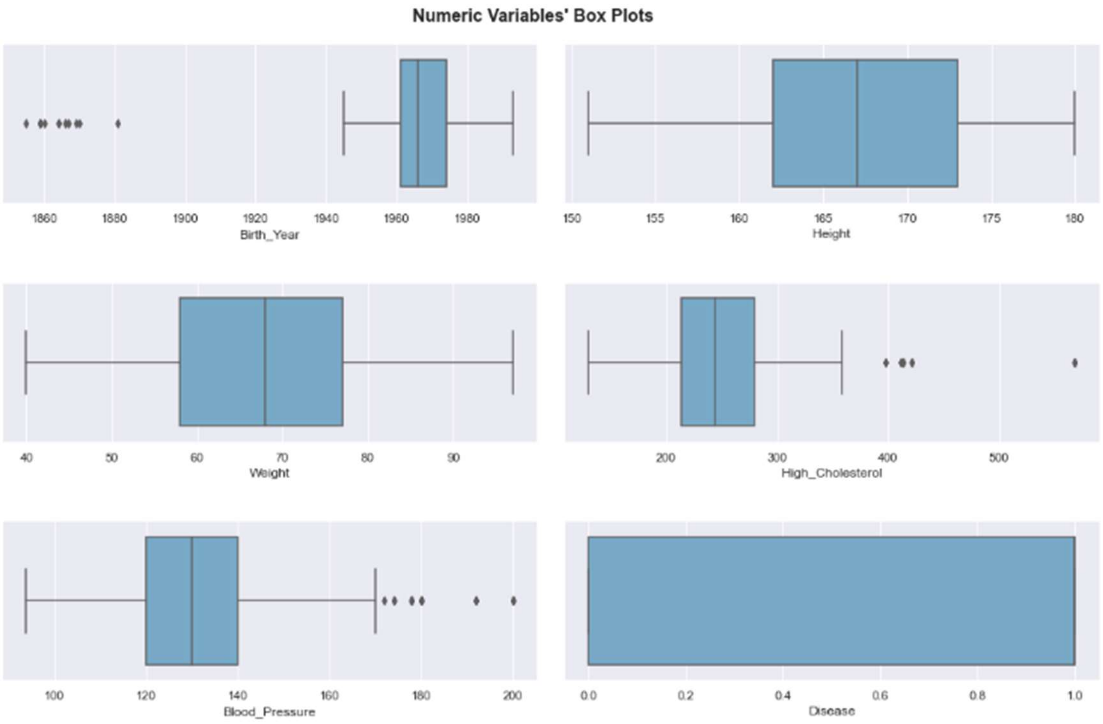
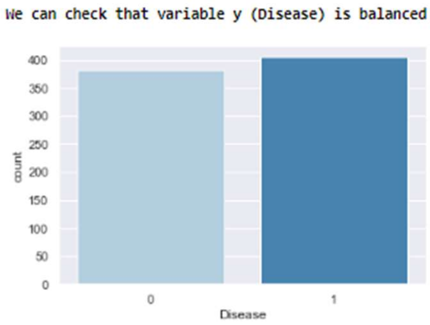
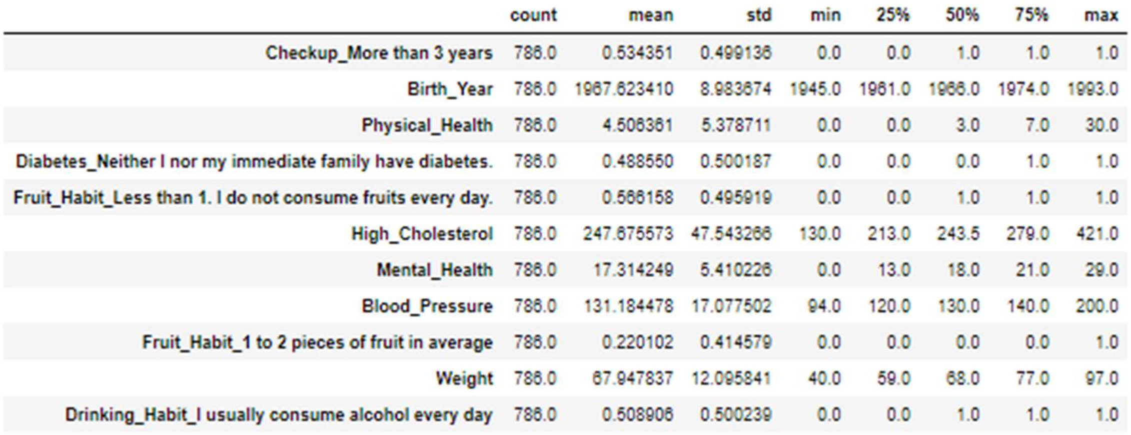


Image 6 – Dataset Balance view



```
We can check that variable y (Disease) is balanced
```

Image 7 – Relevant features with relevance above 2%



	count	mean	std	min	25%	50%	75%	max
Checkup_More than 3 years	786.0	0.534351	0.499136	0.0	0.0	1.0	1.0	1.0
Birth_Year	786.0	1967.623410	8.983674	1945.0	1961.0	1966.0	1974.0	1993.0
Physical_Health	786.0	4.506361	5.378711	0.0	0.0	3.0	7.0	30.0
Diabetes_Neither I nor my immediate family have diabetes.	786.0	0.488550	0.500187	0.0	0.0	0.0	1.0	1.0
Fruit_Habit_Less than 1. I do not consume fruits every day.	786.0	0.566158	0.495919	0.0	0.0	1.0	1.0	1.0
High_Cholesterol	786.0	247.675573	47.543266	130.0	213.0	243.5	279.0	421.0
Mental_Health	786.0	17.314249	5.410226	0.0	13.0	18.0	21.0	29.0
Blood_Pressure	786.0	131.184478	17.077502	94.0	120.0	130.0	140.0	200.0
Fruit_Habit_1 to 2 pieces of fruit in average	786.0	0.220102	0.414579	0.0	0.0	0.0	0.0	1.0
Weight	786.0	67.947837	12.095841	40.0	59.0	68.0	77.0	97.0
Drinking_Habit_I usually consume alcohol every day	786.0	0.508906	0.500239	0.0	0.0	1.0	1.0	1.0

Image 8 – Correlation Heatmap of relevant features

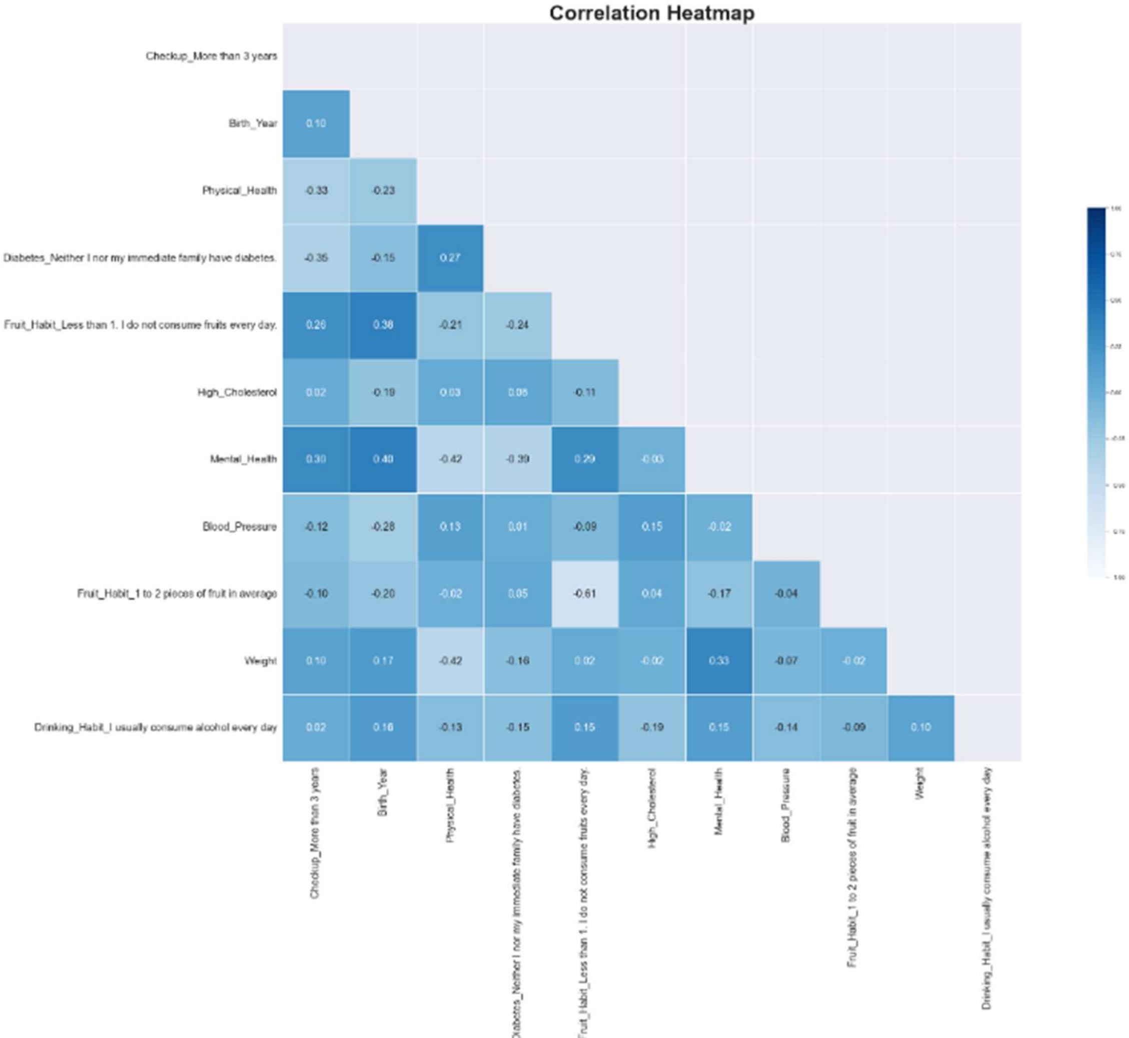


Image 9 – Decision Tree

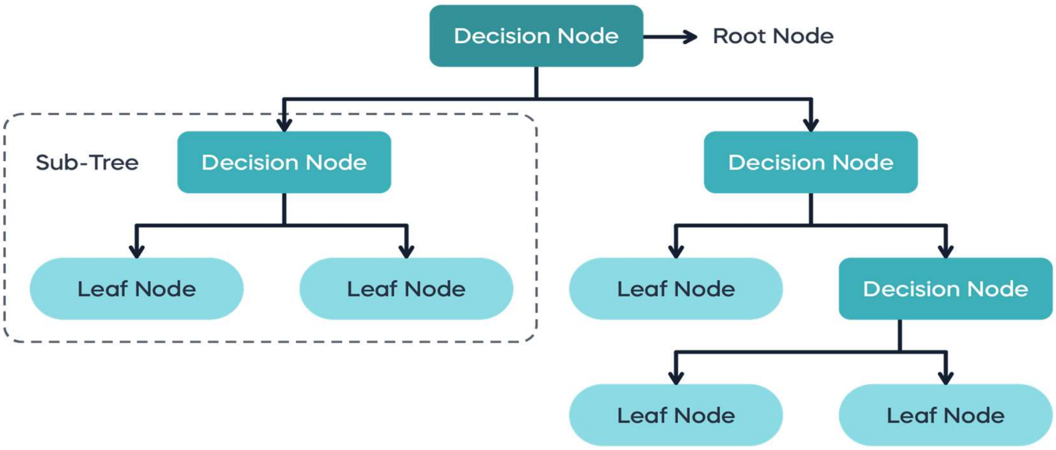


Image 10 – Random Forest Model

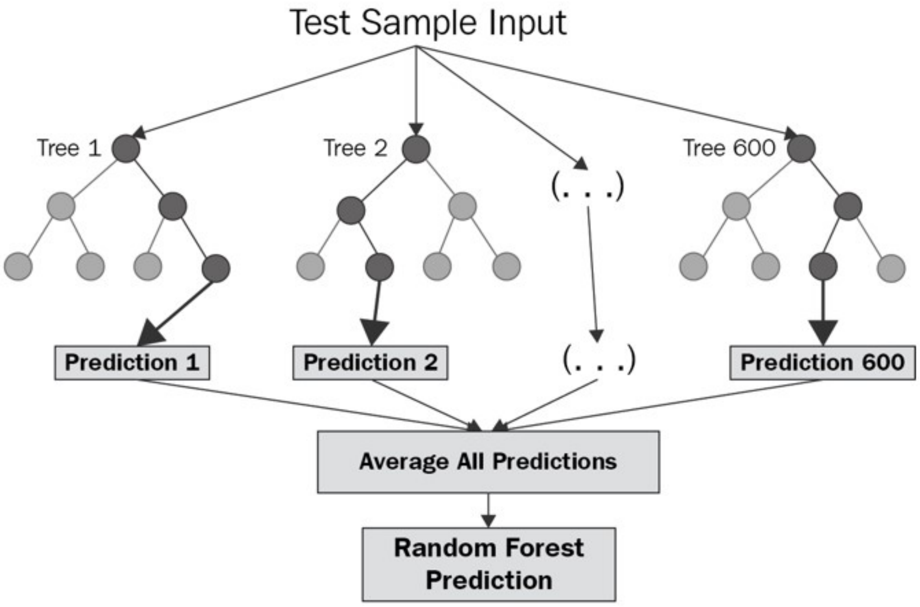
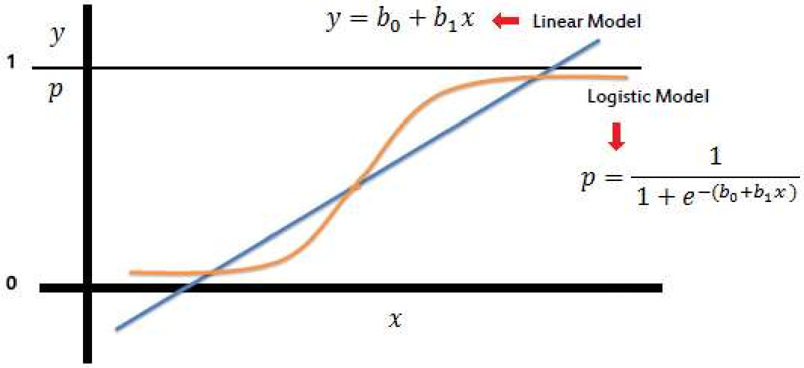
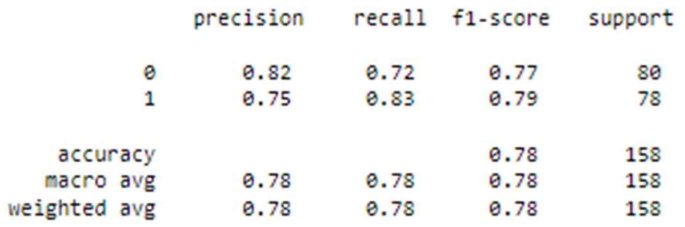


Image 11 – Logistic Regression and Linear Regression



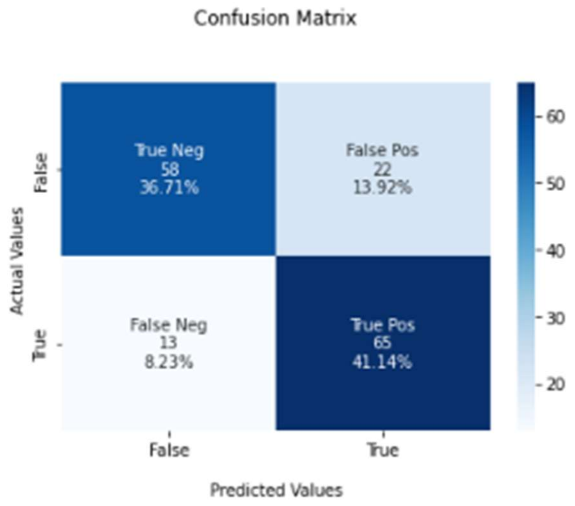
Result 1 – Logistic Regression results



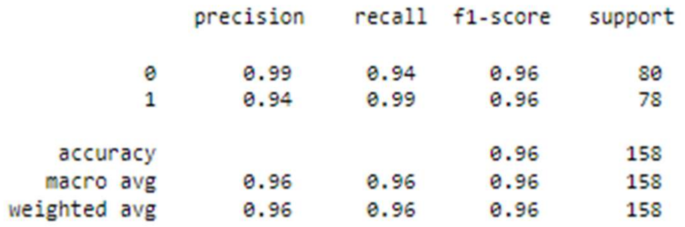
```
              precision    recall  f1-score   support

           0       0.82      0.72      0.77        80
           1       0.75      0.83      0.79        78

    accuracy                           0.78       158
   macro avg       0.78      0.78      0.78       158
weighted avg       0.78      0.78      0.78       158
```



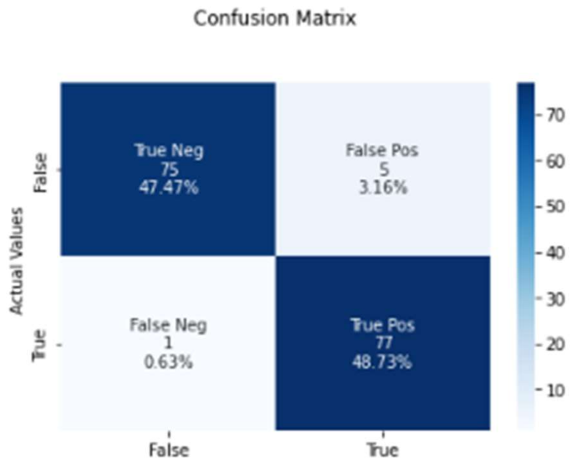
Result 2 – Decision Tree results



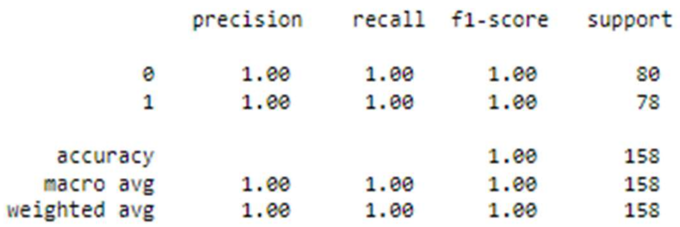
```
              precision    recall  f1-score   support

           0       0.99      0.94      0.96        80
           1       0.94      0.99      0.96        78

    accuracy                           0.96       158
   macro avg       0.96      0.96      0.96       158
weighted avg       0.96      0.96      0.96       158
```



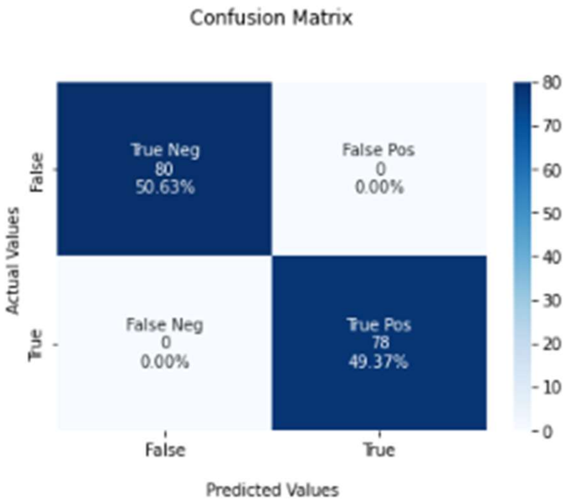
Result 3 – Random Forest results



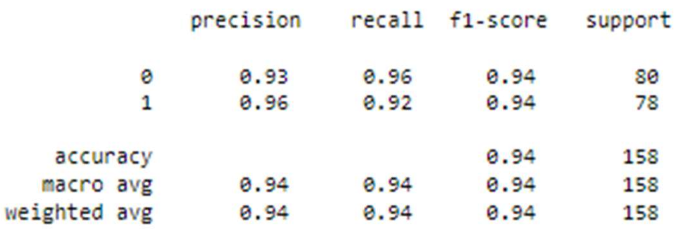
```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        80
           1       1.00      1.00      1.00        78

    accuracy                           1.00       158
   macro avg       1.00      1.00      1.00       158
weighted avg       1.00      1.00      1.00       158
```



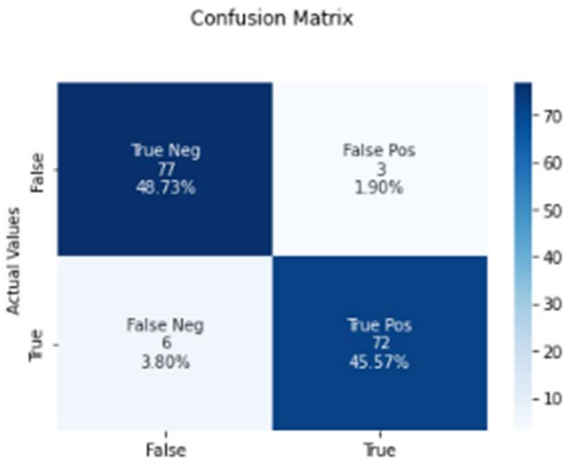
Result 4 – K Nearest Neighbours results



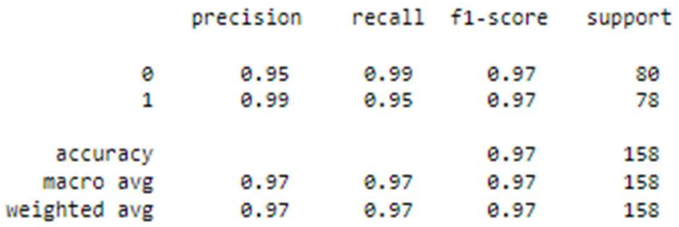
```
              precision    recall  f1-score   support

           0       0.93      0.96      0.94        80
           1       0.96      0.92      0.94        78

    accuracy                           0.94       158
   macro avg       0.94      0.94      0.94       158
weighted avg       0.94      0.94      0.94       158
```



Result 5 – Neural Networks results



```
              precision    recall  f1-score   support

           0       0.95      0.99      0.97        80
           1       0.99      0.95      0.97        78

    accuracy                           0.97       158
   macro avg       0.97      0.97      0.97       158
weighted avg       0.97      0.97      0.97       158
```

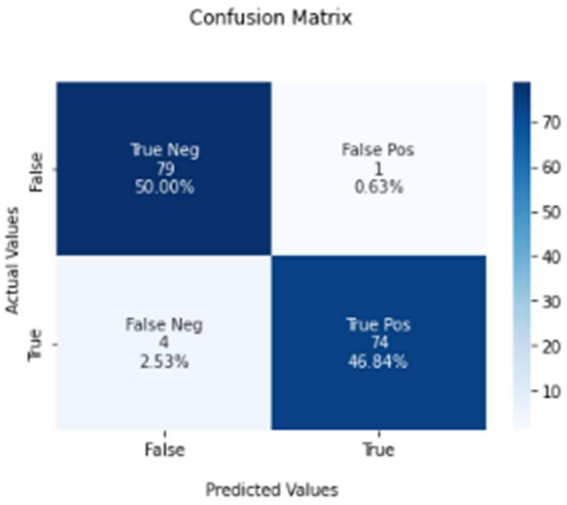
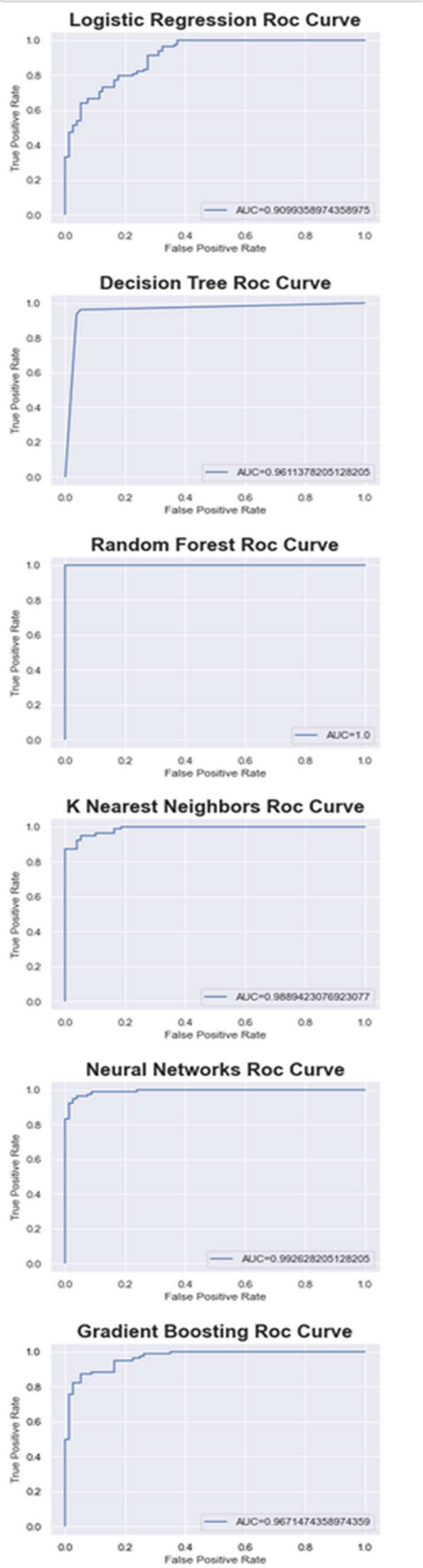


Image 12 – Neural Networks results


Logistic Regression Roc Curve
True Positive Rate
False Positive Rate
AUC=0.9099358974358975
Decision Tree Roc Curve
True Positive Rate
False Positive Rate
AUC=0.9611378205128205
Random Forest Roc Curve
True Positive Rate
False Positive Rate
AUC=1.0
K Nearest Neighbors Roc Curve
True Positive Rate
False Positive Rate
AUC=0.9889423076923077
Neural Networks Roc Curve
True Positive Rate
False Positive Rate
AUC=0.992628205128205
Gradient Boosting Roc Curve
True Positive Rate
False Positive Rate
AUC=0.9671474358974359