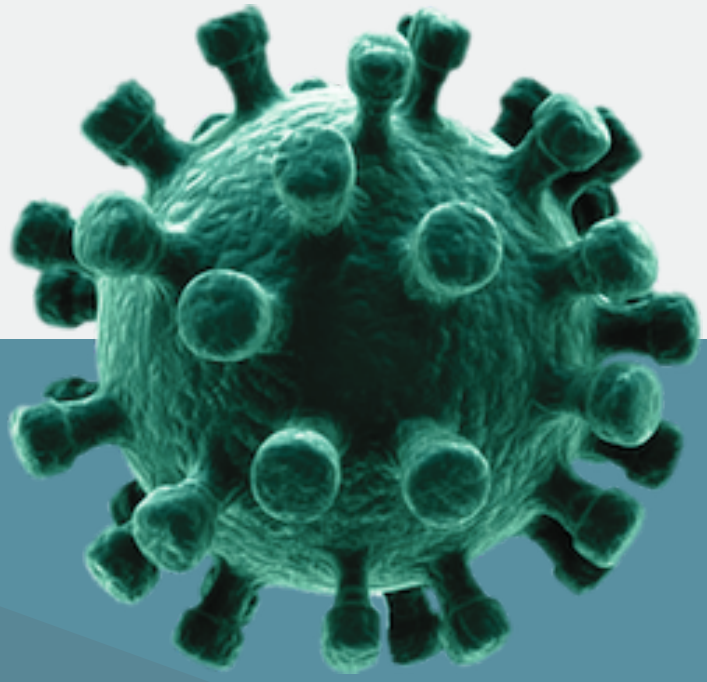
NOVA IMS Information Management School

PROJECT REPORT

The Smith Parasite

- 2022 / 2023 -



Group 41

Carolina Costa	20220715
Inês Castro	20220156
Pedro Pereira	20220684
Rúben Serpa	20221284

Abstract: Recently a new virus was discovered in England and there are no certainties of what leads a patient to suffer or not from it. So, this Machine Learning study was made to analyse and transform the features available as needed and build a model that predict if a patient will suffer, or not, from the Smith Disease. The data was cleaned and treated in several ways, considering the missing values, duplicates, outliers, and some incoherencies. After using different feature selection methods, several models were tested with the selected features until reaching the final model made with Gaussian Process which turned out to be the better performing algorithm for this specific problem, with a F1-Score of 1, on Kaggle.
Keywords: Machine Learning, Feature Selection, Smith Disease, Gaussian Process, F1- Score.

Table of Contents

1. Introduction

Smith Parasite is a new disease, that has recently been discovered and has affected more than 5000 people. Although fever and fatigue are the most common symptoms of the disease, some patients are asymptomatic. The goal of this project is to build a predictive model that can accurately predict which patients are more likely to suffer from the disease. The target variable is *Disease* and can either be 1 – the patient is sick –, or 0 – the patient does not have the disease. The training data, composed of 800 observations, will be the dataset used to train the models and the test dataset (225 observations) will be used to assess how well the model performs on unseen data. Knowing all this, the project will be moved by the goal of understanding what are the relevant variables to predict “Disease”, and what is the best model to apply.

2. Data Exploration

First of all, it was verified that the 3 databases had the same size and the same patients, consequently it was decided to join all of them to facilitate the data manipulation. The database with all data together was called “train_dataset”.

When exploring the available data (Table 1), the first step was to check data types, missing values and duplicates for each variable. Considering that **duplicates** are occurrences that may cause overfitting, meaning that the model will overlearn for the instances that are duplicated more in comparison to the others in the dataset. When looking for these, were revealed 0 duplicated records.

Moving on to the **checking of missing values** in the "train_dataset", after replacing by *NaN* the empty cells, it was possible to conclude that the variable educations had 13 missing values which were filled in later on, reducing bias.

Through the analysis of descriptive statistics of each variable, Table 2 in annex, it was proved that all the variables that needed to be greater than 0 (*Height, Weight*, *High_Cholesterol*, *Blood_Pressure*, *Mental_Health*, *Physical_Health*) have no negative values. Furthermore, it was revealed the variable "Name" had only 799 unique values and the "Mr. Gary Miller" is repeated once, in spite of this after a more detailed analysis, it was verified that they correspond to different people. Regarding the variable called "Birth Year" which has a minimum value of 1855, this was considered an input error because of the difference between the minimum and the 25th percentile is considerably high. Concerning the *High_Cholestrol* variable, it was proved that it has a high standard deviation. Moreover, the difference between the maximum and the 75th percentile on the *High_Cholesterol* and the *Blood_Pressure* variables are noticeable. Therefore, a deeper analysis of outliers is needed. In addition, a count of the various values that each variable could take was carried out and only the variable *Region* presented an inconsistency at the level of the writing of the London value, incorrectly assuming two different regions so this was later adjusted. For the remaining numerical variables, the minimum, maximum, mean and standard deviation seem to be in accordance. For the categorical variables, it is possible to verify the correspondent number of classes.

Another takeout was that the dataset was balanced. The training dataset had 800 observations, 49%

of which had “0” on the target variable *Disease* and 51% had “1”.

Moving on to the data visualization, the dataset was divided into metric and non-metric features since they require different visualization methods. Further the **metric features** are: *Birth_Year, Height, Weight, High_Cholesterol, Blood_Pressure, Mental_Health* and *Physical_Health*. The **non-metric features** are: *Name, Disease, Region, Education, Checkup, Diabetes, Smoking_Habit, Drinking_Habit, Exercise, Fruit_Habit, Water_Habit*.

Conductive to explore metric features visually, histograms were plotted to check distributions and to identify possible **outliers**, resorting to the use of Box-Plots (Figure 1). In statistics, an outlier is an observation point that is distant from other observations and may be due to variability in the measurement or it may indicate experimental error. Therefore, these outliers are sometimes excluded from the dataset. By analysing them, it was assumed that *Birth_Year*, *High_Cholesterol*, and *Blood_Pressure* presented outliers with higher values than expected. Afterwards, the relationships between features were checked through heat map of a **correlation matrix**, also with the target variable (Figure 2). According to the matrix, it is possible to claim that there is no clear strong correlation between the metric variables and most of them have low correlation with the variable *Disease,* whereby these results may be influenced by missing values and outliers.

For the second group of features, bar charts were plotted to check if any value of any variable had any clear relationship with the target variable. With these plots, it was concluded that patients who did a Checkup more than 3 years ago, do not have the habit of eating fruit, do not practice exercise, consume alcohol every day and who have diabetes or had it during pregnancy seem to be more likeable to have the disease (Figure 3).

3. Pre-processing

3.1. Treating missing values and outliers

When pre-processing the available data, it started by filling the missing values. As it was verified that the education variable had 13 missing values, these values were filled with the mode since it is a categorical variable. The handling of missing data is essential during this phase as many machine learning algorithms do not support missing values, it reduces bias, and helps to produce suitable models.

Next, just like missing values, were considered approaches to handle outliers as it is important that the model does not learn from instances that are often not real. Before applying any approach, the graphs of the variables were analysed in detail, in order to verify their distributions and understand which records represented outliers, through Box-Plots and Histograms. For each feature, a range of values was classified as outliers. Although there was an option to exclude them, different approaches were tried to correct them in order to most of the records so as not to miss a high percentage of values.

Using **Manual Detection**, one of the cases considered as an outlier is having a blood pressure greater than 180, since in these cases you suffer from severe third-degree hypertension, running the risk of stroke and therefore it would not make sense to be able to participate in the study. Other assumptions considered as outliers were an individual be born after 1930 and having a cholesterol level greater than 350, as this is

considered untypical. With this approach, 5% of the observations would be removed, which is sustainable for the continuation of the study, but it was decided to check how the IQR could contribute in order to avoid human error.

The **IQR method**, or Interquartile Range, is a statistical concept that describes the spread of all data points within one quartile of the mean, or the mid-50% range, according to Grant (2022). Regarding this method more than approximately 8% of the observations would be removed, which is not sustainable for the continuation of the study, as a large part of the dataset would be eliminated. More than that this, the method did not show results as satisfactory as the Manual approach.

Finally, it was decided to try a combination of both Manual and IQR methods to provide a more robust outlier detection technique, removing only 4% of the data. Although 4% outiler removal would work, two replacing approaches were tested: median and K- Nearest Neighbors imputer. The median replacement approach was chosen since it provided the best F1-Score. After treating the outliers, the data was distributed as illustrated in Figure 4.

3.2. Feature Engineering

According to Patel 2021, feature engineering is a machine learning technique that leverages data to create new variables that are not in the training set with the goal of simplifying and speeding up data transformations while also enhancing model accuracy. In order to promote useful and relevant information, the variable *Birth_Year* was transformed into *Age*. Moreover, to give more significance to the fact of not smoking, the variable *Smoking_Habit* became to *Non-Smoking Habit*, so when the individual has a value equal to 1 it means that he does not smoke and is therefore healthier. Some Information was extracted on the gender of each individual through the variable *Name* being classified as a man if it contains "Mr.", leaving space for the variable *Gender* to be created.

Conforming to Patrick 2019, feature transformations can include aggregating or combining attributes to create new features, depend on the problem at hand but averages, sums and ratios over different groupings can better expose trends to a model. In this sense, the variable *Health Score* was created as the sum of *Mental Health Score* and *Physical Health Score*, taking values between 0 and 2. Furthermore, a variable named *Habit Score* was created, which generally reflects how healthy the individual's habits can be considered, in a range from 0 to 5 where a value equal to 5 would mean that he is very healthy. More specifically, this variable consists of the sum of all the scores of the variables *Drinking Habit*, *Fruit Habit*, *Water Habit*, *Exercise* and *Non-Smoking Habit*. The *BMI (Body Mass Index)* variable was also created, to further reduce the number of variables based on information about each individual's height and weight.

After an analysis, some inconsistencies were found in the region variable, "LONDON" was replaced with London.

Before feature selection, the feature scaling was applied taking into account that the data was at different scales. It was considered important to standardize the data on equal scales in order to avoid major problems if it made sense to use models that take distances into account. The **MinMaxScaler** was used as a standardization method. This method was applied to data versions where outliers were removed. So, in

observations for these datasets, all values which are non-binary numeric were now between 0 and 1. The **Robust Scaler** method was also applied to the metric features before the outliers were removed. Subsequently, a decision was made on which of the two standardization methods was better, taking into account the results measured by the models. It is important to mention that the Standard Scaler was not used because it was not sure that the distributions are normal.

3.3. Feature Selection

Feature selection is a technique whose purpose is to reduce the input space dimensionality. In other words, the relevant features are selected, and redundant features are removed. Accordingly, the model complexity decreases, and the model's generalization ability is enhanced, which results in improved performance and better understanding of the problem.

Considering the purpose of the project and the features available, two main rulings were implemented: carefully select the appropriate methods for metric features and non-metric features and apply three different feature selection methods (Filters, Wrappers, Embedded). The final decision was based on the results from all the methods applied.

In the first place, the **Kendall Correlation Coefficient**, **Chi-squared** and **Mutual Information (MI)** were employed, for the metric and non-metric features respectively, to evaluate the relationship between the independent variables and the target. These are Filter Methods, meaning that they are independent of any learning method, and they rank features based on a certain evaluation criterion.

In what regards the approach used with the metric features, Pearson's method was immediately excluded as requires a linear relationship between two variables. Additionally, although both Kendall's and Spearman's measure monotonic relationships, **Kendall's** is more robust. Figure 5 depicts the correlation matrix between all the metric features and the target variable *Disease*. It is possible to infer that while *Mental_Health* and *Physical_Health* are the variables with the highest correlation with *Disease*; *Height*, *High_Cholestrol* and *Blood_Pressure*, do not present a significant relation. The final decision was to consider as important the variables whose correlation with *Disease* was higher or equal than 0.2.

The **MI** measures the dependency between the variables, this is, how much information one variable gives about other. Figure 6 illustrates the MI gain for each non-metric feature. It is possible to infer that *Checkup* has the largest MI gain (0.14), meaning that *Checkup* gives 14% of the information about the target variable. In conclusion, the variables whose MI score were higher than 0.05 are relevant to predict the target variable.

The **Chi-Square Test** is used to teste the independence of two events. When applied to ML, a higher value of Chi-Square implies that the hypothesis of independence between the dependent and independent variable is incorrected, thus the feature should be selected for the model. Assuming a 95% confidence level, the features selected were the ones whose Chi-square value falls in the rejection region, thus rejecting the Null Hypothesis of independence between the two variables. In summary, the criterion used was to maintain the feature which meets the following condition: p-value lower than alpha=0.05. Table 4 outlines the main conclusions from this test for the dataset.

From these three methods, some insights can be taken to decide which variables are relevant for the model. Nevertheless, filter methods do not consider the interaction with the classifier and ignore interaction effects between the different variables. In the interest of overcoming these two issues, the following step was to apply the **Recursive Feature Elimination (RFE)** method, which is a wrapper-type feature selection algorithm.

The RFE fits a model and eliminates the weakest features until a pre-defined number of features is achieved. The model chosen was the Decision Tree Classifier as it operates with both numerical and categorical data. Since the number of features is not known in advance, a cross-validation technique (StratifiedKFold) is used with RFE to score different feature subsets and then select the bundle of features with the highest score. This technique aims to eliminate dependences and collinearity that exist in the model by repeatedly creating models while keeping aside the best or worst performing feature at each iteration. RFE is effective to select the relevant features; nonetheless, it demands some computational power and is susceptible to over-fitting. Table 5 lists the RFE results for each variable.

Finally, it was applied one Embedded Method- **the Lasso Regression**. In this case, the feature selection process is rooted in the model building phase. The Lasso algorithm is a regularization algorithm, meaning that, a 'penalty' term is added to the least-squared loss function of the linear regression to decrease weights against complexity. The less important features are penalized by the Lasso Regression, which may result in some of the weights becoming zero. Ultimately, the **LassoCV** method was used, as it finds the optimal parameters for a Lasso model using cross-validation. The variables with a coefficient different from 0 are the ones selected by the algorithm (Figure 7).

The main conclusions from the methods previously mentioned are shown in Table 6Table 7. For the cases in which the final decision was not heterogenous among the feature selection methods, different subsets of features were tested using a Decision Tree Model (Table 8*).* Depending on the average train and test accuracy score, the final decision would be to include or not the variable in the final set of features. As an additional assessment, the mean accuracy was also checked with or without the 'Include in the model' features (Table 9). The final selected features were *Age*, *High_Cholesterol*, *Physical_Health_Score*, *Diabetes*, *Fruit_Habit*, and *Checkup*, encoding the nominal *Checkup* feature with One-Hot-Encoding.

4. Modelling

In order to obtain the best predictions, it is recommended to try different models, since its performance depends on the data. Additionally, all classification problems should start with Logistic Regression, since it is the simplest and quickest model, being characterized by using a logistic function to model the dependent variable.

The first step was to test the following models **Logistic Regression**, **K-Nearest Neighbours (KNN) Classifier, Multi-Layer Perceptron Model (MLP)**, **Decision Tree (DT)**, **Support Vector Machine (SVM)**, **Radius Neighbours (RN) Classifier**, and **Gaussian Process (GP) Classifier**, with the default parameters in order to select the best 4 models to hyperparameter tuning. To understand how well each model can predict the outcome of unseen data, the K-fold cross-validation method was used, thus obtaining an

average accuracy and F1 score for the training and validation dataset. By analysing the Figure 8, which compares the scores of the different algorithms, the following decisions were determined: 1) Optimize the DT and the GP Classifier, as both present the best payout between train and validation score; 2) Although MLP performs well, it is computationally expensive and heavy to reparametrize. Therefore, it is more convenient to use SVM, which accomplished similar results; 3) Include RN Classifier instead of KNN. Just like K-Nearest Neighbours (KNN), RN belongs to the family of Instance-based algorithms. Being an extension of KNN, the model stores the entire training dataset and, at prediction time, the most common class label (mode) of the neighbours of each new example is assigned as the prediction. However, unlike KNN, which uses a defined number of neighbours, this model operates with the neighbours that are within a defined radius. To sum up, the DT, GP, SVC and RN are the models chosen for the next phase, which aims to maximize the validation score and reduce or mitigate overfitting. The overfitting issue is a key factor to have accurate predictions for never-before-seen data (like Kaggle's).

In view of optimizing the models' parameter, the Grid Search algorithm was used, which is a process that explores meticulously through a subset of hyperparameters previously defined.

Decision Trees are a powerful tool for classification; however, they are prone to overfitting. Considering this, first a pre-pruning approach was taken, meaning that, the parameters '*criterion*', '*max_depth*', '*min_samples_split'*, '*min_samples_leaf*', '*max_features'*, and '*min_impurity_decrease'* were adjusted. The subset of parameters used in the Grid Search was based on the analysis of the Figure 9, more precisely, by identifying the best values considering the balance between the score for both train and validation and overfitting. Regarding the parameter '*spliter'*, the '*best'* was chosen to assure the best model architecture is obtained. Regarding DT, there is one issue relevant to point out, which is the fact that the model's performance with the default parameters either in terms of validation score or overfitting, is superior to the performance after reparametrizing. This was not expected, as an increase in the deepness of a DT is prone to overfitting. One possible explanation relates to the complexity of the problem and the dataset in question, which requires higher levels of depth to obtain accurate predictions.

For the **GP Classifier model**, the most important hyperparameter is the '*kernel'* (Brownlee, 2020). As such, the performance of the model will be evaluated with the following kernels: RBF, DotProduct, Matern, RationalQuadratic, and WhiteKernel.

On the other hand, in the case of **SVM**, all the parameters were tested within a reasonable range, which was decided based on Figure 10, which depict how the model behaves with different combinations of the 'gamma' and 'C' values.

Lastly, the **RN Classifier** works similarly to the KNN, but instead of using the k nearest neighbours, it uses a radius to make the classification. In what regards the parameter 'algorithm', the brute-force was exclude due to its time expense. Also, for the argument 'metric', the two most well-known methods to compute distance were tested (Manhattan and Euclidean). The Table 10 summarizes the tested parameters and final choices for each one of the formerly referred models.

Finally, in the ambition of improving the results, Ensemble techniques were applied. These techniques normally produce better results, since they combine multiple ML models, aiming to obtain more accurate results. The three commonly used ensemble models are bagging, boosting, and stacking.

In bagging, a Decision Tree model was used since it is an unstable classifier, meaning that small changes in input training samples may cause dramatically changes in output classification rules. In opposition, in boosting, the base models should have low variance but high bias, therefore, a small decision tree was used (with maximum depth equals to 8). Additionally, to perform stacking, the SVM model and RN classifier were used since both offer the best trade-off between performance and time consumption.

5. Assessment

After improving the performance of the best four models and applying ensemble techniques, a comparison of the average **F1 scores** was made attempting to choose the final model for the project. It is important to highlight that the parameter random state was set equal to 41 to consistently obtain the same results every time the code is run.

The F1 score is an "harmonic mean of the model's precision and recall", thus being a good measure of the model's accuracy. Additionally, in the diseases' context, false negatives are of greater importance, and this type of score considers both false positives and false negatives. In order to obtain a more generalized and non-bias result, a 10-fold strategy is applied with stratification. For each fold, the data is split between train and validation and fitted to the model in question to predict the classification result for the validation dataset. Afterwards, the average score is computed from a mean of the individual scores of each fold.

The average scores for the train and validation datasets are illustrated on Figure 11. The models with the highest performance were **Gaussian Process Classifier** and **Stacking Ensemble** on **Support Vector Machine** and **Radius Neighbours Classifier**. Not only both presented high average F1 scores in train and validation dataset, but also do not have results with overfitting. The **Radius Neighbours** and **Bagging** and **Boosting Ensemble** with **Decision Tree** also achieved satisfying results, however, the score difference between train and validation is higher, which reflects in overfitting. Finally, the Decision Tree was the poorest performance model, which was already expected due to the difficulty in reducing overfitting while maintaining the score.

The choice between **GP classifier** and **Stacking Ensemble** on **SVM** and **RN** classifier consisted on: first comparing the results of each model on Kaggle, which was the same (100%) for both models; and second a deeper analysis on some metrics such as **Confusion Matrix**, **ROC curve**, and **Precision-Recall curve**. Aiming to compute the previously mentioned metric, the full dataset was partitioned in train (75%) and in test (25%).

Regarding the **Confusion Matrices** (Figure 12*),* since the context in question concerns diseases, it is important to guarantee that individuals who have the disease are not identified as healthy (False Negatives). In this case, both models performed similarly, having only 2 false negative cases. On the other hand, when predicting positive cases, **GP** was more accurate (94 cases correctly classified compared to 91 cases using **Stacking**). Hence, **GP** obtained better predictions. Furthermore, according to the **ROC curve**, the **GP** has

a better performance, since it is closer to the top-left corner, thus having a larger Area Under the Curve (AUC) (Figure 13*)*. More specifically, while GP has an AUC of 0.97, Stacking has an AUC of 0.96. Although the Precision-Recall curve do not consider the model ability to correctly identify people without the disease, this metric was also used as a complement to the analysis (Figure 14). Once more, the GP is the chosen model.

Notwithstanding the previous analysis, it is important to highlight that the differences in performance between the Gaussian Process and Stacking Ensemble on **SVM** and **RN** classifiers are subtle. Consequently, the time to perform the model should be an issue to contemplate in the analysis, meaning that, in a hypothetical scenario of a larger dataset, GP model would be much more time expensive. For instance, in the case of the project which has 800 observations, it takes 24.6x more time than the Stacking. However, in the project scope the performance will prevail over model's execution time.

In conclusion, it was decided to proceed with the Gaussian Process Classifier as the final model, with an average F1 score of 99.76% in validation and 100% in test (in Kaggle).

6. Conclusion

Throughout the development of this project, it was faced challenges that required not only Machine Learning knowledge acquired during classes but also some investigation. To improve on the initial score of 0.8 in Kaggle, it was necessary to rework most of the project to fix the problems identified through A/B testing at times. It was used a large number of Data Exploration, Pre-Processing and Modelling methods, thus reaching the conclusion that a combination of several methods can produce a better result.

One challenging step of the project was the choice of the six most relevant features, which was accomplished by performing not only different feature selection techniques (Filter, Wrapper, and Embedded methods), but also by manually evaluating the performance of each subset of possible relevant features in a Decision Tree model.

Having the data prepared and cleaned, the modelling stage started by testing multiple default parameterized models (Logistic Regression, KNN, MLP, Decision Tree, Support Vector Machine, Radius Neighbours, and Gaussian Process) to select the ones to optimize. Considering factors such as the validation score, overfitting, and execution time, it was decided to proceed with the DT, Radius Neighbours, Gaussian Process, and SVM. The optimization process consisted of first defining the range of values to test in each parameter, and then applying the Grid Search Algorithm. Furthermore, Ensemble Techniques were also employed aiming to improve the results: Bagging and Boosting with Decision Trees and Stacking with SVM and RN.

In order to assess which models have an enhanced performance, the F1 score for train and validation was compared, resulting in choosing the Gaussian Process and Stacking as the outperforming classifiers. Since both presented a similar score, additional comparison metrics were exploited, such as Confusion Matrix, ROC curve, Precision-Recall curve, and execution timing. By completing this deeper analysis, the conclusion was to elect the Gaussian Process as the classifier which best predicts the patients who have the Parasite Disease.

7. Annexes

7.1. Theoretical Explanation of Algorithms

Gaussian Process Classifier:

Since Naïve Bayes assumes the variables are independent, which does not happen in the dataset, another probabilities-based model, **Gaussian Process Classifier** was tested. Performing under a probabilities-based algorithm, it uses probabilities to decide on which class the data fits in. From the various inputs, unlike other probabilistic algorithms, it samples multiple functions that are likely to describe the data distribution, and to each sample, a noise variance is applied. To account with uncertainty, all the noises are averaged, resulting in a singular distribution (Grant, 2022). In order to control the outputs, the algorithm provides a kernel hyperparameter that, from calculating the similarities between the inputs, through a covariance function, determines what functions are more likely to be sampled.

The kernel by default is a Radial Basis Function. This stationary kernel, also known as the "squared exponential" function, is parameterized by a length scale parameter (l). The function is defined as following:

$$k(x_i, x_j) = exp\left(-\frac{d(x_i,x_j)^2}{2l^2}\right),$$

where $d(x_i, x_j)$ is the Euclidean distance between $x_i \; and \; x_j$.

From this kernel, the Rational Quadratic is the defined, being characterized as an infinite sum of RBF kernels, each with different length scales. Additional to the l parameter, also has a scale mixture parameter (α) as defined in the following formula:

$$k(x_i, x_j) = (1 + \frac{d(x_i, x_j)^2}{2\alpha l^2})^{-\alpha}$$

Besides the kernels explained above, WhiteKernel, DotProduct and Matern are the most common and can be defined as below, respectively:

$$k(x_1, x_2) = noise\_level \text{ if } x_i == x_j \text{ else } 0,$$

where $noise\_level$ is the variance defined by parameter,

$$k(x_i, x_j) = \sigma_0^2 + x_i \cdot x_j,$$

where σ_0^2 controls the inhomogeneity of the kernel,

$$k(x_i, x_j) = \frac{1}{\Gamma(\nu)2^{\nu-1}}(\frac{\sqrt{2\nu}}{l} d(x_i, x_j))^{\nu} K_{\nu}(\frac{\sqrt{2\nu}}{l} d(x_i, x_j)),$$

where $d(x_i, x_j)$ is the Euclidean distance between $x_i \; and \; x_j$, K_{ν} is a modified Bessel function and $\Gamma(\nu)$ is the gamma function.

7.2. Figures

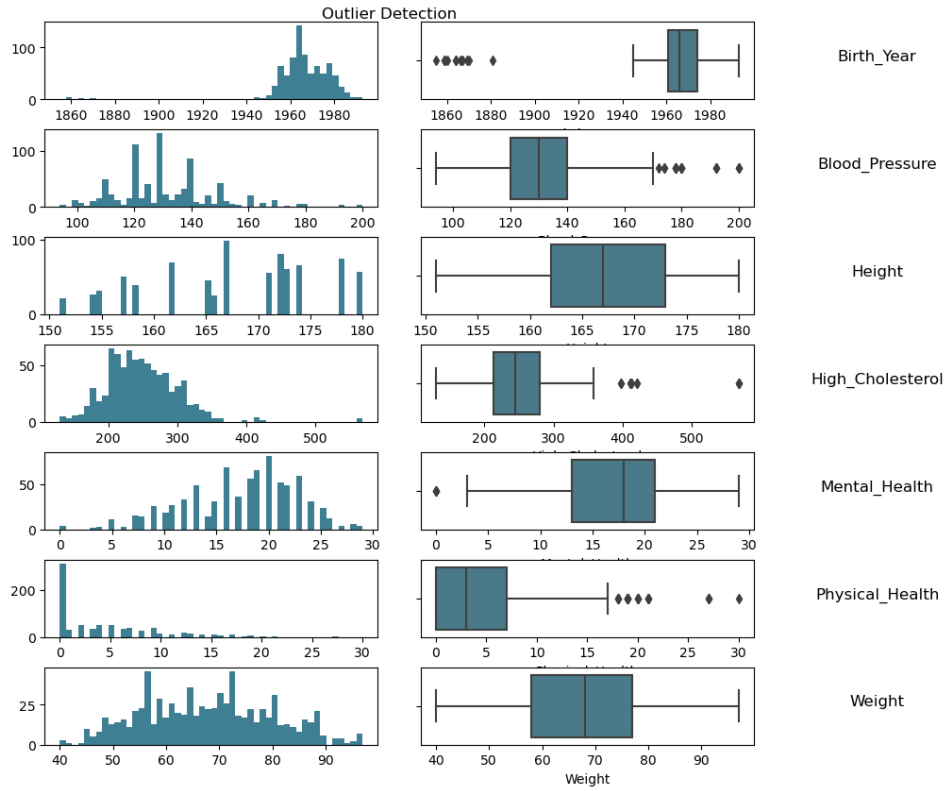


Figure 1: Outlier Detection Plots (Boxplots and Histograms)

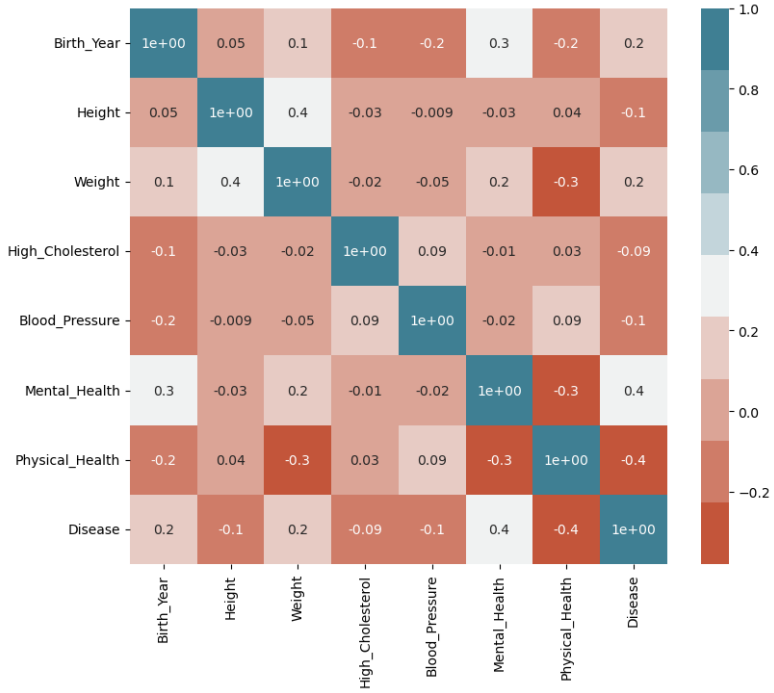
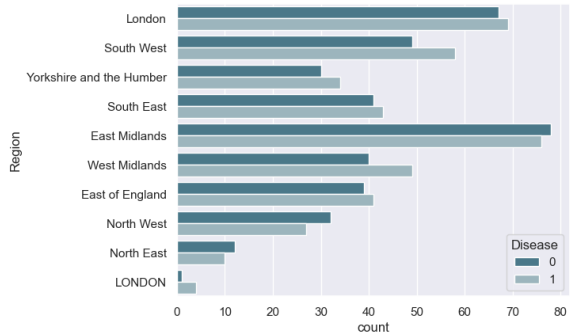
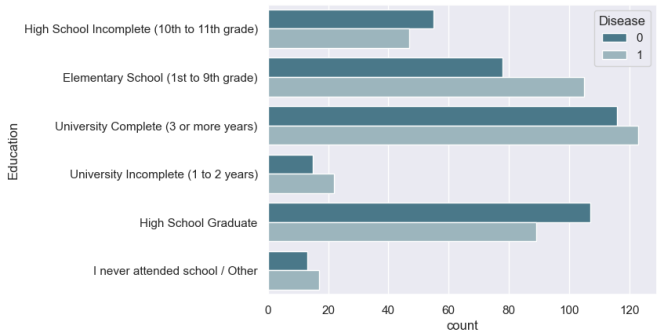
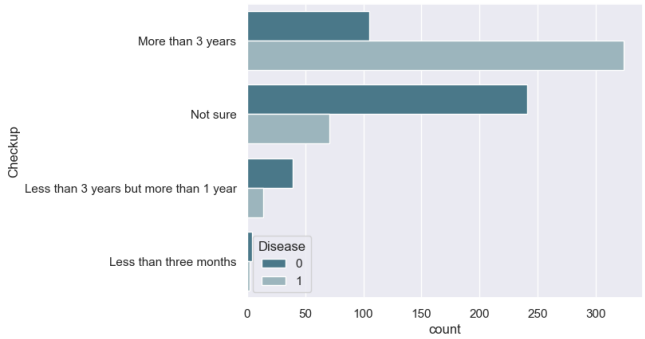
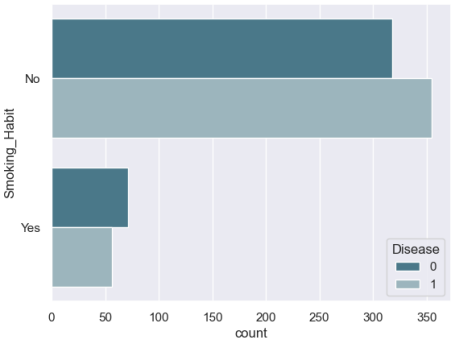


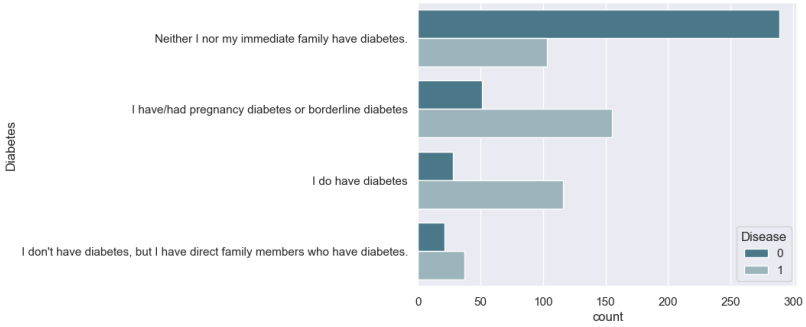
Figure 2: Correlation Matrix representing the correlation between the metric features and the target variable

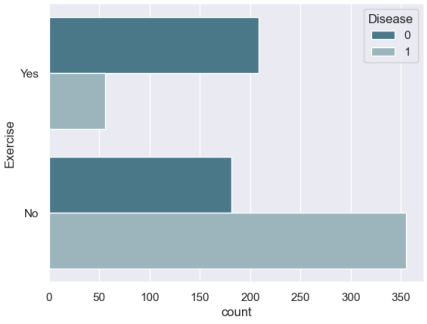

London
South West
Yorkshire and the Humber
South East
East Midlands
West Midlands
East of England
North West
North East
LONDON
Region
Disease
0
1
0
10
20
30
40
50
60
70
80
count

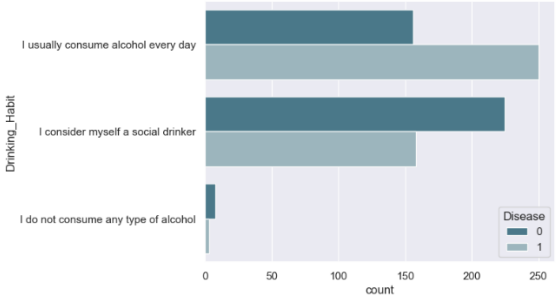

High School Incomplete (10th to 11th grade)
Elementary School (1st to 9th grade)
University Complete (3 or more years)
University Incomplete (1 to 2 years)
High School Graduate
I never attended school / Other
Education
Disease
0
1
0
20
40
60
80
100
120
count

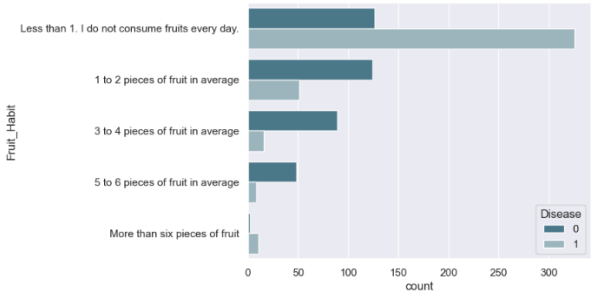

More than 3 years
Not sure
Less than 3 years but more than 1 year
Less than three months
Checkup
Disease
0
1
0
50
100
150
200
250
300
count


No
Yes
Smoking_Habit
Disease
0
1
0
50
100
150
200
250
300
350
count


Neither I nor my immediate family have diabetes.
I have/had pregnancy diabetes or borderline diabetes
I do have diabetes
I don't have diabetes, but I have direct family members who have diabetes.
Diabetes
Disease
0
1
0
50
100
150
200
250
300
count


Yes
No
Exercise
Disease
0
1
0
50
100
150
200
250
300
350
count


I usually consume alcohol every day
I consider myself a social drinker
I do not consume any type of alcohol
Drinking_Habit
Disease
0
1
0
50
100
150
200
250
count


Less than 1. I do not consume fruits every day.
1 to 2 pieces of fruit in average
3 to 4 pieces of fruit in average
5 to 6 pieces of fruit in average
More than six pieces of fruit
Fruit_Habit
Disease
0
1
0
50
100
150
200
250
300
count

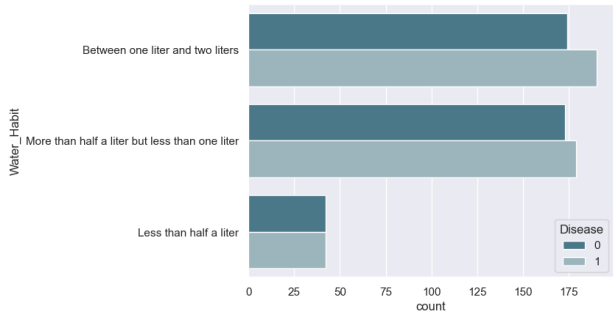


Figure 3: Countplot for Categorical Features

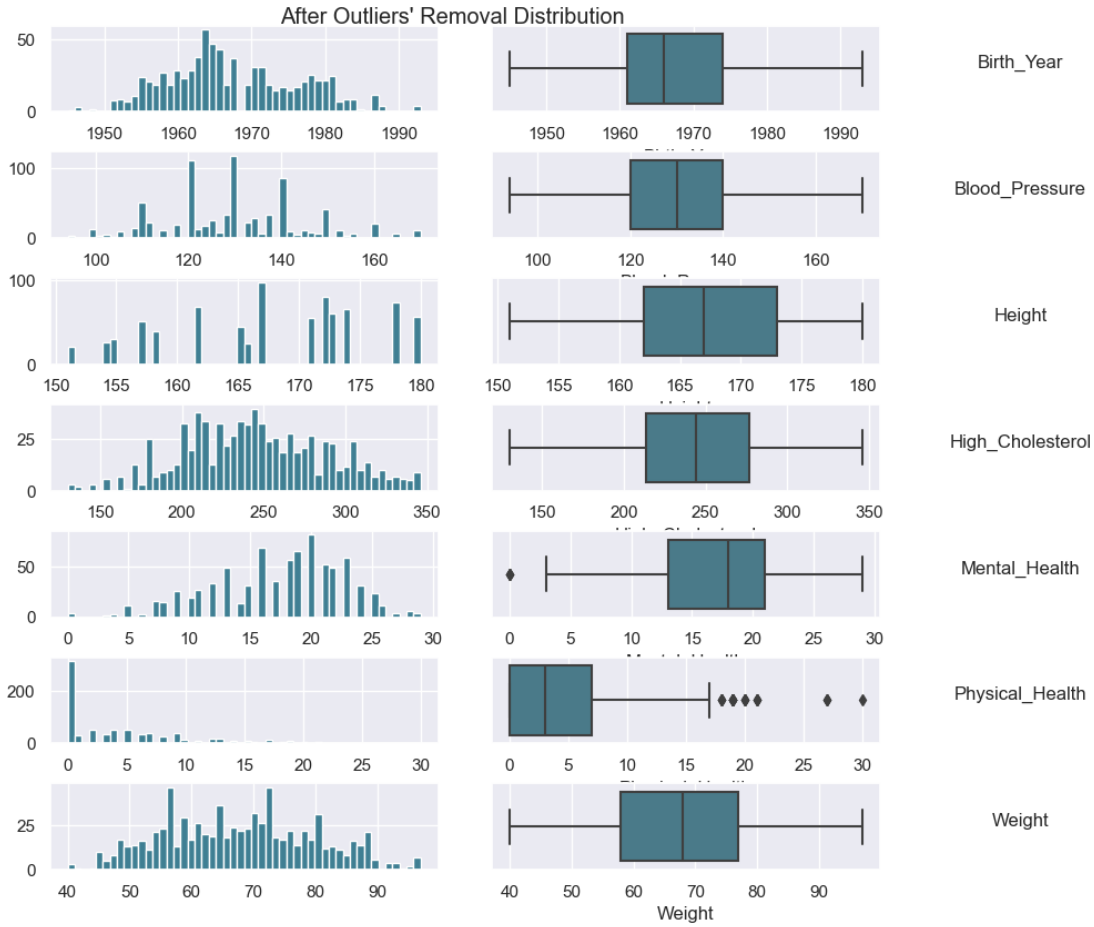


Figure 4: After Outliers' Removal Distribution

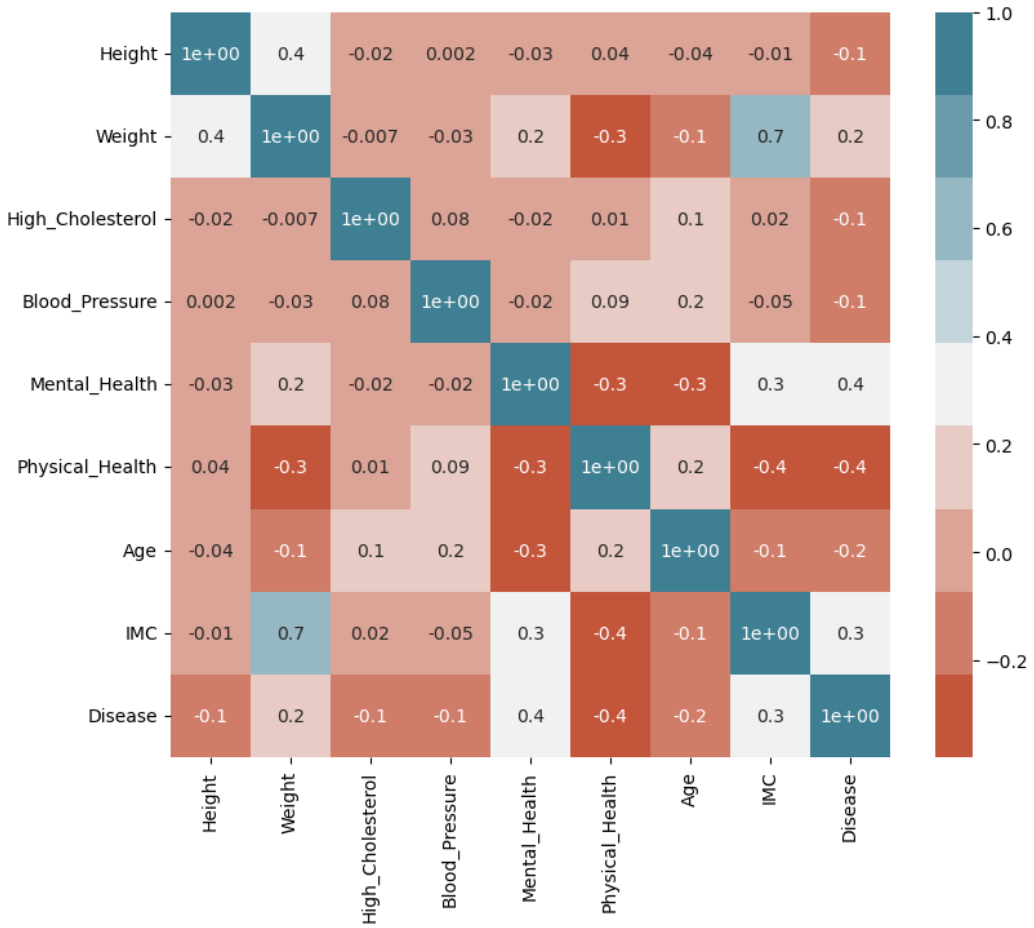


Figure 5: Filter Methods, Kendall Correlation Coefficient Matrix for the metric features

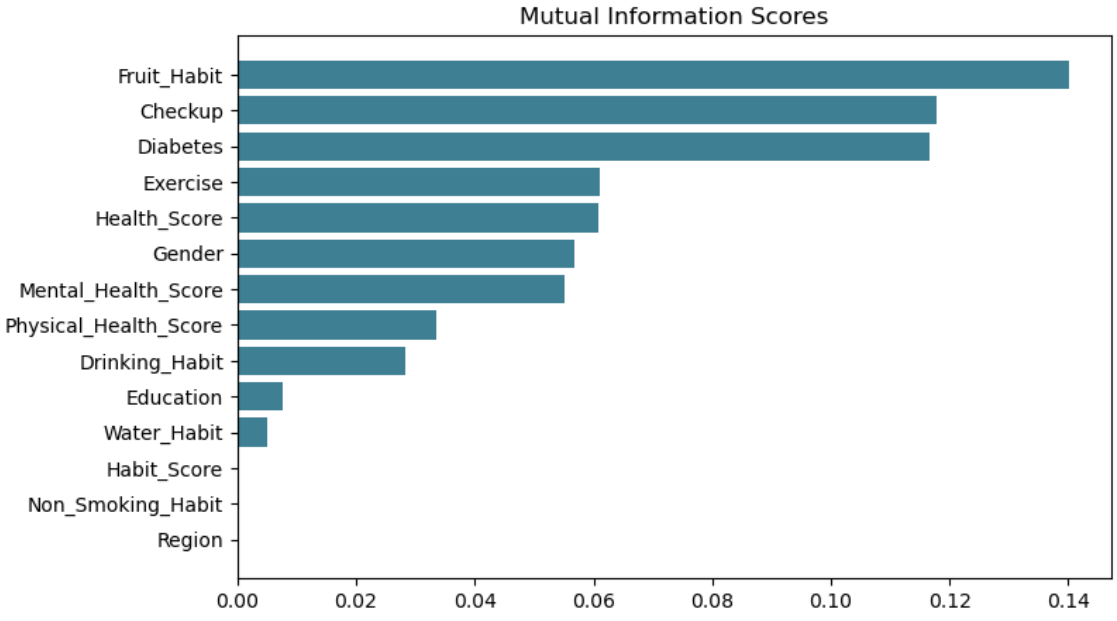


Figure 6: Filter Method: Mutual Information Scores for the non-metric features

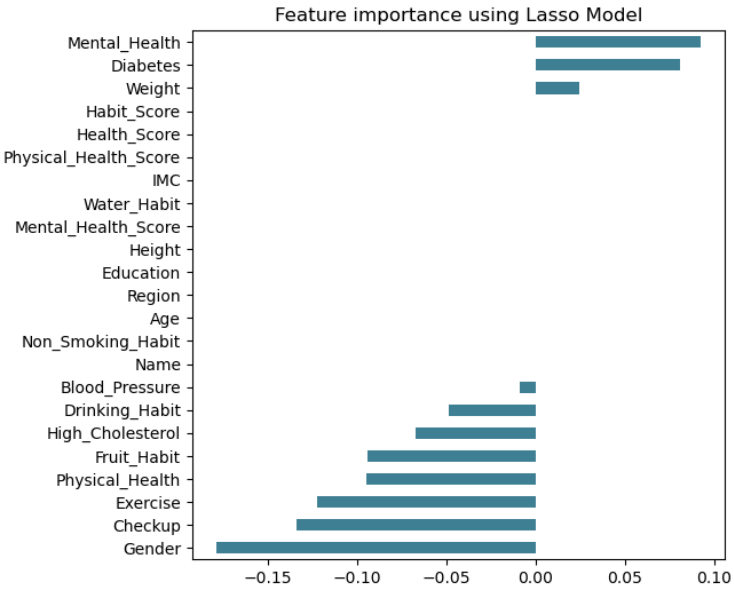


Figure 7: Embedded Methods: Feature importance using Lasso Classification

Figure 8: Training and Validation Scores for default parameterized models

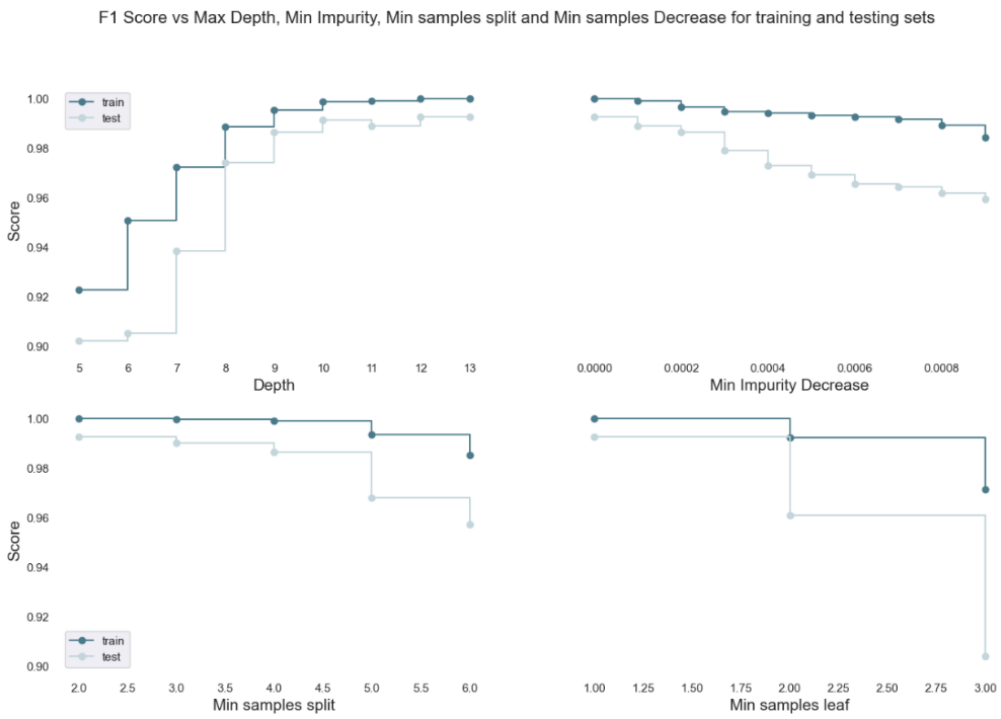


Figure 9: Decision Tree Parameters Optimization

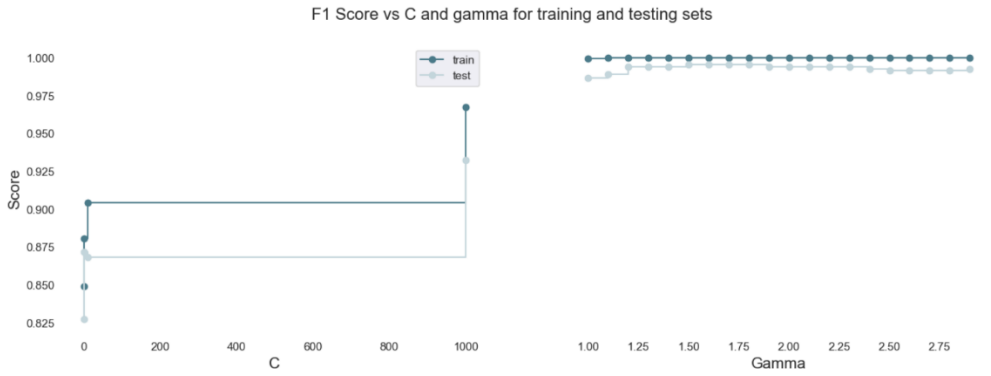


Figure 10: Support Vector Machine parameters optimization

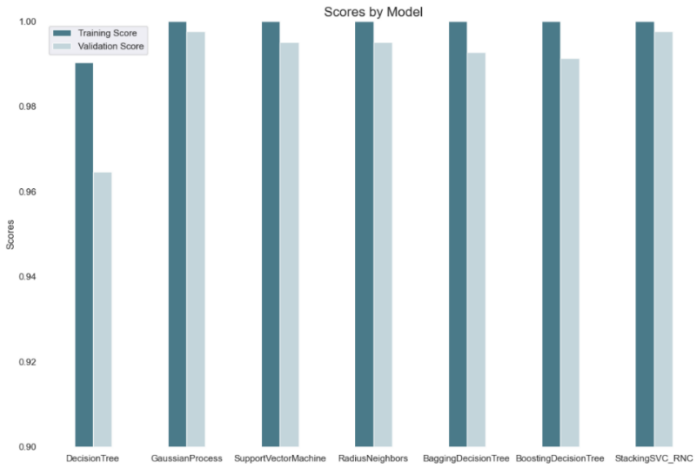


Figure 11: Barplot with the F1 scores of each model

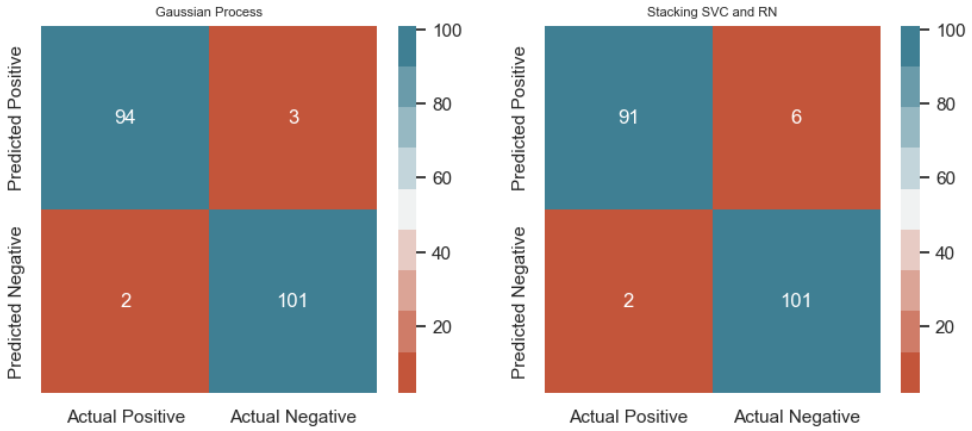


Figure 12: Confusion Matrixes

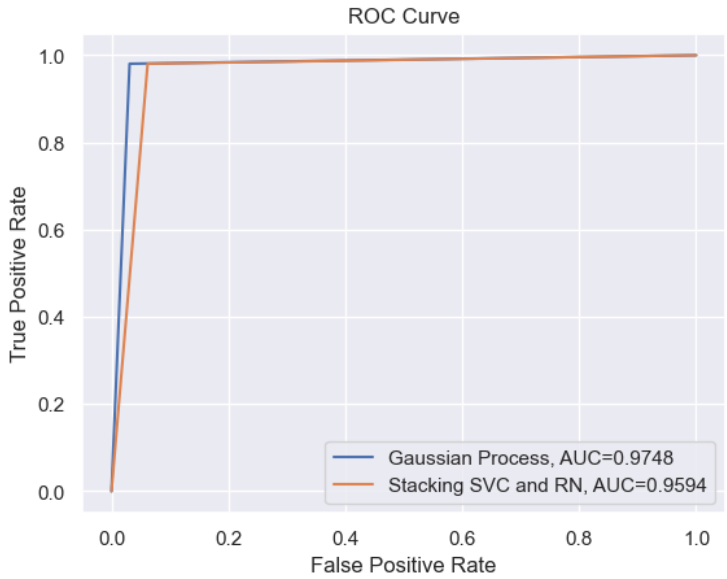


Figure 13: Receiver Operating Characteristic (ROC) Curve

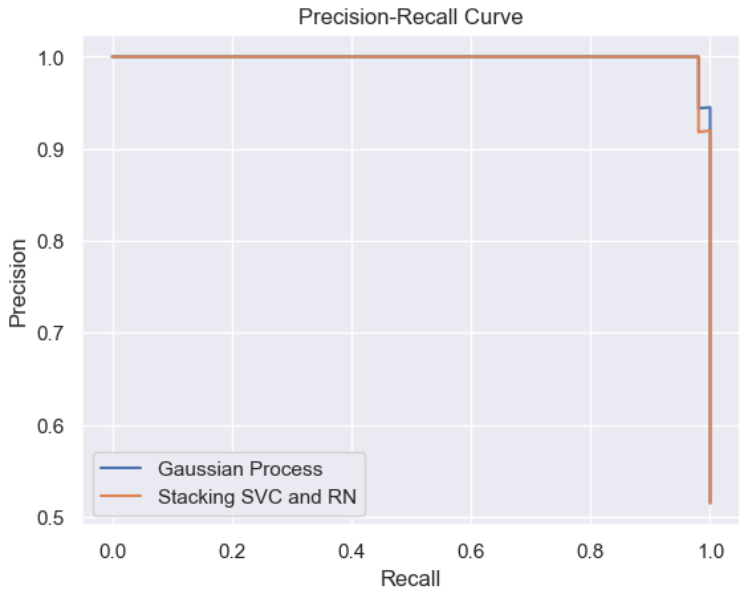


Figure 14: Precision-Recall Curve

7.3. Tables

Attribute	Description
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)
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?
Diabetes	Answer to the question: (Ever told) you or your direct relatives have diabetes?
HighCholesterol	Cholesterol value
BloodPressure	Blood Pressure in rest value
Mental Health	Answer to the question: During the past 30 days, for about how many days did poor physicalor 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?
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?

Table 1: Data Dictionary

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Name	800	799	Mr. Gary Miller	2	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Birth_Year	800.0	NaN	NaN	NaN	1966.04375	15.421872	1855.0	1961.0	1966.0	1974.0	1993.0
Region	800	10	East Midlands	154	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Education	787	6	University Complete (3 or more years)	239	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Disease	800.0	NaN	NaN	NaN	0.51375	0.500124	0.0	0.0	1.0	1.0	1.0
Height	800.0	NaN	NaN	NaN	167.80625	7.976888	151.0	162.0	167.0	173.0	180.0
Weight	800.0	NaN	NaN	NaN	67.8275	12.11347	40.0	58.0	68.0	77.0	97.0
High_Cholesterol	800.0	NaN	NaN	NaN	249.3225	51.566631	130.0	213.75	244.0	280.0	568.0
Blood_Pressure	800.0	NaN	NaN	NaN	131.05375	17.052693	94.0	120.0	130.0	140.0	200.0
Mental_Health	800.0	NaN	NaN	NaN	17.345	5.385139	0.0	13.0	18.0	21.0	29.0
Physical_Health	800.0	NaN	NaN	NaN	4.55875	5.449189	0.0	0.0	3.0	7.0	30.0
Checkup	800	4	More than 3 years	429	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Diabetes	800	4	Neither I nor my immediate family have diabetes.	392	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Smoking_Habit	800	2	No	673	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Drinking_Habit	800	3	I usually consume alcohol every day	406	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Exercise	800	2	No	536	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Fruit_Habit	800	5	Less than 1. I do not consume fruits every day.	452	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Water_Habit	800	3	Between one liter and two liters	364	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Table 2: Descriptive Statistics

Name	Description
Age	Age was extracted from *Birthday* variable.
IMC	The quotient between weight and squared height gives the Body Mass Index of a certain patient.
Gender	Gender was extracted from *Name* variable.
Mental Health Score	Mental Health was filtered and a “1” was assigned if the has a value higher than 15, and “0” otherwise
Physical Health Score	Physical Health was filtered and a “1” was assigned if the has a value higher than 15, and “0” otherwise
Health Score	The sum of the *Mental Health Score* and *Physical Health Score variables.*
Habit Score	The sum of the values of all the habit features.
Non-Smoking Habit	*Non-Smoking Habit* is the transformation of *Smoking Habit* variable

Table 3: Created Features Description

	Dependent with Disease?(Chi-squared)
Disease	True
Checkup	True
Diabetes	True
Drinking_Habit	True
Exercise	True
Fruit_Habit	True
Gender	True
Mental_Health_Score	True
Physical_Health_Score	True
Health_Score	True
Name	False
Region	False
Education	False
Water_Habit	False
Non_Smoking_Habit	False
Habit_Score	False

Table 4: Filter Method: Chi-Square Test Result for the non-metric features

	Relevant?
Name	False
Health_Score	False
Physical_Health_Score	False
Mental_Health_Score	False
Gender	False
IMC	False
Water_Habit	False
Exercise	False
Drinking_Habit	False
Non_Smoking_Habit	False
Habit_Score	False
Physical_Health	False
Mental_Health	False
Blood_Pressure	False
Weight	False
Height	False
Education	False
Region	False
Fruit_Habit	True
High_Cholesterol	True
Age	True
Checkup	True
Diabetes	True

Table 5: RFE Analysis

Predictor	Kendall	RFE	Lasso	What to do? (One possible way to "solve")
Age	Keep	Keep	Discard	Try with and without
Blood_Pressure	Discard	Discard	Keep	Try with and without
Height	Discard	Discard	Discard	Discard
High_Cholesterol	Discard	Keep	Keep	Try with and without
IMC	Keep	Discard	Discard	Try with and without
Mental_Health	Keep	Discard	Keep	Try with and without
Physical_Health	Keep	Discard	Keep	Try with and withou
Weight	Keep	Discard	Keep	Try with and without

Table 6: Feature Selection insights for Numerical Data

Predictor	Mutual Information	Chi-Square	RFE	Lasso	What to do? (One possible way to "solve")
Checkup	Keep	Keep	Keep	Keep	Include in the model
Diabetes	Keep	Keep	Keep	Keep	Include in the model
Drinking_Habit	Discard	Keep	Discard	Keep	Try with and withou
Education	Discard	Discard	Discard	Discard	Discard
Habit_Score	Discard	Discard	Discard	Discard	Discard
Exercise	Keep	Keep	Discard	Keep	Try with and without
Fruit_Habit	Keep	Keep	Keep	Keep	Include in the model
Gender	Discard	Keep	Discard	Keep	Try with and without
Health_Score	Keep	Keep	Discard	Discard	Try with and without
Mental_Health_Score	Keep	Keep	Discard	Discard	Try with and without
Name	NaN	NaN	Discard	Keep	Discard
Non_Smoking_Habit	Discard	Discard	Discard	Discard	Discard
Physical_Health_Score	Keep	Keep	Discard	Discard	Try with and without
Region	Discard	Discard	Discard	Discard	Discard
Water_Habit	Discard	Discard	Discard	Discard	Discard

Table 7: Feature Selection insights for Categorical Data

	Metric Features Testing								Categorical Features Testing					
Checkup	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Diabetes	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Fruit_Habit	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Age		X	X	X	X	X	X	X	X	X	X	X	X	X
Blood_Pressure			X	X	X	X	X	X	X	X	X	X	X	X
High_Cholestrol				X										
IMC					X									
Mental_Health						X								
Physical_Health							X							
Weight								X						
Drinking_Habit									X					
Exercise										X				
Health_Score											X			
M_H_Score												X		
P_H_Score													X	X
Gender														X
Validation Score	**83.78**	**92.74**	**99.26**	**99.02**	**93.46**	**98.52**	**98.88**	**92.84**	**99.14**	**98.91**	**98.52**	**98.77**	**99.38**	**99.13**

Table 8: Testing validation scores with different subsets of features

	Final Features Testing				
Checkup	X	X	X	X	X
Diabetes	X	X	X	X	X
Fruit_Habit	X	X	X	X	X
Age	X	X	X	X	X
Blood_Pressure	X		X	X	
High_Cholestrol		X		X	X
Physical_Health_Score	X	X	X	X	X
Gender		X	X		
Validation Score	**99.38**	**99.13**	**99.13**	**99.14**	**99.27**
Kaggle	**98.90**	**100**	**97.82**	**100**	**100**

Table 9: Testing Validation and Kaggle Score for Final Features

	Parameters	Values	Final Result-GridSearch
Decision Tree	*criterion*	gini, entropy, log_loss	gini
	min_samples_split	range (2,7)	2
	min_samples_leaf	range (1,4)	1
	max_features	auto, sqrt, log2	log2
	min_impurity_decrease	range (0, 0.001) within the interval 0.0001	0
	Max_depth	10	10
Gaussian Process	*kernel*	RBF, DotProduct, Matern, RationalQuadratic, White Kernel	RationalQuadratic
Support Vector Machine	*C*	*10,100,1000*	*1000*
	kernel	*rbf, sigmoid*	*rbf*
	gamma	*range (1,2) within the interval 0.1*	*1.7*
	shrinking	*True, False*	*True*
	probability	*True, False*	*True*
	class_weight	*None, balanced*	*None*
	decision_function_shape	*ovo, ovr*	*ovr*
	break_ties	*True, False*	*False*
Radius-Nearest Neighbours	*radius*	Range (1.3,3)	5
	Weights	Uniform, distance	distance
	Algorithm	Ball_tree, kd_tree	Ball_tree
	Metric	Euclidean, Manhattan	Manhattan
	Leaf_size	Range (10,100)	10

Table 10: GridSearch Hyperparameter tunning for the top 4 models

8. References

Grant, P. (2022, May 12).How to Find Outliers With IQR Using Python | Built In. . Builtin.com. https://builtin.com/data-science/how-to-find-outliers-with-iqr

Patel, H. (2021, September 2). What is Feature Engineering — Importance, Tools and Techniques for Machine Learning. Medium. https://towardsdatascience.com/what-is-feature-engineering-importance-tools-and-techniques-for-machine-learning-2080b0269f10

Remeseiro, B., & Bolon-Canedo, V. (2019). A review of feature selection methods in medical applications. Computers in Biology and Medicine, 112, 103375. https://doi.org/10.1016/j.compbiomed.2019.103375

Jain, D., & Singh, V. (2018). Feature selection and classification systems for chronic disease prediction: A review. Egyptian Informatics Journal, 19(3), 179–189. https://doi.org/10.1016/j.eij.2018.03.002

Zinda, Z. (2021, October 4). Data Science Stats Review: Pearson's, Kendall's, and Spearman's Correlation for Feature Selection. PhData. https://www.phdata.io/blog/data-science-stats-review/

Sampath Kumar Gajawada. (2019, October 4). Chi-Square Test for Feature Selection in Machine learning. Medium; Towards Data Science. https://towardsdatascience.com/chi-square-test-for-feature-selection-in-machine-learning-206b1f0b8223

Brownlee, J. (2020, October). Gaussian Processes for Classification With Python - MachineLearningMastery.com. MachineLearningMastery.com. https://machinelearningmastery.com/gaussian-processes-for-classification-with-python/

Torabi, M., Udzir, N. I., Abdullah, M. T., & Yaakob, R. (2021). A Review on Feature Selection and Ensemble Techniques for Intrusion Detection System. International Journal of Advanced Computer Science and Applications, 12(5). https://doi.org/10.14569/ijacsa.2021.0120566