NOVA IMS Information Management School

# Machine Learning Project

## The Smith Parasite - An Unknown Parasitic Disease


**Group 24**

Álvaro Reis, student number - 20220679

David Martins, student number - 20221006

Diogo Martins, student number - 20221361

Marta Correia, student number - 20220709


**MASTER DEGREE PROGRAM IN DATA SCIENCE AND ADVANCED ANALYTICS**

**2022/2023**

# Index

## Table Index



## Figure Index

## Introduction

The main goal of this project is to build a predictive model that answers the question: “Who are the people more likely to suffer from the Smith Parasite?”. To answer this question, our objective is to build a predictive model that can accurately predict if a patient will suffer, or not, from the Smith Parasite. For this task, we have been provided with a set of sociodemographic, health, and behavioral information obtained from a sample of patients. Throughout the project, the available data will be analyzed and transformed. After that, different algorithms will be tested and accessed to answer the defined question in a more accurate way and find the best model.

## Exploration

We initiated our project by importing all the training and test sets that were made available and created two datasets: “*training_full*” (a dataset that contains all the training sets) and “*test_full*” (a dataset that contains all the test sets). Next, we created a copy of both, to store these original datasets as a way to ensure that we had a reference point for the data and were able to make changes to the data with more confidence.

We started by checking the existing columns in the dataframe, which correspond to the attributes of our data – demographic, health, and habits attributes (Table 1). We noticed the existence of a 'PatientID' and decided to index it in the dataframe as it is an univariate variable (so it would be irrelevant for our model). Alongside this variable, and for similar reasons, we also decided to eliminate the variable 'Name' from our model since it also represents a unique characteristic of each patient.

Through descriptive statistics (Table 2 and Table 3), it was possible to make an initial analysis of the metric and non-metric variables and detect some possible errors. Regarding categorical data, we noticed that there were two values in the variable 'Region' that referred to the same location: ('London' and 'LONDON'). We corrected this problem by changing all the “LONDON” values to “London”. By looking at Table 2, we noticed again the missing values on the variable ‘Education’. It is also worth mentioning the variable ‘Smoking_Habit’ had many observations with the value 'No'. We did not detect any other important anomalies at first sight in the non-metric features.

Regarding the metric features, by looking at Table 3, it was possible to see that the variables 'Weight' and 'Height' seemed to have plausible values in an acceptable range, with no anomalies at first sight. It was also possible to observe that the maximum value of 'High_Cholesterol' was quite high relative to the mode. Furthermore, the 75% quartile value was quite small when compared to this value (75% of the values for 'High_Cholesterol' were below the value of 280.0). However, we decided not to remove this value but rather wait until the outlier analysis. We also observed that the minimum value in “Birth_Year” was probably an outlier, as it was well below the other values for this variable.

Duplicated observations can be problematic and cause model overfitting on the training data and can also cause that the model to not generalize well [1]. As such, we searched for duplicate observations, but found none.

We also checked whether this was a case of imbalanced data and found it was not the case, since the target class did not have an uneven distribution of observations (on the training dataset there were 411 observations where 'Disease' = 1 and 389 observations where 'Disease' = 0). Then, we divided the features into metric (numeric features) and non-metric (categorical features) as they required different approaches in some of the next phases of the project.

After that, we focused on data visualization. Data visualization is an important tool to examine the data for distribution, outliers and anomalies to direct specific testing of your hypothesis, while also providing tools for hypothesis generation by visualizing and understanding the data usually through graphical representation. A good explanatory analysis allied with relevant data visualization is the keys to explore and preprocess the data with success [2] [3]. We used:

- Histograms (Figure 1 and Figure 2) and Boxplots (Figure 3): both were used to check the distribution and skewness (or symmetric) of the data, as well as detecting the presence of outliers. By looking at the histogram and boxplot of the metric features (Figure 1 and Figure 3), it was possible to conclude that all the variables, except 'Weight' and 'Height' probably presented some outliers. By looking at the histogram of the categorical features (Figure 2), at first sight it seemed that there were no problems with these variables.
- Pairwise Relationship (Figure 4) – allowed us to see both the distribution of single variables and relationships between two variables. The values 0 and 1 on "Disease" were displayed in different colors, to make out obvious relationships between certain traits of the population and having the disease.

The pair plot allowed us to visualize the difference in distributions between people with and without the disease in the "Physical_Health" and "Mental_health", suggesting importance of this features. To verify this, two separate heatmaps correlating the numerical and categorical features with the target were later created (Figure 5 and Figure 6). By observing these figures we were able to draw a first insight that there were several variables that seemed crucial to the forecasting process since they had a high Spearman correlation with the target. For example, 'Mental_Health' had a correlation of 0.4 with the target and 'Physical_Health' had a correlation of -0.4. They were followed by the variable 'BMI' which had a correlation of 0.3 with the target (Figure 5). Among the non-metric features (Figure 6), two variables seemed to have a high correlation with the target (0.5): 'Checkup_More than 3 years' and 'Fruit_Habit_less than 3 years but more than 1 year'. This was followed by 'Exercise_No' (with a correlation of 0.4) and 'Checkup_Not_Sure' (with a correlation of 0.3). Based on these findings, we expected that these variables would be included in the final feature selection, and we later confirmed these results during the feature selection phase.

## Preprocessing

The handling of missing data is very important during this phase as many machine learning algorithms do not support missing values, it reduces bias, and is one-step closer to produce powerful suitable models. When we checked for missing values, we noticed that the only variable that had missing values was 'Education' (Figure 7). Since it was a categorical variable and there was a very limited number of missing values, we decided to make the imputation of these values by using the mode, as it is a common statistical missing value technique, that typically requires a short time to compute [4].

We then moved on to handling outliers. Before focusing on applying the outlier removal methods, we checked and analyzed again the graphs of the variables in order to examine their distributions, identify outliers and understand the data better. This was done using box plots and histograms. We used a combination of the Inter-Quartile Range method and a manual removal method. In the manual outlier removal, we used the box plots and histograms to have an idea of which variables were more likely to be outliers. Using the interception of two methods, we hoped we could remove clear outliers, while keeping a good percentage of the data. After the removal we were left with 97.8% of the data.

After that, we proceeded with feature engineering. Age and body mass index (BMI) were created. We considered BMI important since it's known that the immune system is compromised by obesity, as well as other forms of malnutrition [5]. Age was created to replace "Birth_Year" with a feature with more readability (Table 4).

In the next phase we focused on standardizing the numeric data (as having data with different scales could result in impactful problems when working with models that take into account distances) and encoding the categorical data (because most of the models we would use do not work with categorical data). To encode the non-metric variables, we used "One Hot Encoder" (in this categorical encoding method, each category value is converted into a new column and assigned a 1 or 0 – notation for true or false – value to the column [6]). Finally, to standardize the metric features we used "MinMaxScaler" (MinMaxScaler transforms the data by scaling the values to a specific value range – in our case of 0 (minimum of feature) to 1 (maximum of feature) – without changing the shape of the original distribution [7]). After this phase, we used the concatenate pandas' function to join the One Hot Encoder dataframe and the Standard Scaler dataframe into a single one (we did it both for train data and validation data). We ended up having two different dataframes (already scaled and encoded): X_train_norm and X_test_norm.

At this stage of the project, having our final data set already preprocessed, we moved to feature selection. Feature selection is the process of selecting a subset of relevant features for use in model construction. It is important because it can help improve the performance of the model by reducing the complexity and overhead of processing irrelevant or redundant features, and by avoiding

overfitting [8]. Additionally, feature selection can help to identify the most important features in the dataset, which can provide insights into the underlying relationships and patterns in the data. We start this phase by looking for unitary variables by checking the variance. If the variance of any variable was 0 this would mean that this variable was unitary (has the same value in every observation) and, therefore, it would not be important for the model. However, we did not find any unitary variable.

In this phase, we decided to test multiple feature selection methods as it could be beneficial because it would help to confirm the robustness of the results. Different feature selection methods may identify different sets of relevant features, and by comparing the results across multiple methods, we can get a more comprehensive understanding of which features are important for your problem. Additionally, using multiple methods can help to reduce the risk of selecting a suboptimal set of features due to the limitations of a single method. By using a variety of methods, we could potentially identify a more reliable set of features. We tested different models with all of them to see in which combination of model and feature selection method we would have the highest accuracy, using the F1 score. We had seven different feature selection methods that selected different features. (see the features selected by each model on Table 5).

Our first feature selection method was a mix between Chi-Square (for non-metric features) and Spearman correlation (for metric features). By looking at the Spearman correlation heatmap (Figure 8), we could see that the variables 'BMI' and 'Weight' were very highly correlated (correlation of 0.85). We have decided to drop one of them since they would be given very similar information to our models, which could lead to some problems. We have decided to drop 'Weight' since it had higher correlations with other variables and a smaller correlation with the target (Figure 5). We have conjugated this method with Chi-Square to have a final feature selection with numeric and categorical features. In addition to the Chi-Square and Spearman. We have also used Recursive Feature Elimination (RFE) (for Random Forest and Gradient Boosting). Although this feature selection method is used to specific models we thought it would be interesting to test it in other models. In the end, we could see that we achieved very good results with this feature selection method. Lasso, Decision Tree (Figure 9) and Ridge Regression (Figure 10) and ANOVA (see theoretical explanation in Annexes) were also used on our work.

We tested each model with default parameters and used k-fold cross-validation to evaluate each performance. In this situation, the decision of using k-fold over train/test split was mainly to mitigate the lack of accuracy that comes with splitting the data into a single train/test split. By using k-fold, we allow our model to be evaluated on different subsets of data, providing a more robust and reliable estimate of its performance and a better analysis when it comes to identifying any issues with overfitting. We decided to use k-fold cross-validation with k=10, which means that the data will be divided into 10 folds and the model will be trained and evaluated 10 times, with each fold serving as

the evaluation set once, and the final performance scores were then calculated as the average performance across all these 10 folds. Only the best methods for each model were used in the modeling phase. The models that proceed to the 'Modeling' phase, were chosen based on two criteria:

- Highest F1 scores
- Lowest likelihood overfitting (smaller gap between train and validation scores)

Based on the insights we gained from this phase, which included the best train and validation performance for all models with default parameters and the most effective feature selection methods for each model, we progressed to the next stage with a selection of candidate models and their corresponding feature selection methods that we believed had the potential to improve performance and lead us to the final model.

A train-test split was then applied, with a data partition of 85%/15% for the training and validation datasets respectively. We used the training set to fit our models, the validation set to get an unbiased evaluation of our models fit on the training dataset while tuning model hyperparameters and the test set to have an unbiased evaluation of a final model fit on the training dataset. This step allowed us to simulate how our model would perform on unseen data later [9]. The numerical variables in the train, validation, and test datasets were minmax scaled, and the categorical variables were one-hot encoded. We decided to make the split on the X_train without being normalized or scaled. We could then encode (with One Hot Encoder) and normalize (with Min Max Scaler) by fitting the model to the training set and transforming the training, validation, and test sets. By doing this, we could ensure that the model had been trained on the same transformation that would be applied to the validation and test sets, which would provide a more accurate evaluation of the model's performance.

## Modeling

Out of the 12 models used in feature selection (Logistic Regression, Support Vector Machine, Decision Tree (DT), Random Forest (RF), K-Nearest Neighbors Classifier (KNN), Neural Networks, Ada Boost Classifier, Extreme Gradient Boosting Classifier (XGBoost), Gradient Boosting Classifier (GBC), Ridge Classifier, Extra Trees Classifier (ETC) and Bagging Classifier), only the 7 best (Table 6) moved on to hyper parametrization.

For each model, we created a parameter space, using GridSearchCV with F1 as the scoring parameter, and obtained the best performing ones. We considered GridSearchCV a useful tool for this phase, since it would help us to fine-tune the performance of each model and hopefully, achieve better results. We opted for GridSearchCV for a range of different reasons that include its versatility (as we could use it with a wide range of models and parameter space) and consistency (as it ensures that the parameter space is conducted in a systematic and unbiased manner). With GridSearchCV

we could then ensure that all the possible combinations of our interest were being considered and evaluated.

We then ran the models and evaluated them using the “metrics” function that outputs a confusion matrix and a classification report containing the precision, recall, F1 and support scores for the train and validation data (Figure 11-19). Best performance was determined by how high both the train and validation scores were, and by how big the gap between them was, to avoid overfitting.

## Assessment

Most models produced great results, achieving 1 in all metrics for both the train and validation datasets. Only Decision Trees (Figure 11) and Extra Trees Classifier (Figure 16) had lower scores, both averaging 0.99 for all metrics in the validation data. With the models that received the highest scores, we performed stacking, creating the 2 stacking models one with KNN and GBC (st_model_2) and one with RF and BC (st_model_3), which both achieved a score of 1 in the train and validation data. (Figure 17 and Figure 18). With this, we can conclude that, in terms of score, the models seemed to have very good results with little (in some models) to almost no overfitting.

We then plotted a ROC curve (Figure 19), to visualize the overall performance of the classifier across all possible discrimination thresholds. However, since the areas under the curve were the same for our models (AUC = 1), it didn’t help our decision making, so for our final assessment method we performed a k-fold cross-validation with k=10 with the 7 models, since it provides a more trustworthy assessment of the models’ performances. A visualization of the results (Figure 20) was created, clearly showing that GBC and the st_model_2 were the best performing ones. The GBC model obtained 0.977 on the validation data, st_model_2 obtained a score of 0.982.

Upon selecting our final models, we trained them on the complete train dataset, to maximize their accuracy and robustness before submitting them to Kaggle. We felt that this step was particularly important given the relatively small size of the data. We decided to submit both the models, st_model 2 got a score of 0.989 and GBC got a score of 1.

We decided to select GBC as our final submission, since we believe that a better score on data in Kaggle was more important than on the validation data. The data on Kaggle is unseen data, which our model is built to predict, so we deemed an entirely correct prediction on 40% of the data to be very important.

## Conclusion

In this study, we addressed a classification issue by completing a full machine learning project with the goal of predicting individuals who are more likely to suffer from the "Smith Parasite" disease. After thorough data preprocessing and modeling, we achieved excellent scores on all assessment metrics, as well as in the Kaggle competition.

Even before hyperparameterization, our models performed exceptionally well, with some able to correctly predict almost all values in both the training and validation datasets. After selecting the best parameters, the predictions improved even further. This was surprising as such performances are not common. We attribute this to several factors. Firstly, the quality of the data was exceptional - it was balanced and had very few missing values, outliers, or inconsistencies. Secondly, during data exploration, we observed that certain metric variables such as "Physical_Health," "Mental_Health," and certain categorical variables related to fruit habits, checkups, exercise, and diabetes had strong correlation with the target. These features were frequently chosen by our feature selection methods, indicating that the models had access to highly relevant data for predicting the target variable.

Finally, there is a less likely hypothesis that data leakage (the use of data from outside the dataset during model creation) may have occurred, although we took steps to prevent this. Our score of 1 in the Kaggle competition, which is based on unseen data, reduces the probability of the overfitting hypothesis, although it does not entirely eliminate it as we currently only have access to 30% of the data, which means the score may change.

## References


[1] Zhao, Y., Li, L., Wang, H., Cai, H., Bissyande, T., Klein, J., & Grundy, J. (2021). On the Impact of Sample Duplication in Machine Learning based Android Malware Detection. ACM Transactions on Software Engineering and Methodology, 30(3), 1-38.

[2] Komorowski, M., Marshall, D.C., Salciccioli, J.D., & Crutain, Y. (2016). Exploratory Data Analysis. In Secondary Analysis of Electronic Health Records (pp. 257-277). Springer, Cham. https://doi.org/10.1007/978-3-319-43742-2_15

[3] Codecademy. (n.d.). EDA: Data Visualization. Retrieved on 10/10/2022 from https://www.codecademy.com/article/eda-data-visualization

[4] Xu, X., Xia, L., Zhang, Q., & others. (2020). The ability of different imputation methods for missing values in mental measurement questionnaires. BMC Medical Research Methodology, 20, 42. https://doi.org/10.1186/s12874-020-00932-0

[5] De Heredia, F., Gómez-Martínez, S., & Marcos, A. (2012). Obesity, inflammation and the immune system. Proceedings of the Nutrition Society, 71(2), 332-338. doi:10.1017/S0029665112000092

[6] Yadav, D. (2019). Categorical encoding using label encoding and one hot encoder. Towards Data Science. Retrieved on 10/10/2022 from https://towardsdatascience.com/categorical-encoding-using-label-encoding-and-one-hot-encoder-911ef77fb5bd

[7] Hemavatisabu. (2022). Data pre-processing wit sklearn using standard and minmax scaler. GeeksforGeeks. Retrieved on 6/11/2022 from https://www.geeksforgeeks.org/data-pre-processing-wit-sklearn-using-standard-and-minmax-scaler/

[8] Roepke, B. (2022, April 18). Feature Selection. dataknowsall. Retrieved on 15/12/2022 from https://www.dataknowsall.com/featureselection.html

[9] RP, S. (2018). Getting Started. Kaggle. Retrieved on 15/12/2022 from https://www.kaggle.com/discussions/getting-started/143685

[10] Alpaydin, E. (2010). An introduction to machine learning. Cambridge University Press.

[11] scikit-learn. (n.d.). Linear Model. Retrieved on 12/12/2022 from https://scikit-learn.org/stable/modules/linear_model.html

[12] Hastie, T., Tibshirani, R., & Friedman, J. (n.d.). Introduction to Boosting. In The Elements of Statistical Learning. Retrieved from https://web.stanford.edu/~hastie/ElemStatLearn/

[13] Aron, A., Coups, E. J., & Aron, A. (2016). Statistics for the behavioral and social sciences. Routledge.

[14] Qualtrics. (n.d.). Analysis of Variance (ANOVA). Retrieved on 18/11/2022 from https://www.qualtrics.com/uk/experience-management/research/anova/

[15] Alpaydin, E. (2010). An introduction to machine learning. Cambridge University Press. Retrieved on 18/11/2022 from https://scikit-learn.org/stable/modules/linear_model.html#ridge-regression

[16] Scikit-learn. (n.d.). RidgeClassifier. Retrieved on 6/11/2022 from https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeClassifier.html

[17] GeeksforGeeks. (2020). ML Extra Tree Classifier for Feature Selection. on 10/11/2022 from https://www.geeksforgeeks.org/ml-extra-tree-classifier-for-feature-selection/

[18] XGBoost. (n.d.). Parameters. Retrieved on 1/12/2022 from https://xgboost.readthedocs.io/en/latest/parameter.html

[19] Scikit-learn. (n.d.). Gradient Boosting Classifier. Retrieved on 6/11/2022 from https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeClassifier.html

[20] Scikit-learn. (n.d.). Random Forest Classifier. Retrieved on 6/11/2022 from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier

[21] Dietterich, T. G., & Efron, B. (1979). Evaluating the accuracy of diagnostic systems. Clinical Chemistry, 25(4), 657-665.

## Annexes

### *Theoretical explanation of techniques and models not covered in class*

- ***Feature Selection Techniques***
    - *L1 Regularization*

L1 Regularization: This is a method used to reduce the complexity of a model by adding a penalty term to the objective function being minimized. The penalty term is the sum of the absolute values of the model's weights, and it forces some of the weights to be exactly equal to zero, effectively removing those features from the model [10],[11].

- *RFE (Recursive Feature Elimination):*

This is a feature selection method that uses a model (such as a random forest or gradient boosting model) to select the most relevant features. It works by recursively eliminating the least important features one by one until the desired number of features is reached. In our work we use Recursive Feature Elimination for Gradient Boosting and Recursive Feature Elimination for Random Forest. Both RFE with gradient boosting and RFE with random forests are feature selection methods that use a model to select the most relevant features for a particular task. They work by recursively eliminating the least important features one by one until the desired number of features is reached, resulting in a model that is simpler and potentially more interpretable, while still maintaining good predictive performance. [12]

- *ANOVA (Analysis of Variance):*

This is a statistical method used to test whether the mean of a quantitative response variable is the same across different levels of a categorical predictor variable. It can be used for feature selection by selecting the features that have the highest F-score, which indicates that they have a significant effect on the response variable [13], [14].

- *Ridge Regression:*

This is a type of linear model that uses L2 regularization to reduce the complexity of the model. It works by adding a penalty term to the objective function being minimized, which shrinks the model's weights towards zero but does not force them to be exactly equal to zero. This helps to prevent overfitting and improve the generalization ability of the model [15].

- ***Models and parameters not discussed in class***
    - *Ridge Classifier:*

This is a linear model that can be used for classification. Like other linear models, the Ridge Classifier makes predictions based on linear combination of input features, using a set of weights

(coefficients) learned from the training data. It includes a regularization term that penalizes large coefficients and consequently helps to prevent overfitting and can improve the model's generalization to unseen data [16]

- *ExtraTrees Classifier:*

This Classifier is an ensemble algorithm that belongs to the family of decision tree algorithms, more specifically, an extension of the Random Forests algorithm. In ExtraTrees Classifier, each tree in the ensemble is trained using a random subset of the training data, and a random subset of features are used to make predictions at each split in the tree. This makes the model more robust and less prone to overfitting, as it can capture a wide range of interactions among the features and the target variable [17].

- *XGBoost Classifier*

XGBoost is a powerful algorithm used for supervised learning problems, where we use the training data (with multiple features) to predict a target variable. It works by building an ensemble of decision trees, where each tree is trained to correct the errors made by the previous tree in the ensemble. This is done by minimizing a loss function through gradient descent, which helps to improve the overall predictive accuracy of the model [18].

- Parameter *"colsample_bytree"* – determines the fraction of columns (features) to be randomly subsampled for each tree in the model.
- Parameter *"gamma"* – controls the minimum loss reduction required to make a split.
- Parameter *"min_child_weight"* – controls the minimum sum of weights of all observations required in a child.
- Parameter *"subsample"* – determines the fraction of observations to be randomly subsampled for each tree in the model

- *Gradient Boosting Classifier – parameters*

- Parameter "*loss*" – specifies the loss function that is used to evaluate the model's predictions. The loss function measures the difference between the predicted output and the true output. In other words, it quantifies how well the model is able to predict the target variable [19].

- *Random Forest Classifier - parameters*

- Parameter *"ccp_alpha"* – controls the complexity of the model. It is used to specify the amount of complexity penalty to be used when building the model. The complexity penalty is a measure of how much the model is being penalized for being more complex. In the case of a random forest classifier, the complexity of the model is determined by the number of decision trees in the forest and the depth of those trees [20].

- ***ROC curve***

A receiver operating characteristic (ROC) curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting the true positive rate (TPR) on the y-axis against the false positive rate (FPR) on the x-axis. The true positive rate is the proportion of actual positive cases that are correctly identified as such, while the false positive rate is the proportion of actual negative cases that are incorrectly identified as positive [21]

## *Tables*

*Table 1 - Original features and their descriptions*

| Feature | 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? |
| **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? |
| **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? |
| **Disease** | The dependent variable. If the patient has the disease (Disease = 1) or not (Disease = 0) |

*Table 3 - Descriptive Statistics of the non-metric features*

| | count | unique | top | freq | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **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 |
| **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 |
| **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 |
| **Disease** | 800.0 | NaN | NaN | NaN | 0.51375 | 0.500124 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 |

*Table 2 - Descriptive Statistics of the metric features*

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **Birth_Year** | 800.0 | 1966.04375 | 15.421872 | 1855.0 | 1961.00 | 1966.0 | 1974.0 | 1993.0 |
| **Height** | 800.0 | 167.80625 | 7.976888 | 151.0 | 162.00 | 167.0 | 173.0 | 180.0 |
| **Weight** | 800.0 | 67.82750 | 12.113470 | 40.0 | 58.00 | 68.0 | 77.0 | 97.0 |
| **High_Cholesterol** | 800.0 | 249.32250 | 51.566631 | 130.0 | 213.75 | 244.0 | 280.0 | 568.0 |
| **Blood_Pressure** | 800.0 | 131.05375 | 17.052693 | 94.0 | 120.00 | 130.0 | 140.0 | 200.0 |
| **Mental_Health** | 800.0 | 17.34500 | 5.385139 | 0.0 | 13.00 | 18.0 | 21.0 | 29.0 |
| **Physical_Health** | 800.0 | 4.55875 | 5.449189 | 0.0 | 0.00 | 3.0 | 7.0 | 30.0 |

*Table 4 - Created variables and their descriptions*

| Feature | **Description** |
|---|---|
| **Age** | The age of the patient |
| **BMI** | Body Mass Index: Weight (kg) / Height**2 (m) |

*Table 5 - Feature Selection (different models)*

| Features \ Feature selection | Spearman correlation + Qui-square | L1 Regularization | RFE (Random Forest + Gradient Boosting) | ANOVA | Lasso | Decision Tree | Ridge Regression |
|---|---|---|---|---|---|---|---|
| **Discard Region_East Midlands** | Discard | Keep | Discard | Discard | Keep | Discard | Discard |
| **Region_East of England** | Discard | Keep | Discard | Discard | Keep | Discard | Discard |
| **Region_London** | Discard | Keep | Discard | Discard | Keep | Discard | Discard |
| **Region_North East** | Discard | Keep | Discard | Discard | Keep | Discard | Discard |
| **Region_North West** | Discard | Keep | Discard | Discard | Keep | Discard | Discard |
| **Region_South East** | Discard | Keep | Discard | Discard | Keep | Discard | Discard |
| **Region_South West** | Discard | Discard | Discard | Discard | Discard | Discard | Discard |
| **Region_West Midlands** | Discard | Keep | Discard | Discard | Keep | Discard | Keep |
| **Region_Yorkshire and the Humber** | Discard | Keep | Discard | Discard | Keep | Discard | Discard |
| **Education_Elementary School (1st to 9th grade)** | Discard | Keep | Discard | Discard | Keep | Discard | Discard |
| **Education_High School Graduate** | Discard | Keep | Discard | Discard | Keep | Discard | Discard |
| **Education_High School Incomplete (10th to 11th grade)** | Discard | Keep | Discard | Discard | Keep | Discard | Discard |
| **Education_I never attended school / Other** | Discard | Discard | Discard | Discard | Keep | Discard | Keep |
| **Education_University Complete (3 or more years)** | Discard | Discard | Discard | Discard | Keep | Discard | Discard |
| **Education_University Incomplete (1 to 2 years)** | Discard | Discard | Discard | Discard | Keep | Discard | Keep |
| **Smoking_Habit_No** | Discard | Keep | Discard | Discard | Keep | Discard | Discard |
| **Smoking_Habit_Yes** | Discard | Keep | Discard | Discard | Discard | Keep | Discard |
| **Drinking_Habit_I consider myself a social drinker** | Keep | Keep | Discard | Discard | Keep | Discard | Keep |
| **Drinking_Habit_I do not consume any type of alcohol** | Keep | Discard | Discard | Discard | Discard | Discard | Discard |
| **Drinking_Habit_I usually consume alcohol every day** | Keep | Discard | Discard | Discard | Discard | Discard | Discard |
| **Exercise_No** | Keep | Discard | Discard | Keep | Keep | Keep | Keep |
| **Exercise_Yes** | Keep | Discard | Discard | Keep | Discard | Discard | Keep |
| **Fruit_Habit_1 to 2 pieces of fruit in average** | Keep | Keep | Discard | | Keep | Keep | Keep |
| **Fruit_Habit_3 to 4 pieces of fruit in average** | Keep | Keep | Discard | Keep | Discard | Keep | Keep |
| **Fruit_Habit_5 to 6 pieces of fruit in average** | Keep | Discard | Discard | | Keep | Keep | Keep |
| **Fruit_Habit_Less than 1. I do not consume fruits every day.** | Keep | Keep | Keep | Keep | Keep | Keep | Keep |
| **Fruit_Habit_More than six pieces of fruit** | Keep | Keep | Discard | Discard | Keep | Discard | Keep |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Water_Habit_Between one liter and two liters** | Remove | Discard | Discard | Discard | Keep | Discard | Discard |
| **Water_Habit_Less than half a liter** | Remove | Discard | Discard | Discard | Keep | Discard | Discard |
| **Water_Habit_More than half a liter but less than one liter** | Remove | Discard | Discard | Discard | Discard | Discard | Discard |
| **Checkup_Less than 3 years but more than 1 year** | Keep | Discard | Discard | Discard | Discard | Keep | Discard |
| **Checkup_Less than three months** | Keep | Keep | Discard | Discard | Keep | Discard | Keep |
| **Checkup_More than 3 years** | Keep | Keep | Keep | Keep | Keep | Keep | Keep |
| **Checkup_Not sure** | Keep | Keep | Discard | Keep | Keep | Discard | Keep |
| **'Diabetes_I do have diabetes** | Keep | Keep | Discard | Discard | Discard | Discard | Discard |
| **Diabetes_I don't have diabetes, but I have direct family members who have diabetes.** | Keep | Remove | Discard | Discard | Discard | Discard | Discard |
| **Diabetes_I have/had pregnancy diabetes or borderline diabetes** | Keep | Keep | Discard | Discard | Keep | Keep | Keep |
| **Diabetes_Neither I nor my immediate family have diabetes.** | Keep | Keep | Keep | Keep | Keep | Keep | Keep |
| **Height** | Keep | Keep | Discard | Discard | Keep | Keep | Keep |
| **Weight** | Discard | Discard | Discard | Discard | Discard | Keep | |
| **High_Cholesterol** | Keep | Keep | Keep | Discard | Keep | Keep | Keep |
| **Blood_Pressure** | Keep | Keep | Discard | Discard | Keep | Keep | Keep |
| **Mental_Health** | Keep | Keep | Keep | Keep | Keep | Keep | Keep |
| **Physical_Health** | Keep | Keep | Keep | Keep | Keep | Keep | Keep |
| **Age** | Keep | Keep | Keep | Discard | Keep | Keep | Keep |
| **BMI** | Keep | Keep | Keep | Keep | Keep | Keep | Keep |

*Table 6 - Scores of different models using the default parameters. The performance of the best models are coloured with green. We did Hyperparameterization Tunning in the best models*

| Features | | Spearman correlation + Qui-square | L1 Regularization | RFE (Random Forest + Gradient Boosting) | ANOVA | Lasso | Decision Tree | Ridge Regression |
|---|---|---|---|---|---|---|---|---|
| **Logistic Regression** | Train | 0.8649 | 0.8627 | 0.8534 | 0.8470 | 0.8637 | 0.8553 | 0.8607 |
| | Val | 0.8289 | 0.8314 | 0.8468 | 0.8391 | 0.8314 | 0.8416 | 0.8518 |
| **Support Vector Machine** | Train | 0.9167 | 0.9142 | 0.8786 | 0.8661 | 0.9121 | 0.8901 | 0.9063 |
| | Val | 0.8570 | 0.8531 | 0.8647 | 0.8544 | 0.8582 | 0.8634 | 0.8659 |
| **Decision Tree** | Train | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| | Val | 0.9425 | 0.9463 | 0.9539 | 0.9017 | 0.9386 | 0.9591 | 0.9438 |
| **Random Forest** | Train | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| | Val | 0.9668 | 0.9664 | 0.9783 | 0.9311 | 0.9548 | 0.9642 | 0.9732 |
| **Extra Trees Classifier** | Train | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| | Val | 0.9451 | 0.9579 | 0.9873 | 0.9502 | 0.9451 | 0.9859 | 0.9860 |
| **KNN (n_neighbors=5)** | Train | 0.9184 | 0.9093 | 0.9362 | 0.9170 | 0.8922 | 0.9395 | 0.9298 |
| | Val | 0.8558 | 0.8532 | 0.8353 | 0.8520 | 0.8430 | 0.8596 | 0.8775 |
| **KNN (n_neighbors=1)** | Train | 0.9184 | 0.9093 | 0.9362 | 0.9170 | 0.8922 | 0.9395 | 0.9298 |
| | Val | 0.8558 | 0.8532 | 0.8353 | 0.8519 | 0.8430 | 0.8596 | 0.8775 |
| **Neural Networks** | Train | 0.9690 | 0.9537 | 0.8708 | 0.8615 | 0.9486 | 0.9064 | 0.9281 |
| | Val | 0.8621 | 0.8597 | 0.8558 | 0.8442 | 0.8672 | 0.8787 | 0.8825 |
| **Ada Boost Classifier** | Train | 0.9218 | 0.9190 | 0.9105 | 0.8798 | 0.9195 | 0.9154 | 0.9177 |
| | Val | 0.8672 | 0.8685 | 0.8723 | 0.8518 | 0.8646 | 0.8659 | 0.8659 |
| **XGBoost Classifier** | Train | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| | Val | 0.9617 | 0.9745 | 0.9719 | 0.9132 | 0.9656 | 0.9681 | 0.9745 |
| **Gradient Boosting Classifier** | Train | 0.9888 | 0.9862 | 0.9845 | 0.9442 | 0.9870 | 0.9857 | 0.9855 |
| | Val | 0.9362 | 0.9336 | 0.9335 | 0.8838 | 0.9361 | 0.9336 | 0.9361 |
| **Ridge Classifier** | Train | 0.8679 | 0.8680 | 0.8532 | 0.8473 | 0.8682 | 0.8614 | 0.8636 |
| | Val | 0.8442 | 0.8430 | 0.8442 | 0.8429 | 0.8480 | 0.8519 | 0.8404 |
| **Bagging Classifier** | Train | 0.9987 | 0. 9987 | 0. 9987 | 0. 9987 | 0. 9987 | 0. 9987 | 0. 9987 |
| | Val | 0.9477 | 0. 9477 | 0. 9477 | 0. 9477 | 0. 9477 | 0. 9477 | 0. 9477 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Bagging Classifier (ETC)** | Train | 0.9987 | 0.9982 | 0.9987 | 0.9963 | 0.9982 | 0.9979 | 0.9976 |
| | Val | 0.9477 | 0.9452 | 0.9490 | 0.9094 | 0.9413 | 0.9490 | 0.9490 |

## *Figures*

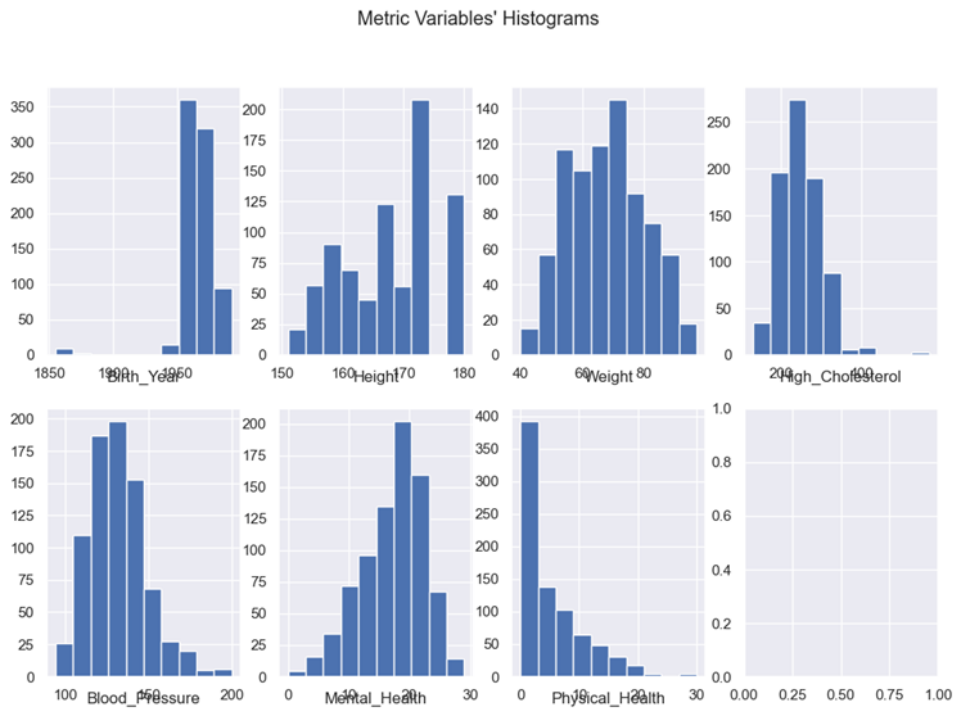
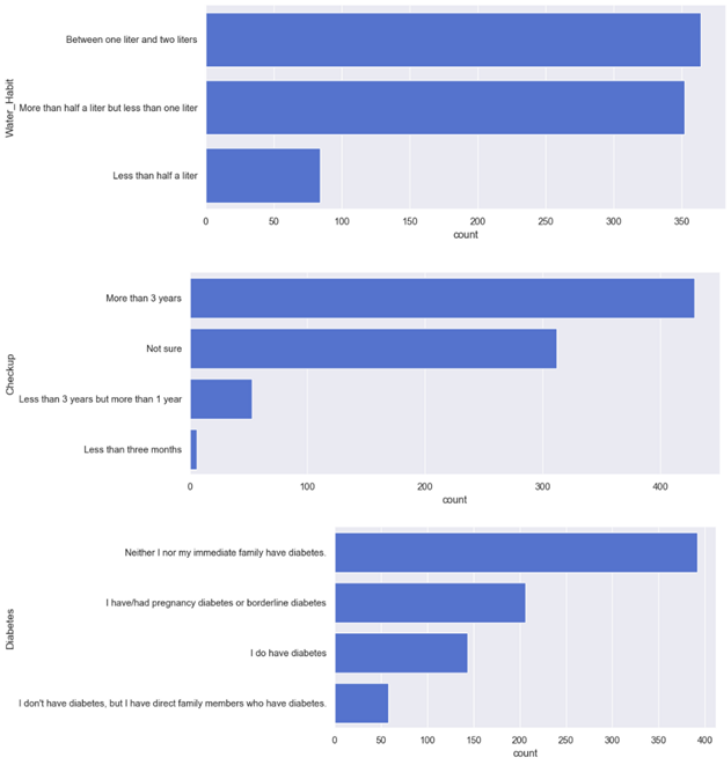



*Figure 1 - Metric Variables' Histograms before removing outliers*



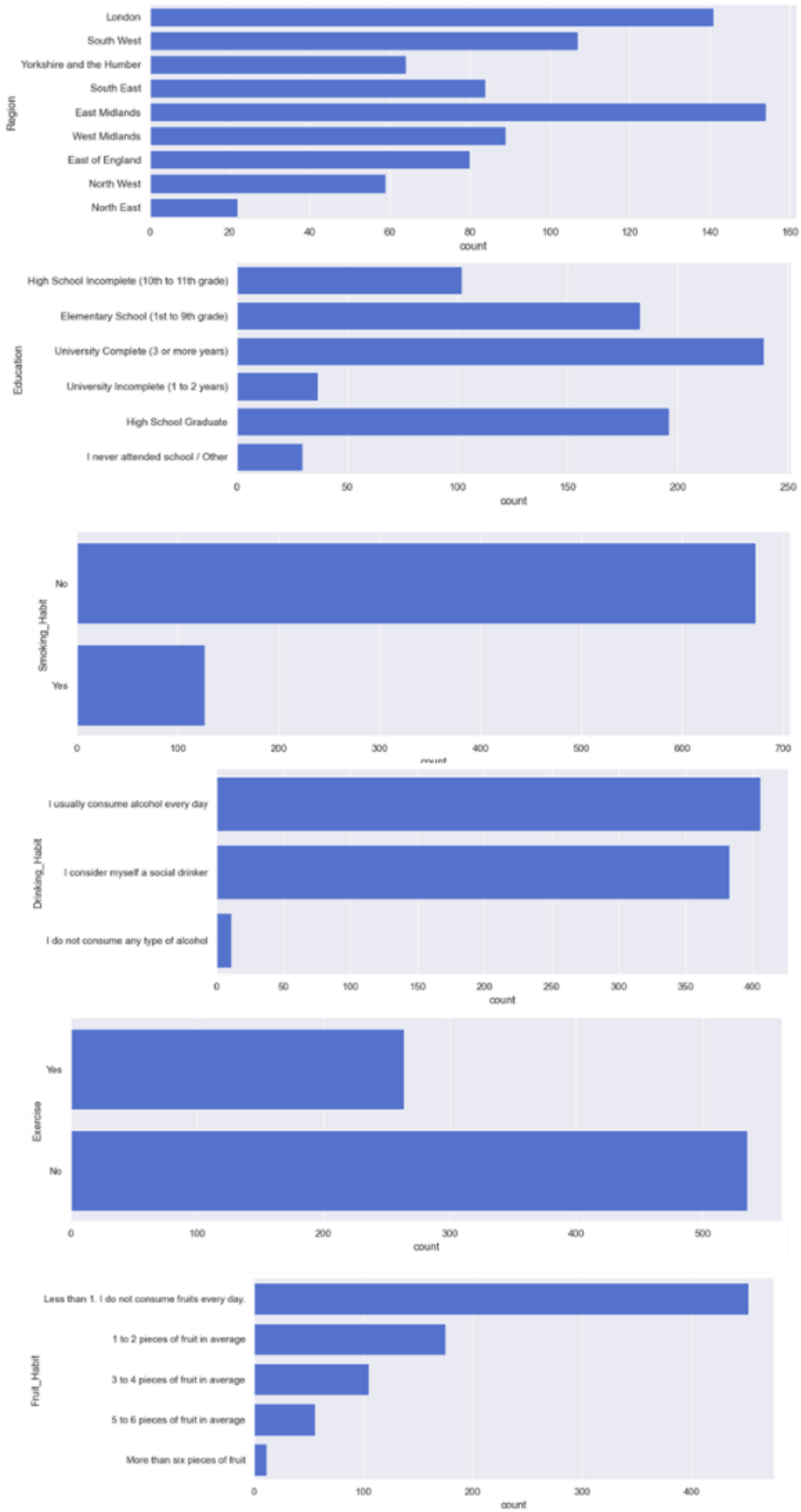



*Figure 2 - Non-metric Variables' Histograms before removing outliers*

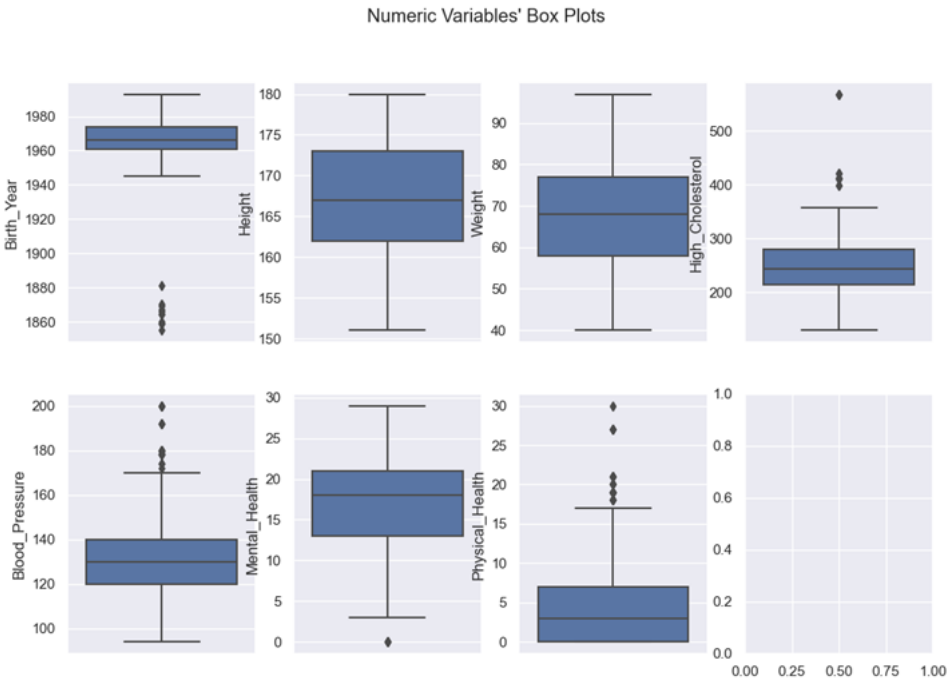



*Figure 3 - Metric Variables' Box Plots before removing outliers*

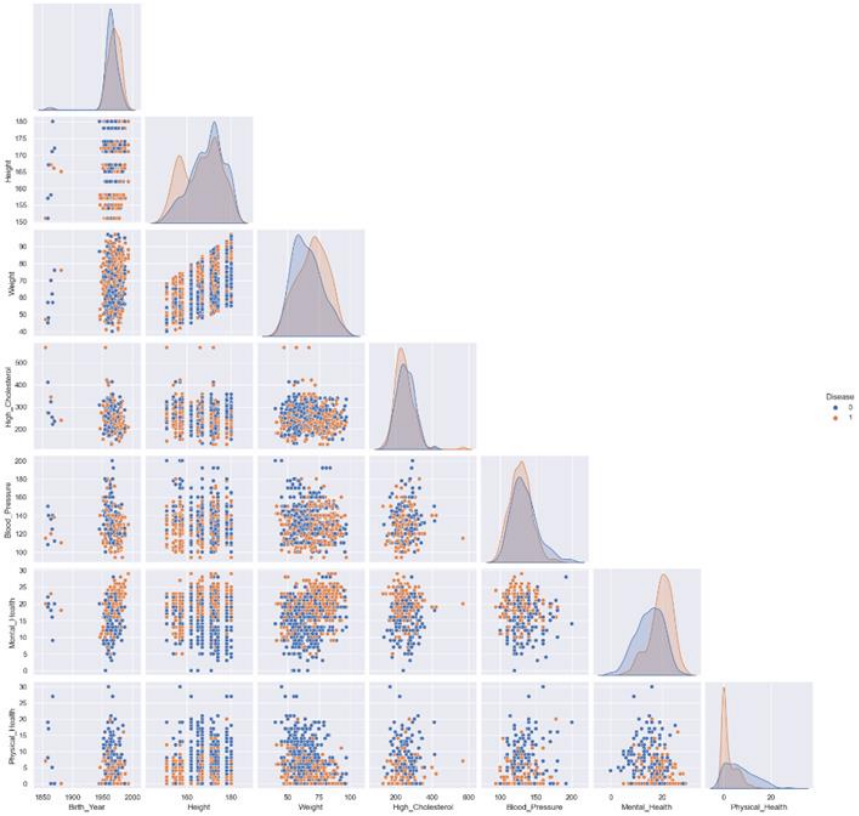


*Figure 4 - Metric Variable's Pairwise bivariate distributions*

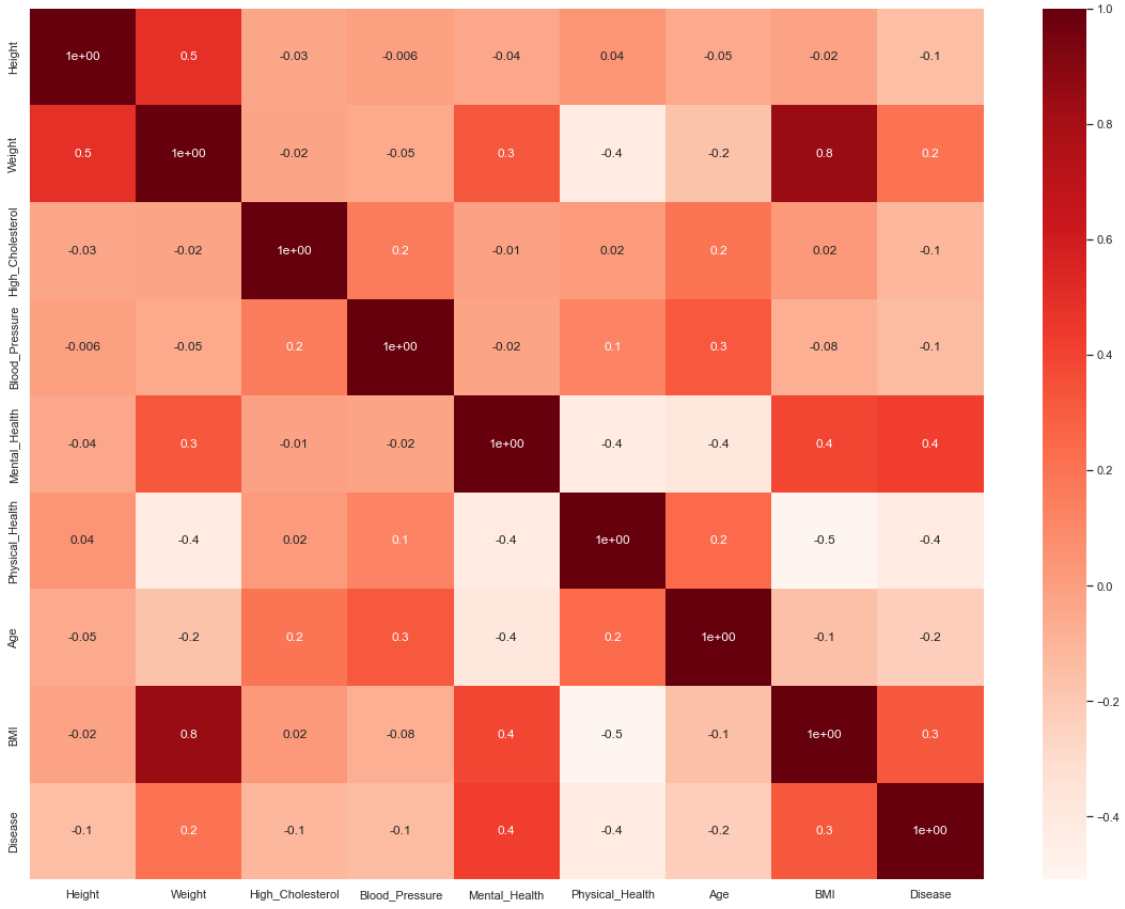



*Figure 5 - Metric Features' correlation with the target ("Disease")*

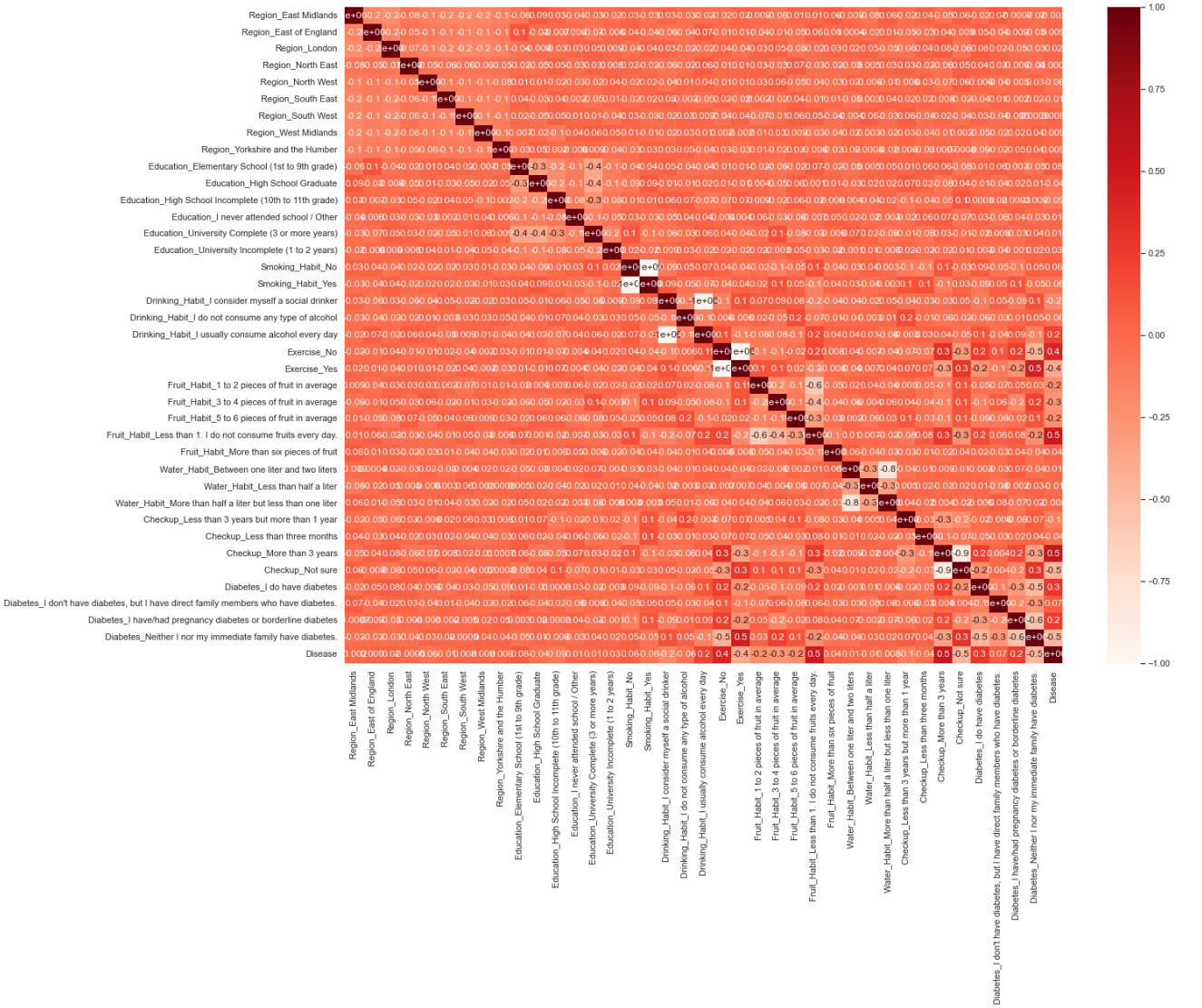



*Figure 6 - Non-Metric Features' correlation with the target ("Diisease")*

```
University Complete (3 or more years)          239
High School Graduate                           196
Elementary School (1st to 9th grade)           183
High School Incomplete (10th to 11th grade)    102
University Incomplete (1 to 2 years)            37
I never attended school / Other                 30
NaN                                             13
Name: Education, dtype: int64
```

*Figure 7 - Education Value_counts()*

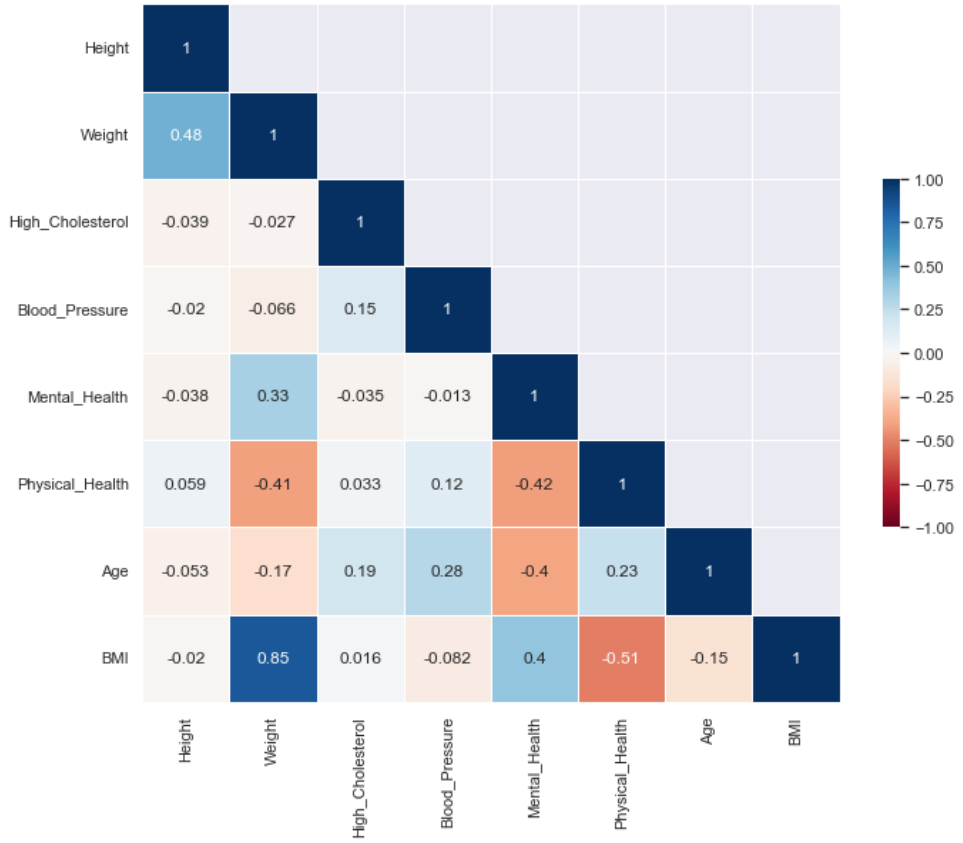



*Figure 8 - Spearman correlation between all metric features*

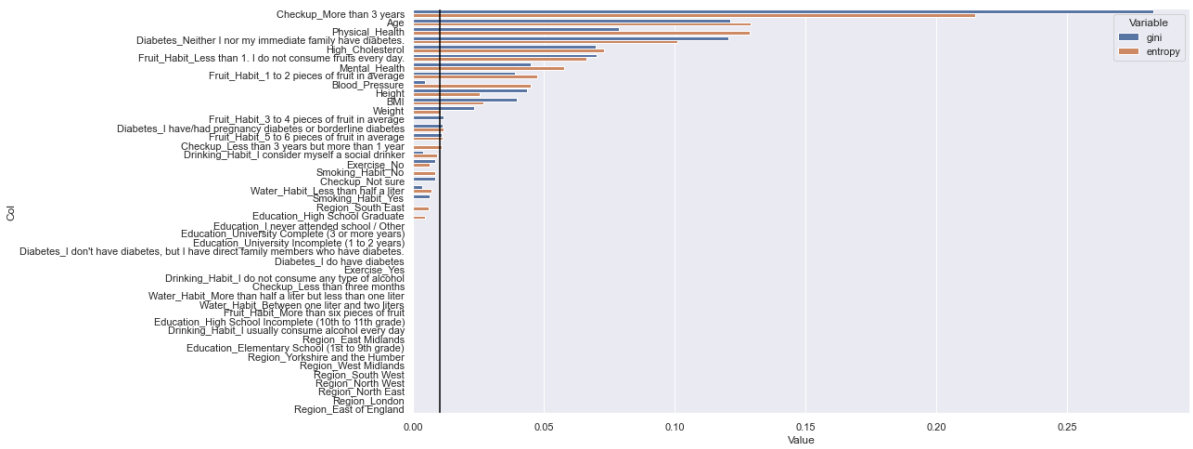



*Figure 9 - Feature Selection with Decision Tree*

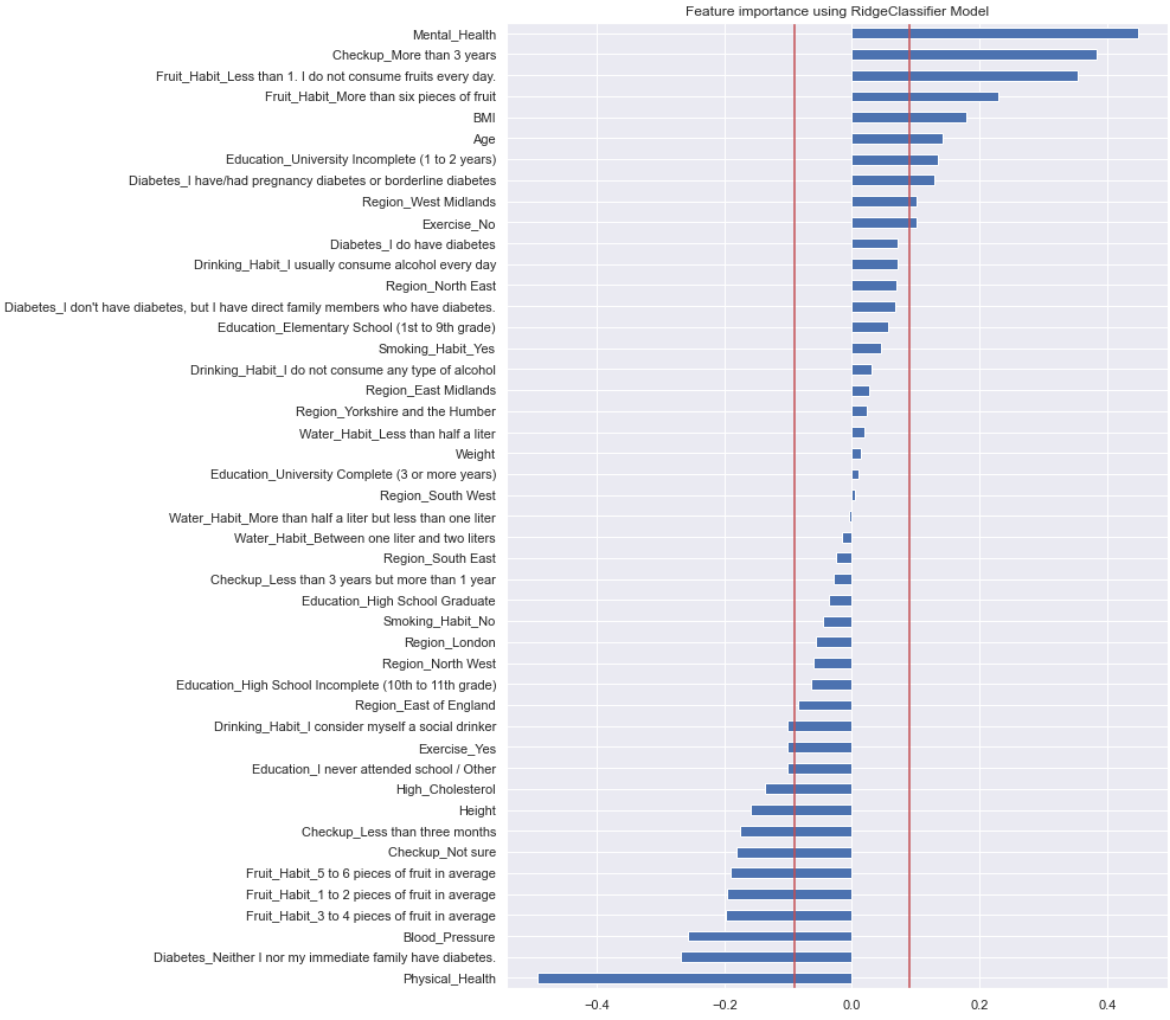



*Figure 10 - Feature Selection with Ridge Classifier*

```
                                                  TRAIN
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       312
           1       1.00      1.00      1.00       353

    accuracy                           1.00       665
   macro avg       1.00      1.00      1.00       665
weighted avg       1.00      1.00      1.00       665

[[312   0]
 [  0 353]]
```

```
                                                VALIDATION
              precision    recall  f1-score   support

           0       0.99      1.00      0.99        66
           1       1.00      0.98      0.99        52

    accuracy                           0.99       118
   macro avg       0.99      0.99      0.99       118
weighted avg       0.99      0.99      0.99       118

[[66  0]
 [ 1 51]]
```

*Figure 11 - Classification report and confusion matrix for Decision Tree*

```
                                                  TRAIN
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       312
           1       1.00      1.00      1.00       353

    accuracy                           1.00       665
   macro avg       1.00      1.00      1.00       665
weighted avg       1.00      1.00      1.00       665

[[312   0]
 [  0 353]]
```

```
                                                VALIDATION
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        66
           1       1.00      1.00      1.00        52

    accuracy                           1.00       118
   macro avg       1.00      1.00      1.00       118
weighted avg       1.00      1.00      1.00       118

[[66  0]
 [ 0 52]]
```

*Figure 12 - Classification report and confusion matrix for Random Forest*

```
_____
                                                TRAIN
------------------------------------------------------------------------------------------------------------
------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       312
           1       1.00      1.00      1.00       353

    accuracy                           1.00       665
   macro avg       1.00      1.00      1.00       665
weighted avg       1.00      1.00      1.00       665

[[312   0]
 [  0 353]]
_____
                                              VALIDATION
------------------------------------------------------------------------------------------------------------
------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        66
           1       1.00      1.00      1.00        52

    accuracy                           1.00       118
   macro avg       1.00      1.00      1.00       118
weighted avg       1.00      1.00      1.00       118

[[66  0]
 [ 0 52]]
```

*Figure 13 - Classification report and confusion matrix for KNN*

```
_____
                                                TRAIN
------------------------------------------------------------------------------------------------------------
------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       312
           1       1.00      1.00      1.00       353

    accuracy                           1.00       665
   macro avg       1.00      1.00      1.00       665
weighted avg       1.00      1.00      1.00       665

[[312   0]
 [  0 353]]
_____
                                              VALIDATION
------------------------------------------------------------------------------------------------------------
------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        66
           1       1.00      1.00      1.00        52

    accuracy                           1.00       118
   macro avg       1.00      1.00      1.00       118
weighted avg       1.00      1.00      1.00       118

[[66  0]
 [ 0 52]]
```

*Figure 14 - Classification report and confusion matrix for XGBoost*

```
_____
                                                TRAIN
-------------------------------------------------------------------------------------------------------
------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       312
           1       1.00      1.00      1.00       353

    accuracy                           1.00       665
   macro avg       1.00      1.00      1.00       665
weighted avg       1.00      1.00      1.00       665

[[312   0]
 [  0 353]]
_____
_____
                                             VALIDATION
-------------------------------------------------------------------------------------------------------
------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        66
           1       1.00      1.00      1.00        52

    accuracy                           1.00       118
   macro avg       1.00      1.00      1.00       118
weighted avg       1.00      1.00      1.00       118

[[66  0]
 [ 0 52]]
```

*Figure 15 - Classification report and confusion matrix for Gradient Boosting*

```
_____
_____
                                                TRAIN
-------------------------------------------------------------------------------------------------------
------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       312
           1       1.00      1.00      1.00       353

    accuracy                           1.00       665
   macro avg       1.00      1.00      1.00       665
weighted avg       1.00      1.00      1.00       665

[[312   0]
 [  0 353]]
_____
_____
                                             VALIDATION
-------------------------------------------------------------------------------------------------------
------
              precision    recall  f1-score   support

           0       1.00      0.98      0.99        66
           1       0.98      1.00      0.99        52

    accuracy                           0.99       118
   macro avg       0.99      0.99      0.99       118
weighted avg       0.99      0.99      0.99       118

[[65  1]
 [ 0 52]]
```

*Figure 16 - Classification report and confusion matrix for Extra Tree Classifier*

```
_____
                                                TRAIN
------------------------------------------------------------------------------------------------------------
------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       312
           1       1.00      1.00      1.00       353

    accuracy                           1.00       665
   macro avg       1.00      1.00      1.00       665
weighted avg       1.00      1.00      1.00       665

[[312   0]
 [  0 353]]
_____
_____
                                              VALIDATION
------------------------------------------------------------------------------------------------------------
------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        66
           1       1.00      1.00      1.00        52

    accuracy                           1.00       118
   macro avg       1.00      1.00      1.00       118
weighted avg       1.00      1.00      1.00       118

[[66  0]
 [ 0 52]]
```

*Figure 17 - Classification report and confusion matrix for Bagging Classifier*

```
_____
                                                TRAIN
------------------------------------------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       312
           1       1.00      1.00      1.00       353

    accuracy                           1.00       665
   macro avg       1.00      1.00      1.00       665
weighted avg       1.00      1.00      1.00       665

[[312   0]
 [  0 353]]
_____
                                              VALIDATION
------------------------------------------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        66
           1       1.00      1.00      1.00        52

    accuracy                           1.00       118
   macro avg       1.00      1.00      1.00       118
weighted avg       1.00      1.00      1.00       118

[[66  0]
 [ 0 52]]
```

*Figure 18 - Classification report and confusion matrix for the ST (Stacking with KNN and Gradient Boosting) Model*

```
                                                    TRAIN
-------------------------------------------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       312
           1       1.00      1.00      1.00       353

    accuracy                           1.00       665
   macro avg       1.00      1.00      1.00       665
weighted avg       1.00      1.00      1.00       665

[[312   0]
 [  0 353]]
-------------------------------------------------------------------------------------------------------------
                                                  VALIDATION
-------------------------------------------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        66
           1       1.00      1.00      1.00        52

    accuracy                           1.00       118
   macro avg       1.00      1.00      1.00       118
weighted avg       1.00      1.00      1.00       118

[[66  0]
 [ 0 52]]
```

*Figure 19 - Classification report and confusion matrix for ST2 (Stacking with Random Forest and Bagging) Model*

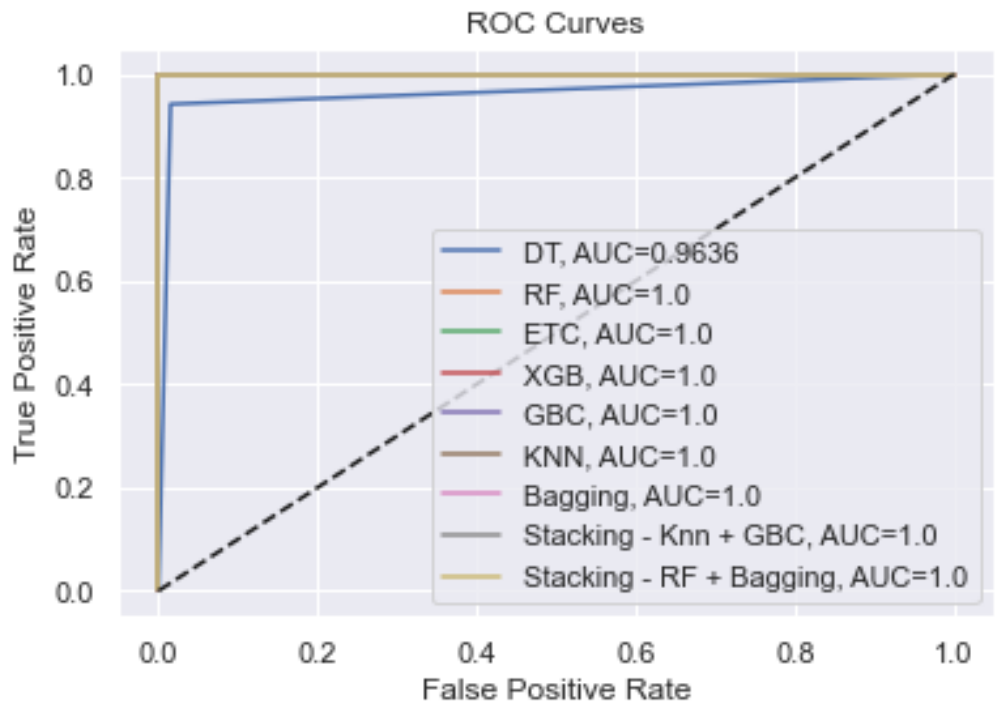



*Figure 20 - ROC Curves of different models*

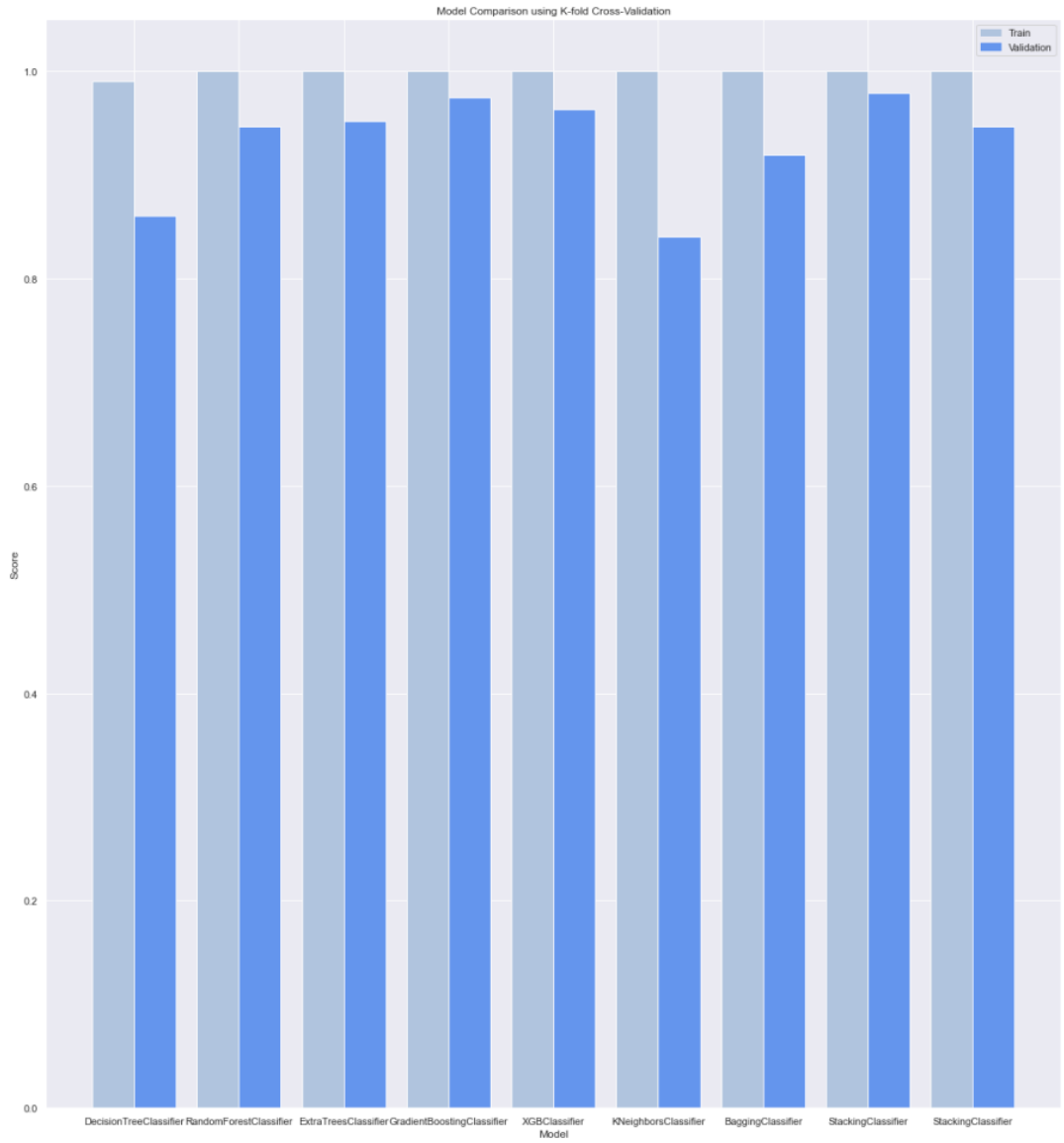



*Figure 21 - Model Comparison using K-fold Cross Validation*