THE SMITH PARASITE PROJECT

MACHINE LEARNING – GROUP 33

Ana Miguel Monteiro, 20221645

Ana Rita Viseu, 20220703

Miguel Cruz, 20221391

Rodrigo Brigham, 20221607

Sara Galguinho, 20220682

Abstract

This project aims to build a predictive model to accurately determine if a patient will suffer from a recently discovered disease, the Smith Parasite. In order to do this, we were granted access to 800 training observations and 225 test observations obtained from a set of patients.

We performed data exploration, using data visualization and descriptive statistics, and then preprocessed the data by handling inconsistencies, incoherences, missing values and outliers. Afterwards, we performed feature engineering along with feature selection using different methods. Then, we advanced to the modelling stage where we implemented various models, optimized their hyperparameters and found the best combination of features and scaling method for each. Lastly, we analyzed their respective results (training, validation and score in Kaggle).

By considering the context of this problem, the characteristics of the models and their performance in F1 score, we chose the decision tree as our final classifier. With this model we achieved a training score of 100%, a validation score of approximately 99% and a score in Kaggle of 100%.

Keywords: Machine Learning, Data Exploration, Data Preprocessing, Feature Engineering, Feature Selection, Decision Tree Classifier, F1 Score, Smith Parasite, Kaggle

Index

Index of Tables

Index of Figures

Introduction

In 2022, Dr. Smith discovered a new disease in England that had already impacted numerous individuals. Common symptoms included fever and fatigue, although some people were asymptomatic. Of greater concern were the post-disease conditions, which included impaired speech, confusion, chest pain, and shortness of breath. Despite the lack of knowledge on how the disease is transmitted and the absence of any discernible link between those infected, certain demographics appeared to be more susceptible to infection.

We have been provided with data regarding different patients, some of which have already been infected with the parasite. The provided data was split into 6 separate datasets: 3 of them belonging to a training set that was used to build our machine learning models, and the other 3 belonging to a test set that was used to see how well each model performs on unseen data. This data contained:

- Sociodemographic attributes – name, year of birth, living region and highest education level. In the training set, it was also known whether the patient had the disease or not.
- Health related attributes – height, weight, last known checkup consultation, history of diabetes on themselves or on their family, cholesterol level, blood pressure level, mental health state and physical health state.
- Habits related attributes – smoking habits, alcohol consumption habits, exercise habits, fruit consumption habits and water consumption habits.

The aim of this research is to answer the question “Who are the people more likely to suffer from the Smith Parasite?”.

By pinpointing our target variable as whether a patient has contracted the disease or not (1 if yes, 0 otherwise), we were able to generate hypotheses on what factors are more relevant to predict whether someone is more inclined to be infected. To further this cause, we tested various algorithms and strategies to create a predictive model that accurately and consistently predicts if an individual has contracted the disease, so that an appropriate counter strategy can be implemented to reduce the spread of the parasite.

Exploration

Our first step was to perform an initial exploration of the available information and try to identify data patterns, structural problems and take early conclusions. We merged the provided data into two datasets: a “train” dataset and a “test” dataset after confirming that the “PatientID” had the same values in the respective initial datasets.

Afterwards, we used descriptive statistics and data visualization tools (such as pandas profiling) to generate an overall report of the data and discover inconsistencies and potential insights such as:

- 800 observations in our training dataset and 225 in our test dataset. This could be considered a low number of observations but we also have to take into account that these come from patients and from a recently discovered disease;
- High Spearman Correlation values between “Weight” and “Height” on the training dataset as well as on the test dataset;
- High Cramer’s V correlation values between “Exercise” and “Diabetes” on both datasets;
- Looking at the correlations with the target variable, the least important predictors seem to be “Region” and “Water_Habit”, while the most are “Mental_Health”, “Checkup”, “Fruit_Habit” and “Diabetes”;
- In general, we seem to have predictors that give us an overview of the patients (17 in total, not counting “PatientID”) and what we would consider to be the most important information. Therefore, we expect to have a strong performance in our final model;
- A balanced training dataset with approximately 51% of 1’s and 49% of 0’s.
- 13 missing values in the variable “Education” on the training dataset;
- Extremely high values in the variable “High_Cholesterol” on the training dataset;
- Extremely low values in the variable “Birth_Year” (1800’s) on the training dataset;
- An extremely high percentage of individuals that did a checkup more than 3 years ago or that didn’t know when they last did one on the training (92.6%) and test (94.7%) datasets;
- 5 instances where the variable “Region” had a value of “LONDON” instead of “London” on the training dataset;
- 2 instances of a patient named “Mr. Gary Miller” on the training dataset;
- 2 instances of a patient named “Mr. Michael Williams” on the test dataset;
- The feature “Name” gives us the gender of the patient (considering the Mr. and Mrs.).

These potential problems will be solved in the data preprocessing stage. We then confirmed the datatypes were appropriate for our purposes and created a table with the most common descriptive statistics for all numeric variables of both datasets to systematize the variable information (annex, Tables 1 and 2).

Preprocessing

Regarding the preprocessing stage, we corrected some of the incoherences/ inconsistencies identified above (or decided that they were not incoherences but instead a normal part of the data), encoded categorical data, renamed some columns, imputed missing values and removed outliers. We also made sure to create numerous copies of the different versions of the training dataset throughout this stage in case we decided to change something in our preprocessing strategy later.

Treating Inconsistencies/ Incoherences

We started by correcting the value “LONDON” by replacing it with “London” in the variable “Region”. We then modified the birth years from the 1800’s by making the years that started with “18” start with “19” instead, assuming it was a mistake in data input. This only occurred 12 times, in none of the changes did the birth year become unreasonable (no patient with more than 100 years) and our assumption, although quite strong, should not significantly alter the overall variance of the data.

Preparing Data

In this phase of preprocessing, it is important to note that every change was applied both in the training dataset and in the test dataset.

We began by encoding the categorical variables. We wanted to keep “Education”, “Drinking_Habit”, “Fruit_Habit” and “Water_Habit” as ordinal variables. However, by using ordinal encoder, we would lose the possibility to define the specific order of the observed values, so we decided to do it manually.

In the categorical variables “Region”, “Checkup” and “Diabetes”, we decided that it would be too strong of an assumption or simply would not make sense to define an order, so we used OneHotEncoder to transform them into dummy variables (annex, Table 3). We also used OneHotEncoder on the variables “Exercise” and “Smoking_Habit” because they can be naturally seen as binary variables (annex, Table 3).

After this, we dropped the variable “Name” because it didn’t add any relevant information to the models we would be training, although we will use it later during the feature engineering stage.

We then renamed some of the features to make them shorter and clearer.

Finally, we defined the following features as metric: “Height”, “Weight”, “High_Cholesterol”, “Blood_Pressure”, “Mental_Health”, “Physical_Health” and “Birth_Year”. All other features were listed as non-metric (annex, Table 4).

Filling in Missing Values

There were no missing values in the test dataset that needed to be treated.

In the training dataset we had 13 missing values in the variable “Education”. We decided to try some different options for the imputation such as Mode, K-Nearest Neighbor with uniform weights and weights based on distance and Iterative Imputer using Logistic Regression and Random Forest.

I. Mode

Using the mode, all missing values in “Education” were replaced by “5”, which corresponds to “University Complete (3 or more years)”. Although simple and easy to use, we found it to not be suitable considering that the categories for “Education” had a similar number of observations.

II. KNN with weights=’uniform'

Since it is a distance-based method and each variable has its own unique scale we must normalize the data, which we did by applying the MinMaxScaler. We applied this method using the Euclidean distance and 5 neighbours as parameters and then we reverted scale to get the unscaled original values back. We only used the metric features. We were able to apply this method because “Education” was encoded as an ordinal variable and we rounded the results of the KNN.

III. KNN with weights='distance'

The difference in the weights makes it an interesting option since it helps to reduce the influence of outliers and focuses the prediction on the more “representative” data points.

IV. Iterative Imputer using Logistic Regression

We had difficulties in the convergence of the logistic regression, so we filtered the features using the correlation to “Education”.

V. Iterative Imputer using Random Forest

We started by removing the features with a correlation to “Education” lower than 1% (since they are not relevant). After that, we applied the Iterative Imputer using Random Forest with the remaining features and in the end added the removed ones back to the dataset.

We decided to use the Random Forest Classifier as our final choice because it is a multivariate imputer but also takes into account the information given by the mode. As well as that, it did not have the problems in convergence that the Logistic Regression had.

As a final step, we checked the absolute frequencies of the variable “Education” after data imputation (annex, Figure 1).

Outlier Detection and Removal

When preparing data, detecting and processing outliers is an important step to consider. To ensure the data is treated in the most beneficial manner, a particular treatment must be adopted. We tried six different ways to detect outliers and then chose the most effective one and with as little loss of relevant information as possible.

I. Manual Filtering

We defined filters based on visualizations of our data. We kept 97.62% of the original data.

II. IQR (Interquartile Range) Method

In this method we defined an upper and lower bound using the inter quartile range (IQR) multiplied by 2 (we chose the value 2 based on visualizations of our data and the percentage of data that was kept). We then excluded all observations outside of that boundary and kept 97.88% of the original data.

III. Manual & IQR

Here we combined the two previous methods and kept 97.88% of the data.

IV. Z-Score

Z-score standardizes the data and removes all values that are 3.5 standard deviations apart from the mean (0). We chose 3.5 for a similar reason to the value in the IQR method. We kept 98.38% of the data.

V. DBSCAN

Using DBSCAN we excluded all points that are not considered core or border. Firstly, we had to determine the optimum epsilon to use as a parameter (0.35) using the Elbow Method and considering both visualizations of our data and the percentage of data that was kept. We kept 96.88% of the data.

VI. Isolation Forest

We defined the contamination as 0.03 as we intended to exclude 3% of the data. We chose 3% considering, again, visualizations of our data and trying to transform as little as possible of our dataset. Naturally, we kept 97% of the original data.

We decided to proceed with the IQR method as our final choice after considering visualizations, the amount of data removed and testing we have done in the modelling stage.

Before proceeding, we made some visualizations of our data to see how the outlier removal impacted the distribution of the metric variables (annex, Figures 2, 3, 4 and 5).

Feature Engineering

After completing data exploration and preprocessing, we performed feature engineering.

Using the name prefixes “Mrs.” and “Mr.”, we created a new variable “Gender”. A categorical variable where the “Mrs.” is associated with 1, meaning it is a female, and “Mr” is associated with 0, meaning it is a male. This way we can identify a patient’s gender.

We also created the variable “Age”, which tells us the current age of the person by subtracting the year from the current date to the “Birth_Year”.

By combining height and weight we created the variable “BMI” (Body Mass Index). BMI is an international measure used to calculate whether a person is at an ideal weight.

With these new features (annex, Table 5) we have more information to achieve our project’s goal and we are able to improve our machine learning model, leading to better performance.

Feature Selection

In order to not have redundant variables (that would give the same type of information) we chose to eliminate the feature “Birth_Year”. Once we already have “Age”, it doesn’t make sense to have another feature that serves precisely the same purpose. We decided to keep “Age” because “Birth_Year” would maintain the same values, despite changing meaning as the year changes.

Afterwards, we proceeded to apply feature selection methods to reduce the input space in our model, keeping only relevant data and getting rid of any noise.

Before applying the feature selection methods, we defined the independent variables as “x” and the dependent variable (target) as “y”. As we had few observations (only 783), we decided not to split the data set into training and validation data. Instead, we used Stratified k-fold Cross Validation, with k=10 as recommended in the theoretical materials for this class.

I. Filter Methods

After scaling the data using MinMaxScaler, we used Spearman Correlation (annex, Figure 6), ANOVA, Kendall’s Correlation and the variance of the numerical variables. For the categorical variables, we reversed the OneHotEncoding so that it was possible to check the value counts and use the Chi-Squared method. These methods helped us measure the relevance and the redundancy of the variables.

II. Wrapper Methods

Using Wrapper Methods, we measured the usefulness of a subset of features by training a model on it. First, we had to apply OneHotEncoding for categorical features and scale the data. After that, we applied Recursive Feature Elimination with Cross Validation (RFECV) using Logistic Regression as an estimator. We also applied Recursive Feature Elimination with Cross Validation (RFECV) using Random Forest as an estimator.

III. Embedded Methods

From the existing Embedded Methods, we used Lasso Regression (annex, Figure 7) and Ridge Regression (annex, Figure 8), both with cross validation to measure the importance of each feature. Finally, we also computed the Gini Importance and Entropy Importance based on decision trees (annex, Figure 9).

After applying the various feature selection methods and analysing the results, we finished our conclusions about which features we were going to use in our model training, which features we wouldn’t use and which ones we would test our models with and without (annex, Tables 6 and 7). We then tested different combinations of features during the modelling phase.

Modelling

In order to find the best solution for our problem, we used the following models: Logistic Regression, Logistic Regression using Polynomial Features, Gaussian Naïve Bayes, Categorical Naïve Bayes, KNN Classifier, MLP Classifier, Gradient Boosting, AdaBoost Classifier, Ridge Classifier, Passive Aggressive Classifier, Decision Tree Classifier, Random Forest Classifier, Support Vector Machine, Bagging Classifier, Stacking Classifier and Voting Classifier. For the Bagging Classifier, we tried three different estimators: Decision Tree Classifier, KNN Classifier and Logistic Regression. For the Stacking Classifier, we tried three different combinations: one with Support Vector Machine, Gradient Boosting and AdaBoost Classifier, another with Support Vector Machine and KNN Classifier and finally one with Support Vector Machine, KNN Classifier and Decision Tree Classifier. For the Voting Classifier, we used two different combinations: Support Vector Machine and Decision Tree Classifier as estimators and KNN Classifier and Decision Tree Classifier as estimators. In these ensemble methods, we tried to choose models for them with strong performance and with the parameters we had found before. For instance, Decision Tree Classifier had very good scores, therefore we used it along with the hyperparameters we had optimized for it.

We followed a similar workflow when optimizing the results. Firstly, we used a function that gave us an overview of the best parameters to use in Grid Search. Secondly, we applied Grid Search and decided on the best combination of parameters based on the training score, the validation score and the difference between those two (overfitting or underfitting). Thirdly, using the chosen parameters, we compared the different combinations of features and different scaling methods (when the model required the data to be scaled) to determine our best solution. Finally, we tested the model in Kaggle to check if our performance was very distinct from the one obtained in the validation score. For Logistic Regression using Polynomial Features, Gaussian Naïve Bayes, Categorical Naïve Bayes, Ridge Classifier, Passive Aggressive Classifier, Bagging Classifier using Logistic Regression and Bagging Classifier using Decision Tree we skipped certain steps because we had already decided that these would not be selected as our final model. It is also important to mention that, similarly to before, we always used Stratified k-fold Cross Validation with k=10.

Assessment

By analysing the models' train, validation and Kaggle scores (annex, Table 8 and Figure 10), we can see that the models that had the best performance in all scores were (in no particular order): Voting Classifier (the two combinations tried), Stacking Classifier (also the three combinations tried), Gradient Boosting, Support Vector Machine, AdaBoost Classifier, Decision Tree Classifier, Random Forest Classifier, KNN Classifier, Bagging Classifier with KNN as estimator and MLP Classifier. As expected, we had very strong performances in a lot of models, nearing the 100% both in the training and validation scores.

Taking this into consideration, we decided to use a Decision Tree with the following parameters: ccp_alpha=0.001, criterion='entropy', max_depth=None, max_features=15 and splitter='random'.

The best combination of features for this model (annex, Table 9) was: "High_Cholesterol", "Blood_Pressure", "Mental_Health", "Physical_Health", "Age", "BMI", "Drinking_Habit", "Fruit_Habit", "Exercise", "Checkup_LessThan3Months", "Checkup_MoreThan3Years", "Checkup_NotSure", "Diabetes_A", "Diabetes_B", "Diabetes_C" and "Gender".

A Decision Tree is a simple model from which we can extract many insights regarding the conditions of the transmission of the disease and what leads a patient to suffer or not from it. Moreover, it is a technique that can be much more easily explained than other models, which is a significant attribute in the medical context. Based on the model we selected, we observed that the most relevant features for our Decision Tree Classifier are "Checkup_MoreThan3Years", "Fruit_Habit", "Diabetes_A" and "Physical_Health", whereas the least relevant features are "Checkup_LessThan3Months", "Checkup_NotSure" and "Diabetes_C" (annex, Figure 11).

Conclusion

To begin our project, we analysed the provided patient data to identify any wrong data types, missing values, inconsistencies and outliers. We then applied various techniques to rectify the issues we found. Subsequently, we examined the impact of each variable on the target and reduced the input variables by utilizing multiple feature selection methods. During the model selection stage, we tested a variety of models from both class and our own research to attain the highest F1 score possible for the train and validation sets. Based on these scores and the specificities of each algorithm, we chose the Decision Tree as our final model.

Considering the predictive model we created, we assessed that the most relevant factors that need to be monitored in order to efficiently predict if a patient contracts the disease or not are whether they did a check-up in the last 3 years, their fruit consumption habits as well as if neither the patient nor his immediate family relatives have diabetes. Monitoring these factors will help prevent further infections from the Smith Parasite and aid in more accurate diagnoses.

Finally, using this final model, we predicted from our test dataset that 116 individuals would suffer from the disease and 109 would not.

References

Bhuvaneswari Gopalan (November 4, 2020), *MICE algorithm to Impute missing values in a dataset*. Accessed November 9, 2022, <https://www.numpyninja.com/post/mice-algorithm-to-impute-missing-values-in-a-dataset>.

Idil Ismiguzel (May 12, 2022), *Imputing Missing Data with Simple and Advanced Techniques*. Accessed November 11, 2022, <https://towardsdatascience.com/imputing-missing-data-with-simple-and-advanced-techniques-f5c7b157fb87>.

Dhiraj K (2019), *Anomaly Detection Using Isolation Forest in Python*. Accessed November 20 2022, <https://blog.paperspace.com/anomaly-detection-isolation-forest/>.

Cory Maklin (June 30, 2019), *DBSCAN Python Example: The Optimal Value For Epsilon (EPS)*. Accessed November 25, 2022, <https://towardsdatascience.com/machine-learning-clustering-dbscan-determine-the-optimal-value-for-epsilon-eps-python-example-3100091cfbc>.

Unknown, *Anomaly Detection Example with DBSCAN in Python*. Accessed November 25, 2022, <https://www.datatechnotes.com/2020/04/anomaly-detection-with-dbscan-in-python.html>.

Jason Brownlee (May 29, 2020), *How to Use Polynomial Feature Transforms for Machine Learning*. Accessed December 12, 2022, <https://machinelearningmastery.com/polynomial-features-transforms-for-machine-learning/>.

Avinash Navlani (November 2018), *Understand the ensemble approach, working of the AdaBoost algorithm and learn AdaBoost model building in Python*. Accessed December 13, 2022, <https://www.datacamp.com/tutorial/adaboost-classifier-python>

Mate Pocs (May 16, 2021), *Hyperparameter tuning in Lasso and Ridge Regressions*. Accessed December 14, 2022, <https://towardsdatascience.com/hyperparameter-tuning-in-lasso-and-ridge-regressions-70a4b158ae6d>

Alokesh985 (July 17, 2020), *Passive Aggressive Classifiers.* Accessed December 23, 2022*,* <*https*://www.geeksforgeeks.org/passive-aggressive-classifiers/>

Unknown (July 30, 2020), *Classification Example with Ridge Classifier in Python*. Accessed December 23, 2022 <https://www.datatechnotes.com/2020/07/classification-example-with-ridge-classifier-in-python.html>

Annexes

Methods Not Taught in Class:

- Iterative Imputer with Logistic Regression: This is a technique used for multivariate data imputation where the missing values in a dataset are replaced considering the values generated from a logistic regression model and an initial strategy (for instance, mean or mode). The logistic regression model is trained on the existing values in the dataset and then used to predict the missing values.
- Iterative Imputer with Random Forest Classifier: This is a technique used for multivariate data imputation where the missing values in a dataset are replaced considering the values generated from a random forest classifier and an initial strategy (for instance, mean or mode). The random forest classifier is trained on the existing values in the dataset and then used to predict the missing values.
- Isolation Forest: This is an unsupervised learning algorithm used for anomaly detection. It works by randomly selecting a feature and then splitting the data based on that feature. It then calculates the isolation score of each data point, which determines if the data point is an outlier or not.
- ANOVA: ANOVA stands for Analysis of Variance and is a statistical technique used to test whether there is a significant difference between two or more means.
- Kendall's Correlation: This is a measure of the strength of ordinal associations between two variables. It determines the degree to which two variables are related and is often used in statistical data analysis.
- Ridge Classifier: This is a type of machine learning algorithm used for classification. It converts the target variable into –1 or 1 and then applies the Ridge Regression model. If the regression's predicted value is positive then we predict 1. If it is negative, we predict –1.
- Passive Aggressive Classifier: This is an algorithm used for classification that is based on the passive aggressive online learning algorithm. The name comes from: if the prediction is correct, we do not make any changes (passive), but if the prediction is wrong, the model changes (aggressive).
- Voting Classifier: The voting classifier works by training an ensemble of methods and predicting an output based on either soft or hard voting. In hard voting, the output is the class that had the highest probability of being predicted by each of the classifiers (other models). In soft voting, the output class is the prediction based on the average of probability given to that class. In our project, we used soft voting because it led us to better results.

Table 1: Descriptive Statistics of the Training Dataset's Numeric Variables

	count	mean	std	min	25%	50%	75%	max
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
Birth_Year	800.0	1966.04375	15.421872	1855.0	1961.00	1966.0	1974.0	1993.0
Disease	800.0	0.51375	0.500124	0.0	0.00	1.0	1.0	1.0

Table 2: Descriptive Statistics of the Test Dataset's Numeric Variables

	count	mean	std	min	25%	50%	75%	max
Height	225.0	167.422222	8.014743	151.0	162.0	167.0	173.0	180.0
Weight	225.0	67.800000	12.758750	42.0	57.0	68.0	77.0	97.0
High_Cholesterol	225.0	252.408889	51.727410	135.0	217.0	244.0	278.0	421.0
Blood_Pressure	225.0	133.595556	18.983098	94.0	120.0	130.0	144.0	200.0
Mental_Health	225.0	17.546667	4.902514	3.0	15.0	18.0	22.0	27.0
Physical_Health	225.0	5.377778	6.061032	0.0	0.0	4.0	9.0	30.0
Birth_Year	225.0	1967.644444	9.438607	1945.0	1961.0	1967.0	1975.0	1988.0

Table 3: Binary variables created from original variables of the dataset

Variable	Description
Region_EastOfEngland	Binary, 1 if the patient lives in the East of England
Region_London	Binary, 1 if the patient lives in London
Region_NorthEast	Binary, 1 if the patient lives in the North East of England
Region_NorthWest	Binary, 1 if the patient lives in the North West of England
Region_SouthEast	Binary, 1 if the patient lives in the South East of England
Region_SouthWest	Binary, 1 if the patient lives in the South West of England
Region_WestMidlands	Binary, 1 if the patient lives in the West Midlands
Region_YorkshireAndTheHumber	Binary, 1 if the patient lives in Yorkshire and the Humber
Checkup_LessThan3Months	Binary, 1 if the patient's last checkup was less than 3 months ago
Checkup_MoreThan3Years	Binary, 1 if the patient's last checkup was more than 3 months ago
Checkup_NotSure	Binary, 1 if the patient isn't sure of when his last checkup was
Diabetes_A	Binary, 1 if neither the patient nor their immediate family have diabetes
Diabetes_B	Binary, 1 if the patient doesn't have diabetes but they have direct family members who do
Diabetes_C	Binary, 1 if the patient has/ had pregnancy diabetes or borderline diabetes
Smoking_Habit	Binary, 1 if the patient smokes more than 10 cigars daily
Exercise	Binary, 1 if the patient exercises (more than 30 minutes) 3 times per week or more

Table 4: Metric and Non-Metric Features

Metric Features	Non-Metric Features
Height Weight High_Cholesterol Blood_Pressure Mental_Health Physical_Health Birth_Year	Drinking_Habit; Fruit_Habit; Water_Habit; Education; Smoking_Habit; Exercise; Region_EastOfEngland; Region_London; Region_NorthEast; Region_NorthWest; Region_SouthEast; Region_SouthWest; Region_WestMidlands; Region_YorkshireAndTheHumber; Checkup_LessThan3Months; Checkup_MoreThan3Years; Checkup_NotSure; Diabetes_A; Diabetes_B; Diabetes_C

Table 5: New Variables Created

Variable	Description
Gender	Binary, 1 if the patient is female and 0 if the patient is male
Age	Current age of the patient
BMI	Body Mass Index of the patient

Table 6: Numerical Variables' Feature Importance

Predictor	Spearman	Variance	ANOVA	RFE (Logistic Regression)	RFE (Random Forest)	Lasso	Ridge	Kendall's	Gini	Entropy	What to do? (One possible way to "solve")
Height	Keep	Keep	Keep	Keep	Discard	Keep	Keep	Keep	Keep?	Keep?	Try with and without
Weight	Discard	Keep	Keep	Discard	Discard	Discard	Keep	Keep	Keep?	Keep?	Discard
High_Cholesterol	Keep	Keep	Keep	Keep	Keep	Keep	Keep	Keep?	Keep	Keep	Keep
Blood_Pressure	Keep	Keep	Keep	Discard	Keep	Keep	Keep	Keep?	Keep	Keep	Keep
Mental_Health	Keep?	Keep	Keep	Keep	Keep	Keep	Keep	Keep	Keep	Keep	Keep
Physical_Health	Keep?	Keep	Keep	Keep	Keep	Keep	Keep	Keep	Keep	Keep	Keep
Age	Keep	Keep	Keep	Discard	Keep	Keep?	Keep?	Keep	Keep	Keep	Try with and without
BMI	Keep?	Keep	Keep	Discard	Keep	Keep?	Keep	Keep	Keep?	Keep?	Try with and without

Table 7: Categorical Variables' Feature Importance

Predictor	Value Counts	Chi-Square	What to do? (One possible way to "solve")
Drinking_Habit	Keep	Keep	Keep
Fruit_Habit	Keep	Keep	Keep
Water_Habit	Keep	Discard	Discard
Education	Keep	Discard	Discard
Smoking_Habit	Keep	Discard	Discard
Exercise	Keep	Keep	Keep
Region	Keep	Discard	Discard
Checkup	Keep	Keep	Keep
Diabetes	Keep	Keep	Keep
Gender	Keep	Keep	Keep

Table 8: Models' Scores

Models	Train score	Val Score	Kaggle Score
Logistic Regression	0.8666	0.8610	0.8188
Logistic Regression with Polynomial features	0.8328	0.8315	0.8041
Naïve Bayes	0.7633	0.8495	NA
Categorical Naive Bayes	0.8306	0.8291	N/A
KNN Classifier	1.0	0.9875	0.9677
Neural Networks	1.0	0.9901	1.0
Gradient Boosting	1.0	0.9801	0.9892
ADA Boosting	1.0	0.9852	1.0
Ridge Classifier	0.8715	0.8633	N/A
Passive Agressive Classifier	0.8693	0.8655	N/A
Decision Tree	1.0	0.9865	1.0
Random Forest Classifier	1.0	0.9867	1.0
Bagging base=default	0.8834	0.8676	N/A
Bagging base=KNN	1.0	0.9890	0.9890
Bagging base=Logistic Regression	0.8741	0.8680	N/A
SVM	1.0	0.9852	0.9485
Stacking	1.0	0.9878	0.90
SVM and KNN	1.0	0.9914	0.9583
SVM, KNN and Decision Trees	1.0	0.9902	0.9574
Voting Classifier (SVM and Decision Trees)	0.9917	0.9769	0.9451
Voting Classifier (KNN and Decision Trees)	1.0	0.9903	0.9565

Table 9: Decision Tree Classifier's Combinations of Features and Scaling Methods

	Train Score not scaled	Val Score not scaled	Train Score MinMax	Val Score MinMax	Train Score Standard	Val Score Standard	Train Score Robust	Val Score Robust
All	1.0	0.976793	1.0	0.976793	1.0	0.976793	1.0	0.976793
Without Height	1.0	0.986508	1.0	0.986508	1.0	0.986508	1.0	0.986508
Without Age	1.0	0.961295	1.0	0.961295	1.0	0.961295	1.0	0.961295
Without BMI	1.0	0.971960	1.0	0.971960	1.0	0.971960	1.0	0.971960
Without Age and Height	1.0	0.975351	1.0	0.975351	1.0	0.975351	1.0	0.975351
Without BMI and Height	1.0	0.979152	1.0	0.979152	1.0	0.979152	1.0	0.979152
Without Age and BMI	1.0	0.965359	1.0	0.965359	1.0	0.965359	1.0	0.965359
Without Age, Height and BMI	1.0	0.982643	1.0	0.982643	1.0	0.982643	1.0	0.982643

Figure 1: Absolute frequencies of the variable “Education”

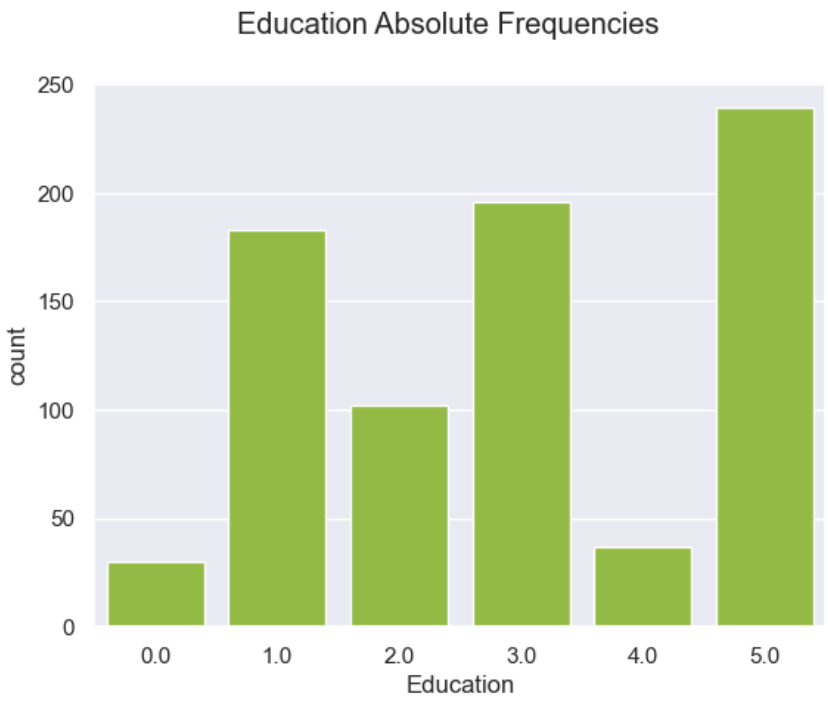

Figure 2: Numeric Variables’ Distributions Histograms before outlier removal

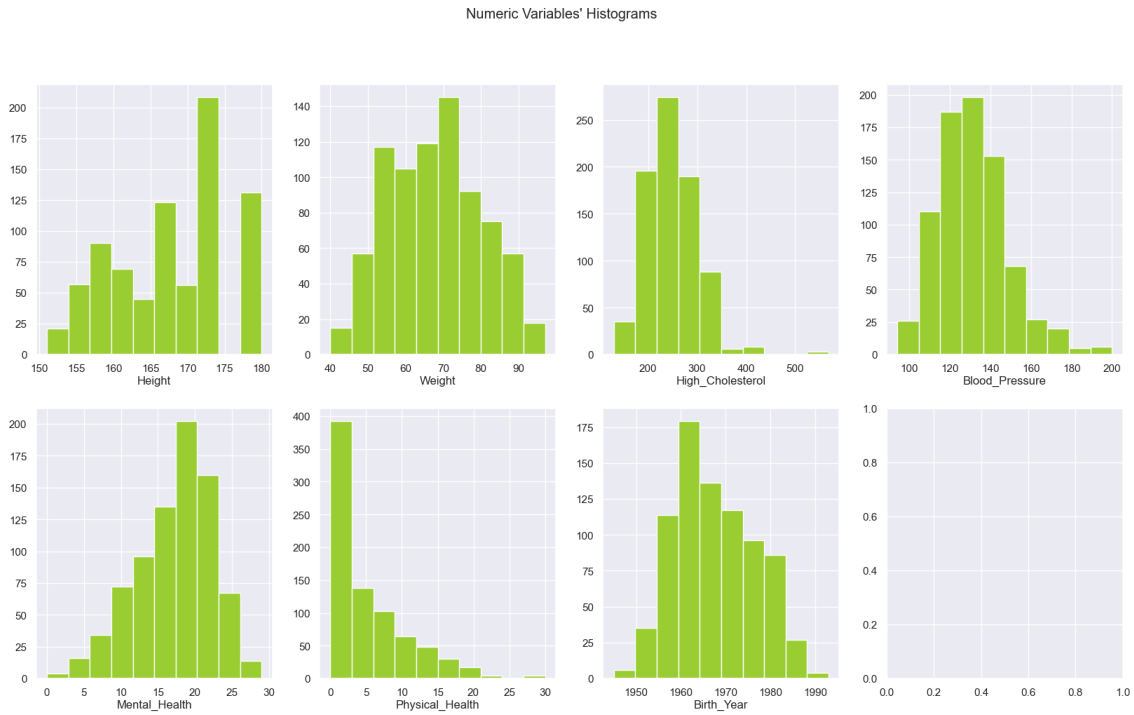

Figure 3: Numeric Variables' Distributions Box Plots before outlier removal

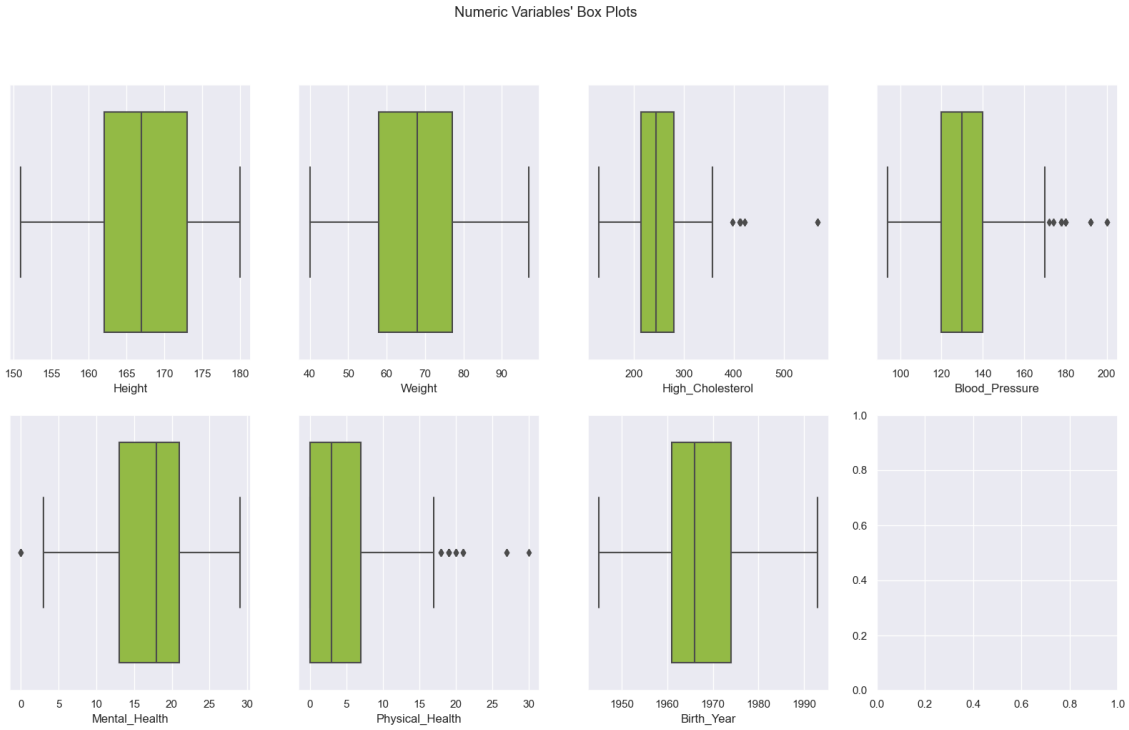

Figure 4: Numeric Variables' Distributions Histograms after outlier removal (IQR Method)

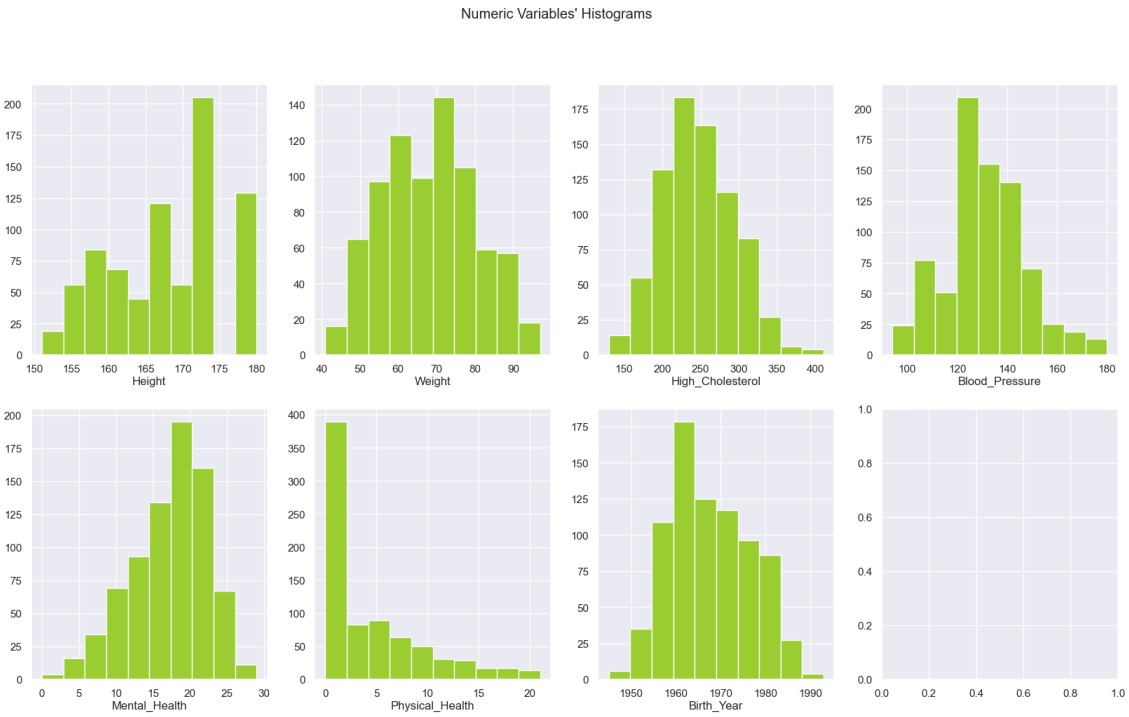

Figure 5: Numeric Variables' Distributions Box Plots after outlier removal (IQR Method)

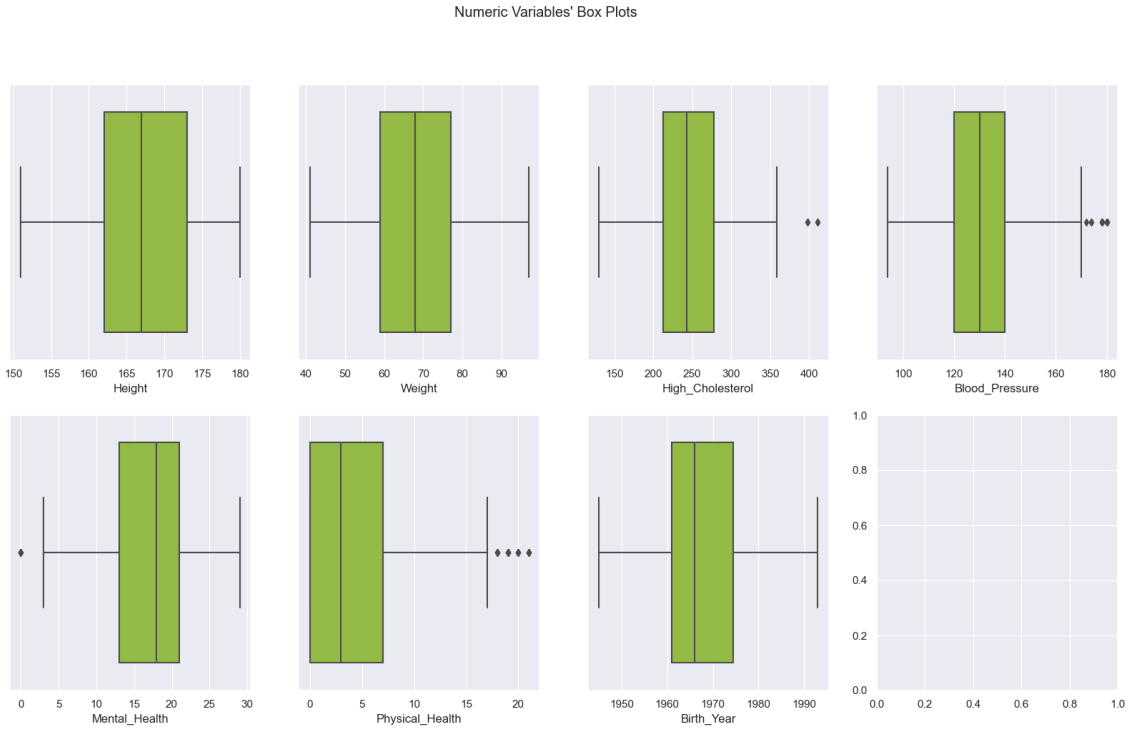

Figure 6: Metric Variables' Spearman Correlation Matrix

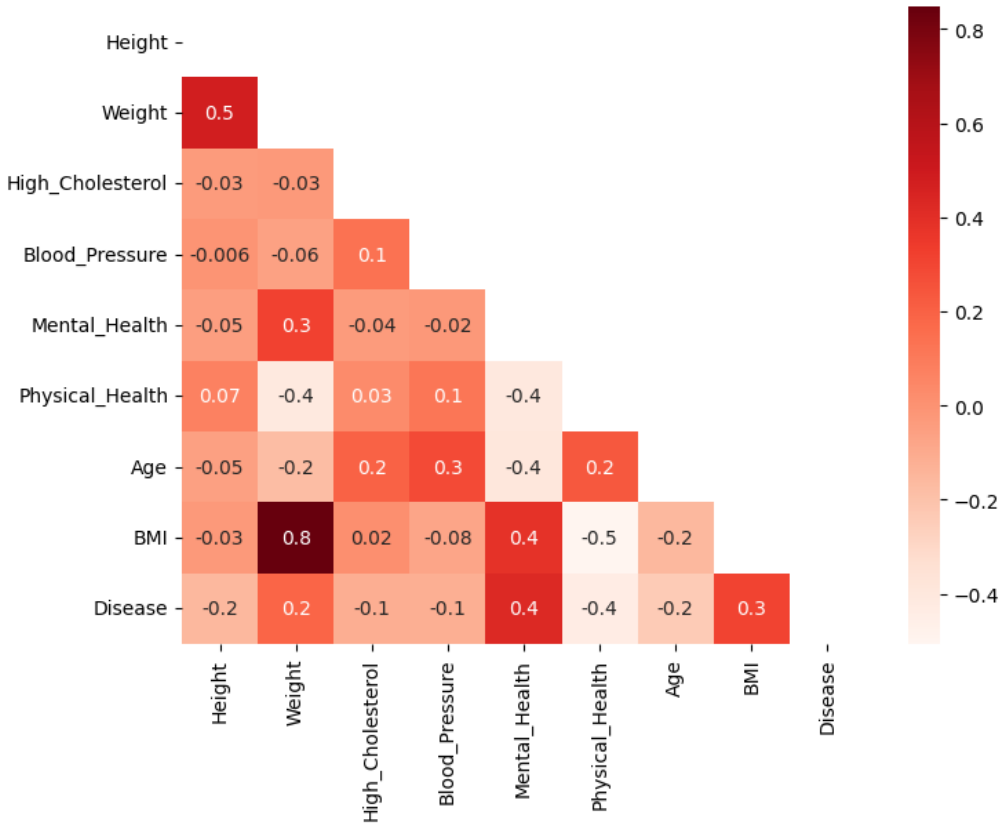

Figure 7: Feature Importance using Lasso Regression

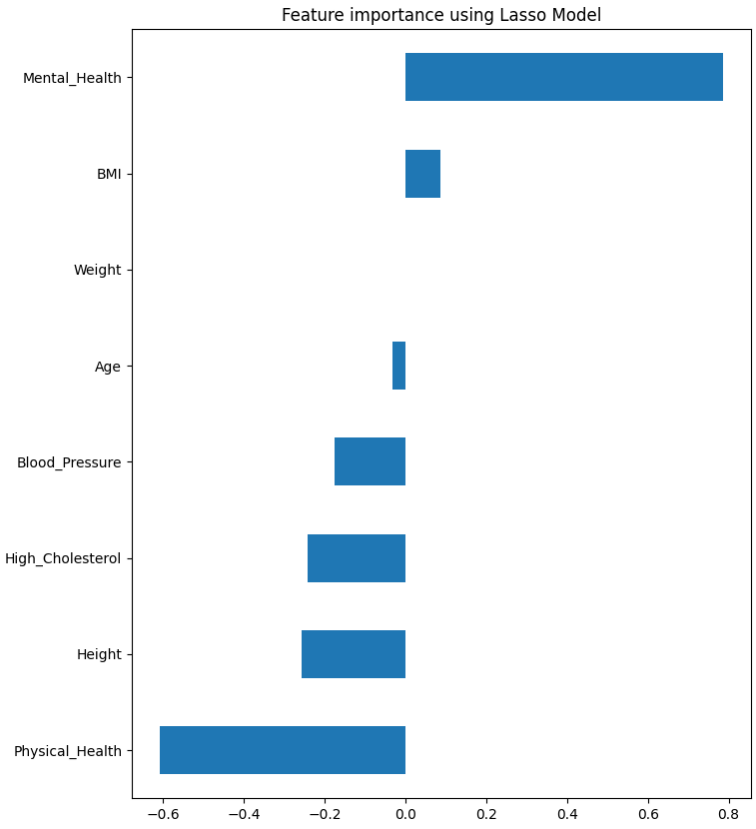

Figure 8: Feature Importance using Ridge Regression

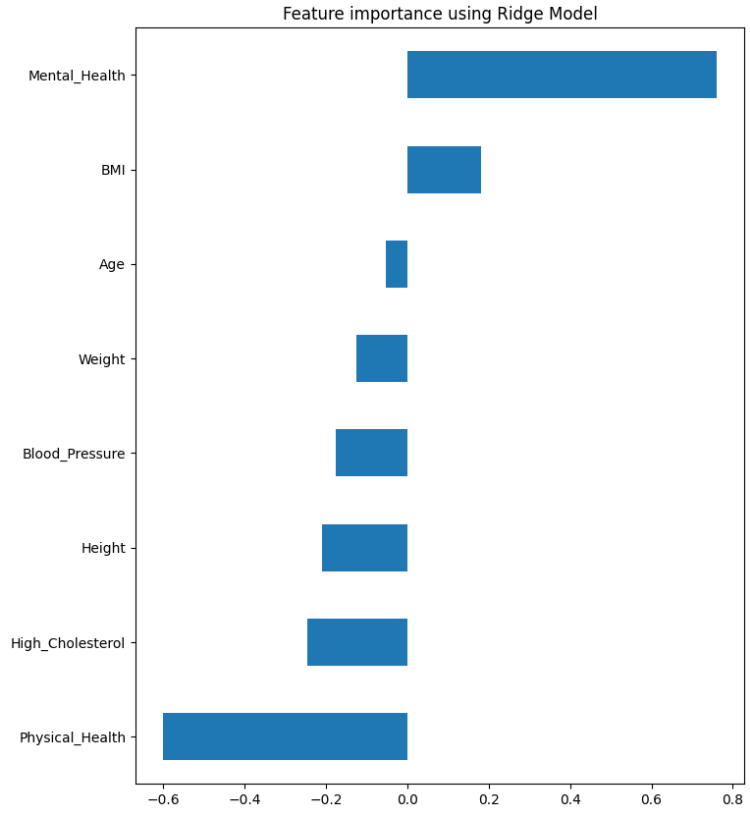

Figure 9: Gini and Entropy Feature Importance based on Decision Trees

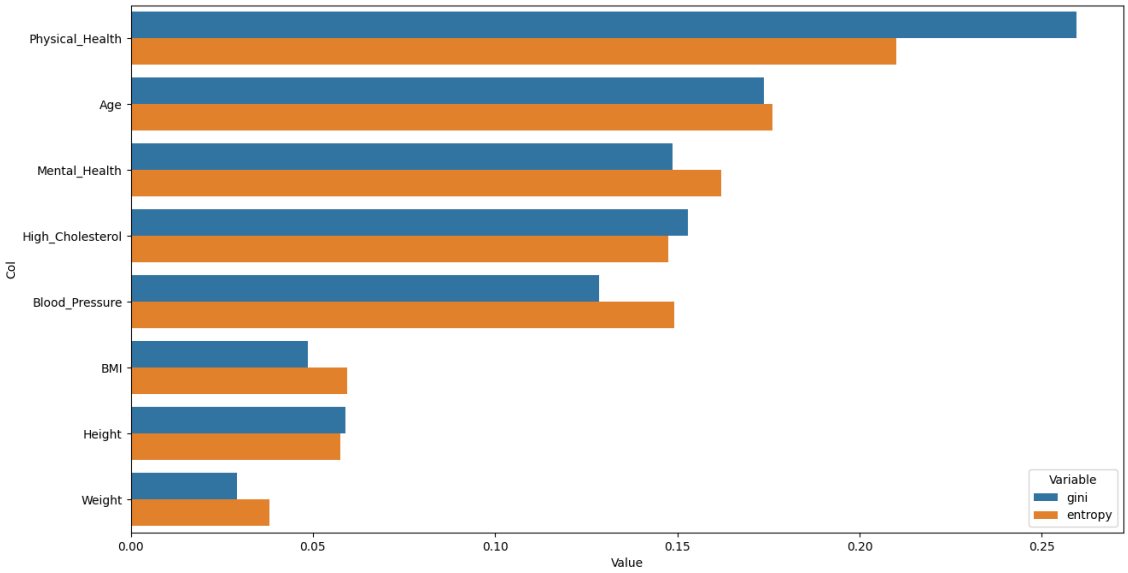

Figure 10: Models' F1 Scores Comparison

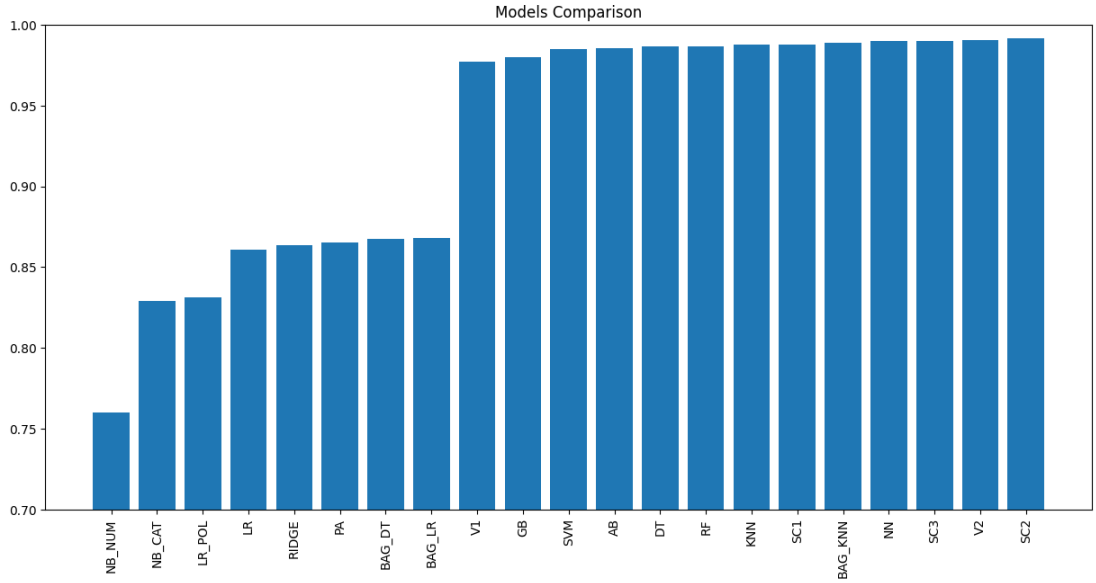

Figure 11: Feature Importance in the Final Decision Tree Classifier