Springboard - Data Science Track

Capstone Project 2

Predicting Coronary Artery Disease in the United States Population

Hiram G. Menendez

February 2023

1. Introduction

1.1 Problem

Coronary artery disease (CAD, also known as “coronary heart disease” and from here on referred to as “heart disease”) is the leading cause of death in the United States. Research has established various health conditions and lifestyle behaviors that are associated with a higher risk of having this disease, including high blood pressure, diabetes, and tobacco use, among others. It should therefore be possible to predict whether an individual will develop heart disease based on information about his or her medical history and lifestyle.

This project aimed to use publicly provided data from the Centers for Disease Control and Prevention (CDC) to develop machine learning models that can accurately predict whether an individual has heart disease given information about his or her medical history and lifestyle behaviors.

1.2 Relevance

Heart disease develops gradually over time. In fact, many don’t realize they have it until they experience a heart attack. Therefore, being able to identify whether an individual is at a higher risk of heart disease is of immense value because it can lead to early medical intervention before the condition worsens. The models developed in this project can be used by the medical community for use with their patients to predict if they have heart disease.

If the models that get developed are interpretable, it can also potentially lead to insights into the relative importance of its features in predicting heart disease. This is useful because it can inform the physician what are the patient’s behaviors that are potentially leading them to be at high risk of heart disease, and he or she can then inform the patient what lifestyle changes are needed to minimize the risk. Also, the knowledge about feature importance gained from the models can help the medical community understand what are the main predictors of heart disease.

1.3 Summary of Results

The trained models performed well when evaluated using cross-validation. The one identified as “the best” had an average recall score for the positive class of 0.82, which means that among the subsample of individuals that had heart disease, it was able to correctly predict they had it 82% of the time. However, it had an average precision score of 0.22, which means that among all of the individuals that got classified as having heart disease, only 22% of them had it. In other words, the model frequently makes false positives.

It is therefore recommended that physicians not use the models to get a final determination on whether the patient has heart disease, but rather use it to determine which patients are good candidates to refer to more accurate diagnostic tests.

The implementation details can be found in the following link: https://github.com/Hiram32/Springboard/tree/main/Capstone_Project_2

2. Approach

2.1 Summary of Approach

The approach was to use heath features of individuals living in the USA found in the Behavioral Risk Factor Surveillance System (BRFSS) - an annual telephone survey made by the Centers for Disease Control and Prevention (CDC) - in order to train models that predict the presence or absence of heart disease in individuals. The survey collects information on individuals such as whether they have hypertension, high cholesterol, reliable access to healthcare, etc.

2.2 Data Acquisition and Description

The dataset consists of the BRFSS survey for the year 2015. It was downloaded from the CDC website as a .csv file.

The survey had a total of 441456 respondents and asked up to a total of 330 questions. Therefore, the dimensions of the tabular data were 441456 rows by 330 columns. Its survey questions are related to topics such as general health, diabetes, sodium intake, etc. The responses to the questions are either categorical or non-categorical. An example of a categorical answer is the one to the question “Do you smoke cigarettes every day, some days, or not at all?”, which can be either “1” (Every day), “2” (Some days), “3” (Not at all), “7” (Don’t know/Not sure), or “9”(Refused). The responses could also be left blank, which means that the question was not asked or is missing for some other unknown reason. On the other hand, the responses to, for example, the question “...how many days per week or month did you have at least one drink or any alcoholic beverage…” can be, for example, “113” (where the first digit means “days per week” and the other two mean “13 days”) or “205 (where the first digit means “days in the past 30 days” and the other two mean “5 days”).

The column names are somewhat unintuitive. For example, the column name for the survey question “Do you smoke cigarettes every day, some days, or not at all?” was given the name “SMOKDAY2”. The meaning of each of the features can be found in a separate document provided by the CDC called “the codebook”. At the start of the project, it was not feasible to change the names of 330 columns, but later in the project I reduced the number of features and gave them intuitive names.

2.3 Data Wrangling

The following steps were taken in order to wrangle the data:

1. Sting entries in date columns - For columns where the entries were meant to be dates, the entries were instead string. Each entry also began with the character “b” for some reason. These were changed with datetime objects.
2. Four columns were empty - This is because some questions are optional during the administration of the survey. These columns were dropped.
3. Fixing the entry value 5.3976e-79 - This value showed up frequently in some categorical columns. I managed to discover by consulting the codebook that this value was meant to be the category-encoding value 0, so I changed it to that.
4. Removing redundant columns - Some columns were redundant. For example, one column has the height of respondents in inches while another has their heights in meters.
5. Categorical values in numeric columns - In some numeric columns, some of the entries were numbers that represent a category. For example, the column “PHYSHLTH” has entries with numeric values between 1 and 30, but it also has entries 77, 88, and 99 which represent categories. I decided to set these categorical entries to null values for later imputation (even though some information is lost due to the categorical nature of these entries).
6. Features that use more than one measuring unit - In these columns, I had to convert the units of some of the entries so that the entire column used the same units.

2.4 Exploratory Data Analysis (EDA)

These are the steps I took to explore some of the characteristics of the data:

1. Plot the distributions of all features - I plotted the distributions for all 330 features. The primary purpose was to verify that, given what is known about the feature, the distributions made sense. For some features, the distributions showed suspiciously large values. For example, some entries for the question "How many alcoholic beverages do you drink each day on average?" had values in the order of 100s. However, these values occurred rarely enough that the most likely explanation is an input error, e.g., a value of 100 was entered instead of an intended 10. Other than this observation, no feature was observed to have unusual values.
2. Explore correlations between features - In order to more easily visualize correlations between features, I decided to make plots using only a subset of the 330 features. There was a large correlation between the feature for "Weight" and "BMI index" but this is due to the mathematical definition between the two. There was also a moderate correlation between the feature for "Weight" and "Height", which is already well known. After calculating the correlation matrix for the entire data, I discovered that only a handful of features have correlations between 0.30 and 0.40, while the rest are below 0.30. Therefore, I did not discover any significant correlations between pairs of features.
3. Explore distributions grouped by different demographics - I decided to explore how different features were distributed among different demographic groups, such as race, age, income level, and level of education. I managed to discover certain trends. For example, the proportion of people having good general health seems to increase as the income level increases, and the proportion of people who smoke tends to decrease as the level of education increases.

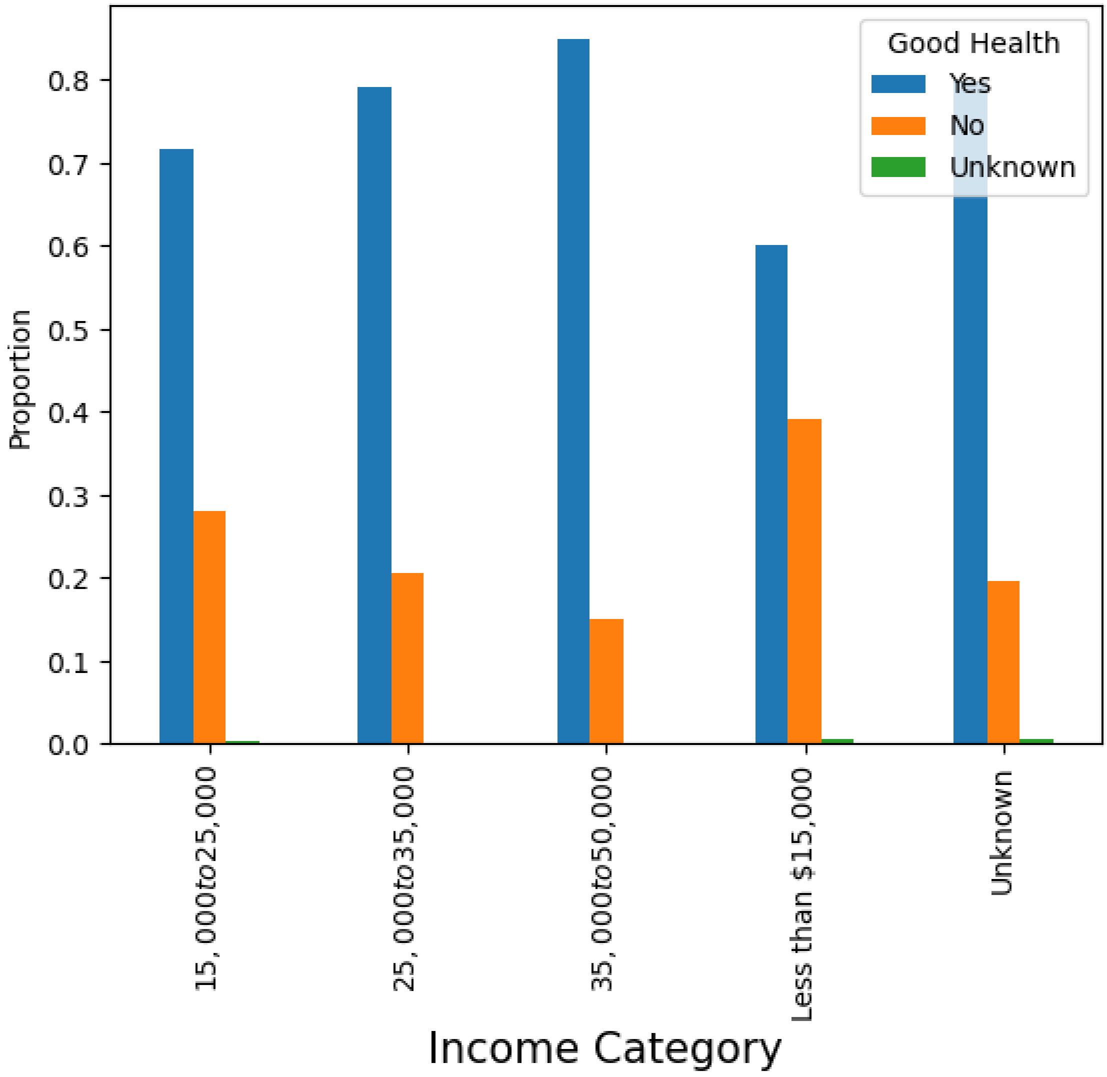


Figure 1: Proportion of samples in the categories of the feature “Good Health” within different income categories

4. Check if the dataset is imbalanced - I discovered that about 8% of the samples have a positive value for the target class, i.e., they have heart disease. This can lead to the models underperforming, so resampling methods had to be applied before training the models in order to remedy this.

2.5. Preprocessing and Training

In this step, I preprocessed the data to make it ready to use in building the predictive models. The following are the steps that I took.

1. Imputation - I imputed the missing values as follows:
    a. Missing values for the target feature - all rows for which the target feature was missing were dropped.
    b. Missing categorical entries - I gave these missing entries their own category.
    c. Missing numeric entries from approximately Gaussian distributions - I replaced the missing entries with the mean of the column.
    d. Missing numeric entries from distributions with long one-sided tails - I replaced the missing entries with the median of the column.

2. Splitting the data frame into feature matrix and target vector
3. One-hot encoding of all categorical features - since numbers represent different categories in the data, the machine learning algorithm might behave as if a higher number for a category means that it is somehow more important. I prevented this behavior by applying one-hot encoding to all categorical features.
4. Train-test spit of dataset - Since the data is imbalanced, a bigger proportion of samples should be used for the training set than would be used otherwise. The fraction of samples used for the test set was therefore set to 20%. I also made sure that the split was stratified so that the proportion of samples in the positive class is roughly the same between the training and test set.
5. Scaling the data - since many machine learning algorithms rely on calculating distances between samples in feature space, I had to ensure that all features' numeric values are roughly of the same order of magnitude. I applied standard scaling on numeric features that are roughly Gaussian-distributed while I used quantile transformation on data that has long one-sided tails in their distribution.

2.6 Modeling

The steps during the modeling step of the project were as follows:

1. Preparing two separate dataframes - The raw data set has 330 features, which became about 1500 after the one-hot encoding of the categorical features. This could potentially lead to overfitting and it also makes the models harder to interpret. Therefore, I decided to also utilize a much smaller subset of the dataframe to train separate models. My main criteria guiding my selection of features was to make the models as simple to interpret as possible. Therefore, for the smaller dataframe, I selected as features only those that correspond to the presence or absence of a particular health condition (for example, having or not having diabetes) or those related to demographics (for example, age group). I ended up selecting a total of 25 features, all

of them categorical, plus the target. After one-hot encoding of the categorical features, I ended up with 107 features.

2. Training a baseline model using both the large and small data set - I wanted to compare the performance using each of the two dataframes when using a baseline model. I chose to build the baseline model with a logistic regression algorithm that used balanced classes (i.e., it assigned weights such that the set became balanced).
3. Analyzing results from baseline models - When the performance between the two previously described models is compared using the metrics precision, recall, f1 score, and roc auc, I discovered that the model trained on the large dataframe was only marginally better. For example, the roc auc score with the large dataframe was 0.86 while the one for the model trained on the small datafame was 0.85. I concluded that this small gain in performance is not worth the dramatic increase in complexity gained when using the large dataframe. Therefore, I decided to carry on the project by only training models with the smaller dataframe.
4. Training the models and evaluating their performance through cross-validation - There are various things to consider when building machine learning models:

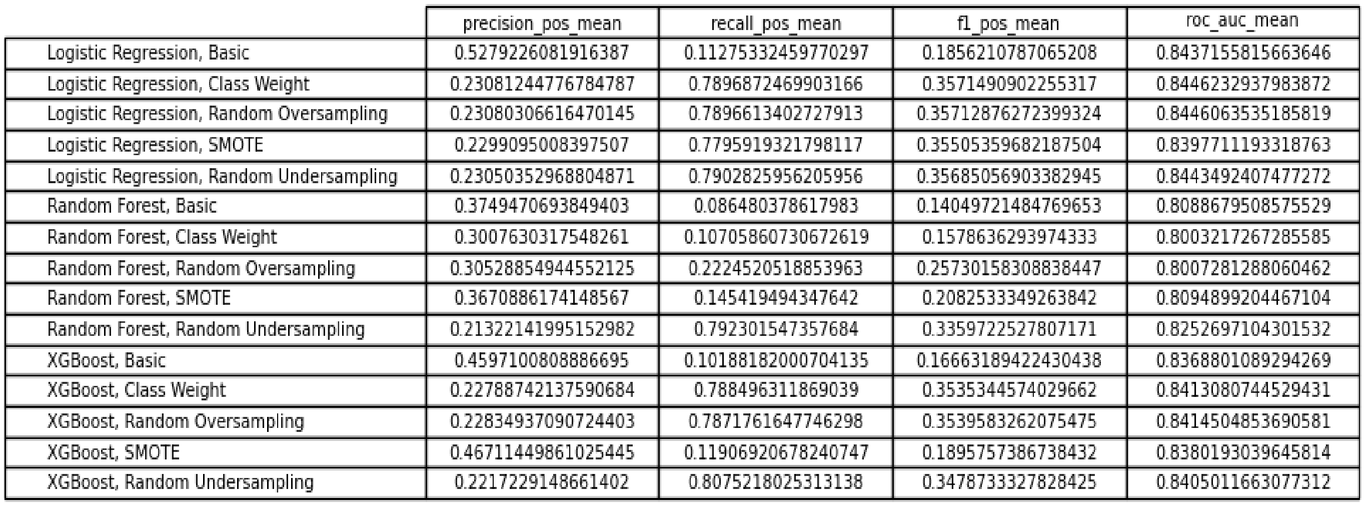
a. The type of machine learning algorithm.
b. The hyperparameters.
c. The type of resampling technique to use, if any.
d. The scoring metrics.

Different choices for the first three items on the previous list will lead to different models. For the type of machine learning algorithm to use, I decided to use three: Logistic Regression, Random Forest, and XGBoost. Since the number of possible combinations of parameters for each algorithm is large, I decided to initially use the default parameters of each of the three algorithms. For the resampling techniques, I decided to choose among five possibilities:

a. No resampling technique
b. Class Weight - Adding weights to the samples so that the data becomes balanced.
c. Random Oversampling
d. SMOTE
e. Random Undersampling

Once a model is built, it is evaluated with cross-validation using the metrics precision, recall, f1 score, and roc auc. Since there are three possible algorithms, each of them is built using their default parameters, and there are five oversampling techniques to choose from, this leads to a total of 15 models to evaluate and compare.

5. Evaluating and comparing the models - The table and figure below summarize the performance results of the models.



	precision_pos_mean	recall_pos_mean	f1_pos_mean	roc_auc_mean
Logistic Regression, Basic	0.5279226081916387	0.11275332459770297	0.1856210787065208	0.8437155815663646
Logistic Regression, Class Weight	0.23081244776784787	0.7896872469903166	0.3571490902255317	0.8446232937983872
Logistic Regression, Random Oversampling	0.23080306616470145	0.7896613402727913	0.35712876272399324	0.8446063535185819
Logistic Regression, SMOTE	0.2299095008397507	0.7795919321798117	0.35505359682187504	0.8397711193318763
Logistic Regression, Random Undersampling	0.23050352968804871	0.7902825956205956	0.35685056903382945	0.8443492407477272
Random Forest, Basic	0.3749470693849403	0.086480378617983	0.14049721484769653	0.8088679508575529
Random Forest, Class Weight	0.3007630317548261	0.10705860730672619	0.1578636293974333	0.8003217267285585
Random Forest, Random Oversampling	0.30528854944552125	0.2224520518853963	0.25730158308838447	0.8007281288060462
Random Forest, SMOTE	0.3670886174148567	0.145419494347642	0.2082533349263842	0.8094899204467104
Random Forest, Random Undersampling	0.21322141995152982	0.792301547357684	0.3359722527807171	0.8252697104301532
XGBoost, Basic	0.4597100808886695	0.10188182000704135	0.16663189422430438	0.8368801089294269
XGBoost, Class Weight	0.22788742137590684	0.788496311869039	0.3535344574029662	0.8413080744529431
XGBoost, Random Oversampling	0.22834937090724403	0.7871761647746298	0.3539583262075475	0.8414504853690581
XGBoost, SMOTE	0.46711449861025445	0.11906920678240747	0.1895757386738432	0.8380193039645814
XGBoost, Random Undersampling	0.2217229148661402	0.8075218025313138	0.3478733327828425	0.8405011663077312

Table 1: Evaluation metrics for each of the 15 models built.

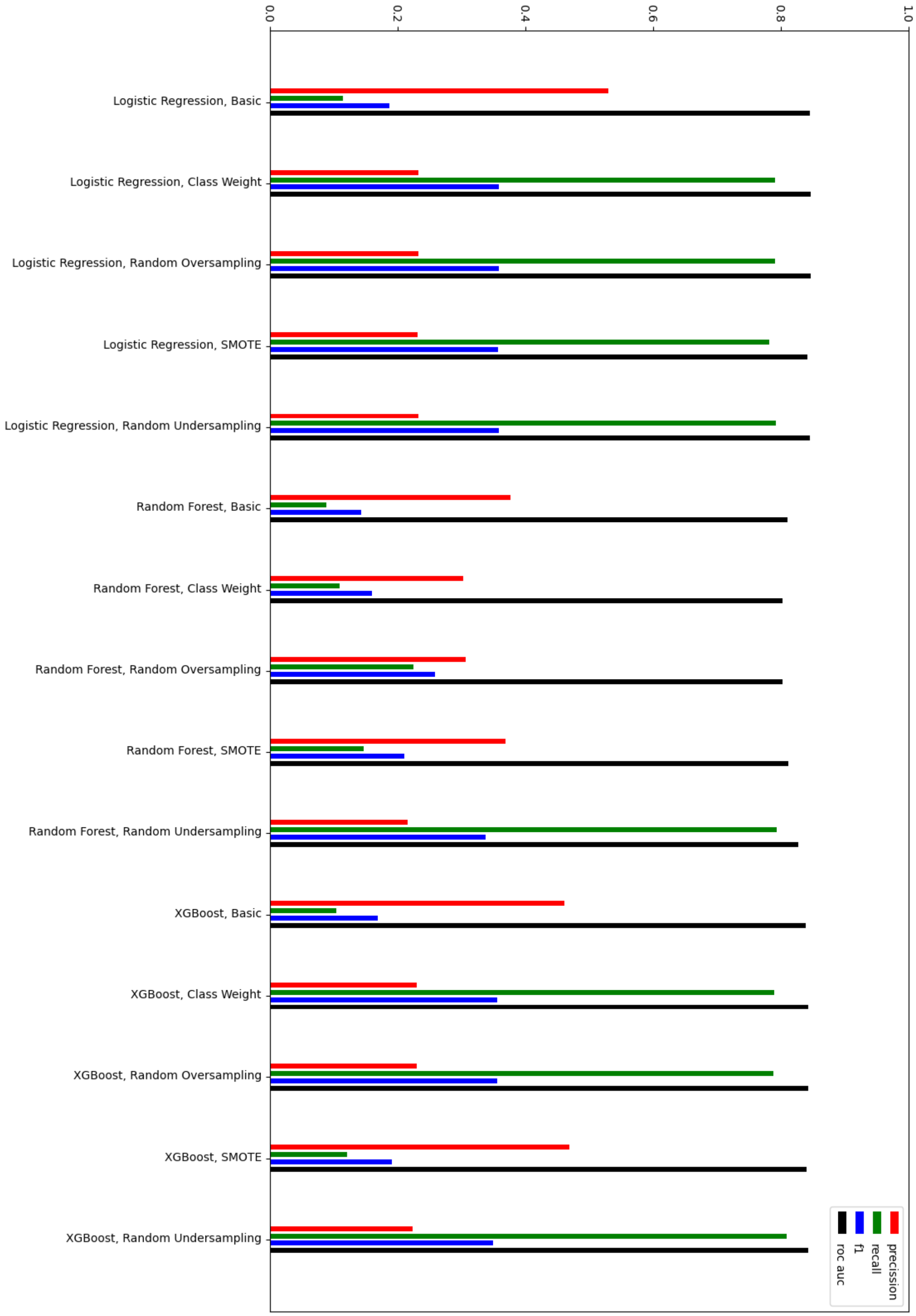


Figure 2: Bar graph for the evaluation metrics of each of the 15 models built.

6. Choosing the best model - based on the performance metric results, I chose the best model. The criteria I used was the recall score because it minimizes the incidences of false negatives, which is highly undesirable in the context of life-threatening diseases. The chosen best model was XGBoost with random undersampling, which had a recall score of 0.81.

7. Hyperparameter tuning - I then proceed to perform a grid search for the parameters that give the best model performance. The parameters that I decided to fine-tune for my XGBoost model with random undersampling were the learning rate, the max depth of each tree, and the regularization constant. After the grid search, the best parameters found for the model were

   a. Learning rate - 0.3
   b. The max depth of each tree - 3
   c. The regularization constant - 0.2

With these parameters, the best model had a cross-validation mean recall score of 0.81 and an roc auc score of 0.85.

3. Findings

3.1 Final Model Results

The following figures summarize the results of the final model.

Confusion Matrix

The values in each cell are normalized by the total number of samples belonging to that class. Therefore, the diagonal cells contain the true negative and true positive rates, while the off-diagonal cells contain the false negative and positive rates. It can be seen that the model does a decent job of correctly predicting that an individual has heart disease, but not as good of a job of correctly predicting when an individual does not have heart disease. In other words, the model is good at detecting individuals that have heart disease but at the cost of predicting too many false positives. However, in this current context concerning a life-threatening disease, it is much more desirable to minimize as much as possible the false negative rate. As can be seen, the model only has a false negative rate of 0.18.

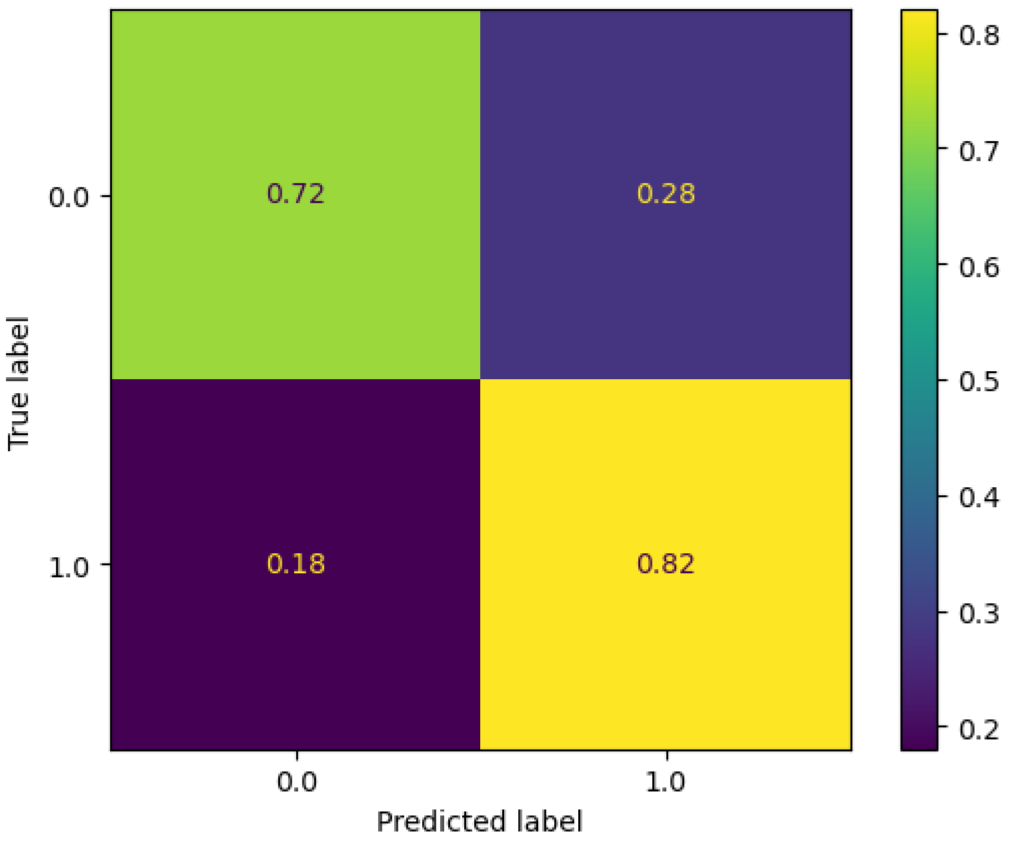
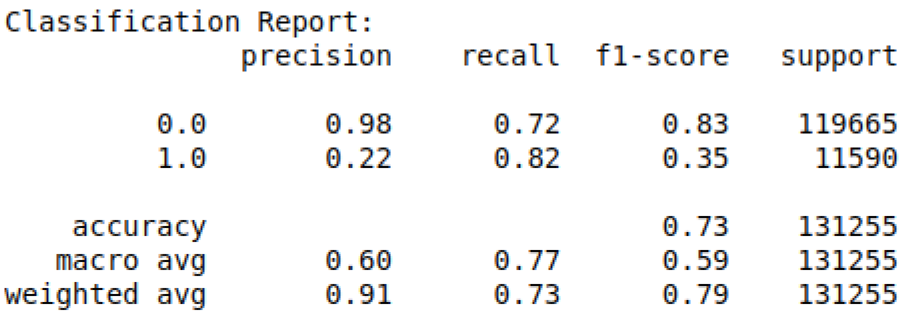


Figure 3: Confusion matrix of the best model.

Classification Report



```
Classification Report:
              precision    recall  f1-score   support

         0.0       0.98      0.72      0.83    119665
         1.0       0.22      0.82      0.35     11590

    accuracy                           0.73    131255
   macro avg       0.60      0.77      0.59    131255
weighted avg       0.91      0.73      0.79    131255
```

Figure 4: Classification report of the best model.

The classification report shows how, for the positive class, the model has a good recall score. However, the positive class also has a very low precision score. Therefore, the model is useful for detecting when an individual has heart disease, but the user of the model must be aware that this comes at the cost of getting many false positives.

ROC Curve

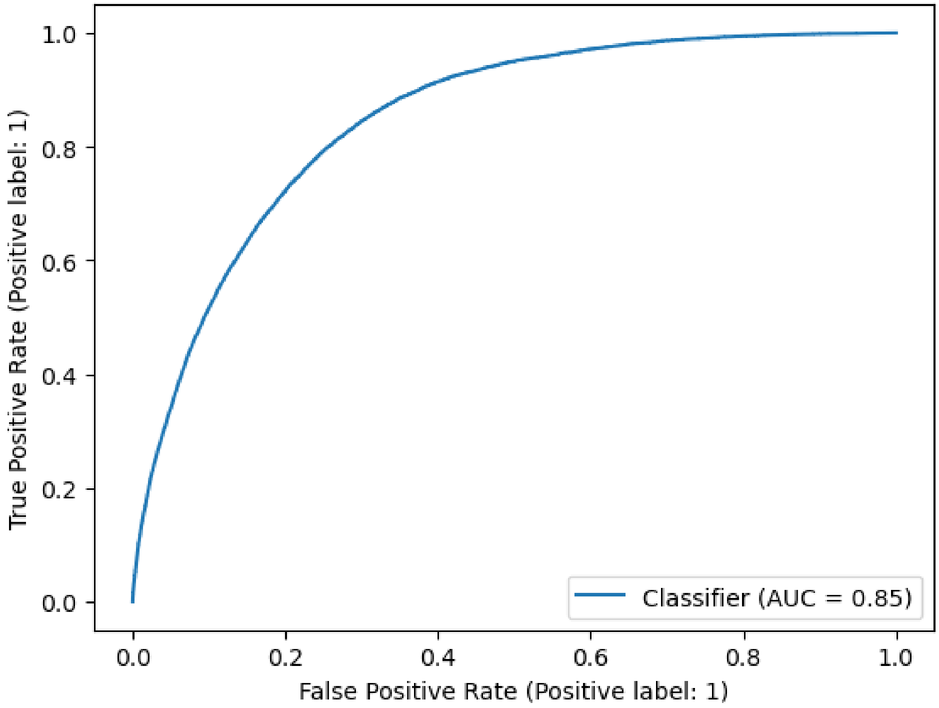


Figure 5: ROC curve of the best model

The area under the curve of the ROC curve is a performance metric that is independent of the probability cutoff that is used to make the classification. A value of 0.85 is considered good, showing that the model does a good job of predicting the presence of heart disease in individuals.

3.2 Interpreting the Model Using SHAP Values

While the built model gives good results, it is also useful to be able to understand why the model makes the prediction that it makes. The SHAP values can be calculated per feature per sample. Their magnitude represents the degree to which each feature contributes to a prediction, and the sign of the value corresponds to whether it leads to an increase or decrease in the prediction of a probability of having heart disease. Therefore, SHAP values with higher magnitudes are considered “more important” in contributing to a prediction for a particular sample.

The plot below shows the top 20 features by order of average SHAP values. Therefore, these are the features that, on average, contribute the most to a prediction and should therefore be considered the most “important”. The top three features that are important in predicting the prevalence of heart disease are “not having hypertension” (Hypertension_2.0), “being in the age group of 65 years old and above” (Age_Cat_6.0), and being male (SEX_1.0).

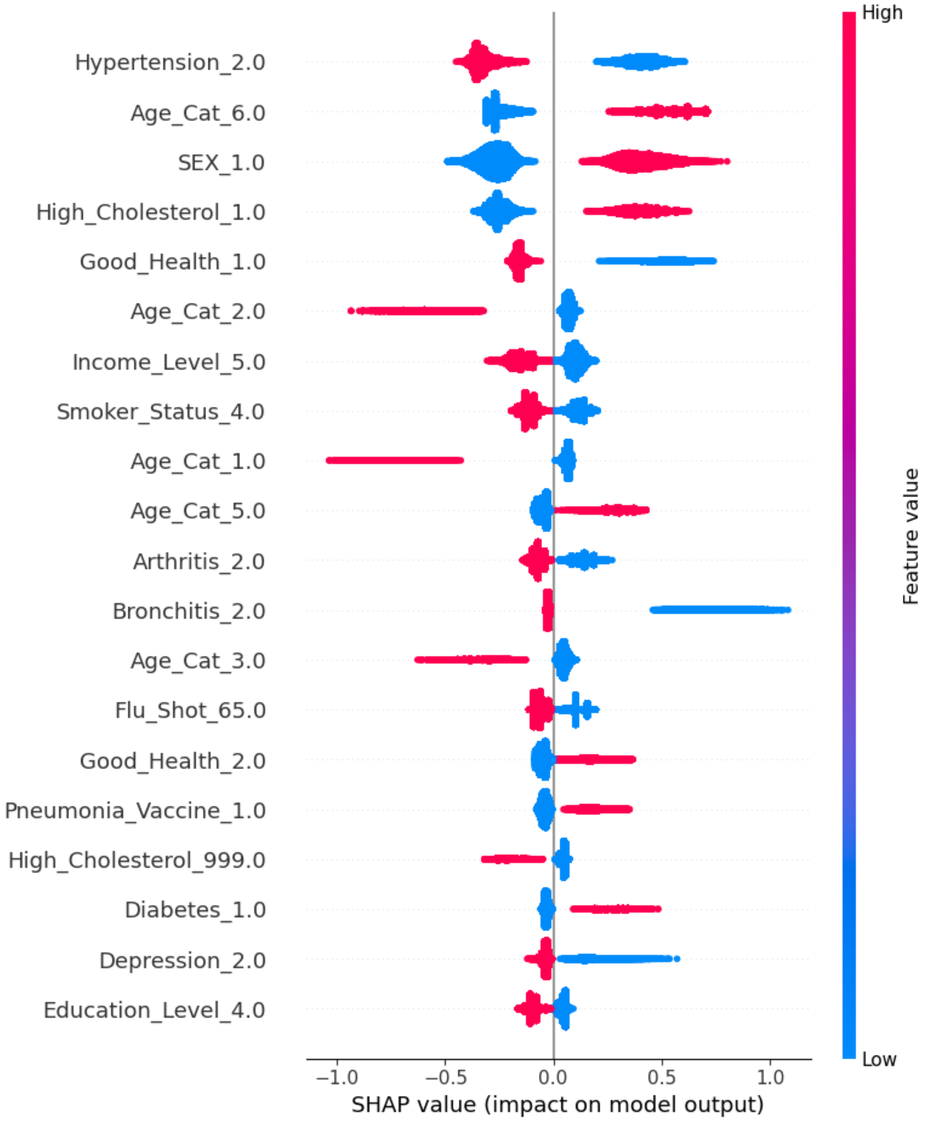


Figure 6: Beeswarm plot for the top 20 features sorted by the average magnitude of the SHAP values.

One interesting observation can be made about the feature Hypertension_2.0. In the plot below, red points represent individuals that don't have hypertension, while blue points represent those that do. Each sample will have different calculated SHAP values, but note how the average value for the samples that do have hypertension (the blue points) appears to be higher (the blue points, on average, tend to be farther away from the black vertical line representing a SHAP value of 0). This means that if an individual has hypertension, that fact will be more important in predicting whether he or she has heart disease than if he or she did not have hypertension. This observation for this particular feature can also be made for others, such as the feature Age_Cat_6.0.

4. Recommendations, Further Research, and Summary

4.1 Recommendations

1. The models are useful if used by physicians to predict if a patient has heart disease. The physician would need to get the patient's information about the 25 features used by the models either from the patient's medical record or by interviewing him or her. The physician would then input those values into the model and he would get as an output a prediction on whether the patient has heart disease, as well as a probability for the prediction.
2. It is very important to stress that the predictions of the models should not be taken as final. One reason is that, even though the best model correctly predicted 82% of the time that an individual has heart disease for the subset of patients that do have it, this health condition has such vast implications for the life of the patient that the physician has a responsibility to be as certain as possible that this is the case. Another reason is that the models have a very high rate of incorrectly predicting that an individual has heart disease when he or she does not: among the total of individuals that were predicted to have heart disease by the best model, only 22% of them had it.

   Therefore, the models should not be used to make a final prediction about whether an individual has heart disease or not, but rather to determine which individuals are the best candidates for more specialized testing. This will be even most useful in situations where diagnostic tools are limited such that only a few individuals can be tested within a given period. If the models predict that the individual does not have heart disease, he or she may not be considered a good candidate for more specialized testing because the false negative rate is low. However, if the models predict that the individual has heart disease, the prediction should be taken seriously, and the individual should be referred to more specialized testing.

3. The physician can also use the models and the dataset used to train them to calculate the SHAP values of each feature for a patient. This can help the physician explain to the patient why he or she is considered to be likely to have heart disease. For example, if a particular patient has a large and positive SHAP value for the feature “Age Category” and the patient belongs to the age category of “65 years old or above”, the physician can explain to the patient that the most important consideration that went into deciding that he or she is at high risk of heart disease is his or her age.

4.2 Further Research

Due to time constraints, only a small number of machine learning algorithms were used in this project. A further avenue of research is to see if better-performing models can be built using other machine learning algorithms.

Another potential avenue of research is to generate counterfactual explanations for the built models. This will allow us to give answers to questions such as “what lifestyle changes could a patient have changed about him or herself in order to not have been considered at high risk of heart disease”. This is useful in patient care because it allows the physician to explain to the patient the kinds of lifestyle changes that he or she can implement in order to lower his risk of heart disease.

4.1 Summary

In this project, I was able to build machine learning models capable of predicting the presence of Coronary Heart Disease (CAD) in individuals based on their health characteristics and lifestyle. The models were trained on data from the Centers for Disease Control and Prevention’s (CDS) Behavioral Risk Factor and Surveillance System (BRFSS) survey for the year 2015. They used 25 features, all of them binary classifiers, such as whether the individual has hypertension, is a smoker, has diabetes, or has high cholesterol.

The models tested well when their performances were evaluated using cross-validation: the best one had an average recall score of 0.82 for the positive class (has heart disease) and an roc auc score of 0.85. One limitation of the models is that it has a very low precision score for the positive class, which means that their predictions generate a lot of false positives.

SHAP values were calculated for each of the samples to provide an interpretation of the models. From these values, I was able to determine which features, on average, were the most important in determining whether an individual has heart disease.