Aaron Leung, Yuze Zhang, Jeremiah Asala CS 6220
CS 6220 Final Project Report

Predictive Model for Early Detection of Heart Disease

Introduction

According to the Centers for Disease Control and Prevention (CDC), 6 in 10 adults in the US have a chronic disease and 4 in 10 adults have two or more chronic diseases (Centers for Disease Control and Prevention, 2022). The list of chronic diseases includes heart disease, cancer, chronic lung diseases, diabetes, to name a few. These chronic diseases account for the leading causes of death and disability and are the leading drivers of the nation's annual health care cost (Centers for Disease Control and Prevention, 2022). Heart disease, in particular, is the leading cause of deaths in the United States and has a significant impact on healthcare cost. According to the CDC, 697,000 people in the United States died from heart disease in 2020. On top of that, CDC also reported that heart disease cost the United States $229 billion during 2017-2018 (Centers for Disease Control and Prevention, 2022). Many individuals only discover they have heart disease after experiencing symptoms thus highlighting the need for preventive care and early detection. Research in the field has identified important risk factors such as high blood pressure, high cholesterol, and smoking that play a role in predicting heart disease. Given the prevalence and serious nature of heart disease, there is an urgent need for predictive measures that can accurately assess an individual's risk of developing heart disease. By being able to predict which patients are at high risks of developing heart disease, we can hopefully intervene at earlier stages before the condition becomes severe. As we learned during this semester, machine learning classification models such as logistic regression offer a promising approach to develop predictive models. For our final project, we will be building two classification prediction models to predict heart disease: Logistic Regression and Random Forest. We will then compare the two models and analyze which model offers better performance. Additionally, we are interested in identifying which risk factors are good predictors of our outcome variable.

Data Description

Our data comes from the Heart Disease Health Indicators Dataset hosted on Kaggle. This data is sourced from public health surveys released by the Center for Disease Control and Prevention (CDC). Specifically, our dataset comes from The Behavioral Risk Factor Surveillance System (BRFSS), which is the nation's premier system of health-related telephone surveys that collect state data about U.S. residents regarding their health-related risk behaviors, chronic health conditions, and use of preventive services. The BRFSS completes more than 400,000 adult interviews each year and collects data in all 50 states as well as the District of Columbia and three U.S. territories.
While our dataset originates from the BRFSS 2015 dataset, this dataset has already been preprocessed and cleaned, containing only features that are relevant to diagnosing heart disease. Overall, this dataset contains 253,680 survey responses and keeps 22 of 330 columns of the original dataset which include the following:

Table 1

HeartDiseaseOrAttack	HighBP	HighChol	CholCheck	BMI
Smoker	Stroke	Diabetes	PhysActivity	Fruits
Veggies	HvyAlcoholConsump	AnyHealthcare	NoDocbcCost	GenHlth
MentHlth	PhysHlth	DiffWalk	Sex	Age
Education	Income			

Before proceeding further, we reviewed the raw 2015 dataset to see what columns were removed and came to the conclusion that the dataset we planned to use contained the most relevant columns. As previously mentioned, we planned to create two classification prediction models. Before doing so, we first performed exploratory data analysis in order to gain a better understanding of our data such as relationships between our predictor variable and target variable (HeartDiseaseOrAttack). More importantly, this allowed us to ascertain the balance of classes and gave us a preliminary idea of which features may be good predictors for our target variable.

The dataset that we will be using as well as original BRFSS 2015 dataset can be found at the following links:

Heart Disease Health Indicators Dataset:
https://www.kaggle.com/datasets/alexteboul/heart-disease-health-indicators-dataset

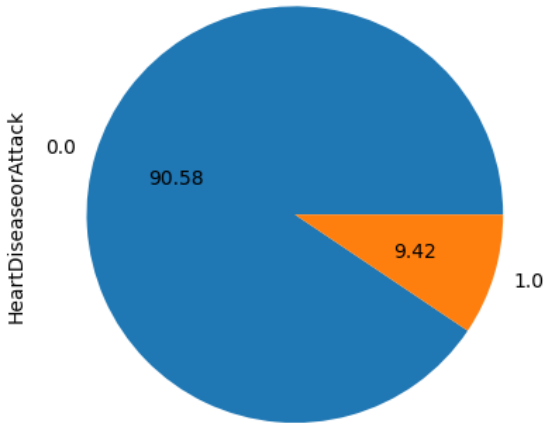
BRFSS 2015 codebook and dataset:
https://www.cdc.gov/brfss/annual_data/2015/pdf/codebook15_llcp.pdf
https://www.kaggle.com/datasets/cdc/behavioral-risk-factor-surveillance-system?select=2015.csv

Exploratory Data Analysis

Target Variable:

The first feature that we looked at was the target variable that we want to predict (HeartDiseaseOrAttack). To visualize the distribution of the values in our target variable we used a pie plot using built-in pandas function.

Figure 1



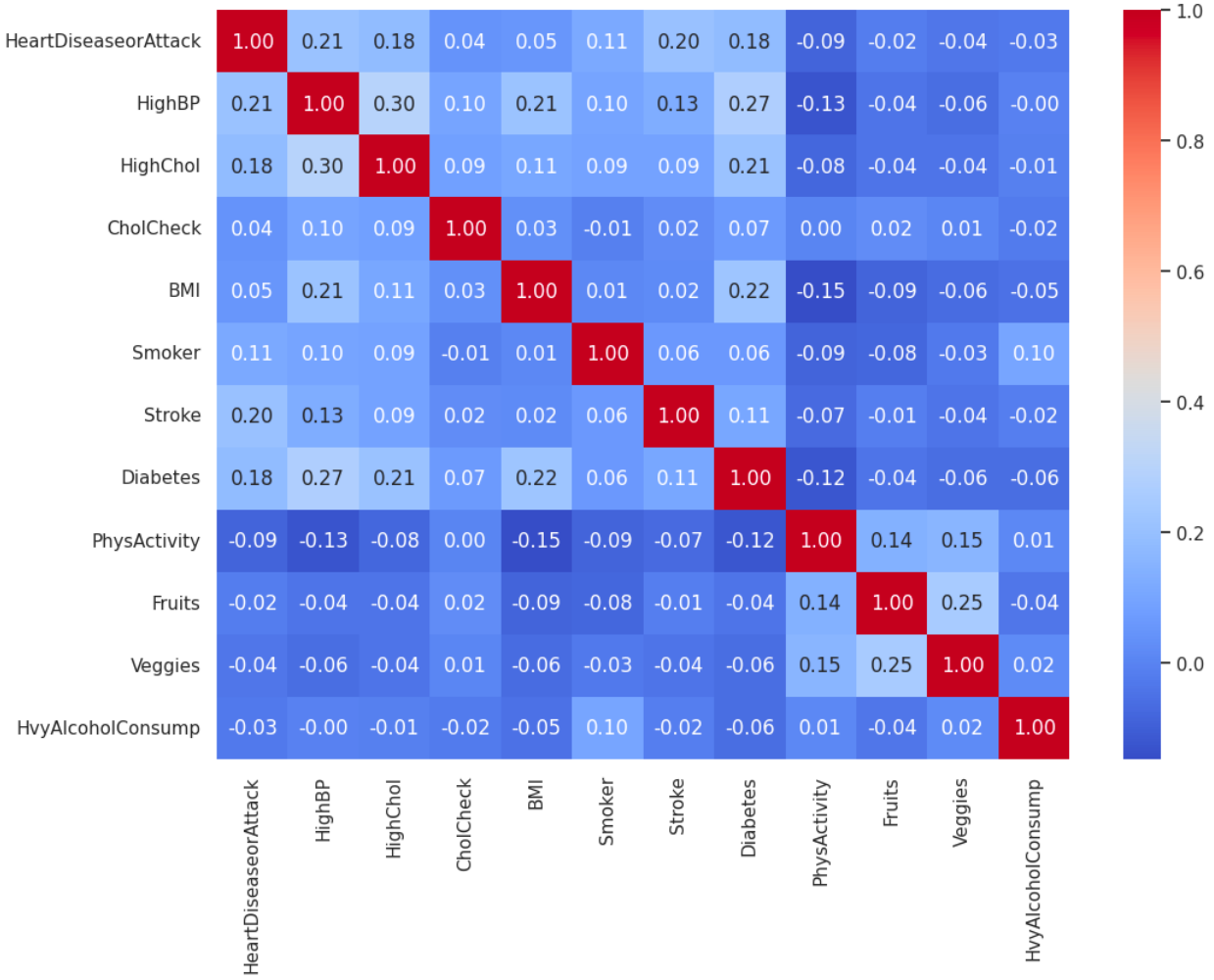
Pie chart showing class distribution between classes 0.0 and 1.0

As we can see, there is a huge class imbalance with ~90% of people having no heart disease or attack and ~10% people having heart disease or attack. We refer to class 0.0 as the majority class and class 1.0 as the minority class. In general, class imbalance can significantly affect the performance of our prediction model. We will further explore and discuss how class imbalance can affect our accuracy, the reasons behind it and how to address it in the following section.

Predictor Variables:

For the predictor variables, we used a combination of bar plots and heatmap to visualize correlation between each respective predictor variable and the target variable. We repeated this process for each predictor variable and noted our observations. Below are heatmaps showing correlation between the target variable and all predictor variables.

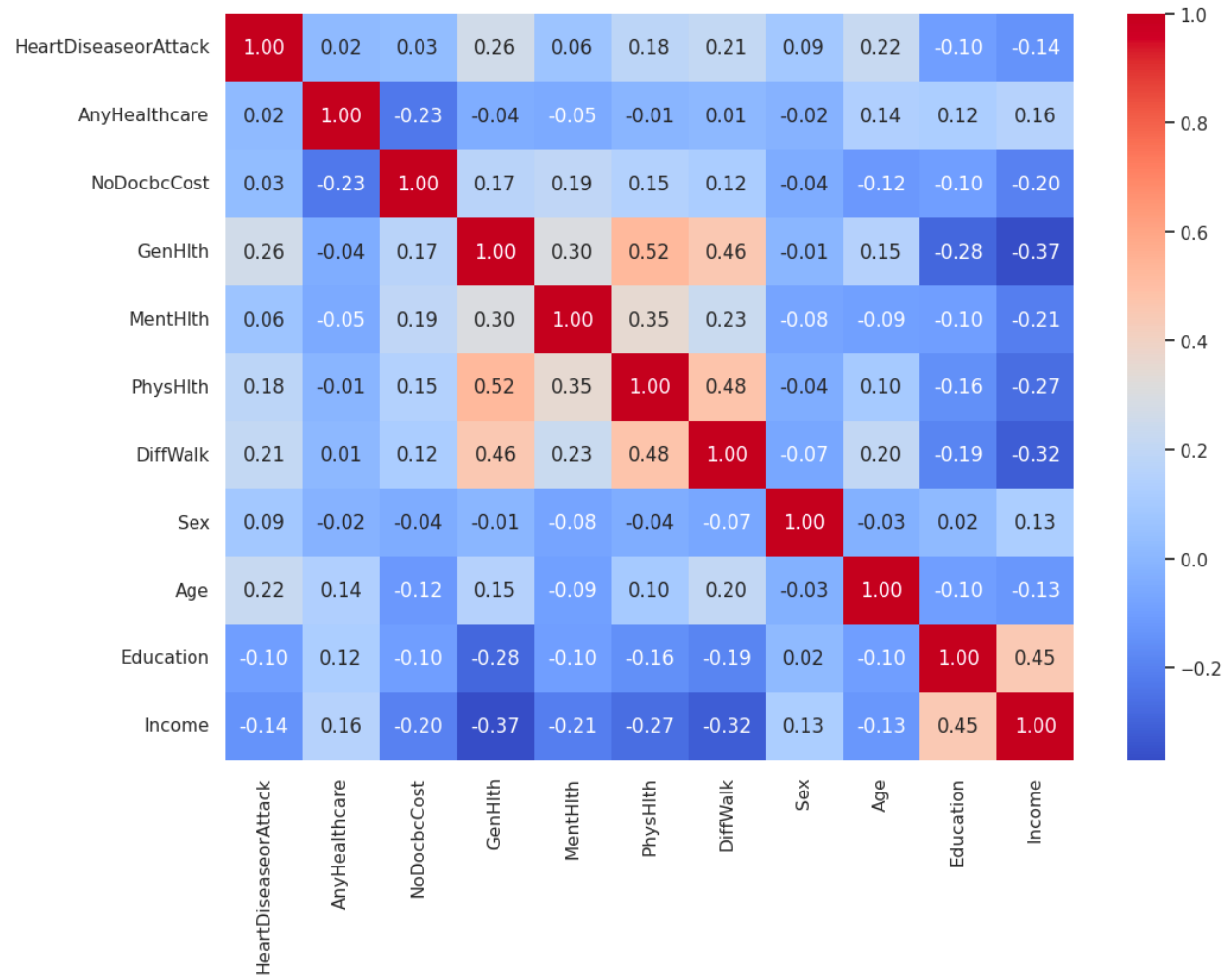
Figure 2



Heatmap showing correlation between target variable and *HighBP, HighChol, CholCheck, BMI, Smoker, Stroke, Diabetes, PhysActivity, Fruits, Veggies*, and *HvyAlcoholConsump*

From the above heatmap, *HighBP, HighChol, BMI, Smoker, Stroke,* and *Diabetes* show the highest correlation with *HeartDieseaseorAttack*. These predictor variables may be good predictors of the outcome variable.

Figure 3

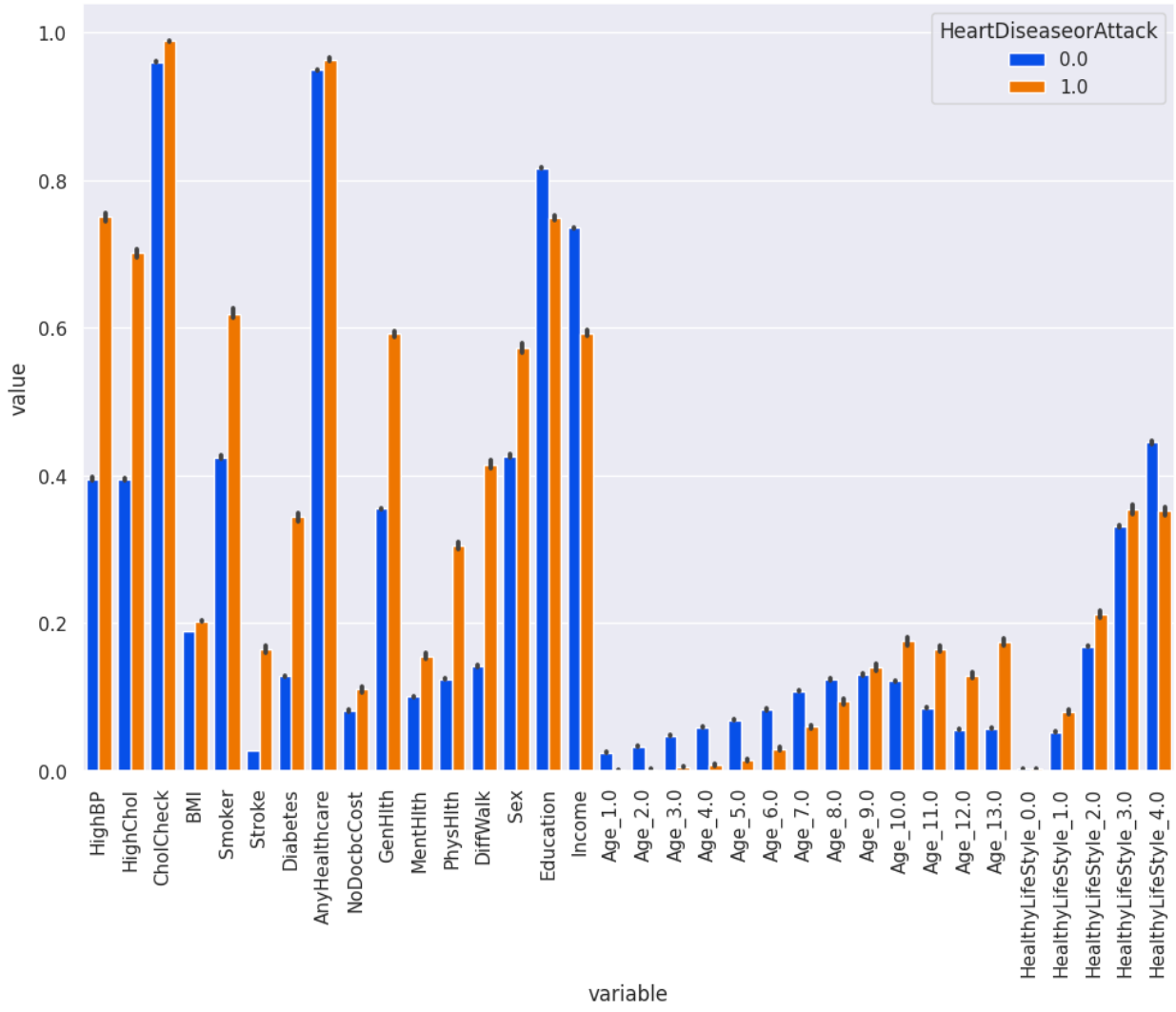


Heatmap showing correlation between target variable and *AnyHealthcare, NoDocbcCost, GenHlth, MentHlth, PhysHlth, DiffWalk, Sex, Age, Education*, and *Income*

In Figure 3 above, *GenHlth, PhysHlth, DiffWalk*, and *Age* show the highest correlation with the target variable. This suggests that the aforementioned features could be good predictors of the target variable.

Additionally, we were interested in visualizing the difference in mean between both classes (0.0 and 1.0) for each predictor variable. To do this, we grouped the entire dataset by the target variable *HeartDiseaseorAttack*, and computed the mean of each predictor variable within each target class (0.0 and 1.0). The purpose of this approach was to get a sense of how the mean of each feature differs between the two target classes (0.0 and 1.0). A higher mean for a particular feature in one of the two target classes might suggest some correlation between the feature and the target variable.

Figure 4



Bar chart showing the difference in mean between classes 0.0 and 1.0 for each predictor variable

From the above plot, we were able to ‘eyeball’ and identify some predictor variables that showed significant difference in mean values between the target classes. For example, the feature *HighBP* shows a significant difference in the height of the bars. Specifically, the mean value of class 1.0 (having a heart disease or attack) is higher than the mean value of class 0.0, which suggests that *HighBP* is likely an important predictor of the outcome variable. Other features or predictor variables that show a significant difference in mean values between the two target classes are *HighChol*, *Smoker*, *Stroke*, *Diabetes*, *PhysHlth*, and *GenHlth*. Through data exploration, we were able to discover a class imbalance and achieve one of our project goals, which was to identify features that might be good predictors of the outcome variable.

Feature Engineering

In this section, we provide a brief overview of the feature engineering that we conducted. Based on the heatmap from our data exploration, we found that PhysActivity, Fruits, Veggies and HvyAlcoholConsump were weakly correlated with HeartDiseaseorAttack with correlation coefficients of -0.09, -0.02, -0.04 and -0.03 respectively. As these four features are related to an individual's lifestyle, we decided to consolidate these four features into a single feature called healthy lifestyle with a score range from 0 to 4. A score of 4 denotes a highly healthy lifestyle while a score of 0 denotes a highly unhealthy lifestyle. To calculate this score, we simply sum up the value of PhysActivity, Fruits, Veggies and LghtAlcoholConsump. Before calculating the score, we need to first create the complementary feature (LghtAlcoholConsump) from HvyAlcoholConsump. This is due to the fact that a value 1.0 for HvyAlcoholConsump means an individual is a heavy drinker, and therefore, this column's value of 1.0 should not be included in calculation of the healthy lifestyle score. After creating the healthy lifestyle feature, we dropped the original four columns (PhysActivity, Fruits, Veggies, HvyAlcoholConsump) along with the complementary feature (LghtAlcoholConsump) from the dataset.

Technical Discussion

Our project's goals were:

1. Create two classification models (Logistic Regression and Random Forest) for predicting heart disease and compare both model performances.
2. Determine which features are good predictors for our target variable.

Prediction Models

Purpose

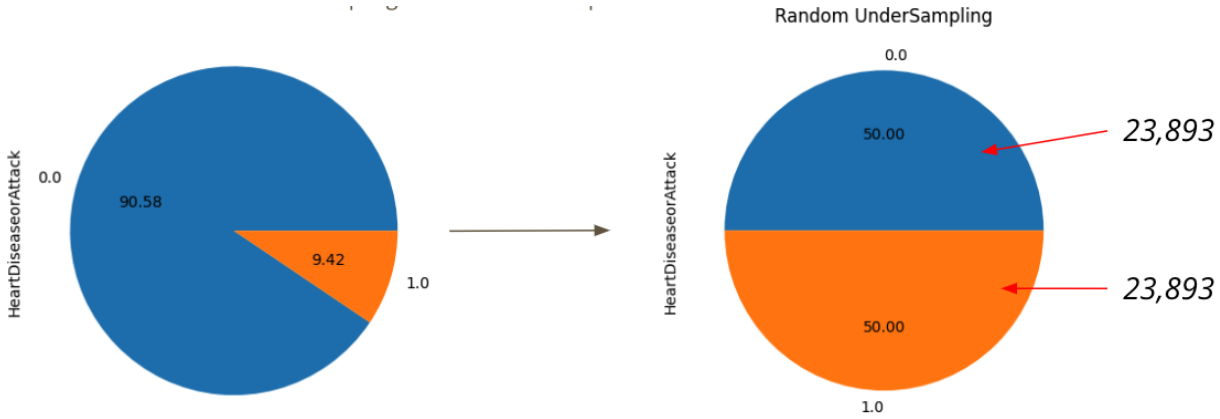
Train two classification models that can accurately assess an individual's risk of developing heart disease based on a patient's medical history/ survey responses. We utilize the sklearn Python library to implement both classifications: logistic regression and random forest.

In the exploratory data analysis section, we observed a heavy class imbalance in our target variable (HeartAttackOrDisease). As previously mentioned, a class imbalance could affect the performance of our prediction models and lead to misleading results. For instance, a model trained on an imbalanced dataset, such as the one in our case, could yield high accuracy. This high accuracy would be attributed to the model's tendency to predict class 0.0 a majority of the time (due to an overwhelming amount of negative sample) thus giving us the impression the model is effective. However, this model would be ineffective and unhelpful in achieving our goal as it is failing to identify the positive cases (minority class).

There are several techniques that can be used to address class imbalance. For our project, we chose to address this issue using an approach called random undersampling. Random undersampling essentially reduces the size of the majority class until it becomes proportional to

the size of the minority class. In other words, after random undersampling is performed, there should be an equal number of samples from both the majority and minority class. To perform random undersampling, we utilized the RandomUndersSampler function from the imblearn library which undersamples the majority class by randomly selecting samples until the size of the majority class matches that of the minority class. Furthermore, we performed random undersampling before the train/ train split in order to preserve all of the positive cases from our dataset.

Figure 5: Visual representation of Random UnderSampling



Pie charts showing the class distributions after performing random undersampling.

For discussion purposes, we decided to also train our classification models using the imbalance dataset in order to establish a baseline for comparison. To evaluate each model's performance we examined various metrics included in the classification report from sklearn.metrics. The results for both logistic regression and random forest were as follows:

Logistic Regression

Baseline Model:

	precision	recall	f1-score	support
0.0	0.92	0.99	0.95	46010
1.0	0.55	0.13	0.21	4726
accuracy			0.91	50736
macro avg	0.74	0.56	0.58	50736
weighted avg	0.88	0.91	0.88	50736

Random Undersample Model:

	precision	recall	f1-score	support
0.0	0.77	0.76	0.77	4741
1.0	0.77	0.78	0.77	4817
accuracy			0.77	9558
macro avg	0.77	0.77	0.77	9558
weighted avg	0.77	0.77	0.77	9558

Random Forest:

Baseline Model:

	precision	recall	f1-score	support
0.0	0.91	0.98	0.95	45911
1.0	0.43	0.12	0.18	4825
accuracy			0.90	50736
macro avg	0.67	0.55	0.56	50736
weighted avg	0.87	0.90	0.87	50736

Random Undersample Model:

	precision	recall	f1-score	support
0.0	0.79	0.70	0.74	4839
1.0	0.73	0.80	0.76	4719
accuracy			0.75	9558
macro avg	0.76	0.75	0.75	9558
weighted avg	0.76	0.75	0.75	9558

Prediction Model Analysis

In order to assess the performance of our models, we need to first understand key metrics from the classification report which are precision, recall and f1-score. Our analysis primarily focuses on precision and recall as f1-score essentially combines the former two metrics into one. Beginning with precision, precision is defined as how good the model is at correctly predicting a positive result. In other words, out of all the samples that are predicted as positive, how many are actually positive. Recall, on the other hand, is defined as how good is the model at capturing the actual positive results in the set (Smolic, 2022). Put another way, it's an indicator of how effective the model is at minimizing situations where an actual positive result is predicted as negative. In general, it is desirable for both precision and recall to be as high as possible. However, precision and recall are often in tension. That is, as precision increases, recall decreases and vice versa (Google, 2022). The choice of which metric (precision or recall) to prioritize ultimately depends on the problem statement. In the context of our problem statement, since our goal is to predict patients who are at a risk of developing heart disease our goal is to prioritize recall over precision. We will now analyze our classification model performances beginning with logistic regression. Please note that the following discussion will primarily revolve around analyzing the precision and recall of class 1.0 of HeartDiseaseorAttack.

Logistic regression analysis:

Starting with the baseline model, we observed that it produced a 91% accuracy. However, upon closer inspection, we noticed that the recall for class 1.0 (minority class) was only 13% indicating poor performance in detecting the minority class. Precision, on the other hand, was moderately better at 55%. As discussed earlier, the high accuracy of the baseline model is mainly attributed to its ability to correctly predict class 0.0 (majority class) rather than its ability to correctly identify the minority class. Therefore, this model is unhelpful in accomplishing our project goal of identifying individuals who have heart disease or attack.

When examining the model that employed random undersampling, we noticed a significant improvement in recall from 13% to 79%. Additionally, we saw improvements in precision from 55% to 77%. Accuracy, on the other hand, had a moderate drop from 91% to 77%. Although the accuracy for the second model (random undersampling) is much lower, this model offers a more accurate depiction of our model’s performance.

Random forest analysis:

When examining our random forest models, we recognized a similar pattern to that of the logistic regression models. More specifically, the baseline model produced a high accuracy of 90% while recall for the minority class remained low at 12%. Precision was slightly lower at 43%. Overall, reasons for similar performances are due to the same reasons described above. Similarly, when examining the model that employed random undersampling, we noted improvements in both precision and recall: from 43% to 73% and 12% to 80% respectively. These increases were accompanied with a decrease in accuracy from 90% to 75%.

Logistic regression versus random forest:

Lastly, we compared the logistic regression model to the random forest model. Both models produced comparable results in terms of precision, recall and accuracy. Therefore, we concluded that one model does not perform better than the other.

Feature Importance

To find out whether some features are not important in prediction, we use coef_ attribute of the model and bar plot to visualize the importance of every feature. Since there are some negative values in the coefficient of the Logistic Regression model, we use Random Forest model's coefficients to prune the dataset.

Use percentile function to get the top 20% important features only (HighBP, HighChol, BMI, GenHlth, MentHlth, PhysHlth, Education and Income) and remove the rest of features from the dataset. Train the model again and get the prediction, the accuracy drops from 76% to 69%, the precision of minority class drops from 74% to 68% and the recall of it drops from 81% to 71%. Since the influences are so big, we could conclude that 8 left features are not enough for prediction. Smoker, stroke, diabetes, age and many other features should also be considered into risk factors.

Random Forest 80th percentile Features:

	precision	recall	f1-score	support
0.0	0.69	0.66	0.68	4789
1.0	0.67	0.71	0.69	4769
accuracy			0.68	9558
macro avg	0.68	0.68	0.68	9558
weighted avg	0.68	0.68	0.68	9558

Conclusion

- The importance of class imbalance. A big class imbalance could lead to misleading results such as 91% accuracy at the beginning. It will be poor in prediction of minority class which is really bad since our goal is aiming on minority class of HeartDiseaseorAttack feature.
- The importance of considering precision and recall ratio when evaluating models, the former one measures the accuracy of every label and latter one is the indicator of how good the model is at minimizing false negative prediction, which could avoid missing early treatment for heart disease.

- Logistic Regression model and Random Forest model yield comparable results in those indicators (precision, recall, f1-score, accuracy and so on).
- In feature importance analysis, based on the coefficient of Random Forest model, top 20% features are not enough for prediction. Smoker, diabetes and stroke should also be considered as strong risk factors.

Future Considerations

- Do random oversampling and compare the model report with random undersampling to get deeper understanding in class imbalance and its different solutions.
- Train more machine learning or deep learning models over our dataset to compare with Logistic Regression and Random Forest. There are several other popular algorithms like decision trees, support vector machines and so on.
- Find a dataset that could include more features that have been proved to have an effect on heart disease, such as family history.
- Find a dataset with better class balance so that no need to introduce oversample or undersample which may have an impact on the dataset.

Bibliography (APA citations)

1. Brownlee, J. (2020, August 20). *How to calculate feature importance with python*. MachineLearningMastery.com. Retrieved April 23, 2023, from https://machinelearningmastery.com/calculate-feature-importance-with-python/

2. Centers for Disease Control and Prevention. (2022, July 21). *About chronic diseases*. Centers for Disease Control and Prevention. Retrieved April 22, 2023, from https://www.cdc.gov/chronicdisease/about/index.htm

3. Centers for Disease Control and Prevention. (2022, October 14). *Heart disease facts*. Centers for Disease Control and Prevention. Retrieved April 22, 2023, from https://www.cdc.gov/heartdisease/facts.htm

4. Google. (2022, July 18). *Classification: Precision and recall.* Machine Learning. Retrieved April 22, 2023, from https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall

5. Smolic, H. (2022, September 16). *Precision versus recall - essential metrics in machine learning*. Graphite Note. Retrieved April 22, 2023, from https://graphite-note.com/precision-versus-recall-machine-learning