2022Spring STA160

Midterm Project

(Heart Disease Health Indicators)

Group member:

Yixin Xie (yxxie@ucdavis.edu)

Renyu Yang (yryyang@ucdavis.edu)

Zixuan Mi (leomi@ucdavis.edu)

Yihao Zheng (yhzheng@ucdavis.edu)

Zehao Li (azhli@ucdavis.edu)

Introduction

According to the CDC, heart disease refers to several types of heart conditions, and coronary artery disease, which affects the blood flow to the heart, is the most common type of heart disease in the United States. The shortness of blood flow to the heart caused by the coronary artery disease can further lead to a heart attack, leading the patient to a dangerous and even fatal situation. In fact, heart disease is the main cause of death for people in the United States: one person dies every 36 seconds in the United States from heart disease; about 659,000 people, 1 in every 4 deaths, in the US die from heart disease each year. Lifestyle is one of the major effects that cause heart disease. People who consume high carbohydrates, oil and salt are more likely to get heart disease. Studies have shown that high blood pressure, high blood pressure, and smoking are the leading causes for heart disease. There are also some chronic diseases like diabetes and stroke that are likely to cause heart disease. In this project, we aimed to classify individuals regarding whether an individual has a heart disease attack, using the information from a survey of 22 lifestyle or identity questions from the Behavioral Risk Factor Surveillance System (BRFSS). We will first study the characteristics of the data through in-depth exploration, then we would select the appropriate metrics and methods to be used based on this information. Eventually, we hope to gain profound insights into heart disease attack classification by comparing the performance of all candidate methods.

Dataset Description in General

In this dataset, we have 22 features in total. The feature “HeartDiseaseorAttack” is the response variable. There are 253680 observations. Thus, this is a dataset with dimension 253680*22. After examination, we found that the dataset does not contain any missing values. There are 14 binary categorical variables including the response variable. The other 8 variables are also categorical but have more than 2 levels.

Linear Correlation Examination

From figure 1 we can see that; majority of variables have weak or even no linear correlation with each other. However, there are few exceptions. There are several pairs that have moderate linear correlation. The variables “PhysHlth” and “GenHlth” have moderate linear correlation 0.52. The variables “DiffWalk” and “GenHlth” have moderate linear correlation 0.46. The variables “DiffWalk” and “PhysHlth” have moderate linear correlation 0.48. The variables “Education” and “Income” have moderate linear correlation 0.45. For the response variable “HeartDiseaseorAttack”, it has very weak linear correlation with all other variables. In conclusion, there is no strong linear correlation between any variables in this dataset.

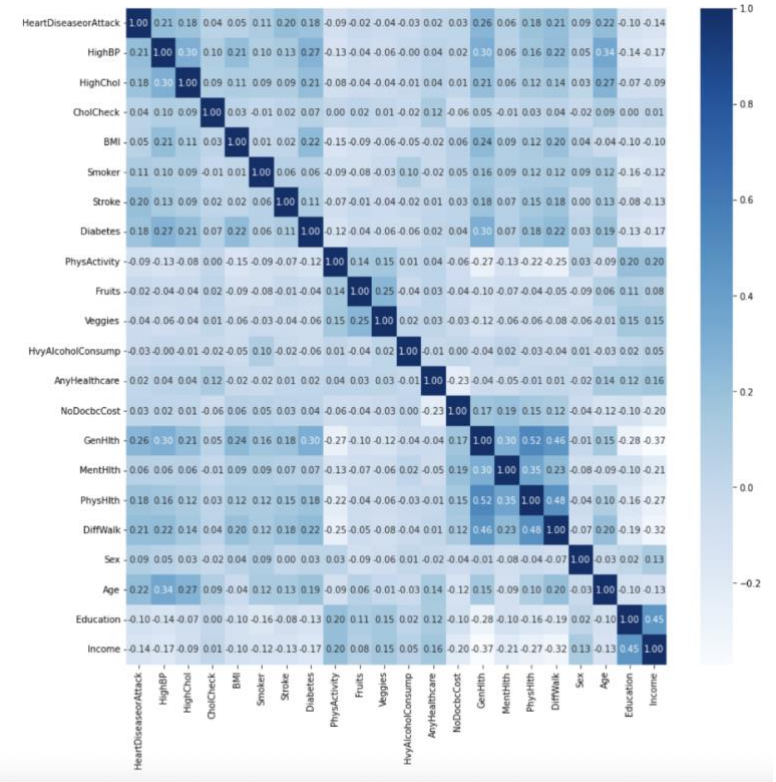

Figure 1.Correlation heatmap for full dataset

Feature Analysis and Selection

For the response variable, it is a binary categorical variable with level “0” and “1”. “0” represents the individual never reported having heart disease. “1” represents the

individual who has reported. After checking the distribution of the variable, we found that the dataset suffers from imbalance. The number of individuals who never reported is about ten times the amount of individuals who have reported.

We first analyze the binary categorical variables' importance through the use of odd ratio and stacked bar plot visualization.

For the variable "HighBP", it has two levels. The level "1" represents adults who have been diagnosed with high blood pressure by health professionals. The level "0" represents adults who haven't. From figure 2 we can see that the level "0" has 0.04 odd ratio and level "1" has 0.2 odd ratio. Level "1" is 5 times more than level "0". This tells us that high blood pressure is definitely more likely to cause heart related diseases. Thus, we regard the variable "HighBP" as important for predicting the response variable.

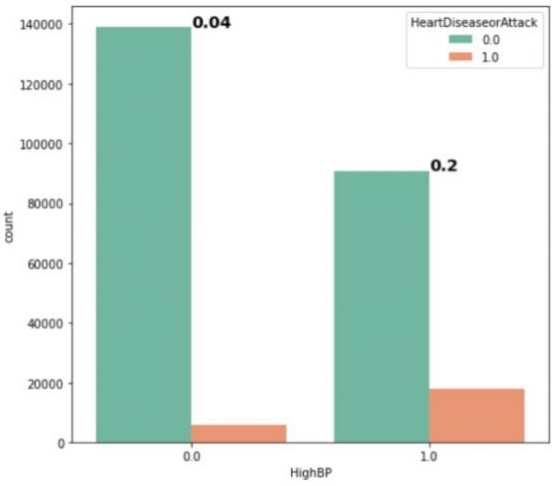

Figure 2.Odd ratio comparison for levels of variable "HighBP"

For the variable "HighChol", it has two levels. The level "1" represents adults who have been diagnosed with high blood cholesterol by health professionals. The level "0" represents adults who haven't. From figure 3 we can see that the level "0" has 0.05 odd ratio and level "1" has 0.18 odd ratio. Level "1" is 3.6 times higher than level "0". This tells us that high blood cholesterol is more likely to cause heart related diseases. Thus, we regard the variable "HighChol" as important for predicting the response variable.

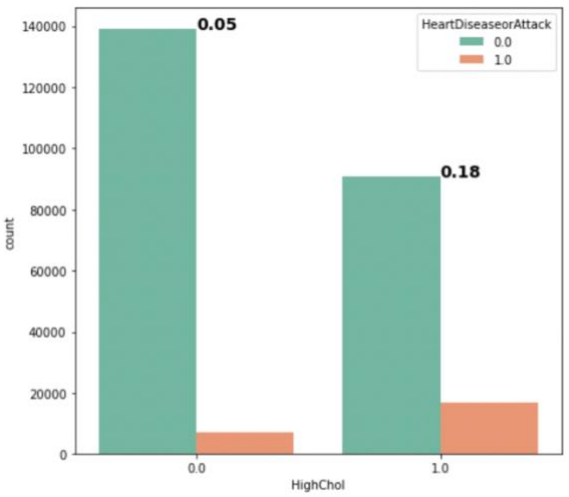

Figure 3.Odd ratio comparison for levels of variable "HighChol"

For the variable "CholCheck", level "0" means the individual did not check their cholesterol within five years. Level "1" represents the individual checked cholesterol level within five years. From figure 4 we can see that the level "0" has 0.03 odd ratio and level "1" has 0.11 odd ratio. Level "1" is about 3.7 times higher than level "0". This tells us that

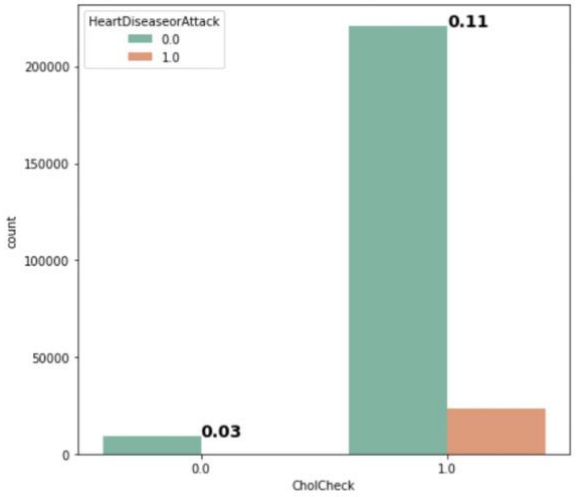

Figure 4.Odd ratio comparison for levels of variable "CholCheck"

individuals who checked their cholesterol within five years are more likely to have heart related diseases. However, the figure indicates that the amount of data is imbalanced. The number of individuals who are both labeled as Level "0" and suffering from heart related disease is relatively small. But we decided to regard this variable as important for predicting the response variable since it relates to the variable "HighChol" and the odd ratio between two levels of this variable is evident.

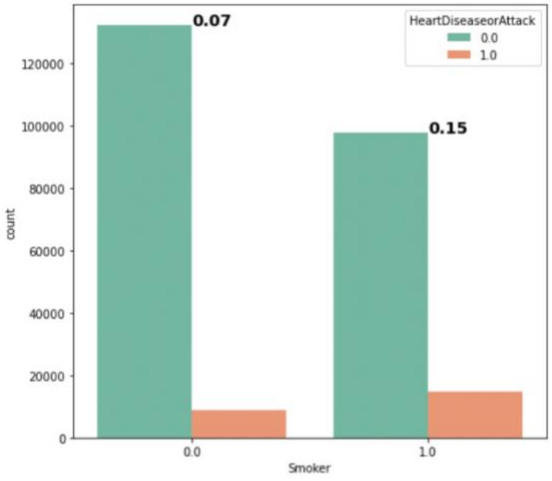

Figure 5.Odd ratio comparison for levels of variable "Smoker"

For the variable "Smoker", level "0" means the individual did not smoke at least 100 cigarettes for the entire life. Level "1" represents the individual who smoked at least 100 cigarettes. From figure 5 we can see that the level "0" has 0.07 odd ratio and level "1" has 0.15 odd ratio. Level "1" is about 2.1 times higher than level "0". This tells us that smokers are more likely to have heart related diseases. Thus, this variable is important.

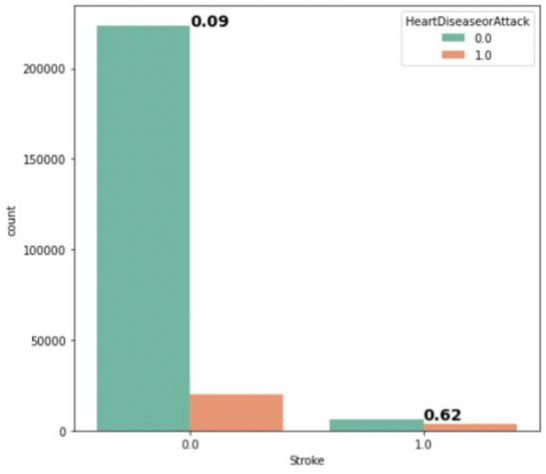

Figure 6.Odd ratio comparison for levels of variable "Stroke"

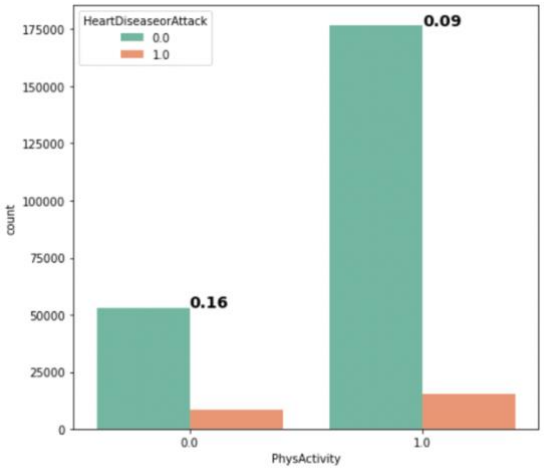

Figure 7.Odd ratio comparison for levels of variable "PhysActivity"

For the variable “Stroke”, level “1” means the individual had a stroke before. Level “0” represents the individual who never had a stroke. From figure 6 we can see that the level “0” has 0.09 odd ratio and level “1” has 0.62 odd ratio. Level “1” is about 6.9 times higher than level “0”. This tells us that individuals who experienced a stroke before are more likely to have heart related diseases. The figure also indicates that the amount of data for individuals belonging to level “1” is small compared with level “0”. However, the huge, odd ratio differences cannot be neglected. Thus, we decided to include this variable for prediction. For the variable “PhysActivity”, level “1” means the adult had reported doing physical activities. Level “0” represents adults who didn’t. From figure 7 we can see that the level “0” has 0.16 odd ratio and level “1” has 0.09 odd ratio. Level “0” is about 1.8 times higher than level “1”. This tells us that individuals who reported doing physical activities are less likely to have heart related diseases. Thus, we think this variable is important. For the variable “Fruits”, level “1” means the individual consumed fruits for one or more times per day. Level “0” represents the individual who didn’t. From figure 8 we can see that the level “0” has a 0.11 odd ratio and level “1” has a 0.1 odd ratio. Level “0” is about the same as level “1”. Since this is a large dataset, and the similar odd ratio indicates that individuals who consumed fruits one or more times per day may have little influence in terms of predicting the response variable. Thus, we think this variable is not important. We will drop this variable when fitting machine learning models.

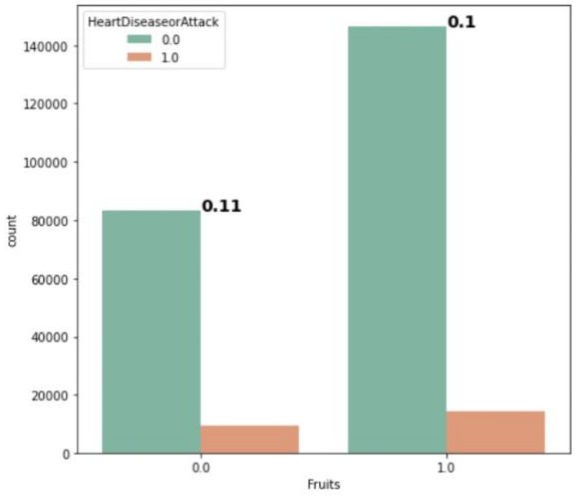

Figure 8.Odd ratio comparison for levels of variable “Fruits”

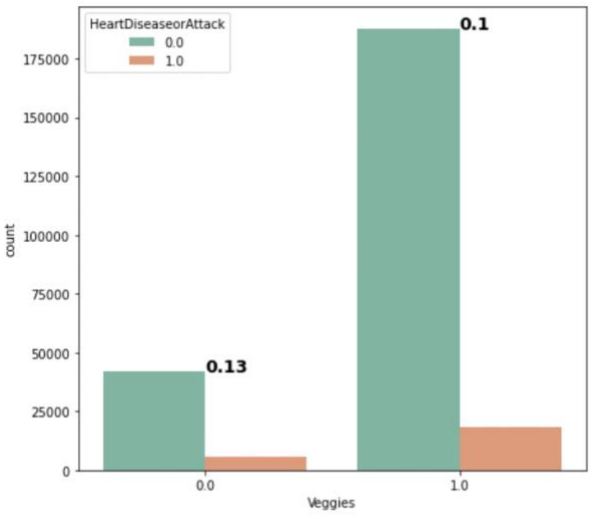

Figure 9.Odd ratio comparison for levels of variable “Veggies”

For the variable “Veggies”, level “1” means the individual consumed vegetables one or more times per day. Level “0” represents the individual who didn’t. From

figure 9 we can see that the level “0” has a 0.13 odd ratio and level “1” has a 0.1 odd ratio. Level “0” is about 1.3 times higher than level “1”. Since this is a large dataset and the odd ratio difference is not evident, it indicates that individuals who consumed vegetables one or more times per day may have little influence in terms of predicting the response variable. Thus, we think this variable is not important and it will be dropped.

For the variable “HvyAlcoholConsump”, level “1” means the individual is regarded as a heavy drinker. Level “0” represents the individual who is not. From figure 10 we can see that the level “0” has a 0.11 odd ratio and level “1” has a 0.06 odd ratio. The odd ratio of level “0” is about 1.8 times higher than level “1”, however, the figure indicates that the amount of individual who belonging to level “1” is very small compared to level “0” and the amount of individuals who both belonging to level “1” and have heart related diseases are extremely small. Therefore, the imbalance of the dataset makes this variable less important. We decided to drop this variable.

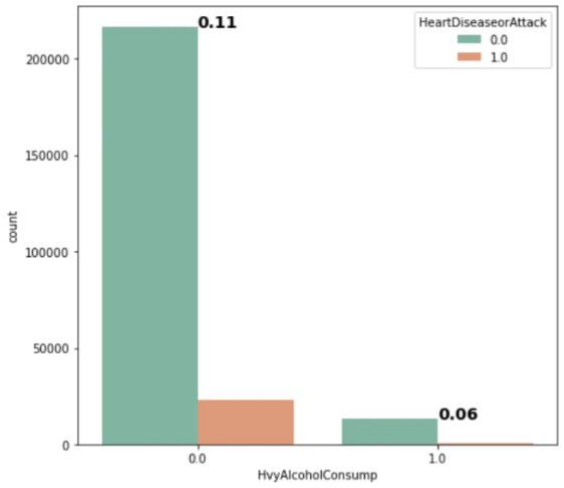

Figure 10.Odd ratio comparison for levels of variable “HvyAlcoholConsump”

For the variable “AnyHealthcare”, level “1” means the individual has any kind of healthcare plan. Level “0” represents the individual who doesn’t. From figure 11 we can see that the level “0” has a 0.08 odd ratio and level “1” has a 0.11 odd ratio. The odd ratio of level “1” is about 1.4 times higher than level “0”, however, the figure indicates that the amount of individual who belonging to level “0” is very small compared to level “1” and the amount of individuals who both belonging to level “0” and have heart related diseases are extremely small. Therefore, the imbalance of the dataset makes this variable less important. We decided to drop this variable.

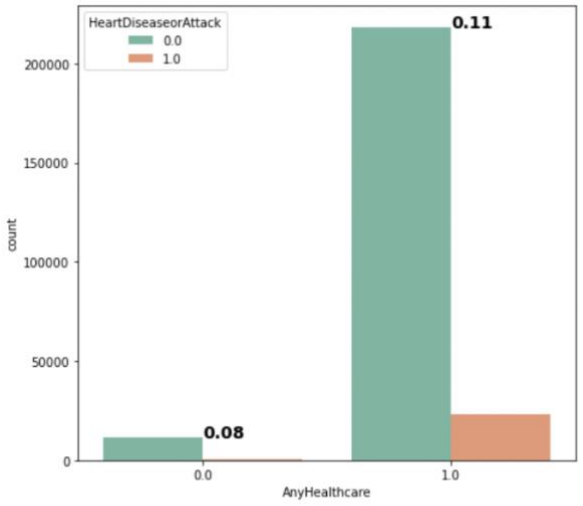

Figure 11.Odd ratio comparison for levels of variable “AnyHealthcare”

For the variable “NoDocbcCost”, level “1” means the individual had the need to see a doctor but didn’t because of the cost within 12 months. Level “0” represents the individual who didn’t have this situation. From figure 12 we can see that the level “0” has a 0.1 odd ratio and level “1” has a 0.14 odd ratio. The odd ratio of level “1” is about 1.4 times higher than level “0”. The small amount of data in level “1” makes us decide to drop this variable. In addition, the odd ratio difference is not that high to make the variable seem important. For the variable “DiffWalk”, the level “1” represents individuals who have difficulties walking or climbing stairs. The level “0” means individuals who didn’t have this issue. From figure 13 we can see that the level “0” has 0.07 odd ratio and level “1” has 0.3 odd ratio. Level “1” is about 4.3 times higher than level “0”. This tells us that having those difficulties is more likely to cause heart related diseases. Thus, we regard the variable “DiffWalk” as important for predicting the response variable.

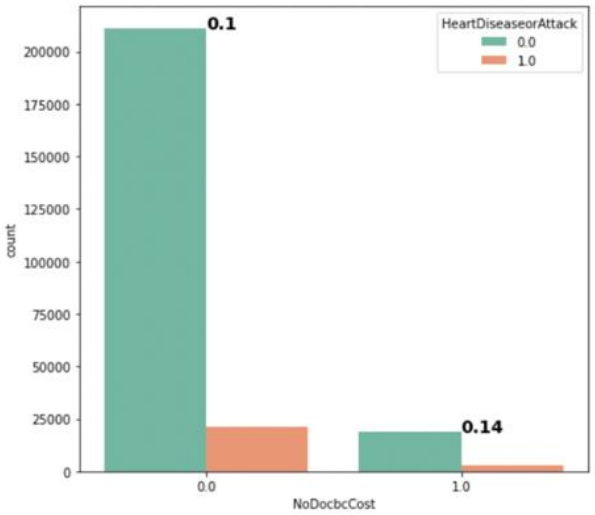

Figure 12.Odd ratio comparison for levels of variable “NoDocbcCost”

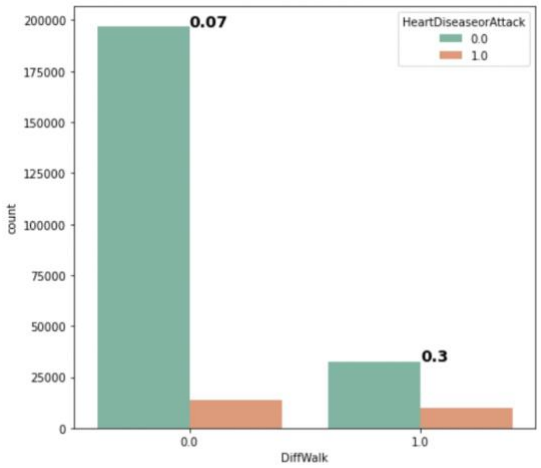

Figure 13.Odd ratio comparison for levels of variable “DiffWalk”

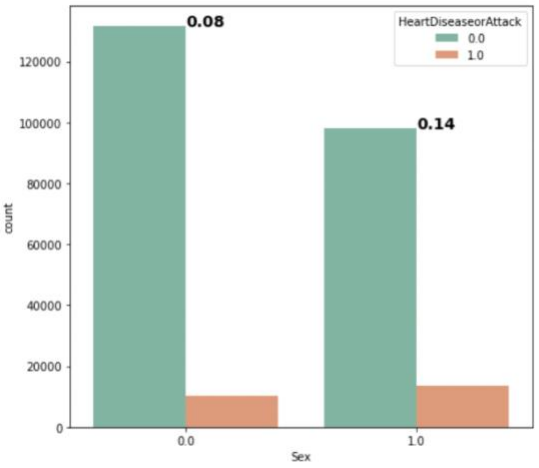

Figure 14.Odd ratio comparison for levels of variable “sex”

For the variable “sex”, the level “0” represents individuals who are female. The level “1” means individuals who are male. From figure 14 we can see that the level “0” has 0.08 odd ratio and level “1” has 0.14 odd ratio. Level “1” is about 1.8 times higher than level “0”. This tells us that male are more likely to have heart related diseases than females. Thus, we regard the variable “sex” as important for predicting the response variable.

After analyzing the binary categorical variables' importance, we then focused on other variables. For those variables who have more than two categories, we decided to still use stacked barplot and odds ratio to analyze the importance. However, some of the variables needed feature engineering to further explore their importance. To choose the best feature engineering methods, we also used catplot to see the distribution of data for those variables.

For the variable "Diabete", there are 3 levels. Level "0" represents the individual who does not have diabetes. Level "1" and level "2" indicates the severity of diabetes. From figure 15 we find that the odd ratio increases as the level increases (the line indicates the change of odd ratio). Even though the amount of data in level 1 is relatively small, the increased amount of odd ratio is evident for each level. Therefore, we conclude that diabetes is a good explanatory variable for predicting the response variable.

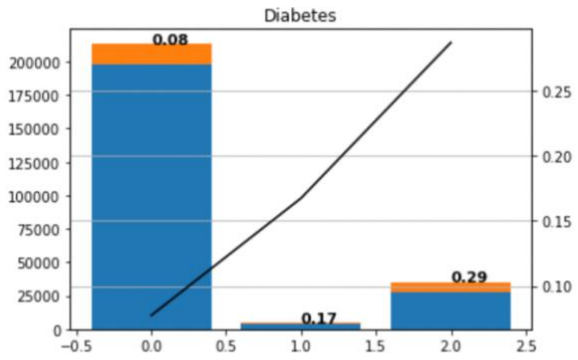

Figure 15. Odd ratio comparison for variable "Diabetes"

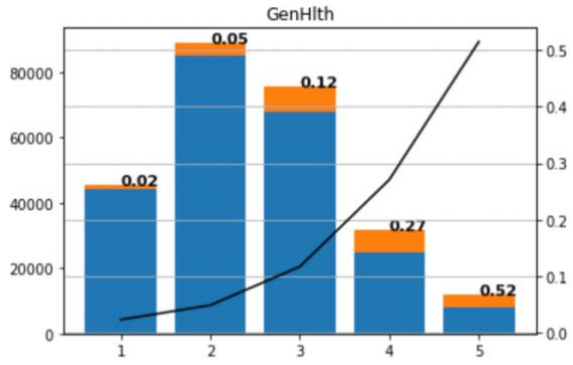

Figure 16. Odd ratio comparison for variable "GenHlth"

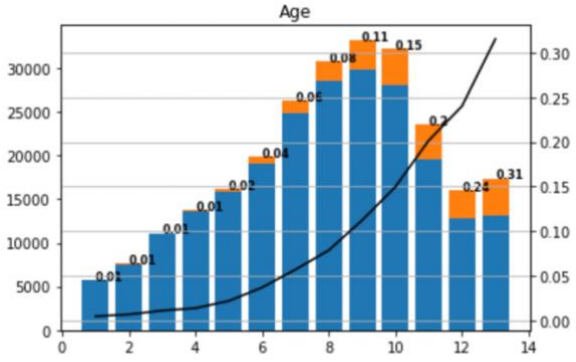

Figure 17. Odd ratio comparison for variable "Age"

For the variable "GenHlth", there are 5 levels. Each individual rates their general health condition from level "0" to level "4". As the level increases, individuals think their general health condition gets worse. From figure 16 we can see that the odd ratio increases as the level number increases and there is enough data for each level of this

variable. So, this shows that there are higher possibilities for the individual to have heart related diseases as the level of “GenHlth” increases. Thus, the variable “GenHlth " will be included. For the variable “age”, there are 13 levels. Each level represents an age group. As the level increases, individuals belonging to that group are older than the previous group. From figure 17 we can see that the odd ratio increases as the level number increases and there is enough data for each level of this variable. So, this shows that there are higher possibilities for the individual to have heart related diseases as age increases. Thus, the variable “age” is important for predicting the response variable.

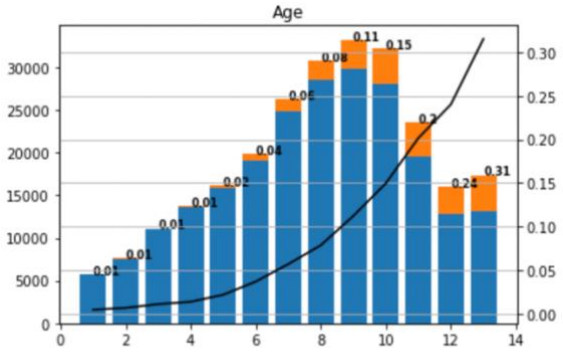

Figure 17. Odd ratio comparison for variable “Age”

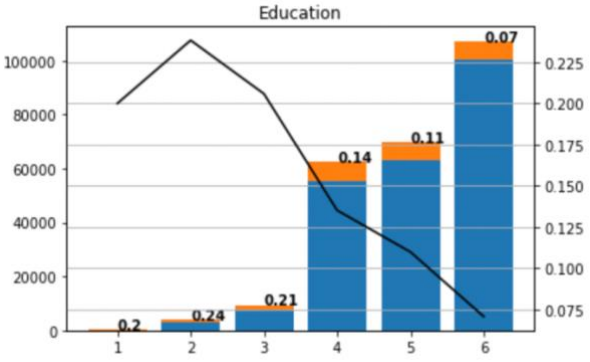

Figure 18. Odd ratio comparison for variable “Education”

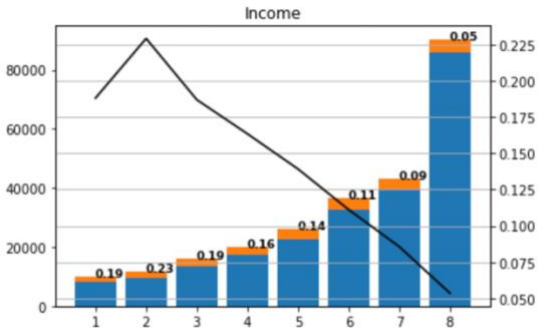

Figure 19. Odd ratio comparison for variable “Income”

For variables “Education” and “Income”, they are similar in many ways. As the level increases, individuals' education level and amount of money they earn per year increase. From figure 18 and figure 19 we can see that the overall trend for odd ratio is decreasing for both variables as variables’ level increases. This indicates that higher education level and income level may lead to lower possibility of the individual having heart related disease. For variables “MentHlth” and “PhysHlth”, both variables have 31 levels. The level number represents the number of days individuals did not feel good about their physical and mental health conditions. Both variables suffer from insufficiency and imbalance of data. Thus, we performed feature engineering to reduce the number of levels for both variables. From figure 22 and figure 23 we can see that most data is concentrated from level 0 to level 5 and level 30 for both variables. Level 0 has the most amount of data. Thus, we decided to reduce the number of levels from 31 to 4 and assign each new level with enough data according to the

distribution. Figure 20 and figure 21 indicate the change of odd ratio after the feature engineering. The odd ratio increases as the level increases for both variables. Thus, we consider both variables as important.

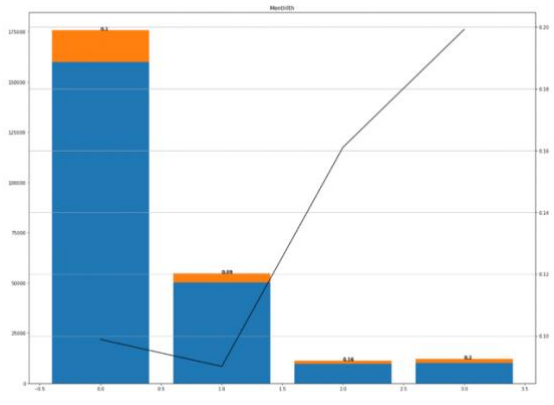

Figure 20. Odd ratio comparison for variable "MentHlth"

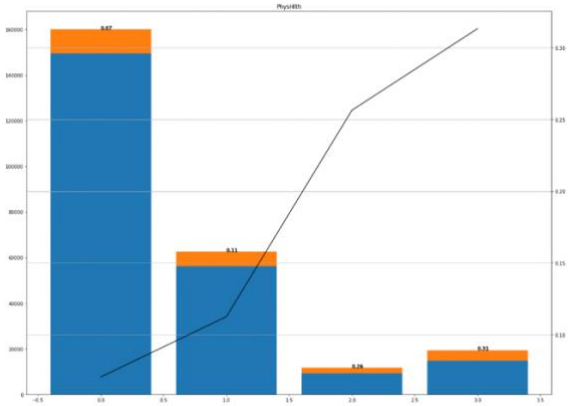

Figure 21. Odd ratio comparison for variable "PhysHlth"

For the variable "BMI", it has 84 levels, and the level number ranges from 12 to 98. Each level number represents the Body Mass Index for the individual. However, the majority of levels suffer from data insufficiency. Therefore, we performed feature engineering. From figure 24 we can see that most data concentrated from level 15 to 45. After careful analysis, we decided to reduce the number of levels from 84 to 4. Figure 25 shows that the odd ratio increases as the level increases after the feature engineering. Thus, this variable will be included for prediction.

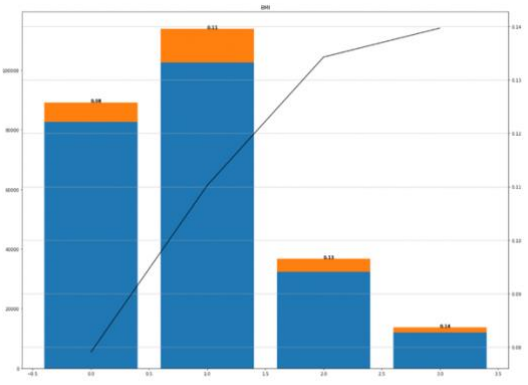

Figure 25. Odd ratio comparison for variable "BMI"

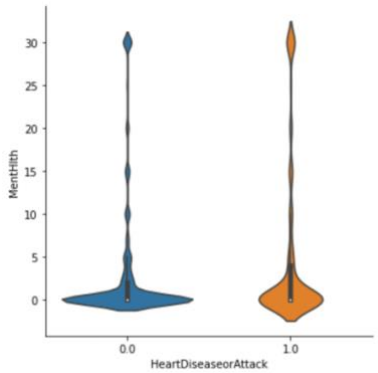

Figure 22. Data distribution for variable "MentHlth"

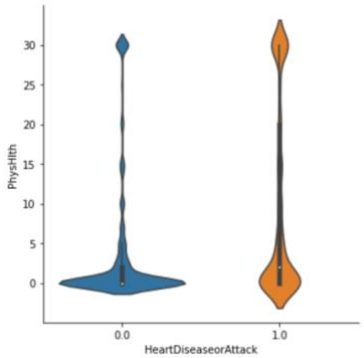

Figure 23. Data distribution for variable "PhysHlth"

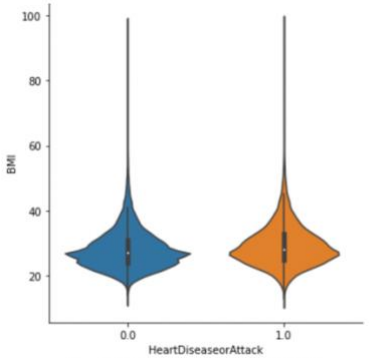

Figure 24. Data distribution for variable "BMI"

In conclusion, we decided to drop variables "Fruits", "Veggies", "HvyAlcoholConsump", "NoDocbcCost", and "AnyHealthcare". Those variables will not be included during the prediction.

Logistic Regression Model Assumption Examination

First, the binary logistic regression model requires the dependent variable to be binary. In our dataset, the response variable “HeartDiseaseorAttack” is a binary variable with levels “0” and “1”. Second, logistic regression requires observations to be independent of each other. From our dataset we knew that each observation is an individual’s response toward questions the researcher provided. They cannot influence other people’s responses. And those observations did not come from repeated measurements. Every observation represents a unique individual. Third, logistic regression requires independent variables to not highly correlate with each other. From the correlation heatmap we can see that there are no pairs that have strong linear association, and only few have moderate linear association. In addition, logistic regression requires large sample sizes. In our case, we have 253680 observations which is enough data for fitting a logistic regression model according to the general guideline.

Machine Learning Matrix Selection

In general, there are 4 most popular metrics regarding the performance of machine learning classification algorithms, there are accuracy, precision, recall and F1 score. For this project, we will be heavily focusing on the optimization of the F1 score. The reasoning is as follows. First, from the visualization figures, the data display a severe imbalance between the two categories of heart disease or attack and not having heart disease, for which the ratio is close to 1:10. This is likely linked to the reality that heart attack is a small portion of the whole population, and people that have experienced heart attack are a minority group in the sampling population. This observation also confirms that our sample is randomly selected. Therefore, the accuracy no longer serves as a reliable standard for measuring classification performance, for which a classifier that only classifies observations into 0 (no attacks) would still achieve an accuracy score above 90. Second, from a reality perspective, we tend to emphasize FN (false negative) more than FP (false positive) in this scenario, which is because underestimating the likelihood of heart attack could potentially produce more harmful effects to patients compared to overestimating it. Therefore, we would like to punish FN (false

negative) while rewarding TP (True positive). The recall score might seem to be a viable option because it rewards TP while punishing FN; yet in practice, we found that it leads to extreme results during parameter tuning. For example, in the case of weighted logistic regression, the gridsearch result shows that the optimized weight for 1 (heart attack) is 1, meaning that it suggests removing all 0 data to maximize recall (See figure 1a). Thus, it is problematic to optimize the parameters using recall as the main metric. On the other hand, the F1 score is a harmonic mean of recall and precision. Noticeably, it punishes both FN and FP while maximizing TP. As a result, F1 score finds a balance between controlling FN and FP, which makes it the most appropriate metric to be used in this project.

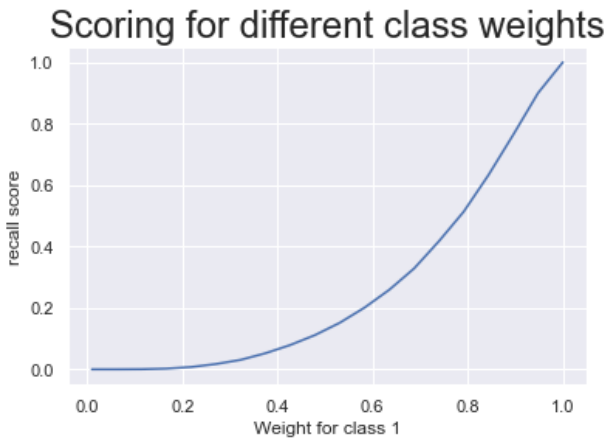

Figure 1a

Algorithms

To resolve the extreme imbalance problem, we could do classification from three approaches: Reweighting observations, resampling observation, and random forest.

Weighted Logistic Regression

The logistic regression classifies samples using ordinary least squares, we assume the following model:

$$P(Y = y \mid X = x) = \frac{1}{1 + exp^{-y(x^T \beta)}}$$

And its log form is:

$$log(P(Y = y \mid X = x)) = log(1 + exp^{-y(x^T \beta)})$$

To optimize the classification, we minimize the log loss:

$$-\frac{1}{N}\sum_{i=1}^{N} y_i log(p(y_i)) + (1-y_i) log(1-p(y_i))$$

For which y_i is the actual class, while $p(y_i)$ is the probability of classifying observations to 1. When y_i=1, the penalty is $-logp$, and the penalty is $log(1-p)$ when y_i= 0. As previously mentioned, the data is severely imbalanced, and we have more samples that have $y_i = 0$. Therefore, the log loss would heavily emphasize on minimizing FP (false positive) to reduce the total loss. To resolve this problem, we need to penalize FN (false negative) more to balance the learning. To be more specific, we could assign more weight on the penalty when $y_i = 1$. And the formula becomes:

$$-\frac{1}{N}\sum_{i=1}^{N} w_0(y_i log(p(y_i))) + w_1((1-y_i) log(1-p(y_i)))$$

w_0 is weight for class 0, and w_1 is the weight for class 1. In practice, any weight from 0 to 1 could possibly improve the performance, we can assign weight by grid searching the best weight distribution within the [0,1] interval by checking cross validation scores of each trial. The metrics for grid searching weight is the F1 score. Since there is only one parameter "weight" to be tuned in the weighted logistic regression model, the running time is significantly less than some other popular learning methods, such as the random forest. Eventually the gridsearch indicates that the optimal weight is w_0= 0.208, w_1= 0.791. As shown in figure 2a, the cross validation F1 score performance is an increasing convex curve from w_1 = 0 to w_1=0.6; it turns concave after 0.6 and reaches maximum at 0.791. The result also agrees with our intuition of penalizing more on FN predictions since $w_1 > w_0$. Finally, we could use this new model to fit and classify the test set.

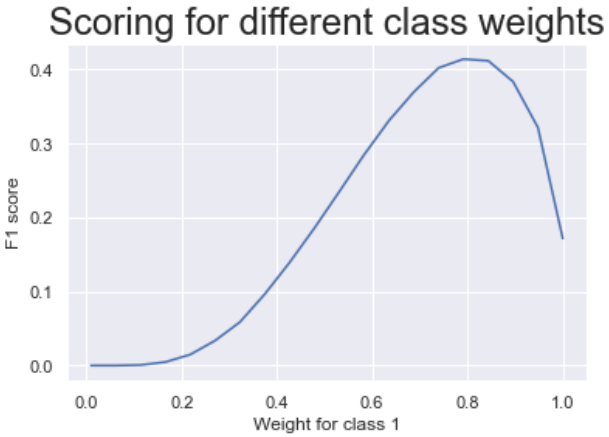

Figure 2a

Resampling Observation

To deal with the imbalance dataset, the resampling technique provides us with two options: Undersampling and oversampling. For undersampling, we start with separating the train set and test set. We only resample the training set and leave the test set unchanged. The idea of undersampling is to sample from the majority class to get the same number of observations as the minority class. We won't have repeated observations if we choose to sample without replacement. Undersampling will bring us a perfectly balanced dataset. In our heart disease dataset, the dataset is reduced to a very small dimension with undersampling technique. This leads to significant improvement in training time. However, we have to be aware of the fact that a large portion of the dataset is being removed by undersampling. The reduced dataset might not be able to represent the original dataset. We got 0.37 for our F1 score. F1 score is a combination of precision and recall. Precision is defined as the probability of true positive results in prediction among all observations that are classified to be positive. Recall is defined as the probability of true positive results in prediction among the whole population of people having heart disease in the sample space. Accuracy is defined as maximizing the probability of making correct classification out of the whole population in the sample space. By comparing to the original dataset, we observe a tradeoff between accuracy and F1 score. We sacrifice overall accuracy to improve on correctly identifying people with heart disease attacks, while controlling the total number of people being classified wrongly.

For Oversampling, we are doing the opposite of undersampling. We resample the minority class to have the same number of observations as the majority class. For the heart disease dataset, we need to increase the size of the minority class by ten times to create a balanced dataset. In this case, we must sample with replacement. Our duplicate observations will influence prediction accuracy and slow down the process of training our prediction model. Besides using the basic oversampling function, we also tried Synthetic Minority Oversampling Technique (SMOTE). The idea of SMOTE is to oversample the minority class. The number of neighbors is user-determined. For a specific observation, a random neighbor is selected. By forming a line between the two, a new point is selected to our oversampled observation. We keep repeating this process to get enough samples for the minority class. With this process, our oversampled observations are close to the actual observations. We

apply gridsearch to get the optimal number of neighbors that would lead to the highest F1 score. We restricted our range to be less than or equal to ten because the number of neighbors larger than ten will not likely improve our model. With a higher number of neighbors, we are more likely to draw a line with two observations that are far from each other and obtain a new observation far from the original dataset. This will increase the bias of our model. After conducting the grid search parameter tuning, we observe that k-neighbors equal to five will bring us the best performance.

Surprisingly, we observe similar results compared to the undersampling technique. In terms of performance, we conclude that oversampling and undersampling are quite similar. However, in terms of efficiency, we would prefer undersampling over oversampling. Dealing with a smaller size of dataset saves us time and memory.

Random Forest Method Observation

Random forest classifier is also a powerful machine learning method to classify our data into two groups: having a heart disease or heart attack and having neither. The basic idea of this method is to ensemble many decision trees that split our data based on different features. Each decision tree results in a class prediction and the class with the most observations is the prediction of our model. Random forest method, as a combination of many relatively uncorrelated decision trees, outperforms any individual decision tree. To optimize our prediction performance, we need to apply a hyperparameter tuning process. Hyperparameters are settings of our random forest model, including the number of decision trees and the number of features in each tree while splitting a node. The hyperparameters are consist of the following: number of trees in the forest, the max number of features considered for splitting a node, the max number of levels in each decision tree, the minimum number of data points placed in a node before the node is split, the minimum number of data points that are allowed in a leaf node, and the method for sampling data points (to decide whether with or without replacement).

In our study, we apply randomized grid search to find our optimal hyperparameters. The idea of this method is to randomly select samples from a wide range of hyperparameters and compare the performance of the model with the selected combination of hyperparameters and result in the combination that outperforms the rest. The most important settings of the randomized grid search are “n_iter”, the number of different combinations (random sample)

to try, and “cv,” the number of folds for cross validation. “cv” stands for cross validation, which is a powerful method to help us avoid overfitting of our train set. Although we would obtain higher performance with greater numbers of “n_iter” and “cv”, we need to balance the performance and computation cost. Due to the limit of our computational ability, we set “n_iter” to be 100 and “cv” to be 3. In most conditions, we use accuracy as the performance of hyperparameters. However, we have unbalanced data. Thus, we will use F1 score as our scoring method for tuning hyperparameters.

Results and Interpretation

method	F1 score	Accuracy
Simple logistic	**0.197**	**0.907**
Weighted logistic	**0.418**	**0.86**
Undersampling logistic	**0.3779**	**0.7497**
Oversampling logistic	**0.3773**	**0.7488**
Smote oversampling logistic	**0.378**	**0.7525**
Random Forest	**0.2102**	**0.8993**

After implementation of different approaches of classification with imbalance data, we fit the model on the test data set and collect F1 score and accuracy performance of each method (the test set is a fraction that is randomly sampled from the data, controlled by the same seed in each trial). From the result table above, we can observe that re-weighted and re-sampling methods display a tradeoff between F1 score and accuracy, for which they improve F1 score significantly while having a decrease in accuracy. On the other hand, random forest regression reveals a similar performance compared to the simple logistic regression.

Regarding Weighted Logistic Regression

The weighted logistic regression boosts up the F1 score the most out of all variations of the logistic regression models. Compared to the simple model, it raises the F1 score by 0.21 while only having 0.04 decrease in accuracy, which implies that it sacrifices minor performance on accuracy to trade for significant improvements over correctly identifying people with true positive results while controlling the overall false prediction. In addition, weighted logistic regression also displays efficiency in running time compared to SMOTE and random forest, which is due to its simple parameter tuning process. Using F1 score as a main metric, the weighted logistic regression is the most economic method on classifying people with heart disease attack out of the models we selected.

Regarding Re-sampled Logistic Regression

The resampled logistic regression techniques have the second-best performance regarding all variations of logistic regression models. Compared to the simple model, it raises the F1 score by 0.181 while having 0.1545 drop in accuracy. This is a fair amount of trade between correctly identifying people with true positives while having a higher chance of making false predictions. These three methods share similar performance regarding F1 score and accuracy. Therefore, we select the undersampling logistic regression method because it requires the least amount of memory and runtime. Compared to weighted logistic regression, undersampling logistic regression has slightly lower accuracy and F1 score. However, in terms of runtime efficiency, undersampling has the best performance among all of the logistic models.

Regarding Random Forest

The random forest method only outperforms the simple logistic regression based on the F1 score, raising 0.0132 F1 score in comparison to the simple logistic model. However, it has the second-best performance in accuracy. The performance result of the random forest method is not consistent with our expectation. The reason for this observation may be caused by the insufficient value of “n_iter,” the number of different combinations of hyperparameters to select. Although random forest is one of the best and most powerful methods for

classification, it consumes huge computational power during the process of tuning hyperparameters.

Conclusion

In this project, we first conducted detailed interpretation of the characteristics of the dataset, which guided us to our feature engineering and metrics selection decisions. Then, we selected three approaches aiming to resolve the imbalance issue, which are re-weighting, re-sampling, and random forest classification. We discovered that each method has an advantage in some respects. Although the prediction results demonstrate the tradeoff between accuracy and F1 score, the weighted logistic regression stood out to be the most economical option of maximizing the F1 score while maintaining the overall accuracy.

However, there also exists some limitations regarding this report. First, most methods are variations of the logistic regression methods, thus we reach the outcome relying on the assumption that variables are independent of each other. Despite the fact that we show there are no linear correlations, in reality, the variables could be related in manners other than linear combination. Secondly, we are also limited to our computational power. In the random forest model, the large set of possible combinations of hyperparameter choices make it problematic to find an optimal solution for F1 score: the computation ended early before convergence due to the limitation of max iteration. Thus, if we have more time to continue our study, we will increase the number of iterations for selecting possible combinations of hyperparameters and increase the number of folds for cross validation to increase our random forest model performance.

Citation

Binary Cross Entropy aka Log Loss-The cost function used in Logistic Regression. (2020, November 9). Megha270396. https://www.analyticsvidhya.com/blog/2020/11/binary-cross-entropy-aka-log-loss-the-cost-function-used-in-logistic-regression/

Kamaldeep Singh. (2020, October 6). How to Improve Class Imbalance using Class Weights in Machine Learning. https://www.analyticsvidhya.com/blog/2020/10/improve-class-imbalance-class-weights/

SMOTE — Version 0.9.0. SMOTE. From https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html

Yiu, T. (2021, December 10). Understanding Random Forest - Towards Data Science. Medium. https://towardsdatascience.com/understanding-random-forest-58381e0602d2

Koehrsen, Will. “Hyperparameter Tuning the Random Forest in Python,” Towards Data Science, https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74.