Detection of Heart Disease using personal key indicators

Jonah Rockey, Alankrit Gupta, Bala Akhil Rajdeep Battula, Rathnapriya Gopalakrishnan

Indiana University-Purdue University, Indianapolis, IN 46202, USA

jorockey@iu.edu, alangupt@iu.edu, bbattula@iu.edu, rgopala@iu.edu

1 Introduction

Heart disease is currently the leading cause of death in the United States today. According to the Center for Disease Control (CDC), around 659,000 people die from heart related diseases every year in America, which is one in four deaths overall (CDC, 2021). From this number alone we can see that this is a major problem causing the majority of deaths. Medically, heart disease arises when a layer of plaque blocks the arteries or blood vessels connected to the heart. This congests the arteries and does not allow the necessary nutrients and oxygen to reach the heart (Roth, 2018). Furthermore, there are many factors that make an individual more likely to suffer from heart disease. Some major risk factors include high blood pressure, smoking, obesity, and physical inactivity. While heart disease is very dangerous, many of the risk factors can be prevented with actions such as exercising and maintaining a healthy diet. That is why it is important to be able to predict possible heart disease when it is still preventable. This leads us to the problem in our project.

1.2 Problem Statement

As we have seen, heart disease is a dangerous problem in the United States today, so it is extremely useful to be able to predict future heart disease based on an individual's traits. Our group set out to do this given a data set containing pertinent information on individuals including whether they have had a heart disease. By using various data analytics techniques, we were able to construct models to attempt this prediction.

1.3 Data

To analyze this problem, we utilized a data set found on Kaggle titled "Personal Key Indicators of Heart Disease" (https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease). This data was collected by the CDC as a part of the Behavioral Risk Factor Surveillance System (BRFSS). This is a large system that conducts telephone surveys of adults in the United States, and it is one of the most extensive health surveys in the country with about 400,000 surveys every year. For this project, we are using data collected from the 2020 survey. The original CDC dataset has about 400,000 entries and more

than 300 columns containing survey questions on different demographic and health topics. This data was then reduced to approximately 320,000 entries and 18 columns by the creator of the Kaggle dataset. This was done to include only the data that is relevant to heart disease.

2. Methodology

2.1 Data Cleaning

Initially, we followed the cleaning of data using columns of the dataframe after collecting from the dataset. We observed and found that there are no missing or null values to be removed.

2.2 Data Exploration

We organized the data based on the structure using exploratory analysis. Different types of data can be seen from the explanation of dataset variables. The summary of the data shown gives us the description of the common attributes. This tells us that there are 18 columns with 319795 values.

2.3 Feature Engineering

Features are distributed as continuous and categorical based on the class that is numeric or character leading us to categorize the data and store them with respective levels using factors. The vectors of type string and integer for the unique values here are converted into numeric with levels “Yes” or “No” having its labels as 1 and 0. We clearly see that the variables AgeCategory has factors with 13 levels, Race with 6 levels, GenHealth with 5 levels, Diabetic with 4 levels and all the other variables consist of factors with 2 levels. The correlation values of the numeric variables helps us to visualize the data further to predict the closeness of the truth value.

2.4 Data Visualization

The data is explored through various types of graphs and charts such as boxplot, histogram, stacked bar charts etc. The ggplot library is used for the graphic mapping which gives us aesthetically pleasing plots. The correlation plot in Figure 1, shows us that there are no strong correlations among the continuous variables. From the box plot we identified that those with heart disease experienced physical injuries/ illness for a higher number of days than those without heart disease. From the histograms, we identified that many did not report any physical injuries or bad mental health. From the stacked bar charts, we found that heart disease is lower among those without risk factors.

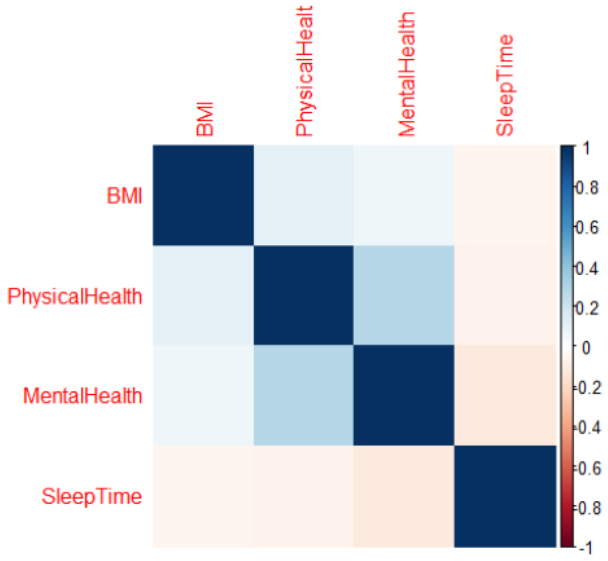

Figure 1: Correlation between the risk identification factors

2.5 Sampling

Due to the imbalanced dataset we use various sampling methods by considering the unequal distribution with the dependent variable. To improve the overall classification performance we use undersampling and oversampling to treat the imbalanced data. We use the ROSE (Random Over Sampling Examples) package as it is a bootstrap technique by random selection of the data to deal with the binary classification problems in our dataset. We also explored smote methodology for some model builds as it was easily integrated into the pipeline.

2.5.1 Under Sampling

Through under-sampling, we eliminate a subset of the data from the majority class to match its proportion with the minority class, to get a balanced dataset. However, this process could lead to loss of important information of the training data relevant to the majority class.

2.5.2 Over Sampling

The proportion of data in the minority class is increased to match the size of the majority class to get a balanced dataset. Unlike under-sampling, there is no loss of information in this method. We used ovum.sample() function provided by the ROSE package to obtain these samples.

2.5.3 Synthetic Minority Over-Sampling Technique (SMOTE)

The SMOTE algorithm generates artificial data instead of replicating and adding more observations from the minority class . It overcomes the imbalances in the data with respect to synthetic data generation with

the help of the ROSE package based on similarities of feature vector sample space that focus the classifier learning bias towards the minority class.

2.6 Machine Learning

We built machine learning classification models such as Logistic Regression, Random Forest, Decision Tree, KNN and SVM. We split the under-sampled dataset into 80% training and 20% test data. We fitted the model on the training data and observed its performance on test data. We calculated the performance metrics compared to these models. Our target variable is the HeartDisease variable which tells whether a person has a heart disease or not. Our independent variables are our risk factors.

2.6.1 Logistic Regression

In Logistic Regression, we used the glm function to fit the training data. We evaluated its performance on the test data. We calculated the sensitivity, specificity, accuracy, and test error values that are shown in results with the help of the confusion matrix in Table 1 and we also plotted ROC curves.

2.6.2 Random Forest

For Random Forest, we used cross validation with the number of folds being 3, to identify the best tuning parameters. We found that the best value for mtry to be 2, mtry is the number of features considered at each split point. We can clearly see that the ROC curve in Figure 2 is like that obtained by the result of the logistic regression model.

2.6.3 Decision Tree

We see that AgeCategory and GenHealth variables help in construction of the classification tree with the number of terminal nodes as 4 as shown in Figure 3. The plot of the ROC area curve clearly shows that it is not smooth.

2.6.4 K-Nearest Neighbors

In KNN, we found that the k value with 9 is the best optimal value, which uses 17 predictors rather than k having values 5 and 7. The performance of this model can be seen in Table 2.

2.6.5 SVM

We have also constructed an SVM classification model and evaluated its performance.

2.7 Performance Evaluation

The evaluation metrics for all the models have been plotted in Figure 2. The inferences are made in the results section.

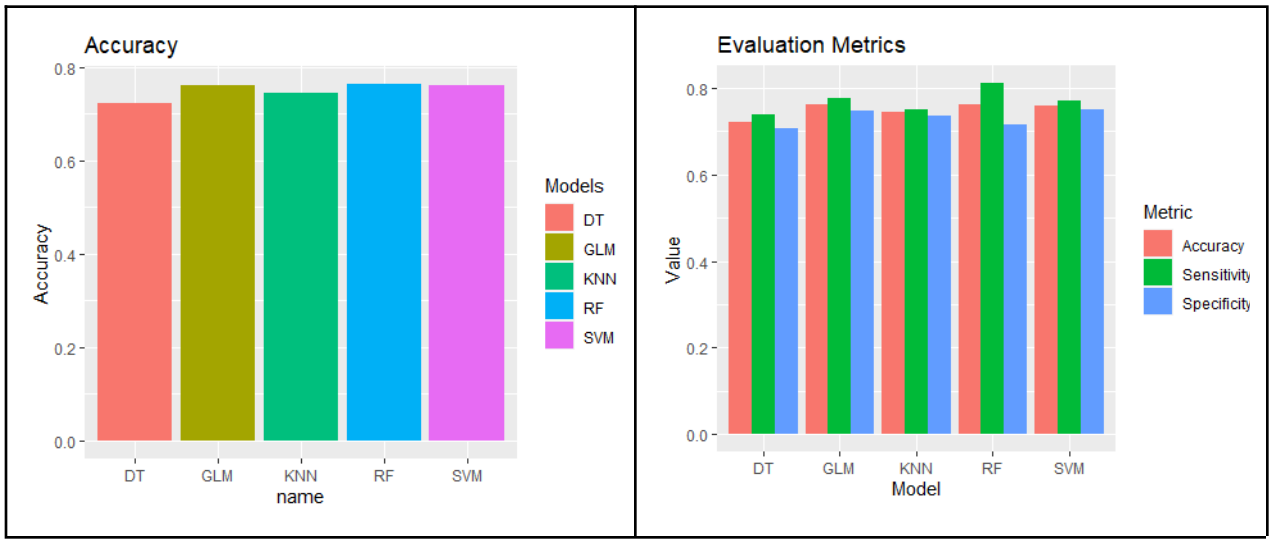

Figure 2: Accuracy, Sensitivity and Specificity of the models

3. Results

After the process of data wrangling, visualization, and analysis, we obtained a logistic regression model, a random forest model, a k-nearest neighbors model, and a decision tree model for predicting the risk of heart disease. In this section, we will look at the results of each model both numerically and with visualizations along with how they compared to each other. The logistic regression model constructed on the whole dataset gives us the best accuracy so far. Its confusion matrix for the test data is as stated in table 1.

	No	Yes
No	86998	728
Yes	7294	917

Table 1: Confusion Matrix for Logistic Regression

From this we can see that the overall accuracy on the test data is 0.9164. Additionally, the sensitivity is 0.9226 and the specificity is 0.5574.

Next, for our random forest model, after cross validation we found our model to have a test error value of 0.2362. This model found that the most important predictors were AgeCategory (factor with 13 levels) , GenHealth (factor with 5 levels), and Stroke (factor with 2 levels) in that order. The confusion matrix for the random forest gave an accuracy of 0.7638 with a sensitivity of 0.8118 and a specificity of 0.7167. Figure 3 shows the ROC curve for our random forest model.

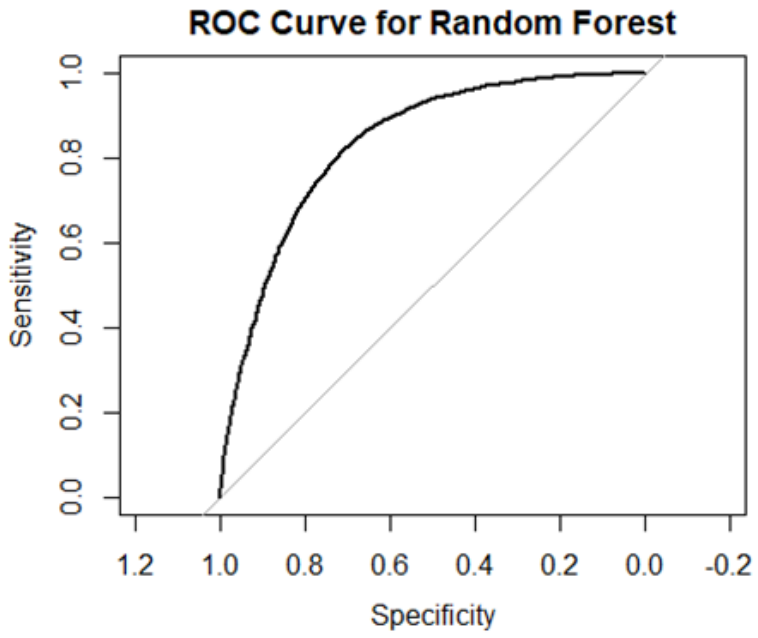

Figure 3: ROC curve

Next, for our k-nearest neighbors (KNN) model, ROC was used to choose k=9 for our final model. This yielded a confusion matrix shown in table 2 below.

	No	Yes
No	4070	1346
Yes	1458	4078

Table 2: Confusion Matrix for KNN

This gives us an accuracy on the test data of 0.744. Additionally, the sensitivity is 0.7518 and the specificity is 0.7363.

Finally, our last model was a decision tree. Figure 4 shows the decision tree below.

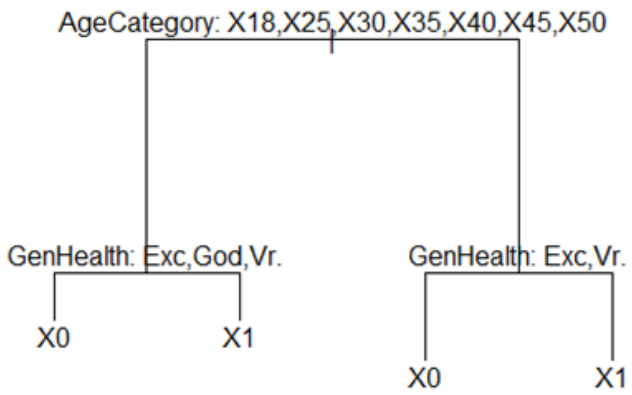

Figure 4: Decision Tree

When predicting the test data, the decision tree yields an accuracy of 0.7225 with a sensitivity of 0.7390 and a specificity of 0.7058.

Overall, when we look at our four models, the logistic regression model performs the best in terms of accuracy with a value of 0.9164. However, it has a much lower specificity at 0.5574. This may be an issue as false negatives are much more dangerous in this situation as it predicts that someone is not at risk of heart disease while they are in real time. The sampling methods for the logistic regression model also contributed to the accuracy following the sensitivity much more than the specificity. With that being said, we can also consider the random forest model to be one of the best because it has both a fairly high sensitivity and specificity in predicting heart disease risk.

4. Discussion

We believe that as more and more research is poured into understanding medical data it will revolutionize how we monitor our health and stay active for longer. The work we have done aimed at helping predict heart disease based on health metrics will be beneficial to millions of people. We expect this to be further developed and fine tuned for specific devices such as apple watches and fit bits where these predictions will be more accurate given more granular data.

Some limitations we discovered :

1. These models however would have to be optimized for scaling as it does not perform well with large data. For this analysis we explored several machine learning models such as logistic

regression, tree-based models such as random forest, c5.0 and other models such as KNN. The biggest learning from all of this was that having too much data is not always a plus point. During several runs of our analysis we ran into troubles with models getting too big to fit into system requirements and not converging even after overnight analysis(KNN). Random forest often requested too much RAM to fit the entire model and it became a challenge at times and we couldn't experiment a lot (grid search CV) because of the hardware limitations of our machines.

2. We started with a challenging problem because of the class imbalance and since there is never a single solution that works for all datasets, we had to experiment with SMOTE and ROSE. Despite these methodologies sampling does not solve the problem as even though we might train the model in a protected balanced environment the real test data always suffer from performance issues.

Therefore, in a future study we would like to spend more time working on fine tuning the balance of the data and measure a more accurate performance on the validation or test data. A more powerful machine with a GPU would be a bonus as it would cut down time and allow more experimentation.

5. References

R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL

https://www.R-project.org/.

Centers for Disease Control and Prevention. (2021, September 27). About Heart Disease. Centers for Disease Control and Prevention. Retrieved from https://www.cdc.gov/heartdisease/about.htm

Pytlak, K. (2022, February 16). Personal key indicators of heart disease. Kaggle. Retrieved May 6, 2022, from https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease

Roth, E. (2018, September 17). Heart disease causes and risk factors. Healthline. Retrieved from https://www.healthline.com/health/heart-disease/causes-risks#causes

Taiyun Wei and Viliam Simko (2021). R package 'corrplot': Visualization of a Correlation Matrix (Version 0.92). Available from https://github.com/taiyun/corrplot

H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

Nicola Lunardon, Giovanna Menardi, and Nicola Torelli (2014). ROSE: a Package for Binary Imbalanced Learning. R Journal, 6(1), 82-92.

Greg Snow (2020). TeachingDemos: Demonstrations for Teaching and Learning. R package version 2.12. https://CRAN.R-project.org/package=TeachingDemos

Brandon Greenwell, Bradley Boehmke, Jay Cunningham and GBM Developers (2020). Gbm: Generalized Boosted Regression Models. R package version 2.1.8. https://CRAN.R-project.org/package=gbm

Max Kuhn (2022). caret: Classification and Regression Training. R package version 6.0-91. https://CRAN.R-project.org/package=caret

Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frédérique Lisacek, Jean-Charles Sanchez and Markus Müller (2011). pROC: an open-source package for R and S+ to analyze and compare ROC curves. BMC Bioinformatics, 12, p. 77. DOI: 10.1186/1471-2105-12-77 <http://www.biomedcentral.com/1471-2105/12/77/>

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. R News 2(3), 18--22.

Brian Ripley (2021). tree: Classification and Regression Trees. R package version 1.0-41. https://CRAN.R-project.org/package=tree

David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel and Friedrich Leisch (2021). e1071: Misc Functions of the Department of Statistics, Probability

Theory Group (Formerly: E1071), TU Wien. R package version 1.7-9. https://CRAN.R-project.org/package=e1071

Emil Hvitfeldt (2022). themis: Extra Recipes Steps for Dealing with Unbalanced Data. R package version 0.2.1. https://CRAN.R-project.org/package=themis

Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686

Hadley Wickham and Jim Hester (2021). readr: Read Rectangular Text Data. R package version 2.0.1. https://CRAN.R-project.org/package=readr

Yachen Yan (2016). MLmetrics: Machine Learning Evaluation Metrics. R package version 1.1.1. https://CRAN.R-project.org/package=MLmetrics

Max Kuhn (2021). caret: Classification and Regression Training. R package version 6.0-90. https://CRAN.R-project.org/package=caret

Microsoft Corporation and Stephen Weston (2022). doSNOW: Foreach Parallel Adaptor for the 'snow' Package. R package version 1.0.20. https://CRAN.R-project.org/package=doSNOW

Ben Hamner and Michael Frasco (2018). Metrics: Evaluation Metrics for Machine Learning. R package version 0.1.4. https://CRAN.R-project.org/package=Metrics