Heart Disease Classifier

By: Grace Panos,
Neftali Lemus,
Cameron Chatfield
(Stats 101C, L2).

Table of Contents

1. Heart Disease Introduction
    a. Dataset Discussion
2. Data Processing
    a. Exploratory analysis and visualizations
    b. Data Transformation
3. Modeling
    a. Introduction to glm()
    b. Variable selection
    c. Methodology
4. Limitations and conclusions

Heart Disease Overview

Heart Disease is one of the leading causes of death in the United States, responsible for about 1 in 4 deaths. Unhealthy lifestyle decisions account for a variety of the factors that can lead to them such as unbalanced diets, lack of physical exercise, and stress. Early diagnosis is the key for its treatment and with classifiers like this, we can recognize the early signs and help millions of patients around the world.

Data Set Description

Training Data Set:

-4220 Observations

- 20 predictors and ‘HeartDisease’ as the outcome.

-2524 missing observations

Test Data Set:

-1808 observations

-with 20 predictors

-1148 missing observations

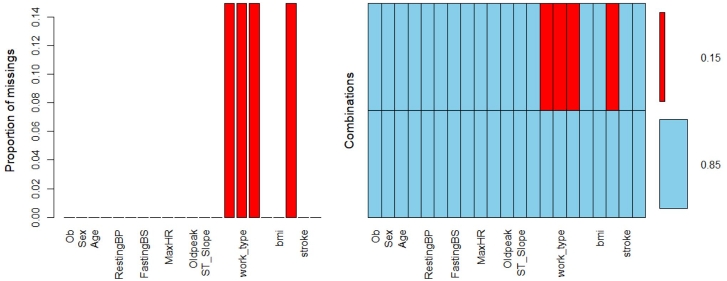
EDA & Further Visualization of Raw Data

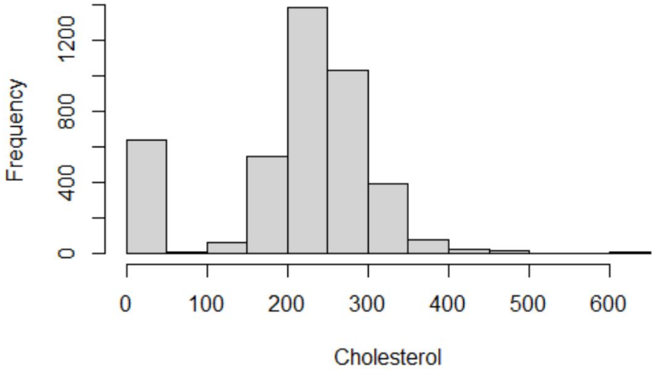
The initial structure of the data shows the following problems:

- **Missing categorical data** in work type, Residence Type, smoking status, and if the patient has ever married. They have 631 NA's each, accounting for about 15% of the information for the mentioned variables.

- **Mislabeling and Unbalanced categorical data** such as Gender ("Male", "Female", "M", "F") and Work Type ("Never worked" with 12 obs)

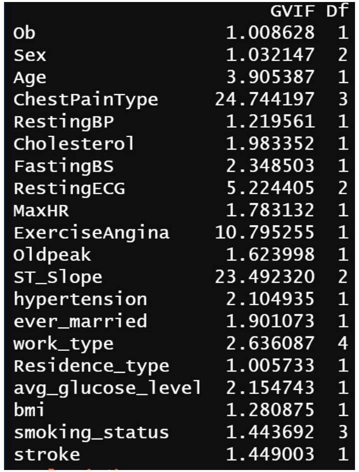
- **Non-Random Observations** in numerical predictors such as Cholesterol with 600 observations at 0.





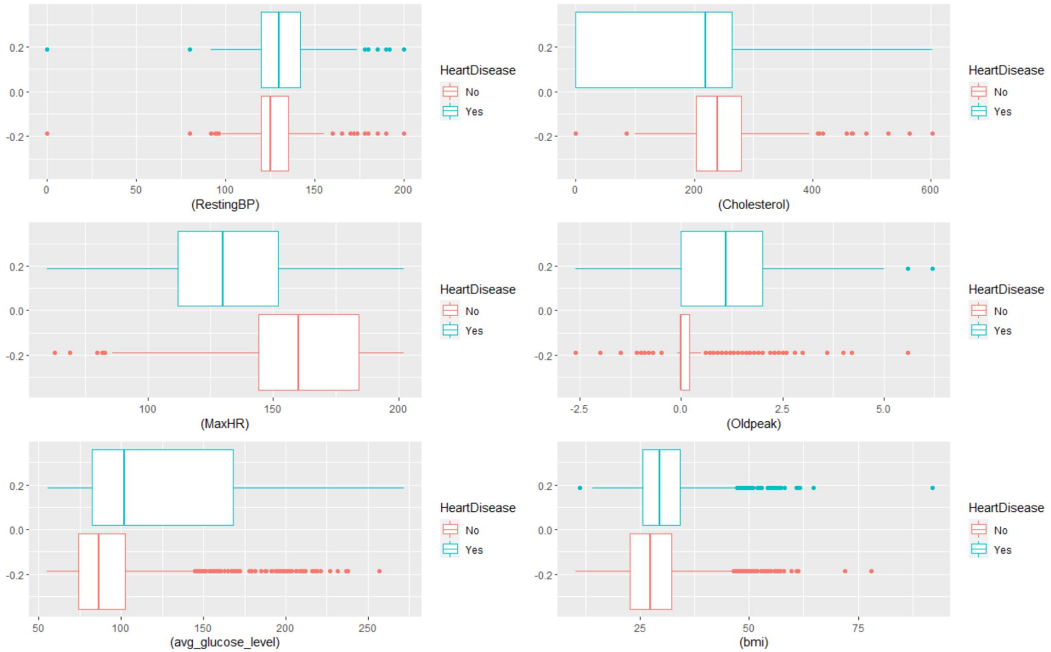
EDA & Further Visualization of Raw Data

- **Multicollinearity** among predictors that lead to redundancy and overfitting.



```
                      GVIF Df
Ob                1.008628  1
Sex               1.032147  2
Age               3.905387  1
ChestPainType    24.744197  3
RestingBP         1.219561  1
Cholesterol       1.983352  1
FastingBS         2.348503  1
RestingECG        5.224405  2
MaxHR             1.783132  1
ExerciseAngina   10.795255  1
Oldpeak           1.623998  1
ST_Slope         23.492320  2
hypertension      2.104935  1
ever_married      1.901073  1
work_type         2.636087  4
Residence_type    1.005733  1
avg_glucose_level 2.154743  1
bmi               1.280875  1
smoking_status    1.443692  3
stroke            1.449003  1
```

- **Outliers and Extreme Values** in most numerical predictors that severely affect the distribution of the data.

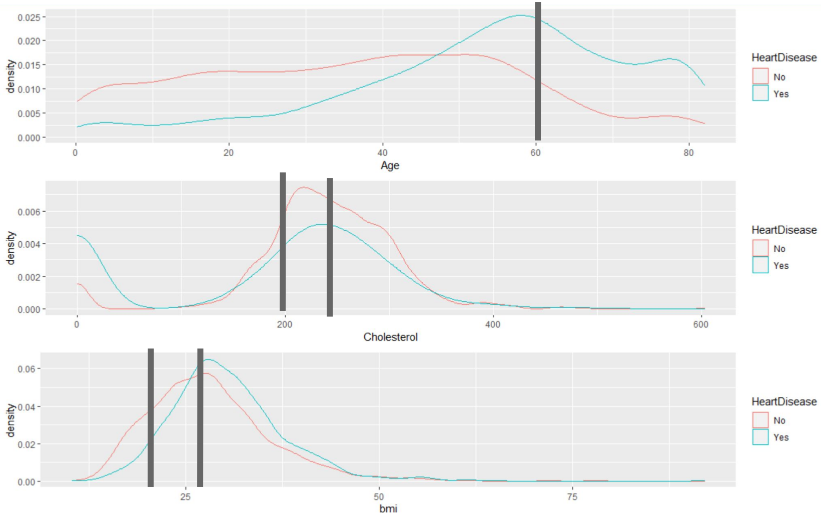


Data Processing & Cleaning

- **MissForest Imputation:** Used to impute missing observations, converting categorical to factors and binding the resultant data with all numerical predictors. We could not just omit them as this would get rid of 15% the entire data.

- **Transformed Numerical to Categorical:** Since the original HD.train was full of outliers, cutting the data in specific ranges would simplify our working information and normalize extreme values. To find the best places to cut our data, we did a small research about healthy/unhealthy parameters for each predictor. We split the Age, Cholesterol, and bmi variables into 2 to 3 level categorical predictors.

- **Recoding categorical variables:** Categorical data was also unbalanced, there were predictors that had categories with less that 20 observations and others were just mislabeled. We decided that it was better to merge categories such as worktype and gender and st slope. This would reduce noise and ensure a better distribution of the data.



Choosing A model

We tried many different classification models before settling on our final model including lda, qda, glm, decision trees, and random forest. Here were our most successful models:

- glm()
  - 18.9% testing missclassification rate
- Random forest (xgboost)
  - 18.2% testing missclassification rate

As a next step, we split our training data into "training" (70%) and "testing" (30%) subsets.

Random Forest

Using the xgboost package, we fit a random forest model on our subsetted training dataset. After tuning our parameters, here were our results:

	No	Yes
No	451	100
Yes	53	394

missclassification rate = 15.5%

Random Forest (downfalls)

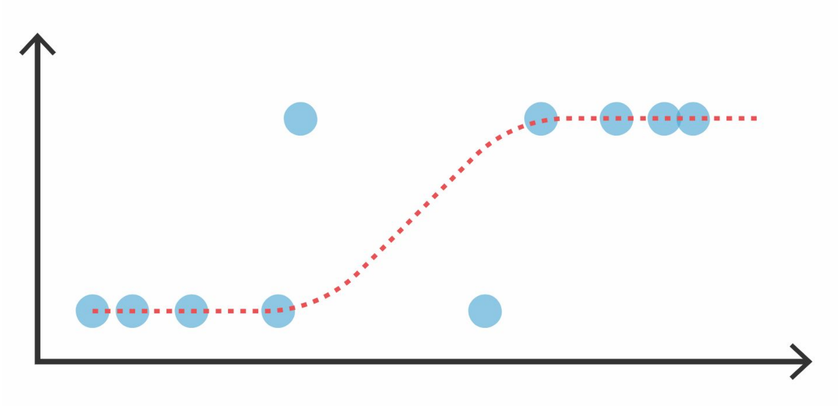
We ended up not going with random forest due to a few key issues

- Tendency to overfit the training data
- Unpredictable testing results
  - Training missclassification rate: 15.5%
  - Testing missclassification rate: 18.2%
- Did not perform well on kaggle data set

All of these issues led us to favor the glm() model due to greater flexibility and more predictable results.

GLM Classification (Description)

- Generalized linear models (glms) include both linear and logistic regression
- HeartDisease is a binary outcome variable with levels (Y, N)
- A glm() of family binomial() would fit a logistic classification model



Methodology: (Best Predictors, subsetting, significance)

We ran a summary of our glm() model, did backward stepwise selection using AIC and BIC, and used a lasso model to select our best predictors. From here we did the following to ensure we were using the best subset of predictors:

- Categorized the “strength” of our predictors based on how many feature selection models they appeared in
- Ran our model with our top 5 “strong” predictors
  - Added and subtracted additional “medium” predictors to find the best combination of predictors

Methodology (Feature Selection)

Color	Predictor Strength
🟨	Strong
🟧	Medium

full model significant coefficient estimates

	estimate
maxhr	-0.018
oldpeak	1.321
avg_glucose_level	0.024
exerciseangia	0.164
stslope	0.2559
stroke	-2.48

forward stepwise (BIC)

	estimate
strokeYes	-0.3822
Oldpeak	0.209
ExerciseAngiaY	0.175
ST_SlopeUp	-0.162
FastingBSYes	0.1136
Cholesterol	0.0387
MaxHR	0.0034
avg_glucose_level	0.0031

forward stepwise (AIC)

	estimate
strokeYes	-0.38
Oldpeak	0.2032
ExerciseAngiaY	0.1911
ST_SlopeUp	-0.155
FastingBSYes	0.097
ChestPainTypeNAP	-0.047
Cholesterol	0.041
Ageyoung	0.04
smoking_statusnever smoked	0.035
gvtjob	0.035

lasso model

	estimate
strokeYes	-0.275
Oldpeak	0.18933
ExerciseAngiaY	0.1089
FastingBSYes	0.088
ST_SlopeUp	-0.086
Ageyoung	0.026
SexMale	0.015
ever_marriedYes	0.015
ChestPainTypeATA	0.012
smoking_statusnever smoked	0.0119

Methodology (Tuning Parameters - ROCR)

To ensure that we were using the best p to determine the cutoff for Yes or No, we used the ROCR library to find the best parameter. Here was our output:

accuracy	cutoff
0.8165877	0.5026786

This shows that the best cutoff was 0.5, which confirmed that our parameter was the best choice.

Results(prediction, Confusion Matrix, Training MSE)

The best combination of predictors were stroke, ExerciseAngina, ST_Slope, Oldpeak, FastingBS, avg_glucose_level, Sex, MaxHR, and Cholesterol.

- missclassification rate: 18.4%
- Confusion matrix:

	No	Yes
No	1915	468
Yes	314	1523

Model Limitations and Ways to Further Improve

Limitations:

- GLM tends to oversimplify without appropriate tuning of parameters. This led to a loss in significance once we used regsubsets to find the best predictors.
- This classifier is sensitive to outliers.
- Variables like Old Peak were hard to normalize due to the structure of the data.

Ways to Further improve:

- Merge data with new information.
- Boost predictors.
- Redefine categorical boundaries.

Final Conclusions and kaggle ranking

- Throughout this project we were able to understand the importance behind a clean data set. Independence, significance, and normalization were some of the key features that had our main focus in order to come up with the best classifier.

- The idea of a “right model” does not exists in statistics,all of the different methods learned in 101C are subject to the bias variance tradeoff that we extensively discuss throughout this quarter. We must consider error as part of life and minimize its effects as we aim to reduce it along the way.

“Team 100% ” 6th place in lecture 2 , with a final accuracy of 0.79742