project_writeup

2023-04-30

###Introduction

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
## Warning: package 'DPpack' was built under R version 4.2.3
```

```
## Warning: package 'MASS' was built under R version 4.2.2
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
## Warning: package 'e1071' was built under R version 4.2.3
```

```
## Warning: package 'randomForest' was built under R version 4.2.3
```

```
## randomForest 4.7-1.1
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## Warning: package 'xgboost' was built under R version 4.2.3
```

```
##
## Attaching package: 'xgboost'
##
## The following object is masked from 'package:dplyr':
##
##     slice
```

```
## Warning: package 'GGally' was built under R version 4.2.2
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

The data set for this project is a record of 735 patients with Heart Disease. Each patient is recorded with related attributes.

Variable description:

Age: age of the patient [years]
Sex: sex of the patient [M: Male, F: Female]
ChestPainType: chest pain type [TA: Typical Angina, ATA: Atypical Angina,
NAP: Non-Anginal Pain, ASY: Asymptomatic]
RestingBP: resting blood pressure [mm Hg]
Cholesterol: serum cholesterol [mm/dl]
FastingBS: fasting blood sugar [1: if FastingBS $>$ 120 mg/dl, 0: otherwise]
RestingECG: resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of $>$ 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria]
MaxHR: maximum heart rate achieved [Numeric value between 60 and 202]
ExerciseAngina: exercise-induced angina [Y: Yes, N: No]
Oldpeak: oldpeak = ST [Numeric value measured in depression]
ST_Slope: the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]
HeartDisease: output class [1: heart disease, 0: Normal]

Our goal for this project is predict future patients if they have heart disease based on the patient's attributes. We will use these past patient data to train a learning model that accurately and precisely predict future cases.

###Exploration The ratio of patients with heart disease and without is 396 to 339, which is a fairly balanced ratio. From the histograms, none of the univariate distribution of continuous features are not heavily skewed that needs a data transformation or heavy outliers that needed to omit. From the conditional distributions of featured given the heart disease outcome shows some interesting patterns. Current features types have higher proportions of having heart disease such being Male (62%), having asymptomatic ChestPainType (79%), having fast blood sugar (78%), having exercise Angina (83%), and having ST_Slope of downward sloping (78%) or flat (82%). This insight led us to believe that categorical variables can be quite predictive of the heart disease outcome, so we decide to perform one-hot encoding on all categorical variables.

```
## null device
##           1
```

#Multivariate EDA

#One-hot Encoding on categorical variables

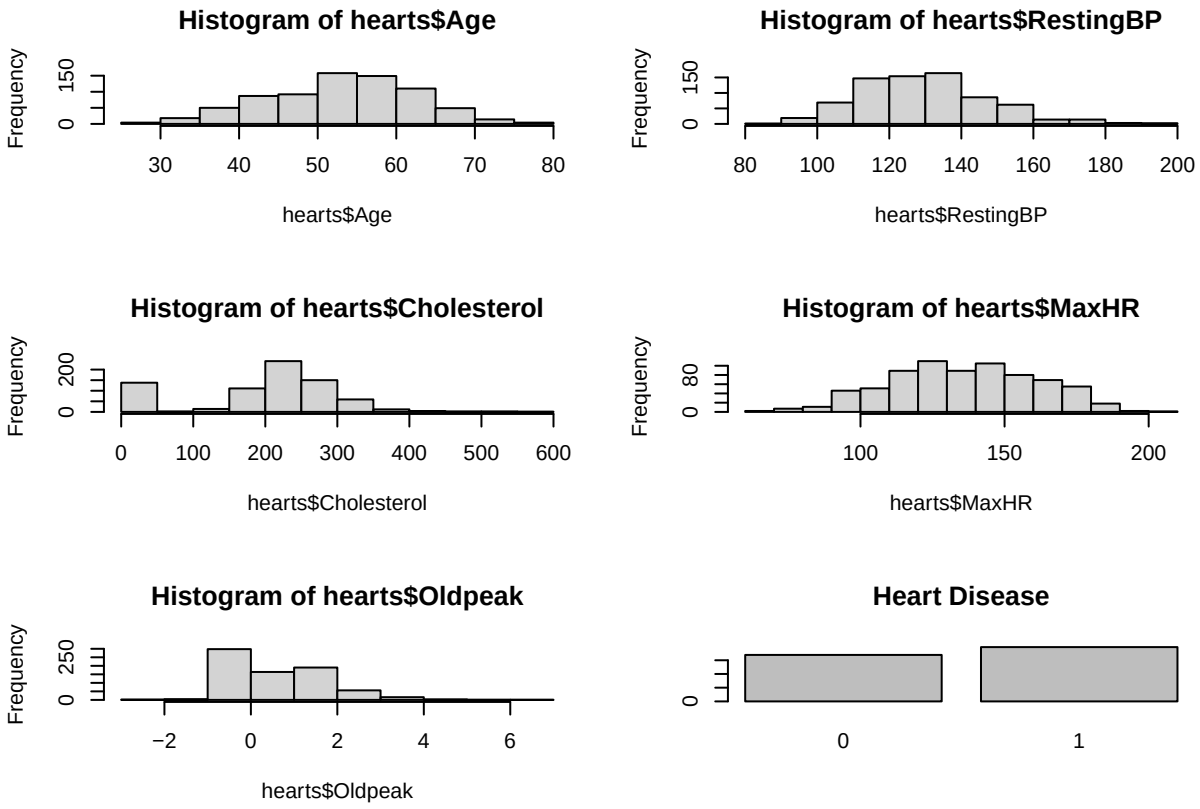


Figure 1: Continuous Predictor Univariate Distribution

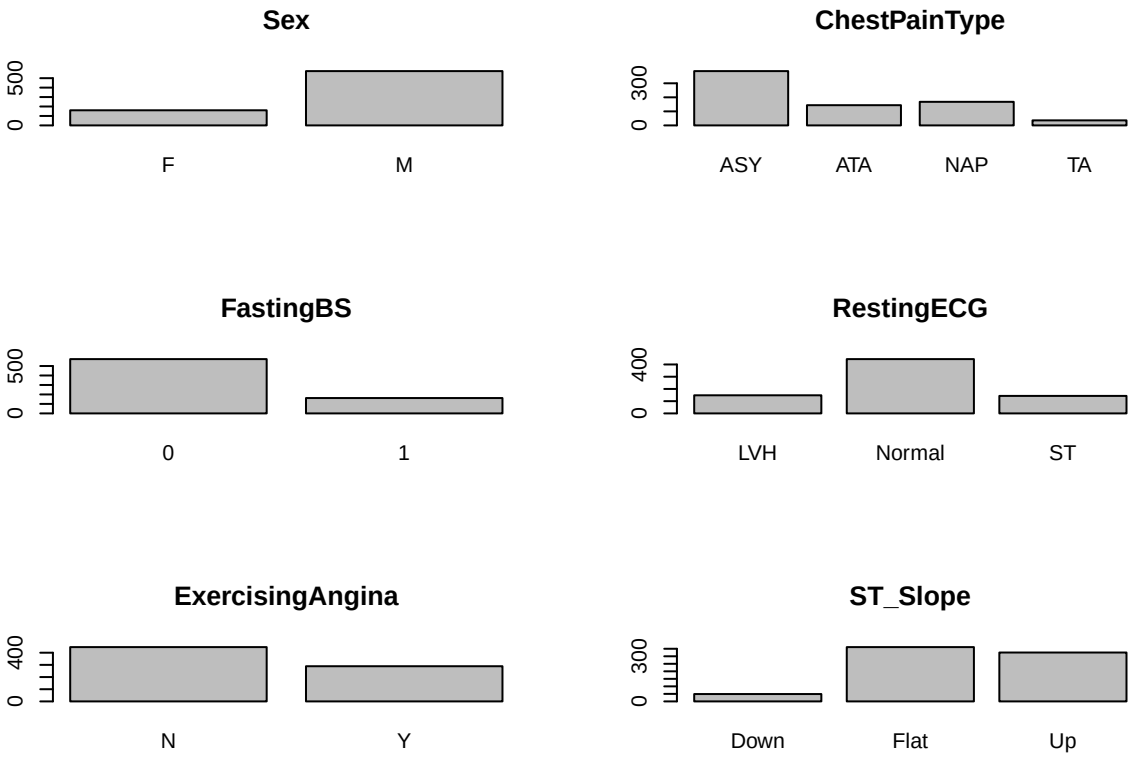


Figure 2: Binary Predictor Univariate Distribution

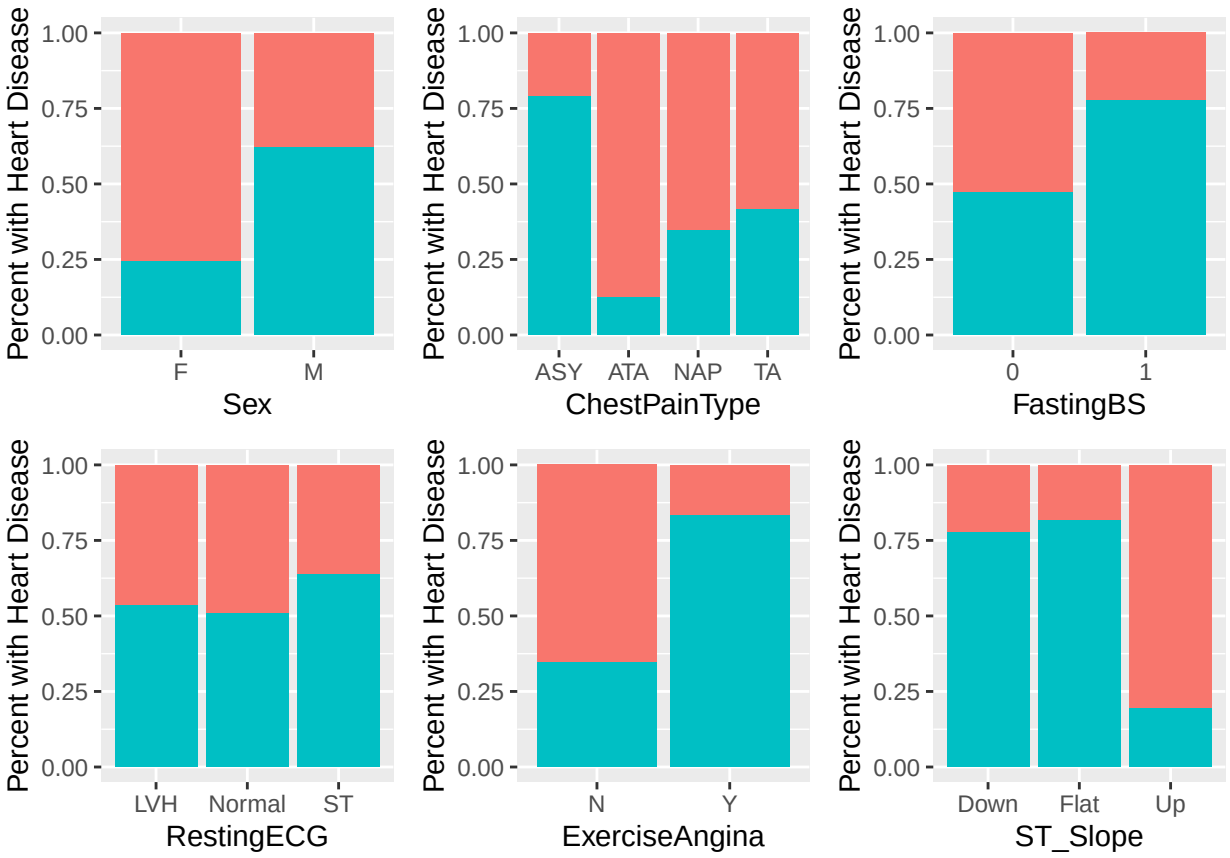


Figure 3: Binary Predictor Conditional Distribution on Heart Disease

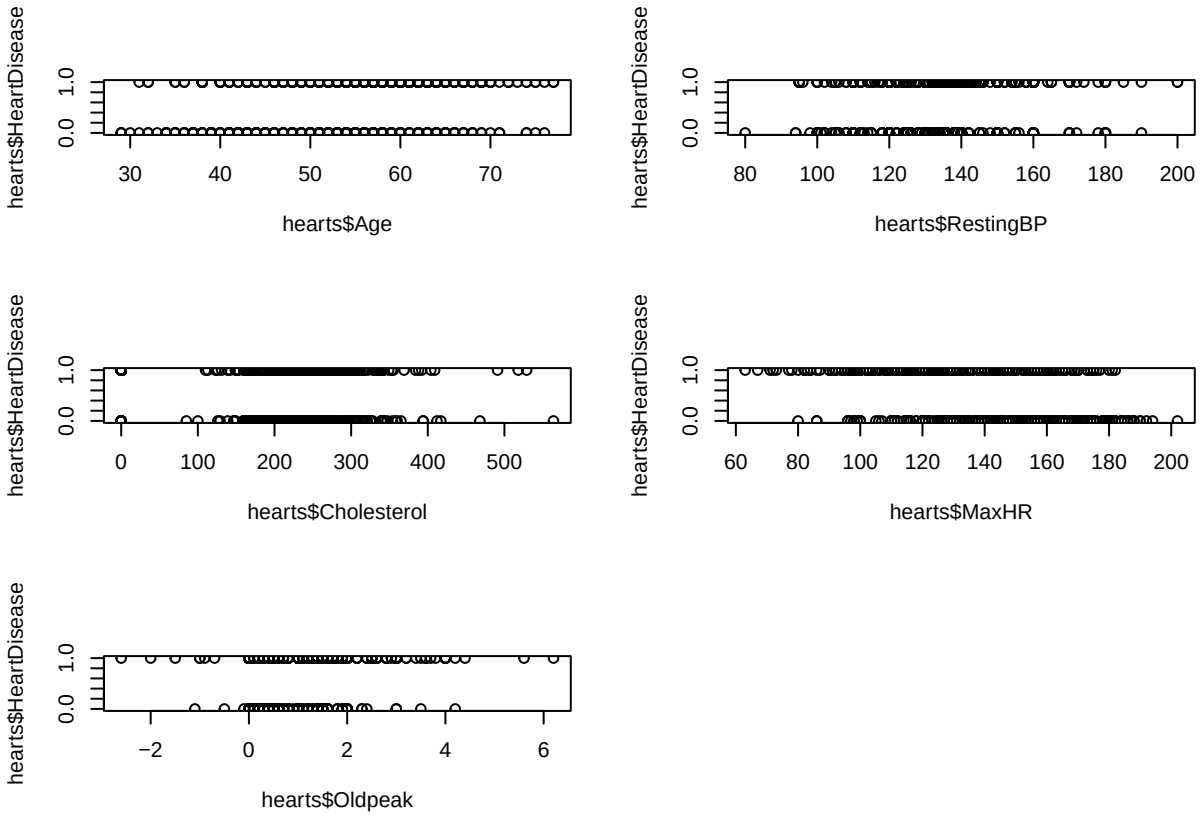


Figure 4: Continuous Predictor Conditional Distribution on Heart Disease

One-Hot Encoding Note: To limit degrees of freedom, k categorical variable is transformed to k-1 features Sex == F corresponds to Is_Male == 0 ChestPainType == TA corresponds to CPT_ASY, CPT_ATA, CPT_NAP == (0, 0, 0) Resting_ECG == Normal corresponds to ECG_LVH, ECG_ST == (0, 0) ExerciseAngina == N corresponds to ExerciseAngina == 0 ST_Slope == Flat corresponds to (ST_Slope_Flat_Up, ST_Slope_Flat_Down) == (0, 0)

Dropping Excess Variables

###Supervised Analysis

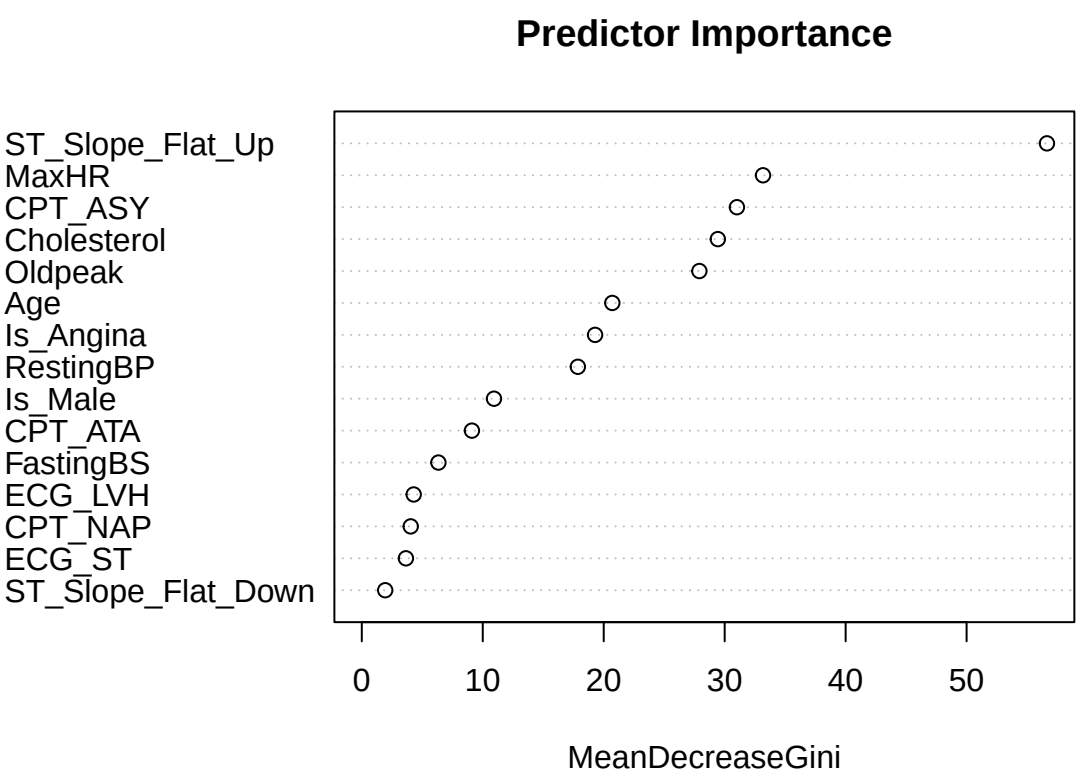
We decided to include all features in the learning model. This is because even though a feature may have no association with our response variable, including it in our model does not significantly increase the bias into our model. For feature engineering, we simply implement one-hot encoder on all categorical variables, and manually centering and scaling all our continuous variables. After one-hot encoding, we have a total of 15 predictors. One important detail is we scale our test data set with our training data set's mean and standard deviation when evaluating our model performance with cross validation. This is because our models was trained with training data set; therefore, it should be on the same scale as our training data set. We test five learning algorithms to compare their sample expected test errors, which are logistics regression, QDA, SVM, random forest, and gradient boosting. Each of these learning models, I tune their hyper parameter with cross validation because comparing them with other learning models. For logistics and QDA regression, there is no tuning parameter. For SVM, we use gridSearch library cross-validation to select the best margin C, γ, and kernel for the data set. For random forest, we tested different number of trees (100 to 1000 with 100 per intervals) in a forest and evaluated their test error using 0/1 lost function. We found 500 trees per forest to have the lowest test error. Next, I use Out-of-bag error to tune the bext number of variables selected per split, find 3 variables at each split to have lowest OOB error at 0.233. For gradient boosting, we intentionally have a shallow tree depth of 8 for high computational speed, low variance, but high bias.

Once we selected models for the different algorithms, we ran a 5-fold cross validation of these five learning methods, evaluate them using cross entropy and 0/1 lost separately. In both lost functions, the random forest emerge as the best performing method with lowest cross-entropy loss function of 0.3335, and lowest accuracy rate of 0.8775.

To explain relationship between predictor and prediction, we plot the mean decrease gini score of each predictor. The higher the mean decrease gini score, the higher the importance of the variable in the model. Although one drawback of mean decrease gini score method as a measure of variable importance is it favor features with high cardinality, we engineered our features with one-hot encoding, thus eliminating this drawback. From the variable importance plot, we see ST_Slope_Flat_Up has the highest mean decrease gini score followed by MaxHR, CPT_ASY, Cholesterol, and Oldpeak. This result reinforces our earlier conditional distribution graphs that shows a high proportion (76%) of no heart disease patients for ST_Slope Up, but high proportion of heart disease patients for patients with ST_Slope Up (78%) and ST_Slope Flat (82%).

```
##         Logistics               QDA               SVM Gradient_Boosting
##         0.3439973         0.7405972         0.3366983         0.3419108
##     Random_Forest
##         0.3335229
```

```
##         Logistics               QDA               SVM Gradient_Boosting
##         0.8585034         0.8476190         0.8761905         0.8666667
##     Random_Forest
##         0.8775510
```



###Analysis of results

We compute the confusion matrix, sensitivity, and specificity of model, which are 84.87% and 88.01% respectively. Our model is slightly more specific than sensitive, meaning there are fewer false positive than false negative rate. From a practical point of view, this is a preferable outcome as low specificity may not be feasible for screening, since many people without the disease will screen positive, and potentially receive unnecessary diagnostic procedures.

We calculate 95% pivotal confidence interval of our model sensitivity and specificity by bootstrapping samples with replacement from training data. The pivotal 95% confidence interval is (0.8222 0.8630) for sensitivity, and (0.8699 0.9051) for specificity. Overall, our model's true specificity is statistically different and higher than sensitivity, thus we can say our model does better on false positive than false negative samples.

```
##                 actual_values
## predicted_values   0   1
##                0 230  38
##                1  41 279
```

Our prediction model when deployed on actual test data set has an accuracy of 0.1202, and an accuracy error of 0.0087. With a precision error of less than 1%, we believe our model has low variance. We can work on reducing our model's bias, ideally lowering our test error to less than 10%. If given more time, we can collect more data and refit our models tuning parameters, which will in theory yield a model with lower bias and variance. Another idea for improvement is experimenting with more engineered features by adding and testing interaction terms and transformations of our current features. This can also improve our model's fit.