Predicting heart disease

Mike Miemczok

5 5 2021

Introduction

The dataset used in this analysis is the Heart-Disease dataset provided by following creators:

Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D. University Hospital, Zurich, Switzerland: William Steinbrunn, M.D. University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D. V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D. Donor: David W. Aha (aha '@' ics.uci.edu) (714) 856-8779

and reachable under following web adress: "https://www.kaggle.com/ronitf/heart-disease-uci".

The main goal of this project it is to predict heart disease by the given attributes of the dataset. More informations to the dataset will be given in the analysis section.

To achieve this goal the project was built up in following steps:

1. Exploring the dataset.
2. Data preprocessing.
3. Visualization of data.
4. Testing multiple models to get the best three performing.
5. Optimize the three given models by tuning parameters and correlations.
6. Present the results.

Analysis

Data exploration

In the first step it is helpful to check the raw dataset and get comfortable with it before starting any analysis. We started with getting an overview of the structure of the dataset:

```
## 'data.frame':    303 obs. of  14 variables:
##  $ ï..age  : int  63 37 41 56 57 57 56 44 52 57 ...
##  $ sex     : int  1 1 0 1 0 1 0 1 1 1 ...
##  $ cp      : int  3 2 1 1 0 0 1 1 2 2 ...
##  $ trestbps: int  145 130 130 120 120 140 140 120 172 150 ...
##  $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
##  $ fbs     : int  1 0 0 0 0 0 0 0 1 0 ...
##  $ restecg : int  0 1 0 1 1 1 0 1 1 1 ...
##  $ thalach : int  150 187 172 178 163 148 153 173 162 174 ...
##  $ exang   : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
##  $ slope   : int  0 0 2 2 2 1 1 2 2 2 ...
##  $ ca      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ thal    : int  1 2 2 2 2 1 2 3 3 2 ...
##  $ target  : int  1 1 1 1 1 1 1 1 1 1 ...
```

Followed by getting an overview over the number of columns and rows:

```
## [1] 14
```

```
## [1] 303
```

As we can see the dataset contains only numerical data. So it's useful to understand the meaning of the numercial numbers for each column:

Column	Meaning	Values
Age	Age in years	29 - 77
Sex	Gender of the patient	0: Female; 1: Male
Cp	Chest Pain Type	0: asymptomatic; 1: atypical angina; 2: non-anginal pain; 3: typical angina
Trestbps	Resting Bloog Pressure in mm Hg	94 - 200
Chol	Serum Cholosterol in mg/dl	126 - 564
Fbs	Fasting Blood Sugar > 120 mg/dl	0: False; 1: True
Restecg	Resting Electrocardiographic Results	Value 0: showing probable or definite left ventricular hypertrophy; 1: normal; 2: having ST-T wave abnormality
Thalach	Maximum Heart Rate Achieved	71 - 202
Exang	Exercise Induced Angina	0: No; 1: Yes
Oldpeak	ST depression induced by exercise relative to rest	0 - 6.2
Slope	Slope of the peak exercise ST segment	0: Downsloping; 1: Flat; 2: Upsloping
Ca	number of major vessels (0-3) colored by flourosopy	0 - 3
Thal	Thallium Stress Test	1: fixed defect; 2: normal; 3: reversible defect
Target	Heart Disease present	0: heart disease; 1: no heart disease

Preprocessing

Before we start to get an better insight in the dataset we need to make sure the dataset doesn't contains any missing values like N/As:

```
##   ï..age      sex       cp trestbps     chol      fbs  restecg  thalach
##        0        0        0        0        0        0        0        0
##    exang  oldpeak    slope       ca     thal   target
##        0        0        0        0        0        0
```

Also we need to make sure that the dataset is balanced. That means that there is a good balance between patients who have a heart disease and those who don't.

```
##
##   0   1
## 138 165
```

We see, that we have 138 records of patients who do have a heart disease and 165 records of patients who don't. All in all we have a balanced datset so we don't need to use any balancing techniques.

It makes also sense for following visualizations to translate the numerical number of gender into human readable -> 1 = Male and 0 = Female.

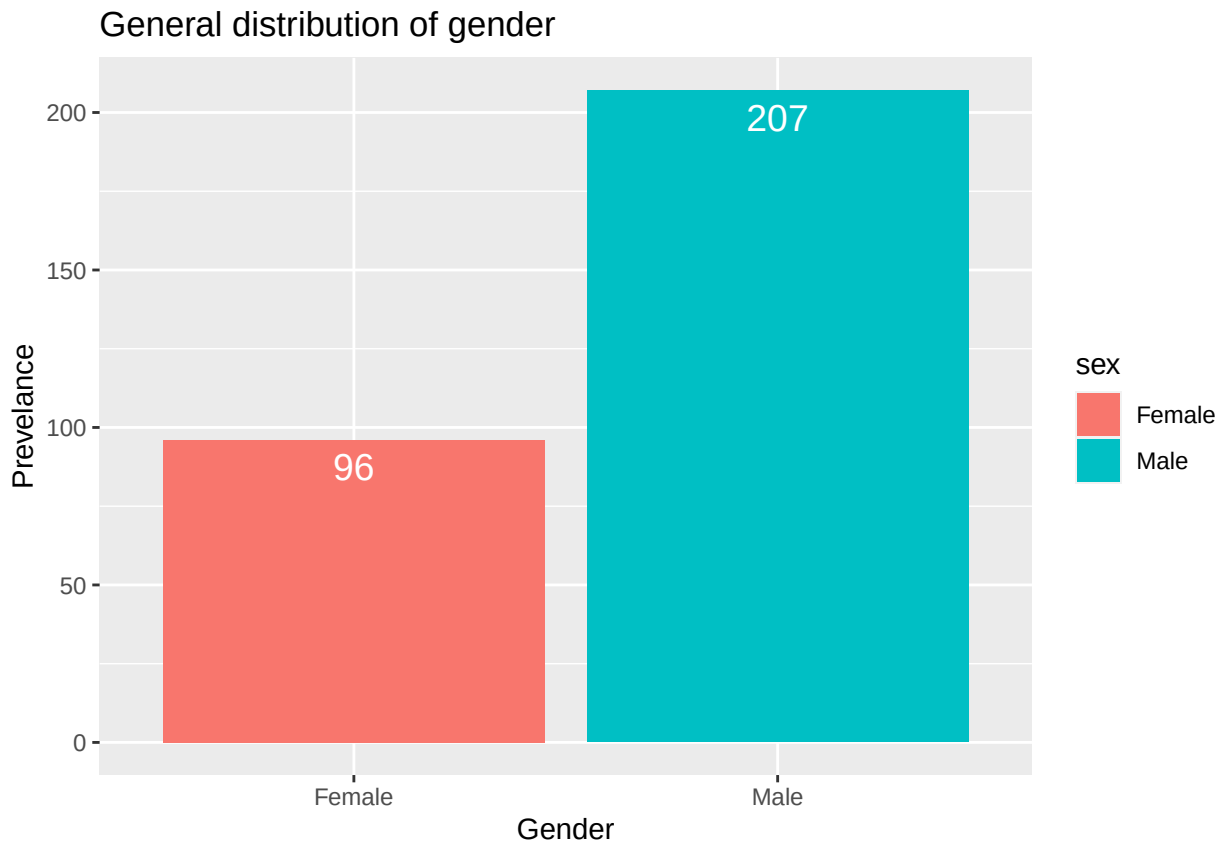
```
## [1] 1 1 0 1 0 1
```

```
## [1] "Male"   "Male"   "Female" "Male"   "Female" "Male"
```

Visualization

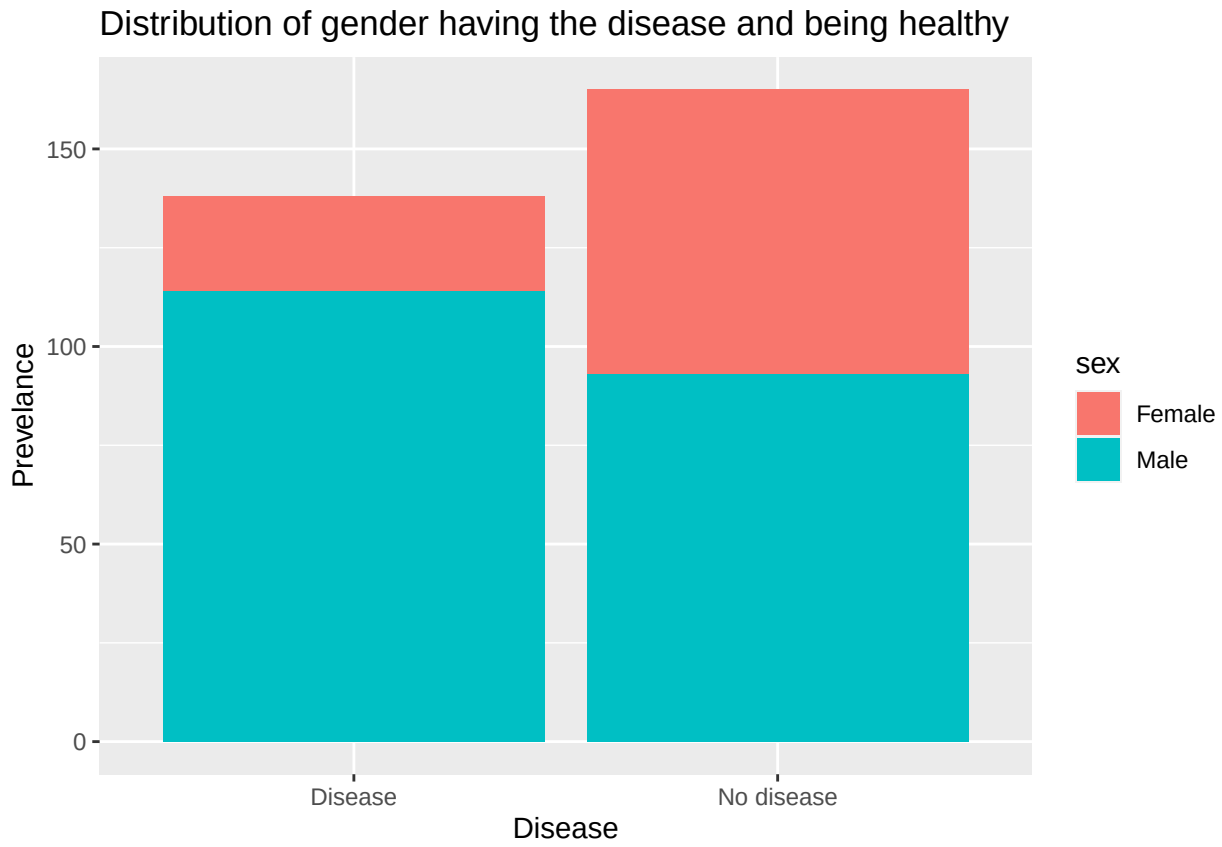
To get more insight into the data we start with some visualization.

The first visualization shows the general distribution of gender in the heart disease dataset.



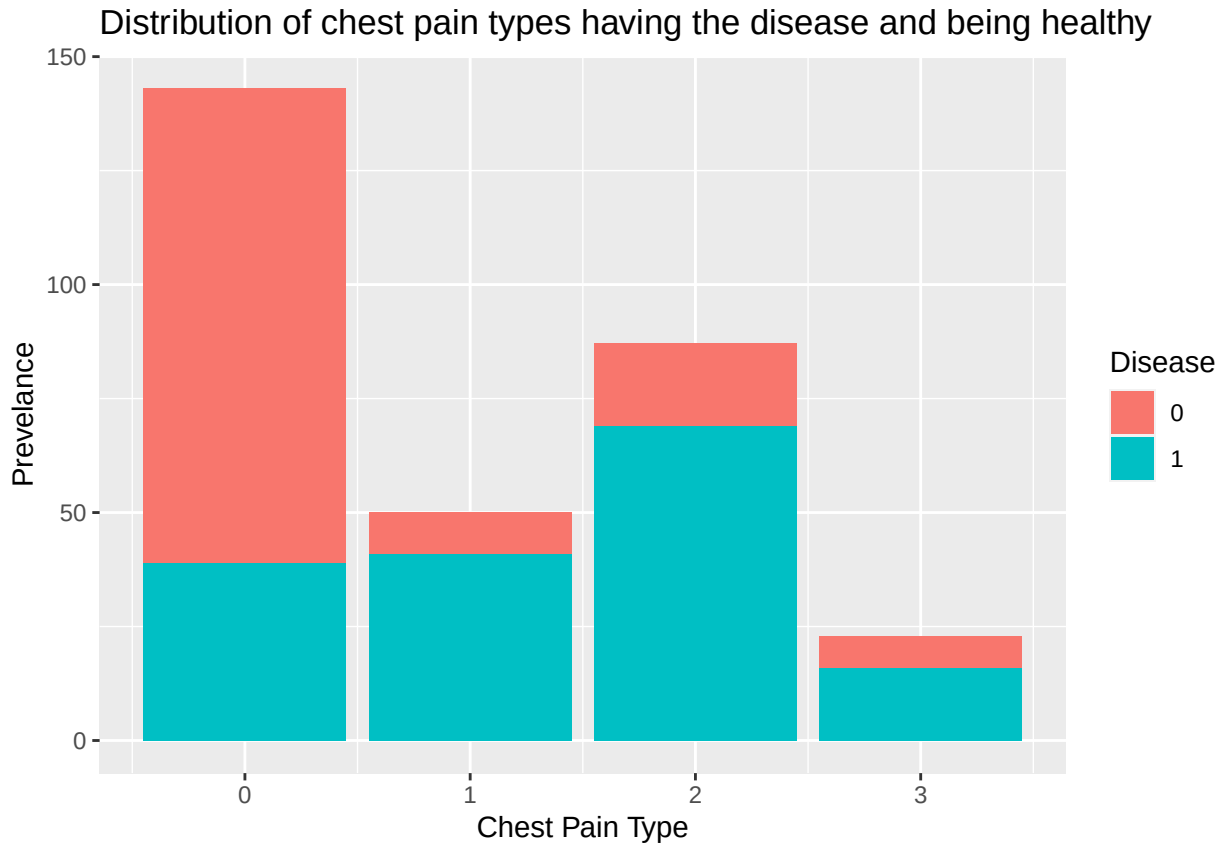
We see that we have twice as much males in the dataset as we have females.

It makes also sense to get an overview over the distribution of diseases by gender.



If we remember the balance section we already know that we have 138 records with disease and 165 without disease. In this visualization we see that of the 138 records with heart disease men tend to be more likely to have a heart disease than women. The bar showing without disease is relatively balanced between men and women.

With the next graph we are going to examine how the different types of chest pains correspond to the chance of having a heart disease.



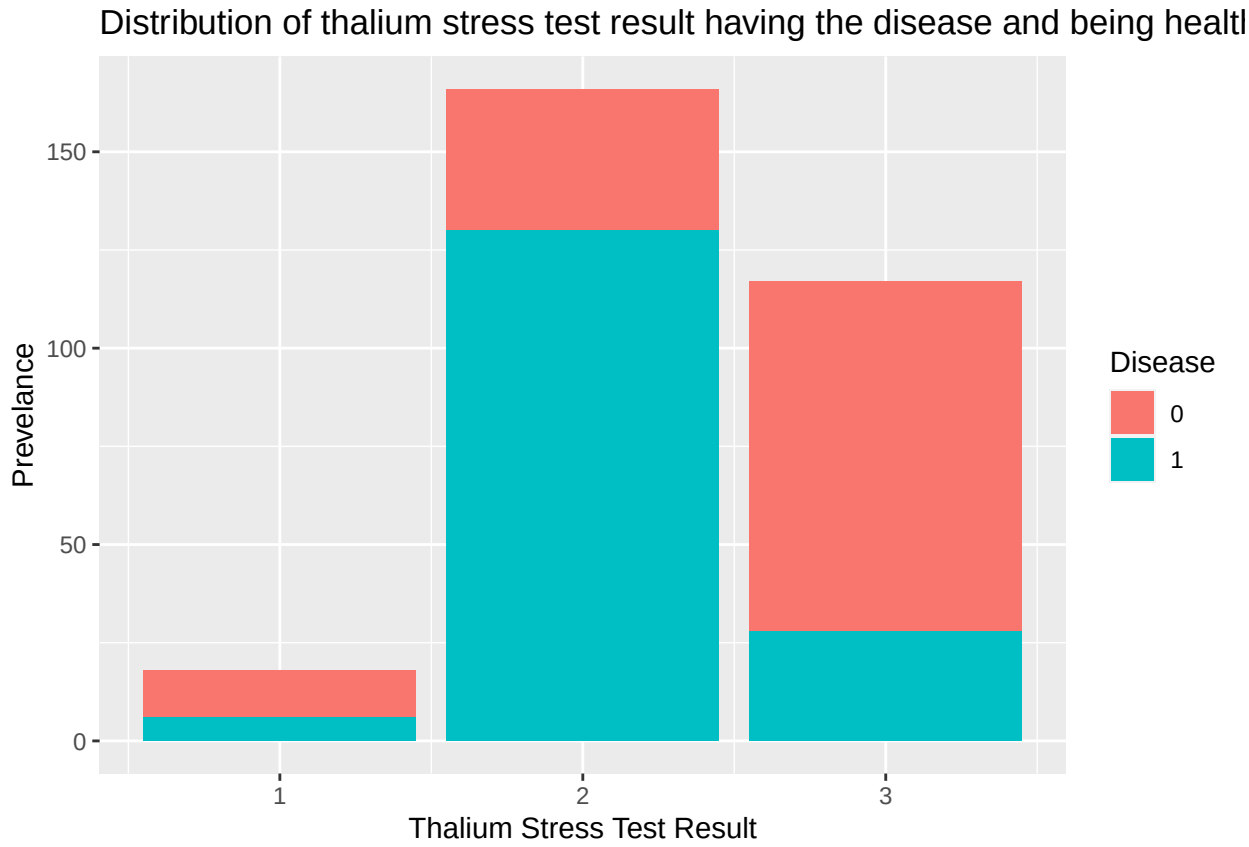
We would expect more heart disease cases in category 1 (atypical angina) and 3 (typical angina) or maybe even 2 (non-anginal pain), but contradicting to our expectations most cases of heart disease occur in category 0 (asymptomatic). So if a patient experiences chest pain it is not necessarily an indicator for a heart disease.

As next we will inspect the impact of Thalium stress test result which is categorized in three sections:

1: fixed defect

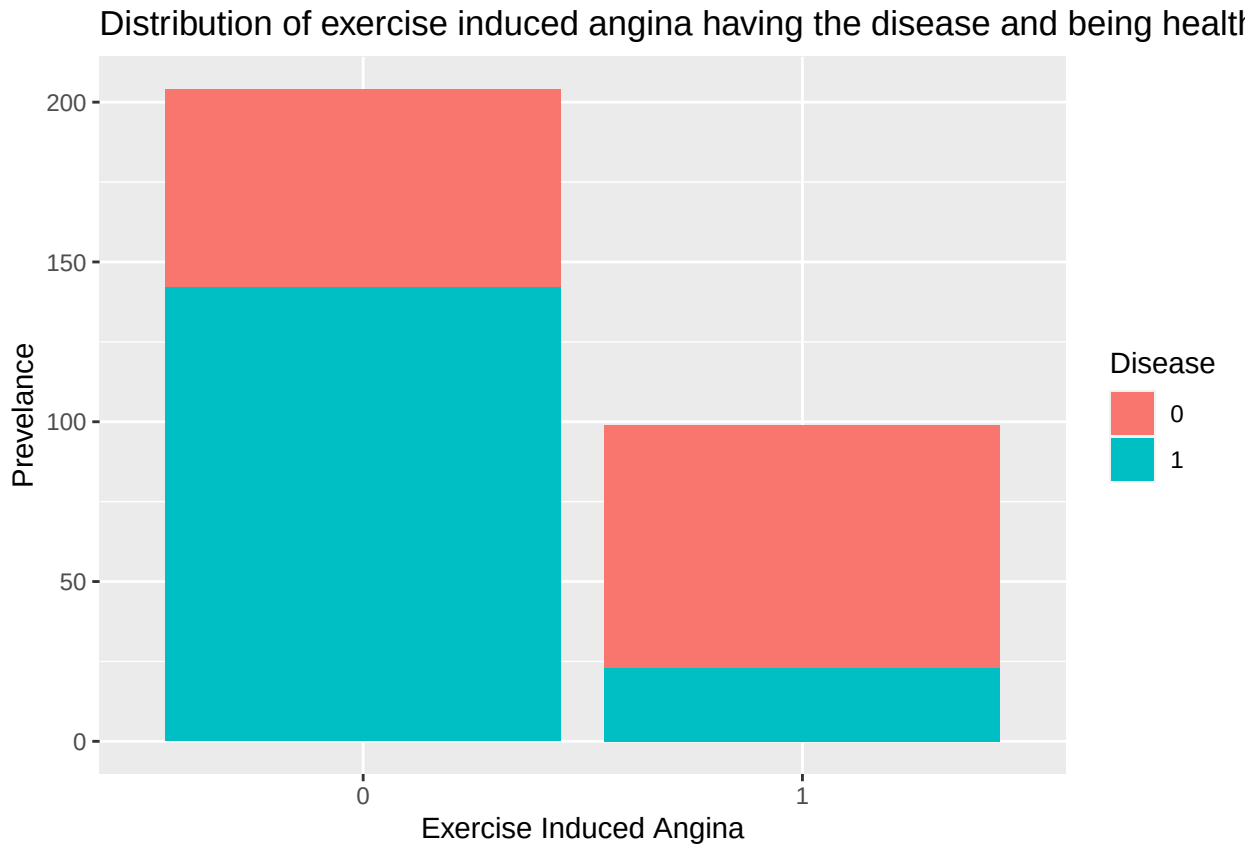
2: normal

3: reversable defect



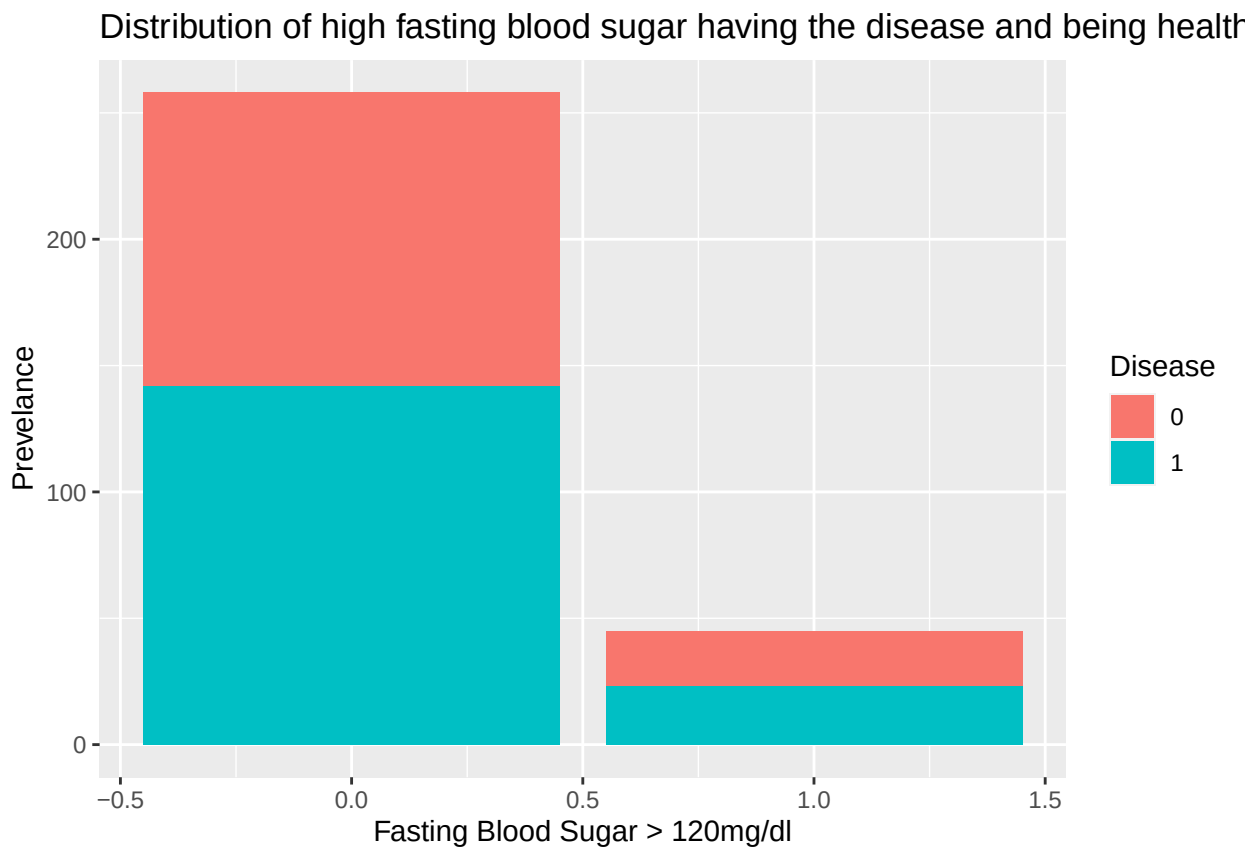
We can see that the normal type (2) includes much fewer heart disease patients than the two other types 1 and 3.

The next column we will check is exang. That means will check the impact of exercises induced angina on heart disease:



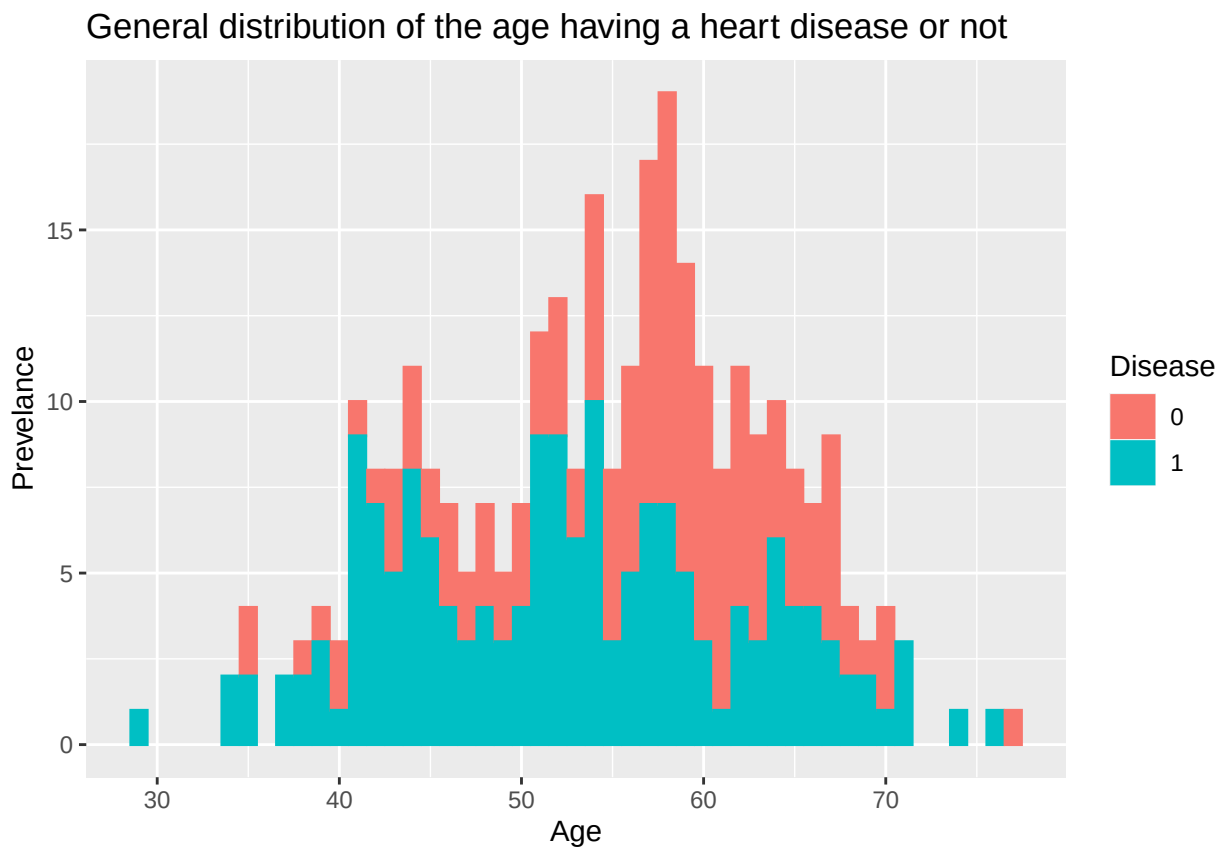
We can see that patients with an exercise induced angina were much more likely to have a heart disease.

We will also check the impact of fasting blood sugar on heart disease:



As we can see the impact of fasting blood sugar has nearly no impact because the bars are balanced whether the fasting blood sugar is low or high.

In this visualization we inspect the impact of the age on heart disease.



As expected higher ages had more heart disease cases than the lower ages. Especially in the range between 50 and 70 years we see increased spikes of heart disease cases.

Model selection

In the next step we need to prepare the dataset for the machine learning algorithms and choose the right model. We will split the heart dataset into a 80 / 20 set. That means we will use 80% of the data to train our models and 20% to validate it after we have a final model. We will also split the train dataset into a 80 / 20 split to test our models without using the validation set.

I choose the seven following models to test how they perform native on the dataset:

```
models <- c("glm", "lda", "rpart", "naive_bayes", "svmLinear", "knn", "rf")
```

GLM - Generalized liner model

LDA - Linear discriminant analysis

RPART - Classification and Regression Trees

Naive_Bayes - Naive bayes

svmLinear - Suport Vector Machine with a linear kernel

knn - K-Nearest-Neighbor

rf - Random Forest

```
## [1] "glm"
## [1] "lda"
## [1] "rpart"
## [1] "naive_bayes"
## [1] "svmLinear"
## [1] "knn"
## [1] "rf"
```

Variables	Accuracy
GLM in general	0.7142857
LDA in general	0.6938776
RPART in general	0.6326531
Naive bayes in general	0.7755102
SVM in general	0.7142857
KNN in general	0.5510204
RF in general	0.7346939

Based on the results I decided to take those models: naive_bayes, rf and GLM.

In the next step it is necessary to optimize all three selected models. That means finding good values for every models tuning parameters. The tuning parameters needs to get optimized to get the best performing model results.

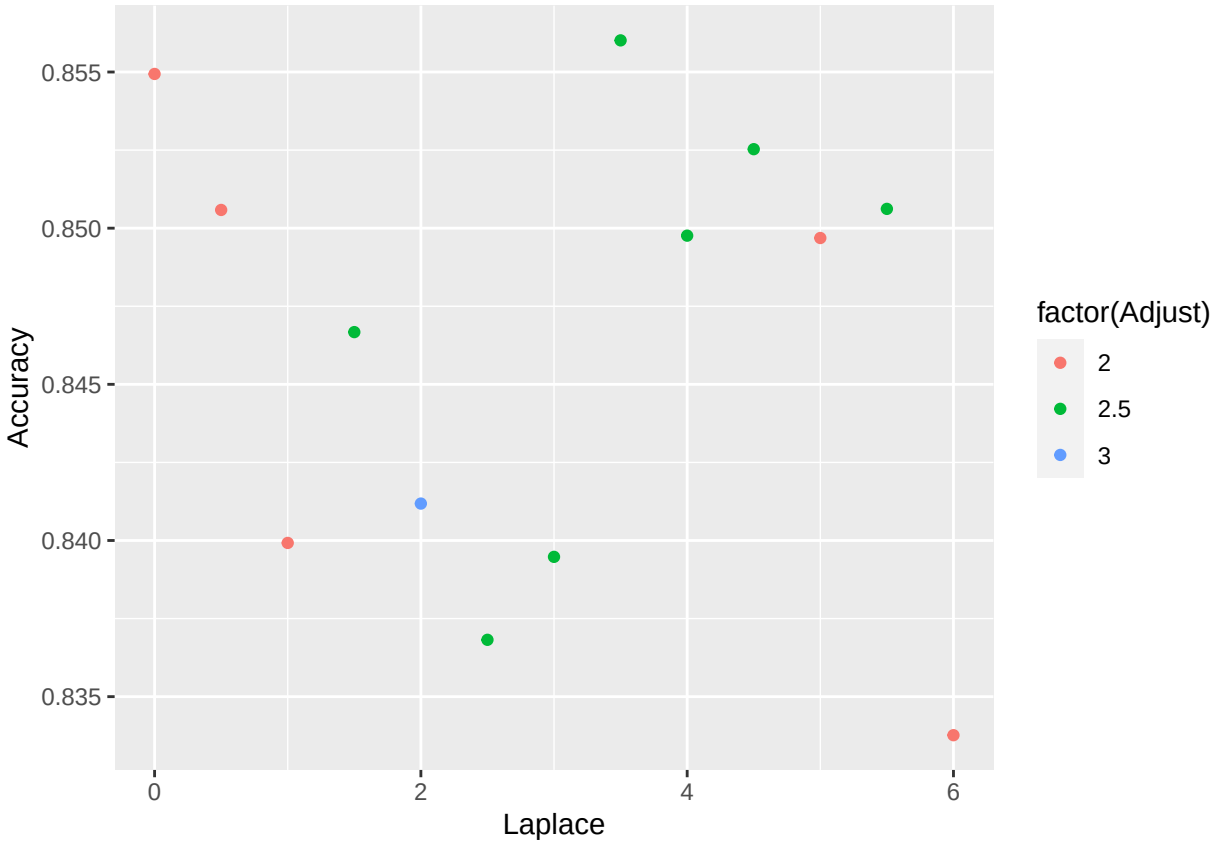
We will start first with the naive bayes model. The basic idea of this model is to predict heart disease using conditional probability. In this model we need to tune the parameters usekernel, laplace and adjust. The arguments are good for:

laplace: gives the amount of laplace smoothing. That means that it handles the problem of zero probaility in naive bayes.

usekernel: Allows to use kernel density estimation for numeric values instead of gaussian distribution.

adjust: Allows to adjust the flexibility of the kernel density estimate.

To get the best parameters we have to use them in combination to get the best value combination of the parameters.



As we can see the best parameter combination is:

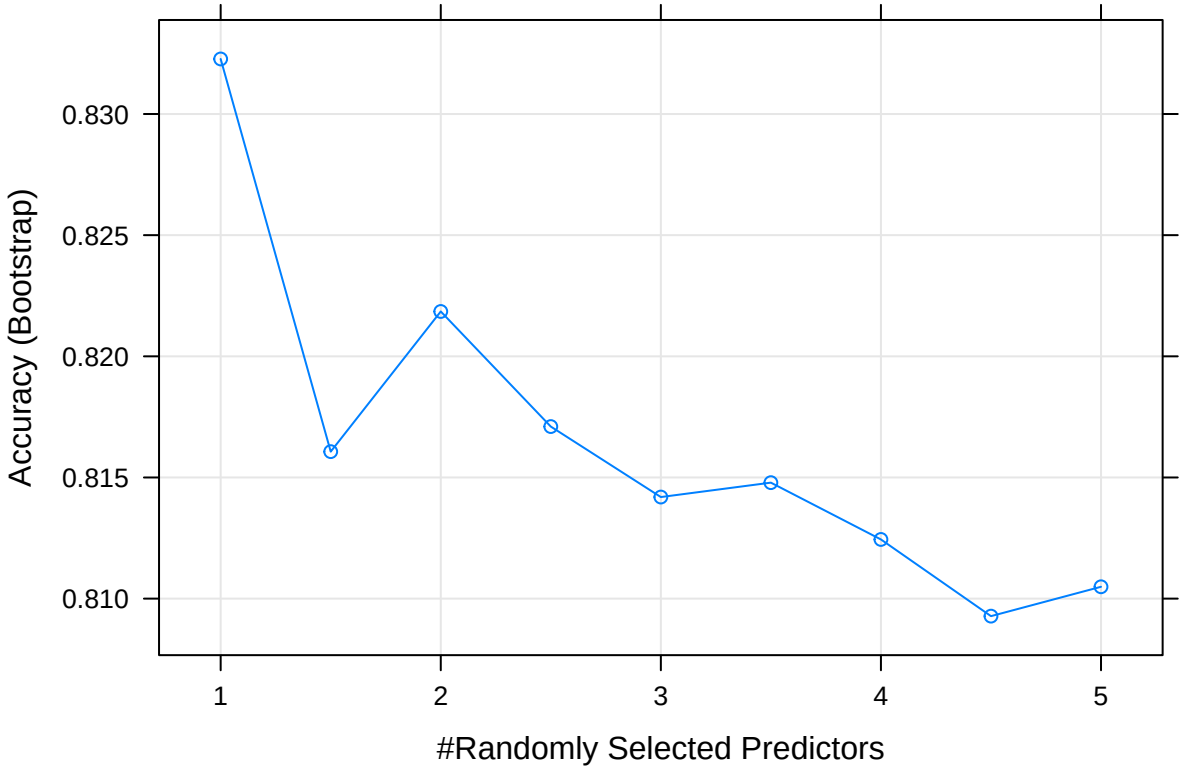
adjust: 2.5 and **laplace**: 3.5

So lets fit the naive bayes model with the optimized parameters again.

Variables	Accuracy
Naive Bayes after optimization	0.7755102

The next model we need to optimize is the random forest model.

The basic idea of the random forest model it is to build multiple decision trees which all use different predictors. They don't use the entire dataset but only different parts of it for each decision tree. In this model we need to tune the parameter mtry. The parameter mtry means how many splitcandidates are looked at at each decision tree.



As we can see the best parameter is:

mtry: 1

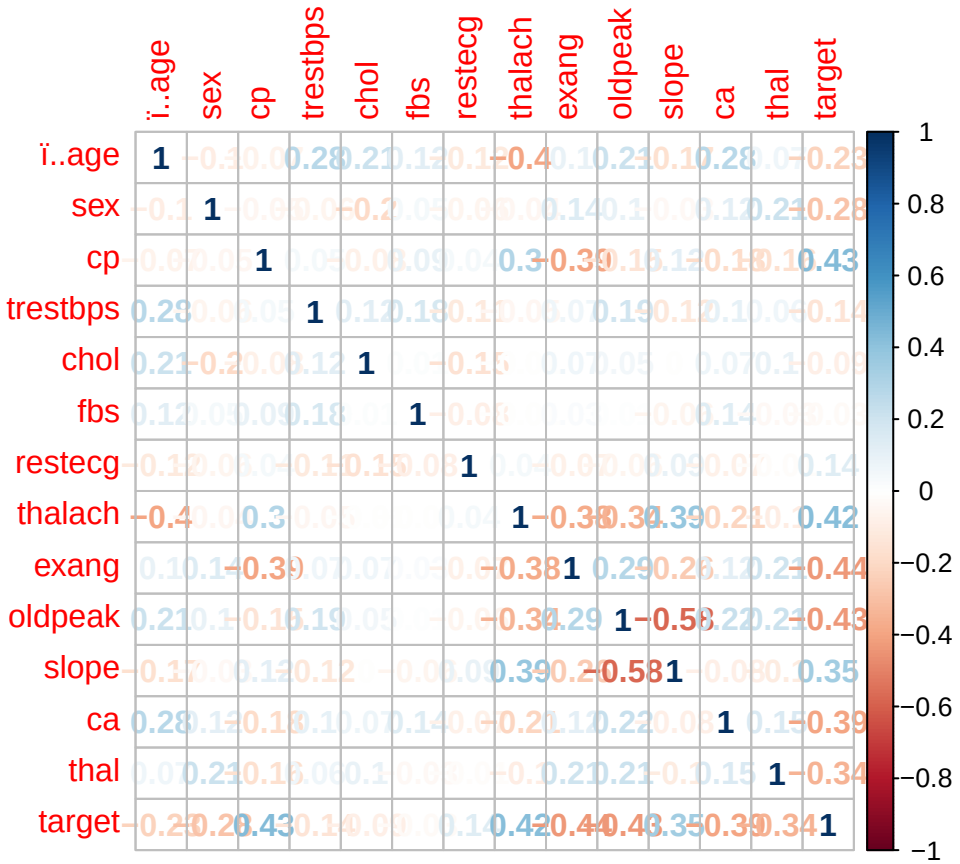
So lets fit the random forest model with the optimized parameters again:

Variables	Accuracy
Naive Bayes after optimization	0.7755102
Random forest after optimization	0.7551020

The next model is the generalized linear model. This model doesn't have any parameters to tune with the train function. The basic idea of the generalized linear model it is to predict values along a linear regression line.

Variables	Accuracy
Naive Bayes after optimization	0.7755102
Random forest after optimization	0.7551020
GLM after optimization	0.7142857

Another aspect to optimize the model is to check existing correlations and delete those columns not correlating with the target (heart disease) implying those columns do not add any valuable information to the model.



As we can see there are few cloumns (fbs, chol, trestbps and restecg) that do not correlate with the target (heart disease).

In the next step we will delete those columns to see if this improves the models. We will also check the ensemble method where we use the three improved models to gather a majority vote on the target value (heart disease).

Variables	Accuracy
Naive Bayes after optimization - correlation	0.7959184
Random Forest after optimization - correlation	0.7551020
GLM after optimization - correlation	0.7142857
Ensemble	0.7755102

Based on the Accuracy the Naive bayes model is the best performing model to predict heart disease.

Results

We saw already in analysis section that the Naive bayes model is the best performing model. To get the the final results we need to deploy the algorithm on the validation dataset.

Variables	Accuracy
Naive Bayes - final model	0.852459

The final accuracy is 0.852459.

Conclusion

As we can see our final model has a decent accuracy to predict heart disease in patients. Although the model would be more stable and better at predicting with more data in the dataset. Also while the Accuracy is decent our sensitivity to find the a heart disease is just okay.

As we saw in the graphs above some predictors do not really have an impact on the outcome, meaning the patient has a heart disease. One of these is the column fbs - Fasting Blood Sugar. Other columns, like exercise induced angina, increased the risk for having a heart disease substantially.

Feel free to network with me on LinkedIn: https://www.linkedin.com/in/mike-miemczok-432206209/

or Xing: https://www.xing.com/profile/Mike_Miemczok

Also check out my github for future work: https://github.com/mikemiemczok