Heart Attack Risk Prediction: Case Study

George Urgiles

Spring 2025

Load Libraries

Load & Clean Data

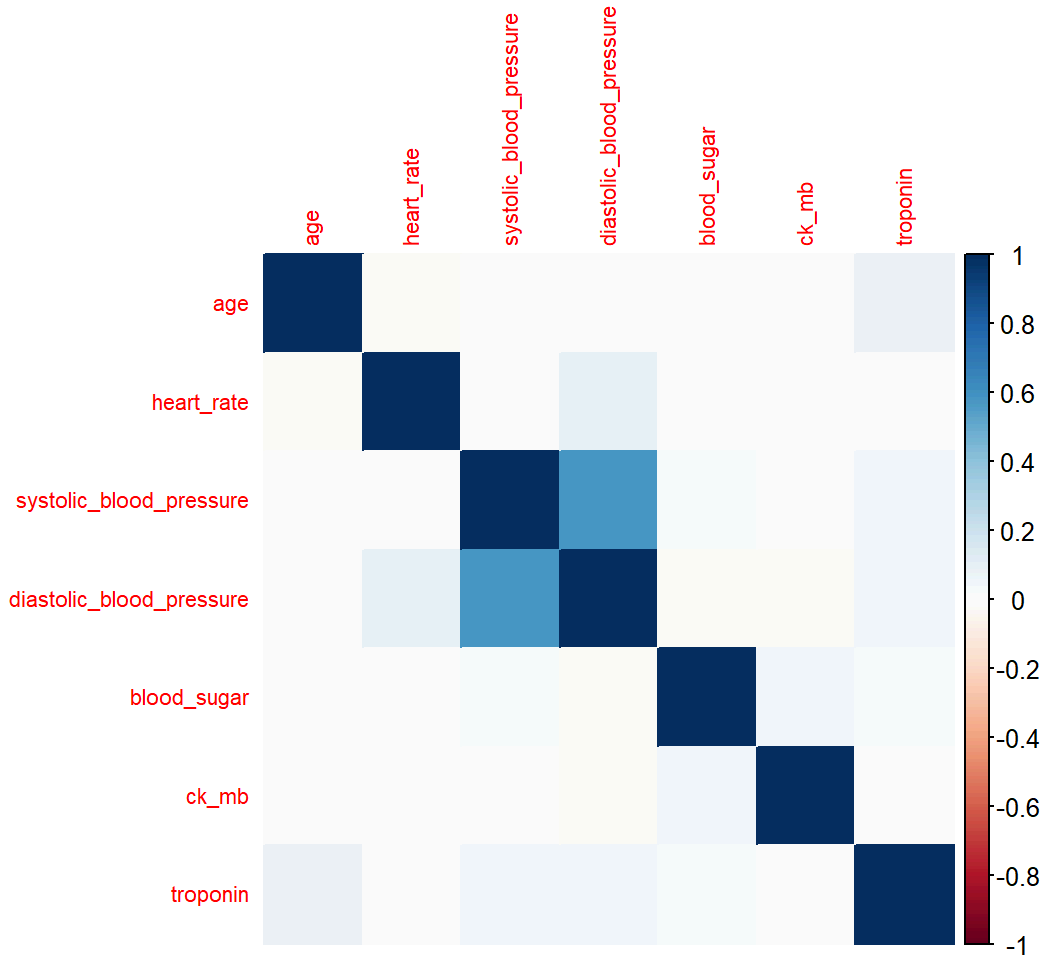
```
## Rows: 1,319
## Columns: 11
## $ age                      <int> 63, 20, 56, 66, 54, 52, 38, 61, 49, 65, 45, 6…
## $ gender                   <fct> Female, Female, Female, Female, Female, Male,…
## $ heart_rate               <int> 66, 94, 64, 70, 64, 61, 40, 60, 60, 61, 60, 6…
## $ systolic_blood_pressure  <int> 160, 98, 160, 120, 112, 112, 179, 214, 154, 1…
## $ diastolic_blood_pressure <int> 83, 46, 77, 55, 65, 58, 68, 82, 81, 95, 90, 8…
## $ blood_sugar              <dbl> 160, 296, 270, 270, 300, 87, 102, 87, 135, 10…
## $ ck_mb                    <dbl> 1.800, 6.750, 1.990, 13.870, 1.080, 1.830, 0.…
## $ troponin                 <dbl> 0.012, 1.060, 0.003, 0.122, 0.003, 0.004, 0.0…
## $ result                   <chr> "negative", "positive", "negative", "positive…
## $ risk_level               <fct> Moderate, High, Moderate, High, Moderate, Low…
## $ recommendation           <chr> "Monitor closely and consult doctor", "Immedi…
```

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

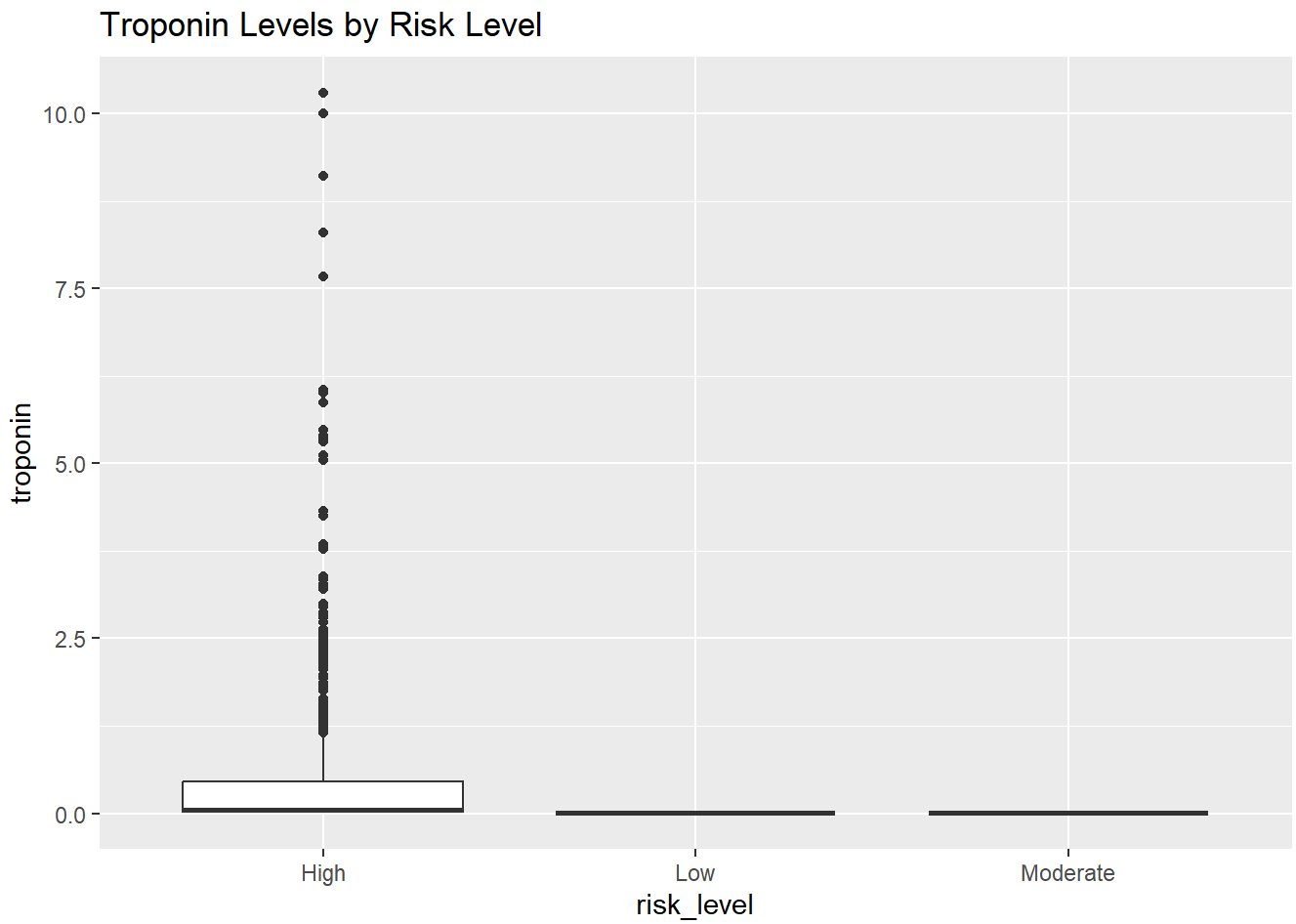
Descriptive Statistics

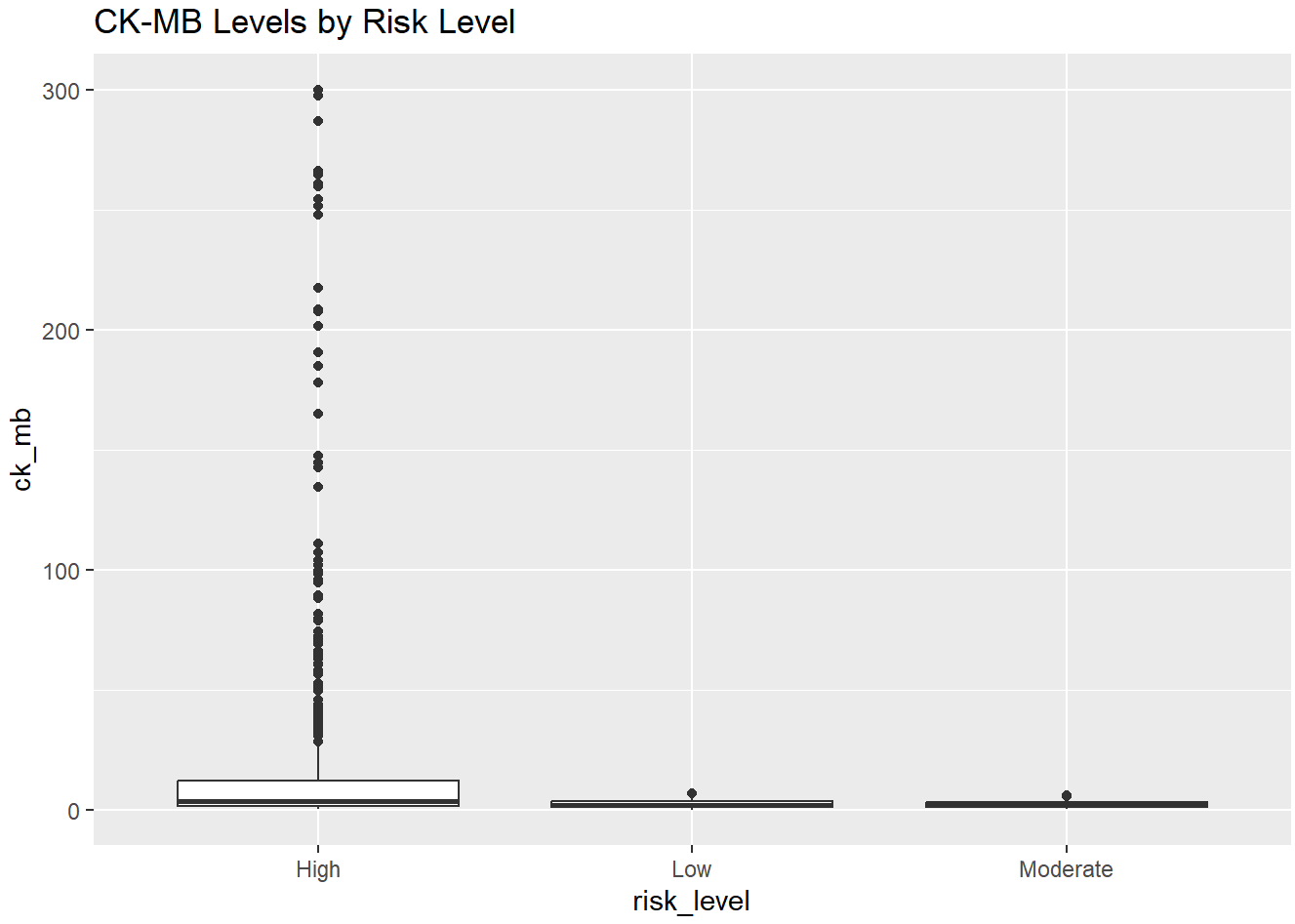
```
##       age            gender      heart_rate      systolic_blood_pressure
##  Min.   : 14.00   Male  :449   Min.   :  20.00   Min.   : 42.0
##  1st Qu.: 47.00   Female:870   1st Qu.:  64.00   1st Qu.:110.0
##  Median : 58.00                Median :  74.00   Median :124.0
##  Mean   : 56.19                Mean   :  78.34   Mean   :127.2
##  3rd Qu.: 65.00                3rd Qu.:  85.00   3rd Qu.:143.0
##  Max.   :103.00                Max.   :1111.00   Max.   :223.0
##  diastolic_blood_pressure  blood_sugar        ck_mb            troponin
##  Min.   : 38.00           Min.   : 35.0   Min.   :  0.321   Min.   : 0.0010
##  1st Qu.: 62.00           1st Qu.: 98.0   1st Qu.:  1.655   1st Qu.: 0.0060
##  Median : 72.00           Median :116.0   Median :  2.850   Median : 0.0140
##  Mean   : 72.27           Mean   :146.6   Mean   : 15.274   Mean   : 0.3609
##  3rd Qu.: 81.00           3rd Qu.:169.5   3rd Qu.:  5.805   3rd Qu.: 0.0855
##  Max.   :154.00           Max.   :541.0   Max.   :300.000   Max.   :10.3000
##     result             risk_level  recommendation
##  Length:1319        High    :812   Length:1319
##  Class :character   Low     :275   Class :character
##  Mode  :character   Moderate:232   Mode  :character
##
##
##
```

Correlation Matrix for Numeric Features



Boxplots for Key Variables


Troponin Levels by Risk Level
troponin
10.0
7.5
5.0
2.5
0.0
High
Low
Moderate
risk_level


CK-MB Levels by Risk Level
ck_mb
300
200
100
0
High
Low
Moderate
risk_level

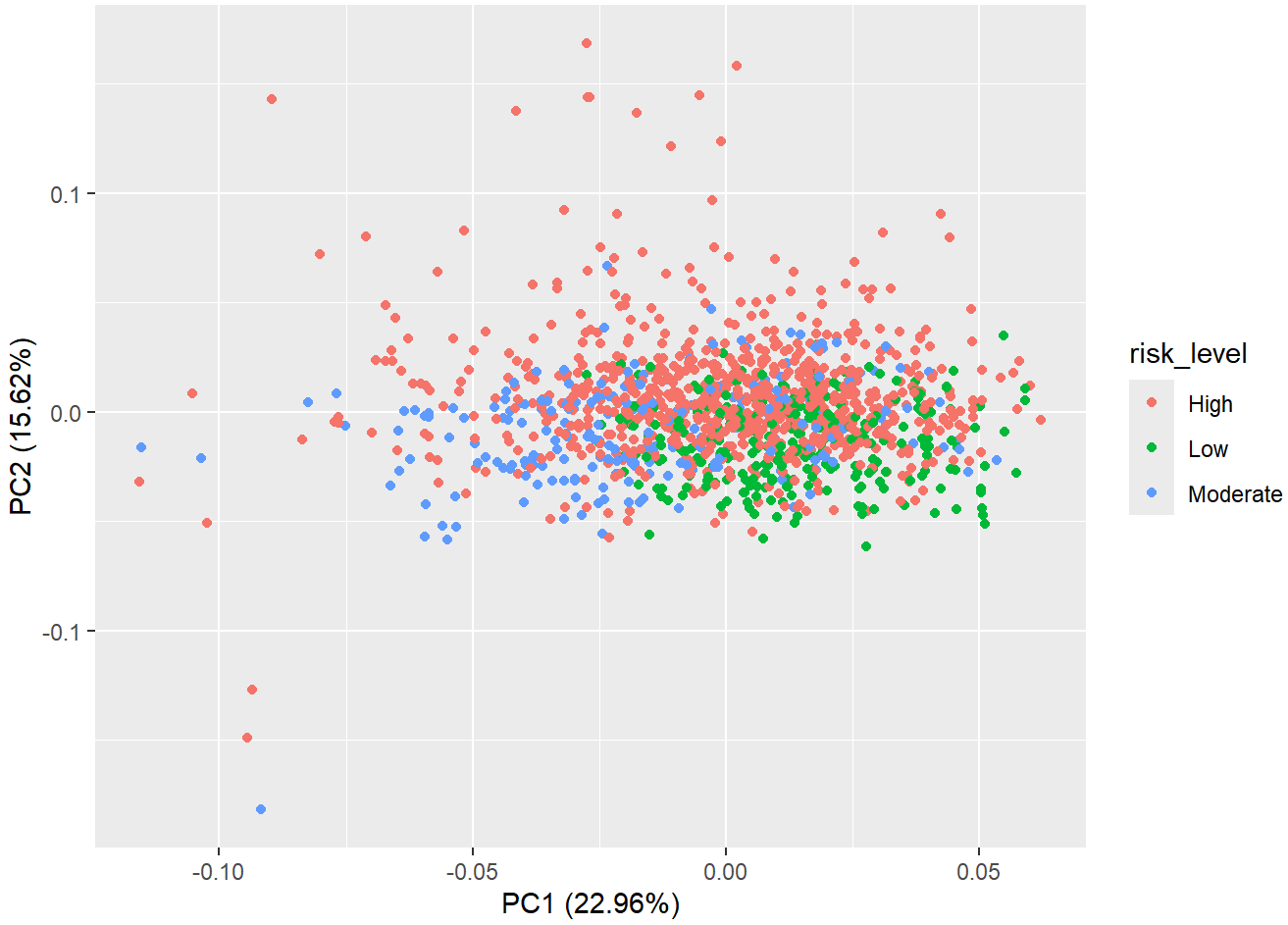
Explore Class Imbalance

```
##
##     High      Low Moderate
##      812      275      232
```

```
##
##      High       Low  Moderate
## 0.6156179 0.2084913 0.1758908
```

A pca plot a scatter plot used to visualize high-dimensional data by reducing it to two dimensions, typically using the first two principal components

##information accured from https://cran.r-project.org/web/packages/ggfortify/vignettes/plot_pca.html (https://cran.r-project.org/web/packages/ggfortify/vignettes/plot_pca.html)



Train/Test Split

Logistic Regression

```
## # weights:  39 (24 variable)
## initial  value 814.071706
## iter  10 value 668.559799
## iter  20 value 23.327156
## iter  30 value 0.117021
## iter  40 value 0.000714
## final  value 0.000000
## converged
## # weights:  39 (24 variable)
## initial  value 814.071706
## iter  10 value 668.559933
## iter  20 value 37.623591
## iter  30 value 17.367300
## iter  40 value 15.933406
## iter  50 value 15.839948
## iter  60 value 15.659851
## iter  70 value 15.476047
## final  value 15.476046
## converged
## # weights:  39 (24 variable)
## initial  value 814.071706
## iter  10 value 668.559799
## iter  20 value 23.342852
## iter  30 value 0.222027
## iter  40 value 0.128441
## iter  50 value 0.121685
## iter  60 value 0.104817
## iter  70 value 0.100121
## iter  80 value 0.088256
## iter  90 value 0.081425
## iter 100 value 0.077891
## final  value 0.077891
## stopped after 100 iterations
## # weights:  39 (24 variable)
## initial  value 812.973094
## iter  10 value 666.757718
## iter  20 value 9.767478
## iter  30 value 0.181654
## iter  40 value 0.007276
## final  value 0.000057
## converged
## # weights:  39 (24 variable)
## initial  value 812.973094
## iter  10 value 666.758074
## iter  20 value 28.311962
## iter  30 value 16.500780
## iter  40 value 15.933859
## iter  50 value 15.857314
## iter  60 value 15.593791
## iter  70 value 15.440272
## final  value 15.440270
## converged
```

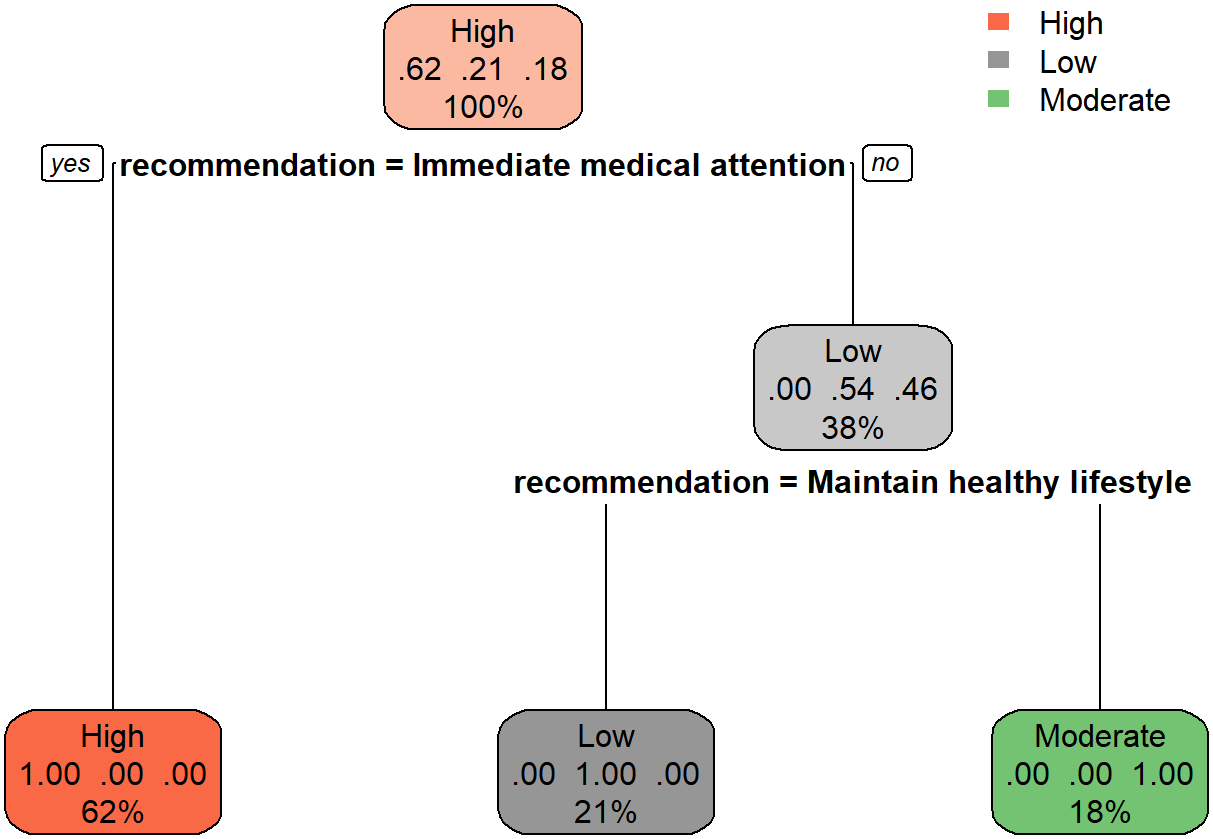
```
## # weights:  39 (24 variable)
## initial  value 812.973094
## iter  10 value 666.757719
## iter  20 value 9.788696
## iter  30 value 0.299149
## iter  40 value 0.155498
## iter  50 value 0.139292
## iter  60 value 0.117521
## iter  70 value 0.113731
## iter  80 value 0.098580
## iter  90 value 0.076976
## iter 100 value 0.074436
## final  value 0.074436
## stopped after 100 iterations
## # weights:  39 (24 variable)
## initial  value 812.973094
## iter  10 value 656.688596
## iter  20 value 22.887012
## iter  30 value 0.598455
## iter  40 value 0.038102
## iter  50 value 0.001155
## final  value 0.000072
## converged
## # weights:  39 (24 variable)
## initial  value 812.973094
## iter  10 value 656.689043
## iter  20 value 45.072813
## iter  30 value 16.781545
## iter  40 value 15.636221
## iter  50 value 15.541296
## iter  60 value 15.358635
## iter  70 value 15.108017
## final  value 15.107955
## converged
## # weights:  39 (24 variable)
## initial  value 812.973094
## iter  10 value 656.688596
## iter  20 value 22.908944
## iter  30 value 0.648340
## iter  40 value 0.134502
## iter  50 value 0.114544
## iter  60 value 0.095308
## iter  70 value 0.086058
## iter  80 value 0.077320
## iter  90 value 0.071393
## iter 100 value 0.064261
## final  value 0.064261
## stopped after 100 iterations
## # weights:  39 (24 variable)
## initial  value 812.973094
## iter  10 value 657.705773
## iter  20 value 33.546987
```

```
## iter  30 value 0.280284
## iter  40 value 0.001900
## final  value 0.000013
## converged
## # weights:  39 (24 variable)
## initial  value 812.973094
## iter  10 value 657.706288
## iter  20 value 40.958243
## iter  30 value 16.341525
## iter  40 value 15.875431
## iter  50 value 15.795469
## iter  60 value 15.593063
## iter  70 value 15.288777
## final  value 15.288741
## converged
## # weights:  39 (24 variable)
## initial  value 812.973094
## iter  10 value 657.705773
## iter  20 value 33.585404
## iter  30 value 0.338689
## iter  40 value 0.095347
## iter  50 value 0.090026
## iter  60 value 0.082117
## iter  70 value 0.078219
## iter  80 value 0.074256
## iter  90 value 0.073221
## iter 100 value 0.071257
## final  value 0.071257
## stopped after 100 iterations
## # weights:  39 (24 variable)
## initial  value 811.874481
## iter  10 value 657.421591
## iter  20 value 18.263124
## iter  30 value 0.038217
## iter  40 value 0.000109
## iter  40 value 0.000066
## iter  40 value 0.000050
## final  value 0.000050
## converged
## # weights:  39 (24 variable)
## initial  value 811.874481
## iter  10 value 657.422140
## iter  20 value 34.555598
## iter  30 value 16.588996
## iter  40 value 15.914183
## iter  50 value 15.846572
## iter  60 value 15.601860
## iter  70 value 15.402848
## final  value 15.402841
## converged
## # weights:  39 (24 variable)
## initial  value 811.874481
```

```
## iter  10 value 657.421592
## iter  20 value 18.280379
## iter  30 value 0.136197
## iter  40 value 0.116286
## iter  50 value 0.099147
## iter  60 value 0.090426
## iter  70 value 0.084151
## iter  80 value 0.075208
## iter  90 value 0.074157
## iter 100 value 0.071308
## final  value 0.071308
## stopped after 100 iterations
## # weights:  39 (24 variable)
## initial  value 1016.216367
## iter  10 value 814.402349
## iter  20 value 41.592581
## iter  30 value 0.429927
## iter  40 value 0.156357
## iter  50 value 0.138865
## iter  60 value 0.112163
## iter  70 value 0.108910
## iter  80 value 0.094251
## iter  90 value 0.089156
## iter 100 value 0.083781
## final  value 0.083781
## stopped after 100 iterations
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction High Low Moderate
##   High      243   0        0
##   Low         0  82        0
##   Moderate    0   0       69
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9907, 1)
##     No Information Rate : 0.6168
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: High Class: Low Class: Moderate
## Sensitivity               1.0000     1.0000          1.0000
## Specificity               1.0000     1.0000          1.0000
## Pos Pred Value            1.0000     1.0000          1.0000
## Neg Pred Value            1.0000     1.0000          1.0000
## Prevalence                0.6168     0.2081          0.1751
## Detection Rate            0.6168     0.2081          0.1751
## Detection Prevalence      0.6168     0.2081          0.1751
## Balanced Accuracy         1.0000     1.0000          1.0000
```

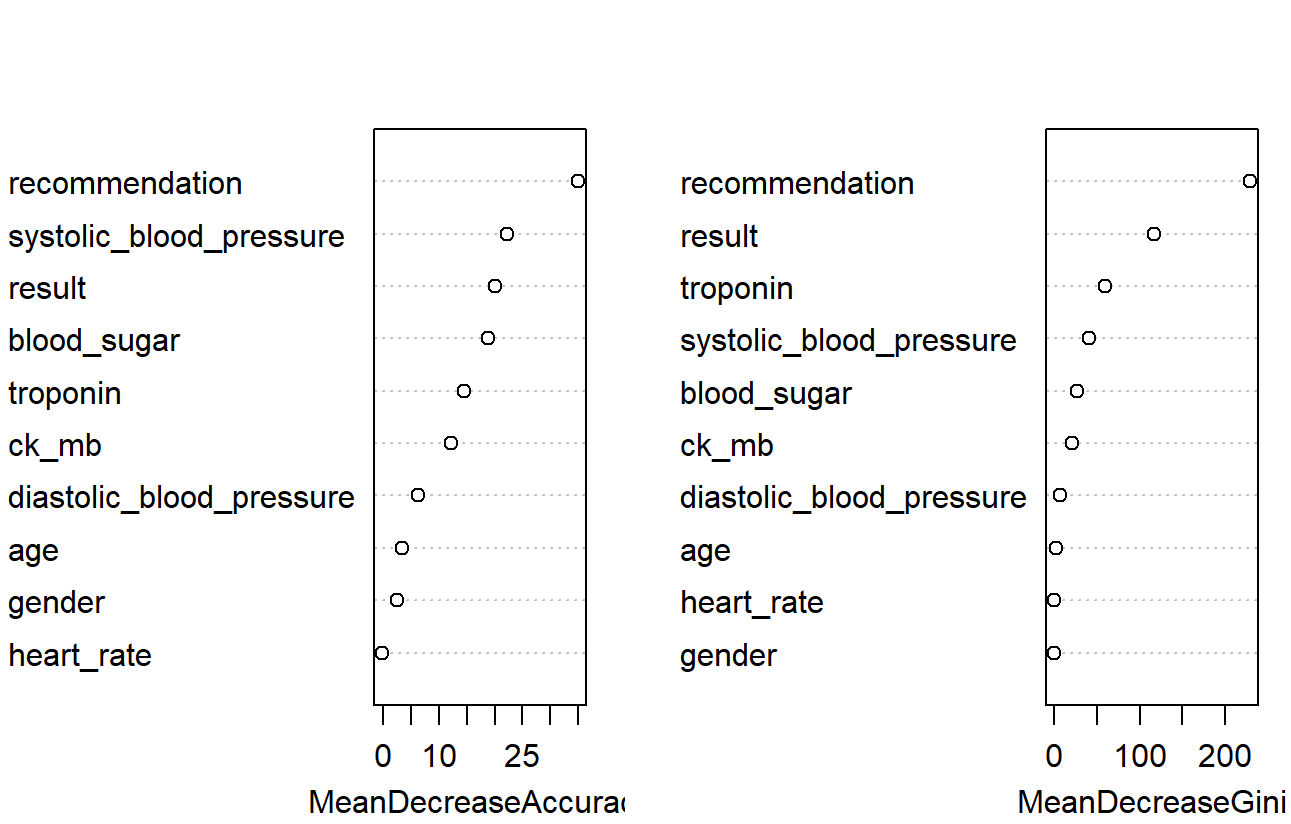
Decision Tree


High
Low
Moderate
High
.62 .21 .18
100%
yes
recommendation = Immediate medical attention
no
Low
.00 .54 .46
38%
recommendation = Maintain healthy lifestyle
High
1.00 .00 .00
62%
Low
.00 1.00 .00
21%
Moderate
.00 .00 1.00
18%

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction High Low Moderate
##   High      243   0        0
##   Low         0  82        0
##   Moderate    0   0       69
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9907, 1)
##     No Information Rate : 0.6168
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: High Class: Low Class: Moderate
## Sensitivity               1.0000     1.0000          1.0000
## Specificity               1.0000     1.0000          1.0000
## Pos Pred Value            1.0000     1.0000          1.0000
## Neg Pred Value            1.0000     1.0000          1.0000
## Prevalence                0.6168     0.2081          0.1751
## Detection Rate            0.6168     0.2081          0.1751
## Detection Prevalence      0.6168     0.2081          0.1751
## Balanced Accuracy         1.0000     1.0000          1.0000
```

Random Forest

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction High Low Moderate
##   High      243   0        0
##   Low         0  82        0
##   Moderate    0   0       69
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9907, 1)
##     No Information Rate : 0.6168
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: High Class: Low Class: Moderate
## Sensitivity               1.0000     1.0000          1.0000
## Specificity               1.0000     1.0000          1.0000
## Pos Pred Value            1.0000     1.0000          1.0000
## Neg Pred Value            1.0000     1.0000          1.0000
## Prevalence                0.6168     0.2081          0.1751
## Detection Rate            0.6168     0.2081          0.1751
## Detection Prevalence      0.6168     0.2081          0.1751
## Balanced Accuracy         1.0000     1.0000          1.0000
```



Model Comparison

```
##                 Model Accuracy
## 1 Logistic Regression        1
## 2       Decision Tree        1
## 3       Random Forest        1
```

Conclusion

We explored and modeled heart attack risk levels using logistic regression, decision trees, and random forests. We enhanced exploration with correlation plots, boxplots, and PCA to visualize data distribution. Random forests typically performed best, underlining the importance of key biomarkers like Troponin and CK-MB. Future work can include advanced tuning or deeper medical insights.