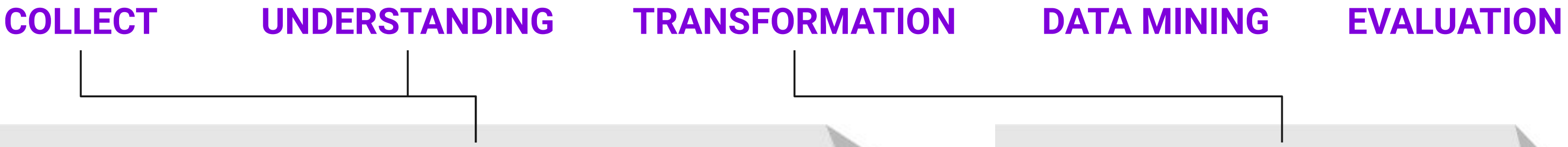
Heart Disease Causes Analysis

Jaeweon Kim

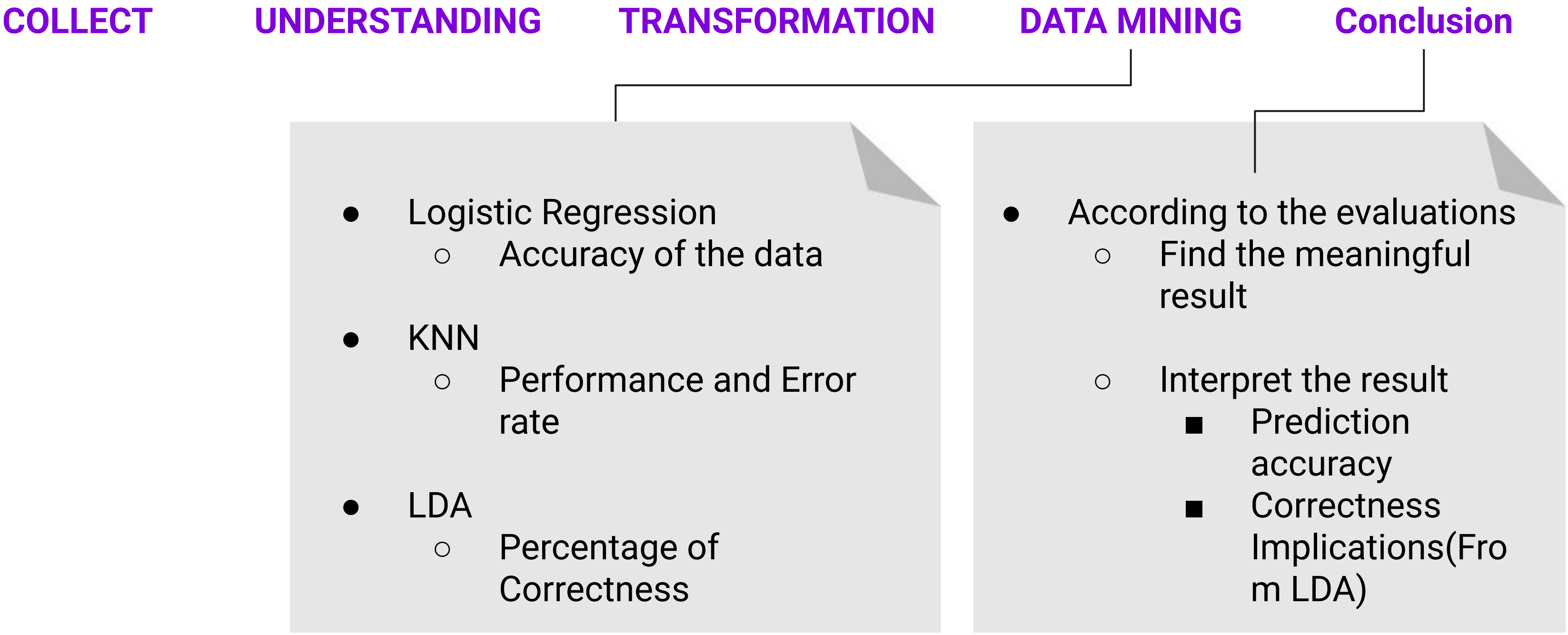
Introduction



- Heart Disease Data Set
  - Abstract: 4 databases: Cleveland, Hungary, Switzerland, and the VA Long Beach
  - Number of Instances: 303
  - Attribute Characteristics: Categorical, Integer, Real
  - Database contains 76 attributes, but the published experiment dataset refer to using a subset of 14 of them.
- Through the 14 attributes, we decide the predicted attribute, target value from 0 and 1.

- Backward Stepwise Processing
  - Determine relevant independent variables from the data.

Introduction



COLLECT UNDERSTANDING TRANSFORMATION DATA MINING Conclusion

- Logistic Regression
  - Accuracy of the data
- KNN
  - Performance and Error rate
- LDA
  - Percentage of Correctness

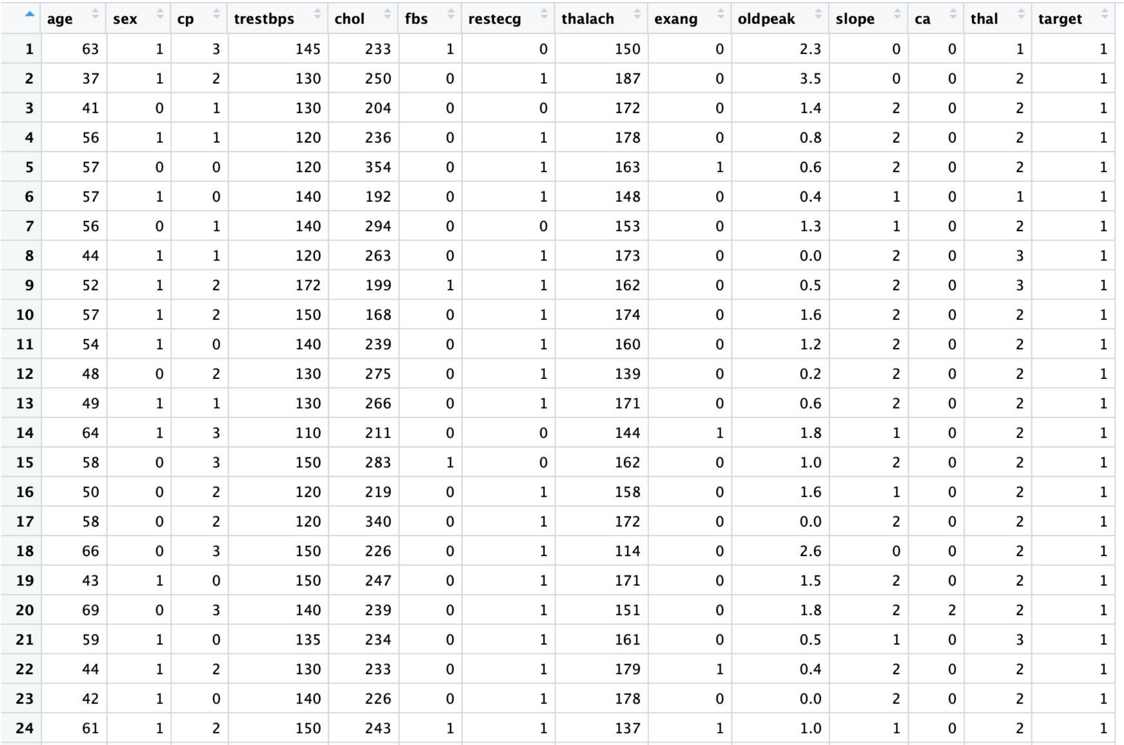
- According to the evaluations
  - Find the meaningful result
  - Interpret the result
    - Prediction accuracy
    - Correctness Implications(From LDA)

Data Description

303 observations

13 factors: age, sex, chest pain type, resting blood pressure, serum cholesterol, fasting blood sugar, resting electrocardiographic results, max heart rate, exercise induced angina, oldpeak, the slope of the peak exercise, number of major vessels, thal

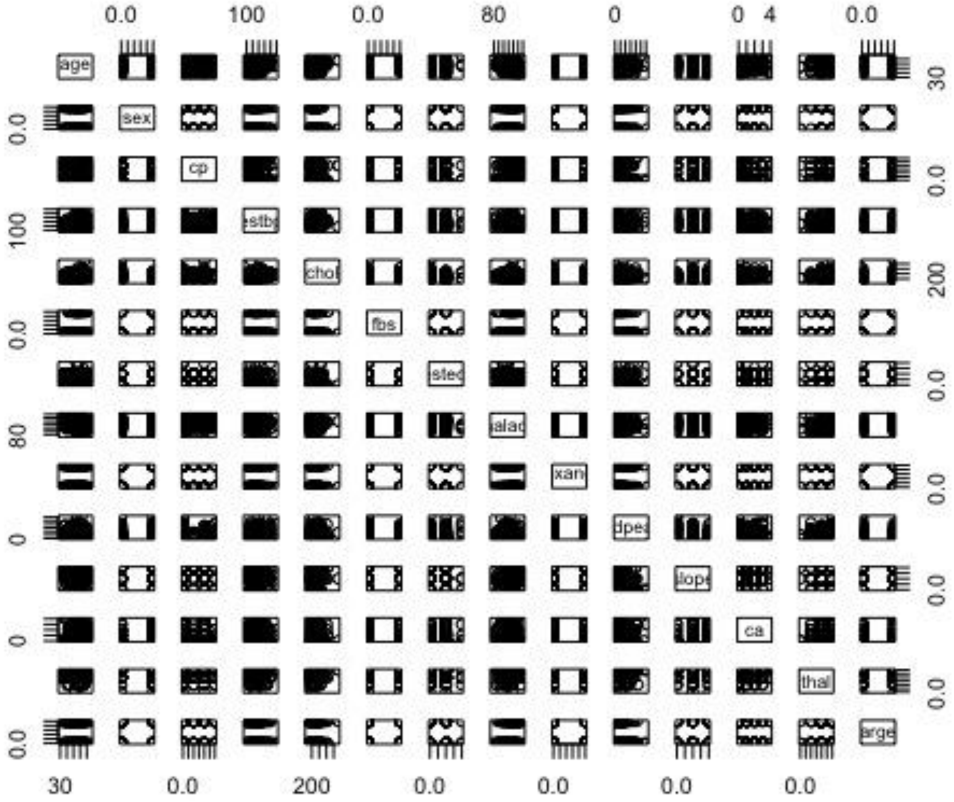
1: disease   0: healthy



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
1	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
2	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
3	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
4	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
5	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
6	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
7	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
8	44	1	1	120	263	0	1	173	0	0.0	2	0	3	1
9	52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
10	57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
11	54	1	0	140	239	0	1	160	0	1.2	2	0	2	1
12	48	0	2	130	275	0	1	139	0	0.2	2	0	2	1
13	49	1	1	130	266	0	1	171	0	0.6	2	0	2	1
14	64	1	3	110	211	0	0	144	1	1.8	1	0	2	1
15	58	0	3	150	283	1	0	162	0	1.0	2	0	2	1
16	50	0	2	120	219	0	1	158	0	1.6	1	0	2	1
17	58	0	2	120	340	0	1	172	0	0.0	2	0	2	1
18	66	0	3	150	226	0	1	114	0	2.6	0	0	2	1
19	43	1	0	150	247	0	1	171	0	1.5	2	0	2	1
20	69	0	3	140	239	0	1	151	0	1.8	2	2	2	1
21	59	1	0	135	234	0	1	161	0	0.5	1	0	3	1
22	44	1	2	130	233	0	1	179	1	0.4	2	0	2	1
23	42	1	0	140	226	0	1	178	0	0.0	2	0	2	1
24	61	1	2	150	243	1	1	137	1	1.0	1	0	2	1

Data Description

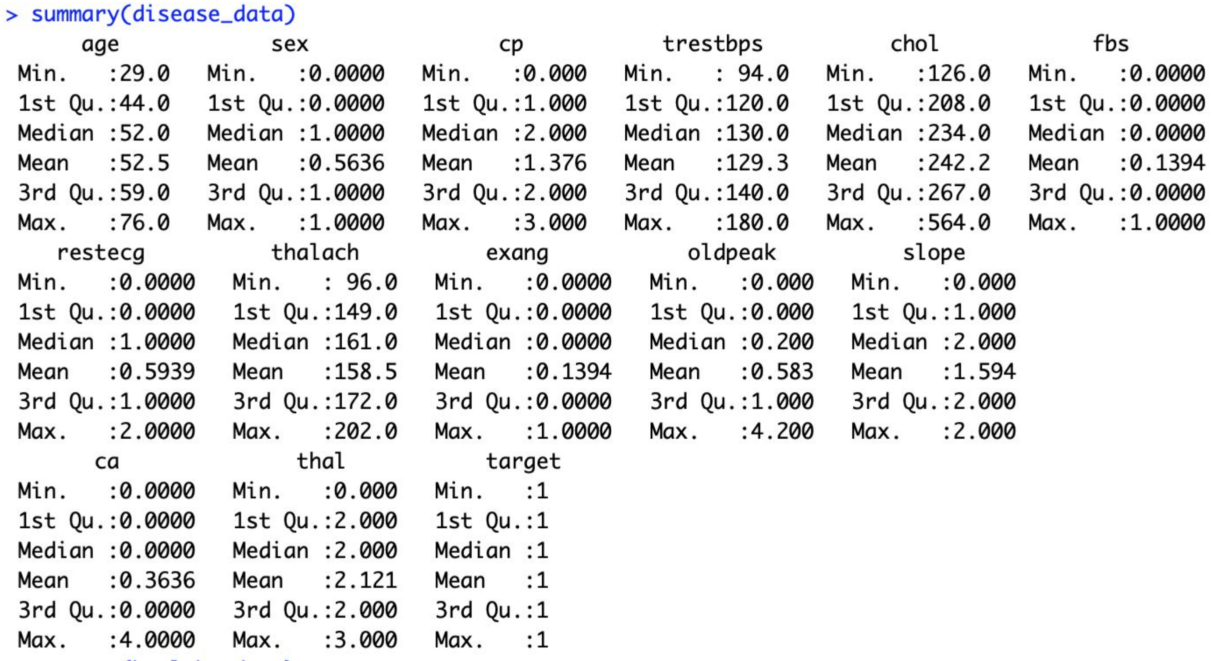
```
> pairs(heart)

> dim(heart)

[1] 303  14
```



Data Description

disease_data <- heart[1:165,]

summary(disease_data)

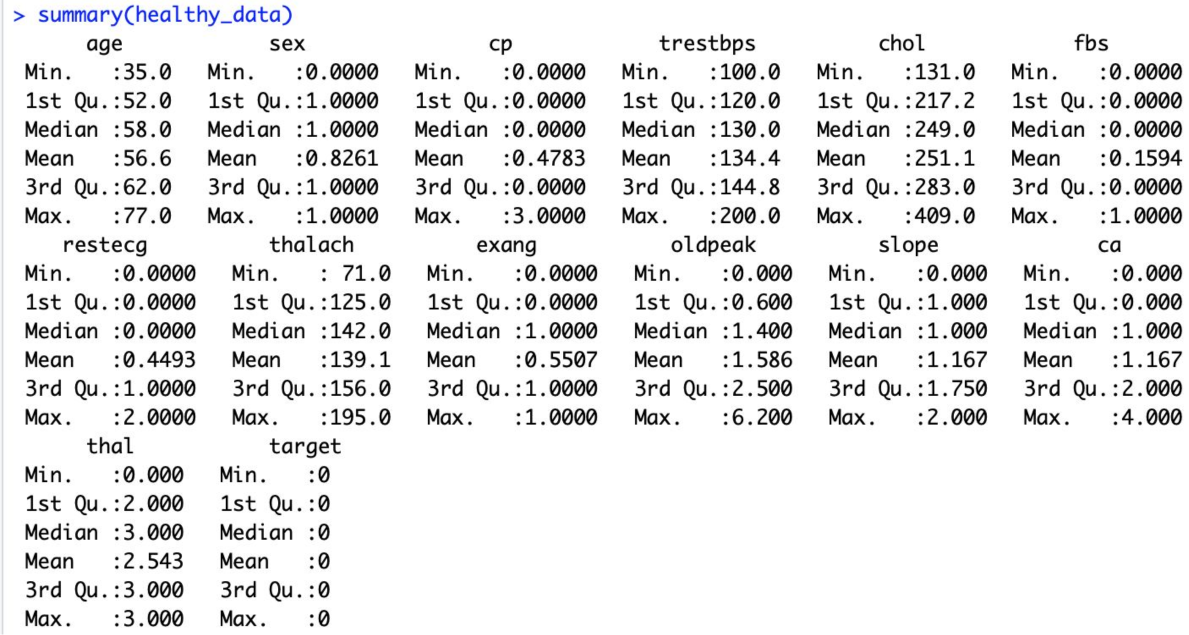


```
> summary(disease_data)
      age             sex               cp            trestbps          chol            fbs        
 Min.   :29.0   Min.   :0.0000   Min.   :0.000   Min.   : 94.0   Min.   :126.0   Min.   :0.0000  
 1st Qu.:44.0   1st Qu.:0.0000   1st Qu.:1.000   1st Qu.:120.0   1st Qu.:208.0   1st Qu.:0.0000  
 Median :52.0   Median :1.0000   Median :2.000   Median :130.0   Median :234.0   Median :0.0000  
 Mean   :52.5   Mean   :0.5636   Mean   :1.376   Mean   :129.3   Mean   :242.2   Mean   :0.1394  
 3rd Qu.:59.0   3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:140.0   3rd Qu.:267.0   3rd Qu.:0.0000  
 Max.   :76.0   Max.   :1.0000   Max.   :3.000   Max.   :180.0   Max.   :564.0   Max.   :1.0000  
    restecg          thalach          exang           oldpeak          slope      
 Min.   :0.0000   Min.   : 96.0   Min.   :0.0000   Min.   :0.000   Min.   :0.000  
 1st Qu.:0.0000   1st Qu.:149.0   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:1.000  
 Median :1.0000   Median :161.0   Median :0.0000   Median :0.200   Median :2.000  
 Mean   :0.5939   Mean   :158.5   Mean   :0.1394   Mean   :0.583   Mean   :1.594  
 3rd Qu.:1.0000   3rd Qu.:172.0   3rd Qu.:0.0000   3rd Qu.:1.000   3rd Qu.:2.000  
 Max.   :2.0000   Max.   :202.0   Max.   :1.0000   Max.   :4.200   Max.   :2.000  
       ca              thal           target 
 Min.   :0.0000   Min.   :0.000   Min.   :1  
 1st Qu.:0.0000   1st Qu.:2.000   1st Qu.:1  
 Median :0.0000   Median :2.000   Median :1  
 Mean   :0.3636   Mean   :2.121   Mean   :1  
 3rd Qu.:0.0000   3rd Qu.:2.000   3rd Qu.:1  
 Max.   :4.0000   Max.   :3.000   Max.   :1  
```

Data Description

healthy_data <- heart[166:303,]

summary(healthy_data)



```
> summary(healthy_data)
      age             sex               cp            trestbps          chol            fbs        
 Min.   :35.0   Min.   :0.0000   Min.   :0.0000   Min.   :100.0   Min.   :131.0   Min.   :0.0000  
 1st Qu.:52.0   1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.:120.0   1st Qu.:217.2   1st Qu.:0.0000  
 Median :58.0   Median :1.0000   Median :0.0000   Median :130.0   Median :249.0   Median :0.0000  
 Mean   :56.6   Mean   :0.8261   Mean   :0.4783   Mean   :134.4   Mean   :251.1   Mean   :0.1594  
 3rd Qu.:62.0   3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:144.8   3rd Qu.:283.0   3rd Qu.:0.0000  
 Max.   :77.0   Max.   :1.0000   Max.   :3.0000   Max.   :200.0   Max.   :409.0   Max.   :1.0000  
    restecg          thalach           exang           oldpeak          slope             ca       
 Min.   :0.0000   Min.   : 71.0   Min.   :0.0000   Min.   :0.000   Min.   :0.000   Min.   :0.000  
 1st Qu.:0.0000   1st Qu.:125.0   1st Qu.:0.0000   1st Qu.:0.600   1st Qu.:1.000   1st Qu.:0.000  
 Median :0.0000   Median :142.0   Median :1.0000   Median :1.400   Median :1.000   Median :1.000  
 Mean   :0.4493   Mean   :139.1   Mean   :0.5507   Mean   :1.586   Mean   :1.167   Mean   :1.167  
 3rd Qu.:1.0000   3rd Qu.:156.0   3rd Qu.:1.0000   3rd Qu.:2.500   3rd Qu.:1.750   3rd Qu.:2.000  
 Max.   :2.0000   Max.   :195.0   Max.   :1.0000   Max.   :6.200   Max.   :2.000   Max.   :4.000  
      thal           target 
 Min.   :0.000   Min.   :0  
 1st Qu.:2.000   1st Qu.:0  
 Median :3.000   Median :0  
 Mean   :2.543   Mean   :0  
 3rd Qu.:3.000   3rd Qu.:0  
 Max.   :3.000   Max.   :0  
```

Data Description

Splitting the data into training and testing:

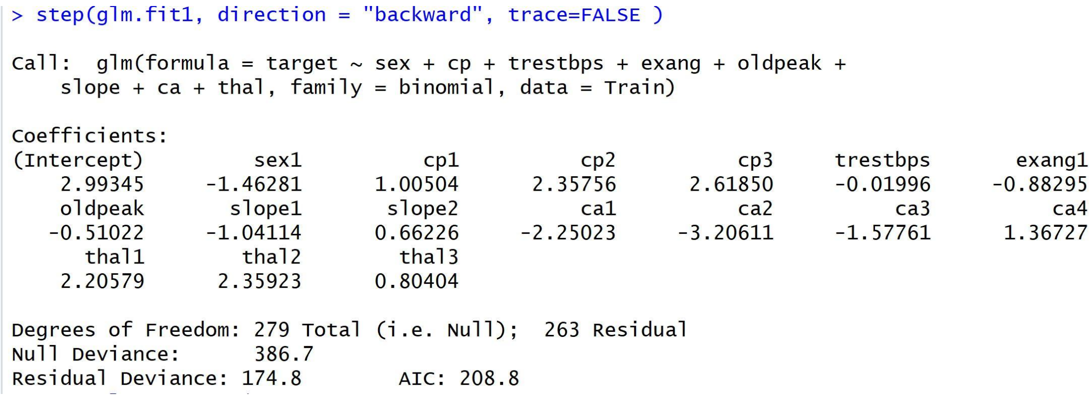
```
sampledata <- sample(c(1:303),280,replace = FALSE, prob = NULL)

Train <- heart[c(sampledata),]

Test <- heart[-c(sampledata),]
```

Logistic Regression

> glm.fit1 = glm(target~age+sex+cp+trestbps+chol+fbs+restecg+thalach+exang+oldpeak+slope+ca+thal, data=Train, family = binomial)

>step(glm.fit1, direction = "backward", trace=FALSE)

#we get final model of glm as target ~ sex + cp + trestbps + exang + oldpeak + slope + ca + thal
>glm.final <- glm(formula = target ~ sex + cp + trestbps + exang + oldpeak + slope + ca + thal, family = binomial, data = Train)
#AIC = 208.8



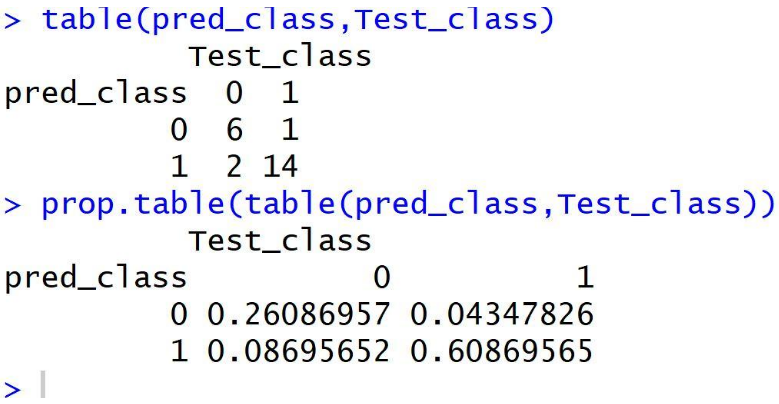
```
> step(glm.fit1, direction = "backward", trace=FALSE )

Call:  glm(formula = target ~ sex + cp + trestbps + exang + oldpeak + 
    slope + ca + thal, family = binomial, data = Train)

Coefficients:
(Intercept)         sex1          cp1          cp2          cp3     trestbps       exang1  
    2.99345     -1.46281      1.00504      2.35756      2.61850     -0.01996     -0.88295  
    oldpeak       slope1       slope2          ca1          ca2          ca3          ca4  
   -0.51022     -1.04114      0.66226     -2.25023     -3.20611     -1.57761      1.36727  
      thal1        thal2        thal3  
    2.20579      2.35923      0.80404  

Degrees of Freedom: 279 Total (i.e. Null);  263 Residual
Null Deviance:      386.7 
Residual Deviance: 174.8        AIC: 208.8
```

Logistic Regression

> From the result, the model is 86.94% correct overall

> The model has 95.35% accuracy for identifying people with disease

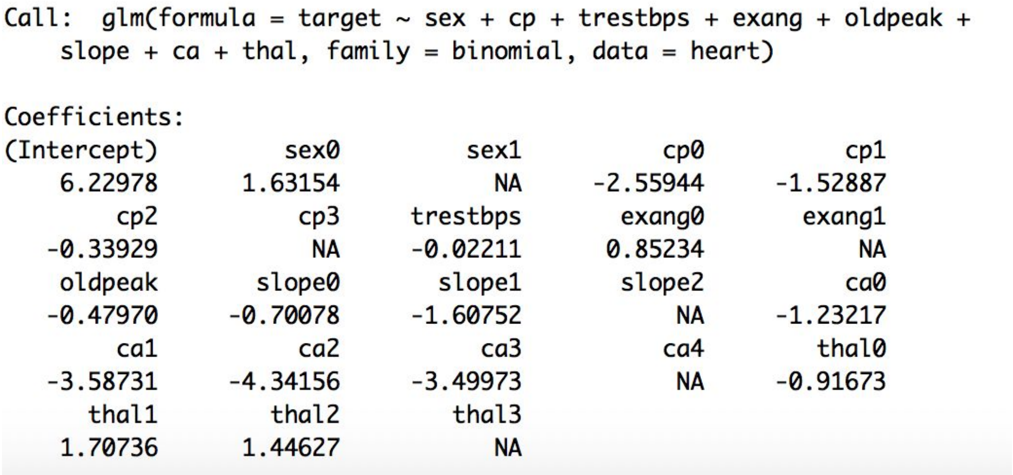
> The model has 76% accuracy for identifying people without disease



```
> table(pred_class,Test_class)
          Test_class
pred_class  0  1
         0  6  1
         1  2 14
> prop.table(table(pred_class,Test_class))
          Test_class
pred_class          0          1
         0 0.26086957 0.04347826
         1 0.08695652 0.60869565
> |
```

KNN

Step 1: Selecting significant variables using Backward selection



```
Call:  glm(formula = target ~ sex + cp + trestbps + exang + oldpeak +
    slope + ca + thal, family = binomial, data = heart)

Coefficients:
(Intercept)         sex0         sex1          cp0          cp1
    6.22978      1.63154           NA     -2.55944     -1.52887
        cp2          cp3     trestbps       exang0       exang1
   -0.33929           NA     -0.02211      0.85234           NA
    oldpeak       slope0       slope1       slope2          ca0
   -0.47970     -0.70078     -1.60752           NA     -1.23217
        ca1          ca2          ca3          ca4        thal0
   -3.58731     -4.34156     -3.49973           NA     -0.91673
      thal1        thal2        thal3
    1.70736      1.44627           NA
```

KNN

Step 2: Find the best K

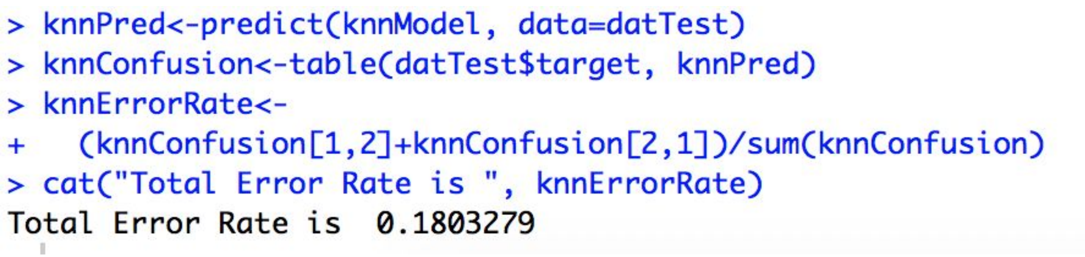
```
Outer Loop:for (k in c(1:floor(0.8*nrow(heart))
    Inner Loop: repeat ı times
        Choose a new training and testing set, Fit a knn model with k = k
        Calculate error rate/
    Calculate average error rate for each k/
```

Best K obtained is 21

KNN

How well the model perform?

- Error rate in testing is 0.18



```
> knnPred<-predict(knnModel, data=datTest)
> knnConfusion<-table(datTest$target, knnPred)
> knnErrorRate<-
+   (knnConfusion[1,2]+knnConfusion[2,1])/sum(knnConfusion)
> cat("Total Error Rate is ", knnErrorRate)
Total Error Rate is  0.1803279
```

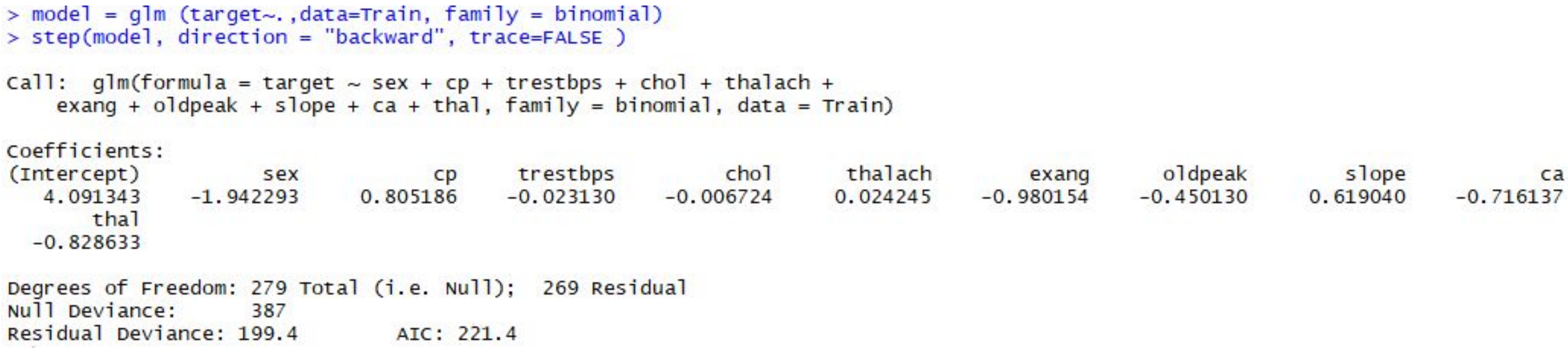
KNN

How we evaluate the result?

- $N << 2^d$, where $N = 303$, $d = 8$ ⇒ sample is not big enough
- The result of the best K is not always consistent

LDA

Step 1: Selecting significant variables using Backward selection

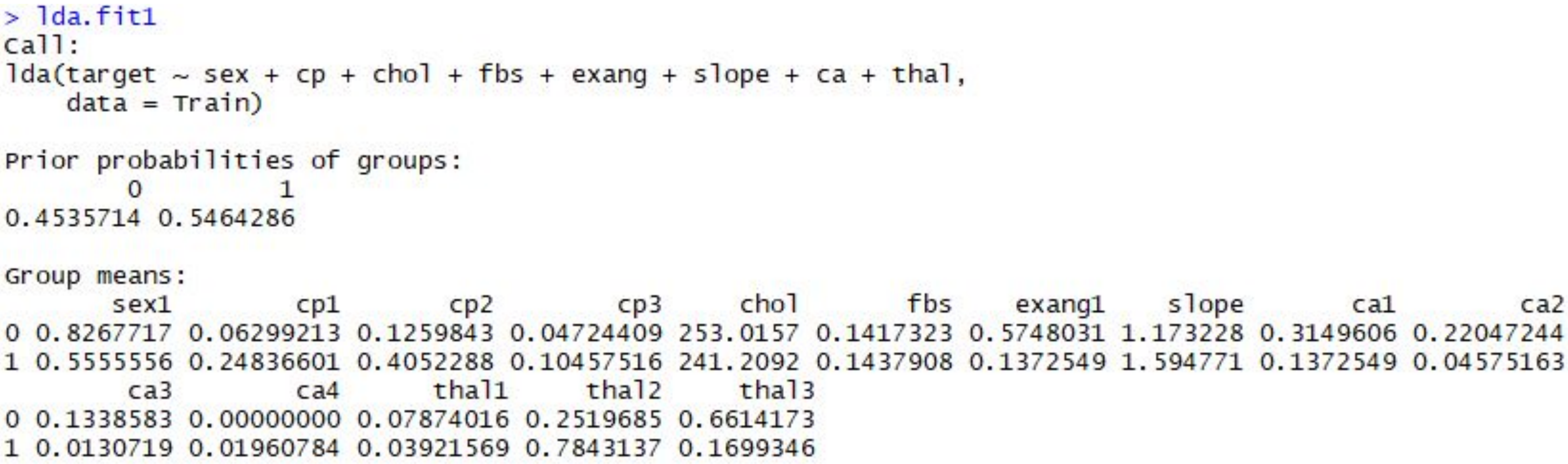


```
> model = glm (target~.,data=Train, family = binomial)
> step(model, direction = "backward", trace=FALSE )

Call:  glm(formula = target ~ sex + cp + trestbps + chol + thalach + 
    exang + oldpeak + slope + ca + thal, family = binomial, data = Train)

Coefficients:
(Intercept)          sex           cp     trestbps         chol      thalach        exang      oldpeak        slope           ca  
   4.091343    -1.942293     0.805186    -0.023130    -0.006724     0.024245    -0.980154    -0.450130     0.619040    -0.716137  
       thal  
  -0.828633  

Degrees of Freedom: 279 Total (i.e. Null);  269 Residual
Null Deviance:	    387 
Residual Deviance: 199.4 	AIC: 221.4
```

LDA

Step 2: Apply LDA

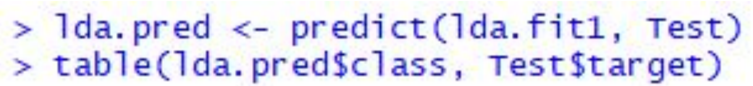
library(MASS)
> lda.fit1 <- lda(target~sex+cp+chol+fbs+exang+slope+ca+thal, data=Train)



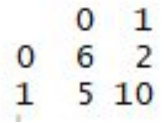
```
> lda.fit1
Call:
lda(target ~ sex + cp + chol + fbs + exang + slope + ca + thal,
    data = Train)

Prior probabilities of groups:
        0         1 
0.4535714 0.5464286 

Group means:
       sex1        cp1       cp2        cp3     chol       fbs    exang1    slope       ca1        ca2
0 0.8267717 0.06299213 0.1259843 0.04724409 253.0157 0.1417323 0.5748031 1.173228 0.3149606 0.22047244
1 0.5555556 0.24836601 0.4052288 0.10457516 241.2092 0.1437908 0.1372549 1.594771 0.1372549 0.04575163
        ca3        ca4      thal1     thal2     thal3
0 0.1338583 0.00000000 0.07874016 0.2519685 0.6614173
1 0.0130719 0.01960784 0.03921569 0.7843137 0.1699346
```

LDA

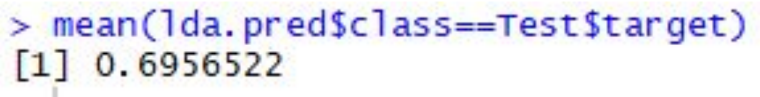
Step 2: Generate table and the result



```
> lda.pred <- predict(lda.fit1, Test)
> table(lda.pred$class, Test$target)
```



```
    0  1
0   6  2
1   5 10
```



```
> mean(lda.pred$class==Test$target)
[1] 0.6956522
```

- This suggests that LDA predictions are accurate around 69.57% of the time, which is much lower than the previous two model.
- The test error rate is 30.43%, which is too high among the heart disease tests
- These results are very different from those obtained with the logistic regression model.

Results

- Correctness Implications from each correctness scenarios in LDA result table
  - Percentage of correct predictions: 69.56%
  - The correctness of Target is right most of the time by 83.33%
  - The correctness of Target is wrong most of the time by 45.45%
  - Since the best correctness rate and the worst correctness rate have a big difference, we can conclude that this database model is not sufficient enough and LDA method is not the best result model among the types of methods we used.

Results

- Significant factors that may lead to heart disease (by backward selection)
  - sex, chest pain type, resting blood pressure, resting electrocardiographic, thalach, exercise induced angina, oldpeak, number of major vessels
-

Models	Accuracy	Comments
Logistic Regression	86.94%	Works well when response variable is Binary and sample size is small
KNN	82%	The small size of data would influence model accuracy
LDA	69.57%	LDA works better for normally distributed data

Conclusions

- Among the three machine learning methods
  - Logistic regression is so far presenting the most accurate data
  - quite satisfying to reach a classification percentage of 86%

- The classification percentage changes depend on the application in which data mining is used
  - Still, there are total of 75 attributes that can affect the target value
  - But, all published experiments refer to using a subset of 14 of them.

- Not enough data entries to predict more accurate result (303 entries)
  - To have a good model, the data entries need to be larger than 2^8

References

Data source: https://www.kaggle.com/ronitf/heart-disease-uci

https://stats.stackexchange.com/questions/248812/when-does-logistic-regression-not-work-properly