Prediction of next-day rain in Australia

In this report, we will use the data set "Rain in Australia" from Kaggle.com to predict the next-day rain in Australia.

Link: https://www.kaggle.com/jsphyg/weather-dataset-rattle-package

1 Data preparation

1.1 overall glimpse of the data set

```
> head(rain)
        Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine WindGustDir WindGustSpeed WindDir9am
1 2008-12-01   Albury    13.4    22.9      0.6          NA       NA           W            44          W
2 2008-12-02   Albury     7.4    25.1      0.0          NA       NA         WNW            44        NNW
3 2008-12-03   Albury    12.9    25.7      0.0          NA       NA         WSW            46          W
4 2008-12-04   Albury     9.2    28.0      0.0          NA       NA          NE            24         SE
5 2008-12-05   Albury    17.5    32.3      1.0          NA       NA           W            41        ENE
6 2008-12-06   Albury    14.6    29.7      0.2          NA       NA         WNW            56          W
  WindDir3pm WindSpeed9am WindSpeed3pm Humidity9am Humidity3pm Pressure9am Pressure3pm Cloud9am Cloud3pm
1        WNW           20           24          71          22      1007.7      1007.1        8       NA
2        WSW            4           22          44          25      1010.6      1007.8       NA       NA
3        WSW           19           26          38          30      1007.6      1008.7       NA        2
4          E           11            9          45          16      1017.6      1012.8       NA       NA
5         NW            7           20          82          33      1010.8      1006.0        7        8
6          W           19           24          55          23      1009.2      1005.4       NA       NA
  Temp9am Temp3pm RainToday RainTomorrow
1    16.9    21.8        No           No
2    17.2    24.3        No           No
3    21.0    23.2        No           No
4    18.1    26.5        No           No
5    17.8    29.7        No           No
6    20.6    28.9        No           No
```

The size of the data set:

```
> print(c(nrow(rain), ncol(rain)))
[1] 145460     23
```

1.2 data cleaning

```
> colSums(df_null)
         Date      Location       MinTemp       MaxTemp      Rainfall   Evaporation      Sunshine
            0             0          1485          1261          3261         62790         69835
 WindGustDir WindGustSpeed    WindDir9am    WindDir3pm WindSpeed9am WindSpeed3pm Humidity9am
        10326         10263         10566          4228          1767          3062          2654
Humidity3pm   Pressure9am   Pressure3pm     Cloud9am     Cloud3pm      Temp9am      Temp3pm
         4507         15065         15028         55888         59358          1767          3609
   RainToday  RainTomorrow
         3261          3267
```

We exclude the variables that have 50,000+ missing values, which are Evaporation, Sunshine, Cloud9am and Cloud3pm.

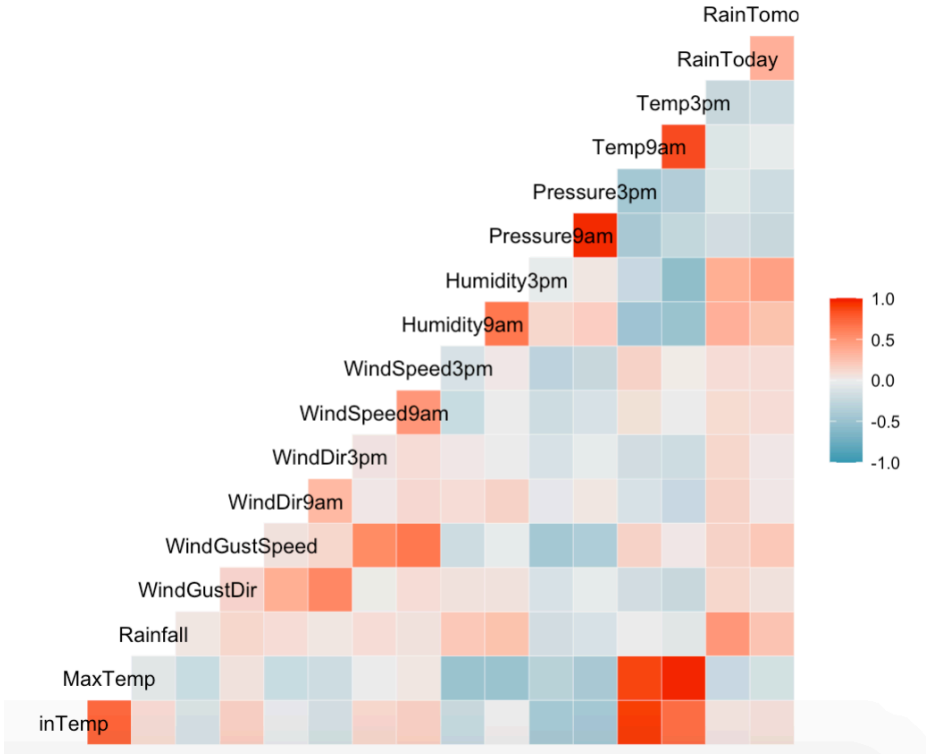
Then we deal with the missing values. For variable of numeric type, we fill the missing

values with its mean. For variable of factor type, we fill the missing values with the value that appears the most frequently.

```
> summary(rain)
         Date           MinTemp          MaxTemp         Rainfall        WindGustDir    WindGustSpeed
 2013-03-01:    49   Min.   :-8.50   Min.   :-4.80   Min.   :  0.000   W      : 9915   Min.   :  6.00
 2013-03-02:    49   1st Qu.: 7.60   1st Qu.:17.90   1st Qu.:  0.000   SE     : 9418   1st Qu.: 31.00
 2013-03-03:    49   Median :12.00   Median :22.60   Median :  0.000   N      : 9313   Median : 39.00
 2013-03-04:    49   Mean   :12.19   Mean   :23.22   Mean   :  2.361   SSE    : 9216   Mean   : 40.03
 2013-03-05:    49   3rd Qu.:16.90   3rd Qu.:28.20   3rd Qu.:  0.800   E      : 9181   3rd Qu.: 48.00
 2013-03-06:    49   Max.   :33.90   Max.   :48.10   Max.   :371.000   (Other):88091   Max.   :135.00
 (Other)   :145166   NA's   :1485    NA's   :1261    NA's   :3261      NA's   :10326   NA's   :10263
  WindDir9am      WindDir3pm     WindSpeed9am     WindSpeed3pm    Humidity9am      Humidity3pm
 N      :11758   SE     :10838   Min.   :  0.00   Min.   : 0.00   Min.   :  0.00   Min.   :  0.00
 SE     : 9287   W      :10110   1st Qu.:  7.00   1st Qu.:13.00   1st Qu.: 57.00   1st Qu.: 37.00
 E      : 9176   S      : 9926   Median : 13.00   Median :19.00   Median : 70.00   Median : 52.00
 SSE    : 9112   WSW    : 9518   Mean   : 14.04   Mean   :18.66   Mean   : 68.88   Mean   : 51.54
 NW     : 8749   SSE    : 9399   3rd Qu.: 19.00   3rd Qu.:24.00   3rd Qu.: 83.00   3rd Qu.: 66.00
 (Other):86812   (Other):91441   Max.   :130.00   Max.   :87.00   Max.   :100.00   Max.   :100.00
 NA's   :10566   NA's   : 4228   NA's   :1767     NA's   :3062    NA's   :2654     NA's   :4507
  Pressure9am      Pressure3pm       Temp9am         Temp3pm       RainToday     RainTomorrow
 Min.   : 980.5   Min.   : 977.1   Min.   :-7.20   Min.   :-5.40   No  :110319   No  :110316
 1st Qu.:1012.9   1st Qu.:1010.4   1st Qu.:12.30   1st Qu.:16.60   Yes : 31880   Yes : 31877
 Median :1017.6   Median :1015.2   Median :16.70   Median :21.10   NA's:  3261   NA's:  3267
 Mean   :1017.6   Mean   :1015.3   Mean   :16.99   Mean   :21.68
 3rd Qu.:1022.4   3rd Qu.:1020.0   3rd Qu.:21.60   3rd Qu.:26.40
 Max.   :1041.0   Max.   :1039.6   Max.   :40.20   Max.   :46.70
 NA's   :15065    NA's   :15028    NA's   :1767    NA's   :3609
```

For example, we fill the missing values in WindGustDir with "W". Since the number of Yes in RainToday and RainTomorrow is extremely close, we fill all the missing values in RainToday and RainTomorrow with "Yes".

1.3 correlation



From the heatmap, we see that:

Temp3pm and Temp9am are highly correlated;

Temp3pm and MaxTemp are highly correlated;

Temp3pm and MinTemp are highly correlated;

Temp9am and MaxTemp are highly correlated;

Temp9am and MinTemp are highly correlated;

Humidity3pm and Humidity9am are highly correlated.

Therefore, we drop the variables that are highly correlated: Temp9am, Temp3pm and Humidity9am.

Here, we have our data set for the analysis.

2 logistic model

2.1 fit the logistic model

We split the data set into train data set and test data set. Then we use the train data set to fit the model and use the model to predict the result in test data set.

```
#split train and test dataset
train = sample(1:nrow(rain_logi), nrow(rain_logi)/2)
rain.train = rain_logi[train, ]
rain.test = rain_logi[-train, ]
RainTomorrow.test = rain_logi$RainTomorrow[-train]
rain.train
```

```
> logi_model<-glm(RainTomorrow~., family = "binomial",data = rain.train)
> summary(logi_model)

Call:
glm(formula = RainTomorrow ~ ., family = "binomial", data = rain.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.2987  -0.6289  -0.3885  -0.1594   3.0341

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)   57.2681138  1.9059063  30.048  < 2e-16 ***
MinTemp        0.0306362  0.0034975   8.759  < 2e-16 ***
MaxTemp       -0.0410770  0.0036371 -11.294  < 2e-16 ***
Rainfall       0.0274000  0.0014816  18.494  < 2e-16 ***
WindGustDir2   0.0768610  0.0617334   1.245 0.213114
WindGustDir3   0.0181485  0.0632801   0.287 0.774268
WindGustDir4   0.1301823  0.0661582   1.968 0.049098 *
WindGustDir5  -0.0158134  0.0682169  -0.232 0.816685
WindGustDir6   0.0899560  0.0711671   1.264 0.206226
WindGustDir7   0.0720441  0.0718626   1.003 0.316090
WindGustDir8   0.0891340  0.0695666   1.281 0.200097
WindGustDir9   0.1376567  0.0650591   2.116 0.034356 *
WindGustDir10  0.0938761  0.0618417   1.518 0.129013
WindGustDir11  0.0157987  0.0643909   0.245 0.806181
WindGustDir12  0.1107093  0.0665032   1.665 0.095969 .
WindGustDir13  0.1024314  0.0666009   1.538 0.124051
WindGustDir14  0.0967257  0.0563394   1.717 0.086008 .
WindGustDir15  0.0415984  0.0688789   0.604 0.545887
WindGustDir16  0.0830774  0.0662919   1.253 0.210130
WindGustSpeed  0.0517309  0.0011614  44.540  < 2e-16 ***
WindDir9am2   -0.0437019  0.0612643  -0.713 0.475639
WindDir9am3    0.0451197  0.0609631   0.740 0.459230
WindDir9am4   -0.0485119  0.0514981  -0.942 0.346186
WindDir9am5   -0.0548730  0.0631621  -0.869 0.384976
WindDir9am6   -0.0123047  0.0626933  -0.196 0.844400
WindDir9am7   -0.0274191  0.0638945  -0.429 0.667828
WindDir9am8   -0.0160914  0.0618302  -0.260 0.794669
WindDir9am9   -0.0039946  0.0619583  -0.064 0.948594
WindDir9am10  -0.0090758  0.0596743  -0.152 0.879116
WindDir9am11   0.0038069  0.0605825   0.063 0.949896
WindDir9am12   0.0569935  0.0634457   0.898 0.369024
WindDir9am13  -0.0568120  0.0624483  -0.910 0.362957
WindDir9am14  -0.0738350  0.0627544  -1.177 0.239367
WindDir9am15  -0.0702666  0.0645737  -1.088 0.276524
WindDir9am16   0.0011416  0.0651950   0.018 0.986030
WindDir3pm2   -0.0092003  0.0631489  -0.146 0.884164
WindDir3pm3    0.0692106  0.0606321   1.141 0.253668
WindDir3pm4   -0.0453711  0.0663289  -0.684 0.493955
WindDir3pm5    0.0457716  0.0652554   0.701 0.483039
WindDir3pm6    0.0124651  0.0697363   0.179 0.858137
WindDir3pm7   -0.0905009  0.0689428  -1.313 0.189286
WindDir3pm8    0.0062821  0.0664234   0.095 0.924651
WindDir3pm9   -0.0248566  0.0637808  -0.390 0.696744
WindDir3pm10  -0.0315169  0.0569422  -0.553 0.579928
WindDir3pm11  -0.1147536  0.0642059  -1.787 0.073893 .
WindDir3pm12  -0.0983543  0.0672658  -1.462 0.143694
WindDir3pm13  -0.0931972  0.0654073  -1.425 0.154193
WindDir3pm14  -0.0203444  0.0635488  -0.320 0.748863
WindDir3pm15  -0.0060029  0.0664592  -0.090 0.928029
WindDir3pm16  -0.0688950  0.0647255  -1.064 0.287139
WindSpeed9am  -0.0059109  0.0015904  -3.717 0.000202 ***
WindSpeed3pm  -0.0292753  0.0015831 -18.493  < 2e-16 ***
Humidity3pm    0.0561595  0.0007989  70.298  < 2e-16 ***
Pressure9am    0.0959332  0.0057977  16.547  < 2e-16 ***
Pressure3pm   -0.1578702  0.0058819 -26.840  < 2e-16 ***
RainToday1    -0.0001850  0.0245784  -0.008 0.993994
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 80245  on 72729  degrees of freedom
Residual deviance: 59099  on 72674  degrees of freedom
AIC: 59211

Number of Fisher Scoring iterations: 5
```

2.2 Stepwise Selection:

```
> step.model<-stepAIC(logi_model, direction = "both", trace = FALSE)
> summary(step.model)

Call:
glm(formula = RainTomorrow ~ MinTemp + MaxTemp + Rainfall + WindGustSpeed +
    WindSpeed9am + WindSpeed3pm + Humidity3pm + Pressure9am +
    Pressure3pm, family = "binomial", data = rain.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.2933  -0.6298  -0.3889  -0.1602   3.0383

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)   57.3351568  1.9039022  30.115  < 2e-16 ***
MinTemp        0.0304937  0.0034942   8.727  < 2e-16 ***
MaxTemp       -0.0409227  0.0036339 -11.261  < 2e-16 ***
Rainfall       0.0273386  0.0014801  18.471  < 2e-16 ***
WindGustSpeed  0.0516891  0.0011606  44.538  < 2e-16 ***
WindSpeed9am  -0.0058553  0.0015891  -3.685 0.000229 ***
WindSpeed3pm  -0.0292770  0.0015815 -18.512  < 2e-16 ***
Humidity3pm    0.0561523  0.0007984  70.328  < 2e-16 ***
Pressure9am    0.0958261  0.0057916  16.546  < 2e-16 ***
Pressure3pm   -0.1578065  0.0058755 -26.858  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 80245  on 72729  degrees of freedom
Residual deviance: 59139  on 72720  degrees of freedom
AIC: 59159

Number of Fisher Scoring iterations: 5
```

Thus, we exclude WindGustDir, WindDir9am, WindDir3pm and RainToday in our model, and get the final logistic model:

```
> logi_model1<-glm(RainTomorrow~.-WindGustDir-WindDir9am-WindDir3pm-RainToday, family = "binomial",data = rain.train)
> summary(logi_model1)

Call:
glm(formula = RainTomorrow ~ . - WindGustDir - WindDir9am - WindDir3pm -
    RainToday, family = "binomial", data = rain.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.2933  -0.6298  -0.3889  -0.1602   3.0383

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)   57.3351568  1.9039022  30.115  < 2e-16 ***
MinTemp        0.0304937  0.0034942   8.727  < 2e-16 ***
MaxTemp       -0.0409227  0.0036339 -11.261  < 2e-16 ***
Rainfall       0.0273386  0.0014801  18.471  < 2e-16 ***
WindGustSpeed  0.0516891  0.0011606  44.538  < 2e-16 ***
WindSpeed9am  -0.0058553  0.0015891  -3.685 0.000229 ***
WindSpeed3pm  -0.0292770  0.0015815 -18.512  < 2e-16 ***
Humidity3pm    0.0561523  0.0007984  70.328  < 2e-16 ***
Pressure9am    0.0958261  0.0057916  16.546  < 2e-16 ***
Pressure3pm   -0.1578065  0.0058755 -26.858  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 80245  on 72729  degrees of freedom
Residual deviance: 59139  on 72720  degrees of freedom
AIC: 59159

Number of Fisher Scoring iterations: 5
```

Then we use the model to predict the data in the test data set and calculate the prediction accuracy:

```
> rain.pred = predict(logi_model1, rain.test, type = "response")
> rain.pred.class <-ifelse(rain.pred>0.5, 1,0)
> mean(rain.pred.class == rain.test$RainTomorrow)
[1] 0.8213392
```

The prediction accuracy of our logistic model is 82.13%

3 SVM

I tried in this part, using the train set in logistic regression to conduct SVM. However, the dataset is too large. And R runs forever. Thus, I choose randomly 10000 observations in train set to fit the model. And we still going to use the model to predict the next-day rain in test data set.

3.1 SVM with linear kernel

```
> summary(svmfit)

Call:
svm(formula = RainTomorrow ~ ., data = rain.train_svm, kernel = "linear", cost = 10, scale = FALSE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  10

Number of Support Vectors:  3054

 ( 1524 1530 )


Number of Classes:  2

Levels:
 0 1



> ypred = predict(svmfit, rain.test)
> table(predict = ypred, truth = rain.test$RainTomorrow)
       truth
predict     0     1
      0 51968 10117
      1  3110  7535
```

Therefore, we have the prediction accuracy:

$$\frac{51968 + 7535}{51968 + 10117 + 3110 + 7535} = 84.82\%$$

3.2 SVM with polynomial kernel

```
> svmfit_poly = svm(RainTomorrow~., data = rain.train_svm, kernel = "polynomial", cost = 10, scale = FALSE)

WARNING: reaching max number of iterations
> summary(svmfit_poly)

Call:
svm(formula = RainTomorrow ~ ., data = rain.train_svm, kernel = "polynomial", cost = 10, scale = FALSE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  polynomial
       cost:  10
     degree:  3
     coef.0:  0

Number of Support Vectors:  2941

 ( 1470 1471 )


Number of Classes:  2

Levels:
 0 1

> table(predict = ypred_poly, truth = rain.test$RainTomorrow)
       truth
predict     0     1
      0 50683  8780
      1  4395  8872
```

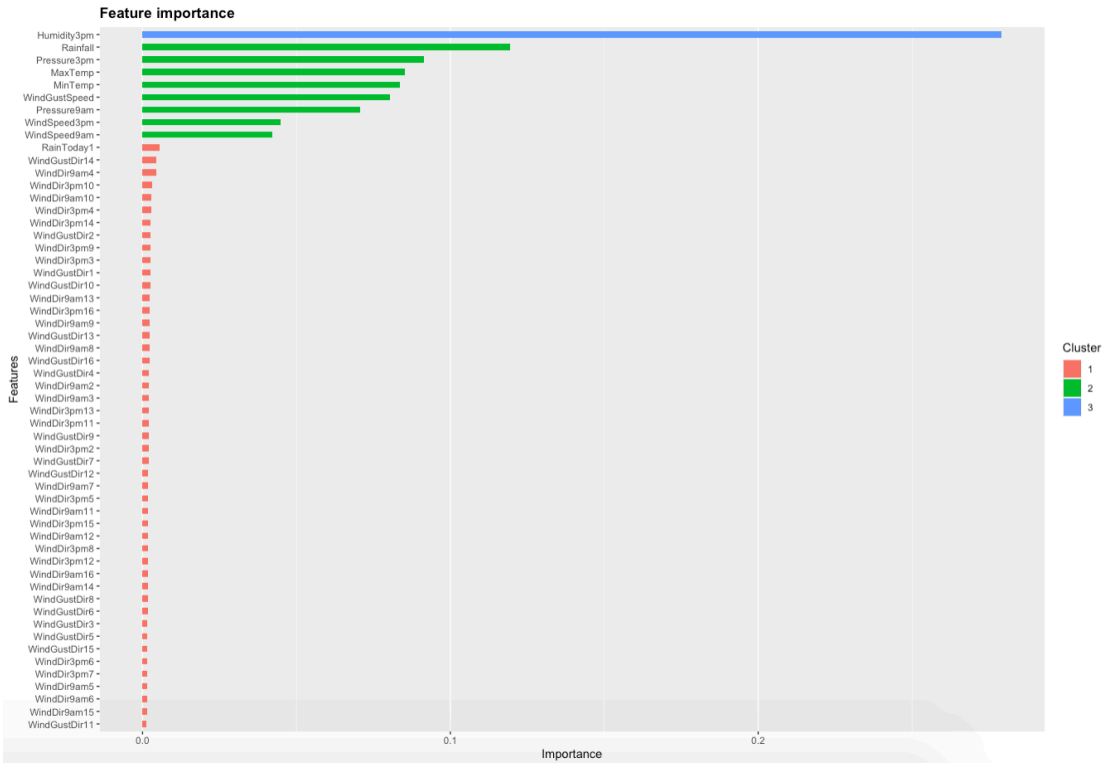
We have the prediction accuracy:

$$\frac{50683 + 8872}{50683 + 8780 + 4395 + 8872} = 81.89\%$$

4 Free Style Part

We use xgboost algorithm to predict the next-day rain

```
> train_matrix <- sparse.model.matrix(RainTomorrow~.-1, data = rain.train)
> test_matrix <- sparse.model.matrix(RainTomorrow~.-1, data = rain.test)
> train_label <- as.numeric(rain.train$RainTomorrow ==1)
> test_label <- as.numeric(rain.test$RainTomorrow == 1)
> train_fin <-list(data = train_matrix, label = train_label)
> test_fin <- list(data = train_matrix, label = test_label)
> dtrain <- xgb.DMatrix(data = train_fin$data, label = train_fin$label)
> dtest <- xgb.DMatrix(data = test_fin$data, label = test_fin$label)
> xgb<- xgboost(data = dtrain, max_depth = 15, eta = 0.5, objective = 'binary:logistic', nround = 25)
```



The feature importance of the variables here conforms with the significance level of the variables in logistic regression.

```
> library(Ckmeans.1d.dp)
> importance <- xgb.importance(train_matrix@Dimnames[[2]], model = xgb)
> head(importance)
         Feature       Gain      Cover  Frequency
1:   Humidity3pm 0.27908200 0.17476025 0.08891203
2:      Rainfall 0.11934642 0.08275172 0.07507306
3:   Pressure3pm 0.09147530 0.14961218 0.10090356
4:       MaxTemp 0.08521745 0.09224860 0.12616304
5:       MinTemp 0.08362210 0.08522160 0.12754022
6: WindGustSpeed 0.08038052 0.10649113 0.07728998
> xgb.ggplot.importance(importance)
> pre_xgb = round(predict(xgb, newdata = dtest))
> tt <- table(test_label, pre_xgb, dnn = c("true","pre"))
> (tt[1,1]+tt[2,2])/sum(tt)    #the rate of correct predictions
[1] 0.6379761
> 1-(tt[1,1]+tt[2,2])/sum(tt)
[1] 0.3620239
```

The prediction accuracy is about 63.80%, which is pretty low.