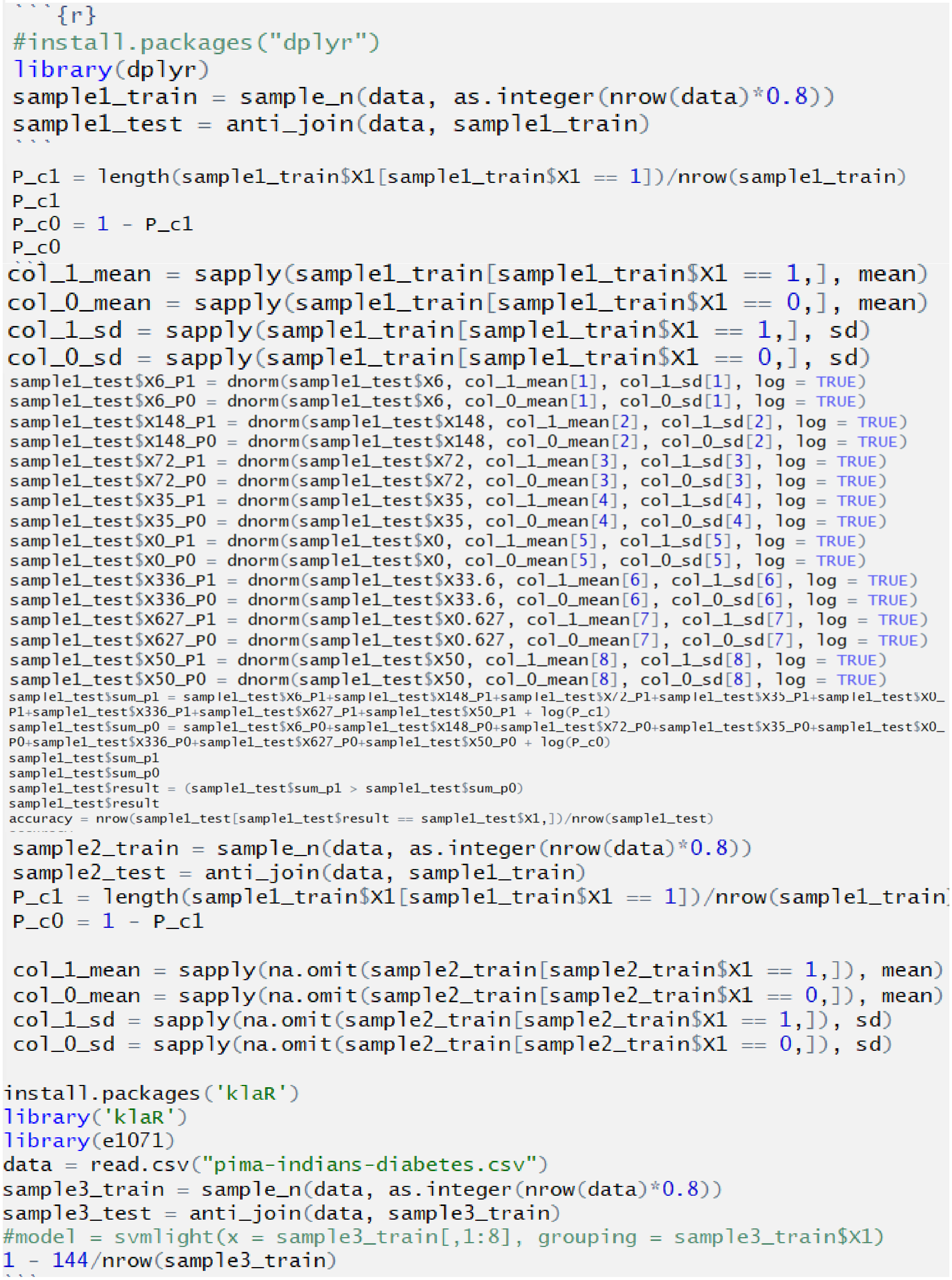
1a. 0.7538961

1b. 0.7053127

1d. 0.7650897



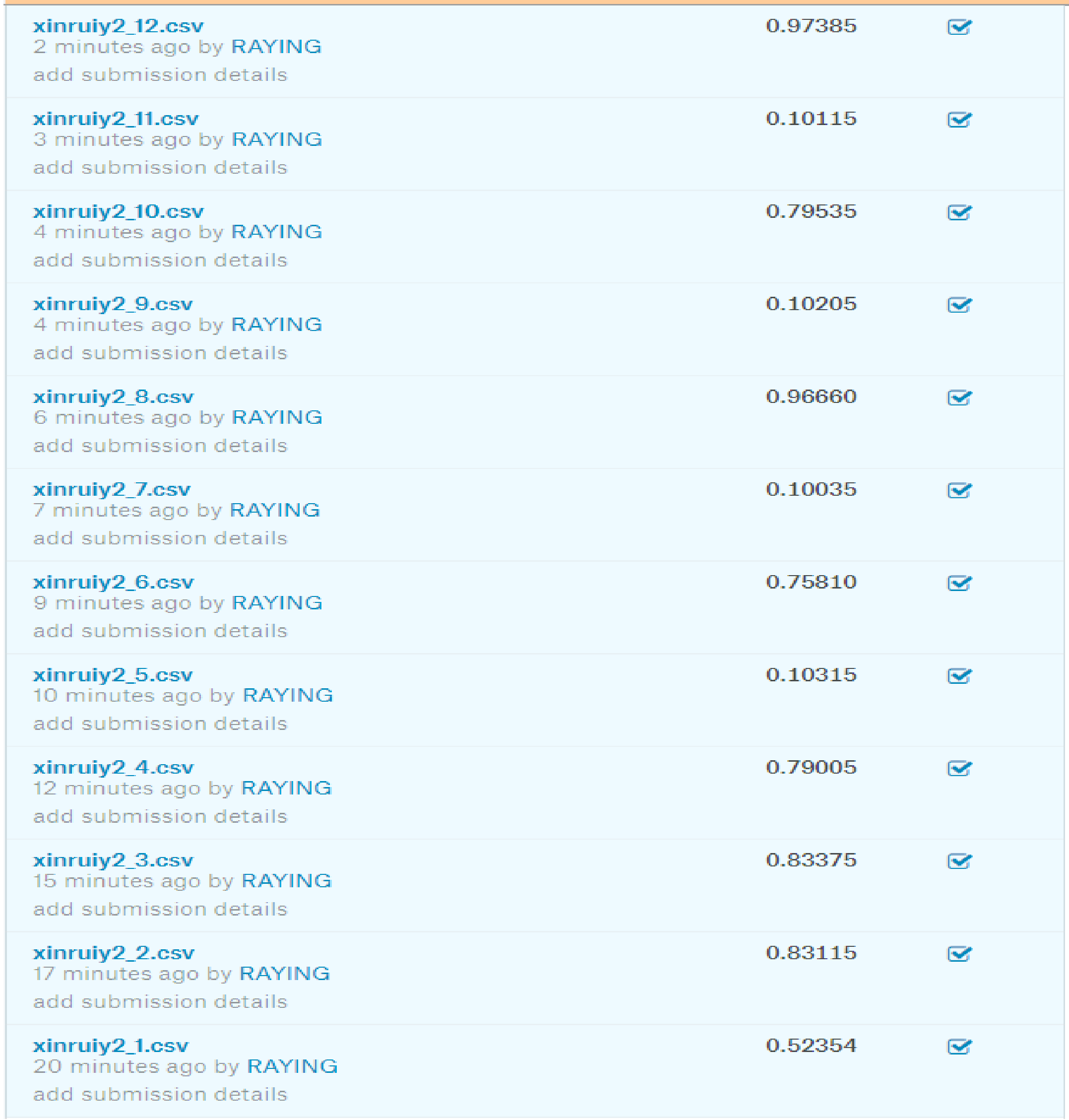
~~~~
```{r}
#install.packages("dplyr")
library(dplyr)
sample1_train = sample_n(data, as.integer(nrow(data)*0.8))
sample1_test = anti_join(data, sample1_train)
```

P_c1 = length(sample1_train$X1[sample1_train$X1 == 1])/nrow(sample1_train)
P_c1
P_c0 = 1 - P_c1
P_c0
col_1_mean = sapply(sample1_train[sample1_train$X1 == 1,], mean)
col_0_mean = sapply(sample1_train[sample1_train$X1 == 0,], mean)
col_1_sd = sapply(sample1_train[sample1_train$X1 == 1,], sd)
col_0_sd = sapply(sample1_train[sample1_train$X1 == 0,], sd)
sample1_test$X6_P1 = dnorm(sample1_test$X6, col_1_mean[1], col_1_sd[1], log = TRUE)
sample1_test$X6_P0 = dnorm(sample1_test$X6, col_0_mean[1], col_0_sd[1], log = TRUE)
sample1_test$X148_P1 = dnorm(sample1_test$X148, col_1_mean[2], col_1_sd[2], log = TRUE)
sample1_test$X148_P0 = dnorm(sample1_test$X148, col_0_mean[2], col_0_sd[2], log = TRUE)
sample1_test$X72_P1 = dnorm(sample1_test$X72, col_1_mean[3], col_1_sd[3], log = TRUE)
sample1_test$X72_P0 = dnorm(sample1_test$X72, col_0_mean[3], col_0_sd[3], log = TRUE)
sample1_test$X35_P1 = dnorm(sample1_test$X35, col_1_mean[4], col_1_sd[4], log = TRUE)
sample1_test$X35_P0 = dnorm(sample1_test$X35, col_0_mean[4], col_0_sd[4], log = TRUE)
sample1_test$X0_P1 = dnorm(sample1_test$X0, col_1_mean[5], col_1_sd[5], log = TRUE)
sample1_test$X0_P0 = dnorm(sample1_test$X0, col_0_mean[5], col_0_sd[5], log = TRUE)
sample1_test$X336_P1 = dnorm(sample1_test$X33.6, col_1_mean[6], col_1_sd[6], log = TRUE)
sample1_test$X336_P0 = dnorm(sample1_test$X33.6, col_0_mean[6], col_0_sd[6], log = TRUE)
sample1_test$X627_P1 = dnorm(sample1_test$X0.627, col_1_mean[7], col_1_sd[7], log = TRUE)
sample1_test$X627_P0 = dnorm(sample1_test$X0.627, col_0_mean[7], col_0_sd[7], log = TRUE)
sample1_test$X50_P1 = dnorm(sample1_test$X50, col_1_mean[8], col_1_sd[8], log = TRUE)
sample1_test$X50_P0 = dnorm(sample1_test$X50, col_0_mean[8], col_0_sd[8], log = TRUE)
sample1_test$sum_p1 = sample1_test$X6_P1+sample1_test$X148_P1+sample1_test$X72_P1+sample1_test$X35_P1+sample1_test$X0_
P1+sample1_test$X336_P1+sample1_test$X627_P1+sample1_test$X50_P1 + log(P_c1)
sample1_test$sum_p0 = sample1_test$X6_P0+sample1_test$X148_P0+sample1_test$X72_P0+sample1_test$X35_P0+sample1_test$X0_
P0+sample1_test$X336_P0+sample1_test$X627_P0+sample1_test$X50_P0 + log(P_c0)
sample1_test$sum_p1
sample1_test$sum_p0
sample1_test$result = (sample1_test$sum_p1 > sample1_test$sum_p0)
sample1_test$result
accuracy = nrow(sample1_test[sample1_test$result == sample1_test$X1,])/nrow(sample1_test)
accuracy
sample2_train = sample_n(data, as.integer(nrow(data)*0.8))
sample2_test = anti_join(data, sample1_train)
P_c1 = length(sample1_train$X1[sample1_train$X1 == 1])/nrow(sample1_train
P_c0 = 1 - P_c1

col_1_mean = sapply(na.omit(sample2_train[sample2_train$X1 == 1,]), mean)
col_0_mean = sapply(na.omit(sample2_train[sample2_train$X1 == 0,]), mean)
col_1_sd = sapply(na.omit(sample2_train[sample2_train$X1 == 1,]), sd)
col_0_sd = sapply(na.omit(sample2_train[sample2_train$X1 == 0,]), sd)

install.packages('klaR')
library('klaR')
library(e1071)
data = read.csv("pima-indians-diabetes.csv")
sample3_train = sample_n(data, as.integer(nrow(data)*0.8))
sample3_test = anti_join(data, sample3_train)
#model = svmlight(x = sample3_train[,1:8], grouping = sample3_train$X1)
1 - 144/nrow(sample3_train)
```
~~~~

x	Method	Probability
1	Gaussian + untouched	0.52354
2	Gaussian + stretched	0.83115
3	Bernoulli + untouched	0.83375
4	Bernoulli + stretched	0.79005
5	10 trees + 4 depth + untouched	0.10315
6	10 trees + 4 depth + stretched	0.75810
7	10 trees + 16 depth + untouched	0.10035
8	10 trees + 16 depth + stretched	0.96660
9	30 trees + 4 depth + untouched	0.10205
10	30 trees + 4 depth + stretched	0.79535
11	30 trees + 16 depth + untouched	0.10115
12	30 trees + 16 depth + stretched	0.97385



xinruiy2_12.csv 0.97385 ☑
2 minutes ago by RAYING
add submission details

xinruiy2_11.csv 0.10115 ☑
3 minutes ago by RAYING
add submission details

xinruiy2_10.csv 0.79535 ☑
4 minutes ago by RAYING
add submission details

xinruiy2_9.csv 0.10205 ☑
4 minutes ago by RAYING
add submission details

xinruiy2_8.csv 0.96660 ☑
6 minutes ago by RAYING
add submission details

xinruiy2_7.csv 0.10035 ☑
7 minutes ago by RAYING
add submission details

xinruiy2_6.csv 0.75810 ☑
9 minutes ago by RAYING
add submission details

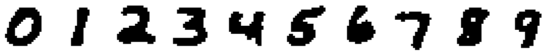
xinruiy2_5.csv 0.10315 ☑
10 minutes ago by RAYING
add submission details

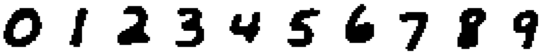
xinruiy2_4.csv 0.79005 ☑
12 minutes ago by RAYING
add submission details

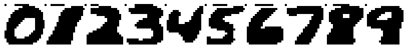
xinruiy2_3.csv 0.83375 ☑
15 minutes ago by RAYING
add submission details

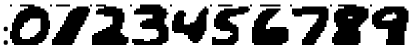
xinruiy2_2.csv 0.83115 ☑
17 minutes ago by RAYING
add submission details

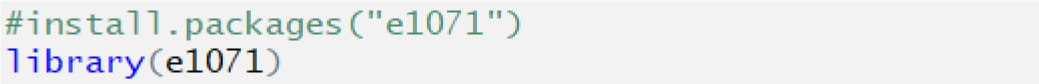
xinruiy2_1.csv 0.52354 ☑
20 minutes ago by RAYING
add submission details

Gaussian + untouched : 

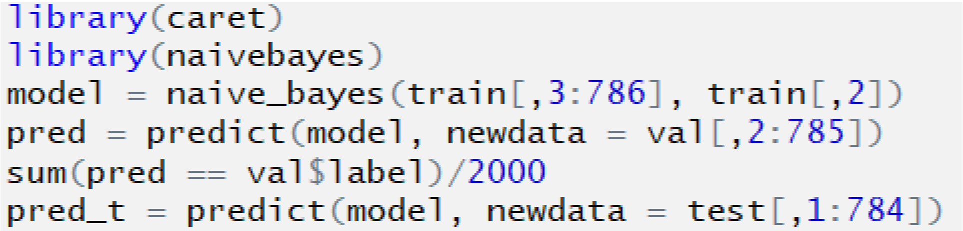
Bernoulli + Untouched: 

Gaussian + Stretched: 

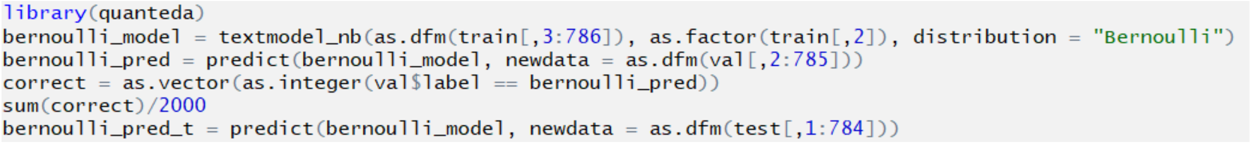
Bernoulli + Stretched: 



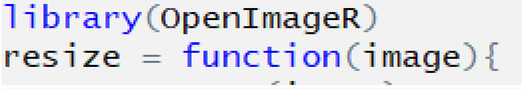
```r
#install.packages("e1071")
library(e1071)
```



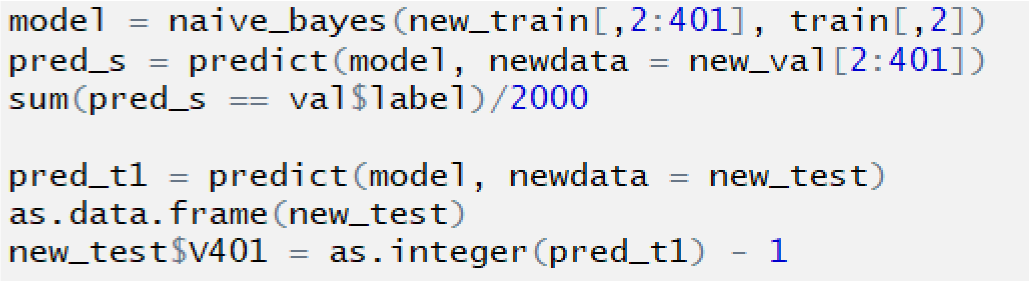
```r
library(caret)
library(naivebayes)
model = naive_bayes(train[,3:786], train[,2])
pred = predict(model, newdata = val[,2:785])
sum(pred == val$label)/2000
pred_t = predict(model, newdata = test[,1:784])
```



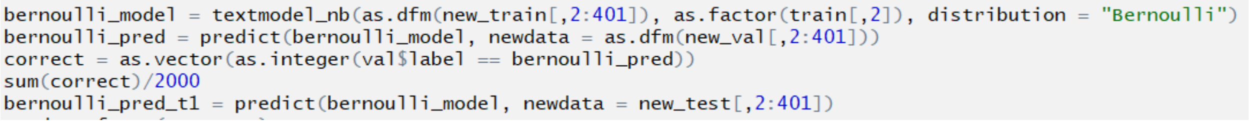
```r
library(quanteda)
bernoulli_model = textmodel_nb(as.dfm(train[,3:786]), as.factor(train[,2]), distribution = "Bernoulli")
bernoulli_pred = predict(bernoulli_model, newdata = as.dfm(val[,2:785]))
correct = as.vector(as.integer(val$label == bernoulli_pred))
sum(correct)/2000
bernoulli_pred_t = predict(bernoulli_model, newdata = as.dfm(test[,1:784]))
```



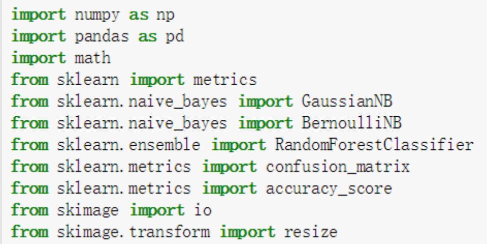
```r
library(OpenImageR)
resize = function(image){
```



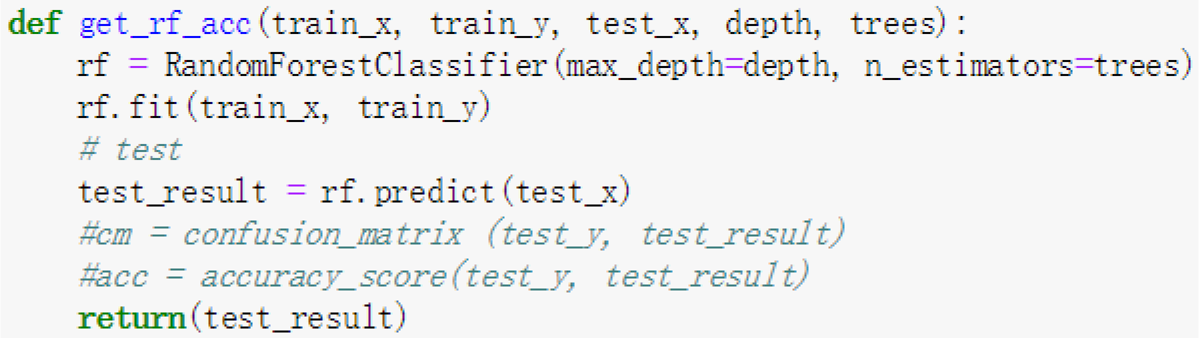
```r
model = naive_bayes(new_train[,2:401], train[,2])
pred_s = predict(model, newdata = new_val[2:401])
sum(pred_s == val$label)/2000

pred_t1 = predict(model, newdata = new_test)
as.data.frame(new_test)
new_test$V401 = as.integer(pred_t1) - 1
```



```r
bernoulli_model = textmodel_nb(as.dfm(new_train[,2:401]), as.factor(train[,2]), distribution = "Bernoulli")
bernoulli_pred = predict(bernoulli_model, newdata = as.dfm(new_val[,2:401]))
correct = as.vector(as.integer(val$label == bernoulli_pred))
sum(correct)/2000
bernoulli_pred_t1 = predict(bernoulli_model, newdata = new_test[,2:401])
```



```python
import numpy as np
import pandas as pd
import math
from sklearn import metrics
from sklearn.naive_bayes import GaussianNB
from sklearn.naive_bayes import BernoulliNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from skimage import io
from skimage.transform import resize
```



```python
def get_rf_acc(train_x, train_y, test_x, depth, trees):
    rf = RandomForestClassifier(max_depth=depth, n_estimators=trees)
    rf.fit(train_x, train_y)
    # test
    test_result = rf.predict(test_x)
    #cm = confusion_matrix (test_y, test_result)
    #acc = accuracy_score(test_y, test_result)
    return(test_result)
```