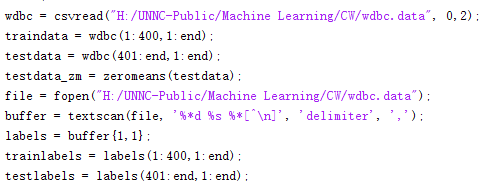
Machine Learning Course Work Report

Like Chen 16518695 [zy18695@nottingham.edu.cn](mailto:zy18695@nottingham.edu.cn)

Task 1:



Using csvread jump 2 columns read rest 30 columns of data,

Using fopen and textscan to read the labels,

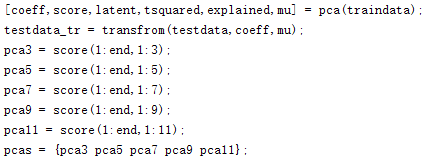
Then split them into train set and test set.

Task 2:

Task is to use the wdbc data to train decision tree.

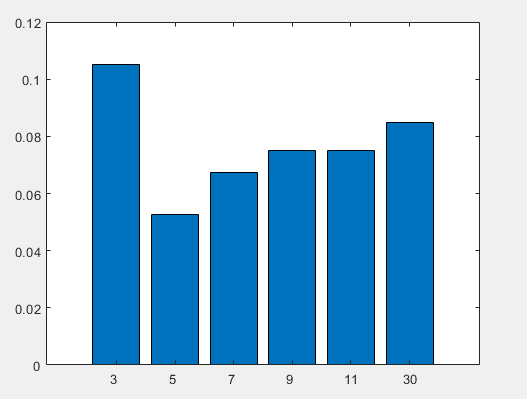
Process:

Step 1: Prepare data



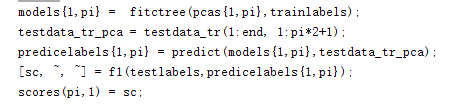
Do PCA to the train data and use transform function to translate testdata to pca train data’s space.

Step 2: Use 10-fold cross validation to evaluate the performance

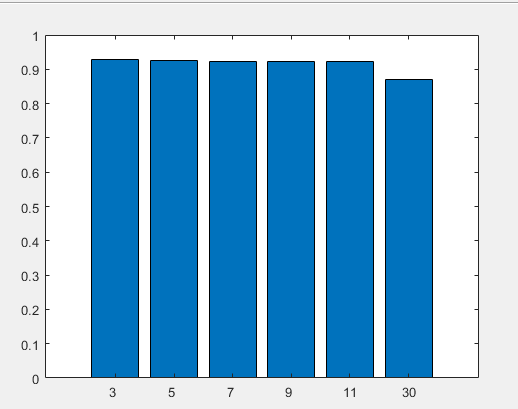


Plot the data and compare performance.

Step 3: Train tree and calculate the f1 value



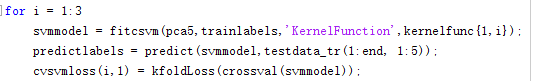
Using whole training set to train the decision tree, then use transformed testing data to predict the labels, then we can use the predict labels and truth labels to calculate the f1 value.

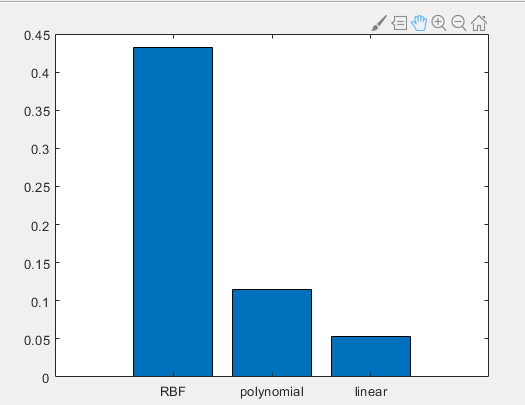


Task 3:

In this part we use the best performance feature which we evaluate in task 2 which is pca 5 as the input features in this task.

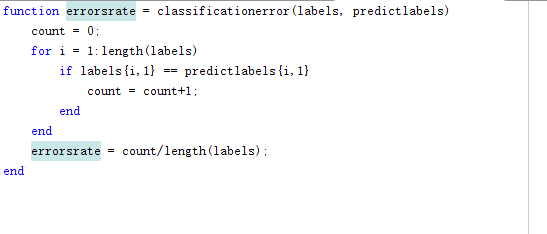
Do the 10-fold cross validation to train and validate model:

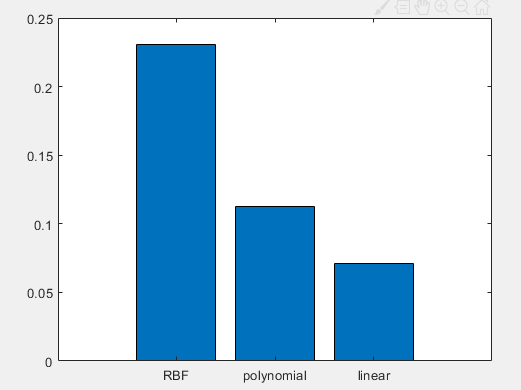




We can see that input default value RBF has highest loss.

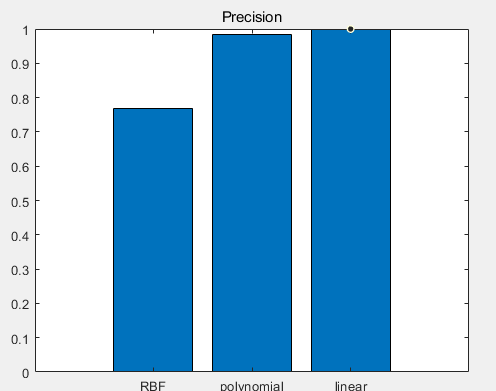
Then calculate classification error:

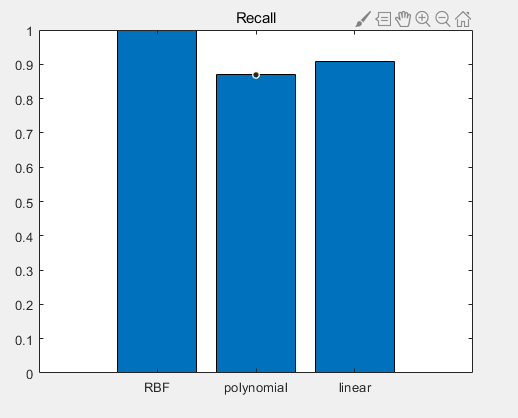


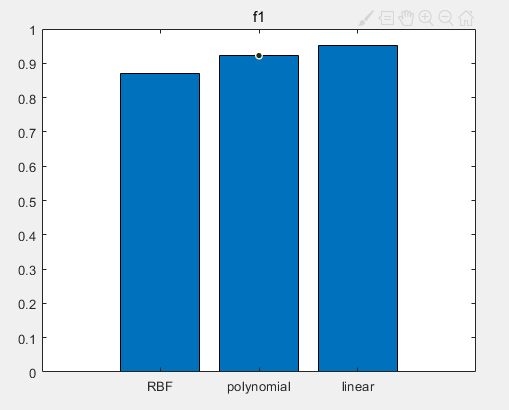


RBF has the highest error rate too.

Then calculate precision, recall and f1 value:





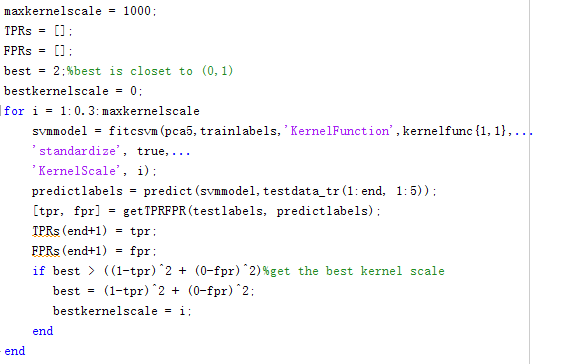


We can see that RBF still has lowest f1.

RBF has worse performance in default hyperparameter.

Then we trying to find the best hyperparameter which can give the best performance.

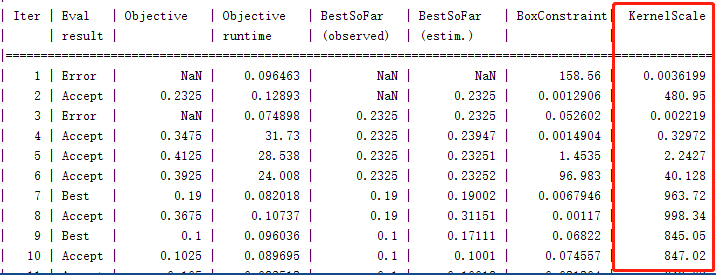
So, we trying to use the ROC curve, first we should calculate TPR and FPR for each hyperparameters we change:



I use 1000 as max kernel scale because when I use matlab automate hyperparameter optimize, I found that the kernel scale is always below 1000, and it start from a number between -0.003 to 0.003, but matlab raise an error when I use 0, hence, I set it start from 1.

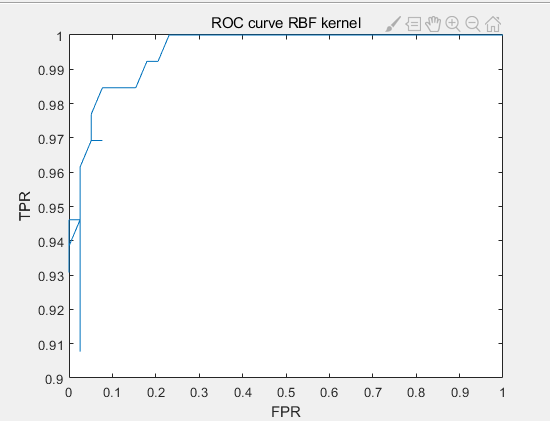


This pic is from matlab documentation of fitcsvm.



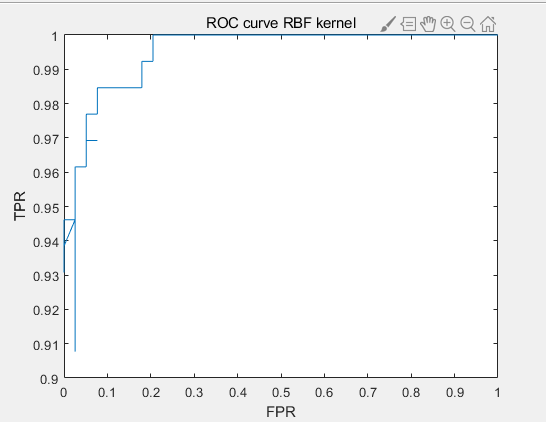
I tried different step.

From 1 to 1000 step 0.3



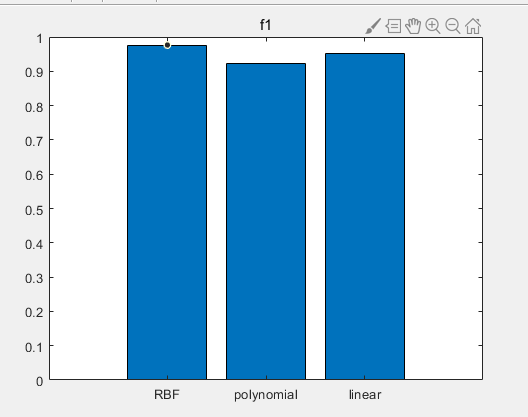
The best kernel scale is 16.3.

From 1 to 1000 step 0.1

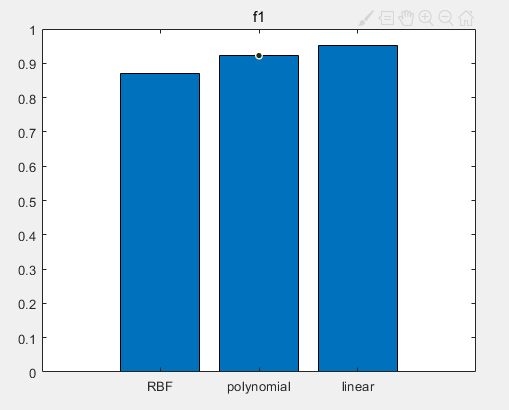


The best kernel scale is 16.3.

Then I use 16.3 as kernel scale.



New value



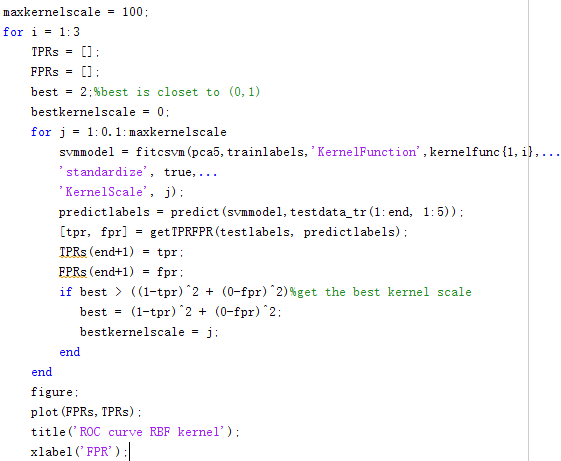
Old value

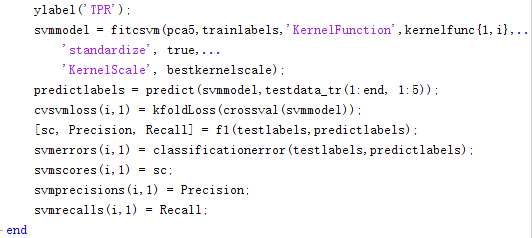
We can see that RBF now have the best performance compare to other two using the default value.

Task 4:

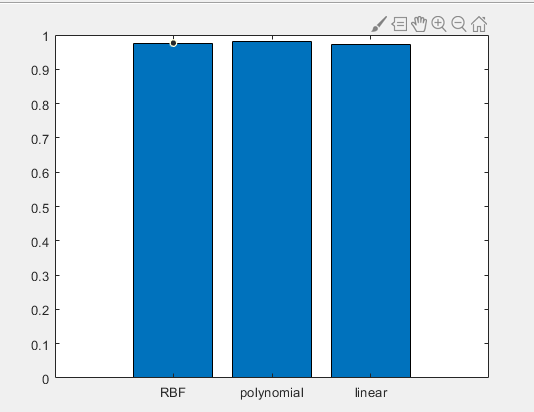
Trying to use the same method before and apply to other two kernel.

I found that in this case I does not need to calculate up to 1000, so I change max kernel scale to 100.





Then apply best kernel scale to each kernel function we get:

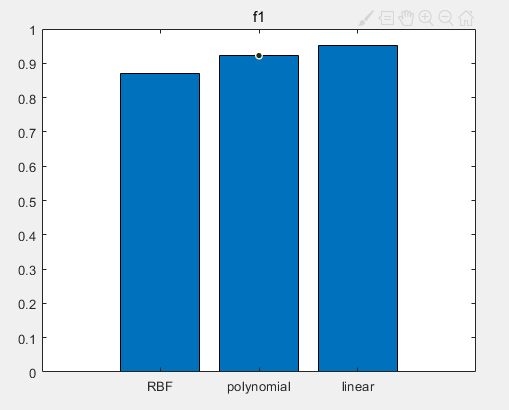


RBF: 0.9766

Polynomial:0.9804

Linear: 0.9725

Which is obviersly higher than before.



RBF: 0.8696

Polynomial:0.9224

Linear: 0.9516

Task 5: