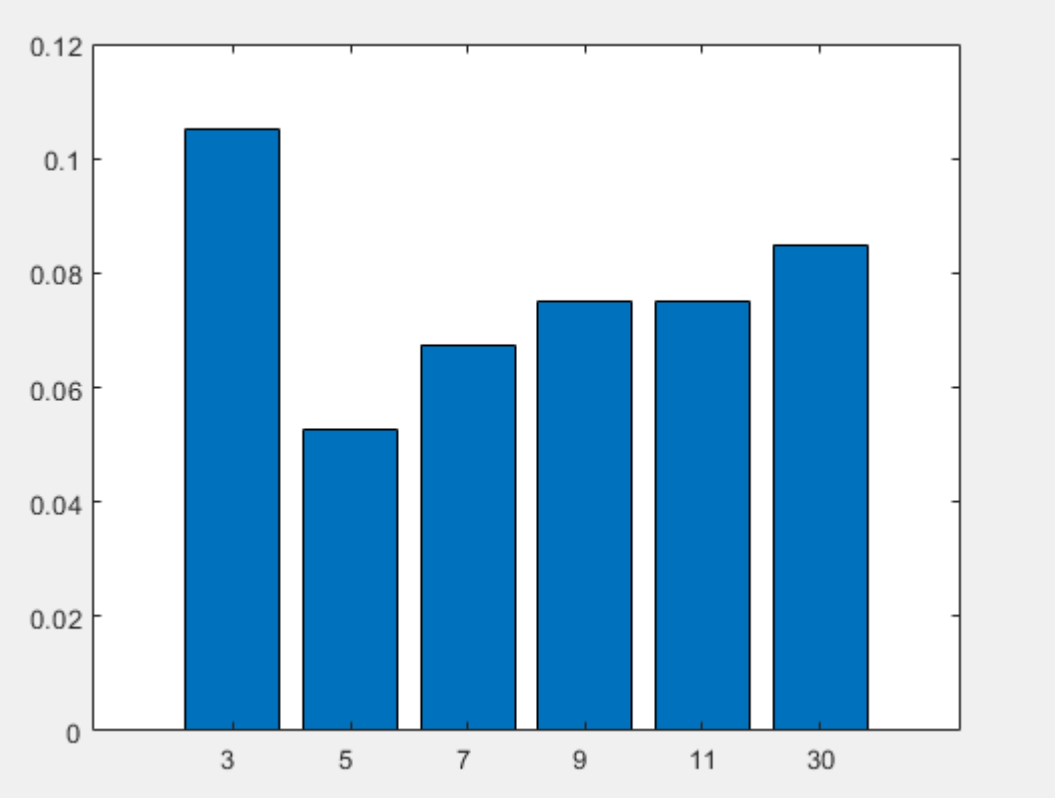
Machine Learning Course Work Report

Like Chen 16518695 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.

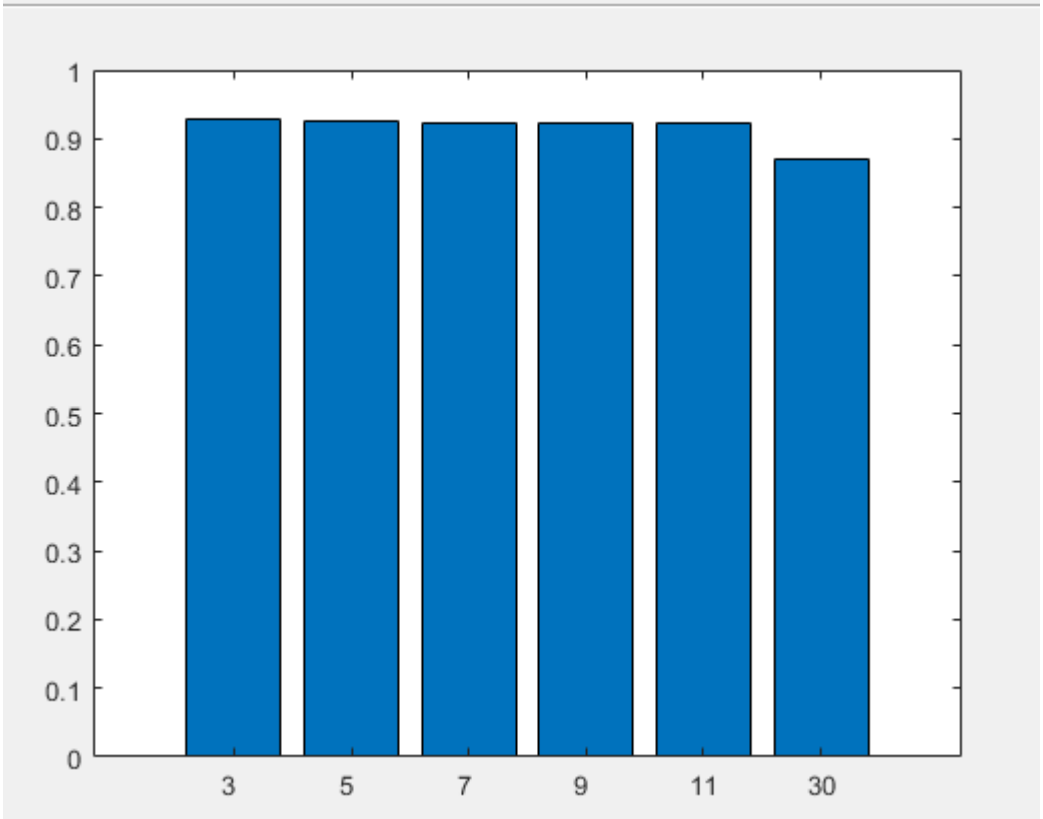
Task 2:
Task is to use the wdbc data to train decision tree.
Process:
Step 1: Prepare data
Do PCA to the train data and use transform function to translate testdata to pca train data’ s space.
Step 2: Use 10-fold cross validation to evaluate the performance
LOSS:



Pca3: 0.105
Pca5: 0.052
Pca7: 0.067
Pca9: 0.075
Pca11: 0.075
Origin-data: 0.085
Plot the data and compare performance.
In this case we just familiar with the process of building a precision tree.
In real world we should test more hyperparameters to find if there is better performance.

Step 3: Train tree and calculate the f1 value

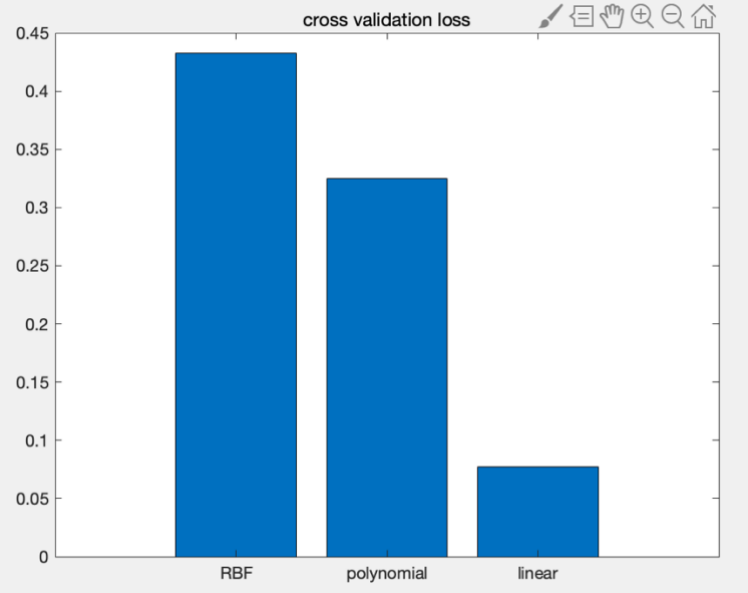
Using whole training set to train the decision tree, then use transformed testing data to predict the labels, then we can use the predict labels and truth labels to calculate the f1 value.



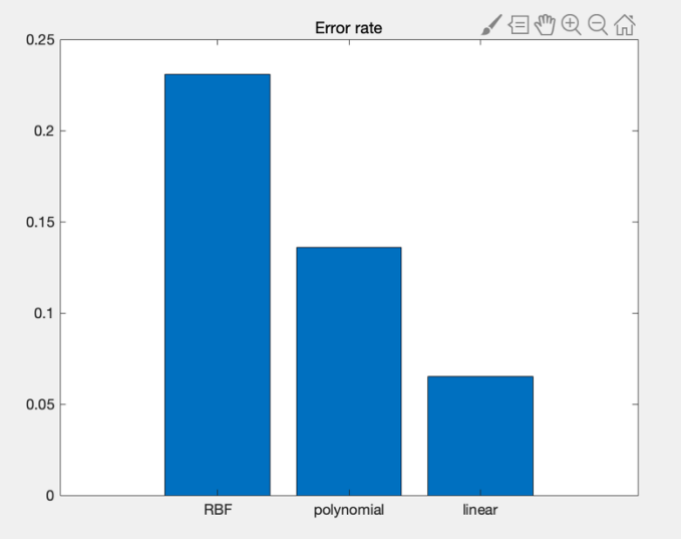
Pca3: 0.9274
Pca5: 0.9268
Pca7: 0.9224
Pca9: 0.9224
Pca11: 0.9224
Origin-data: 0.8696

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:
LOSS:

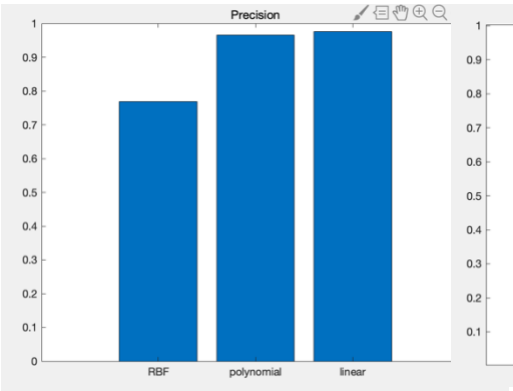


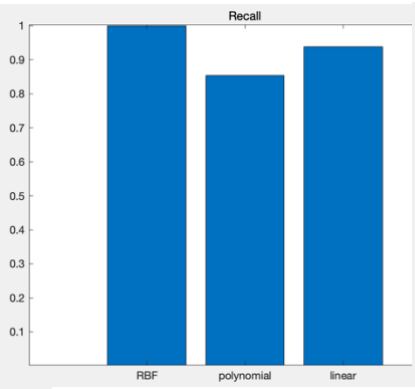
We can see that input default value RBF has highest loss.
Then calculate classification error:

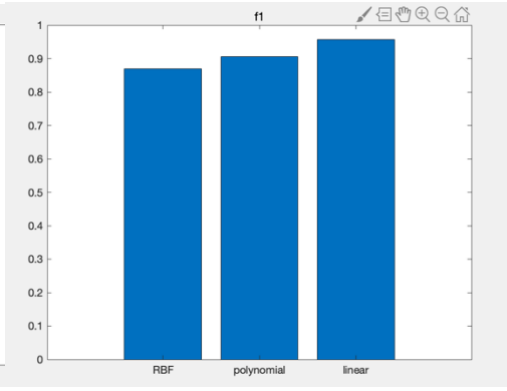


RBF has the highest error rate too.

Then calculate precision, recall and f1 value:





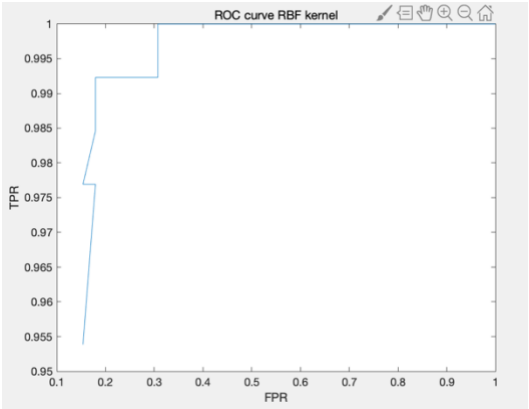


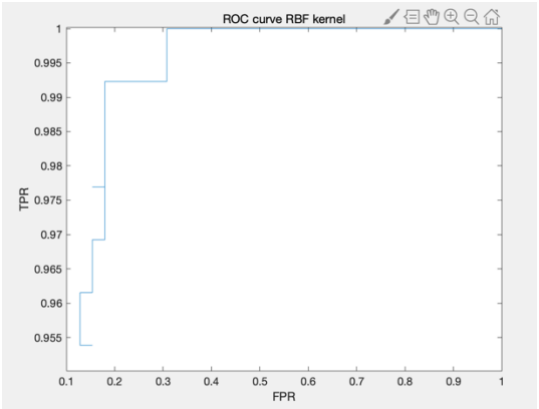
F1 :
RBF: 0.8696
Polynomial:0.9224
Linear: 0.9516

We can see that RBF still has lowest f1.

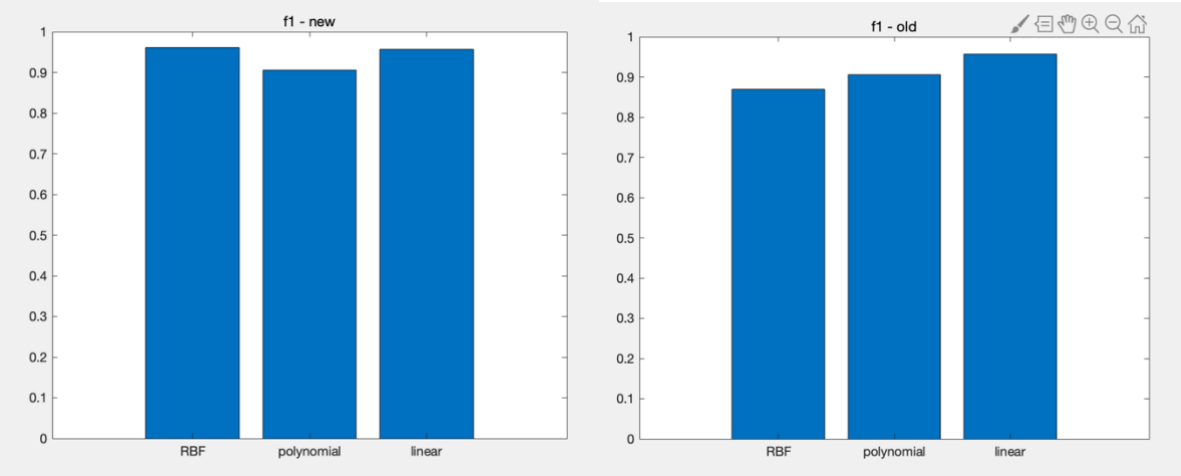
RBF has worse performance in default hyperparameter.

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.
I tried different step.





From 1 to 1000 step 0.3 (left figure)
The best kernel scale is 3.
From 1 to 1000 step 0.1 (right figure)
The best kernel scale is 1.2.



Then I use 1.2 as kernel scale.
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 2, choose 2 because in task3 we get 2 and then 1.2 when we use 1 step and 0.1 step, we know that best value should between 1 and 2.
But when I test this part, I found that the ROC curve of linear kernel is strange, then I realize that this value is for rbf kernel only. And I decide to test again, I found that best kernel scale is become 1.681 when step is 0.001, I realize that the value of step is effect if I can find the local maximum. But I don’ t have so much resource to calculate small step for each kernel, hence, I use step 0.1 and maxkernelscale 100, for other two kernel.

And then I get:

RBF: 0.9655
Polynomial: 0.9612
Linear: 0.9690
Which is obviersly higher than before.

RBF: 0.8696
Polynomial: 0.9061
Linear: 0.9569
Compare to matlab automately optimize:
RBF: 0.9575
Polynomial: 0.9568
Linear: 0.9568
I did not expect this, but my value is better than auto optimize.

Task 5:
Experience:
1, Pca can actrally help model to get higher performance by selecting difference features
2, If I want to apply a model to a data, I should transfrom the data to the data which use to train the model.
3, Cross validation, ROC curve and f1 value is highly related.
For compare decision tree and svm I record 4 value:
tree training time: 2.5947e-07,
tree predict time: 3.7253e-08,
svm trainning time: 1.6276e-06,
svm predict time: 1.5957e-07.
Svm is better than decision tree.
For f1 value:
Pca3: 0.9274
Pca5: 0.9268
Pca7: 0.9224
Pca9: 0.9224
Pca11: 0.9224
Origin-data: 0.8696

RBF: 0.9575
Polynomial: 0.9568
Linear: 0.9568
Svm is better.
For the level of overfitting, I decide to use the average of training set predict f1 minus test set predict which is

```
oflvltree = mean(ofscores-scores);
oflvlsvm = mean(ofsvmscores - svmscores);
```

And I get decision tree over fitting level is 0.0708, svm over fitting level is -0.0171. Which means that decision tree training set predict is on average bigger than test set predict 7% which is a very higher level over fitting. But svm get negative number which means svm get a low level of over fitting.

Hence, I think SVM is better in this case, since it beat decision tree on every aspect in this case.