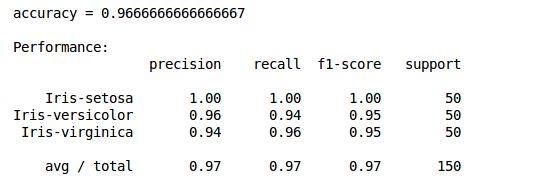
Machine Learning HW1

1. What environments the members are using:

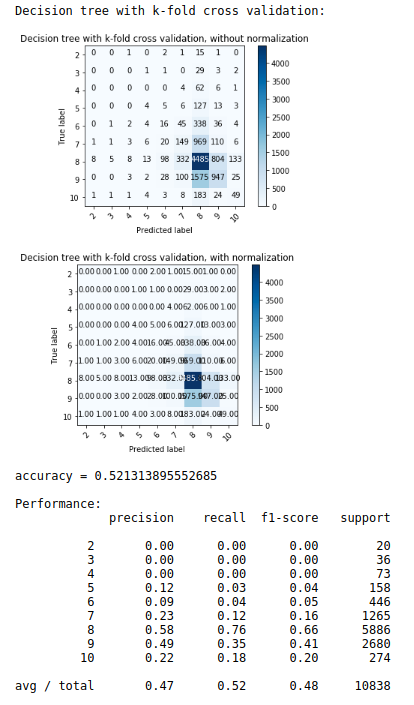
Use Jupyter as the environment

Screenshots from all the members:

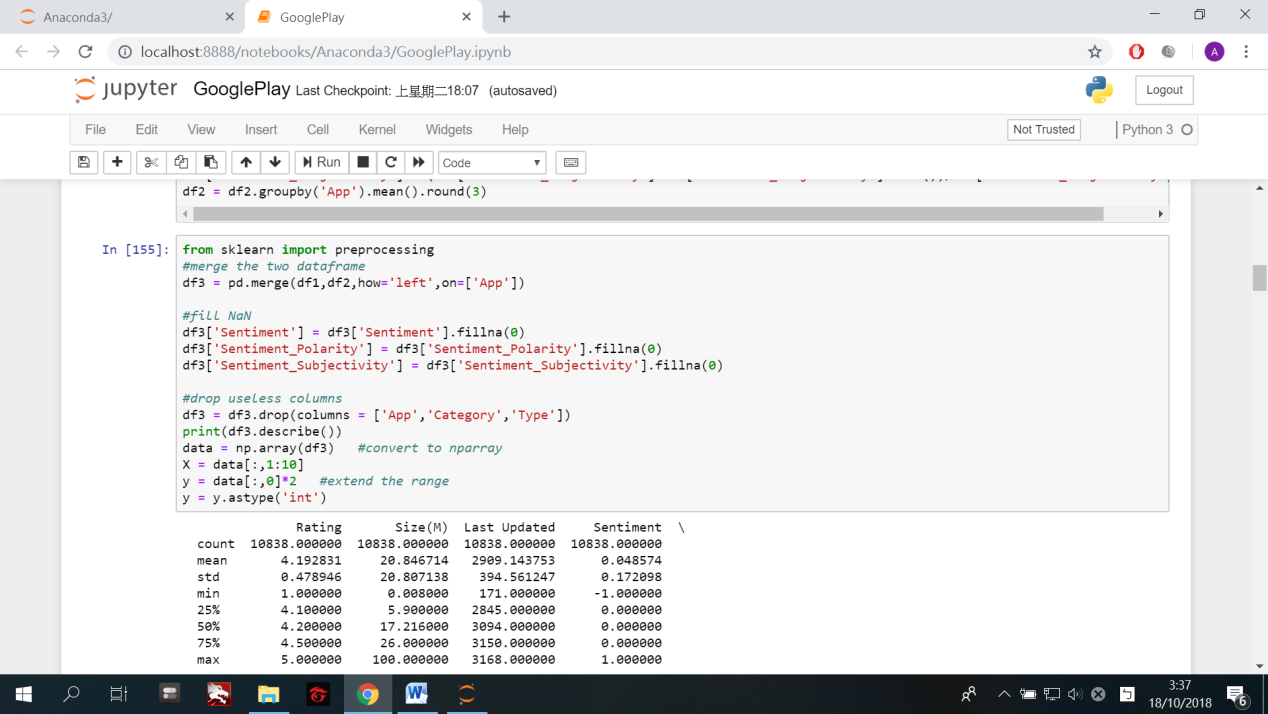
0516017 李柏毅:



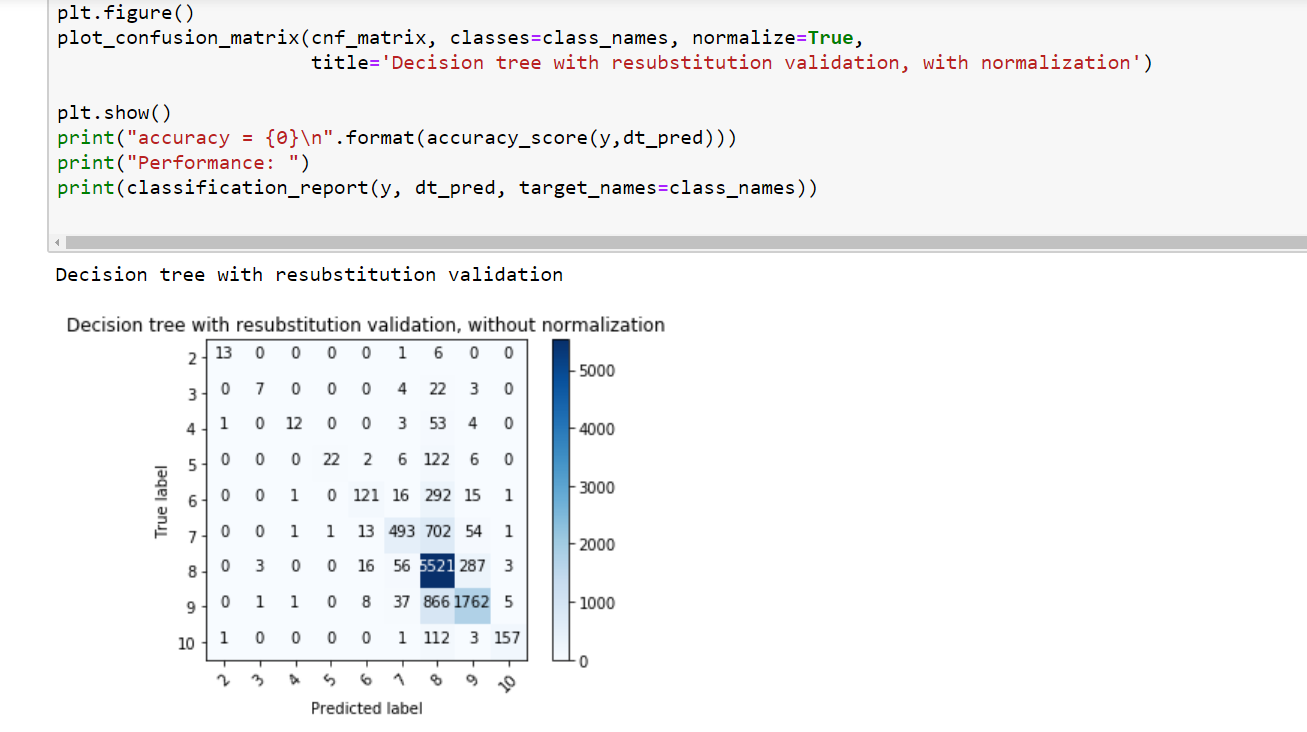
0516059 劉嘉豪:



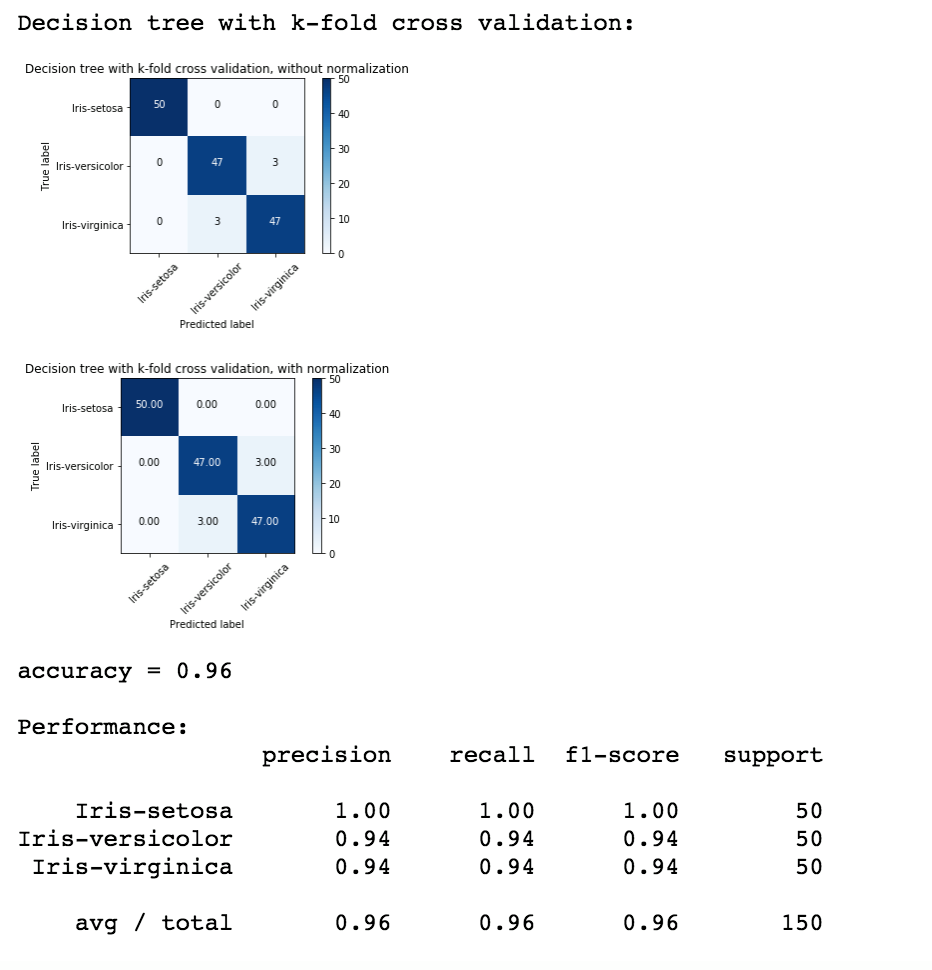
0516306 尤健羽:



0516319 傅信瑀:

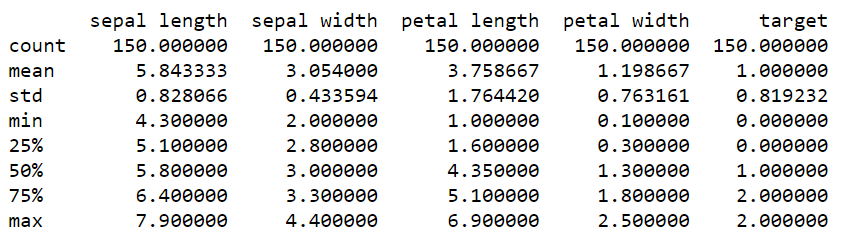


0516322 朱蝶:

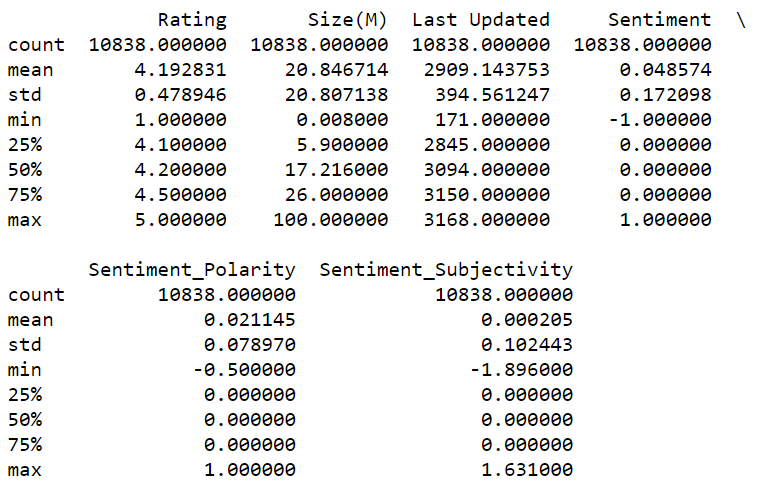


1. Basic statistic visualization of the data

Iris:



Googleplaystore:



Last updated is the number of days count from 2010/1/1

1. Data preprocessing methods:

Iris:

Only change the iris’s names from string to number 0,1,2

Googleplaystote:

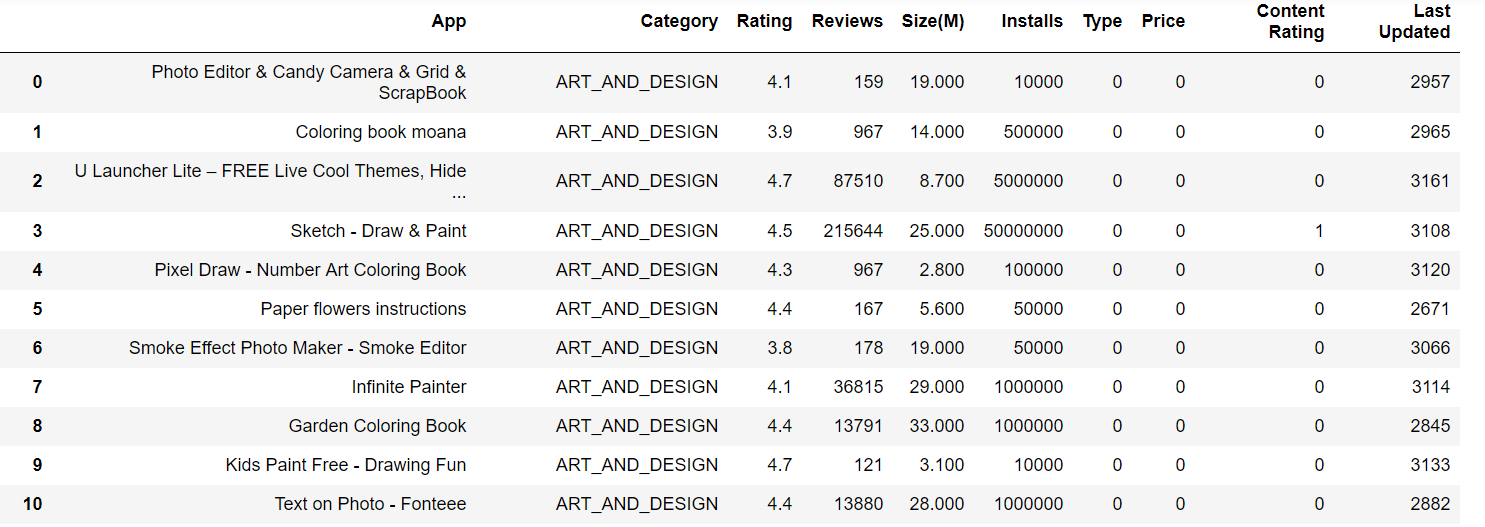
For the file ‘googleplaystore’

1. Drop column ‘App’, ‘Category’, ‘Type’, ’Genres’, ‘Current Ver’, ‘Android Ver’
2. Extend ‘Rating’ range to 2~10
3. Size: Remove end character, such as ‘M’, ‘k’.

When removing k, the number in the field divide 1000，to make sure it has the same unit with those end with’ M’

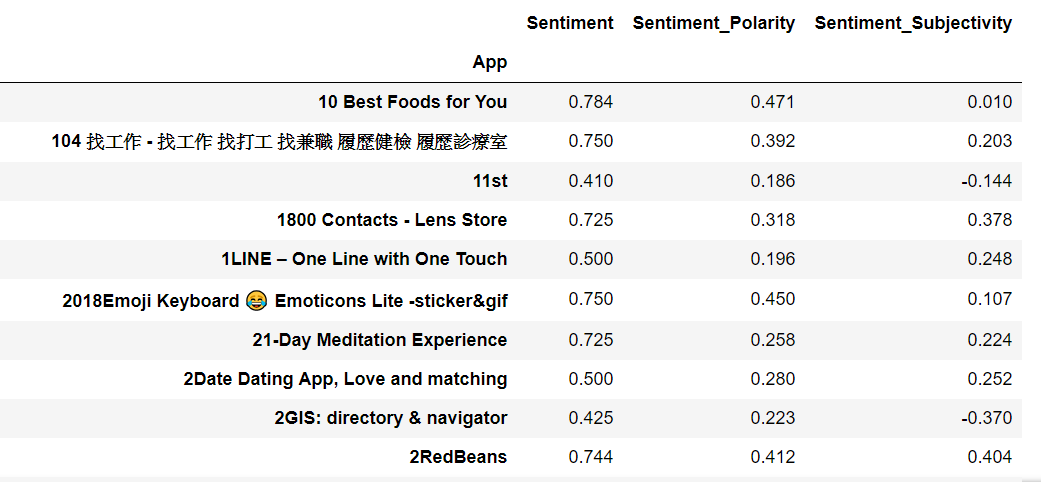
Fill ‘nan’ and ‘Varies with device’ with mean without outliers

1. Installs: remove the end character ‘+’, and ‘,’ between numbers
2. Type: ‘Free’ means ‘Price’ = 0, ‘Paid’ means ‘Price’ > 0, it has the same meaning with ‘Price’, drop this column
3. Price: remove the end character ‘$’
4. Content Rating: classified the field into six groups, rename them with number 0 to 6
5. Last updated: number of days count from 2010/1/1



For the file ‘googleplay\_users\_reviews’

1. Drop column ‘Translated\_Review’
2. Sentiment: convert ‘positive’ to 1, ‘neutral’ to 0, ‘negative’ to -1
3. Sentiment\_Subjectivity: standardize this column
4. Calculate the mean of each column group by the App name



Merge the two dataframe base on App name

1. Sentiment, Sentiment\_Polarity, Sentiment\_Subjectivity: Fill ‘nan’ with 0, because there are some App that didn’t have users reviews



Then we can start training

1. How you generate decision tree and random forest models:

Iris:

Decision tree: Use the model “DecisionTreeClassifier” from sklearn.

Random forest: Choose three out of four features everytime and create four decision trees.

Googleplaystore:

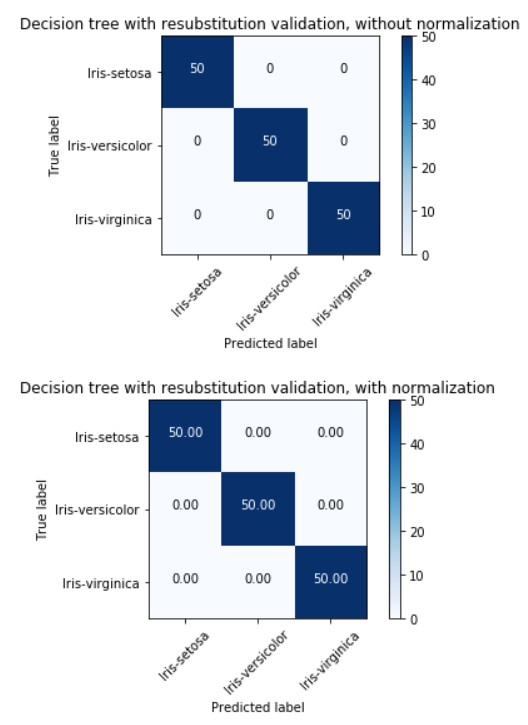
Decision tree: Use the model “DecisionTreeClassifier” from sklearn.

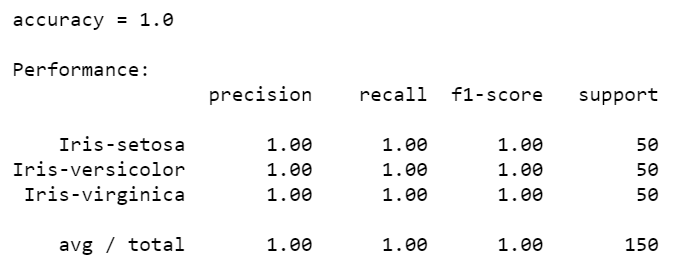
Random forest: Choose seven out of nine features everytime and create thirty-six decision trees.

1. The performance:

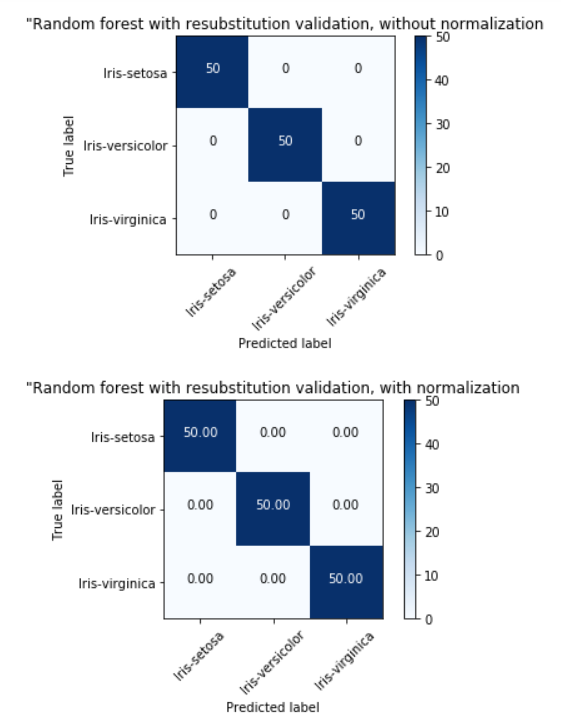
Iris:

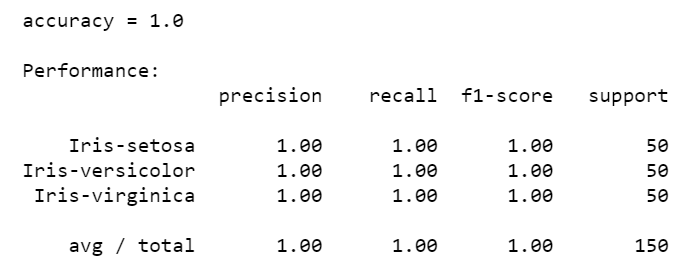
1. Decision tree with resubstitution validation:



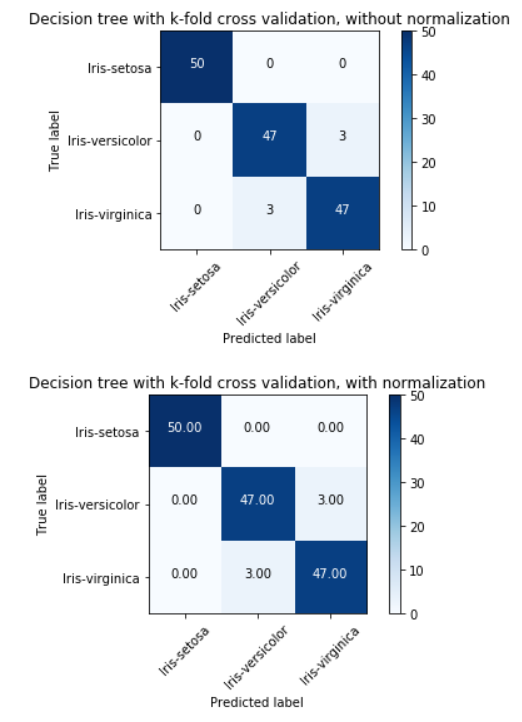


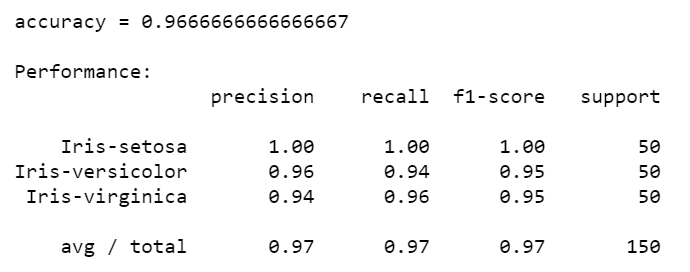
1. Random forest with resubstitution validation:



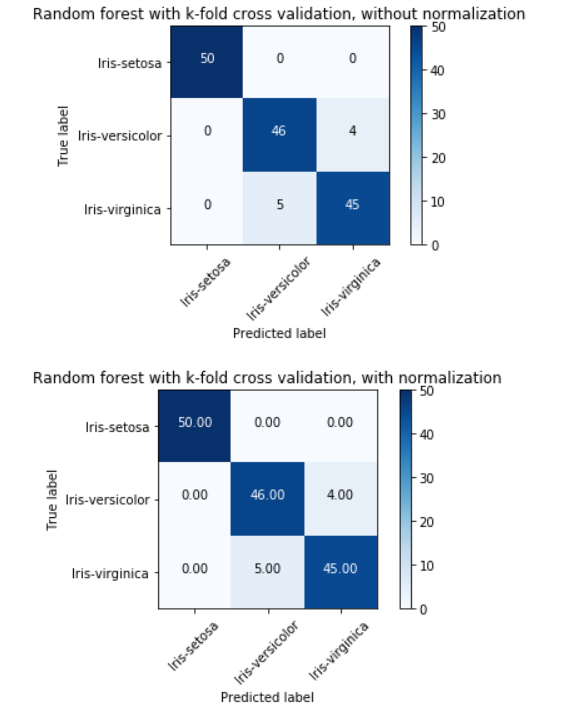


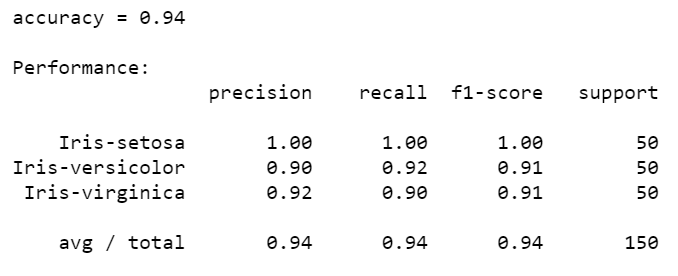
1. Decision tree with k-fold cross validation:





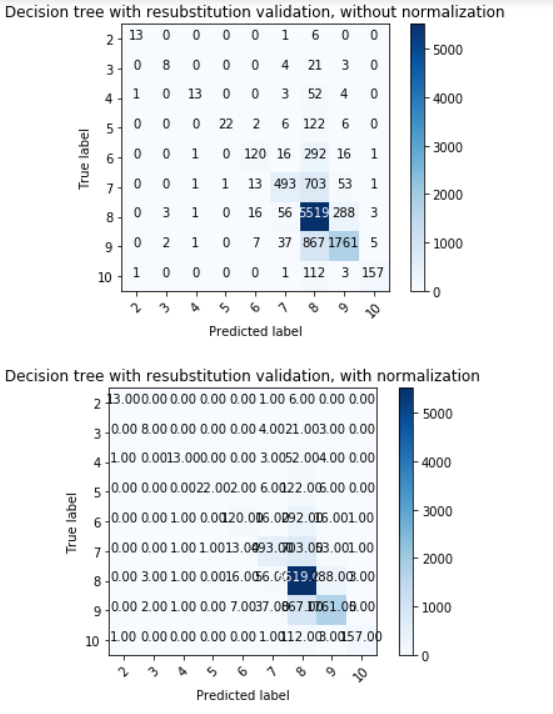
1. Random forest with k-fold cross validation:

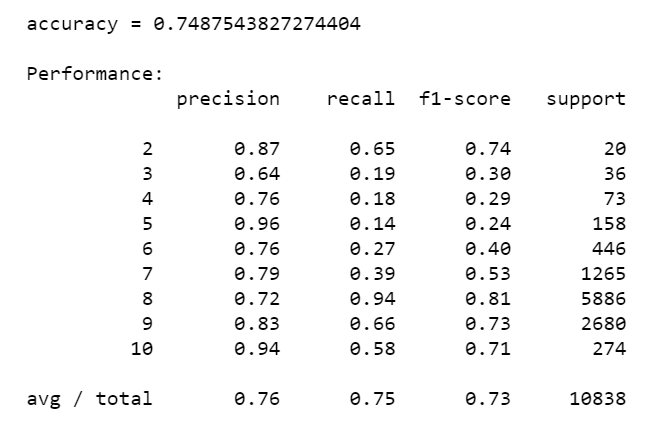




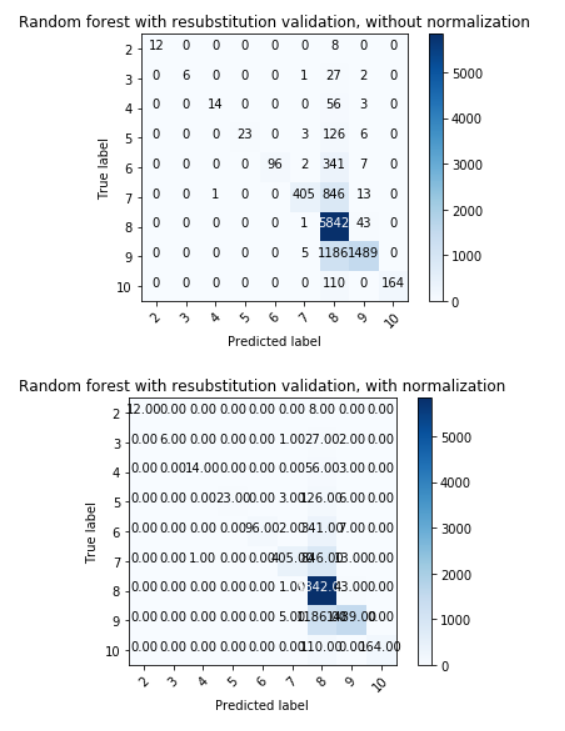
Googleplaystore:

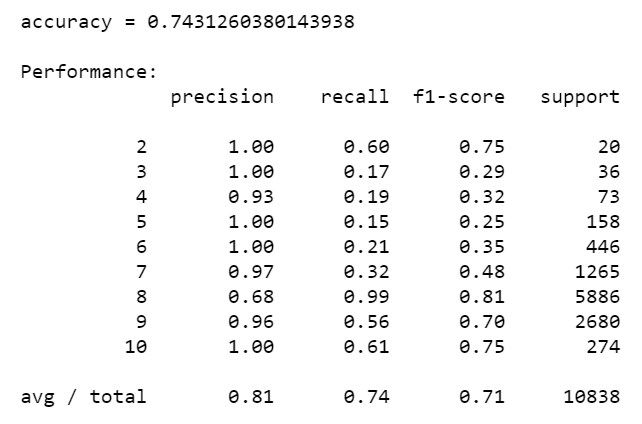
1. Decision tree with resubstitution validation: (tree max\_depth = 15)



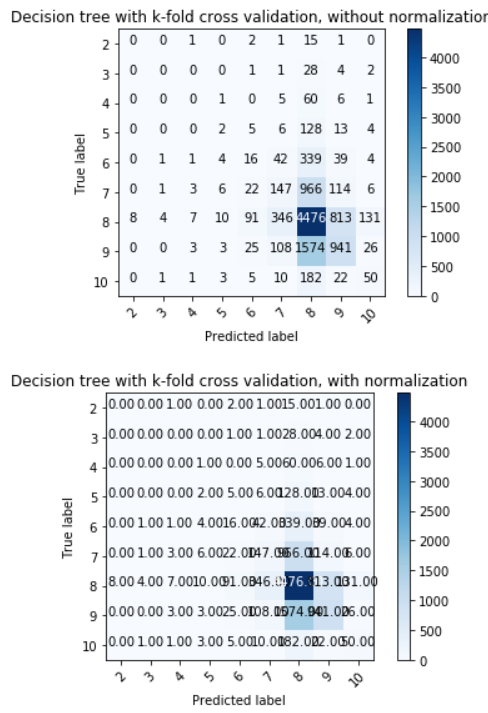


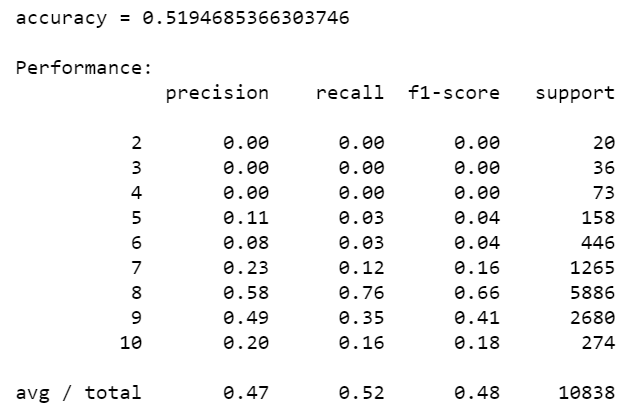
1. Random forest with resubstitution validation: (tree max\_depth = 15)



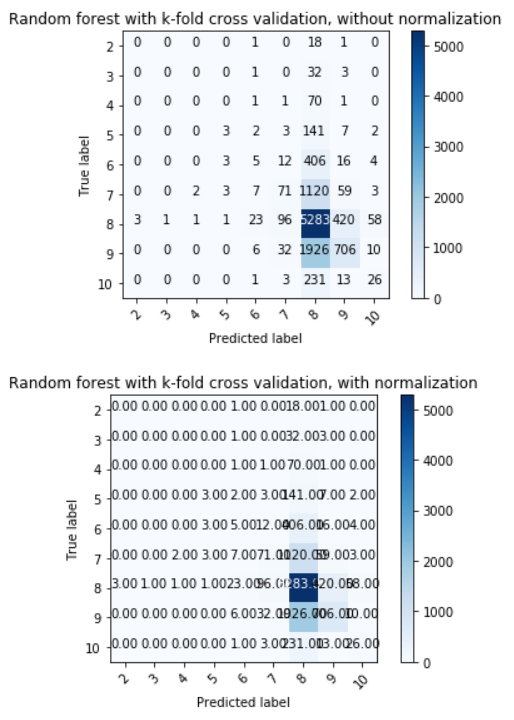


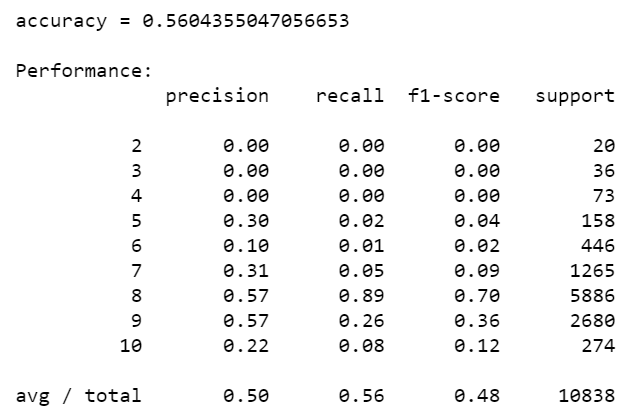
1. Decision tree with k-fold cross validation: (tree max\_depth = 15, k = 10)





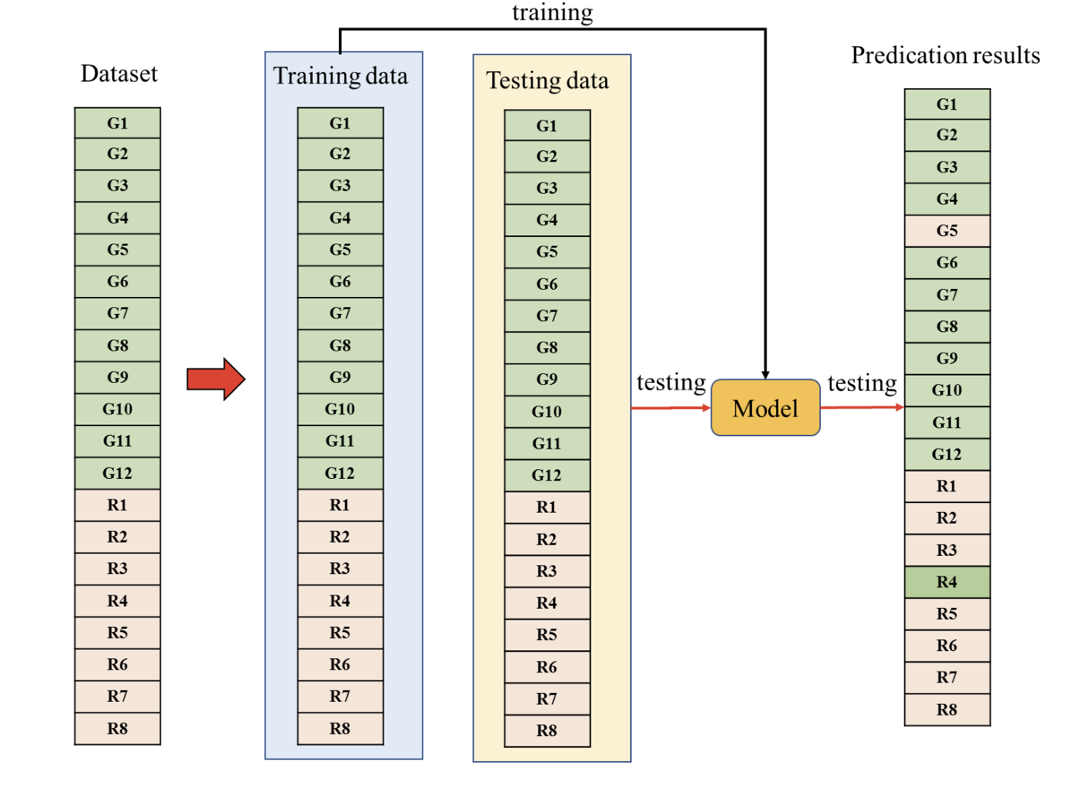
1. Random forest with k-fold cross validation: (tree max\_depth = 15, k = 10)



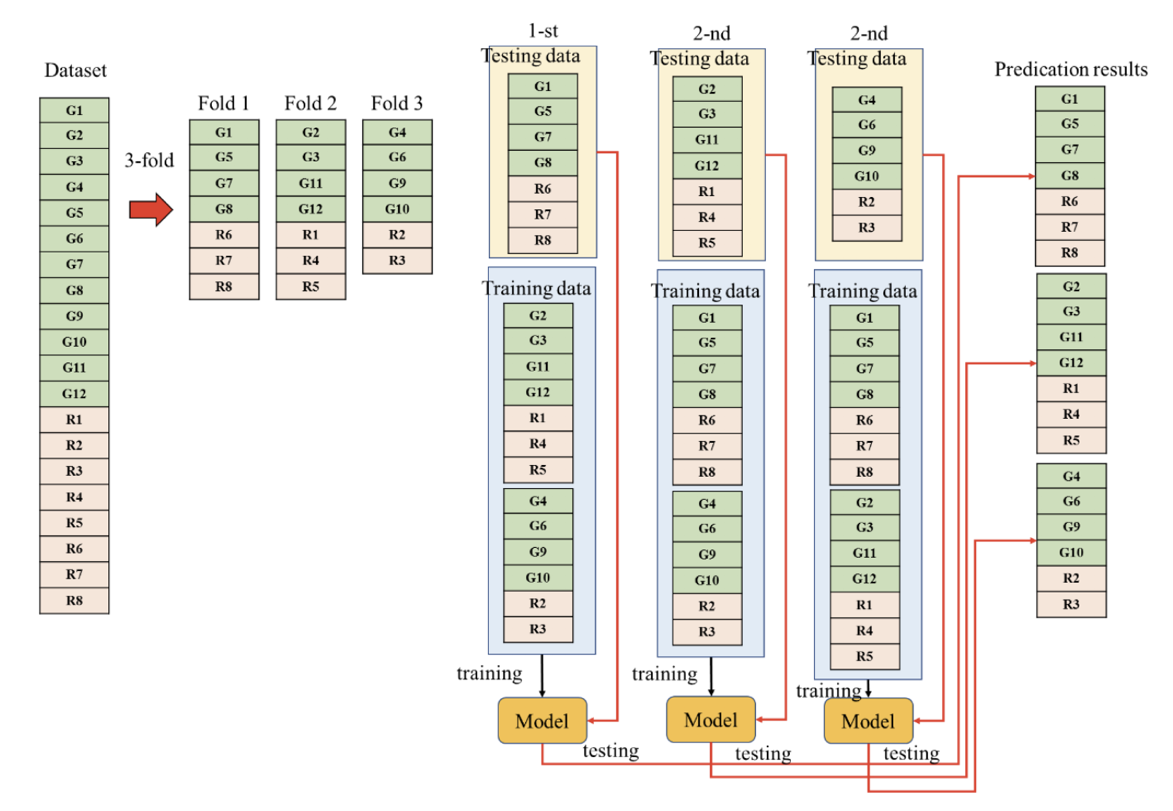


1. Conclusion:

From the result of GooglePlay dataset, we can observe that the accuracy using resubstitution is much higher than the one using K-fold CV. We think that it is because the training and testing data are the same in resubstitution method, just like the picture below.



On the other hand, the K-fold method is like the following picture.



We can see that the data are randomly divided into k groups (in the example above, k is 3), and use one group to be the testing data, the remaining k-1 groups be the training data, until each group is used to be the testing data once.

In comparison, we can conclude that the accuracy of the resubstitution method will certainly be higher since the training data is used as the testing data, so if the model is well-trained, the result will certainly be good. Although this method is faster, we think it is vulnerable to new data, since it just know how to deal with familiar ones. As for the K-fold method, we use not just once, but many times’ cross validation to train our model, that’s why the accuracy is lower.

Compare the differences between decision tree and random forest, the last two pictures of GooglePlayStore show that random forest has better accuracy than decision tree. Although there’s only a tiny gap, we can still see the difference, perhaps next time we can do better on the data preprocessing part.