Machine Learning HW1

1. What environments the members are using:

Use Jupyter as the environment

Screenshots from all the members:

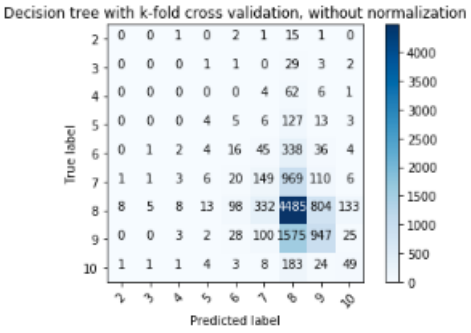
0516017 李柏毅:

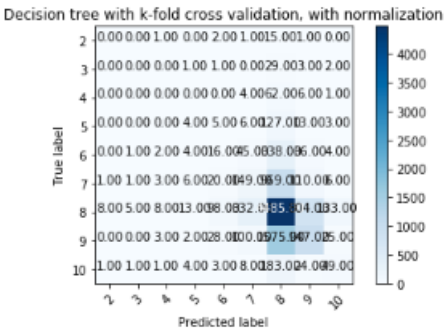
```
accuracy = 0.9666666666666667

Performance:
```

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	50
Iris-versicolor	0.96	0.94	0.95	50
Iris-virginica	0.94	0.96	0.95	50
avg / total	0.97	0.97	0.97	150

0516059 劉嘉豪:

```
Decision tree with k-fold cross validation:
```

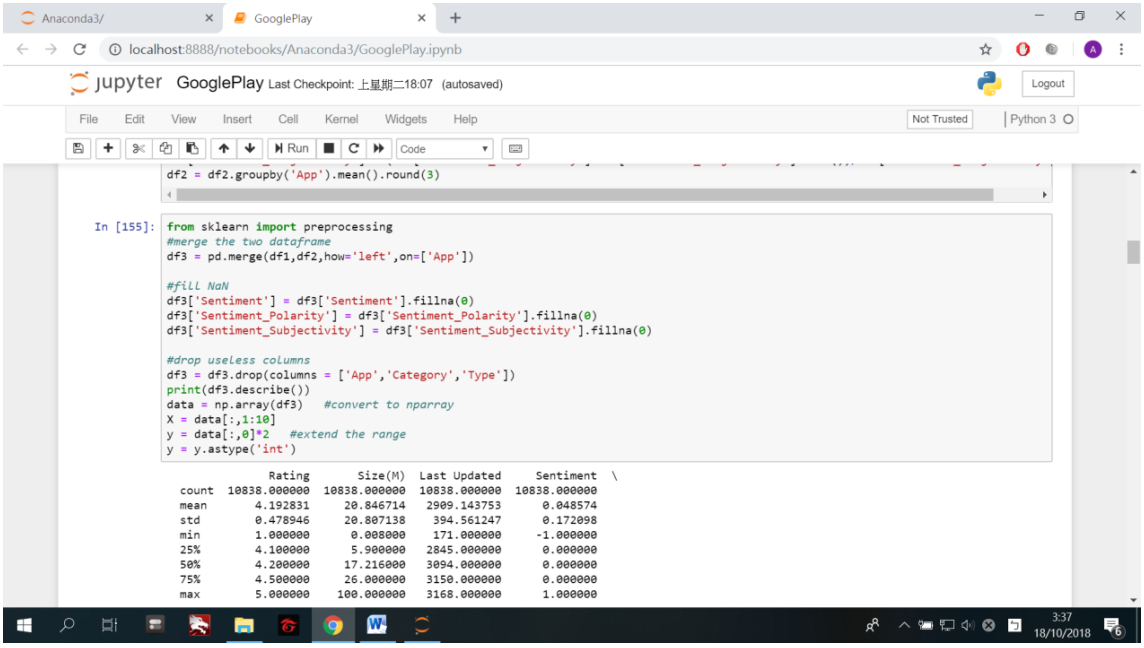





```
accuracy = 0.521313895552685

Performance:
```

	precision	recall	f1-score	support
2	0.00	0.00	0.00	20
3	0.00	0.00	0.00	36
4	0.00	0.00	0.00	73
5	0.12	0.03	0.04	158
6	0.09	0.04	0.05	446
7	0.23	0.12	0.16	1265
8	0.58	0.76	0.66	5886
9	0.49	0.35	0.41	2680
10	0.22	0.18	0.20	274
avg / total	0.47	0.52	0.48	10838

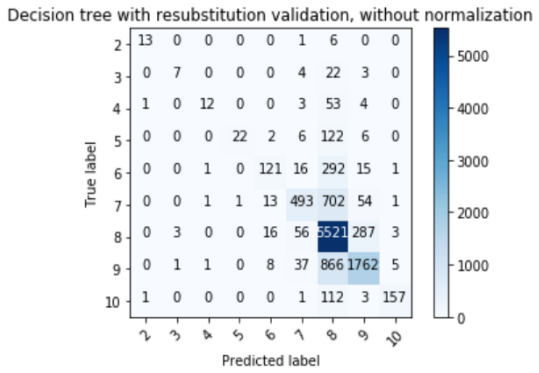
0516306 尤健羽:



0516319 傅信瑀:

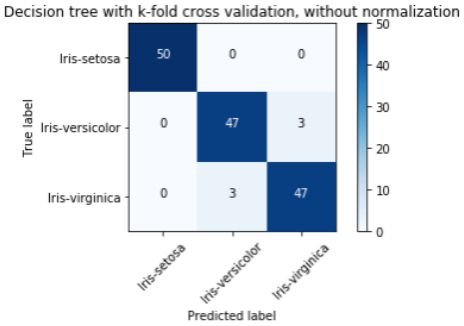
```
plt.figure()
plot_confusion_matrix(cnf_matrix, classes=class_names, normalize=True,
                      title='Decision tree with resubstitution validation, with normalization')

plt.show()
print("accuracy = {0}\n".format(accuracy_score(y,dt_pred)))
print("Performance: ")
print(classification_report(y, dt_pred, target_names=class_names))
```

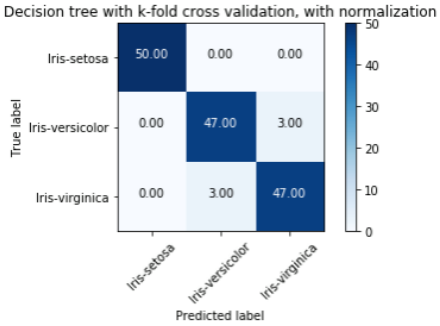
```
Decision tree with resubstitution validation
```



0516322 朱蝶:

```
Decision tree with k-fold cross validation:
```






```
accuracy = 0.96

Performance:
                precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        50
Iris-versicolor       0.94      0.94      0.94        50
 Iris-virginica       0.94      0.94      0.94        50

    avg / total       0.96      0.96      0.96       150
```

2. Basic statistic visualization of the data

Iris:

	sepal length	sepal width	petal length	petal width	target
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667	1.000000
std	0.828066	0.433594	1.764420	0.763161	0.819232
min	4.300000	2.000000	1.000000	0.100000	0.000000
25%	5.100000	2.800000	1.600000	0.300000	0.000000
50%	5.800000	3.000000	4.350000	1.300000	1.000000
75%	6.400000	3.300000	5.100000	1.800000	2.000000
max	7.900000	4.400000	6.900000	2.500000	2.000000

Googleplaystore:

	Rating	Size(M)	Last Updated	Sentiment	\
count	10838.000000	10838.000000	10838.000000	10838.000000	
mean	4.192831	20.846714	2909.143753	0.048574	
std	0.478946	20.807138	394.561247	0.172098	
min	1.000000	0.008000	171.000000	-1.000000	
25%	4.100000	5.900000	2845.000000	0.000000	
50%	4.200000	17.216000	3094.000000	0.000000	
75%	4.500000	26.000000	3150.000000	0.000000	
max	5.000000	100.000000	3168.000000	1.000000	

	Sentiment_Polarity	Sentiment_Subjectivity
count	10838.000000	10838.000000
mean	0.021145	0.000205
std	0.078970	0.102443
min	-0.500000	-1.896000
25%	0.000000	0.000000
50%	0.000000	0.000000
75%	0.000000	0.000000
max	1.000000	1.631000

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

3. Data preprocessing methods:

Iris:

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

Googleplaystote:

For the file 'googleplaystore'

(a) Drop column 'App', 'Category', 'Type', 'Genres', 'Current Ver', 'Android Ver'

(b) Extend 'Rating' range to 2~10

(c) 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

(d) Installs: remove the end character '+', and ',' between numbers

(e) Type: 'Free' means 'Price' = 0, 'Paid' means 'Price' > 0, it has the same meaning with 'Price', drop this column

(f) Price: remove the end character '$'

(g) Content Rating: classified the field into six groups, rename them with number 0 to 6

(h) Last updated: number of days count from 2010/1/1

	App	Category	Rating	Reviews	Size(M)	Installs	Type	Price	Content Rating	Last Updated
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19.000	10000	0	0	0	2957
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14.000	500000	0	0	0	2965
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.700	5000000	0	0	0	3161
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25.000	50000000	0	0	1	3108
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.800	100000	0	0	0	3120
5	Paper flowers instructions	ART_AND_DESIGN	4.4	167	5.600	50000	0	0	0	2671
6	Smoke Effect Photo Maker - Smoke Editor	ART_AND_DESIGN	3.8	178	19.000	50000	0	0	0	3066
7	Infinite Painter	ART_AND_DESIGN	4.1	36815	29.000	1000000	0	0	0	3114
8	Garden Coloring Book	ART_AND_DESIGN	4.4	13791	33.000	1000000	0	0	0	2845
9	Kids Paint Free - Drawing Fun	ART_AND_DESIGN	4.7	121	3.100	10000	0	0	0	3133
10	Text on Photo - Fonteee	ART_AND_DESIGN	4.4	13880	28.000	1000000	0	0	0	2882

For the file ‘googleplay_users_reviews’

(a) Drop column ‘Translated_Review’

(b) Sentiment: convert ‘positive’ to 1, ‘neutral’ to 0, ‘negative’ to -1

(c) Sentiment_Subjectivity: standardize this column

(d) Calculate the mean of each column group by the App name

App	Sentiment	Sentiment_Polarity	Sentiment_Subjectivity
10 Best Foods for You	0.784	0.471	0.010
104 找工作 - 找工作 找打工 找兼職 履歷健檢 履歷診療室	0.750	0.392	0.203
11st	0.410	0.186	-0.144
1800 Contacts - Lens Store	0.725	0.318	0.378
1LINE – One Line with One Touch	0.500	0.196	0.248
2018Emoji Keyboard 😂 Emoticons Lite -sticker&gif	0.750	0.450	0.107
21-Day Meditation Experience	0.725	0.258	0.224
2Date Dating App, Love and matching	0.500	0.280	0.252
2GIS: directory & navigator	0.425	0.223	-0.370
2RedBeans	0.744	0.412	0.404

Merge the two dataframe base on App name

(a) Sentiment, Sentiment_Polarity, Sentiment_Subjectivity: Fill ‘nan’ with 0, because there are some App that didn’t have users reviews

	Rating	Reviews	Size(M)	Installs	Type	Price	Content Rating	Last Updated	Sentiment	Sentiment_Polarity	Sentiment_Subjectivity
0	4.1	159.0	19.000	10000.0	0.0	0.0	0.0	2957.0	0.000	0.000	0.000
1	3.9	967.0	14.000	500000.0	0.0	0.0	0.0	2965.0	0.273	0.153	0.572
2	4.7	87510.0	8.700	5000000.0	0.0	0.0	0.0	3161.0	0.000	0.000	0.000
3	4.5	215644.0	25.000	50000000.0	0.0	0.0	1.0	3108.0	0.000	0.000	0.000
4	4.3	967.0	2.800	100000.0	0.0	0.0	0.0	3120.0	0.000	0.000	0.000
5	4.4	167.0	5.600	50000.0	0.0	0.0	0.0	2671.0	0.000	0.000	0.000
6	3.8	178.0	19.000	50000.0	0.0	0.0	0.0	3066.0	0.000	0.000	0.000
7	4.1	36815.0	29.000	1000000.0	0.0	0.0	0.0	3114.0	0.000	0.000	0.000
8	4.4	13791.0	33.000	1000000.0	0.0	0.0	0.0	2845.0	0.600	0.268	0.118

Then we can start training

4. 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:

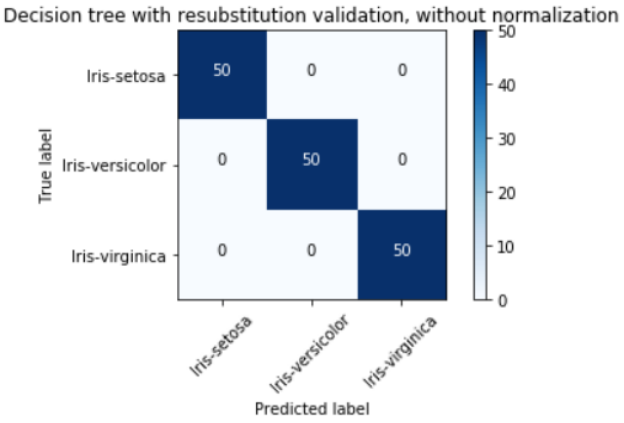
Decision tree: Use the model “DecisionTreeClassifier” from sklearn.

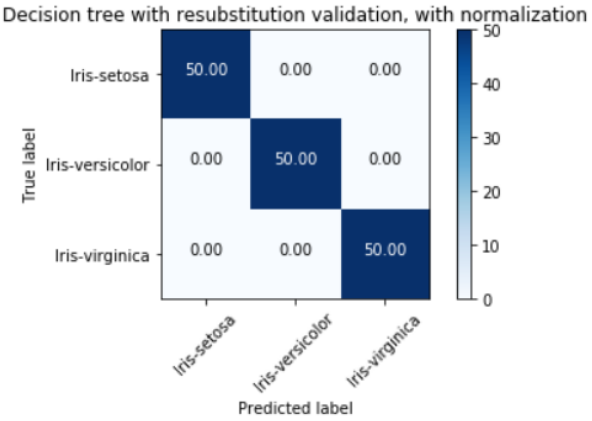
Random forest: Choose seven out of nine features everytime and create thirty-six decision trees.

5. The performance:

Iris:

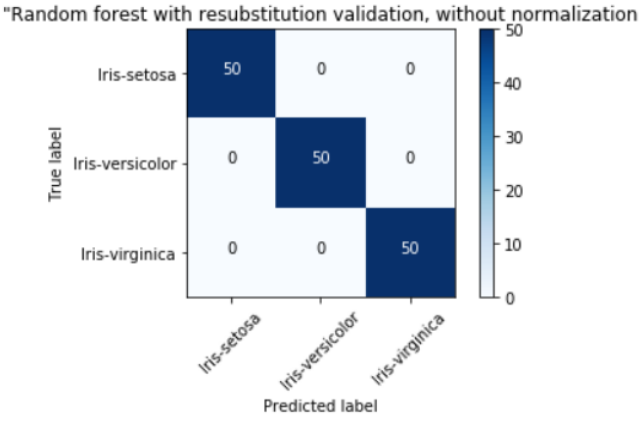
1. Decision tree with resubstitution validation:

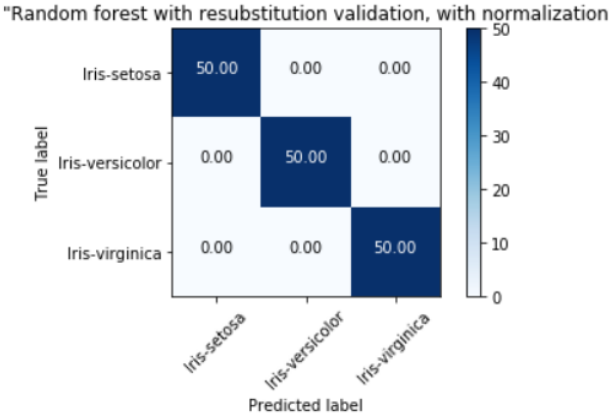





```
accuracy = 1.0

Performance:
                precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        50
Iris-versicolor       1.00      1.00      1.00        50
 Iris-virginica       1.00      1.00      1.00        50

    avg / total       1.00      1.00      1.00       150
```

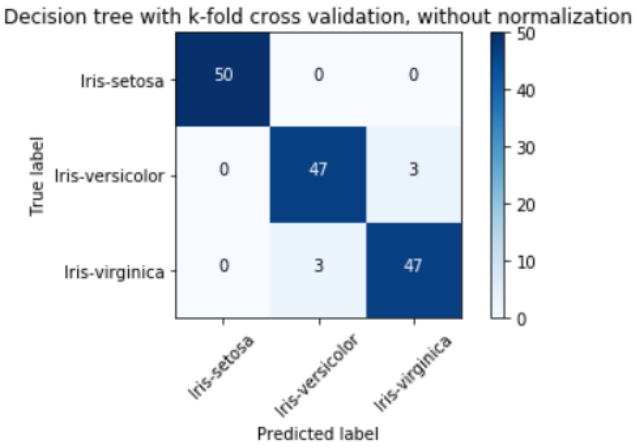
2. Random forest with resubstitution validation:

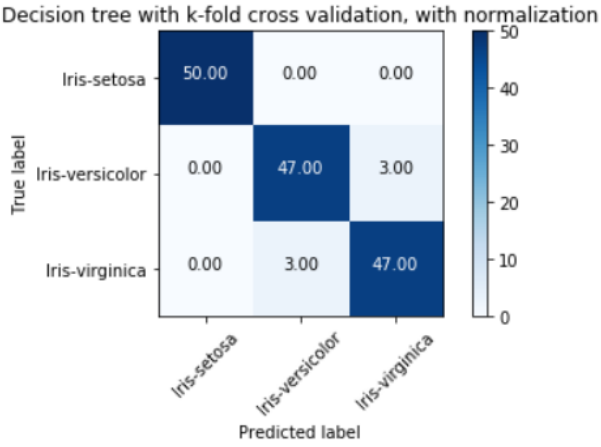





```
accuracy = 1.0

Performance:
                precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        50
Iris-versicolor       1.00      1.00      1.00        50
 Iris-virginica       1.00      1.00      1.00        50

    avg / total       1.00      1.00      1.00       150
```

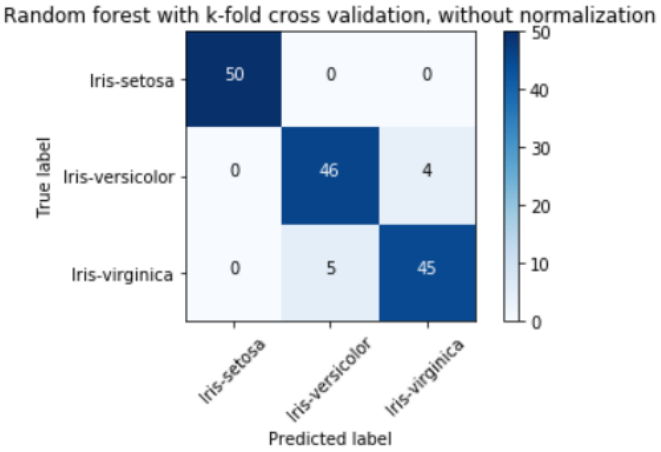
3. Decision tree with k-fold cross validation:

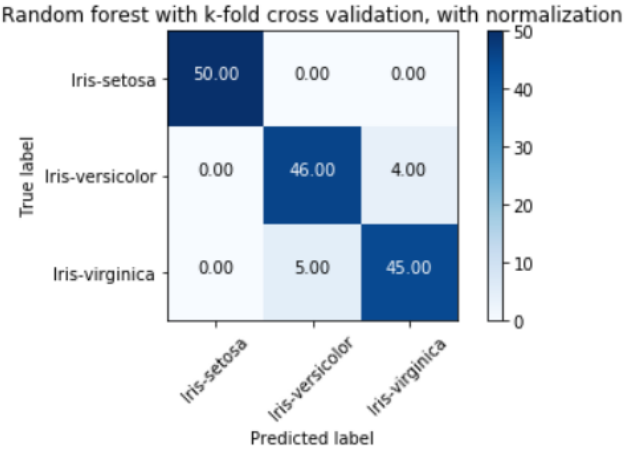





```
accuracy = 0.9666666666666667

Performance:
                precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        50
Iris-versicolor       0.96      0.94      0.95        50
 Iris-virginica       0.94      0.96      0.95        50

    avg / total       0.97      0.97      0.97       150
```

4. Random forest with k-fold cross validation:

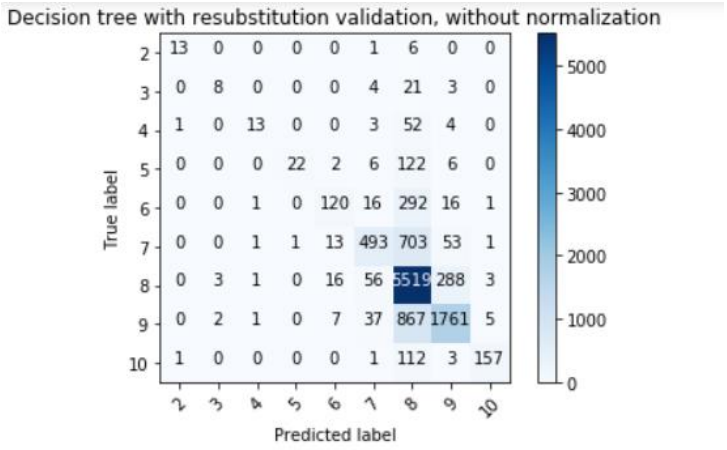


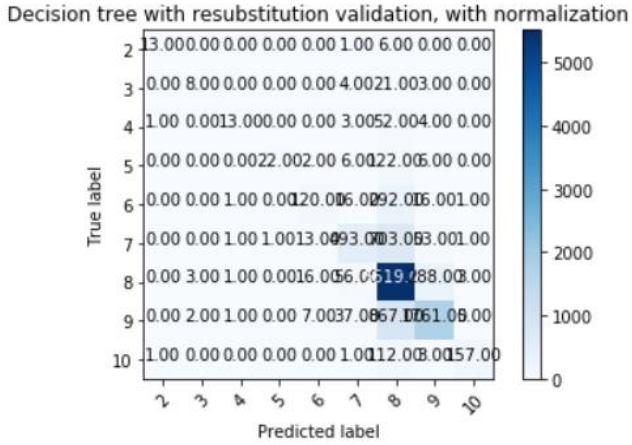



```
accuracy = 0.94

Performance:
                precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        50
Iris-versicolor       0.90      0.92      0.91        50
 Iris-virginica       0.92      0.90      0.91        50

    avg / total       0.94      0.94      0.94       150
```

Googleplaystore:

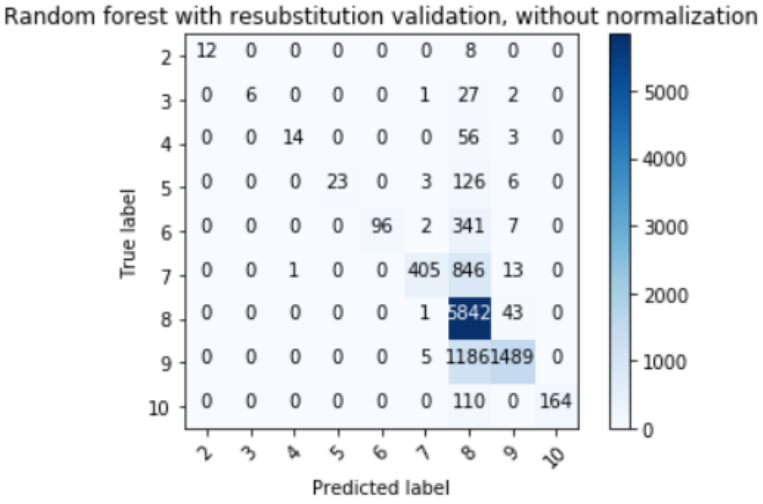
1. Decision tree with resubstitution validation: (tree max_depth = 15)

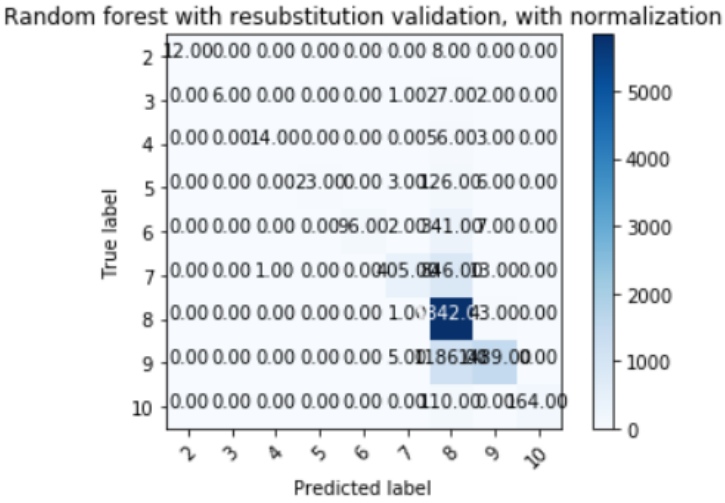





```
accuracy = 0.7487543827274404

Performance:
              precision    recall  f1-score   support

           2       0.87      0.65      0.74        20
           3       0.64      0.19      0.30        36
           4       0.76      0.18      0.29        73
           5       0.96      0.14      0.24       158
           6       0.76      0.27      0.40       446
           7       0.79      0.39      0.53      1265
           8       0.72      0.94      0.81      5886
           9       0.83      0.66      0.73      2680
          10       0.94      0.58      0.71       274

avg / total       0.76      0.75      0.73     10838
```

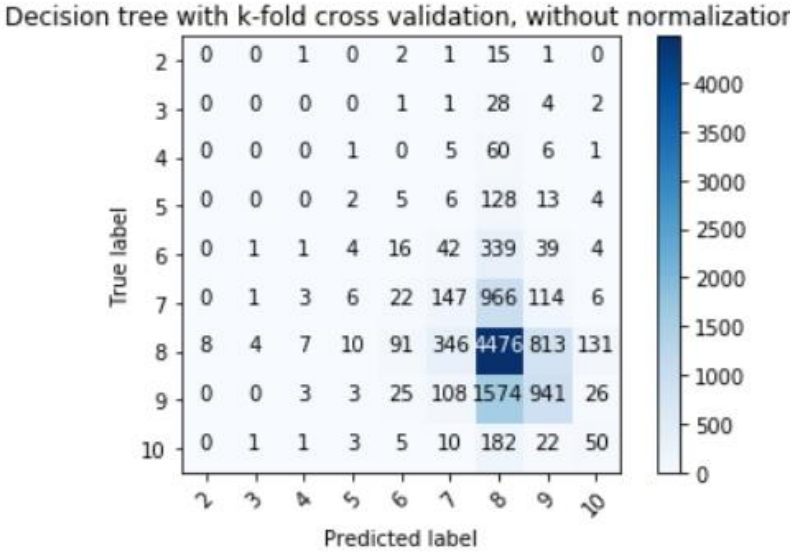
2. Random forest with resubstitution validation: (tree max_depth = 15)

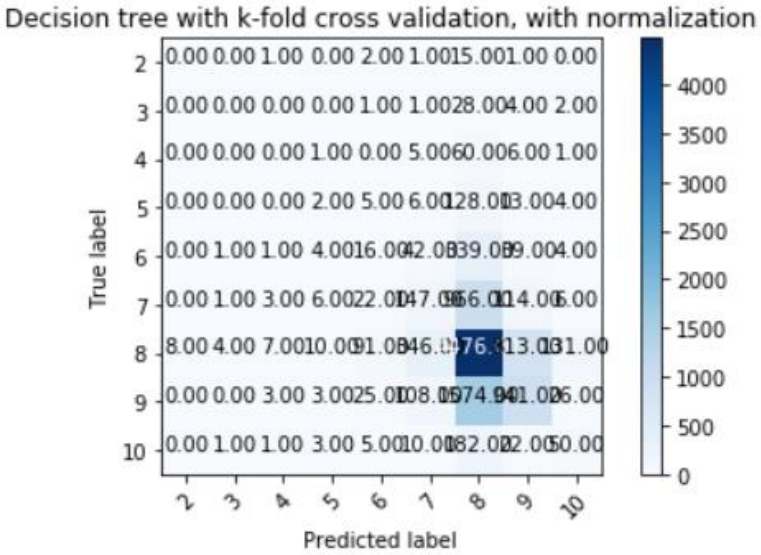





```
accuracy = 0.7431260380143938

Performance:
              precision    recall  f1-score   support

           2       1.00      0.60      0.75        20
           3       1.00      0.17      0.29        36
           4       0.93      0.19      0.32        73
           5       1.00      0.15      0.25       158
           6       1.00      0.21      0.35       446
           7       0.97      0.32      0.48      1265
           8       0.68      0.99      0.81      5886
           9       0.96      0.56      0.70      2680
          10       1.00      0.61      0.75       274

avg / total       0.81      0.74      0.71     10838
```

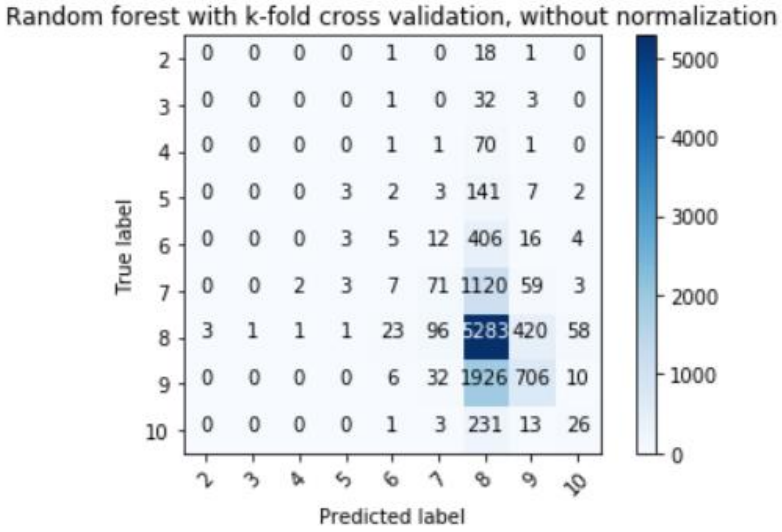
3. Decision tree with k-fold cross validation: (tree max_depth = 15, k = 10)

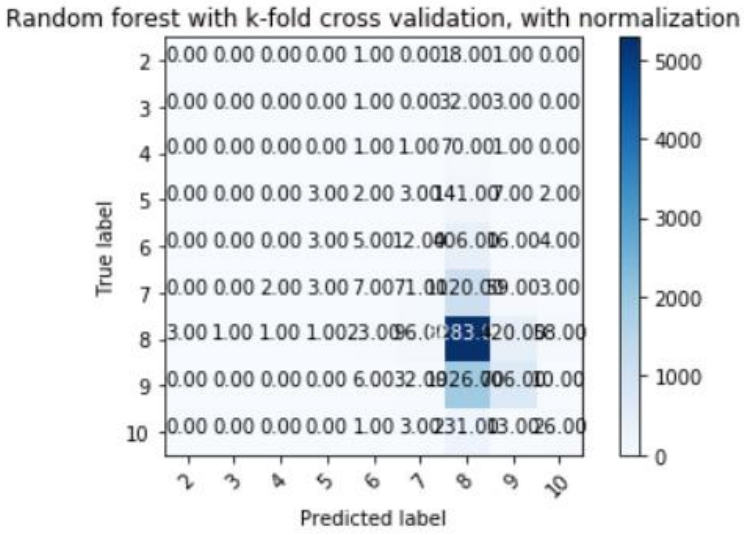





```
accuracy = 0.5194685366303746

Performance:
              precision    recall  f1-score   support

           2       0.00      0.00      0.00        20
           3       0.00      0.00      0.00        36
           4       0.00      0.00      0.00        73
           5       0.11      0.03      0.04       158
           6       0.08      0.03      0.04       446
           7       0.23      0.12      0.16      1265
           8       0.58      0.76      0.66      5886
           9       0.49      0.35      0.41      2680
          10       0.20      0.16      0.18       274

avg / total       0.47      0.52      0.48     10838
```

4. Random forest with k-fold cross validation: (tree max_depth = 15, k = 10)

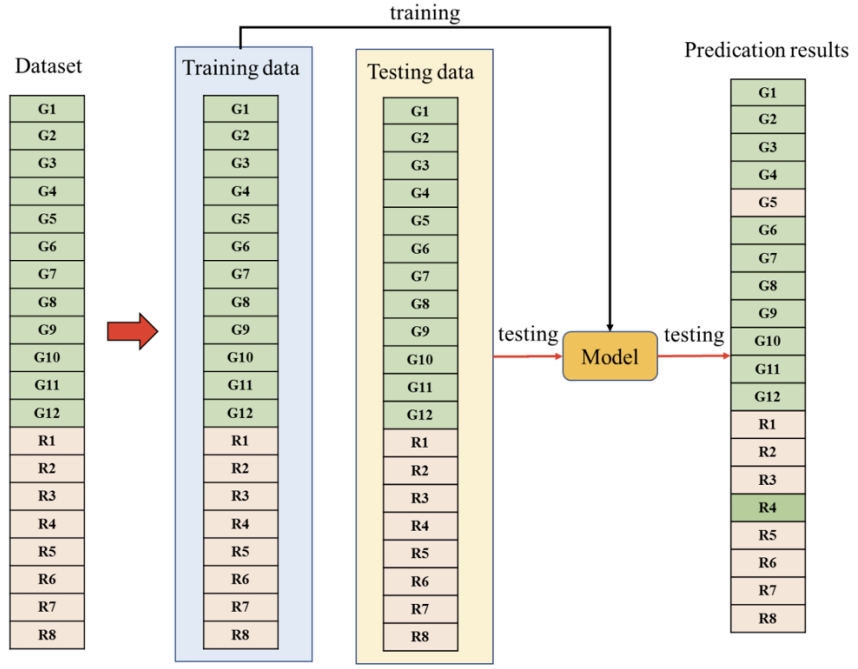




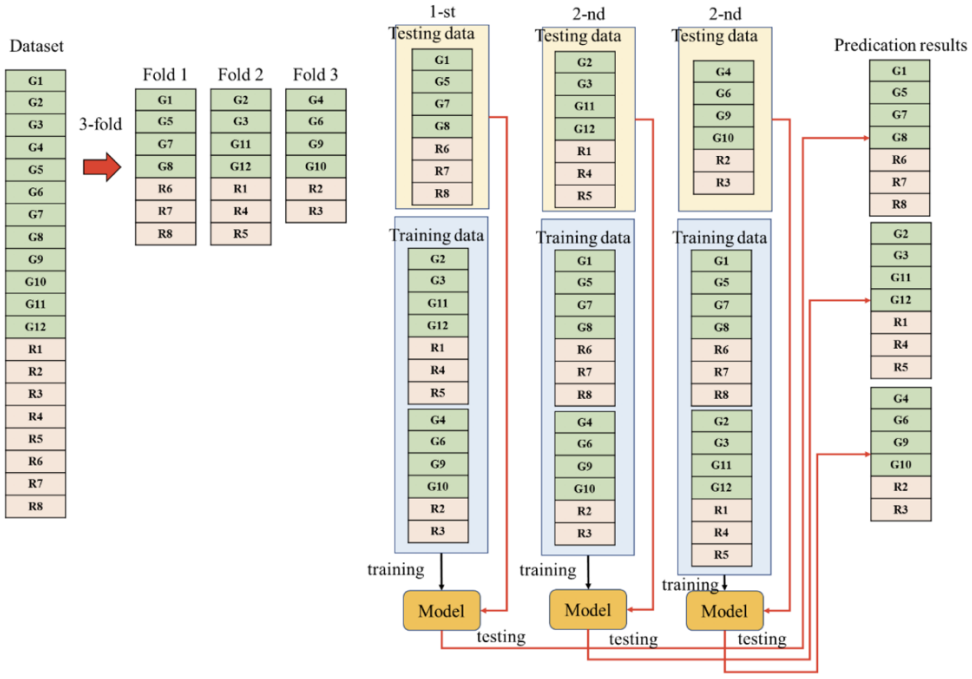

```
accuracy = 0.5604355047056653

Performance:
              precision    recall  f1-score   support

           2       0.00      0.00      0.00        20
           3       0.00      0.00      0.00        36
           4       0.00      0.00      0.00        73
           5       0.30      0.02      0.04       158
           6       0.10      0.01      0.02       446
           7       0.31      0.05      0.09      1265
           8       0.57      0.89      0.70      5886
           9       0.57      0.26      0.36      2680
          10       0.22      0.08      0.12       274

avg / total       0.50      0.56      0.48     10838
```

6. 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.