Report

第 25 組：吳東鴻、曾靖驊、孫念恩、謝富丞、鄭益和

1. What environments the members are using?

- 使用 Python 3

2. Basic statistic visualization of the data?

(1) Iris

- 原始資料：iris.csv
- 程式：iris.py
- 執行程式所產生的檔案：iris_data_quality_report.csv，preprocessed_iris.csv
- Basic statistic visualization：

iris_data_quality_report.csv 資料如下：

1	Continuous Features									
2	Feature	Count	Miss%	Card	Min	1st Qrt	Mean	Median	3rd Qrt	Max	Std Dev
3	sepal length	150	0	35	4.3	5.1	5.84	5.8	6.4	7.9	0.83
4	sepal width	150	0	23	2	2.8	3.05	3	3.3	4.4	0.43
5	petal length	150	0	43	1	1.6	3.76	4.35	5.1	6.9	1.76
6	petal width	150	0	22	0.1	0.3	1.2	1.3	1.8	2.5	0.76
7											
8	Categorical Features										
9	Feature	Count	Miss%	Card	Mode	Mode Freq	Mode%	2nd Mode	2nd Mode	2nd Mode%	
10	class	150	0	3	Iris-setosa	50	33.33	Iris-versicolor	50	33.33	

說明：計算 Continuous Features 之個數、資料遺失率、相異值個數、最小值、第一四分位數、算術平均數、中位數、第三四分位數、最大值及標準差。計算 Categorical Features 之個數、資料遺失率、相異值個數、最常出現的值、最常出現的值之個數、最常出現的值之百分比、第二常出現的值、第二常出現的值之個數、第二常出現的值之百分比。

- Training data and testing data：preprocessed_iris.csv 為 iris.csv 經前處理過資料，可做為 training data 與 testing data

(2) Google play store

- 原始資料：googleplaystore.csv
- 程式：googleplaystore.py
- 執行程式所產生的檔案：googleplaystore_data_quality_report.csv，preprocessed_googleplaystore.csv
- Basic statistic visualization：

googleplaystore_data_quality_report.csv 資料如下：

1	Continuous Features										
2	Feature	Count	Miss%	Card	Min	1st Qrt	Mean	Median	3rd Qrt	Max	Std Dev
3	Rating	10841	13.6	39	1	4	4.19	4.3	4.5	5	0.52
4	Reviews	10841	0	6001	0	38	444111.9	2094	54768	78158306	2927494
5	Size	10841	0	477	0.01M	4.9M	21.51M	13.0M	30.0M	100.0M	22.59M
6	Installs	10841	0	21	0.0+	1000.0+	15462912.	100000.0+	5000000.0	10000000(	85021647.42+
7	Price	10841	0	92	$0.00	$0.00	$1.03	$0.00	$0.00	$400.00	$15.95
8											
9	Categorical Features										
10	Feature	Count	Miss%	Card	Mode	Mode Freq	Mode%	2nd Mode	2nd Mode	2nd Mode%	
11	App name	10841	0	9660	ROBLOX	9	0.08	CBS Sports	8	0.07	
12	Category	10841	0.01	33	FAMILY	1972	18.19	GAME	1144	10.55	
13	Type	10841	0.01	2	Free	10040	92.62	Paid	800	7.38	
14	Content Ra	10841	0	6	Everyone	8715	80.39	Teen	1208	11.14	
15	Genres	10841	0.01	119	Tools	842	7.77	Entertainm	623	5.75	
16	Last Updat	10841	0	1377	3-Aug-18	326	3.01	2-Aug-18	304	2.8	
17	Current Ve	10841	0.07	2831	Varies with	1459	13.47	1	809	7.47	
18	Android Ve	10841	0.02	33	4.1 and up	2451	22.61	4.0.3 and u	1501	13.85	

- Data：preprocessed_googleplaystore.csv 為 googleplaystore.csv 經第一步前處理過的 data

(3) Google play store user reviews

- 原始資料：googleplaystore_user_reviews.csv
- 程式：googleplaystore_user_reviews.py
- 執行程式所產生的檔案：googleplaystore_user_reviews_data_quality_report.csv，preprocessed_googleplaystore_user_reviews.csv
- Basic statistic visualization：

googleplaystore_user_reviews_data_quality_report.csv 資料如下：

1	Continuous Features									
2	Feature	Count	Miss%	Card	Min	1st Qrt	Mean	Median	3rd Qrt	Max	Std Dev
3	Sentiment Polarity	64295	41.78	6492	-1	0	0.18	0.15	0.4	1	0.35
4	Sentiment Subjectivity	64295	41.78	4694	0	0.36	0.49	0.51	0.65	1	0.26
5											
6	Categorical Features										
7	Feature	Count	Miss%	Card	Mode	Mode Freq	Mode%	2nd Mode	2nd Mode	2nd Mode%	
8	App name	64295	0	1074	CBS Sports	320	0.5	Angry Bird	320	0.5	
9	Translated Review	64295	41.78	27995	Good	247	0.66	Nice	173	0.46	
10	Sentiment	64295	41.78	3	Positive	23998	64.11	Negative	8271	22.1	

- Data：preprocessed_googleplaystore_user_reviews.csv 為 googleplaystore_user_reviews.csv 經第一步前處理過的 data

接著我們針對(2)中的 preprocessed_googleplaystore.csv 和(3)中的 preprocessed_googleplaystore_user_reviews.csv 進行第二步的前處理以產生最後的 training data 與 testing data：

- 原始資料：preprocessed_googleplaystore.csv，preprocessed_googleplaystore_user_reviews.csv

- 程式：Preprocess_csvs.py
- 執行程式所產生的檔案：preprocessed_googleplaystore.csv，preprocessed_googleplaystore_user_reviews.csv（會覆蓋掉以前的）
- Training data and testing data：preprocessed_googleplaystore.csv 及 preprocessed_googleplaystore_user_reviews.csv 經過第二步的前處理，可做為 training data 與 testing data

3. Data preprocessing methods?

檢查所有的 data quality reports，根據理論，因為每個 feature 的 missing rate 都沒有大於 60%，且每個 feature 的 Cardinality 都比 1 大，因此沒有必要把任何的 feature 刪除，且每個 feature 的 missing rate 都小於 50%，故採取 Imputation 的方法，也就是對於每個 instance，若碰到有任何 feature 是 missing 的，continuous feature 用算術平均數（Mean），categorical feature 用最常出現的值（Mode）填補上去，以上為第一步的前處理，主要目的是讓所有的 missing value 消失。至於第二步的前處理，是將非必要的一些符號拿掉以讓資料看起來乾淨一點（例如我們將 preprocessed_googleplaystore.csv 中，Installs 的+拿掉，Price 的$拿掉等），並且把 preprocessed_googleplaystore.csv 的 Installs，Price，Category，Rating 及 Reviews 這些 features 放到 preprocessed_googleplaystore_user_reviews.csv（用 App 來做對應），以利之後的分析。

4. How you generate decision tree and random forest models?

- 原始資料：
  i. preprocessed_iris.csv
  ii. preprocessed_googleplaystore.csv
  iii. preprocessed_googleplaystore_user_reviews.csv
- 程式：
  i. Data_visualization.py 圖像化分析
  ii. iris_dtree.py 分析 iris dataset
  iii. google_dtree_install.py 分析 google play store app 的 installs
  iv. google_dtree_rating.py 分析 google play store app 的 rating
- 執行程式所產生的檔案：
  i. iris_output.png
  ii. google_output[1-4].png

首先對兩個 dataset 做圖像化分析，

(1) Iris dataset

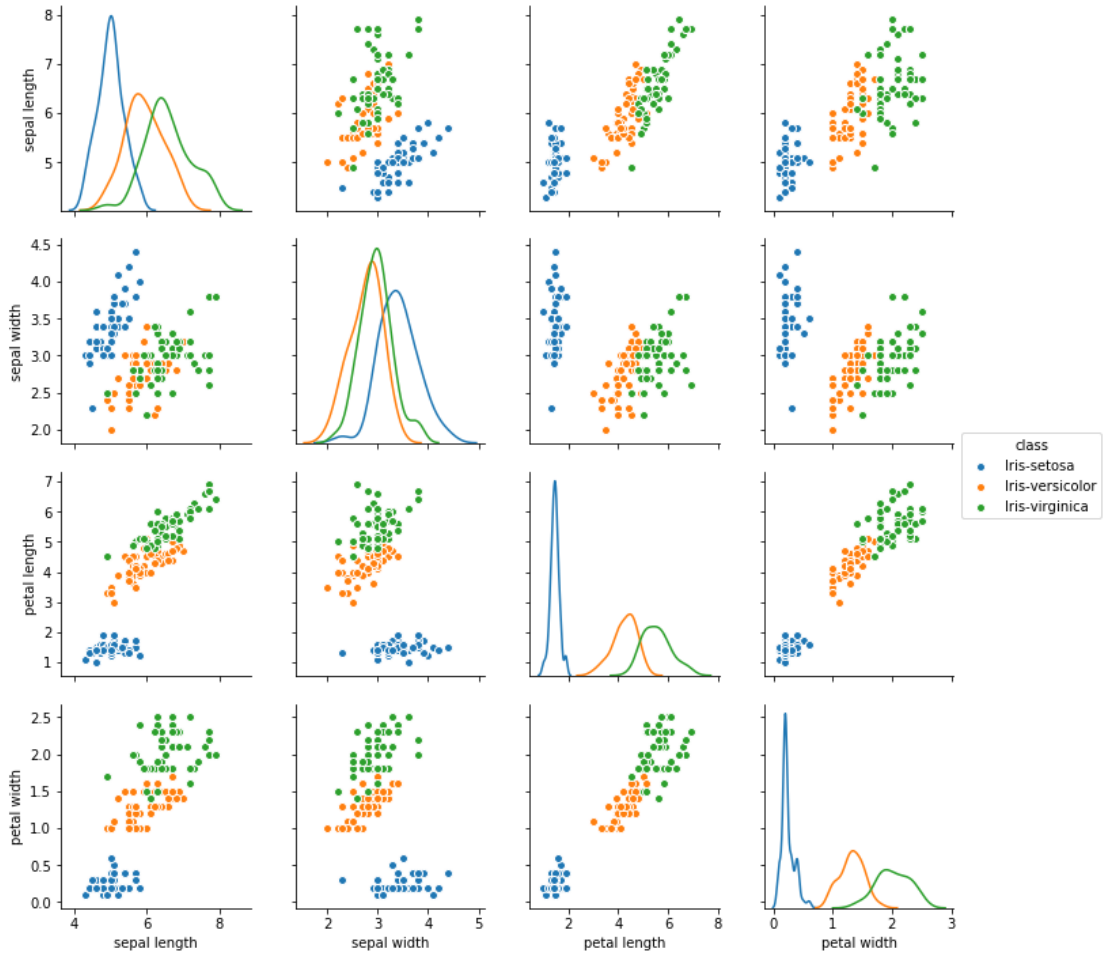


圖 1. 利用 Python Seaborn 產生三種不同 class 在各 feature 下的散布圖

- 利用圖像化分析 Iris dataset，可以發現三筆不同的 class (Iris 的種類) 在各種 feature 下的分布算是相當分散，應能順利分辨三種 class，因此能用所有 feature 來產生 decision tree。
- Decision tree 在這組 data 中使用到的是 Classifier，因為只有三種 class；criterion 設定為 entropy，max_depth 設為 3 避免 overfitting。
- 切割 Training data 和 Testing data 的比例為 7:3，利用 sklearn 中的 train_test_split 做分割。
- 最後以 sklearn.metrics 中的 accuracy_score 簡單評估本 model 的好壞。

(2) Google play store dataset

和 Iris dataset 一樣做圖像化分析處理完畢的 preprocessed_googleplaystore.csv (以下簡稱 googleplaystore.csv)以及 preprocessed_googleplaystore_user_reviews.csv (以下簡稱 googleplaystore_user_review.csv)。

在這裡做兩種預測 **Installs** 以及 **Rating**。

1. Installs 分析

首先對 googleplaystore.csv 用 Seaborn 做圖像化分析，

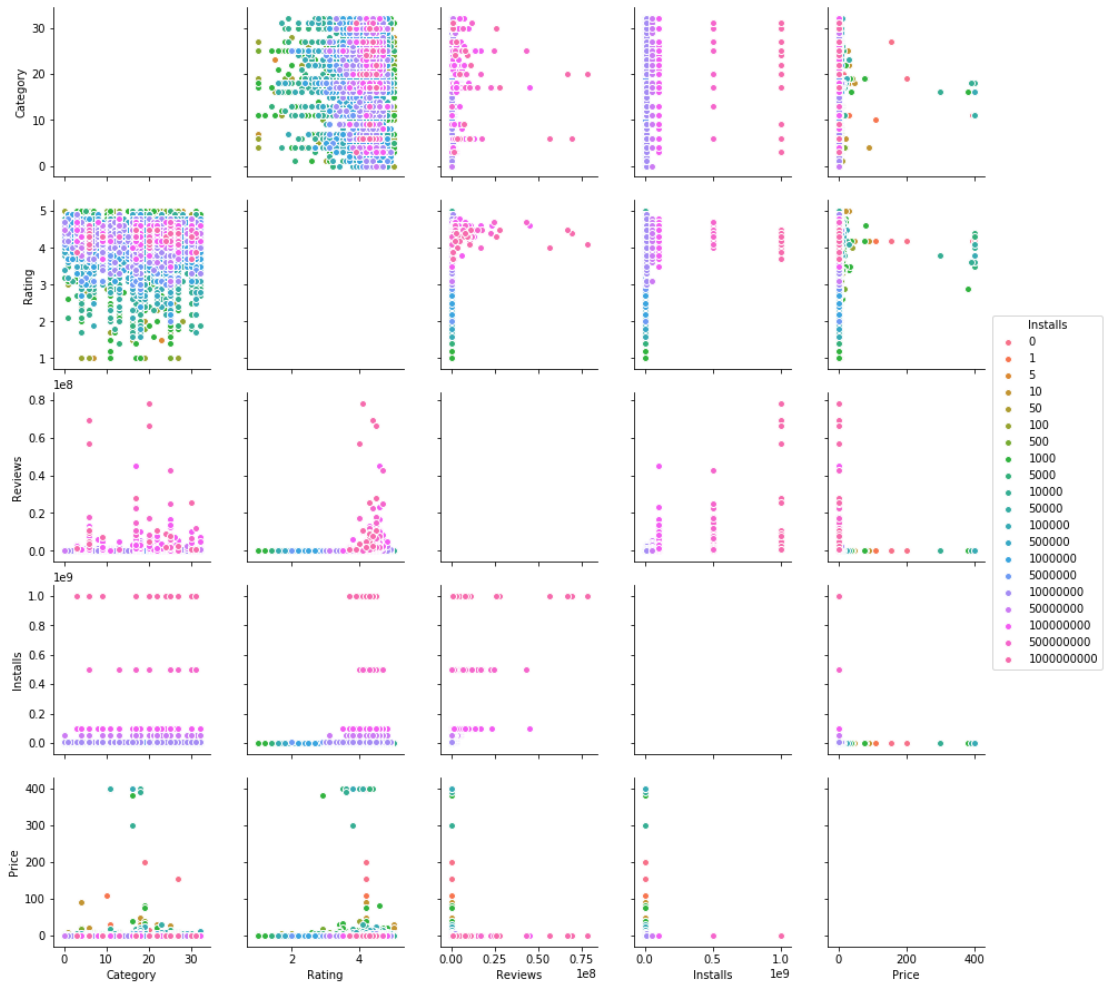


圖 2. googleplaystore.csv 中，各 feature 和 Install 數的關聯散布圖

註: 這裡 feature 只取可數值化的部分，Category 中共有 33 筆不同資料，以 0~32 作編號。

接著我們分析 googleplaystore_user_review.csv，觀察其中 App column，可以發現此 data 可能為 googleplaystore.csv 中的部分 App 的 review 內容以及語意分析。故將 googleplaystore.csv 中 Installs 數、Rating、Category 等資料以 App 為 index 加在 googleplaystore_user_review.csv 各 row 的後面方便作分析。

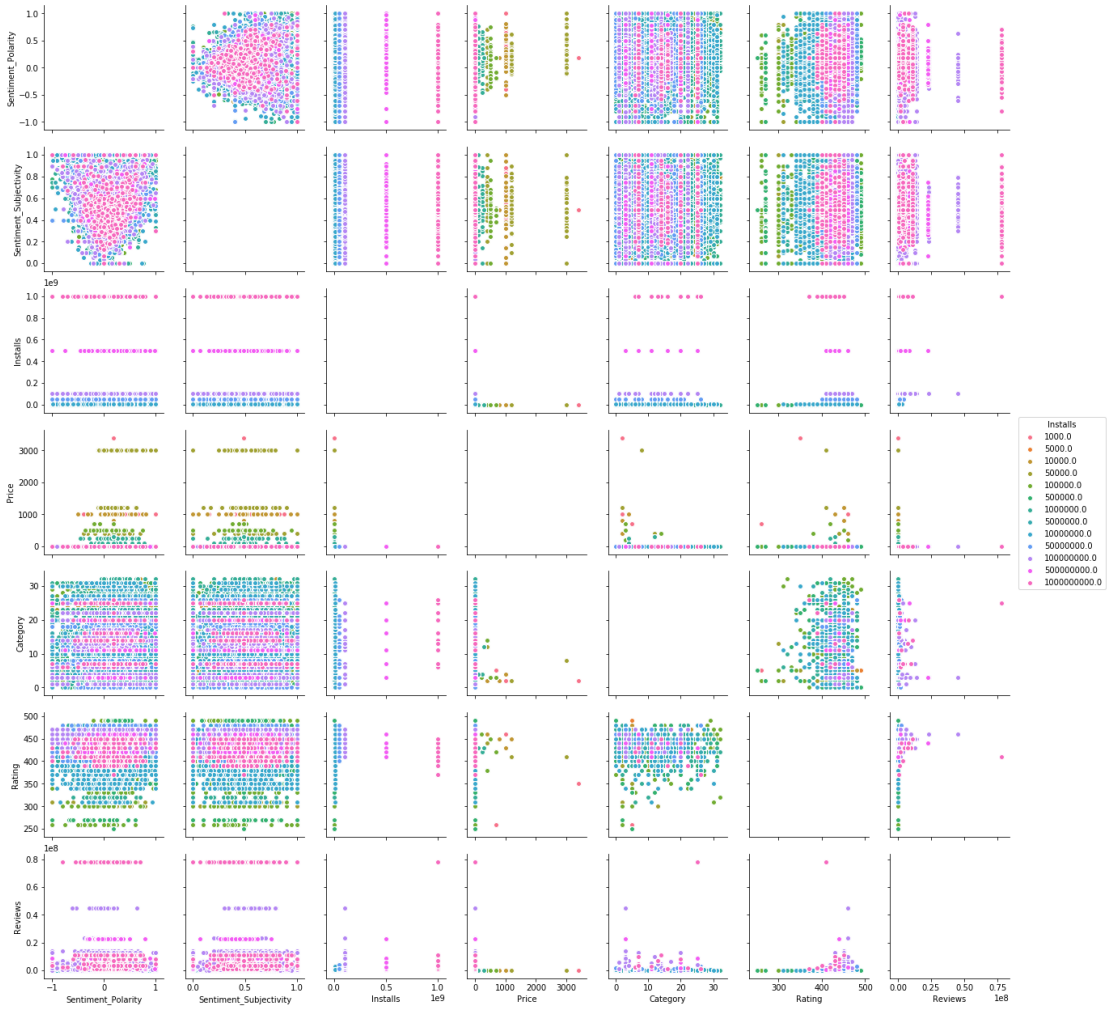


圖 3. googleplaystore_user_review.csv 中，各 feature 和 Install 數的關聯散布圖

註: Rating 經過處理 (*100)

- Installs 數為一離散值 (因原本資料無詳細下載數)，以 Classifier 做 Decision Tree
- 可以看到，高 Installs 數的 data (圖 2.中偏向紫色的 data)，較可能為 Rating 較高 (4~5)、Reviews 較多、Price 為免費 (Price=0)的部分 data，和 Category、Sentiment (Review 語意的正向性)並無明顯關聯。
- 圖 3. 的結果和圖 2. 的預估相同，因資料的重複性可能造成有不太一樣的結果，需兩方比較。
- 因此取 Rating、Reviews、Price 做 Decision Tree model。

2. Rating 分析

對兩個 csv file 做和 Installs 相同的分析步驟

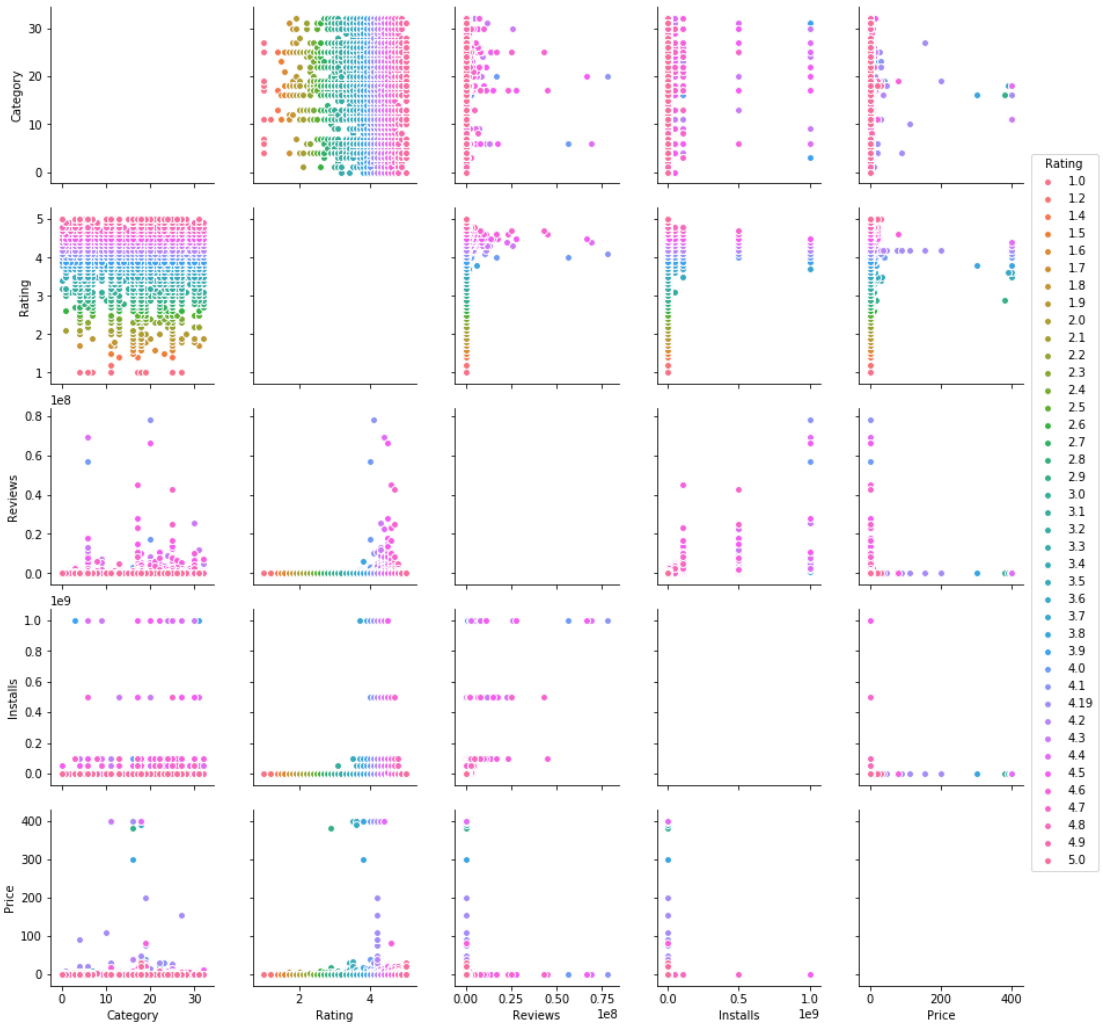


圖 3. googleplaystore.csv 中，各 feature 和 Rating 的關聯散布圖

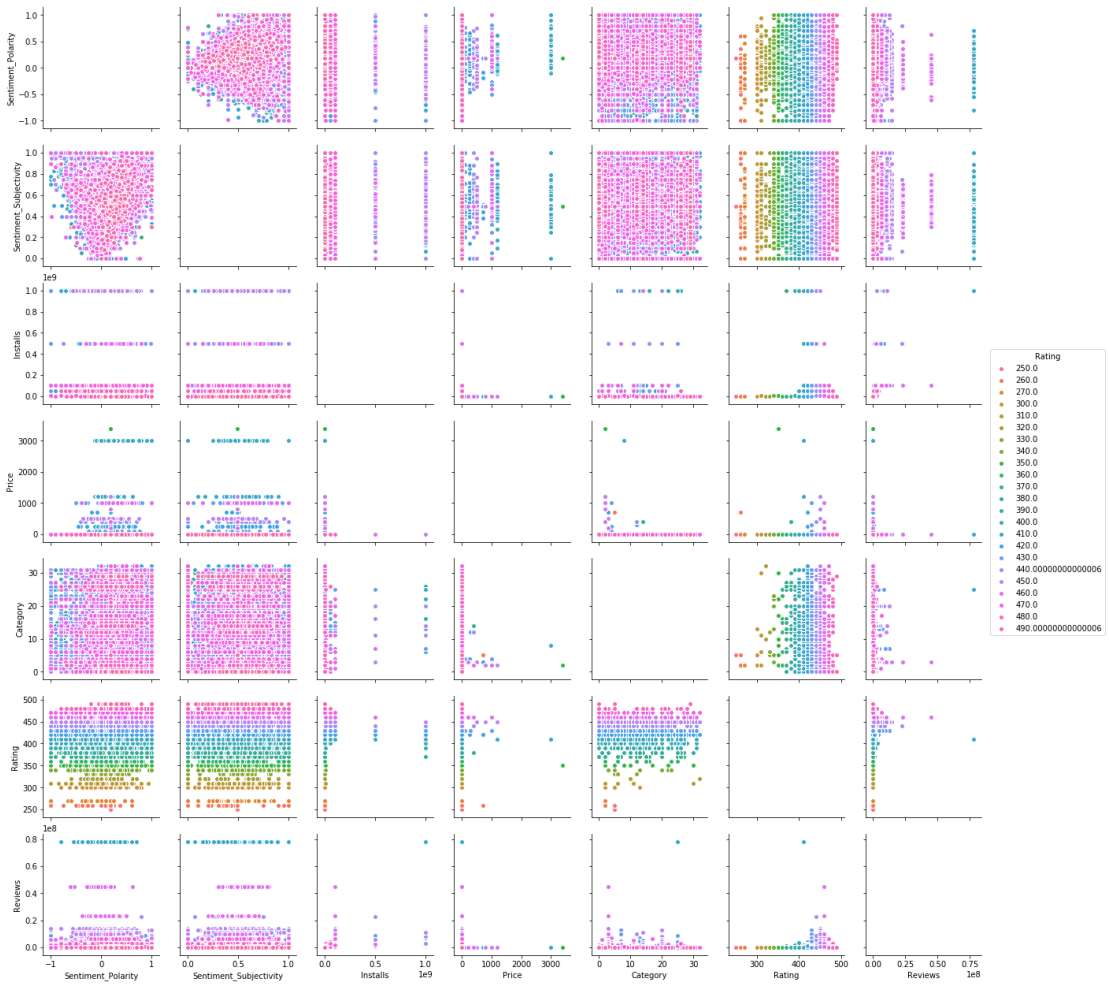


圖 4. googleplaystore_user_review.csv 中，各 feature 和 Rating 的關聯散布圖

- 從圖 3.、圖 4.中，一樣可以看到和 Installs 類似的結果
- 即 Rating 和 Installs、Price、Reviews 的相關性較高
- 故一樣以此三 Features 做 Decision Tree model
- 因 Rating 為一連續的資料，故以 Regressor 建 Decision Tree model
- 最後以 r2 score 簡單評估本 model 的好壞

因為 Installs 和 Rating 都與 Sentiment 無明顯關聯，故以 googleplaystore.csv 的資料建 decision tree model。

5. Generate Random Forest

- 原始資料：
  - i. preprocessed_iris.csv
  - ii. preprocessed_googleplaystore.csv
- 程式：
  - i. Iris_RF.py 建 Iris dataset 的 Random forest

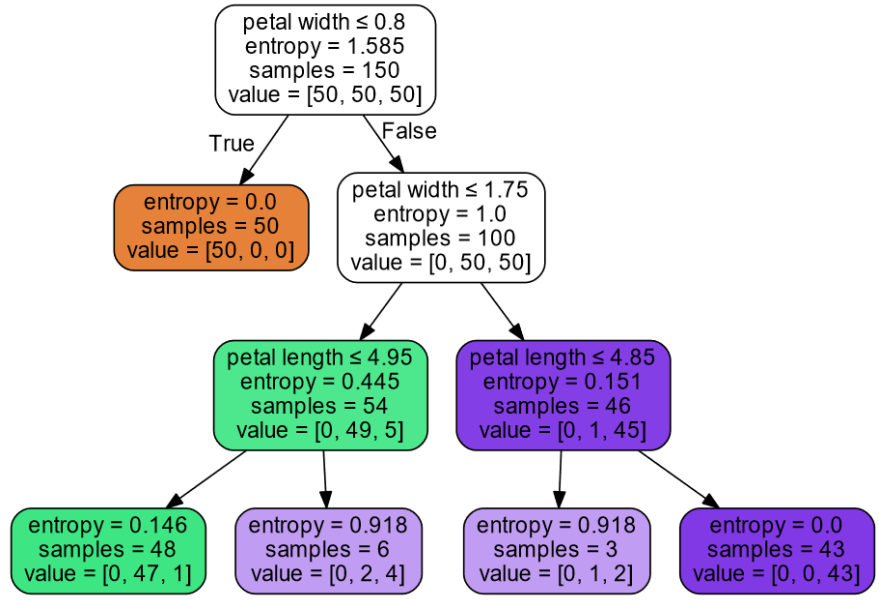
ii. Google_Rating_RF_Install.py 建 Google dataset 的 Install 數 RF model
iii. Google_Rating_RF_Rating.py 建 Google dataset 的 Rating 數 RF model

兩個 Dataset 皆以以下方法建 Random Forest model：

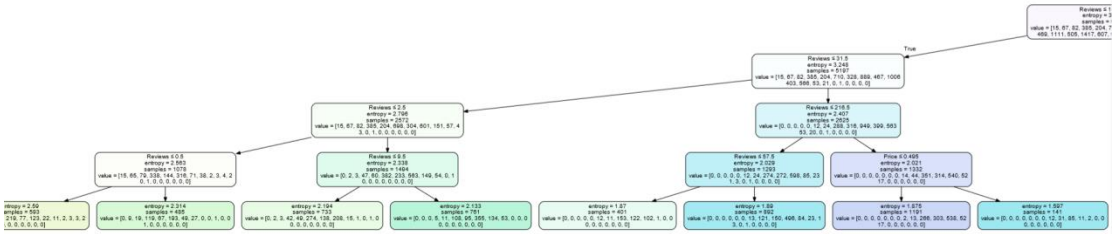
- 在這裡我應用的是 Bootstrapping 做 resample 後產生許多 subsample。
- 此 subsample 中的 data row，因取樣後重新放回後再次取樣，因此可能會有重複 data row 的情況發生，也因此在同一個 training set 下建立了許多不同的 decision tree。
- 用這些 subsample，並從 Features 中隨機取幾個 Feature，搭配產生出許多 Decision Tree，這裡我產生出 100 個隨機樹。
- 對每個 Decision Tree 餵入這 100 個 Testing data，得到各 data row 的預測結果，因此每個 data row 會有 100 個預測結果。
- 接著利用眾數的方式取 100 個結果中出現最多的數字，以此做為該 data row 的最終預測值。即以 100 張票投票的方式產生最終預測結果。
- Price 的部分雖然有 Installs 數高集中在 Price = 0 的趨勢，但推測此 data 中大部分為免費(Price 為 0)的 APP，和 Installs 數較低的免費 App 混雜在一起，因此無法做很好的分割。經測試後發現在 RF model 下移除 Price 這個 feature 會有比較好的效果(accuracy 將近 1.5 倍)，故我們在這裡(RF model)決定不採用 Price 這個 feature。

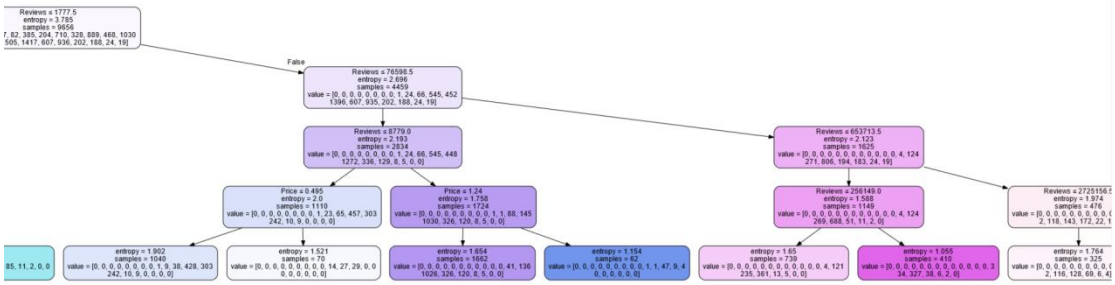
6. The performance?

- Decision Tree:
  - iris
  - 讀入檔案名稱：preprocessed_iris.csv
  - 程式名稱：iris_dtree.py
  - 執行後產生檔案：iris_.pdf

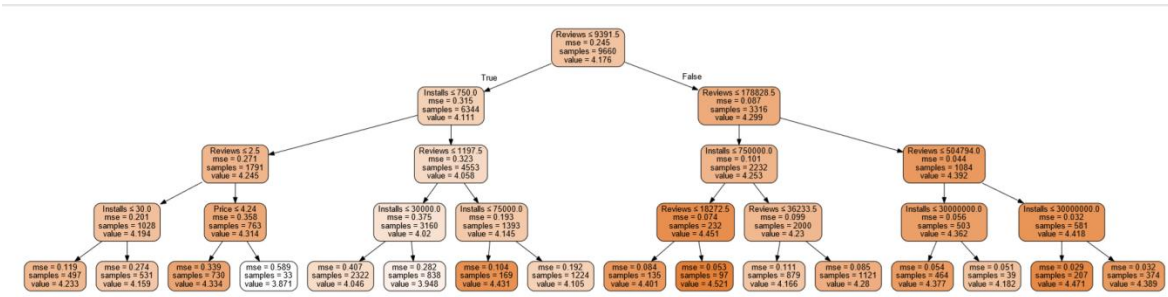


-
- Google play store (Installs)
- 讀入檔案名稱：preprocessed_googleplaystore.csv
- 程式名稱：google_dtree_install.py
- 執行後產生檔案：Google_play_store_(Installs) .pdf





- Google play store (Rating)
- 讀入檔案名稱：preprocessed_googleplaystore.csv
- 程式名稱：google_dtree_rating.py
- 執行後產生檔案：Google_play_store_(Rating).pdf

- 

- Random Forest:
  - 讀入檔案：
    - preprocessed_iris.csv
    - preprocessed_googleplaystore.csv
  - 程式：
    - Iris_RF.py
    - Google_Rating_RF_Install.py
    - Google_Rating_RF_Rating.py
  - Iris, evaluated by accuracy score

```
In [9]: ### Evaluate accuracy of RF ###

        print("Accuracy Evaluate: ", end="")
        print(accuracy_score(y_test, vote_list))

Accuracy Evaluate: 1.0
```

  - Google play store (Installs), evaluated by accuracy score

```
In [9]: ### Evaluate accuracy of RF ###

        print("Evaluate by accuracy score: ", end="")
        print(accuracy_score(y_test, vote_list))

Evaluate by accuracy score: 0.3128023919543354
```

  - Google play store (Rating), evaluated by r2-score
  - 

```
142
143 # In[9]:
144
145
146 ### Evaluate accuracy of RF ###
147
148 print("Evaluate by r2-score: ", end="")
149 print(r2_score(y_test, vote_list))
150
```

```
In [92]: runfile('C:/Users/k8567/Desktop/untitled2.py', wdir='C:/
Users/k8567/Desktop')

In [93]: runfile('C:/Users/k8567/Desktop/program4最新/program4/
Google_Rating_RF_Rating.py', wdir='C:/Users/k8567/Desktop/program4最
新/program4')
Evaluate by r2-score: -0.4128612224616237
```

Decision tree Part

iris_K-fold

validation_dtree_iris_K_fold.py

```
class = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']

[[50  0  0]
 [ 0 47  4]
 [ 0  3 46]]
precision = [1.         0.92156863 0.93877551]
recall = [1.   0.94 0.92]
accuracy = 0.9533333333333334
```

		True Condition			
		Iris-setosa	Iris-versicolor	Iris-virginica	Precision
Predicted Condition	Iris-setosa	50	0	0	1
	Iris-versicolor	0	47	4	0.9215
	Iris-virginica	0	3	46	0.9387
	Recall	1	0.94	0.92	

Accuracy = 0.9533

- iris_resub

validation_dtree_iris_resub.py

```
class = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']

[[50  0  0]
 [ 0 47  1]
 [ 0  3 49]]
precision = [1.         0.97916667 0.94230769]
recall = [1.   0.94 0.98]
accuracy = 0.9733333333333334
```

		True Condition			
		Iris-setosa	Iris-versicolor	Iris-virginica	Precision
Predicted Condition	Iris-setosa	50	0	0	1
	Iris-versicolor	0	47	1	0.9791
	Iris-virginica	0	3	49	0.9423
	Recall	1	0.94	0.98	

Accuracy = 0.9733

- Google_install_K-fold

validation_dtree_google_install_K_fold.py

```
class = ['<=1000', '<=10000', '<=100000', '<=1000000', '<=10000000', 'other']

[[2528  425    8    1    0    0]
 [ 150  848  273   12    1    0]
 [   2  224 1205  545   15    0]
 [   0    2   95 1245  474   16]
 [   0    0    0  119  952  100]
 [   0    0    0    0  102  318]]
precision = [0.85347738 0.66043614 0.60522351 0.67958515 0.81298036 0.75714286]
recall = [0.94328358 0.56571047 0.76217584 0.64776275 0.61658031 0.73271889]
accuracy = 0.734575569358178
```

-

		True Condition						
		<=1000	<=10000	<=100000	<=1000000	<=10000000	>10000000	Precision
Predicted Condition	<=1000	2528	428	8	1	0	0	0.8534
	<=10000	150	848	273	12	1	0	0.6604
	<=100000	2	224	1205	545	15	0	0.6052
	<=1000000	0	2	95	1245	474	16	0.6795
	<=10000000	0	0	0	119	952	100	0.8129
	>10000000	0	0	0	0	102	318	0.7571
	Recall	0.9432	0.5657	0.7621	0.6477	0.6165	0.7327	

Accuracy = 0.7345

- ## Google_install_ resub

- validation_dtree_google_install_resub.py

```
class = ['<=1000', '<=10000', '<=100000', '<=1000000', '<=10000000', 'other']

[[2532  432    8    1    0    0]
 [ 146  763  120    4    1    0]
 [   2  304 1412  627   19    0]
 [   0    0   41 1162  446   13]
 [   0    0    0  128  958   64]
 [   0    0    0    0  120  357]]
precision = [0.85166498 0.73791103 0.59729272 0.69915764 0.83304348 0.74842767]
recall = [0.94477612 0.509006   0.89310563 0.60457856 0.62046632 0.82258065]
accuracy = 0.7436853002070394
```

-

		True Condition						
		<=1000	<=10000	<=100000	<=1000000	<=10000000	>10000000	Precision
Predicted Condition	<=1000	2532	432	8	1	0	0	0.8516
	<=10000	146	763	120	4	1	0	0.7379
	<=100000	2	304	1412	627	19	0	0.5972
	<=1000000	0	0	41	1162	446	13	0.6991
	<=10000000	0	0	0	128	958	64	0.8330
	>10000000	0	0	0	0	120	357	0.7484
	Recall	0.9447	0.5090	0.8931	0.6045	0.6204	0.8225	

Accuracy = 0.7436

- ## rating_K-fold

validation_dtree_google_rating_K_fold.py

```
class = ['<=3', '<=3.5', '<=4', '<=4.5', '<=5']

[[   0    0    0    0    0]
 [   0    0    0    0    0]
 [  73  101  275  494  158]
 [ 288  413 1268 5016 1513]
 [   0    0    6   24   31]]
C:\Python\P5\validation_dtree_google_rating_K_fold.py:85: RuntimeWarni
  precision[i] = confusion_matrix_rating[i][i] / sum(confusion_matrix_
precision = [       nan        nan 0.24977293 0.59025653 0.50819672]
recall = [0.         0.         0.17753389 0.90639682 0.01821387]
accuracy = 0.5509316770186335
```

		True Condition					
		<=3	<=3.5	<=4	<=4.5	<=5	Precision
Predicted Condition	<=3	0	0	0	0	0	nan
	<=3.5	0	0	0	0	0	nan
	<=4	73	101	275	494	158	0.2497
	<=4.5	288	413	1268	5016	1513	0.5902
	<=5	0	0	6	24	31	0.5081
	Recall	0	0	0.1775	0.9063	0.018	

Accuracy = 0.5509

- rating_ resub

validation_dtree_google_rating_resub.py

```
class = ['<=3', '<=3.5', '<=4', '<=4.5', '<=5']

[[   0    0    0    0    0]
 [   0    0    0    0    0]
 [  66   99  248  388   70]
 [ 295  415 1295 5105 1582]
 [   0    0    6   41   50]]
C:\Python\P5\validation_dtree_google_rating_resub.py:82: RuntimeWarni
  precision[i] = confusion_matrix_rating[i][i] / sum(confusion_matrix
precision = [       nan        nan 0.2847302  0.58732168 0.51546392]
recall = [0.         0.         0.16010329 0.92247922 0.0293772 ]
accuracy = 0.5593167701863354
```

		True Condition					
		<=3	<=3.5	<=4	<=4.5	<=5	Precision
Predicted Condition	<=3	0	0	0	0	0	nan
	<=3.5	0	0	0	0	0	nan
	<=4	66	99	248	388	70	0.2847
	<=4.5	295	415	1295	5105	1582	0.5873
	<=5	0	0	6	41	50	0.5154
	Recall	0	0	0.1601	0.9224	0.0293	

Accuracy = 0.5593

- Random forest Part

- iris_K-fold

validation_RF_iris_K_fold.py

```
class = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']

confusionm matrix
[[50.  0.  0.]
 [ 0. 45.  2.]
 [ 0.  5. 48.]]
precision = [1.         0.95744681 0.90566038]
recall = [1.   0.9  0.96]
accuracy = 0.9533333333333334
```

		True Condition			
		Iris-setosa	Iris-versicolor	Iris-virginica	Precision
Predicted Condition	Iris-setosa	50	0	0	1
	Iris-versicolor	0	45	2	0.9574
	Iris-virginica	0	5	48	0.9056
	Recall	1	0.9	0.96	

Accuracy = 0.9533

- iris_resub

validation_RF_iris_resub.py

```
class = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']

[[50  0  0]
 [ 0 47  2]
 [ 0  3 48]]
precision = [1.         0.95918367 0.94117647]
recall = [1.   0.94 0.96]
accuracy = 0.9666666666666667
```

		True Condition			
		Iris-setosa	Iris-versicolor	Iris-virginica	Precision
Predicted Condition	Iris-setosa	50	0	0	1
	Iris-versicolor	0	47	2	0.9591
	Iris-virginica	0	3	48	0.9411
	Recall	1	0.94	0.96	

Accuracy = 0.9666

- Google_install_K-fold

validation_RF_google_install_K_fold.py

```
class = ['<=1000', '<=10000', '<=100000', '<=1000000', '<=10000000', 'other']

[[2323  349   55   31   12    0]
 [  97  512  227   33    7    0]
 [   8  144  499  194   17    0]
 [ 239  468  782 1520  868  205]
 [  13   26   18  140  586   79]
 [   0    0    0    4   54  150]]
precision = [0.83862816 0.58447489 0.57888631 0.37236649 0.67981439 0.72115385]
recall = [0.86679104 0.34156104 0.31562302 0.79084287 0.37953368 0.34562212]
accuracy = 0.5786749482401656
```

-

		True Condition						
		<=1000	<=10000	<=100000	<=1000000	<=10000000	>10000000	Precision
Predicted Condition	<=1000	2323	349	55	31	12	0	0.8386
	<=10000	97	512	227	33	7	0	0.5844
	<=100000	8	144	499	194	17	0	0.5788
	<=1000000	239	468	782	1520	868	205	0.3723
	<=10000000	13	26	18	140	586	79	0.6798
	>10000000	0	0	0	4	54	150	0.7211
	Recall	0.8667	0.3415	0.3156	0.7908	0.3795	0.3456	

Accuracy = 0.5786

- Google_install_ resub

validation_RF_google_install_resub.py

```
class = ['<=1000', '<=10000', '<=100000', '<=1000000', '<=10000000', 'other']

[[2099  239   90   52   25    2]
 [  63  375  130   64   63    7]
 [  16   42  310   59   40    4]
 [ 502  843 1051 1747 1113  371]
 [   0    0    0    0  303    6]
 [   0    0    0    0    0   44]]
precision = [0.83725568 0.53418803 0.6581741  0.31046739 0.98058252 1.        ]
recall = [0.78320896 0.25016678 0.19607843 0.90894901 0.19624352 0.10138249]
accuracy = 0.5049689440993789
```

-

		True Condition						
		<=1000	<=10000	<=100000	<=1000000	<=10000000	>10000000	Precision
Predicted Condition	<=1000	2099	239	90	52	25	2	0.8372
	<=10000	63	375	130	64	63	7	0.5341
	<=100000	16	42	310	59	40	4	0.6581
	<=1000000	502	843	1051	1747	1113	371	0.3104
	<=10000000	0	0	0	0	303	6	0.9805
	>10000000	0	0	0	0	0	44	1
	Recall	0.7832	0.2501	0.1960	0.9089	0.1962	0.1013	

Accuracy = 0.5049

- Google_rating_K-fold

validation_RF_google_rating_K_fold.py

```
 class = ['<=3', '<=3.5', '<=4', '<=4.5', '<=5']

[[   4   12   35   53   14]
 [  21   16   72  181   48]
 [  63  105  324  778  249]
 [ 253  322  932 3878 1204]
 [  20   59  186  644  187]]
precision = [0.03389831 0.04733728 0.21329822 0.58855669 0.17062044]
recall = [0.01108033 0.0311284  0.2091672  0.70075894 0.10987074]
accuracy = 0.45641821946169775
```

		True Condition					
		<=3	<=3.5	<=4	<=4.5	<=5	Precision
Predicted Condition	<=3	4	12	35	53	14	0.0338
	<=3.5	21	16	72	181	48	0.0473
	<=4	63	105	324	778	249	0.2132
	<=4.5	253	322	932	3878	1204	0.5885
	<=5	20	59	186	644	187	0.1706
	Recall	0.0110	0.0311	0.2091	0.7007	0.1098	

Accuracy = 0.4564

- Google_rating_ resub

validation_RF_google_rating_K_fold.py

```
 class = ['<=3', '<=3.5', '<=4', '<=4.5', '<=5']

[[   4   15   25   45   17]
 [  13   17   78  174   50]
 [  51   92  306  777  242]
 [ 271  327  955 3943 1198]
 [  22   63  185  595  195]]
precision = [0.03773585 0.05120482 0.20844687 0.58903496 0.18396226]
recall = [0.01108033 0.03307393 0.1975468  0.71250452 0.11457109]
accuracy = 0.46221532091097306
```

		True Condition					
		<=3	<=3.5	<=4	<=4.5	<=5	Precision
Predicted Condition	<=3	4	15	25	45	17	0.0377
	<=3.5	13	17	78	174	50	0.0512
	<=4	51	92	306	777	242	0.2084
	<=4.5	271	327	955	3943	1198	0.5890
	<=5	22	63	185	595	195	0.1839
	Recall	0.0110	0.0330	0.1975	0.7125	0.1145	

- Accuracy = 0.4622

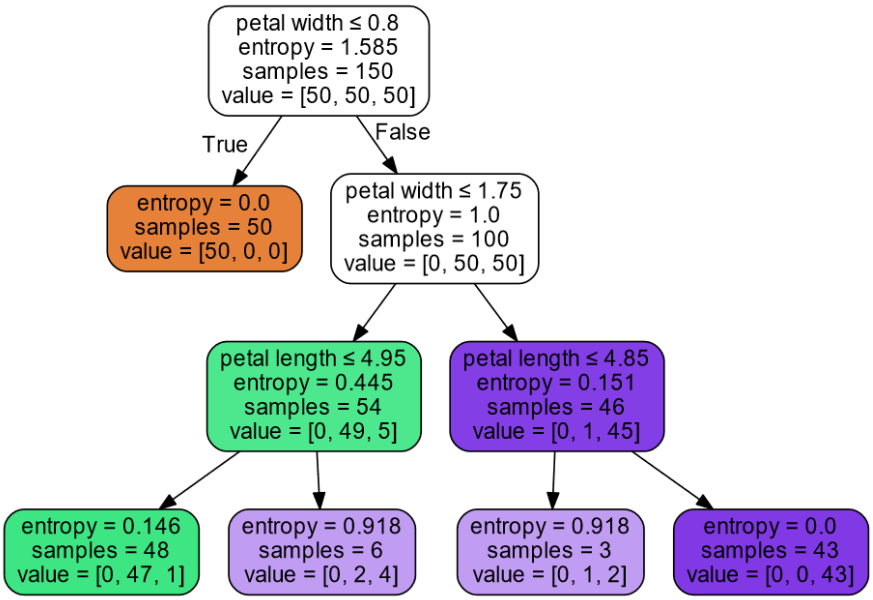
7. Conclusion?

Decision tree iris 的部分，我們藉由 decision tree 可以得知，petal 是決定不同的三種花的主要關鍵，我們判斷可能是因為 petal 的長度及寬度才是決定不同種類 iris 大小的原因，所以同一個品種的花瓣最大的長度以及寬度是類似的，而不同種的則會有些微的差異；至於 decision tree 的 google_play_store 部分，因為有先看過散步圖之後，才決定拿 install rating 做預測目標並進行分析，一開始 depth 設成 3 的時候，會只有 review 決定，而到了 depth=4 的時候才有其他的參數來決定這棵 tree，故我們的 decision tree 不同於 iris 的 depth=3，而選擇了 4。而由上述可知，不管目標是 install 還是 rating，review 都是決定分類的關鍵因素，為此我們推估為：因為 review 的次數越多，代表有安裝這 app 的人數也越多，所以才會有足夠的 review 數量。而 rating 之所以也會有關係，也是因為其實 review 的個數也會影響 rating 的高低，因為畢竟不同人使用相同的 app 感官也不同，但能夠從大家給的 rating 中得知這個 app 的好壞。至於 random forest 與 confusion matrix 的部分，iris 的預測效果非常顯著，正確率非常高，但反觀 google_play_store 的部分，預測的結果卻只有約 5 成，但我們覺得這已經算是非常高的預測準確率了。因為就一般人而言，去做評價的個數越多，但表其中也有可能會有反串的人，或著是刁民故意雞蛋裡挑骨頭，因此我們認為能夠達到五成的命中率是已經非常高的了，因為若是舊表面上來看的話，代表我們已經能夠預測接近 50%的人的思考，而超過五成也代表能預測一般人的想法，所以我們覺得這個也算是成功的例子，畢竟人的想法不可能跟花一樣都是固定的。

附錄：螢幕截圖，因為第一跟第二部分都只會產生.csv 檔，所以無法截圖，因此每個成員的圖都將從第三部分的 dicision tree 開始截

成員一：吳東鴻

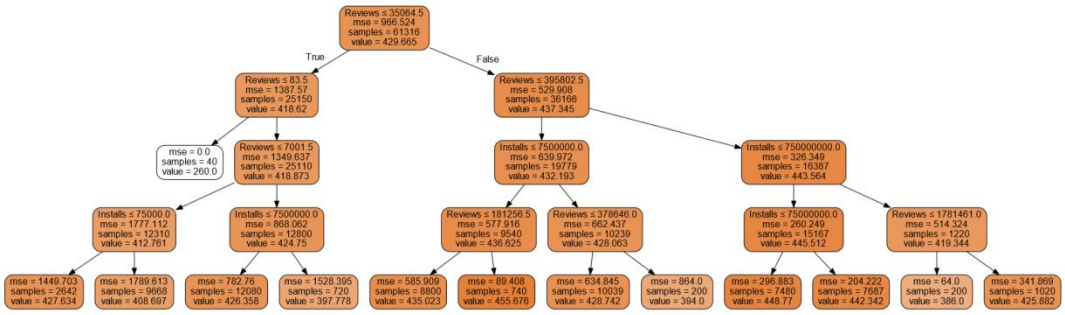
P3

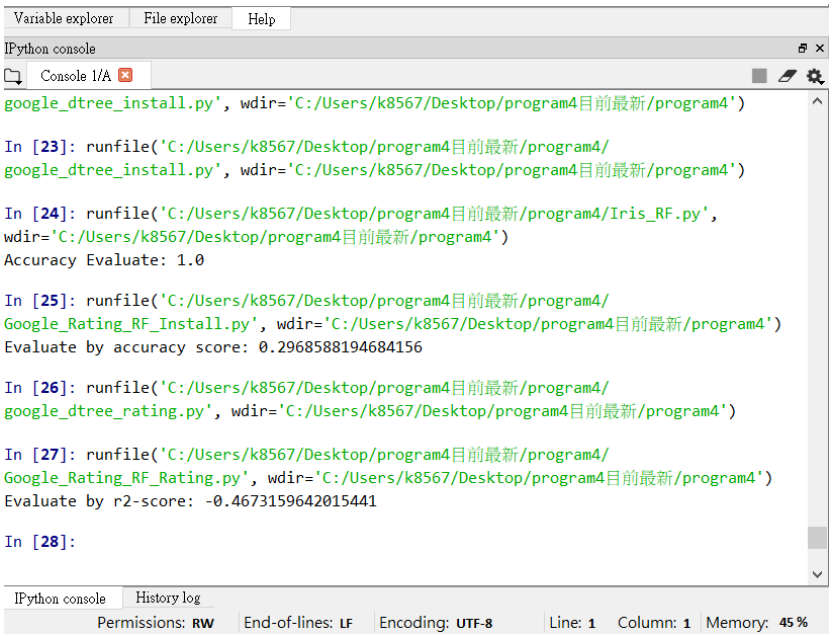


iris：

Google play store (Installs)

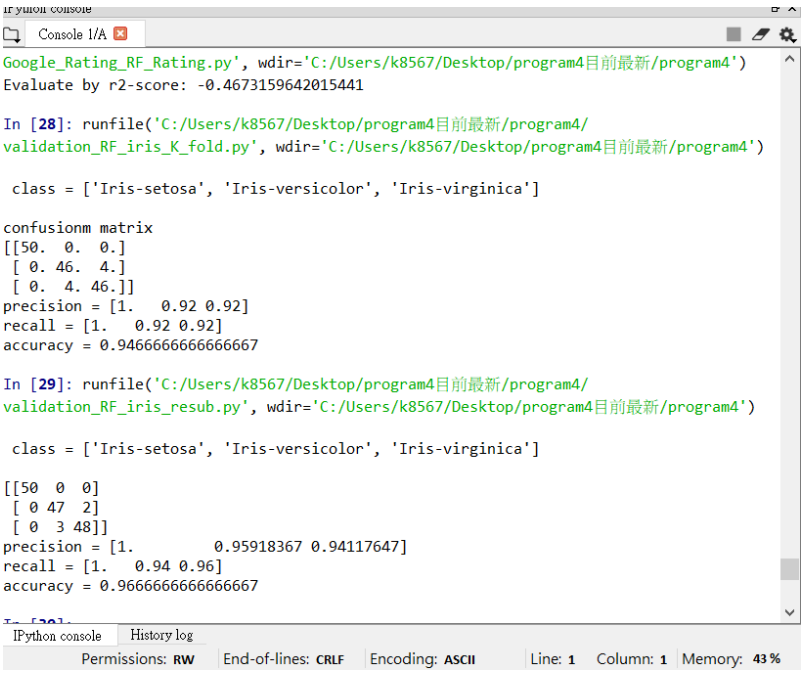
Google play store (rating)





P4

24 25 27 分別代表：iris、Google play store (Installs)、Google play store (rating)



P5

28 29 代表 random forest 的 iris_K-fold、iris_resub

```
In [31]: runfile('C:/Users/k8567/Desktop/program4目前最新/program4/
validation_RF_google_install_K_fold.py', wdir='C:/Users/k8567/Desktop/program4目前最新/
program4')

 class = ['<=1000', '<=10000', '<=100000', '<=1000000', '<=10000000', 'other']

[[2242  321   69   30   19    2]
 [  97  461  215   38   16    2]
 [  15  125  417  195   27    0]
 [ 322  575  865 1554  966  256]
 [   4   16   13  102  475   60]
 [   0    0    1    3   40  113]]
precision = [0.83563176 0.55609168 0.53530167 0.3424416  0.70895522 0.71974522]
recall = [0.83656716 0.30774366 0.26392405 0.80853278 0.30784187 0.26096998]
accuracy = 0.5449461474730737

In [32]: runfile('C:/Users/k8567/Desktop/program4目前最新/program4/
validation_RF_google_install_resub.py', wdir='C:/Users/k8567/Desktop/program4目前最新/
program4')

 class = ['<=1000', '<=10000', '<=100000', '<=1000000', '<=10000000', 'other']

[[2493  348   68   27   13    2]
 [  66  767  141   11    1    0]
 [   3   28  796   49    8    0]
 [ 118  355  575 1835  548  159]
 [   0    0    0    0  972   11]
 [   0    0    0    0    1  261]]
precision = [0.84479837 0.77789047 0.90045249 0.51114206 0.98880977 0.99618321]
recall = [0.93022388 0.51201602 0.50379747 0.95473465 0.62994167 0.60277136]
accuracy = 0.7377796188898095

In [33]:
```

IPython console History log

Permissions: RW End-of-lines: CRLF Encoding: ASCII Line: 1 Column: 1 Memory: 44 %

31 32 代表 random forest 的 Google_install_K-fold、Google_install_resub

```
In [33]: runfile('C:/Users/k8567/Desktop/program4目前最新/program4/
validation_RF_google_rating_K_fold.py', wdir='C:/Users/k8567/Desktop/program4目前最新/
program4')

 class = ['<=3', '<=3.5', '<=4', '<=4.5', '<=5']

[[   6   10   24   54   19]
 [  11   14   76  154   51]
 [  63  104  310  775  258]
 [ 262  328  957 3925 1176]
 [  19   58  182  623  197]]
precision = [0.05309735 0.04575163 0.20529801 0.59040313 0.18257646]
recall = [0.0166205  0.02723735 0.20012912 0.70963659 0.11581423]
accuracy = 0.4610604805302403

In [34]: runfile('C:/Users/k8567/Desktop/program4目前最新/program4/
validation_RF_google_rating_resub.py', wdir='C:/Users/k8567/Desktop/program4目前最新/
program4')

 class = ['<=3', '<=3.5', '<=4', '<=4.5', '<=5']

[[  88    5    9   23    7]
 [   9  226   19   31    8]
 [  20   58  946  126   33]
 [ 237  217  548 5308  748]
 [   7    8   27   43  905]]
precision = [0.66666667 0.77133106 0.79966188 0.75205441 0.91414141]
recall = [0.24376731 0.43968872 0.61071659 0.95968179 0.53203998]
accuracy = 0.7739229494614748

In [35]:
```

IPython console History log

Permissions: RW End-of-lines: CRLF Encoding: ASCII Line: 1 Column: 1 Memory: 45 %

33 34 代表 random forest 的 Google_rating_K-fold Google_rating_resub

```
In [35]: runfile('C:/Users/k8567/Desktop/program4目前最新/program4/
validation_dtree_iris_K_fold.py', wdir='C:/Users/k8567/Desktop/program4目前最新/
program4')

 class = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']

[[50  0  0]
 [ 0 45  3]
 [ 0  5 47]]
precision = [1.         0.9375     0.90384615]
recall = [1.   0.9  0.94]
accuracy = 0.9466666666666667

In [36]: runfile('C:/Users/k8567/Desktop/program4目前最新/program4/
validation_dtree_iris_resub.py', wdir='C:/Users/k8567/Desktop/program4目前最新/
program4')

 class = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']

[[50  0  0]
 [ 0 47  1]
 [ 0  3 49]]
precision = [1.         0.97916667 0.94230769]
recall = [1.   0.94 0.98]
accuracy = 0.9733333333333334

In [37]:
```

IPython console | History log

Permissions: RW | End-of-lines: CRLF | Encoding: UTF-8 | Line: 1 | Column: 1 | Memory: 44 %

35 36 代表 decision tree 的 iris_K-fold、iris_resub

```
In [37]: runfile('C:/Users/k8567/Desktop/program4目前最新/program4/
validation_dtree_google_install_K_fold.py', wdir='C:/Users/k8567/Desktop/program4目前最
新/program4')

 class = ['<=1000', '<=10000', '<=100000', '<=1000000', '<=10000000', 'other']

[[2532  433    9    1    0    0]
 [ 146  878  395   16    1    0]
 [   2  187 1079  505   16    0]
 [   0    0   97 1283  486   16]
 [   0    0    0  117  974  140]
 [   0    0    0    0   66  277]]
precision = [0.85109244 0.61142061 0.60313024 0.68172157 0.79122665 0.80758017]
recall = [0.94477612 0.58611482 0.68291139 0.66753382 0.63123785 0.63972286]
accuracy = 0.7273198011599006

In [38]: runfile('C:/Users/k8567/Desktop/program4目前最新/program4/
validation_dtree_google_install_resub.py', wdir='C:/Users/k8567/Desktop/program4目前最
新/program4')

 class = ['<=1000', '<=10000', '<=100000', '<=1000000', '<=10000000', 'other']

[[2532  432    8    1    0    0]
 [ 146  762  120    4    1    0]
 [   2  304 1411  627   19    0]
 [   0    0   41 1162  446   13]
 [   0    0    0  128  957   64]
 [   0    0    0    0  120  356]]
precision = [0.85166498 0.73765731 0.5971223  0.69915764 0.83289817 0.74789916]
recall = [0.94477612 0.50867824 0.89303797 0.60457856 0.62022035 0.8221709 ]
accuracy = 0.7435791217895609

In [39]:
```

IPython console | History log

Permissions: RW | End-of-lines: CRLF | Encoding: UTF-8 | Line: 1 | Column: 1 | Memory: 45 %

37 38 代表 decision tree 的 Google_install_K-fold、Google_install_resub

```
validation_dtree_google_rating_K_fold.py:85: RuntimeWarning: invalid value encountered
in longlong_scalars
  precision[i] = confusion_matrix_rating[i][i] / sum(confusion_matrix_rating[i])

In [40]: runfile('C:/Users/k8567/Desktop/program4目前最新/program4/
validation_dtree_google_rating_K_fold.py', wdir='C:/Users/k8567/Desktop/program4目前最
新/program4')

 class = ['<=3', '<=3.5', '<=4', '<=4.5', '<=5']

[[   0    0    0    0    0]
 [   0    0    0    0    0]
 [  99  152  373  716  201]
 [ 262  362 1168 4788 1464]
 [   0    0    8   27   36]]
precision = [       nan        nan 0.24205062 0.59522626 0.50704225]
recall = [0.         0.         0.24080052 0.86566624 0.02116402]
accuracy = 0.5382145816072909
C:/Users/k8567/Desktop/program4目前最新/program4/
validation_dtree_google_rating_K_fold.py:85: RuntimeWarning: invalid value encountered
in longlong_scalars
  precision[i] = confusion_matrix_rating[i][i] / sum(confusion_matrix_rating[i])
```

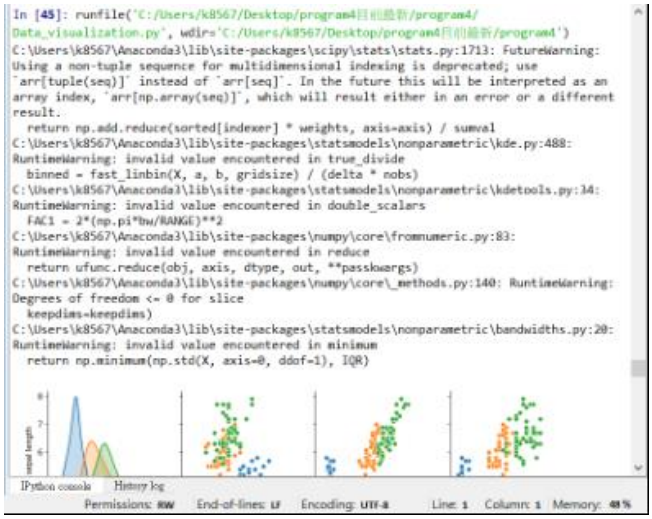
40 代表 decision tree 的 Google_rating_K-fold

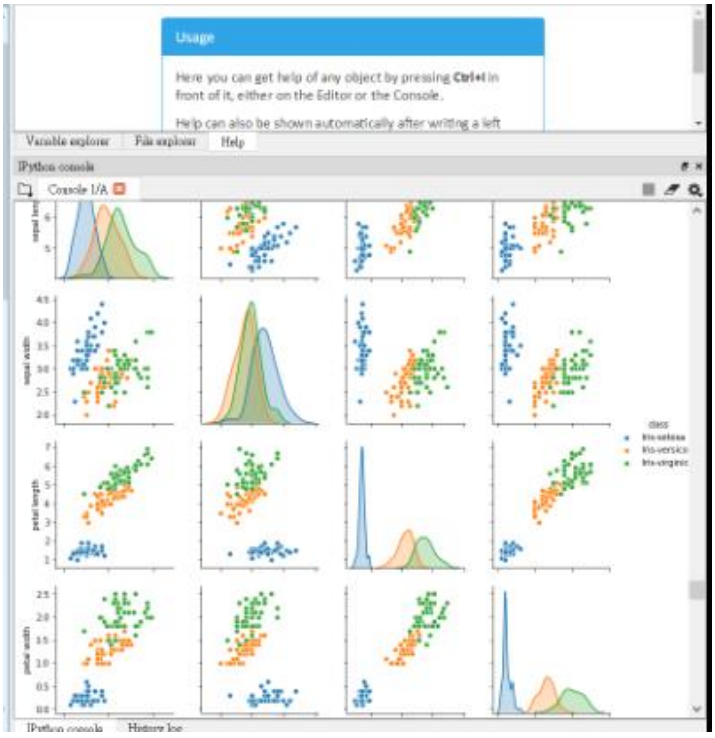
```
In [44]: runfile('C:/Users/k8567/Desktop/program4目前最新/program4/
validation_dtree_google_rating_resub.py', wdir='C:/Users/k8567/Desktop/program4目前最新/
program4')

 class = ['<=3', '<=3.5', '<=4', '<=4.5', '<=5']

[[   0    0    0    0    0]
 [   0    0    0    0    0]
 [   5    5    5   11    7]
 [ 356  509 1544 5520 1694]
 [   0    0    0    0    0]]
precision = [       nan        nan 0.15151515 0.57362569        nan]
recall = [0.         0.         0.00322789 0.99801121 0.        ]
accuracy = 0.5721830985915493
C:/Users/k8567/Desktop/program4目前最新/program4/
validation_dtree_google_rating_resub.py:82: RuntimeWarning: invalid value encountered
in longlong_scalars
  accurate = 0

In [45]:
```

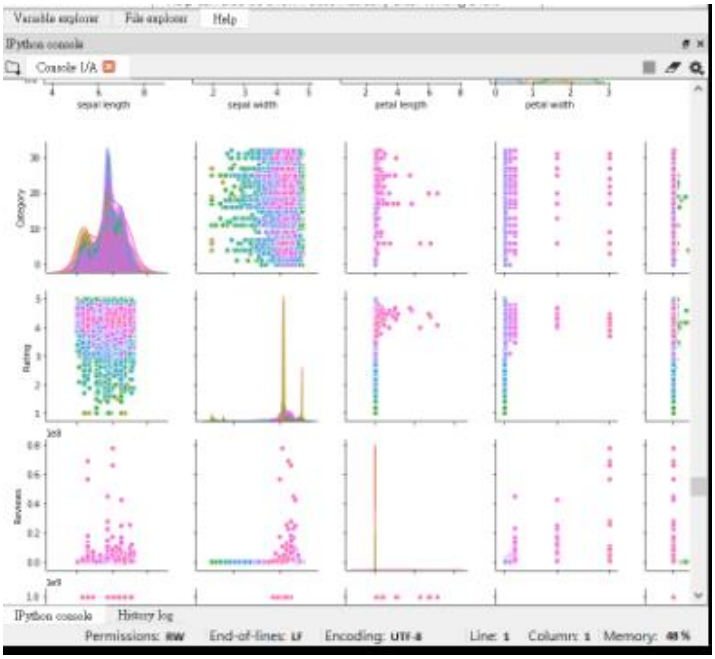
IPython console | History log

Permissions: RW | End-of-lines: CRLF | Encoding: UTF-8 | Line: 91 | Column: 1 | Memory: 44 %

44 代表 decision tree 的 Google_rating_resub



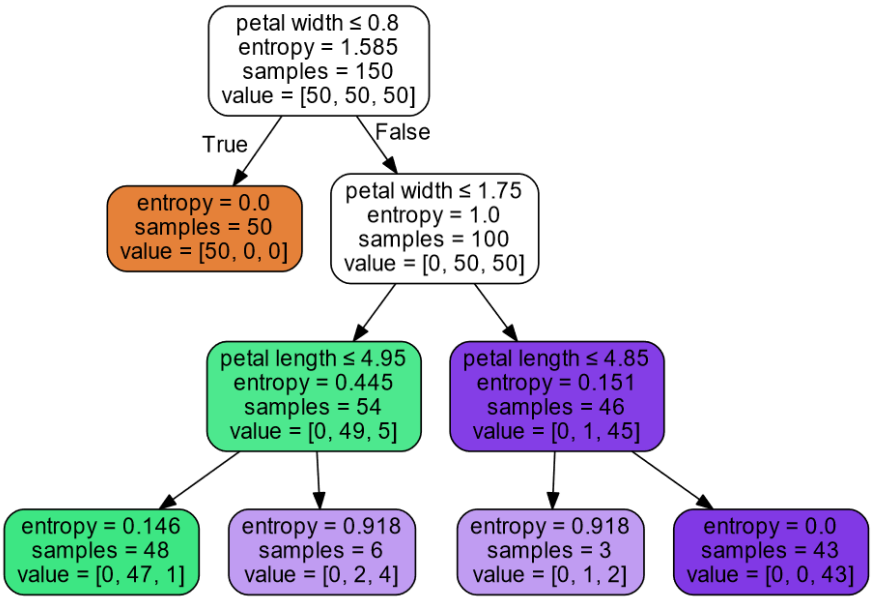




45 為 data visualization 的部分

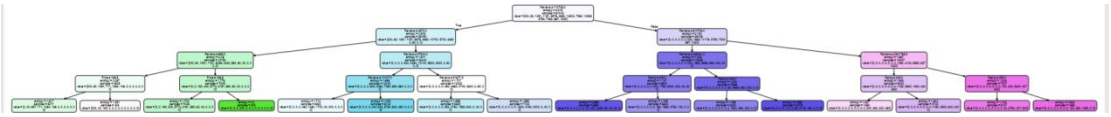
成員二：孫念恩

P3

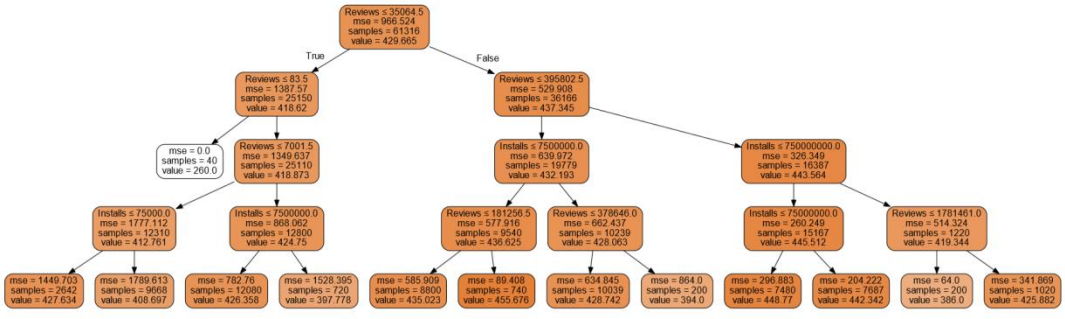


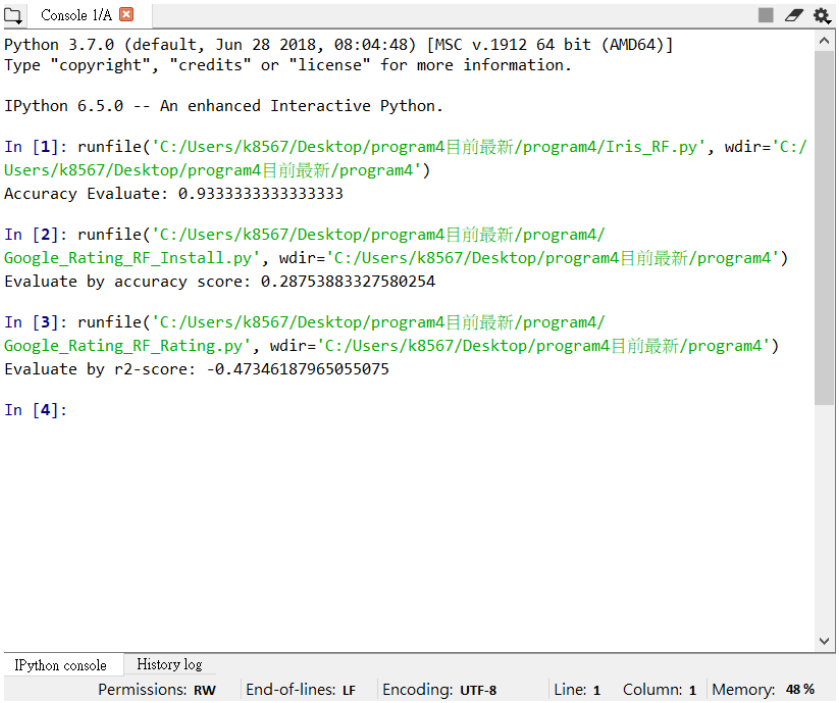
iris：

Google play store (Installs)



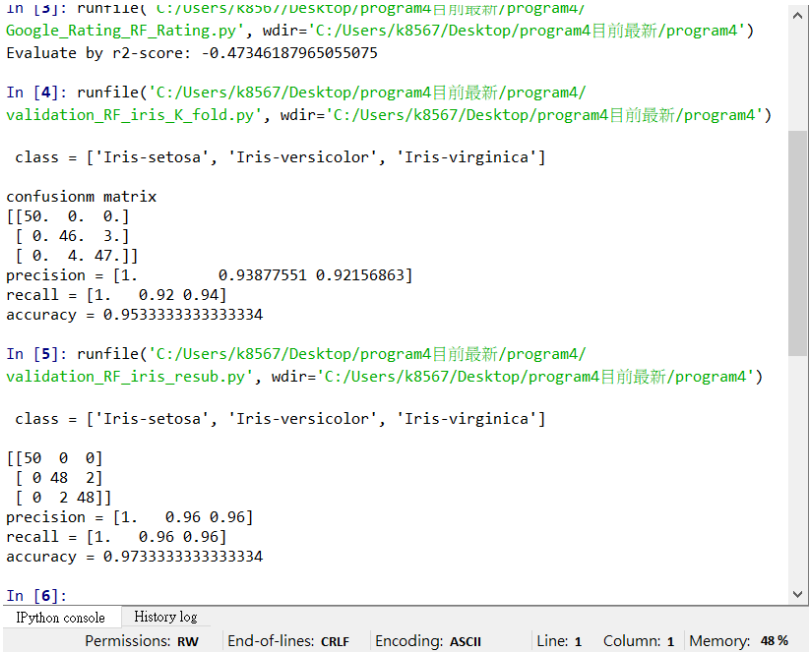
Google play store (rating)





P4

1 2 3 分別代表：iris、Google play store (Installs)、Google play store (rating)



P5

4 5 代表 random forest 的 iris_K-fold、iris_resub

```
In [8]: runfile('C:/Users/k8567/Desktop/program4目前最新/program4/
validation_RF_google_install_K_fold.py', wdir='C:/Users/k8567/Desktop/program4目前最新/
program4')

 class = ['<=1000', '<=10000', '<=100000', '<=1000000', '<=10000000', 'other']

[[2318  345   63   27   12    2]
 [ 100  480  228   34   10    3]
 [   9  132  446  176   19    0]
 [ 250  535  837 1555  922  225]
 [   3    6    6  126  542   68]
 [   0    0    0    4   38  135]]
precision = [0.83773039 0.56140351 0.57033248 0.35962072 0.72170439 0.76271186]
recall = [0.86492537 0.32042724 0.28227848 0.80905307 0.35126377 0.31177829]
accuracy = 0.5671085335542668

In [9]: runfile('C:/Users/k8567/Desktop/program4目前最新/program4/
validation_RF_google_install_resub.py', wdir='C:/Users/k8567/Desktop/program4目前最新/
program4')

 class = ['<=1000', '<=10000', '<=100000', '<=1000000', '<=10000000', 'other']

[[2163  273   91   52   25    2]
 [  61  463  126   33   17    2]
 [   8   28  414   56   29    4]
 [ 448  734  948 1780 1128  329]
 [   0    0    1    1  344   20]
 [   0    0    0    0    0   76]]
precision = [0.83000767 0.65954416 0.76808905 0.33165642 0.93989071 1.        ]
recall = [0.80708955 0.30907877 0.26202532 0.92611863 0.22294232 0.17551963]
accuracy = 0.5426677713338857

In [10]:
```

IPython console | History log

Permissions: RW | End-of-lines: CRLF | Encoding: ASCII | Line: 1 | Column: 1 | Memory: 47 %

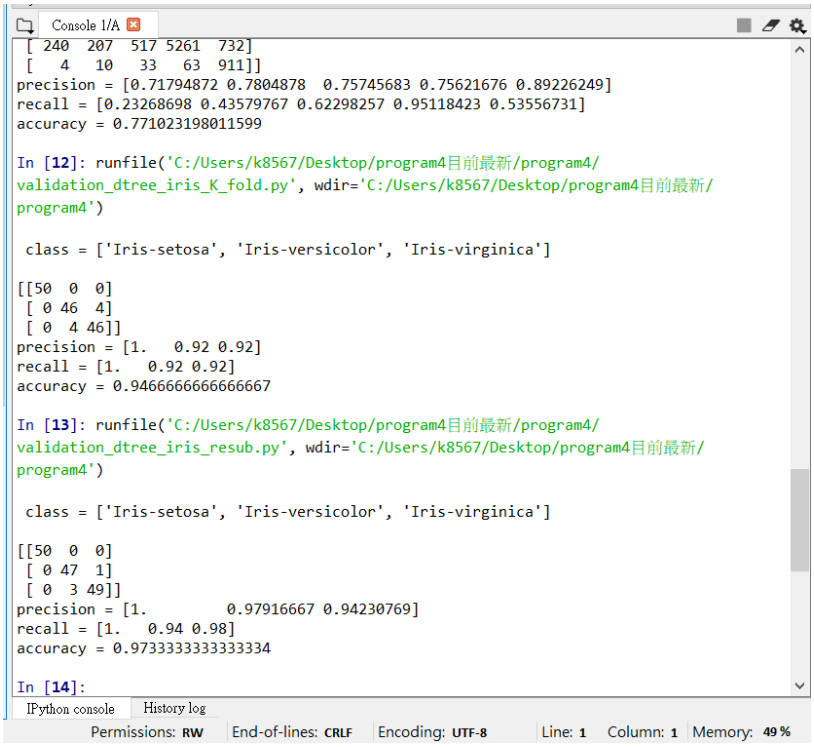
8 9 代表 random forest 的 Google_install_K-fold、Google_install_resub

```
In [10]: runfile('C:/Users/k8567/Desktop/program4目前最新/program4/
validation_RF_google_rating_K_fold.py', wdir='C:/Users/k8567/Desktop/program4目前最新/
program4')

 class = ['<=3', '<=3.5', '<=4', '<=4.5', '<=5']

[[   4   16   23   48   14]
 [  14   20   69  154   51]
 [  54   96  308  774  249]
 [ 275  324  962 3928 1206]
 [  14   58  187  627  181]]
precision = [0.03809524 0.06493506 0.20796759 0.5867065  0.16963449]
recall = [0.01108033 0.03891051 0.19883796 0.71017899 0.106408  ]
accuracy = 0.4599212924606462

In [11]: runfile('C:/Users/k8567/Desktop/program4目前最新/program4/
validation_RF_google_rating_resub.py', wdir='C:/Users/k8567/Desktop/program4目前最新/
program4')

 class = ['<=3', '<=3.5', '<=4', '<=4.5', '<=5']

[[  84    4    7   20    2]
 [   8  224   27   23    5]
 [  25   69  965  164   51]
 [ 240  207  517 5261  732]
 [   4   10   33   63  911]]
precision = [0.71794872 0.7804878  0.75745683 0.75621676 0.89226249]
recall = [0.23268698 0.43579767 0.62298257 0.95118423 0.53556731]
accuracy = 0.771023198011599

In [12]:
```

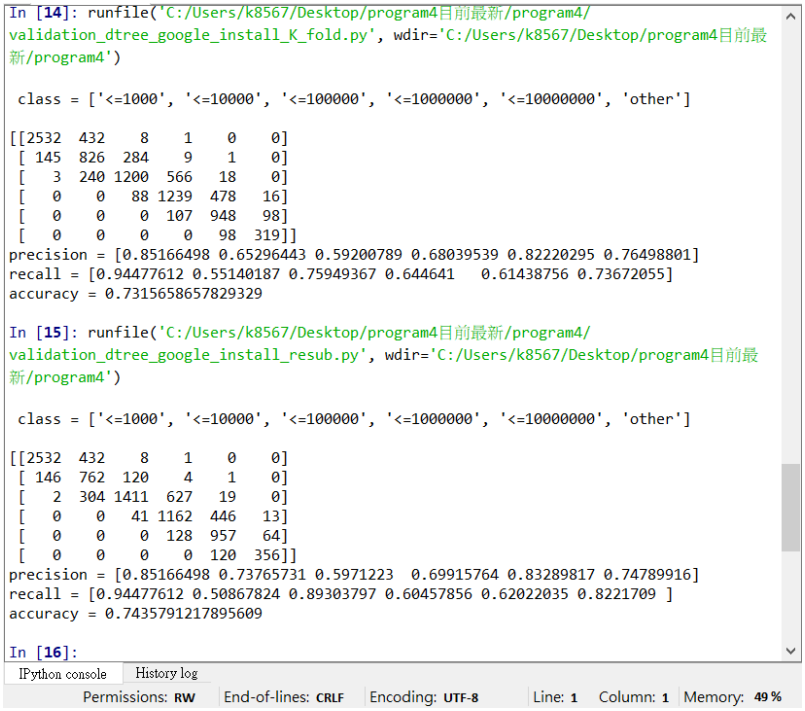
IPython console | History log

Permissions: RW | End-of-lines: CRLF | Encoding: ASCII | Line: 1 | Column: 1 | Memory: 49 %

10 11 代表 random forest 的 Google_rating_K-fold Google_rating_resub



12 13 代表 decision tree 的 iris_K-fold、iris_resub



14 15 代表 decision tree 的 Google_install_K-fold、Google_install_resub

```
In [16]: runfile('C:/Users/k8567/Desktop/program4目前最新/program4/
validation_dtree_google_rating_K_fold.py', wdir='C:/Users/k8567/Desktop/program4目前最
新/program4')

 class = ['<=3', '<=3.5', '<=4', '<=4.5', '<=5']

[[   0    0    0    0    0]
 [   0    0    0    0    0]
 [  63   76  207  403  121]
 [ 298  437 1336 5091 1557]
 [   0    1    6   37   23]]
precision = [       nan        nan 0.23793103 0.58389724 0.34328358]
recall = [0.         0.         0.1336346  0.92044838 0.01352146]
accuracy = 0.551056338028169
C:/Users/k8567/Desktop/program4目前最新/program4/
validation_dtree_google_rating_K_fold.py:85: RuntimeWarning: invalid value encountered
in longlong_scalars
  precision[i] = confusion_matrix_rating[i][i] / sum(confusion_matrix_rating[i])

In [17]: runfile('C:/Users/k8567/Desktop/program4目前最新/program4/
validation_dtree_google_rating_resub.py', wdir='C:/Users/k8567/Desktop/program4目前最新/
program4')

 class = ['<=3', '<=3.5', '<=4', '<=4.5', '<=5']

[[   0    0    0    0    0]
 [   0    0    0    0    0]
```

IPython console | History log

Permissions: RW | End-of-lines: CRLF | Encoding: UTF-8 | Line: 1 | Column: 1 | Memory: 49 %

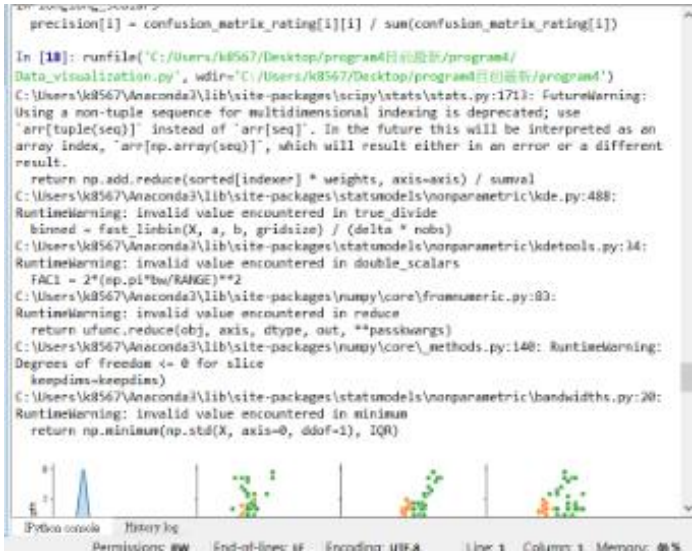
16 代表 decision tree 的 Google_rating_K-fold

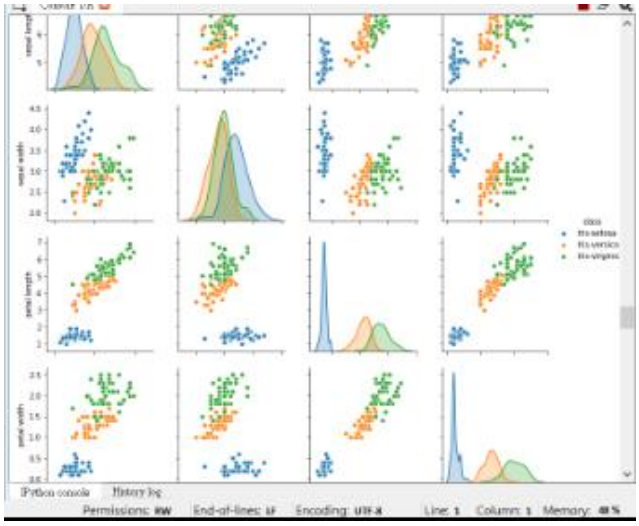
```
In [17]: runfile('C:/Users/k8567/Desktop/program4目前最新/program4/
validation_dtree_google_rating_resub.py', wdir='C:/Users/k8567/Desktop/program4目前最新/
program4')

 class = ['<=3', '<=3.5', '<=4', '<=4.5', '<=5']

[[   0    0    0    0    0]
 [   0    0    0    0    0]
 [   5    5    5   11    7]
 [ 356  509 1544 5520 1694]
 [   0    0    0    0    0]]
precision = [       nan        nan 0.15151515 0.57362569        nan]
recall = [0.         0.         0.00322789 0.99801121 0.        ]
accuracy = 0.5721830985915493
C:/Users/k8567/Desktop/program4目前最新/program4/
validation_dtree_google_rating_resub.py:82: RuntimeWarning: invalid value encountered
in longlong_scalars
  precision[i] = confusion_matrix_rating[i][i] / sum(confusion_matrix_rating[i])

In [18]:
```

IPython console | History log

Permissions: RW | End-of-lines: CRLF | Encoding: UTF-8 | Line: 1 | Column: 1 | Memory: 49 %

17 代表 decision tree 的 Google_rating_resub

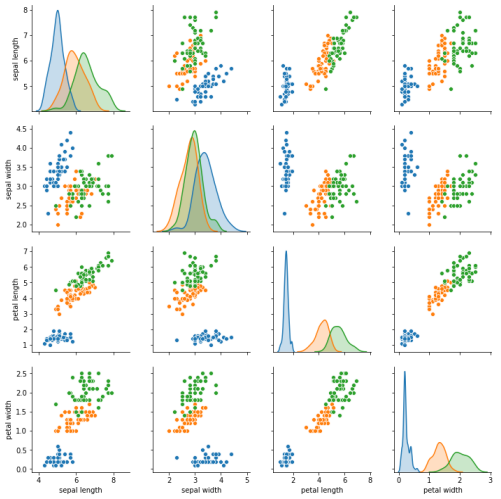


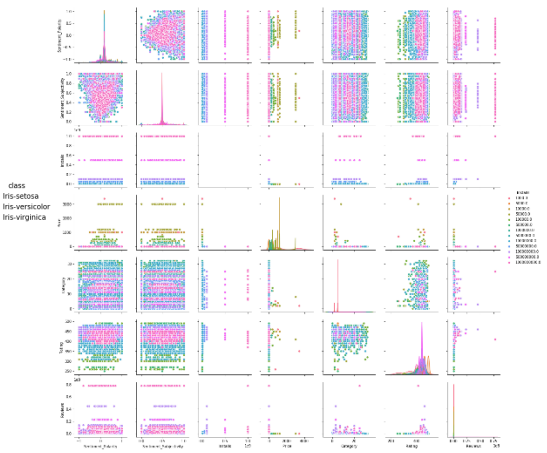


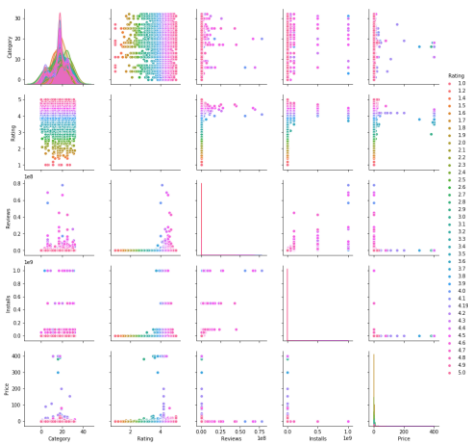
18 為 data visualization 的部分

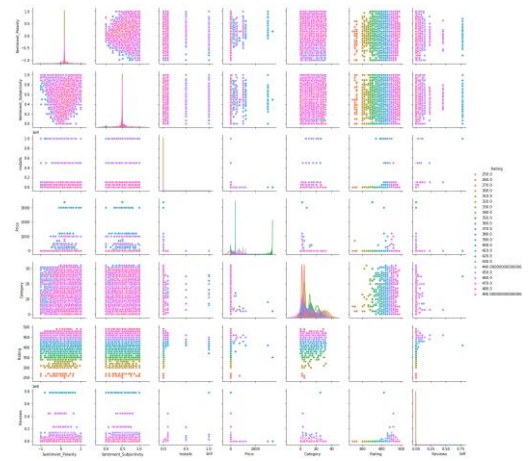
成員三：謝富丞

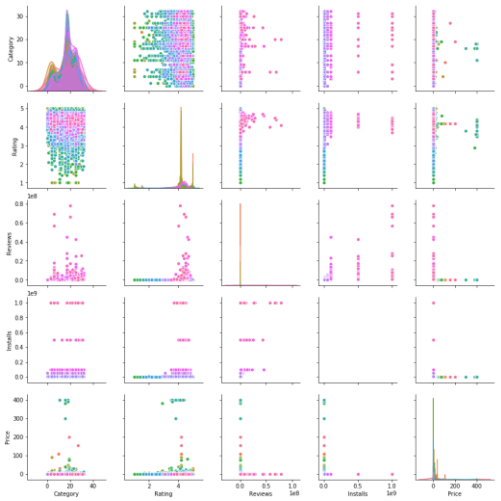
Data visualization



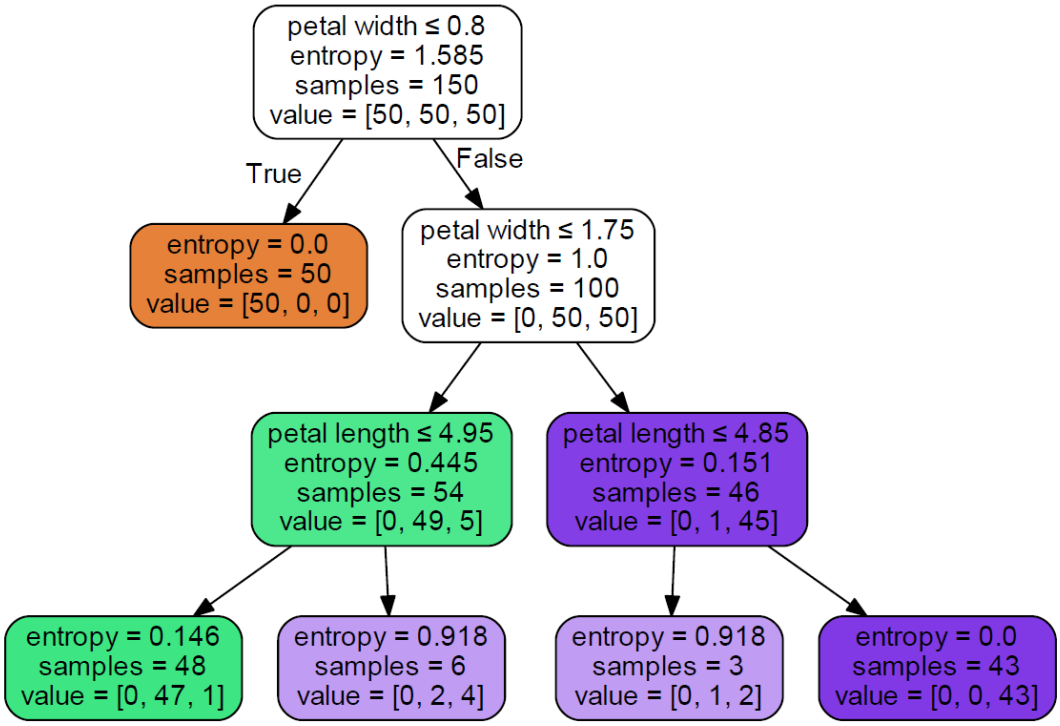




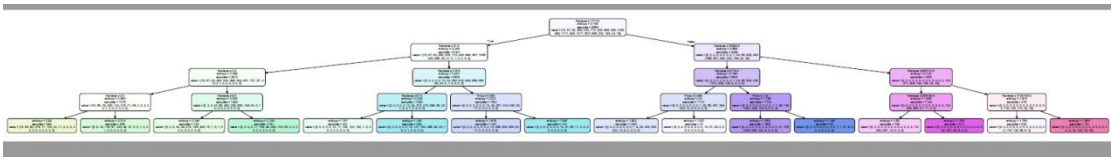




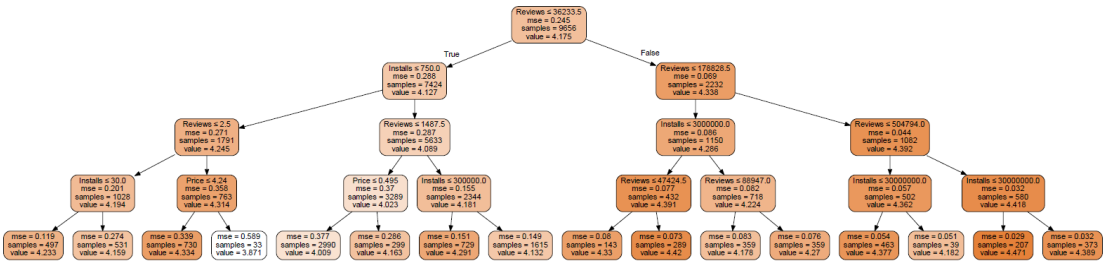
Iris_.pdf



Google_play_store_(Installs) .pdf



Google_play_store_(Rating).pdf



Random forest

1. Iris

```
(base) D:\Downloads\Project1>python Iris_RF.py
Accuracy Evaluate: 0.9333333333333333
```

2. Google app Install

```
(base) D:\Downloads\Project1>
(base) D:\Downloads\Project1>python Google_Rating_RF_Install.py
Evaluate by accuracy score: 0.31377286848463926
```

3. Google app Rating

```
(base) D:\Downloads\Project1>python Google_Rating_RF_Rating.py
Evaluate by r2-score: -0.49458338920986855
```

Validation

1. Iris
   - Decision tree
     - Resubstitution validation

```
(base) D:\Downloads\Project1>python validation_dtree_iris_resub.py

 class = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']

[[50  0  0]
 [ 0 47  1]
 [ 0  3 49]]
precision = [1.         0.97916667 0.94230769]
recall = [1.   0.94 0.98]
accuracy = 0.9733333333333334
```

     - K-fold cross validation

```
(base) D:\Downloads\Project1>python validation_dtree_iris_K_fold.py

 class = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']

[[50  0  0]
 [ 0 43  4]
 [ 0  7 46]]
precision = [1.         0.91489362 0.86792453]
recall = [1.   0.86 0.92]
accuracy = 0.9266666666666666
```

   - Random forest
     - Resubstitution validation

```
(base) D:\Downloads\Project1>python validation_RF_iris_resub.py

 class = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']

[[50  0  0]
 [ 0 48  2]
 [ 0  2 48]]
precision = [1.   0.96 0.96]
recall = [1.   0.96 0.96]
accuracy = 0.9733333333333334
```

- K-fold cross validation

```
(base) D:\Downloads\Project1>python validation_RF_iris_K_fold.py

 class = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']

confusionm matrix
[[50.  0.  0.]
 [ 0. 45.  4.]
 [ 0.  5. 46.]]
precision = [1.         0.91836735 0.90196078]
recall = [1.   0.9  0.92]
accuracy = 0.94
```

2. Google app install
   - Decision tree
     - Resubstitution validation

```
(base) D:\Downloads\Project1>python validation_dtree_google_install_resub.py

 class = ['<=1000', '<=10000', '<=100000', '<=1000000', '<=10000000', 'other']

[[2532  432    8    1    0    0]
 [ 146  762  120    4    1    0]
 [   2  304 1411  627   19    0]
 [   0    0   41 1162  446   13]
 [   0    0    0  128  957   64]
 [   0    0    0    0  120  356]]
precision = [0.85166498 0.73765731 0.5971223  0.69915764 0.83289817 0.74789916]
recall = [0.94477612 0.50867824 0.89303797 0.60457856 0.62022035 0.8221709 ]
accuracy = 0.7435791217895609
```

     - K-fold cross validation

```
(base) D:\Downloads\Project1>python validation_dtree_google_install_K_fold.py

 class = ['<=1000', '<=10000', '<=100000', '<=1000000', '<=10000000', 'other']

[[2522  421    8    1    0    0]
 [ 155  823  278   10    1    0]
 [   3  253 1214  562   17    0]
 [   0    1   80 1235  480   17]
 [   0    0    0  114  950  101]
 [   0    0    0    0   95  315]]
precision = [0.85433604 0.6495659  0.59248414 0.6811914  0.81545064 0.76829268]
recall = [0.94104478 0.5493992  0.76835443 0.64255983 0.61568373 0.72748268]
accuracy = 0.7310480530240265
```

   - Random forest
     - Resubstitution validation

```
(base) D:\Downloads\Project1>python validation_RF_google_install_resub.py

 class = ['<=1000', '<=10000', '<=100000', '<=1000000', '<=10000000', 'other']

[[2255  292   89   43   20    2]
 [  53  508  125   24   13    1]
 [   4   22  481   45   11    1]
 [ 368  676  883 1809 1012  297]
 [   0    0    2    1  487   21]
 [   0    0    0    0    0  111]]
precision = [0.83487597 0.70165746 0.85283688 0.35857284 0.95303327 1.        ]
recall = [0.84141791 0.33911883 0.30443038 0.94120708 0.31561892 0.25635104]
accuracy = 0.5852319801159901
```

- K-fold cross validation

```
(base) D:\Downloads\Project1>python validation_RF_google_install_K_fold.py

 class = ['<=1000', '<=10000', '<=100000', '<=1000000', '<=10000000', 'other']

[[2290  331   74   32   16    2]
 [  95  468  204   37    7    1]
 [  19  128  429  168   20    3]
 [ 273  565  864 1560  965  223]
 [   3    6    9  122  488   61]
 [   0    0    0    3   47  143]]
precision = [0.83424408 0.57635468 0.55932203 0.3505618  0.70827286 0.74093264]
recall = [0.85447761 0.31241656 0.27151899 0.81165453 0.31626701 0.33025404]
accuracy = 0.5569594034797017
```

3. Google app rating
   - Decision tree
     - Resubstitution validation

```
(base) D:\Downloads\Project1>python validation_dtree_google_rating_resub.py

 class = ['<=3', '<=3.5', '<=4', '<=4.5', '<=5']

[[   0    0    0    0    0]
 [   0    0    0    0    0]
 [   5    5    5   11    7]
 [ 356  509 1544 5520 1694]
 [   0    0    0    0    0]]
validation_dtree_google_rating_resub.py:82: RuntimeWarning: invalid value encountered in longlong_scalars
  precision[i] = confusion_matrix_rating[i][i] / sum(confusion_matrix_rating[i])
precision = [       nan        nan 0.15151515 0.57362569        nan]
recall = [0.         0.         0.00322789 0.99801121 0.        ]
accuracy = 0.5721830985915493
```

     - K-fold cross validation

```
accuracy = 0.7455791217895609

(base) D:\Downloads\Project1>python validation_dtree_google_rating_K_fold.py

 class = ['<=3', '<=3.5', '<=4', '<=4.5', '<=5']

[[   0    0    0    0    0]
 [   0    0    0    0    0]
 [  34   54  144  229   45]
 [ 327  460 1399 5283 1637]
 [   0    0    6   19   19]]
validation_dtree_google_rating_K_fold.py:85: RuntimeWarning: invalid value encountered in longlong_scalars
  precision[i] = confusion_matrix_rating[i][i] / sum(confusion_matrix_rating[i])
precision = [       nan        nan 0.28458498 0.58016692 0.43181818]
recall = [0.         0.         0.0929632  0.95516182 0.0111699 ]
accuracy = 0.5640016570008285
```

   - Random forest
     - Resubstitution validation

```
(base) D:\Downloads\Project1>python validation_RF_google_rating_resub.py

 class = ['<=3', '<=3.5', '<=4', '<=4.5', '<=5']

[[  79    2    7   14    0]
 [  11  221   19   23    8]
 [  34   68  998  147   67]
 [ 232  212  500 5293  701]
 [   5   11   25   54  925]]
precision = [0.7745098  0.78368794 0.75951294 0.76289997 0.90686275]
recall = [0.21883657 0.42996109 0.64428664 0.95696981 0.54379777]
accuracy = 0.7783761391880696
```

- K-fold cross validation

```
(base) D:\Downloads\Project1>python validation_RF_google_rating_K_fold.py

 class = ['<=3', '<=3.5', '<=4', '<=4.5', '<=5']

[[   5   13   27   46   17]
 [  21   15   62  163   53]
 [  60   98  323  770  251]
 [ 258  337  946 3934 1180]
 [  17   51  191  618  200]]
precision = [0.0462963  0.0477707  0.2150466  0.59113449 0.18570102]
recall = [0.01385042 0.02918288 0.20852163 0.71126379 0.1175779 ]
accuracy = 0.4636495443247722
```

成員四：曾靖驊

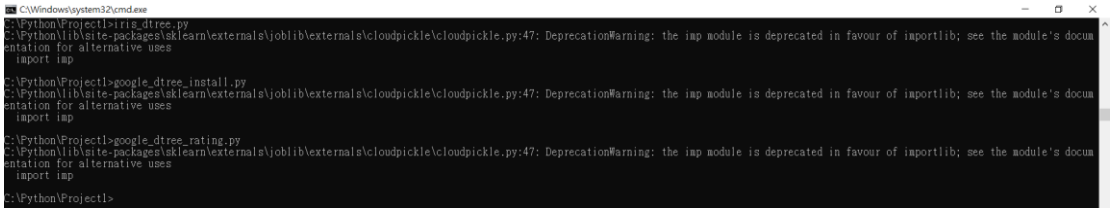
P3

執行以下.py 檔

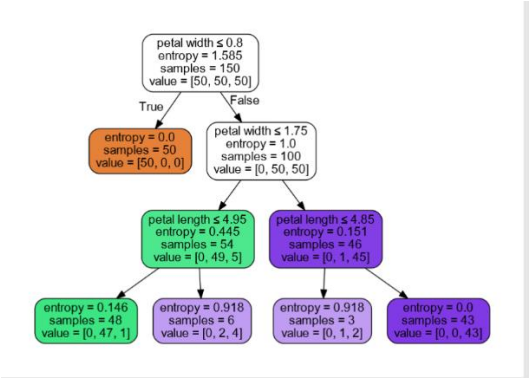
iris_dtree.py

google_dtree_install.py

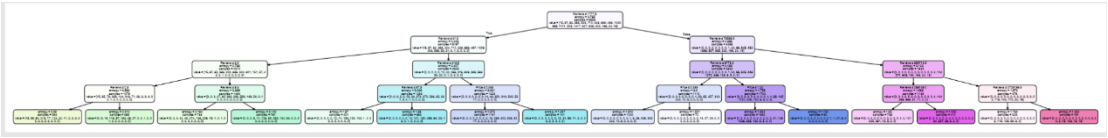
google_dtree_rating.py



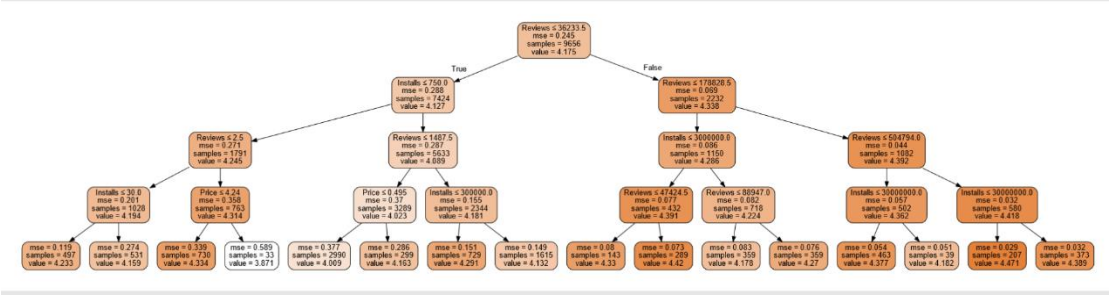
iris_.pdf:



Google_play_store_(Installs) .pdf:



Google_play_store_(Rating).pdf:



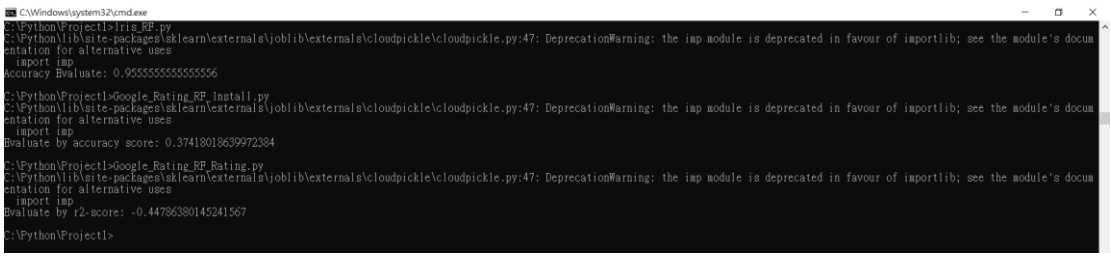
P4

執行以下.py 檔

Iris_RF.py

Google_Rating_RF_Install.py

Google_Rating_RF_Rating.py



P5

執行以下.py 檔

validation_dtree_google_install_K_fold.py

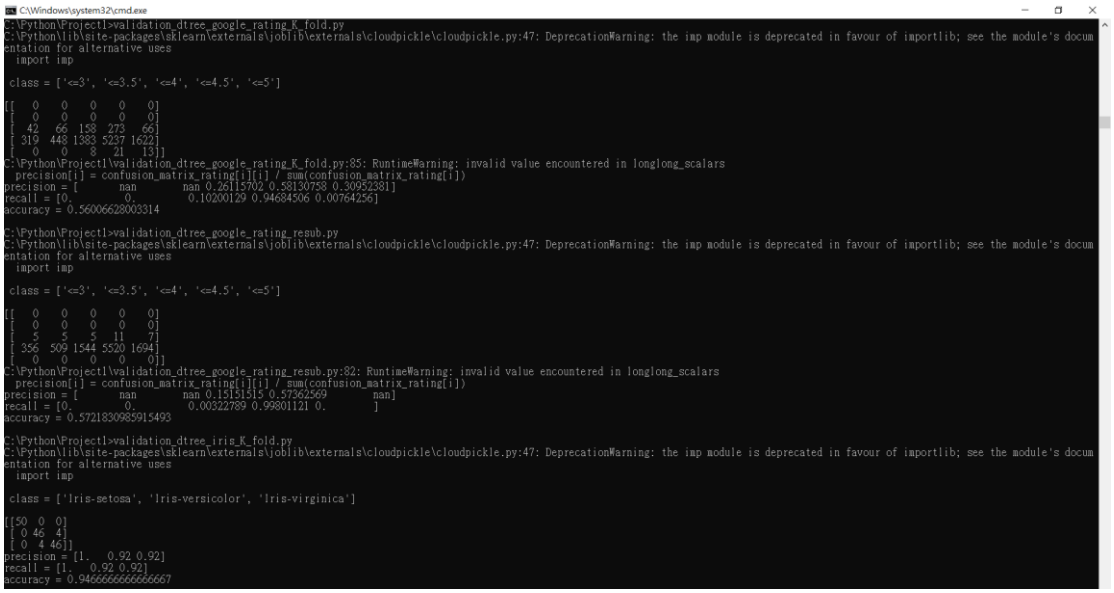
validation_dtree_google_install_resub.py

執行以下.py 檔

validation_dtree_google_rating_K_fold.py

validation_dtree_google_rating_resub.py

validation_dtree_iris_K_fold.py

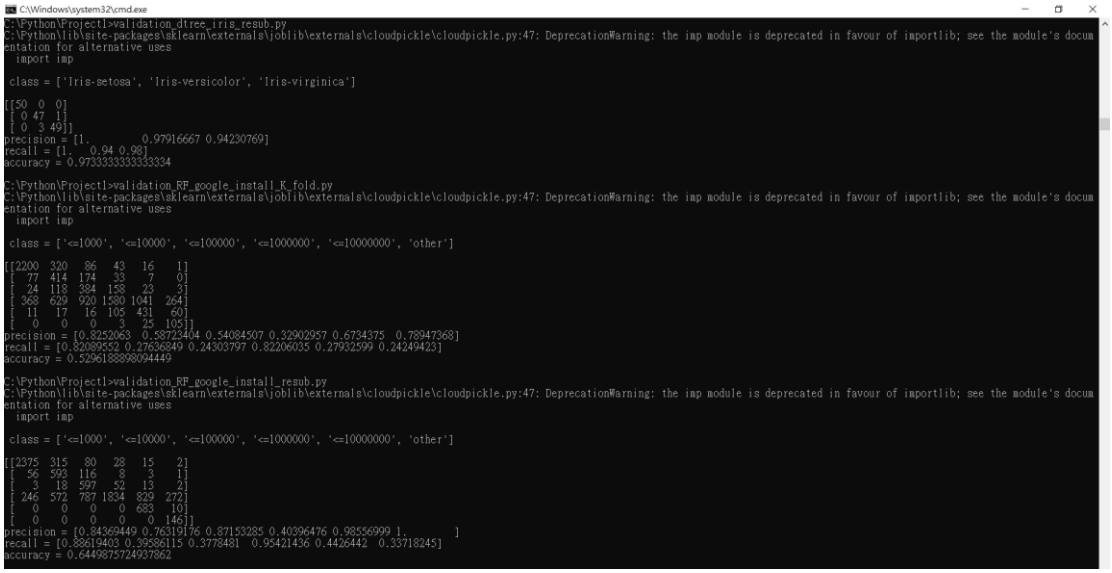


執行以下.py 檔

validation_dtree_iris_resub.py

validation_RF_google_install_K_fold.py

validation_RF_google_install_resub.py

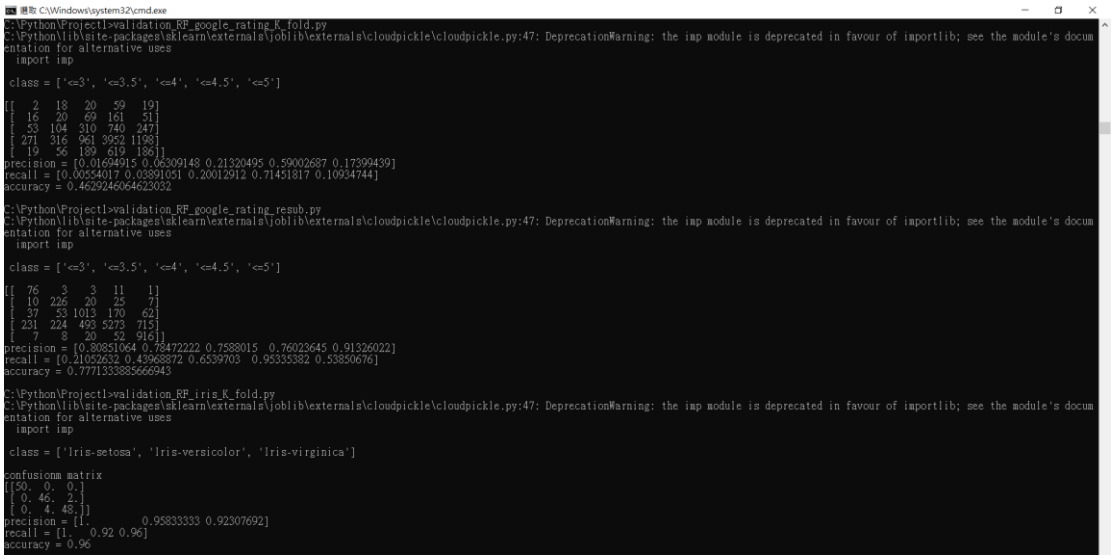


執行以下.py 檔

validation_RF_google_rating_K_fold.py

validation_RF_google_rating_resub.py

validation_RF_iris_K_fold.py



執行以下.py 檔

validation_RF_iris_resub.py