



# 機器學習期末報告

Airbnb New User Bookings

<https://www.kaggle.com/c/airbnb-recruiting-new-user-bookings>



巨資四 B

05170282

陳盈君

英文四 B

05121232

羅苡禎

# 目錄

一、 比賽說明

二、 過程說明

三、 整體分析及檢討

四、 Kaggle 分數

五、 心得

六、 參考資料

## 一、比賽說明

Airbnb旅行者沒有醒來就忽略了“請勿打擾”的標誌，而是發現自己在異想天開的樹屋中與鳥類一同起床，在船屋甲板上喝咖啡，或與房東共享區域早餐。Airbnb的新用戶可以預訂在190多個國家/地區的34,000多個城市中居住的地方。通過準確預測新用戶將在這裡預訂他們的第一次旅行體驗，Airbnb可以與社區共享更多個性化的內容，減少首次預訂的平均時間，並更好地預測需求。

## 二、過程說明

將呈現兩個檔案的執行流程

兩個檔案的原因:第一個檔案基本上是按照老師課堂所教的方法再加上網路的資料參考而完成的，有可能因為資料清楚的不夠乾淨或是最後索取的變數不夠精確，所以分數不高；基於已經修改多次程式碼，於是又製作一份新的檔案。不過第二份程式碼參考較多網路來源(原創部分較少)，因此我們決定將兩份檔案都放上來比較，最後也會附註參考資料來源。

### ■ 大綱:

讀取檔案-理解檔案內容(如:缺失值、資料型態等)-繪製視覺化圖表(幫助自己更好的理解數據分布狀況以及了解某些特殊狀況)-資料清楚(將離群值或是缺失值做刪除或填補的動作)-選擇要放入模型的變數(這步非常重要，是否所選的每個變數與最終結果是有正向關聯)-模型篩選(選擇適合的模型以便有更精確的預測)-檔案儲存-上傳Kaggle-事後檢討、分析-心得(反思)

### ■ 自行製作程式:

檔案讀取	<pre>from sklearn import preprocessing import pandas as pd import numpy as np import matplotlib.pyplot as plt import seaborn as sns  train = pd.read_csv("C:/Users/user/Desktop/train_users_2.csv") test = pd.read_csv("C:/Users/user/Desktop/test_users.csv") age_gen = pd.read_csv("C:/Users/user/Desktop/age_gender_bkts.csv") ses=pd.read_csv("C:/Users/user/Desktop/sessions.csv") country=pd.read_csv("C:/Users/user/Desktop/countries.csv") submit=pd.read_csv("C:/Users/user/Desktop/sample_submission_NDF.csv")</pre>
訓練、測試及資料檢視	<pre>train.info() test.info()  &lt;class 'pandas.core.frame.DataFrame'&gt; RangeIndex: 213451 entries, 0 to 213450 Data columns (total 16 columns): id                213451 non-null object date_account_created  213451 non-null object timestamp_first_active  213451 non-null int64 date_first_booking  88908 non-null object gender            213451 non-null object age              125461 non-null float64 signup_method     213451 non-null object signup_flow      213451 non-null int64 language          213451 non-null object affiliate_channel  213451 non-null object affiliate_provider  213451 non-null object first_affiliate_tracked  213451 non-null object signup_app        213451 non-null object first_device_type  213451 non-null object first_browser     213451 non-null object country_destination  213451 non-null object dtypes: float64(1), int64(2), object(13) memory usage: 26.1+ MB &lt;class 'pandas.core.frame.DataFrame'&gt; RangeIndex: 62096 entries, 0 to 62095 Data columns (total 15 columns): id                62096 non-null object date_account_created  62096 non-null object timestamp_first_active  62096 non-null int64 date_first_booking  0 non-null float64 age              33220 non-null float64 gender            62096 non-null object signup_method     62096 non-null object signup_flow      62096 non-null int64 language          62096 non-null object</pre>

將訓練及測試資料合併(並重設index)

```
df = train.append(test)
df
```

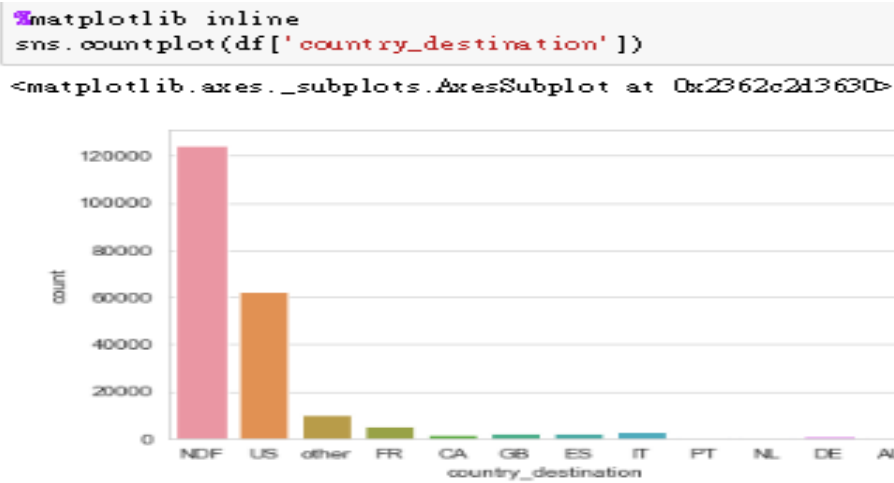
C:\Users\user\Anaconda3\lib\site-packages\pandas\core\frame.py:6211: FutureWarning: Sorting because non-concatenation axis is not aligned. A future version of pandas will change to not sort by default.  
To accept the future behavior, pass 'sort=False'.  
To retain the current behavior and silence the warning, pass 'sort=True'.  
sort=sort)

	affiliate_channel	affiliate_provider	age	country_destination	date_account_created	date_first_booking	first_affiliate_tracked	first_browser	first_device_h
0	direct	direct	NaN	NDF	2010-06-28	NaN	untracked	Chrome	Mac Desk
1	seo	google	38.0	NDF	2011-05-25	NaN	untracked	Chrome	Mac Desk
2	direct	direct	56.0	US	2010-09-28	2010-08-02	untracked	IE	Windows Desk
3	direct	direct	42.0	other	2011-12-05	2012-09-08	untracked	Firefox	Mac Desk

```
df.reset_index(inplace=True,drop=True)
df
```

	affiliate_channel	affiliate_provider	age	country_destination	date_account_created	date_first_booking	first_affiliate_tracked	first_browser	first_device_h
0	direct	direct	NaN	NDF	2010-06-28	NaN	untracked	Chrome	Mac Desk
1	seo	google	38.0	NDF	2011-05-25	NaN	untracked	Chrome	Mac Desk
2	direct	direct	56.0	US	2010-09-28	2010-08-02	untracked	IE	Windows Desk
3	direct	direct	42.0	other	2011-12-05	2012-09-08	untracked	Firefox	Mac Desk

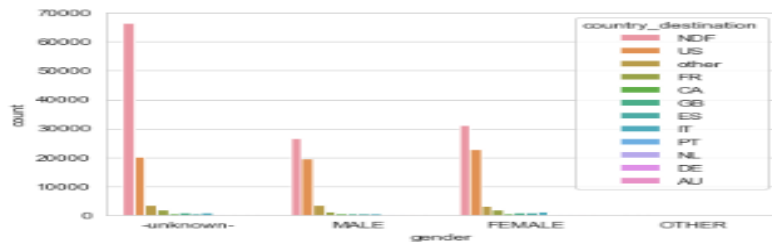
繪製視覺化圖表  
Country\_destination



將部分特徵都做  
成視覺化圖表(觀察  
是否有特殊數  
值)

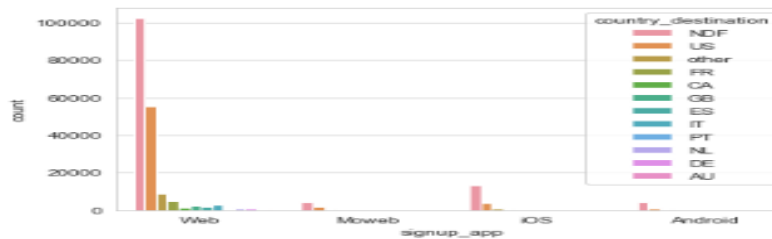
```
sns.countplot(df['gender'],hue=df['country_destination'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2363a6fa198>
```



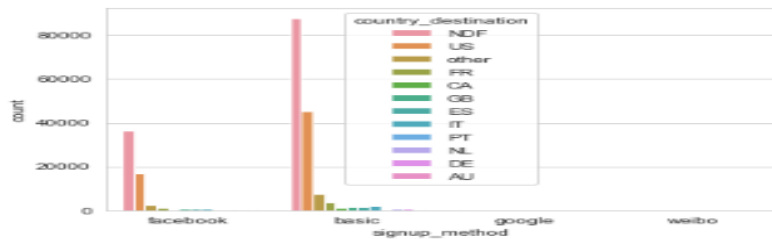
```
sns.countplot(df['signup_app'],hue=df['country_destination'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x23621ac1b00>
```



```
sns.countplot(df['signup_method'],hue=df['country_destination'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2362f96e9b0>
```



檢視合併數據集  
某些變數-  
觀察用戶註冊時  
間到訂房的時間  
差(可看出有少數  
離群值)

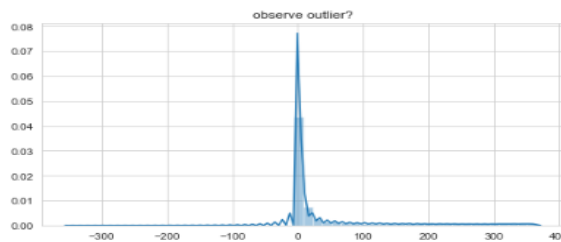
```
##查看用戶註冊後到使用的時間
```

```
df.date_account_created = pd.to_datetime(df.date_account_created)
df.date_first_booking = pd.to_datetime(df.date_first_booking)
day = (df[df.date_first_booking.notna()].date_first_booking - df[df.date_first_booking.notna()].date_account_created)
```

```
day_collect = []
for d in day:
    day_collect.append(int(d.days))
plt.figure(figsize=(8,4))
plt.title('observe outlier?')
sns.distplot(day_collect)
```

```
C:\Users\user\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for n
is deprecated; use 'arr[tuple(seq)]' instead of 'arr[seq]'. In the future this will be interpreted as an array index
which will result either in an error or a different result.
    return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x23620420e48>
```



```
#有些outlier是創帳戶很久之後才用，但大部分的都是沒創多久就使用了(可能是臨時要訂房間所以辦個帳戶這樣)
```

針對年齡這項變數做初步檢視(觀察是否有離群值或是不合理的數值)  
進行年齡數據初步調整(將不合理的值替換掉)

```
df.age.describe()
```

```
count    158681.000000
mean      47.145310
std       142.629468
min        1.000000
25%       28.000000
50%       33.000000
75%       42.000000
max       2014.000000
Name: age, dtype: float64
```

```
df[df.age>1000]['age'].describe()
```

```
count      828.000000
mean     2007.117150
std        22.219408
min     1920.000000
25%     2014.000000
50%     2014.000000
75%     2014.000000
max     2014.000000
Name: age, dtype: float64
```

```
df[df.age<= 18].age.describe()
```

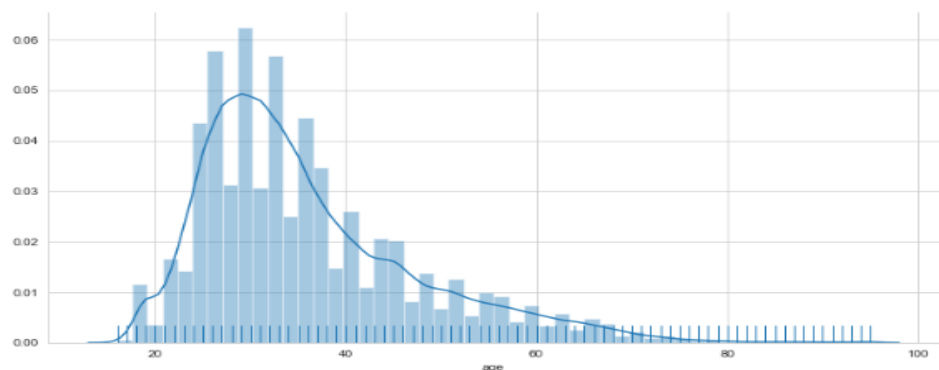
```
count      188.000000
mean       12.718085
std         5.764569
min         1.000000
25%         5.000000
50%        16.000000
75%        17.000000
max        17.000000
Name: age, dtype: float64
```

```
##16歲其實還算合理，5歲不太合理，我們可以將16歲當作一個filter(閾值)把數字濾掉。  
#1歲有可能是會是以公司名義創辦的帳號，有的公司會以創立年份當作歲數(但依舊屬於離群值可清除)
```

年齡數據分佈圖(可看出普遍用戶年齡落點位置)

```
##做年齡分佈圖  
plt.figure(figsize = (12 , 6))  
sns.distplot(df.age.dropna() , rug = True)  
sns.despine()
```

```
C:\Users\user\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple  
is deprecated; use 'arr[tuple(seq)]' instead of 'arr[seq]'. In the future this will be interpreted as  
which will result either in an error or a different result.  
    return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



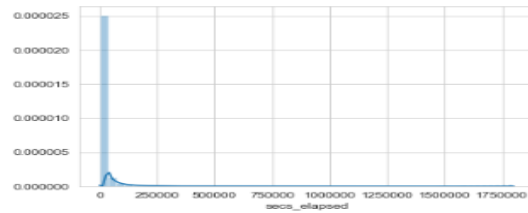
```
##觀察年齡分佈  
#發現25-40歲左右是主要客群
```

Sessions資料讀取後，取變數停留時間做視覺化圖表(了解用戶普遍停留時長)

```
sns.distplot(ses[ses['secs_elapsed'].notnull()][['secs_elapsed']])

C:\Users\user\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple
is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as
which will result either in an error or a different result.
    return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

<matplotlib.axes._subplots.AxesSubplot at 0x2362404bc88>
```

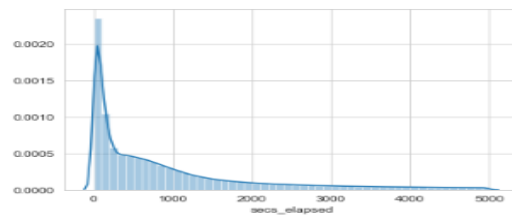


#幾乎所有會話的時間都少於5000秒

```
sns.distplot(ses[(ses['secs_elapsed'].notnull()) & (ses['secs_elapsed'] < 5000)][['secs_elapsed']])

C:\Users\user\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple
is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as
which will result either in an error or a different result.
    return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

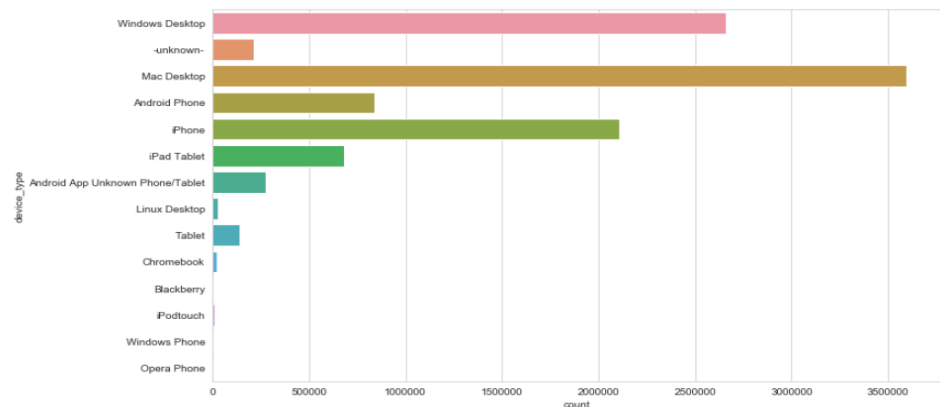
<matplotlib.axes._subplots.AxesSubplot at 0x2363ff32d30>
```



#如上所示，人們正在使用13種類型的設備。

```
plt.figure(figsize=(12,7))
sns.countplot(y='device_type', data=ses)
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x2362215ff60>



使用設備分佈圖

#觀察男女目的地數據 不考慮從未預訂過Airbnb的用戶，或從未在列入行程的國家（NDF和其他）進行預訂的用戶。

```
df_inf = df[(df['country_destination'] != 'NDF') & (df['country_destination'] != 'other') & (df['gender'] != 'OTHER') & (df['gender'].notnull())]
df_inf = df[['id', 'gender', 'country_destination']]
df_inf.head()
```

	id	gender	country_destination
0	gx3p5htnn	-unknown-	NDF
1	820tgsxq7	MALE	NDF
2	4ft3gnwmtx	FEMALE	US
3	bjlt8pjhuk	FEMALE	other
4	87mebub9p4	-unknown-	US

```
observed = df_inf.pivot_table('id', ['gender'], 'country_destination', aggfunc='count').reset_index()
del observed.columns.name
observed = observed.set_index('gender')
observed
```

	AU	CA	DE	ES	FR	GB	IT	NDF	NL	PT	US	other
gender												
-unknown-	143	461	284	715	1713	758	1040	66870	227	69	20106	3469
FEMALE	207	455	358	853	1962	881	1091	31048	254	78	22894	3160
MALE	188	477	416	677	1335	682	699	26719	278	69	16457	3443
OTHER	1	5	3	4	13	3	5	106	3	1	116	22

觀察男女目的地數據(可用做後面分析變數時，性別是否會與結果有正向關係)

觀察註冊方法和註冊設備關係

#註冊方法和註冊設備之間的關係。  
 #大多數用戶通過標準的基本方法或通過Facebook進行註冊。  
 #多數使用桌面瀏覽器，移動瀏覽器或移動應用程序登錄。

#設備的類型（移動或計算機）是否會影響Airbnb的註冊方法  
 #忽略Google註冊方法，因為實例較少。  
 #我們將iOS，Noweb和Android視為移動設備類型。

```
df_signup = df[(df['signup_method'] != 'google')][['id', 'signup_method', 'signup_app']]
df_signup['device'] = df_signup['signup_app'].apply(lambda x: 'Computer' if x == 'Web' else 'Mobile')
df_signup.head()
```

	id	signup_method	signup_app	device
0	gx3p5htnn	facebook	Web	Computer
1	820tgsjq7	facebook	Web	Computer
2	4ft3gnwmtx	basic	Web	Computer
3	bjlt8pjhuk	facebook	Web	Computer
4	87mebub0p4	basic	Web	Computer

```
df_signup['signup_method'].value_counts()
```

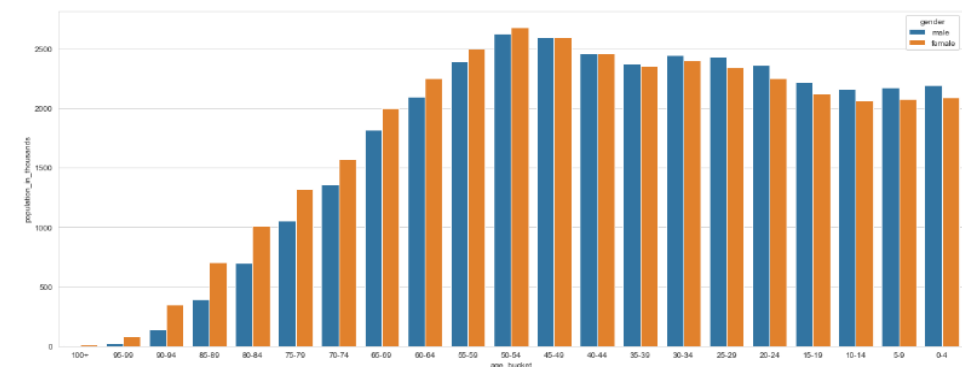
```
basic      198222
facebook   74864
weibo       23
Name: signup_method, dtype: int64
```

```
df_signup['device'].value_counts()
```

```
Computer    219917
Mobile      53192
Name: device, dtype: int64
```

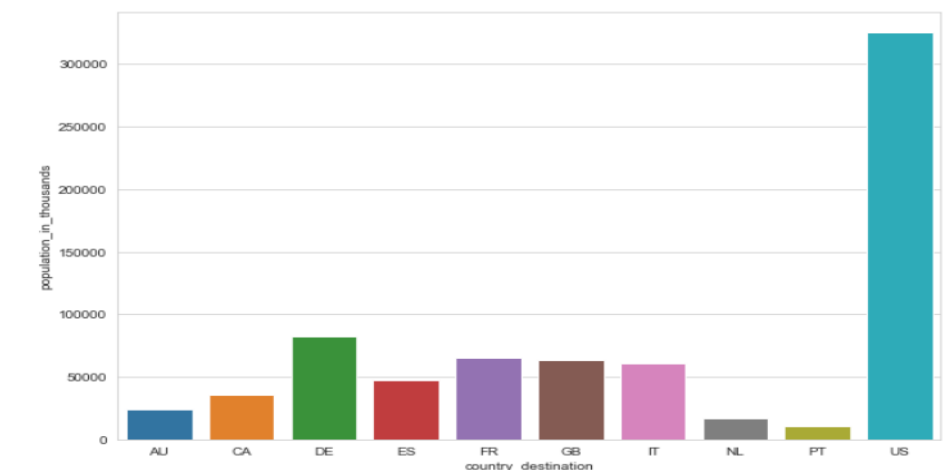
#將各個國家的年齡和性別統計數據可視化。

```
plt.figure(figsize=(20,8))
sns.barplot(x='age_bucket', y='population_in_thousands', hue='gender', data=age_gen, ci=None)
<matplotlib.axes._subplots.AxesSubplot at 0x2961f7d3278>
```



#觀察結果  
 #該統計數據中所代表的國家主要是人口老龄化，最大的人群是50-54歲的人群。  
 #繪製每個國家的人口圖。

```
sns.set_style('whitegrid')
plt.figure(figsize=(10,7))
pop_stats = age_gen.groupby('country_destination')['population_in_thousands'].sum()
sns.barplot(x=pop_stats.index, y=pop_stats)
<matplotlib.axes._subplots.AxesSubplot at 0x236619add68>
```



觀查各國人口統計(了解最大群體的歲數)

透過視覺化圖表-得知美國是用戶最愛前往的國家



模型變數篩選並進行資料類別型態轉換	<pre>df['gender'] = df['gender'].astype('category').cat.codes df['signup_method'] = df['signup_method'].astype('category').cat.codes df['language'] = df['language'].astype('category').cat.codes df['affiliate_provider'] = df['affiliate_provider'].astype('category').cat.codes df['first_browser'] = df['first_browser'].astype('category').cat.codes  df.loc[df.age&gt;120,'age'] = np.nan df.age.fillna(df.age.mean(),inplace=True)  dataTrain = df[pd.notnull(df['country_destination'])].sort_values(by=["id"]) dataTest = df[~pd.notnull(df['country_destination'])].sort_values(by=["id"])  dataTrain.columns  dataTrain = dataTrain[['country_destination','age','gender','language','signup_method','affiliate_provider','first_browser']] dataTest = dataTest[['age','gender','language','signup_method','affiliate_provider','first_browser']] dataTrain</pre>
模型測試分數-採用決策樹	<pre>from sklearn.tree import DecisionTreeClassifier dt = DecisionTreeClassifier() dt.fit(dataTrain.iloc[:,1:], dataTrain.iloc[:,0]) dt.score(dataTrain.iloc[:,1:], dataTrain.iloc[:,0])</pre> <p>0.6568486444195624</p>

## ■ 網路參考程式：

檔案讀取	<pre># Load the data into DataFrames from sklearn import preprocessing import pandas as pd import numpy as np import matplotlib.pyplot as plt import seaborn as sns import matplotlib as mpl import seaborn.apionly as sns  train_users = pd.read_csv("C:/Users/user/Desktop/train_users_2.csv") test_users = pd.read_csv("C:/Users/user/Desktop/test_users.csv") age_gender = pd.read_csv("C:/Users/user/Desktop/age_gender_bkts.csv") sessions=pd.read_csv("C:/Users/user/Desktop/sessions.csv") countries=pd.read_csv("C:/Users/user/Desktop/countries.csv")</pre>
將訓練和測試資料做合併	<pre>users = pd.concat((train_users, test_users), axis=0, ignore_index=True)</pre>
將類別資料做處理	<pre>categorical_features = [     'affiliate_channel',     'affiliate_provider',     'country_destination',     'first_affiliate_tracked',     'first_browser',     'first_device_type',     'gender',     'language',     'signup_app',     'signup_method' ]  for categorical_feature in categorical_features:     users[categorical_feature] = users[categorical_feature].astype('category')</pre>
字串轉換成時間格式	<pre>users['date_account_created'] = pd.to_datetime(users['date_account_created']) users['date_first_booking'] = pd.to_datetime(users['date_first_booking']) users['date_first_active'] = pd.to_datetime(users['timestamp_first_active'], format='%Y%m%d%H%M%S')</pre>
將年齡資料做處理-再以視覺化圖表呈現 可看出部分數值集中在零歲(基本來看不合	<pre>users.loc[users.age &gt; 85, 'age'] = np.nan users.loc[users.age &lt; 18, 'age'] = np.nan users['age'].fillna(-1,inplace=True) bins = [-1, 0, 4, 9, 14, 19, 24, 29, 34,39,44,49,54,59,64,69,74,79,84,89] users['age_group'] = np.digitize(users['age'], bins, right=True)</pre>

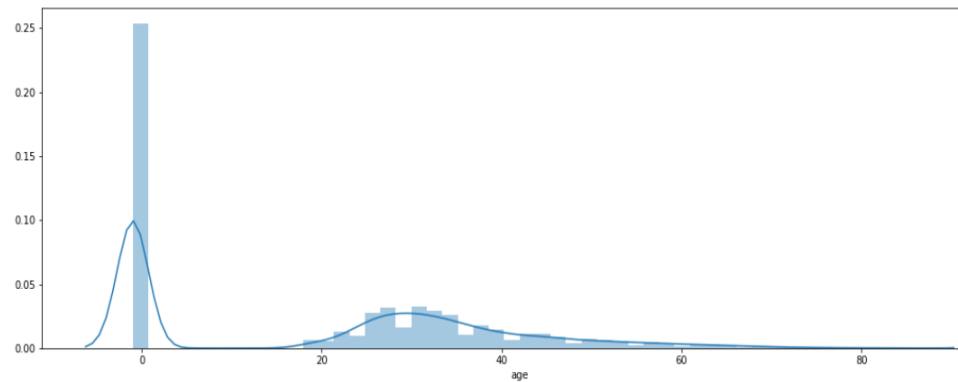
理)

不過也可看出  
主要用戶年齡  
落點在 20-40 之  
間

```
plt.figure(figsize=(18,6))
sns.distplot(users.age.dropna())
```

```
C:\Users\user\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use 'arr[tuple(seq)]' instead of 'arr[seq]'. In the future this will be interpreted as an array index, 'arr[np.array(seq)]', which will result either in an error or a different result.
return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

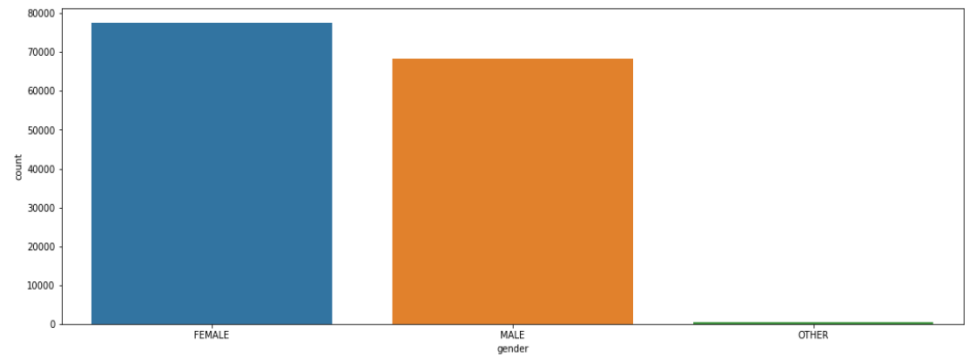
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a701066278>
```



性別資料移除  
unknown 值之後  
呈現狀況

```
plt.figure(figsize=(18,6))
sns.countplot(x='gender', data=users)
```

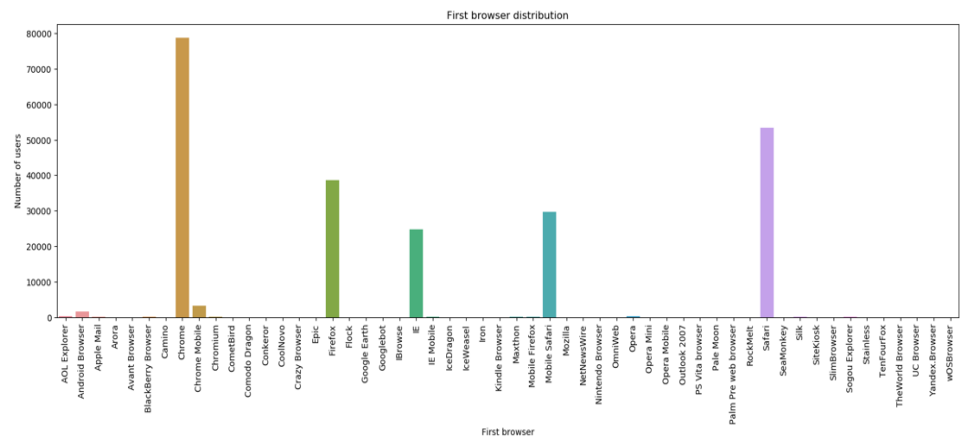
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a7032cfe80>
```



檢視 first  
browser-  
發現最多人使  
用 Chrome，再  
來是 safari

```
# Most of them use:Chrome browser to access Airbnb website.
# Next favourite seems to be Safari browser (Fig.13).
plt.figure(figsize=(20,6))
sns.countplot(x='first_browser', data=users)
plt.xlabel('First browser')
plt.ylabel('Number of users')
plt.title('First browser distribution')
plt.xticks(rotation=90)
```

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
        51, 52, 53]), <a list of 54 Text xticklabel objects>)
```



檢視 sessions 資料(檢查空值)	<pre>display(sessions.isnull().sum())</pre> <pre> user_id          34496 action          79626 action_type     1126204 action_detail   1126204 device_type      0 secs_elapsed    136031 dtype: int64 </pre>																														
檢視使用設備的情況-最多人使用的是 mac desktop	<pre> #the most popular device to access Airbnb seems to be Mac Desktop plt.figure(figsize=(18,6)) sns.countplot(x='device_type', data=sessions) plt.xlabel('Device type') plt.ylabel('Number of sessions') plt.title('Device type distribution') plt.xticks(rotation=90) </pre> <p>(array([ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]), &lt;a list of 14 Text xticklabel objects&gt;)</p>  <table border="1"> <caption>Device type distribution data (approximate values from chart)</caption> <thead> <tr> <th>Device type</th> <th>Number of sessions</th> </tr> </thead> <tbody> <tr><td>Windows Desktop</td><td>2,600,000</td></tr> <tr><td>unknown</td><td>200,000</td></tr> <tr><td>Mac Desktop</td><td>3,500,000</td></tr> <tr><td>Android Phone</td><td>800,000</td></tr> <tr><td>iPhone</td><td>2,100,000</td></tr> <tr><td>iPad Tablet</td><td>600,000</td></tr> <tr><td>Android App Unknown Phone/Tablet</td><td>300,000</td></tr> <tr><td>Linux Desktop</td><td>100,000</td></tr> <tr><td>Tablet</td><td>150,000</td></tr> <tr><td>Chromebook</td><td>50,000</td></tr> <tr><td>BlackBerry</td><td>10,000</td></tr> <tr><td>iPodtouch</td><td>10,000</td></tr> <tr><td>Windows Phone</td><td>10,000</td></tr> <tr><td>Opera Phone</td><td>10,000</td></tr> </tbody> </table>	Device type	Number of sessions	Windows Desktop	2,600,000	unknown	200,000	Mac Desktop	3,500,000	Android Phone	800,000	iPhone	2,100,000	iPad Tablet	600,000	Android App Unknown Phone/Tablet	300,000	Linux Desktop	100,000	Tablet	150,000	Chromebook	50,000	BlackBerry	10,000	iPodtouch	10,000	Windows Phone	10,000	Opera Phone	10,000
Device type	Number of sessions																														
Windows Desktop	2,600,000																														
unknown	200,000																														
Mac Desktop	3,500,000																														
Android Phone	800,000																														
iPhone	2,100,000																														
iPad Tablet	600,000																														
Android App Unknown Phone/Tablet	300,000																														
Linux Desktop	100,000																														
Tablet	150,000																														
Chromebook	50,000																														
BlackBerry	10,000																														
iPodtouch	10,000																														
Windows Phone	10,000																														
Opera Phone	10,000																														
檢視有修正過部分數據的 countries 資料	<pre>countries.info()</pre> <pre> &lt;class 'pandas.core.frame.DataFrame'&gt; RangeIndex: 10 entries, 0 to 9 Data columns (total 7 columns): country_destination    10 non-null object lat_destination         10 non-null float64 lng_destination         10 non-null float64 distance_km            10 non-null float64 destination_km2         10 non-null float64 destination_language    10 non-null object language_levenshtein_distance 10 non-null float64 dtypes: float64(5), object(2) memory usage: 640.0+ bytes </pre>																														
Sessions- 做 rename 的步驟	<pre>sessions.rename(columns = {'user_id': 'id'}, inplace=True)</pre>																														

<p>針對- 每種動作類型的計數、 每個動作經過的時間總和、 每種操作類型經過的時間與用戶經過的總時間的關係.. 等數據。 將所有數據與用戶數據合併</p>	<pre> from sklearn import preprocessing # Create a minimum and maximum processor object min_max_scaler = preprocessing.MinMaxScaler()  action_count = sessions.groupby(['id'])['action'].nunique()  #action_count = pd.DataFrame(min_max_scaler.fit_transform(action_count.fillna(0)),columns=action_count.columns) action_type_count = sessions.groupby(['id', 'action_type'])['secs_elapsed'].agg(len).unstack() action_type_count.columns = action_type_count.columns.map(lambda x: str(x) + '_count') #action_type_count = pd.DataFrame(min_max_scaler.fit_transform(action_type_count.fillna(0)),columns=action_type_count.columns) action_type_sum = sessions.groupby(['id', 'action_type'])['secs_elapsed'].agg(sum)  action_type_pcts = action_type_sum.groupby(level=0).apply(lambda x:  100 * x / float(x.sum())).unstack() action_type_pcts.columns = action_type_pcts.columns.map(lambda x: str(x) + '_pct') action_type_sum = action_type_sum.unstack() action_type_sum.columns = action_type_sum.columns.map(lambda x: str(x) + '_sum') action_detail_count = sessions.groupby(['id'])['action_detail'].nunique()  #action_detail_count = pd.DataFrame(min_max_scaler.fit_transform(action_detail_count.fillna(0)),columns=action_detail_count.columns)  device_type_sum = sessions.groupby(['id'])['device_type'].nunique()  #device_type_sum = pd.DataFrame(min_max_scaler.fit_transform(device_type_sum.fillna(0)),columns=device_type_sum.columns)  sessions_data = pd.concat([action_count, action_type_count, action_type_sum, action_type_pcts, action_detail_count, device_type_sum],axis=1) action_count = None action_type_count = None action_detail_count = None device_type_sum = None  #users = users.join(sessions_data, on='id')  users= users.reset_index().join(sessions_data, on='id') </pre>
<p>Encode categorical features- 對分類特徵進行編碼: 編碼為數值 使用: one hot encoding</p>	<pre> from sklearn.preprocessing import LabelEncoder categorical_features = [     'gender', 'signup_method', 'signup_flow', 'language',     'affiliate_channel', 'age_group', 'weekday_account_created', 'month_account_created', 'weekday_first_active', 'month_first_active', 'hour_first_a',     'signup_app', 'affiliate_provider', 'first_affiliate_tracked', 'first_device_type', 'first_browser' ] users_sc = users.copy(deep=True) encode = LabelEncoder() for j in categorical_features:     users_sc[j] = encode.fit_transform(users[j].astype('str')) </pre>
<p>Feature Selection- 從用戶和 sessions 提取了 54 個功能 因此，執行特徵選擇，減少數量 消除了具有低閾值的變量，並產生最終列表</p>	<pre> colx = users_sc.columns.tolist() rm_list = ['id', 'country_destination'] for x in rm_list:     colx.remove(x) X = users_sc[~(users_sc['country_destination'].isnull())][colx] X.fillna(0, inplace=True) from sklearn.feature_selection import VarianceThreshold sel = VarianceThreshold(threshold=(0.8)) sel.fit_transform(X) idxs = sel.get_support(indices=True) colo = [X.columns.tolist()[i] for i in idxs] print ('\n'.join(colo)) for y in rm_list:     colo.append(y) </pre>

	<pre> affiliate_channel affiliate_provider first_affiliate_tracked first_browser first_device_type gender language signup_flow age_group weekday_account_created month_account_created weekday_first_active month_first_active month_first_book hour_first_active time_lag_create time_lag_active action -unknown-_count click_count data_count message_post_count submit_count view_count -unknown-_sum booking_request_sum booking_response_sum click_sum data_sum message_post_sum partner_callback_sum submit_sum view_sum -unknown-_pct booking_request_pct click_pct data_pct message_post_pct </pre> <hr/> <pre> categorical_features_1 = [val for val in categorical_features if val in colo] users_encode = pd.get_dummies(users[colo], columns=categorical_features_1) </pre>
<p>Countries and Age Bkds dataset- 部分有價值的 信息(但目前可使用部分不多)</p>	<pre> from time import time from math import sqrt import logging import os import sys import csv import datetime  total = {} started = {} model_perf={}  def start(key):     started[key]=time()  def stop(key):     stop=time()     start=started.pop(key,None)     if start:         if key in total:             total[key].append(stop-float(start))         else:             total[key]=[stop-float(start)]     else:         logging.error("stopping non started timer: %s"%key) </pre>

<p>切割訓練、測試數據集 其中， time_lag_create 和 time_lag_active 通過方差閾值方法以及我們構建的決策樹和隨機森林模型都被標為關鍵特徵。 但因為目前模型較不需要，因此排除了這兩項變數</p>	<pre>from sklearn.model_selection import train_test_split users = users_encode users.set_index('id', inplace=True) users.drop([col for col in users.columns if 'pct_booking_request' in col], axis=1, inplace=True) users.drop([col for col in users.columns if 'booking_request_count' in col], axis=1, inplace=True) colx = users.columns.tolist() #colx_1 = users_data_1.columns.tolist() rm_list = ['country_destination', 'month_first_book', 'time_lag_create', 'time_lag_active'] for x in rm_list:     colx.remove(x) X_1 = users[(users['country_destination'].isnull())][colx] X_1.fillna(0, inplace=True) X = users[~(users['country_destination'].isnull())][colx] #X_1 = users_data_1[~(users_data_1['country_destination'].isnull())][colx_1] Y = users[~(users['country_destination'].isnull())]['country_destination'] #Y_1 = users_data_1[~(users_data_1['country_destination'].isnull())]['country_destination'] X.fillna(0, inplace=True) #X_1.fillna(0, inplace=True) #X_res, Y_res = ada.fit_sample(X, Y) X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42, stratify=Y) #X_train_1, X_test_1, Y_train_1, Y_test_1 = train_test_split(X_1, Y_1, test_size=0.2, random_state=42, stratify=Y)</pre>
<p>模型使用隨機森林</p>	<pre>from sklearn.tree import DecisionTreeClassifier dt = DecisionTreeClassifier() dt.fit(X_train, Y_train)</pre> <p>DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None, max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, presort=False, random_state=None, splitter='best')</p>
<p>最後結果預測及存成檔案-便可上傳 kaggle</p>	<pre>y_pred_prob_1 = dt.predict_proba(X_1) id_test = X_1.reset_index()['id'] ids = [] #list of ids cts = [] #list of countries for i in range(len(id_test)):     idx = id_test[i]     ids += [idx] * 5     arr = [dt.classes_.tolist()[k] for k in np.argsort(y_pred_prob_1[i])[::-1]]     cts += arr[:5]  #Generate submission sub = pd.DataFrame(np.column_stack((ids, cts)), columns=['id', 'country']) sub.to_csv('sub_dt.csv', index=False)</pre>

### 三、整體分析及檢討

主要針對自行製作程式方面(網路參考程式的過程都在第二點有介紹):

整體而言，最一開始的數據讀取到做基本視覺化圖表，不管是何種寫法呈現出的結果都是大同小異的。但開始進行細項變數清整後就有落差了。

讀取資料時:

- 發現date\_first\_booking的缺失值很多，而且在測試集中的數據為零，這時便可判斷最終執行時不會採用這項變數。
- 進行年齡讀取時發現有部分極端值(例如:超過100歲或是只有1歲)，某些層面來說是有人故意填寫非真實數值的。這點我曾在實習時遇過，事後發

現會有以公司名義註冊的帳號，所以年齡便是公司成立年數；但這並非個人名義，所以此種數值依舊需要清整(以平均值替代或是刪除)

- 有一點關於設備的特殊發現是多數有訂房的用戶都是使用蘋果相關的產品(舉凡:手機、平板、電腦)，而且美國是最熱門的地點；因此，我們天馬行空的發想，是否普遍用戶都對美國有憧憬(也對他們產品特別偏愛)

最後選取變數要進行模型測試時，我們自行製作的程式選擇了較多認為有相關的變數實測，至於模型的選擇，我們發現決策樹所測出的分數都較好，不管是自行製作還是網路參考的都是。而網路參考的程式也使用蠻多變數，但都有進行更進階的清整，所以最終結果較我們製作的好。而根據我們爬文所了解關於使用模型的資訊是，普遍的人推薦xgboost或是做久了自然能了解何種模型會有較佳的結果，但這次我們都是使用決策樹。


- 關於自行製作程式的反思:我們認為在進行數據清整時應該更仔細(可能要把數據分得更細)，以及再更準確的判斷何種變數是真正和結果有正相關的。
- 關於網路參考程式的反思:見識到許多較複雜的寫法，花了不少時間才能釐清他們如此使用的原因；儘管無法很好地應用到自行製作的程式，但至少知道如何將優點彙整起來且稍微精簡，做出一個較好的模型。

## 四、Kaggle 分數

自行製作程式分數：

您最近的提交				
名稱	已提交	等待時間	執行時間處理時間	得分了
Submit.csv	2分鐘前	0 秒	0 秒	0.57708
完成				
<a href="#">跳至排行榜上的位置</a> ▼				

網路參考程式分數：



airbnb

Airbnb New User Bookings

Where will a new guest book their first travel experience?

1,462 teams · 4 years ago

Overview

Data

Notebooks

Discussion

Leaderboard

Rules

Team

My Submissions

Late Submission

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
sub_dt.csv	just now	0 seconds	6 seconds	0.75418
Complete				
<a href="#">Jump to your position on the leaderboard</a>				



## 五、心得

### ■ 羅苡禎:

這次的期末報告，確實不好寫。自己要做出像樣的資料清楚、分類回歸，對我來說很不容易。所以，一邊參考老師的上課講義，一邊參考網路上高手們的程式碼。一邊做一邊學，也學到很多資料分析的邏輯和好方法，收穫不少。

學習理論需要真正的應用和實踐，才是一件有意義的事情。做這次的期末報告，像幫自己做了個總複習。身為外系的我程式碼非常待加強。幸好在同組夥伴的幫助下，雖然費了很大的心力，作業最終完成了。

### ■ 陳盈君:

這次的期末報告對我來說是困難的，雖然平時上課有跟上老師的進度、回家作業也有確實完成，但要完全寫出一個原創的程式碼還是有難度的。但在寫的過程中一邊寫一邊思考、了解哪部分的程式碼出了問題應如何修正及整體分析文件的邏輯，這都是期末報告能學到的。雖然自己寫出的程式碼分數有點淒慘，必須參考網路上的資料才能完成報告，但能順利跑出結果也是很欣慰了。這學期學習到的模型、分析方法也透過報告比較理解該如何運用了，期末報告真的讓我獲益良多。

## 六、參考資料

<https://towardsdatascience.com/predicting-destination-countries-for-new-users-of-airbnb-eb0d7db7579f>

<https://towardsdatascience.com/predicting-airbnb-prices-using-machine-learning-in-vancouver-1b42ca52eece>

<https://www.dataquest.io/blog/machine-learning-tutorial/>

<https://github.com/karvenka/kaggle-airbnb>

<https://www.kaggle.com/raghavendrakotala/airbnb-springboard-eda>

<https://medium.com/finformation%E7%95%B6%E7%A8%8B%E5%BC%8F%E9%81%87%E4%B8%8A%E8%B2%A1%E5%8B%99%E9%87%91%E8%9E%8D/%E7%B6%B2%E7%AB%99%E6%8E%A8%E8%96%A6%E7%B3%BB%E7%B5%B1%E5%BB%BA%E7%BD%AE-%E6%88%91%E6%80%8E%E9%BA%BC%E5%88%86%E6%9E%90airbnb%E7%9A%84%E8%B3%87%E6%96%99-1bbac91ba723>

<https://blog.csdn.net/Datawhale/article/details/80847662>

<https://github.com/karvenka/kaggle>

[https://github.com/karvenka/kaggle/blob/master/notebooks/Venkatesan\\_Karthick\\_Final\\_Project\\_Report.ipynb](https://github.com/karvenka/kaggle/blob/master/notebooks/Venkatesan_Karthick_Final_Project_Report.ipynb)

<https://codertw.com/%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80/369869/>