資料分析與學習基石
HW 1 - First visit in Kaggle data

基本資料集描述

資料集名稱：

SpaceX Missions, 2006 - Present

資料集介紹：

SpaceX 是一家民營航太製造商和太空運輸公司，擁有製造、發射、回收、復用運載火箭的技術，其目的是在為了降低太空運輸之成本。
除了 NASA 向 SpaceX 簽約定期提供運送貨物至國際太空站的服務，SpaceX 也提供商業衛星的發射，甚至是美國政府的官方任務都會交由這間成長速度極快的太空運輸公司，使得 SpaceX 在近幾年的宇宙航班迅速增加，因此產生了此份資料集。
在這份資料集之中，每一筆資料代表著一次發射火箭的任務過程，而這筆資料裡面包括了發射火箭時的日期、載重、發射成功與否等16項資訊，各欄位內容細節與描述說明如下所述。

資料欄位：

1. Flight Number : 次此任務所發射火箭之航班編號
2. Launch Date : 發射火箭之日期 (日 月 年)
3. Launch Time : 發射火箭時間 (24小時制)
4. Launch Site : 發射火箭的地點
5. Vehicle Type : SpaceX 用來執行此次任務的火箭種類名稱
6. Payload Name : 火箭上所搭載之執行任務的物體編號名稱
7. Payload Type : 火箭上所搭載之物體種類 (如：飛行船、衛星、太空站補給品等...)
8. Payload Mass : 搭載物體的載重 (KG)

9. Payload Orbit : 執行任務之軌道地點 (如 : Low Earth Orbit (低空地球軌道)、Polar orbit (繞極軌道) ...)
10. Customer Name : 任務客戶名稱
11. Customer Type : 客戶型別 (如 : Business、Government …)
12. Customer Country : 客戶所屬國家
13. Mission Outcome : 紀錄任務為成功或是失敗
14. Failure Reason : 任務失敗原因
15. Landing Type : 降落的型態
16. Landing Outcome : 降落成功與否

資料特性

觀察以上的欄位，可以發現其提供了多項客觀資料，而大致上可以分為兩類。
(一) 與火箭發射本身的相關資料(如 : 火箭種類、搭載之物體)
(二) 與這次發射任務相關的其他資訊(如: 客戶、國家)
利用這些所有的周邊資訊或是發射成功與否等結果，我們可以對 SpaceX 的火箭發射任務進行各種面向的分析，甚至預測。

例如我們可以藉由分析資料得知 :
任務成功與否和各項其他因素是否具有關聯性、降落成功與否和火箭型號之間的關係、哪種火箭較被受青睞、發射失敗主要原因為哪些、SpaceX 的客戶主要來自那些國家或是哪些機構、預測 SpaceX 接下來的發射規劃等...都是我們可以從資料去進一步分析得知的資訊。

Notebook方法介紹與比較

(一) SpaceX missions over time

在這份 Notebook 之中，對於了各項數據資料做了以橫軸為時間，縱軸為個不同項目的作圖分析。通過作圖，我們可以觀察各項資料在近幾年間的變化，如 : 火箭型號使用的變化、降落技術進步而使得主要 Landing Type 近幾年不同、或是隨著時間發展，任務成功的比率是否增加等等...。

方法：

先取出各筆資料的 Lunch Date (省略了 Lunch Time) 並自行寫成一個 function 轉成 timestamp，以便之後作圖可以作為標記時間使用。

```
import calendar
import time
import datetime

month = {v: k for k,v in enumerate(calendar.month_name)}

def dateToTimestamp(d):
    t = d.split()
    d = '{0}/{1}/{2}'.format(month[t[1]], t[0], t[2])
    return time.mktime(datetime.datetime.strptime(d, "%m/%d/%Y").timetuple())

data['timestamp'] = data['Launch Date'].apply(dateToTimestamp)
```

再寫成一個 function (plotOverTime()) ，參數為讀進資料的某一欄位 (如 : Lunch Site)，然後以這個欄位之中，相異的資料項目作為縱軸座標和時間作為橫軸座標來作圖。使得每次呼叫此 function 時可以給定不同欄位資料作為參數，並做出一張與時間的關係圖。

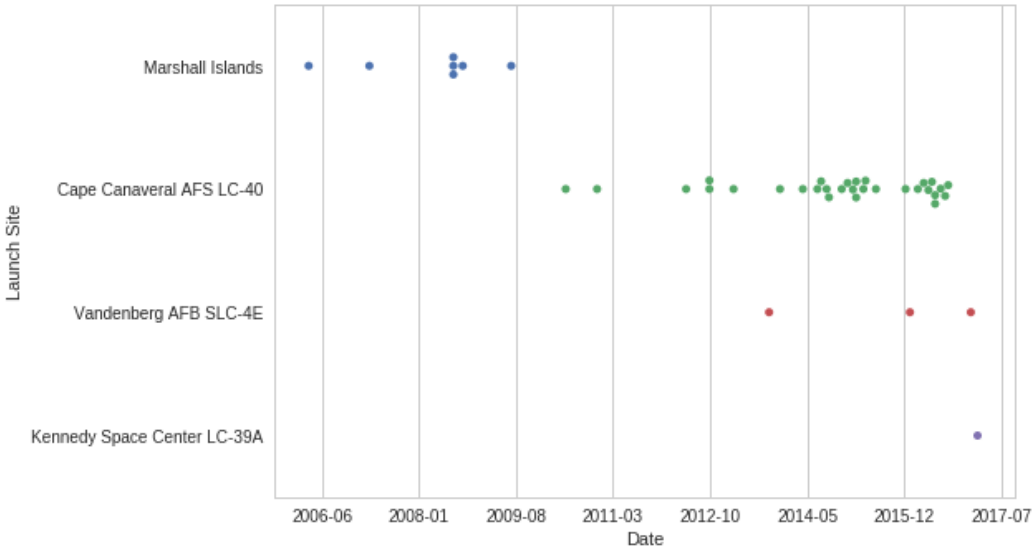
```
def myFormatter(x, pos):
    return datetime.datetime.fromtimestamp(x).strftime('%Y-%m')

def plotOverTime(col):
    ax = sns.swarmplot(x="timestamp", y=col, data=data)
    ax.xaxis.set_major_formatter(matplotlib.ticker.FuncFormatter(myFormatter))
    ax.set(xlabel='Date')
```

For Exemple :

(1) 根據 **發射地點** 所做出的與時間關係圖

```
plotOverTime('Launch Site')
```

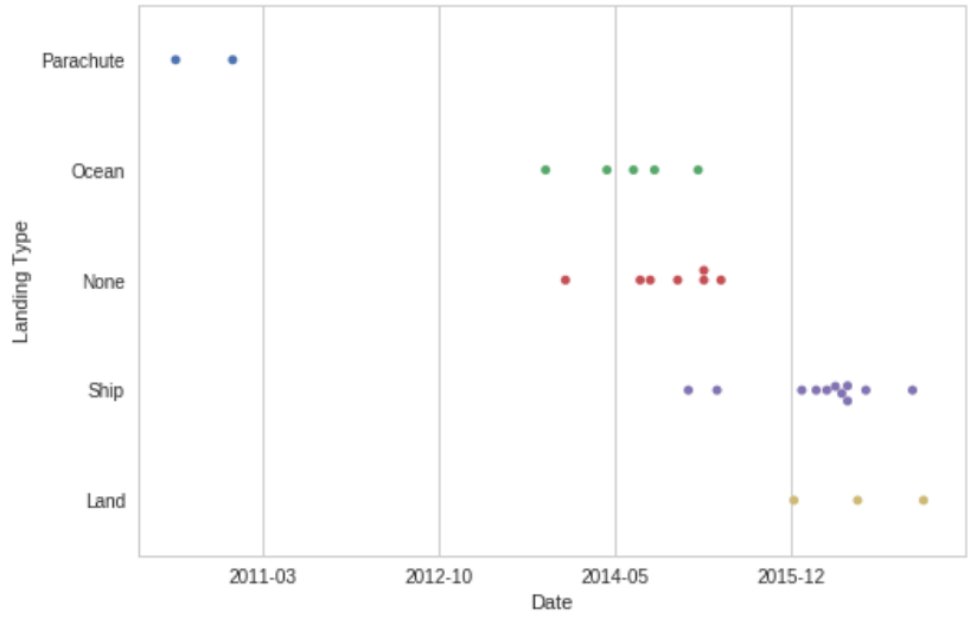


觀察：

可以看到 SpaceX 公司在2005至2009年間大都選擇在Marshall Islands(馬紹爾群島)發射火箭，而後幾年則多在Cape Canaveral AFS LC-40(卡納維爾角空軍基地)執行航太發射。

(2) 根據 **降落型態** 所做出的時間關係圖

```
plotOverTime('Landing Type')
```

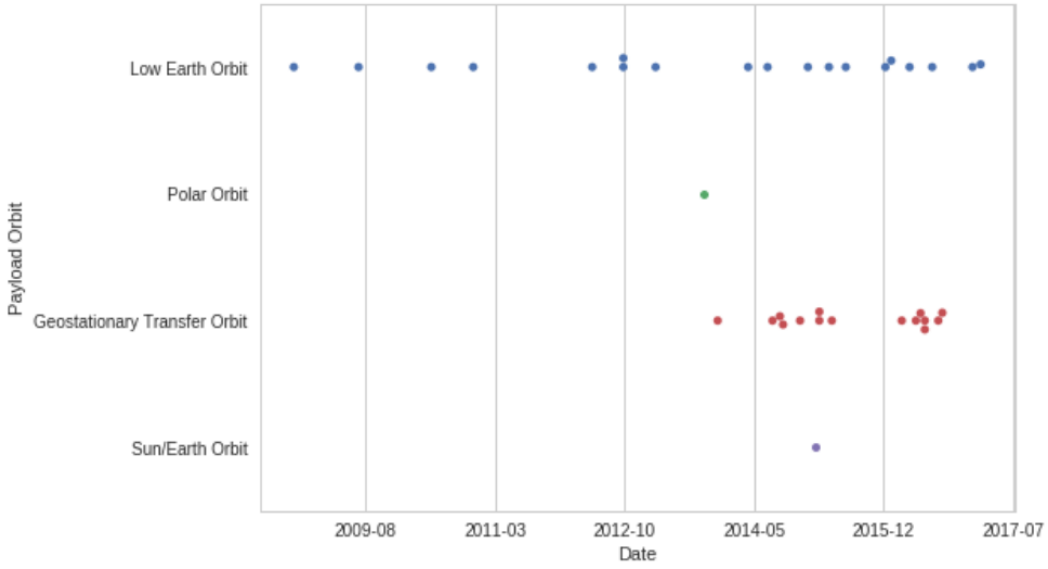


觀察：

從圖中可以看到隨著時間的演進和技術的進步，SpaceX從2011年的使用降落傘(Parachute)降落，到中間令設施降落在海洋(Ocean)，現在則能嘗試在陸地上成功降落(Land)。

(3) 根據 **執行任務之軌道地點** 所做出的時間關係圖

```
plotOverTime('Payload Orbit')
```



觀察：

我們可以看到 Low Earth Orbit (低空地球軌道) 是從2009年就持續都有在接觸的太空軌道且近幾年也不曾減少，或許是因為絕大多數衛星、太空站等都採用此軌道。而 Geostationary Transfer Orbit (地球同步轉移軌道)則是近幾年才有在此軌道上執行任務的紀錄，也可以說明技術的進步。

(二) Data Preparation,categorical data

在這份 Notebook 之中，根據各個欄位數據之間的關聯性做出了相依關係的強度分析，或許可以試著找出個欄位間的因果關係，如：不同的客戶與 Payload Orbit 間的關聯性如何、任務結果是否會和某些特定因素具有強烈關聯性等等。

方法：

先將欄位主題中的空白字元改為 " _ "，以便之後使用。

```
columns = df.columns
col = [i.replace(' ', '_') for i in columns]
df.columns = col
```

不同於第一個 Notebook，這裡會處理 **Missing Value** (缺失值) 的部分。

先計算 Missing Value 數量，我們可以發現這份資料中有許多的地方是沒有值的，因此需要去做處理，為的是之後將項目轉成數字時不會出錯。

```
missing_values_count = df.isnull().sum()
total_cells = np.product(df.shape)
total_missing = missing_values_count.sum()
total_cells,total_missing
```

```
(656, 88)
```

由上圖， 可以看到應填的資料格共有656格，但有88格的資料為缺失。

在這裡，他選擇了利用 Pandas 的 fillna() function 來把缺失/未定義的值都補成0。

```
df.fillna(0)
```

接下來就可以準備畫圖的相關處理了。

利用一個 function (convert_categorical_to_int)，將所有欄位內的資料(Dtype : Object)都轉成相對應的 INT 數字，以便之後可以進行 Correlation 的計算，最後作圖。以下為其步驟：

1. 先取出各個資料有哪些欄位

```
objet_columns = df.select_dtypes(include=[object])
colonne = objet_columns.columns
```

2. 此 function 主要會針對單一 column 來看裡面有哪些不同的項目，相同項目給予同值，不同項目給予不同值，也就是一個數字代表其中一個項目。(如 Customer Type 中：所有的 Government = 0, 而 Business = 1)，利用這個 function 來將所有資料完成轉換。

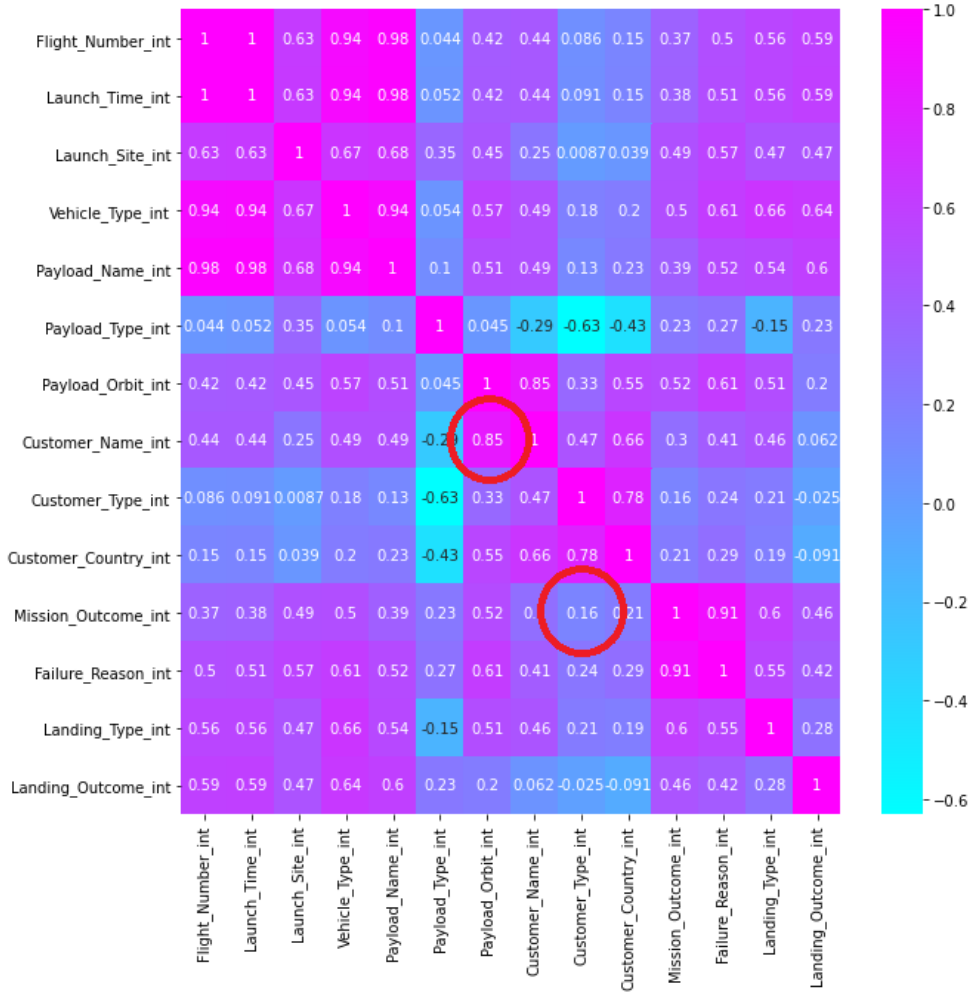
```
def convert_categorical_to_int(col,df) :
    for i in col:
# list unique value for each column
        tab  = df[i].unique()
#     transform to dataframe
        data = pd.DataFrame(tab)
#      add primary key for each dataframe
        data[i+'_int'] = range(len(data))
#     rename the first column
        data = data.rename(columns={0: i})
#     lookup in df to get the primary key
        result = pd.merge(df, data, on=i)
        df = result
    return result

df = convert_categorical_to_int(colonne,objet_columns)
df.info()
```

3. 畫相關係數圖 (df.corr())

```
plt.subplots(figsize = (10 , 10))
sns.heatmap(df.corr(),annot=True,cmap = 'cool')
```

最後結果：



觀察：

相關係數的取值範圍為 [-1, 1]，當接近1時，表示兩者具有強烈的正相關性；當接近 -1 時，表示有強烈的的負相關性；而若值接近0，則表示相關性很低。

可以看到對角線關聯性必定為1，因為對角線為相同欄位。而我們可以比較其他某些組別的關聯性，如 Customer Name 和 Payload Orbit 的關聯程度0.85，兩者相關程度大，顯示客戶與其所選的任務軌道有相當關係存在；但 Mission Outcome 和 Customer Type 關聯程度只有0.16，表示這兩者之間較無直接關係。

Insight :

透過以上兩種方法，我們可以了解一些單純查看資料本身是無法直接觀察出的資訊，但當然 Notebook 方法並會只侷限在上面這兩種方法以及產出。除了以上兩種，我們也可以使用如次數的分析，統計 SpaceX 的最大客戶是誰、最常在哪個地點發射火箭等；亦或是我們可以透過畫分布圖等工具，來觀察哪種火箭型號最容易使任務成功、載重多少是否會影響火箭發射，甚至是發射失敗的最大原因為何等等。
而我們也可以跟其他 Dataset 做結合，如：想更進一步了解火箭發射與當時的天氣之間的關聯，我們可以結合這份資料跟火箭發射地點、時間等的天氣資料(如：溫度、風向...)來一起做分析，也許可以挖掘到更多資訊。
研究以上的 Notebook 之後，我們發現通過資料分析的過程，能把看似平淡無奇的一筆筆資料，轉換為有用的資訊。以 SpaceX 為例，公司可以透過分析這些統計資料，進而從失敗的任務之中找尋最大的問題並且對症下藥，或許就可以逐漸增加未來的發射成功率，減低花費成本，也相對就能增加公司收益，這也是資料分析的價值之一。