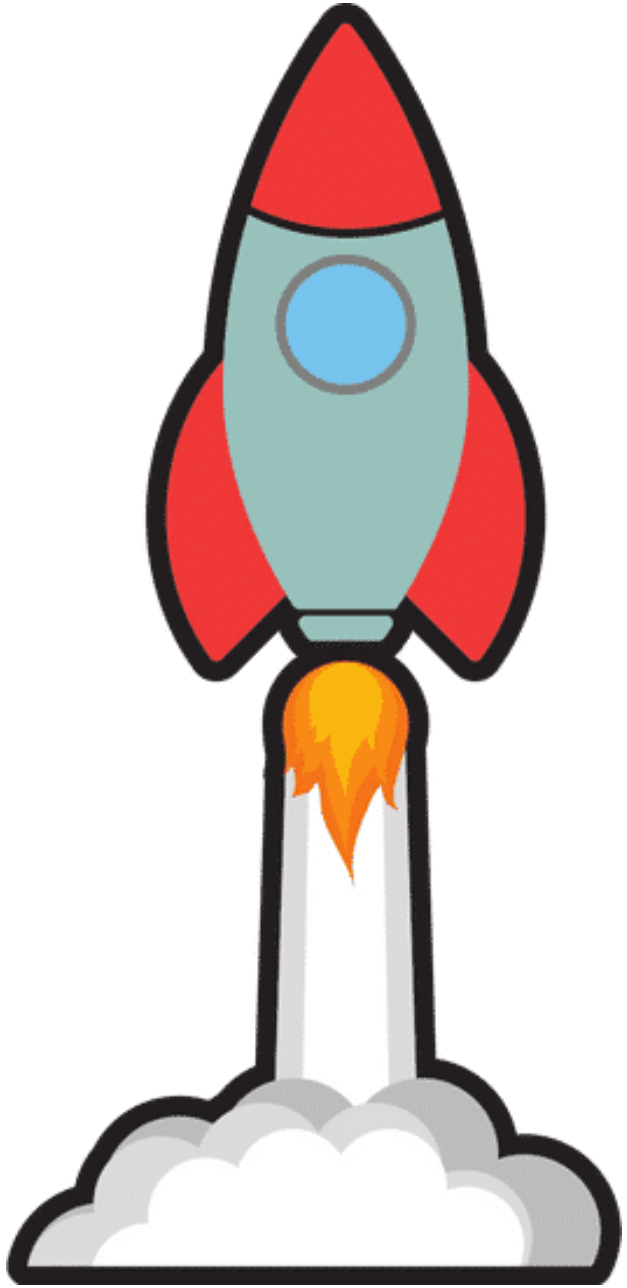


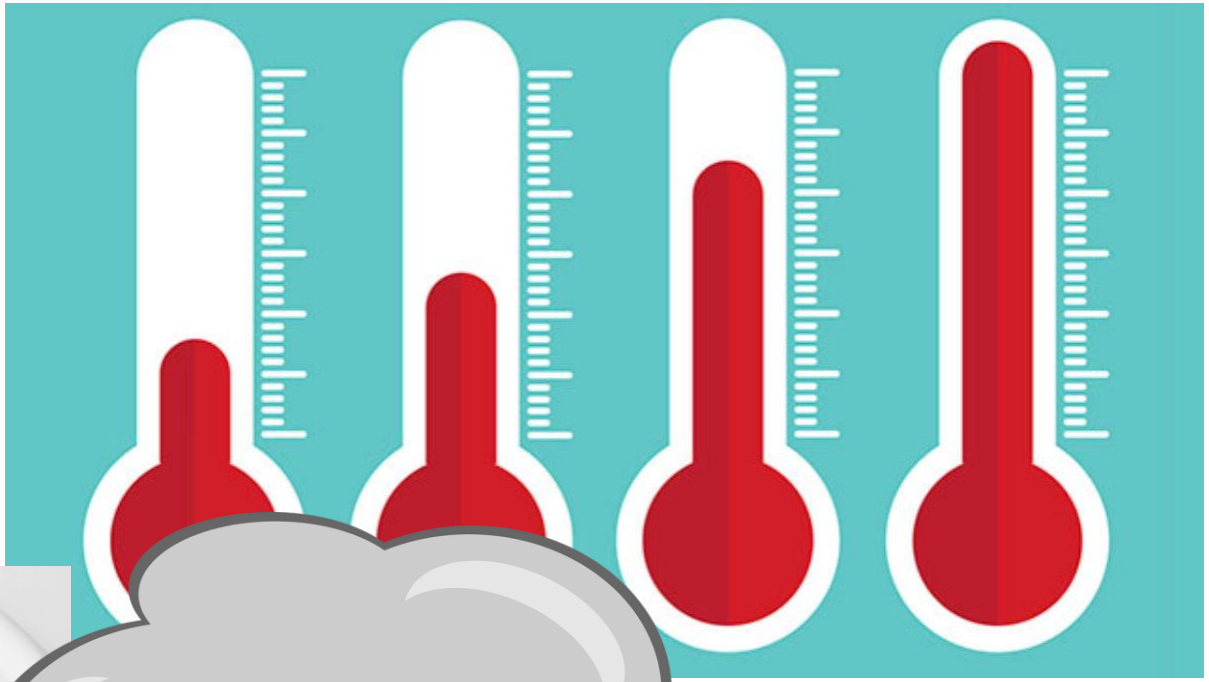
canva.cn

使用機器學習預測火箭發射是否如期

Ryan Chung

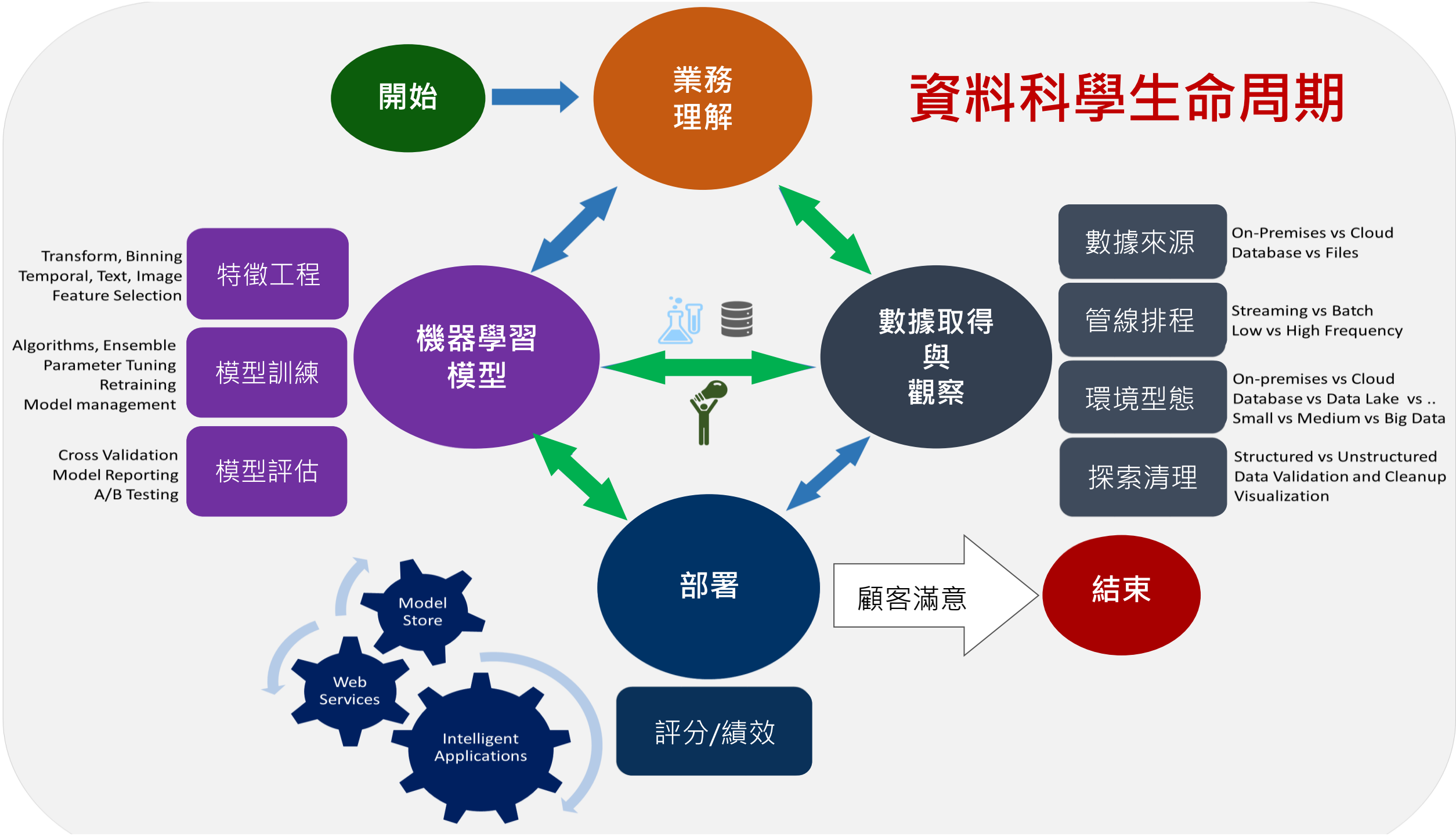
Ryan@MobileDev.TW

火箭發射需要考量哪些因素?

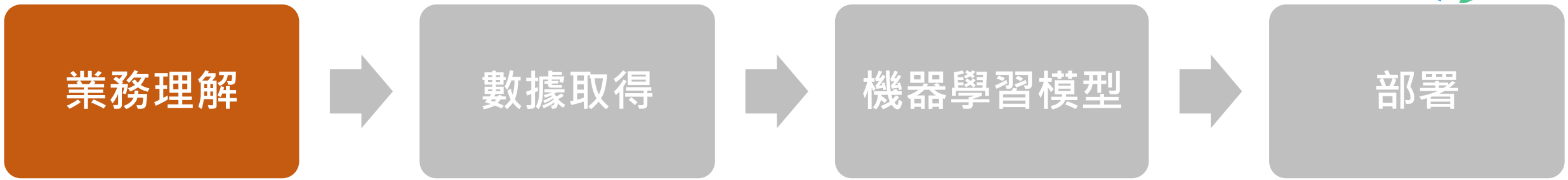


Ryan@MobileDev.TW

資料科學生命周期


開始
業務
理解
機器學習
模型
數據取得
與
觀察
部署
結束
顧客滿意
特徵工程
Transform, Binning
Temporal, Text, Image
Feature Selection
模型訓練
Algorithms, Ensemble
Parameter Tuning
Retraining
Model management
模型評估
Cross Validation
Model Reporting
A/B Testing
數據來源
On-Premises vs Cloud
Database vs Files
管線排程
Streaming vs Batch
Low vs High Frequency
環境型態
On-premises vs Cloud
Database vs Data Lake vs ..
Small vs Medium vs Big Data
探索清理
Structured vs Unstructured
Data Validation and Cleanup
Visualization
Model
Store
Web
Services
Intelligent
Applications
評分/績效

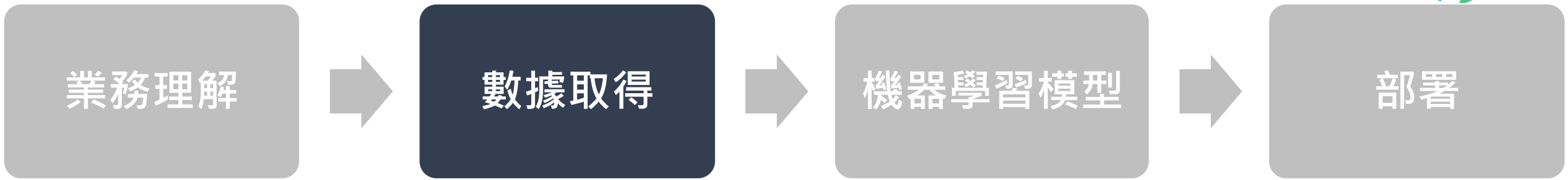
資料科學



- 提高預期火箭發射日是天候良好的可能性
- 瞭解哪些狀況必須一定要停止發射

Ryan@MobileDev.TW

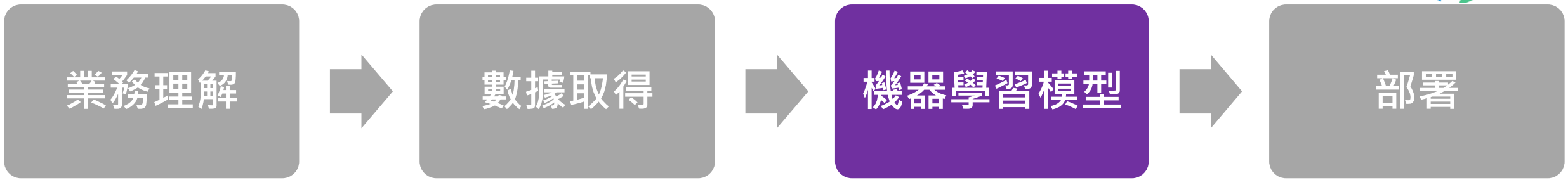
資料科學



- 困難點：實驗成本高，很難不斷嘗試
- 若只有成功發射的案例，識別不良狀況效果會較差

Ryan@MobileDev.TW

資料科學



- 數據中的那些特徵是最關鍵的影響因素?
- 火箭發射
  - 溫度
  - 降水量
  - 濕度
- 目標
  - 使用過去的發射/天氣資料，判斷未來發射是否可能成功

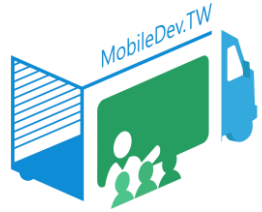
Ryan@MobileDev.TW

數據觀察

- 狀況 (多雲、局部多雲、晴朗、降雨、雷電、暴風雨)
- 溫度
- 濕度
- 風速
- 風向
- 降雨
- 可見度
- 海平面
- 氣壓

Ryan@MobileDev.TW

模組匯入



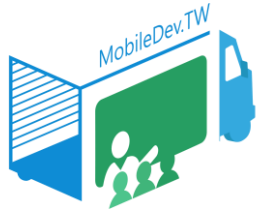

```
# Pandas library is used for handling tabular data
import pandas as pd
# NumPy is used for handling numerical series operations
import numpy as np
# Sklearn library contains all the machine learning packages we need
from sklearn import linear_model, model_selection, metrics
from sklearn.model_selection import train_test_split

# Machine learning libraries used to build a decision tree
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree

# Sklearn's preprocessing library is used for processing and cleaning the data
from sklearn import preprocessing

# for visualizing the tree
import pydotplus
from IPython.display import Image
```

數據匯入與觀察




```
launch_data = pd.read_excel('RocketLaunchDataCompleted.xlsx')
launch_data.head()
launch_data.columns
```

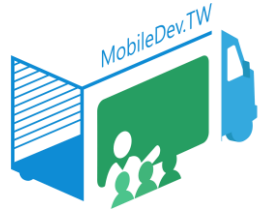
	Name	Date	Time (East Coast)	Location	Crewed or Uncrewed	Launched?	High Temp	Low Temp	Ave Temp	Temp at Launch Time	...	Max Wind Speed	Visibility	Wind Speed at Launch Time	Hist Ave Max Wind Speed	Hist Ave Visibility	Sea Level Pressure	Hist Ave Sea Level Pressure	Day Length	Condition	Notes
0	NaN	1958-12-04	NaN	Cape Canaveral	NaN	NaN	75.0	68.0	71.00	NaN	...	16.0	15.0	NaN	NaN	NaN	30.22	NaN	10:26:00	Cloudy	NaN
1	NaN	1958-12-05	NaN	Cape Canaveral	NaN	NaN	78.0	70.0	73.39	NaN	...	14.0	10.0	NaN	NaN	NaN	30.2	NaN	10:26:00	Cloudy	NaN
2	Pioneer 3	1958-12-06	01:45:00	Cape Canaveral	Uncrewed	Y	73.0	0.0	60.21	62.0	...	15.0	10.0	11.0	NaN	NaN	30.25	NaN	10:25:00	Cloudy	NaN
3	NaN	1958-12-07	NaN	Cape Canaveral	NaN	NaN	76.0	57.0	66.04	NaN	...	10.0	10.0	NaN	NaN	NaN	30.28	NaN	10:25:00	Partly Cloudy	NaN
4	NaN	1958-12-08	NaN	Cape Canaveral	NaN	NaN	79.0	60.0	70.52	NaN	...	12.0	10.0	NaN	NaN	NaN	30.23	NaN	12:24:00	Partly Cloudy	NaN

5 rows × 26 columns

```
Index(['Name', 'Date', 'Time (East Coast)', 'Location', 'Crewed or Uncrewed',
       'Launched?', 'High Temp', 'Low Temp', 'Ave Temp', 'Temp at Launch Time',
       'Hist High Temp', 'Hist Low Temp', 'Hist Ave Temp',
       'Percipitation at Launch Time', 'Hist Ave Percipitation',
       'Wind Direction', 'Max Wind Speed', 'Visibility',
       'Wind Speed at Launch Time', 'Hist Ave Max Wind Speed',
       'Hist Ave Visibility', 'Sea Level Pressure',
       'Hist Ave Sea Level Pressure', 'Day Length', 'Condition', 'Notes'],
      dtype='object')
```


Ryan@MobileDev.TW

數據匯入與觀察



launch_data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 26 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Name                            60 non-null     object
 1   Date                            300 non-null    datetime64[ns]
 2   Time (East Coast)               59 non-null     object
 3   Location                        300 non-null    object
 4   Crewed or Uncrewed              60 non-null     object
 5   Launched?                       60 non-null     object
 6   High Temp                       299 non-null    float64
 7   Low Temp                        299 non-null    float64
 8   Ave Temp                        299 non-null    float64
 9   Temp at Launch Time             59 non-null     float64
 10  Hist High Temp                  299 non-null    float64
 11  Hist Low Temp                   299 non-null    float64
 12  Hist Ave Temp                   299 non-null    float64
 13  Percipitation at Launch Time    299 non-null    float64
 14  Hist Ave Percipitation          299 non-null    float64
 15  Wind Direction                  299 non-null    object
 16  Max Wind Speed                  299 non-null    float64
 17  Visibility                      299 non-null    float64
 18  Wind Speed at Launch Time       59 non-null     float64
 19  Hist Ave Max Wind Speed         0 non-null      float64
 20  Hist Ave Visibility             0 non-null      float64
 21  Sea Level Pressure              299 non-null    object
 22  Hist Ave Sea Level Pressure     0 non-null      float64
 23  Day Length                      298 non-null    object
 24  Condition                       298 non-null    object
 25  Notes                           3 non-null      object
dtypes: datetime64[ns](1), float64(15), object(10)
memory usage: 61.1+ KB
```


Ryan@MobileDev.TW

遺漏值處理

```
launch_data.isnull().sum()
```

```
Name                                240
Date                                  0
Time (East Coast)                   241
Location                              0
Crewed or Uncrewed                  240
Launched?                           240
High Temp                             1
Low Temp                              1
Ave Temp                              1
Temp at Launch Time                 241
Hist High Temp                        1
Hist Low Temp                         1
Hist Ave Temp                         1
Percipitation at Launch Time          1
Hist Ave Percipitation                1
Wind Direction                        1
Max Wind Speed                        1
Visibility                            1
Wind Speed at Launch Time           241
Hist Ave Max Wind Speed             300
Hist Ave Visibility                 300
Sea Level Pressure                    1
Hist Ave Sea Level Pressure         300
Day Length                            2
Condition                             2
Notes                               297
dtype: int64
```


Ryan@MobileDev.TW

遺漏值處理

- 沒有發射資料的也是當天沒發射，補上N

```
launch_data['Launched?'].value_counts()
launch_data['Launched?'].fillna('N',inplace=True)
launch_data['Launched?'].value_counts()
```

```
launch_data['Launched?'].value_counts()
```

```
Y    59
N     1
Name: Launched?, dtype: int64
```

填補前

```
launch_data['Launched?'].value_counts()
```

```
N    241
Y     59
Name: Launched?, dtype: int64
```

填補後

Ryan@MobileDev.TW

遺漏值處理

- 有太空人的任務比較少，所以都補上沒有太空人的

```
launch_data['Crewed or Uncrewed'].value_counts()
launch_data['Crewed or Uncrewed'].fillna('Uncrewed', inplace=True)
launch_data['Crewed or Uncrewed'].value_counts()
```

```
launch_data['Crewed or Uncrewed'].value_counts()
```

填補前

```
Uncrewed    44
Crewed      16
Name: Crewed or Uncrewed, dtype: int64
```

```
launch_data['Crewed or Uncrewed'].value_counts()
```

填補後

```
Uncrewed    284
Crewed       16
Name: Crewed or Uncrewed, dtype: int64
```


Ryan@MobileDev.TW

遺漏值處理

- 天氣概況僅兩筆無數據，使用一般晴天 "Fair"

```
launch_data['Condition'].value_counts()
launch_data['Condition'].isnull().sum()
launch_data['Condition'].fillna('Fair',inplace=True)
```

```
Cloudy            113
Fair               68
Partly Cloudy      68
Rain               24
T-Storm            12
Thunder             7
Mostly Cloudy       2
Heavy T-Storm       1
Partly Cloudly      1
Windy               1
Light Rain          1
Name: Condition, dtype: int64
```

```
launch_data['Condition'].isnull().sum()
```

```
2
```

填補前

```
Cloudy            113
Fair               70
Partly Cloudy      68
Rain               24
T-Storm            12
Thunder             7
Mostly Cloudy       2
Heavy T-Storm       1
Partly Cloudly      1
Windy               1
Light Rain          1
Name: Condition, dtype: int64
```

```
launch_data['Condition'].isnull().sum()
```

```
0
```

填補後

Ryan@MobileDev.TW

遺漏值處理

- 風向僅1筆無數據，標示為 “unknown"

```
launch_data['Wind Direction'].value_counts()
launch_data['Wind Direction'].isnull().sum()
launch_data['Wind Direction'].fillna('unknown',inplace=True)
```

```
E     80
W     54
NE    42
SE    38
S     28
NW    25
N     19
SW    13
Name: Wind Direction, dtype: int64
```

```
launch_data['Wind Direction'].isnull().sum()
```

```
1
```

填補前

```
E          80
W          54
NE         42
SE         38
S          28
NW         25
N          19
SW         13
unknown     1
Name: Wind Direction, dtype: int64
```

```
launch_data['Wind Direction'].isnull().sum()
```

```
0
```

填補後

Ryan@MobileDev.TW

遺漏值處理

- 其餘數值類的，全部補上 0

```
launch_data.isnull().sum()
launch_data.fillna(0, inplace=True)
```

```
Name                                240
Date                                  0
Time (East Coast)                   241
Location                              0
Crewed or Uncrewed                    0
Launched?                             0
High Temp                             1
Low Temp                              1
Ave Temp                              1
Temp at Launch Time                 241
Hist High Temp                        1
Hist Low Temp                         1
Hist Ave Temp                         1
Percipitation at Launch Time          1
Hist Ave Percipitation                1
Wind Direction                        0
Max Wind Speed                        1
Visibility                            1
Wind Speed at Launch Time           241
Hist Ave Max Wind Speed             300
Hist Ave Visibility                 300
Sea Level Pressure                    1
Hist Ave Sea Level Pressure         300
Day Length                            2
Condition                             0
Notes                               297
dtype: int64
```

填補前

```
Name                                0
Date                                0
Time (East Coast)                   0
Location                            0
Crewed or Uncrewed                  0
Launched?                           0
High Temp                           0
Low Temp                            0
Ave Temp                            0
Temp at Launch Time                 0
Hist High Temp                      0
Hist Low Temp                       0
Hist Ave Temp                       0
Percipitation at Launch Time        0
Hist Ave Percipitation              0
Wind Direction                      0
Max Wind Speed                      0
Visibility                          0
Wind Speed at Launch Time           0
Hist Ave Max Wind Speed             0
Hist Ave Visibility                 0
Sea Level Pressure                  0
Hist Ave Sea Level Pressure         0
Day Length                          0
Condition                           0
Notes                               0
dtype: int64
```

填補後

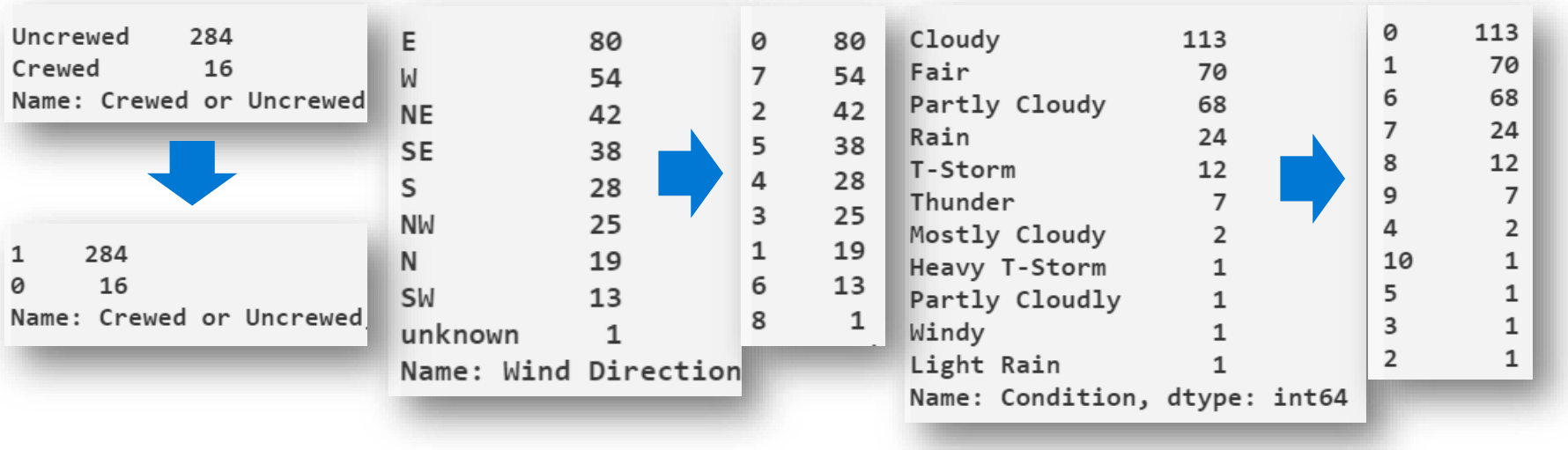
Ryan@MobileDev.TW

數值轉換

- 將三個關鍵但非數值的特徵，進行轉換

```
launch_data.info()
label_encoder = preprocessing.LabelEncoder()
# Three columns have categorical text info, and we convert them to numbers
launch_data['Crewed or Uncrewed'].value_counts()
launch_data['Wind Direction'].value_counts()
launch_data['Condition'].value_counts()

launch_data['Crewed or Uncrewed'] = label_encoder.fit_transform(launch_data['Crewed or Uncrewed'])
launch_data['Wind Direction'] = label_encoder.fit_transform(launch_data['Wind Direction'])
launch_data['Condition'] = label_encoder.fit_transform(launch_data['Condition'])
```



Ryan@MobileDev.TW

數據切割與特徵篩選

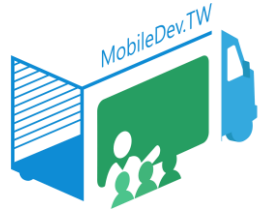
- Launched? 爲預測目標
- 非數值或較不相關的特徵可先剔除

```
y = launch_data['Launched?']
# Removing the columns we are not interested in
X = launch_data.drop(['Name','Date','Time (East Coast)','Location','Launched?',
'Hist Ave Sea Level Pressure','Sea Level Pressure','Day Length','Notes','Hist A
ve Visibility', 'Hist Ave Max Wind Speed'],axis=1)
X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 15 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Crewed or Uncrewed            300 non-null    int64
 1   High Temp                     300 non-null    float64
 2   Low Temp                      300 non-null    float64
 3   Ave Temp                      300 non-null    float64
 4   Temp at Launch Time           300 non-null    float64
 5   Hist High Temp                300 non-null    float64
 6   Hist Low Temp                 300 non-null    float64
 7   Hist Ave Temp                 300 non-null    float64
 8   Percipitation at Launch Time  300 non-null    float64
 9   Hist Ave Percipitation        300 non-null    float64
 10  Wind Direction                300 non-null    int64
 11  Max Wind Speed                300 non-null    float64
 12  Visibility                    300 non-null    float64
 13  Wind Speed at Launch Time     300 non-null    float64
 14  Condition                     300 non-null    int64
dtypes: float64(12), int64(3)
memory usage: 35.3 KB
```


Ryan@MobileDev.TW

模型選擇與數據切割



- 使用DecisionTreeClassifier
- 並依80/20比例進行訓練/測試數據切割
- 進行訓練

```
tree_model = DecisionTreeClassifier(random_state=0, max_depth=5)

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2, random_state=99)

tree_model.fit(X_train, y_train)
```


Ryan@MobileDev.TW

知識檢查

1. 為什麼必須清理資料？

- ○ 如此其他人才可更輕鬆地讀取資料
- ○ 不一致的資料會使電腦混淆
- ○ 使用資料建立更佳視覺效果

2. 下列哪一項不是正在使用的程式庫？

- ○ PyTorch
- ○ pandas
- ○ Sklearn
- ○ NumPy

Ryan@MobileDev.TW

模型使用與成效評估

- 使用模型進行預測
- 評估成效

```
tree_model.fit(X_train, y_train)

predictions = tree_model.predict(X_test)
predictions
metrics.accuracy_score(y_test, predictions)

# is equal to
tree_model.score(X_test, y_test)
```

```
predictions
```

```
array(['N', 'N', 'N', 'Y', 'N', 'N', 'N', 'N', 'Y', 'Y', 'N', 'Y', 'N',
       'N', 'N', 'N', 'N', 'N', 'N', 'N', 'N', 'Y', 'Y', 'N', 'N',
       'N', 'N', 'N', 'N', 'N', 'Y', 'N', 'N', 'N', 'N', 'N', 'N', 'N',
       'N', 'N', 'N', 'Y', 'N', 'N', 'N', 'Y', 'N', 'N', 'N', 'N', 'N',
       'N', 'N', 'N', 'N', 'N', 'N', 'N', 'Y'], dtype=object)
```

```
metrics.accuracy_score(y_test, predictions)
```

```
0.9833333333333333
```

```
tree_model.score(X_test, y_test)
```

```
0.9833333333333333
```


Ryan@MobileDev.TW

數據視覺化

- Graphiz 安裝
  - pip install graphviz
  - 下載 https://graphviz.org/download/
  - 設定系統路徑，讓graphviz在任意路徑均可執行
- 參數準備

數據視覺化

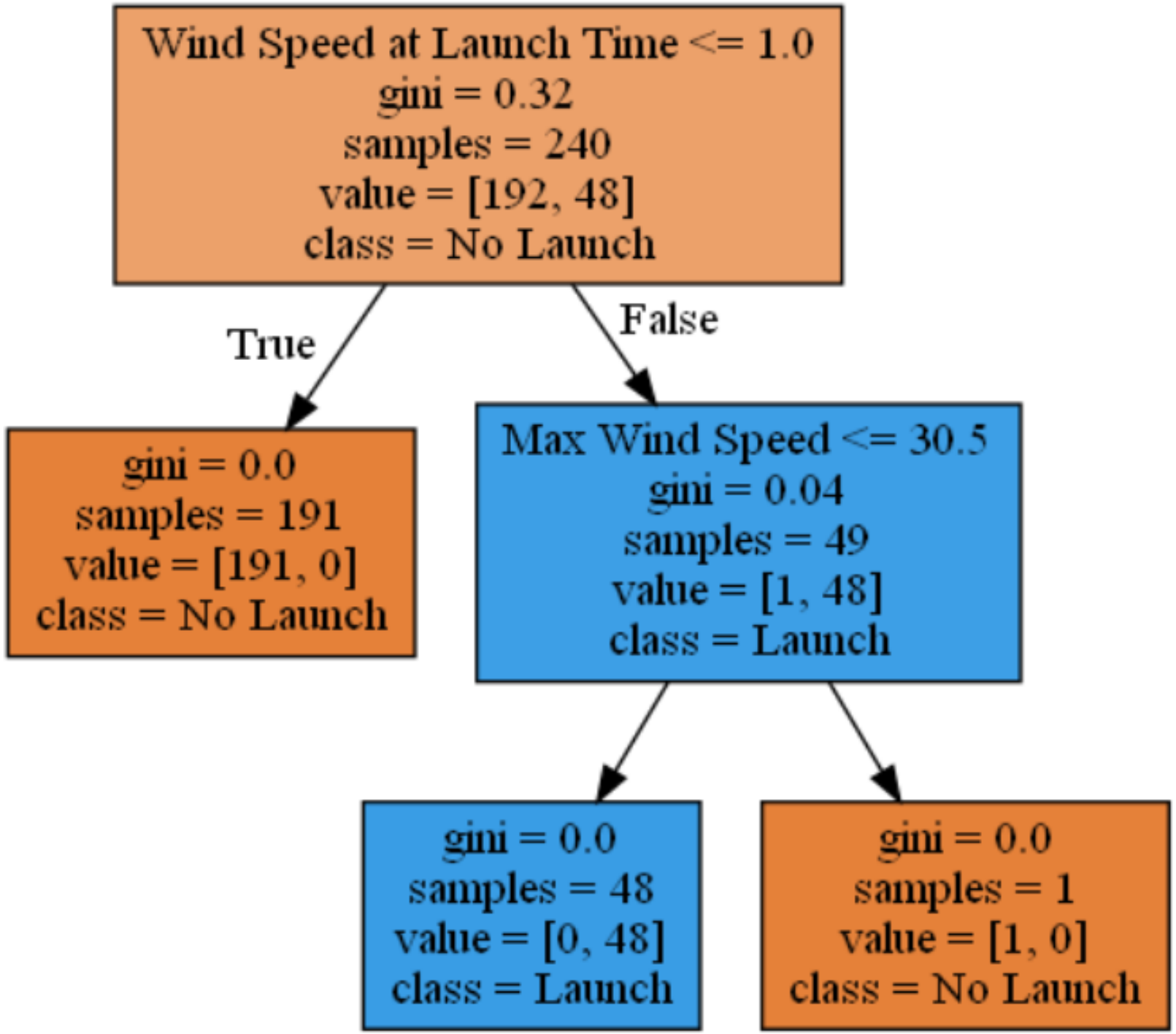
```
# Let's import a library for visualizing our decision tree.
from sklearn.tree import export_graphviz

tree_str = export_graphviz(tree_model, feature_names=X.columns.values,
 class_names=['No Launch', 'Launch'],filled=True, out_file=None)
graph = pydotplus.graph_from_dot_data(tree_str)
Image(graph.create_png())
```

filled ：是否將主要分類節點著色

Ryan@MobileDev.TW

數據視覺化



Ryan@MobileDev.TW

單筆測試

```
X.info()
data_input = [1, 75.0, 68.0, 71.0, 0.0, 75.0, 55.0, 65.0, 0.0, 0.08, 0, 16.0, 15.0, 0.0, 0]
tree_model.predict([data_input])
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 15 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Crewed or Uncrewed            300 non-null    int32
 1   High Temp                     300 non-null    float64
 2   Low Temp                      300 non-null    float64
 3   Ave Temp                      300 non-null    float64
 4   Temp at Launch Time           300 non-null    float64
 5   Hist High Temp                300 non-null    float64
 6   Hist Low Temp                 300 non-null    float64
 7   Hist Ave Temp                 300 non-null    float64
 8   Percipitation at Launch Time  300 non-null    float64
 9   Hist Ave Percipitation        300 non-null    float64
 10  Wind Direction                300 non-null    int32
 11  Max Wind Speed                300 non-null    float64
 12  Visibility                    300 non-null    float64
 13  Wind Speed at Launch Time     300 non-null    float64
 14  Condition                     300 non-null    int32
dtypes: float64(12), int32(3)
memory usage: 31.8 KB
```

```
tree_model.predict([data_input])
```

```
array(['N'], dtype=object)
```


Ryan@MobileDev.TW

單筆測試

- 請參考決策樹模型圖，製造一筆模型會判斷要如期發射的資料

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 15 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Crewed or Uncrewed            300 non-null    int32
 1   High Temp                     300 non-null    float64
 2   Low Temp                      300 non-null    float64
 3   Ave Temp                      300 non-null    float64
 4   Temp at Launch Time           300 non-null    float64
 5   Hist High Temp                300 non-null    float64
 6   Hist Low Temp                 300 non-null    float64
 7   Hist Ave Temp                 300 non-null    float64
 8   Percipitation at Launch Time  300 non-null    float64
 9   Hist Ave Percipitation        300 non-null    float64
 10  Wind Direction                300 non-null    int32
 11  Max Wind Speed                300 non-null    float64
 12  Visibility                    300 non-null    float64
 13  Wind Speed at Launch Time     300 non-null    float64
 14  Condition                     300 non-null    int32
dtypes: float64(12), int32(3)
memory usage: 31.8 KB
```

```
tree_model.predict([data_input2])
```

```
array(['Y'], dtype=object)
```


Ryan@MobileDev.TW

其他探索方向

- 發射前的天氣考量通常包含哪些
- 發射日期是否有在特定的季節?日期?
- 其他國家的發射火箭資料?
- 遺漏值的填補是否有更好的選擇?
- 是否延期的決策重點是什麼?
- 飛機延遲是否考量方式也類似?

Ryan@MobileDev.TW

知識檢查

1. 為什麼我們要針對機器學習演算法選擇決策樹？

- ○ 決策樹是最複雜但最正確的演算法。
- ○ 決策樹很容易視覺化。 因為模型只能進行兩個選擇：[是] 或 [否]，所以相當適合。
- ○ 決策樹有許多分支，而模型則可進行許多選擇。

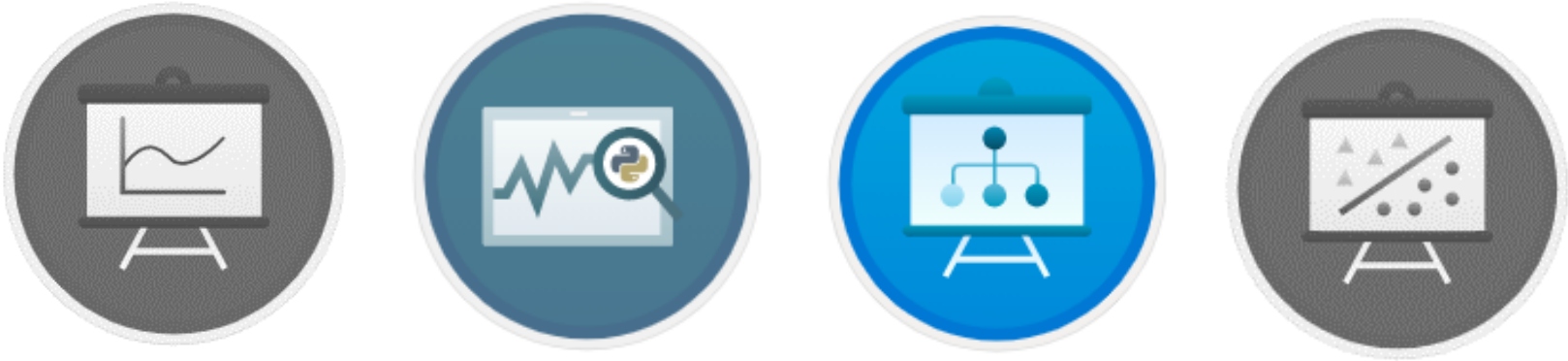
2. 分割資料集的目的為何？

- ○ 透過去除不正確的資料來使模型更加準確。
- ○ 嘗試搭配不同的資料使用不同的演算法。
- ○ 使用不同的資料來對模型進行定型及測試。

Ryan@MobileDev.TW

小結

- 釐清問題種類，決定使用的機器學習算法
- 剔除沒有要用到的特徵，也將要用的特徵數值化
- 使用決策樹來分類，持續兵分兩路，產生最後結果



Ryan@MobileDev.TW