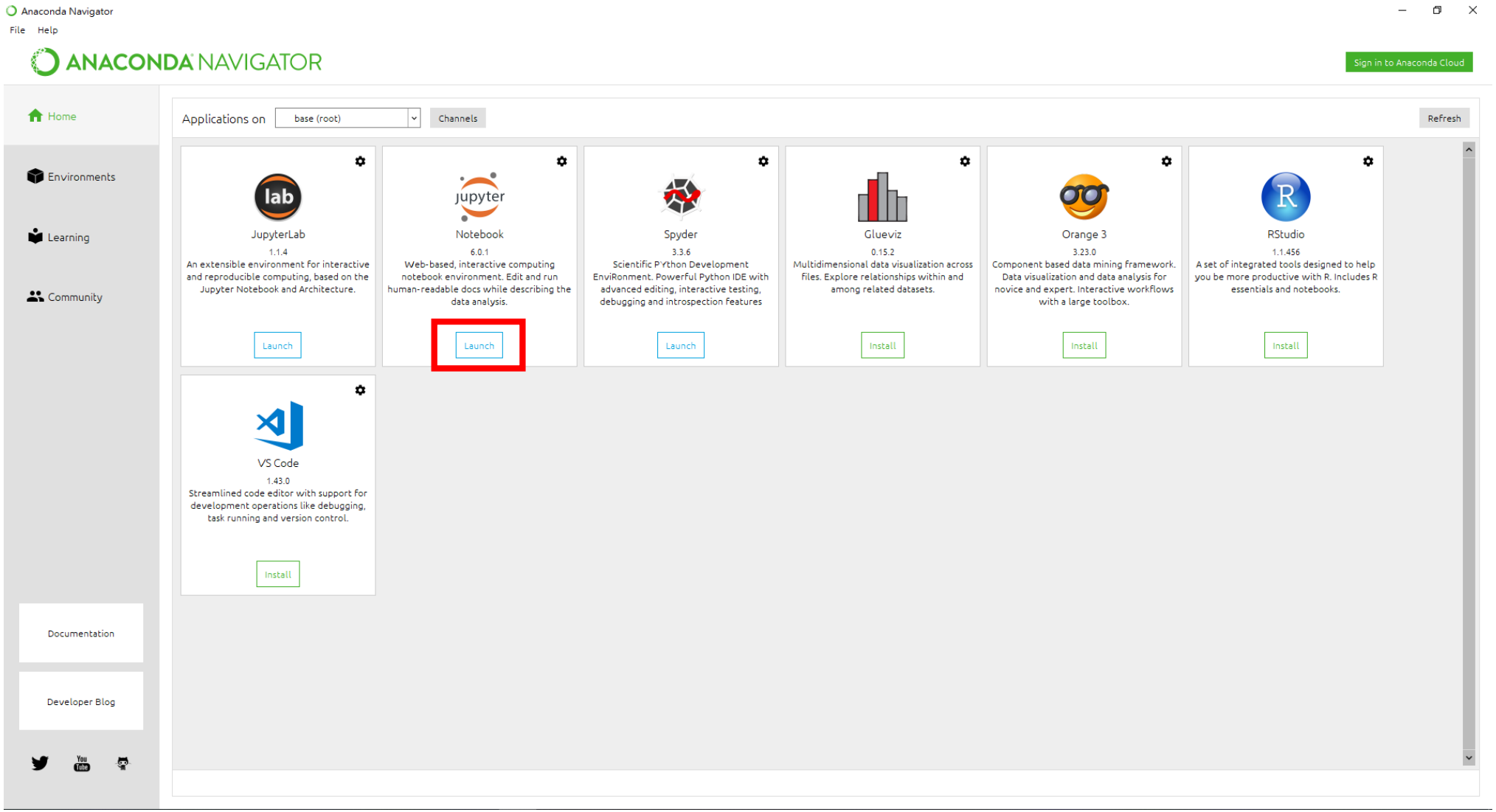
Python example

啟動jupyter notebook



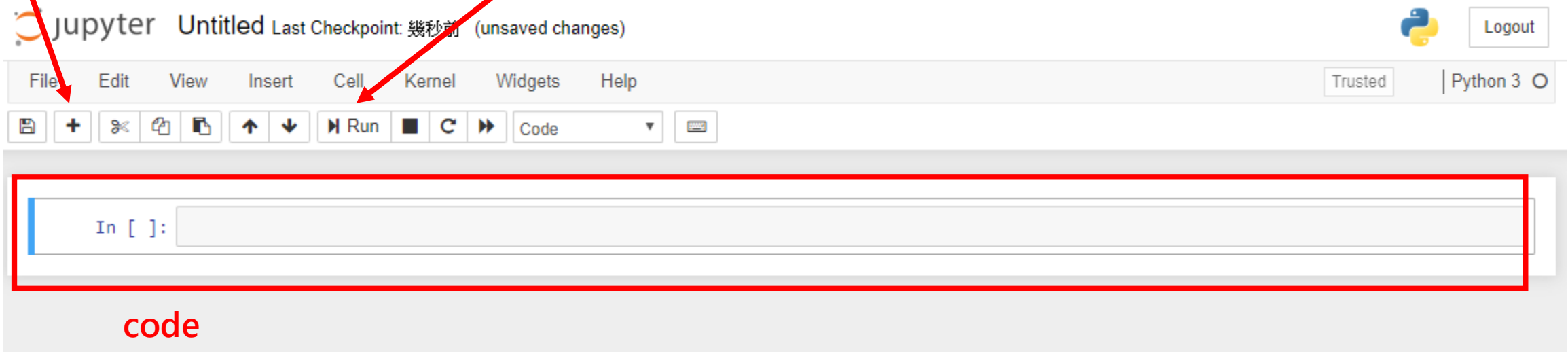
啟動jupyter notebook

啟動jupyter notebook

啟動jupyter notebook

新增cell

執行cell, 可用shift+enter代替



code

Naïve Bayes範例 - weather

- 資料集weather.csv
- 將資料集放至jupyter目錄
  - 預設為 C:\Users\USERNAME

Naïve Bayes範例 - weather

In [1]:
```
import pandas as pd
#讀取CSV檔案
data = pd.read_csv('weather.csv')
```

info ( )

主要可以看有幾筆資料、每個欄位的資料型別是什麼**(int, float..)**、有無空值**(null)**的存在、佔據多少記憶體

In [2]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14 entries, 0 to 13
Data columns (total 5 columns):
outlook        14 non-null object
temperature    14 non-null object
humidity       14 non-null object
windy          14 non-null bool
play           14 non-null object
dtypes: bool(1), object(4)
memory usage: 590.0+ bytes
```

Naïve Bayes範例 - weather

describe ( )

主要是看資料的平均值、分佈情況、是否有資料傾斜**Skew**的問題

```
In [4]: data.describe ( )
```

Out[4]:

	outlook	temperature	humidity	windy	play	
count	14	14	14	14	14	資料數
unique	3	3	2	2	2	幾筆不同資料
top	rainy	mild	normal	False	yes	數量最多的資料
freq	5	6	7	8	9	出現頻率

Naïve Bayes範例 - weather

切分input 和output

```
#x:input
x=data.loc[:,['outlook','temperature','humidity','windy']]  特徵欄位
#y:output
y=data.loc[:,['play']]  目標欄位
```

sklearn: Naive Bayes Classifier

In [6]:
```
from sklearn import preprocessing
#將屬性轉為數字label
le = preprocessing.LabelEncoder()
#outlook: overcast: 0, rainy: 1, sunny: 2
X_outlook_encoded=le.fit_transform(x.outlook)
X_temp_encoded=le.fit_transform(x.temperature)
X_humidity_encoded=le.fit_transform(x.humidity)
X_windy_encoded=le.fit_transform(x.windy)

#將play轉為數字label
#play: no: 0 ,yes: 1
Y_play_label=le.fit_transform(y.play)

#將屬性合併
#變成list
feature=list(zip(X_outlook_encoded, X_temp_encoded,X_humidity_encoded,X_windy_encoded))

#轉成array
import numpy as np
features=np.asarray(feature)
```

Naïve Bayes範例 - weather

訓練模型：訓練集

In [7]:
```
#Import Gaussian Naive Bayes 模型 (高斯樸素貝氏)
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()

# 訓練集訓練模型
# model.fit(x, y)
model.fit(features, Y_play_label)
```

Out[7]:
```
GaussianNB(priors=None, var_smoothing=1e-09)
```

測試集測試模型

In [8]:
```
expected = Y_play_label
predicted = model.predict(features)
from sklearn import metrics
print(metrics.classification_report(expected, predicted))
```

```
              precision    recall  f1-score   support

           0       1.00      0.80      0.89         5
           1       0.90      1.00      0.95         9

    accuracy                           0.93        14
   macro avg       0.95      0.90      0.92        14
weighted avg       0.94      0.93      0.93        14
```

Naïve Bayes範例 - weather

```
In [9]: print(metrics.confusion_matrix(expected, predicted))
```

```
[[4 1]
 [0 9]]
```