GAI Project 1

顏呈育 F64096106

Numpy

練習 1：softmax 函數

實作 Deep Learning 中常用的機率輸出函數 softmax。

若當前任務為 c 分類問題，輸入資料 D 經過神經網路計算後得到的 logits 為 $xinR^c$，則分類 i 的 logits x_i 經過 softmax 得到的結果 p_i 為：

$$p_i = \text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{c} e^{x_j}}$$

```
[0.08243482 0.13231661 0.1009803  0.10385545 0.10038973 0.12891227
 0.08117735 0.11233675 0.0763328  0.0812639 ]
```

練習 2：Linear Layer + ReLU Activation

實作 Deep Learning 中線性神經網路與常用的激發函數（Activation Function）ReLU。

令當前神經網路權重（Weight）為 $W \in R^{d_{out} \times d_{in}}$，偏差值（Bias）為 $b \in R^{d_{out}}$。若神經網路輸入值為 $x \in R^{d_{in}}$，則輸出值 $y \in R^{d_{out}}$ 為：

$$y = \text{ReLU}(Wx + b) = \max(0, Wx + b)$$

```
[[ 0.          0.          5.63686124  0.          0.          6.36785861
   0.         14.14558999 10.38071167  0.          0.         18.90131427
  10.04692259  8.20470951  0.          3.70528603  4.07315001  0.
   0.          0.          0.          0.          0.         13.11209157
   0.          6.77434588  0.          0.          0.          0.        ]]
```

pandas

練習 1：數值轉換

透過台南站氣象觀測站（467410）資料集，將 2022 年 8 月將給定的日最高紫外線指數轉換為相應的紫外線強度等級（低、中、高、甚高、極高），並新增一個欄位表示紫外線強度等級，然後計算每個等級的出現次數：

低：0 ~2、中：3 ~5、高：6 ~7、甚高：8 ~10、極高：11+

```
    day     UV level
0     1  11.36    極高
1     2   4.04     中
2     3  11.09    極高
3     4  11.20    極高
4     5  14.32    極高
5     6  12.18    極高
6     7   7.86     高
7     8  10.59    甚高
8     9   7.48     高
9    10  13.28    極高
10   11  10.20    甚高
11   12  14.75    極高
12   13  13.58    極高
13   14  11.64    極高
14   15   8.33    甚高
15   16  10.73    甚高
16   17  12.77    極高
17   18   7.37     高
18   19  13.84    極高
19   20  13.46    極高
20   21  12.87    極高
21   22  13.90    極高
22   23  13.35    極高
23   24   8.20    甚高
24   25  13.88    極高
25   26  13.17    極高
26   27  12.94    極高
27   28  11.96    極高
28   29  10.57    甚高
29   30  12.94    極高
30   31  12.50    極高
紫外線強度等級「低、中、高、甚高、極高」分別出現 0 、 1 、 3 、 6 、 21 次
```

練習 2：條件篩選

根據給定的降水量（mm）和降水時數（hour），計算出降水強度（mm/hr），並找出降水強度大於平均強度的日期及其相關資訊 (將整個 row print 出)。

(113/3/5 update) 抱歉資料部分沒有規定完善，雨量若為 T 請以 0 取代即可

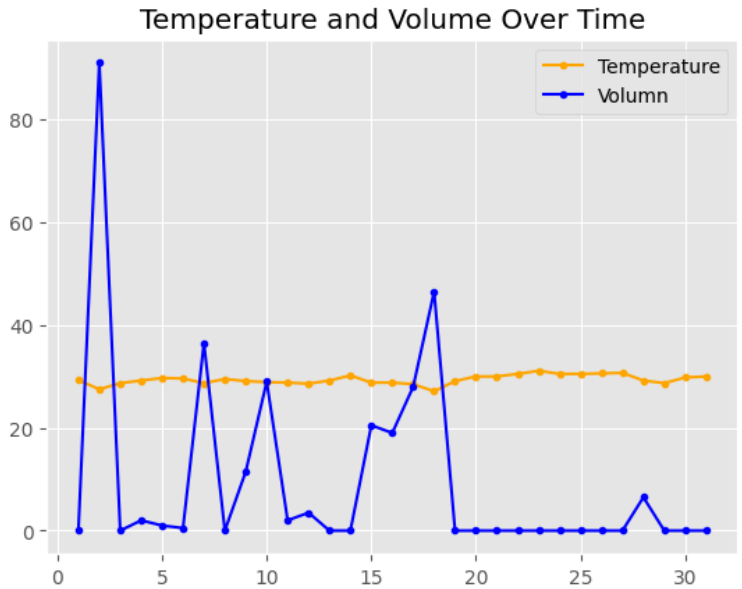
```
平均降水強度 =  5.230778673822601
降水強度大於平均降水強度：
    day  volumn  hour     UV  temperature  speed  direction        str
1     2    91.0   3.6   4.04         27.5    1.9        200  25.277778
6     7    36.5   2.8   7.86         28.7    1.8        100  13.035714
8     9    11.5   1.7   7.48         29.1    1.9         20   6.764706
9    10    29.0   3.1  13.28         28.9    2.4        360   9.354839
14   15    20.5   2.5   8.33         28.8    2.0        120        8.2
15   16    19.0   2.1  10.73         28.8    2.0         20   9.047619
16   17    28.0   4.3  12.77         28.5    2.6        220   6.511628
17   18    46.5   3.6   7.37         27.1    1.9         90  12.916667
```

matplotlib

使用與 pandas 練習相同的氣象資料。

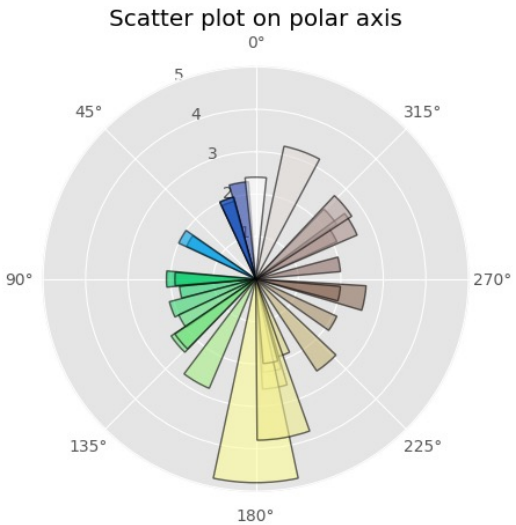
練習 1：折線圖

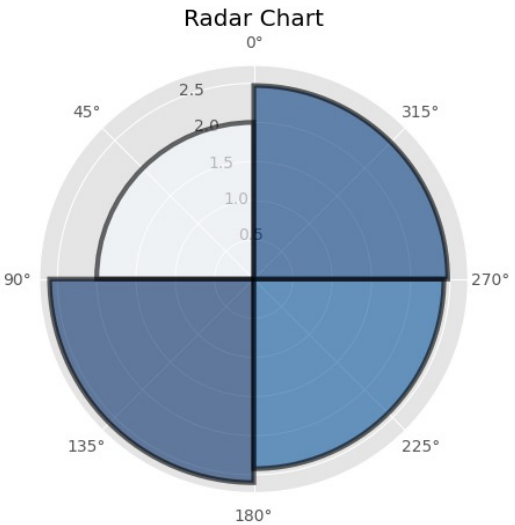
請依照日期畫出氣溫以及雨量的變化，並以折線圖的方式呈現。



練習 2：雷達圖

分析風速和風向之間的關係，對於每個風向角度區間（0-90度、90-180度、180-270度、270-360度），計算相應的平均風速，並繪製成雷達圖以可視化四種風向的風速分佈情況。





scikit-learn

本練習使用 Kaggle Titanic 所提供的資料，根據鐵達尼號乘客資料**預測生還者**。

點選資料集分頁後，點擊 `Download All` 下載所有資料並解壓縮，或是只下載 `train.csv`。

- (113/3/1 update) 關於最終預測資料，請使用 sklearn.model_selection.train_test_split 將 train.csv 分割，參數請設置成：train_size=0.8, random_state=1012，以分割後的 test_data 做最終預測結果的分析基準

- 分割後 test acc 預測原始分數為：0.7262，請改進到超越此分數。

練習 1：改善決策樹分類模型

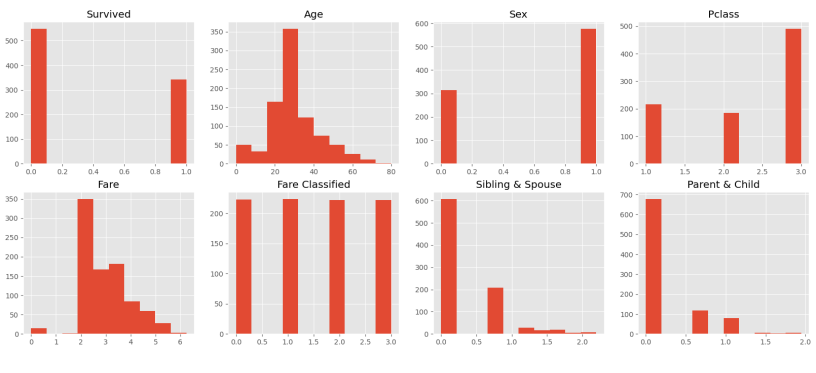
請改進 `sklearn.tree.DecisionTreeClassifier` 模型分類效果，可以嘗試：

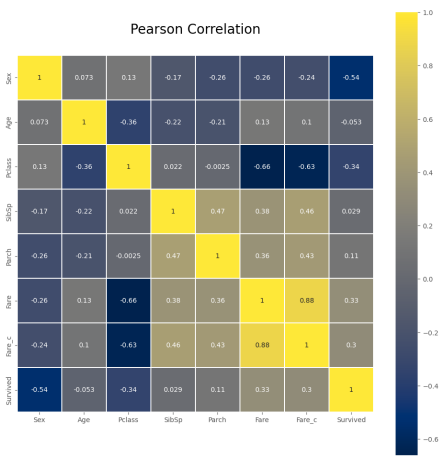
- 增加更多的輸入特徵
- 使用不同的前處理方法
- 調整超參數

```
train accuracy: 0.9143258426966292
valid accuracy: 0.7821229050279329
```

- 經由Pearson相關係數挑選出相關程度較高的四類作為輸入特徵，分別為‘Sex‘ , ’Pclass‘ , ’Fare‘ , ’Fare_c‘ ， 其中Fare_c為以Fare經過四分位數分級的結果。
- 前處理方式包含計算相關係數挑選、OneHot encoding、填補缺失值
- 超參數的調整:

```python
model = DecisionTreeClassifier(
    random_state=1012,
    criterion='entropy',
    max_depth=25,
    max_leaf_nodes=2 ** 25
)
```





練習 2：使用不同的模型

請使用不同的模型打敗使用 `sklearn.tree.DecisionTreeClassifier` 模型分類效果，可以嘗試課堂所提過的方法：
以下列舉只是提供參考，可以自行查訊表現更好的模型

模型	名稱
`sklearn.naive_bayes`	樸素貝氏分類器（Naive Bayes Classifier）
`sklearn.svm`	支援向量機（Support Vector Machines）
`sklearn.neighbors`	近鄰演算法（Nearest Neighbors）
`sklearn.ensemble`	集合學習（Ensemble）

- SVM使用SVC模型：
  - 核函數：高斯核函數RBF(Radial Basis Function)
  - 懲罰係數 C：1.0（過小可能效果差）
  - 擬合半徑 gamma：自動

```
train accuracy: 0.8132022471910112
valid accuracy: 0.7597765363128491
```

- Naive.Bayes使用GaussianNB模型：

```
train accuracy: 0.7401685393258427
valid accuracy: 0.7094972067039106
```

- 鄰近演算法採KNeighborsClassifier模型
  - n_neighbors=3

```
train accuracy: 0.875
valid accuracy: 0.7821229050279329
```

- Ensemble採RandomForestClassifier
  - n_estimators=100

```
train accuracy: 0.9143258426966292
valid accuracy: 0.7821229050279329
```