a.

import numpy as np

import pandas as pd

df = pd.read\_csv('C:/Users/user/Desktop/data\_science/train.csv')

-----🡪讀取train.csv，並命為df

df.count().sort\_values()

-----🡪檢查df每個column非null的值有幾個

df = df.drop(columns = ['Attribute7','Attribute6','Attribute18', 'Attribute19','Attribute2','Attribute1'], axis=1)

df = df.dropna(how='any')

-----🡪藉由剛剛的檢查，丟掉null很多的column(6、7、18、19)，以及column1和2，因為其為地區及日期，不必要之資訊。而dropna則是把有null的row丟掉，確保剩下的row皆沒有null值

from scipy import stats

z = np.abs(stats.zscore(df.\_get\_numeric\_data()))

print(z)

df = df[(z < 3).all(axis=1)]

------🡪這邊是要把較突出的數據刪掉，以免影響之後判斷

df['Attribute22'].replace({'No': 0, 'Yes': 1},inplace = True)

df['Attribute23'].replace({'No': 0, 'Yes': 1},inplace = True)

categorical\_columns = ['Attribute8', 'Attribute10', 'Attribute11']

for col in categorical\_columns:

print(np.unique(df[col]))

df = pd.get\_dummies(df, columns=categorical\_columns)

------🡪把為文字的數據轉為數字，像是22 23的會不會下雨(yes or no)，跟8 10 11吹甚麼風，每個風分出來，有吹這個方向的風就為1。

X = df[['Attribute3','Attribute4','Attribute5','Attribute9','Attribute12','Attribute13','Attribute14','Attribute15','Attribute16','Attribute17','Attribute20','Attribute21','Attribute22']]

y = df[['Attribute23']]

------🡪選取要做模型的資料，X為觀測數據，y為明天下雨結果

模型有用LogisticRegression、RandomForestClassifier、DecisionTreeClassifier、svm

from sklearn.linear\_model import LogisticRegression

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

import time

from imblearn.over\_sampling import SMOTE

t0=time.time()

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,test\_size=0.25)

sm = SMOTE(random\_state=12, ratio = 1.0)

X\_train\_res, y\_train\_res = sm.fit\_sample(X\_train, y\_train)

clf\_logreg = LogisticRegression(random\_state=0)

clf\_logreg.fit(X\_train\_res,y\_train\_res)

y\_pred = clf\_logreg.predict(X\_test)

score = accuracy\_score(y\_test,y\_pred)

print('Accuracy :',score)

print('Time taken :' , time.time()-t0)

------🡪把X和y皆拆成3/4的train和1/4的test，再做oversample的部分，輸出為X\_train\_res和y\_train\_res，再train\_res下去訓練(clf\_logreg.fit(X\_train\_res,y\_train\_res))，再用X\_test下去預測出y\_pred，在拿y\_pred跟實際上的y\_test作準確率比較。

發現採用LogisticRegression準確率較高，所以之後test.csv就用這個演算法來預測

df2 = pd.read\_csv('C:/Users/user/Desktop/data\_science/test.csv')

df2 = df2[['Attribute3','Attribute4','Attribute5','Attribute9','Attribute12','Attribute13','Attribute14','Attribute15','Attribute16','Attribute17','Attribute20','Attribute21','Attribute22']]

------🡪讀取test.csv為df2，並指取當初train.csv所選擇的attribute(上面的X)

df2['Attribute22'].replace({'No': 0, 'Yes': 1},inplace = True)

------🡪df2有選到attribute22，把yes or no轉為數字

df2\_pred = clf\_logreg.predict(df2)

------🡪用LR來預測做好處理的df2

df2\_pred = pd.DataFrame(df2\_pred, dtype = np.int)

output = pd.DataFrame(index = df2.index, columns = ['id', 'ans'], )

output['id'] = df2.index + 0.0

output['ans'] = df2\_pred

------🡪當時預測出來為float，把他轉成int。然後輸出所想要格式的output

filename = 'output.csv'

output.to\_csv(filename,index=False)

------🡪輸出output.csv

b.

numpy為處理多維陣列的套件，對於有多種天氣資料這種龐大資料庫是好用的套件。

pandas為專門處理資料的套件，像這次要讀入csv檔就必須需要他，pandas也可以快速瞭叫一個data frame的外觀，也可以幫我們整理資料，像是sort\_value()、dropna()，這些都可以調整我們所輸入的資料。

Logistic Regression藉由各種回歸來(像是線性回歸、簡單回歸、多項式回歸)，來預測出想要的結果。

imblearn中的SMOTE為要做oversample的套件，做了oversample可讓資料不要特別偏向某種結果，進而導致預測有瑕疵。