Data Science HW1

B10815057 廖聖郝

1. 執行程式:

   (1) pip install pandas

   (2) pip install sklearn

   (3) python judge_rain_tomorrow.py

2. 程式架構流程:

   (1) 讀取檔案，包括訓練與測試資料

```
train_data = pandas.read_csv("train.csv")
test_data = pandas.read_csv("test.csv")
```

   (2) 資料預處理(preprocess 函式內)

      i. 把不是數字的資料建立虛擬數值

```
labelencoder = preprocessing.LabelEncoder()
not_num = [8,10,11,22]
#建立虛擬數值
for i in not_num:
    data["Attribute" + str(i)] =
labelencoder.fit_transform(data["Attribute" + str(i)].fillna('0'))
```

      ii. 填補空缺資料

```
#填補空缺資料
for i in range(2,23):
    median = numpy.nanmedian(data["Attribute" + str(i)])
    newData = numpy.where(data["Attribute" +
str(i)].isnull(),median,data["Attribute" + str(i)])
    data["Attribute" + str(i)] = newData
```

iii. 日期欄位只取月份(因年份或幾號影響不大)

```
data['Attribute1'] = pandas.to_datetime(data['Attribute1'])
data['Attribute1'] = pandas.DatetimeIndex(data['Attribute1']).month
```

iv. 把有 NAN 的 row 都丟掉

```
data.dropna()
```

v. 將 Yes No 轉為 0 1，因測試資料無 Attribute23，所以要先判斷有沒有(hasattr)再做修改

```
data['Attribute22'] = labelencoder.fit_transform(data['Attribute22'] )
if hasattr(data,'Attribute23'):
    data['Attribute23'] = labelencoder.fit_transform(data['Attribute23'] )
```

vi. 風向欄位做 one hot encoding

```
data = pandas.get_dummies(data)
```

vii. 將訓練與測試資料都執行預處理

```
train_data = preprocess(train_data)
test_data = preprocess(test_data)
```

(3) 平衡資料(resample_data 函式內)

由於下雨跟不下雨的次數相差過大，用 resample 將資料平衡，然後把資料打亂，得到更好的效果

```
majority = data[data.Attribute23==0]
minority = data[data.Attribute23==1]


majority_down_sampled = resample(majority,replace=False,n_samples=3000,random_state=7414)
data = pandas.concat([majority_down_sampled,minority])
data = shuffle(data)
```

(4) 建立模型

分析模型採用神經網路模組: MLPClassifier(多層感知分類器)

```
    train_data_output = pandas.DataFrame(train_data["Attribute23"])
    train_data_input = train_data.drop(columns=["Attribute23"],axis=0)
    model = MLPClassifier(solver='adam', activation='logistic',alpha=
0.0001,learning_rate= 'adaptive' , hidden_layer_sizes=(50,50),
random_state=1,max_iter=1000,verbose=10,learning_rate_init=0.001)
    model.fit(train_data_input, train_data_output)
```

(5) 預測

```
    result = model.predict(test_data)
```

(6) 輸出結果到 csv 檔

```
    output = pandas.DataFrame(result)
    output.index = output.index.astype(float)
    output.to_csv("submit.csv",header=['ans'],index_label='id')
```