2019 ECT 作業四

一、請用 python 依照步驟對 Titanic.csv 進行前處理整理出 Weka 可執行 logistic 的資料，順便準備 python 的前處理(40%)：

請將各步驟程式碼截圖並簡單說明

首先，import 需要的套件與讀取 Titanic.csv 檔案。

```
import pandas as pd
import numpy as np
from sklearn import preprocessing
```

```
#讀取CSV檔案
data = pd.read_csv('Titanic.csv')
```

(a) 利用 np.nanmeduan 函數計算資料集中 Age 的中位數

```
mid = np.nanmedian(data['Age'])
mid
```

```
28.0
```

(b) 利用 np.where 將 dataframe 中空值的 Age 替換成上述的中位數

```
data['Age'] = np.where(pd.isnull(data['Age']), mid, data['Age'])
```

(c) 運用 drop 函數刪除 Cabin 欄位

(d) 運用 drop 函數刪除 PassengerId 欄位

```
data.drop(['Cabin', 'PassengerId'], axis=1, inplace=True)
```

(e) 運用 dropna()函數去掉資料的空值

```
data.dropna()
```

(f) 將 name 轉為 encoded 的資料

```
#轉換屬性型態
#將屬性轉為數字label
le = preprocessing.LabelEncoder()

#將 Name 轉為數字label
X_Name_encoded=le.fit_transform(data['Name'])
data['Name'] = X_Name_encoded
```

(g) 運用 replace 將原資料中 Survived 欄位中的[0,1]轉為 [No,Yes](Python 部分不用)

```
data['Survived'] = np.where(data['Survived']==0 , 'No', 'Yes')
```

(h) 運用 to_csv 另存為 TitanicClean.csv

```
data.to_csv('TitanicClean.csv', index = False)
```

Python 前處理

首先切分 input 和 output

```
#x:input
x=data.loc[:,['Pclass','Name','Sex','Age','SibSp','Parch','Ticket','Fare','Embarked']]
#y:output
y=data.loc[:,['Survived']]
```

(i) 除上述步驟外將 Name、Sex、Ticket、Embarked 進行 encoded

```
#將 Sex 轉為數字label
X_Sex_encoded=le.fit_transform(x.Sex)
#將 Ticket 轉為數字label
X_Ticket_encoded=le.fit_transform(x.Ticket)
#將 Embarked 轉為數字label
X_Embarked_encoded=le.fit_transform(x.Embarked.astype(str))
```

此時需註解 「運用 replace 將原資料中 Survived 欄位中的[0,1]轉為[No,Yes]」此(f)步驟的 code，並將所有 Cells 重新 Run 一遍，以避免之後在做 Logistic Regression 時讀取到字串型態而發生錯誤。

(j) 將 'Pclass','Name','Sex','Age','SibSp','Parch','Ticket','Fare','Embarked' 切為 feature

```
#將屬性合併
#變成list
feature=list(zip(x.Pclass,X_Name_encoded,X_Sex_encoded,x.Age,x.SibSp,x.Parch,X_Ticket_encoded,x.Fare,X_Embarked_encoded))

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

(k) 將 Survived 切為 Target

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(features, y['Survived'], test_size=0.34)
```

二、請用 python 對 Titanic.csv，Weka 對第一大題整理的 TitanicClean.csv 進行 logistic regression 的分類。

1. 請用 python 對 Titanic.csv 進行資料集 66%及訓練集 34%切分，並運用 linear_model.LogisticRegression (solver= 'liblinear ')進行訓練，並運用 LogisticRegression 函式庫中的 score 印出模型準確度。請將程式碼及準確度一同截圖。(30%)

i. python 對 Titanic.csv 進行資料集 66%及訓練集 34%切分

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(features, y['Survived'], test_size=0.34)
```

ii. 運用 linear_model.LogisticRegression (solver= 'liblinear ')進行訓練

```
from sklearn.linear_model import LogisticRegression
clf = LogisticRegression(random_state=0, solver='liblinear').fit(X_train, y_train)
clf.predict(X_test)
clf.predict_proba(X_test)
```

iii. 運用 LogisticRegression 函式庫中的 score 印出模型準確度。

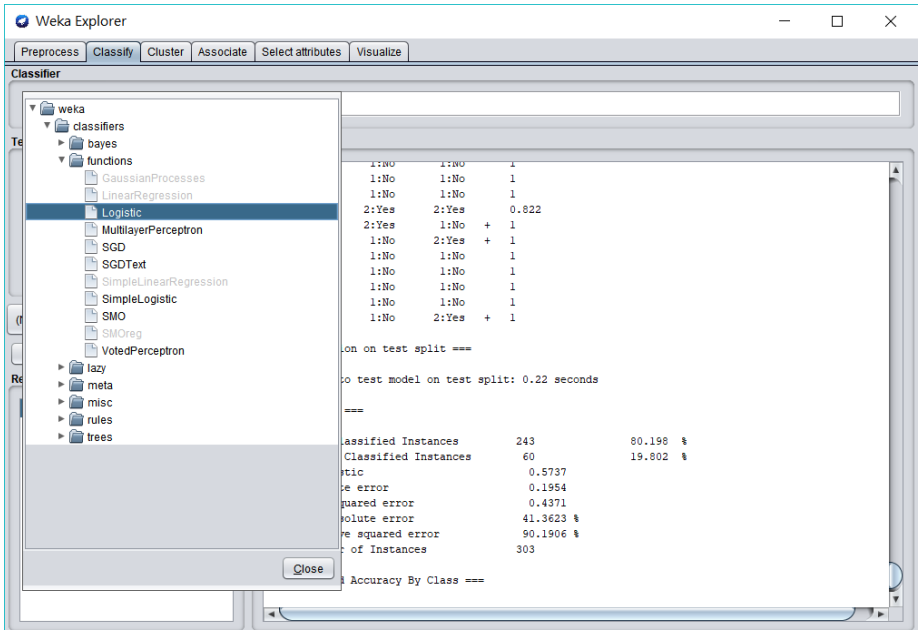
```
print('測試集分數： ',  clf.score(X_test, y_test))
print('訓練集分數： ',  clf.score(X_train, y_train))
```

```
測試集分數：  0.7920792079207921
訓練集分數：  0.8129251700680272
```

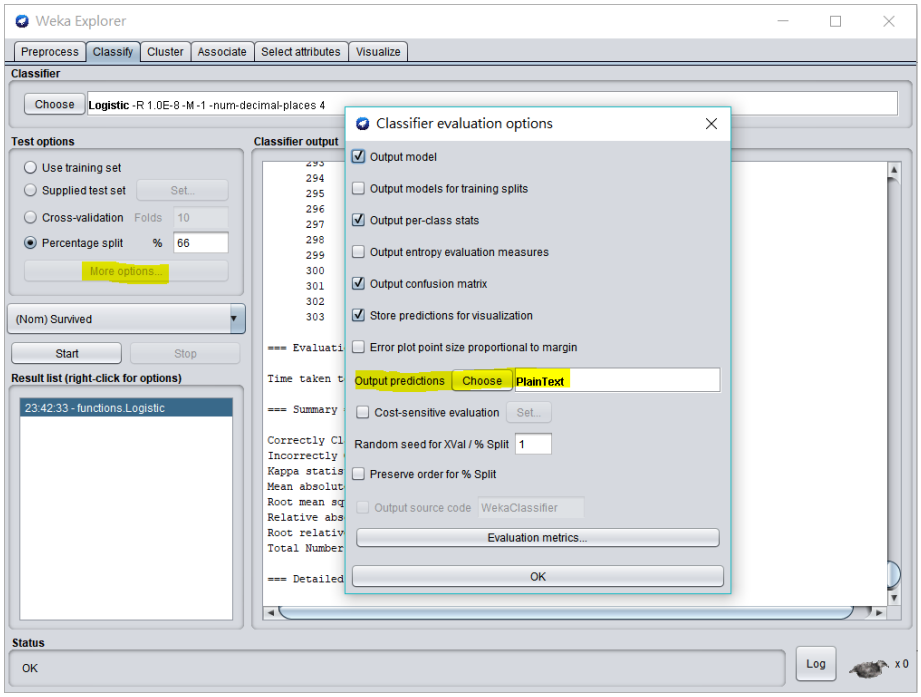
2. 請用 WEKA 對 TitanicClean.csv 同上等分切分資料集進行 LogisticRegression 訓練，同時在 Classifier output 中找出兩個與分類結果為 No 正相關的因素。 截圖並附上過程、答案及準確率。(30%)

i. 首先下載 jupyter 上的 TitanicClean.csv，並在 Weka 中開啟。

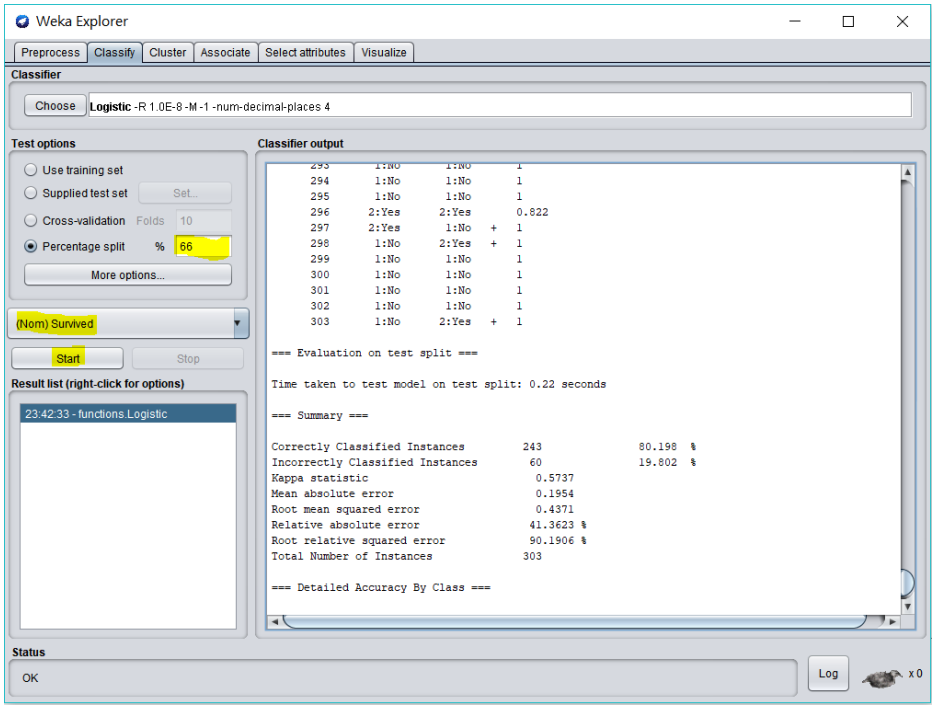
ii. 在 Classify 面板中，在 Classifier 選擇「weka / classifiers / functions / Logistics」。



iii. 在「More options」中的「output predictions」選擇「PlainText」。



iv. 在「Test options」中選取「Percentage split」，並設定為 66%；選擇預測「(Nom)Survived」，並點選「Start」。



v. 觀看 Classifier output 中的 Classifier model，在此會顯示各個 X(Variable)對預測 Y(Survived=No)的相關係數，只要從從右側的數值正負即可判斷為正相關或負相關。由圖可知與分類結果為 No 正相關因素包含「Pclass」、「Ticket=A/5 21171」。