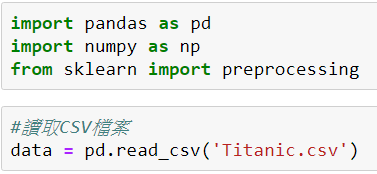
**2019 ECT 作業四**

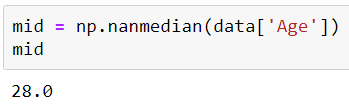
1. **請用 python 依照步驟對 Titanic.csv 進行前處理整理出 Weka 可執行 logistic 的資料，順便準備 python 的前處理(40%)：**

**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。



1. **利用 np.nanmeduan 函數計算資料集中 Age 的中位數**



1. **利用 np.where 將 dataframe 中空值的 Age 替換成上述的中位數**



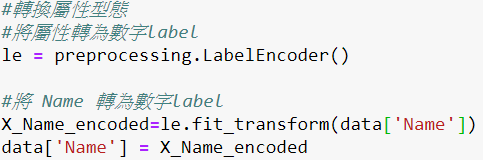
1. **運用 drop 函數刪除 Cabin 欄位**
2. **運用 drop 函數刪除 PassengerId 欄位**



1. **運用 dropna()函數去掉資料的空值**



1. **將 name 轉為 encoded 的資料**



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

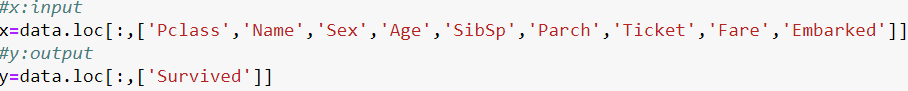


1. **運用 to\_csv 另存為 TitanicClean.csv**

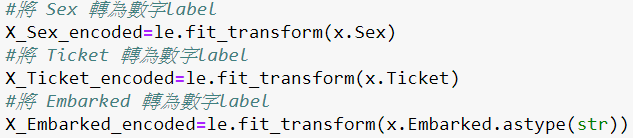


**Python 前處理**

首先切分input和output

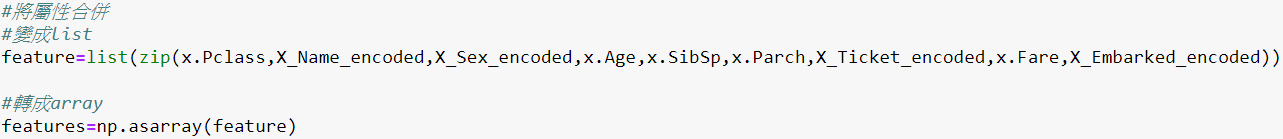


1. **除上述步驟外將 Name、Sex、Ticket、Embarked 進行 encoded**



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。

1. **將 'Pclass','Name','Sex','Age','SibSp','Parch','Ticket','Fare','Embarked' 切為 feature**



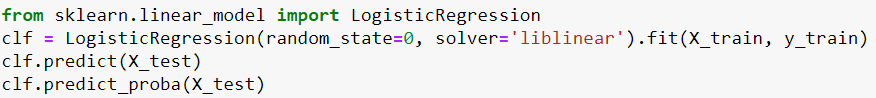
1. **將 Survived 切為 Target**



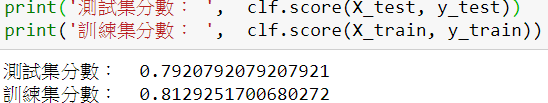
1. **請用 python 對 Titanic.csv，Weka 對第一大題整理的 TitanicClean.csv 進行 logistic regression 的分類。**
2. **請用 python 對 Titanic.csv 進行資料集 66%及訓練集 34%切分，並運用 linear\_model.LogisticRegression (solver=‘liblinear‘)進行訓練，並運用 LogisticRegression 函式庫中的 score 印出模型準確度。請將程式碼及準確度一同截圖。(30%)**
3. python 對 Titanic.csv 進行資料集 66%及訓練集 34%切分



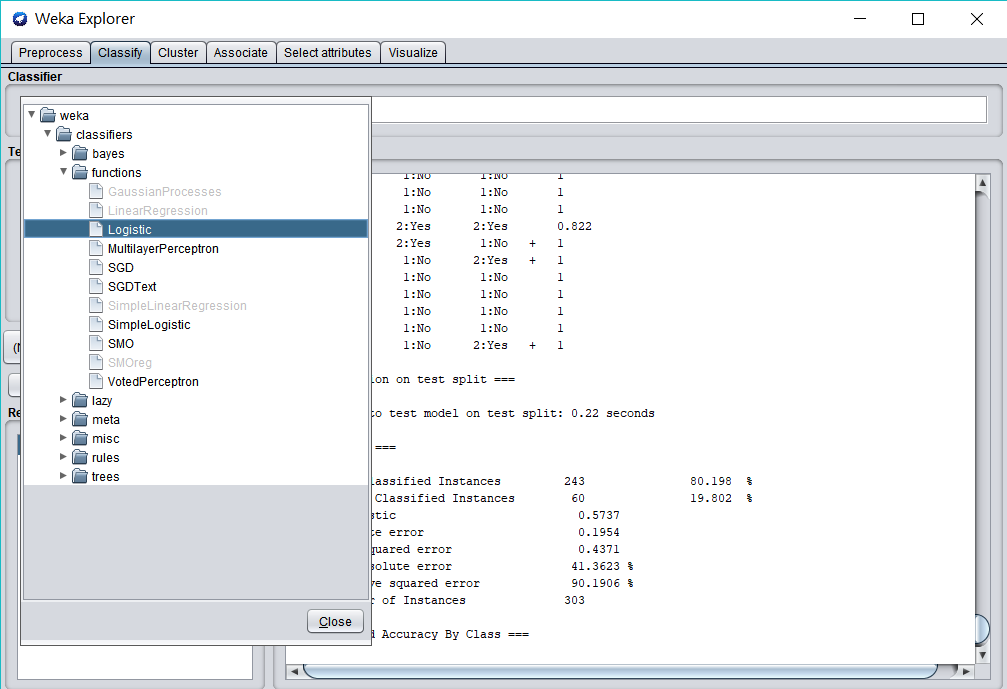
1. 運用 linear\_model.LogisticRegression (solver=‘liblinear‘)進行訓練



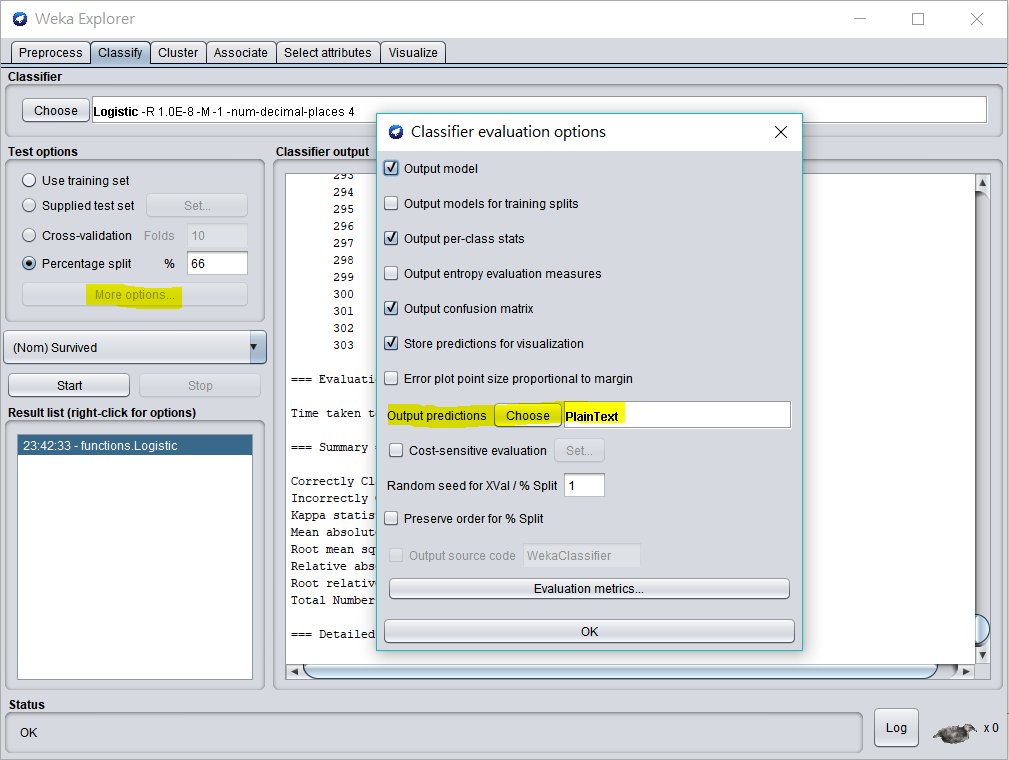
1. 運用 LogisticRegression 函式庫中的 score 印出模型準確度。



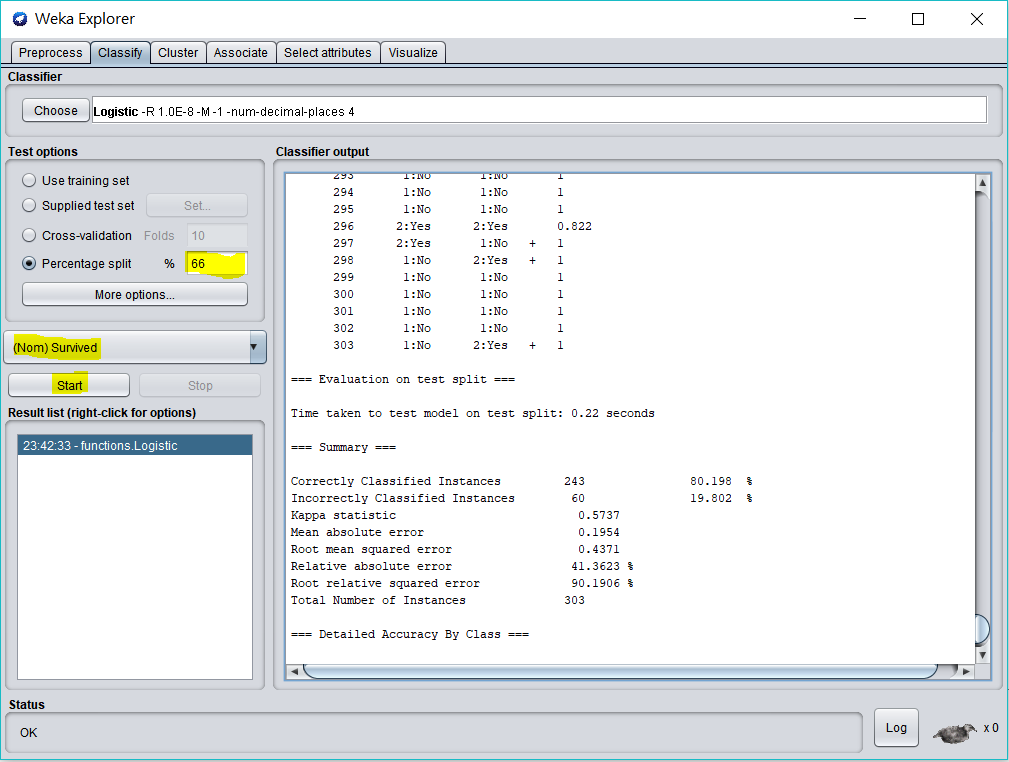
1. **請用 WEKA 對 TitanicClean.csv 同上等分切分資料集進行 LogisticRegression 訓練，同時在 Classifier output 中找出兩個與分類結果為 No 正相關的因素。 截圖並附上過程、答案及準確率。(30%)**
2. 首先下載jupyter上的TitanicClean.csv，並在Weka中開啟。
3. 在Classify面板中，在Classifier選擇「weka / classifiers / functions / Logistics」。



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



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



1. 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」。

