ECT_HW4

2019

第一大題 – 前處理

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

題目一(40%)

請將各步驟程式碼截圖並簡單說明

1. 利用np.nanmeduan函數計算資料集中Age的中位數
2. 利用np.where將dataframe中空值的Age替換成上述的中位數
3. 運用drop函數刪除Cabin欄位
4. 運用drop函數刪除PassengerId欄位
5. 運用dropna()函數去掉資料的空值
6. 將name轉為encoded的資料
7. 運用replace將原資料中Survived欄位中的0,1轉為No,Yes(Python部分不用)
8. 運用to_csv轉為TitanicClean.csv

Python

1. 除上述步驟外將Name、Sex、Ticket、Embarked進行encoded
2. 將'Pclass','Name','Sex','Age','SibSp','Parch','Ticket','Fare','Embarked’ 切為feature
3. 將Survived切為Target

題目一解答(40%)

WEKA處理

In [119]:
```
data = pd.read_csv('Titanic.csv')
```

In [120]:
```
age_median = np.nanmedian(data["Age"])
new_Age = np.where(data["Age"].isnull(), age_median, data["Age"])
data["Age"] = new_Age
data = data.drop("Cabin",axis = 1)
data = data.drop("PassengerId",axis = 1)
data = data.dropna()
```

In [121]:
```
new_Name=le.fit_transform(data.Name)
data.Name = new_Name
data.Survived.replace([0, 1],['No','Yes'],inplace = True)
data.to_csv("TitanicClean.csv",index = 0)
```

題目一解答(40%)

Python

In [122]:
```
#讀取CSV檔案
data = pd.read_csv('Titanic.csv')
```

In [123]:
```
data = data.drop("Cabin",axis = 1)
data = data.drop("PassengerId",axis = 1)
```

In [124]:
```
age_median = np.nanmedian(data["Age"])
new_Age = np.where(data["Age"].isnull(), age_median, data["Age"])
data["Age"] = new_Age

data = data.dropna()
```

In [125]:
```
#轉換屬性型態
#將屬性轉為數字label
le = preprocessing.LabelEncoder()

new_Name=le.fit_transform(data.Name)
new_Sex=le.fit_transform(data.Sex)
new_Ticket=le.fit_transform(data.Ticket)
new_Embarked = le.fit_transform(data.Embarked)

data.Name = new_Name
data.Sex = new_Sex
data.Ticket = new_Ticket
data.Embarked = new_Embarked
```

In [126]:
```
x=data.loc[:,['Pclass','Name','Sex','Age','SibSp','Parch','Ticket','Fare','Embarked']]
y=data.loc[:,['Survived']]
```

第二大題 – logistic regression

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

題目一(30%)

請用python對Titanic.csv隨意進行資料及訓練及切分，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

題目一(30%)

In [127]:
```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.34, random_state = 1)
```

In [128]:
```
logistic_regr = linear_model.LogisticRegression(solver='liblinear')
logistic_regr.fit(X_train, y_train.values.ravel())
```

Out[128]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='warn',
          n_jobs=None, penalty='l2', random_state=None, solver='liblinear',
          tol=0.0001, verbose=0, warm_start=False)
```

In [129]:
```
survived_predictions = logistic_regr.predict(X_test)
accuracy = logistic_regr.score(X_test, y_test.values.ravel())
print(accuracy)
```

```
0.8316831683168316
```

題目二(30%)

請用WEKA對TitanicClean.csv隨意切分資料集進行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。

題目二(30%)

```
=== Summary ===

Correctly Classified Instances         127               71.3483 %
Incorrectly Classified Instances        51               28.6517 %
Kappa statistic                          0.4204
Mean absolute error                      0.2873
Root mean squared error                  0.5354
Relative absolute error                 60.4259 %
Root relative squared error            109.2399 %
Total Number of Instances              178
```

```
=== Classifier model (full training set) ===

Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                                        Class
Variable                                   No
=============================================
Pclass                                5.2233
Name                                  0.0058
Sex=female                            -7.972
Age                                   0.1642
SibSp                                 1.5358
```