下載鐵達尼號旅客資料集

1.匯入下載所需模組

```
import  urllib.request
import  os
```

2.上傳鐵達尼號旅客資料集

```
from  google.colab  import  files
uploaded  =  files.upload()
```

選擇檔案 未選擇任何檔案 Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving 鐵達尼號資料集.xlsx to 鐵達尼號資料集 (1).xlsx

3.設定儲存檔案路徑

```
filepath="鐵達尼號資料集.xlsx"
```

使用Pandas dataframe讀取資料並進行欲處理

1.匯入所需模組

```
import  numpy
import  pandas  as  pd
```

2.讀取鐵達尼號資料

```
all_df  =  pd.read_excel(filepath)
```

3.查看前2筆資料

```
all_df[:2]
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292
				Wilkes,						

4.選取需要的欄位至dataframe

```
cols=['Survived','Name','Pclass','Sex','Age','SibSp','Parch','Fare','Embarked']
all_df  =  all_df[cols]
```

5.選取欄位後，顯示前2筆資料

```
all_df[:2]
```

	Survived	Name	Pclass	Sex	Age	SibSp	Parch	Fare
0	0	Kelly, Mr. James	3	male	34.5	0	0	7.8292
1	1	Wilkes, Mrs. James (Ellen Needs)	3	female	47.0	1	0	7.0000

使用Pandas dataframe進行資料預處理

1.將name欄位移除

```
df  =  all_df.drop(['Name'],axis=1)
```

2.找出那些欄位含有null值

```
all_df.isnull().sum()
```

```
Survived     0
Name         0
Pclass       0
Sex          0
Age         86
SibSp        0
Parch        0
Fare         1
Embarked     0
dtype: int64
```

3.age欄位null的資料填上平均值

```
age_mean  =  df['Age'].mean()
df['Age']  =  df['Age'].fillna(age_mean)
```

4.fare欄位null的資料填上平均值

```
fare_mean  =  df['Fare'].mean()
df['Fare']  =  df['Fare'].fillna(fare_mean)
```

5.轉換性別欄位為0與1

```
df['Sex']  =  df['Sex'].map({'female':  0  ,  'male':  1}).astype(int)
```

6.將Embarked欄位以Onehot Encoding轉換

```
x_OneHot_df  =  pd.get_dummies(data=df,columns=["Embarked"])
```

7.查看轉換後的data frame

```
x_OneHot_df[:2]
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked_C	Embarked_Q	Embarkec
0	0	3	1	34.5	0	0	7.8292	0	1	
1	1	3	0	47.0	1	0	7.0000	0	0	

將data frame轉換為array

1.dataframe 轉換為array轉換為array

```
ndarray  =  x_OneHot_df.values
```

2.查看ndarry 的shape

```
ndarray.shape
```

```
(418, 10)
```

3.查看ndarry的前2筆資料

```
ndarray[:2]
```

```
array([[ 0.    ,  3.    ,  1.    , 34.5   ,  0.    ,  0.    ,  7.8292,
         0.    ,  1.    ,  0.    ],
       [ 1.    ,  3.    ,  0.    , 47.    ,  1.    ,  0.    ,  7.    ,
         0.    ,  0.    ,  1.    ]])
```

4.擷取features與label擷取features與label

```
Label  =  ndarray[:,0]
Features  =  ndarray[:,1:]
```

5.查看前2筆labels標籤欄位

```
Label[:2]
```

```
array([0., 1.])
```

6.查看前2筆features特徵欄位

```
Features[:2]
```

```
array([[ 3.    ,  1.    , 34.5   ,  0.    ,  0.    ,  7.8292,  0.    ,
         1.    ,  0.    ],
       [ 3.    ,  0.    , 47.    ,  1.    ,  0.    ,  7.    ,  0.    ,
         0.    ,  1.    ]])
```

將ndarray特徵欄位進行標準化

1.匯入sklearn的資料預處理模組

```
from  sklearn  import  preprocessing
```

2.建立MinMaxScaler標準化刻度minmax_scale

```
minmax_scale  =  preprocessing.MinMaxScaler(feature_range=(0,1))
```

3.使用minmax_scale.fit_transform進行標準化

```
scaledFeatures  =  minmax_scale.fit_transform(Features)
```

4.查看標準化之後的特徵欄位前2筆資料

```
scaledFeatures[:2]
```

```
array([[1.        , 1.        , 0.4527232 , 0.        , 0.        ,
        0.01528158, 0.        , 1.        , 0.        ],
       [1.        , 0.        , 0.61756561, 0.125     , 0.        ,
        0.01366309, 0.        , 0.        , 1.        ]])
```

將資料分為訓練資料與測試資料

1.將資料以隨機方式分為訓練資料與測試資料

```
msk  =  numpy.random.rand(len(all_df))<  0.8        #依照8:2的比例使用numpy.random.rand產生msk
train_df  =  all_df[msk]                            #產生訓練資料，為全部資料的80%
test_df  =  all_df[~msk]                            #產生測試資料，為全部資料的20%
```

2.顯示訓練資料與測試資料筆數

```
print('total:',len(all_df),
      'train:',len(train_df),
      'test:',len(test_df))
```

```
total: 418 train: 334 test: 84
```

3.建立PreprocessData函數進行資料的預處理

```
def  PreprocessData(raw_df):
    df  =  raw_df.drop(['Name'],axis=1)
    age_mean  =  df['Age'].mean()
    df['Age']  =  df['Age'].fillna(age_mean)
    fare_mean  =  df['Fare'].mean()
    df['Fare']  =  df['Fare'].fillna(fare_mean)
    df['Sex']  =  df['Sex'].map({'female':0,'male':1}).astype(int)
    x_OneHot_df  =  pd.get_dummies(data=df,columns=["Embarked"])

    ndarray  =  x_OneHot_df.values
    Features  =  ndarray[:,1:]
    Label  =  ndarray[:,0]

    minmax_scale  =  preprocessing.MinMaxScaler(feature_range=(0,1))
    scaledFeatures  =  minmax_scale.fit_transform(Features)

    return  scaledFeatures,Label
```

4.將訓練資料與測試資料進行預處理

```
train_Features,train_Label=PreprocessData(train_df)
test_Features,test_Label=PreprocessData(test_df)
```

5.查看資料預處理後，訓練資料特徵欄位

```
train_Features[:2]
```

```
array([[1.        , 1.        , 0.44850498, 0.        , 0.        ,
        0.02976882, 0.        , 1.        , 0.        ],
       [1.        , 0.        , 0.61461794, 0.125     , 0.        ,
        0.02661597, 0.        , 0.        , 1.        ]])
```

6.查看資料預處理後，訓練資料標籤欄位

```
train_Label[:2]
```

```
array([0., 1.])
```