Overfitting

train\_x = df[['Sex', 'Age', 'Pclass', 'SibSp', 'Parch', 'Embarked']].copy()                   # 取出訓練資料需要分析的資料欄位

#

train\_y = df['Survived'].copy()                       # 取出訓練資料的答案

from sklearn.impute import SimpleImputer       # 匯入填補缺失值的工具

from sklearn.preprocessing import LabelEncoder # 匯入 Label Encoder

'''

imputer = SimpleImputer(strategy='median')     # 創造 imputer 並設定填補策略

age = train\_x['Age'].to\_numpy().reshape(-1, 1)

imputer.fit(age)                               # 根據資料學習需要填補的值

train\_x['Age'] = imputer.transform(age)        # 填補缺失值

'''

le = LabelEncoder()                            # 創造 Label Encoder

le.fit(train\_x['Sex'])                         # 給予每個類別一個數值

train\_x['Sex'] = le.transform(train\_x['Sex'])  # 轉換所有類別成為數值

le2 = LabelEncoder()

le.fit(train\_x['Embarked'])

train\_x['Embarked'] = le.transform(train\_x['Embarked'])

Pass81

# 資料分析與前處理

#, 'SibSp', 'Parch', 'Embarked'

train\_x = df[['Sex', 'Age', 'Pclass']].copy()                   # 取出訓練資料需要分析的資料欄位

#

train\_y = df['Survived'].copy()                       # 取出訓練資料的答案

from sklearn.impute import SimpleImputer       # 匯入填補缺失值的工具

from sklearn.preprocessing import LabelEncoder # 匯入 Label Encoder

'''

imputer = SimpleImputer(strategy='median')     # 創造 imputer 並設定填補策略

age = train\_x['Age'].to\_numpy().reshape(-1, 1)

imputer.fit(age)                               # 根據資料學習需要填補的值

train\_x['Age'] = imputer.transform(age)        # 填補缺失值

'''

le = LabelEncoder()                            # 創造 Label Encoder

le.fit(train\_x['Sex'])                         # 給予每個類別一個數值

train\_x['Sex'] = le.transform(train\_x['Sex'])  # 轉換所有類別成為數值

'''

le2 = LabelEncoder()

le.fit(train\_x['Embarked'])

train\_x['Embarked'] = le.transform(train\_x['Embarked'])'''

train\_x = train\_x.fillna(-1)