GAI project1 report 資訊 113 F74096027 吳瀚惟

練習 1：改善決策樹分類模型

本次練習中，透過採用不同的數據前處理方法、增加更多輸入特徵以及調整超參數，成功的提升了決策樹分類模型在測試集上的準確度。以下我將具體介紹改進的過程以及它們對模型性能的影響：

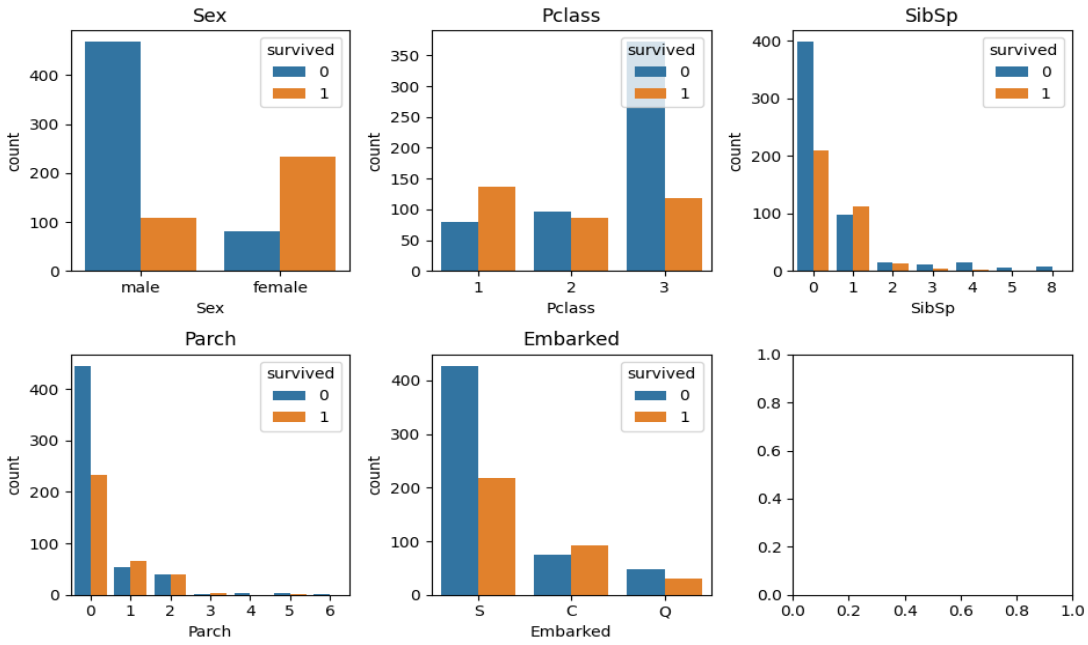
不同的數據前處理方法

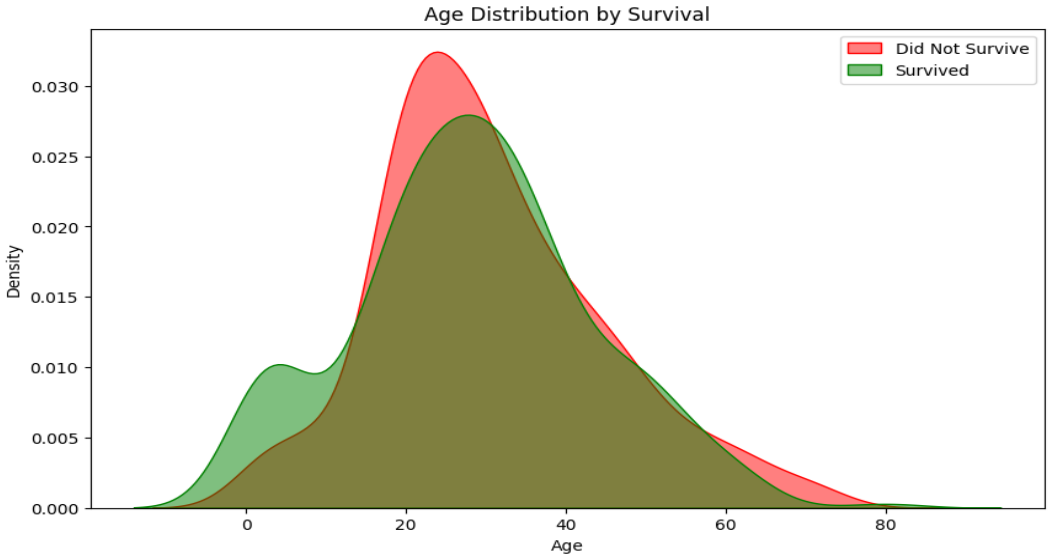
1. **數值型特徵處理：**選取了'Age', 'SibSp', 'Parch', 'Fare'作為數據型特徵，並透過**中位數**進行缺失值填補('Age'欄位)。之後使用了 **MinMaxScalar** 進行特徵縮放，將數據縮放到 0 至 1 之間，以消除不同量級特徵帶來的影響。
2. **類別型特徵處理：**選取'Sex', 'Embarked', 'Pclass'作為類別型特徵，並利用**眾數**進行缺失值填補('Embarked'欄位)，之後採取 **one-hot encoding** 將這些類別型特徵轉化為數值型，以便後續的模型處理。

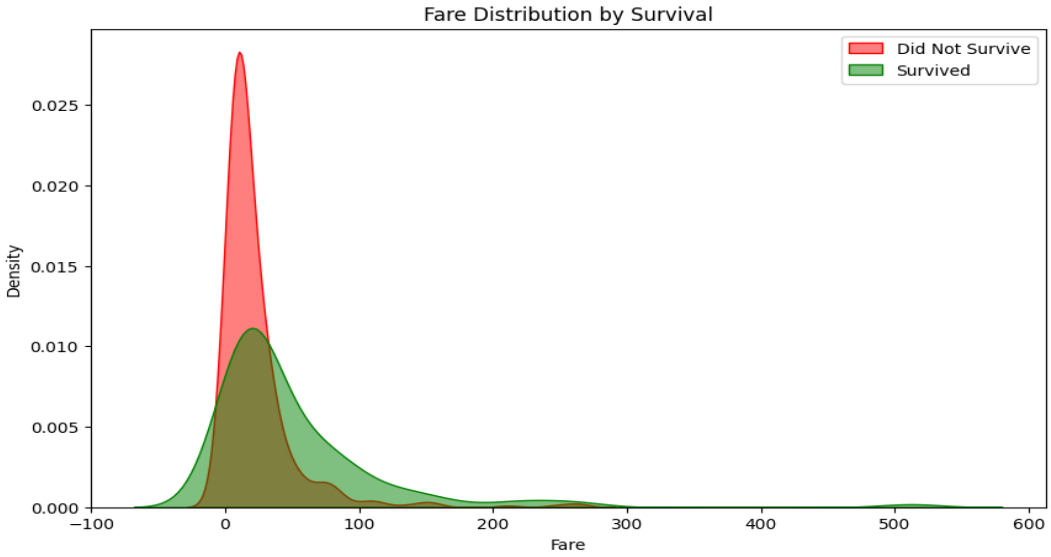
特徵選擇

繪製圖表初步觀察存活狀況，經過分析，我挑選出對預測生還機率有較大影響的特徵：'Pclass(艙等)', 'Sex', 'Age', 'SibSp(兄弟姊妹+老婆丈夫的數量)', 'Parch(父母子女的數量)', 'Fare(票價)', 和'Embarked(出發港口)'。此外，我捨棄了缺失值過多的'Cabin(房間號碼)'欄位，並捨棄掉了'PassengerId', 'Name' 以及'Ticket'欄位，因為這三者與看似與生還機率並沒有太大關聯。

[繪製圖表]


Sex
survived
0
1
count
male
female
Sex
Pclass
Pclass
SibSp
SibSp
Parch
Parch
Embarked
S
C
Q
Embarked


Age Distribution by Survival
Did Not Survive
Survived
Density
Age


Fare Distribution by Survival
Did Not Survive
Survived
Density
Fare

[初步觀察]

根據圖表數據初步推斷:

1. 女性存活率較高
2. 頭等艙乘客存活率較高
3. 與兄弟姊妹或老婆丈夫同行的乘客存活機率較高
4. 與父母或子女同行的乘客存活機率較高
5. 於C碼頭出發者存活機率較高
6. 未成年乘客存活機率較高
7. 票價較高的乘客存活機率較高

調整超參數

在建模階段，我利用 **Grid Search** 網格搜索對各種參數組合進行交叉驗證，找到了最佳超參數配置如下：

```
Best parameters found: {'decisiontreeclassifier__max_depth': 4,
'decisiontreeclassifier__min_samples_leaf': 2,
'decisiontreeclassifier__min_samples_split': 2}
```

套用此配置後得到了 **0.7988** 的 test accuracy，原先我以為這就是當下最好的準確率，然而我測出了一個特殊的現象：當我僅調整 DecisionTreeClassifier 的 **max_depth=3** 來控制模型的複雜度，竟然得到了 **0.8156** 的 test accuracy，只做這個簡單的設定，比利用 GridSearchCV 或是 RandomizedSearchCV 尋找更複雜的超參數設定提供了更良好的結果。上網查詢資料後，我才了解利用 Grid Search 並不是每次都能找出最佳超參數配置，原因可能是因為搜索的範圍不夠廣泛，或是僅針對這個數據集，簡單的決策樹反而更適合捕捉數據的趨勢，套用

別的測試集說不定就沒有辦法比 Grid Search 的結果來的高了。因此，我還是選擇先採用”利用 Grid Search 方法”找出的超參數配置而得出的 test accuracy(0.7988)，進行後續的分析。

以上的調整也都成功避免了 overfitting，同時維持足夠的學習能力，以對新數據做出準確的預測。此外，固定的 random_state 確保了模型訓練的可重複性。

模型性能評估

經過以上的改進，可以明顯觀察到 test accuracy 從原先的 **0.7262 提高到了 0.7988**，證明了上述前處理方法的有效性，特徵選擇的合理性以及超參數調整的必要性。

[Before]

```
train accuracy: 0.9831460674157303
test accuracy: 0.7262569832402235
```

[After]

```
training accuracy: 0.8426966292134831
test accuracy: 0.7988826815642458
```

[Special case]

```
train accuracy: 0.8314606741573034
test accuracy: 0.8156424581005587
```

練習 2：使用不同的模型

本次練習中，我一共套用了五個模型，分別是 GaussianNB（Naive Bayes Classifier）, SVC(Support Vector Machines), KNeighborsClassifier(Nearest Neighbors) , RandomForestClassifier（Ensemble），以及 GradientBoostingClassifier（Ensemble）。除了 GaussianNB 以外，針對其他四個模型，分別使用 Grid Search 搭配手動調整來優化超參數配置。其中，我觀察到 **Random Forest** 以及 **Gradient Boosting** 的準確率達到 **0.7989**，是我測試的五個模型中表現最好的。至於 GaussianNB，SVC 以及 KNeighborsClassifier 也都達到了 0.77 以上的準確率。總結來說，套用這五個 model，經過超參數優化，**他們的 test accuracy 皆有大於 0.7262**，打敗了改進前的決策樹分類模型的執行結果。

[長條圖比較]