The way to improve

說明對哪個部分進行改進，解釋方法與前後結果差異：

- **給予更多的資料欄位給模型訓練**
  - 邏輯上，越多資料就越好，但是是不是還是得看實驗結果
  - ```
    df_x = df[['Sex', 'Age', 'Fare', 'Pclass', 'Embarked']]
    ```
  - ```
    df_x = df[['Sex', 'Age', 'Fare', 'Pclass']]
    ```
  - 以上兩種訓練資料是我測試到結果最好的兩組，它們的最終結果是相同的
  - 若我所使用的是助教所給的例子：
  - ```
    df_x = df[['Sex', 'Age', 'Fare']]
    ```
  - 所有結果的比較的前提是在只假設該部分作爲變數的情況下，意思是說其他已經做好的部分我沒改，以上面的作爲例子，我只改了上面這個部分從四個、五個資料欄位改成三個，其他部分沒改動，只是想比較這個部分的影響：
  - 沒改進前的結果：train accuracy: 0.8196969696969697, test accuracy: 0.7818181818181819
  - 改進後的結果就是最後結果，放在這題的最後面
- **去除離群值(outliers)**
  - 因差別過大的值（離群值）可能會影響到整個結果，所以試著刪了那些數值
  - ```
    #Remove outliers
    q1, q3 = np.quantile(df_x['Age'], [0.25, 0.75])
    low = q1 - 1.5 * (q3 - q1)
    high = q3 + 1.5 * (q3 - q1)
    outlier_mask = (df_x['Age'] >= low) & (df_x['Age'] <= high)
    df_x = df_x[outlier_mask]
    df_y = df_y[outlier_mask]
    ```
  - 沒改進前的結果：train accuracy: 0.8426966292134831, test accuracy: 0.7932960893854749
- **調整超參數**
  - ```
    model = DecisionTreeClassifier(
    criterion='gini',
    #splitter='random',
    max_depth=4,
    max_leaf_nodes=12,
    min_samples_leaf=2, #2-5
    random_state=1012)
    ```
  - 經過不斷地試錯，所找到的最終最好的超參數
  - Criterion='gini'，其實可以不用寫，因爲 default 便是這個
  - 爲什麼我#掉 splitter='random'，是因爲雖然有這個超參數的結果是很不錯，不過我認爲沒有的結果會更好（有這行的結果：train accuracy: 0.8121212121212121, test accuracy: 0.8121212121212121，train accuracy 和 test accuracy 完全相同）
  - 至於 max_depth, max_leaf_nodes, min_samples_leaf 都是經過不斷地嘗試所得出的在我這個情況下最好的超參數
- **最終結果：** train accuracy: 0.843939393939394, test accuracy: 0.8303030303030303
    ```

Different model comparison

測試不同模型的訓練結果，至少兩種以上:

1) `from sklearn.naive_bayes import GaussianNB, MultinomialNB, BernoulliNB, CategoricalNB`

GaussianNB: train accuracy: 0.7863636363636364, test accuracy: 0.8242424242424242

MultinomialNB: train accuracy: 0.7772727272727272, test accuracy: 0.8181818181818182

BernoulliNB: train accuracy: 0.7848484848484848, test accuracy: 0.8181818181818182

CategoricalNB: train accuracy: 0.8196969696969697, test accuracy: 0.806060606060606

- 不知道爲什麼出現那麼多 test accuracy > train accuracy 的情況，可是這個模組并沒有多少超參數給你調，所以最好的情況也就是以上的情況了

2) `from sklearn.tree import ExtraTreeClassifier`

- 經過測試，最好的最終結果:
- train accuracy: 0.8318181818181818
- test accuracy: 0.8303030303030303

3) `from sklearn.ensemble import ExtraTreesClassifier`

- 經過測試，最好的最終結果:
- train accuracy: 0.9454545454545454
- test accuracy: 0.8363636363636363
- 這個結果 overfitting 了，可是這個是我測試到最高的 test accuracy 的結果了，也沒辦法降低 train accuracy 了，其他結果雖然 train accuracy 和 test accuracy 的數值是更接近，可是 test accuracy 沒有這個來得高，所以我最終選擇這個，不知道這個選擇會不會比較不好?

4) `from sklearn.ensemble import RandomForestClassifier`

- 經過測試，最好的最終結果:
- train accuracy: 0.8393939393939394
- test accuracy: 0.8363636363636363

5) `from sklearn.neighbors import KNeighborsClassifier`

- 經過測試，最好的最終結果:
- train accuracy: 0.8166666666666667
- test accuracy: 0.7636363636363637
- 結果沒有比用 sklearn.tree.DecisionTreeClassifier 來的好，可是還是有比 test acc 預測原始分數 0.7262 來得高