Please choose **two** datasets for Homework 2, and **please tell me why you selected these two dataset**.

我會選擇Dataset 1跟Dataset 2來做作業2，因為Dataset 3都是文字感覺處理起來很麻煩，Dataset 4則是有太多資料集，可能要花很多時間在做資料的前處理。

Dataset 1: maintenance_prediction.csv

Q1. How many unique device IDs are there in this dataset?

1169種裝置

```
device_unique_count = df['device'].nunique()
print(f'有{device_unique_count}種裝置')
```

✓ 0.0s

```
有1169種裝置
```

Q2. You are asked to do data analysis. What will you find?

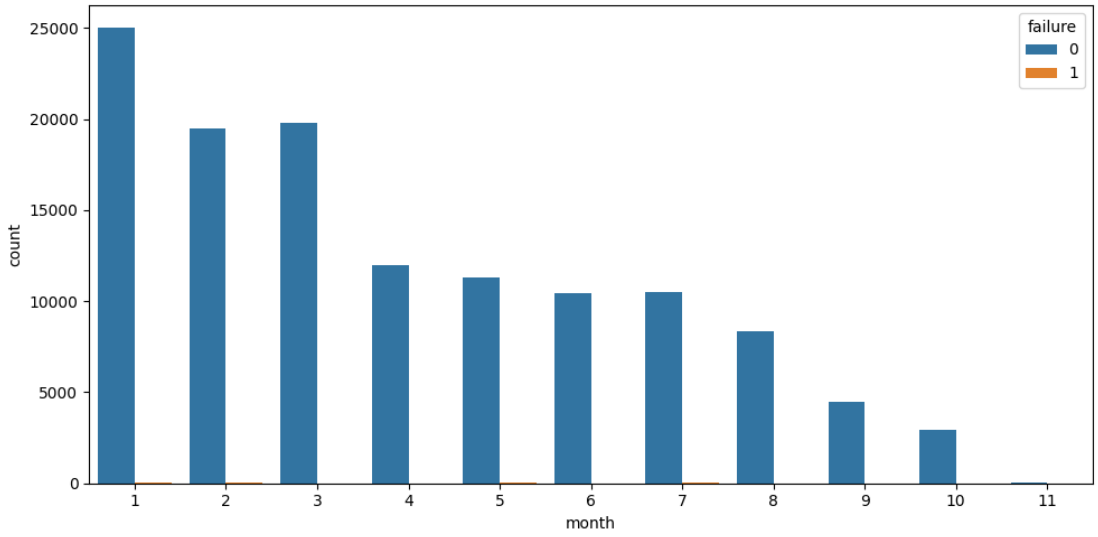
1. **這是個不平衡的數據集**

```
normal_device = df[df['failure'] == 0].value_counts().count()
abnormal_device = df[df['failure'] == 1].value_counts().count()
print(f'正常運作: {normal_device}\n發生故障: {abnormal_device}')
```

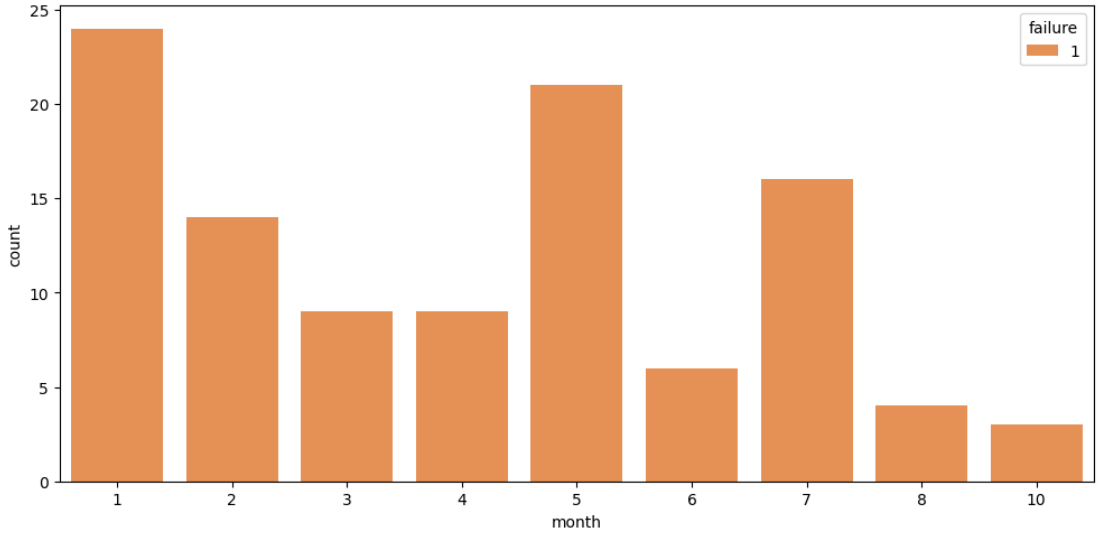
✓ 0.1s

```
正常運作: 124387
發生故障: 106
```

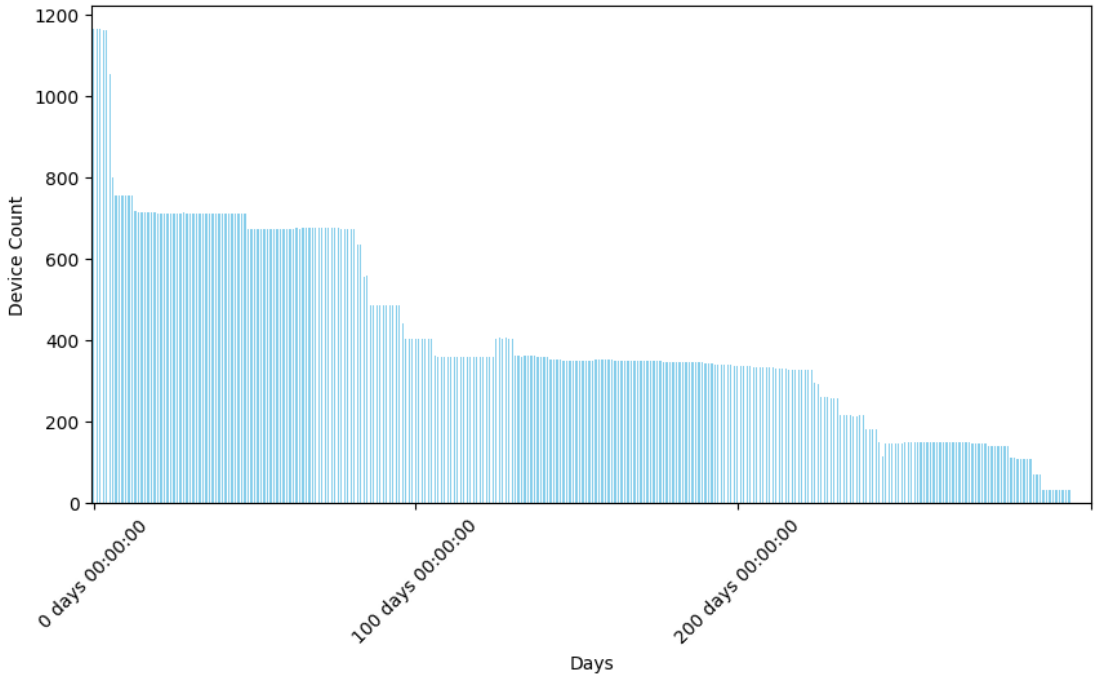
2. **每個月裝置正常與故障的數量**



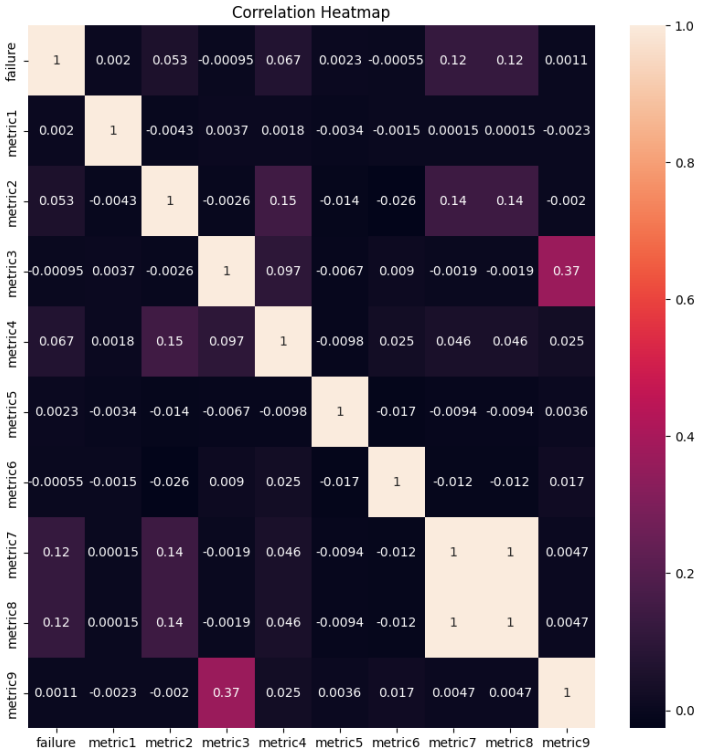
3. 有幾個月裝置故障的數量很高



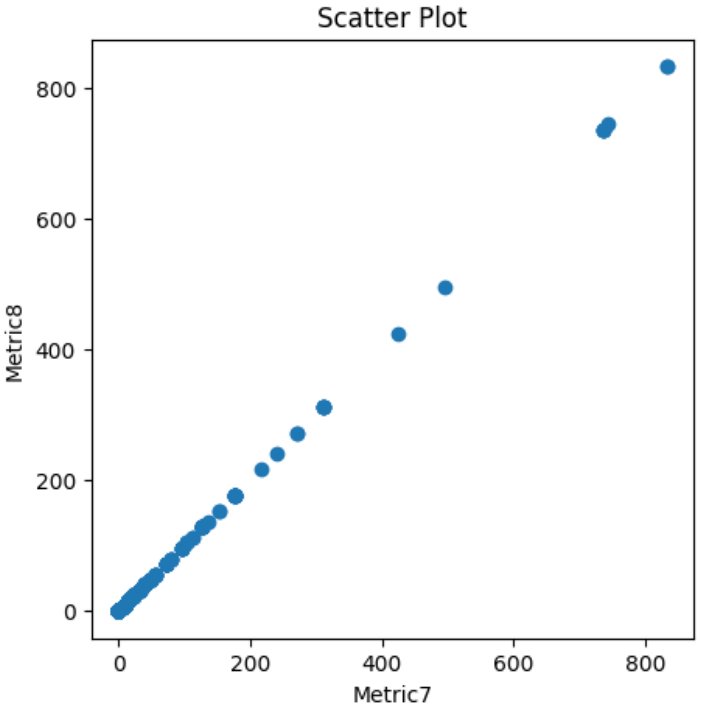
4. 從年初到年末運行的裝置越來越少



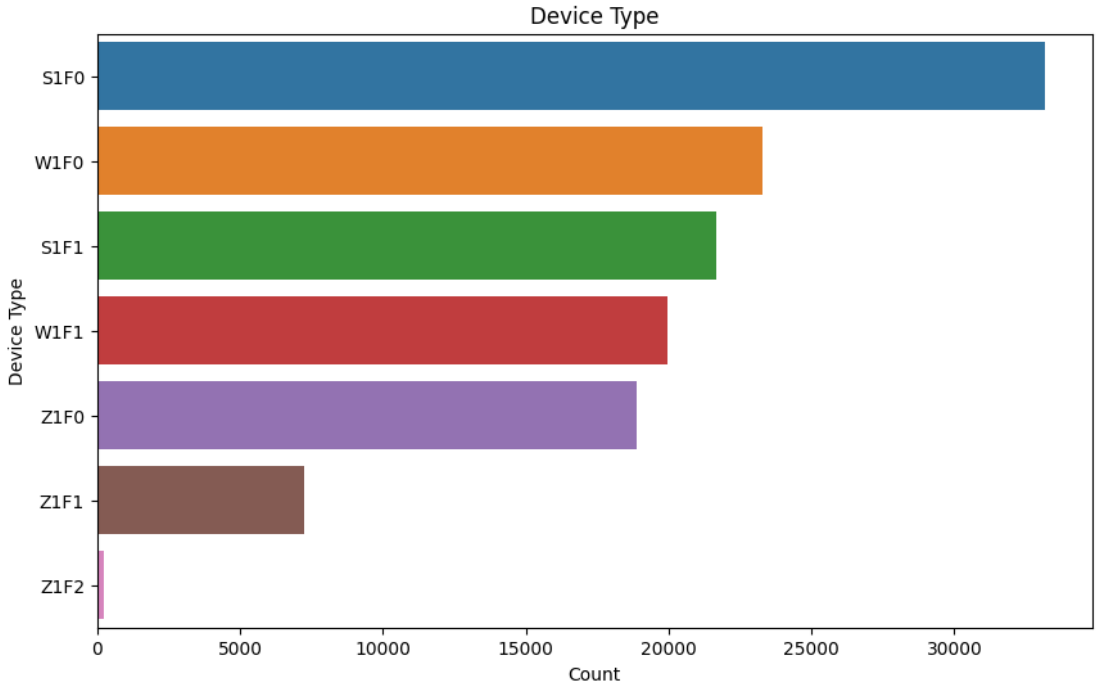
5. 繪製相關性熱圖 發現指標7跟指標8相關性很高



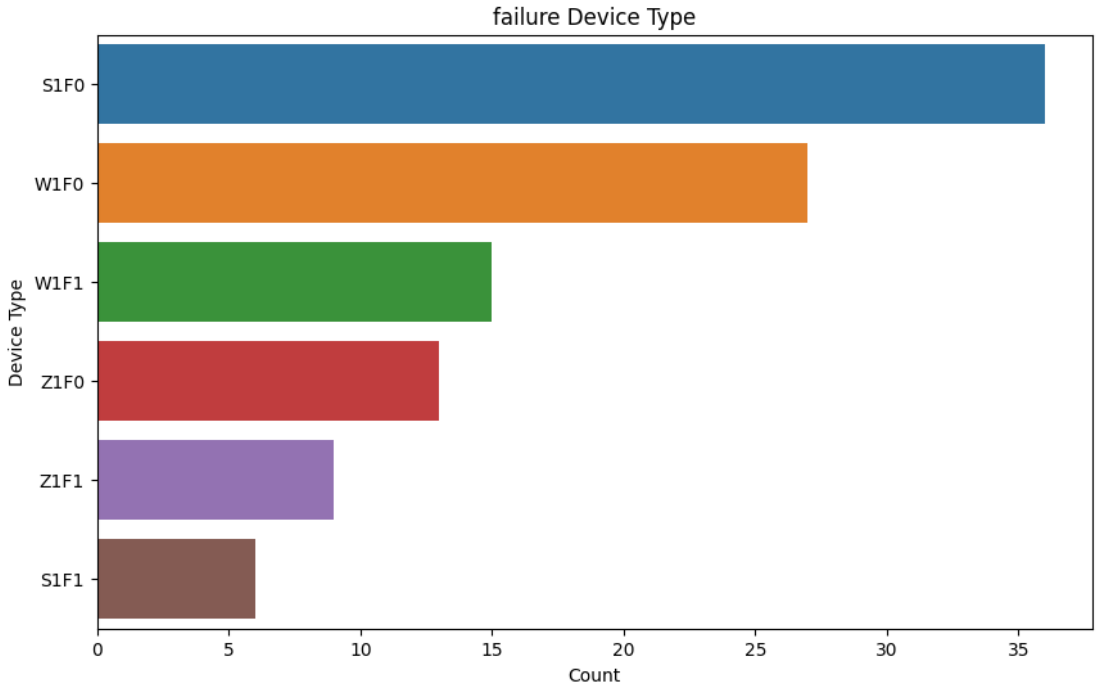
6. 繪製指標7(X軸)跟指標8(Y軸)的散點圖 看起來兩個指標有相同的數據



7. S1F0 開頭的裝置是最多的



8. S1F0 也是最多故障的裝置



Q3. You are asked to build a prediction model. Which kind of machine learning will be used and why? Supervised learning or Unsupervised learning? Regression, classification, or clustering? Which model will you use?

我會選擇監督式學習的分類模型，因為這是一項二分類任務，回歸跟非監督式學習分群等方式就不考慮使用。

我選擇**XGBoost**來預測資料集，以下是我選擇**XGBoost**的原因及過程:

1. 把上課教的八種分類模型都拿來跑暴力找最佳解。由於是個不平衡的資料集，找最佳解的過程，結果都偏向樣本數較多的類別，所以我用SMOTE把正負樣本平衡再重新找最佳解。**(SVM找最佳解的過程跑太久了，我只用原本不平衡的資料集+GridSearchCV跑SVM)**

```
from imblearn.over_sampling import SMOTE
from collections import Counter

smo = SMOTE(random_state=42)
X_smo, y_smo = smo.fit_resample(X, y)
print(Counter(y_smo))
```

✓ 0.0s

```
Counter({0: 124388, 1: 124388})
```

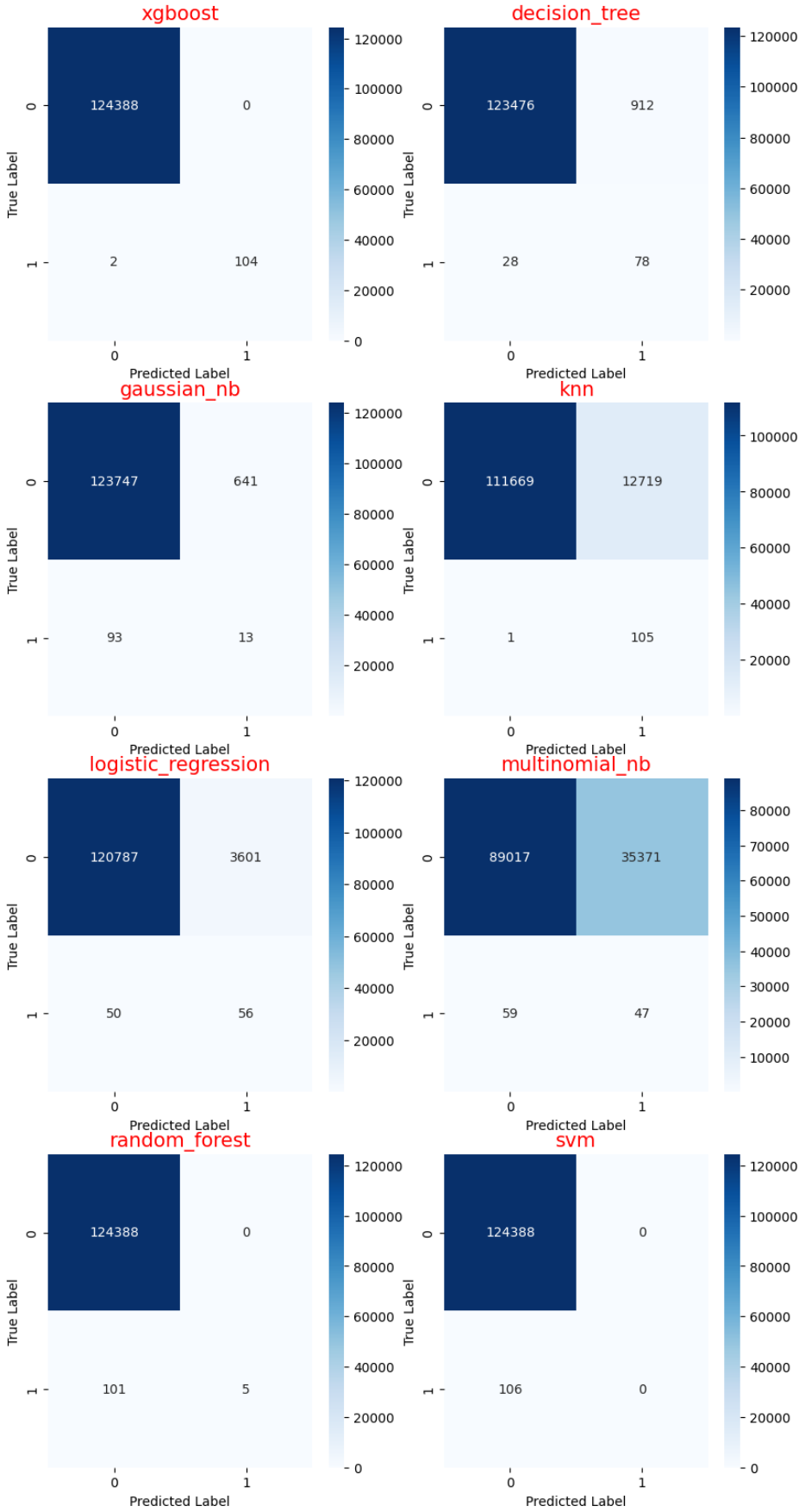
2. 除了精確率、召回率、F1分數外，多新增了一項指標 **ROC曲線**。以**AUC**作為標準來更新最佳參數。

   訓練下來表現最好的是**XGBoost(測試資料10%, n_estimators=200, max_depth=5, learning_rate=1, random_state=42)**

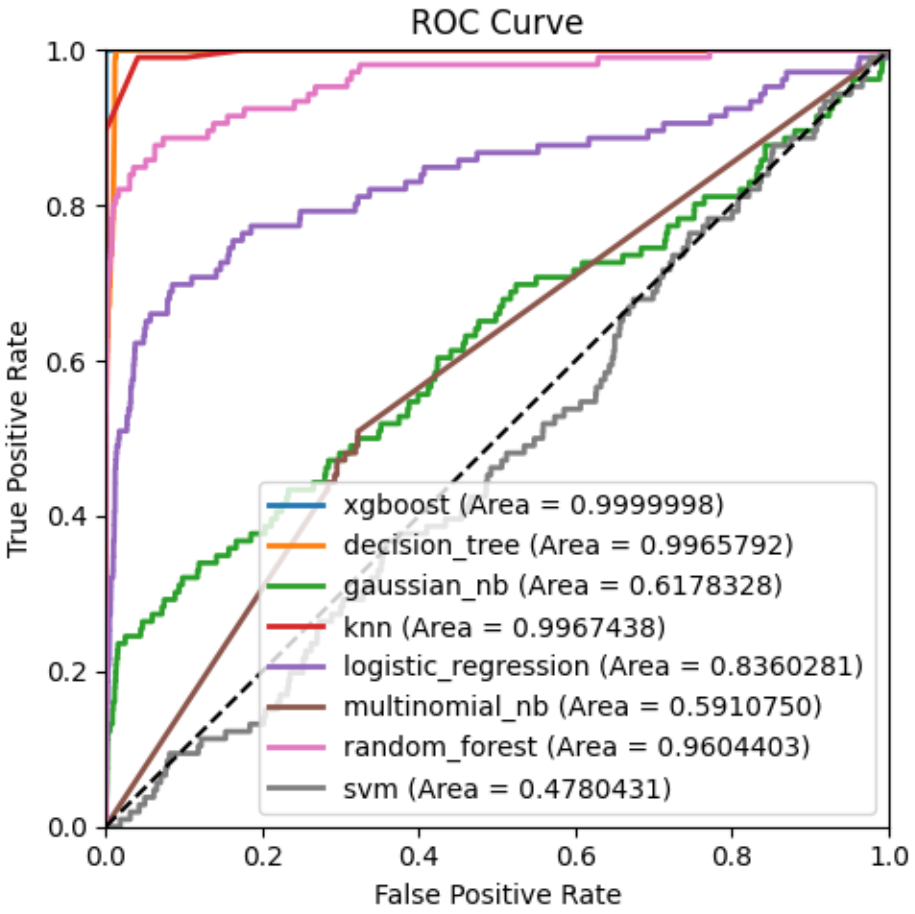
   每個跑出最佳解的模型在對原始資料集做預測，**XGBoost**的各項指標及混淆矩陣圖的表現是最好的。

	Model	Accuracy	F1 Score	Recall	Precision	ROC AUC
0	xgboost	0.999984	0.999984	0.999984	0.999984	1.000000
1	decision_tree	0.992449	0.995481	0.992449	0.998989	0.996579
2	gaussian_nb	0.994104	0.996223	0.994104	0.998415	0.617833
3	knn	0.897826	0.945323	0.897826	0.999147	0.996744
4	logistic_regression	0.970673	0.984298	0.970673	0.998748	0.836028
5	multinomial_nb	0.715408	0.833316	0.715408	0.998488	0.591075
6	random_forest	0.999189	0.998820	0.999189	0.999189	0.960440
7	svm	0.999149	0.998723	0.999149	0.998298	0.478043

3. 從混淆矩陣看XGBoost比起其他模型更能區分原資料集的正負例

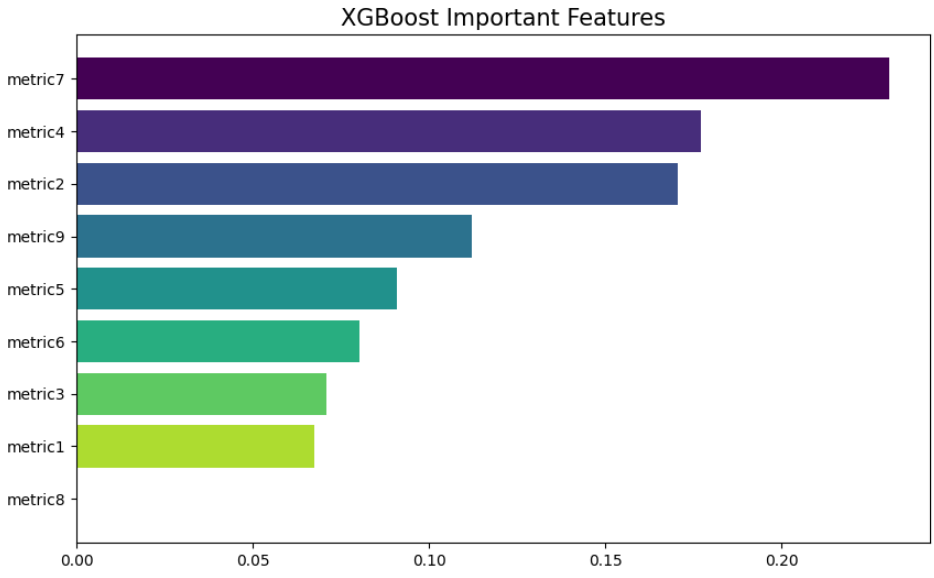


4. 八個最佳參數模型的ROC Curve, XGBoost最靠左上



Q4. Can you find the important features, informative features, or coefficient values? (Note: the answer will depend on your selected machine learning model.)

使用XGBoost的.feature_importances_找出重要特徵並排序



Q5. There are two data from two devices, please predict the corresponding failure values.

	metric								
device	1	2	3	4	5	6	7	8	9
AA	127175526	4109.434	3.90566	54.63208	15.46226	258303.5	30.62264	30.62264	23.08491
BB	4527376	0	0	0	3	24	0	0	0

表現最好的XGBoost模型預測兩個裝置為0

```
AA = best_xgb_model.predict([[127175526, 4109.434, 3.90566, 54.63208, 15.46226,
                              258303.5, 30.62264, 30.62264, 23.08491]])

BB = best_xgb_model.predict([[4527376, 0, 0, 0, 3, 24, 0, 0, 0]])

print('Device AA 預測為:', AA)

print('Device BB 預測為:', BB)
```

✓ 0.0s

```
Device AA 預測為: [0]
Device BB 預測為: [0]
```

Dataset 2: Insurance_dataset.csv and Insurance_validation.csv

Q1. Which kind of data selection method will you use to split csv data to training and testing datasets? sequential or random? WHY?

我會選擇**random**分割資料集的方式，因為接收到的資料集沒有時間序列的數據，就不用**sequential**分割來保持資料的時間連續性。

Q2. In class, we learned many model evaluation methods, such as confusion matrix, accuracy score, precision score, recall score, and so on. In addition to the confusion matrix and accuracy score, which must be used in the Q3 and Q4, if you were to choose two evaluation metrics/scores, which two would you choose? Why?

我會選擇用**confusion matrix**還有**ROC Curve**，因為在寫**dataset 1**時我發現**accuracy score, precision score, recall score, F1**分數雖然都很高，但在不平衡的數據集上結果會都偏向樣本較多的那一類，所以又找了**ROC Curve**這項二分類指標以它的面積來更新目前最佳模型參數設置，但是**ROC Curve**的面積越接近**1**也不代表能準確分類，所以最後還加入**confusion matrix**來一起評估模型的表現。

Q3. Please use eight classification models taught in the class and find their own best parameters' settings.

1. **Decision Tree**

```
best ratio of testing data: 30
best depth: 11
best min_sample_leaf: 9
Training score: 0.8420031720313552
Testing score: 0.8296371236122574
F1 score: 0.8284240801193259
Recall: 0.8296371236122574
Precision: 0.8397769637966541
ROC AUC: 0.8850614136821033
Confusion Matrix:
 [[14036  4814]
 [ 1585 17126]]
```

2. KNN

```
best ratio of testing data: 10
best neighbors: 9
best p: 1
Training score: 0.839981895794322
Testing score: 0.8147112850411309
F1 score: 0.8136729639041218
Recall: 0.8147112850411309
Precision: 0.8243198161209053
ROC AUC: 0.8711584997716304

confusion matrix:
 [[4684 1685]
 [ 635 5517]]
```

3. RandomForest

```
best ratio of testing data: 10
best no. of estimators: 100
best depth: 7
best min_sample_leaf: 3
Training score: 0.8313646488760306
Testing score: 0.8320421691558182
F1 score: 0.8298206380285986
Recall: 0.8320421691558182
Precision: 0.8500936866482824
ROC AUC: 0.8901613249890212

confusion matrix:
 [[4490 1763]
 [ 340 5928]]
```

4. XGBoost

```
best ratio of testing data: 20
best no. of estimators: 100
best depth: 2
best learning_rate: 1
Training score: 0.8397280378590469
Testing score: 0.8357094365241005
F1 score: 0.8346064543667611
Recall: 0.8357094365241005
Precision: 0.8459757025370953
ROC AUC: 0.8950133721642721

confusion matrix:
 [[ 9485  3126]
 [  988 11442]]
```

5. Logistic Regression Gaussian

```
best ratio of testing data: 90
best C: 0.1
best class weight: None
best solver: liblinear
Training score: 0.8118210862619808
Testing score: 0.8147175236506274
F1 score: 0.8091832373386998
Recall: 0.8147175236506274
Precision: 0.8557980292097587
ROC AUC: 0.8549500770366629

confusion matrix:
 [[36288 20014]
 [  864 55516]]
```

6. Gaussian Naive bayes

```
best ratio of testing data: 60
Training score: 0.8139376996805112
Testing score: 0.815087457735417
F1 score: 0.8097577620276823
Recall: 0.815087457735417
Precision: 0.855714733264389
ROC AUC: 0.8642359461228912

confusion matrix:
 [[24378 13285]
 [  606 36853]]
```

7. Multinomail Naive bayes

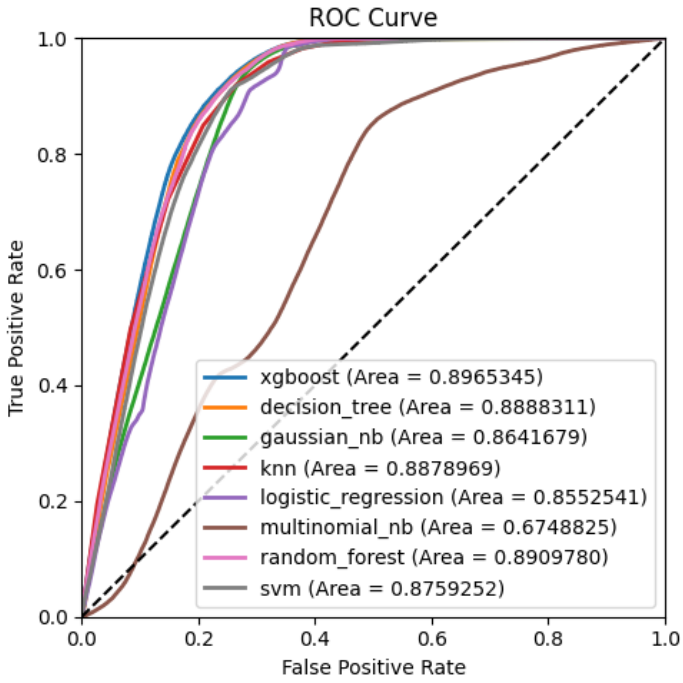
```
best ratio of testing data: 60
best alpha: 0.001
best fit_prior: True
best class_prior: [0.5, 0.5]
Training score: 0.5814097444089457
Testing score: 0.5860466973722744
F1 score: 0.5814772366029491
Recall: 0.5860466973722744
Precision: 0.5896317828768873
ROC AUC: 0.6763821170320119

confusion matrix:
 [[25988 11675]
 [19422 18037]]
```

8. SVM

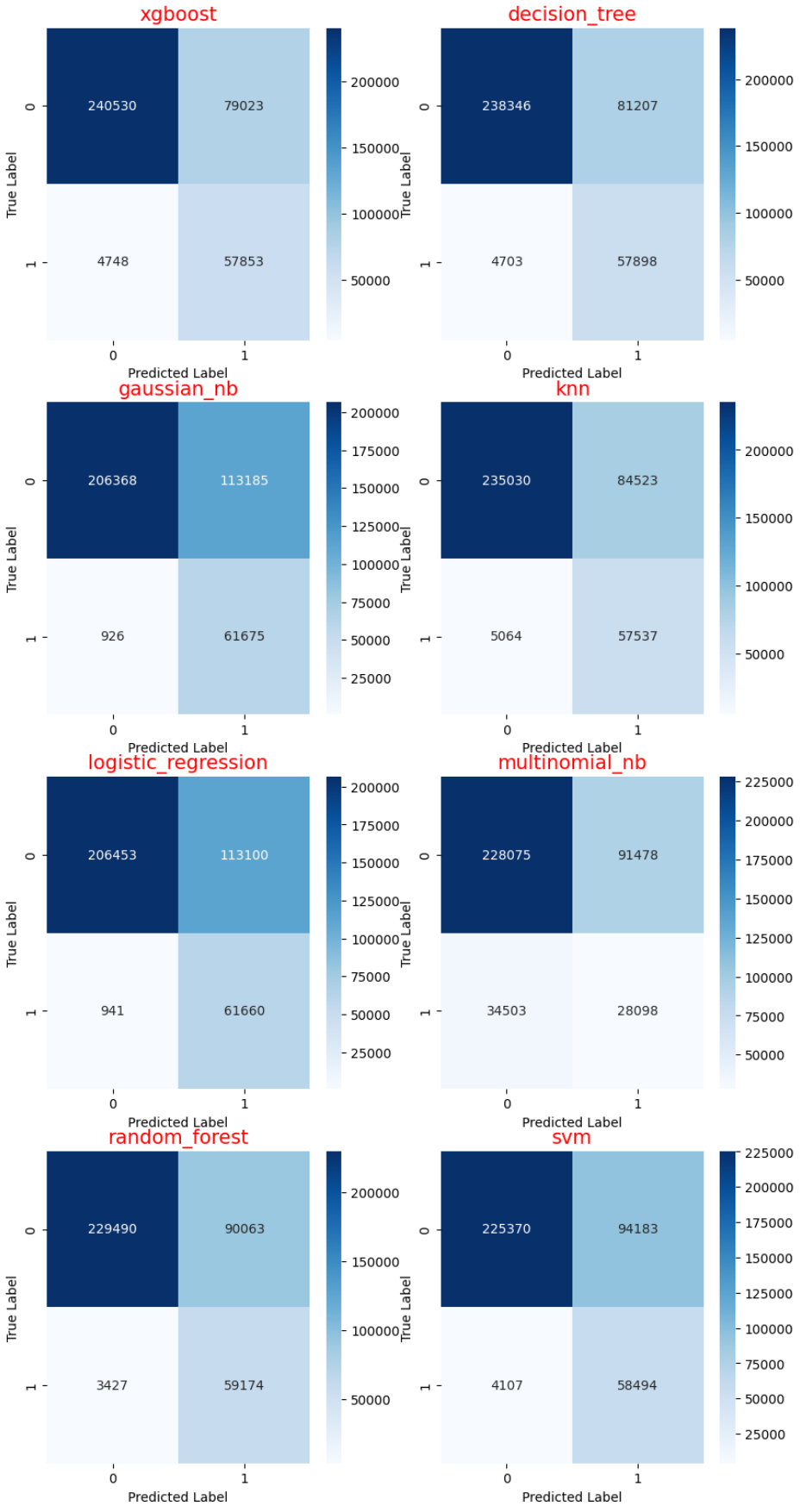
```
Best parameters found: {'gamma': 'scale', 'kernel': 'rbf'}
ROC AUC: 0.8763535620264856

confusion matrix:
 [[ 8945  3666]
 [  826 11604]]
```

Q4. Which prediction model is the best between these eight classification models? WHY?

XGBoost是表現最好的，因為它的ROC的AUC最接近1，而且在用各模型最佳參數對原始資料集預測時，XGBoost的混淆矩陣分類錯誤的數量較少



	Model	Accuracy	F1 Score	Recall	Precision	ROC AUC
0	xgboost	0.780793	0.807191	0.780793	0.889240	0.896535
1	decision_tree	0.775195	0.802543	0.775195	0.888190	0.888831
2	gaussian_nb	0.701400	0.740169	0.701400	0.890232	0.864168
3	knn	0.765574	0.794439	0.765574	0.884899	0.887897
4	logistic_regression	0.701584	0.740330	0.701584	0.890192	0.855254
5	multinomial_nb	0.670340	0.705757	0.670340	0.764805	0.674882
6	random_forest	0.755360	0.786204	0.755360	0.888839	0.890978
7	svm	0.742800	0.775510	0.742800	0.883983	0.875925


xgboost
240530
79023
4748
57853
True Label
Predicted Label
decision_tree
238346
81207
4703
57898
True Label
Predicted Label
gaussian_nb
206368
113185
926
61675
True Label
Predicted Label
knn
235030
84523
5064
57537
True Label
Predicted Label
logistic_regression
206453
113100
941
61660
True Label
Predicted Label
multinomial_nb
228075
91478
34503
28098
True Label
Predicted Label
random_forest
229490
90063
3427
59174
True Label
Predicted Label
svm
225370
94183
4107
58494
True Label
Predicted Label

Q5. Insurance_validation.csv is a validation dataset. Please use your best prediction model to get the "Response" and output the results to a csv file. The format of the csv file is the following sample.

顯示預測結果的前20個

	id	Response
0	57782	0
1	286811	1
2	117823	0
3	213992	0
4	324756	0
5	425764	0
6	2934	1
7	99098	1
8	120076	0
9	272687	0
10	267856	1
11	72816	0
12	333425	0
13	340300	1
14	109149	0
15	56564	0
16	327695	0
17	121415	1
18	135475	0
19	123111	0

```
import joblib

model = joblib.load('d2_best_xgboost_model.pkl')

validation_id = df_val['id']
validation_no_id = df_val.drop(columns=['id'])

predictions = model.predict(validation_no_id)

results = pd.DataFrame({'id': validation_id, 'Response': predictions})
results.to_csv('D2_val_predict.csv', index=False)

results.head(20)
```