

專題報告

# 合金生產品質預測



# 使用資料集

資料集來源：

Kaggle：Metal-Furnace

描述：

此資料集紀錄了金屬熔爐中冶金製程的28 個匿名因素  
(編號為f0至f27)與其產品的合金品質。

# 專題目標與流程

01

找出冶金製程中與產品品質高度相關的關鍵因子

02

選擇不同的機器學習模型訓練並評估結果

03

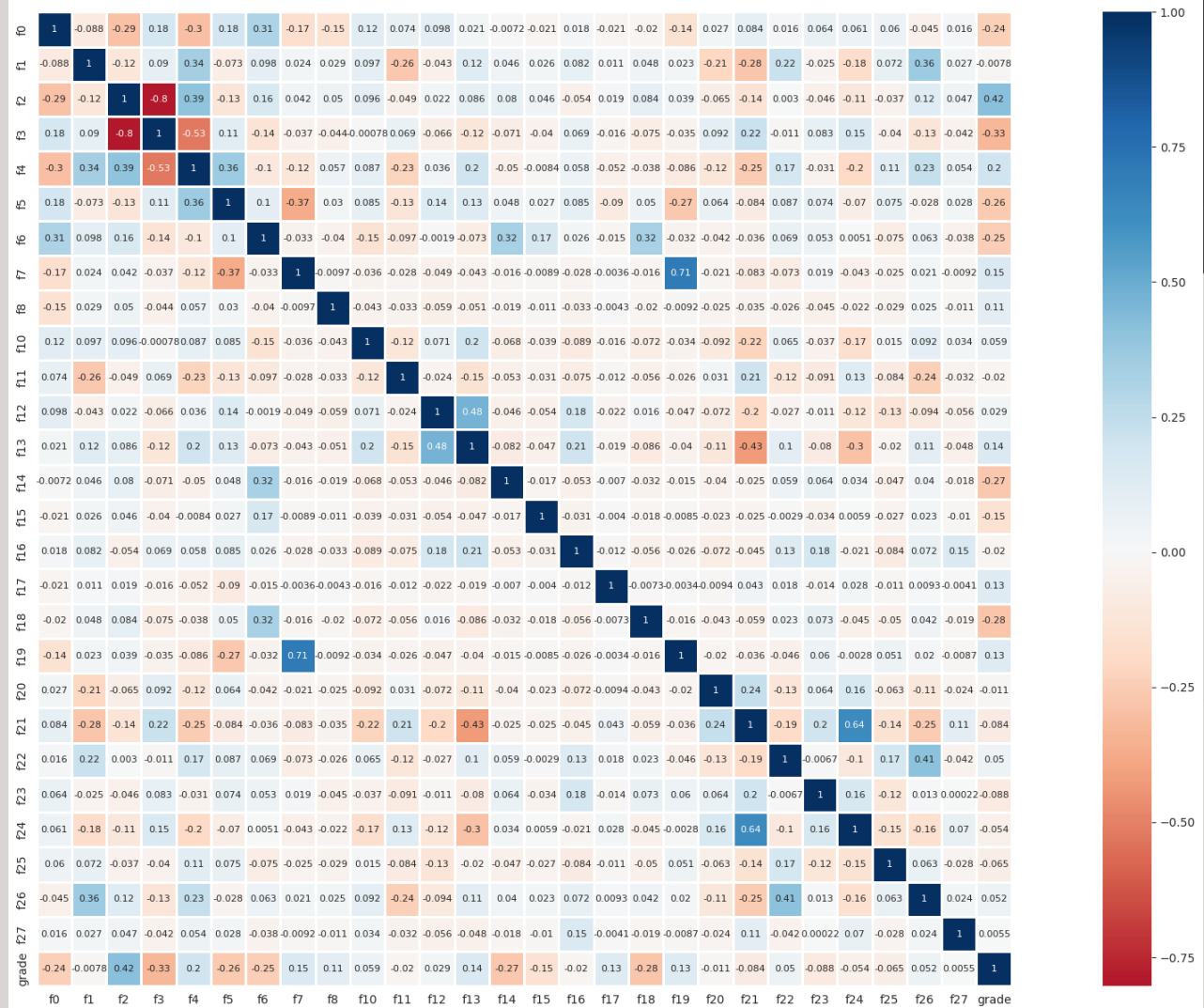
使用精準度較高的模型對新資料集進行預測

# 資料清洗

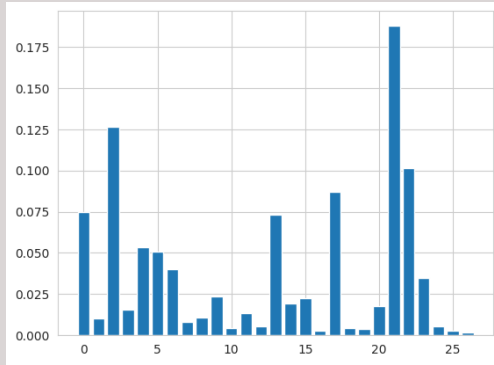
f9的欄位0值太多，選擇drop不放入模型訓練

```
[ ] 1 data = data.drop(['f9'], axis=1)
```

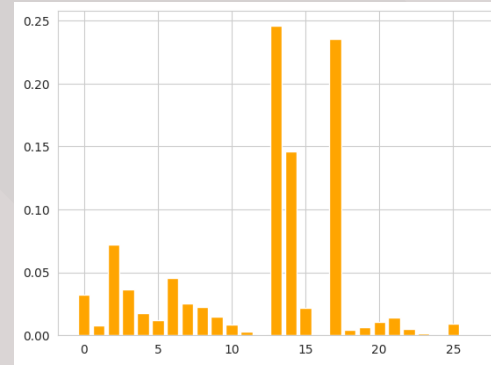
# Correlation Matrix



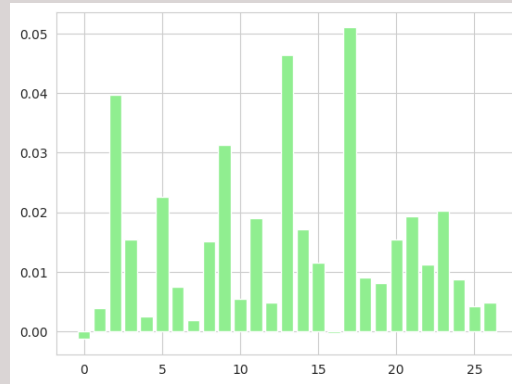
# Feature Importance



**RandomForestClassifier**



**XGBClassifier**



**KNeighborsClassifier**

# 與產品品質高度相關的因子

Correlation Matrix	f2
Feature Importance	f12 f13 f16

# 切割資料集與載入模型

```
[ ] 1 from sklearn.model_selection import train_test_split
    2 from sklearn.tree import DecisionTreeClassifier
    3 from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
    4 X = data.iloc[:, :-1]
    5 y = data.iloc[:, -1]
    6 train_X, test_X, train_y, test_y=train_test_split(X, y, test_size=0.2, random_state=17, shuffle=True, stratify=y)
```



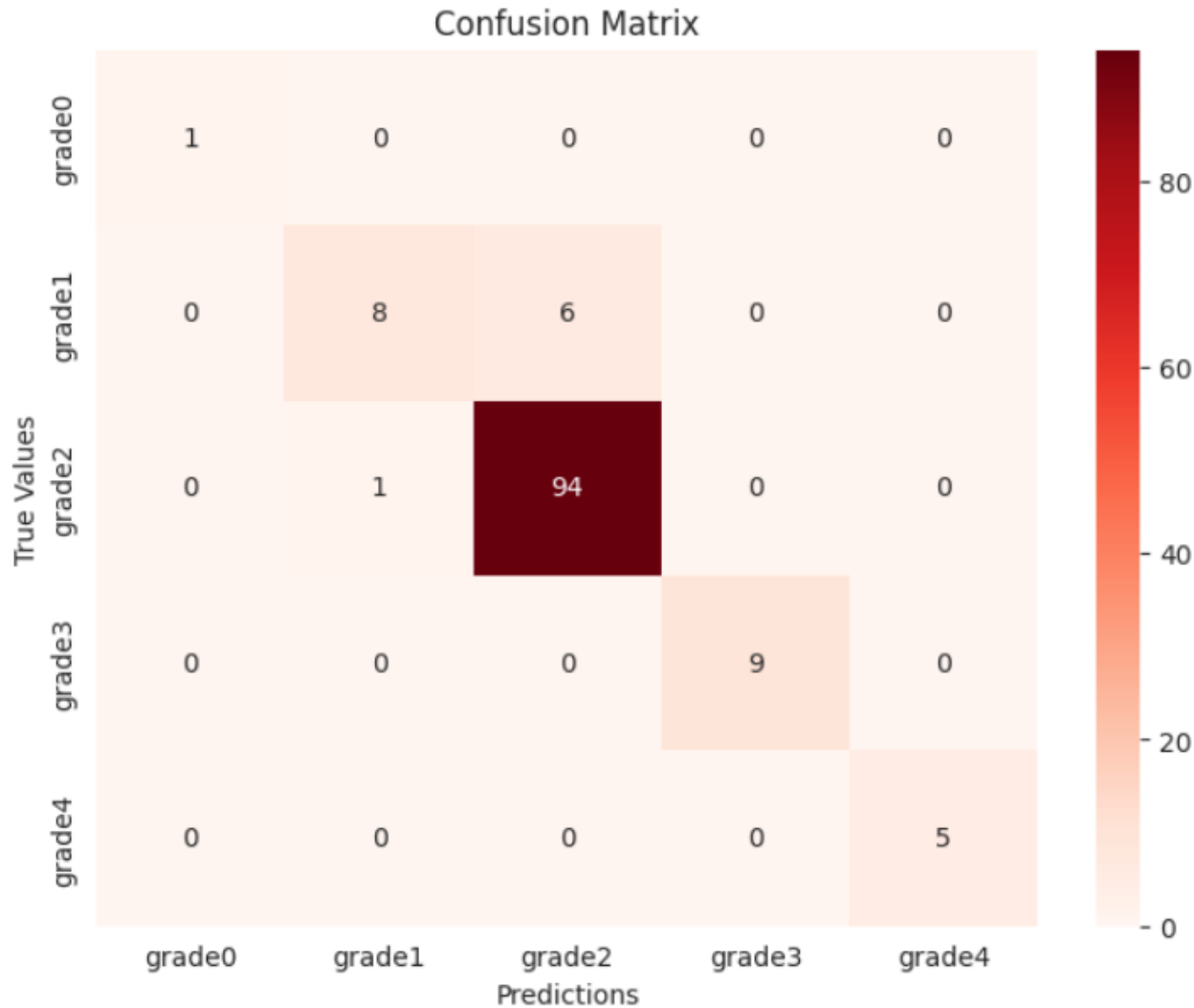
# 模型選擇

RandomForestClassifier

GradientBoostingClassifier

XGBoostClassifier

# Confusion Matrix



# 預測新資料集

## 載入驗證集進行預測

+ 程式碼

+ 文字

```
1 pre_data = pd.read_csv("/content/gdrive/My Drive/Test1.csv")  
2 pre_data
```



	f0	f1	f2	f3	f4	f5	f6	f7	f8	f9	...	f18	f19	f20	f21	f22	f23	f24	f25	f26	f27
0	-0.837812	-0.273636	1.276580	0.463262	-0.585142	-0.24287	0.349804	0.12356	0.166795	0.06143	...	0.197642	0.06143	0.27735	0.886135	-0.568935	1.100428	-0.244589	0.229718	-0.217109	0.087039
1	2.078087	-0.273636	-0.496119	0.463262	-2.438092	-0.24287	0.349804	0.12356	0.166795	0.06143	...	-5.059644	0.06143	0.27735	0.886135	0.504299	-0.434268	-0.244040	0.229718	-0.217109	0.087039
2	-0.837812	-0.273636	1.276580	0.463262	-0.585142	-0.24287	0.349804	0.12356	0.166795	0.06143	...	0.197642	0.06143	0.27735	-1.128496	-0.568935	-0.434268	-0.662763	0.229718	-0.217109	0.087039
3	-0.837812	-0.273636	-0.496119	0.463262	1.267808	-0.24287	-2.858743	0.12356	0.166795	0.06143	...	-5.059644	0.06143	0.27735	-1.128496	-0.449819	-1.918647	-0.662763	0.229718	-0.217109	0.087039
4	-0.837812	-0.273636	-0.496119	0.463262	-0.585142	-0.24287	-2.858743	0.12356	0.166795	0.06143	...	0.197642	0.06143	0.27735	-1.128496	-0.568935	-0.434268	-0.662763	0.229718	-0.217109	0.087039
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
261	0.411859	-0.273636	-0.496119	0.463262	-0.585142	-0.24287	0.349804	0.12356	0.166795	0.06143	...	0.197642	0.06143	0.27735	0.886135	-0.567744	-1.025755	-0.244040	0.229718	-0.217109	0.087039
262	-0.837812	-0.273636	-0.496119	0.463262	-0.585142	-0.24287	0.349804	0.12356	0.166795	0.06143	...	0.197642	0.06143	0.27735	-1.128496	2.410154	-0.434520	-0.662763	0.229718	-0.217109	0.087039
263	0.411859	-0.273636	-0.496119	0.463262	-0.585142	-0.24287	0.349804	0.12356	0.166795	0.06143	...	0.197642	0.06143	0.27735	0.886135	-0.448628	-0.434268	-0.662213	0.229718	-0.217109	0.087039
264	0.828416	4.569609	-0.496119	0.463262	1.267808	-0.24287	0.349804	0.12356	0.166795	0.06143	...	0.197642	0.06143	0.27735	0.886135	0.504299	1.352018	-0.244040	0.229718	-0.217109	0.087039
265	-0.837812	-0.273636	-0.496119	0.463262	-0.585142	-0.24287	0.349804	0.12356	0.166795	0.06143	...	0.197642	0.06143	0.27735	-1.128496	2.410154	1.352018	-0.662763	0.229718	-0.217109	0.087039

266 rows x 28 columns

```
[ ] 1 pre_data = pre_data.drop(['f9'],axis=1)
```

# 輸出預測結果

## ▼ 使用預測準確率最高的模型(RandomForestClassifier)進行預測

```
1 TestPredictions = rfcgrid.predict(pre_data)
2 PredictResult = {'grade': TestPredictions}
3 Result = pd.DataFrame(PredictResult)
4 Result
```



grade

0 2

1 3

2 2

3 3

4 2

...

261 2

262 2

263 1

264 2

265 2

266 rows x 1 columns

```
[ ] 1 Result.to_csv("/content/gdrive/My Drive/predict_result.csv")
```

		grade
1		
2	0	2
3	1	3
4	2	2
5	3	3
6	4	2
7	5	2
8	6	2
9	7	2
10	8	2
11	9	2
12	10	1
13	11	2
14	12	2
15	13	2
16	14	3
17	15	2
18	16	1
19	17	2
20	18	2
21	19	2
22	20	2
23	21	2
24	22	2
25	23	2
26	24	2
27	25	2
28	26	2
29	27	2
30	28	2
31	29	2



**Thank You For  
Watching!**