空氣品質氣體濃度
預測之研究

吳杰峰
M11123015
國立雲林科技大學資訊管理所

梁淯愷
M11123036
國立雲林科技大學資訊管理所

王敬翔
M11123051
國立雲林科技大學資訊管理所

劉昱辰
M11123057
國立雲林科技大學資訊管理所

摘要

隨著近年工業化及科技進步，空氣汙染對全球暖化與生物健康上產生危害，世界各國政府亦意識到此問題，如本次研究本是利用義大利政府空氣汙染所偵測之數據進行分析，除指定資料集 Adult 資料集外，自選題使用 Air Quality Data Set，本組利用多種演算法如 KNN、RandomForest、XGboost 等，分析的結果呈現本組認為 KNN 演算法能符合本組的預期。期盼透過此次的資料集分析與預測能夠給環保及政府單位，在做出空氣汙染相關政策或決策時，能夠當成有效的佐證數據。

關鍵字: 空氣汙染、資料分析、決策樹

Air Quality Gas Concentration Forecasting Research

WU, JIE-FENG
M11123015
Department of Information Management
National Yunlin University of Science and Technology

LIANG, YU-KAI
M11123036
Department of Information Management
National Yunlin University of Science and Technology

WANG, CHING-HSIANG
M11123051
Department of Information Management
National Yunlin University of Science and Technology

LIU, YU-CHEN
M11123057
Department of Information Management
National Yunlin University of Science and Technology

Abstract

With industrialization and technological progress in recent years, air pollution is harmful to global warming and biological health. Governments around the world are also aware of this problem. For example, this study uses the data detected by the Italian government to analyze air pollution. In addition to the designated data set, the Adult data set, the self-selected questions use the Air Quality Data Set. This group uses a variety of algorithms such as KNN, RandomForest, XGboost, etc. The analysis results show that the group believes that the KNN algorithm can meet the expectations of the group. It is hoped that the analysis and prediction of this data set can be used as effective supporting data for environmental protection and government agencies when making policies or decisions related to air pollution.

Keywords: air pollution, data analysis, decision tree

一、動機

隨著科技的進步，世界各國面臨的工業汙染與有毒氣體的破壞日益嚴重，其中，義大利政府有關部門為了探查有害氣體對於環境的破壞影響，以及有害氣體於空氣的濃度，在義大利國內街頭、道路、工廠旁設置多種類之氣體感應器，觀察各氣體數據的資料。因此，本組認為將環境氣體數據進行分析是有利於未來政府有關部門對於有害氣體做數據的統整、預測與分析，對於未來的氣體上，公司與政府所做出的控管，以及對有害氣體所做出的決策、政策等，皆提供有效的資料科學化的佐證。

二、目的

為了探討有害氣體生成濃度的預測，提供數據化的有害氣體濃度資料，便於政府做決策時的數據依據本次研究利用兩個資料集，除一個是指定的 Adult Data Set 資料集以外，另一個是自選的 Air Quality Data Set 資料集，該數據集包含來自空氣質量化學多傳感器設備中嵌入的 5 個金屬氧化物化學傳感器陣列的每小時平均響應的 9358 個實例。該設備位於意大利城市內道路高度污染嚴重區域的田野上。數據是從 2004 年 3 月到 2005 年 2 月（一年）記錄的，代表了現場部署的空氣質量化學傳感器設備響應的最長免費記錄。 Ground Truth CO、非金屬碳氫化合物、苯、總氮氧化物 (NOx) 和二氧化氮 (NO2) 的每小時平均濃度，由位於同一地點的參考認證分析儀提供。此數據集專門用於研究目的。完全排除商業目的。

三、方法

在第一筆實驗為 Adult 資料集，來預測測試資料集中 hours-per-week 之欄位值。第二筆實驗為空氣質量數據集。而我們在這兩項實驗中都使用了 KNN、RandomForest 和 XGBoost 這三種演算法來建構出數值的預測模型。這兩項研究的操作平台都是在 JupyterLab 上執行。

四、實驗

4.1 資料集

此實驗第一筆資料集為 Census Income ，此資料集共有 14 欄位[Age、Workclass、fnlwgt、Education、Education-num、Marital_Status、Occupation、Relationship、Race、Sex、Capital-gain、Capital-loss、hrs_per_week、Native-Country、Earning_potential]。

連續型格式有 ['Age', 'fnlwgt', 'Education-num', 'Capital-gain', 'Capital-loss', 'hrs_per_week']。

分類型格式有 ['Workclass', 'Education', 'Marital_Status', 'Occupation', 'Relationship', 'Race', 'Sex', 'Native-Country', 'Earning_potential']。

工作類：私人、自僱非公司、自僱公司、聯邦政府、地方政府、州政府、無薪、從未工作過。
教育：學士、大學、11、HS-grad、Prof-school、Assoc-acdm、Assoc-voc、9th、7-8、12、碩士、1-4、10、博士、5-6、學前班。
婚姻狀況：已婚-公民-配偶、離婚、未婚、分居、喪偶、已婚-配偶-缺席、已婚-AF-配偶。
職業：技術支持、工藝維修、其他服務，銷售、執行管理、專業教授、處理清潔工、機器操作檢查、行政人員、農業捕魚、運輸移動、私人住宅保護服務、武裝部隊。
關係：妻子、親子、丈夫、非親人、其他親屬、未婚。
種族：白人、亞洲太平洋島民、美洲印第安人-愛斯基摩人、黑人、其他種族。
性別：女性、男性。
居住國家：美國、柬埔寨、英國、波多黎各、加拿大、德國、美國邊遠地區（關島-美屬維爾京群島等）、印度、日本、希臘、南方、中國、古巴、伊朗、洪都拉斯、菲律賓、意大利、波蘭、牙買加 、越南、墨西哥、葡萄牙、愛爾蘭、法國、多米尼加共和國、老撾、厄瓜多爾、台灣、海地、哥倫比亞、匈牙利、危地馬拉、尼加拉瓜、蘇格蘭、泰國、南斯拉夫、薩爾瓦多、特立尼達和多巴哥、秘魯、香港、荷蘭-荷蘭。

	Age	Workclass	fnlwgt	Education	Education-num	Marital_Status	Occupation	Relationship	Race	Sex	Capital-gain	Capital-loss	hrs_per_week	Native-Country	Earning_potential
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	United-States	<=50K
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	United-States	<=50K
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	United-States	>50K
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40	United-States	>50K
4	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Female	0	0	30	United-States	<=50K

圖一、Adult 資料集

第二筆資料集為 Air Quality Data Set，此資料集共有 15 個欄位，分別為 [Date, Time, CO(GT), PT08.S1(CO), NMHC(GT), C6H6(GT), PT08.S2(NMHC), NOx(GT), PT08.S3(NOx), NO2(GT), PT08.S4(NO2), PT08.S5(O3), T, RH, AH]。
1. 日期 (DD/MM/YYYY)
2. 時間 (HH.MM.SS)
3. 以 mg/m^3 為單位的真實每小時平均濃度 CO（參考分析儀）
4. PT08.S1（氧化錫）每小時平均傳感器響應（標稱 CO 目標)
5. 以 microg/m^3 為單位的真實每小時平均整體非金屬碳氫化合物濃度(參考分析儀）
6. 以 microg/m^3 為單位的真實每小時平均苯濃度（參考分析儀）
7. PT08.S2（二氧化鈦）每小時平均傳感器響應（標稱 NMHC
8. 以 ppb 為單位的真實每小時平均 NOx 濃度（參考分析儀）

9. PT08.S3（氧化鎢）每小時平均傳感器響應（標稱 NOx 目標）
10. 以 microg/m^3 為單位的真實每小時平均 NO2 濃度（參考分析儀）
11. PT08.S4（氧化鎢）每小時平均傳感器響應（標稱 NO2 目標）
12. PT08.S5（氧化銦）每小時平均傳感器響應（標稱 O3 目標）
13. ˚C 溫度
14. 相對濕度 (%)
15. AH 絕對濕度
而這 15 個欄位有 14 個欄位為數值特徵，1 個欄位為類別特徵。

Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)	T	RH	AH
2004/3/10	18:00:00	2.6	1360	150	11.9	1046	166	1056	113	1692	1268	13.6	48.9	0.7578
2004/3/10	19:00:00	2	1292	112	9.4	955	103	1174	92	1559	972	13.3	47.7	0.7255
2004/3/10	20:00:00	2.2	1402	88	9.0	939	131	1140	114	1555	1074	11.9	54.0	0.7502
2004/3/10	21:00:00	2.2	1376	80	9.2	948	172	1092	122	1584	1203	11.0	60.0	0.7867
2004/3/10	22:00:00	1.6	1272	51	6.5	836	131	1205	116	1490	1110	11.2	59.6	0.7888
2004/3/10	23:00:00	1.2	1197	38	4.7	750	89	1337	96	1393	949	11.2	59.2	0.7848
2004/3/11	00:00:00	1.2	1185	31	3.6	690	62	1462	77	1333	733	11.3	56.8	0.7603
2004/3/11	01:00:00	1	1136	31	3.3	672	62	1453	76	1333	730	10.7	60.0	0.7702
2004/3/11	02:00:00	0.9	1094	24	2.3	609	45	1579	60	1276	620	10.7	59.7	0.7648
2004/3/11	03:00:00	0.6	1010	19	1.7	561	-200	1705	-200	1235	501	10.3	60.2	0.7517
2004/3/11	04:00:00	-200	1011	14	1.3	527	21	1818	34	1197	445	10.1	60.5	0.7465
2004/3/11	05:00:00	0.7	1066	8	1.1	512	16	1918	28	1182	422	11.0	56.2	0.7366
2004/3/11	06:00:00	0.7	1052	16	1.8	553	34	1738	48	1221	472	10.5	58.1	0.7353
2004/3/11	07:00:00	1.1	1144	29	3.2	667	98	1490	82	1339	730	10.2	59.6	0.7417
2004/3/11	08:00:00	2	1333	64	8.0	900	174	1136	112	1517	1102	10.8	57.4	0.7408
2004/3/11	09:00:00	2.2	1351	87	9.5	960	129	1079	101	1583	1028	10.5	60.6	0.7691
2004/3/11	10:00:00	1.7	1233	77	6.3	827	112	1218	98	1446	860	10.8	58.4	0.7552
2004/3/11	11:00:00	1.5	1179	43	5.0	762	95	1328	92	1362	671	10.5	57.9	0.7352
2004/3/11	12:00:00	1.6	1236	61	5.2	774	104	1301	95	1401	664	9.6	66.8	0.7951
2004/3/11	13:00:00	1.9	1286	63	7.3	869	148	1162	112	1537	799	8.3	76.4	0.8393
2004/3/11	14:00:00	2.9	1371	164	11.6	1034	207	983	128	1730	1037	8.0	81.1	0.8736
2004/3/11	15:00:00	2.2	1310	79	8.8	933	184	1082	126	1647	946	8.3	79.8	0.8778
2004/3/11	16:00:00	2.2	1292	95	8.3	912	193	1103	131	1591	957	9.7	71.2	0.8569
2004/3/11	17:00:00	2.9	1383	150	11.2	1020	243	1008	135	1719	1104	9.8	67.6	0.8185
2004/3/11	18:00:00	4.8	1581	307	20.6	1319	291	799	151	2083	1409	10.3	64.2	0.8066
2004/3/11	19:00:00	6.9	1776	481	27.4	1488	383	702	172	2333	1704	9.7	69.3	0.8319
2004/3/11	20:00:00	5.1	1640	401	24.0	1404	351	743	165	2191	1634	9.6	67.8	0.8133
2004/3/11	21:00:00	3.9	1313	197	12.8	1076	240	967	136	1707	1285	9.1	64.0	0.7419
2004/3/11	22:00:00	1.5	966	61	4.7	749	94	1325	86	1333	821	8.2	63.4	0.6905

圖二、Air Quality Data Set 資料集

4.2 前置處理

第一筆資料集此實驗所做的前置處理是把表格裡有出現 ”?“ 的都去除掉，並用 IQR 方式來處理這些箱型圖裡的異常值。使用最大最小值標準化(Min_Max_Scale)來針對資料進行線性轉換，等比例縮放資料，讓最小值為 0、最大值為 1。

而第二筆資料集所做的前置處理為把表格有出現 “?” 以及空白的表格都去除掉，並使用 IQR 方式來把資料集裡面的異常值給去除掉。

4.3 實驗設計

第一筆實驗先做預處理，再把 train_data 裡的連續型特徵欄位的箱型圖顯示出來，並算出各個欄位之間的信度，再把 Earning_potential 資料刪除掉，重複的動作用 test_data 再做一次。最後先用第一個模型 KNN，來找出最好的 K 值並輸入再找出最低錯誤率，並用 x_train 和 y_train 訓練出新模型並顯示新模型的 train, test 的準確率為 0.99, 0.99，用測試資料集測試出來的 Y 值和 X 值算出絕對平均誤差(MAE)為 0.0018，再用 MEA 算出均方根誤差(RMSE)為 0.0429，最後再用測試資料集測試出來的 Y 值和 X 值算出平均絕對百分比誤差(MAPE)為 7.5893。在用第二個 RandomForest 此研究設定的值為此決

策樹的個數為 20，至少要再有 15 筆資料才能再繼續做分類，某節點再劃分時的信息增益要大於 0.05 才能繼續劃分。接著算出用此決策樹所算出來的 train, test 的準確度為 0.46, 0.46。用 RandomForest 測試資料集預測出來的 Y 值和 X 值算出絕對平均誤差(MAE)為 7.2941，再用 MEA 算出均方根誤差(RMSE)為 2.7007，最後再用測試資料集測試出來的 Y 值和 X 值算出平均絕對百分比誤差(MAPE)為 0.3111。最後再用第三個演算法 XGboost 來做演算，此研究決策樹設定為決策樹的個數為 20 個，學習率(值越小，訓練越慢)為 0.3，而其他的設定都為預設值，再算出用此決策樹所算出來的 train, test 的準確率為 0.99, 0.96。最後用測試資料集測試出來的 Y 值和 X 值算出絕對平均誤差(MAE)為 0.0979，再用 MEA 算出均方根誤差(RMSE)為 0.3129，最後再用測試資料集測試出來的 Y 值和 X 值算出平均絕對百分比誤差(MAPE)為 0.0031。

對第二筆實驗做完預處裡後，算出各個欄位之間的信度關係。接下來使用三個演算法個別是 KNN、RandomForest 和 Xgboost，並算出他們 MSE、MEA 和 MAPE 這三種績效。在 KNN 演算法裡，先訓練出適合的模型，並對測試資料集做測試，在用訓練資料集的 Y 值和用 KNN 所訓練出來的 y 值帶入這三個函式 mean_squared_error 、 mean_absolute_error 、 Mean_absolute_percentange_error 算出 MSE、MEA 和 MAPS 的績效，最後用 cross_val_score 來驗證評分資料準確度。在 RandomForest 演算法裡，而此次研究的 RandomForest 設定為決策樹的各數為 10，其餘為預設值。接著用測試數據上使用森林的預測方法來算出 MSE、MEA 和 MAPS 和了解每個欄位從 RandomForest 模型中獲取的特徵重要性。最後在 Xgboost 演算法裡，先用 XGBRegressor 建立新的模型並使用訓練資料集來訓練此模型和對此模型做預測，並算出他們的 MSE、MEA 和 MAPS 和了解他們的特徵重要性，接這最後再用最初的測試和訓練資料集來用 XGboost 模型做預測並並算出他們的 MSE、MEA 和 MAPS 和了解他們的特徵重要性，最後發現用不同的 X 測試資料集和 Y 測試資料集結果是不同的

4.4 實驗結果

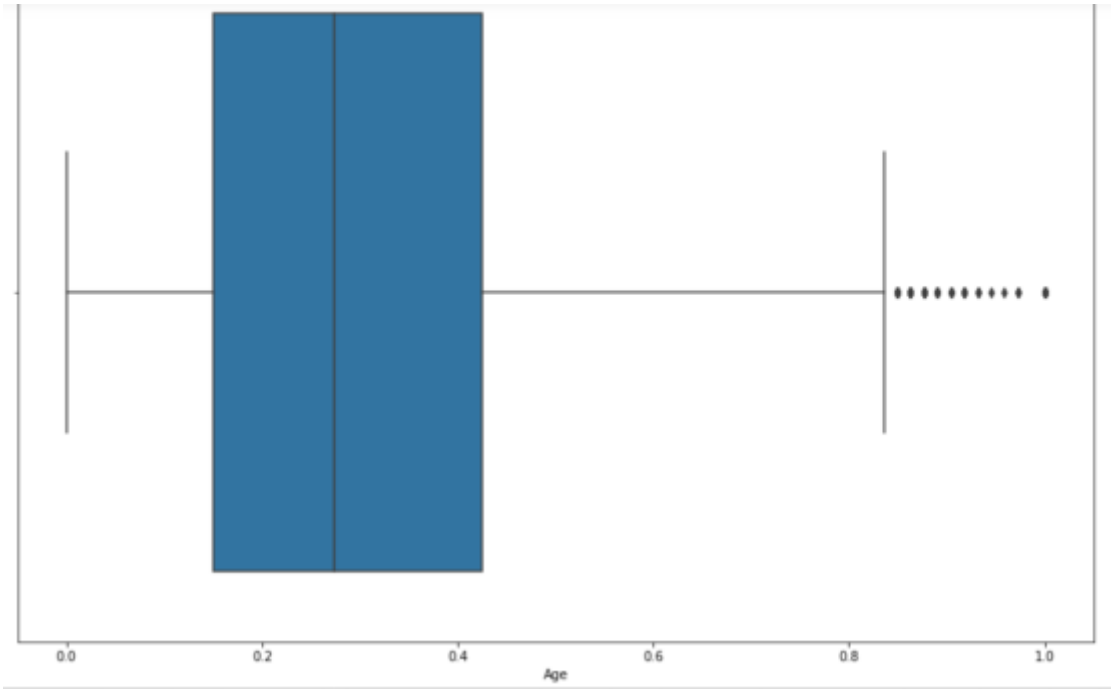
第一筆資料集

```
min_max_scaler = MinMaxScaler()

scaled_encoded_adult_data_train = pd.DataFrame()

column_values = encoded_adult_data_train.columns.values
column_values = column_values[:-1]
print(column_values[-1])

scaled_values = min_max_scaler.fit_transform(encoded_adult_data_train[column_values])

for i in range(len(column_values)):
    scaled_encoded_adult_data_train[column_values[i]] = scaled_values[:,i]

scaled_encoded_adult_data_train['hrs_per_week'] = encoded_adult_data_train['hrs_per_week']
scaled_encoded_adult_data_train.sample(10)
```

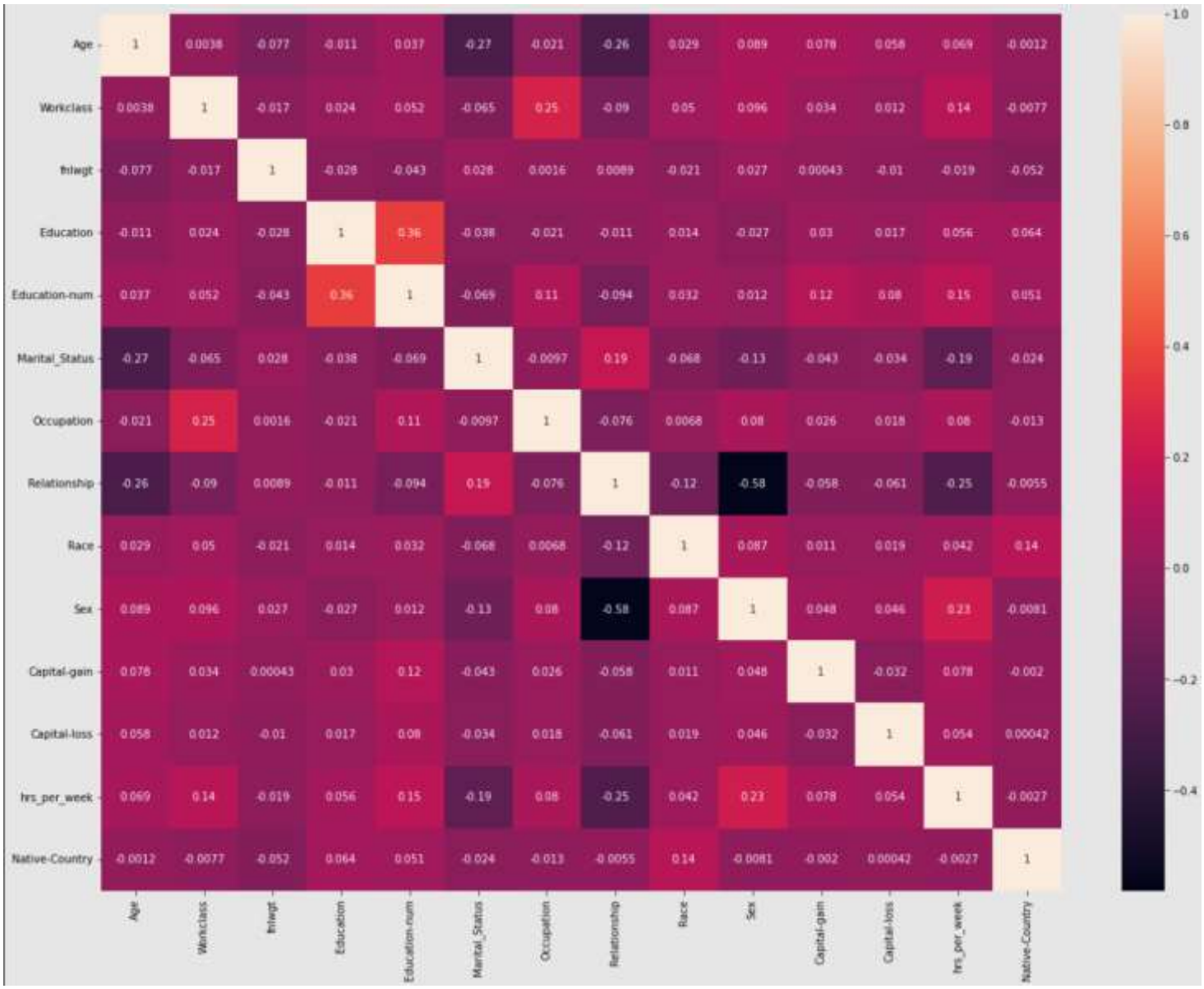
Native-Country

	Age	Workclass	fnlwgt	Education	Education-num	Marital_Status	Occupation	Relationship	Race	Sex	Capital-gain	Capital-loss	hrs_per_week	Native-Country
788	0.575342	0.000	0.122142	0.733333	0.533333	0.333333	0.000000	0.0	1.00	1.0	0.0	0.000000	16	0.951220
16951	0.342466	0.500	0.111144	0.733333	0.533333	0.333333	0.500000	0.0	1.00	1.0	0.0	0.000000	40	0.951220
26606	0.506849	0.500	0.110686	0.666667	0.000000	0.333333	0.214286	0.0	0.25	1.0	0.0	0.000000	40	0.073171
27061	0.383562	0.500	0.142279	1.000000	0.800000	0.000000	0.428571	0.2	1.00	1.0	0.0	0.000000	30	0.951220
20729	0.315068	0.500	0.129436	1.000000	0.800000	0.000000	0.071429	0.8	1.00	0.0	0.0	0.000000	24	0.951220
21428	0.027397	0.000	0.140032	0.733333	0.533333	0.666667	0.000000	0.6	1.00	0.0	0.0	0.367769	30	0.951220
17849	0.301370	0.750	0.064484	0.733333	0.533333	0.666667	0.214286	0.4	0.00	1.0	0.0	0.000000	40	0.951220

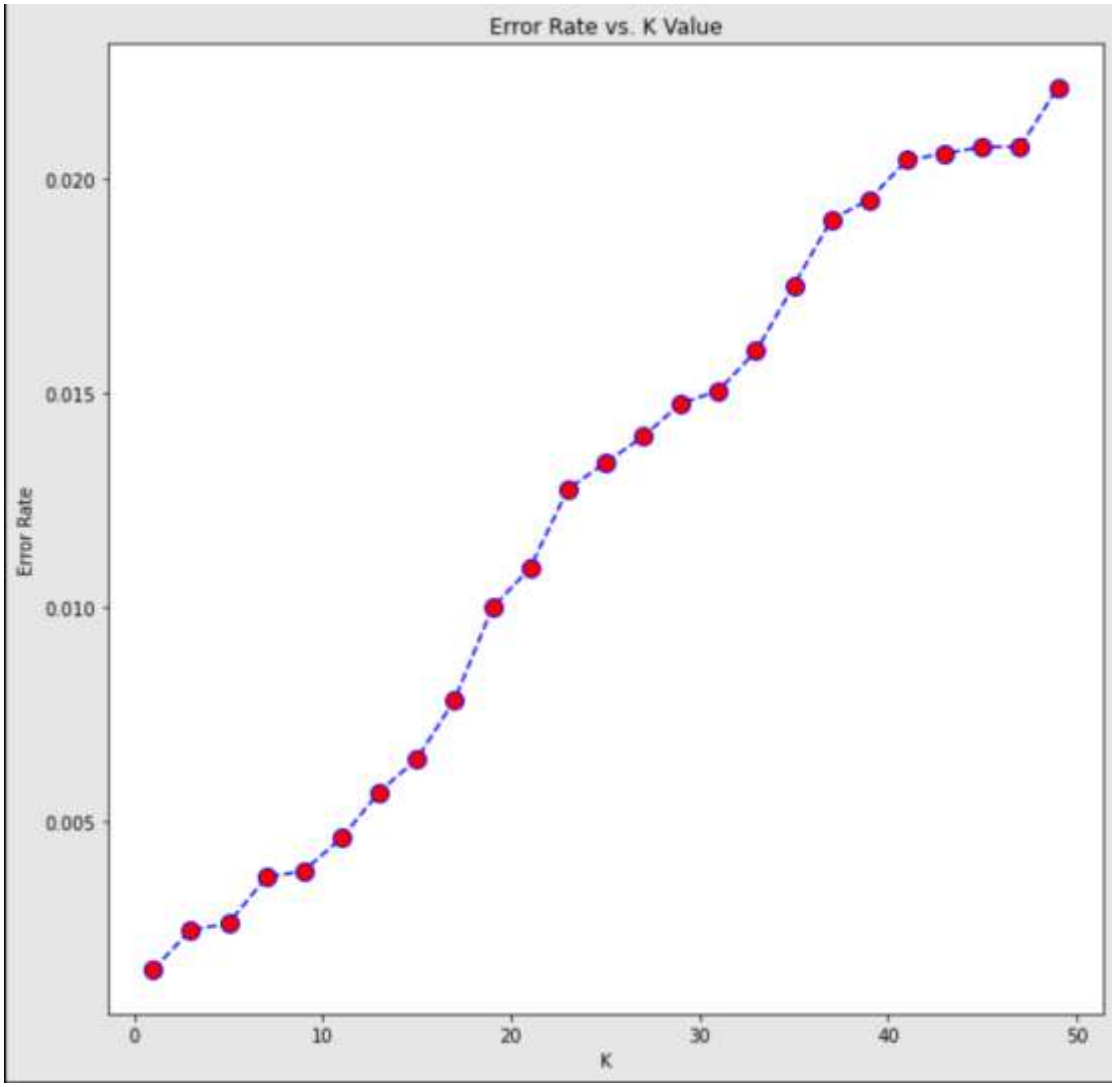
圖三、最大最小值標準化



圖四、箱型圖



圖五、各個欄位之間信度關係



圖六、K 值對錯誤率之間的關係

```
knn_train_score = knn_classifier.score(x_train, y_train)
knn_test_score = knn_classifier.score(x_test, y_test)

print('Train score: {}\nTest score: {}'.format(knn_train_score, knn_test_score))
```

```
Train score: 0.9994625307125307
Test score: 0.9981575310916628
```

```
knn_prediction = knn_classifier.predict(x_test)

knn_classifier_mae = mean_absolute_error(y_test, knn_prediction)
knn_classifier_rmse = np.sqrt(knn_classifier_mae)
knn_classifier_mape = mean_absolute_percentage_error(y_test, knn_prediction)

print('MAE: {}\nRMSE: {}\nMAPE: {}'.format(knn_classifier_mae, knn_classifier_rmse, knn_classifier_mape))
```

```
MAE: 0.0018424689083371719
RMSE: 0.042923989893032685
MAPE: 7.589319018664595e-05
```

圖七、KNN 的準確率和績效指標

```
random_forest_train_score = random_forest_classifier.score(x_train,y_train)
random_forest_test_score = random_forest_classifier.score(x_test,y_test)
print('Train score: {}\nTest score: {}'.format(random_forest_train_score, random_forest_test_score))
```

```
Train score: 0.4680589680589681
Test score: 0.4644557059766621
```

```
random_forest_prediction = random_forest_classifier.predict(x_test)

random_forest_mae = mean_absolute_error(y_test, random_forest_prediction)
random_forest_rmse = np.sqrt(random_forest_mae)
random_forest_mape = mean_absolute_percentage_error(y_test, random_forest_prediction)

print('MAE: {}\nRMSE: {}\nMAPE: {}'.format(random_forest_mae, random_forest_rmse, random_forest_mape))
```

```
MAE: 7.294188869831169
RMSE: 2.7007741240302137
MAPE: 0.311151795149642
```

圖八、RandomForest 的準確率和績效指標

```
xgboost_train_score = xgboostModel.score(x_train,y_train)
xgboost_test_score = xgboostModel.score(x_test,y_test)
print('Train score: {}\nTest score: {}'.format(xgboost_train_score, xgboost_test_score))
```

```
Train score: 0.9999232186732187
Test score: 0.9691386457853524
```

```
xgboost_prediction = xgboostModel.predict(x_test)

xgboost_mae = mean_absolute_error(y_test, xgboost_prediction)
xgboost_rmse = np.sqrt(xgboost_mae)
xgboost_mape = mean_absolute_percentage_error(y_test, xgboost_prediction)

print('MAE: {}\nRMSE: {}\nMAPE: {}'.format(xgboost_mae, xgboost_rmse, xgboost_mape))
```

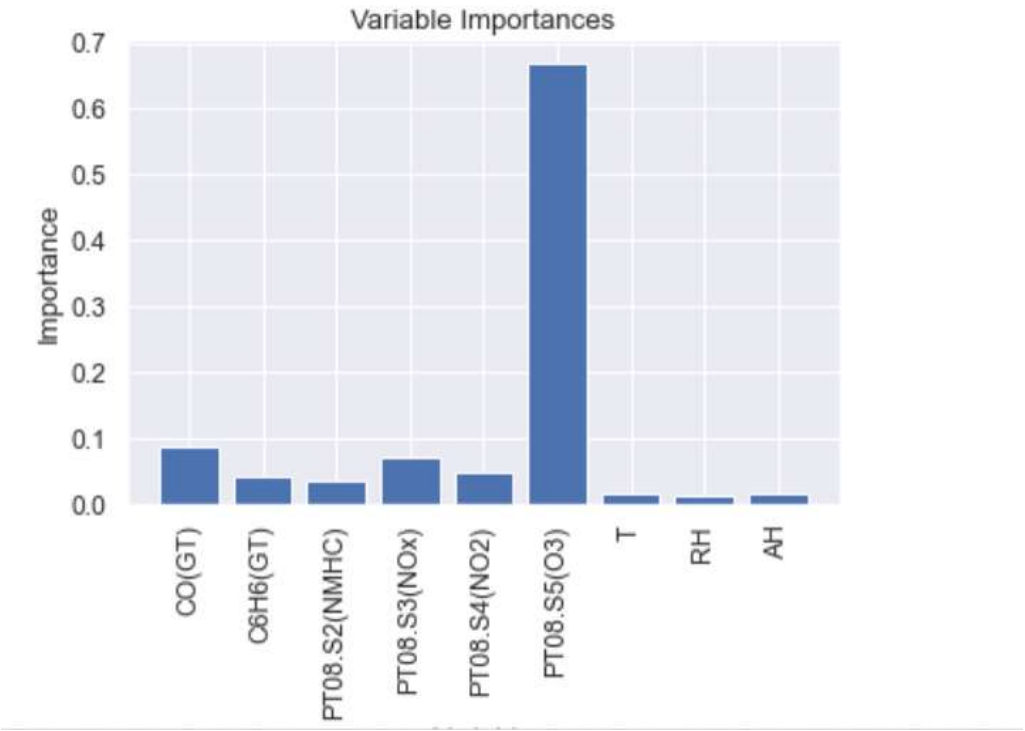
```
MAE: 0.09795793829325964
RMSE: 0.31298231626285156
MAPE: 0.0813334475862860553
```

圖九、XGboost 的準確率和績效指標

第二筆資料集

```
df_values=list(range(len(importances)))
plt.bar(df_values,importances,orientation='vertical')
plt.xticks(df_values,feature_list,rotation='vertical')
plt.ylabel('Importance')
plt.xlabel('Variable')
plt.title('Variable Importances')
```

```
Text(0.5, 1.0, 'Variable Importances')
```



圖十、欄位特徵重要性

```
from sklearn.model_selection import GridSearchCV
params = {'n_neighbors':[2,3,4,5,6,7,8,9,10,15,20,25,30]}

knn = KNeighborsRegressor()

model = GridSearchCV(knn, params, cv=10)
model=model.fit(df_train_std,y_train)
model.best_params_
y_pred = model.predict(df_test_std)
print(model.best_score_)
print('Mean squared error: %.2f'
      % mean_squared_error(y_test, y_pred))
print('Mean absolute error: %.2f'
      % mean_absolute_error(y_test, y_pred))
print('Mean absolute percentange error: %.2f'
      % mean_absolute_percentage_error(y_test, y_pred))
print('Root Mean squared error: %.2f'
      % sqrt(mean_squared_error(y_test, y_pred)))
print('R squared value: %.2f'%r2_score(y_test, y_pred))
scores = cross_val_score(model, df_train_std, y_train, scoring='r2', cv=10).mean()#cross_val_score這是驗證用來評分資料準確度的
scores
```

```
0.9192354508009908
Mean squared error: 3106.78
Mean absolute error: 40.32
Mean absolute percentange error: 0.04
Root Mean squared error: 55.74
R squared value: 0.93

0.9186889526976465
```

圖十一、KNN 的準確率和績效指標

```
model = RandomForestRegressor(n_estimators=10,random_state=1,n_jobs=-1)
model.fit(df_train1,y_train)
# 使用訓練資料預測
y_pred=model.predict(df_test1)
print('model feature importances:',model.feature_importances_)
print('Mean squared error: %.2f'
      % mean_squared_error(y_test, y_pred))
print('Mean absolute error: %.2f'
      % mean_absolute_error(y_test, y_pred))
print('Mean absolute percentange error: %.2f'
      % mean_absolute_percentage_error(y_test, y_pred))
print('Root Mean squared error: %.2f'
      % sqrt(mean_squared_error(y_test, y_pred)))
print('R squared value: %.2f'%r2_score(y_test, y_pred))
```

```
model feature importances: [0.08844051 0.07729774 0.07437211 0.05151866 0.6721306  0.02149773
 0.01494265]
Mean squared error: 3177.29
Mean absolute error: 40.83
Mean absolute percentange error: 0.04
Root Mean squared error: 56.37
R squared value: 0.92
```

圖十二、Random Forest 的準確率和績效指標

```
df_train1=df_train[['CO(GT)','C6H6(GT)','PT08.S3(NOx)','PT08.S4(NO2)','PT08.S5(O3)','T','RH']]
df_test1=df_test[['CO(GT)','C6H6(GT)','PT08.S3(NOx)','PT08.S4(NO2)','PT08.S5(O3)','T','RH']]
model=xgb.XGBRegressor()
model.fit(df_train1,y_train)
# 使用訓練資料預測
y_pred=model.predict(df_test1)
print('model feature importances:',model.feature_importances_)
print('Mean squared error: %.2f'
      % mean_squared_error(y_test, y_pred))
print('Mean absolute error: %.2f'
      % mean_absolute_error(y_test, y_pred))
print('Mean absolute percentange error: %.2f'
      % mean_absolute_percentage_error(y_test, y_pred))
print('Root Mean squared error: %.2f'
      % sqrt(mean_squared_error(y_test, y_pred)))
print('R squared value: %.2f'%r2_score(y_test, y_pred))
```

```
model feature importances: [0.09834594 0.0611056  0.04658936 0.03736635 0.72213095 0.02027939
 0.0141825 ]
Mean squared error: 5052.68
Mean absolute error: 48.38
Mean absolute percentange error: 0.04
Root Mean squared error: 55.25
R squared value: 0.93
```

圖十三、XGboost 的準確率和績效指標

五、結論

此研究在使用兩種內容有明顯差異的資料集來做三種分類器的預測和三種不同的績效指標，從實驗結果可以發現，本組的自選資料集在 KNN 還有 Random forest 有比較低的 MSE 的值還有比較高的 R square 值，代表使用這兩個演算法去跑會是比較適合本組的資料集的，也可以從 feature importance 發現我們做特徵篩選的時候會跟原本的數值有些許的差別。本組期盼未來的研究者可以利用本組的資料為基礎，未來空氣汙染的相關政策時，能更投入相關的研究。

六、參考資料

UCI Adult Data for Earning Potential of people ; 2020/12/10
Retrieved from: Adult UCI Dataset Analysis (aniket-mishra.github.io)

Air Quality Data Set ; 2021/10/12
Retrieved from: https://archive.ics.uci.edu/ml/datasets/Air+Quality#