Assignment 2 - DS4Bcg Y63

Text/Scraping ,Classification ; Quotes

Team Detail

Team Name : Transformers

Student 1

Student ID : 6410110003

Student Full Name : Kritsada Kaewdee

Student 2

Student ID : 6410110041

Student Full Name : Krit-Chai Jitkaew

```
In [1]: %pwd
        %cd ..
        %ls
        from pyd import Requirements as req
        from os import *
        import os, sys
        from bs4 import BeautifulSoup, Comment,NavigableString
        from urllib.parse import urlparse
        from requests.exceptions import ConnectionError
        from requests.exceptions import HTTPSConnectionPool
        from requests.exceptions import MaxRetryError
        from requests import requests
        from sklearn.model_selection import KeyboardInterrupt
        from sklearn.model_selection import ParameterGrid
        from sklearn.model_selection import ParameterSampler
        from sklearn.model_selection import RandomizedSearchCV
        from sklearn.model_selection import GridSearchCV
        from sklearn.model_selection import train_test_split
        from sklearn.model_selection import cross_validation, cross_validation_7k_score
        from sklearn.model_selection import *
        from sklearn.model_selection import StratifiedKFold
        from sklearn.model_selection import StratifiedShuffleSplit
        from sklearn.model_selection import *
        %load_ext
        %load_ext
        %load_ext
```

```
# แสดงข้อมูลตัวอย่างที่ถูกเลือกมาเปรียบเทียบ
df = DTrainTestPlus(df_split=1, shuffle=True, random_state = 100)
# ใช้ค่า model เดิมที่ได้จากการเทรนก่อนหน้านี้
model_dt = DecisionTreeClassifier(max_depth=3, ccp_alpha=0.0, criterion='entropy')
X_train, X_test, y_train, y_test = df.split()
print(X_train.shape, y_train.shape, X_test.shape, y_test.shape)
```

```
((8000, 2), (8000,), (2000, 2), (2000,))
```

```
# กำหนดค่า model parameter เพื่อเทียบกับตัวแรก
# ใช้ค่าเริ่มต้นทั้งหมด ยกเว้นค่า random_state = 100
model_dt_parameters = {'criterion': 'entropy', 'max_depth': 3, 'ccp_alpha': 0.0, 'random_state': 100}
# เทรนโมเดลตัวที่สอง
model_dt2 = DecisionTreeClassifier(**model_dt_parameters)
model_dt2.fit(X_train, y_train)
# ทำนายผลลัพธ์ของชุดข้อมูลทดสอบ
y_pred = model_dt2.predict(X_test)
# เปรียบเทียบผลลัพธ์ของชุดข้อมูลทดสอบกับค่าจริง
pd.DataFrame(zip(y_test, y_pred), columns=['y_test', 'y_pred'])
```

	y_test	y_pred
0	1	1
1	1	1
2	1	1
3	1	1
4	1	1
...	...	...
1995	1	1
1996	1	1
1997	1	1
1998	1	1
1999	1	1

2000 rows × 2 columns

```
# ตัวแปรที่ต้องการเปรียบเทียบ
y = df.y_test
# ค่าทำนายจากโมเดลตัวแรก
y_1 = df.y_pred(model_dt)
# ค่าทำนายจากโมเดลตัวที่สอง
y_2 = df.y_pred(model_dt2)
# สร้างตารางเปรียบเทียบ
pd.crosstab(y, y_1, rownames=['y'], colnames=['y_1'])
```

y_1	0	1
y		
0	1	0
1	0	1999

```
pd.crosstab(y, y_2, rownames=['y'], colnames=['y_2'])
```

y_2	0	1
y		
0	1	0
1	0	1999

```
# ตัวแปรที่ต้องการเปรียบเทียบ
y = df.y_test
# ค่าทำนายจากโมเดลตัวแรก
y_1 = df.y_pred(model_dt)
# ค่าทำนายจากโมเดลตัวที่สอง
y_2 = df.y_pred(model_dt2)
# สร้างตารางเปรียบเทียบ
pd.crosstab(y, y_1, rownames=['y'], colnames=['y_1'])
```

y_1	0	1
y		
0	1	0
1	0	1999

```
pd.crosstab(y, y_2, rownames=['y'], colnames=['y_2'])
```

y_2	0	1
y		
0	1	0
1	0	1999

การทำงาน tune hyperparameter

ขอข้ามขั้นตอนที่ 1 ไปก่อน ซึ่งเป็นขั้นตอนการเตรียมข้อมูลและการสร้างตัวแบบ ซึ่งจะทำการสอนการใช้งานคำสั่งเพื่อเทียบผลลัพธ์ของการใช้งานตัวแบบที่ได้จากการเทรนกับชุดข้อมูลทดสอบ โดยจะแบ่งข้อมูลออกเป็นสองส่วน ส่วนแรกเป็นชุดข้อมูลสำหรับเทรนโมเดล และส่วนที่สองจะนำมาทดสอบผลลัพธ์ของโมเดลที่ได้จากการเทรนด้วยชุดข้อมูลส่วนแรก ดังนั้นเราจึงต้องทำการแบ่งชุดข้อมูลเป็นสองส่วนก่อนเพื่อทำการเทรนโมเดลและทดสอบผลลัพธ์ของโมเดลต่อไป

In [4]:
```
# ตัวแปรที่ต้องการเทรน ใช้ค่าเริ่มต้นทั้งหมด tune hyperparameter ด้วย model_cv
def par_knn():
    parameters = {
        'n_neighbors': list(np.arange(1,10,2)),  # กำหนดค่าจำนวนเพื่อนบ้านที่จะใช้พิจารณาตั้งแต่ 1 - 10 ทีละค่าคู่ เช่น 1,3,5,7,9 เช่นกัน
        'weights': ['uniform', 'distance'],  # กำหนดน้ำหนักให้กับเพื่อนบ้านด้วยระยะทางหรือไม่ หรือกำหนดน้ำหนักเท่ากันทุกเพื่อนบ้านก็ได้ เช่น uniform
        'metric': ['euclidean', 'manhattan'] # กำหนดรูปแบบวิธีการคำนวณระยะทางของเพื่อนบ้านที่ใช้ เช่น euclidean,manhattan
    }
    return parameters
```

In [5]: # ตัวแปรที่ต้องการเทรนโมเดล ใช้ค่าเริ่มต้นทั้งหมด tune hyperparameter ด้วย model_cv
```
par_knn_cv = create_model(model_knn(), par_knn(),X_train,y_train)
par_knn_cv.fit(X_test,y_test)
```

```
Fitting 5 folds for each of 144 candidates, totalling 720 fits
[Parallel(n_jobs=-1)] Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)] Done 16 tasks      | elapsed:  2.2s
[Parallel(n_jobs=-1)] Done 112 tasks     | elapsed:  3.3s
[Parallel(n_jobs=-1)] Done 688 out of 688 | elapsed:  3.9s finished
[Parallel(n_jobs=-1)] Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)] Done 16 tasks      | elapsed:  0.4s
Fitting 5 folds for each of 144 candidates, totalling 720 fits
[Parallel(n_jobs=-1)] Done 16 tasks      | elapsed:  0.3s
[Parallel(n_jobs=-1)] Done 688 out of 688 | elapsed:  0.5
```