CUAI BASIC 스터디 6팀

2022.05.30

발표자 : 고가연, 김동우

스터디원 소개 및 만남 인증

팀원 1 : 정은진

팀원 2 : 김동우

팀원 3 : 오규안

팀원 4 : 고가연

목차

- 대회 소개
- 선택한 알고리즘 소개 및 이유
- 코드 소개

대회 소개

데이콘 Basic 범죄 유형 분류 AI 경진대회

알고리즘 | 정형 | 분류 | 사회 | Macro F1 Score

2023.05.15 ~ 2023.05.29 10:00

...

주제: 사건 발생 장소 및 기후 데이터 분석을 통해 세 가지 범죄 유형을 분류하는 AI 모델 개발

효과: 범죄 관련 데이터를 분석하여 어떤 유형의 범죄가 발생할지 예측하고, 예측 결과를 바탕으로 범죄에 대한 대응을 빠르게 할 수 있도록 도움을 줄 수 있음

대회 소개

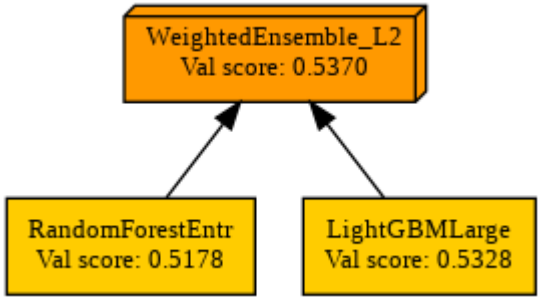
Dataset Info.

train.csv [파일] test.csv [파일]
84406개의 데이터 17289개의 데이터

ID : 샘플 별 고유 id
월 : 사건 발생월
요일 : 월요일 ~ 일요일
시간 : 사건 발생 시각
소관경찰서 : 사건 발생 구역의 담당 경찰서
소관지역 : 사건 발생 구역
사건발생거리 : 가장 가까운 경찰서에서 사건 현장까지의 거리
강수량(mm)
강설량(mm)
적설량(cm)
풍향 : 범죄발생지에서 바람이 부는 방향(최대 360도)
안개 : 가시거리가 1km 미만인 경우
짙은안개 : 가시거리가 200m 미만인 경우
번개
진눈깨비
서리
연기/연무 : 먼지, 연기가 하늘을 가리는 현상
눈날림
범죄발생지 : 범죄가 발생한 장소
TARGET : 범죄타입 [0 : 강도, 1: 절도, 2: 상해]

선택한 알고리즘

autogloun, pycaret 등의 autoML모델을 사용해보았고
데이콘 점수가 더 높았던 autogloun 선택



autogloun

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.5200	0.6440	0.5200	0.5134	0.5095	0.2459	0.2499	20.1420
ridge	Ridge Classifier	0.4824	0.0000	0.4824	0.3862	0.3802	0.1158	0.1754	1.1500
dummy	Dummy Classifier	0.4319	0.5000	0.4319	0.1865	0.2605	0.0000	0.0000	1.5800
knn	K Neighbors Classifier	0.4206	0.5542	0.4206	0.4025	0.3984	0.0739	0.0769	12.5480
svm	SVM - Linear Kernel	0.3650	0.0000	0.3650	0.4410	0.2975	0.0737	0.0899	14.7980
nb	Naive Bayes	0.3501	0.6434	0.3501	0.4822	0.2511	0.0543	0.1080	1.5420
dt	Decision Tree Classifier	0.3009	0.5000	0.3009	0.0905	0.1392	0.0000	0.0000	1.4180
rf	Random Forest Classifier	0.3009	0.5234	0.3009	0.0905	0.1392	0.0000	0.0000	9.7120
ada	Ada Boost Classifier	0.3009	0.5000	0.3009	0.0905	0.1392	0.0000	0.0000	5.9660
gbc	Gradient Boosting Classifier	0.3009	0.5019	0.3009	0.0905	0.1392	0.0000	0.0000	27.7900
lda	Linear Discriminant Analysis	0.3009	0.5000	0.3009	0.0905	0.1392	0.0000	0.0000	2.2820
et	Extra Trees Classifier	0.3009	0.5716	0.3009	0.0905	0.1392	0.0000	0.0000	6.2820
xgboost	Extreme Gradient Boosting	0.3009	0.5657	0.3009	0.0905	0.1392	0.0000	0.0000	13.2780
lightgbm	Light Gradient Boosting Machine	0.3009	0.4915	0.3009	0.0905	0.1392	0.0000	0.0000	4.3160
qda	Quadratic Discriminant Analysis	0.2942	0.5000	0.2942	0.0867	0.1339	0.0000	0.0000	1.9460

pycaret

선택한 알고리즘

앙상블 학습 : 여러 개의 분류기를 생성하고 그 예측을 결합 → 정확한 최종 예측 도출

- Random Forest, Gradient Boosting : 뛰어난 성능과 쉬운 사용, 다양한 활용도

  Random Forest : 앙상블 알고리즘 중 비교적 빠른 수행 속도

  기반 알고리즘: 결정 트리 → 쉽고 직관적인 장점 유지

- XGBoost, LightGBM

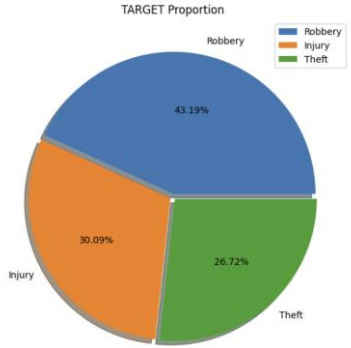
  LightGBM : XGBoost보다 학습에 걸리는 시간 단축, 메모리 사용량이 적음.

  리프 중심 트리 분할 방식을 사용 → 오류 손실 최소화

코드 소개 - EDA

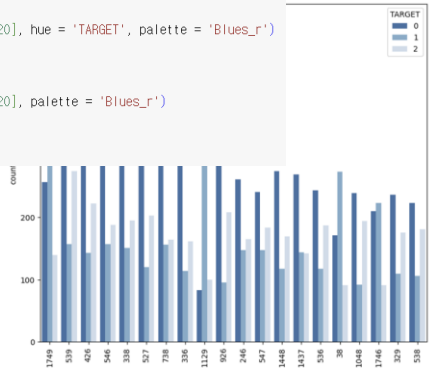
Multiclass Classification 시 가장 중요하게 판단해야 할 것 : Target 값의 분포

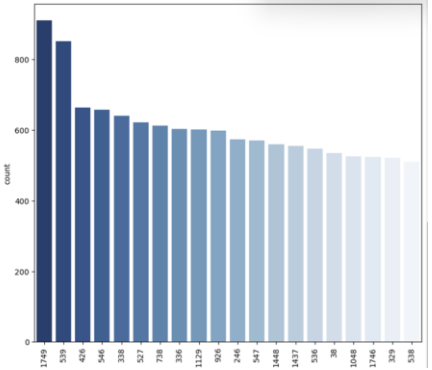
```
import matplotlib.pyplot as plt

plt.figure(figsize = (8,7), facecolor = 'w')
plt.pie(x = train_df['TARGET'].value_counts(), autopct = '%.2f%%',
        labels = ['Robbery', 'Injury', 'Theft'], shadow = True, explode = [0.02, 0.02, 0.02])
plt.legend()
plt.title("TARGET Proportion")
plt.show()
```



소관경찰서별 범죄발생 빈도 확인

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize = (20, 8))

plt.subplot(121)
sns.countplot(x = '소관경찰서', data = train_df, order = train_df['소관경찰서'].value_counts().index[:20], hue = 'TARGET', palette = 'Blues_r')
plt.xticks(rotation = 90)

plt.subplot(122)
sns.countplot(x = '소관경찰서', data = train_df, order = train_df['소관경찰서'].value_counts().index[:20], palette = 'Blues_r')
plt.xticks(rotation = 90)

plt.show()
```





코드 소개 - Data Preprocessing

```python
### 풍향 구분
import numpy as np

cardinal_directions = np.zeros(len(train_df))
cardinal_directions[train_df.query('풍향 <= 45').index] = 1 # 북 ~ 북동
cardinal_directions[train_df.query('풍향 > 45 and 풍향 <= 90').index] = 2 # 북동 ~ 동
cardinal_directions[train_df.query('풍향 > 90 and 풍향 <= 135').index] = 3 # 동 ~ 남동
cardinal_directions[train_df.query('풍향 > 135 and 풍향 <= 180').index] = 4 # 남동 ~ 남
cardinal_directions[train_df.q
cardinal_directions[train_df.q
cardinal_directions[train_df.q
cardinal_directions[train_df.q

# 카테고리형으로 변환한다.
train_df['풍향'] = cardinal_dir
train_df['풍향'] = train_df['풍
```

```python
### 요일 구분 + 주말여부 추가
from sklearn.preprocessing import LabelEncoder
le_week = LabelEncoder()
le_week.fit(train_df['요일'].astype('category'))

train_df['요일'] = le_week.transform(train_df['요일'].astype('category'))
train_df['주말여부'] = train_df['요일'].isin(['토요일','일요일']).astype('int')
```

```python
### 범죄발생지 카테고리형으로
from sklearn.preprocessing i
le_loc = LabelEncoder()
le_loc.fit(train_df['범죄발생지'].astype('category'))

train_df['범죄발생지'] = le_loc.transform(train_df['범죄발생지'].astype('category'))
```

Python

```python
# 정수형으로 변환
for k in ['소관경찰서', '소관지역', '풍향', '안개', '짙은안개', '번개', '진눈깨비', '서리', '연기/연무',  '눈날림', '범죄발생지', '계절',
    train_df[k] = train_df[k].astype('int')
```

Python

코드 소개 - autogluon

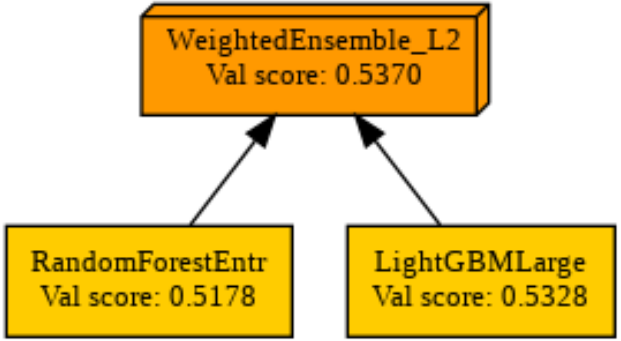
```
from autogluon.tabular import TabularDataset, TabularPredictor
predictor = TabularPredictor(label='TARGET', eval_metric='f1_macro').fit(train)
```

```
leaderboard_sorted = leaderboard.sort_values(by='score_val', ascending=False)
print(leaderboard_sorted)
```

	model	score_val	pred_time_val	fit_time
0	WeightedEnsemble_L2	0.536981	0.564614	59.159169
1	LightGBMLarge	0.532819	0.054394	4.780814
2	LightGBM	0.532459	0.067288	5.154675
3	LightGBMXT	0.531581	0.074468	7.683348
4	CatBoost	0.528073	0.010865	102.142849
5	RandomForestGini	0.527462	0.289740	46.974932
6	XGBoost	0.524959	0.017458	2.837328
7	NeuralNetFastAI	0.524135	0.070301	131.981363
8	NeuralNetTorch	0.521672	0.021761	88.053609
9	RandomForestEntr	0.517812	0.507262	51.591808
10	ExtraTreesEntr	0.517146	0.265100	21.802649
11	ExtraTreesGini	0.514585	0.281868	21.006346
12	KNeighborsDist	0.384812	0.064479	0.142520
13	KNeighborsUnif	0.373498	0.056506	0.160294

코드 소개

pygraphviz를 이용하여 앙상블 모델 시각화



RandomForest와 LightGBM을 앙상블

115	kdwkdw	kd	0.52817	14	35분 전

감사합니다.