Fire Prediction Model

학번: 2018005

이름: 김민성

Github address: https://github.com/minsungkim333333/Fire-Prediction-Model

1. **안전 관련 머신러닝 모델 개발의 목적**
    a. 화재유형별 피해를 분석하여 화재 예방에 도움을 준다.
    b. 학습 모델 활용 대상 : 소방처 등
    c. 사망, 부상자, 재산피해, 인명피해, 발생 시간을 독립변수로 하고 화재 유형을 종속변수로 하였다.
    d. 개발의 의의: 화재 발생을 예측하여 피해를 입기 전에 예방할 수 있다.

2. **안전 관련 머신러닝 모델의 네이밍의 의미**
    a. Fire Prediction Model 은 화재 예측 모델의 영문명으로 사망, 부상자, 재산피해, 인명피해, 발생 시간 등을 이용하여 화재유형을 예측한다. 직관적인 이름이 좋을 것 같아 화재 예측 모델이라고 네이밍을 하였다.

3. **개발 계획**

    a. 데이터 전처리 계획

i. 데이터의 정보를 확인하였다.

```
>>> df=pd.read_csv(".\data\소방청_화재현황.csv")
... df.info()
...
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40113 entries, 0 to 40112
Data columns (total 25 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   연번          40113 non-null  int64
 1   시도          40113 non-null  object
 2   시군구         40113 non-null  object
 3   읍면동         40113 non-null  object
 4   화재발생(년)     40113 non-null  int64
 5   화재발생(월)     40113 non-null  int64
 6   화재발생(일)     40113 non-null  int64
 7   화재발생(시)     40113 non-null  int64
 8   화재발생(분)     40113 non-null  int64
 9   요일          40113 non-null  object
 10  인명피해(명)소계  40113 non-null  int64
 11  사망          40113 non-null  int64
 12  부상          40113 non-null  int64
 13  재산피해소계      40113 non-null  int64
 14  화재유형        40113 non-null  object
 15  발화요인대분류     40113 non-null  object
 16  발화요인소분류     40113 non-null  object
 17  최초착화물대분류    40113 non-null  object
 18  최초착화물소분류    40113 non-null  object
 19  발화관련기기대분류  16481 non-null  object
 20  발화관련기기소분류  16481 non-null  object
 21  장소대분류       40113 non-null  object
 22  장소중분류       40113 non-null  object
 23  장소소분류       40113 non-null  object
 24  차량장소        4669 non-null   object
dtypes: int64(10), object(15)
memory usage: 7.7+ MB
```

ii. 화재 유형별 일시, 인명피해(명)소계 등 정수형 자료들을 사용한다.

b. 머신러닝 모델 : 화재 유형별 수치들을 분류하는 작업이기 때문에 **RandomForestClassifier** 라는 분류 작업에 특화된 모델을 사용하였다. 이 모델은 의사결정 트리(Decision Tree)를 기반으로 하며, 여러 개의 의사결정 트리를 조합하여 높은 정확도와 일반화 성능을 제공한다.

i. 앙상블 모델: 여러 개의 결정 트리를 함께 사용하는 앙상블 모델이다. 이러한 결정 트리들을 랜덤한 서브셋으로 학습하고 그 결과를 종합하여 예측을 수행한다.
ii. 부트스트랩 샘플링: 각 트리는 원본 데이터셋에서 랜덤하게 선택된 부트스트랩 샘플(복원추출)을 사용하여 학습한다. 이로써 각 트리는 서로 다른 관측치들을 학습하게 된다.
iii. 랜덤 특성 선택: 각 트리는 랜덤하게 선택된 특성들을 사용하여 분할을 수행한다. 이는 특성의 다양성을 증가시켜 모델이 더욱 강건하게 되도록 도와준다.
iv. 다양한 의사결정 트리의 결합: 각 트리가 독립적으로 학습하고 예측하므로, 다양한 관점에서 데이터를 해석하고 학습할 수 있다. 이를 통해 과적합(Overfitting)을 줄이고 일반화 성능을 향상시킨다.
v. Out-of-Bag 평가: 부트스트랩 샘플링을 통해 생성되지 않은 샘플들을 사용하여 각 트리의 성능을 평가하는데 사용됩니다. 이를 통해 교차 검증을 수행하지 않고도 모델의 성능을 추정할 수 있다.

c. 사용할 성능 지표
i. Accuracy, precision, recall, f1-score 을 이용하여 성능을 확인하려고 한다.

d. 성능 검증 방법 계획 등
K-fold 교차검증을 사용하여 성능을 확인할 예정이다.

4. 개발 과정

a. 화재 유형의 종류 확인

```
>>> import pandas as pd
>>> df=pd.read_csv(".\data\소방청_화재현황.csv")
... df.info()
... unique_values = df['화재유형'].unique()
... print(unique_values)
['건축,구조물' '기타(쓰레기 화재등)' '임야' '자동차,철도차량' '선박,항공기' '위험물,가스제조소등']
```

독립, 종속 변수 설정

```
features = ['사망', '부상', '재산피해소계', '인명피해(명)소계', '화재발생(년)', '화재발생(월)', '화재발생(일)', '화재발생(시)']
target = '화재유형'
```

데이터 분할, 모델 선택 및 학습

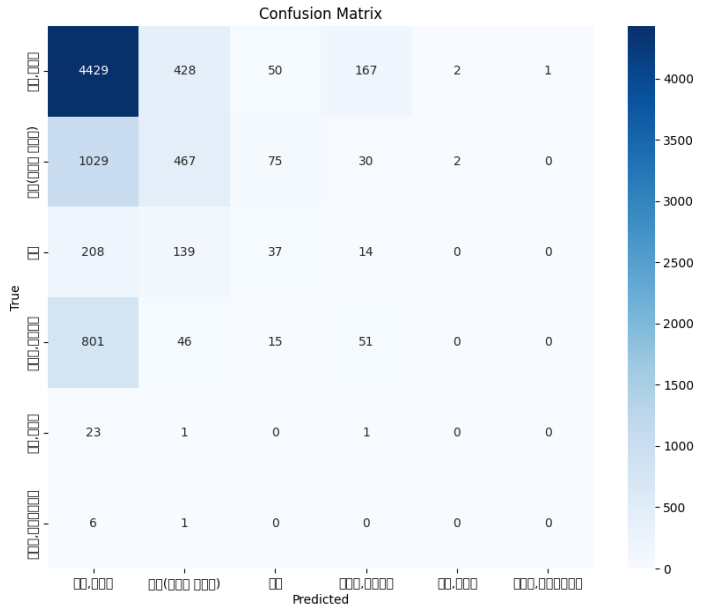
```
>>> X_train, X_test, y_train, y_test = train_test_split(df[features], df[target], test_size=0.2, random_state=42)
Backend TkAgg is interactive backend. Turning interactive mode on.
>>> model = RandomForestClassifier(n_estimators=100, random_state=42)
... model.fit(X_train, y_train)
```

모델 평가

```
>>> y_pred = model.predict(X_test)
... print("Classification Report:")
... print(classification_report(y_test, y_pred))
...
Classification Report:
              precision    recall  f1-score   support

      건축,구조물       0.68      0.87      0.77      5077
 기타(쓰레기 화재등)       0.43      0.29      0.35      1603
      선박,항공기       0.00      0.00      0.00        25
  위험물,가스제조소등       0.00      0.00      0.00         7
          임야       0.21      0.09      0.13       398
    자동차,철도차량       0.19      0.06      0.09       913

    accuracy                           0.62      8023
   macro avg       0.25      0.22      0.22      8023
weighted avg       0.55      0.62      0.57      8023

>>>
```

혼동 행렬 계산 및 시각화

```
cm = confusion_matrix(y_test, y_pred, labels=unique_values)

# 혼동 행렬 시각화
plt.figure(figsize=(10, 8))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=unique_values, yticklabels=unique_values)
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.show()
```

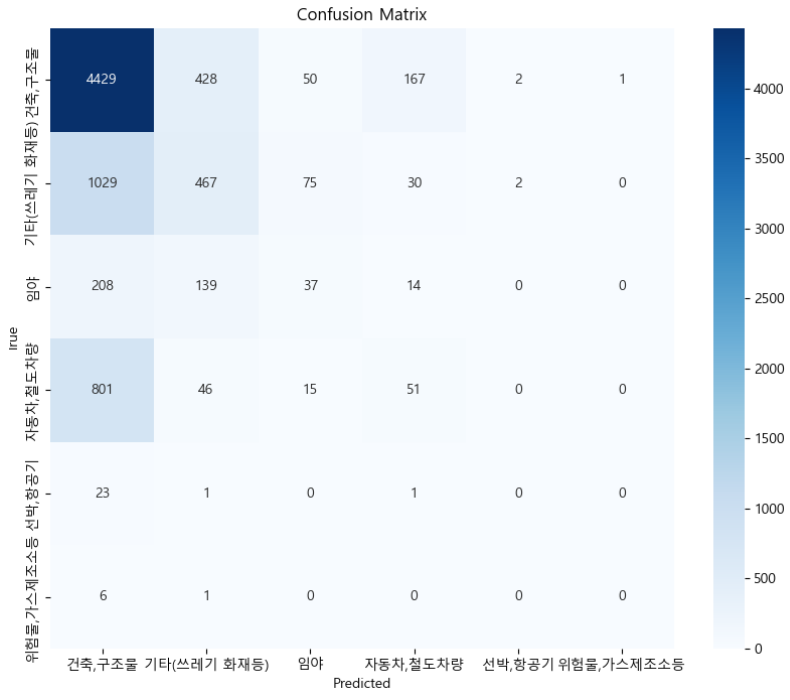


혼동행렬 항목들이 박스로 뜨는 오류 발생

```
C:\Program Files\Python38\lib\site-packages\seaborn\utils.py:84: UserWarning: Glyph 44148 (\N{HANGUL SYLLABLE GEON}) missing from current font.
  fig.canvas.draw()
```

한글 폰트가 없다고 판단, 가능한 한글 폰트 적용

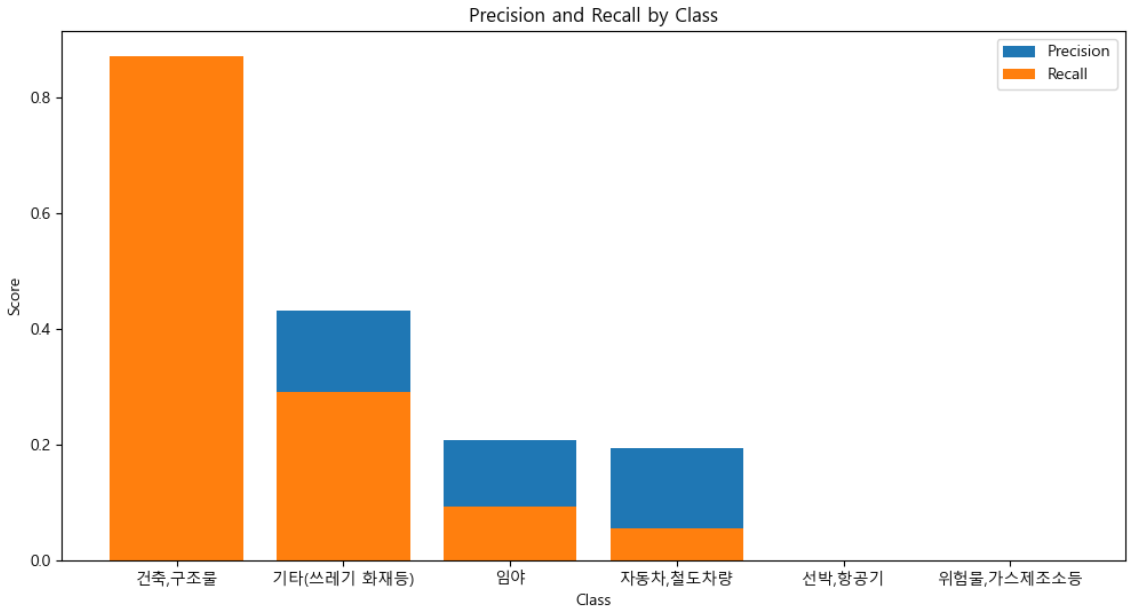
```
plt.rcParams["font.family"] = "Malgun Gothic"
```



오류 해결

클래스 별 정밀도,재현율 f1 계산 후 시각화

```
precision, recall, fscore, _ = precision_recall_fscore_support(y_test, y_pred, labels=unique_values)
plt.figure(figsize=(12, 6))
plt.bar(unique_values, precision, label='Precision')
plt.bar(unique_values, recall, label='Recall')
plt.xlabel('Class')
plt.ylabel('Score')
plt.title('Precision and Recall by Class')
plt.legend()
plt.show()
```



b. K-fold 교차 검증

```
>>> X = pd.get_dummies(df[features])
>>> X_train, X_test, y_train, y_test = train_test_split(X, df[target], test_size=0.2, random_state=42)
>>> model = RandomForestClassifier(n_estimators=100, random_state=42)
>>> model.fit(X_train, y_train)
>>> y_pred = model.predict(X_test)
>>> print("Classification Report:")
... print(classification_report(y_test, y_pred))
...
RandomForestClassifier(random_state=42)
>>> cv_scores = cross_val_score(model, X, df[target], cv=5, scoring='accuracy')
... print(f"\n평균 정확도: {cv_scores.mean()}")
```

```
Classification Report:
              precision    recall  f1-score   support

      건축,구조물       0.68      0.87      0.77      5077
 기타(쓰레기 화재등)       0.43      0.29      0.35      1603
      선박,항공기       0.00      0.00      0.00        25
  위험물,가스제조소등       0.00      0.00      0.00         7
         임야       0.21      0.09      0.13       398
    자동차,철도차량       0.19      0.06      0.09       913

    accuracy                           0.62      8023
   macro avg       0.25      0.22      0.22      8023
weighted avg       0.55      0.62      0.57      8023


평균 정확도: 0.2301607324218365
```

c. 결과

i. 전체적으로 정확도가 높지 않아, 더 많은 자료나 다른 모델을 사용하여 정확도를 개선할 필요가 있다고 생각된다.

ii. 결과를 보면 건축, 구조물에서 높은 정확도를 보여준다. 그렇기 때문에 이 모델은 건축 구조물을 대상으로 활용할 수 있다고 생각한다.

iii. 건축, 구조물에서 화재 원인을 예측하는 모델로서 활용이 가능하다.

5. 개발 후기

데이터를 조사하는 과정에서부터 어려움을 느꼈다. 활용성이 높은 데이터, 데이터 간의 연관성이 높은 데이터들의 가치가 높은 이유를 알게 되었다. 데이터 사이의 연관성을 파악하는 과정들이 생각보다 흥미롭고 즐거웠다. 시각화를 통해서 직관적으로 확인할 수 있어, 미래에 데이터를 분석하는 기회가 온다면 잘 활용할 수 있을 것이라고 생각했다.