RTAPM

학번: 2018068

이름: 손승현

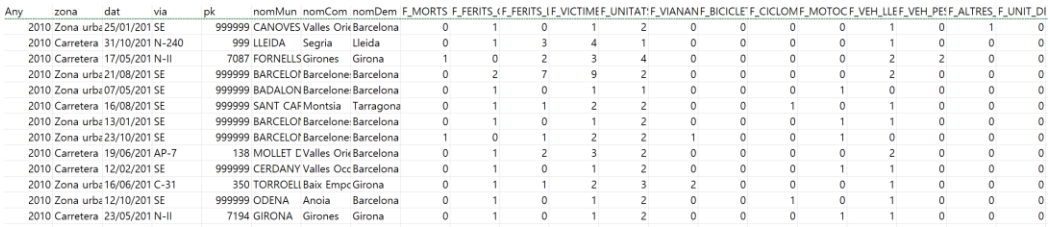
Github address: SonseungHyun1

1. **안전 관련 머신러닝 모델 개발의 목적**
   a. **학습 모델 활용 대상:** 스페인의 교통사고 통계 데이터셋을 이용해 학습 및 예측 모델을 개발 / 도로의 속도, 안개 여부, 기후 상황, 가시성의 여부 등의 독립 변수를 설정하고 피해자 수를 종속변수로 두어 사고의 규모에 대해 예측하는 머신러닝 모델을 개발
   b. 데이터의 어떠한 독립 변수를 사용하여 어떠한 종속 변수를 예측하는 지
      독립 변수 : 도로의 속도, 안개 여부, 기후 상황, 가시성의 여부, 심각성 (중상, 경상의 사고)
      종속 변수 : 피해자 수 -> 사고의 규모 예측
   c. 개발의 의의: 해당 모델을 개발 함으로써 기후,안개, 가시성과 도로 속도 등의 상관관계와 그로 발생하는 사고의 규모를 예측해 교통사고를 예방할 수 있는 플랫폼 구축하기 위함.

2. **안전 관련 머신러닝 모델의 네이밍의 의미**
   a. RTAPM 은 Road Traffic Accident Prediction Model 의 약자로
      도로 교통사고 예측 모델이라는 뜻을 의미한다.

3. **개발 계획**
   a. 데이터에 대한 요약 정리 및 시각화
      데이터 셋은 kaggle 에서 수집 했으며 Road Traffic Injuries & Deaths (Catalonia 2010/20) 라는 데이터셋이다.

      전처리 전 데이터는 총 57 개의 컬럼을 가지고 있으며 사고 지역, 날짜, 킬로미터 포인트, 사고의 규모, 사고와 관련된 차량의 댓수 등 2010 년부터 2020 년까지 카탈루냐주에서 발생한 자동차 사고와 관련된 모든 내용을 담고 있는 데이터이다.



Any	zona	dat	via	pk	nomMun	nomCom	nomDem	F_MORTS	F_FERITS_(	F_FERITS_I	F_VICTIME	F_UNITAT	F_VIANAN	F_BICICLE	F_CICLOM	F_MOTOC	F_VEH_LLE	F_VEH_PES	F_ALTRES_	F_UNIT_DI
2010	Zona urba	25/01/201	SE	999999	CANOVES	Valles Ori	Barcelona	0	1	0	1	2	0	0	0	0	1	0	1	0
2010	Carretera	31/10/201	N-240	999	LLEIDA	Segria	Lleida	0	1	3	4	1	0	0	0	0	1	0	0	0
2010	Carretera	17/05/201	N-II	7087	FORNELLS	Girones	Girona	1	0	2	3	4	0	0	0	0	2	2	0	0
2010	Zona urba	21/08/201	SE	999999	BARCELOI	Barcelone	Barcelona	0	2	7	9	2	0	0	0	0	2	0	0	0
2010	Zona urba	07/05/201	SE	999999	BADALON	Barcelone	Barcelona	0	1	0	1	1	0	0	0	1	0	0	0	0
2010	Carretera	16/08/201	SE	999999	SANT CAF	Montsia	Tarragona	0	1	1	2	2	0	0	1	0	1	0	0	0
2010	Zona urba	13/01/201	SE	999999	BARCELOI	Barcelone	Barcelona	0	1	0	1	2	0	0	0	1	1	0	0	0
2010	Zona urba	23/10/201	SE	999999	BARCELOI	Barcelone	Barcelona	1	0	1	2	2	1	0	0	1	0	0	0	0
2010	Carretera	19/06/201	AP-7	138	MOLLET D	Valles Ori	Barcelona	0	1	2	3	2	0	0	0	0	2	0	0	0
2010	Carretera	12/02/201	SE	999999	CERDANY	Valles Occ	Barcelona	0	1	0	1	2	0	0	0	1	1	0	0	0
2010	Zona urba	16/06/201	C-31	350	TORROELI	Baix Empc	Girona	0	1	1	2	3	2	0	0	0	1	0	0	0
2010	Zona urba	12/10/201	SE	999999	ODENA	Anoia	Barcelona	0	1	0	1	2	0	0	1	0	1	0	0	0
2010	Carretera	23/05/201	N-II	7194	GIRONA	Girones	Girona	0	1	0	1	2	0	0	0	1	1	0	0	0

[엑셀 파일 원문]

```
    Any         zona  ...                                  tipAcc    tipDia
0  2010  Zona urbana  ...           Col.lisió de vehicles en marxa  dill-dij
1  2010    Carretera  ...  Sortida de la calcada sense especificar        dg
2  2010    Carretera  ...           Col.lisió de vehicles en marxa  dill-dij
3  2010  Zona urbana  ...           Col.lisió de vehicles en marxa       dis
4  2010  Zona urbana  ...                    Bolcada a la calcada       div

[5 rows x 58 columns]
                Any             pk  ...  C_VELOCITAT_VIA           hor
count  16774.000000   16773.000000  ...     14654.000000  16774.000000
mean    2013.897580  523111.778811  ...       136.754402    949.659533
std        2.567239  498936.376803  ...       209.862794    740.394816
min     2010.000000       0.000000  ...         0.000000      0.000000
25%     2012.000000     155.000000  ...        80.000000    162.000000
50%     2014.000000  999999.000000  ...       100.000000   1009.000000
75%     2016.000000  999999.000000  ...       100.000000   1618.000000
max     2018.000000  999999.000000  ...       999.000000   2359.000000

[8 rows x 17 columns]
```

총 58 개의 컬럼을 가지고 있음.

b. 데이터 전처리 계획

데이터 전처리 계획으로는 머신러닝 모델이 학습할 수 있는 숫자형 데이터들과 예측하려고 하는 종속변수와 관련된 데이터들을 추출해내려고 계획함,

c. 어떠한 머신러닝 모델을 사용할 것인지 (해당 머신러닝 모델의 이론 추가)

랜덤 포레스트 모델을 사용할 예정으로

선정 사유는 랜덤 포레스트는 여러 개의 결정 트리를 사용하여 학습하고, 이들의 예측을 결합함으로써 과적합을 줄이고 일반화 성능을 향상시킬 수 있다는 강점이 있어 선정하게 되었다.

랜덤 포레스트(Random Forest)는 앙상블 학습 방법 중 하나로, 여러 개의 결정 트리를 조합하여 높은 성능의 모델을 만드는 알고리즘이다.
랜덤 포레스트는 각 트리를 독립적으로 학습하고, 예측을 결합하여 더 정확하고 안정적인 예측을 제공하는 모델이다.

d. 머신러닝 모델 예측 결과가 어떠할 지

사실 머신러닝에 대해 학습하면서 과제를 수행해 나가게 되면서 결과에 대해 예측을 하는 부분에서 어려움을 느꼈다.
다만 일반적인 의사결정 나무를 사용 할 경우 과적합에 대한 우려라는 단점을 보완 할 수 있다는 장점이 있다.

e. 사용할 성능 지표

성능 지표는 정확도를 사용해 전체 예측 중 올바르게 예측한 비율에 대해 검증하는 방법을 이용할 계획이다,

f. 성능 검증 방법 계획 등

테스트 데이터로 랜덤 포레스트 모델을 평가하고
분류보고서를 출력시켜 성능에 대한 검증을 할 예정이다.

4. 개발 과정

a. 계획 후 실제 학습 모델 개발 과정을 기록 (*개발 과정 캡쳐 필수)

모델을 개발 중 코드를 추가하면 파이썬 자체의 오류가 생기면서 결괏값이 도출이 되지 않는 오류가 생겨 그래프를 확인하는 코드와 전처리 로직 데이터를 시각화 하는 코드, 마지막으로 머신러닝 모델을 구축하는 코드 를 각각의 파일로 나누어 진행하게 되었다.

1. 데이터 확인

```
5. # 데이터 셋 불러오기
   import pandas as pd
   data_path = r"..\data\Catalunya Accidents data.csv"
   data = pd.read_csv(data_path)
```

```
print(data.head())
print(data. describe())
```

위 코드를 작동시켜

```
    Any         zona  ...                                tipAcc    tipDia
0  2010  Zona urbana  ...          Col.lisió de vehicles en marxa  dill-dij
1  2010    Carretera  ...  Sortida de la calcada sense especificar        dg
2  2010    Carretera  ...          Col.lisió de vehicles en marxa  dill-dij
3  2010  Zona urbana  ...          Col.lisió de vehicles en marxa       dis
4  2010  Zona urbana  ...                    Bolcada a la calcada       div

[5 rows x 58 columns]
                Any             pk  ...  C_VELOCITAT_VIA           hor
count  16774.000000   16773.000000  ...     14654.000000  16774.000000
mean    2013.897580  523111.778811  ...       136.754402    949.659533
std        2.567239  498936.376803  ...       209.862794    740.394816
min     2010.000000       0.000000  ...         0.000000      0.000000
25%     2012.000000     155.000000  ...        80.000000    162.000000
50%     2014.000000  999999.000000  ...       100.000000   1009.000000
75%     2016.000000  999999.000000  ...       100.000000   1618.000000
max     2018.000000  999999.000000  ...       999.000000   2359.000000

[8 rows x 17 columns]

Process finished with exit code 0
```

다음과 같이 데이터셋이 구성되어있는 것을 확인했다.

데이터 시각화와 모델 구현에서 사용될 전처리 로직은 공통적으로 사용하였다.

```
# 데이터 전처리 과정
# D_BOIRA
data['D_BOIRA'] = data['D_BOIRA'].map({"No n'hi ha": 0, "Si": 1})

# D_GRAVETAT
data['D_GRAVETAT'] = data['D_GRAVETAT'].map({'Accident greu': 1, 'Accident
mortal': 2})

# D_CLIMATOLOGIA
data['D_CLIMATOLOGIA'] = data['D_CLIMATOLOGIA'].map({'Bon temps': 0, 'Pluja
dèbil': 1})

# D_INFLUIT_VISIBILITAT
data['D_INFLUIT_VISIBILITAT'] = data['D_INFLUIT_VISIBILITAT'].map({'No': 0,
'Sense especificar': 1, 'Si': 2})

# D_INFLUIT_VISIBILITAT를 이진 분류로 변환 (예: 0 또는 1)
data['D_INFLUIT_VISIBILITAT'] = (data['D_INFLUIT_VISIBILITAT'] >
```

```
0).astype(int)

# 필요한 특징 선택 (피해자 수만 남김)
selected_features = ["F_VICTIMES", "C_VELOCITAT_VIA", "D_BOIRA",
"D_GRAVETAT", "D_CLIMATOLOGIA", "D_INFLUIT_VISIBILITAT"]
data = data[selected_features]

# 데이터 전처리 - 필요한 특징 선택 및 누락된 값 처리
selected_features = ["F_VICTIMES", "C_VELOCITAT_VIA", "D_BOIRA",
"D_GRAVETAT", "D_CLIMATOLOGIA", "D_INFLUIT_VISIBILITAT"]
data = data[selected_features]
data = data.fillna(data.mean())
```

2, 데이터 시각화

전처리 이후 숫자형 데이터만 선택하여 상관관계 분석 및 히트맵과 시각화 그래프를 생성 및 저장해 사고 심각성 분포를 확인

```
# "results" 디렉토리 생성
os.makedirs("results", exist_ok=True)

# 숫자형 데이터만 선택하여 상관관계 분석 및 히트맵
numeric_data = data.select_dtypes(include='number')
correlation_matrix = numeric_data.corr()
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f",
linewidths=.5)
plt.title('Correlation Heatmap')

# 파일 경로 지정
heatmap_filepath = "results/correlation_heatmap.png"

# 파일 저장
plt.savefig(heatmap_filepath)
plt.close()  # 플롯 창 닫기


# 시각화 그래프 - 사고 심각성 분포
plt.figure(figsize=(10, 6))
sns.countplot(data['F_VICTIMES'])
plt.title('Distribution of Accident Severity')
plt.xlabel('Severity (0: No impact, 1: Impact without specifying, 2:
Impact)')
plt.ylabel('Count')

# 파일 경로 지정
distribution_filepath = "results/severity_distribution.png"

# 파일 저장
```

```
plt.savefig(distribution_filepath)
plt.close()  # 플롯 창 닫기


# 두 파일의 경로 출력
print("Correlation Heatmap saved at:", heatmap_filepath)
print("Severity Distribution saved at:", distribution_filepath)
```

3. 분석 모델 구축

```
# 특징과 레이블 분리
X = data.drop("F_VICTIMES", axis=1)
y = data["F_VICTIMES"]

# 학습용과 테스트용으로 데이터 분할
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# 랜덤 포레스트 모델 구축 및 학습
rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train, y_train)

# 테스트 데이터로 랜덤 포레스트 모델 평가
y_pred_rf = rf_model.predict(X_test)
accuracy_rf = accuracy_score(y_test, y_pred_rf)
print(f"랜덤 포레스트 모델 정확도: {accuracy_rf}")

# 분류 보고서 출력
classification_rep_rf = classification_report(y_test, y_pred_rf,
zero_division=1)
print("분류 보고서:\n", classification_rep_rf)

import matplotlib.pyplot as plt

# 특성 중요도 가져오기
feature_importances = rf_model.feature_importances_

# 특성 중요도를 내림차순으로 정렬
indices = sorted(range(len(feature_importances)), key=lambda k:
feature_importances[k], reverse=True)

# 특성 이름 가져오기
feature_names = X.columns

# 시각화
plt.figure(figsize=(12, 8))
plt.bar(range(len(feature_importances)), feature_importances[indices],
align="center")
plt.xticks(range(len(feature_importances)), [feature_names[i] for i in
```

```
indices], rotation=45)
plt.xlabel("Feature")
plt.ylabel("Importance")
plt.title("Random Forest Feature Importance")
plt.show()
```

a. 각 함수는 어떻게 동작하는 지 구체적으로 설명

각 코드 주석 표기

b. 에러 발생 지점 및 해결 과정

클래스의 불균형으로 인해 정확도는 높지만 정밀도와 재현율이 떨어진다. 해결하는 법을 찾기 어려운 부분이다.

c. 학습 모델의 성능 평가

```
랜덤 포레스트 모델 정확도: 0.6882265275707898
분류 보고서:
              precision    recall  f1-score   support

           1       0.69      0.99      0.82      2321
           2       0.24      0.02      0.03       610
           3       1.00      0.00      0.00       203
           4       1.00      0.00      0.00       104
           5       1.00      0.00      0.00        59
           6       1.00      0.00      0.00        26
           7       1.00      0.00      0.00        13
           8       1.00      0.00      0.00         9
           9       1.00      0.00      0.00         3
          10       1.00      0.00      0.00         3
          11       1.00      0.00      0.00         2
          16       1.00      0.00      0.00         1
          20       1.00      0.00      0.00         1

    accuracy                           0.69      3355
   macro avg       0.92      0.08      0.07      3355
weighted avg       0.65      0.69      0.57      3355


Process finished with exit code 0
```

클래스 1 에 대한 Precision 은 0.69, Recall 은 0.99, F1-score 는 0.82 로 나타났고 클래스 1 을 잘 예측하고 있다는 것을 알 수있다.

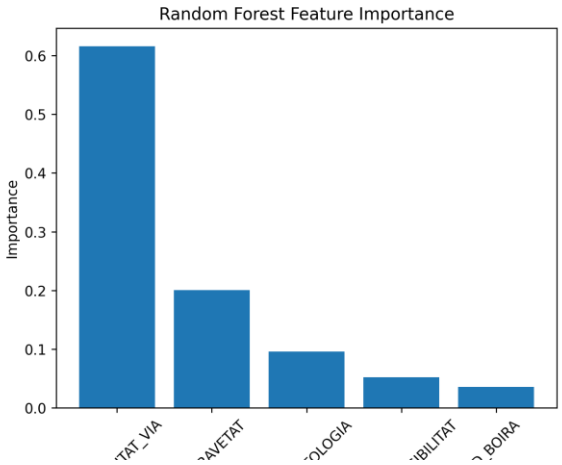
클래스 2 에 대한 Precision 은 0.24, Recall 은 0.02, F1-score 는 0.03 으로 나타났다. 이는 클래스 2 에 대한 예측이 낮은 정확도를 가지고 있음을 알 수 있다.

그 외에 클래스 3, 4, 5, 6, 7, 8, 9, 10, 11, 16, 20 에 대한 Precision, Recall, F1-score 는 모두 1.00 또는 0.00 으로 나타났다.

해당 클래스들에 대한 데이터가 부족하여 정확한 평가가 어렵다는 것을 나타낸다.

해당 모델은 클래스 1 에 대해서는 좋은 예측을 하고 있지만, 다른 클래스들에 대한 예측은 제한적인 예측이라고 볼 수 있다. 클래스 불균형이나 다양한 클래스 간 데이터의 부족이 이러한 결과에 영향 미치는 것이다. 모델의 성능을 향상시키기 위해 클래스 불균형 처리나 다양한 특성을 활용하는 방법을 고려해야한다.

d. 결과 시각화



6. 개발 후기

a. 개발 후 느낀 점 설명

최근 AI 산업이 발달하면서 머신 러닝에 대한 중요성이 높게 평가되고 있다. 직접 머신 러닝 모델을 구축하는 과제를 수행하면서 얼마나 어렵고 복잡한 작업인지 체감할 수 있었다.

이번 컴퓨터 프로그래밍 2 실습과 이론 수업을 통해 코딩에 대한 관심도가 많이 높아졌으며 프로그래밍에 대한 공부도 놓지 않고 꾸준히 해야겠다. 라는 점을 느낀 과제가 된 거 같다.