항공안전사고예측 학습 모델

학번: 1818035

이름: 이해찬

Github address:
https://github.com/LHC0121/homework_Aviation_Safety_Accident_Predictior

1. 항공안전사고예측 학습 모델 개발의 목적

항공사고예측 학습 모델을 개발하여 항공사별로 주당 가용한 좌석의 수, 발생한 경미한 사고 건 수와 중대한 사고 건 수, 그로 인한 사망자 수와 같은 데이터 통계를 이용하여 어떤 방식으로 운영해야 사고와 사망자 수를 감소시킬 수 있는지 등의 방안을 항공사에서 유용할 수 있다.
또한 이 학습 모델을 사용한다면 생명 뿐 아니라 항공사의 자산, 사고로 발생되는 기타 부수적인 문제 등 여러 방면에서 나올 수 있는 부정적 가치들을 예방할 수 있다.

항공안전사고예측 학습 모델의 네이밍의 의미

지금까지 발생한 사고들로 인한 사망자 수, 두 자료들의 인과관계를 학습하여 미래에 야기될 수 있는 크고작은 사고들을 예방 할 수 있기에 항공안전사고예측 이라는 이름으로 네이밍 하였다.

2. 개발 계획

우선 데이터의 정보를 확인한 다음 결측치 그리고 이상치가 있는지 확인합니다. 그런 다음 평균, 표준편차, 최대,최솟값을 확인하고 항공사별 사고발생현황, 연도별 사고 및 사망자 수 추이를 시각화, 변수 간의 상관 관계 분석을 위한 히트맵을 작성합니다.

데이터 전처리는 결측치를 처리하고, 범주형 변수가 있을 때에는 레이블 인코딩을 해서 숫자로 변환, 변수 간 스케일 차이를 줄이기 위해 표준화를 수행합니다.

머신러닝 모델은 변수들 간 복잡한 관계를 잘 처리할 수 있는 랜덤 포레스트(Random forest) 모델을 선택합니다
랜덤포레스트 모델은 여러 개의 결정 트리를 만들어 예측들을 결합함으로써 보다 정확하고 안정적인 결과를 얻게 합니다 분류와 회귀문제에서 모두

사용가능하며 단일 결정 트리의 한계와 과적합 문제를 해결하는데 주로 쓰입니다.

주어진 데이터를 이용하여 학습모델의 예측을 수행합니다
도출된 결과를 해석하여 항공안전사고 예측과 주요 영향을 미치는 변수를 확인합니다.

성능지표에는 정확도(Accuracy)를 중점으로 사용합니다 전체 예측한 경우 중 올바르게 예측된 비율을 계산해 정확도를 산출해 낼 수 있습니다

성능 검증방법은 K-fold Cross Validation 을 통해서 모델의 일반화 성능을 평가해 교차 검증 합니다 모델의 성능을 더 심도있게 평가하기 위해 혼동 행렬을 분석합니다

3. 개발 과정

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split, KFold, cross_val_score
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

머신러닝에 필요한 라이브러리 호출(import)

```
# 데이터 로드
filename = "/Users/noooo/Desktop/Aviation safety.csv"
column_names = ['airline', 'avail_seat_km_per_week', 'incidents_85_14', 'fatal_accident_85_14',
data = pd.read_csv(filename, names=column_names)
```

저장한 필요한 데이터를 불러온다

```
# 데이터 정보 확인
print(data.info())

# 결측치 확인
print("Missing Values:\n", data.isnull().sum())

# 기술 통계량 확인
print("Descriptive Statistics:\n", data.describe())
```

불러온 데이터와 데이터에 결측치가 있는지 확인하고, 평균 표준편차 분산 등 통계량을 확인한다

```
# 항공사별 사고발생 현황 시각화
plt.figure(figsize=(12, 6))
sns.countplot(x='airline', data=data, hue='target')
plt.title('Accident Occurrence by Airline')
plt.show()
```

항공사별 사고 현황을 시각화 자료로 나타낸다

```
# 연도별 사고 및 사망자 수 추이 시각화
data['year'] = data['incidents_85_14'].astype(str).str[:2]
plt.figure(figsize=(12, 6))
sns.countplot(x='year', data=data, hue='target')
plt.title('Accident and Fatality Trends by Year')
plt.show()
```

그 다음으로 연도별 사고, 사망자수 추이를 시각화 자료로 나타낸다

```
# 변수 간의 상관 관계 분석을 위한 히트맵 작성
plt.figure(figsize=(10, 8))
sns.heatmap(data.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()
```

변수간의 상관관계를 분석하기 위해 히트맵을 그린다

```
# 데이터 전처리
# 레이블 인코딩
label_encoder = LabelEncoder()
data['airline'] = label_encoder.fit_transform(data['airline'])
```

```
# 변수 간 스케일링
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data.drop(['target', 'year'], axis=1))
scaled_data = pd.DataFrame(scaled_data, columns=data.columns[:-2])
```

데이터 전처리 과정을 거치고 변수 간의 스케일링을 해준다

Year 열은 임시로 추가한 것이므로 제외한다

```
# 머신러닝 모델 선택
model = RandomForestClassifier(n_estimators=100, random_state=42)

# 데이터 분할
X_train, X_test, y_train, y_test = train_test_split(scaled_data, data['target'], test_size=0.2, random_state=42)
```

선택한 러닝머신 모델(Random forest)을 선택하고 데이터들을 분할한다

```
# 모델 학습
model.fit(X_train, y_train)

# 모델 평가
y_pred = model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

머신러닝 모델을 학습시키고 평가하는 과정을 거친다

```
# 성능 검증
kfold = KFold(n_splits=5, shuffle=True, random_state=42)
cv_results = cross_val_score(model, scaled_data, data['target'], cv=kfold)
print("Cross-validated Accuracy:", cv_results.mean())
```

성능검증 코드를 작성해 코드를 마무리한다

4. 개발 후기

초기에 데이터 설정단계에서부터 어떤 데이터를 가지고 머신러닝 코드를 작성할지 많은 고민이었다 데이터가 너무 방대해서도 안되고 표본이 너무 적어 충분하지 못한 결과값을 얻을 수 있어서였다 필자가 선택한 데이터는 많은 데이터는 아니지만 단편적으로 분석하기에 괜찮은 데이터셋이라고 생각해 이 데이터를 선택하게 되었다.

머신러닝 코드를 작성하며 수치의 범위 설정에도 어려움이 있었다 어느정도로 기준을 잡아야 합리적인 지표가 될지에 대해 많이 생각해보았다 머신러닝 코드를 개발해봄으로써 우리가 실생활에서 많은 부분과 접목되어 활용가능할 수 있다는 생각에 미쳐 미래에는 더 정교하고 체계적인 머신러닝, ai 기술들에 기대를 걸어 볼 수 있을 것 같다는생각이 들었다.