

Smoke Detection Using Machine Learning

학번:2118305

이름:민정운

Github address: <https://github.com/Aqau2/Quiz8>

1. 안전 관련 머신러닝 모델 개발의 목적

- a. 각 데이터 온도, 습도, 이산화탄소, 공기 압력 등으로 화재 예측하기
- b. 연기가 감지되었을 때 화재인지 아닌지를 감지함
- c. 이러한 데이터는 기업, 가정, 어디서든 사용이 가능함

2. 안전 관련 머신러닝 모델의 네이밍의 의미

- a. 연기를 감지하여 화재인지 구별할 수 있다.

3. 개발 계획

- a. 성능 검증 방법 계획 등
- b. 데이터는 시간, 기온, 공기 습도, 휘발성 유기화합물, co2, 원시 분자 수소, 에탄올 가스, 미립자 크기의 데이터를 받습니다.
- c. 데이터에 따른 화재 여부를 조사할 수 있습니다.
- d. `RandomForestClassifier` 모델을 사용하여 예측했습니다.

4. 개발 과정

```

1 import pandas as pd
2
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 from sklearn.model_selection import train_test_split
6 from sklearn.ensemble import RandomForestClassifier
7 from sklearn.metrics import r2_score, mean_absolute_error, accuracy_score
8 from sklearn.preprocessing import StandardScaler
9
10
11
12 df=pd.read_csv("smoke_detection_iot.csv")
13 df.head()
14 df.columns
15 df.shape
16 df.drop(columns=['Unnamed: 0'],inplace=True)
17 df.describe()
18 df.isnull().sum()
19 df.duplicated().sum()
20 df.info()
21 usage new *
22 def uniquecounts(df):
23     for x in df.columns:
24         print(x, len(df[x].unique()), df[x].unique())
25 uniquecounts(df)
26 df.tail()

```

a.

개발을 진행하면서 컴퓨터 저장공간의 부족으로 정말 힘들었습니다.
 룰 서든어택을 삭제하면서 개발을 진행하였습니다.
 도중에 plt.show()를 하였고 저장하는 함수까지 사용하였지만 지속적인
 블루스크린으로 하나씩 캡처 떴가면서 데이터를 비교하였습니다.
 구글코랩으로 시도도 하였다가 파일을 경로지정하는 법을 모르겠어서
 포기하였습니다. 또한 재공해주는 데이터들이 무슨 역할을 하는지는
 알아냈어서 공부도 하였습니다.
 개발을 진행하면서 chat gpt 에게 많은 것을 물오보면서 진행했습니다.
 그러던중 plt.savefig(tes.png) 로 진행하면서 데이터를 저장하려했는데
 이게 여러사진이면 블루스크린이 뜨는걸 알았습니다.

```

def violinplot(df):
    for x in df.columns:
        sns.violinplot(x='Fire Alarm',y=x,data=df)
        plt.savefig(f'test1_{x}.png') # 각 그래프를 다른 파일로 저장
        plt.show()

```

violinplot(df)

다음은 제가 gpt 를통하여 바꾼 내용입니다.

```
# 필요한 라이브러리들을 임포트합니다.
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler

# CSV 파일을 DataFrame 으로 읽어옵니다.
df = pd.read_csv("smoke_detection_iot.csv")

# DataFrame 의 첫 부분을 출력하여 데이터를 확인합니다.
df.head()

# DataFrame 의 열(컬럼) 이름을 확인합니다.
df.columns

# DataFrame 의 행과 열의 개수를 확인합니다.
df.shape

# 'Unnamed: 0' 열을 삭제합니다.
df.drop(columns=['Unnamed: 0'], inplace=True)

# 수치형 데이터에 대한 기술 통계 정보를 출력합니다.
df.describe()

# 각 열에 대해 결측치의 개수를 확인합니다.
df.isnull().sum()

# 중복된 행의 개수를 확인합니다.
df.duplicated().sum()

# DataFrame 의 정보(열 데이터 타입, 비어있지 않은 값의 개수 등)를 출력합니다.
df.info()

# 각 열에 대해 고유한 값들의 개수와 실제 값들을 출력하는 함수를 정의합니다.
def uniquecounts(df):
    for x in df.columns:
        print(x, len(df[x].unique()), df[x].unique())

# 정의한 함수를 이용하여 DataFrame 의 고유한 값들을 출력합니다.
uniquecounts(df)

# DataFrame 의 마지막 부분을 출력하여 데이터를 확인합니다.
df.tail()
```

```

# 각 열들 간의 상관 관계를 계산합니다.
df.corr()

# 'TVOC[ppb]' 열의 각 값들의 개수를 확인합니다.
df['TVOC[ppb]'].value_counts()

# 각 특성(feature)에 대해 'Fire Alarm'에 따른 분포를 바이올린 플롯으로 시각화합니다.
def violinplot(df):
    for x in df.columns:
        sns.violinplot(x='Fire Alarm', y=x, data=df)
        plt.savefig(f'test1_{x}.png') # 각 그래프를 다른 파일로 저장
    plt.show()

violinplot(df)

# 특성과 타겟 변수를 나누어줍니다.
X = df.drop('Fire Alarm', axis=1) # 특성
y = df['Fire Alarm'] # 타겟 변수

# 데이터를 학습용과 평가용으로 분할합니다.
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
                                                    random_state=42)

# RandomForestClassifier 모델을 초기화하고 학습합니다.
model = RandomForestClassifier()
model.fit(X_train, y_train)

# 학습된 모델을 사용하여 평가용 데이터를 예측합니다.
y_pred = model.predict(X_test)

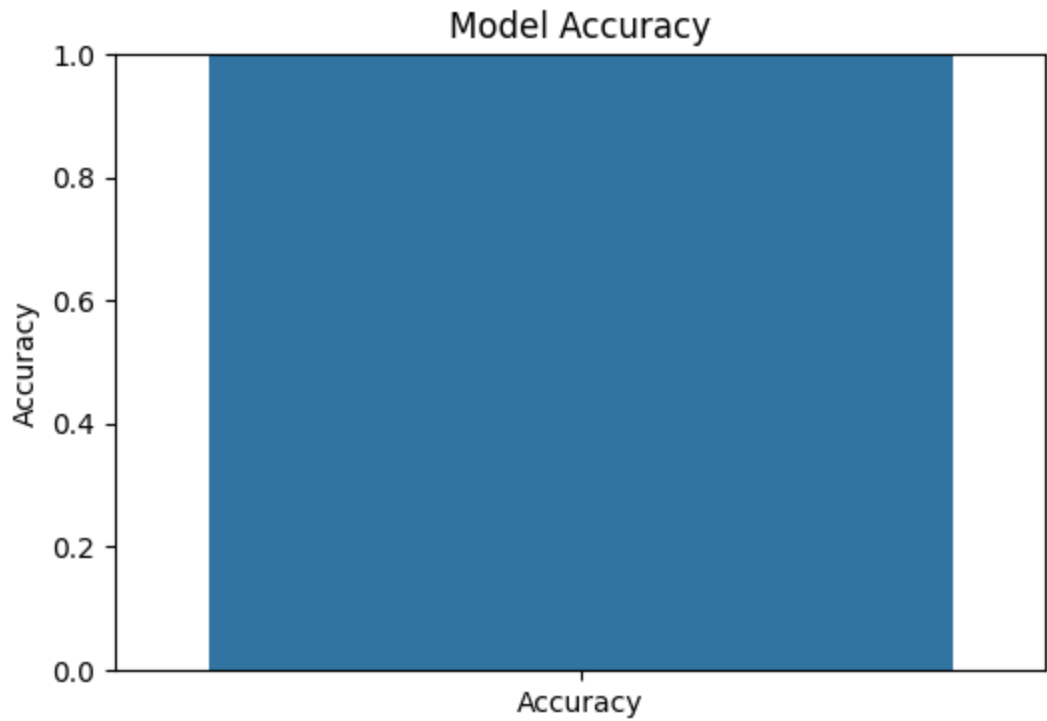
# 모델의 정확도를 계산합니다.
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# 모델의 정확도를 막대 그래프로 시각화합니다.
plt.figure(figsize=(6, 4))
sns.barplot(x=['Accuracy'], y=[accuracy])
plt.ylim(0, 1) # y 축 범위 설정 (0 에서 1 사이)
plt.title('Model Accuracy')
plt.ylabel('Accuracy')
plt.show()

```

각 함수별로 하는 역할들을 정리하였습니다.

결과값은 잘 나오는거 같습니다.



5. 개발 후기

- 수업을 들으면서 배우는것보다 직접 하면서 배우는게 얻어가는게 훨씬 많다는걸 다시 깨달았습니다.
- 구글링을 하면서 제일 어려웠던게 내가 모르는게 뭔지를 몰랐을 때였습니다. 내가 부족한 부분을 채워나가는 과제였습니다.
- 실제로 화재감지기를 iot 센서들로 데이터를 받아 학습을 시킨다면 정확도가 높은 화재감지기가 나올 거 같았습니다.