

typhoon_expected_model

학번 : 2018045

이름: 정태경

Github address: jtk0817

1. 운전자 모니터링 모델 개발의 목적

b. kaggle사이트에서 데이터를 기반으로 만들었다.

c. 위 코드에서는 태풍의 경로와 관련된 데이터를 활용하여 다양한 분석을 수행하고 있다. 여기서 사용된 독립 변수(Features)와 종속 변수(Target)는 다음과 같다.

독립 변수 (Features):

'Time of analysis': 분석 시간

'Latitude of the center': 태풍 중심의 위도

'Longitude of the center': 태풍 중심의 경도

'Central pressure': 태풍 중심의 중심 압력

'Maximum sustained wind speed': 최대 지속 풍속

'Direction of the longest radius of 50kt winds or greater': 50노트 이상의 최장 반경의 방향

'The longest radius of 50kt winds or greater': 50노트 이상의 최장 반경

'The shortest radius of 50kt winds or greater': 50노트 이상의 최단 반경

'Direction of the longest radius of 30kt winds or greater': 30노트 이상의 최장 반경의 방향

'The longest radius of 30kt winds or greater': 30노트 이상의 최장 반경

'The shortest radius of 30kt winds or greater': 30노트 이상의 최단 반경

'Indicator of landfall or passage': 상륙 또는 통과 지표

종속 변수 (Target): 'Grade': 태풍 등급

이러한 데이터를 사용하여 태풍의 등급을 예측하는 것으로 보입니다.
태풍 등급을 종속 변수로 보고, 해당 등급을 예측하기 위해 태풍의
경로와 관련된 다양한 정보를 활용하고 있다.

2. 안전 관련 머신러닝 모델의 네이밍의 의미

a. 태풍이 일어난 데이터를 통해서 태풍이 어디서 일어날지
예상하는 머신러닝이라 말그대로 `typhoon_expected_model`이라 칭한다.

3. 개발 계획

a. 데이터 요약:

`df0.info()`: 데이터프레임의 요약 정보를 출력합니다. 열 이름, 데이터
타입, 누락된 값 등을 확인할 수 있다.

`print(df0.columns.tolist())`: 데이터프레임의 열 목록을 출력한다.

지도 시각화:

각 연도별로 태풍의 경로를 지도 상에 표시한다.

`GeoDataFrame`, `geopandas`를 사용하여 태풍 경로를 지도에 표현한다.

`world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))`

코드는 세계 지도를 가져오는데 사용되었다.

그래프 시각화:

태풍의 중심 압력 및 최대 지속 풍속과 같은 수치형 데이터에 대한 막대
그래프를 출력한다.

태풍의 중심 압력과 같은 값들을 막대 그래프로 시각화하여 비교한다.

b. 전처리 계획:

데이터 불러오기: `pd.read_csv('파일경로')`: 태풍 관련 데이터 파일을

CSV 형식으로 불러옵니다.

데이터 확인:

`df0.info()`, `print(df0.columns.tolist())`: 데이터프레임의 기본 정보 및 열 목록을 확인하여 데이터 구조를 이해합니다.

날짜 데이터 처리:

'Time of analysis' 열에서 연도를 추출하여 'year' 열을 생성합니다.
'year' 열을 기준으로 필터링하여 연도별로 데이터를 분할합니다.

지리 정보 처리:

위도('Latitude of the center')와 경도('Longitude of the center') 데이터를 활용하여 지리적 위치를 나타내는 지오메트리 데이터를 생성합니다.

지오메트리 데이터를 활용하여 지도 상에 태풍의 경로를 시각화합니다.

결측치 처리:

필요한 열에서 결측치를 확인하고, 필요에 따라 결측치를 처리합니다.
(`dropna()`, `fillna()` 등)

수치 데이터 처리:

수치형 열의 분포를 확인하기 위해 히스토그램, 상자 그림 등을 사용하여 데이터를 시각화합니다.

수치형 열 간의 관계를 산점도 등을 통해 시각적으로 확인합니다.

범주형 데이터 처리:

범주형 열의 빈도를 확인하기 위해 카운트 플롯 등을 사용하여 데이터를 시각화합니다.

시각화 및 저장:

`plt.savefig()`를 사용하여 그래프를 이미지 파일로 저장합니다.

c. 태풍 등급을 예측하는 것이 목적인 머신러닝을 목적으로 하였다.

d. 북태평양 서부와 남중국해의 열대성 저기압에 대한 정보, 현재 및 예측 위치, 열대성 저기압의 이동 및 강도 등을 제공할 것이다.

4. 개발 과정

a.

1. 각 라이브러리들은 코드에서 특정 기능을 수행하기 위해 사용하기 위한 코드를 작성한다.

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import geopandas as gpd
from geopandas import GeoDataFrame
from shapely.geometry import Point
import seaborn as sns
```

2. 데이터 불러오기 및 전처리

```
# 파일 경로 변경
df0 = pd.read_csv('C:/Users/82105/Documents/GitHub/Homework02/RSMC_Best_Track_Data.csv')
df0['year'] = df0['Time of analysis'].apply(lambda x: int(x[0:4]))
```

외부 CSV 파일을 Pandas 데이터프레임으로 불러옵니다. 연도 정보 생성을 위해 시간 데이터에서 연도 정보를 추출하여 새로운 열('year')을 생성한다.

3.데이터탐색및시각화

```
for y in range(2020, 2024):
    df = df0[df0['year'] == y].copy()
    geometry = [Point(xy) for xy in zip(df['Longitude of the center'], df['Latitude of the center'])]
    gdf = GeoDataFrame(df, geometry=geometry)
    world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
    gdf.plot(ax=world.plot(figsize=(12, 12)), color='orange', markersize=10)
    plt.title(str(y))
    plt.show()
```

연도별 태풍 위치 시각화를 위한 연도별 태풍 위치를 지도 위에 표시하여 공간적 분포를 확인하기 위한 코드를 작성한다.

```
maxcols = ['Maximum sustained wind speed',
            'The longest radius of 50kt winds or greater',
            'The longest radius of 30kt winds or greater']
mincols = ['Central pressure']
print(len(maxcols))

for i in range(len(mincols)):
    df = df0[['Name of the storm', mincols[i]].dropna()
    data = df.sort_values(mincols[i], ascending=True)[0:40]
    plt.figure(figsize=(15, 6))
    plt.title(mincols[i])
    g = sns.barplot(x=data['Name of the storm'], y=data[mincols[i]], color='seagreen', data=data)
    g.tick_params(axis='x', labelrotation=45)
    plt.show()
```

태풍 속성 분석 및 시각화를 위해 태풍의 최대 바람 속도, 중심 기압 등 속성을 분석하고 막대 그래프를 통해 태풍 간 비교를 시각화하기 위한 코드를 작성한다.

b. `pd.read_csv('파일경로')`

사용 용도: CSV 파일을 Pandas의 DataFrame 형식으로 읽어온다.

`df0['열이름'].apply(func)`

사용 용도: 특정 열에 함수를 적용하여 새로운 값을 생성하거나 변환한다.

`GeoDataFrame(df, geometry=geometry)`

사용 용도: Geopandas의 GeoDataFrame을 생성하여 지리적 데이터를 처리한다.

```
gdf.plot(ax=world.plot(figsize=(가로, 세로)), color='색상',  
markersize=크기)
```

사용 용도: 지리 데이터를 지도 위에 시각화합니다.

```
plt.show()
```

사용 용도: Matplotlib으로 생성한 그래프를 화면에 출력합니다.

```
sns.barplot(x='x축 데이터', y='y축 데이터', data=데이터)
```

사용 용도: Seaborn을 사용하여 막대 그래프를 생성합니다.

c. 파일 경로 오류: 파일을 찾을 수 없거나 경로가 잘못된 경우 오류가 발생할 수 있어 파일 경로를 정확하게 지정해야 한다.

데이터 타입 변환 오류: 날짜와 같은 데이터를 처리할 때 데이터 형식이 일치하지 않으면 오류가 발생하기 때문에 데이터 형 변환 시 주의가 필요하다.

데이터 결측치: 코드에서 데이터를 처리할 때, 결측치를 제거하거나 적절하게 처리하지 않으면 오류가 발생할 수 있다.

시각화 오류: 지도 데이터를 시각화하거나 그래프를 그릴 때 데이터의 형식이나 변수를 잘못 사용하면 오류가 발생할 수 있다.

라이브러리 버전 호환성: 사용하는 라이브러리의 버전이 호환되지 않거나 각 라이브러리 간의 충돌이 발생할 수 있다.

d. 정확성은 csv파일을 불러와 그 파일에 대한 정보를 막대그래프와 지도 점 그래프로 나타내 정확성이 좋다고 생각한다.

e. github에 올려두었습니다.

5. 개발후기

만들면서 다시한번 컴퓨터 프로그래밍의 세계는 어려우면서도 다양한 분야에서 많은 부분을 차지한다고 느꼈고 1년동안 멍청한 저같은 놈들 알려주신다고 고생많으셨습니다. 정말 수업을 교수와 학생이 아니라 사람과 사람 관계느낌으로 알려주신 느낌이 들어서 좋았습니다. 정말 고생하셨습니다!!