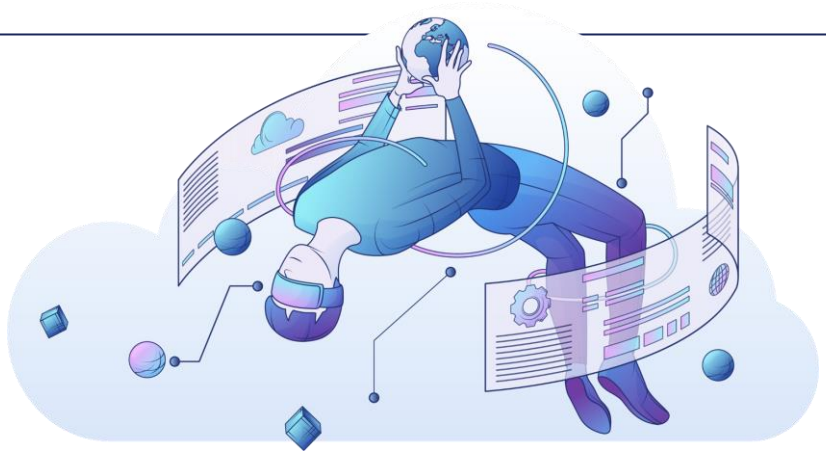


일일 태양 흑점 수 예측

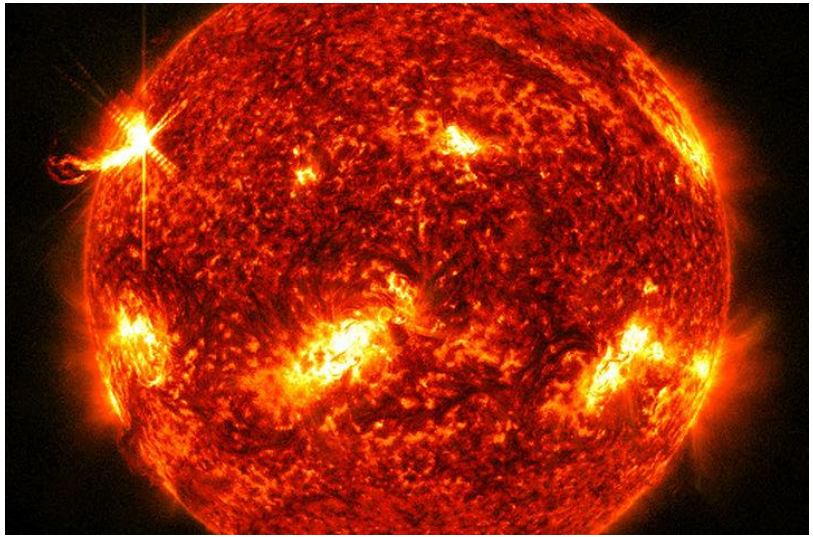
메카트로닉스 공학 전공 2017108071 부준호

목차

01

개요 및 필요성

개요 및 필요성

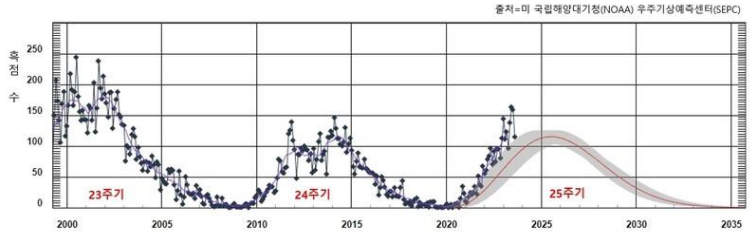


2023년 1월 태양에서 발생한 태양 플레어 현상. (출처: NASA)

흑점이 왜 중요한가?

- 태양 활동이 활발해질수록 지구에 당도하는 태양 에너지 입자가 증가해 통신과 전력망, 항공기 운항 시스템, 우주선 등에 장애를 일으킬 수 있기 때문이다.

2000년 이후 태양 활동 주기와 흑점 수 변화

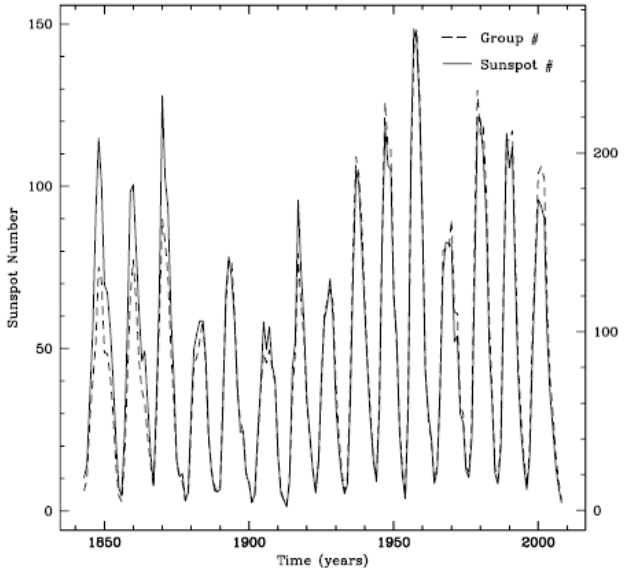


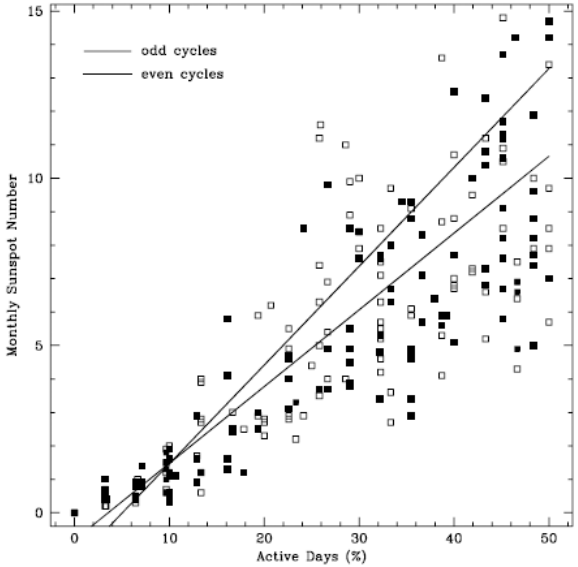
02

관련 연구 내용

관련 연구 내용

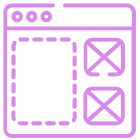
논문 이름 : Active Days around Solar Minimum and Solar Cycle Parameter
저자 : 장헌영(경북대학교 천문대기과학과)





H.-Y. Chang, “Active Days around Solar Minimum and Solar Cycle Parameter,” *Journal of Astronomy and Space Sciences*, vol. 38, no. 1. The Korean Space Science Society, pp. 23–29, Mar-2021.

관련 연구와 차이점



본 연구와 차이점1

- 관련 연구에 사용된 일일 태양 흑점 데이터는 1843년부터 2008년까지의 과거 데이터를 가지고 연구 했다. 하지만 본 연구에서는 1850년 부터 2023년 8월 31일까지의 최신 데이터를 가지고 일일 태양 흑점을 예측한다.

본 연구와 차이점2

- 관련 연구는 태양 자기 매개변수와 월간 흑점 수의 선형 기울기 사이의 관계를 통계적으로 탐구하는 주제를 가지고 연구했다면, 본 연구는 미래 특점 시점에 발생한 태양 흑점수를 예측하는 연구이다.

03

내용 요약

내용 요약

- 태양 활동이 활발해질수록 지구에 당도하는 태양 에너지 입자가 증가해 통신과 전력망, 항공기 운항 시스템, 우주선 등에 장애를 일으킬 수 있다. 연구를 위해 1850년부터 2023년 8월까지 일일 태양 흑점 수를 기록한 데이터를 기반으로 했다. 본 연구에서는 다양한 시계열 예측 모델과 머신러닝 모델을 통해 미래 태양 흑점 수를 예측하고, 시각화 한다.

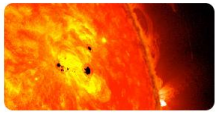
04

데이터 설명

데이터 설명

Daily Sunspots Dataset (1850 - 2023)

Can you forecast the number of sunspot?



- 브뤼셀의 벨기에 왕립천문대(SILSO)에서 제공한 1850년부터 2023년 8월 까지의 일일 태양흑점 데이터

df

	date	year	month	day	date_frac	counts	std	nobs	indicator
0	1850-01-01	1850	1	1	1850.001	253	19.6	1	NaN
1	1850-01-02	1850	1	2	1850.004	162	15.5	1	NaN
2	1850-01-03	1850	1	3	1850.007	217	18.1	1	NaN
3	1850-01-04	1850	1	4	1850.010	99	12.0	1	NaN
4	1850-01-05	1850	1	5	1850.012	108	12.6	1	NaN
...	...	...	...	...	...	...	...	...	...
63425	2023-08-27	2023	8	27	2023.653	84	10.5	31	*
63426	2023-08-28	2023	8	28	2023.656	84	12.3	36	*
63427	2023-08-29	2023	8	29	2023.659	88	11.8	28	*
63428	2023-08-30	2023	8	30	2023.662	104	14.8	32	*
63429	2023-08-31	2023	8	31	2023.664	90	11.5	37	*

63430 rows × 9 columns

https://www.sidc.be/SILSO/datafiles

Column 설명

- Date : 날짜
- Year : 년도
- Month : 월
- Day : 일
- Date_frac : 연도의 분수
- Counts : 일일 총 흑점 수
- Std : 흑점 수의 일일 표준 편차
- Nobs : 일일 값을 계산하는데 사용된 관측치 수
- Indicator : 확정적/잠정적 지표

```
df.dtypes
```

```
date          object
year           int64
month          int64
day            int64
date_frac    float64
counts         int64
std          float64
nobs           int64
indicator     object
dtype: object
```

05

데이터 전처리

데이터 결측치 분석

```
df.isnull().sum()
```

```
date            0
year            0
month           0
day             0
date_frac       0
counts          0
std             0
nobs            0
indicator    63277
dtype: int64
```

- isnull().sum() Method를 통해 해당 데이터에 예측에 필요 없는 결측치가 있는지 확인한다.

데이터 전처리

```
df=df.drop(columns = ["date_frac","std","nobs","indicator"])
df
```

	date	year	month	day	counts
0	1850-01-01	1850	1	1	253
1	1850-01-02	1850	1	2	162
2	1850-01-03	1850	1	3	217
3	1850-01-04	1850	1	4	99
4	1850-01-05	1850	1	5	108
...	...	...	...	...	...
63425	2023-08-27	2023	8	27	84
63426	2023-08-28	2023	8	28	84
63427	2023-08-29	2023	8	29	88
63428	2023-08-30	2023	8	30	104
63429	2023-08-31	2023	8	31	90

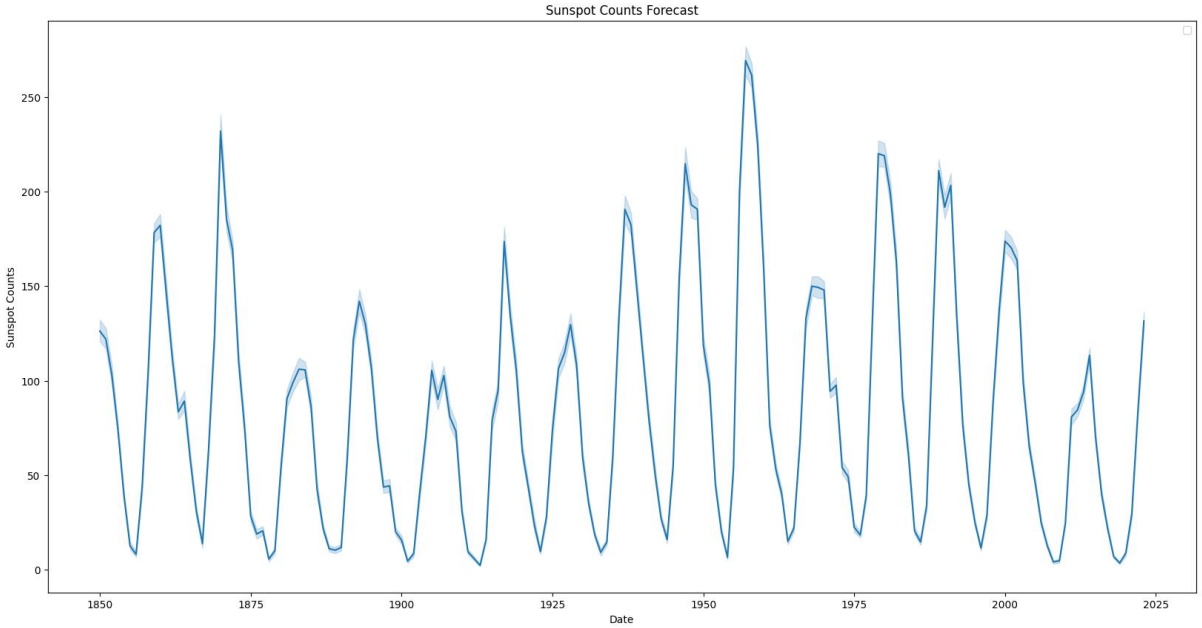
63430 rows × 5 columns

- 데이터 설명 페이지에서 나타난 Column 중 예측에 필요 없는 Column인 "date_frac","std","nobs","indicator"를 drop() Method를 통해 제거한다.

06

시각화 / 분석

시각화/분석



- 전처리가 완료된 데이터와 시각화 모듈인 “seaborn”을 가지고 시각화한 모습

07

데이터 분할

데이터 분할

```
train, test = train_test_split(df,train_size = 0.8)

train_X = train[['year','month','day']]
train_y = train.counts

test_X = test[['year','month','day']]
test_y = test.counts
```

- 예측을 위한 train과 test 데이터를 분할합니다.

08

학습 및 테스트 결과

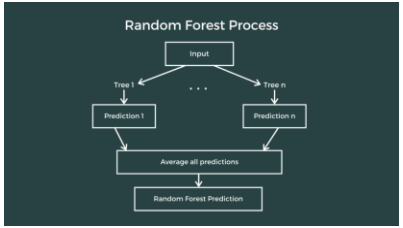
사용 모델 소개

Prophet

Time-series를 다루기 위해서 만든 Library이다. 통계적인 지식 없이 Time-series 데이터를 기반으로 자동으로 Forecast를 수행해주며, 아웃라이어, 데이터 부재 등에도 비교적 강건하게 모델링을 수행하는 장점이 있다.

ARIMA

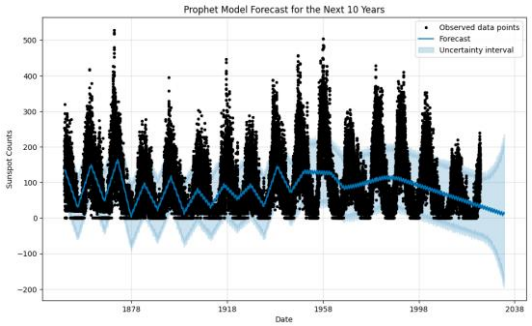
데이터의 현제 값과 이전 값의 차이, 이전 예측 오차의 평균을 고려하여 미래 값을 예측하는 Library

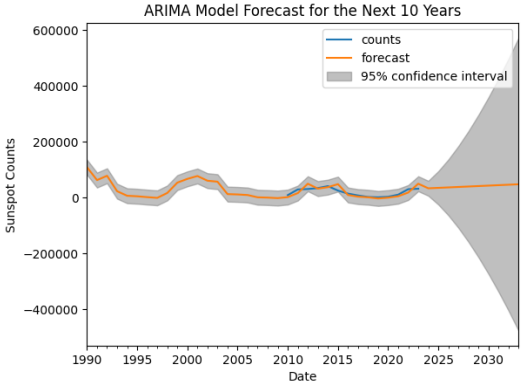


Random Forest Regressor

여러 트리들이 예측을 수행하는데, 각 트리는 다양한 특성을 고려하여 예측하고, 이러한 다양성을 통해 모델은 안정적이고 강력한 예측을 할 수 있게 된다.

모델 평가






```
n_estimators =  5  인식률 : 90.476
Root Mean Squared Error (RMSE): 23.763494287001688
n_estimators =  10  인식률 : 91.622
Root Mean Squared Error (RMSE): 22.287096413879578
n_estimators =  20  인식률 : 92.061
Root Mean Squared Error (RMSE): 21.695539510516273
n_estimators =  30  인식률 : 92.308
Root Mean Squared Error (RMSE): 21.355308834293673
n_estimators =  40  인식률 : 92.394
Root Mean Squared Error (RMSE): 21.23609371696747
n_estimators =  100  인식률 : 92.612
Root Mean Squared Error (RMSE): 20.92856103385554
n_estimators =  150  인식률 : 92.654
Root Mean Squared Error (RMSE): 20.86983224106198
n_estimators =  200  인식률 : 92.607
Root Mean Squared Error (RMSE): 20.936800107013276
n_estimators =  250  인식률 : 92.631
Root Mean Squared Error (RMSE): 20.902328897624642
n_estimators =  300  인식률 : 92.628
Root Mean Squared Error (RMSE): 20.90600111089442
n_estimators =  350  인식률 : 92.643
Root Mean Squared Error (RMSE): 20.885887398464586
```

Prophet

ARIMA

Random Forest Regressor

Prophet

```python
import pandas as pd
from prophet import Prophet
import matplotlib.pyplot as plt

# 데이터프레임을 Prophet 모델에 맞게 변환
df_prophet = df[['date', 'counts']].rename(columns={'date': 'ds', 'counts': 'y'})

# Prophet 모델 생성
model_prophet = Prophet()

# 모델 학습
model_prophet.fit(df_prophet)

# 미래 10년(3650일)에 대한 데이터프레임 생성
future_prophet = model_prophet.make_future_dataframe(periods=3650)

# 예측 수행
forecast_prophet = model_prophet.predict(future_prophet)

# 예측 결과 시각화
fig_prophet = model_prophet.plot(forecast_prophet)
plt.title('Prophet Model Forecast for the Next 10 Years')
plt.legend()
plt.xlabel('Date')
plt.ylabel('Sunspot Counts')
plt.show()
```

ARIMA

```
import pandas as pd
from statsmodels.tsa.arima.model import ARIMA
import matplotlib.pyplot as plt
from statsmodels.graphics.tsaplots import plot_predict

dta = df_year_month_counts
dta.index = pd.date_range(start='1850', end='2024', freq='A')
res = ARIMA(dta, order=(0,2,0)).fit()
fig, ax = plt.subplots()
ax = dta.loc['1950':].plot(ax=ax)
plot_predict(res, '1950', '2033', ax=ax)

plt.title('ARIMA Model Forecast for the Next 10 Years')
plt.xlabel('Date')
plt.ylabel('Sunspot Counts')
plt.legend()
plt.show()
```

Random Forest Regressor

```
from math import sqrt
for i in (5,10,20,30,40,100,150,200,250,300,350):
    rf = RandomForestRegressor(n_estimators=i ,n_jobs=-1)
    rf.fit(train_X, train_y)
    result = rf.predict(test_X)
    rf_rate = rf.score(test_X,test_y) * 100
    print("n_estimators = ",i, " 인식률 : {0:.3f}".format(rf_rate))
    rmse = sqrt(mean_squared_error(test_y, result))
    print(f'Root Mean Squared Error (RMSE): {rmse}')
```

09

결론 및 소감

결론

- 태양 흑점 수 예측 가능 -> 태양풍 강도 예측 가능
- 태양 활동 주기 예측 가능

소감

- 비록 Prophet 모델과 ARIMA모델 결과가 인상적이게 나오지 못했지만, 강의시간에 배우지 못한 모델들을 사용해보고 평가하고, 결과를 도출해 나가면서 인공지능의 재미를 맛볼 수 있었고, 더 나아가 Pycaret Library을 통해 더 많은 모델들을 비교해보고 싶다.

- 또한, 데이터 정제 과정에서 전처리 과정과 시각화 부분에서 내 실력의 부족함을 깨달았으며, 부족한 부분은 추후 데이터 전처리와 시각화 등 추가적인 공부를 통해 메워나갈 예정이다.

QnA

저자 : 부준호
bm1549@naver.com
010-9419-1549