EXPERIMENT 7

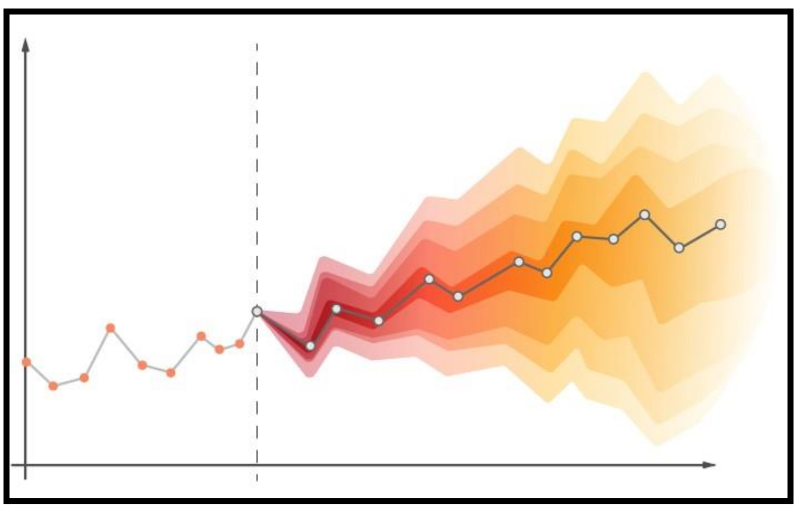
Aim: Implement time series forecasting.(example Weather data ,Rainfall measurements)

Theory:

Time series forecasting is a critical tool for decision-making and planning across various industries. By leveraging historical data patterns, time series forecasting enables organizations to anticipate future trends and make proactive decisions. Let's delve deeper into the nuances of time series forecasting:

Types of Time Series Forecasting Methods:

1. **Statistical Methods:**
   - ❖ **ARIMA** (Autoregressive Integrated Moving Average): A popular method for modeling and forecasting time series data based on its autocorrelation and moving average components.
   - ❖ **Exponential Smoothing:** A family of methods that assign exponentially decreasing weights to past observations, with different variations like Simple Exponential Smoothing.
2. **Machine Learning Methods:**
   - ❖ **Random Forests:** Ensemble learning technique that builds multiple decision trees and combines their predictions to make accurate forecasts.
   - ❖ **Support Vector Machines (SVM):** Supervised learning algorithm that finds the hyperplane that best separates classes in high-dimensional space.
   - ❖ **Long Short-Term Memory (LSTM) Networks**: A type of recurrent neural network (RNN) designed to capture long-term dependencies in time series data.



Use Cases :

- **Weather Forecasting**: Predicting temperature, rainfall, wind speed, and other meteorological parameters.
- **Financial Forecasting**: Forecasting stock prices, exchange rates, economic indicators, and financial market trends.
- **Demand Forecasting:** Predicting sales, inventory levels, customer demand, and supply chain requirements.
- Energy Forecasting: Forecasting energy demand, renewable energy production, energy prices, and consumption patterns.

Advantages of Time Series Forecasting:

- **Informed Decision Making:** Provides insights into future trends, allowing organizations to make data-driven decisions and strategic plans.
- **Proactive Measures:** Enables proactive measures and resource allocation to meet anticipated demand or mitigate potential risks.
- **Operational Efficiency:** Improves operational efficiency and cost optimization by optimizing inventory levels, production schedules, and resource utilization.

Disadvantages:

- **Data Quality Dependence:** Accuracy of forecasts heavily relies on the quality and quantity of historical data available.
- **Vulnerability to Unexpected Events:** Unexpected events or structural changes in the underlying data-generating process can lead to inaccurate forecasts.

- **Computational Complexity:** Complex forecasting models may be computationally expensive and require significant parameter tuning and optimization.

Code :

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.statespace.sarimax import SARIMAX
from sklearn.metrics import mean_squared_error
# Load the dataset
try:
data = pd.read_csv('london_weather.csv', parse_dates=['date'], index_col='date')
except FileNotFoundError:
print("Error: The file 'london_weather.csv' not found.")
exit()
print(data.head())
plt.figure(figsize=(10, 6))
plt.plot(data.index, data['mean_temp'], label='Mean Temperature')
plt.title('London Weather Data')
plt.xlabel('Date')
plt.ylabel('Mean Temperature')
plt.legend()
plt.show()
train_data = data.loc[:'2010-12-31', 'mean_temp'] # Data up to 2010 for training
test_data = data.loc['2011-01-01':, 'mean_temp'] # Data from 2011 onwards for testing
order = (1, 1, 1) # ARIMA order
seasonal_order = (1, 1, 1, 12) # Seasonal order
# Fit SARIMA model
try:
model = SARIMAX(train_data, order=order, seasonal_order=seasonal_order)
result = model.fit(disp=False)
except Exception as e:
print("Error:", e)
exit()
forecast_steps = 12
forecast = result.get_forecast(steps=forecast_steps)
plt.figure(figsize=(10, 6))
plt.plot(train_data.index, train_data, label='Training Data', color='blue')
plt.plot(test_data.index, test_data, label='Test Data', color='green')
plt.plot(forecast.predicted_mean.index, forecast.predicted_mean, color='red', label='Forecast')
plt.fill_between(forecast.predicted_mean.index, forecast.conf_int().iloc[:, 0],
```

```
forecast.conf_int().iloc[:, 1], color='pink', alpha=0.3)
# Add labels and legend
plt.title('London Weather Forecast (SARIMA)')
plt.xlabel('Date')
plt.ylabel('Mean Temperature')
plt.legend()
plt.show()
```

Output :

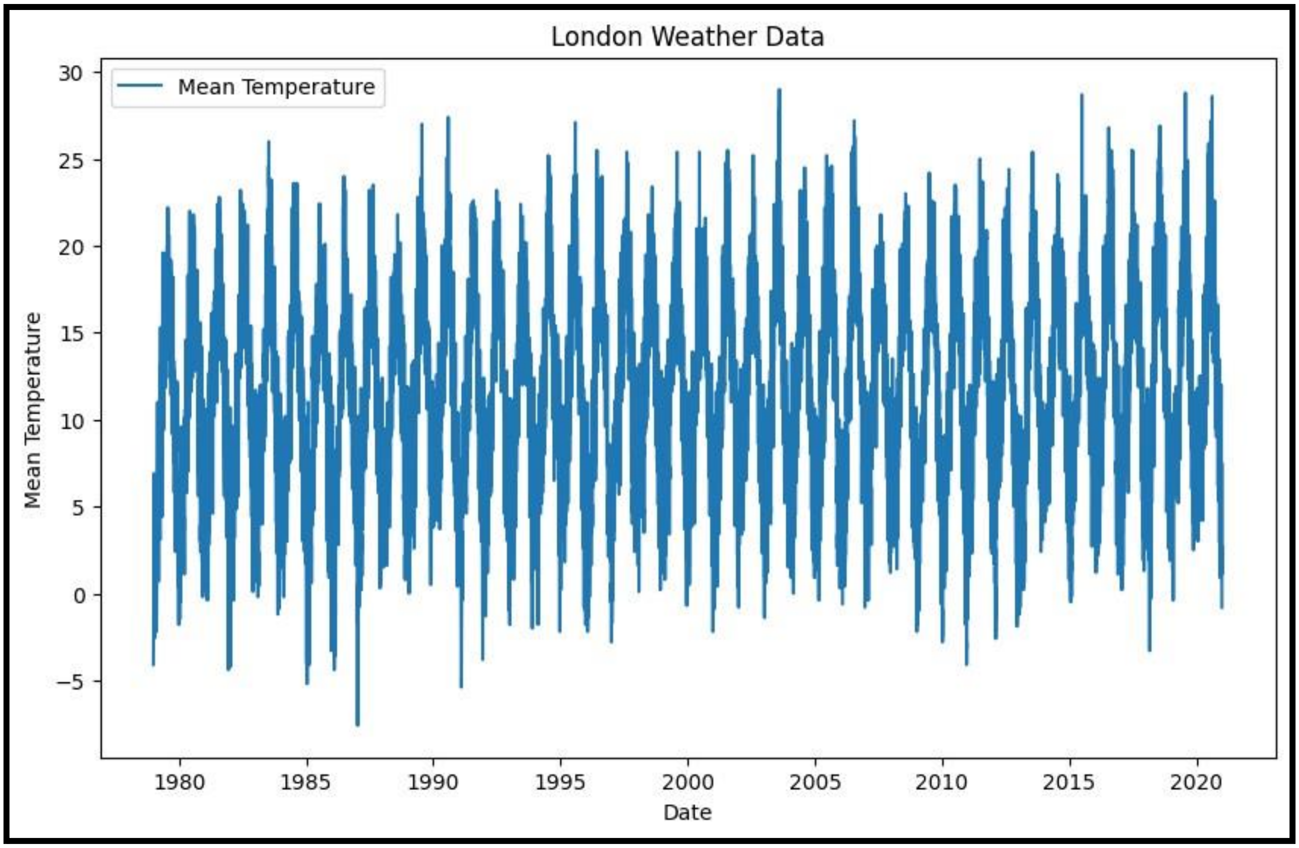
Dataset

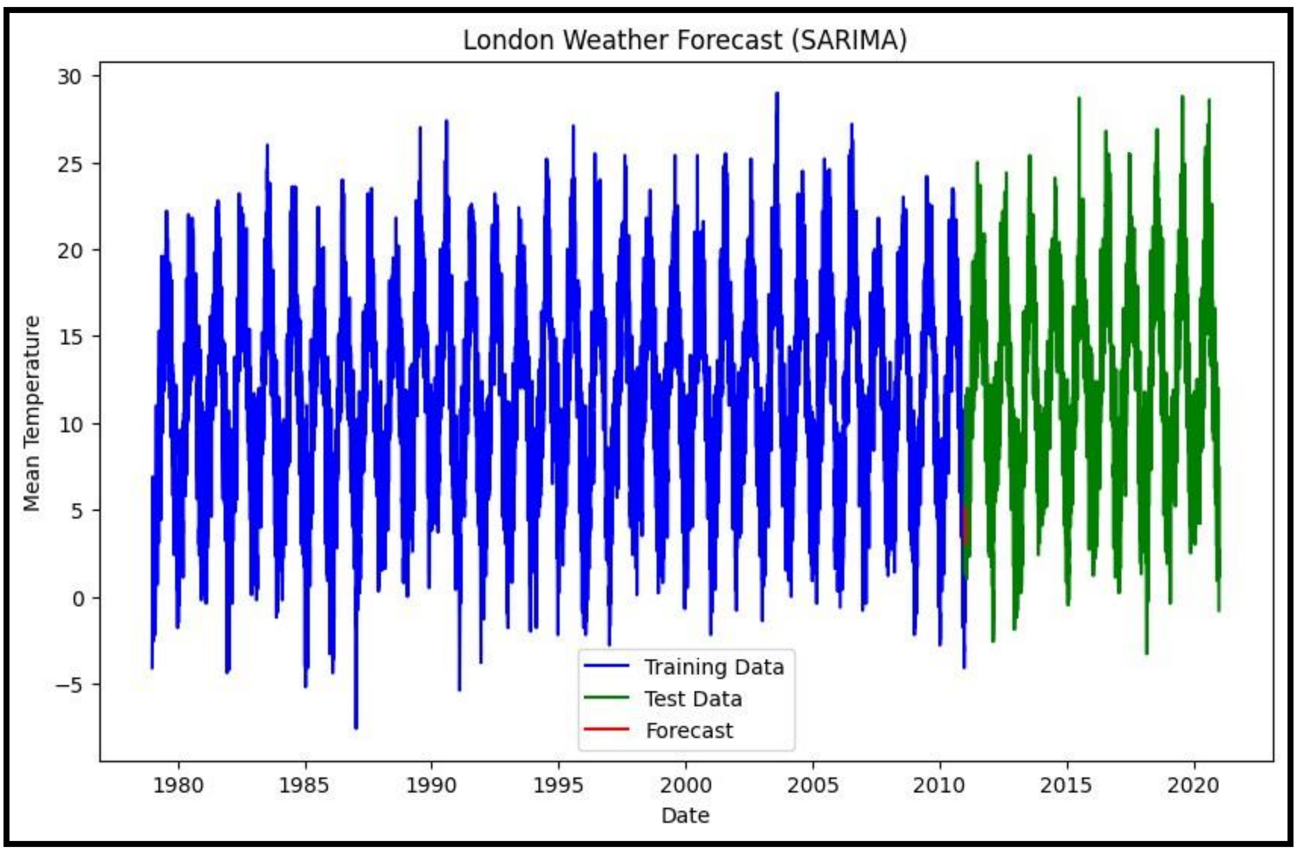
P19

	A	B	C	D	E	F	G	H	I	J
1	date	cloud_cove	sunshine	global_rad	max_temp	mean_tem	min_temp	precipitatio	pressure	snow_dept
2	19790101	2.0	7.0	52.0	2.3	-4.1	-7.5	0.4	101900.0	9.0
3	19790102	6.0	1.7	27.0	1.6	-2.6	-7.5	0.0	102530.0	8.0
4	19790103	5.0	0.0	13.0	1.3	-2.8	-7.2	0.0	102050.0	4.0
5	19790104	8.0	0.0	13.0	-0.3	-2.6	-6.5	0.0	100840.0	2.0
6	19790105	6.0	2.0	29.0	5.6	-0.8	-1.4	0.0	102250.0	1.0
7	19790106	5.0	3.8	39.0	8.3	-0.5	-6.6	0.7	102780.0	1.0
8	19790107	8.0	0.0	13.0	8.5	1.5	-5.3	5.2	102520.0	0.0
9	19790108	8.0	0.1	15.0	5.8	6.9	5.3	0.8	101870.0	0.0
10	19790109	4.0	5.8	50.0	5.2	3.7	1.6	7.2	101170.0	0.0
11	19790110	7.0	1.9	30.0	4.9	3.3	1.4	2.1	98700.0	0.0
12	19790111	1.0	6.8	55.0	2.9	2.6	0.3	2.3	98960.0	0.0
13	19790112	3.0	6.4	54.0	2.0	0.4	-2.0	0.0	100650.0	1.0
14	19790113	1.0	7.0	57.0	4.3	-2.6	-7.1	0.0	102350.0	1.0
15	19790114	7.0	0.0	14.0	6.7	-0.6	-5.6	0.8	102700.0	1.0
16	19790115		0.0	15.0	5.9	3.8	1.0	0.1	102990.0	0.0
17	19790116	8.0	0.0	15.0	2.6	5.0	4.1	3.9	103100.0	0.0
18	19790117	8.0	0.0	15.0	1.9	2.1	1.6	2.5	102220.0	0.0
19	19790118	8.0	0.0	15.0	3.0	0.8	-0.2	0.2	101860.0	0.0
20	19790119	8.0	0.0	16.0	7.2	0.8	-1.4	5.2	100910.0	0.0
21	19790120	7.0	0.0	16.0	3.5	3.1	-1.0	0.0	100920.0	0.0

```
            cloud_cover  sunshine  global_radiation  max_temp  mean_temp  \
date
1979-01-01          2.0       7.0              52.0       2.3       -4.1
1979-01-02          6.0       1.7              27.0       1.6       -2.6
1979-01-03          5.0       0.0              13.0       1.3       -2.8
1979-01-04          8.0       0.0              13.0      -0.3       -2.6
1979-01-05          6.0       2.0              29.0       5.6       -0.8

            min_temp  precipitation  pressure  snow_depth
date
1979-01-01      -7.5            0.4  101900.0         9.0
1979-01-02      -7.5            0.0  102530.0         8.0
1979-01-03      -7.2            0.0  102050.0         4.0
1979-01-04      -6.5            0.0  100840.0         2.0
1979-01-05      -1.4            0.0  102250.0         1.0
```


London Weather Data
Mean Temperature
Mean Temperature
Date
30
25
20
15
10
5
0
−5
1980
1985
1990
1995
2000
2005
2010
2015
2020


London Weather Forecast (SARIMA)
Training Data
Test Data
Forecast
Mean Temperature
Date
30
25
20
15
10
5
0
−5
1980
1985
1990
1995
2000
2005
2010
2015
2020