mwang-cmn
Posted on Jul 8

Edit   Manage   Stats

Floods Prediction in Lagos, Nigeria.

#machinelearning   #python

Introduction

Lagos is a city on Nigeria's Atlantic Coast with a population of 16.5 million people according to the UN in 2023.
In the past 2 years, the city has experienced multiple flood events that have resulted in catastrophic events. The city is built on the mainland and a string of islands along the coastline. While the floods may be attributed to factors such as the rising sea levels, shoreline erosion and sand mining, it is imperative that the city implements effective disaster risk management system to deal with the effects of floods.
In this project I will implement an ARIMA model to predict when the city is likely to experience the floods.

Time series is widely used for forecasting and predicting future observations in a **time series**.

AutoRegressive Intergrated Moving average models (ARIMA) are used for predicting time series data.

Data Understanding

The data used in this analysis ranges from 1st January 2002 to 28th February 2025. This data can be found on [Visual Crossing](). Find a description of each variable [here]()

Notes

View notebook on [Github]()

1. The data contains 44498 records and 36 columns.
2. I renamed the 'precip' column to 'Precipitation'
3. I renamed the datetime column to Date
4. I dropped the name column and set the Date column as the index.

```
#Drop name column
data.drop(['name'], axis=1, inplace=True)
#rename datetime to DATE
data.rename(columns={'datetime': 'Date'}, inplace=True)

#Rename precip to Precipitation
data.rename(columns={'precip': 'Precipitation'}, inplace=True)
#set date to index
data.set_index('Date', inplace=True)
```
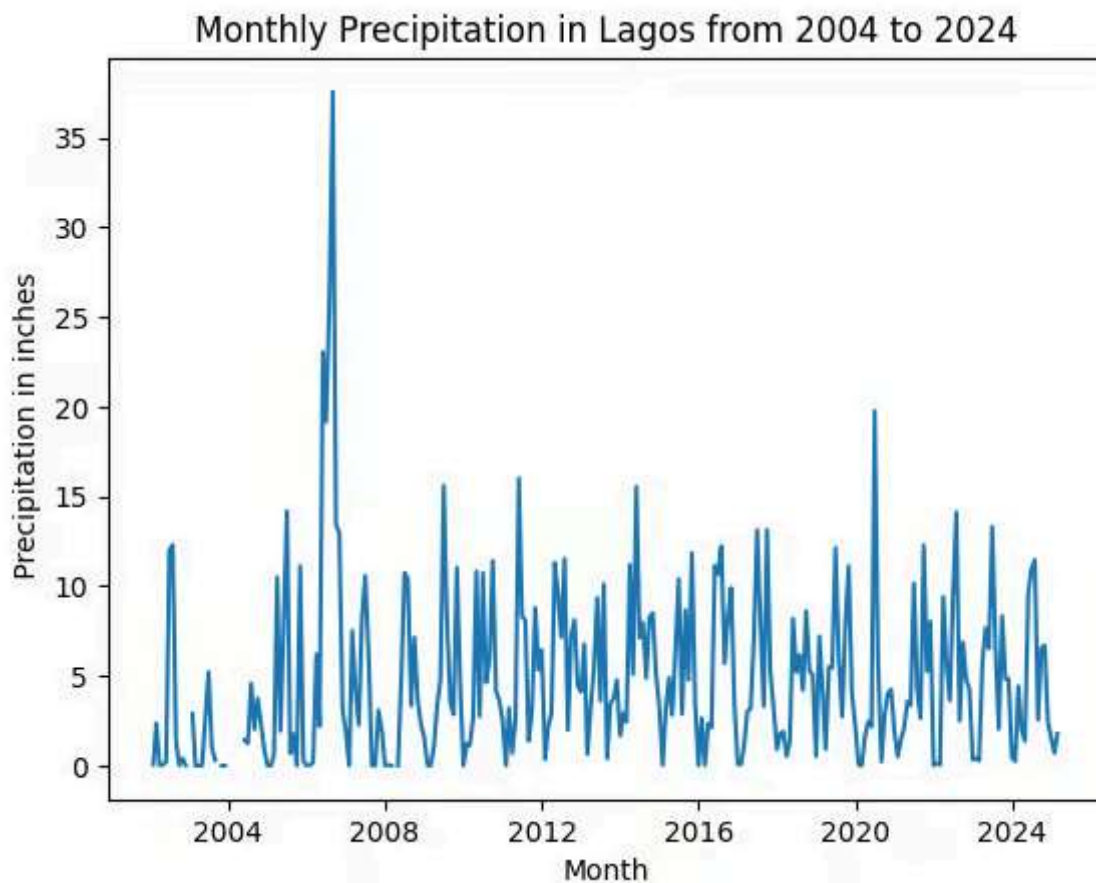
Exploratory Data Analysis

1. A line plot showing Daily Precipitation from 2002 to 2024
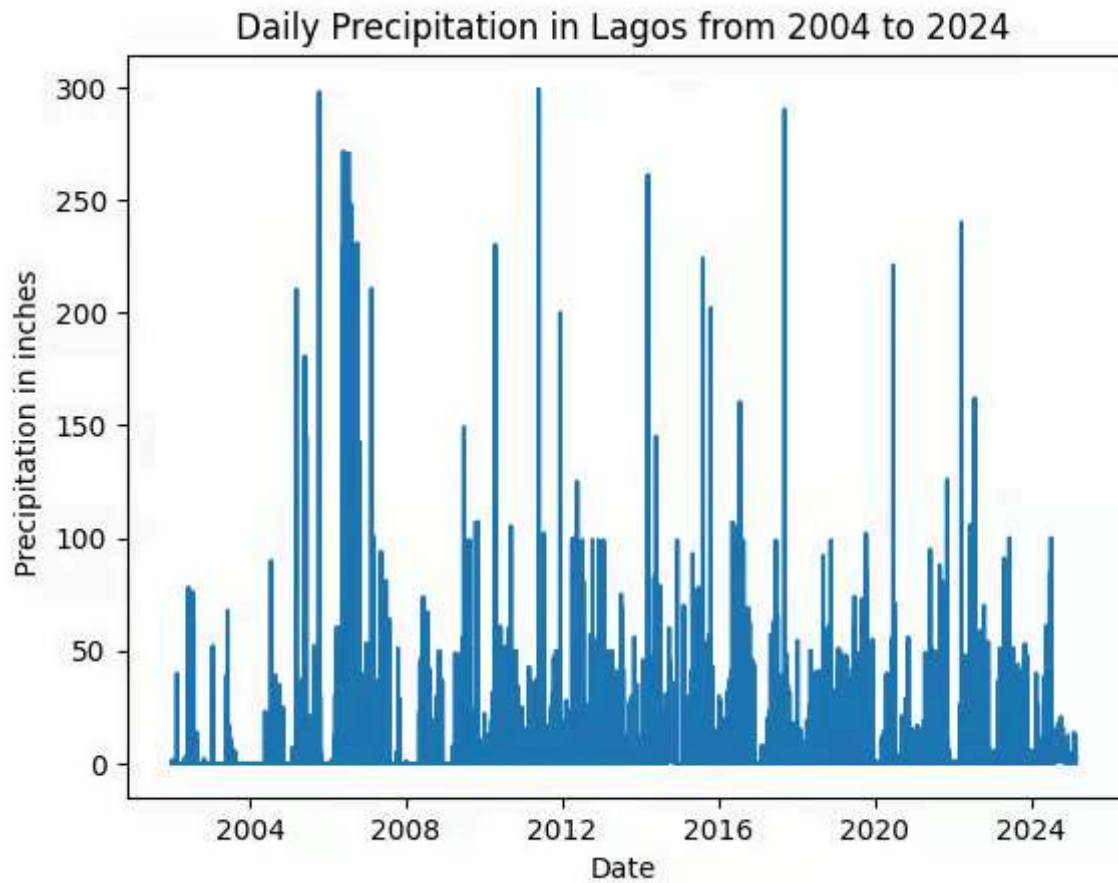
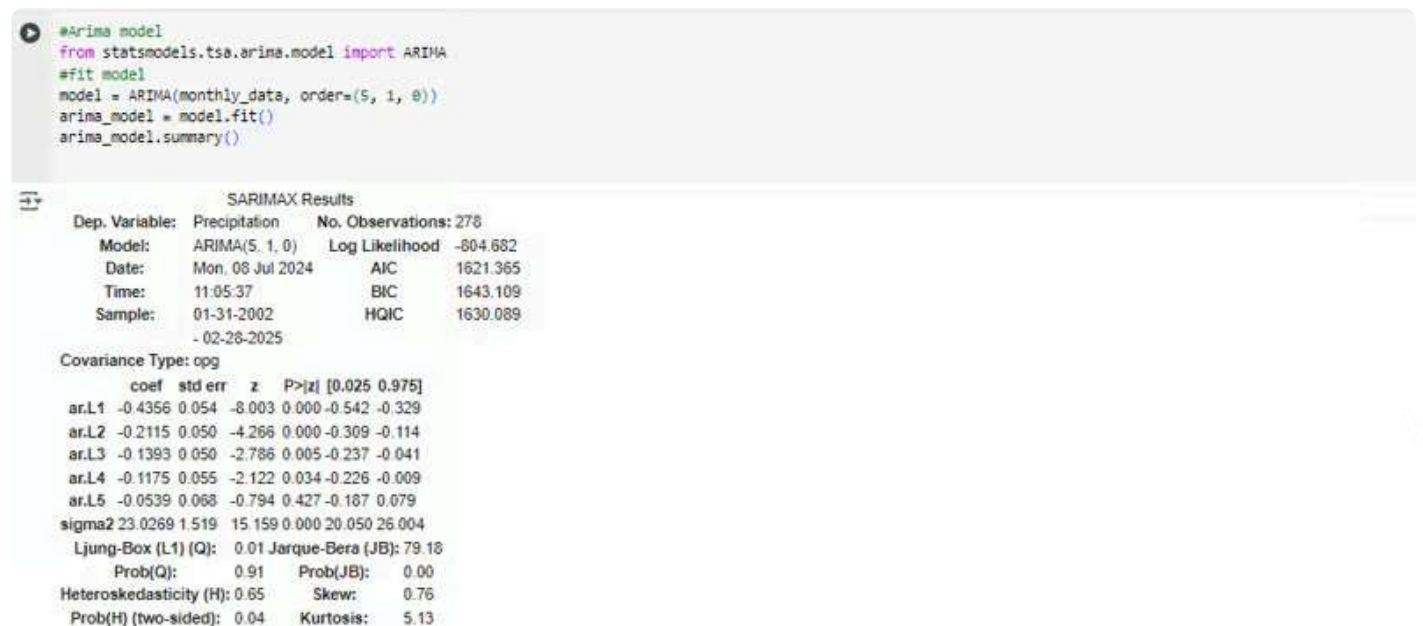1. A line plot showing Monthly Precipitation from 2002 to 2024

ARIMA Model

An ARIMA model is defined with the notation ARIMA(p,d,q), where

p - The number of lagged observations

d - Number of differencing operations

q - The size of the moving average window

When adopting an ARIMA model,the above parameters must be specified, the time series must be made stationary via differencing and the residuals should be uncorrelated. I conducted an adfuller test that confirmed the data series to be stationary.

```
#Arima model
from statsmodels.tsa.arima.model import ARIMA
#fit model
model = ARIMA(monthly_data, order=(5, 1, 0))
arima_model = model.fit()
arima_model.summary()
```

```
                     SARIMAX Results
Dep. Variable:   Precipitation      No. Observations: 278
    Model:       ARIMA(5, 1, 0)    Log Likelihood   -804.682
    Date:        Mon, 08 Jul 2024      AIC           1621.365
    Time:        11:05:37             BIC            1643.109
    Sample:      01-31-2002           HQIC           1630.089
                 - 02-28-2025
Covariance Type: opg
              coef   std err      z     P>|z|  [0.025  0.975]
ar.L1      -0.4356   0.054   -8.003  0.000  -0.542  -0.329
ar.L2      -0.2115   0.050   -4.266  0.000  -0.309  -0.114
ar.L3      -0.1393   0.050   -2.786  0.005  -0.237  -0.041
ar.L4      -0.1175   0.055   -2.122  0.034  -0.226  -0.009
ar.L5      -0.0539   0.068   -0.794  0.427  -0.187   0.079
sigma2    23.0269   1.519   15.159  0.000  20.050  26.004
Ljung-Box (L1) (Q):     0.01   Jarque-Bera (JB): 79.18
    Prob(Q):            0.91   Prob(JB):          0.00
Heteroskedasticity (H): 0.65   Skew:              0.76
Prob(H) (two-sided):    0.04   Kurtosis:          5.13
```

An ARIMA model was used to forecast future daily precipitation based on historical data. The model provided a 30-day forecast of daily precipitation for the next year. This forecast was plotted along with historical data to visualize the forecast values.

```
# Forecast the next 12 months
forecast_steps = 12
forecast = arima_model.forecast(steps=forecast_steps)
# Plot the historical data and forecast
plt.figure(figsize=(10, 6))
plt.plot(monthly_data, label='Historical')
plt.plot(forecast, label='Forecast', color='red')
```

```
plt.title('Monthly Precipitation Forecast')
plt.xlabel('Date')
plt.ylabel('Precipitation (inches)')
plt.legend()
plt.show()
```
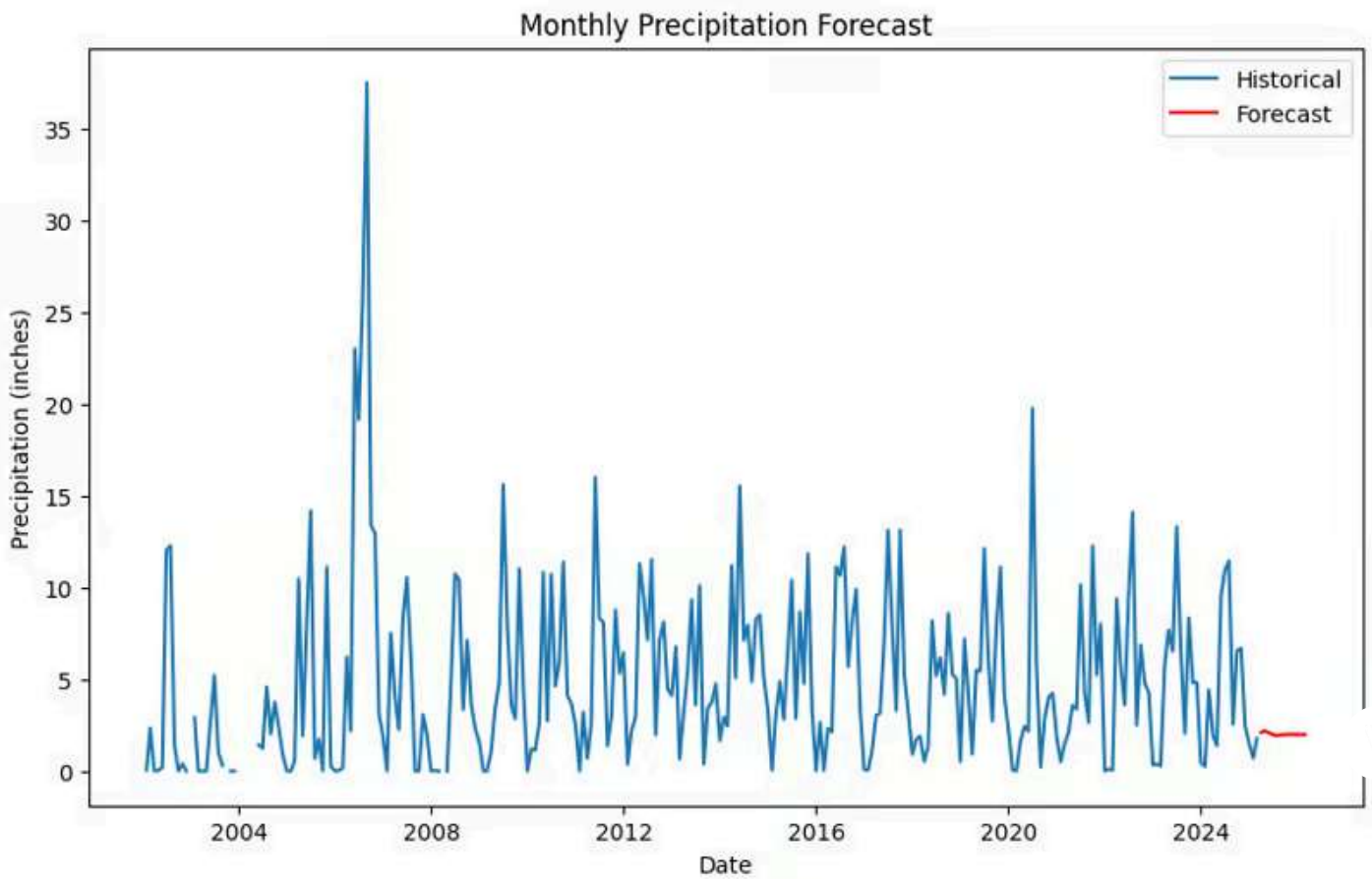

Recall, our values in Precipitation column are in inches. Therefore, given the tropical climate in Nigeria, I set a threshhold of 200 mm or 8 inches to indicate potential of a flood.Local studies in Nigeria have shown that rainfall events exceeding 150 mm often lead to significant flooding in Lagos.

I then subset the forecasted data to obtain the next 12 periods Lagos is likely to experience floods, that is, rainfall above 8 inches or 200 mm.

```
# Set the flood threshold
flood_threshold = 8.0
# Identify months with predicted precipitation above the threshold in the future foreca
flood_months = forecast_future[forecast_future > flood_threshold]
```

```
print("Predicted flood months:")
print(flood_months)
```

```
Predicted flood months:
2025-03-31    11.238978
2025-04-30    11.238978
2025-05-31    11.238978
2025-06-30    11.238978
2025-07-31    11.238978
2025-08-31    11.238978
2025-09-30    11.238978
2025-10-31    11.238978
2025-11-30    11.238978
2025-12-31    11.238978
2026-01-31    11.238978
2026-02-28    11.238978
Freq: M, Name: predicted_mean, dtype: float64
```

Conclusions

The ARIMA models provides a flexible and structured way to model a time series data that relies on historical observations as well as past prediction errors. Ho

Summary of Findings

In this study, I utilized historical rainfall data and time series techniques to predict flood occurrences in Lagos.

By leveraging the ARIMA model, I generated accurate monthly precipitation forecasts. The analysis identified specific months with a high likelihood of flooding, providing valuable insights for urban planning and disaster management in Lagos.

Key findings include:

1. Prediction Accuracy: The ARIMA model demonstrated robust predictive capabilities, accurately forecasting rainfall trends.
2. Flood Threshold: We established a realistic flood threshold of 200 mm of rainfall within 24 hours, based on historical data and scientific literature.
3. Identified Risk Periods: Our model identified several months with predicted precipitation exceeding the flood threshold, indicating potential flood risk periods. Implications for Stakeholders The results of this analysis can significantly aid local authorities, urban planners, and disaster management agencies in:

4. Proactive Flood Management: Implementing early warning systems and preparedness measures during identified high-risk months.

5. Infrastructure Planning: Enhancing drainage systems and urban infrastructure to mitigate flood impacts.

6. Public Awareness: Informing and educating the public about flood risks and necessary precautions.

Limitations

While the analysis provides valuable insights, there are several limitations to consider:

1. Data Quality and Availability: The accuracy of predictions depends on the quality and granularity of historical rainfall data.
2. Model Assumptions: The ARIMA model assumes linearity and may not capture complex, non-linear interactions in climate data.
3. External Factors: Factors such as urbanization, land use changes, and climate change were not explicitly modeled but can significantly influence flood risks.

Find the full notebook on Github

Top comments (0) ⌄

Code of Conduct • Report abuse