PREDICTIVE MODELLING FOR FLOOD PREDICTION IN LAGOS

Summary

This project aimed to develop a predictive model for identifying the potential timing of future flood events in Lagos, Nigeria. By analyzing historical weather data and flood events, the project successfully identified key weather factors associated with flooding. A machine learning model was then trained to predict future flood risks based on these factors.

Introduction

Lagos, situated on the North Atlantic Ocean, is one of Nigeria's major cities frequently experiencing torrential downpours. The city often faces pluvial flooding, characterized by flash floods that occur suddenly after heavy storms, posing significant concerns. Numerous efforts from both industrial and academic perspectives have been made to address flood-related issues, but these efforts are often hindered by the lack of comprehensive flood datasets, which are crucial for analyzing flood risks.

A review of LIDAR data from 2014 to 2017 indicates that floods, storms, temperature extremes, and droughts together account for about 30% of the country's total economic losses, with floods alone responsible for 5% of these losses. Among natural hazards in Nigeria, flooding is the most prevalent, causing severe impacts on people and property. While natural causes of flooding include heavy rainstorms and ocean storms along the coast, human-induced factors such as burst water mains, ineffective drainage systems, dam failures, and spills also significantly contribute to the problem.

The purpose of this project is to analyze weather data from Lagos to identify patterns and correlations that could help predict next flood occurrences. This involves cleaning and preparing the data, conducting exploratory data analysis (EDA), and building predictive models to forecast flooding events based on various weather parameters.

The Study Area

This study focuses on Lagos state. Lagos is one of the world's major cities and is the most populous city in Africa, ahead of Cairo. Lagos City in Lagos State is Nigeria's largest city and it is located in the south west of Nigeria alongside the Atlantic Ocean. The latitude of Lagos, Nigeria is **6.465422**, and the longitude is 3.406448. Below is the map depicting Lagos State.

Data Collection

A comprehensive weather dataset for Lagos was obtained, encompassing various weather parameters like temperature, humidity, wind speed, and precipitation etc.. Likewise a dataset on Lagos Future Weather Forecast was obtained. The two datasets were needed to train and test the data respectively.

The data overview

The dataset comprises weather observations from Lagos, including temperature, precipitation, wind speed, humidity, and other meteorological variables. Key columns include datetime, temperature metrics (maximum, minimum, and average), various precipitation metrics, wind-related metrics, humidity level, sea level pressure, and an indicator of flood occurrence. The primary focus is on understanding how these variables interact and contribute to flood events.

```python
# Displaying the first few rows of the dataset
print(data.head())
```

```
    name    datetime  tempmax  tempmin  temp  feelslikemax  feelslikemin  \
0  Lagos  2004-05-13     31.7     28.2  30.0          38.9          32.6
1  Lagos  2004-05-14     26.6     25.7  26.3          26.6          25.7
2  Lagos  2004-05-15     25.1     25.1  25.1          25.1          25.1
3  Lagos  2004-05-16     22.9     22.9  22.9          22.9          22.9
4  Lagos  2004-05-17     31.4     25.1  28.6          38.7          25.1

   feelslike   dew  humidity  ...  severerisk              sunrise  \
0       35.8  24.9      74.7  ...         NaN  2004-05-13T06:30:34
1       26.3  25.9      97.5  ...         NaN  2004-05-14T06:30:27
2       25.1  25.1     100.0  ...         NaN  2004-05-15T06:30:20
3       22.9  22.2      95.8  ...         NaN  2004-05-16T06:30:14
4       33.6  25.5      84.2  ...         NaN  2004-05-17T06:30:08

               sunset moonphase        conditions  \
0  2004-05-13T18:55:02      0.80  Partially cloudy
1  2004-05-14T18:55:10      0.83  Partially cloudy
2  2004-05-15T18:55:18      0.87  Partially cloudy
3  2004-05-16T18:55:27      0.90  Partially cloudy
4  2004-05-17T18:55:36      0.94  Partially cloudy

                           description               icon     stations  \
0  Becoming cloudy in the afternoon.  partly-cloudy-day  65201099999
1   Partly cloudy throughout the day.  partly-cloudy-day  65201099999
2          Clearing in the afternoon.  partly-cloudy-day  65201099999
3          Clearing in the afternoon.  partly-cloudy-day  65201099999
4   Partly cloudy throughout the day.  partly-cloudy-day  65201099999

   windspeedmax  windspeedmin
0           NaN           NaN
1           NaN           NaN
2           NaN           NaN
3           NaN           NaN
4           NaN           NaN

[5 rows x 34 columns]
```

Data Cleaning and Preparation

Data cleaning involved several crucial steps to ensure the dataset was suitable for analysis. Columns were renamed for consistency and ease of use, and missing values in key columns were addressed using appropriate strategies. For instance, missing values in the 'preciptype' column were filled with 'no rain,' while median values were used to fill missing entries in the 'wind gust' and 'Sea level pressure' columns. The 'datetime' column was converted to a datetime type for time series analysis. These steps ensured a robust and complete dataset for subsequent analysis.

```
# Selecting specific columns for analysis
weather_data = data.filter(['datetime','precip','precipprob',  'precipcover', 'preciptype','windgust', 'windspeed', 'winddir', 'tempmax', 'tempmin', 'temp','dew', 'humidity', 'sealevelpressure', 'moonphase', 'flood_occur
weather_data.head()
```

	datetime	precip	precipprob	precipcover	preciptype	windgust	windspeed	winddir	tempmax	tempmin	temp	dew	humidity	sealevelpressure	moonphase
0	2004-05-13	0.0	0.0	0.0	NaN	NaN	20.5	128.2	31.7	28.2	30.0	24.9	74.7	NaN	0.80
1	2004-05-14	0.0	0.0	0.0	NaN	NaN	20.5	280.4	26.6	25.7	26.3	25.9	97.5	NaN	0.83
2	2004-05-15	0.0	0.0	0.0	NaN	NaN	16.6	180.0	25.1	25.1	25.1	25.1	100.0	NaN	0.87
3	2004-05-16	0.0	0.0	0.0	NaN	NaN	11.2	350.0	22.9	22.9	22.9	22.2	95.8	NaN	0.90
4	2004-05-17	0.0	0.0	0.0	NaN	NaN	31.3	242.3	31.4	25.1	28.6	25.5	84.2	NaN	0.94

```
# Filling missing values in the 'preciptype' column with 'no_rain'
weather_data.preciptype.fillna('no_rain', inplace=True)
```

```
# Filling missing values in the 'windgust' column with the median value
median = weather_data.windgust.median()
 weather_data.windgust.fillna(median, inplace=True)
```

```
# Filling missing values in the 'sealevelpressure' column with the median value
median = weather_data.sealevelpressure.median()
median
```

```
1011.9
```

```
weather_data.sealevelpressure.fillna(median, inplace=True)
```

```
weather_data.head()
```

	datetime	precip	precipprob	precipcover	preciptype	windgust	windspeed	winddir	tempmax	tempmin	temp	dew	humidity	sealevelpressure	moonphase
0	2004-05-13	0.0	0.0	0.0	no_rain	28.1	20.5	128.2	31.7	28.2	30.0	24.9	74.7	1011.9	0.80
1	2004-05-14	0.0	0.0	0.0	no_rain	28.1	20.5	280.4	26.6	25.7	26.3	25.9	97.5	1011.9	0.83
2	2004-05-15	0.0	0.0	0.0	no_rain	28.1	16.6	180.0	25.1	25.1	25.1	25.1	100.0	1011.9	0.87
3	2004-05-16	0.0	0.0	0.0	no_rain	28.1	11.2	350.0	22.9	22.9	22.9	22.2	95.8	1011.9	0.90
4	2004-05-17	0.0	0.0	0.0	no_rain	28.1	31.3	242.3	31.4	25.1	28.6	25.5	84.2	1011.9	0.94

```
# Handling flood occurrence data
flood_occurences = data.flood_occurrence
flood_occurences.head()
```

```
0    0.0
1    0.0
2    0.0
3    0.0
4    0.0
Name: flood_occurrence, dtype: float64
```

```
flood_occurences.fillna(0, inplace=True)
```

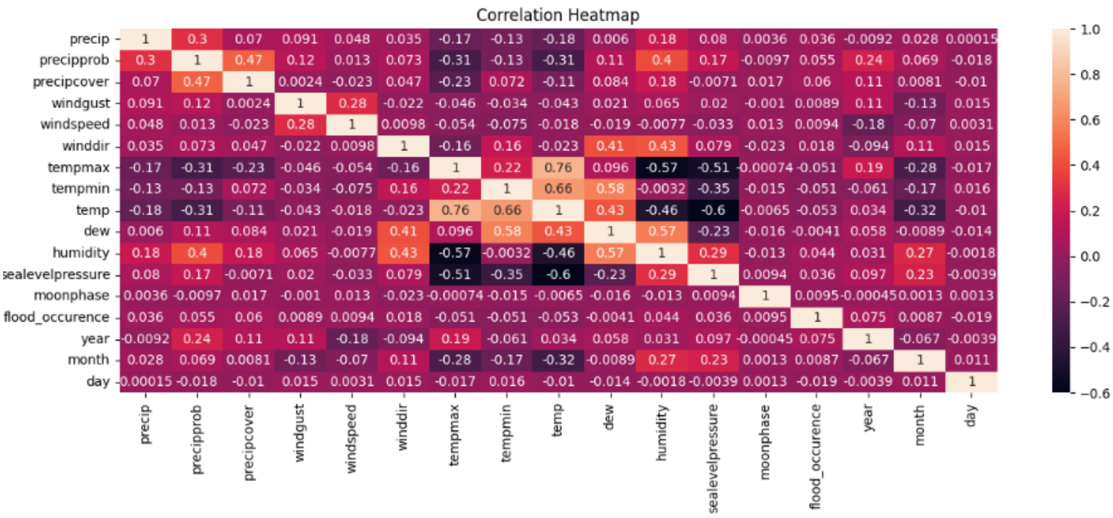
```
# Dropping any remaining rows with missing values
weather_data.dropna(inplace=True)
```

```
# Converting 'datetime' column to datetime type
datetime = pd.to_datetime(weather_data['datetime'])
```

Feature Engineering

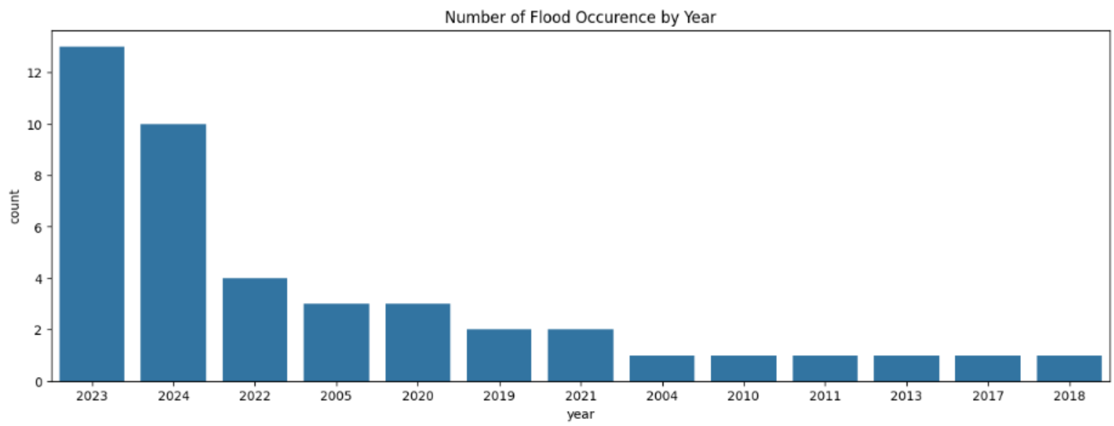
To enhance the model's learning capabilities, relevant features were extracted from the dataset. This included creating new features like year, month, and day from the original date/time column. Correlation analysis explored the relationships between weather variables, aiding in selecting the most informative features for flood prediction.

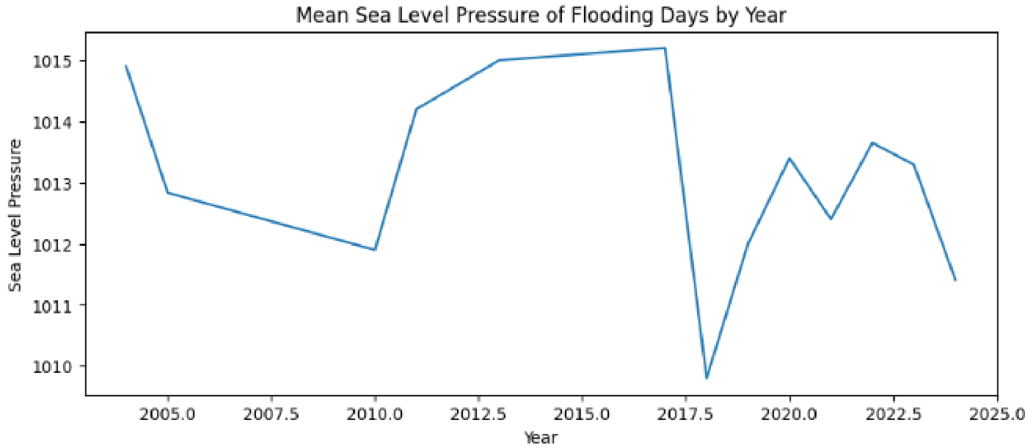
```
# Extracting year, month, and day from the 'datetime' column
weather_data['year'] = datetime.dt.year
weather_data['month'] = datetime.dt.month
weather_data['day'] = datetime.dt.day
```

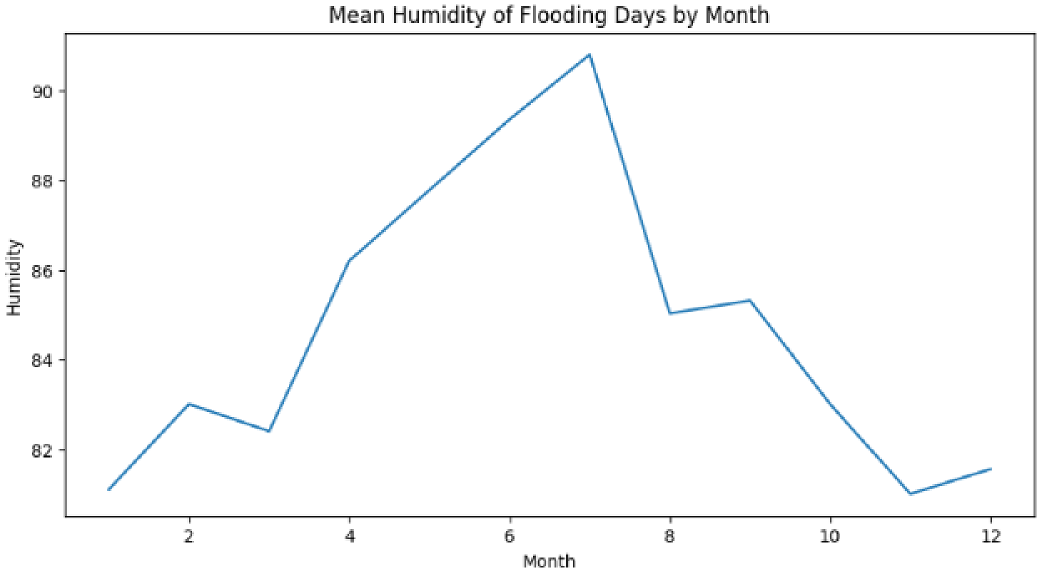


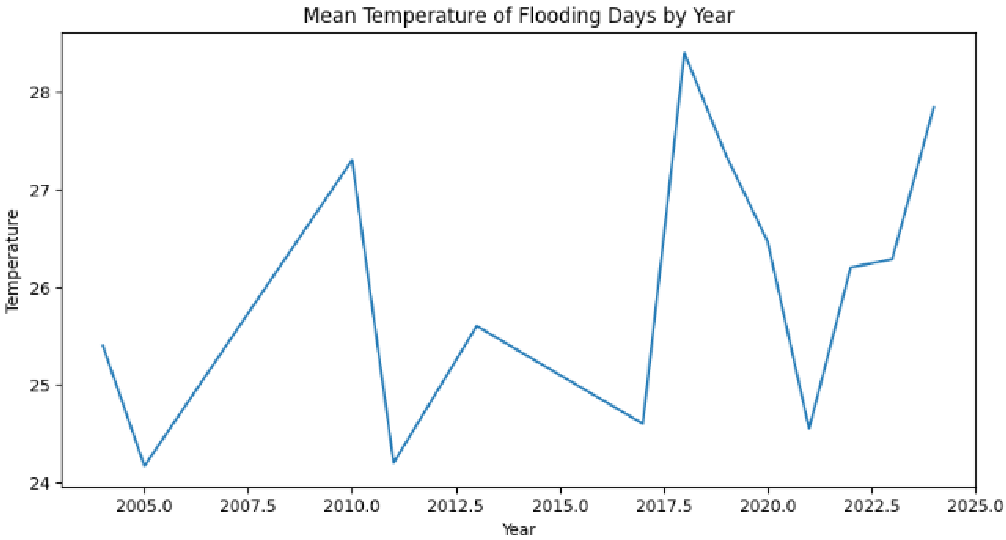
Exploratory Data Analysis (EDA)

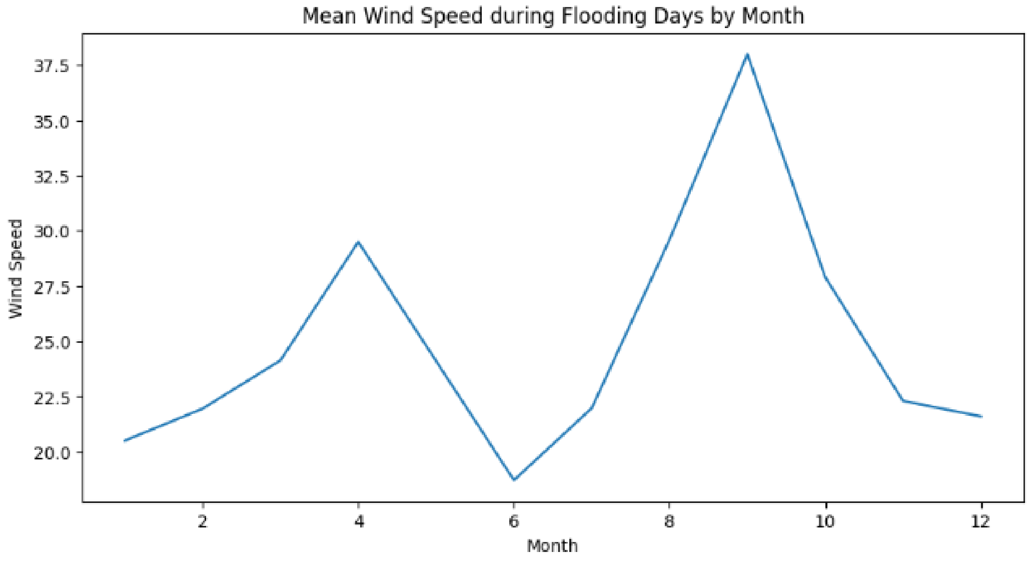
The exploratory data analysis provided valuable insights into historical flood patterns. Visualizations like heatmaps and time series plots revealed trends in weather conditions during flood events. Features like sea level pressure, humidity, and temperature exhibited distinct patterns during flooding periods compared to non-flooding periods.




Mean Sea Level Pressure of Flooding Days by Year
Sea Level Pressure
1015
1014
1013
1012
1011
1010
2005.0
2007.5
2010.0
2012.5
2015.0
2017.5
2020.0
2022.5
2025.0
Year


Mean Humidity of Flooding Days by Month
Humidity
90
88
86
84
82
2
4
6
8
10
12
Month


Mean Temperature of Flooding Days by Year
Temperature
28
27
26
25
24
2005.0
2007.5
2010.0
2012.5
2015.0
2017.5
2020.0
2022.5
2025.0
Year



Model Development

- Several machine learning algorithms were explored, including Logistic Regression, KNN, Support Vector Machines, Random Forest and XGBoost.
- To address class imbalance (unequal distribution of flood and non-flood events), an under-sampling technique was employed to balance the training data.

```
Counter(y_train)
```

```
Counter({'no_flood': 6600, 'flood': 43})
```

```
# Initializing the undersampler
undersampler = RandomUnderSampler()
```

```
# Apply undersampling
x_train_res, y_train_res = undersampler.fit_resample(x_train, y_train)
```

```
x_train_res
```

	precip	precipprob	precipcover	preciptype	windgust	windspeed	winddir	tempmax	tempmin	temp	dew	humidity	sealevelpressure	moonphase
0	-0.276042	-0.907467	-0.422498	0	-0.124897	0.867535	0.275960	-1.626762	-0.978817	-1.299822	-0.864101	0.133193	1.622162	0.92
1	7.528545	1.101968	0.179745	1	-0.124897	-0.444550	1.259647	-1.237168	-1.241901	-2.312818	-0.556765	1.715822	0.165852	0.15
2	1.553999	1.101968	-0.121015	1	-0.124897	-0.105667	2.520517	-2.483867	-0.926200	-2.439442	-0.403097	2.128682	-0.058196	0.18
3	-0.276042	-0.907467	-0.422498	0	-0.124897	0.519962	0.588572	-2.016355	-0.189565	-1.489758	-0.966547	0.119431	1.286090	0.08
4	0.644361	1.101968	6.807303	1	-0.124897	-0.418482	0.105065	-1.431965	1.388940	-0.096889	0.365244	0.449719	-0.058196	0.32
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
81	-0.276042	-0.907467	-0.422498	0	-1.171955	-0.600958	-0.013728	0.554961	0.810155	0.726170	0.416467	-0.224619	-0.506291	0.79
82	-0.249129	1.101968	-0.121015	1	-1.171955	0.181080	0.211353	-0.886534	0.283987	-0.666699	0.211576	0.890103	0.725971	0.40
83	-0.270659	1.101968	-0.121015	1	-1.171955	-0.600958	-0.013728	0.710798	0.862772	1.106044	0.570135	-0.417287	-0.506291	0.37
84	-0.276042	-0.907467	-0.422498	0	0.652300	-0.792122	-0.007476	-0.496941	0.178753	-0.666699	-0.659210	-0.307191	1.734185	0.47
85	-0.098420	1.101968	0.179745	1	-0.319196	-0.262075	0.223857	0.671839	-0.189565	0.662858	-0.146983	-0.871433	0.669959	0.86

86 rows × 14 columns

```
[ ] Counter(y_train_res)
```

```
Counter({'flood': 43, 'no_flood': 43})
```

Modeling

```
[ ] # Initializing classifiers
    log_reg = LogisticRegression()
    knn = KNeighborsClassifier()
    svc = SVC()
    rf = RandomForestClassifier()
    xgb = XGBClassifier()
```

Using Logistic Regression

```
[86] log_reg.fit(x_train_res, y_train_res)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
[88] prediction= log_reg.predict(x_train_res)
```

```
[89] confusion_matrix(y_train_res, prediction)
```

```
array([[30, 13],
       [11, 32]])
```

```
print(classification_report(y_train_res, prediction))
```

```
              precision    recall  f1-score   support

       flood       0.73      0.70      0.71        43
    no_flood       0.71      0.74      0.73        43

    accuracy                           0.72        86
   macro avg       0.72      0.72      0.72        86
weighted avg       0.72      0.72      0.72        86
```

Using SVC

[91]
```
svc.fit(x_train_res, y_train_res)
```

```
▾ SVC
SVC()
```

[92]
```
predy = svc.predict(x_train_res)
```

[93]
```
confusion_matrix(y_train_res, predy)
```

```
array([[33, 10],
       [ 9, 34]])
```

[94]
```
print(classification_report(y_train_res, predy))
```

```
              precision    recall  f1-score   support

       flood       0.79      0.77      0.78        43
    no_flood       0.77      0.79      0.78        43

    accuracy                           0.78        86
   macro avg       0.78      0.78      0.78        86
weighted avg       0.78      0.78      0.78        86
```

Using Random Forest

[95]
```
rf.fit(x_train_res, y_train_res)
```

```
▾ RandomForestClassifier
RandomForestClassifier()
```

[96]
```
predy = rf.predict(x_train_res)
```

[97]
```
confusion_matrix(y_train_res, predy)
```

```
array([[43,  0],
       [ 0, 43]])
```

[98]
```
print(classification_report(y_train_res, predy))
```

```
              precision    recall  f1-score   support

       flood       1.00      1.00      1.00        43
    no_flood       1.00      1.00      1.00        43

    accuracy                           1.00        86
   macro avg       1.00      1.00      1.00        86
weighted avg       1.00      1.00      1.00        86
```

[99]
```
# Random Forest is the best performing model
```

Results and Evaluation

- The trained Random Forest model exhibited promising results in predicting potential flood events based on future weather forecasts.
- Flood likelihood was forecasted for a specific period, providing valuable insights for proactive flood mitigation efforts.

Conclusion

From the model's prediction, the next flooding days in Lagos will be the 10th and 11th July, 2024.). Factors like humidity, temperature, and sea level pressure forecasts for these days are consistent with historical flood events. This suggests a period of heightened vigilance and potential need for proactive flood mitigation measures.

The computational environment for these tasks was facilitated by Google Colab. The analysis steps and code can be provided upon request for further exploration of the technical details.

Recommendation

It is important to bear in mind that flooding cannot be constrained within human environment and the menace will worsen in the future, and as such there is the need to create social systems that are resilient to hazard. This study recommends developing of a repository of flood data to be useful for future investigations.

REFERENCES

Nkwunonwo, U. C. (2016). A Review of Flooding and Flood Risk Reduction in Nigeria. Global Journal of Human Social Science, 1.

Google pictures

Prediction of rainfall using MLP classifier of Neural Network: https://youtu.be/-hYVjCPe0ho?si=tdQQAMCT8iyUwe7l

Weather Data Services: https://www.visualcrossing.com/weather/weather-data-services

Colab link: https://colab.research.google.com/drive/1F4uoKt2MOknzbat-X3ZJ1ODUJX4huf_C?usp=sharing

Weather Data and Weather API: https://www.visualcrossing.com/