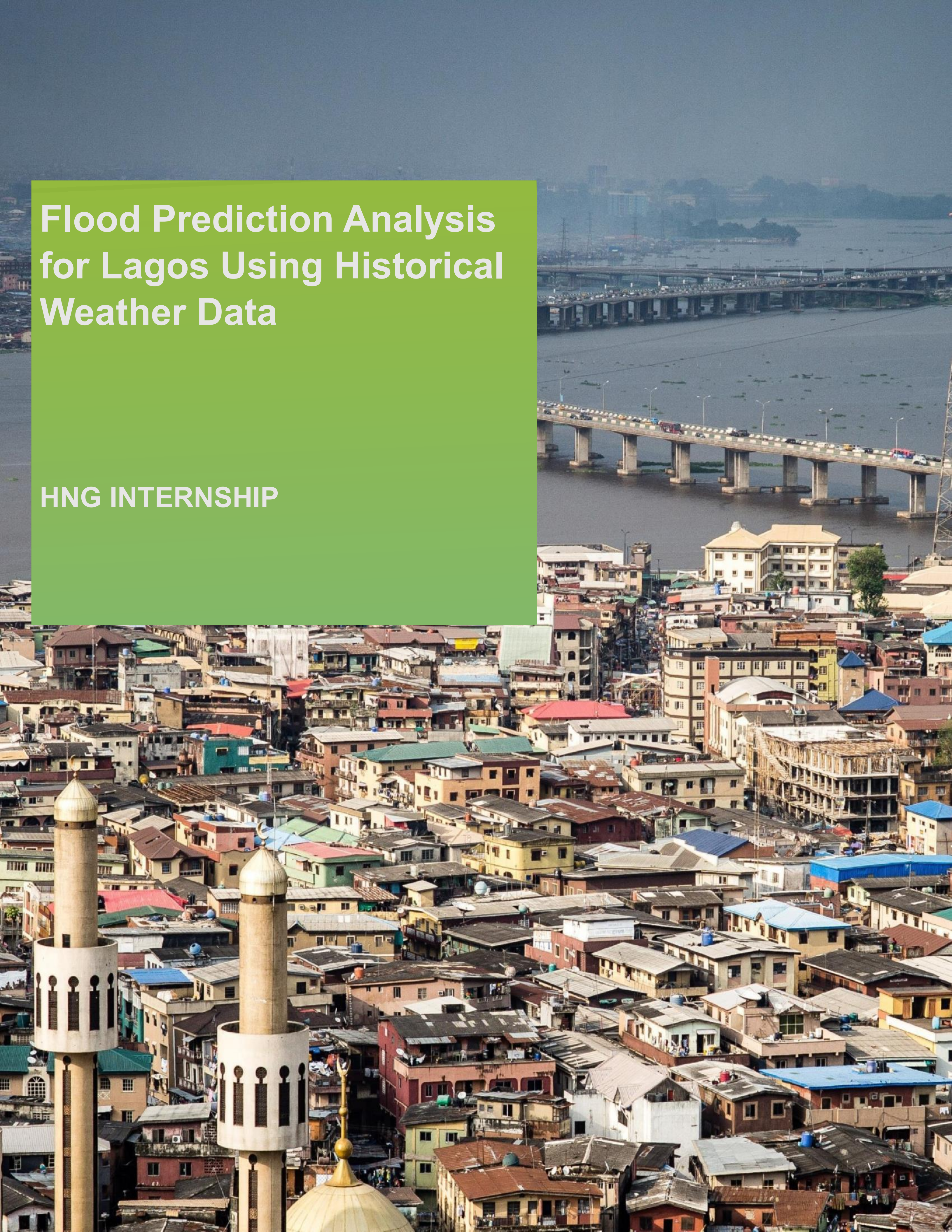
Flood Prediction Analysis for Lagos Using Historical Weather Data

HNG INTERNSHIP



INTRODUCTION

Flooding, the most devastating global natural disaster, occurs when excess water inundates dry land, often due to heavy rainfall. It impacts 70 million people annually, with 800 million in flood-prone areas. In developing countries, causes include climate change, poor infrastructure, rapid population growth, and inadequate preparedness.

Lagos State, Nigeria, has a long history of flooding, primarily due to its coastal location and heavy seasonal rainfall. The city's rapid urbanization, inadequate drainage systems, and poor waste management exacerbate the problem. Notable floods occurred in 2011, 2012, and 2020, causing significant property damage, displacement, and loss of life. Climate change and rising sea levels further threaten the city's low-lying areas. Efforts to mitigate flooding include improved infrastructure, flood warning systems, and urban planning reforms. However, challenges remain in effectively managing and reducing the risk of recurrent flooding in Lagos.

Predicting flooding events accurately can help in mitigating its adverse effects and preparing effective response strategies. This report presents the use of Python to analyze weather data and predict flooding events. The analysis includes data preparation, exploratory data analysis (EDA), predictive modeling, and visualization of results.

CAUSES OF FLOODING IN NIGERIA

- **Heavy Rainfall**: Intense and prolonged rain can overwhelm drainage systems, leading to water accumulation and flooding.
- **Poor Drainage Systems**: Inadequate or poorly maintained drainage infrastructure can cause water to back up and flood streets and properties.
- **Urbanization**: Rapid development reduces natural land that can absorb rainwater, increasing runoff and the risk of urban flooding.
- **Deforestation**: Removal of trees reduces the land's ability to absorb water, increasing runoff into rivers and urban areas, leading to floods.
- **Climate Change**: Rising sea levels and increased frequency of extreme weather events contribute to higher flooding risks.

STUDY AREA

Lagos State is situated close to the Atlantic Ocean, with its coastline extending from Badagry to the border with Ogun State, approximately 180 km long, and stretching inland for about 32 km. Approximately 17% of the state consists of lagoons, creeks, and coastal estuaries (Adefuy et al., 2002; Onyekwelu et al., 2003). Geographically, the state lies between longitudes 2° 42' E to 3° 42' E and latitudes 6° 22' N to 6° 42' N (Akinsanya, 2003; Onyekwelu et al., 2003).

Despite being one of the smallest states in Nigeria, covering only 0.4% of the total landmass, Lagos State hosts the highest number of industries and serves as the commercial hub of the country. As the leading port city in West Africa and the second largest metropolitan city in Africa, Lagos accounts for more than 70% of Nigeria's maritime cargo and seaport activities, as well as 80% of the country's international air traffic.

The city's population has been growing significantly, with a rise of 2,129,126 people, reflecting an annual growth rate of 3.26%. It is expected that Lagos's population will reach 20 million by 2030 (National Population Commission Nigeria, 2019). This growth is mainly attributed to the concentration of economic activities in the city. The population density in Lagos is around 6,871 people per square kilometer, which far exceeds the global average population density for coastal zones of 112 persons per square kilometer (IHDP, 2012) (National Statistics Bureau, 2019).

DATA PREPARATION

The dataset used was obtained from the Humanitarian Data Exchange. It contains various weather details recorded from 2019 to 2024, such as temperature, rainfall, humidity, wind speed, and more. The data was imported into Python from an Excel file.

Key Steps

Data Import:

The weather dataset was imported from an Excel file and loaded into a Pandas DataFrame.

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

```
df = pd.read_excel("PROJECT 2.xlsx")
df.head()
```

Data Cleaning:

Unnecessary columns was removed using the drop () function and managing missing values by employing methods like fillna() to substitute them with mean, median, or other suitable strategies.

```
df = df.drop(["name","feelslikemax","feelslikemin", "feelslike", "dew", "solarenergy", "uvindex"    ,"severerisk", "sunrise", "sunset","moonphase","snow"
```

```
df.head()
```

Data Transformation:

Data formats were standardized, such as converting date columns to datetime format using pd.to_datetime(), and new columns (such as flooding_event in this case) are generated to facilitate additional analysis based on specified conditions.

```
df["datetime"] = pd.to_datetime(df.datetime)
```

```
df.head()
```

	datetime	tempmax	tempmin	temp	humidity	precip	precipprob	precipcover	preciptype	windgust	windspeed	winddir	sealevelpressure	visibility
0	2019-07-01	27.5	25.0	25.7	92.3	26.0	100	4.17	rain	33.5	16.9	225.9	1014.3	8.4
1	2019-07-02	31.0	24.0	26.5	89.1	0.0	0	0.00	NaN	33.1	18.4	236.7	1012.4	9.1
2	2019-07-03	30.0	24.8	26.9	88.0	0.7	100	12.50	rain	28.1	18.4	220.5	1012.5	9.4
3	2019-07-04	27.3	25.0	25.8	95.2	8.0	100	4.17	rain	24.8	14.8	237.6	1013.5	8.6
4	2019-07-05	28.0	25.0	25.8	94.3	3.0	100	4.17	rain	24.1	16.6	246.8	1014.6	8.8

```
df.describe()
```

Important Parameters and their uses in the analysis

I. **Precipitation amount (precip)**: It directly measures the volume of rain falling in a given period, which is the most direct indicator of potential flooding.

II. **Probability of precipitation (precipprob)**: It indicates the likelihood of rain occurring. Higher probabilities suggest a greater chance of rain, which is crucial for predicting flooding.

III. **Percentage of the day covered by precipitation (precipcover)**: It shows how much of the day is expected to have rain. Prolonged periods of rain increase the risk of flooding.

IV. **Humidity percentage (humidity)**: High humidity levels indicate saturated air, which can lead to more intense rainfall. It also affects evaporation rates, which can influence how quickly water levels rise.

V. **Maximum temperature (tempmax)**: It affects evaporation rates. Higher temperatures can increase evaporation, which might temporarily mitigate flooding but can also lead to more intense storms when the water vapor condenses.

VI. **Minimum temperature (tempmin)**: It influences the dew point and condensation rates, affecting rainfall intensity and duration.

VII. **Sea level pressure (sealevelpressure)**: Low pressure systems are often associated with stormy weather and heavy precipitation. Changes in sea level pressure can be an indicator of incoming weather systems that could lead to flooding.

VIII. **Maximum wind gust speed (windgust)**: Strong wind gusts can exacerbate storm surges and increase the impact of heavy rainfall, leading to more severe flooding.

IX. **Average wind speed (windspeed)**: It influences how weather systems move and develop. Persistent winds can lead to prolonged precipitation over an area.

X. **Cloud cover percentage (cloudcover)**: High cloud cover is often associated with low pressure systems and prolonged periods of precipitation, which can increase flooding risk.

XI. **Visibility distance (visibility)**: Reduced in visibility can indicate heavy rain or fog, which are conditions often associated with flooding.

CALCULATION

The flooding event was calculated using an optimum precipitation value of 20 mm, which is the standard for Lagos' coastal precipitation. This benchmark helps in accurately predicting potential flood risks and preparing necessary mitigation measures.

```
# Set frequency (D for daily data, adjust as needed)
df = df.asfreq('D')


# Debug: Check frequency
#print(df.index.freq)

# Create 'flood' column based on precipitation threshold
threshold = 56  # Precipitation threshold in mm
df['flood'] = (df['precip'] > threshold)
```

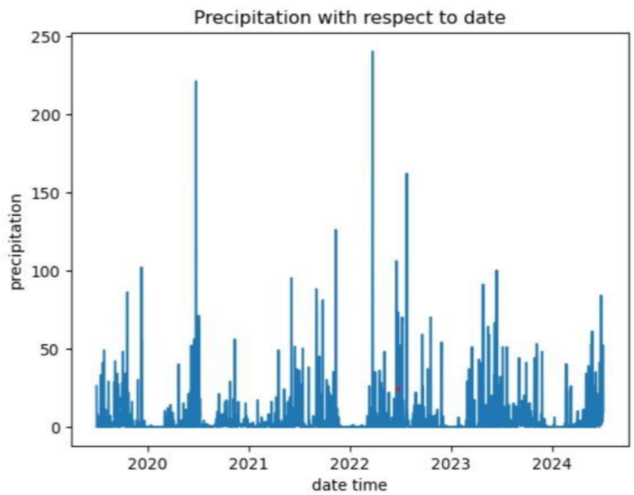
ANALYSIS

Exploratory Data Analysis (EDA) was conducted using Python to uncover underlying patterns and relationships within the weather dataset. This initial phase is crucial for identifying significant factors influencing flooding and guiding subsequent predictive modeling efforts. Python's libraries such as Pandas, Matplotlib, and Seaborn were utilized for data manipulation, visualization, and analysis. Visualizations were created to gain insights into the data

- **Date was plotted against precipitation amount**

  The plot shows that in certain year there are consistently higher and lower precipitation levels which shows extreme weather events, such as heavy storms or flooding incidents.

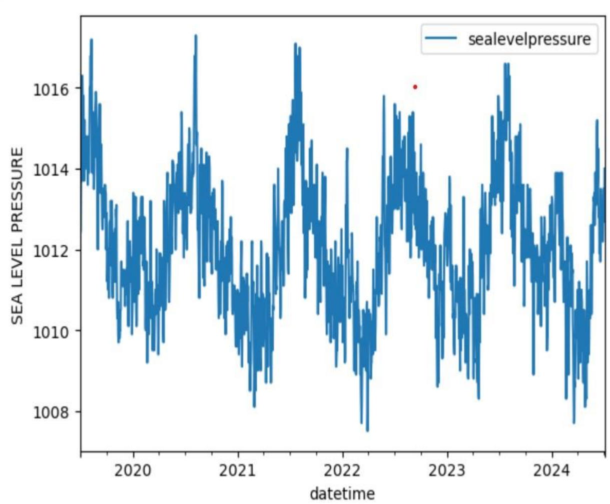
```
[202]: plt.plot(df["datetime"], df["precip"])
plt.xlabel("date time")
plt.ylabel("precipitation")
plt.title("Precipitation with respect to date");
```



- **Date was plotted against sea level pressure**

  The plot of date against sea level pressure reveals the trends in atmospheric pressure changes over time. The pressure fluctuations insights help to identify weather events like storms or calm period.

```
[147]: df.plot(x ="datetime", y = "sealevelpressure", ylabel = "SEA LEVEL PRESSURE");
```



RESULT AND DISCUSSION

The dataset is a time series data, ARIMA plot was used to predict the future outcome. From my analysis, it was found that by 2024-10-04 there will likely be a flooding event with forecasted precipitation of more than 20 mm This indicates a significant risk of heavy rainfall and highlights the urgent need for proactive flood management and preparedness measures to mitigate potential impacts.

```
from statsmodels.tsa.arima.model import ARIMA
model = ARIMA(df['precip'], order=(5, 1, 0))
# Example order, adjust as needed
model_fit = model.fit()
```

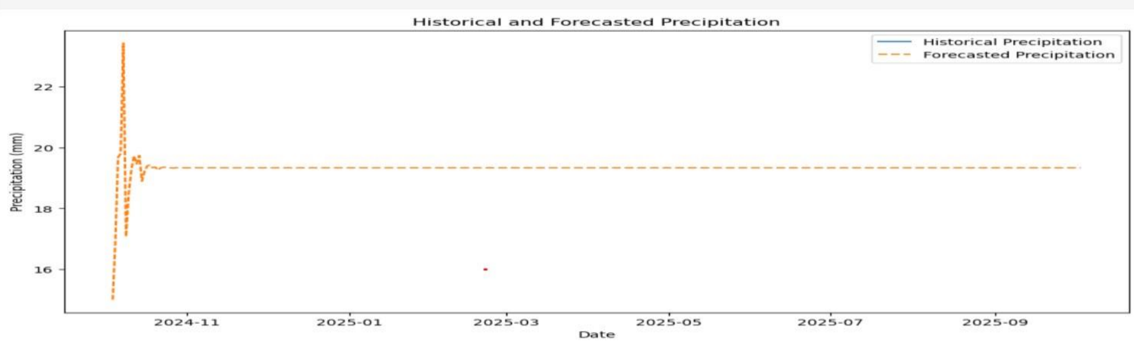
```
model_fit.summary()
```

SARIMAX Results

```
# Set frequency (D for daily data, adjust as needed)
df = df.asfreq('D')


# Debug: Check frequency
#print(df.index.freq)

# Create 'flood' column based on precipitation threshold
threshold = 56  # Precipitation threshold in mm
df['flood'] = (df['precip'] > threshold)
```

```


# Forecast future precipitation
forecast_steps = 365  # Number of future days to predict
forecast = model_fit.forecast(steps=forecast_steps)


# Debug: Print forecasted values
print(forecast)
# Handle last_date as a datetime object
last_date = df.index[-1]  # Using the last datetime from the index
```

```
# Plot historical precipitation and forecasted precipitation
plt.figure(figsize=(12, 6))
plt.plot(df.index, df['precip'], label='Historical Precipitation')
plt.plot(future_data.index, future_data['precip_forecast'], label='Forecasted Precipitation', linestyle='--')
plt.xlabel('Date')
plt.ylabel('Precipitation (mm)')
plt.title('Historical and Forecasted Precipitation')
plt.legend()
plt.show()
```



CONCLUSION

The analysis using Python effectively identified key weather parameters that influence flooding and demonstrated a method to predict flooding events. Plotting date against precipitation amount revealed seasonal patterns, long-term trends, and extreme weather events. This highlights the impact of climate variability, identifies potential data issues, and aids in forecasting and planning. The analysis provides essential insights for weather analysis, climate studies, and resource management, supporting better preparedness and informed decision-making.

REFERENCE

1. Humanitarian data exchange (2023), Nigeria : Flood data Retrieved from: https://data.humdata.org/dataset/nigeria-nema-flood-affected-geographicalareasnorth-east-nigeria-flood-affected-geographical-areas
2. Punch Nigeria, (July, 2024): Lagos, Edo, A'Ibom, Ebonyi plan camps for flood victims Retrived from, https://www.msn.com/en-us/news/us/lagos-edo-a-ibomebonyi-plan-camps- for-flood-victims/ar-BB1puK6n?ocid=BingNewsSerp
3. Visual crossing weather, (march, 23): Weather Data Documentation https://www.visualcrossing.com/resources/documentation/weather-data/weather-datadocumentation/