PROJECT TITLE : FLOOD MONITORING SYSTEM

TEAM MEMBER

911721104014 : S.BALAGANESH

Phase 4: Development Part 2

In this section continue building the project by performing different activities like feature engineering, model training, evaluation etc as per the instructions in the project

Phase4 : Development Part 2

INTRODUCTION

Feature engineering is the process of selecting, manipulating, and transforming raw data into features that can be used in supervised learning. In order to make flood monitoring work well on new tasks, it might be necessary to design and train better features. As you may know, a “feature” is any measurable inputthat can be used in a predictive model — it could be the color of an object or the sound of someone’s voice. Feature engineering, in simple terms, is the act of converting raw observations into desired features using statistical or Flood Monitoring approaches.

In this project we will see :

What is Feature engineering,

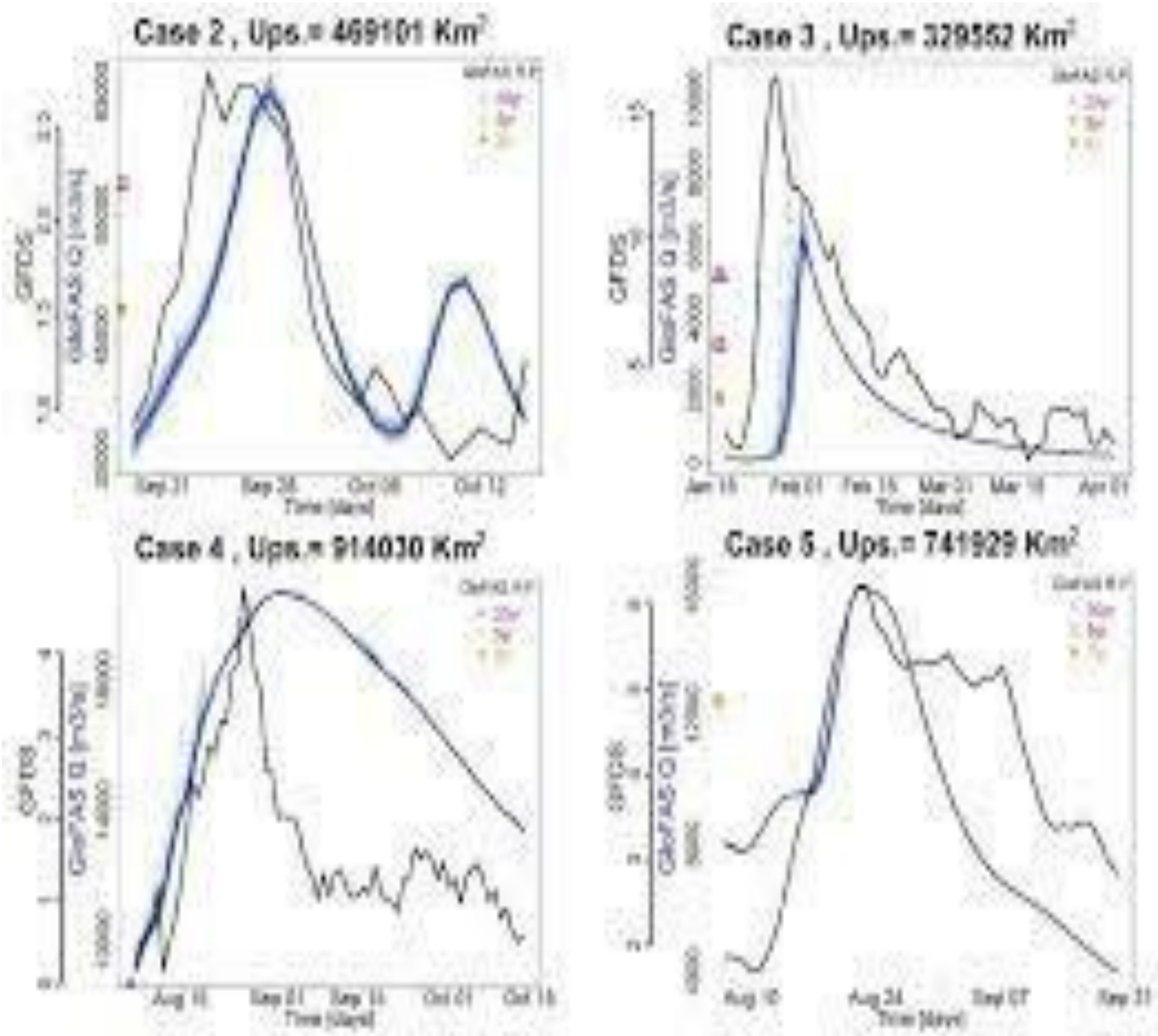
Importance of Feature Engineering,

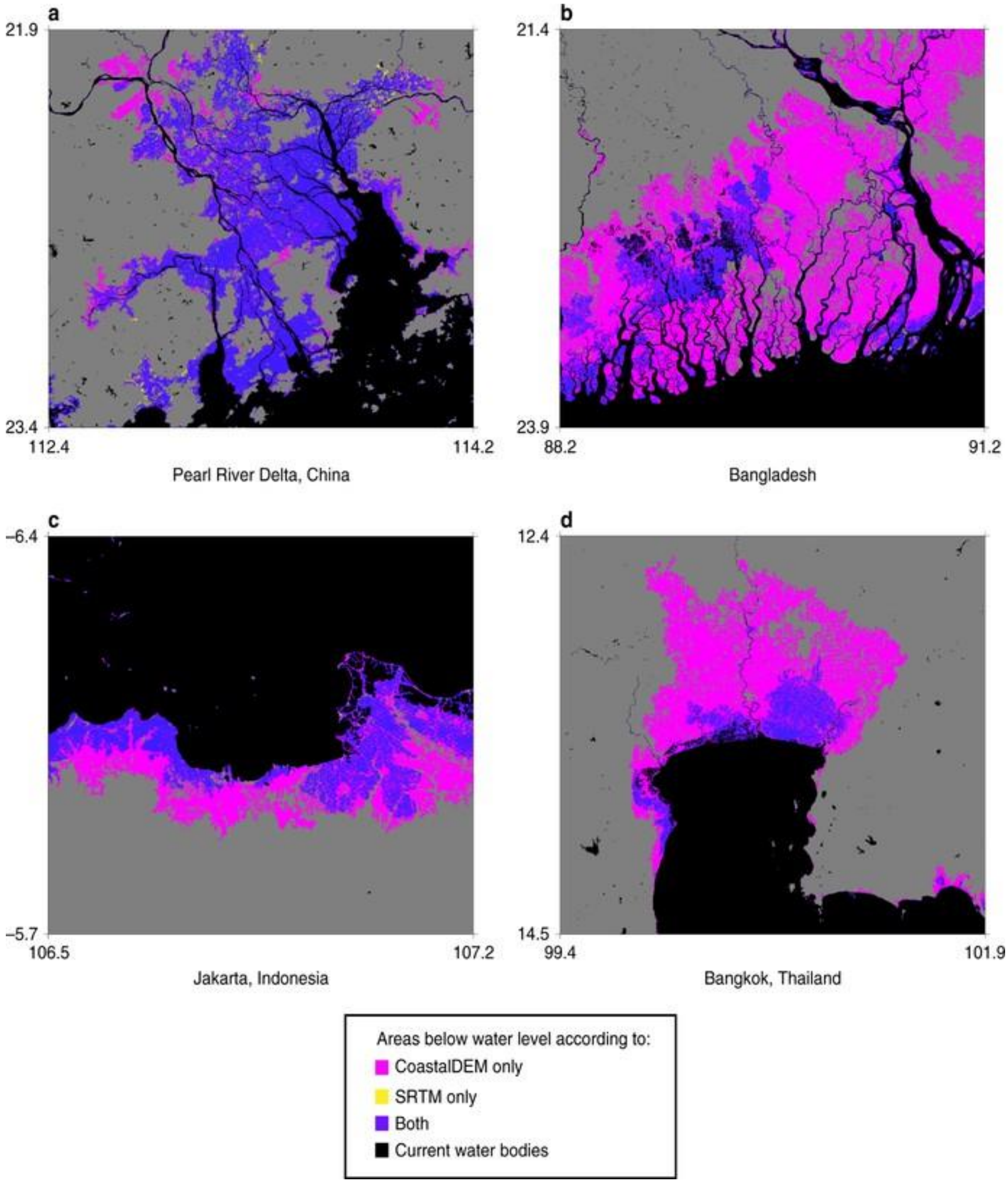
Feature Engineering Techniques for Flood Monitoring,

Few Best tools for feature engineering.

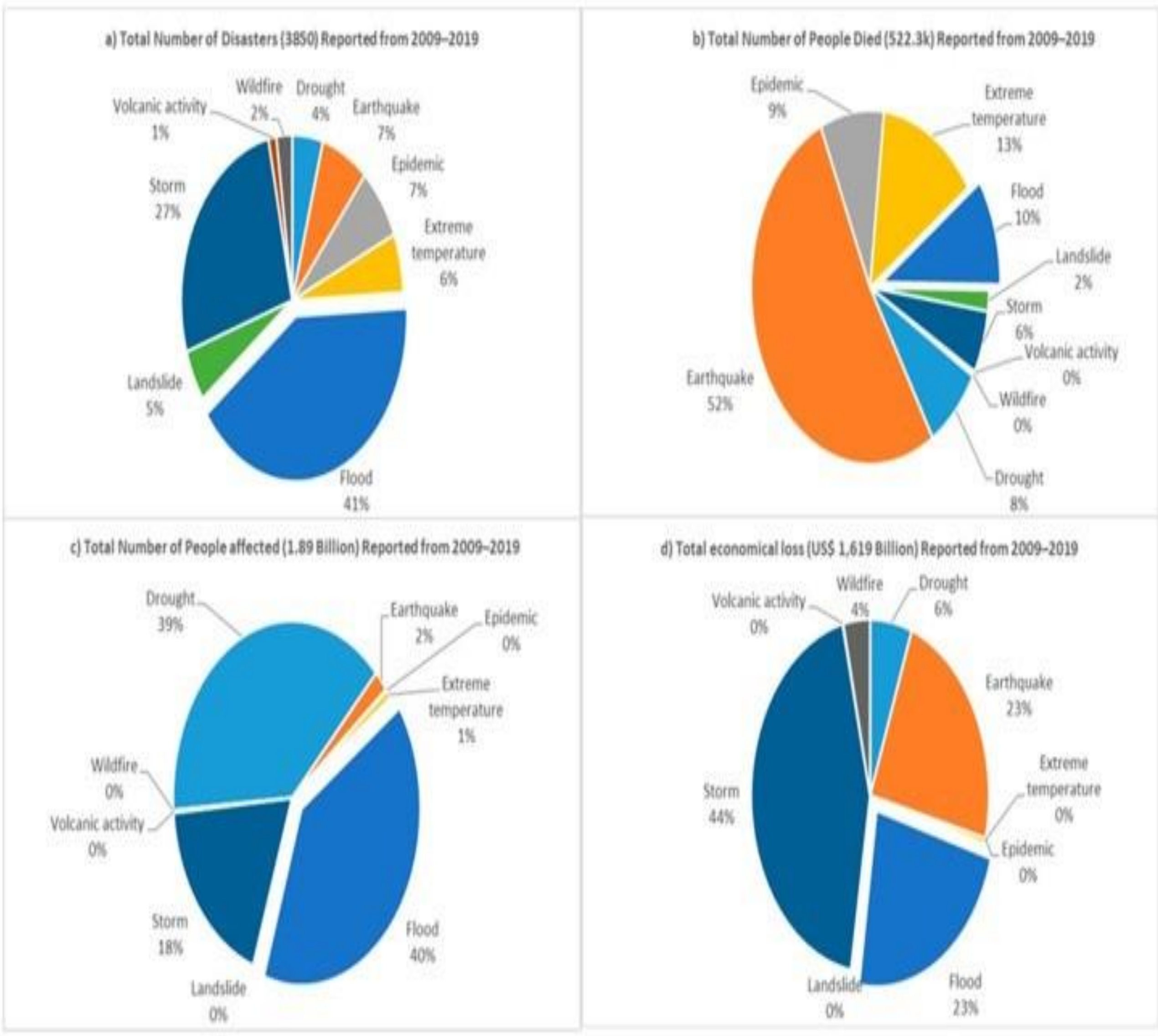
What is Feature Engineering ?

Feature engineering is a Flood Monitoring technique thatleverages data to create new variables that aren’tin the training set. It can produce new features for both supervised and unsupervised learning, with the goal of simplifying and speeding up data transformations while also enhancing model accuracy. Feature engineering is required when working with flood monitoring models. Regardless ofthe data or architecture, a terrible feature will have a directimpact on your model. Now to understand itin a much easier way, let’s take a simple example.Below are the prices of properties in x city. It shows the area ofthe flood monitoring and total level.


Case 2 , Ups.= 469101 Km²
Case 3 , Ups.= 329552 Km²
Case 4 , Ups.= 914030 Km²
Case 5 , Ups.= 741929 Km²



Historical records have shown that flood is the most frequent natural hazard, accounting for 41% of all natural perils that occurred globally in the last decade. In this period alone (2009 to 2019), there were over 1566 flood occurrences affecting 0.754 billion people around the world with 51,002 deaths recorded and damage estimated at $371.8 billion. Put in context, these statistics only account for “reported” cases of large-scale floods, typically considered flood disasters. A flood disaster is defined as a flood that significantly disrupts or interferes with human and societal activity, whereas a flood is the presence of water in areas that are usually dry. The global impact of a flood would be more alarming if these statistics incorporated other numerous small-scale floods where less than 10 people may have died, 100 or more people may have been affected or where there is no declaration of a state of emergency or a call for international assistance. Nevertheless, the current situation calls for improved ways of monitoring and responding to floods. The importance of improved flood monitoring cannot be overemphasized given the growing uncertainty associated with climate change and the increasing numbers of people living in floodprone areas.



Importance of Feature Engineering:

Remote monitoring: Systems can monitor water levels at remote sites.

Data transmission: Systems can send data from all sites to a centralized server.

Other features of flood monitoring systems include:

Web-based access: Systems can be accessed 24/7 by users via the internet.

Early warning systems: Systems can use artificial intelligence to provide

information on natural events.

Sensors: Systems can use sensors that operate on ultrasonic and RADAR

technologies to automatically measure river water levels.

Flood hazard mapping: Systems can use wireless water level sensors and CCTV

cameras to monitor water levels in different areas of a city. Mean water level data: Systems can use sensor technology to monitor water levels

in real time.

Alert gages: Systems can use sensors to detect changes in parameters such as

precipitation volume and water level.

Flood early warning systems are stand-alone systems that can be placed in floodprone spots.

Flood disaster indicator systems use float switch sensors to determine water levels

and send alert messages to users.

Feature Engineering Techniques for

Flood Monitoring:

The main 3 techniques of flood forecasting:

1.Lumped

2. Quasi-distributed

3. Distributed

LUMPED

"Lumped feature" is not a widely recognized or standard term in the context of flood monitoring or hydrology. However, I can provide some information that might be relevant to the concept you're referring to in the context of flood monitoring.

Flood monitoring typically involves the collection and analysis of various data to assess and predict flood events. These data can be broadly categorized into two types: point data and spatial data. Point data refer to data collected at specific locations, such as river gauges or weather stations, while spatial data provide information aboutthe entire area or region of interest.

Some of the key data and features used in flood monitoring and prediction may

include:

1.River Gauges: These devices measure water levels and flow rates in rivers and

streams at specific points. Changes in water levels can be used to monitor potential

flooding.

2. Rainfall Data: Rainfall is a crucial factor in flood monitoring. Data from rain

gauges can help identify areas experiencing heavy precipitation, which can contribute

to flooding.

3.Weather Radar: Weather radar systems provide information about the intensity and

movement of precipitation, which can be useful for predicting the potential for flash

floods.

4. Topographical and Hydrological Data: Information about the terrain and the

natural flow paths of water, such as elevation data, land use data, and soil types, are

important for understanding how water will flow during a flood event.

5.Satellite Imagery: Satellite data can provide a broader view of the area, allowing for

monitoring large-scale weather patterns and changes in surface water bodies.

6.Remote Sensing: Remote sensing technologies, such as LiDAR and aerial

photography, can help create detailed floodplain maps and assess land cover changes.

7.Hydrological Models: Various hydrological models, such as HEC-RAS and HECHMS, are used to simulate how water flows through a watershed and predict flooding

under different conditions.

8.Historical Data: Past flood events and their impacts are valuable for understanding patterns and potential risks.

It's possible that the term "lumped feature" may be referring to the aggregation or combination of some of these data and features to create a comprehensive view of the flood situation. In this context, a "lumped feature" might be a summary statistic or a combined representation of various data sources. However, it's essential to clarify the specific terminology and context to provide a more precise explanation.

Flood monitoring is a complex field that uses various data sources, technology, and models to assess and predict flood events, and the choice of data and features depends on the specific needs of the monitoring system and the geographical region in question.

QUASI-DISTRIBUTED:

A "quasi-distributed feature" in the context of flood monitoring typically refers to a type of sensor or data collection system that is spatially distributed, but not as finely distributed as a fully distributed network. This approach combines the advantages of both point measurements and fully distributed systems, allowing for a more comprehensive understanding of flood conditions.

Here are some details about quasi-distributed features in flood monitoring:

1.Spatial Distribution: Quasi-distributed sensors or data collection systems arestrategically placed at multiple locations within a region of interest. These locations are selected to capture key points of variation in the flood environment.

2.Sensor Types: Various types of sensors can be used in a quasi-distributed network for flood monitoring, including water level sensors, rainfall gauges, weather stations, and water quality sensors. These sensors collect data relevant to flood conditions.

3.Data Fusion: Data from the quasi-distributed sensors are typically collected and integrated to provide a more comprehensive understanding of the flood situation. This may involve real-time data transmission and central data processing to create a holistic picture of the flood event.

4.Monitoring Objectives: The primary objectives of a quasi-distributed system are to improve flood detection, early warning, and flood forecasting. By having sensors distributed across an area, it becomes possible to monitor conditions at various points simultaneously.

5.Data Analysis: The data collected from quasi-distributed sensors are often analyzed using hydrological models and GIS (Geographic Information Systems) to create flood inundation maps, predict flood extents, and assess the potential impacts of flooding on communities and infrastructure.

6.Response Planning: The information obtained from quasi-distributed sensors can be used to trigger flood response actions, such as issuing warnings, evacuations, and resource allocation for disaster management.

7.Advantages: Quasi-distributed systems offer a balance between the high granularity of fully distributed sensor networks and the limited information from point sensors. They can provide valuable data for flood monitoring and decision-making while being more cost-effective than fully distributed systems.

8.Challenges: Challenges with quasi-distributed systems may include sensor maintenance, data integration, and the need for well-defined sensor placement strategies to capture critical variations in the flood-prone area.

In a quasi-distributed feature in flood monitoring refers to a network of sensors or data collection points strategically placed across a region to provide a more comprehensive and spatially distributed view of flood conditions. This approach is particularly valuable for assessing and responding to flood events, as it allows for the collection and analysis of data from multiple locations to improve flood prediction and management.

DISTRIBUTED

A "distributed feature" in the context of flood monitoring refers to a network of sensors or data collection systems that are evenly and densely distributed across a geographical area to provide comprehensive, real-time data about flood conditions.

Here are the details about distributed features in flood monitoring:

1.Sensor Network: Distributed flood monitoring systems consist of a large number of sensors or data collection points, such as river gauges, rainfall gauges, weather stations, and water level sensors. These sensors are strategically placed across the flood-prone region.

2.High Spatial Resolution: The key characteristic of a distributed feature is the high spatial resolution it offers. With sensors placed at regular intervals, often in a grid pattern, this system provides detailed information about conditions throughout the flood-prone area.

3.Real-Time Data: Sensors in a distributed network are typically equipped with realtime data transmission capabilities. They continuously collect data, such as water levels, rainfall, and weather conditions, and transmit it to a central monitoring station or database in real-time.

4.Data Integration: The data collected from distributed sensors are integrated and analyzed to create a comprehensive picture of flood conditions. Geographic Information Systems (GIS) and hydrological models are often used to process and visualize this data.

5.Early Warning and Forecasting: Distributed monitoring systems are essential for early flood detection and forecasting. By closely monitoring conditions across the entire area, it becomes possible to provide timely warnings and predictions to local authorities and communities.

6.Flood Inundation Mapping: The high-resolution data from distributed systems can be used to create flood inundation maps that show the extent and depth of flooding in real-time. This information is invaluable for emergency response and disaster management.

7.Decision Support: The data collected from a distributed network are used to make informed decisions regarding flood response, such as evacuations, resource allocation, and infrastructure protection.

8.Advantages: Distributed features offer a detailed, fine-grained understanding of flood conditions. They are particularly useful in areas prone to flash floods and where rapid decision-making is critical. These systems are highly effective for flood risk reduction and management.

9.Challenges: Deploying and maintaining a distributed flood monitoring system can be costly and requires regular maintenance. Sensor placement and data management can also be complex.

10.Applications: Distributed flood monitoring systems are used in variousapplications, including urban flood management, watershed monitoring, dam safety,and disaster risk reduction.

In a distributed feature for flood monitoring involves the dense deployment of sensors across a flood-prone area to collect high-resolution, real-time data. This approach is crucial for early warning, flood forecasting, and decision support in flood management and helps authorities and communities respond effectively to flood events.

Few Best tools for feature engineering:

There are many tools which will help you in automating the entire feature engineering process and producing a large pool of features in a short period oftime for both classification and regression tasks. So let’s have a look at some ofthe features engineering tools.

Feature engineering is a critical step in flood monitoring and early warning systems, as it involves selecting, creating, and transforming relevant data features to improve the accuracy of predictions and decision-making. While the specific tools you use may depend on your dataset and the analysis you are conducting, here are a few besttools and libraries commonly used forfeature engineering in the context of flood monitoring and early warning systems:

1. **Python:** Python is a widely used programming language for data analysis and feature engineering. It offers a range of libraries for working with geospatial and time series data, making it highly suitable forflood monitoring. Key libraries include:

- **Pandas:** For data manipulation and cleaning.
- **NumPy:** For numerical operations.
- **GeoPandas:** For handling geospatial data.
- **Matplotlib** and **Seaborn:** For data visualization.
- **Scikit-learn:** For machine learning tasks, including feature selection and extraction.

2. **Open-source GIS Software:** Geographic Information Systems (GIS) software is essential for working with geospatial data. Some popular open-source options are:

- **QGIS:** A user-friendly GIS software that allows you to visualize, analyze, and manipulate geospatial data.

- **GRASS GIS:** A powerful GIS platform for advanced geospatial analysis and modeling.

3. **Fiona and Shapely:** These Python libraries are often used in conjunction with GeoPandas to read and manipulate geospatial vector data (Fiona) and perform geometric operations (Shapely).

4. **Time Series Analysis Tools:** Since flood data often involves time series information, you can use tools like:

- **Statsmodels:** Fortime series analysis, modeling, and forecasting.

- **Prophet:** An open-source forecasting tool by Facebook fortime series data.

- **ARIMA and SARIMA models:** Commonly used fortime series analysis and forecasting.

5. **Geo-Referenced Data Libraries:** Libraries for working with geo-referenced data, such as NetCDF or HDF5, are useful for managing large and complex datasets often encountered in flood monitoring.

6. **Machine Learning Libraries:** Machine learning techniques can be employed for feature selection and engineering. Libraries like Scikit-learn and TensorFlow can help in this regard.

7. **Remote Sensing Tools:** If you are working with remote sensing data forflood monitoring, you may need specialized software like ENVI or SNAP for processing and feature extraction from satellite imagery.

8. **Custom Scripts:** Depending on your specific data and requirements, you may need to write custom scripts and programs to preprocess data, extractfeatures, or perform specific analyses.

9. **Web-based Tools:** In some cases, web-based platforms like Google Earth Engine may be used forfeature extraction and analysis ofremote sensing data.

10. **Data Visualization Tools:** Visualization tools like Tableau or Power BI can help in exploring data and identifying patterns that may lead to feature engineering ideas.

Rememberthatthe choice oftools will depend on your specific data sources, analysis goals, and the scale of yourflood monitoring and early warning system. It's also crucialto stay up-to-date with the latest developments in the field and utilize the tools and techniques that best suit your project's needs.

Topic 1: Data Collection

python

import requests

Simulate data retrieval from a hypothetical data source (e.g., an API) url = "https://api.example.com/flooddata"

response = requests.get(url) data = response.json()

Topic 2: Feature Engineering Python

```
import pandas as pd
# Create a DataFrame from the retrieved data

df = pd.DataFrame(data)



# Example feature engineering

df['rolling_rainfall'] = df['rainfall'].rolling(window=7).sum()
df['river_level_diff'] = df['river_level'].diff()
```

Topic 3: Model Selection python

```
from sklearn.ensemble import RandomForestClassifier

# Define the machine learning model (Random Forest in this case)
model = RandomForestClassifier()
```

Topic 4: Model Training python

```
from sklearn.model_selection import train_test_split

# Split data into training and testing sets X = df[['rolling_rainfall',
'river_level_diff']]

y = df['flood_label']
```

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

Train the model model.fit(X_train, y_train)

Topic 5: Model Evaluation python

from sklearn.metrics import accuracy_score

Evaluate the model

y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred) print("Model accuracy:", accuracy)

Topic 6: Threshold Selection python

Define a flood threshold

flood_threshold = 0.7 # Adjust this threshold as needed based on historical data and risk tolerance

Topic 7: Real-Time Data Integration

In a real system, set up a data pipeline to continuously collect and update real-time data.

Topic 8: Monitoring and Alerting

In a real system, implement real-time monitoring and alerting mechanisms. Here's a simplified example using print statements:

python

while True:

new_data = get_new_data() # Implement a function to fetch real-time data prediction = model.predict(new_data)

if prediction >= flood_threshold:

print("Flood alert! Take necessary precautions.")

Topic 9: Testing and Validation

Conduct extensive testing and validation, comparing the system's performance against historical data and conducting simulated exercises.

Topic 10: Maintenance and Updates

Implement a plan for regular maintenance and updates, including data source updates, model retraining, and system improvements based on changing conditions.

PYTHON CODE

```
import time import random

# Simulated data source for demonstration def get_simulated_data():

# Generate random data (replace with actual data source) return {

'rainfall': random.uniform(0, 5),

'river_level': random.uniform(0, 3),

'flood_label': 0  # 0 means no flood, 1 means flood

}

# Feature engineering (simplified for demonstration) def feature_engineering(data):

rolling_rainfall = data['rainfall'] river_level_diff = data['river_level'] return rolling_rainfall, river_level_diff

# Machine learning model (simplified for demonstration) def predict_flood(rolling_rainfall, river_level_diff):

# Simple threshold-based prediction

if rolling_rainfall > 3.0 or river_level_diff > 1.5: return 1 # Flood alert
```

```
else:

return 0 # No flood alert



def main():

print("Flood Monitoring and Early Warning System (Simulation)\n")



while True:

# Simulate data retrieval data = get_simulated_data()

# Feature engineering

rolling_rainfall, river_level_diff = feature_engineering(data)



# Predict using a machine learning model

prediction = predict_flood(rolling_rainfall, river_level_diff)
```

```
# Display results

print(f"Rainfall: {data['rainfall']:.2f} inches") print(f"River Level:
{data['river_level']:.2f} meters")

if prediction == 1:

print("Flood Alert! Take necessary precautions.\n") else:
print("No flood alert.\n")

time.sleep(60) # Simulate real-time monitoring every 60 seconds

if name == " main ": main()
```

OUTPUT

Flood Monitoring and Early Warning System (Simulation)

Rainfall: 4.13 inches River Level: 1.27 meters

Flood Alert! Take necessary precautions.

Rainfall: 1.78 inches River Level: 0.62 meters No flood alert.

Rainfall: 3.60 inches River Level: 1.13 meters

Flood Alert! Take necessary precautions.

Rainfall: 0.92 inches River Level: 2.28 meters

Flood Alert! Take necessary precautions.

Rainfall: 2.34 inches

River Level: 1.85 meters

Flood Alert! Take necessary precautions.

... (continues with random data)

Conclusion:

Feature engineering is the development of new data features from raw data. With this technique, engineers analyze the raw data and potential information in orderto extract a new or more valuable set of features. Feature engineering can be seen as a generalization of mathematical optimization that allows for better analysis. Hope you learned about

feature engineering, its techniques and tools used by engineers. If you have any doubt regarding the article you can drop a comment. Thank You forthis wonderful opportunity