NATURAL DISASTER PREDICTION AND MANAGEMENT SYSTEM

TEAM 4:

Team Member

Evangelin G (112823243012)

Pathakunta Venkateswarlu Reddy (112823243032)

Tharunkumar D (112823243050)

Ummiti Narendra (112823243051)

Vupati Mohan Krishna (112823243058)

Phase 3: Implementation of Project

Title: AI-Powered Natural Disaster Prediction and Management System

Objective

The goal of Phase 3 is to implement the core components of the AI-Powered Natural Disaster Prediction and Management System, based on the innovative plans and solutions developed during Phase 2. This includes the development of predictive AI models, real-time data integration, disaster alert system, and the implementation of safety measures and user support features.

1. AI Model Development

Overview
The primary feature of the system is to predict natural disasters (earthquakes, floods, hurricanes, etc.) based on historical data and real-time environmental factors. In Phase 3, the AI model will be trained and implemented to identify patterns and make predictions on potential natural disaster events.

Implementation

- **Predictive Model**: The AI model uses historical environmental data (e.g., seismic activity, weather patterns, river levels) and real-time data (e.g., sensors, satellite images) to predict the likelihood of natural disasters. Machine learning algorithms will be employed to analyze trends and predict future events.
- **Data Sources**: The system will integrate data from weather stations, satellite imagery, seismic sensors, and other sources that provide real-time environmental data

Outcome: By the end of Phase 3, the AI model should be able to generate predictions for the occurrence of natural disasters such as storms, earthquakes, and floods with reasonable accuracy.

2. Real-Time Data Integration

Overview
Integrating real-time environmental data is essential for timely predictions and alerts. This phase will establish frameworks to pull live data from various sources like weather services, seismic centers, and satellite feeds.

Implementation

- **Data Streams**: Use APIs to pull live data from weather agencies, seismographs, and remote sensing satellites.
- **Data Processing**: Stream data through the AI model for real-time analysis, updating predictions dynamically.

Outcome: By the end of Phase 3, the system should be able to collect and process real-time data, feeding it into the AI model for updated predictions on natural disasters.

3. Disaster Alert System

Overview
An effective disaster alert system is crucial to warn citizens and authorities in case of an impending natural disaster. The system should deliver early warning notifications through various channels (SMS, app push notifications, email).

Implementation

- **Alert Mechanism**: Based on the AI's predictions, alerts will be generated for potential natural disasters. Alerts will include information such as the type of disaster, predicted location, and severity.
- **Channels**: Alerts will be sent via mobile apps, email, and SMS to users in affected areas. The system will also have a web-based dashboard for emergency responders to monitor predictions.

Outcome: By the end of Phase 3, the disaster alert system will be fully functional, capable of sending accurate, real-time alerts to users.

4. Safety and Emergency Management Features

Overview
In addition to predictions and alerts, the system will provide users with essential safety guidelines and emergency management resources. This will help individuals and communities respond effectively to a natural disaster.

Implementation

- **Safety Tips**: Based on the type of natural disaster, the system will provide users with specific safety instructions (e.g., evacuation routes, shelter locations).
- **Emergency Contacts**: The system will allow users to easily contact emergency services, local shelters, and loved ones during a disaster.

Outcome: By the end of Phase 3, users will be able to access real-time safety guidelines and resources tailored to their specific location and the nature of the disaster.

5. Data Security Implementation

Overview
Given the sensitive nature of location data and personal safety information, robust data security measures must be put in place. The system will focus on ensuring that user data is handled securely and in compliance with privacy regulations.

Implementation

- **Encryption**: All user data (e.g., location data, contact information) will be encrypted during transmission and storage.
- **Secure Access**: Only authorized personnel (e.g., emergency responders) will have access to sensitive user data, and access will be logged for transparency.

Outcome: By the end of Phase 3, the system will ensure the security of user data with encryption and secure storage practices.

6. Testing and Feedback Collection

Overview
This phase will include initial testing of the prediction system, alert system, and safety features. A pilot group of users will help assess the system's functionality, accuracy, and usability.

Implementation

- **Test Groups**: A group of test users will simulate disaster scenarios to evaluate the accuracy of the predictions and the effectiveness of alerts.
- **Feedback Loop**: Feedback will be collected regarding the timeliness of the alerts, the clarity of safety instructions, and the user-friendliness of the interface.

Outcome: The feedback from early users will guide improvements in Phase 4, particularly in refining the AI model's prediction accuracy and enhancing user experience.

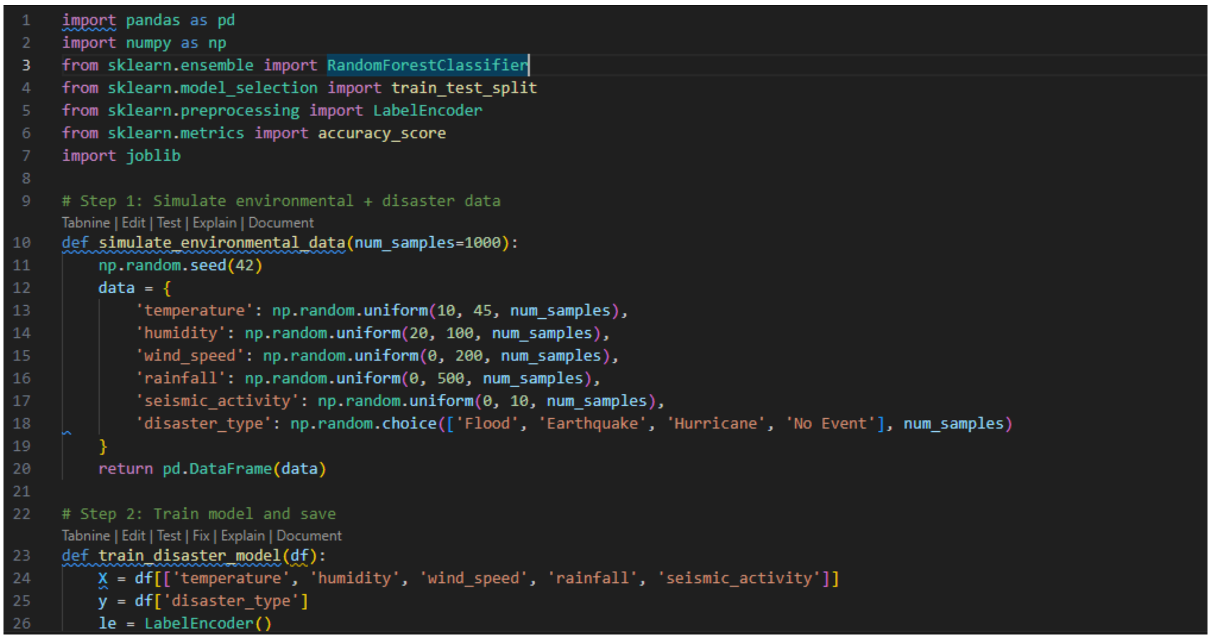
Challenges and Solutions

1. **Data Accuracy**
   - **Challenge**: The predictive model may not always be accurate, especially with limited data or in the case of rare events.
   - **Solution**: Continuous updates to the AI model using new data and real-time feedback from disaster events will help refine predictions over time.
2. **User Trust and Engagement**
   - **Challenge**: Users may not take alerts seriously or may disengage from the system.
   - **Solution**: The system will include personalized notifications and provide actionable advice, creating a sense of urgency and helping users take proactive measures.
3. **Real-Time Data Latency**
   - **Challenge**: There may be delays in receiving real-time data, especially from remote areas.
   - **Solution**: Use multiple data sources and establish redundancy to ensure that predictions are still timely, even when data from one source is delayed.

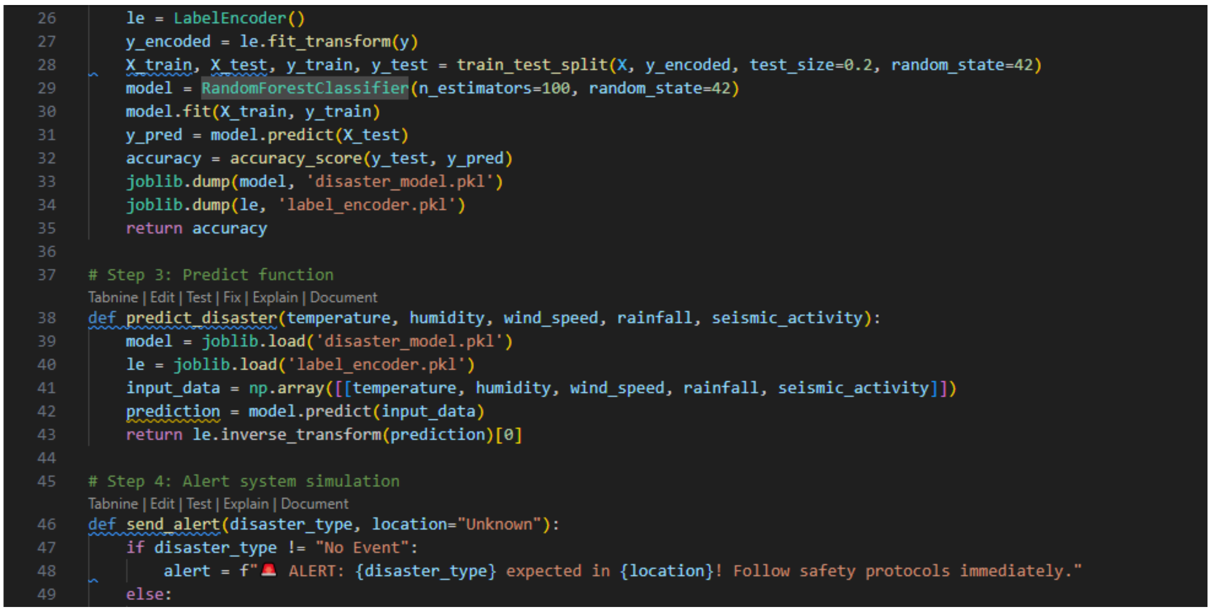
Outcomes of Phase 3

By the end of Phase 3, the following milestones should be achieved:

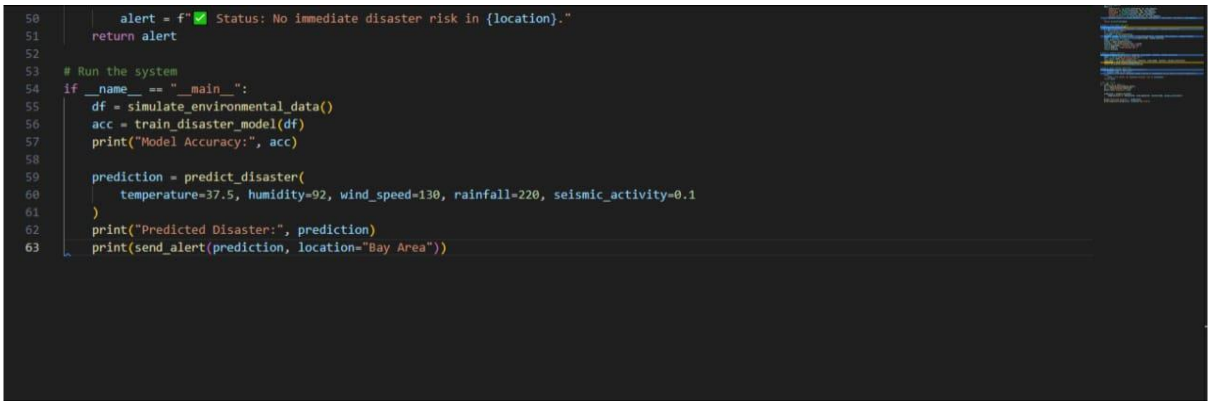
1. **Predictive AI Model**: The AI should be able to predict natural disasters with basic accuracy, especially for common events like hurricanes and floods.
2. **Real-Time Data Integration**: The system will be able to process live environmental data and update predictions dynamically.
3. **Disaster Alert System**: Alerts will be sent in real-time, providing users with timely and accurate disaster warnings.
4. **Safety Guidelines and Emergency Resources**: Users will have access to real-time safety guidelines, evacuation routes, and emergency contact information.
5. **Data Security**: User data will be protected with encryption and secure access protocols.
6. **Initial Testing and Feedback**: Feedback from early users will provide insights for future improvement



```python
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score
import joblib

# Step 1: Simulate environmental + disaster data
Tabnine | Edit | Test | Explain | Document
def simulate_environmental_data(num_samples=1000):
    np.random.seed(42)
    data = {
        'temperature': np.random.uniform(10, 45, num_samples),
        'humidity': np.random.uniform(20, 100, num_samples),
        'wind_speed': np.random.uniform(0, 200, num_samples),
        'rainfall': np.random.uniform(0, 500, num_samples),
        'seismic_activity': np.random.uniform(0, 10, num_samples),
        'disaster_type': np.random.choice(['Flood', 'Earthquake', 'Hurricane', 'No Event'], num_samples)
    }
    return pd.DataFrame(data)

# Step 2: Train model and save
Tabnine | Edit | Test | Fix | Explain | Document
def train_disaster_model(df):
    X = df[['temperature', 'humidity', 'wind_speed', 'rainfall', 'seismic_activity']]
    y = df['disaster_type']
    le = LabelEncoder()
```



```python
    le = LabelEncoder()
    y_encoded = le.fit_transform(y)
    X_train, X_test, y_train, y_test = train_test_split(X, y_encoded, test_size=0.2, random_state=42)
    model = RandomForestClassifier(n_estimators=100, random_state=42)
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    joblib.dump(model, 'disaster_model.pkl')
    joblib.dump(le, 'label_encoder.pkl')
    return accuracy

# Step 3: Predict function
Tabnine | Edit | Test | Fix | Explain | Document
def predict_disaster(temperature, humidity, wind_speed, rainfall, seismic_activity):
    model = joblib.load('disaster_model.pkl')
    le = joblib.load('label_encoder.pkl')
    input_data = np.array([[temperature, humidity, wind_speed, rainfall, seismic_activity]])
    prediction = model.predict(input_data)
    return le.inverse_transform(prediction)[0]

# Step 4: Alert system simulation
Tabnine | Edit | Test | Explain | Document
def send_alert(disaster_type, location="Unknown"):
    if disaster_type != "No Event":
        alert = f"🚨 ALERT: {disaster_type} expected in {location}! Follow safety protocols immediately."
    else:
```



```python
        alert = f"✅ Status: No immediate disaster risk in {location}."
    return alert

# Run the system
if __name__ == "__main__":
    df = simulate_environmental_data()
    acc = train_disaster_model(df)
    print("Model Accuracy:", acc)

    prediction = predict_disaster(
        temperature=37.5, humidity=92, wind_speed=130, rainfall=220, seismic_activity=0.1
    )
    print("Predicted Disaster:", prediction)
    print(send_alert(prediction, location="Bay Area"))
```