Big Data Analysis

A Project Submitted

By

AMAN GUPTA

Under the Guidance of

PROF. PRABAKARAN M V

DHANALAKSHMI SRINIVASAN

COLLEGE OF ENGINEERING AND TECHNOLOGY

ECR, MAMALLAPURAM. KANCHIPURAM DISTRICT.

Approved by AICTE, NEW DELHI | Affiliated to Anna University, Chennai.

Table of Contents:-

1. Introduction:-

This document represents Phase 2 of the & quot ; Big Data Analysis & quot ; project, submitted by Aman Gupta , a 3rd-year student of Computer Science and Engineering (CSE). Phase 2 introduces innovation by incorporating advanced machine learning algorithms for predictive analysis and anomaly detection, leveraging the
& quot ; rainfall in India 1901-2015" dataset.

2. Innovation:-

- Advanced Machine Learning

In Phase 2, the project takes a significant leap by integrating advanced machine learning techniques. These techniques will be applied to the existing dataset, & quot ; rainfall in India 1901-2015, & quot ; to enhance the project's analytical capabilities.

Objective:
- Predictive Analysis: The project aims to develop accurate predictive models using advanced machine learning algorithms. These models will forecast future rainfall trends, providing valuable insights for agricultural planning and water resource management.
Approach:
- Data Preparation: The dataset will undergo preprocessing, including handling missing values, time series decomposition, and feature engineering.

- Algorithm Selection: State-of-the-art machine learning algorithms suitable for time series forecasting, such as Long Short-Term Memory (LSTM) and Random Forest, will be considered for
model development.

3. Dataset:-

The dataset used for this phase remains "rainfall in India 1901-2015." This dataset contains historical rainfall records for various regions in India, making it a suitable choice for advanced analysis and modeling.

4. Machine Learning Implementation:-

----------Python code----------

- Data Preparation

```
# Import necessary libraries
import pandas as pd
import numpy as np

# Load the dataset
data = pd.read_csv('rainfall_india_1901-2015.csv')

# Perform data preprocessing (e.g., handling missing values, feature engineering)
# ...

# Split the dataset into training and testing sets
from sklearn.model_selection import train_test_split

train_data, test_data = train_test_split(data, test_size=0.2, random_state=42)
```

- Algorithm Selection

```
# Select machine learning algorithms for time series forecasting
from sklearn.ensemble import RandomForestRegressor
from tensorflow import keras
from tensorflow.keras import layers

# Initialize models
random_forest_model = RandomForestRegressor()
lstm_model = keras.Sequential([
    layers.LSTM(units=64, activation='relu', input_shape=(n_steps, n_features)),
    layers.Dense(1)
])
```

- Predictive Modeling

```
# Import necessary libraries
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Generate synthetic data (Replace this with loading and preprocessing your actual data)
np.random.seed(0)
X = np.random.rand(100, 1) * 10
y = 2 * X + 1 + np.random.randn(100, 1)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create and fit the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions on the test data
y_pred = model.predict(X_test)

# Evaluate the model's performance
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse}")
print(f"R-squared: {r2}")

# Plot the actual vs. predicted values
plt.scatter(X_test, y_test, color='blue', label='Actual')
plt.plot(X_test, y_pred, color='red', linewidth=3, label='Predicted')
plt.xlabel('X')
plt.ylabel('y')
plt.legend()
plt.show()
```

5. Anomaly Detection:-

The advanced machine learning models can also be utilized for anomaly detection. By analyzing the residuals between predicted and actual rainfall values, unusual variations in rainfall patterns can be identified, helping to mitigate risks associated with extreme weather events.

```
- Anomaly detection using residuals
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
# Load the dataset (use your dataset)
data = pd.read_csv('rainfall_india_1901-2015.csv')
# Split the dataset into training and testing sets
train_data, test_data = train_test_split(data, test_size=0.2, random_state=42)

# Feature selection and preprocessing
X_train = train_data[['Year', 'Month', 'Region']] # Adjust features as needed
y_train = train_data['Rainfall']

X_test = test_data[['Year', 'Month', 'Region']]
y_test = test_data['Rainfall']

# Initialize and train a machine learning model (Random Forest, for example)
model = RandomForestRegressor()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate residuals (difference between actual and predicted values)
residuals = y_test - y_pred

# Define a threshold for anomaly detection (adjust as needed)
threshold = 2.0 # Example threshold, you can fine-tune this value

# Identify anomalies based on residuals
anomalies = np.where(np.abs(residuals) > threshold)[0]

# Print the indices of anomalous data points
print("Anomalous Data Points:")
print(anomalies)
```

For all these Operations, we need to connect our IBM DB2 to our VS Code, the following is the procedure to connect the DB2 to VS Code :

Step 1: Install the IBM Db2 Extension for VS Code

1. Open Visual Studio Code.

2. Go to the Extensions view by clicking on the Extensions icon in the Activity Bar on the side of VS Code or by pressing `Ctrl+Shift+X`.

3. Search for "IBM Db2" in the Extensions view search bar.

4. Click the "Install" button next to the "IBM Db2" extension offered by IBM.

Step 2: Configure the IBM Db2 Connection

1. After installing the extension, you will see a new IBM Db2 icon in the sidebar. Click on it.

2. In the "IBM Db2" panel, click the gear icon (⚙) to open the settings.

3. Click on the "Add Connection" button to create a new connection.

4. Fill in the connection details:
   - Connection Name: Give your connection a name for reference.
   - Host Name: The hostname or IP address of your IBM Db2 instance.
   - Port: The port on which your Db2 instance is listening (typically 50000).
   - Database Name: The name of the database you want to connect to.
   - User Name: Your Db2 username.
   - Password: Your Db2 password.

5. Click the "Test Connection" button to ensure the connection works correctly.

6. Click the "Save" button to save the connection.

Step 3: Connect to the IBM Db2 Database

1. In the "IBM Db2" panel, you will now see your saved connection.

2. Click on your connection to connect to the database.

Step 4: Use SQL to Query the Database

1. Once connected, you can right-click on the connected database and select "New SQL File" to open a new SQL editor.

2. Write your SQL queries in the editor.

3. Click the "Run Query" button to execute the query against the connected database.

That's it! We've successfully connected to our IBM Db2 database from VS Code and can now use SQL to interact with the database.

6. Conclusion:-

Phase 2 of the "Big Data Analysis" project, led by Me(Aman Gupta), demonstrates a significant advancement by incorporating advanced machine learning techniques. The integration of predictive analysis and anomaly detection using the "rainfall in India 1901-2015" dataset enhances the project's capabilities, allowing for more accurate forecasting and risk mitigation. This phase marks a critical step in harnessing the full potential of big data and demonstrates the project's commitment to innovative data analysis.