Phase 5: Project Development

5.1 Overview

The Project Development phase focuses on implementing the rainfall analysis and prediction system using Python and related technologies. This phase includes data preprocessing, exploratory data analysis (EDA), machine learning model training, model saving, and development of a Flask-based web application for deployment.

The development was carried out using Jupyter Notebook for analysis and Flask for building the web interface.

5.2 Development Steps

Step 1: Import Required Libraries

The following Python libraries were used during development:

- **pandas** – For data loading, manipulation, and preprocessing
- **numpy** – For numerical computations
- **matplotlib.pyplot** – For data visualization (graphs and plots)
- **seaborn** – For advanced statistical visualizations
- **scikit-learn (sklearn)** – For machine learning model building
    - LinearRegression
    - train_test_split
    - mean_squared_error
    - r2_score
- **pickle** – For saving and loading the trained model
- **flask** – For developing the web application
- **os** – For handling file paths and system operations

Example import statements:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import pickle
from flask import Flask, render_template, request
import os
```

Step 2: Load Rainfall Dataset

The rainfall dataset was downloaded from Kaggle and loaded into the system using pandas.

```
df = pd.read_csv("rainfall.csv")
```

The dataset contains state-wise rainfall data across multiple years.

Step 3: Data Cleaning and Preprocessing

Data preprocessing was performed to ensure accuracy and reliability.

- Checked dataset structure using `df.info()` and `df.describe()`
- Removed extra spaces in column names
- Handled missing values using mean imputation
- Converted dataset from wide format to long format using `melt()` function

```
df.columns = df.columns.str.strip()
df.fillna(df.mean(numeric_only=True), inplace=True)
```

This step ensured that the dataset was clean and ready for analysis.

Step 4: Perform Exploratory Data Analysis (EDA)

Exploratory analysis was performed to understand rainfall patterns.

The following analyses were conducted:

- State-wise average rainfall
- Year-wise rainfall trend
- Rainfall distribution analysis
- Identification of highest and lowest rainfall states

Visualization techniques used:

- Line plots
- Bar charts
- Histograms

EDA helped in identifying rainfall variations across states and years.

Step 5: Train Machine Learning Model

To predict rainfall trends, a machine learning model was developed.

The dataset was divided into training and testing sets:

```
X = df_long[['Year']]
y = df_long['Rainfall']

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

A **Linear Regression model** was trained:

```
model = LinearRegression()
model.fit(X_train, y_train)
```

Model performance was evaluated using:

- Mean Squared Error (MSE)

- R² Score

Step 6: Save the Trained Model Using Pickle

After training, the model was saved using the pickle library for future use in the web application.

```
with open('rainfall_model.pkl', 'wb') as file:
    pickle.dump(model, file)
```

This allows the model to be reused without retraining.

Step 7: Develop Flask Web Application

A Flask web application was created to provide a user-friendly interface.

The project structure included:

- `app.py` – Backend logic
- `templates/index.html` – Frontend interface
- `static/` – CSS and other assets
- `rainfall_model.pkl` – Saved trained model

Flask was used to:

- Load the trained model
- Accept user input (year/state)

- Display rainfall prediction results

Step 8: Deploy the Model for Prediction

The trained model was integrated into the Flask application to provide real-time predictions.

When the user enters input data:

1. The input is processed.
2. The saved model is loaded using pickle.
3. The prediction is generated.
4. The result is displayed on the webpage.

This completes the deployment process.

5.3 Outcome of Development Phase

The development phase successfully implemented:

- Rainfall data preprocessing
- Exploratory data analysis
- Machine learning-based prediction
- Model saving and reuse
- Web-based rainfall prediction system

The system is capable of analyzing rainfall patterns and providing predictions through an interactive web interface.