Software Requirements Specification

For

Weather Prediction System

20th October 2024

Prepared by

Specialization	SAP ID	Name
B.Tech AIML(H) B-1	500107437	Gaurav Chhajer
B.Tech AIML(H) B-1	500106904	Konal Puri

School Of Computer Science
UNIVERSITY OF PETROLEUM & ENERGY STUDIES,
DEHRADUN- 248007. Uttarakhand

Table of Contents

Revision History

Date	Change	Reason for Changes	Mentor Signature
Nil	No change	Nil	

1. Introduction

1.1 Purpose of the Project

The purpose of this project is to develop a predictive model for rainfall based on historical weather data. The model aims to accurately predict whether it will rain tomorrow (`RainTomorrow_endoced`) using various machine learning algorithms, with a focus on

maximizing performance and accuracy. This project will assist meteorological services, farmers, and urban planners in making weather-related decisions.

1.2 Target Beneficiary

- **Farmers**: To plan irrigation and farming activities.
- **Meteorological Departments**: To improve weather forecasting systems.
- **Transportation Authorities**: To anticipate and plan for rain-related delays and accidents.
- **General Public**: To enhance daily planning by providing accurate rainfall forecasts.

1.3 Project Scope

The project will use supervised learning models such as Logistic Regression, Random Forest, Gradient Boosting, SVM, and XGBoost to predict rainfall. The project scope includes data collection, preprocessing, model training, evaluation, and deployment of the most accurate model.

1.4 Assumptions and Dependencies

- The weather data is assumed to be reliable and sufficiently large for training.
- Dependencies include libraries such as `sklearn`, `joblib`, `pandas`, and `xgboost`, along with cloud services for deployment like AWS EC2.
- Accurate predictions are dependent on weather patterns remaining consistent with historical data.

2. Project Description

2.1 Reference Algorithm

Various machine learning algorithms, including Logistic Regression, Random Forest, Gradient Boosting, SVM, and XGBoost, are applied to predict the target variable `RainTomorrow_endoced`. Each model's performance is evaluated based on metrics like accuracy, precision, recall, and ROC-AUC score.

2.2 Data/ Data Structure

The dataset includes the following key features: `MinTemp`, `MaxTemp`, `Rainfall`, `Humidity9am`, `WindSpeed9am`, `Pressure9am`, and categorical variables like `WindGustDir`. The target variable is `RainTomorrow_endoced`.

2.3 SWOT Analysis

- **Strengths**:

    - Accurate weather predictions assist multiple sectors.
    - Large dataset with historical data improves model robustness.
- **Weaknesses**:
    - Complexity in handling missing or imbalanced data.
    - Model performance highly dependent on data quality.
- **Opportunities**:
    - Expansion to real-time weather monitoring and prediction.
    - Integration with IoT-based weather sensors.
- **Threats**:
    - Sudden weather pattern changes may reduce model accuracy.
    - Data privacy and usage concerns related to public datasets.

2.4 Project Features

- **Weather Data Preprocessing**: Cleaning and encoding weather features like wind direction and temperature for better model training.
- **Multiple Model Evaluation**: Applying and comparing five different models to find the best performer.
- **Model Persistence**: Saving the models using `joblib` for future use.
- **Performance Metrics Calculation**: Evaluating models using accuracy, precision, recall, F1-score, confusion matrix, and ROC-AUC.

2.5 User Classes and Characteristics

- **Developers**: Implement and maintain the models.
- **Data Scientists**: Tune and improve model accuracy.
- **End Users (Farmers, Meteorologists)**: Access predictions through a simplified user interface.
- **Administrators**: Monitor the deployment environment and ensure model availability.

2.6 Design and Implementation Constraints

- **Hardware Constraints**: Model training may require high computational power for larger datasets.
- **Software Constraints**: Libraries such as `sklearn` and `xgboost` need to be compatible with the deployment environment.
- **Data Constraints**: Missing or incorrect data may reduce model performance.

2.7 Design Diagrams

- **Data Flow Diagram**: Illustrating the process from data ingestion, preprocessing, model training, to prediction.
- **UML Class Diagrams**: Representing the relationships between model classes and data.
- **System Architecture Diagram**: Showcasing the interaction between front-end, back-end, and the machine learning pipeline.

3. System Requirements

3.1 User Interface

- **Web Interface**: Simple form to input weather parameters for the prediction of rainfall.

3.2 Software Interface

- **AWS EC2**: For deploying models in the cloud.
- **Flask API**: Used for exposing the models as APIs for front-end integration.
- **GitHub**: Version control for code management.

3.3 Database Interface

- **MySQL/SQLite**: Storing historical data and user interaction logs.
- **S3**: Storing and accessing large datasets efficiently.

3.4 Protocols

- **HTTP/HTTPS**: For communication between the front-end and back-end (API calls).
- **REST API**: For making prediction requests and receiving responses from the deployed model.

4. Non-functional Requirements

4.1 Performance Requirements

- The system should predict results in real-time, with a latency of less than 1 second for API responses.
- The model's accuracy should be above 85% based on validation data.

4.2 Security Requirements

- Data should be encrypted in transit and at rest.
- Ensure role-based access control for system administrators and users.

4.3 Software Quality Attributes

- **Maintainability**: Code should be modular to allow for easy updates to the model or data.
- **Scalability**: The system should handle increasing amounts of data without performance degradation.
- **Usability**: The interface should be simple for non-technical users.

5. Other Requirements

- The system should include a monitoring tool to track the performance of models over time and detect potential drifts in predictions.

Appendix A: Glossary

- **ROC-AUC**: Receiver Operating Characteristic - Area Under the Curve.
- **Joblib**: Python library for saving machine learning models.
- **RainTomorrow_endoced**: Target variable for predicting whether it will rain tomorrow.
- **AWS EC2**: Amazon Web Services Elastic Compute Cloud, used for deploying machine learning models.

Appendix B: Analysis Model

A detailed analysis of the feature importance for each machine learning model, highlighting which weather parameters have the highest influence on rainfall prediction.

Appendix C: Issues List

- Handling missing data in the dataset.
- Addressing class imbalance (i.e., fewer rain days compared to non-rain days).
- Fine-tuning models to avoid overfitting on training data.

References

- Scikit-learn Documentation
- XGBoost Library Documentation
- AWS EC2 Documentation
- Historical weather datasets (e.g., weatherAUS.csv)
- Relevant machine learning libraries documentation, including scikit-learn, pandas, NumPy, and seaborn.
- Best practices in data science and machine learning model development literature.