TeamX

Weather Forecasting for Smart Agriculture

Question 1 Documentation

March 9, 2025

1 Introduction

This project predicts daily rain occurrence using historical weather data collected over more than 300 days. The primary objective is to generate a 21-day rain probability forecast, which is critical for agricultural planning.

Key Aspects:

- **Dataset Features:** Average temperature, humidity, average wind speed, cloud cover, pressure, and date (from which month, day, and weekday features are extracted).
- **Target Variable:** `rain_or_not` is encoded as 1 for rain and 0 for no rain.
- **Approach:** Data cleaning and imputation, exploratory data analysis (EDA), logistic regression model training and hyperparameter tuning, and generation of future predictions.

2 Data Preprocessing

2.1 Data Loading

The weather dataset is loaded from a CSV file using `pandas`. Initial checks (using `head()`, `info()`, and `isnull().sum()`) reveal missing values in several numeric columns.

2.2 Data Cleaning

- **Date Conversion:** Convert the date column to a datetime object.
- **Handling Missing Values:** Impute missing values in `avg_temperature`, `humidity`, `avg_wind_speed`, and `cloud_cover` using column means.
- **Target Variable Encoding:** Convert `rain_or_not` to 1 for "Rain" and 0 for "No Rain".
- **Feature Engineering:** Extract month, day, and weekday from the date.

3 Exploratory Data Analysis (EDA)

- **Visualizations:** Histograms and box plots (using `matplotlib` and `seaborn`) are used to visualize the distribution of weather features and their relationship with the target variable.
- **Correlation Analysis:** A correlation heatmap is generated to assess the relationships among features and the target variable.

4 Model Training & Evaluation

4.1 Data Splitting

The dataset is split into training and testing sets (typically 80% training, 20% testing).

4.2 Baseline Model

A logistic regression model is used as the baseline for its simplicity and interpretability. Evaluation metrics such as precision, recall, F1-score, and ROC-AUC are used to assess performance.

5 Hyperparameter Tuning

GridSearchCV is employed to optimize logistic regression hyperparameters (e.g., the regularization parameter `C` and solver type). The best model is selected based on ROC-AUC scores.

6 Generating Future Predictions

- **Future Data Creation:** A DataFrame with 21 future dates is generated. Simulated or forecasted weather data is used to create the necessary features.
- **Prediction:** The tuned logistic regression model predicts rain probabilities for the next 21 days.

7 Suggestions for Improvement

- **Enhanced Feature Engineering:** Use time-series features (e.g., lag variables, rolling averages) and incorporate external data sources (regional forecasts, seasonal indices).
- **Model Alternatives:** Explore ensemble methods (Random Forest, XGBoost) or neural networks if data volume increases.
- **Data Quality:** Improve sensor calibration and redundancy to reduce noise; increase data collection frequency.
- **Advanced Techniques:** Investigate time-series models (e.g., ARIMA, Prophet) for capturing temporal dependencies.

8 Reproducibility & Running Instructions

8.1 Pre-requisites

- **Python Environment:** Python 3.7 or later.
- **Dependencies:** Install required libraries using pip:

```
pip install pandas numpy matplotlib seaborn scikit-learn tabulate
```

8.2 Steps to Run the Program

1. **Download the Dataset:** Place the file `weather_data.csv` in your working directory.
2. **Open the Notebook or Script:** Use the provided Jupyter Notebook (or Python script) structured into sections:
   - Data Loading and Preprocessing
   - Exploratory Data Analysis (EDA)
   - Model Training and Evaluation
   - Hyperparameter Tuning
   - Future Predictions
3. **Run the Notebook Sequentially:** Execute each cell step-by-step. Verify the output at each stage.
4. **Review Outputs:** The notebook will display classification metrics (accuracy, precision, recall, F1-score, ROC-AUC) and a formatted table of 21-day rain probability predictions.
5. **Modifications:** To experiment with different scenarios or extend the prediction horizon, adjust parameters in the Future Data section.

9 Libraries & Dependencies

- **Pandas** for data manipulation.
- **NumPy** for numerical operations.
- **Matplotlib & Seaborn** for data visualization.
- **scikit-learn** for model building, evaluation, and hyperparameter tuning.
- **Tabulate** (optional) for pretty-printing output tables.

10 Conclusion

This documentation outlines the complete workflow for predicting rain using historical weather data. It covers data preprocessing, exploratory analysis, model training with logistic regression (including hyperparameter tuning), and the generation of 21-day predictions. Although the baseline model achieved a ROC-AUC of around 0.60, further improvements are possible through enhanced feature engineering, exploration of alternative models, and integration of advanced time-series techniques.

By following the reproducibility instructions, you can run the program, reproduce the results, and further experiment with the pipeline to adapt it to specific smart agriculture needs.