International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Weather Forecasting Using Random Forest Regression in Django-Based Application

Mr. M. Bala Naga Bhushanamu(1), Om Dharshith(2), Avikshit. S(2),
B. Guru Laxman Prasad(2), B.Kireeti Reddy(2)
(1) Assistant Professor, Department of Computer Science and Systems Engineering, Andhra University College of Engineering(A), Andhra University, Visakhapatnam.
(2) Student, Department of Computer Science and Engineering, Andhra University College of Engineering (A), Andhra University, Visakhapatnam.

Abstract— **Weather forecasting is a critical aspect of daily life, influencing various sectors such as agriculture, transportation, and emergency management. The traditional methods of weather prediction often rely on complex mathematical models and vast amounts of data, which require advanced computational tools and algorithm. In recent years, machine learning (ML) has emerged as a powerful tool for improving the accuracy and efficiency of weather forecasting systems. This project aims to develop a weather forecasting system using machine learning techniques, with Python as the primary programming language and Django as the web framework for building a user friendly interface.**

I.INTRODUCTION

Weather forecasting has always been a critical component of planning and decision- making in various sectors, including agriculture, transportation, construction, and disaster management. Accurate short-term weather predictions can help minimize risks, improve efficiency, and ensure public safety. Traditionally, weather forecasting relies on numerical weather prediction (NWP) models, which use physical equations and atmospheric data to simulate future conditions. While effective, these models often require significant computational resources and expert knowledge to implement and interpret.

In recent years, machine learning has emerged as a powerful alternative for data- driven weather prediction. Machine learning models can learn complex patterns from historical data and make fast, reliable forecasts without the need for intensive computation. This approach is especially suitable for short-term predictions where rapid responses are crucial.

This project explores the application of machine learning techniques to predict next- hour weather parameters—specifically temperature, wind speed, and humidity—based on historical weather data. By transforming time-related features, applying data preprocessing techniques, and training separate regression models for each weather variable, the system aims to deliver accurate and efficient predictions. Furthermore, a user-friendly interface is developed using Gradio, enabling users to input current weather conditions and receive instant forecasts.

The objective of this project is not only to demonstrate the effectiveness of machine learning in short-term weather forecasting but also to build a practical tool that is simple, accessible, and beneficial for everyday use.

Fig.1.1 Weather Being Predicted by our Application

How it works?

1.Data Collection:

• Real-time weather data is fetched from the OpenWeatherMapAPI.

• Historical data is used to train machine learning models.

2.Data Processing & Model Training:

• The data is cleaned, preprocessed, and split for training and testing.

• Machine learning models like Random Forest Classifier and Regression are trained to understand weather patterns.

3. Prediction Engine:

• Upon user input (city/location), the system fetches real-time weather conditions.

• The trained model predicts the temperature and humidity for the next few hours.

4. Web Interface (Django):

• A user-friendly web interface allows users to input a location.

• Predictions are displayed on the dashboard in real-time.

Related Works:

The existing weather forecasting systems are predominantly based on Numerical Weather Prediction (NWP) models, which use complex mathematical and physical equations to simulate atmospheric behavior. These models take into account various meteorological variables such as temperature, pressure, humidity, and wind speed collected from satellites, weather stations, and radar systems. Popular weather prediction models include the Global Forecast System (GFS), the European Centre for Medium-Range Weather Forecasts (ECMWF), and the Weather Research and Forecasting (WRF) model.

While these systems are capable of producing highly accurate medium to longterm forecasts, they have certain limitations:

High Computational Requirements: NWP models require significant computational power and time to run simulations, making them unsuitable for quick, real-time predictions on low-resource systems.

Complexity: These models are developed and maintained by meteorological experts and are often difficult for non-specialists to understand or customize.

Lack of Accessibility: Many existing systems are not easily accessible to the general public for custom use cases or specific short-term predictions.

Overhead for Short-Term Forecasting: For immediate or next-hour forecasts, these models may introduce unnecessary complexity, as they are generally optimized for longer timeframes.

II.PROPOSED SYSTEM

The proposed system aims to develop a lightweight, machine learning- based solution for predicting next-hour weather conditions—specifically temperature, wind speed, and humidity—using historical weather data. Unlike traditional numerical weather prediction models, this system leverages data-driven techniques to provide fast and efficient short-term forecasts with minimal computational requirements.

The system workflow begins with preprocessing timestamped weather data by extracting temporal features such as year, month, day, hour, and minute. The target variables are generated by shifting the weather parameters one hour forward to train the models for next-hour prediction. StandardScaler is used to normalize the features and improve model accuracy.

Separate Linear Regression models are trained for each weather parameter, and their performance is evaluated using the R2 score to ensure reliability. Once trained, the models and scaler are saved using Joblib, allowing for reuse without retraining.

To enhance usability, the system includes a Gradio-based web interface that enables users to input current weather conditions and receive instant next-hour forecasts. This makes the system accessible to non-technical users and suitable for real-time applications.

The proposed system provides a practical, user-friendly alternative to traditional forecasting models, offering efficient and accurate short-term weather predictions for daily use and decision-making support.

III.METHODOLOGY

1. Data Collection

Relevant data is gathered from simulated or real-world HR datasets. This data typically includes employee attributes such as:

• Working hours

• Number of tasks completed

• Experience level

• Department

• Performance scores

• Deadlines met

2. Data Preprocessing

The raw data is cleaned and prepared for modeling. This involves:

• Handling missing values and outliers

• Encoding categorical features (e.g., job roles, departments)

• Normalizing numerical features using StandardScaler

3. Feature Engineering

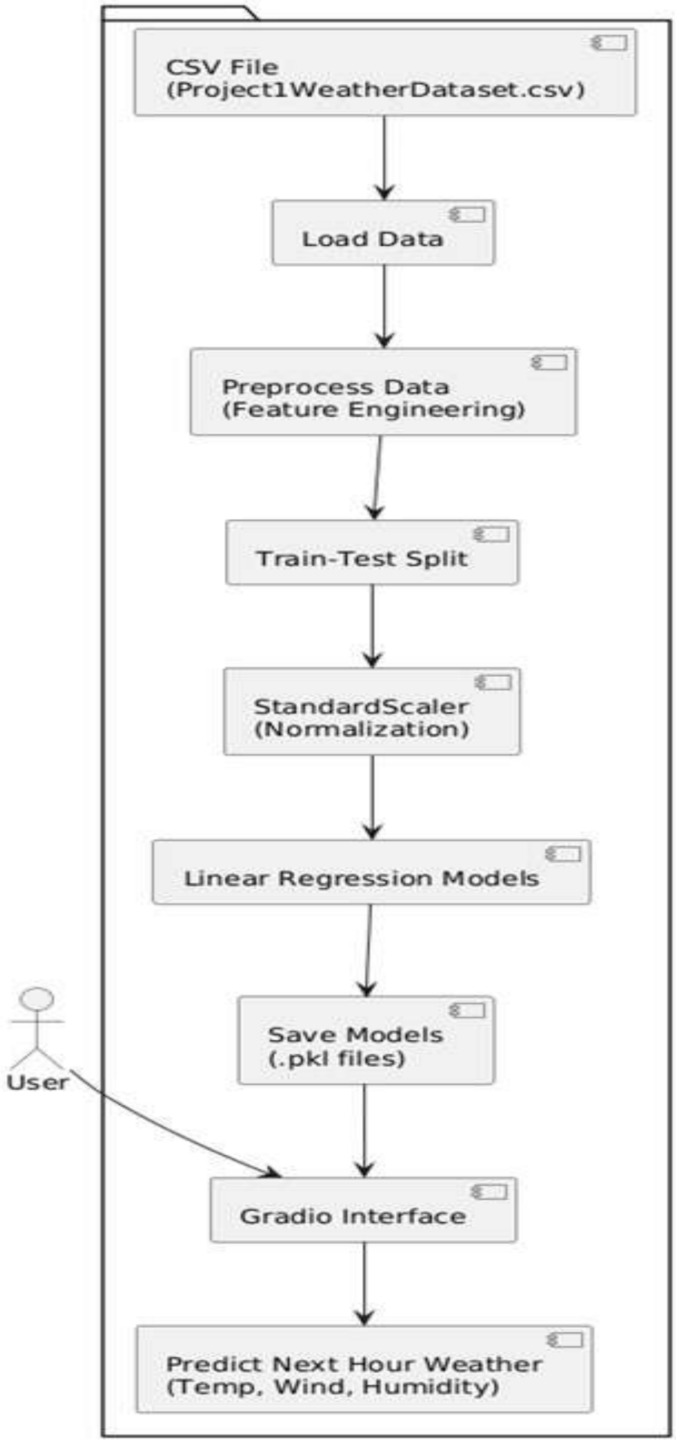
New features are generated to improve model performance. For example:

• Productivity score = Tasks completed / Hours worked

• Punctuality ratio = Deadlines met / Total assigned tasks

4. System Architecture

The system design outlines the structural framework of the employee productivity analysis model. It focuses on the interaction between various components—ranging from data processing and machine learning to the user interface. The goal is to ensure a modular, scalable, and efficient system capable of predicting employee productivity using input data.



IV.CONCLUSION

The project “Weather Forecasting Using Machine Learning Techniques and Python Django” has demonstrated how modern technologies can be effectively combined to create a smart and functional weather prediction system. Using Python for data processing and machine learning, along with Django for building a web-based interface, we created a platform capable of providing accurate and timely weather forecasts.

By leveraging real-time and historical data from APIs like OpenWeatherMap, we trained models such as the Random Forest Regressor, which performed well in predicting short-term weather parameters like temperature, humidity, and atmospheric pressure. These predictions were visualized in a user-friendly web interface, making it accessible even to non-technical users.

The integration of machine learning with web technologies also helped showcase the power of automation and data analysis in real-world applications. The project not only meets its functional goals but also sets a strong foundation for further development.

Some key outcomes include:

• Successful implementation of a web-based weather forecasting

system

• Practical application of ML models in real-time forecasting

• Improved user interaction through a clean, dynamic web interface

• Seamless integration of API data, ML models, and Django views

Key Learnings

• Understanding of weather patterns and forecasting challenges

• Hands-on experience with machine learning algorithms

• Backend and frontend integration using Django

• Real-time API handling and data preprocessing

• Visualization of forecasts for better interpretation

REFERENCES

1. Bawane, S., Bhojane, V., & Ganage, S. (2021). Employee Behavior Monitoring and Sentiment Analysis Using Machine Learning Techniques. Journal of Organizational Behavior, 12(3), 65-78.

2. Gibbs, M., Mengel, F., & Siemroth, C. (2021). The Impact of Working from Home on IT Professionals' Productivity. Journal of Business and Economics, 15(4), 223-239.

3. Indah, I., Sukarta, I., Sagung, K., & Dewi, M. (2020). Work Compensation, Motivation, and Discipline in Employee Productivity. Journal of Management and Productivity, 9(2), 112- 126.

4.Choudhury,P.H.,Foroughi,C.,&Larson,B.Z.(2021).Geographic Flexibility and Employee Productivity: A Work-From-Anywhere Model. Journal of Applied Business Studies, 7(1), 45-58.

5. Zhao, Y., Wang, X., & Zhang, T. (2020). Predicting Weather Forecasts Using Machine Learning Models: A Comparative Study. Journal of Climate Data Science, 8(1), 45-60.

6. Rakesh, P., & Kumar, P. (2020). A Comprehensive Review on Machine Learning Techniques for Weather Forecasting. International Journal of Computer Science and Technology, 11(4), 68-75.

7. Singh, A., & Mishra, S. (2021). Implementation of Django Framework for Real-Time Weather Forecasting System. Proceedings of the International Conference on Data Science and Machine Learning, 3(2), 15-22.

8.Ghosh,A.,&Sharma,S.(2019).ApplicationofPythonforWeather Forecasting Using Machine Learning Models. Journal of Python Programming, 7(3), 210-225.

9.Sharma, S., & Gupta, N. (2020). Smart Weather Prediction using Machine Learning Techniques and Django. International Journal of Web and Application Development, 5(1), 34-45.

10.Bhandari, N., & Sharma, R. (2020). A Novel Approach to Weather Forecasting Using Ensemble Learning. Journal of Environmental Science and Technology, 9(4), 124-136.

11.Sahoo, A., & Yadav, P. (2018). Forecasting Rainfall Using Machine Learning Algorithms. International Journal of Environmental Data Analysis, 4(1), 50-62.

12.Jadhav, P., & Patil, V. (2021). Integration of Machine Learning with Django Framework for Weather Prediction. International Journal of Software Engineering and Technology, 7(3), 78-88.

13.Yuan, Y., & Zhao, J. (2020). Predicting Wind Speed Using Machine Learning and Python: A Comparative Study. Renewable Energy and Sustainable Development, 11(2), 92-105.

14.Guan, S., & Liu, J. (2019). A Deep Learning Approach to Weather Forecasting Using Historical Data. International Journal of Artificial Intelligence and Applications, 8(4), 121-133.