

**A**

**MINI-PROJECT REPORT**

**ON**

**Predictive Analysis of Weather Patterns Using Machine Learning (XGBoost Model)**

**Computer Science and Engineering**

**Walchand Institute of Technology (An Autonomous Institute)**

By

**Aditi Deshpande A-06**

**Trushika Basude A-11**

**Pragati Andewadi A-12**

**Ojas Dhotre A-23**

**Pragati Chavan A-35**

Under Guidance of

**Mr. V. D. Chavan**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**WALCHAND INSTITUTE OF TECHNOLOGY**

**SOLAPUR – 413006**

**(2024-2025)**



**CERTIFICATE**

This is to certify that the Mini-Project entitled

**Predictive Analysis of Weather Patterns Using Machine Learning (XGBoost Model)**

Is

Submitted By

**Aditi Deshpande A-06**

**Trushika Basude A-11**

**Pragati Andewadi A-12**

**Ojas Dhotre A-23**

**Pragati Chavan A-35**

**(Dr. B. R. Solunke) (Dr. Mrs. A. M. Pujar)**

*Project Guide*  *Head*

*Dept of Computer Sc.& Engg*

**(Dr. V. A. Athavale)**

*Principal*

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr no.** | **Topic** | **Page no.** |
| 1 | Abstract | 1 |
| 2 | Introduction | 2 |
| 3 | Problem statement | 3 |
| 4 | Scope Of The Project | 4 |
| 5 | Dataset | 5 |
| 6 | Technologies Used | 7 |
| 7 | Architectural Components | 8 |
| 8 | Methodology | 9 |
| 9 | Implementation | 11 |
| 10 | Results | 12 |
| 11 | Testing Report | 13 |
| 12 | Future Scope | 15 |
| 13 | Conclusion | 16 |
| 14 | References | 17 |

**1. Abstract**

The purpose of this project is to analyze weather data and develop a predictive model capable of forecasting future weather conditions based on historical patterns. Weather prediction plays a crucial role in several fields such as agriculture, transportation, disaster management, aviation, and daily decision-making. Accurate forecasting helps in planning, risk reduction, and improving safety. The dataset used in this project contains multiple meteorological features such as temperature, humidity, visibility, wind speed, pressure, and weather conditions recorded over time.

The project begins with data preprocessing, which includes handling missing values, removing inconsistencies, normalizing feature scales, and converting categorical data into numerical form. Exploratory Data Analysis (EDA) was performed to understand the distribution of variables and correlations between them. Various visualizations such as heatmaps, line plots, scatter plots, and distribution graphs were used to identify underlying patterns and trends in the dataset.

After analysis, multiple machine learning algorithms were tested and compared to build a prediction model. The performance of each model was evaluated using accuracy, precision, recall, and other performance metrics. The best-performing model was selected and further optimized to improve its prediction ability. Finally, the trained model was deployed using Streamlit to create a user-friendly web application. This application enables users to input weather parameter values and instantly receive predicted weather conditions.

This project demonstrates a complete end-to-end machine learning pipeline—from data acquisition and preprocessing to analysis, model training, evaluation, and deployment.

**2. Introduction**

Weather prediction has always been an essential aspect of human planning and decision-making. From agriculture and transportation to event planning and public safety, accurate weather forecasting significantly influences daily activities and long-term strategies. With the rise of data-driven technologies and machine learning, modern weather prediction systems have become more reliable, efficient, and automated. Machine learning models can identify hidden patterns in large volumes of historical weather data that may not be apparent through traditional statistical methods.

In this project, a weather dataset containing multiple atmospheric parameters such as temperature, humidity, wind speed, visibility, and pressure was analyzed to understand its structure and relationships. The data underwent preprocessing steps such as handling missing values, feature transformation, and normalization to ensure high-quality input for the predictive model. Exploratory Data Analysis (EDA) played a central role in identifying trends, seasonal variations, and correlations among features. Visualizations like scatter plots, correlation heatmaps, and time-series graphs helped in gaining insights into the behavior of the data over time.

After gaining a clear understanding of the dataset, different machine learning algorithms were explored to build a predictive model. The performance of these models was compared to select the most accurate and stable prediction technique. The final model was deployed using a Streamlit web application, making it interactive and accessible for real-time forecasting. This project reflects the complete workflow of a data science solution, demonstrating both analytical and practical implementation skills.

**Bottom of Form**

**3. Problem statement**

Weather prediction is a complex and challenging task due to the dynamic and interdependent nature of atmospheric conditions. Factors such as temperature, humidity, wind speed, atmospheric pressure, and visibility change continuously over time, and even minor variations can significantly influence future weather outcomes. Traditional forecasting systems depend heavily on manual interpretation and statistical models, which may not effectively capture hidden or nonlinear patterns present in large-scale historical weather data. This often results in predictions that lack consistency and accuracy, which can negatively impact several sectors such as agriculture, disaster preparedness, aviation, logistics, and daily planning.

With the increasing availability of digital weather records, there is a growing requirement to utilize advanced computational approaches that can analyze large datasets efficiently and provide reliable predictions. Machine learning methods offer the capability to learn meaningful patterns from existing data and use them to forecast future conditions with improved precision. However, building an accurate prediction model requires multiple steps, including data preprocessing, feature engineering, model selection, performance evaluation, and real-time deployment.

Thus, the primary problem addressed in this project is to develop a machine

learning-based model capable of predicting weather conditions using historical meteorological data. The aim is not only to achieve high prediction accuracy but also to provide a system that is practical and accessible. To fulfill this objective, the final trained model is integrated into an interactive Streamlit web application, enabling users to input relevant features and instantly receive predicted weather outputs. This solution ensures both analytical depth and real-world usability.

**4. Scope Of The Project**

The scope of this project covers the complete cycle of data-driven weather prediction, starting from dataset understanding to deploying a working application. The project focuses on analyzing historical weather data, identifying key patterns, building predictive models, and providing real-time predictions through an interactive user interface. The aim is to demonstrate how machine learning can be effectively applied to weather forecasting and integrated into a usable system.

**Scope includes:**

1. **Dataset Utilization:**  
   Using historical weather data containing temperature, humidity, wind speed, pressure, etc.
2. **Data Preprocessing:**  
   Cleaning data, handling missing values, and preparing features for modeling.
3. **Exploratory Data Analysis (EDA):**  
   Visualizing data patterns, trends, and feature correlations.
4. **Model Training:**  
   Applying different machine learning algorithms to build a predictive model.
5. **Model Evaluation:**  
   Comparing model performance and selecting the most accurate one.
6. **Deployment:**  
   Integrating the final model into a Streamlit web application.
7. **User Interaction:**  
   Allowing users to input weather parameters and get prediction results instantly.

**5. Dataset**

The dataset used in this project consists of historical weather observations recorded over a certain period. It includes essential atmospheric variables that influence weather behavior and are commonly used for climate analysis and forecasting. The dataset serves as the basis for building and training the machine learning model, and therefore, understanding its structure and characteristics is an important part of this study.

**5.1 Source of Dataset**

The dataset was obtained from a publicly available weather data repository. It contains real-world weather measurements recorded at regular time intervals. The dataset was downloaded in **CSV (Comma-Separated Values)** format, which allows easy loading and manipulation using Python-based data analysis tools.

**5.2 Dataset Structure**

The dataset is organized in a tabular form, where each row represents a recorded weather instance and each column represents a specific attribute. The major features present in the dataset are:

| **Attribute** | **Description** |
| --- | --- |
| **Temperature** | Air temperature in degrees Celsius |
| **Humidity** | Amount of moisture present in the air (%) |
| **Wind Speed** | Speed of wind in km/h |
| **Visibility** | Distance up to which objects are visible clearly |
| **Pressure** | Atmospheric pressure measured in hPa |
| **Weather Condition** | Descriptive label of the weather (e.g., Clear, Rain, Cloudy) |

These features collectively influence the weather condition and play a key role in prediction.

**5.3 Data Cleaning and Preprocessing**

Before model training, the dataset was examined for quality issues:

* **Missing Values:** Some readings were missing and were either filled using statistical methods (mean/median) or removed if irrelevant.
* **Duplicates:** Repeated records were checked and removed to avoid bias.
* **Outliers:** Extreme and unrealistic values were evaluated and handled carefully.
* **Categorical Encoding:** Text-based weather conditions were converted into numerical form using encoding techniques.
* **Feature Scaling:** Continuous variables like temperature and wind speed were normalized to maintain uniformity across features.

These preprocessing steps improved the consistency and reliability of the dataset.

**5.4 Importance of the Dataset**

The performance of the machine learning model is directly dependent on the quality of the data used for training. A well-prepared dataset enables the model to identify meaningful weather patterns and make accurate future predictions. Therefore, proper dataset understanding, cleaning, and structuring form the foundation of the project’s success.

**6. Technologies Used**

This project utilizes a combination of software tools and hardware resources to perform data analysis, model training, and deployment of the predictive system. The selection of technologies ensures efficient processing, reliable model performance, and a user-friendly interface for real-time weather prediction.

* 1. **Software Requirements**

**1. Python Programming Language**  
 Used for data processing and model development.

1. **Pandas & NumPy**  
   Used for data cleaning, handling missing values, performing numerical calculations, and organizing data in structured formats.
2. **Matplotlib & Seaborn**  
   Utilized for Exploratory Data Analysis (EDA) to visualize patterns, trends, and correlations in the dataset.
3. **Scikit-Learn (sklearn)**  
   Employed for implementing machine learning algorithms, splitting datasets, training models, and evaluating performance.
4. **Streamlit Framework**  
   Used to deploy the final model into a web-based application, allowing interactive user input and real-time prediction output.
5. **Jupyter Notebook / VS Code**  
   Served as the development environment for writing, testing, and debugging the data science code in a structured and visual manner.

**6.2 Hardware Requirements**

The project was executed on a standard personal laptop with a **Dual-Core processor** and **4GB RAM**.

1. **Architectural Components**

The architecture of this project follows a standard **machine learning pipeline** consisting of sequential data processing and model-building stages. Each component has a specific role in transforming raw weather data into meaningful predictions.

**1. Data Source**

* **Input:** Historical weather dataset (CSV).
* Provides raw weather records used for analysis and prediction.

**2. Data Preprocessing Module**

* **Input:** Raw dataset.
* Cleans data, handles missing values, encodes categorical values, and scales numerical features.
* **Output:** Feature set (X) and target variable (y).

**3. Model Training and Validation Module**

* Splits data into training and testing sets.
* Trains machine learning model and selects best-performing one.
* **Output:** Trained prediction model.

**4. Evaluation Module**

* Uses the model to predict on test data.
* Computes metrics like Accuracy, Precision, and Error Rate.
* **Output:** Performance report.

**5. Deployment / User Interface Module**

* The final trained model is integrated into a **Streamlit web app**.
* Users input weather parameters and receive real-time predictions.

In summary, data flows from the dataset → preprocessing → model training → evaluation → deployment

**8.Methodology**

The implementation of Weather Prediction using Machine Learning is carried out through the following steps:

1. **Data Collection:**  
   The weather dataset containing temperature, humidity, wind speed, pressure, visibility, and weather conditions is loaded from a CSV file for analysis.
2. **Data Cleaning:**  
   Missing values are removed and a duplicate-free dataset is prepared to maintain data accuracy. The Date/Time field is converted into useful components such as **Hour** and **Month**.
3. **Feature Engineering:**  
   A new feature **Weather Class** is created by simplifying multiple weather descriptions into three categories — Clear, Cloudy, and Bad Weather. This helps in reducing complexity and improving classification performance.
4. **Label Encoding & Feature Selection:**  
   Categorical labels are encoded numerically using Label Encoder. Selected features used for prediction include:
   * Temperature
   * Humidity
   * Wind Speed
   * Visibility
   * Pressure
   * Hour
   * Month
5. **Handling Class Imbalance:**  
   The **SMOTE (Synthetic Minority Oversampling Technique)** is applied to balance the dataset so that the model learns equally from all weather categories.
6. **Feature Scaling:**  
   Numerical values are normalized using **Standard Scaler** to bring all features to the same scale, improving optimization and model training efficiency.
7. **Model Training – XGBoost Classifier:**  
   The **XGBoost Algorithm** is trained using tuned parameters:
   * n\_estimators = 500
   * learning\_rate = 0.05
   * max\_depth = 8  
     This model is chosen because it is efficient, handles large datasets, reduces overfitting, and provides high accuracy.
8. **Model Evaluation:**  
   The model’s performance is tested on unseen data using:
   * Accuracy Score
   * Confusion Matrix
   * Classification Report (Precision, Recall, F1-Score)  
     The model achieves **90%+ accuracy**, indicating strong predictive performance.
9. **Visualization:**  
   Graphs such as Feature Importance and Confusion Matrix are plotted to understand which features influence predictions and how accurately the classes are identified.

**9. Implementation**

**1.** **Libraries & Dataset:**  
Required libraries are imported, and the weather dataset is loaded using pandas.read\_csv().

**2.** **Data Cleaning:**  
Missing and inconsistent entries are removed for accurate processing.

**3.** **Feature Engineering:**  
The Date/Time field is converted, and **Hour** and **Month** are extracted to capture seasonal patterns.

**4.** **Weather Class Processing:**  
Weather descriptions are simplified into **Clear**, **Cloudy**, and **Bad Weather** and encoded numerically.

**5.** **Feature Preparation:**  
Selected features (Temperature, Humidity, Wind Speed, Visibility, Pressure, Hour, Month) are scaled using **StandardScaler**.

**6.** **Balancing Classes:**  
**SMOTE** is applied to handle class imbalance and improve learning.

**7.** **Train-Test Split:**  
Data is divided into training (80%) and testing (20%) sets.

**8.** **Model Training:**  
The **XGBoost Classifier** is trained using the processed dataset.

**9.** **Model Evaluation & Visualization:**  
Accuracy, Precision, Recall, F1-Score, Confusion Matrix, and Feature Importance graphs are used to assess performance.

**10. Results**

The trained weather prediction model achieved a **final accuracy of 81%**, indicating that the XGBoost classifier performed effectively in classifying weather conditions into *Clear*, *Cloudy*, and *Bad Weather* categories.

The detailed performance based on **precision, recall, and F1-score** is shown below:

| **Weather Class** | **Precision** | **Recall** | **F1-Score** | **Support** |
| --- | --- | --- | --- | --- |
| **Clear (0)** | 0.95 | 0.90 | 0.93 | 755 |
| **Cloudy (1)** | 0.76 | 0.81 | 0.79 | 765 |
| **Bad Weather (2)** | 0.72 | 0.72 | 0.72 | 759 |

* **Overall Model Accuracy:** **81%**
* **Macro Average F1-Score:** 0.81
* **Weighted Average F1-Score:** 0.81

These results show that the model performs strongly for **Clear** weather prediction and maintains a balanced performance across all categories. The classification scores indicate that the model generalizes well and is reliable for practical weather condition prediction tasks.

**11. Testing Report**

| **Test Case No.** | **Test Description** | **Input / Action** | **Expected Output** | **Actual Output** | **Result** |
| --- | --- | --- | --- | --- | --- |
| 1 | Load Dataset | Run script to read CSV file | Dataset loads successfully | Loaded without errors | Pass |
| 2 | Check for Missing Values | Execute data.isnull().sum() | No missing values or handled properly | Missing values removed successfully | Pass |
| 3 | Feature Engineering | Convert Date/Time to Hour & Month | New columns created correctly | Hour and Month columns generated | Pass |
| 4 | Weather Class Simplification | Convert detailed weather text to 3 categories | Labels are simplified correctly | Values converted to Clear / Cloudy / Bad Weather | Pass |
| 5 | Label Encoding | Apply LabelEncoder to target column | Weather class converted to numeric labels | Encoded values generated | Pass |
| 6 | Data Balancing (SMOTE) | Apply SMOTE oversampling | Balanced number of samples across classes | Dataset balanced successfully | Pass |
| 7 | Feature Scaling | Apply StandardScaler | All features normalized | Values scaled correctly | Pass |
| 8 | Train-Test Split | Split data 80:20 | Data splits without error | Data successfully split | Pass |
| 9 | Model Training | Train XGBoost classifier | Model should train successfully | Model trained successfully | Pass |
| 10 | Model Prediction | Predict using test set | Predictions generated | Predictions produced without error | Pass |
| 11 | Accuracy Evaluation | Compare y\_test and y\_pred | Accuracy around expected performance | **Accuracy = 81%** | Pass |
| 12 | Confusion Matrix Display | Visualize confusion matrix | Proper heatmap output | Matrix displayed correctly | Pass |
| 13 | Feature Importance Display | Plot feature importance bar graph | Graph shows top contributing features | Graph displayed correctly | Pass |

**12. Future Scope**

Although the Weather Prediction System performs efficiently and achieves a good level of accuracy, there are several opportunities to enhance and expand the project in the future:

1. **Integration of More Weather Parameters:**  
   Additional environmental factors such as rainfall intensity, cloud density, and solar radiation can be included to improve prediction accuracy and model sensitivity.
2. **Use of Real-Time Weather Data:**  
   The system can be linked with live weather APIs or IoT-based weather stations to provide real-time forecasting instead of relying only on historical data.
3. **Advanced Machine Learning / Deep Learning Models:**  
   Future versions may incorporate models like Random Forest, LSTM (Long Short-Term Memory Networks), or CNN-based atmospheric analysis to capture complex weather patterns and time dependencies.
4. **Deployment as a Web or Mobile Application:**  
   The model can be embedded into a web dashboard or mobile app, allowing users to input current conditions and receive instant weather predictions.
5. **Location-Based Weather Prediction:**  
   By adding GPS or city-based features, the system can provide localized weather forecasts tailored to specific geographic regions.
6. **Continuous Model Training:**  
   The system can be improved by continuously updating the model with new data to ensure that predictions remain accurate even as climate patterns change over time.

**13. Conclusion**

The Weather Prediction System developed in this project successfully demonstrates how Machine Learning techniques can be used to analyze weather data and predict future conditions. By training the model on historical weather datasets, the system is able to identify patterns and generate predictions with a reasonable level of accuracy. This helps in making dependable forecasts that can be useful for planning daily activities, agriculture, travel, and safety measures.

The implementation of the model shows that data preprocessing, feature selection, and model evaluation play an important role in achieving accurate results. The results and classification performance indicate that the system is efficient, but there is still scope for improvement by including more advanced algorithms and real-time weather data sources.

Overall, the project provides a practical and scalable solution for weather forecasting. It highlights the importance of Machine Learning in solving real-world problems and encourages further research and development to enhance prediction accuracy. With future enhancements, this system can become a reliable tool for various industries and users who depend on accurate weather information in their daily lives.

**14. References**

**Research Papers:**

* [**https://www.kaggle.com/datasets**](https://www.kaggle.com/datasets)**(Weather Datasets)**
* [**https://www.weather.gov/forecastmaps**](https://www.weather.gov/forecastmaps)**(Official Weather Forecast Maps)**
* [**https://scikit-learn.org/stable/**](https://scikit-learn.org/stable/)**(Scikit-Learn Machine Learning Library)**
* [**https://pandas.pydata.org/**](https://pandas.pydata.org/)**(Pandas Documentation for Data Handling)**
* [**https://matplotlib.org/stable/**](https://matplotlib.org/stable/)**(Matplotlib Documentation for Visualization)**
* [**https://www.metoffice.gov.uk/research/climate**](https://www.metoffice.gov.uk/research/climate)**(Climate & Weather Research Information)**
* [**https://www.noaa.gov/education/resource-collections/weather-atmosphere**](https://www.noaa.gov/education/resource-collections/weather-atmosphere)**NOAA Weather and Atmospheric Data)**