IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Leveraging Weather Data For Accurate Visibility Prediction

Vadamodalu.Yuva Teja [1], Damera.Anjan Sai Sri Vatsav [2], Palakollu Eswar Sai Karthik [3], Konathala Nithin [4], Mrs. M. Kalyani [5], Mr. A. Venkateswara Rao[6]

[1,2,3,4] **B. Tech Students**, Department of CSE(Artificial Intelligence and Machine Learning), Dadi Institute of Engineering and Technology, NH-16,Anakapalle,
Visakhapatnam-531002, A.P

[5] **Assistant Professor**, Department of CSE (AI & ML), Dadi Institute of Engineering and Technology, NH-16, Anakapalle, Visakhapatnam-531002, A.P

Abstract— Predicting visibility distance using climatic indicators is essential for enhancing safety and operational efficiency across industries, such as aviation and transportation. This project presents a real-time weather data collection and prediction system powered by advanced machine learning techniques. The system processes key meteorological parameters, including temperature, humidity, wind speed, and atmospheric pressure, to accurately forecast visibility distances. To achieve this, the Gradient Boost Regressor, Cat Boost Regressor, and ensemble methods are utilized to model complex nonlinear relationships in weather data. Clustering techniques, such as K-means supported by silhouette analysis, further enhance the data segmentation for effective model training. The system also features a user-friendly web interface that allows users to input geographic locations, generate weather data along specific routes, and visualize results on interactive maps. The backend, implemented in Flask, integrates APIs, such as Weather API and Open Cage, to automate real-time data collection and geocoding. Predictions are seamlessly generated and presented in a downloadable format, supporting critical decision-making processes. The system's ability to deliver reliable visibility forecasts helps to optimize operational efficiency and mitigate risks in safety-sensitive industries. By combining machine learning with real-time weather data, this study provides a robust tool for visibility prediction and decision support.

Keywords— Visibility Prediction, Machine Learning, Gradient Boost Regressor, Cat Boost Regressor, K-Means Clustering, Silhouette Score, Weather Data, Real-time Prediction, Ensemble Techniques, Flask, Transportation, Aviation.

INTRODUCTION

Accurate visibility distance prediction based on climatic factors is essential for ensuring safety and optimizing operations in industries such as transportation, aviation, and environmental monitoring. Visibility plays a pivotal role in minimizing risks and improving efficiency in these sectors. Reliable visibility forecasts can be achieved by analyzing the relationships between climatic indicators, such as temperature, humidity, wind speed, precipitation, and atmospheric pressure [3]. This study introduces a regression-based system to predict visibility distances using a comprehensive dataset of weather indicators. The system combines machine learning models such as Gradient Boosting [1] and Cat Boost [2] with clustering techniques, such as K-means [6], to improve prediction accuracy. Users can input starting and ending locations to generate random geographic points along the route, where weather data are collected using the Weather API [10] and Open Cage Geocoder [11]. This enables real-time prediction of specific routes and locations. The findings of this study are valuable to stakeholders, including meteorologists, transportation operators, and environmental agencies. Reliable visibility predictions support improved weather forecasts [12], safer travel operations [5], and better air quality assessment [4]. This study contributes to the existing knowledge base and provides a practical tool for decision-making and operational planning in diverse climatic scenarios.

Dataset: The dataset used in this study was sourced from NOAA's JFK Weather Dataset, which is available on Kaggle. It includes 114,546 hourly observations of various climatic variables recorded at John F. Kennedy International Airport (JFK) in New York. The key features include visibility, temperature, humidity, wind speed and direction, dew point, and atmospheric pressure.
The dataset spans multiple seasons and offers diverse weather conditions for analysis. Its quality and comprehensiveness make it suitable for developing accurate visibility prediction models for industries, such as aviation and transportation (Weather API, n.d.) [5] , Open Cage Geocoder, n.d.) [6].

Data Description:
VISIBILITY (Target variable): distance (miles) at which an object is visible
Dry Bulb Temp F: Standard air temperature (°F).
Wet Bulb Temp F: Temperature if air cooled to saturation.
Dew Point Temp F: Temperature where air becomes moisture-saturated (°F).
Relative Humidity: Moisture in the air as percentage
Wind Speed: Speed of wind (mph).
Wind Direction: Wind's origin direction (°).
Station Pressure: Atmospheric pressure at the station (inHg).
Sea Level Pressure: Pressure adjusted to sea level (inHg).
Precipitation: Hourly precipitation (inches). [3]

MOTIVATION/ LITERATURE SURVEY

Motivation: Predicting the visibility distance is critical for enhancing safety and operational efficiency in industries such as aviation, transportation, and environmental monitoring. Weather phenomena, such as fog and precipitation, can severely reduce visibility, increase risks, and cause operational delays. Despite advancements in weather forecasting, visibility prediction remains a challenge because of the nonlinear interactions between climatic variables. This research focuses on developing a machine learning-based system to provide real-time visibility predictions, addressing key needs such as Safety. Precise predictions to mitigate risks for operators in transportation and aviation [3]. Efficiency Improved decision making for routes and schedules [4]. Real-time Insights Dynamic data retrieval using APIs ensures accurate location-specific forecasts (Weather API) [10]. Accessibility User-friendly interfaces make predictions usable across technical domains [9].

Literature Survey: Healthcare insurance fraud detection has shifted from traditional rule-based systems, which struggle with modern complexities, to Machine Learning (ML) approaches. Models like Logistic Regression, XG Boost, and Cat Boost excel in analyzing large, imbalanced datasets, identifying patterns, and improving accuracy. While deep learning shows promise for anomaly detection, challenges like interpretability and resource demands remain. ML systems address limitations of traditional methods, offering enhanced efficiency and accuracy in combating fraud, though issues like class imbalance and privacy require ongoing research. This advancement positions ML as a transformative tool in restoring trust and reducing losses in the insurance sector.

ALGORITHMS AND IMPLEMENTATION

K-means Clustering for Data Exploration and Feature Engineering:

In this study, K-means clustering was employed as a supportive technique for data exploration, feature engineering, preprocessing, and visualization. While K-means clustering does not directly contribute to predicting the target variable, visibility distance, it plays a vital role in identifying inherent patterns within the dataset, grouping similar data points, and handling outliers [6]. Additionally, the clusters generated through K-means are utilized as new features for the subsequent regression model, improving its predictive capability. The main prediction task in this work relies on a regression model that uses climatic indicators such as temperature, humidity, wind speed, and atmospheric pressure to forecast visibility distance [4]. The clusters produced by K-means serve as supplementary features in the regression model, enhancing the model's performance. Although K-means clustering offers significant benefits, one of the challenges is determining the optimal number of clusters (K). Selecting an appropriate K is crucial for the success of the algorithm, as it directly influences the clustering results. To address this, various methods exist for determining the optimal K value, and in this study, we focus on two main approaches: the Elbow Method and the Silhouette Score. [7]

Elbow Method for Optimal Cluster Selection:

The Elbow Method was a widely used technique to determine the optimal number of clusters for K-means clustering. The procedure is as follows:

1.K-means clustering is performed on the dataset for a range of K values, typically from 1 to 10.

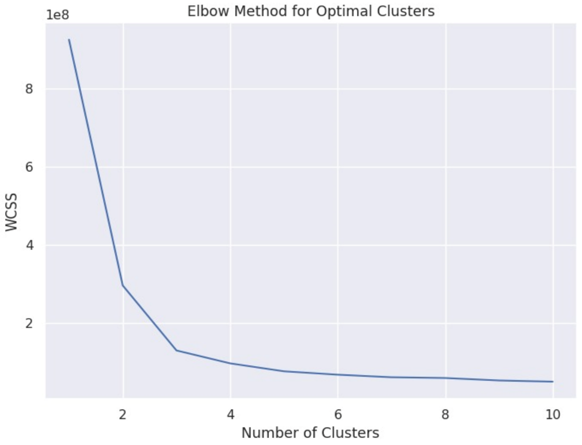
2.The Within-Cluster Sum of Squares (WCSS) is calculated for each K value. WCSS is a measure of the total squared distance between each data point and the centroid of its assigned cluster.

$$WCSS = \sum_{k=1}^{K} \sum_{i=1}^{n_k} (x_i - \mu_k)^2$$

where Xi represents a data point in cluster k, and µk is the centroid of cluster k

3.The WCSS values were plotted against the corresponding K values to generate the curve.

4. The elbow point is identified in the plot, which indicates the optimal number of clusters. This point is characterized by a sharp decrease in the WCSS, followed by a gradual flattening of the curve. The optimal K corresponds to the point where the reduction in WCSS begins to diminish significantly.[8]



Silhouette Score for Cluster Validation:

In addition to the Elbow Method. The Silhouette Score is another valuable metric used to evaluate the cluster quality. The Silhouette Score measures how similar each data point is to its own cluster Compared with other clusters. [7]

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

Where:
a(i)= The average distance of i to all other points within the same cluster (intra-cluster distance).
b(i)= The average distance of i to all points in the nearest neighboring cluster (inter-cluster distance).
1.The score ranges from -1 to +1, where A score close to +1 indicates that the data points are well clustered [7].
2.A score close to zero suggests that the data point lies between the two clusters [8].
3.A score close to -1 indicates that the data point may have been assigned to the wrong cluster. The Silhouette Score for a single data point was calculated [8].

Gradient Boosting Algorithm:
Gradient Boosting is an ensemble learning algorithm utilized for predicting visibility distance based on climatic indicators [1]. It combines the predictions of multiple weak learners, typically decision trees, to improve the accuracy. The process involves data preparation, training the Gradient Boosting model, tuning hyperparameters for optimal performance, and evaluating the model using metrics such as the Mean Squared Error (MSE) or R2squared. Gradient Boosting is well-regarded for its ability to model complex relationships and handle numerical and categorical features effectively [2].

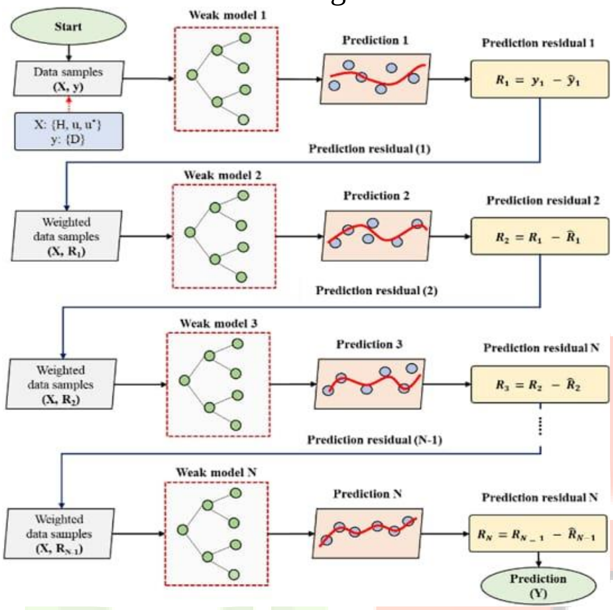


Fig:2- Flow chart of gradient boosting regression model
Step 1: Initialize the Model Initialize predictions with a constant value, usually the mean of the target variable [4].
Step 2: Compute residual residuals computed as the difference between the actual target values and predictions from the previous iteration [5].
Step 3: Train a Weak Learner Fit a weak learner (decision tree) on the residuals to learn the error patterns [13].
Step 4: Update Predictions Update the predictions by combining the previous predictions with the weighted outputs of the new weak learner [14].
Step 5: Regularization:
Regularization parameters such as learning_rate, max_depth and subsample control overfitting [1].
1.Learning Rate: Adjusts the impact of each tree [7].
2.Max Depth: Limits the depth of individual trees [15].
3.Subsample: Controls the fraction of data used for training each tree [12].
Step 6: Loss Function Optimization
1.The loss function (typically MSE for regression) is minimized iteratively using gradient descent [3].
2. Hyperparameter Tuning Process
Learning Rate: Tested values: [0.01,0.1,0.2] [4].
Maximum Depth: Tested values: [3,5,10] [8].
Number of Estimators: Tested values: [50,100,200] [7].
Sub sample: Tested values: [0.8,1.0] [2].

IMPLEMENTATION

1.EDA:

Inspecting Null and Missing Values The dataset was inspected for null and missing values, including Na Ns, empty strings, or placeholder values. No such issues were identified, ensuring the integrity of the data for further analysis [1].

Standard Scaling: Standard scaling was applied to normalize the dataset to a mean of zero and a standard deviation of 1. Although this reduced scale-related disparities, some features deviated from normal distribution [3].

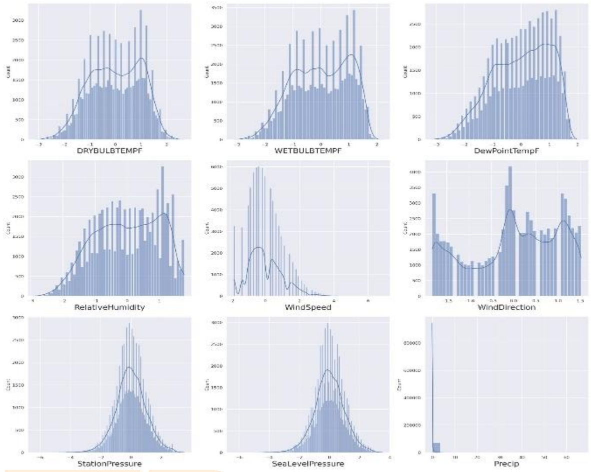
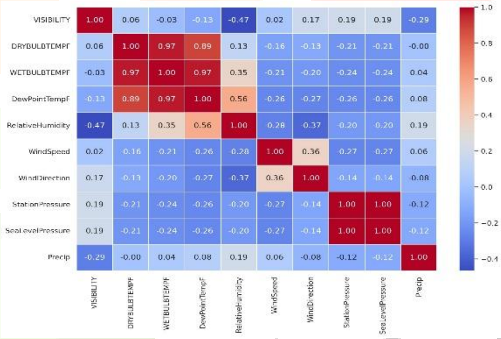


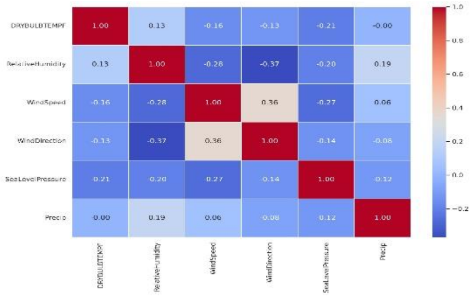
Fig -3: Still some columns are not following normal distribution even after applying standard scaling techniques.

Correlation Analysis: Correlation analysis was conducted to identify the potential redundancy among the features.



	VISIBILITY	DRYBULBTEMPF	WETBULBTEMPF	DewPointTempF	RelativeHumidity	WindSpeed	WindDirection	StationPressure	SeaLevelPressure	Precip
VISIBILITY	1.00	0.06	-0.03	-0.13	-0.47	0.02	0.17	0.19	0.19	-0.29
DRYBULBTEMPF	0.06	1.00	0.97	0.89	0.13	-0.16	-0.13	-0.21	-0.21	-0.00
WETBULBTEMPF	-0.03	0.97	1.00	0.97	0.35	-0.21	-0.20	-0.24	-0.24	0.04
DewPointTempF	-0.13	0.89	0.97	1.00	0.56	-0.26	-0.27	-0.26	-0.26	0.08
RelativeHumidity	-0.47	0.13	0.35	0.56	1.00	-0.28	-0.37	-0.20	-0.20	0.19
WindSpeed	0.02	-0.16	-0.21	-0.26	-0.28	1.00	0.36	-0.27	-0.27	0.06
WindDirection	0.17	-0.13	-0.20	-0.27	-0.37	0.36	1.00	-0.14	-0.14	-0.08
StationPressure	0.19	-0.21	-0.24	-0.26	-0.20	-0.27	-0.14	1.00	1.00	-0.12
SeaLevelPressure	0.19	-0.21	-0.24	-0.26	-0.20	-0.27	-0.14	1.00	1.00	-0.12
Precip	-0.29	-0.00	0.04	0.08	0.19	0.06	-0.08	-0.12	-0.12	1.00

Fig -4: Correlation between the columns (Heat Map)
1.WETBULBTEMPF and DRYBULBTEMPF and Dew Point Temp F
2.Station Pressure and Sea Level Pressure [7].
To avoid multicollinearity, highly correlated features were removed [13].



	DRYBULBTEMPF	RelativeHumidity	WindSpeed	WindDirection	SeaLevelPressure	Precip
DRYBULBTEMPF	1.00	0.13	-0.16	-0.13	-0.21	-0.00
RelativeHumidity	0.13	1.00	-0.28	-0.37	-0.20	0.19
WindSpeed	-0.16	-0.28	1.00	0.36	-0.27	0.06
WindDirection	-0.13	-0.37	0.36	1.00	-0.14	-0.08
SeaLevelPressure	-0.21	-0.20	-0.27	-0.14	1.00	-0.12
Precip	-0.00	0.19	0.06	-0.08	-0.12	1.00

Fig -5: Dropping the columns with high correlation (Heat Map)

ARCHITECTURE

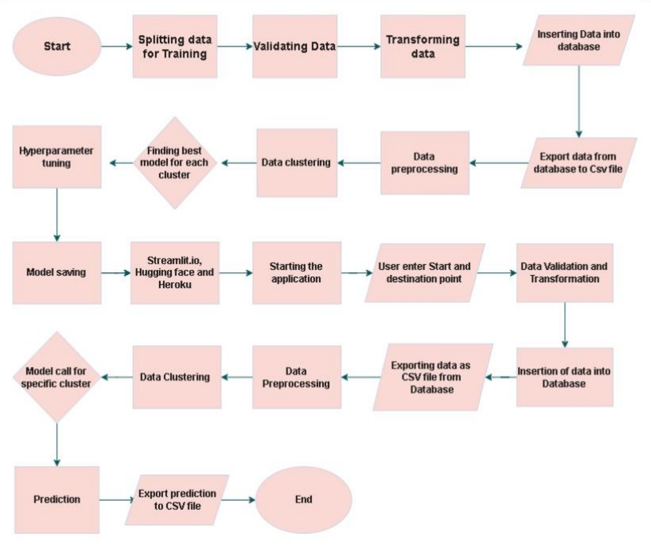

Start
Splitting data for Training
Validating Data
Transforming data
Inserting Data into database
Export data from database to Csv file
Data preprocessing
Data clustering
Finding best model for each cluster
Hyperparameter tuning
Model saving
Streamlit.io, Hugging face and Heroku
Starting the application
User enter Start and destination point
Data Validation and Transformation
Insertion of data into Database
Exporting data as CSV file from Database
Data Preprocessing
Data Clustering
Model call for specific cluster
Prediction
Export prediction to CSV file
End

Fig – 6: Architecture of the project (Work Flow)

WORK FLOW:

This section outlines the end-to-end workflow for developing a system to predict visibility distance based on climatic indicators. The workflow involves data collection, preprocessing, exploratory analysis, model training, evaluation, and deployment using Gradient Boosting Regressor [2].

1.Dataset Collection and Understanding Source:

The dataset was collected from trusted repositories such as the National Oceanic and Atmospheric Administration (NOAA) and contains historical weather data [5].

Variables:

Independent variables: Dry bulb temperature, wet bulb temperature, dew point temperature, relative humidity, wind speed, wind direction, station pressure, sea level pressure, and precipitation.

Target variable: visibility distance.

The dataset comprised hourly records over an extended period, ensuring variability across different climatic conditions [3].

2.Data Cleaning and Preprocessing Handling of Missing Data: Missing values were imputed using median imputation to maintain consistency [1].

Outlier Detection and Removal: Statistical techniques and visualization tools (e.g., box plots) identify and remove outliers [7].

Data Transformation:

Temperature and pressure were standardized, and all features were normalized using Standard Scaler to achieve a mean of 0 and a standard deviation of 1 [13].

3.Exploratory Data Analysis (EDA) Visualization: Distribution plots and histograms were used to study feature distributions, whereas box plots helped identify anomalies [14]. Correlation Analysis Correlation heatmaps revealed high correlations between Dry bulb temperature, Wet bulb temperature and Dew point temperature Station and sea-level pressures [7].

Feature Elimination: Based on the correlation results, highly correlated features (e.g., wet-bulb temperature, dew point temperature, and station pressure) were removed to reduce redundancy and multicollinearity [8].

4.Model Development: The Gradient Boosting Regressor was chosen for its robustness and ability to handle non-linear relationships. Model training [2]. The dataset was split into training (80%) and testing (20%) sets. Gradient boosting Regressor parameters were optimized using a Grid Search to enhance accuracy [4].

Hyperparameter Tuning: The learning rate, maximum depth, and number of estimators were fine-tuned to avoid overfitting and underfitting [12].

5.Model Evaluation: The trained model was evaluated on the testing set using the following regression metrics.

Mean Absolute Error (MAE): Quantified average prediction error.

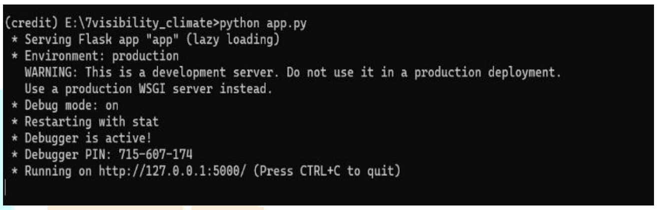
Mean Squared Error (MSE): Penalized larger prediction errors. **R-squared (R^2)**: Assessed the proportion of variance explained by the model. The Gradient Boosting Regressor demonstrated a high accuracy and generalization capability [15].

6.Deployment API Development:

A Flask API was created to deploy the model, enabling real-time prediction [9]. Users input weather parameters such as temperature, humidity, and wind speed to receive visibility predictions [1]. Visualization and User Interaction Predictions for multiple points along a route (starting and ending locations) are displayed using an interactive folium map [16]. A CSV file containing the predictions was generated for download [17].

RESULTS AND DISCUSSIONS

Machine Learning models utilize historical weather data, including temperature, humidity, wind speed, and air quality indicators to forecast visibility conditions. By training these models on past visibility measurements along with related weather features, they identified patterns and relationships that enable accurate predictions of future visibility. These predictions help assess how climatic factors might influence visibility and support more effective planning and decision-making. A Flask-based web application serves as an interactive platform, allowing users to input relevant data and predict visibility distances seamlessly.



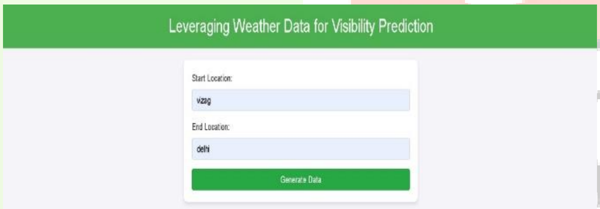
```
(credit) E:\7visibility_climate>python app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 715-607-174
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Fig – 7: Running the application (App.py)

Leveraging Weather Data for Accurate Visibility Prediction was designed to collect real-time weather data, perform predictions, and generate relevant insights based on geographical locations. The web application facilitates two main functions: generating weather data and predicting visibility based on the weather parameters. Below is the breakdown of the system components and workflow.

1.**Weather Data Generation:**

a. User Input: The user provides two geographical locations start and end location via a web form



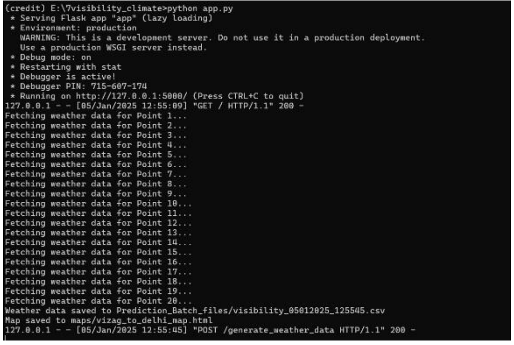
Leveraging Weather Data for Visibility Prediction

Start Location: vizag

End Location: delhi

Geocoding: Using the Open Cage API, the system retrieves the latitude and longitude of the given locations

Weather Data Retrieval: The system uses the Weather API to gather current weather data at various points along the line between the start and end locations. The weather data includes all independent variables. This data is collected for 20 points between the start and end locations

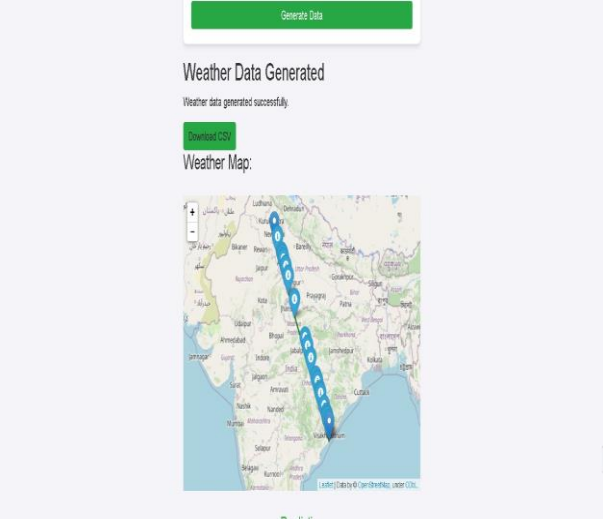


```
(credit) E:\7visibility_climate>python app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 715-607-174
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [05/Jan/2025 12:55:09] "GET / HTTP/1.1" 200 -
Fetching weather data for Point 1...
Fetching weather data for Point 2...
Fetching weather data for Point 3...
Fetching weather data for Point 4...
Fetching weather data for Point 5...
Fetching weather data for Point 6...
Fetching weather data for Point 7...
Fetching weather data for Point 8...
Fetching weather data for Point 9...
Fetching weather data for Point 10...
Fetching weather data for Point 11...
Fetching weather data for Point 12...
Fetching weather data for Point 13...
Fetching weather data for Point 14...
Fetching weather data for Point 15...
Fetching weather data for Point 16...
Fetching weather data for Point 17...
Fetching weather data for Point 18...
Fetching weather data for Point 19...
Fetching weather data for Point 20...
Weather data saved to Prediction_Batch_files/visibility_05012025_125545.csv
Map saved to maps/vizag_to_delhi_map.html
127.0.0.1 - - [05/Jan/2025 12:55:45] "POST /generate_weather_data HTTP/1.1" 200 -
```

Fig – 9: Randomly generated 20 points between starting and destination location.

CSV Export: The weather data is saved in a CSV format and made available for download through a link on the web interface.

Map Generation: A map is generated using the folium library visualize the route between the two locations, with markers for each point along the path.



Weather Data Generated

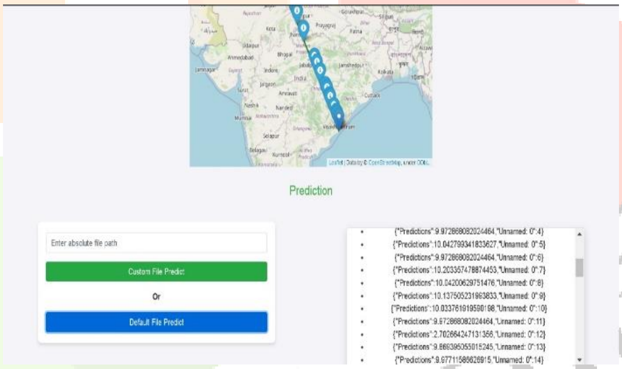
Weather data generated successfully.

Weather Map:

Fig-10: Folium map is generated.

Prediction Model:

- **User Input:** Users upload a CSV file or choose a default dataset for prediction.
- **Prediction Workflow:** Once the file is uploaded, the system validates the data and feeds it into the trained machine learning model to make predictions.
- The model predicts visibility based on the weather data and generates a result file.
- **Prediction Output:** The results are returned as a downloadable file, and a preview of the prediction results is displayed in the interface.



```
• {"Predictions":9.972868082024464,"Unnamed: 0":4}
• {"Predictions":10.042793341833627,"Unnamed: 0":5}
• {"Predictions":9.972868082024464,"Unnamed: 0":6}
• {"Predictions":10.203357478874453,"Unnamed: 0":7}
• {"Predictions":10.04200629751476,"Unnamed: 0":8}
• {"Predictions":10.137505231963833,"Unnamed: 0":9}
• {"Predictions":10.033761919590198,"Unnamed: 0":10}
• {"Predictions":9.972868082024464,"Unnamed: 0":11}
• {"Predictions":2.702664247131356,"Unnamed: 0":12}
• {"Predictions":9.869395055015245,"Unnamed: 0":13}
• {"Predictions":9.977115886626915,"Unnamed: 0":14}
```

Fig –11: Displaying the respective predicted value to user.

FUTURE SCOPE (Future Scope with Fine-Tuned LLMs):

Enhanced Feature Engineering: Use fine-tuned LLMs to derive advanced features and identify subtle relationships between weather variables, thereby improving the prediction accuracy.

Dynamic data preprocessing: LLMs are employed for anomaly detection, context-aware data imputation, and addressing missing or inconsistent weather data.

Natural Language Interfaces: LLMs are integrated to enable users to interact with the system through conversational queries and receive detailed explanations of predictions.

Multi-modal data integration: LLMs combine weather data with external data sources, such as satellite imagery or regional traffic patterns, for comprehensive predictions.

Explainability and scenario modeling: LLMs are utilized to generate textual explanations for predictions and simulate future weather trends under various scenarios, aiding research and decision-making.

CONCLUSION

The visibility prediction system is a cutting-edge solution designed to address the challenges posed by weather-related visibility issues in the aviation industry. This system leverages real-time weather data from APIs and employs advanced machine learning models such as Gradient Boosting Regressor and Cat Boost Regressor to deliver accurate visibility forecasts. By integrating climatic indicators like temperature, humidity, wind speed, and atmospheric pressure, the system ensures precise predictions tailored to aviation requirements. The interactive web-based interface, developed using Flask, provides a seamless user experience. Users can input location details, retrieve real-time weather data, and visualize predictions through intuitive dashboards and interactive maps. This allows aviation stakeholders to assess flight feasibility and make informed decisions quickly and effectively. The solution addresses the critical safety and operational efficiency needs of the aviation sector by offering actionable insights into visibility conditions. It enables aviation operators to optimize flight scheduling, minimize risks, and improve overall safety in adverse weather scenarios. The system's automation, accuracy, and ease of use make it an indispensable tool for aviation operations where visibility plays a crucial role. It enhances decision-making processes and supports resource optimization, ensuring safer and more efficient air travel. By combining real-time data collection, machine learning, and visualization, the visibility prediction system sets a new standard for weather-related decision support in the aviation industry. It highlights the transformative potential of AI-driven solutions in creating a safer and more resilient aviation ecosystem.

REFERENCES

1. Chen, T., & Guestrin, C. (2016). XG Boost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794). DOI: 10.1145/2939672.2939785
2. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). Cat Boost: Unbiased boosting with categorical features. *Advances in Neural Information Processing Systems, 31*.
3. Zhang, Y., & Yang, J. (2018). Visibility prediction based on climatic parameters using machine learning methods. *IEEE Access*, *6*, 20924-20932.
   DOI: 10.1109/ACCESS.2018.2828862
4. Fu, S., & Zhang, G. (2019). Estimation of visibility distance using support vector regression with feature selection. *Theoretical and Applied Climatology*, *136*(3-4), 1505-1517.
   DOI: 10.1007/s00704-018-2532-8

5. Wu, Y., Zhao, Z., & Liu, J. (2017). Visibility distance prediction using random forest regression based on meteorological data. *Journal of Meteorological Research*, *31*(5), 837-850.
   DOI: 10.1007/s13351-017-7043-6
6. Rokach, L., & Maimon, O. (2005). Clustering methods. In *Data Mining and Knowledge Discovery Handbook* (pp. 321-352). Springer.
   DOI: 10.1007/978-0-387-09823-4_15
7. Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53-65.
   DOI: 10.1016/0377-0427(87)90125-7
8. Kaufman, L., & Rousseeuw, P. J. (2009). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley.
   ISBN: 978-0470317488
9. Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media.
   ISBN: 978-1491991732
10. Weather API. (n.d.). Real-time weather and location API for weather data collection.
    Retrieved from https://www.weatherapi.com
11. Open Cage Geocoder. (n.d.). Open Cage API for geocoding and reverse geocoding.
    Retrieved from https://opencagedata.com
12. Singh, R., Kumar, S., & Singh, R. K. (2020). Prediction of visibility using machine learning techniques. *Soft Computing for Problem Solving, 1061-1069*.
    DOI: 10.1007/978-981-15-0035-0_86
13. Karuppiah, R., & Gomathi, R. (2020). Regression-based distance estimation for visibility. In *Proceedings of the11th International Conference on Computing, Communication, and Networking Technologies (ICCCNT).*
    DOI: 10.1109/ICCCNT49239.2020.9225551
14. Sharma, P., Kumar, A., & Garg, K. (2020). Comparative analysis of different regression models for visibility prediction using meteorological parameters. *International Journal of Intelligent Systems*

and Applications, *12*(3), 87-94.
DOI: 10.5815/ijisa.2020.03.09

15. Zhao, Z., & Wang, Q. (2019). Predicting visibility distance using a combined model of wavelet decomposition and long short-term memory neural network. *IEEE Access*, *7*, 25261-25269.
DOI: 10.1109/ACCESS.2019.2899301
16. Lee, S., & Kang, J. (2020). Prediction of atmospheric visibility using ensemble learning methods with meteorological data. Atmospheric Environment, 223, 117273.
DOI: 10.1016/j.atmosenv.2020.117273

17. Li, X., Sun, Z., & Zhang, H. (2021). Visibility forecasting using machine learning techniques: A case study in urban areas. Journal of Atmospheric and Solar-Terrestrial Physics, 213, 105471.DOI: 10.1016/j.jastp.2021.105471