Earth Observation using Machine Learning to Analyze Satellite Data to Monitor and Predict Weather Patterns, Natural Disasters and Climate Change

Dr. Rangaraj B S
Research Professor
Dept. of CSE(AI&ML)
School of Engineering
Dayanada Sagar University

Ratan Ravichandran
B.Tech CSE(AI&ML)
Dept. of CSE(AI&ML)
School of Engineering
Dayanada Sagar University

Sri Bharath Sharma P
B.Tech CSE(AI&ML)
Dept. of CSE(AI&ML)
School of Engineering
Dayanada Sagar University

Sayli Pankaj Bande
B.Tech CSE(AI&ML)
Dept. of CSE(AI&ML)
School of Engineering
Dayanada Sagar University

Shruti Nigam
B.Tech CSE(AI&ML)
Dept. of CSE(AI&ML)
School of Engineering
Dayanada Sagar University

***Abstract* - This research paper investigates the application of machine learning techniques, specifically linear regression, decision trees, and random forests, for weather prediction utilizing satellite data. The study aims to contribute to improved weather forecasting and informed decision-making by analyzing patterns, trends, and predictions related to weather and climate. The methodology involves preprocessing the satellite data and employing the aforementioned algorithms to train and test the models. The results showcase the potential of machine learning in enhancing the accuracy and reliability of weather predictions, thereby supporting efforts to mitigate the impact of natural disasters and climate change.**

I. INTRODUCTION

Machine learning algorithms applied to Earth observation satellite data enable weather pattern, natural disaster, and climate change monitoring and prediction. These algorithms detect patterns and issue early warnings for climate-related events, revolutionizing our understanding of Earth's climate and supporting decision-making. Analysis of historical weather factors like year, month, day, temperature, specific humidity, and relative humidity inform accurate weather predictions, capturing long-term climate variations, seasonal changes, recurring patterns, and their influence on atmospheric conditions and precipitation formation.

By leveraging machine learning techniques and analyzing satellite data, this project aims to enhance our understanding of weather phenomena, leading to improved monitoring and prediction of weather patterns, natural disasters, and climate change. The primary objective is to develop and implement a comprehensive machine learning framework specifically designed for Earth observation data analysis.

II. LITERATURE SURVEY

Random forests excel in accuracy and computational efficiency compared to SVMs and neural networks [1]. They capture complex relationships, handle non-linear patterns, and reduce overfitting by aggregating predictions from multiple decision trees. Parallel processing allows for faster training on large datasets, while their robustness to missing values and outliers reduces data preprocessing requirements.

Satellite data assimilation improves weather prediction accuracy by integrating observations into numerical models [2]. Satellite data supplements numerical models by providing crucial information about the Earth's atmosphere, oceans, and land surface. This integration enhances weather prediction accuracy, enabling more informed decision-making.

Combining Earth observation (EO) with machine learning (ML) offers significant advantages [3]. EO provides extensive global data, while ML excels in analyzing big data and identifying complex patterns. This integration enables prediction of environmental phenomena, land cover classification, optimized data collection, and enhanced decision-making in agriculture, disaster management, and urban planning.

III. METHODOLOGY

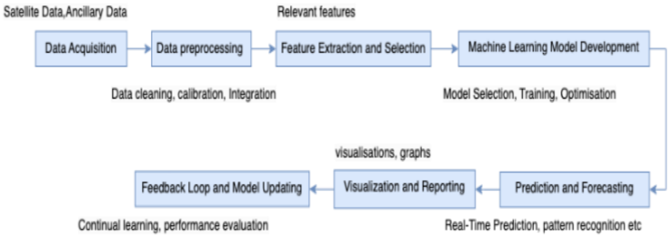

Satellite Data,Ancillary Data
Data Acquisition
Data preprocessing
Data cleaning, calibration, Integration
Relevant features
Feature Extraction and Selection
Machine Learning Model Development
Model Selection, Training, Optimisation
Prediction and Forecasting
Real-Time Prediction, pattern recognition etc
visualisations, graphs
Visualization and Reporting
Feedback Loop and Model Updating
Continual learning, performance evaluation

Figure 3.1 General Architecture

3.1. Dataset Definition

The dataset used comprises data collected over the last 10 years, starting from January 1 2019. It contains values of the minimum and maximum temperatures, UV indices, Sunrise and sunset, moonrise and moonset, heat index, dew point, etc. The features are selected by plotting a correlation heatmap.

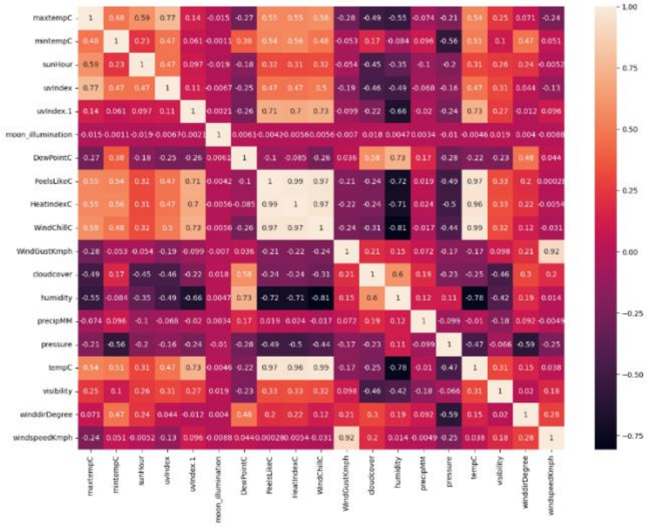


Figure 3.1.1 Correlation Heatmap

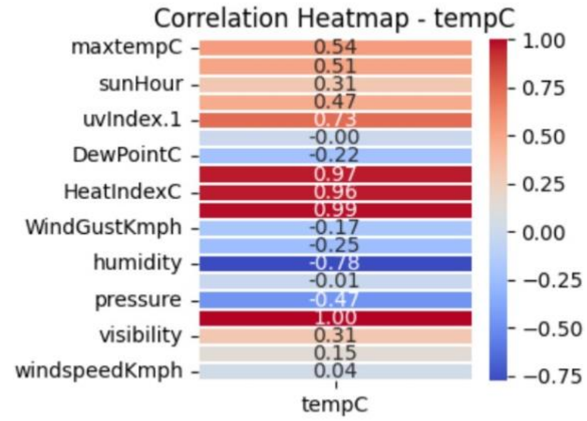


Figure 3.1.2 Correlation Heatmap for selected features

The model utilizes the features and selects ones with the most correlation with the target variable, which are MaxTempC, HeatIndexC, uvIndex1, FeelsLikeC and WindchillC.

3.2. Software/Libraries -

Modules used:

- NumPy: For data manipulation and numerical operations.

- Pandas: For data manipulation and analysis.

- Scikit-learn: For training and evaluating regression models.

- Matplotlib: For creating static and interactive visualizations.

- Seaborn: For enhancing visualizations and analyzing relationships between variables.

3.3. Model Creation

Linear regression is used for predicting continuous numerical values. It establishes a linear relationship between the input features and the target variable. The model assumes that the relationship can be approximated by a straight line in a multi-dimensional space. The mathematical formula for linear regression can be expressed as:

$$y = mx + b$$

where:

y represents the target variable to be predicted,

x denotes the input feature(s),

m represents the slope of the line (indicating the relationship between x and y),

b is the y-intercept (indicating the point where the line intersects the y-axis).

A decision tree is a non-linear algorithm that uses a tree-like structure to make decisions based on the input features. It breaks down the dataset into smaller subsets by repeatedly partitioning the data based on the feature values. Each internal node represents a test on a specific feature, while each leaf node represents the mathematical formula used for decision tree splitting criteria can be explained using the Gini impurity:

$$Gini(D) = 1 - \Sigma (\pi)^2$$

where:

D represents a dataset at a particular node,

pi denotes the proportion of the samples belonging to a specific class in D.ents a predicted value or a class label.

Random forest is an ensemble learning algorithm that combines multiple decision trees to make more accurate predictions. It creates a forest of decision trees, where each tree is trained on a different subset of the data and a random subset of features. The final prediction is obtained by aggregating the predictions of individual trees.

The mathematical formulas used in random forest are based on the decision tree algorithms described above, as random forest builds upon the principles and techniques of decision trees. The random forest algorithm aggregates the predictions from multiple decision trees to improve prediction accuracy and reduce the risk of overfitting.

IV. Experimentation

For linear regression, the LinearRegression model from the sklearn.linear_model library was imported. The model was initialized, trained using the training data (train_X and train_y), and predictions were made on the test data (test_X). The mean absolute error (MAE) was calculated using np.mean(np.absolute(prediction-test_y)), providing an assessment of the average absolute difference between the predicted and actual values. Additionally, the variance score (R^2) was calculated using model.score(test_X, test_y), indicating the proportion of the variance in the target variable explained by the model.

In the case of decision tree regression, the DecisionTreeRegressor model from the sklearn.tree library was imported. The model was initialized with a random_state of 0, trained using the training data, and predictions were made on the test data. The MAE was computed using np.mean(np.absolute(prediction2-test_y)), capturing the average absolute difference between the predicted and actual values. The R^2 score was obtained using regressor.score(test_X, test_y), providing an indication of the goodness-of-fit of the model to the data.

For random forest regression, the RandomForestRegressor model from the sklearn.ensemble library was imported. The model was initialized with specified parameters (max_depth=90, random_state=0, n_estimators=100), trained using the training data, and predictions were made on the test data. The MAE was computed using np.mean(np.absolute(prediction3-test_y)), measuring the average absolute difference between the predicted and actual values. The R^2 score was obtained using regr.score(test_X, test_y), assessing the explained variance in the target variable by the model.

Evaluation:

The model is evaluated on the basis of the following metrics:

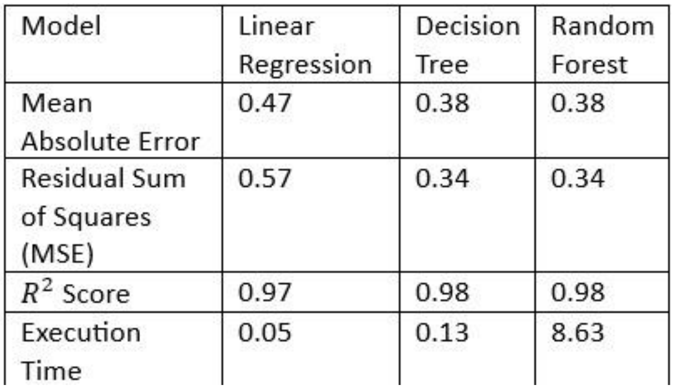
- Variance Score: A measure of how well the regression model captures the variation in the target variable, indicating the proportion of the target's variance explained by the model.

- Mean Absolute Error: The average absolute difference between the predicted and actual values, providing a measure of the model's average prediction error.

- Residual sum of squares: The sum of the squared differences between the predicted and actual values, quantifying the overall model fit by evaluating the total prediction error.

- R2 Score: A statistical measure indicating the proportion of the variance in the target variable that can be explained by the regression model, with higher values indicating better model performance.

V. Results and discussion

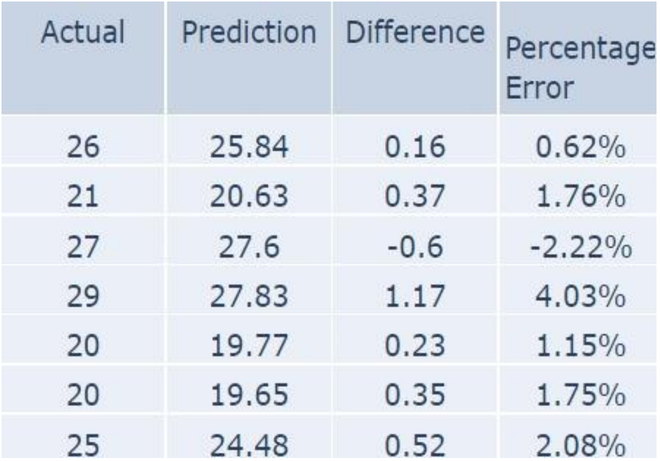


Model	Linear Regression	Decision Tree	Random Forest
Mean Absolute Error	0.47	0.38	0.38
Residual Sum of Squares (MSE)	0.57	0.34	0.34
R^2 Score	0.97	0.98	0.98
Execution Time	0.05	0.13	8.63

Figure 5.1 Comparison of MSE, MAE, R^2 Score, Execution Time

The following percentage error tables only visualize a few rows of actual vs predicted values. The dataset used for analysis consists of over 95,000 rows, making it impractical to display all the values in a table. Instead, a scatterplot illustrating the relationship between the actual and predicted values is presented below. This visualization provides a comprehensive overview of the performance of the model across the entire dataset.

1. Linear Regression



Actual	Prediction	Difference	Percentage Error
26	25.84	0.16	0.62%
21	20.63	0.37	1.76%
27	27.6	-0.6	-2.22%
29	27.83	1.17	4.03%
20	19.77	0.23	1.15%
20	19.65	0.35	1.75%
25	24.48	0.52	2.08%

Figure 5.1.1, Actual vs Predicted values for Linear Regression

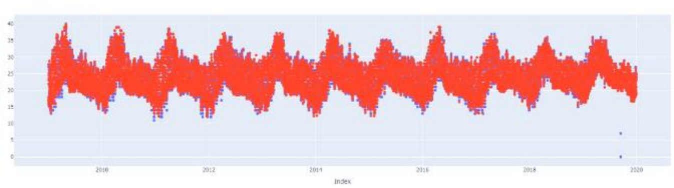
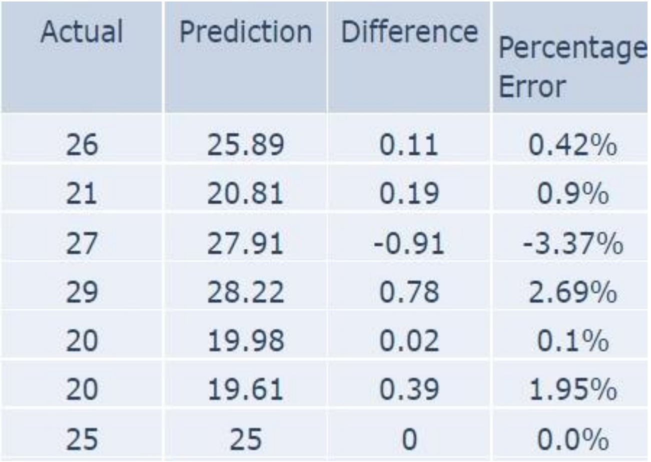


Figure 5.1.2 Scatter plot for Linear Regression

2. Decision Tree



Actual	Prediction	Difference	Percentage Error
26	25.89	0.11	0.42%
21	20.81	0.19	0.9%
27	27.91	-0.91	-3.37%
29	28.22	0.78	2.69%
20	19.98	0.02	0.1%
20	19.61	0.39	1.95%
25	25	0	0.0%

Figure 5.2.1 Actual vs Predicted values for Decision Tree

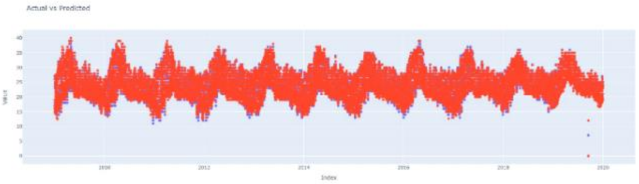
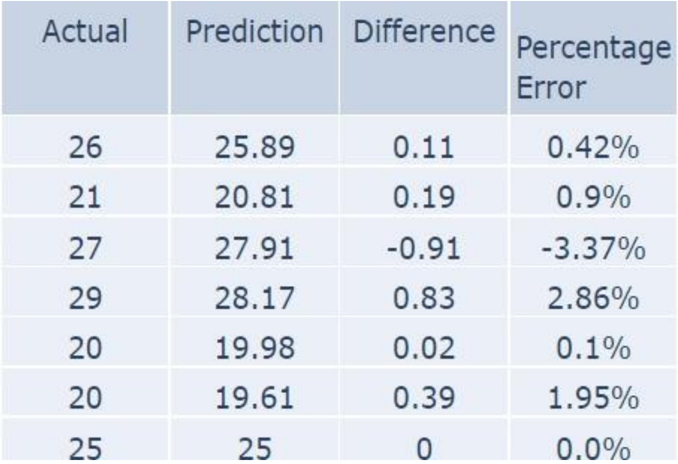


Figure 5.2.2 Scatter plot for Decision Tree

3. Random Forest



Actual	Prediction	Difference	Percentage Error
26	25.89	0.11	0.42%
21	20.81	0.19	0.9%
27	27.91	-0.91	-3.37%
29	28.17	0.83	2.86%
20	19.98	0.02	0.1%
20	19.61	0.39	1.95%
25	25	0	0.0%

Figure 5.3.1 Actual vs Predicted values for Random Forest

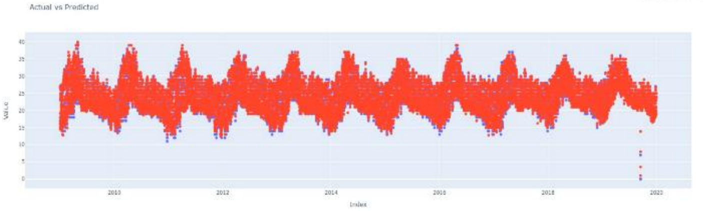


Figure 5.3.2 Scatter Plot for Random Forest

Both RandomForestRegressor and DecisionTreeRegressor models outperform linear regression, with higher variance scores (0.98 vs. 0.97), lower mean absolute error (0.38 vs. 0.48), lower residual sum of squares (0.34 vs. 0.57), and higher R2-scores (0.98 vs. 0.97). The decision tree model is slightly more accurate in predicting temperatures for the past week compared to the random forest. However, both models perform exceptionally well with the historic data. These results indicate that the tree-based models, particularly the decision tree, are effective for temperature prediction

VI. Conclusion

After comparing the results, it can be concluded that decision trees are the most suitable algorithm for implementing the tree on a large scale. While linear regression offers fast computation and lower time complexity, its accuracy is not satisfactory. On the other hand, decision trees provide accurate scores despite having higher complexity than linear regression, yet still lower than that of random forest. Although random forest yields highly accurate results, it has the highest time complexity among the three algorithms. Hence, based on this comparison, decision trees are deemed the optimal choice for large-scale implementation. forecasting and underscore their potential for practical applications in weather prediction and related domains.

References

[1] Aravind M ,Thilak S ,Vigneshwaran B ,Dr.J.B.Jona, Weather prediction Using Random Forest Methods, IJCRT 2022

[2]C. M. KISHTAWAL,Use of satellite observations for weather prediction, MAUSAM 2019

[3]Ferreira, B.; Silva, R.G.; Iten, M. Earth Observation Satellite Imagery Information Based Decision Support Using Machine Learning. Remote Sens. 2019