AIML Challenge II

TEAM Name: NEUROLUX
TEAM Members: R Rajamari, S Saran, Raman Kishore R R
KPRIET, Arasur, Coimbatore - 641048

September 2024

Contents

1 Introduction

The problem addressed in this study revolves around the accurate prediction of temperature using historical weather data from multiple geographical locations. Given the importance of weather forecasting in various sectors such as agriculture, disaster management, and environmental planning, improving the precision of temperature predictions is essential. Current weather prediction models may not fully exploit the relationships between geographical features (latitude, longitude, and elevation) and weather variables (precipitation, snow depth, temperature). By developing a machine learning model that integrates these factors, this study aims to provide more accurate temperature forecasts. The impact of this work extends beyond academic interest; it holds practical implications for improving agricultural planning, enabling better preparedness for natural disasters, and supporting climate monitoring efforts. The insights gained from this model can aid in optimizing resource management and risk mitigation strategies, particularly in regions that are vulnerable to extreme weather conditions. Additionally, the findings may serve as a foundation for future advancements in climate modeling and data-driven weather forecasting systems.

2 Data Preprocessing

2.1 Filling Missing Values

Filling the missing values is an important task in data preprocessing.In the given dataset , there are around 2500(approx.) NULL values.we must impute these NULL values using statistical methods like mean,median,mode.We opted not to use mean imputation for NULL values because geographical features like Latitude and Longitude remain constant for each location. Using the mean could skew the imputation and introduce bias, so we used the median for a more robust approach. we can use mode but imputing NULL values with mode can turn our model biased.So it is better to impute NULL values with median

2.2 Type Conversion

In our given dataset we have date column, which is in String type,but computers can compute using only numbers,So we must split them into fragments of 4 column:Day,Month,Year,DayOfWeek and convert them into interger type

2.3 Data Scaling

There are lot of options available for Scaling.In our dataset we did NULL value imputation based on median,So we do robust scaling in our data.Robust scaling will scale the data based on median and interquartile range(IQR).(Scaling is done for all columns excluding Target column)

$$X_{scaled} = \frac{X - Median(X)}{IQR(X)}$$

2.4 Feature Extraction

After finding the baseline model that gives the highest accuracy,We must select some column from which the model learns the most.To select the columns, we use Permutation importance model from

sklearn.It is based on the idea of shuffling the values of a feature to see how it affects the model's performance.

$$Importance(f) = BaselinePerformance - Performanceaftershufflingfeaturef$$

While other methods like Recursive Feature Elimination (RFE) and LASSO are often used for feature selection, we chose permutation importance due to its ability to capture feature interactions and its straightforward application in tree-based models.

2.5 Anomaly detection and Removal

Isolation Forest is an unsupervised algorithm for anomaly detection. It isolates anomalies by creating random decision trees with splits at random values. Anomalies are isolated quickly with fewer splits, while normal points require more splits. This approach makes Isolation Forest efficient and scalable for large and high-dimensional datasets. The anomaly score $s(x, n)$ is given by:

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}$$

$h(x)$ is the path length of point x. $E(h(x))$ is the average path length in the Isolation Forest. $c(n)$ is the average path length of a Binary Search Tree, given by:

3 Analysing the trends and patterns

3.1 Analysing Target variable

From Figure 1, we observe that the target variable has a trimodal distribution, indicating multiple underlying patterns in the data. Ensemble models like Random Forest or Gradient Boosting are better suited for this type of distribution compared to linear regression, as they handle non-linear relationships and local patterns more effectively. They reduce bias and variance through averaging or sequential learning, capture complex interactions between features, and are more robust to outliers and noise. These factors make ensemble models a more accurate choice for predicting trimodal data than linear regression.

3.2 Anomalies detection

Figures 2 and 3 compare outlier detection using the Isolation Forest algorithm with combination parameters of 0.1 and 0.5. At 0.1, fewer data points are marked as outliers, and the normal points form a compact cluster along the central trend. At 0.5, more data points are classified as outliers, leading to a wider spread of red points across the plot

The reason for these differences lies in the varying sensitivity of the combination parameter. A lower value of 0.1 applies a more conservative approach, flagging fewer anomalies, which is ideal for preserving the integrity of the dataset when only a small number of true outliers are expected. On the other hand, a higher value of 0.5 results in a more aggressive detection approach, which might lead to over-classification of outliers, especially in a dataset of around 800 rows, where the increased number of outliers could be less representative of the actual data structure.

Based on this analysis, the combination parameter of 0.1 is best suited for this dataset. It provides a more balanced approach to outlier detection, ensuring that the normal data points are

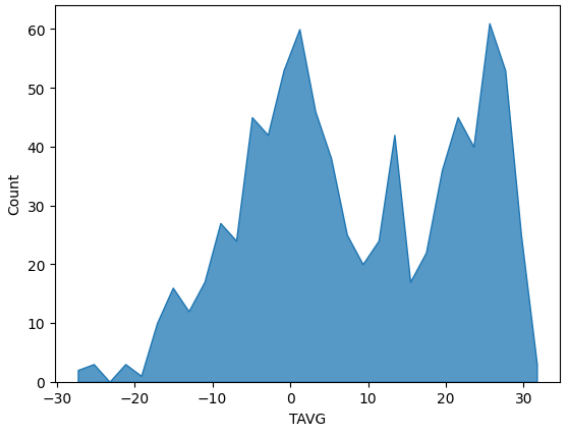

Figure 1: Analysing the type of skewness the target variable have

preserved while identifying genuine anomalies, making it the optimal choice for a dataset of this size.

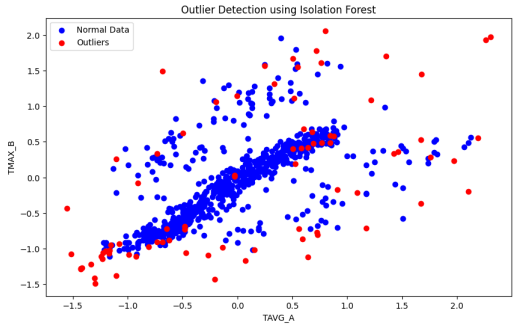

Figure 2: Results when combination = 0.1

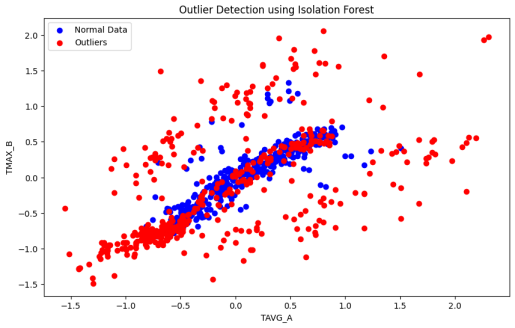

Figure 3: Results when combination = 0.5

3.3 Feature Importance

The graph illustrates the permutation importance of various features in the model. The most influential feature is **Month**, which has the highest impact on the prediction. This is followed closely by weather-related features such as **TMAX_B**, **TAVG_A**, and **TMIN_B**, indicating that maximum and minimum temperatures from different locations (A, B, and C) are key factors in the model's predictive performance. Other features like **Year**, **DayOfWeek**, and **PRCP** (precipitation) contribute less, showcasing that temporal and geographical variables play a secondary role. The chart emphasizes that temperature-related variables across multiple locations are critical for accurate predictions.So it is wise to drop the columns which are contributing to the model's decision.'**TAVG_A', 'TMAX_B', 'TMIN_B','TAVG_B', 'TAVG_C','Month'** are taken and all the other columns are dropped while training.

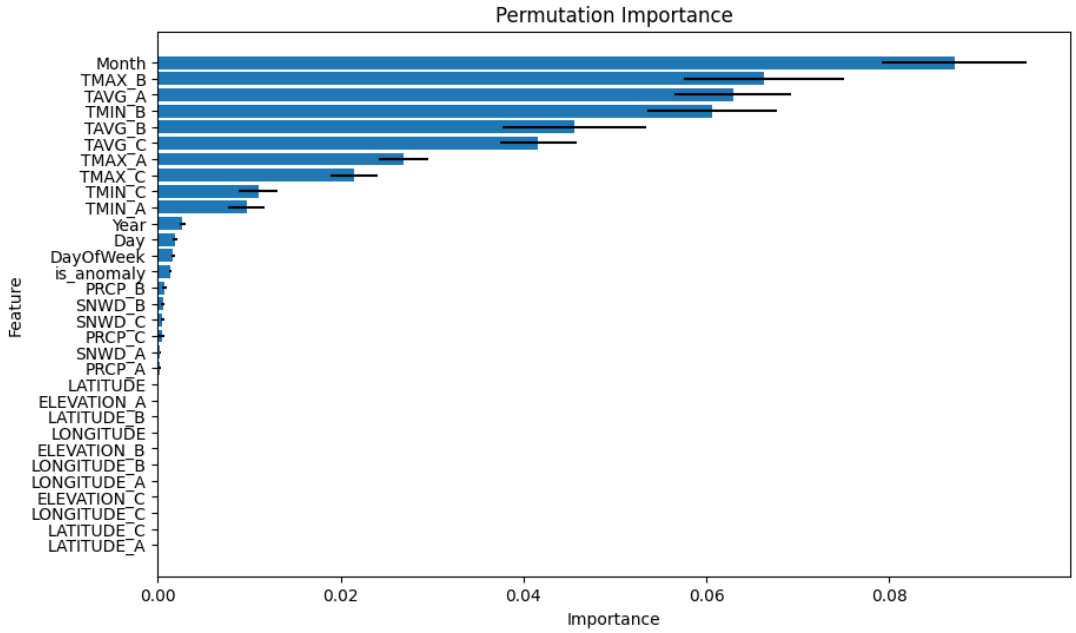

Figure 4: Graph depicts importance given to each column by the model

4 Model Selection

As we came to an conclusion , that our data has trimodal distribution,we must use ensemble(tree based or boosting) models.From ensemble we choose GradientBoosting,RandomForestRegressor,ExtratreesRegressor.

4.1 GradientBoosting

Gradient Boosting is an ensemble learning technique that builds models sequentially, where each new model attempts to correct the errors of the previous one. It works by minimizing a differentiable loss function through gradient descent. Given a dataset (x_i, y_i), the goal is to find a function $F(x)$ that minimizes the loss $L(y, F(x))$. The model starts with an initial prediction $F_0(x)$, and iteratively updates the model by adding new weak learners $h_m(x)$ to correct the residuals:

$$F_m(x) = F_{m-1}(x) + \gamma h_m(x)$$

where γ is a learning rate and $h_m(x)$ is the weak learner fit to the negative gradient of the loss function. Gradient Boosting is highly flexible, capable of handling various loss functions, and effective in reducing both bias and variance, making it suitable for complex, non-linear data.

4.2 Random Forest Regressor

Random Forest Regressor is an ensemble learning method that constructs multiple decision trees during training and averages their predictions to improve accuracy and reduce overfitting. For regression tasks, the final prediction is the mean of the outputs from individual trees. Each tree is trained on a random subset of the data using a technique called bootstrap aggregation (bagging), which helps in reducing variance. Mathematically, given T trees and their predictions $h_1(x), h_2(x), \ldots, h_T(x)$ for a data point x, the Random Forest prediction is:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} h_t(x)$$

This approach provides robustness to noise and is effective in capturing non-linear relationships in the data while preventing overfitting through averaging.

4.3 Extratrees

Extra Trees Regressor (Extremely Randomized Trees) is an ensemble learning method similar to Random Forest but introduces more randomness to reduce variance and improve model robustness. Unlike Random Forest, where trees are built using bootstrap sampling and the best split is selected based on a criterion like Gini or MSE, Extra Trees selects the split points randomly. For regression tasks, the final prediction is the average of the outputs from all individual trees. Mathematically, given T trees and their predictions $h_1(x), h_2(x), \ldots, h_T(x)$ for a data point x, the Extra Trees Regressor prediction is:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} h_t(x)$$

By randomly selecting split points and using the entire dataset (no bootstrapping), Extra Trees reduces variance while maintaining high accuracy, especially for noisy data.

5 Model testing

Model	Accuracy (%)	Error (MAE)	Error (MSE)
Gradient Boosting Regressor	97.8	1.63	4.5
Random Forest Regressor	98.03	1.54	4.3
Extra Trees Regressor	98.44	1.37	3.3

Table 1: Comparison of Accuracy and Error (MAE, MSE) for Random Forest, Gradient Boosting, and Extra Trees Regressors

Note:*All these scores are based on testing that is done after splitting train.csv into train set and test set (internal testing scores)*

The table above presents a performance comparison between three regression models: Gradient Boosting Regressor, Random Forest Regressor, and Extra Trees Regressor. The metrics used for the evaluation include Accuracy, Mean Absolute Error (MAE), and Mean Squared Error (MSE).

"The main reason why gradientboostingRegressor gave low accuracy is due of lack of data,because of this it either overfits the training data or underfits the data. Apart from this multicollinearity also affects the training"

"The reason why Extra Trees Regressor gives great perfomance than Random Forest Regressor is because Extra trees is Extremely Randomized version of Random Forest.Extratrees give more randomness in spliting which decreases the variance of model"

Based on this comparison, the Extra Trees Regressor demonstrates the best balance between accuracy and error reduction, making it the most suitable model for further hyperparameter tuning and optimization.

6 Prediction

To do the prediction in the test.csv (given test data) preprocessing work that is done on train.csv should be repeated.New columns that are not trained during training should be droped.The final prediction should be in same format as training data target variable.

7 Conclusion

In summary, Extra Trees Regressor emerges as the most promising model for temperature prediction, offering high accuracy and reduced error rates due to its Extreme Randomness,Faster tree growth. This model can be further fine-tuned to enhance performance in real-world applications such as agricultural planning and disaster management, where accurate weather forecasting is crucial.