Forecasting UV Index using Machine Learning Methods

• Abhirup Paria • Aradhya Kanth • Kunal Aggarwal • Soumalya Saha • Harsh Priyam

Abstract

Elevating public awareness about the severe health consequences of ultraviolet (UV) radiation is imperative. The risks, ranging from sunburns and premature ageing to more critical conditions like skin cancer and immunosuppression, necessitate urgent attention. Mitigating these risks requires not only a cultural shift towards sun-safe practices but also reliable tools for prediction, such as accurate ultraviolet index (UVI) forecasts. Empowering individuals with this knowledge enables proactive measures, reducing overall solar UV exposures and contributing to a healthier, more informed public.This study aimed to develop and compare the performances of different machine learning approaches with different feature selection methods to forecast the daily UVI of Kolkata city, India. The machine learning algorithms are incorporated coupled with five feature selection algorithms(i.e. LASSO Regression(L1 regularisation), Random Forest Feature Importance method, ANOVA Test, Recursive Feature Elimination with Cross-Validation (RFECV), Gradient Boosting Feature Importance) to understand the diverse combinations of the predictor variables acquired from the dataset. Compared to the counterpart benchmark models, the results demonstrated the excellent forecasting capability (i.e., low error and high efficiency) of the recommended machine learning model in apprehending the complex and non-linear relationships between predictor variables and the daily UVI. This collective approach emphasises the importance of informed public health strategies to combat the severe impact of ultraviolet radiation on our well-being.

1. Introduction

Solar ultraviolet (UV) radiation, constituting a small yet vital fraction (approximately 5–7%) of total radiation, plays a fundamental role in sustaining life on Earth. Over the ages, its benefits have been harnessed for human health, from enhancing immune systems and strengthening bones to treating challenging skin conditions like atopic dermatitis, psoriasis, and localised scleroderma. UV-induced tanning not only has mood-enhancing effects but also contributes to relaxation. Additionally, UV light's role in generating nitrogen oxide (NO) proves beneficial in reducing blood pressure, while its application as a disinfectant in food and water industries is well-established, countering disease-producing microorganisms.

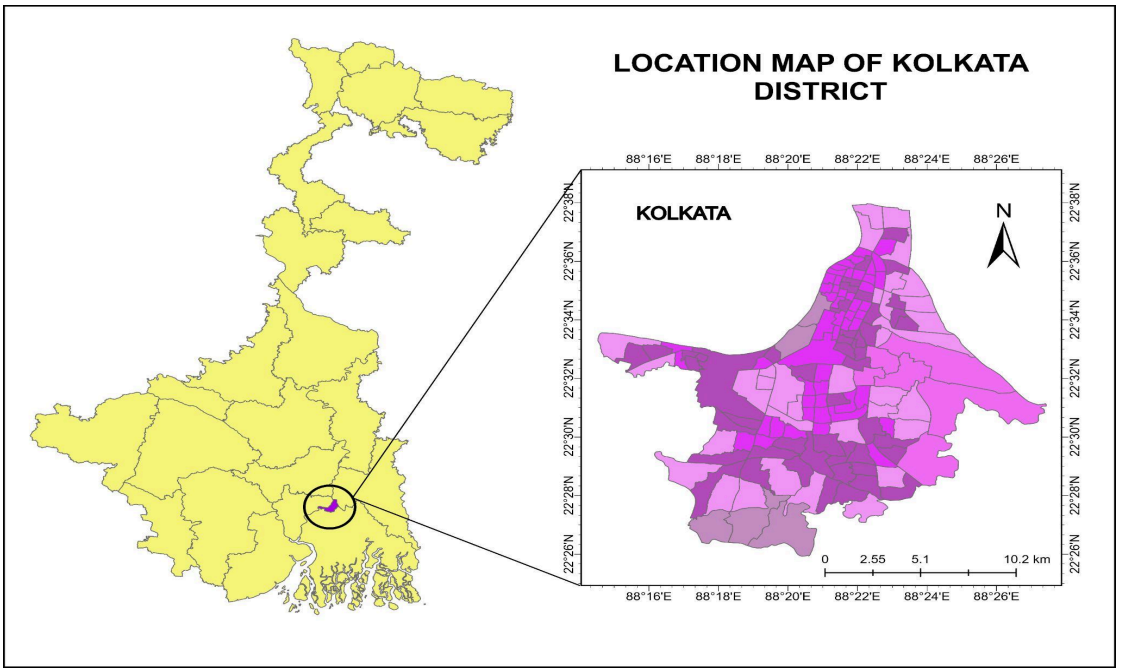
While UV radiation has been extensively utilised for water disinfection in Europe and is gaining traction in developing countries due to its simplicity and cost-effectiveness, its dichotomous nature poses serious concerns. Chronic exposure to UV light, a significant risk factor for melanoma and non-melanoma cancers, underscores the need for accurate information dissemination, especially in regions like India. Providing correct UV irradiance intensity information becomes crucial to protect health, not only for individuals at risk but also for various sectors such as agriculture, medical services, and water management. Efforts to forecast the Ultraviolet Index (UVI) have evolved, with researchers employing machine learning and deep learning techniques alongside deterministic approaches. Notably, the machine learning model demonstrated superior performance, offering promising avenues for enhancing UV index predictions. In contrast to traditional process-driven and empirical models, the adoption of various machine learning (ML) algorithms as data-driven models has emerged as a highly successful approach, leveraging their formidable computational efficiency. Technological progress has substantially boosted computational capabilities, leading to the development of a myriad of ML tools. Notably, studies have extensively employed ML algorithms such as support vector regression (SVR), decision trees, and random forests for estimating UV erythemal irradiance. It is worth highlighting that, while UV irradiance values remain a focus in these investigations, the UV Index (UVI) indicator stands out as a more explicit and user-friendly metric for the general public.

In this study, we employed a model of different machine learning algorithms and compared them to analyse which model best suits for the given dataset. Moreover, we have applied different Feature Selection Techniques and applied machine Learning models in each selection output to take useful insights from it.
Further, only a few data driven models have been applied for UVI forecasting. For example, an ANN was used in modelling UVI on a global scale (Latosin´ska et al. 2015). An extreme learning method (ELM) was applied in forecasting UVI in the Australian context (Deo et al. 2017). There have not been many studies that used many ML methods to forecast UVI. This study incorporates five feature selection algorithms (i.e. L1 Regularisation, Random Forest Feature Importance, ANOVA test, Gradient Boosting Feature Importance, Recursive Feature Elimination with Cross-Validation) to optimise the training procedure and try different predictor variables selected by the feature selection algorithms. Adapting different feature selection approaches would give a diverse understanding of the predictors and effectively quantify the features of UVI.

2. Materials and methods

2.1 Datasets of predictor variables

The model considers a diverse set of predictor variables to estimate the **UV INDEX** in Kolkata City. These variables encompass various environmental and meteorological parameters, providing a comprehensive overview of the factors influencing the UV index.
These variables collectively represent the atmospheric, climatic, and geographical conditions that contribute to the variability in the UV index. The integration of such a diverse set of predictors enhances the model's ability to capture the intricate relationships and nuances associated with UV index prediction in Kolkata City.



PARAMETER	DESCRIPTION
ALLSKY_SFC_LW_DWN	CERES SYN1deg All Sky Surface Longwave Downward Irradiance (W/m^2)
ALLSKY_SFC_PAR_TOT	CERES SYN1deg All Sky Surface PAR Total (W/m^2)
ALLSKY_SFC_SW_DIFF	CERES SYN1deg All Sky Surface Shortwave

	Diffuse Irradiance (MJ/m^2/day)
ALLSKY_SFC_SW_DNI	CERES SYN1deg All Sky Surface Shortwave Downward Direct Normal Irradiance (MJ/m^2/day)
ALLSKY_SFC_SW_DWN	CERES SYN1deg All Sky Surface Shortwave Downward Irradiance (MJ/m^2/day)
ALLSKY_SFC_UVA	CERES SYN1deg All Sky Surface UVA Irradiance (W/m^2)
ALLSKY_SFC_UVB	CERES SYN1deg All Sky Surface UVB Irradiance (W/m^2)
ALLSKY_SFC_UV_INDEX	CERES SYN1deg All Sky Surface UV Index (dimensionless)
ALLSKY_SRF_ALB	CERES SYN1deg All Sky Surface Albedo (dimensionless)
CLOUD_AMT	CERES SYN1deg Cloud Amount (%)
CLRSKY_SFC_PAR_TOT	CERES SYN1deg Clear Sky Surface PAR Total (W/m^2)
CLRSKY_SFC_SW_DWN	CERES SYN1deg Clear Sky Surface Shortwave Downward Irradiance (MJ/m^2/day)
GWETPROF	MERRA-2 Profile Soil Moisture (1)
GWETROOT	MERRA-2 Root Zone Soil Wetness (1)
GWETTOP	MERRA-2 Surface Soil Wetness (1)
PRECTOTCORR	MERRA-2 Precipitation Corrected (mm/day)
PRECTOTCORR_SUM	MERRA-2 Precipitation Corrected Sum (mm)
PS	MERRA-2 Surface Pressure (kPa)
QV2M	MERRA-2 Specific Humidity at 2 Meters (g/kg)
RH2M	MERRA-2 Relative Humidity at 2 Meters (%)
T2M	MERRA-2 Temperature at 2 Meters (C)
T2MDEW	MERRA-2 Dew/Frost Point at 2 Meters (C)
T2MWET	MERRA-2 Wet Bulb Temperature at 2 Meters (C)
T2M_MAX	MERRA-2 Temperature at 2 Meters Maximum (C)
T2M_MIN	MERRA-2 Temperature at 2 Meters Minimum (C
T2M_RANGE	MERRA-2 Temperature at 2 Meters Range (C)
TOA_SW_DWN	CERES SYN1deg Top-Of-Atmosphere Shortwave Downward Irradiance (MJ/m^2/day)
TS	MERRA-2 Earth Skin Temperature (C)

WD10M	MERRA-2 Wind Direction at 10 Meters (Degrees)
WD2M	MERRA-2 Wind Direction at 2 Meters (Degrees)
WS10M	MERRA-2 Wind Speed at 10 Meters (m/s)
WS10M_MAX	MERRA-2 Wind Speed at 10 Meters Maximum (m/s)
WS10M_MIN	MERRA-2 Wind Speed at 10 Meters Minimum (m/s)
WS10M_RANGE	MERRA-2 Wind Speed at 10 Meters Range (m/s)
WS2M	MERRA-2 Wind Speed at 2 Meters (m/s)
WS2M_MAX	MERRA-2 Wind Speed at 2 Meters Maximum (m/s)
WS2M_MIN	MERRA-2 Wind Speed at 2 Meters Minimum (m/s
WS2M_RANGE	MERRA-2 Wind Speed at 2 Meters Range (m/s)

2.2 Feature Selection Technique

2.2.1 Lasso Regression (L1 Regularisation)

With Ridge regression we introduced the idea of penalisation that could result in estimators with smaller MSE, benefiting from a bias-variance trade-off in the estimation process.The penalisation in ridge regression shrinks the estimators towards 0. However, due to the nature of the penalisation, the estimators never reach zero no matter how much penalisation we introduce.

The Lasso uses a similar idea as ridge, but it uses a $\ell 1$ penalisation ($\ell 1$ norm is given by $|\beta| = \sqrt{\sum_{n=1}^{p} |\beta_j|}$) that allows the coefficients to shrink exactly to 0.
This way, the estimation process has embedded a variable selection procedure, because if a coefficient shrinks to 0, it is the same as removing the variable from the model.
To get the Lasso estimates we have to minimise:

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Similar to ridge regression, the amount of penalisation is controlled by the parameter (λ) than can be chosen by cross-validation.

2.2.2 Random Forest Feature Importance

Random forests (RF) construct many individual decision trees at training. Predictions from all trees are pooled to make the final prediction; the mode of the classes for classification or the mean prediction for regression. As they use a collection of results to make a final decision, they are referred to as Ensemble techniques.
Feature importance is calculated as the decrease in node impurity weighted by the probability of reaching that node. The node probability can be calculated by the number of samples that reach the node, divided by the total number of samples. The higher the value the more important the feature.
For each decision tree, Scikit-learn calculates a nodes importance using Gini Importance, assuming only two child nodes (binary tree):

$$ni_j = w_j C_i - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)}$$

- ni sub(j)= the importance of node j
- w sub(j) = weighted number of samples reaching node j
- C sub(j)= the impurity value of node j
- left(j) = child node from left split on node j
- right(j) = child node from right split on node j

The importance for each feature on a decision tree is then calculated as:

$$fi_j = \sum_{j:\ node\ j\ splits\ on\ feature\ i} ni_j \ / \sum_{k \in all\ nodes} ni_k$$

- fi sub(i)= the importance of feature i
- ni sub(j)= the importance of node j

These can then be normalised to a value between 0 and 1 by dividing by the sum of all feature importance values:

$$normfi_j = fi_j / \sum_{j \in all\ features} fi_j$$

The final feature importance, at the Random Forest level, is it's average over all the trees. The sum of the feature's importance value on each trees is calculated and divided by the total number of trees:

$$\text{RFfi}_j = \sum_{j \in all\ trees} normfi_j / T$$

- RFfi sub(i)= the importance of feature i calculated from all trees in the Random Forest model
- normfi sub(ij)= the normalised feature importance for i in tree j
- T = total number of trees.

2.2.3 ANOVA TEST

ANOVA stands for Analysis of variance. As the name suggests it uses variance as its parameter to compare multiple independent groups. ANOVA can be one-way ANOVA or two-way ANOVA. One-way ANOVA is applied when there are three or more independent groups of a variable. ANOVA, is a statistical test used to analyse the difference between the means of more than two groups.

F-statistic can be calculated by

F = MST / MSE

$$\text{MST} = \frac{\sum_{I=1}^{K}\left(T_i^2/n_i\right) - G^2/n}{k-1}$$

$$\text{MSE} = \frac{\sum_{i=1}^{k}\sum_{j=1}^{n_i} Y_{ij}^2 - \sum_{i=1}^{k}(T_i^2/n_i}{n-k}$$

2.2.4 Recursive Feature Elimination with Cross-Validation (RFECV):

Recursive Feature Elimination (RFE) is a feature selection algorithm that is used to select a subset of the most relevant features from a dataset. It is a recursive process that starts with all the features in the dataset and then iteratively removes the least essential features until the desired number of features is reached. The main logic behind RFE is that the most relevant features will have the highest impact on the target variable, and thus will be more useful for predicting the target. RFE uses a model (such as a linear regression or support vector machine) to evaluate the importance of each feature, and the features with the lowest importance are eliminated in each iteration.

Cross-validation is a technique for evaluating the performance of a machine learning model by training it on a subset of the data and then testing it on a different subset. It is a way to assess the generalisation ability of a model, i.e., how well the model is able to make predictions on unseen data.

2.2.5 Gradient Boosting Feature Importance

The mathematical calculation of feature importance in gradient boosting models is typically based on the number of times a feature is selected for splitting across all the trees in the ensemble and the improvement it brings to the model's performance. The specifics may vary slightly between different implementations like XGBoost, LightGBM, or scikit-learn's GradientBoostingRegressor, but the general idea remains the same.

For each split in a tree, a "gain" is computed, which represents the improvement in accuracy brought by a feature.

The feature importance is the average gain across all splits involving that feature.

Importance= average gain across all splits for the feature sum of all average gains for all features / sum of all average gains for all features average gain across all splits for the feature

Cover represents the relative quantity of observations concerned by a feature. It is calculated as the number of observations affected by a feature in all splits. Importance=sum of cover values across all splits for the feature sum of all cover values for all features / sum of cover values across all splits for the feature. Frequency: The number of times a feature is used to split the data across all trees. Importance=frequency of the feature across all splits total number of splits in the ensemble / frequency of the feature across all splits.

2.3 Standalone models

2.3.1 Linear Regression

In linear regression, the coefficients of the model indicate the importance or weight assigned to each feature in predicting the target variable. The mathematical formula for simple linear regression is:

$$y = \beta_0 + \beta_1 x_1 + \in$$

Where:

- y is the target variable,
- x_1 is the feature,
- β_0 is the y-intercept (constant term),
- β_1 is the coefficient of the feature,
- $\in$ is the error term.

In the case of multiple linear regression with multiple features, the formula becomes:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots\ldots + \beta_n x_n + \in$$

where $x_1, x_2, \ldots\ldots, x_n$ *are the features, and* $\beta_0, \beta_1, \beta_2, \ldots\ldots\ldots, \beta_n$ *are the coefficients.*

The coefficients $\beta_0, \beta_1, \beta_2, \ldots\ldots\ldots, \beta_n$ are estimated during the training of the linear regression model. These coefficients are determined in a way that minimises the sum of squared differences between the predicted and actual values (least squares method).

The coefficient β_i represents the change in the target variable for a one-unit change in the corresponding feature x_i, assuming all other variables are held constant.

The ordinary least squares (OLS) method is commonly used to estimate the coefficients in linear regression. The formulas for β_0 *and* β_1 in simple linear regression are:

$$\beta_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

where:

N is the number of data points,

$\bar{x}$ *is the mean of the x values,*

$\bar{y}$ is the mean of the y values.

2.3.2 K-Nearest Neighbors Regressor

K-Nearest Neighbors (KNN) is a non-parametric and lazy learning algorithm used for both regression and classification tasks. In the context of regression, KNN predicts the target variable for a new data point by

averaging the target values of its $\bar{k}$ nearest neighbours. Here's how the prediction is calculated in a KNN Regressor:

Euclidean Distance Calculation

The distance between data points is typically calculated using the Euclidean distance formula. For two points P(x_1, y_1) and Q(x_2, y_2) the Euclidean distance *d* is calculated as: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Finding the K Nearest Neighbours

The algorithm identifies the *k* training instances that are closest to the new data point based on Euclidean distance.

Regression Prediction

For regression, the predicted target value for the new data point is typically the average of the target values of its *k* nearest neighbours. The formula is:

$\hat{y} = \frac{1}{k}\sum_{i=1}^{k} y_i$

where:

- $\hat{y}$ *is the predicted target value,*
- k is the number of neighbours,
- y_i is the target value of the *i*-th neighbour.

2.3.3 Decision Tree Regressor

A decision tree is a predictive model used for classification and regression analysis (Jime´nez-Pe´rez and Mora-Lo´pez 2016). As our data is continuous, we used it for the regression predictions. It is a simple tree-like structure that uses the input observations (i.e., $x_1, x_2, \ldots, x_n$) to predict the target output (i.e., Y). The tree contains many nodes, and at each node, a test to one of the inputs (e.g., x_1) is applied, and the outcome is estimated. The left/right sub branch of the decision tree is selected based on the estimated outcome. After a specific node, the prediction is made, and the corresponding node is termed the leaf node. The prediction averages out all the training points for the leaf node. The model is trained using all input variables and corresponding loss; the mean squared error (MSE) is calculated to determine the best split of the data. The maximum features are set as the total input features during the partition. Therefore, Decision Tree is how to best split the dataset into smaller and smaller subsets to predict the target value. The condition, or test, is represented as the “leaf” (node) and the possible outcomes as “branches” (edges). This splitting process continues until no further gain can be made or a preset rule is met, e.g. the maximum depth of the tree is reached.

Many things would be calculated while evaluating the decision tree model.

These are:

Gini Impurity = $\sum_{i=1}^{c} f_i(1 - f_i)$

Entropy = $\sum_{i=1}^{c} - f_i log(f_i)$

$f_i = frequency\ of\ label\ i\ at\ a\ node\ and\ c\ is\ the\ number\ of\ unique\ labels.$

2.3.4 Random Forest Regressor

The Random Forest algorithm is an ensemble learning technique that builds multiple decision trees and merges their predictions for improved accuracy and generalisation.

Bootstrapped Sampling:

For each tree in the forest, a random subset of the training data is sampled with replacement. This is known as bootstrapped sampling.Random Subset of Features:

At each node split during the construction of each decision tree, a random subset of features is considered. This introduces diversity among the trees.

Decision Trees:

Multiple decision trees are grown independently. Each tree is constructed by recursively partitioning the data based on feature values.

Voting (Classification) or Averaging (Regression):

For regression tasks, the final prediction is the average of the predictions from all trees.

2.3.5 Support Vector Machine Regressor

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression tasks. In the case of regression, it is referred to as Support Vector Machine Regressor. SVM aims to find a hyperplane that best fits the data while minimising the error.

Hyperplane:

In SVM regression, the algorithm seeks to find a hyperplane that best fits the data.For a one-dimensional problem, the hyperplane is a line; for a two-dimensional problem, it's a plane, and so on.The hyperplane is chosen to have the maximum margin from the data points.

Loss Function:

SVM Regressor uses a loss function to penalise errors in prediction.The loss function aims to minimise the sum of the errors while allowing for a margin of tolerance ($\overline{\epsilon}$).

Epsilon-Insensitive Tube:

SVM Regressor defines an epsilon-insensitive tube around the hyperplane.Data points within this tube are not penalised, and points outside the tube contribute to the loss.

Objective Function:

The objective function involves minimising the sum of the errors while keeping them within the epsilon-insensitive tube. It includes a regularisation term to control the complexity of the model.

Kernel Trick:

SVM can use the kernel trick to transform the input features into a higher-dimensional space, potentially making the problem more separable.

Objective function:

The objective function for SVM Regressor with a linear kernel is often written as:

$$min_{w,b,\xi,\xi^*} \frac{1}{2}|w|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$

subject to

$$y_i - w^T x_i - b \leq \in + \xi_i$$

$$w^T x_i + b - y_i \leq \in + \xi_i^*$$

$$\xi_i, \xi^* \geq 0$$

Here,w and b are the parameters of the hyperplane,
$\xi_i \; and \; \xi_i^* \; are \; slack \; variables, \; and \; C \; is \; the \; regularisation \; parameter.$

2.3.6 AdaBoostRegressor

AdaBoost (Adaptive Boosting) is an ensemble learning technique that combines multiple weak learners to create a strong learner. In the case of regression, it is referred to as AdaBoost Regressor. The algorithm focuses more on instances that are poorly predicted by the previous weak learners, adapting its strategy over iterations.

AdaBoost uses a series of weak learners, often decision trees with limited depth. Each weak learner is trained on the dataset, and its predictions are combined to form a strong learner. During training, the algorithm assigns different weights to data points.The weights are adjusted at each iteration, giving more emphasis to data points that are poorly predicted by the current ensemble. The final prediction is a weighted sum of the predictions from all weak learners. The weights are based on the performance of each weak learner, with more accurate learners having higher weights.

Weighted Error of Weak Learner:

- The error of a weak learner is calculated as the weighted sum of misclassified data points.
- For a weak learner with predictions $h(x_i) \; and \; true \; labels \; y_i, \; the \; error(\in) \; is \; given \; by$:

$$\epsilon = \frac{\sum_{i=1}^{n} w_i I(y_i \neq h(x_i))}{\sum_{i=1}^{n} w_i}$$

where I(.) is the indicator function.

Weight of Weak Learner in Ensemble:

- The weight (α) assigned to a weak learner based on its error is calculated as:

$$\alpha = \frac{1}{2} log(\frac{1-\epsilon}{\epsilon})$$

- Accurate learners have higher weights, and those with higher errors have lower weights.

Weight Update for Data Points:

- The weights of the data points are updated to give more emphasis to misclassified points:

$$w_i \leftarrow w_i exp(-\alpha y_i h(x_i))$$

- This update increases the weights of misclassified points.

Final Prediction:

- The final prediction is a weighted sum of the weak learners' predictions:

$$F(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

- *T* is the number of weak learners.

2.3.7 XGBoost Regressor

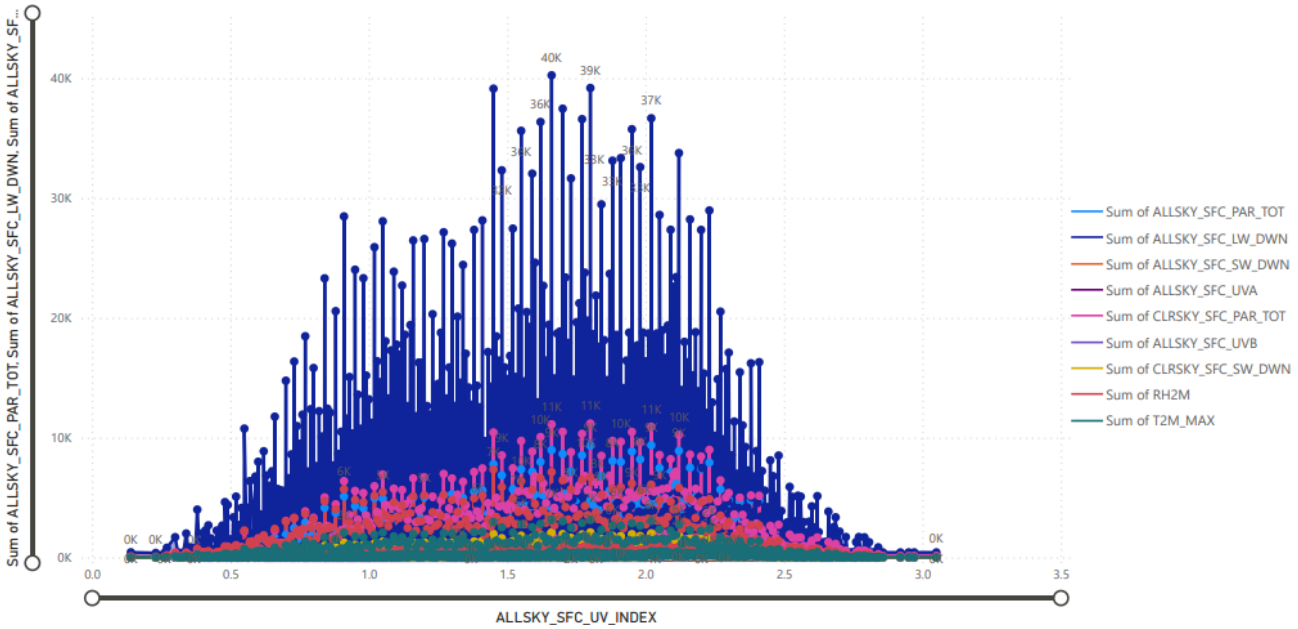
XGBoost (Extreme Gradient Boosting) is an advanced and highly efficient implementation of the gradient boosting algorithm. It has gained popularity for its speed and performance in various machine learning competitions. In the context of regression, it is referred to as XGBoost Regressor.

XGBoost is based on the gradient boosting framework, where weak learners (decision trees) are trained sequentially.Each tree corrects the errors of the ensemble built so far.

The objective function for XGBoost regression is defined as the sum of a loss function and a regularization term:

$$\text{Objective} = \sum_{i=1}^{n} L(y_i, \widehat{y_i}) + \sum_{k=1}^{K} \Omega(f_k)$$

Here, L is the loss function (typically mean squared error for regression), $\widehat{y_i}$ is the predicted value, K is the number of trees, and $\Omega(f_k)$ is the regularisation term for the k-th tree.



.

Figure 1

Figure 1 describes the feature selection technique selects 9 important feature from the dataset. The most important feature extracted through this technique is All Sky Surface Longwave Downward Irradiance which shows the highest correlation with the target variable i.e. UV Index. The least correlated independent feature found from this technique is Temperature at 2 meters.

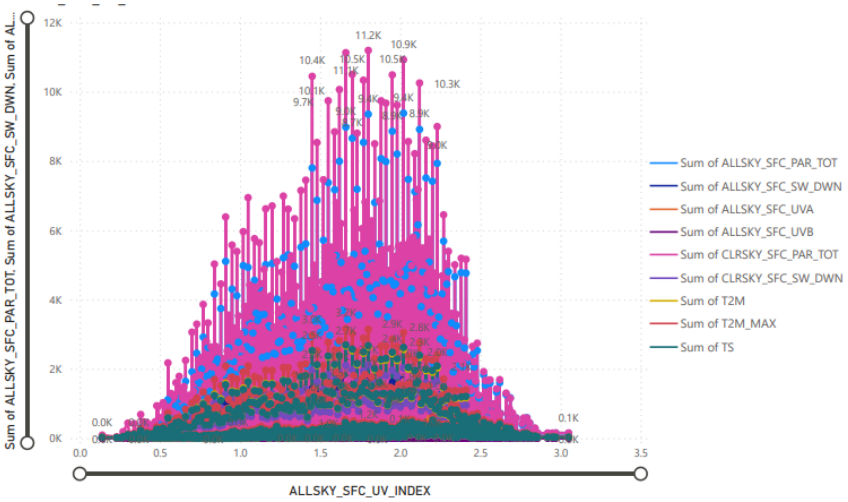


Figure 2

Figure 2 describes the feature selection technique selects 9 important feature from the dataset. The most important feature extracted through this technique is All Sky Surface Par Total which shows the highest correlation with the target variable i.e. UV Index. The independent feature which is least correlated found from this technique is Earth Skin Temperature.

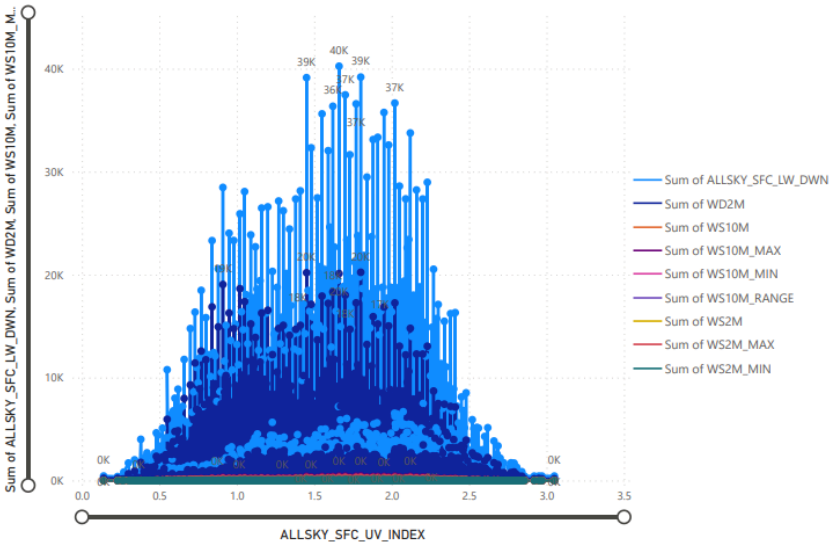


Figure 3

This feature selection technique selects 9 important feature from the dataset. The most important feature extracted through this technique is All Sky Surface Longwave Downward Irradiance which shows the highest correlation with the target variable i.e. UV Index. The least correlated independent feature found from this technique is Wind Speed at 2 meters minimum.

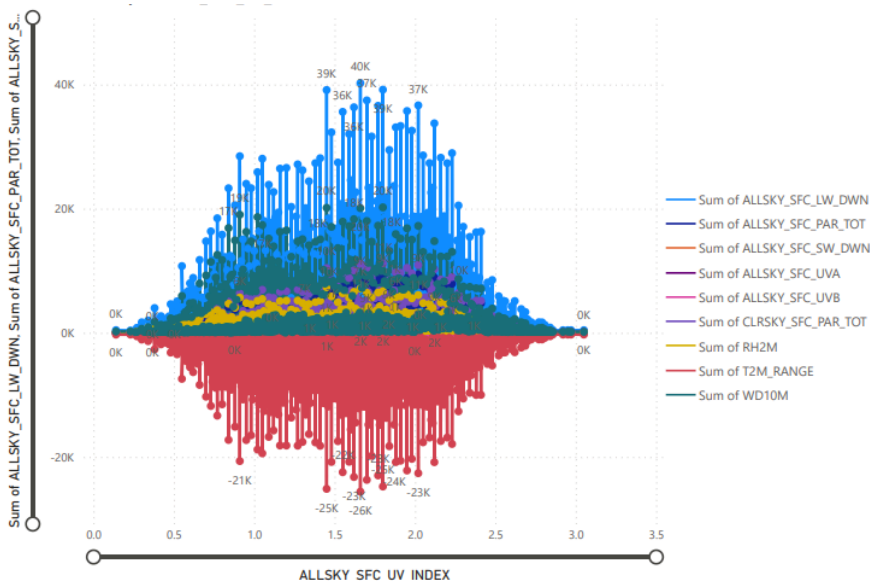


Figure 4

This feature selection technique selects 9 important feature from the dataset. The most important feature extracted through this technique is All Sky Surface Longwave Downward Irradiance which shows the highest correlation with the target variable i.e. UV Index. The least correlated independent feature found from this technique is Temperature at 2 meters Range.

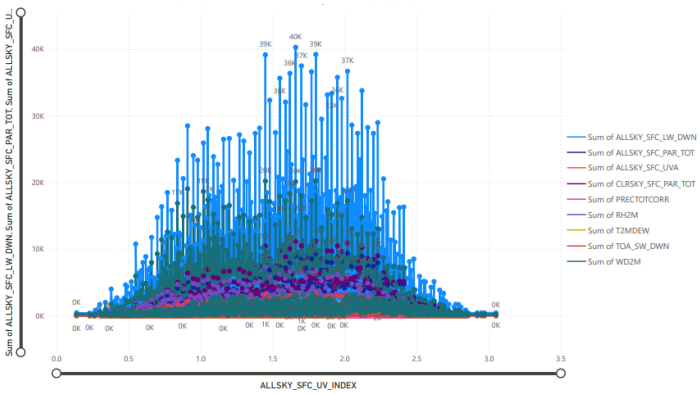
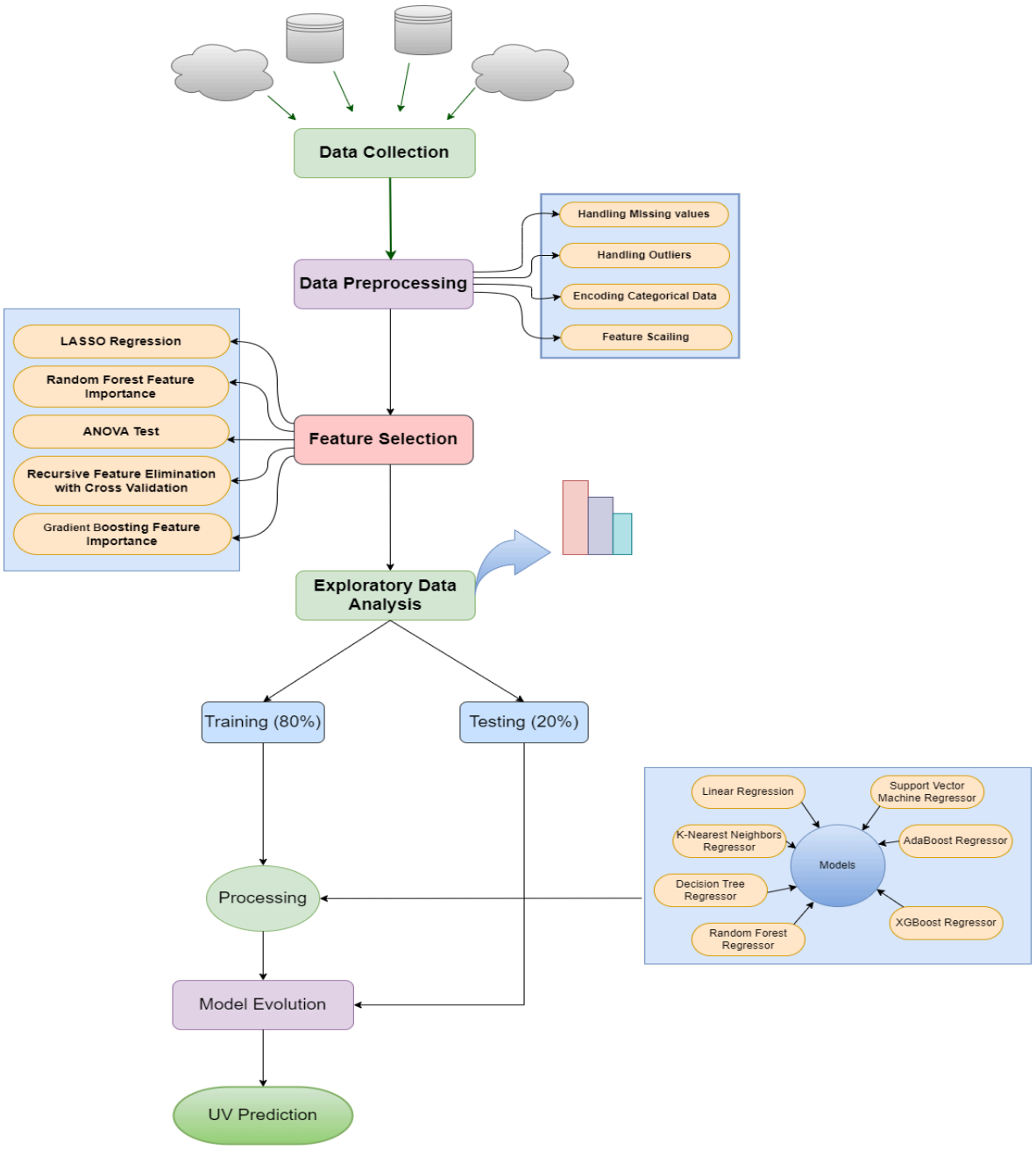


Figure 5

2.4 Model implementation procedure

In this section, we detail the implementation of the predictive modelling approach for estimating the UV INDEX based on the selected features. The model consists of a feature selection step and the application of various regression algorithms. We evaluate the performance of these algorithms using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared.



At first we extract UV INDEX data from various sources and prepare our dataset. After that we are doing various Feature Selection Techniques. Lasso Regression, Applied Lasso regression with an alpha value of 0.01 to select relevant features. Selected features based on non-zero coefficients. Random Forest Regression, Utilised Random Forest to identify feature importances. Selecting the top 9 features based on importances for further analysis. ANOVA Test, Calculated F-statistic and p-values for each feature. Selected the top 9 features based on F-statistic values. Recursive Feature Elimination (RFE), Employed RFE with Linear Regression as the estimator. Retained the top 9 features according to RFE ranking. Gradient Boosting Regression, Utilised Gradient Boosting Regressor to determine feature importances. Select the top 9 features based on importances. After all Feature Selection Technique, split the dataset into 80% training data point and 20% testing data point. Now, Performed model evaluation using the following regression algorithms: Linear Regression, K-Nearest Neighbors Regressor, Decision Tree Regressor, Random Forest Regressor, Support Vector Machine Regressor, AdaBoostRegressor, XGBoost Regressor. For each feature set (X1 to X5), the models were trained and evaluated on metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared. The results were recorded in a structured DataFrame for comprehensive analysis and reporting.

The chosen feature selection techniques and diverse regression models aim to provide a robust understanding of the dataset and contribute to accurate UV INDEX predictions.

3. Results

Five optimization methods (LASSO Regression (L1 Regularization), Random Forest Feature Importance, ANOVA Test, Recursive Feature Elimination with Cross-Validation (RFECV), Gradient Boosting Feature Importance) were employed to extract the crucial features of the target variable (UVI).

In Lasso Regression Feature Selection Technique, Importances array is array([1.30901564e-03, 1.14864973e-03, 0.00000000e+00, 1.27873951e-01, 0.00000000e+00, 3.57951087e-03, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 1.67233428e-03, 0.00000000e+00, 2.98663117e-03, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 4.84935644e-04, 0.00000000e+00, 0.00000000e+00, 2.19259071e-05, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 7.47820410e-05]. We get some useful Features: 'CERES SYN1deg All Sky Surface Shortwave Downward Irradiance (MJ/m^2/day)', 'CERES SYN1deg All Sky Surface PAR Total (W/m^2)', 'CERES SYN1deg All Sky Surface UVA Irradiance (W/m^2)', 'CERES SYN1deg Clear Sky Surface PAR Total (W/m^2)', 'MERRA-2 Relative Humidity at 2 Metres (%)', 'MERRA-2 Dew/Frost Point at 2 Metres (C)', 'CERES SYN1deg Top-Of-Atmosphere Shortwave Downward Irradiance (MJ/m^2/day)', 'MERRA-2 Wind Direction at 2 Metres (Degrees)', 'MERRA-2 Precipitation Corrected (mm/day)'.

In Random Forest Feature Importance, Importance array is Importances array is array([1.01207992e-04, 1.01360400e-04, 9.86379463e-05, 1.90155152e-04, 9.97816864e-01, 2.16141043e-04, 1.01519189e-04, 3.36735866e-05, 3.38679489e-05, 5.98031968e-05, 6.02355040e-05, 3.53644857e-05, 9.38330983e-05, 4.83962462e-05, 4.34290877e-05, 4.83016871e-05, 8.12032896e-05, 5.43346897e-05, 7.31699283e-05, 7.98619673e-05, 4.95705848e-05, 6.80761777e-05, 7.11260911e-05, 3.70567163e-05, 4.25681851e-05, 5.09478852e-05, 6.99640040e-05, 3.78944364e-05, 4.09692991e-05, 5.16049466e-05, 6.61400942e-05, 4.27212674e-05]). We choose the top 9 most important features. So, according to that we get Features: 'MERRA-2 Temperature at 2 Metres Maximum (C)', 'MERRA-2 Relative Humidity at 2 Metres (%)', 'CERES SYN1deg All Sky Surface Shortwave Downward Irradiance (MJ/m^2/day)', 'CERES SYN1deg All Sky Surface Longwave Downward Irradiance (W/m^2)', 'CERES SYN1deg All Sky Surface PAR Total (W/m^2)', 'CERES SYN1deg Clear Sky Surface Shortwave Downward Irradiance (MJ/m^2/day)', 'CERES SYN1deg All Sky Surface UVA Irradiance (W/m^2)', 'CERES SYN1deg Clear Sky Surface PAR Total (W/m^2)', 'CERES SYN1deg All Sky Surface UVB Irradiance (W/m^2)'.

Model metrics Evaluation:

On dataset obtained by Lasso Regression Feature Selection Technique:

Linear Regression MAE: 0.0509

Linear Regression RMSE: 0.0663

Linear Regression R-squared: 0.9829

K-Nearest Neighbors Regressor MAE: 0.0694

K-Nearest Neighbors Regressor RMSE: 0.0917

K-Nearest Neighbors Regressor R-squared: 0.9673

Decision Tree Regressor MAE: 0.0617

Decision Tree Regressor RMSE: 0.0825

Decision Tree Regressor R-squared: 0.9736

Random Forest Regressor MAE: 0.0428

Random Forest Regressor RMSE: 0.0574

Random Forest Regressor R-squared: 0.9872

Support Vector Machine Regressor MAE: 0.0643

Support Vector Machine Regressor RMSE: 0.083

Support Vector Machine Regressor R-squared: 0.9732

AdaBoost Regressor MAE: 0.0656

AdaBoost Regressor RMSE: 0.0821

AdaBoost Regressor R-squared: 0.9738

XGBoost Regressor MAE: 0.0428

XGBoost Regressor RMSE: 0.0575

XGBoost Regressor R-squared: 0.9872

On a dataset obtained by Random Forest Feature Importance…

Linear Regression MAE: 0.0171

Linear Regression RMSE: 0.0242

Linear Regression R-squared: 0.9977

K-Nearest Neighbors Regressor MAE: 0.0636

K-Nearest Neighbors Regressor RMSE: 0.0836

K-Nearest Neighbors Regressor R-squared: 0.9728

Decision Tree Regressor MAE: 0.0221

Decision Tree Regressor RMSE: 0.032

Decision Tree Regressor R-squared: 0.996

Random Forest Regressor MAE: 0.0158

Random Forest Regressor RMSE: 0.0228

Random Forest Regressor R-squared: 0.998

Support Vector Machine Regressor MAE: 0.065

Support Vector Machine Regressor RMSE: 0.0838

Support Vector Machine Regressor R-squared: 0.9727

AdaBoost Regressor MAE: 0.0356

AdaBoost Regressor RMSE: 0.0448

AdaBoost Regressor R-squared: 0.9922

XGBoost Regressor MAE: 0.0164

XGBoost Regressor RMSE: 0.0234

XGBoost Regressor R-squared: 0.9979

On a dataset obtained by Using ANOVA Test…

Linear Regression MAE: 0.0171

Linear Regression RMSE: 0.0242

Linear Regression R-squared: 0.9977

K-Nearest Neighbors Regressor MAE: 0.0631

K-Nearest Neighbors Regressor RMSE: 0.0825

K-Nearest Neighbors Regressor R-squared: 0.9736

Decision Tree Regressor MAE: 0.0217

Decision Tree Regressor RMSE: 0.0313

Decision Tree Regressor R-squared: 0.9962

Random Forest Regressor MAE: 0.0159

Random Forest Regressor RMSE: 0.0229

Random Forest Regressor R-squared: 0.998

Support Vector Machine Regressor MAE: 0.0522

Support Vector Machine Regressor RMSE: 0.0674

Support Vector Machine Regressor R-squared: 0.9823

AdaBoost Regressor MAE: 0.0351

AdaBoost Regressor RMSE: 0.0442

AdaBoost Regressor R-squared: 0.9924

XGBoost Regressor MAE: 0.0166

XGBoost Regressor RMSE: 0.0239

XGBoost Regressor R-squared: 0.9978

On a dataset obtained by Using Recursive Feature Elimination …

Linear Regression MAE: 0.3393

Linear Regression RMSE: 0.4283

Linear Regression R-squared: 0.2876

K-Nearest Neighbors Regressor MAE: 0.3349

K-Nearest Neighbors Regressor RMSE: 0.4274

K-Nearest Neighbors Regressor R-squared: 0.2903

Decision Tree Regressor MAE: 0.4265

Decision Tree Regressor RMSE: 0.5528

Decision Tree Regressor R-squared: -0.187

Random Forest Regressor MAE: 0.3081

Random Forest Regressor RMSE: 0.3946

Random Forest Regressor R-squared: 0.3953

Support Vector Machine Regressor MAE: 0.3367

Support Vector Machine Regressor RMSE: 0.4307

Support Vector Machine Regressor R-squared: 0.2795

AdaBoost Regressor MAE: 0.3412

AdaBoost Regressor RMSE: 0.4242

AdaBoost Regressor R-squared: 0.301

XGBoost Regressor MAE: 0.3122

XGBoost Regressor RMSE: 0.4

XGBoost Regressor R-squared: 0.3785

On a dataset obtained by Using Gradient Boosting Feature Importance

Linear Regression MAE: 0.0171

Linear Regression RMSE: 0.0242

Linear Regression R-squared: 0.9977

K-Nearest Neighbors Regressor MAE: 0.0688

K-Nearest Neighbors Regressor RMSE: 0.0911

K-Nearest Neighbors Regressor R-squared: 0.9677

Decision Tree Regressor MAE: 0.0214

Decision Tree Regressor RMSE: 0.0304

Decision Tree Regressor R-squared: 0.9964

Random Forest Regressor MAE: 0.0158

Random Forest Regressor RMSE: 0.023

Random Forest Regressor R-squared: 0.998

Support Vector Machine Regressor MAE: 0.0706

Support Vector Machine Regressor RMSE: 0.0906

Support Vector Machine Regressor R-squared: 0.9681

AdaBoost Regressor MAE: 0.0356

AdaBoost Regressor RMSE: 0.0448

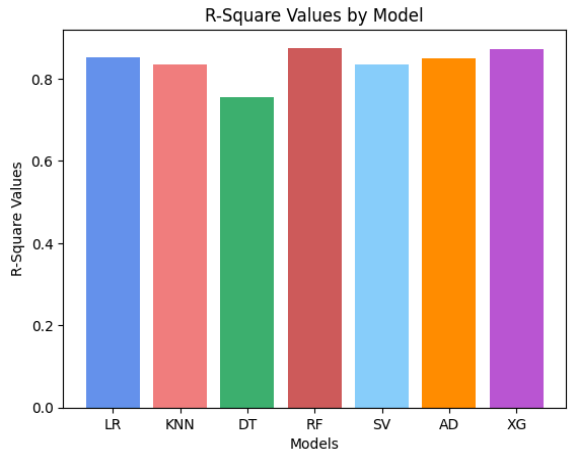
AdaBoost Regressor R-squared: 0.9922

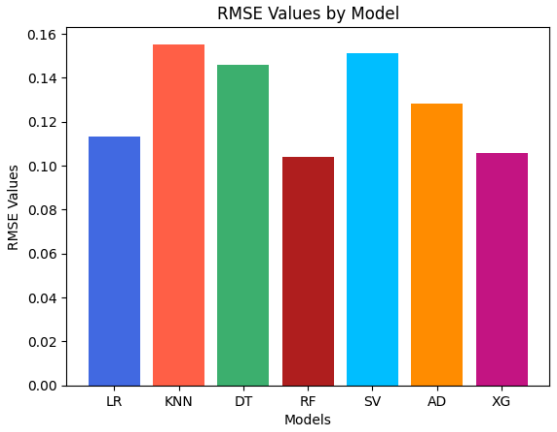
XGBoost Regressor MAE: 0.0163

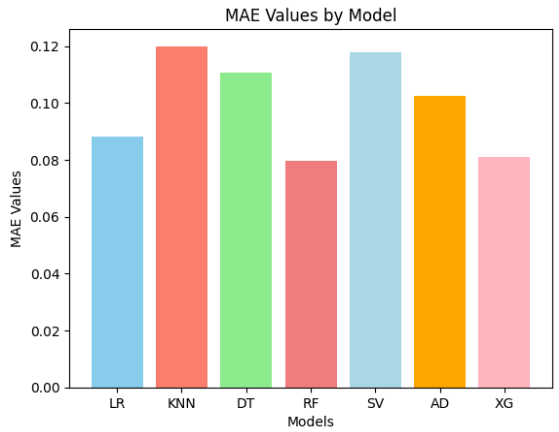
XGBoost Regressor RMSE: 0.0233

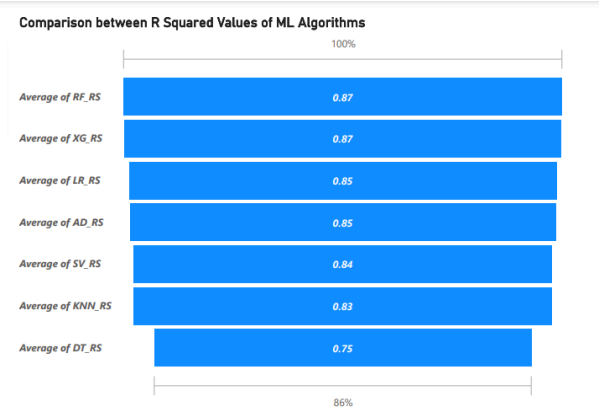
XGBoost Regressor R-squared: 0.9979

	L1 Regularization			**Random Forest Feature Importance**			**ANOVA**			**Recursive Feature Elimination**			**Gradient Boosting Feature Importance**		
	MAE	**RMSE**	**RS**	**MAE**	**RMSE**	**RS**	**MAE**	**RMSE**	**RS**	**MAE**	**RMSE**	**RS**	**MAE**	**RMSE**	**RS**
LR	**0.0509**	**0.0663**	**0.9829**	**0.0171**	**0.0242**	**0.9977**	**0.0171**	**0.0242**	**0.9977**	**0.3393**	**0.4283**	**0.2876**	**0.0171**	**0.0242**	**0.9977**
KNN	**0.0694**	**0.0917**	**0.9673**	**0.0636**	**0.0836**	**0.9728**	**0.0631**	**0.0825**	**0.9736**	**0.3349**	**0.4274**	**0.2903**	**0.0688**	**0.0911**	**0.9677**
DT	**0.0608**	**0.0812**	**0.9744**	**0.0218**	**0/0310**	**0.9963**	**0.0217**	**0.0318**	**0.9961**	**0.4296**	**0.5562**	**-0.2014**	**0.0215**	**0.0309**	**0.9963**
RF	**0.0427**	**0.0572**	**0.9873**	**0.0158**	**0.0229**	**0.9980**	**0.0159**	**0.0230**	**0.9980**	**0.3086**	**0.3956**	**0.3922**	**0.0157**	**0.0229**	**0.9980**
SVR	**0.0643**	**0.0830**	**0.9732**	**0.0650**	**0.0838**	**0.9727**	**0.0522**	**0.0674**	**0.9823**	**0.3367**	**0.4307**	**0.2795**	**0.0706**	**0.0906**	**0.9681**
AD	**0.0656**	**0.0821**	**0.9738**	**0.0356**	**0.0448**	**0.9922**	**0.0351**	**0.0442**	**0.9924**	**0.3412**	**0.4242**	**0.3010**	**0.0356**	**0.0448**	**0.9922**
XG	**0.0428**	**0.0575**	**0.9872**	**0.0164**	**0.0234**	**0.9979**	**0.0166**	**0.0239**	**0.9978**	**0.3122**	**0.4000**	**0.3785**	**0.0163**	**0.0233**	**0.9979**









4. CONCLUSION

Valuable insights can be gleaned from the Ultraviolet Index (UVI) in assessing the impact of solar UV radiation, recognized as a prominent health risk affecting both skin and eyes. This study endeavors to enhance public awareness regarding solar UV levels, aiming to mitigate the associated health risks, including conditions like malignant keratinocyte cancers and sun-related eye diseases. The research employs machine learning models to predict UVI data with a high degree of accuracy, contributing to the overall understanding and management of sun-induced health concerns.Five optimization methods (LASSO Regression (L1 Regularization), Random Forest Feature Importance, ANOVA Test, Recursive Feature Elimination with Cross-Validation (RFECV), Gradient Boosting Feature Importance) were employed to extract the crucial features of the target variable (UVI).

Here we have taken the average of all Mean Squared Error, Root mean Squared error , R Squared values of all machine learning algorithms we have used for the prediction. After looking into the above image result, we can see that **Random Forest** has performed best among all other algorithms in all Metrics Evaluation. After that XGBoost Regressor takes next place, next is Linear Regression, then AdaBoost, then Support Vector Regressor comes, next is K-Nearest Neighbors and at last is Decision Tree.

5. References

Optimization algorithms as training approach with hybrid deep learning methods to develop an ultraviolet index forecasting model. https://link.springer.com/article/10.1007/s00477-022-02177-3

Optimization algorithms as training approach with hybrid deep learning methods to develop an ultraviolet index forecasting model. https://link.springer.com/article/10.1007/s00477-022-02177-3

Deo RC, Downs N, Parisi AV, Adamowski JF, Quilty JM (2017) Very short-term reactive forecasting of the solar ultraviolet index using an extreme learning machine integrated with the solar zenith angle. Environ Res 155:141–166.

https://doi.org/10.1016/ j.envres.2017.01.035

Jiao G, Guo T, Ding Y (2016) A new hybrid forecasting approach applied to hydrological data: a case study on precipitation in Northwestern China. Water 8(9):367

Liu H, Tian H, Li Y (2015) Four wind speed multi-step forecasting models using extreme learning machines and signal decomposing algorithms. Energy Convers Manag 100:16–22. https://doi.org/10.1016/j.enconman.2015.04.057

Decision Tree Algorithm – A Complete Guide. https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/

Understanding Random Forest: How the Algorithm Works and Why it Is So Effective

https://towardsdatascience.com/understanding-random-forest-58381e0602d2

Juzeniene A, Moan J (2012) Beneficial effects of UV radiation other than via vitamin D production. Dermato-Endocrinology 4:109–117. https://doi.org/10.4161/derm.20013

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95: international conference on neural networks. Presented at the Proceedings of ICNN'95: international conference on neural networks, vol 4, pp 1942–1948. https://doi.org/10.1109/ICNN.1995.488968

Feature Importance and Feature Selection With XGBoost in Python. https://machinelearningmastery.com/feature-importance-and-feature-selection-with-xgboost-in-python/