**STA545 STATISTICAL DATA MINING I**

**Exploratory Data Analysis and Predictive Modeling of Temperature Changes**

**Group 7:** Nithyashree Govindarajan, Aseem Salim, Anushka Tiwari, Awnish Shankar

**ABSTRACT**

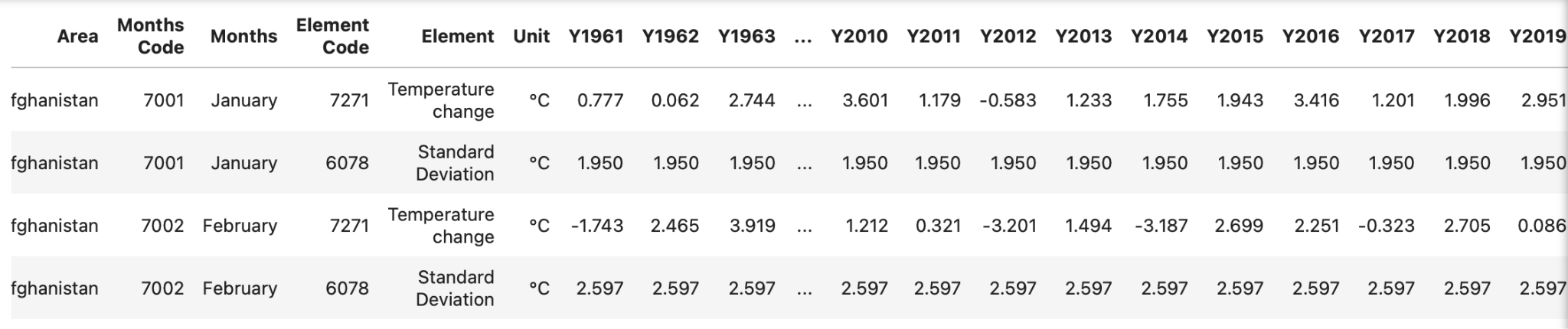
This project delves into predicting temperature changes using historical data through the application of diverse supervised learning models. The dataset, containing information on temperature variations over time, underwent meticulous preprocessing to address missing values and outliers. Employing regression models such as Ridge Regression, Decision Tree, Random Forest, SVR, and XGBoost, the study aimed to visualize trends, compare model performances, and assess robustness. Evaluation metrics including Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) provided a comprehensive understanding of model strengths and weaknesses. The introduction of bootstrap resampling techniques enhanced the robustness assessment, elucidating the stability of the models through metrics like mean MSE and R-squared.

**INTRODUCTION**

In the context of ongoing climate changes, understanding historical temperature variations is crucial for informed decision-making and policy planning. This project centers on a dataset containing temperature change information, spanning different regions and time periods. The primary objectives include visualizing temperature trends over time, developing predictive models using various machine learning algorithms, and assessing model stability through bootstrap resampling. The chosen models encompass a spectrum of complexity, allowing for a comprehensive evaluation of their performance. Through feature importance analysis and stability metrics, we aim to identify key contributors to temperature changes and ensure robustness in our predictions. This analysis holds significance in addressing climate-related challenges and enhancing our ability to anticipate future temperature patterns.

**DATA DESCRIPTION**

The dataset comprises 9,656 records with 66 columns, featuring information on temperature changes and related variables across different regions and time periods. The dataset covers a wide range of areas, from Afghanistan to the Organization for Economic Co-operation and Development (OECD). Each record includes details such as the area code, area name, month, and various elements related to temperature changes. The temporal aspect is captured through the years from 1961 to 2019. Notable variables include temperature change or standard deviation for years from 1961 to 2019, area, month, and other attributes associated with climatic conditions. The dataset provides a comprehensive perspective on temperature variations and is well-structured for exploratory data analysis and potential modeling to derive insights into climate trends over the specified time frame.



**MATERIALS AND METHODS**

In this project, we pursued two distinct approaches for temperature change prediction.

**Approach 1: Mean Temperature Change (2000 to 2019) as Target Variable**

The dataset was loaded into a DataFrame, and an initial exploration was conducted to understand its structure and contents. Following this, the data cleaning and preprocessing steps were implemented. Missing values were addressed by filling them with the mean of their respective columns, and the features were standardized using the StandardScaler from scikit-learn.

The target variable, representing the mean temperature change over the years 2000 to 2019, was calculated and added to the DataFrame. Additionally, a standard deviation column was introduced to capture the variability in temperature change. To enhance interpretability, unnecessary columns were dropped, resulting in a refined dataset for model training.

Subsequently, the dataset was split into training and testing sets, and various regression models were employed for temperature change prediction. These models included Linear Regression, Random Forest Regressor, Gradient Boosting Regressor, Lasso Regression, Support Vector Machine (SVM) Regressor, K-Nearest Neighbors (KNN) Regressor, Ridge Regression, and Decision Tree Regressor. Evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared were utilized to assess the performance of each model. The preprocessing and modeling procedures were orchestrated using the scikit-learn library and visualized through informative plots, allowing for a comprehensive analysis of temperature change trends and model performances.

**CODE**

**Importing libraries**

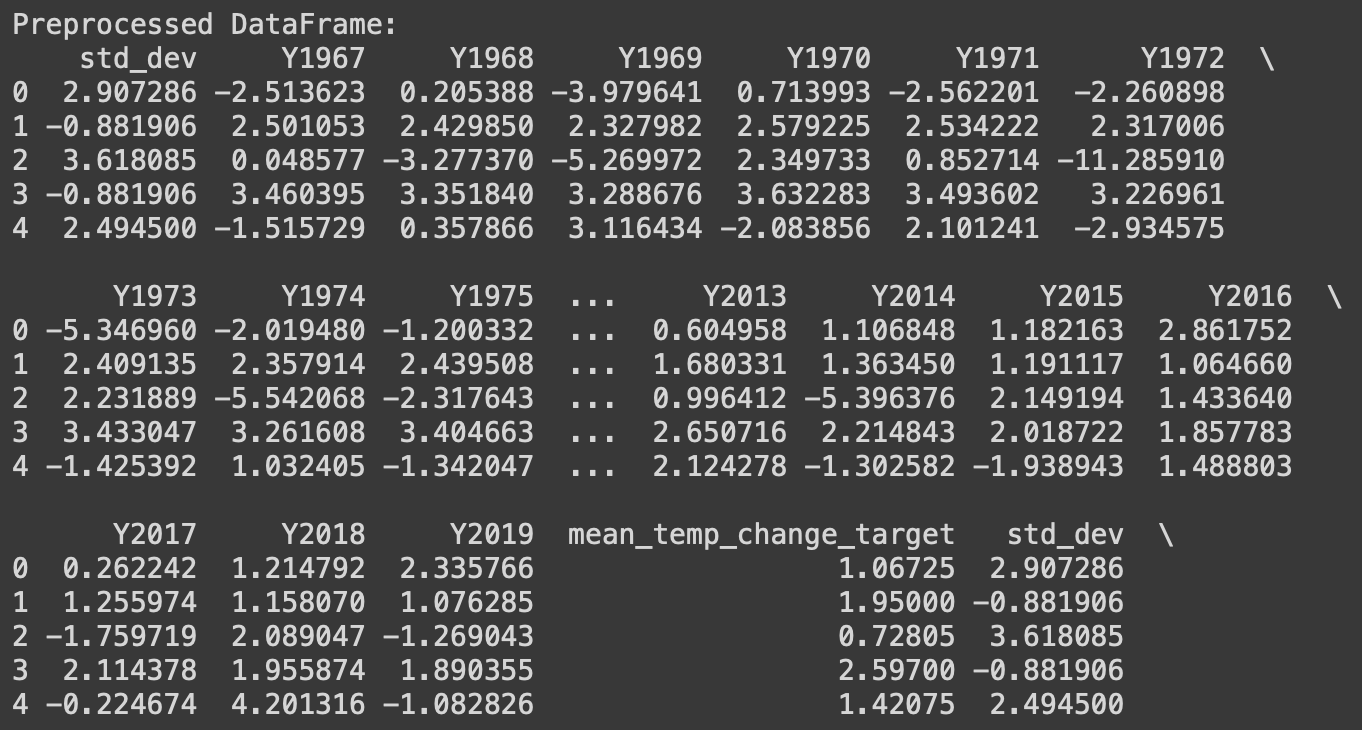
Necessary libraries are imported, including pandas for data manipulation, numpy for numerical operations, seaborn and matplotlib for visualization, and various modules from scikit-learn for machine learning tasks. The dataset is then loaded into a DataFrame called **df** using the **read\_csv** function.

**Exploratory Data Analysis and Target Variable Calculation**

Here the mean temperature change over the years 2000 to 2019 is calculated and stored in a new column called mean\_temp\_change\_target. The standard deviation of temperature change is also calculated and stored in a new column named std\_dev.

**Data Preprocessing**

The relevant columns are selected, checked for missing values, filled with the mean, standardized the features using **StandardScaler**, and created a new DataFrame with the preprocessed data.



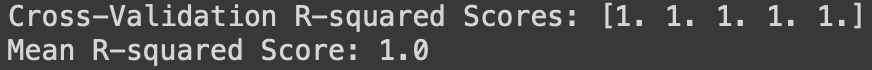
**Data Visualization**

This part visualizes the temperature change trends over time for the first five rows using a line plot. It provides insights into the data distribution and trends.



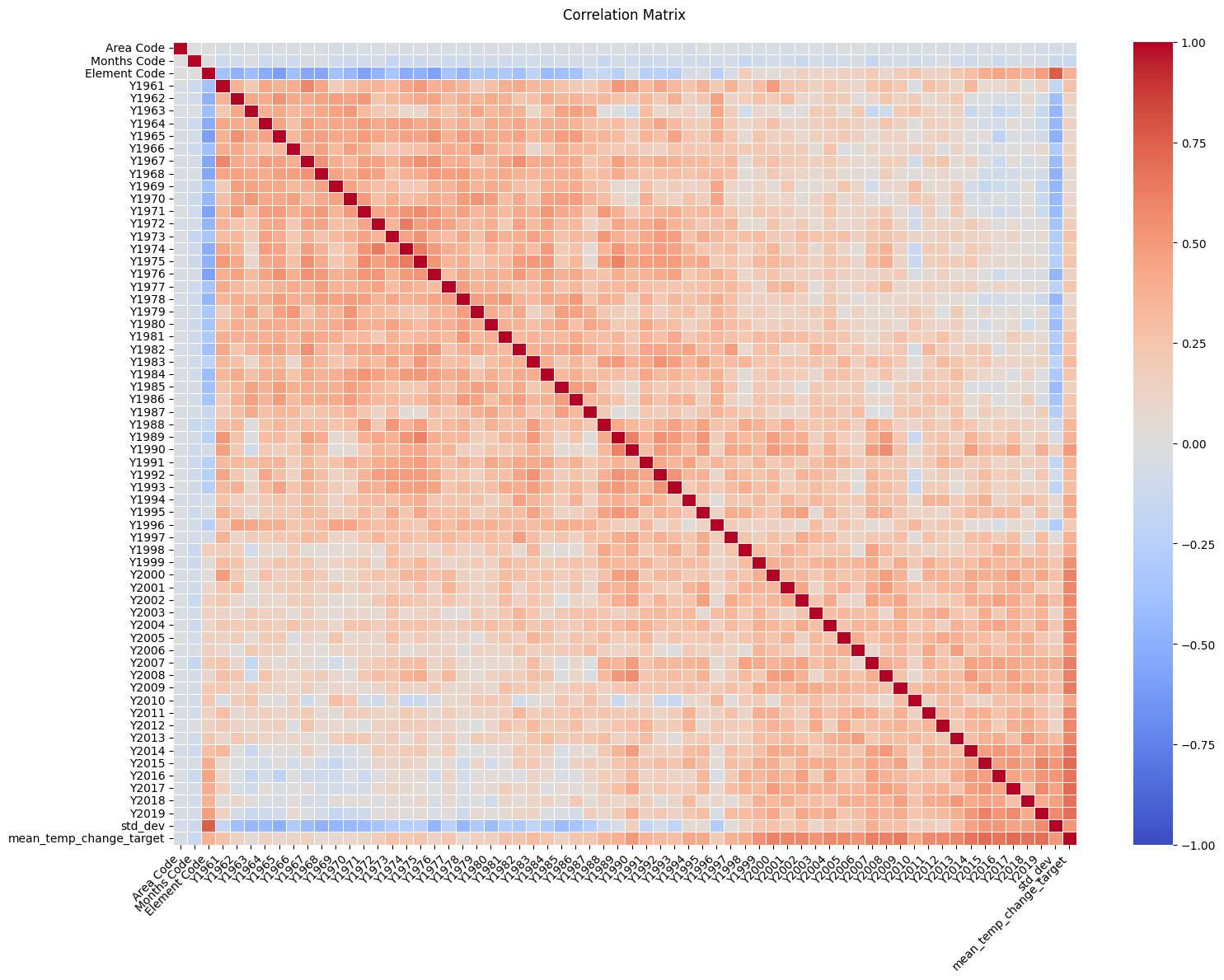
**Cross-Validation**

Cross-validation is performed using k-fold cross-validation with five folds to calculate and display the R-squared scores for model evaluation.



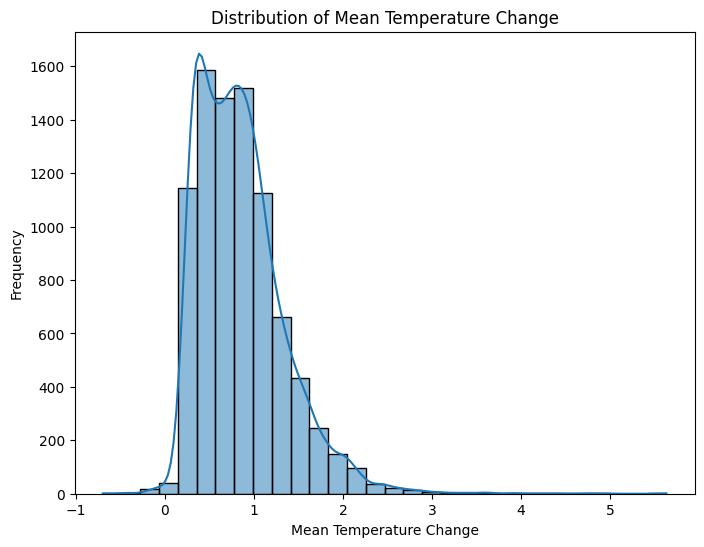
**Correlation Analysis and Visualization**

Statistical summaries and correlation matrices are printed to understand the distribution and relationships between variables.



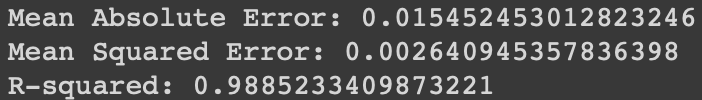
**Histogram and Distribution Visualization**

This part creates a histogram to visualize the distribution of the mean temperature change target variable.

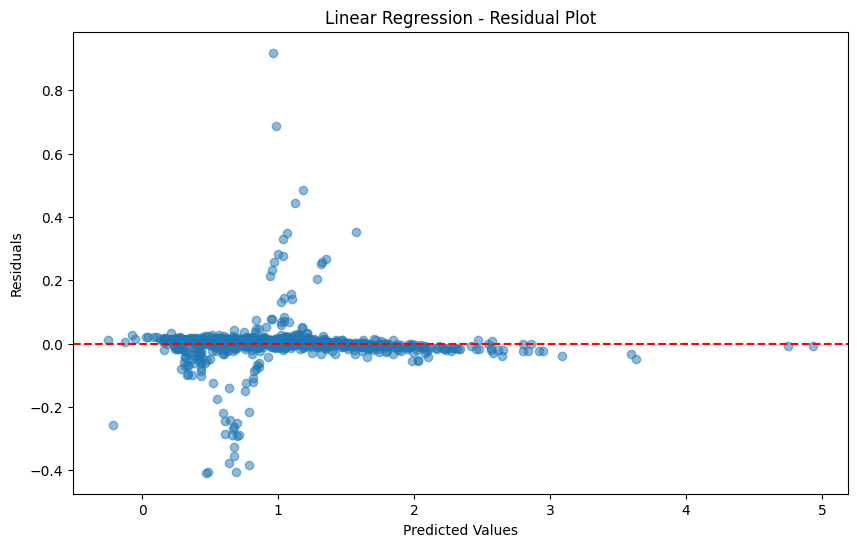


**Linear Regression**

The data is split into training and testing sets, trained a Linear Regression model, made predictions, and evaluated the model's performance using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared.

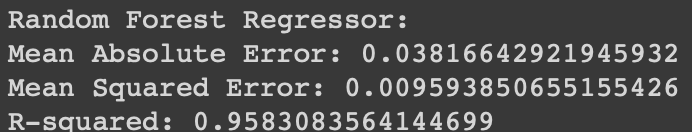


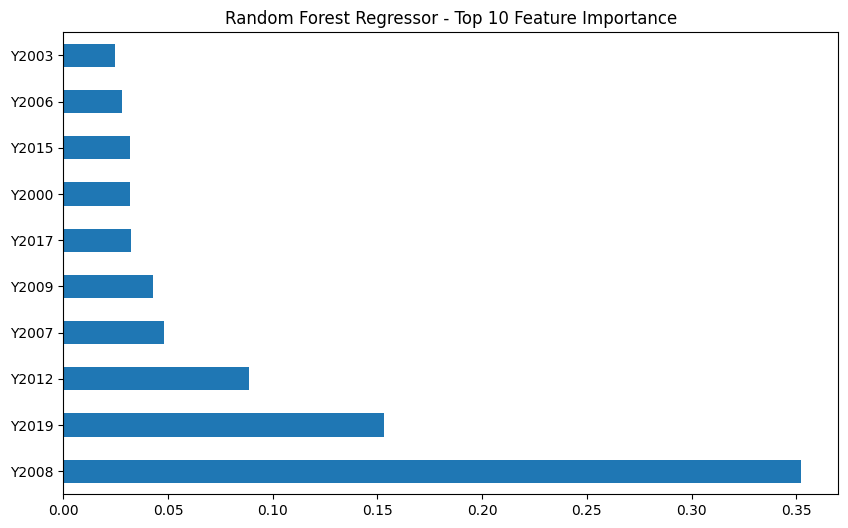




**Random Forest Regressor and Feature Importance**

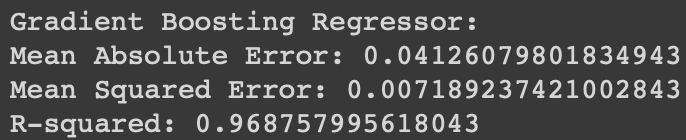
Trained a Random Forest Regressor, evaluated its performance, and visualized the top 15 feature importances.





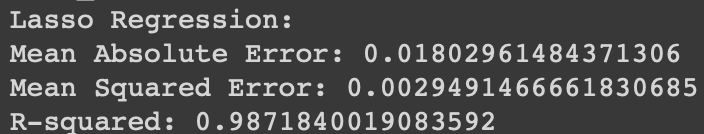
**Gradient Boosting Regressor**

Implemented a Gradient Boosting Regressor, evaluated its performance, and printed the corresponding metrics.

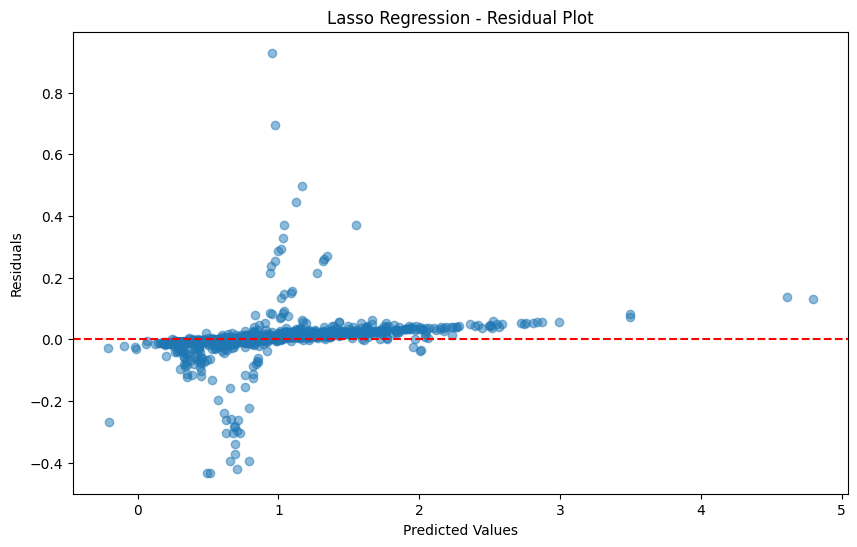


**Lasso Regression**

Implemented a Lasso Regressor, evaluated its performance, and printed the corresponding metrics.

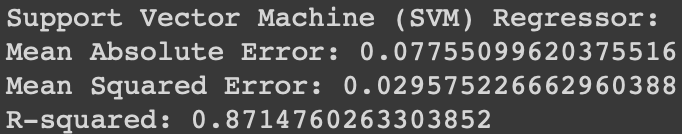


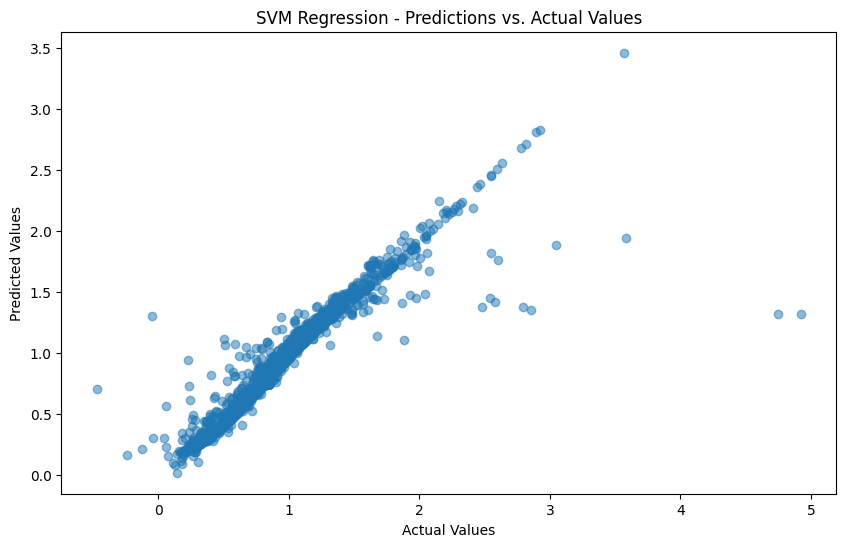


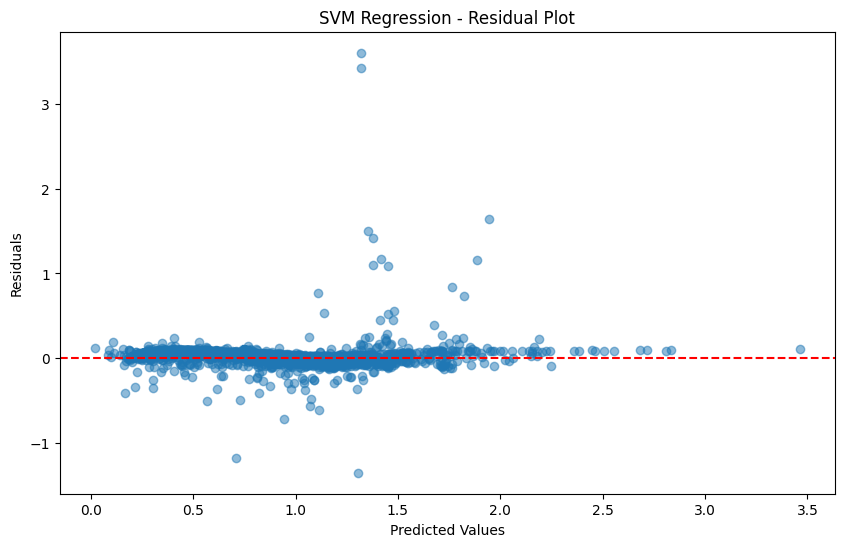


**Support Vector Machine (SVM) Regressor**

Implemented a Support Vector Machine (SVM) Regressor, evaluated its performance, and printed the corresponding metrics.

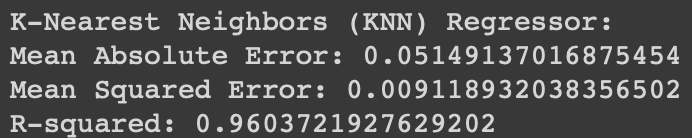


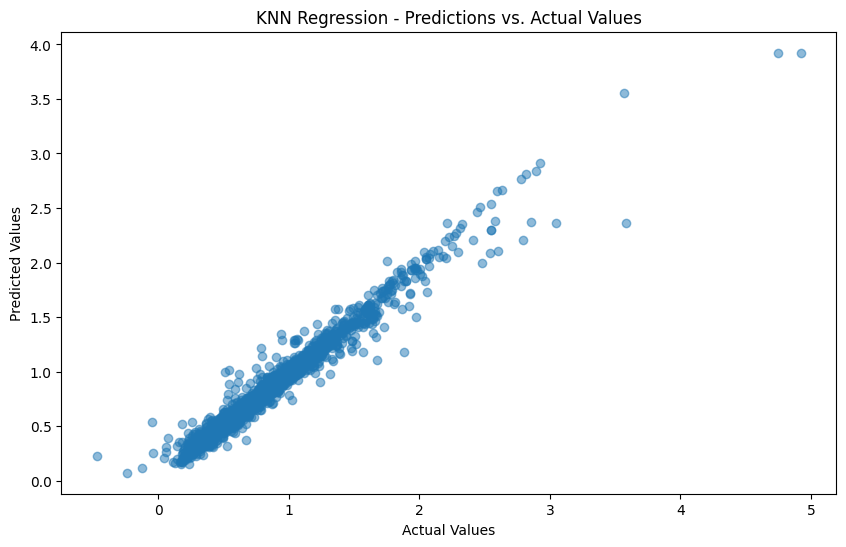


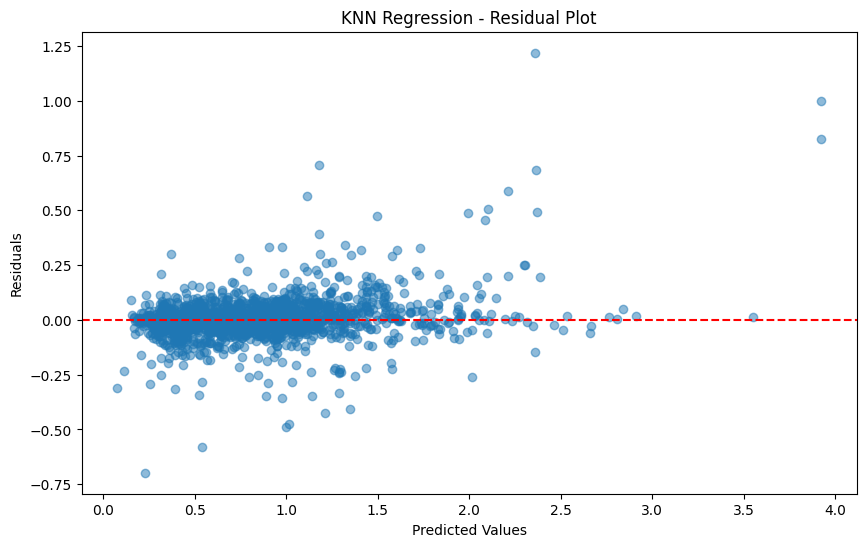


**K-Nearest Neighbors (KNN) Regressor**

Implemented a K-Nearest Neighbors Regressor, evaluated its performance, and printed the corresponding metrics.

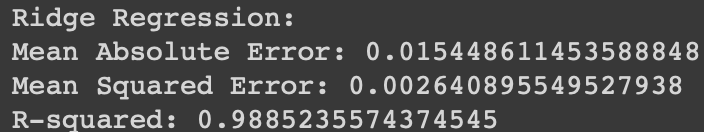


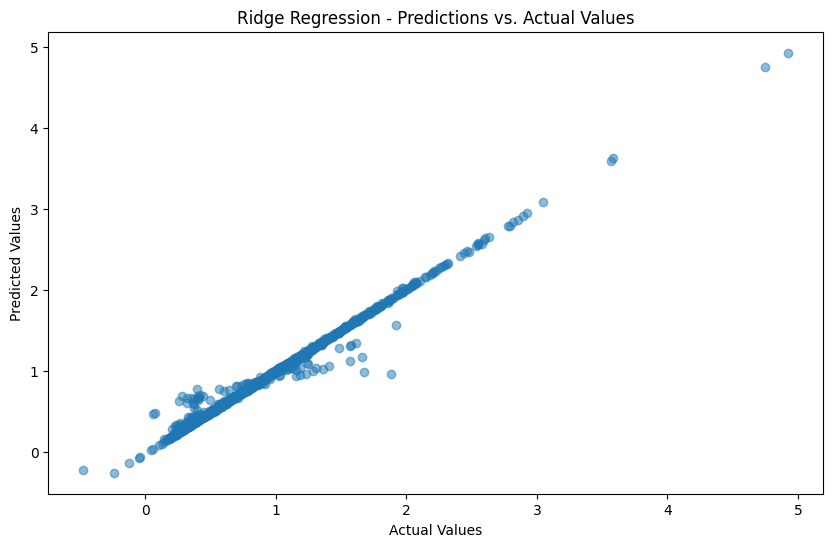




**Ridge Regression**

Implemented a Ridge Regressor, evaluated its performance, and printed the corresponding metrics.

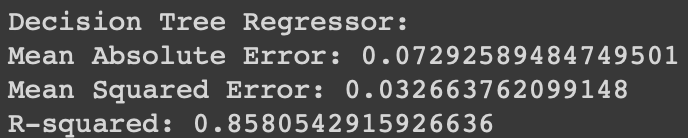


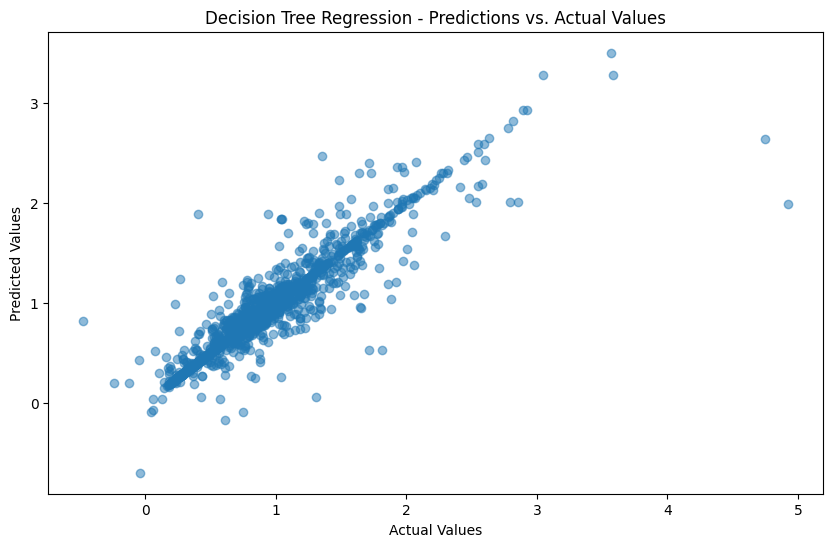


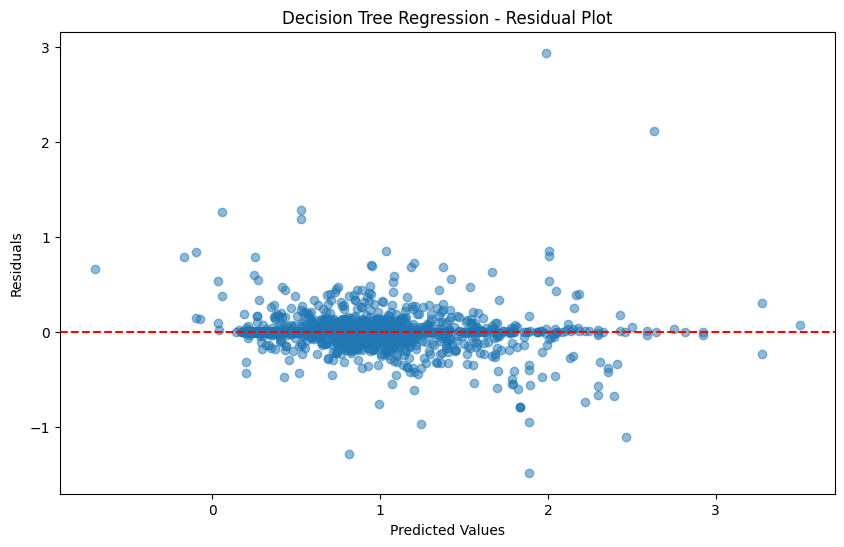


**Decision Tree Regressor**

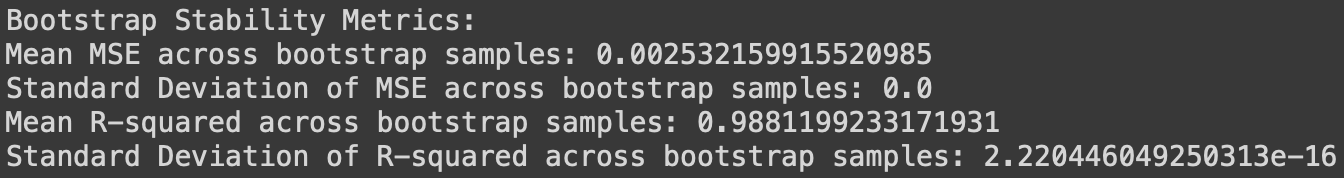
Implemented a Decision Tree Regressor, evaluated its performance, and printed the corresponding metrics.



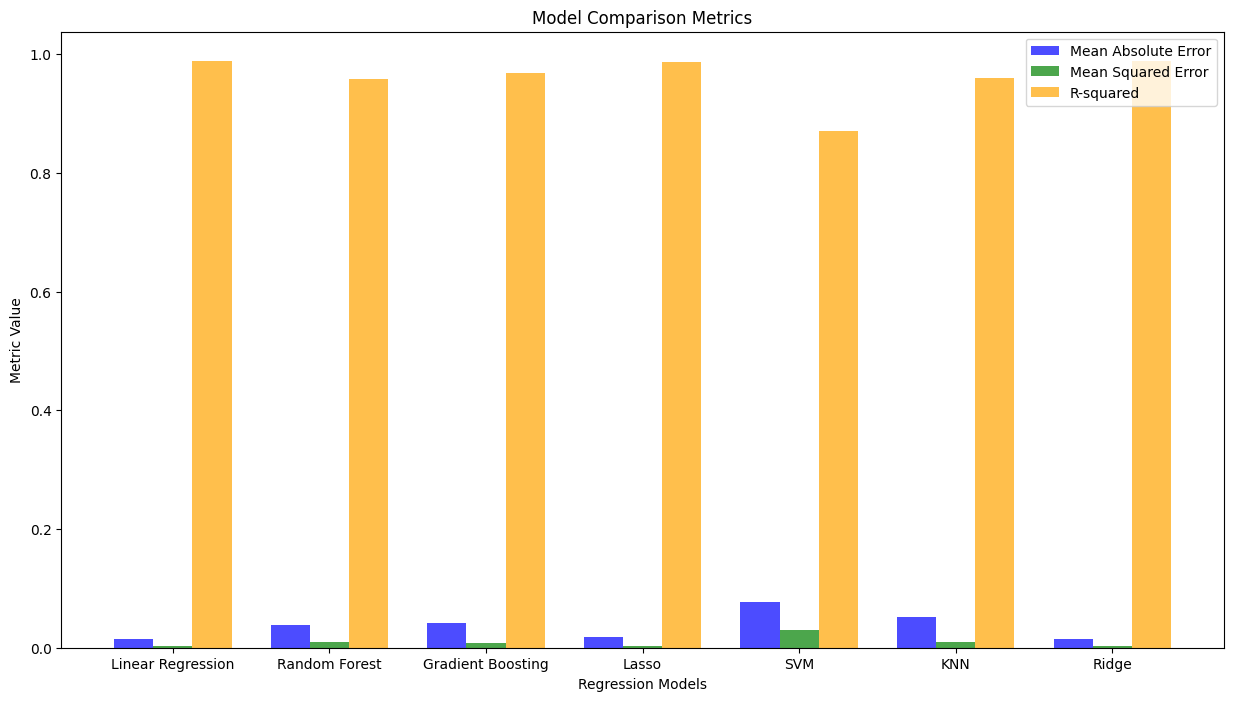




**Bootstrap Stability**



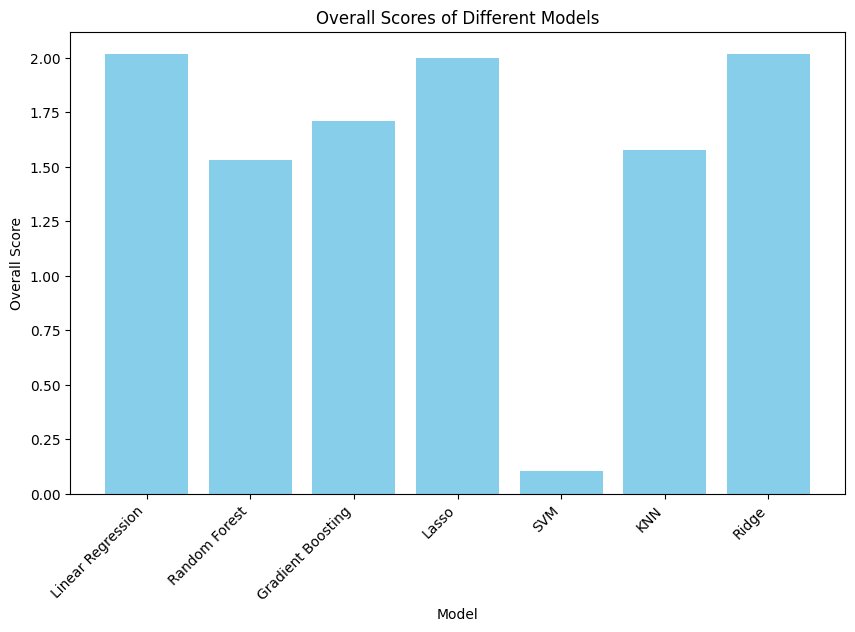
**Model Comparison**



**Conclusion and Visualization**

The final section compiles the results of different regression models into a DataFrame and visualizes the R-squared scores using a bar plot, providing a comprehensive comparison of model performances. The conclusion summarizes the findings from the analysis.





**Approach 2: Y2019 as Target Variable**

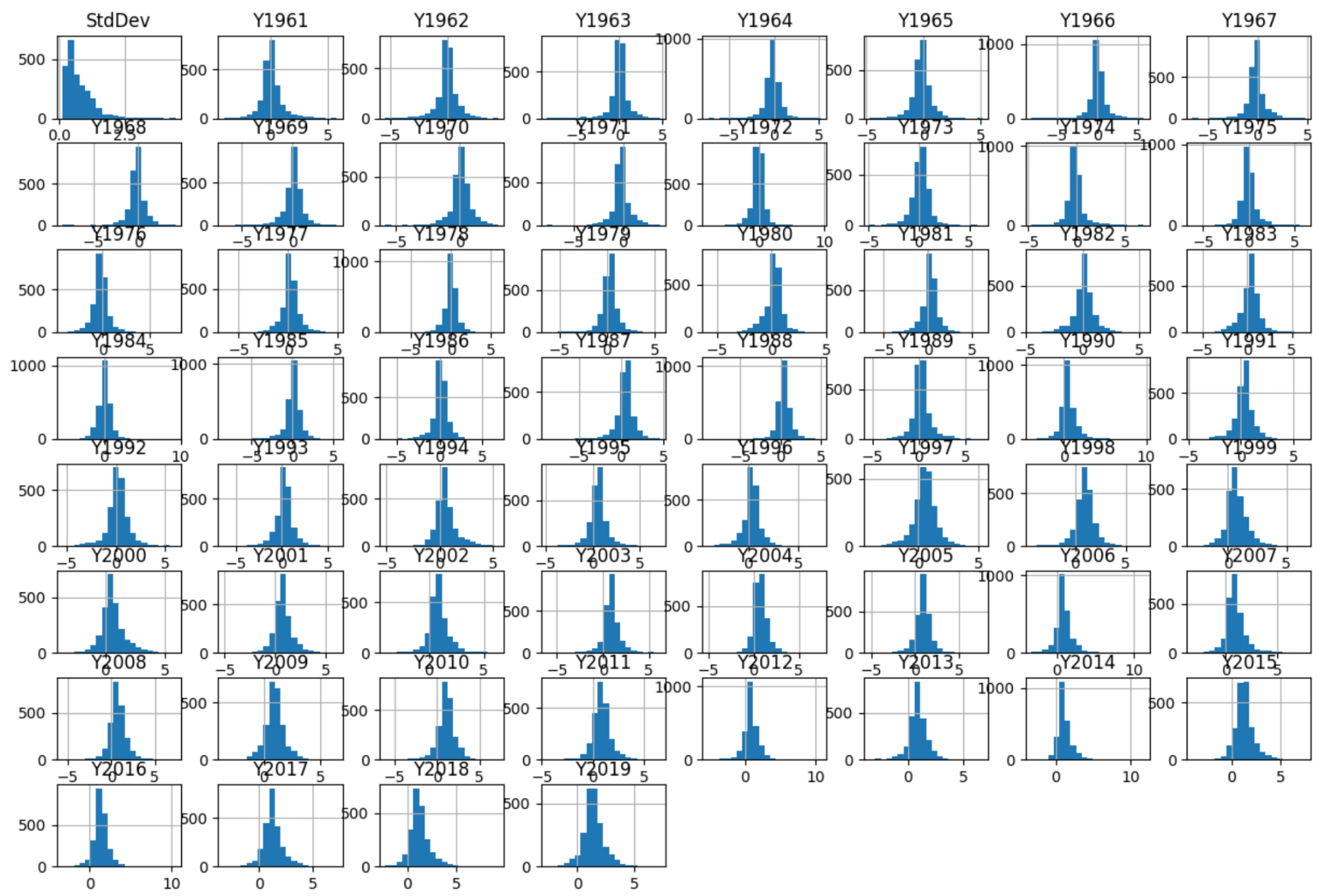
**Data preprocessing**

The dataset was loaded into a DataFrame, and an initial exploration was conducted to understand its structure and contents. Dataset was cleaned by removing similar columns by removing one of those. Area and AreaCode, Months and MonthsCode, and Element and ElementsCode are such similar columns. Removed AreaCode, MonthsCode and ElementsCode columns. Also dropped Unit column as it holds only one value throughout the observations.

When checking the Months column, it’s observed that some of the observations contain a mix of months data. Removed those observations from our dataset. Upon analyzing the Element column, it’s understood that it contains either “Temperature change” or “Standard Deviation” and the value of the year columns from Y1961 to Y2019 is the same when Element type is “Standard Deviation” for the same Area and Months column value. Hence, we have a feature engineered to add a new column StdDev containing a unique Standard Deviation value for a particular Area and Months. Then, drop all observations with Element as “Standard Deviation” and drop the Element column as it contains just one value.

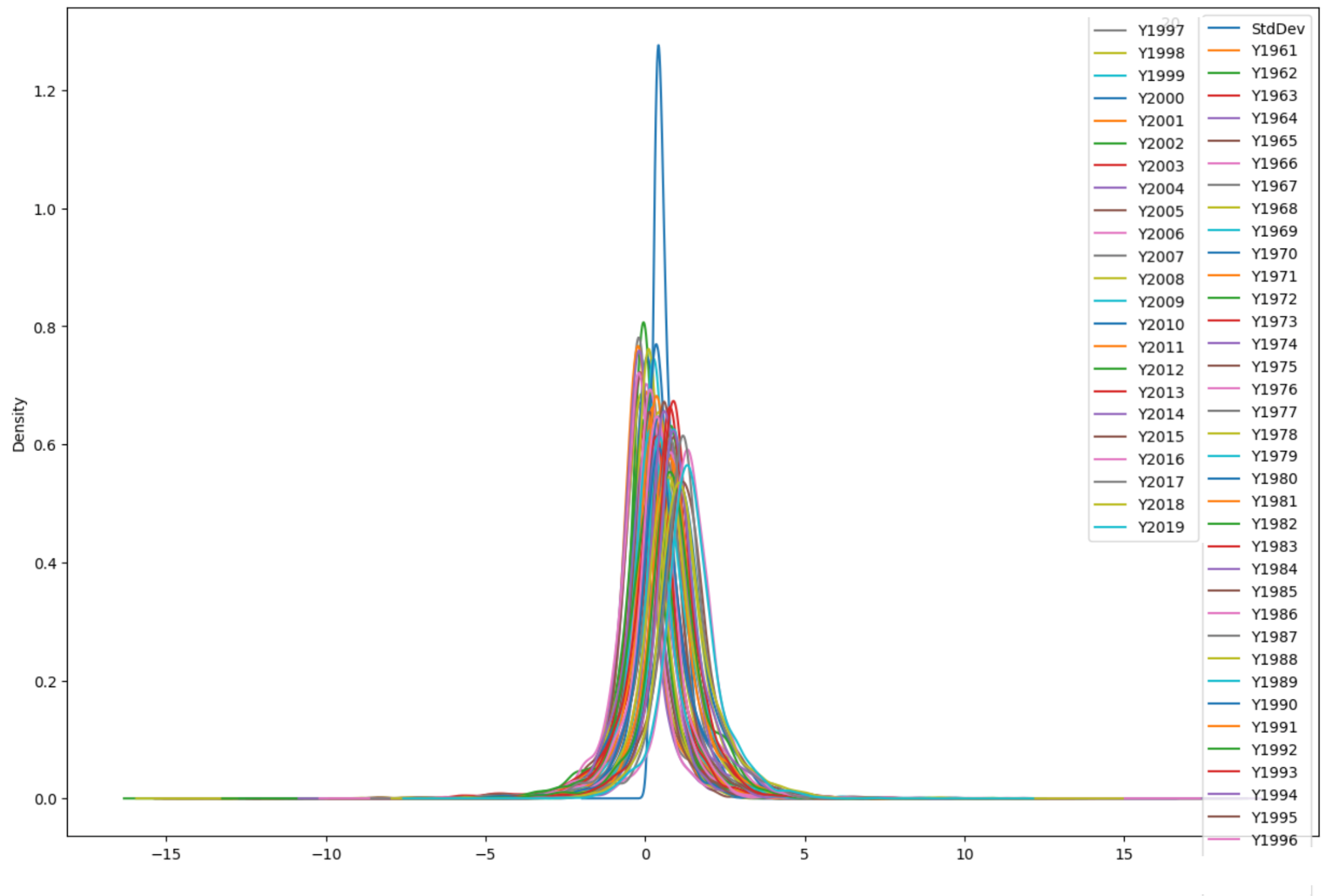
Then, removed all observations having any missing values present.

Distribution of all the remaining features has been verified by plotting histogram.



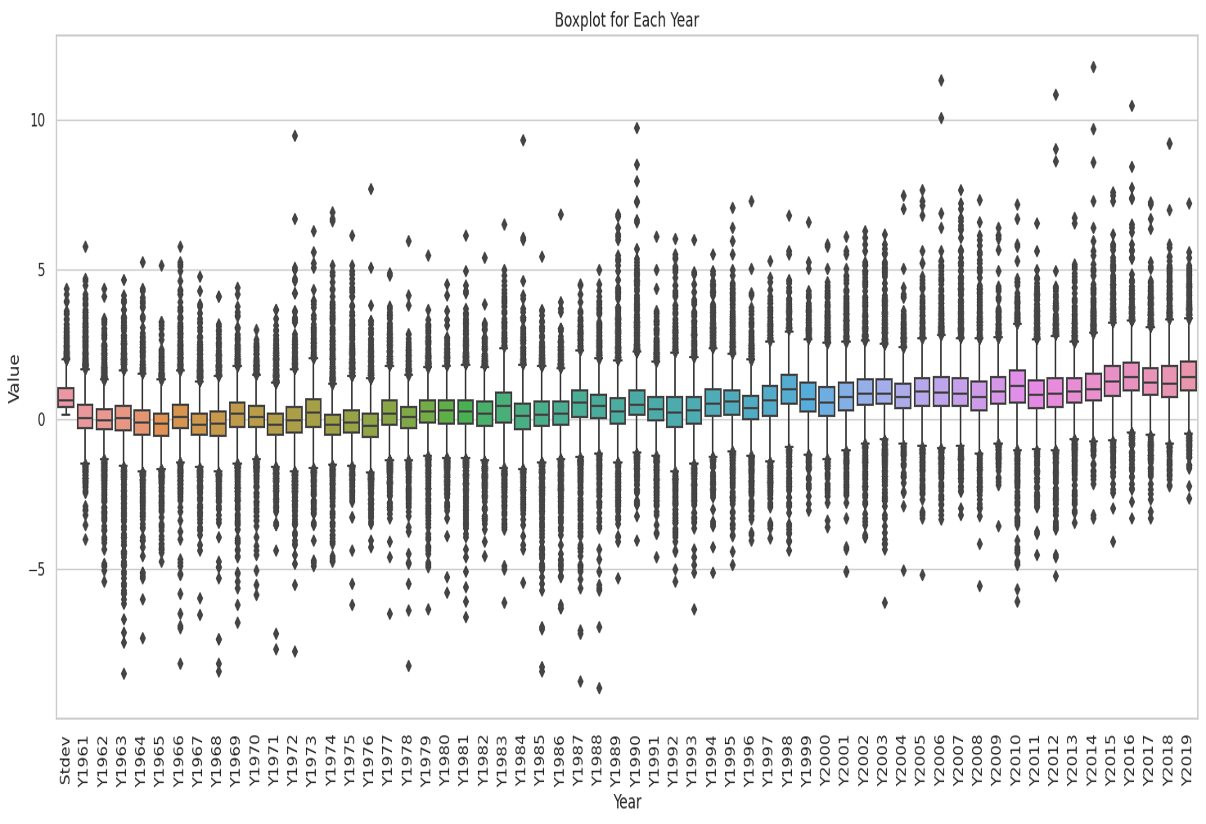
Almost all the features are normally distributed except StdDev, which is slightly right skewed.

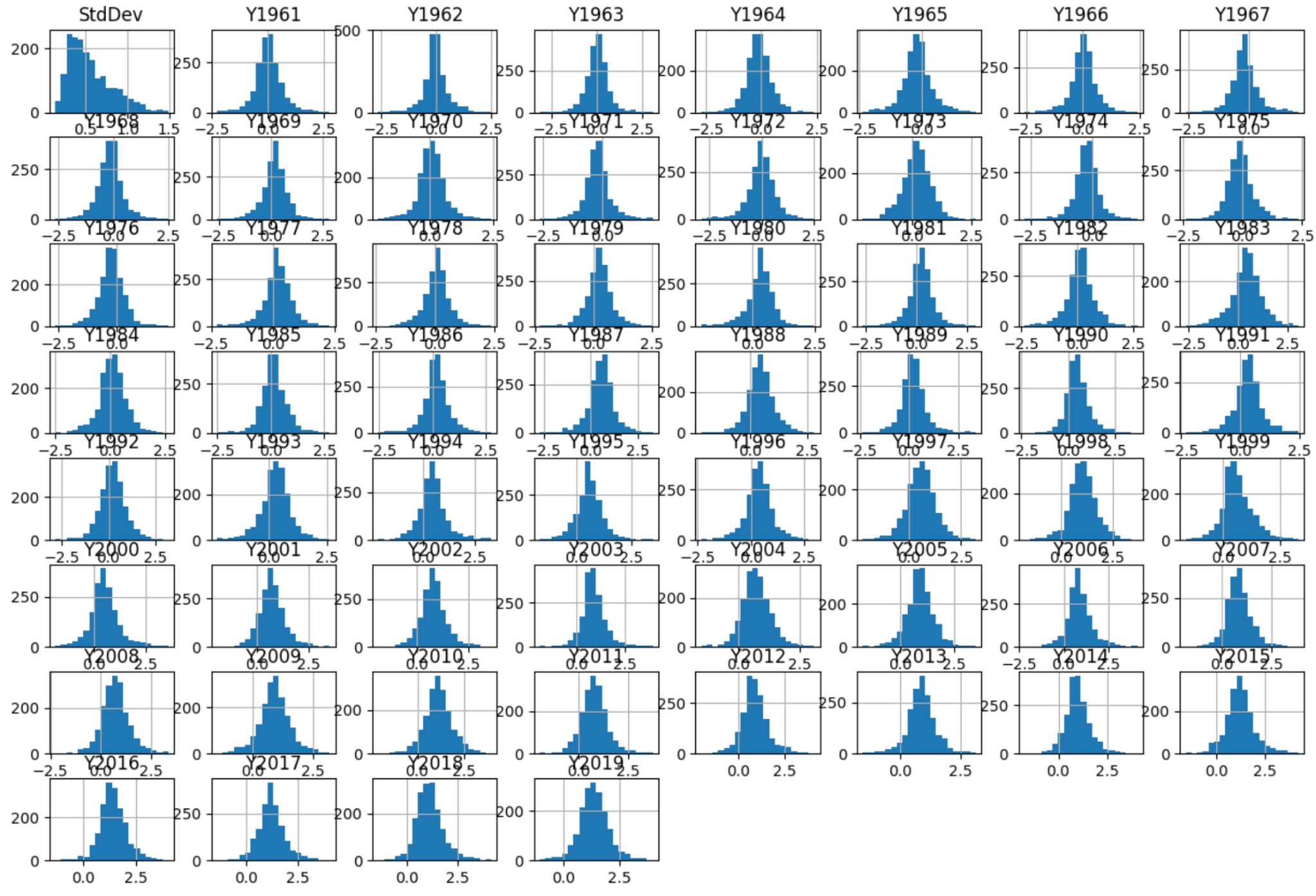
Kernel density plot is also verified for the data.



Kernel density plot shows most of the features are almost normally distributed.

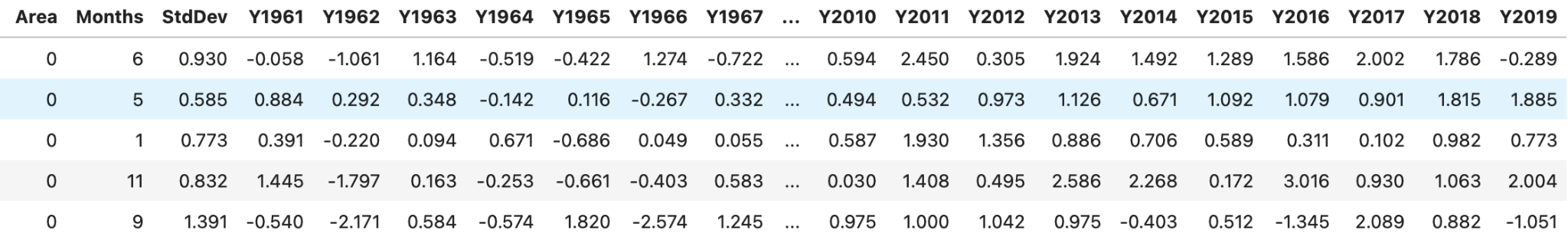
Outliers were analyzed using boxplot. Then removed outliers using the z-score method with absolute threshold as 3.



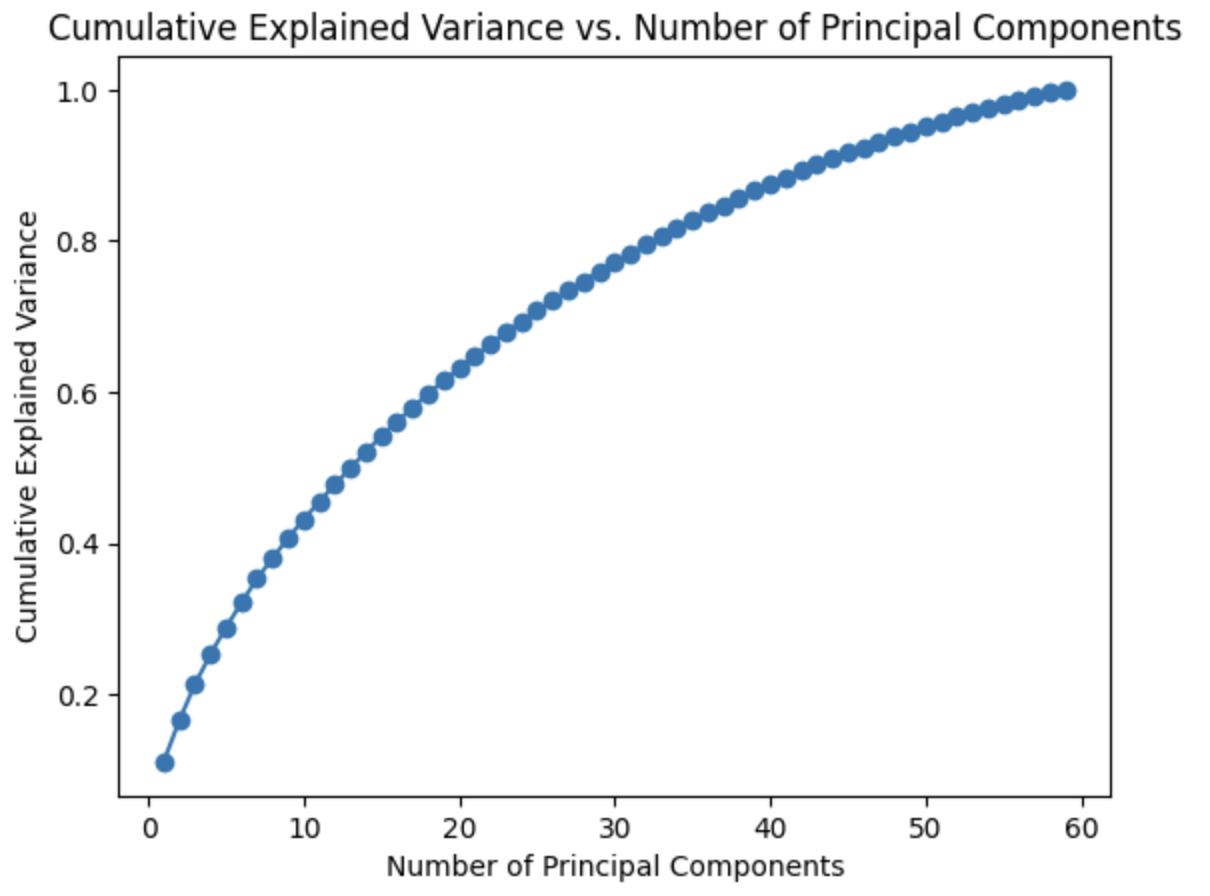


Once outliers are removed, then the skewness of the StdDev feature is reduced.

Label encoded values of categorical variables Area and Months. Now, the final preprocessed dataset is having 1830 observations and 62 features including our target variable Y2019.



As the number of features are high, we have utilized PCA to perform dimensionality reduction.



First 50 principal components explained 95% cumulative variance. Now the final dataset is split into 80% training dataset and 20% test dataset for further modeling.

**Model selection**

We emphasized the type of our problem where we planned to predict the temperature change for 2019 for model selection. In the dataset, the number of predictors is significant after pre-processing(i.e., a total of 50 no. of PC and 2 columns for month and country respectively). Also, there was a slight positive trend found in the avg. temp. change across the year. Considering these points, We selected and applied various linear and non-linear methods with regularisation terms and evaluated their performance on train set( 80% of the total dataset), utilizing MAE(Mean Absolute Error), RMSE(Root Mean Sq. Error), MSE(Mean Sq. Error) and MAPE(Mean Absolute Percentage Error).

The selected models are Ridge, LinearSVM, Decision Tree, Random Forest, XG Boost, and Adaboost. For all the methods we used grid search utilizing 5-fold cross-validation.

**Ridge Regression**:- Ridge Regression introduces an L2 regularization term that penalizes large coefficients, thereby preventing overfitting and improving the model's generalization to new data. To identify the optimal hyperparameter, we first did a random search to find the value closest to the optimal value. The alpha was found to be 233. Then we performed a local search(from 233-234) utilizing Grid Search Cross-Validation. The cross-validation was conducted using a 5-fold strategy. The optimal alpha value was found to be 233.5.

**Linear SVM** - The following hyperparameters were considered for SVM: C representing the regularization parameter, epsilon-"insensitive loss", loss function (squared epsilon insensitive), the maximum number of iterations(1000), and tolerance for stopping the iterative optimization process(.01). The grid search with the incorporation of 5-fold cross-validation was used to find optimal value in the range of values of C(regularization parameter) from .01 to 10, epsilon taking values from 0.01 to 1.

**Decision Tree** - we defined the hyperparameter grid encompassing various configurations for the Decision Tree, including the splitting criterion, maximum depth, minimum samples required to split an internal node, and the minimum samples needed for a leaf node. For the splitting criterion, we considered 'Friedman MSE', 'Squared Error', and 'Absolute Error'. The maximum depth of the decision tree was varied over values 2 to 5. Additionally, the minimum number of samples required to split an internal node and the minimum number of samples required to be at a leaf node were tuned, taking values from 2 to 5.

The identified optimal configuration for the Decision Tree Regressor consisted of the splitting criterion 'Absolute Error', a maximum depth of 2, a minimum number of samples required to split an internal node of 2, and a minimum number of samples required to be at a leaf node of 1. These hyperparameters collectively represent the most effective combination for enhancing the model's predictive performance on the provided training data.

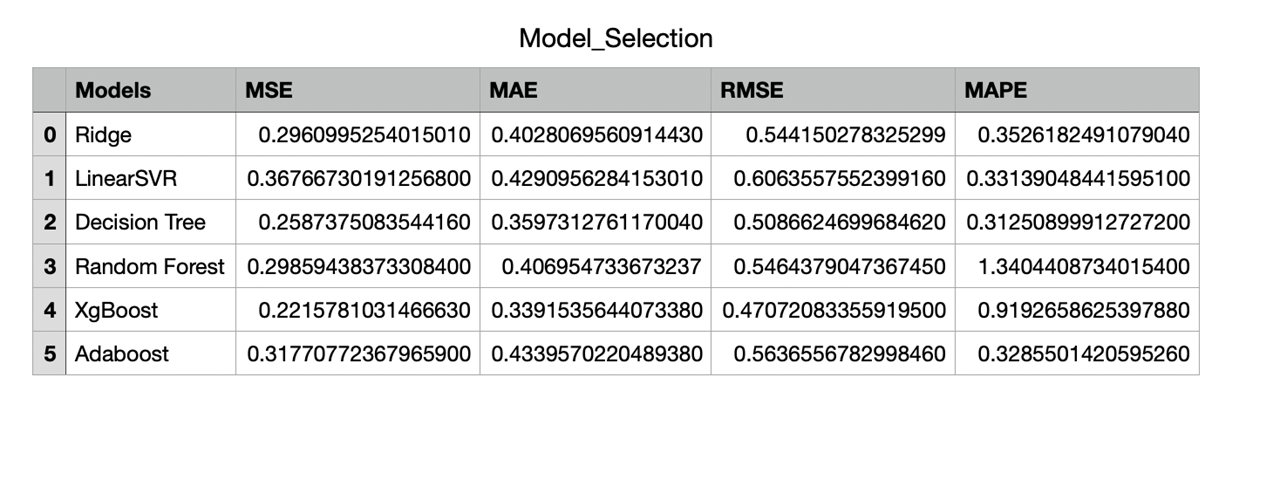
**Random forest**- Here the hyperparameter is no. of Trees. We were able to find out it to be 170 to be the optimal value.

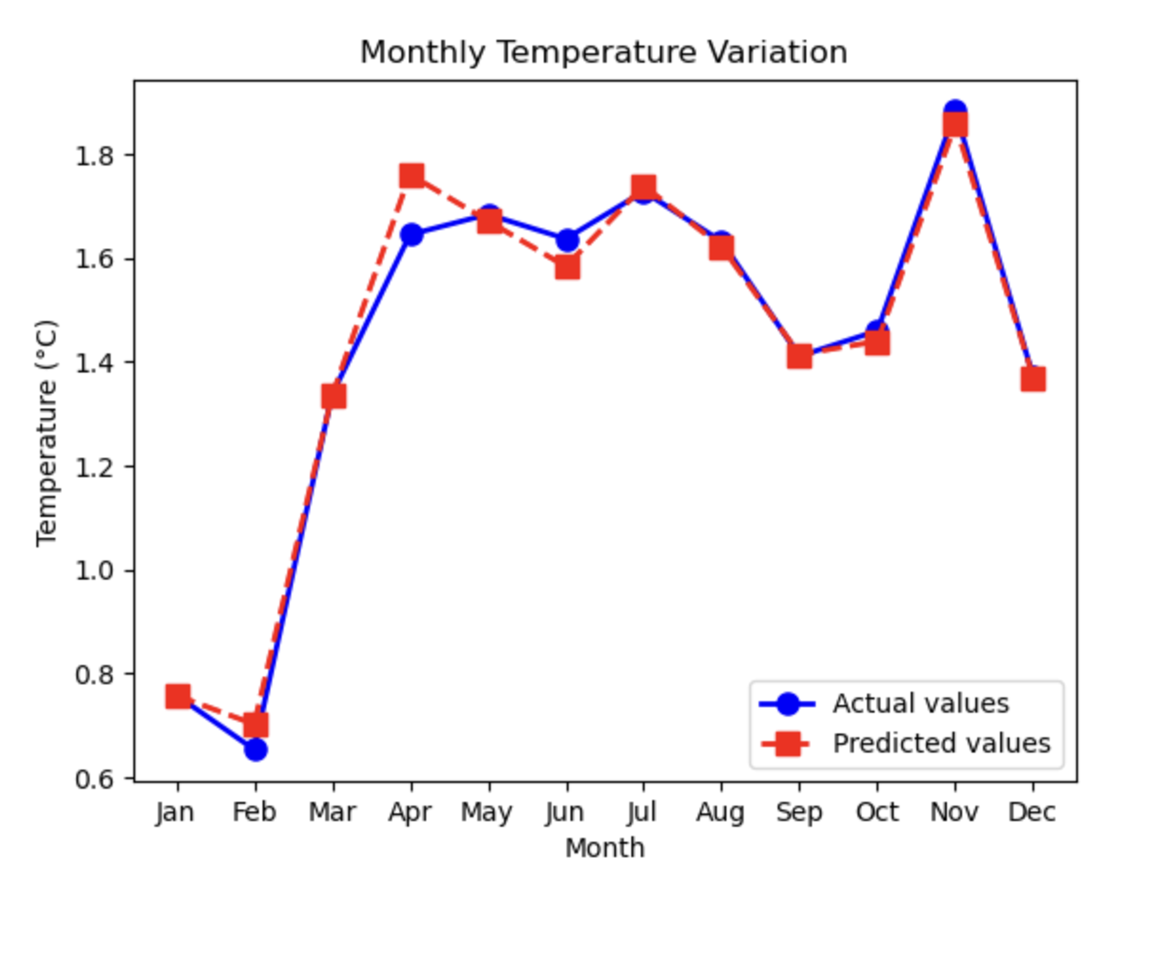
**XGBoost** - the process of tuning hyperparameters for the XGBoost Regressor, a systematic search was conducted across a defined hyperparameter grid. The grid included variations in the number of estimators, learning rate, L1 regularization term, and L2 regularization term. Specifically, the search explored different values for the number of estimators, such as 250, 270, and 300, and considered learning rates of 0.05, 0.09, 0.1, and 0.2. Additionally, the grid encompassed values for L1 regularization in the range of 0.01 to 1 and L2 regularization with values ranging from 0.1 to 2. The best hyperparameter was found to be, learning rate-0.05, no. of estimator- 300, alpha - 0.1, and lambda - 1.

**Adaboost** - The grid encompassed variations in the number of estimators (185, 195, 200, 250), learning rates (0.01, 0.05, 0.09, 0.1), and three potential loss functions: linear, square, and exponential. The grid search, conducted with 5-fold cross-validation, systematically evaluated the model's performance across these hyperparameter combinations.The optimal parameter was found to be 250 no. of estimators, .01- learning rate, and loss function "Square".

**Model Performance Evaluation:**

Among the models evaluated, XGBoost demonstrated the lowest MSE, indicating superior accuracy in predicting target values. It also exhibited the lowest MAEand RMSE, further confirming its robust predictive capabilities. Random Forest and Decision Tree models also performed reasonably well, outperforming Ridge and Linear Support Vector Regression (LinearSVR) in most metrics. AdaBoost, while offering competitive results, showed slightly higher errors compared to XGBoost, placing it in the middle of the ranking.

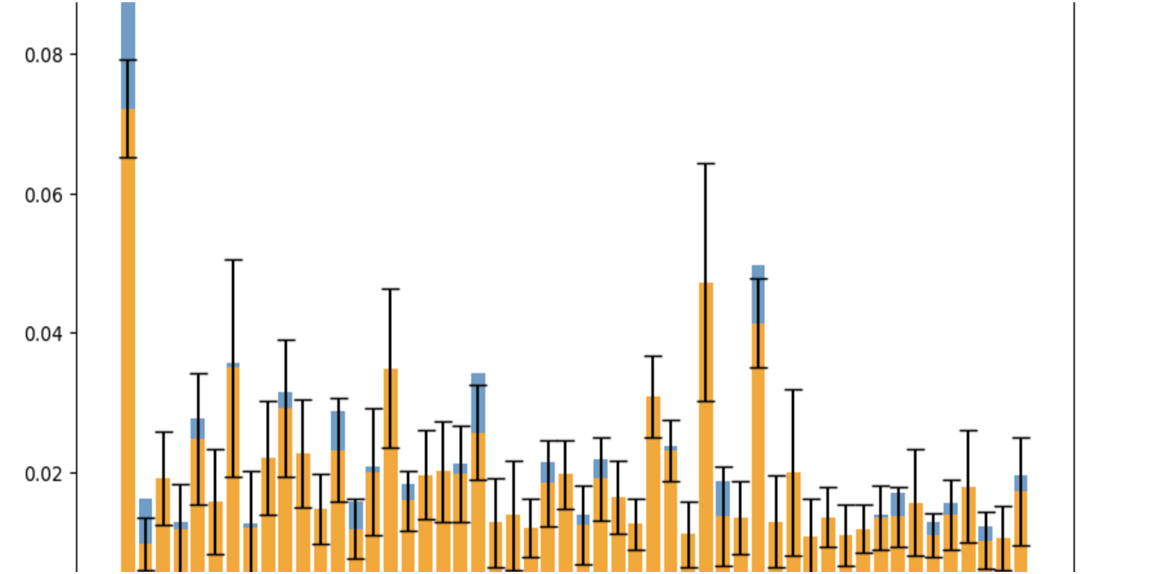
However, it is reasonable to notice that for MAPE, the value of ridge is the lowest and for random forest it has the highest value. This abnormality could occur if the model is overfitting the training set or due to improper range in which the grid search was performed for the hyperparameters.

**Final Model**:

The graph depicting temperature variations for all twelve months in the year 2019 in Albania reveals the impressive accuracy of our predictive model in forecasting these changes. The blue line, representing actual temperature values, harmoniously aligns with the red dashed line, which signifies our model's predictions. This close match between observed and predicted temperatures serves as a testament to the efficacy and reliability of our forecasting model.

**Checking Model Stability** :

The plot below illustrates the stability of feature importance scores in the model, offering valuable insights into the impact of different features on XGBoost model predictions. Blue bars represent feature importance scores derived from the model trained on the original dataset, where higher bars suggest a greater influence on predictions. The introduction of orange bars, depicting mean feature importance scores from bootstrapped samples, provides a measure of stability and robustness against sampling variability. The tight clustering of error bars around the orange bars signifies consistent feature importance across various bootstrap iterations, reinforcing the reliability of the model. This stability not only aids in understanding the relative importance of features but also enhances the interpretability of the model's outcomes. The alignment between blue and orange bars, coupled with minimal variability in the error bars, underscores the model's trustworthiness and highlights the reliability of identified important features for informed decision-making.



**Conclusion:**

This project offers a comprehensive analysis of air temperature differences in the year 2019, leveraging a rich dataset spanning nearly 58 years. The insights gained from this study contribute to our understanding of climate dynamics, informing discussions on environmental changes and facilitating informed decision-making for sustainable practices. Throughout our analysis, we employed a diverse set of regression models, including Ridge regression, Linear Support Vector Regression (SVR), and ensemble methods, to scrutinize temperature differences in the year 2019 based on data spanning 1961 to 2018. Despite the marginal distinctions observed among the results of these models, it is noteworthy that XGBoost emerged as the standout performer. The superiority of XGBoost underscores its efficacy in capturing the complex relationships inherent in the temperature data, demonstrating its adaptability to the intricacies of the dataset. This outcome highlights the importance of leveraging advanced machine learning techniques, such as XGBoost, in enhancing the accuracy and predictive power of climate-related analyses, even when the differences between models might appear subtle at first glance.

**Discussion:**

In light of the observed minimal distinctions among various models, it is reasonable to speculate that the limited variability in their predictions may stem from significant collinearity between features. The presence of high collinearity suggests that multiple predictors in the models are exhibiting strong linear relationships, potentially leading to redundant or overlapping information. In response to this challenge, a strategic approach would involve considering time-dependent models, such as autoregressive models, where predictions are influenced by the immediate past values of the variable of interest. By incorporating temporal dependencies, autoregressive models can capture the sequential patterns and inherent structures within the data, providing a more nuanced understanding of the dynamics at play. This shift towards time-dependent modeling may unravel hidden trends and contribute to improved predictive accuracy, particularly in scenarios where the traditional models exhibit limited discriminative power due to collinearity issues. Additionally, there is one more thing to be considered is that for ensemble learning we do not need to remove outliers. Ensemble models possess a built-in capability to handle outliers on their own. This means we can skip the step of removing outliers before using ensemble methods, making the process simpler and straightforward. Therefore, these are the two things that might be improved upon further.

**Team Contribution**

**Nithyashree Govindarajan -**

Worked on approach 1 as follows:

* Initial data visualization
* Correlation analysis
* Model training:
  + Linear Regression
  + Random Forest
  + Gradient Boosting
  + Lasso Regression
  + Ridge Regression
  + Support Vector Machine (SVM)
  + K-Nearest Neighbors (KNN)
  + Decision Tree Regressor
* Bootstrap stability
* Model comparison

**Aseem Salim -**

* Performed pre-processing steps like redundant variable removal like AreaCode, ElementCode and MonthsCode.
* Removed non relevant feature Unit based on number of unique values.
* Removed rows with month combinations from dataset for Months column.
* Identified new feature Standard deviation from observations and converted that to a column to original dataset.
* Visually verified numerical data distribution using histogram and Kernel Density Estimate plots.
* Visually identified the outliers using box plots and removed outliers using the z-score method.
* Rechecked the data distribution after outlier removal.
* Brainstormed on which target variable to be used. Suggested about using average temperature change of years or take an year.
* Plotted pair plot to see for correlation.
* Performed PCA for approach 2 (target Y2019) and plotted the variance explanation graph. Identified the optimal PCs.
* Helped Anushka in plotting the decision tree of XGBoost approach for Approach 2. But it was not giving proper results.

**Anushka Tiwari -**

* Modeling part for approach 2
* Hyperparameter tuning.
  + Ridge
  + LinearSVM
  + Decision Tree
  + Random Forest
  + Xgboost
  + Adaboost
* Visualization of actual vs predicted values.
* Metric analysis
* Bootstrap sampling.
* Visualization of feature importance stability.
* Tried applying an auto regressive model.

**Awnish Shankar -**

* Performed data preprocessing for approach 1 and 2.
  + Data introduction,identifying and removing missing values,outliers detections.
* Did data exploratory analysis for approach 2.
  + Performed analysis for using barplot KDE plot,Corrplot and Boxplot.