Detecting Climate Change Patterns By Machine Learning.

Introduction

In this project, we delve into a comprehensive analysis of climate change by firstly focusing on the world's seven largest cities: Bombay, Moscow, New York, Shanghai, Sydney, São Paulo, and Toronto. encompassing data from 1970 to 2013, We performed, Time Series Analysis and Regression Modeling To predict and understand the future climate trajectory for the largest cities, we also employ two powerful regression models, the Random Forest Regressor and Gradient Boosting Regression (GBR). Finally Clustering Analysis in order to to identify broader climate trends and regional patterns we employ K-means and MiniBatch K-means clustering methods and , we group geographical regions exhibiting similar temperature behavior.

Climate Change

Climate change refers to long-term alterations in Earth's climate patterns, including temperature, precipitation, and weather events, primarily driven by the increase in greenhouse gas concentrations in the atmosphere, such as carbon dioxide from human

activities like burning fossil fuels and deforestation. These changes result in a range of environmental consequences, such as rising global temperatures, more frequent and severe weather events.

Earth Surface Temperature Data Set

The dataset hosted on Kaggle titled "Climate Change: Earth Surface Temperature Data" is a comprehensive and widely used resource for analyzing historical temperature trends on our planet. This dataset is sourced from the Berkeley Earth Surface Temperature project, which has meticulously compiled temperature records from various sources, including weather stations, ocean buoys, and satellite observations, to create a robust and reliable global temperature dataset. It spans from the mid-18th century to the present day, making it one of the most extensive records of Earth's surface temperature available for research and analysis.

	dt	AverageTemperature	AverageTemperatureUncertainty	City	Country	Latitude	Longitude
0	1743-11-01	6.068	1.737	Århus	Denmark	57.05N	10.33E
1	1743-12-01	NaN	NaN	Århus	Denmark	57.05N	10.33E
2	1744-01-01	NaN	NaN	Århus	Denmark	57.05N	10.33E
3	1744-02-01	NaN	NaN	Århus	Denmark	57.05N	10.33E
4	1744-03-01	NaN	NaN	Århus	Denmark	57.05N	10.33E

The "Climate Change: Earth Surface Temperature Data" dataset offers a rich array of information through its columns. It includes columns such as "dt" (date), "LandAverageTemperature," "LandAverageTemperatureUncertainty," "LandMaxTemperature," "LandMaxTemperatureUncertainty," "LandMinTemperature," "LandMinTemperatureUncertainty," "LandAndOceanAverageTemperature," "LandAndOceanAverageTemperatureUncertainty," and "Country." The "dt" column provides the date in a YYYY-MM-DD format, allowing for precise temporal analysis. Temperature data is presented in both average and extreme values, accompanied by their respective uncertainties, which are crucial for assessing data reliability.

In addition to the temperature-related columns, the dataset also contains a "Country" column, indicating the country or region associated with each temperature measurement. This geographical granularity allows for country-specific climate analysis and comparisons.

Preprocessing

Preprocessing is a crucial part of any machine-learning project, This project performs several

Preprocessing steps Firstly, the dataset is loaded using the Pandas library, and basic statistics about the dataset's structure are examined. By calculating the unique values in the "Country" and "City" columns (159 countries, 3448 cities), the script identifies the number of distinct countries and cities represented in the dataset. Additionally, the code calculates the total number of records in the dataset, providing an overview of the dataset's size and scope.

Furthermore, the code performs a check for missing values in the dataset using the `isna().sum()` method. Two of the columns have missing values AverageTemperatureUncertainty and AverageTemperature, For handling the missing values the code first creates an instance of the `SimpleImputer` class with the specified imputation strategy, which is 'mean' in this case .The 'mean' strategy calculates the mean (average) value of the non-missing data in each column and uses that mean value to fill in the missing entries, The code then applies the imputer to the selected columns in the Data Frame using `imputer.fit_transform(df[cols_with_missing])`

Feature Engineering

During the Feature Engineering phase two new columns, 'Latitude_Direction' and 'Longitude_Direction,' were created from the 'Latitude' and 'Longitude' columns in the "Global Land Temperatures By City" dataset. 'Latitude_Direction' captures the hemisphere information by extracting the last character ('N' for northern or 'S' for southern), while 'Longitude_Direction' signifies the hemisphere's east-west location ('E' for eastern or 'W' for western) based on the last character of the 'Longitude' column. Simultaneously, the numerical portions of 'Latitude' and 'Longitude' are converted into float values, removing the hemisphere indicators..

	dt	AverageTemperature	AverageTemperatureUncertainty	City	Country	Latitude	Longitude	Latitude_Direction	Longitude_Direction
0	1743-11-01	6.068000	1.737000	Århus	Denmark	57.05	10.33	N	E
1	1743-12-01	16.727433	1.028575	Århus	Denmark	57.05	10.33	N	E
2	1744-01-01	16.727433	1.028575	Århus	Denmark	57.05	10.33	N	E
3	1744-02-01	16.727433	1.028575	Århus	Denmark	57.05	10.33	N	E
4	1744-03-01	16.727433	1.028575	Århus	Denmark	57.05	10.33	N	E

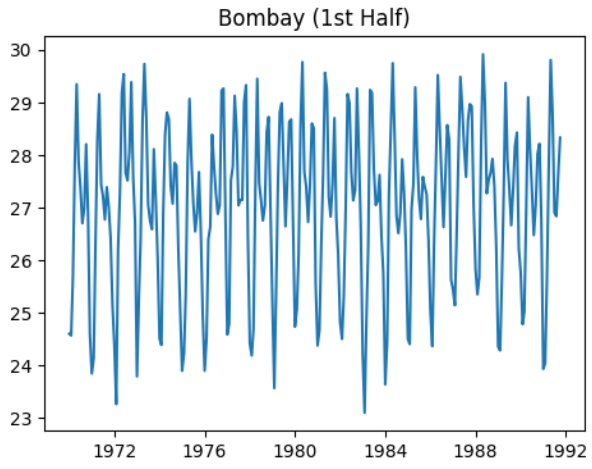
Models

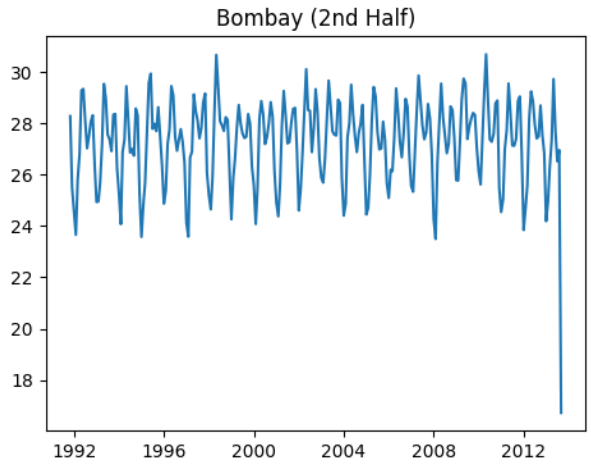
This project has two main parts ,The first part works with df_city which is a subset of the original dataset containing only the 7 largest cities in the world during the last 50 years. and the second part which performs clustering methods on the entire dataset and for the whole timeline of the df dataset.

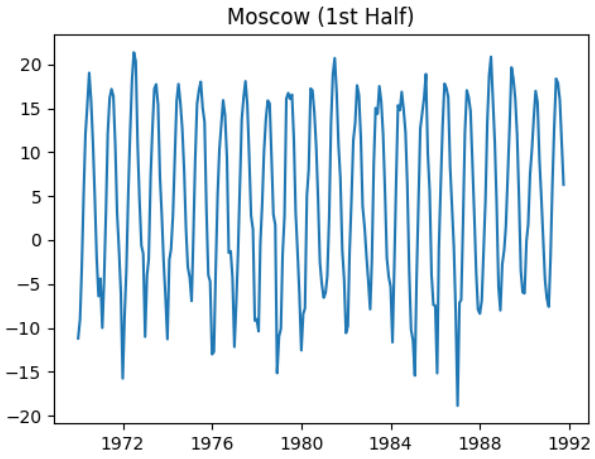
Time Series Analysis on df_city

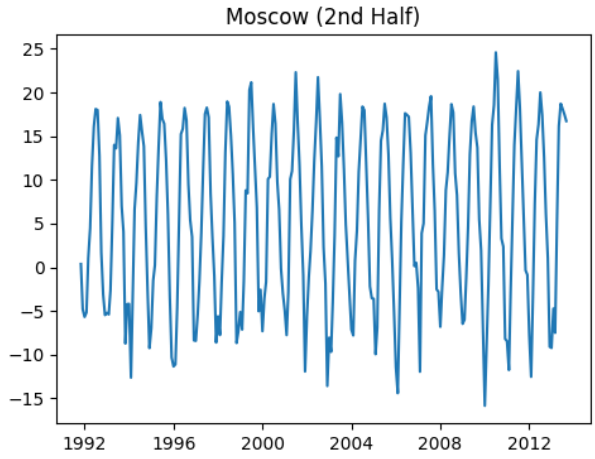
Time series analysis is a statistical technique used to study and model data points collected over time. It involves examining the sequential order of data to identify patterns, trends, and dependencies within the dataset. When applied to the climate change Earth surface temperature dataset , time series analysis can be immensely valuable. This dataset contains historical temperature measurements recorded over time.

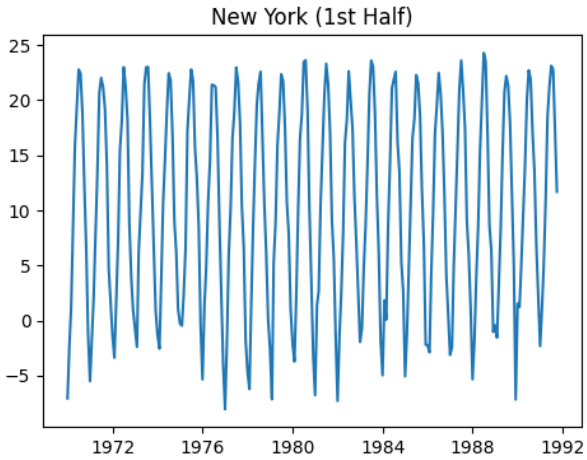
For this project, the focus was specifically on the seven largest cities in the world and a specified time period from 1970 to 2013. This code conducts time series analysis by splitting the temperature data for the seven largest cities into two halves and creating side-by-side plots to visually compare temperature trends within each city over time. The results are shown below.

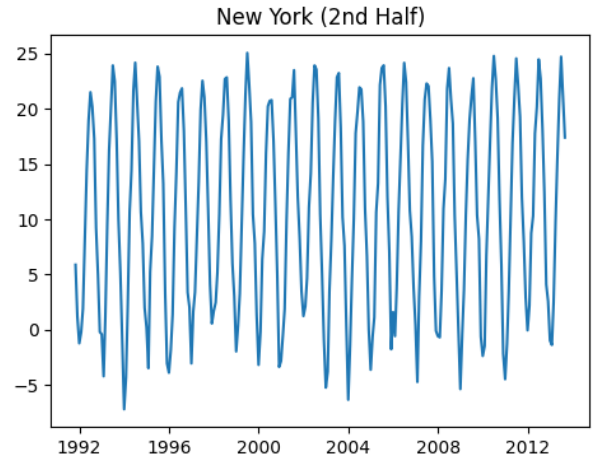


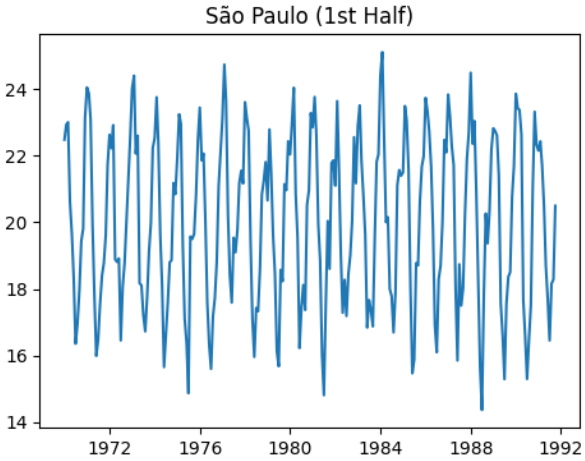


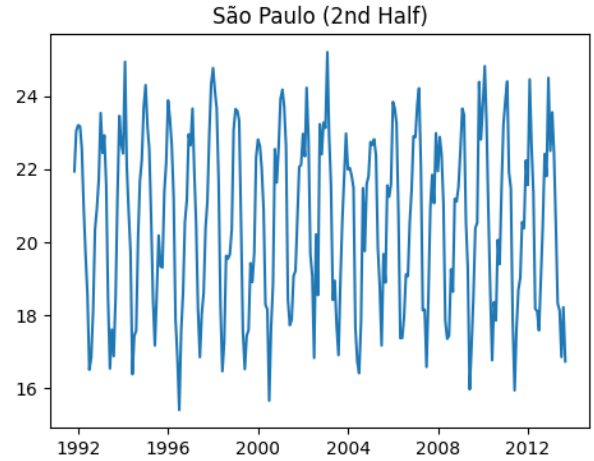

Moscow (1st Half)
20
15
10
5
0
−5
−10
−15
−20
1972
1976
1980
1984
1988
1992

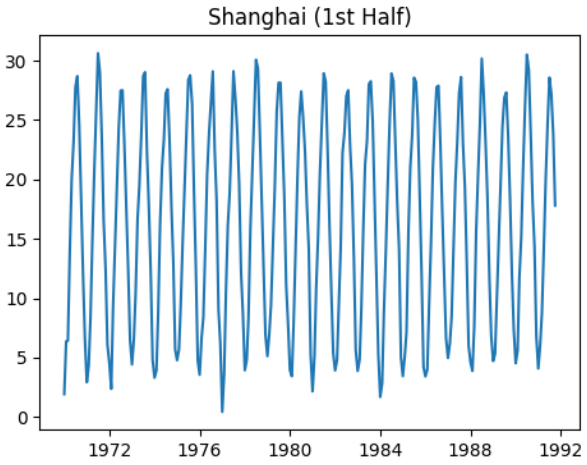

Moscow (2nd Half)
25
20
15
10
5
0
−5
−10
−15
1992
1996
2000
2004
2008
2012

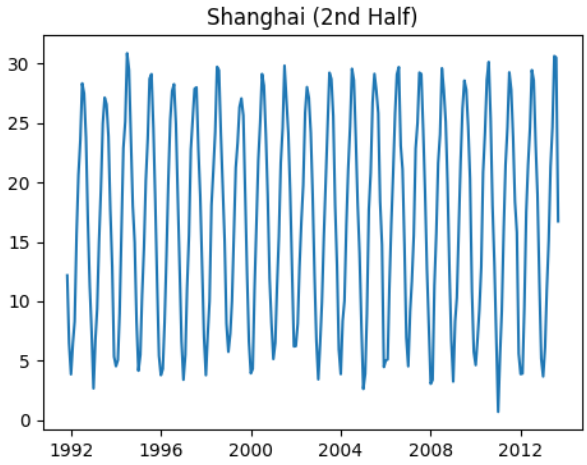

New York (1st Half)
25
20
15
10
5
0
−5
1972
1976
1980
1984
1988
1992

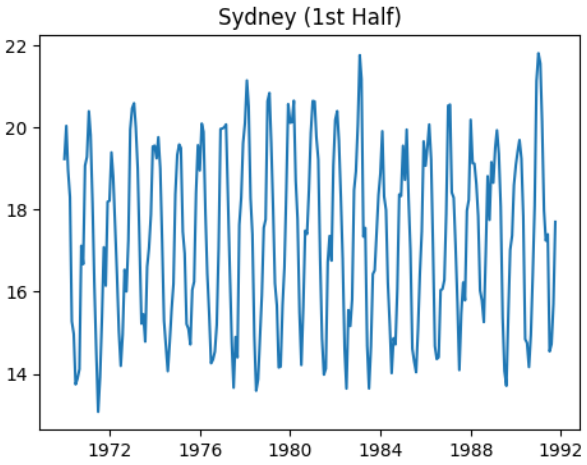

New York (2nd Half)
25
20
15
10
5
0
−5
1992
1996
2000
2004
2008
2012

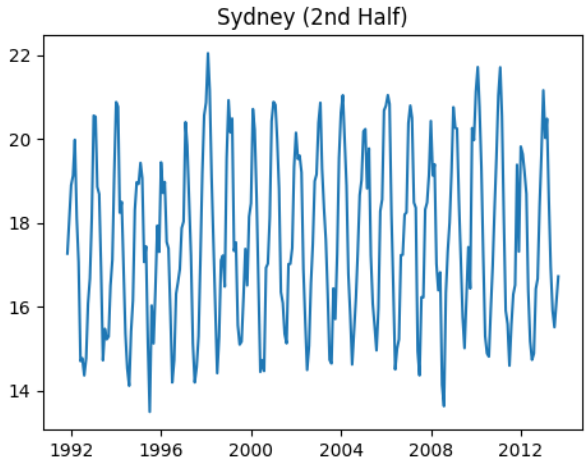

São Paulo (1st Half)
24
22
20
18
16
14
1972
1976
1980
1984
1988
1992

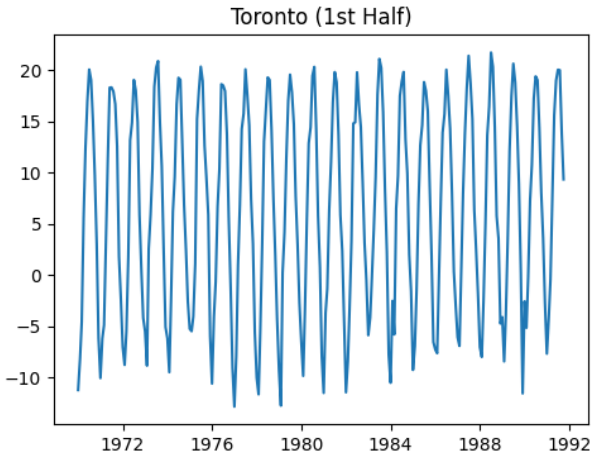

São Paulo (2nd Half)
24
22
20
18
16
1992
1996
2000
2004
2008
2012

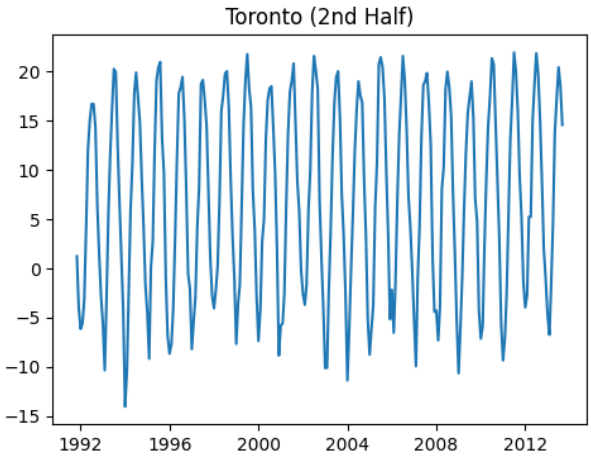








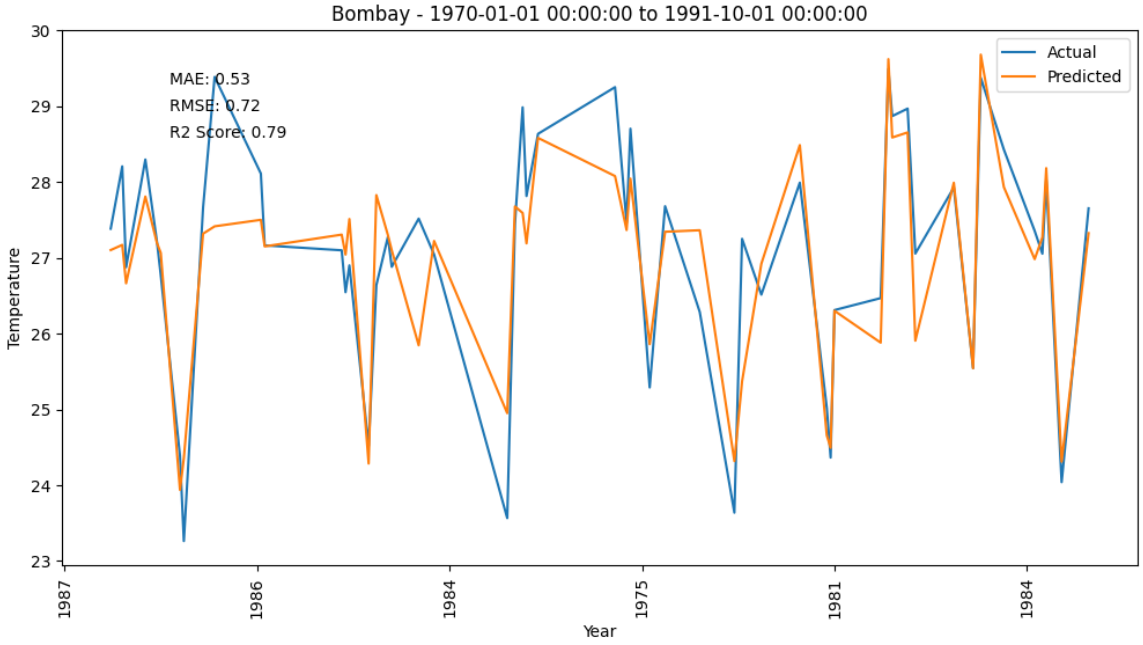


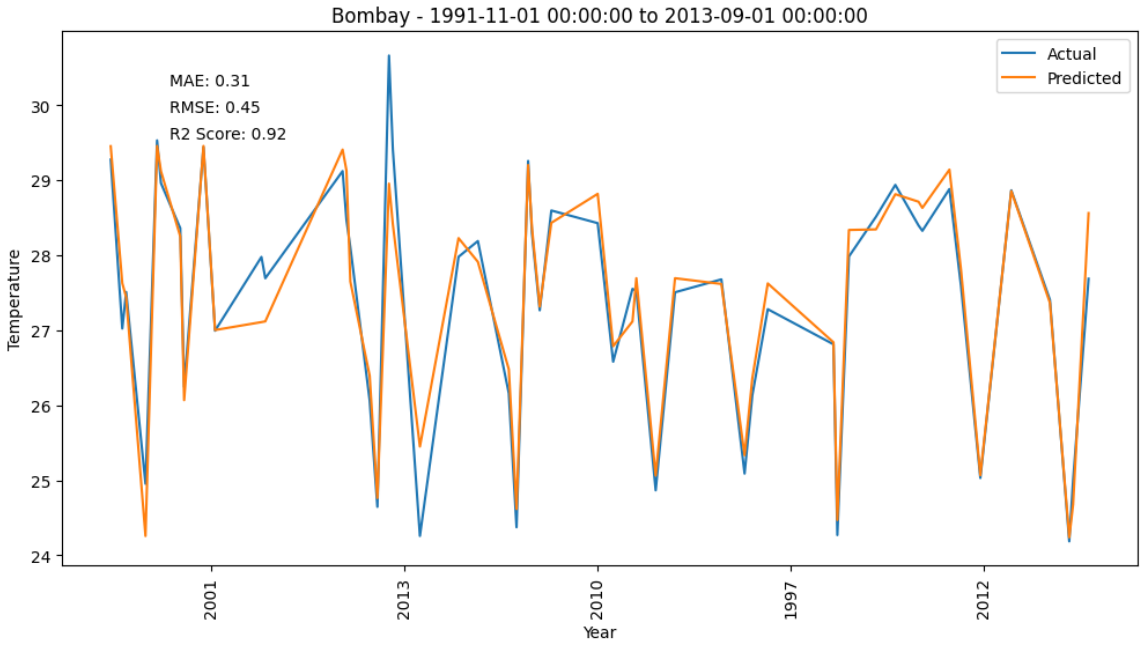


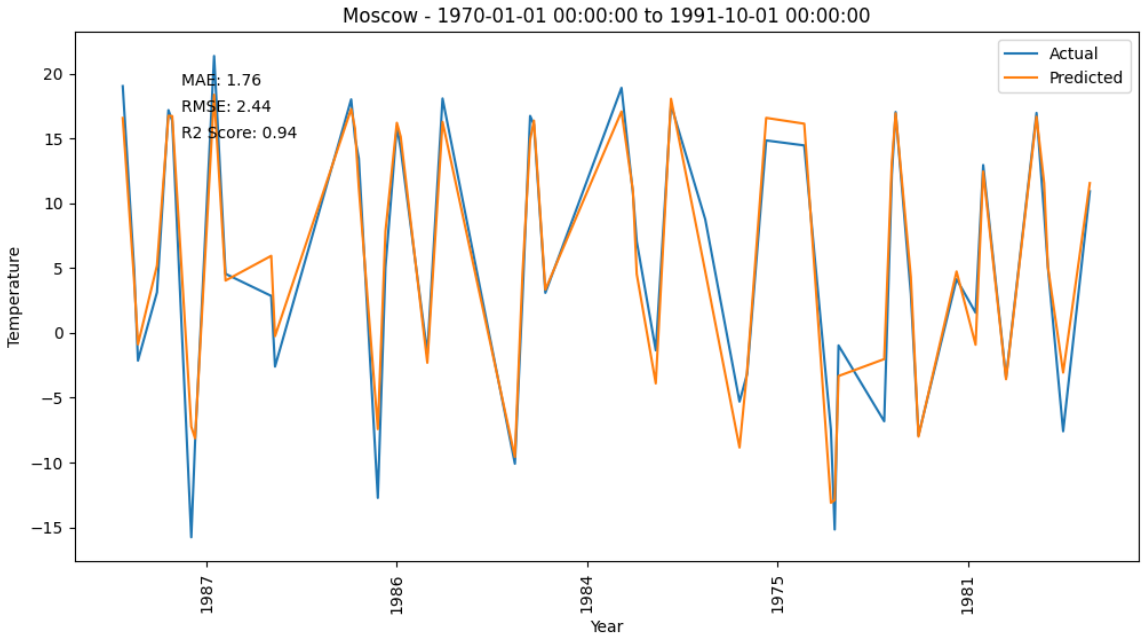
Random Forest Regressor on df_city

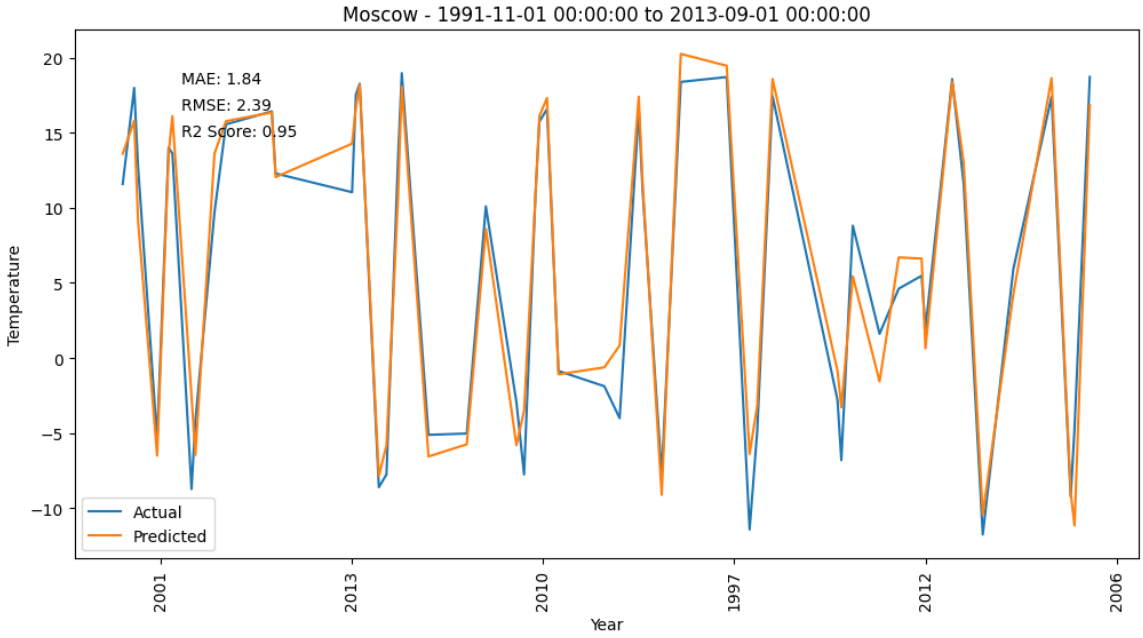
The Random Forest Regressor is a powerful machine learning algorithm widely used for regression tasks, including predictive modeling and forecasting. It is a part of the Random Forest ensemble learning method, which combines multiple decision trees to make accurate predictions. In our project "Climate Change: Earth Surface Temperature Data" dataset, the Random Forest Regressor is employed to build predictive models for temperature prediction on the "Climate Change: Earth Surface Temperature Data" dataset.

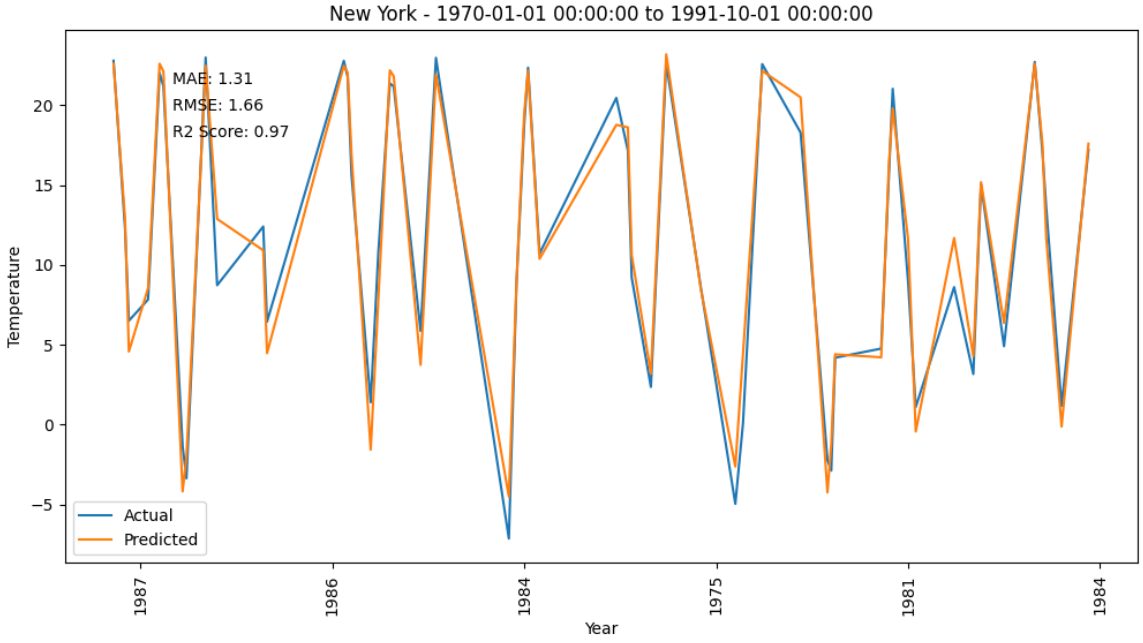
1. **Data Preparation**: Initially we select relevant features for the prediction task, such as 'Year,' 'Month,' 'Country,' and 'City.' then we convert categorical variables ('Country' and 'City') into numeric form using one-hot encoding. Then, we split data into training and testing sets, ensuring that the data is organized by city and temporally split into two halves for each city.
2. **Model**: The model is constructed by a loop that iterates over each city and its two temporal halves, we train a Random Forest Regressor model. The model is trained on 'Year' and 'Month' features to predict 'AverageTemperature.' After training, we evaluate the model's performance using metrics like Mean Absolute Error , Root Mean Squared Error , and R-squared for each city and time period. Finally we create line plots to visualize the predicted and actual temperature values for each city and time period.

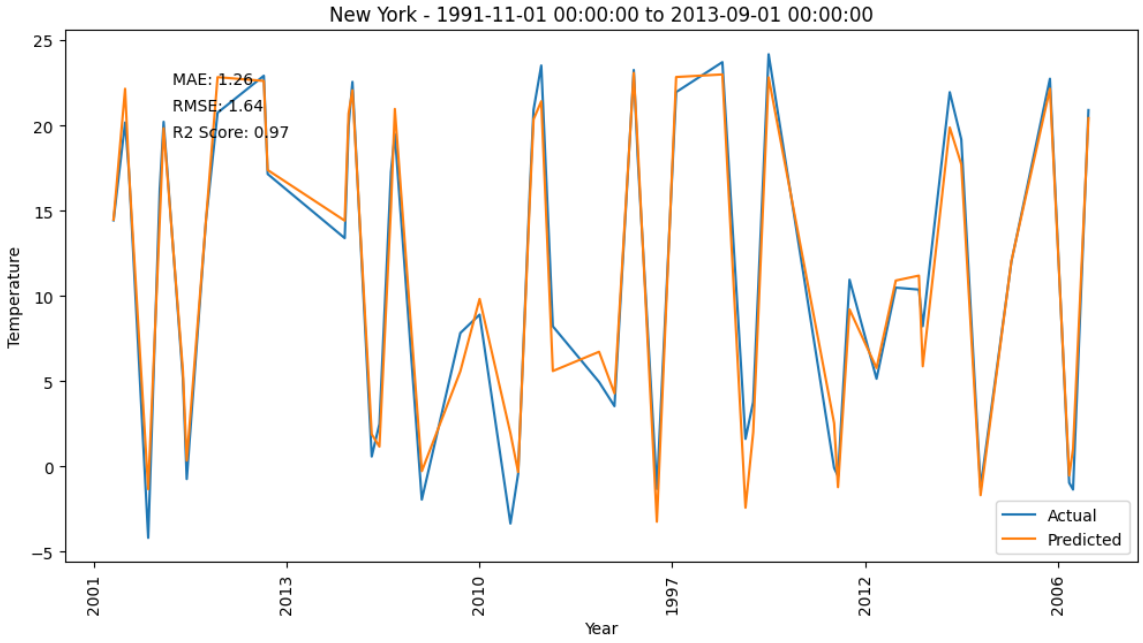


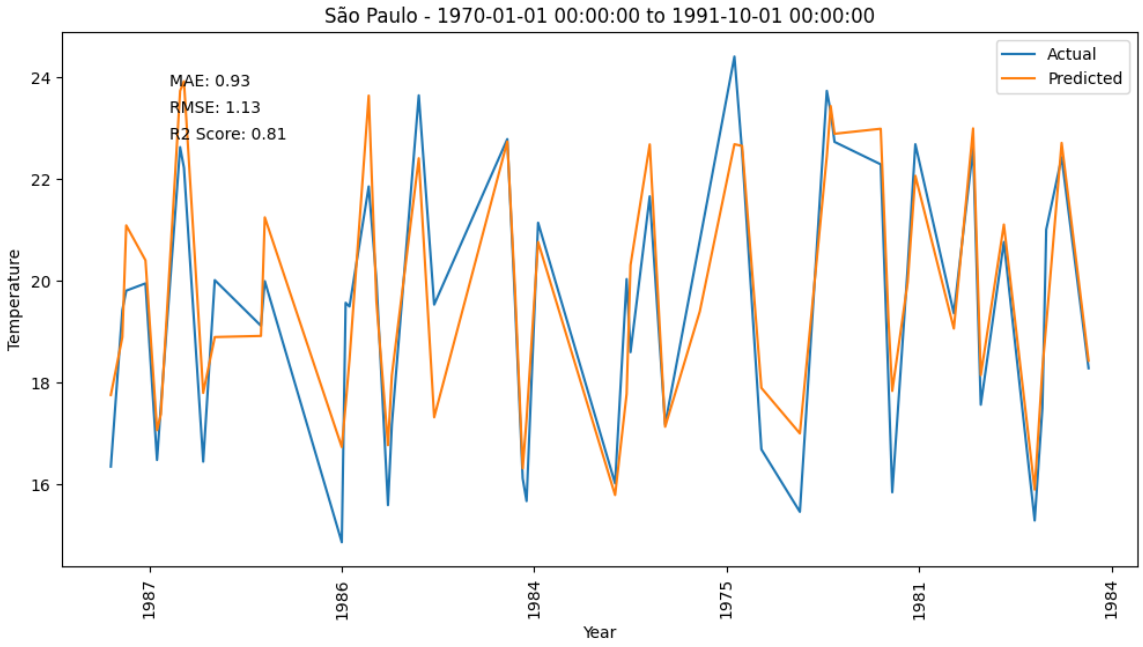

Bombay - 1991-11-01 00:00:00 to 2013-09-01 00:00:00
MAE: 0.31
RMSE: 0.45
R2 Score: 0.92
Actual
Predicted
Temperature
Year
2001
2013
2010
1997
2012

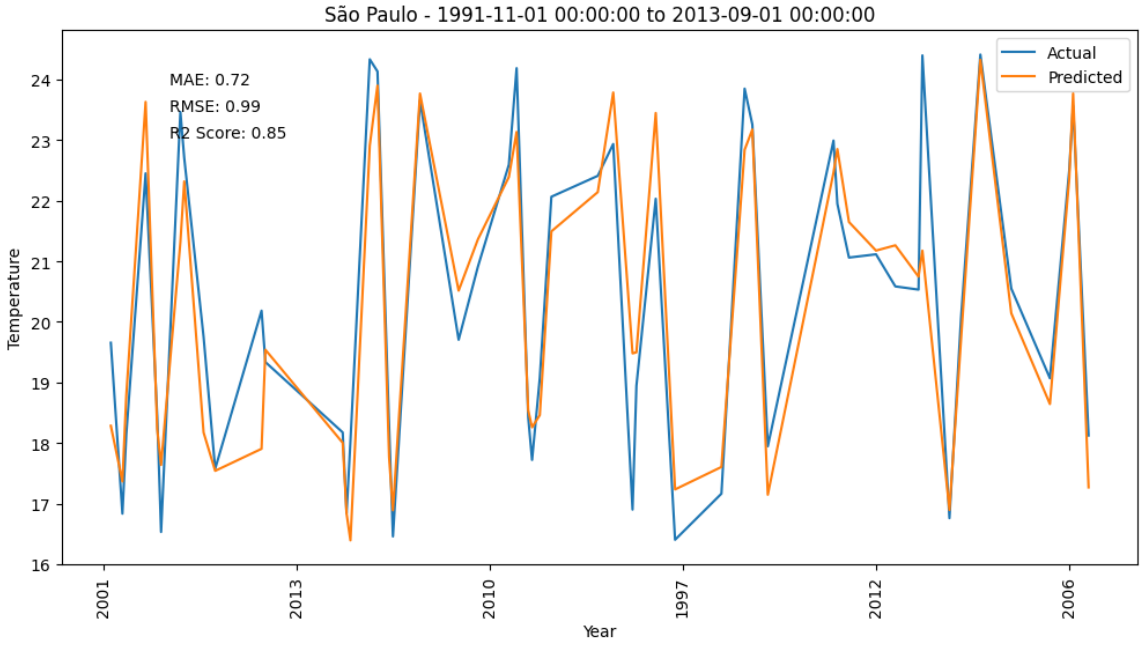

Moscow - 1970-01-01 00:00:00 to 1991-10-01 00:00:00
MAE: 1.76
RMSE: 2.44
R2 Score: 0.94
Actual
Predicted
Temperature
Year
1987
1986
1984
1975
1981

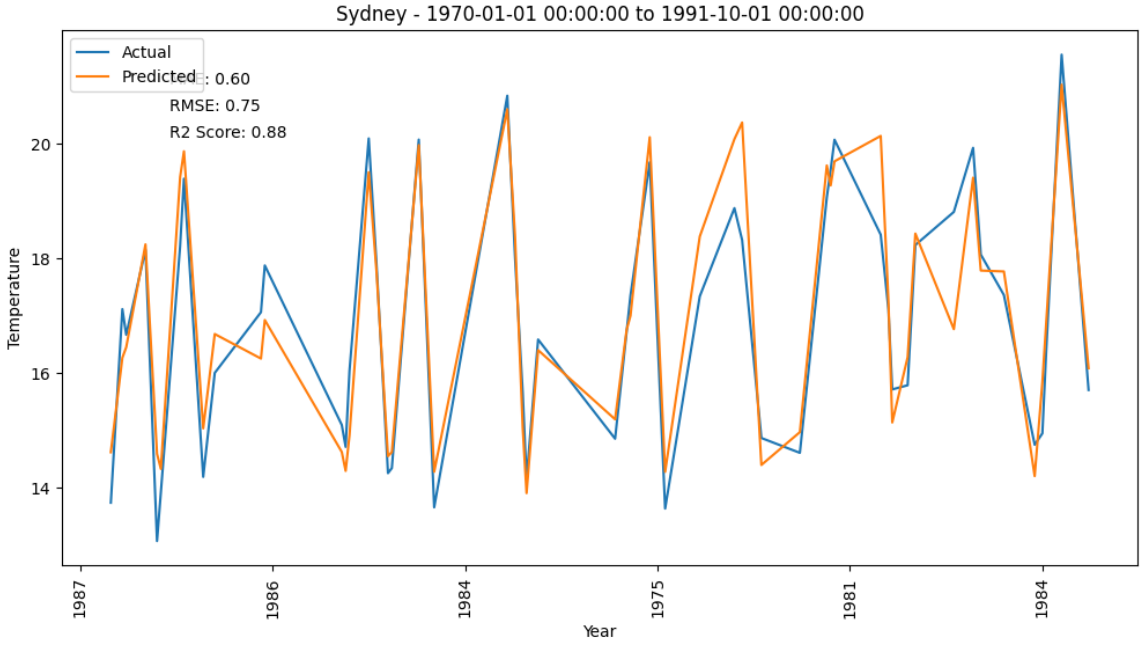

Moscow - 1991-11-01 00:00:00 to 2013-09-01 00:00:00
MAE: 1.84
RMSE: 2.39
R2 Score: 0.95
Actual
Predicted
Temperature
Year
2001
2013
2010
1997
2012
2006

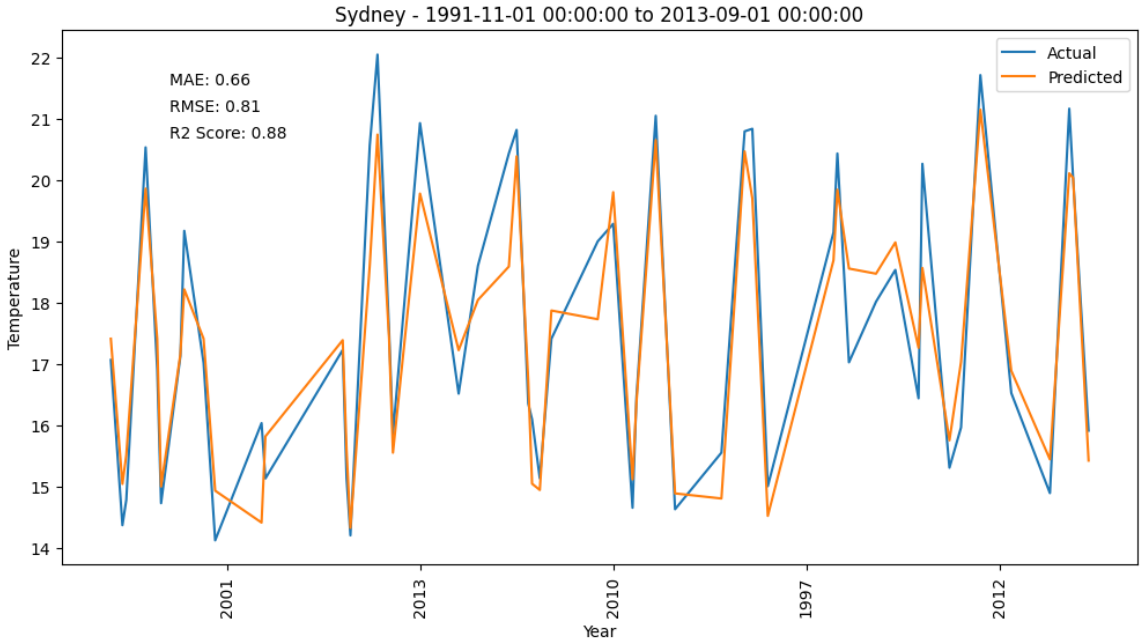

New York - 1970-01-01 00:00:00 to 1991-10-01 00:00:00
MAE: 1.31
RMSE: 1.66
R2 Score: 0.97
Actual
Predicted
Temperature
Year
1987
1986
1984
1975
1981
1984

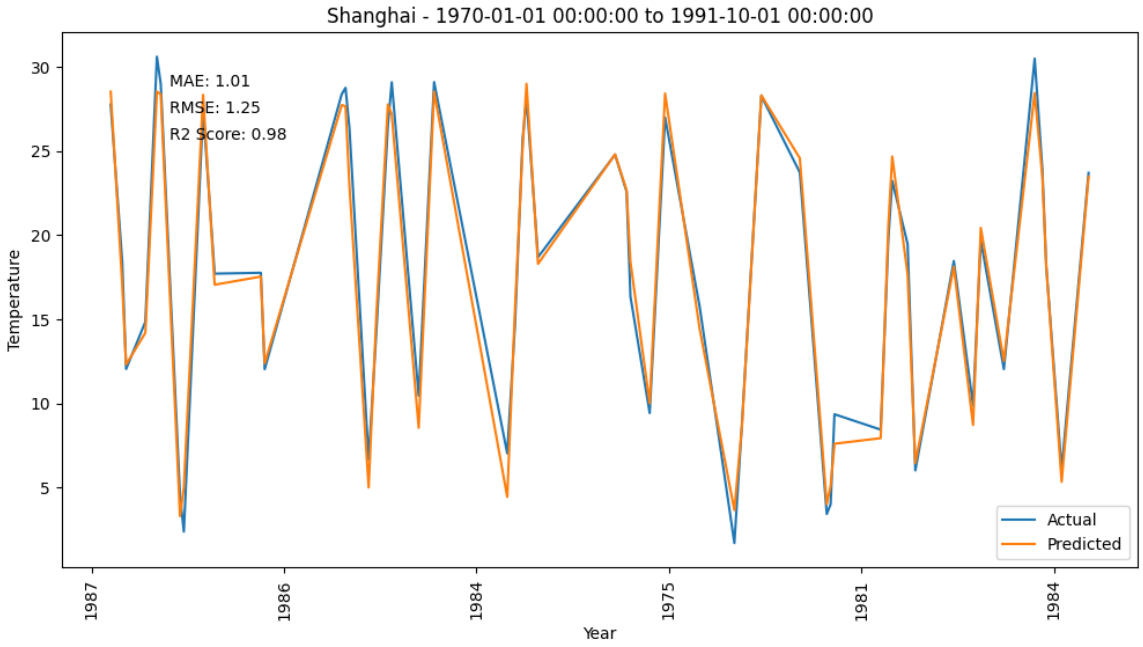

New York - 1991-11-01 00:00:00 to 2013-09-01 00:00:00
MAE: 1.26
RMSE: 1.64
R2 Score: 0.97
Temperature
Year
Actual
Predicted
2001
2013
2010
1997
2012
2006

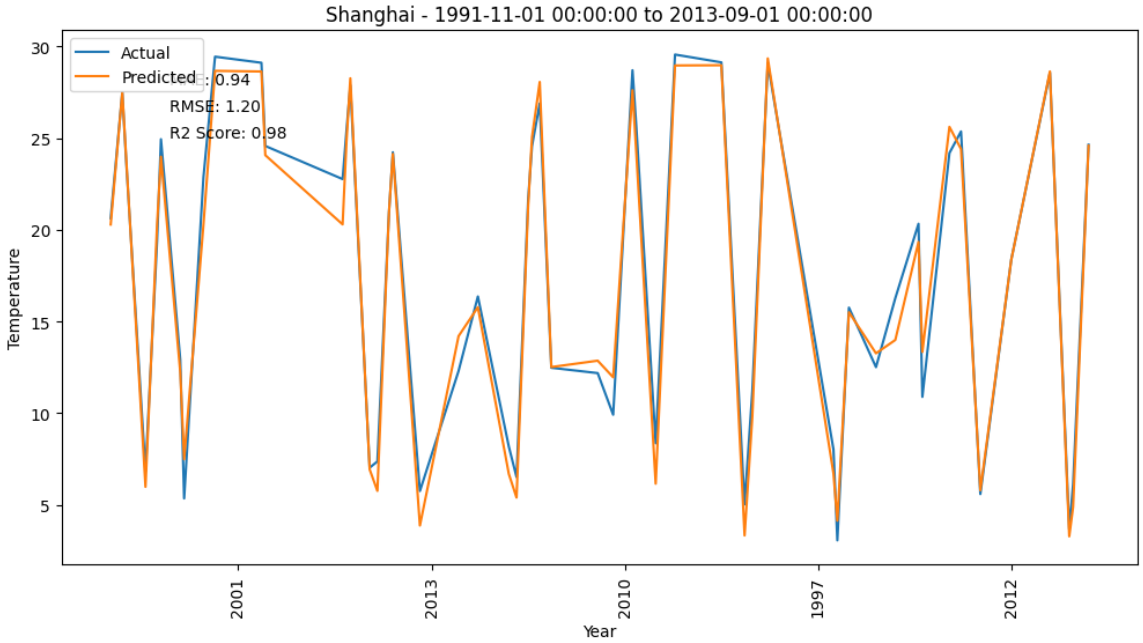

São Paulo - 1970-01-01 00:00:00 to 1991-10-01 00:00:00
MAE: 0.93
RMSE: 1.13
R2 Score: 0.81
Temperature
Year
Actual
Predicted
1987
1986
1984
1975
1981
1984

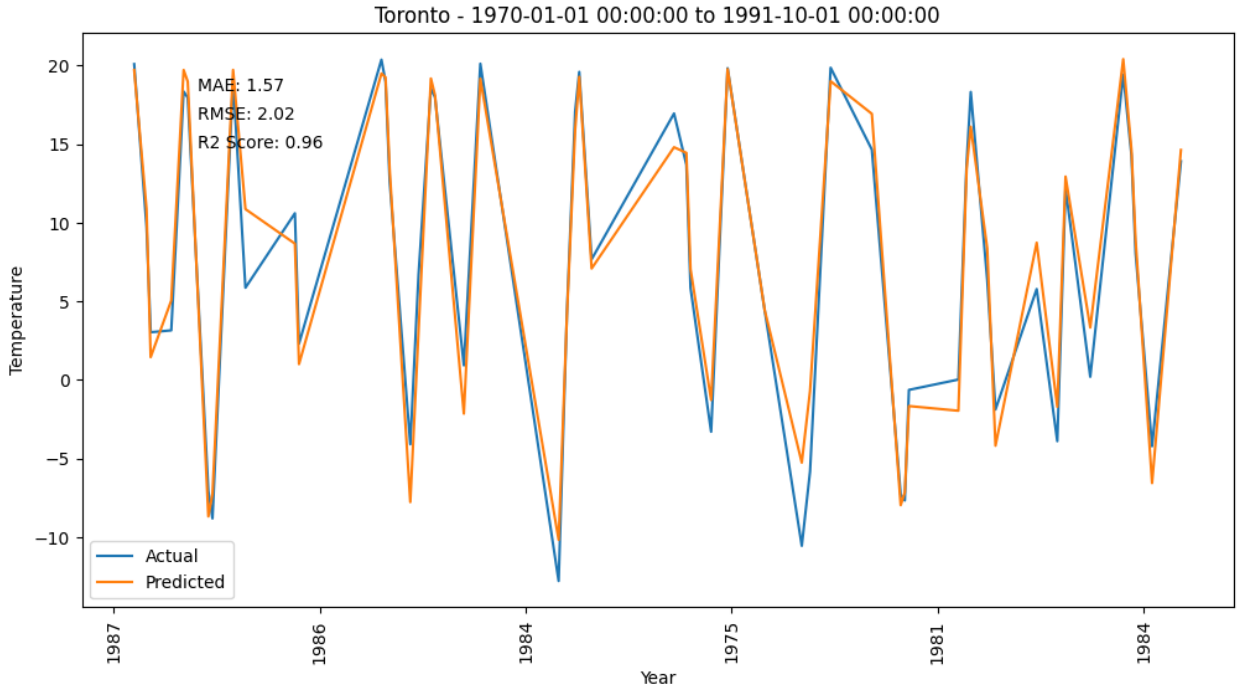

São Paulo - 1991-11-01 00:00:00 to 2013-09-01 00:00:00
MAE: 0.72
RMSE: 0.99
R2 Score: 0.85
Actual
Predicted
Temperature
Year
2001
2013
2010
1997
2012
2006

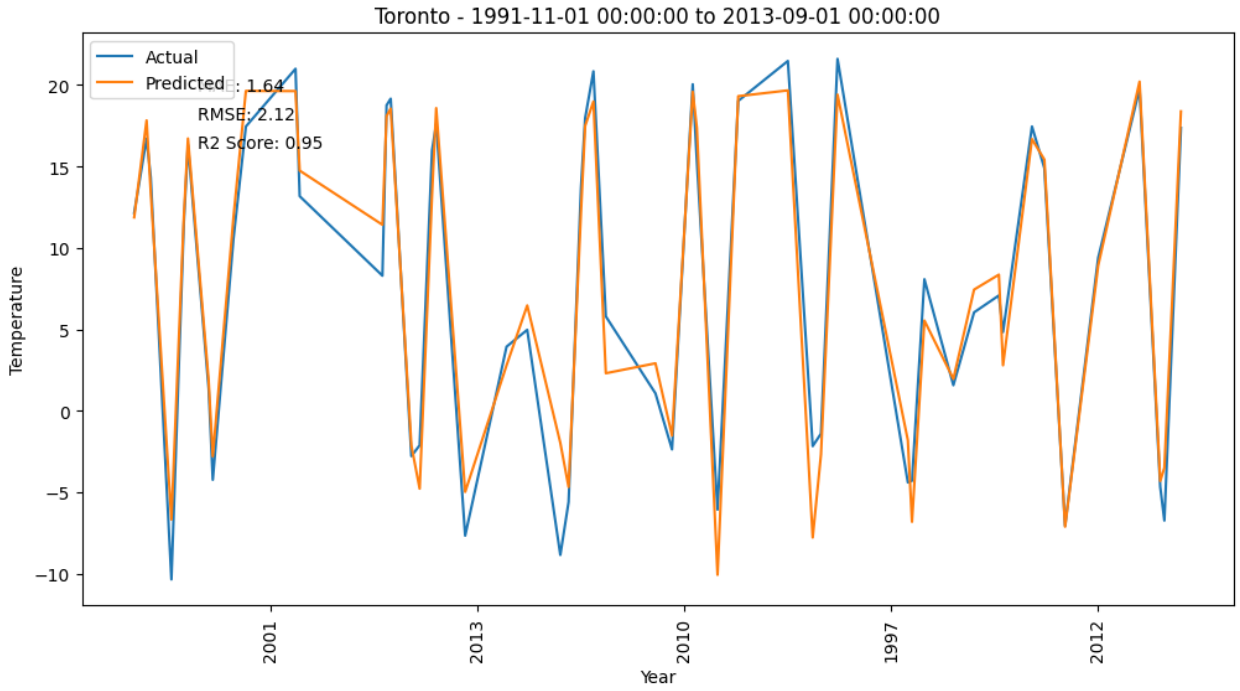

Sydney - 1970-01-01 00:00:00 to 1991-10-01 00:00:00
Actual
Predicted
: 0.60
RMSE: 0.75
R2 Score: 0.88
Temperature
Year
1987
1986
1984
1975
1981
1984


Sydney - 1991-11-01 00:00:00 to 2013-09-01 00:00:00
MAE: 0.66
RMSE: 0.81
R2 Score: 0.88
Actual
Predicted
Temperature
Year
2001
2013
2010
1997
2012


Shanghai - 1970-01-01 00:00:00 to 1991-10-01 00:00:00
MAE: 1.01
RMSE: 1.25
R2 Score: 0.98
Actual
Predicted
Temperature
Year
1987
1986
1984
1975
1981
1984

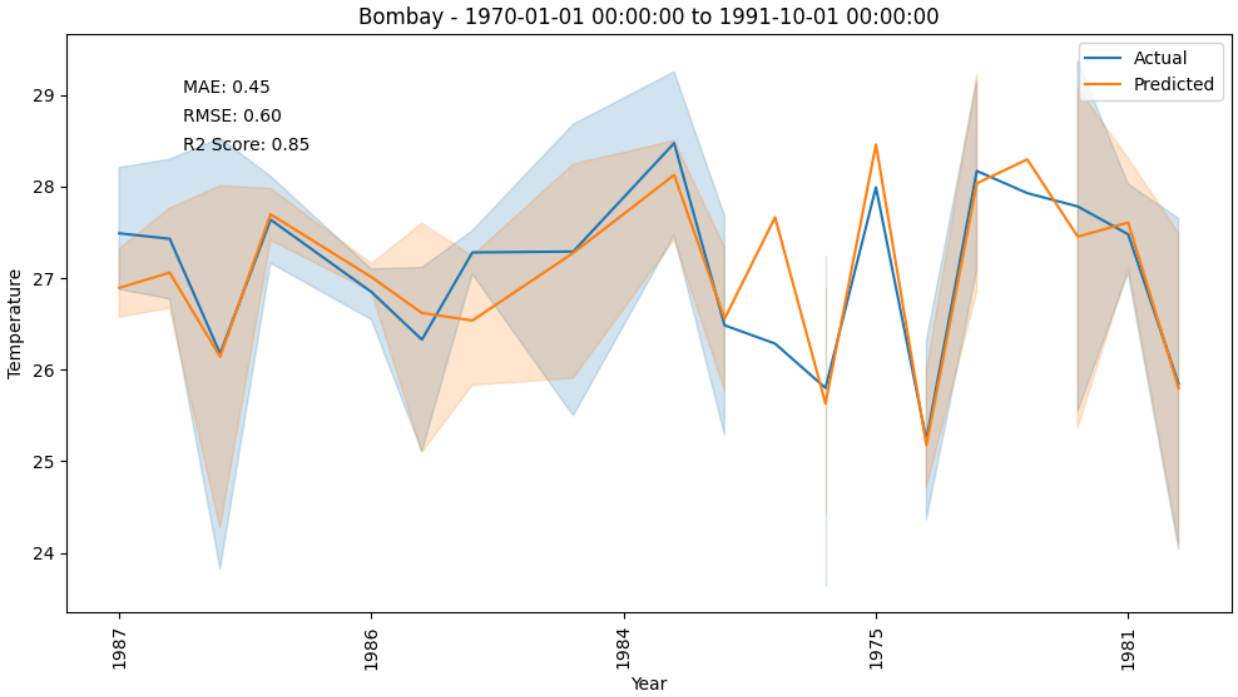

Shanghai - 1991-11-01 00:00:00 to 2013-09-01 00:00:00
Actual
Predicted
MAE: 0.94
RMSE: 1.20
R2 Score: 0.98
Temperature
Year
2001
2013
2010
1997
2012

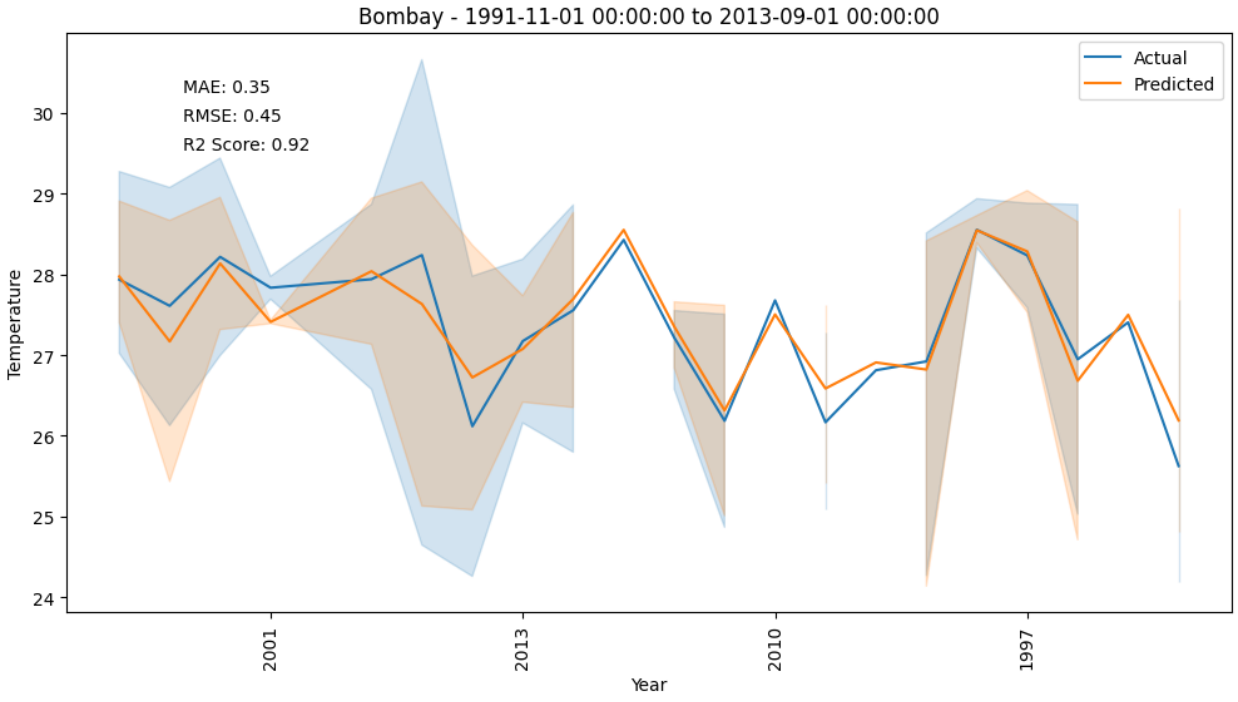

Toronto - 1970-01-01 00:00:00 to 1991-10-01 00:00:00
MAE: 1.57
RMSE: 2.02
R2 Score: 0.96
Actual
Predicted
Temperature
Year
1987
1986
1984
1975
1981
1984

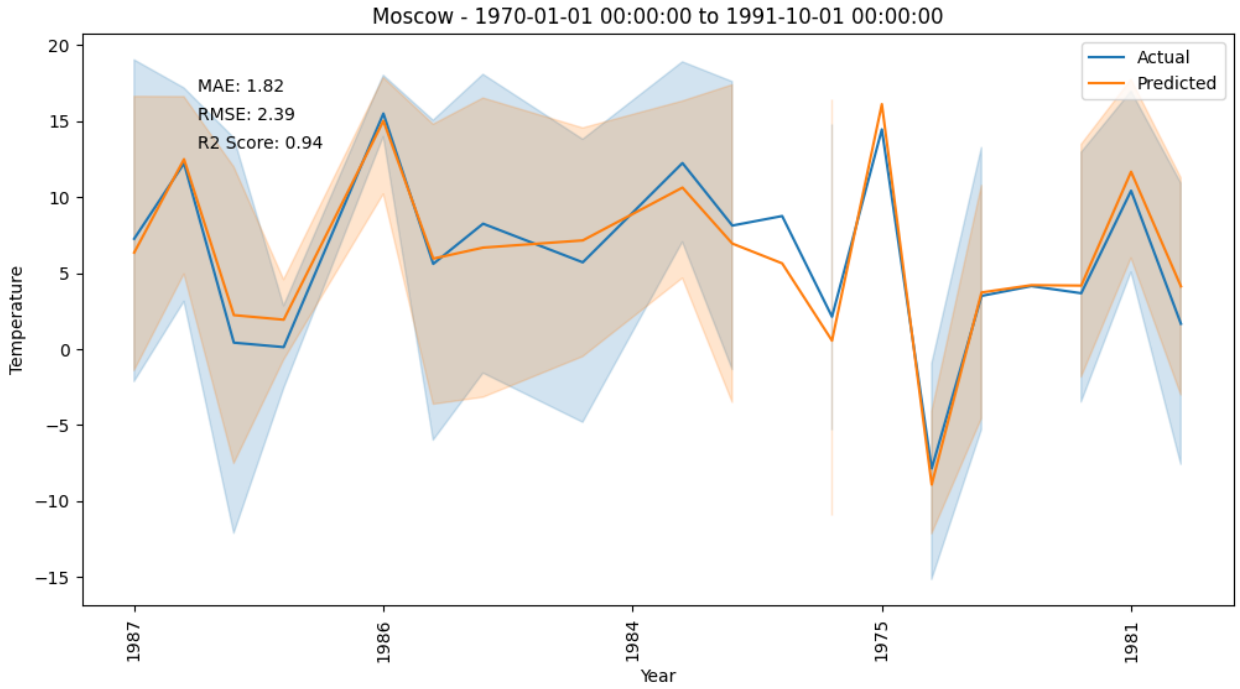


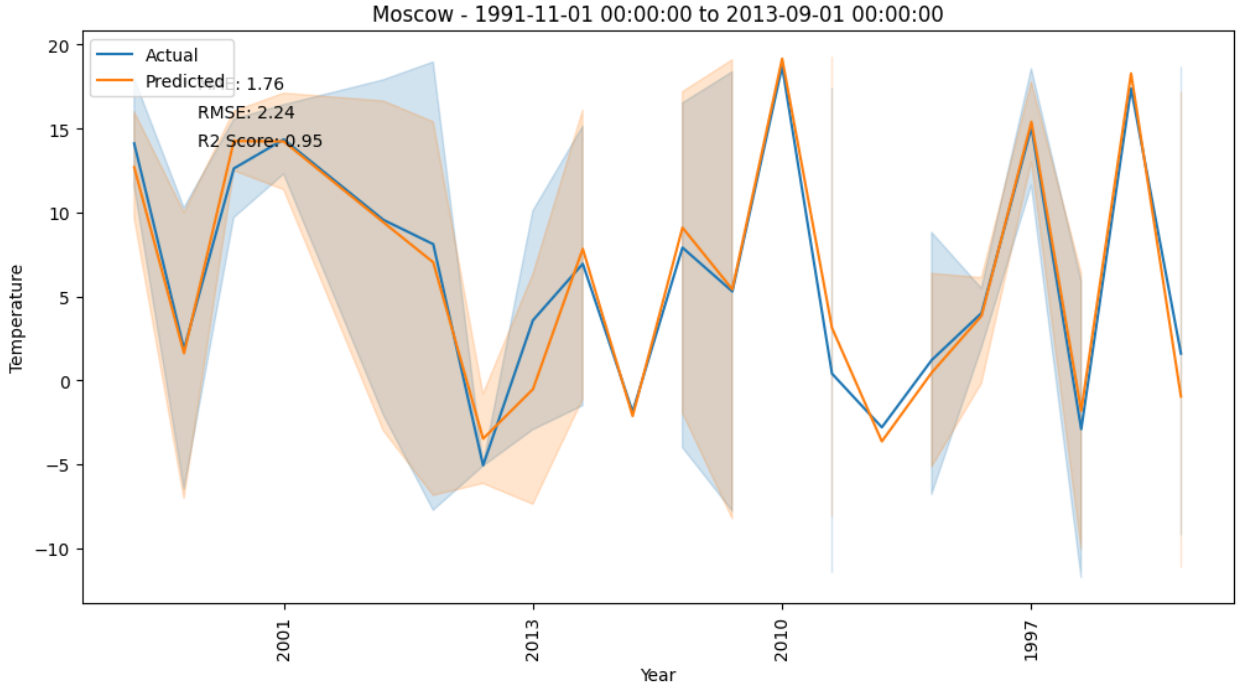
Gradient Boosting Regression on df_city

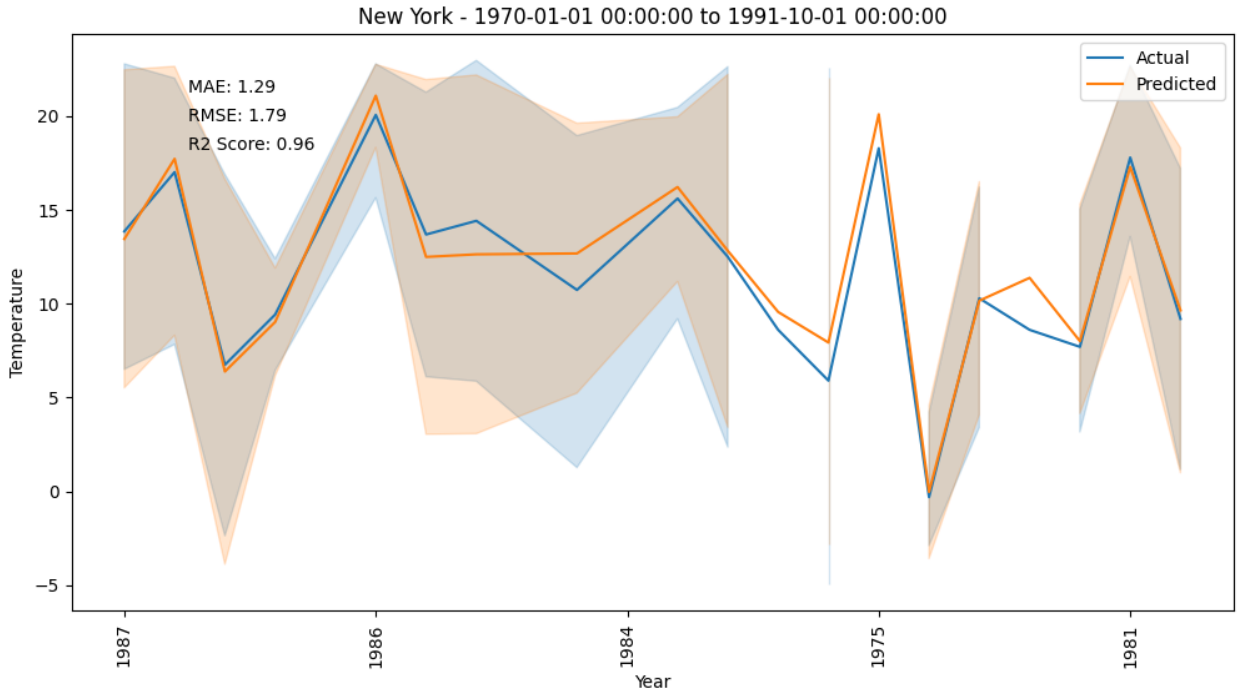
Gradient Boosting Regression is a powerful ensemble machine learning technique used for regression tasks, including predictive modeling and forecasting. Unlike traditional decision tree models, Gradient Boosting builds a strong predictive model by combining multiple weak learners, typically decision trees, in an iterative manner. It aims to minimize the residuals (differences between predicted and actual values) of the previous models by training subsequent models on these residuals. We performed the same set of task we did in the previous section, below you can see the results. Later we shall compare the two methods.

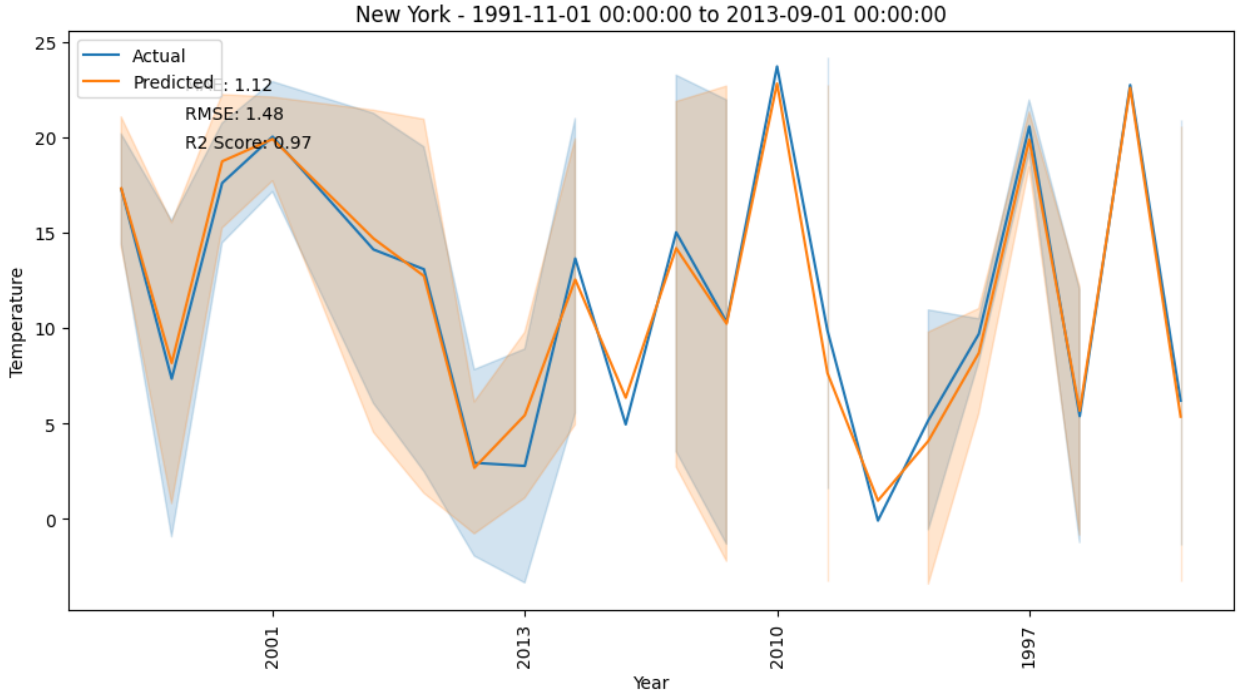

Bombay - 1970-01-01 00:00:00 to 1991-10-01 00:00:00
MAE: 0.45
RMSE: 0.60
R2 Score: 0.85
Actual
Predicted
Temperature
Year
1987
1986
1984
1975
1981

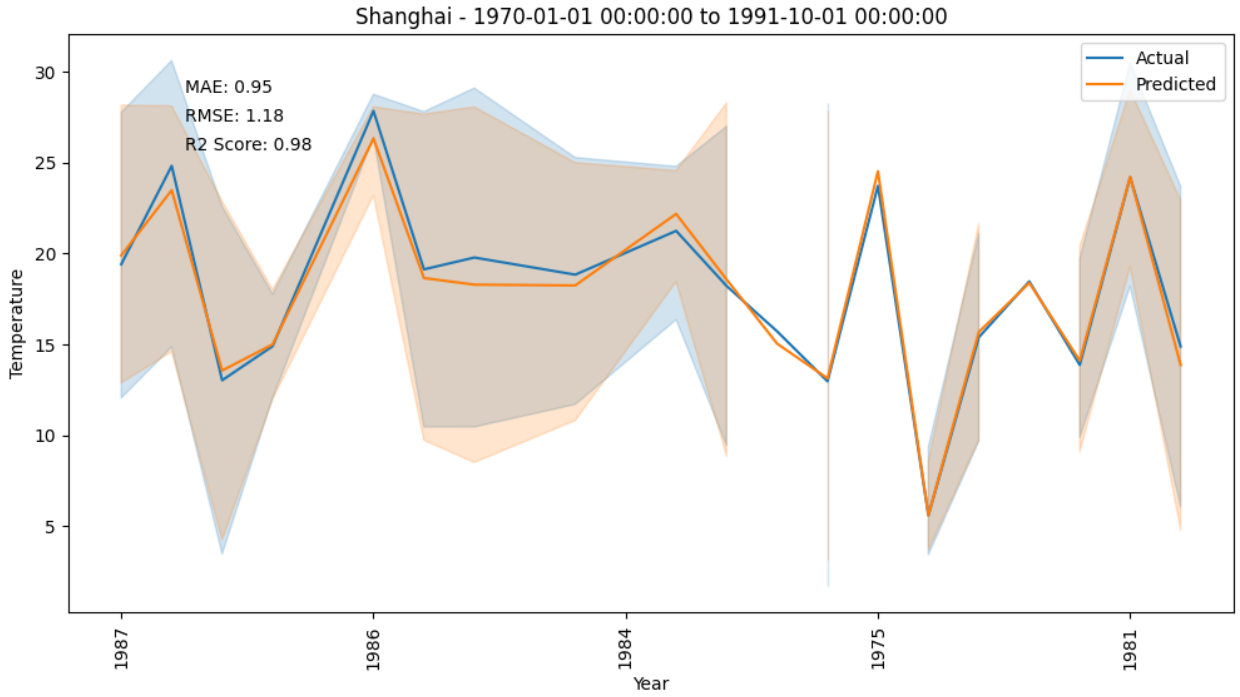

Bombay - 1991-11-01 00:00:00 to 2013-09-01 00:00:00
MAE: 0.35
RMSE: 0.45
R2 Score: 0.92
Actual
Predicted
Temperature
Year
2001
2013
2010
1997

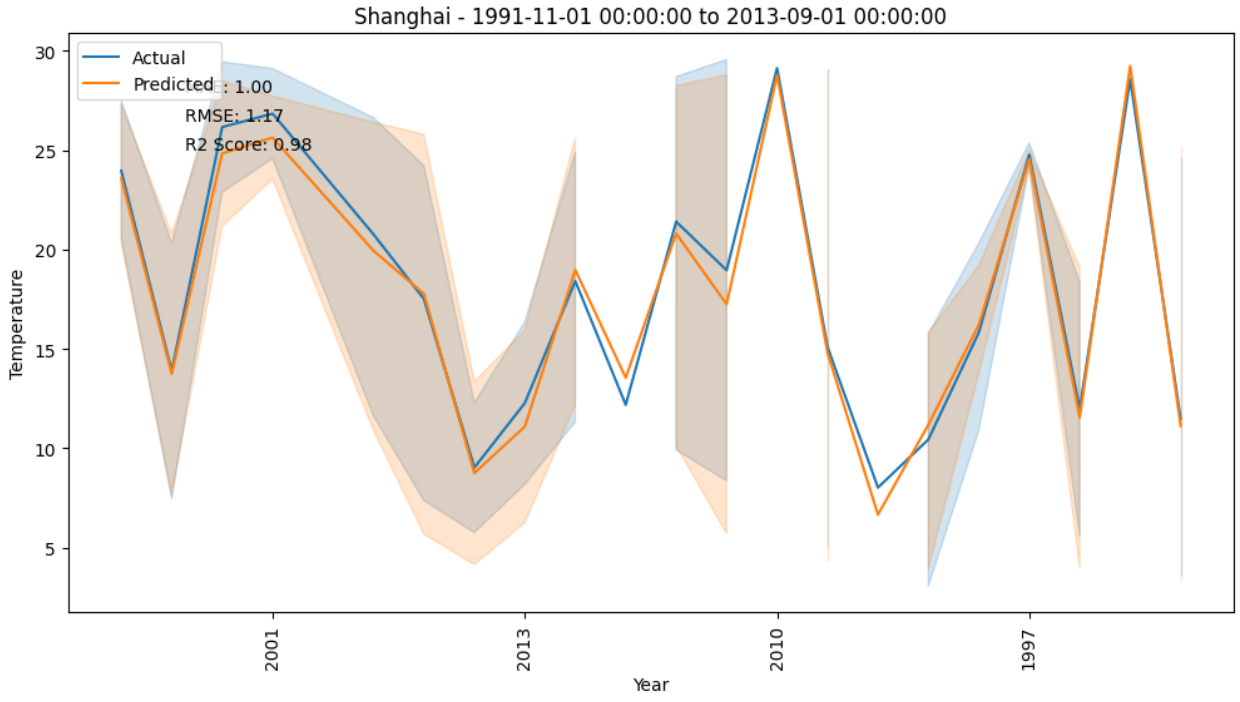

Moscow - 1970-01-01 00:00:00 to 1991-10-01 00:00:00
MAE: 1.82
RMSE: 2.39
R2 Score: 0.94
Actual
Predicted
Temperature
Year
1987
1986
1984
1975
1981

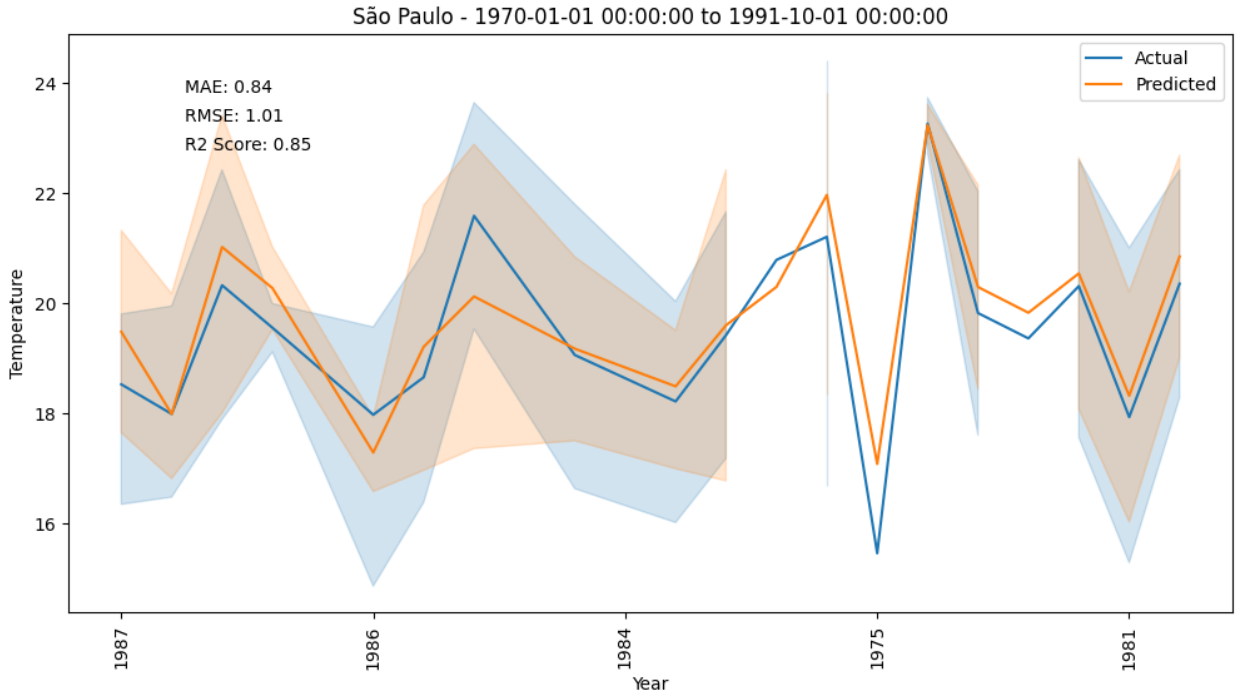

Moscow - 1991-11-01 00:00:00 to 2013-09-01 00:00:00
Actual
Predicted
: 1.76
RMSE: 2.24
R2 Score: 0.95
Temperature
Year
2001
2013
2010
1997

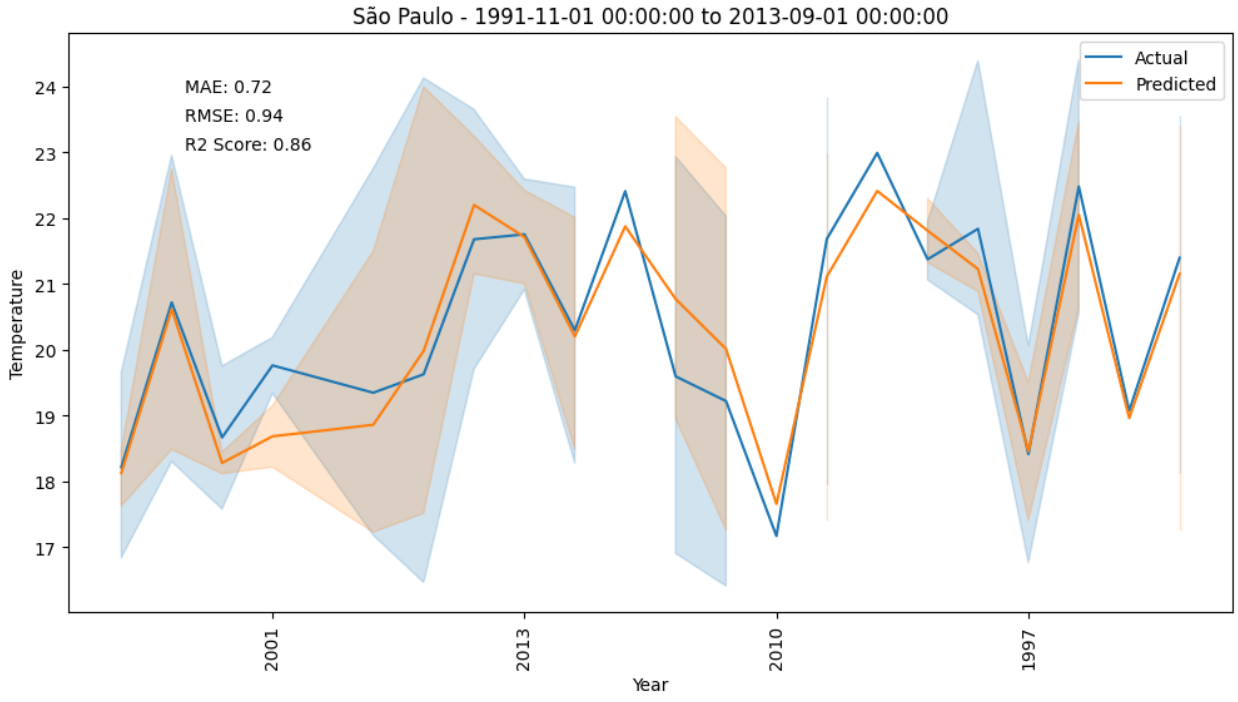

New York - 1970-01-01 00:00:00 to 1991-10-01 00:00:00
Actual
Predicted
MAE: 1.29
RMSE: 1.79
R2 Score: 0.96
Temperature
Year
1987
1986
1984
1975
1981

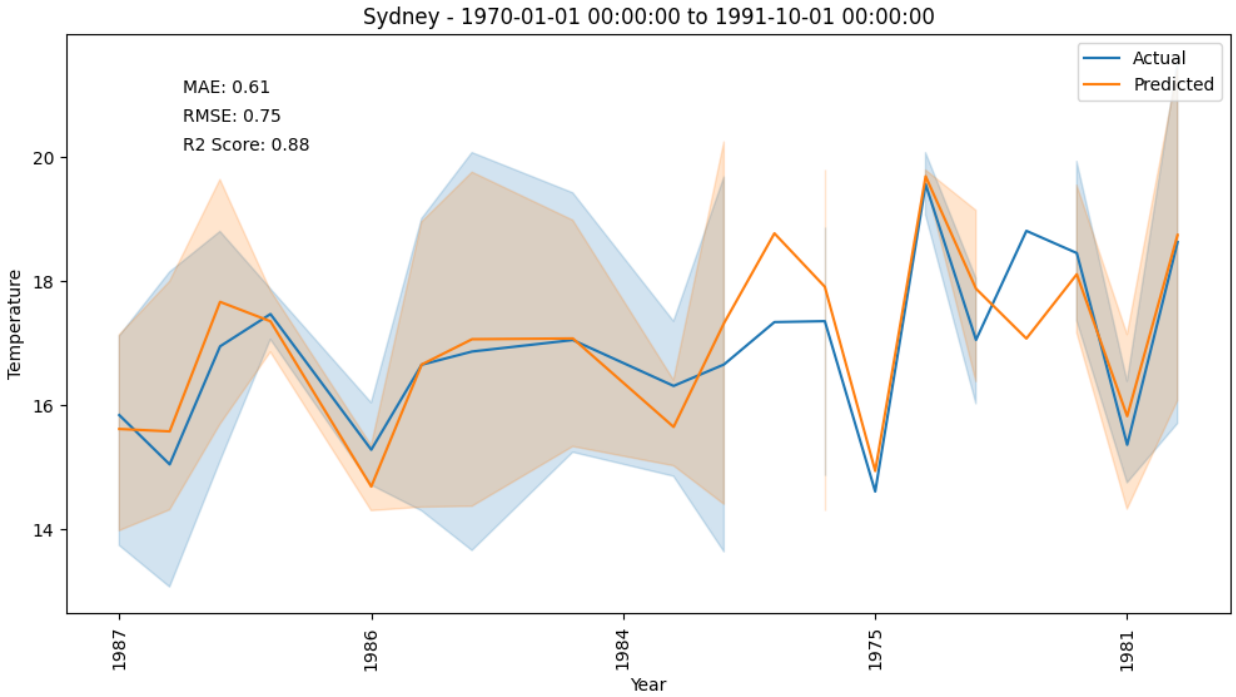

New York - 1991-11-01 00:00:00 to 2013-09-01 00:00:00
Actual
Predicted
RMSE: 1.48
R2 Score: 0.97
Temperature
Year
2001
2013
2010
1997

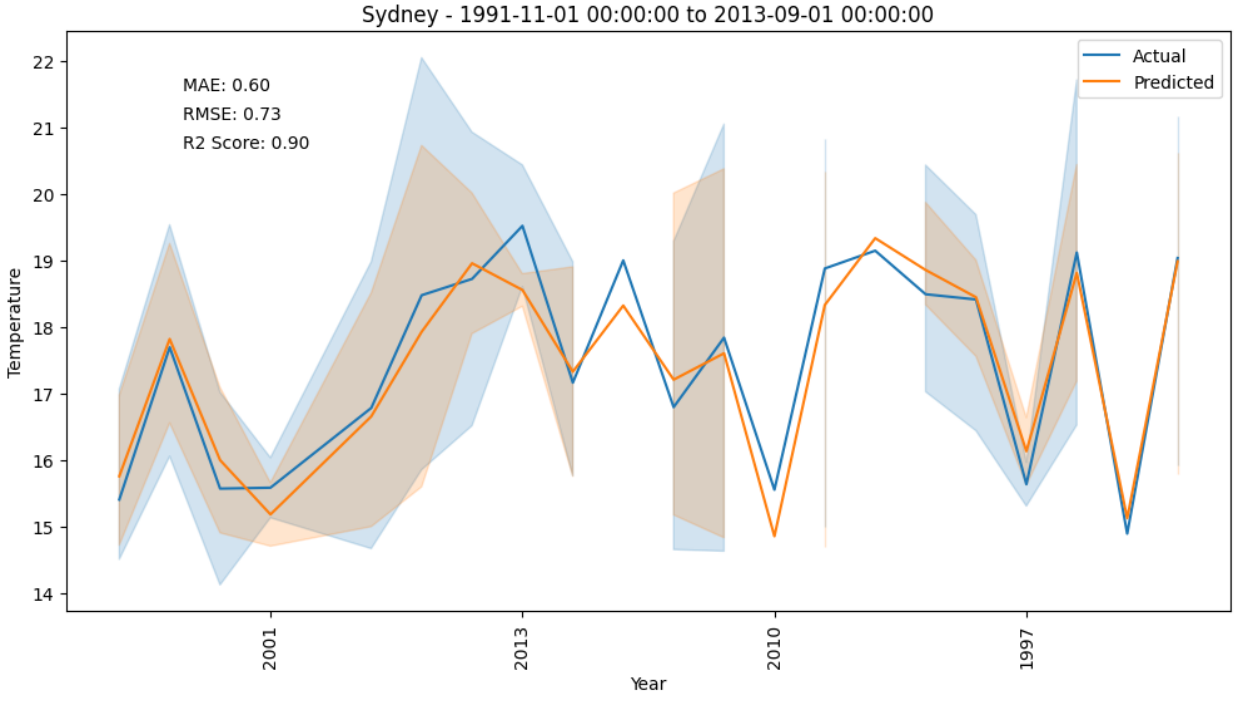

Shanghai - 1970-01-01 00:00:00 to 1991-10-01 00:00:00
MAE: 0.95
RMSE: 1.18
R2 Score: 0.98
Actual
Predicted
Temperature
Year
1987
1986
1984
1975
1981

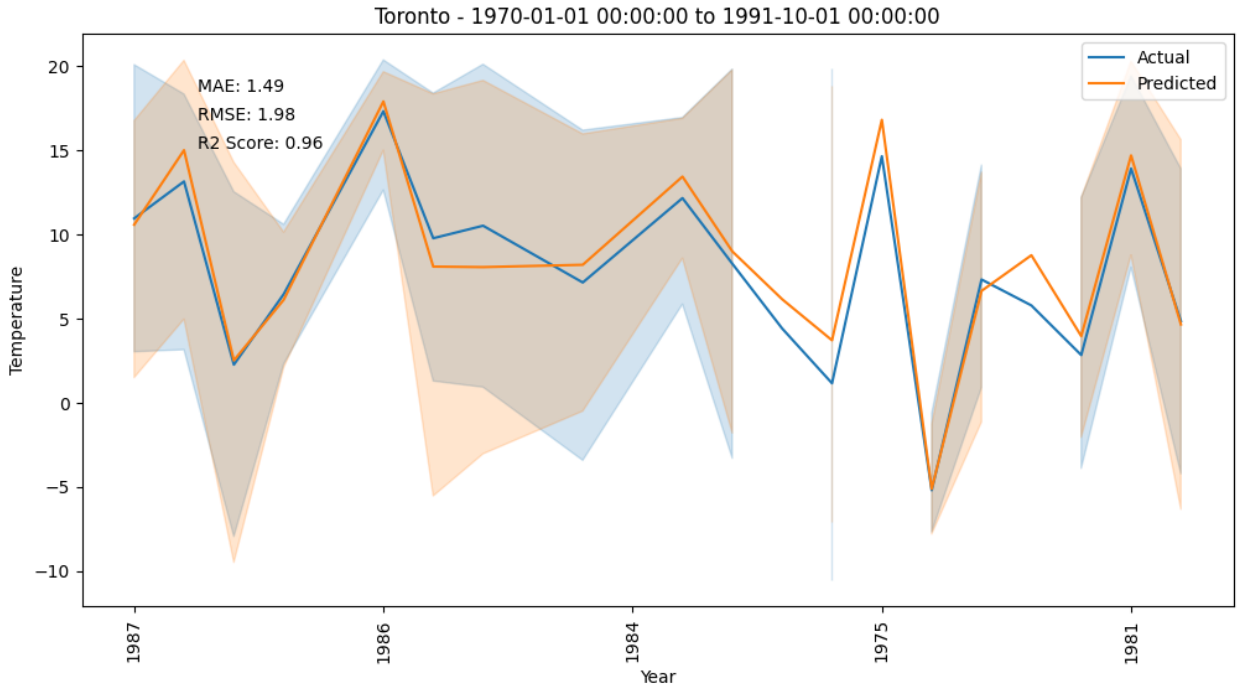

Shanghai - 1991-11-01 00:00:00 to 2013-09-01 00:00:00
Actual
Predicted
RMSE: 1.17
R2 Score: 0.98
Temperature
Year
2001
2013
2010
1997

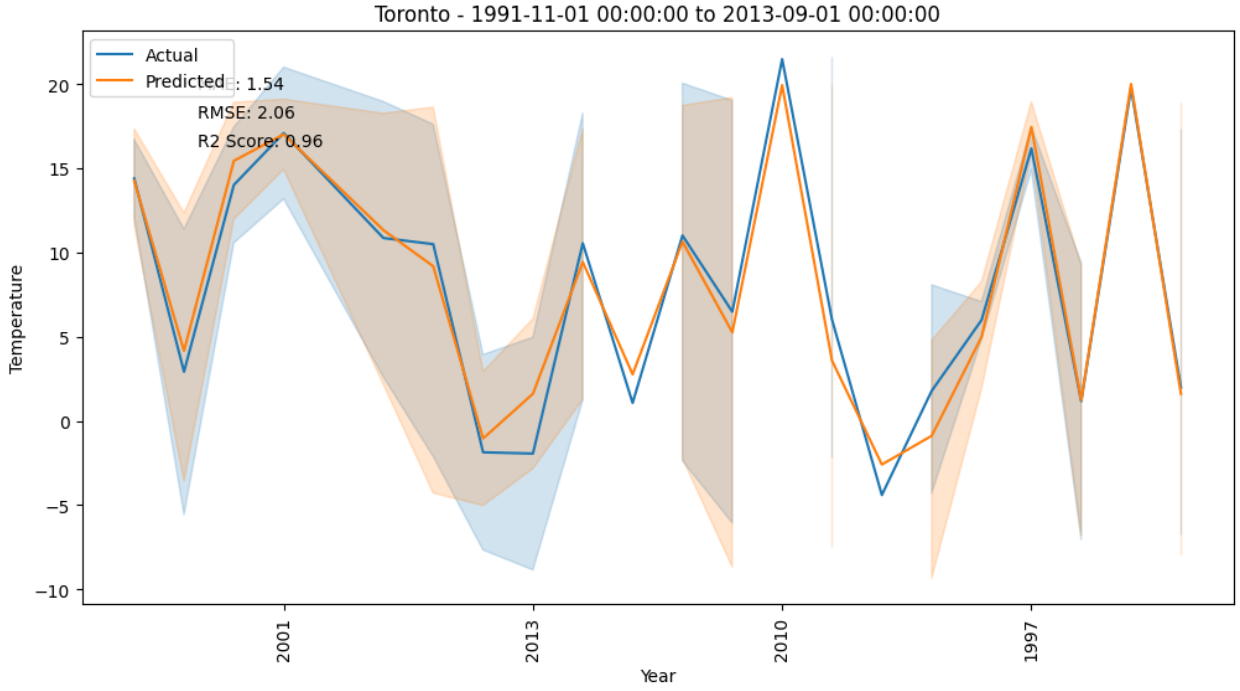

São Paulo - 1970-01-01 00:00:00 to 1991-10-01 00:00:00
MAE: 0.84
RMSE: 1.01
R2 Score: 0.85
Actual
Predicted
Temperature
Year
1987
1986
1984
1975
1981


São Paulo - 1991-11-01 00:00:00 to 2013-09-01 00:00:00
MAE: 0.72
RMSE: 0.94
R2 Score: 0.86
Actual
Predicted
Temperature
Year
2001
2013
2010
1997


Sydney - 1970-01-01 00:00:00 to 1991-10-01 00:00:00
MAE: 0.61
RMSE: 0.75
R2 Score: 0.88
Actual
Predicted
Temperature
Year
1987
1986
1984
1975
1981


Sydney - 1991-11-01 00:00:00 to 2013-09-01 00:00:00
MAE: 0.60
RMSE: 0.73
R2 Score: 0.90
Actual
Predicted
Temperature
Year
2001
2013
2010
1997





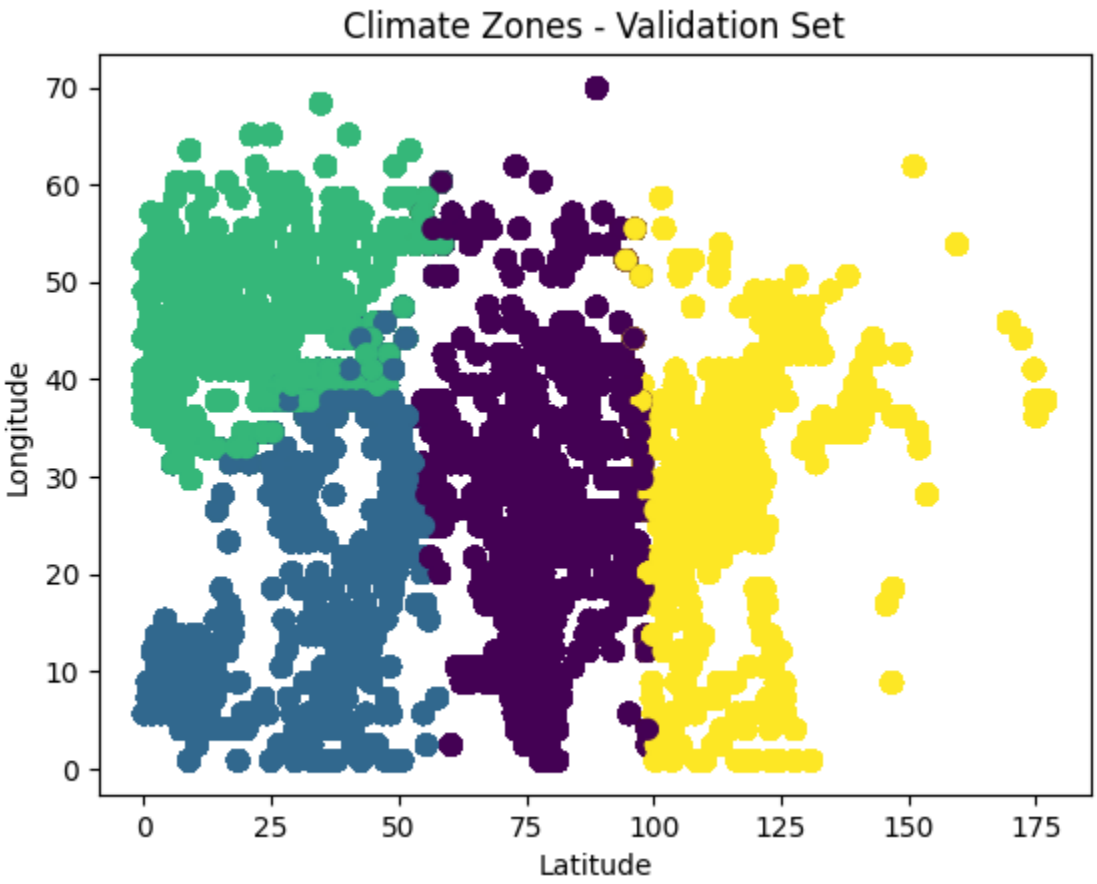
Random Forest Regressor Vs Gradient Boosting Regression

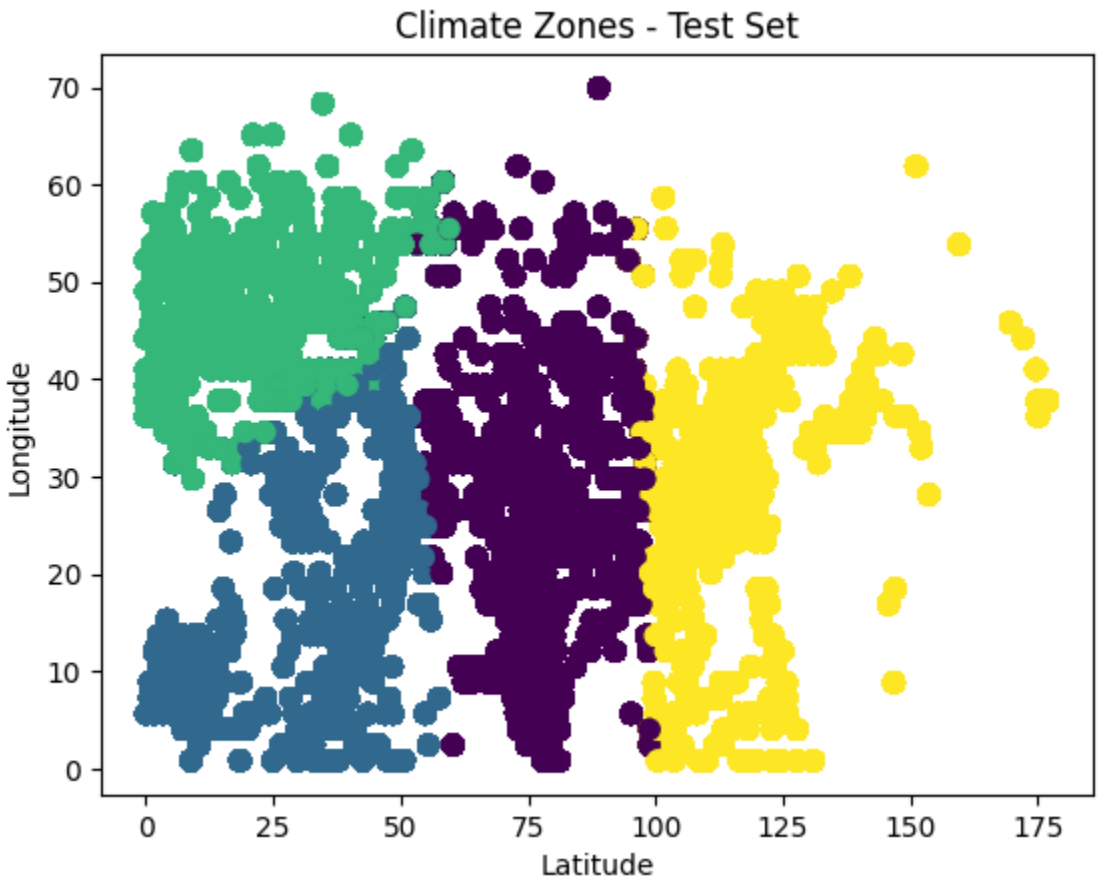
- Random Forest Regressor generally performs well across all cities and time periods.it achieves relatively low (MAE) and (RMSE) values, indicating good predictive accuracy. (R^2) scores are also high, especially for New York, Shanghai, and Toronto, indicating that it captures a significant portion of the variance in the data
- Gradient Boosting Regression, in most cases, outperforms Random Forest Regressor, achieving lower MAE and RMSE values, GBR consistently produces high R^2 scores, indicating that it explains a substantial portion of the variance in the data.The improvement in performance is especially noticeable for cities like Moscow and São Paulo.

KMeans Clustering

K-means clustering is a machine learning algorithm used for unsupervised clustering, where it groups similar data points together into clusters based on their feature similarities. in this project, K-means clustering can was applied to identify distinct patterns or regions with similar temperature trends.

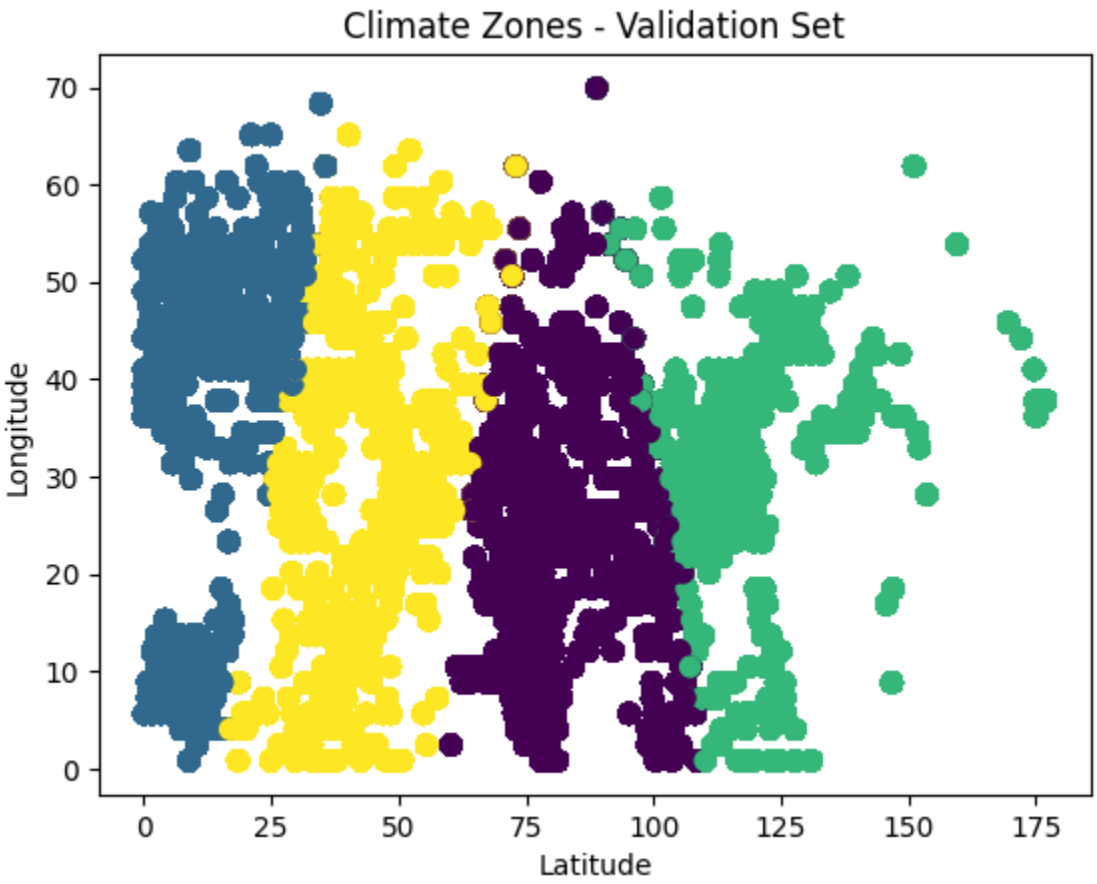
we randomly sampled 40% of the data from each stratum. This stratified sampling approach helps maintain the diversity of countries in the dataset while reducing its size, making it computationally more manageable

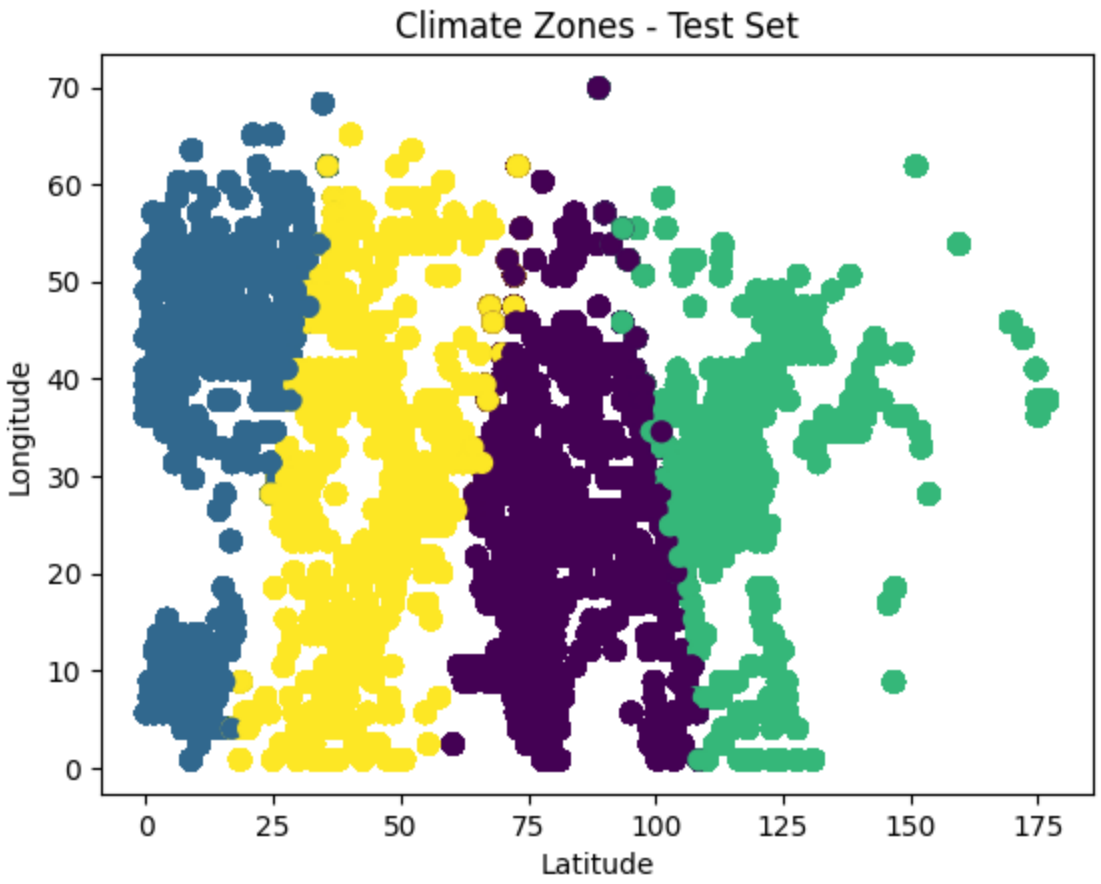

Climate Zones - Validation Set
Longitude
Latitude
0
10
20
30
40
50
60
70
0
25
50
75
100
125
150
175



Mini Batch KMeans Clustering

Mini Batch K-Means is a variant of the traditional K-Means clustering algorithm designed to handle large datasets more efficiently. In the Mini Batch K-Means algorithm, rather than updating centroids after processing the entire dataset (as in the standard K-Means), it updates centroids based on small, random subsets or "mini-batches" of the data. This approach reduces computational complexity and memory requirements, making it suitable for large datasets.


Climate Zones - Validation Set
Longitude
Latitude
0
10
20
30
40
50
60
70
0
25
50
75
100
125
150
175



KMeans Clustering Vs Mini Batch KMeans Clustering

In this section we will compare the performance of these two models to demonstrate the differences between the two. The evaluation metrics provided for both traditional K-means clustering and MiniBatchKMeans clustering indicate some differences in their performance:

1. **(WSS):** For both clustering methods, the WSS on the training set is considerably higher for MiniBatchKMeans compared to traditional K-means. This suggests that traditional K-means may have achieved better compactness on the training data. However, both models have similar and very low WSS values on the validation and test sets, indicating that they perform well in terms of clustering data points together.
2. (CHI)**:** The Calinski-Harabasz Index measures the ratio of between-cluster variance to within-cluster variance. A higher value indicates better separation between clusters. In this case, traditional K-means outperforms MiniBatchKMeans on the test set, suggesting that it might have created more distinct and well-separated clusters.

3. (DBI)**:** The Davies-Bouldin Index measures the average similarity between each cluster and its most similar neighbor, with lower values indicating better clustering. Traditional K-means has slightly lower Davies-Bouldin Index values for all sets, indicating that it might have produced clusters with slightly less overlap compared to MiniBatchKMeans.

In summary, based on these metrics, traditional K-means appears to perform slightly better than MiniBatchKMeans . However, the differences between the two methods are relatively small.The tabels bellow show the evaluation metrics for traditional K-means and MiniBatchKMeans respectively.

KMeans Clustering

Training Set	Validation Set	Test Set
(WSS) 383019882.0437846	(WSS) 17.464956899937203	(WSS) 17.479360421251567
(CHI)1567822.4860365319	(CHI)1569640.8193488847	(CHI)2090215.5601428647
(DBI)0.9538093205811329	(DBI)0.9522577262359173	(DBI)0.9535783669398441

Training Set	Validation Set	Test Set
(WSS) 402705575.8408018	(WSS) 17.741402816960857	(WSS) 17.75196440083742
(CHI)1475955.9691354835	(CHI)1475877.3014649486	(CHI)1966594.4844587562
(DBI)1.0070011174349516	(DBI)1.0063501229513296	(DBI)1.0074369476069112