Data_Ninjas Team (Serial No. D45)

Index

OVERVIEW

TASK

Task is to design and implement a machine learning model that predicts tourist arrivals to any specific destination using Internet search index data as one of the primary features. The model can assist tourism authorities and businesses in forecasting tourist arrivals, thereby enabling better resource allocation, marketing strategies, and overall destination management. We have to extract data on tourist arrivals, internet search indexes, and other relevant features such as hotel booking and flight search and it covers the weekly span between 2010 and 2022. The sample will include any one popular tourist destination, namely, Shimla, Kerala and Goa in our case. The forecasting and time-series analysis process can be used to identify patterns in the data, such as long-term trends and seasonal patterns.

TIME SERIES FORECASTING

Time series forecasting in machine learning is a powerful technique for predicting future values based on historical data ordered by time. It finds applications across various domains, from finance to weather forecasting and from sales prediction to anomaly detection. Time series forecasting models leverage patterns, seasonality, and trends present in past data to make predictions about what may happen in the future. Popular algorithms for time series forecasting include Autoregressive Integrated Moving Average (ARIMA), Exponential Smoothing methods, and more recently, advanced deep learning models like Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks. Time series forecasting plays a crucial role in decision-making, resource allocation, and risk management, helping organizations make informed choices based on historical patterns and trends.

APPROACH

This is found to be a typical problem of time series forecasting. We first collected both monthly and weekly data for regions including Shimla, Kerala, and Goa from various tourism websites and articles. Then, we applied a wide range of predictive models, including Random Forest, XGBoost, CATboost and LSTM with a focus on optimizing their hyperparameters. This step ensured that our models were fine-tuned to deliver the most precise forecasts possible. To further refine our predictions, we employed the prophet to accurately forecast future data based on statistical features. Subsequently, we visualized the forecasted values. To assess the performance of our models, we utilized the R2 Score and RMSE as primary evaluation metrics. This comprehensive process allowed us to make data-driven forecasts and informed decisions in the respective locations.

DATA EXTRACTION

The very first task is data collection and data scraping. We had to collect data from various websites and tourism articles like IndiaStat, Government Websites and other tourism sources to build time series machine learning models. The quality and quantity of the data collected directly impact the performance and reliability of our model. Our target destinations include Shimla , Goa and Kerala. Tourist arrival data for Goa is typically obtained from several sources, including government agencies, tourism departments, and international organizations. For Shimla and Kerala, data is specifically collected from research articles and tourism statistics. Various sites such as prominent travel sites and government databases were used to finally build our dataset.

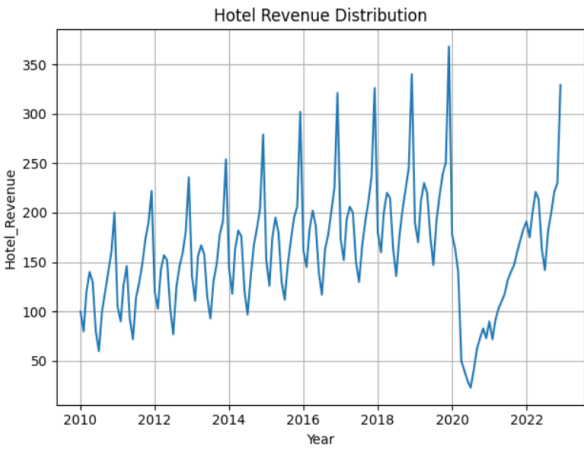
DATA COLLECTION

Web scraping is an essential technique used to extract data from websites. It involves automating the process of sending HTTP requests to a website, retrieving the web page's HTML content, and then parsing that content to extract specific information. Web scraping is a valuable tool for collecting data from the internet for various purposes, such as research, analysis, or data analysis. Web scraping can be done manually by inspecting the website's source code, but it is often automated using programming languages like Python with libraries such as Beautiful Soup, Scrapy, and requests. These libraries provide the tools to send HTTP requests, parse HTML, and extract data from web pages efficiently.

EXPLORATORY DATA ANALYSIS

GOA

A. Hotel Revenue

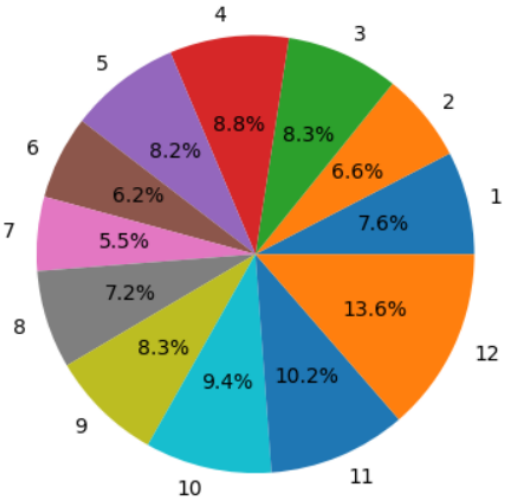


The given plot demonstrates the distribution of Hotel Revenue across time from 2010 - 2022.

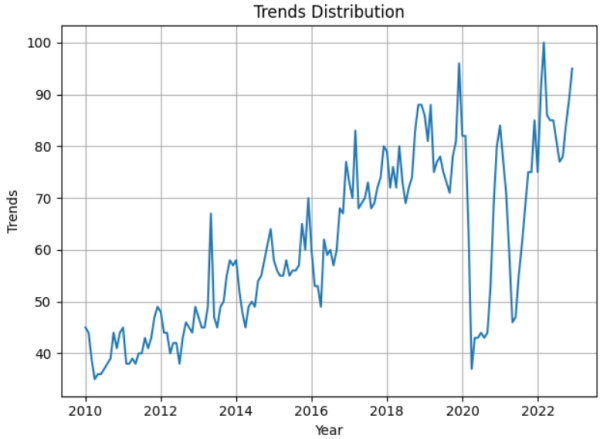
- A seasonality is observed at the end of year in the form of a peak, which might be due to the Christmas holidays.
- In general until 2019, the trend was upward, but with a sudden drop in 2020. This could be due to the COVID-19 pandemic which affected tourist demand thus had a great impact on hotel revenue.
- A recovery is observed at the end of 2021, which is expected to continue.

The given pie-chart gives us an analysis of how the hotel revenue is distributed on an average over the 12 months in a year.

- The maximum hotel revenue comes in between October to December, which can be due to a major portion of Christian population residing there.
- Also, being a prominent sea beach, the months April and May see a lot of traffic.
- Goa is known for its rich culture and diverse population. Hence, a nearly uniform distribution in the can hotel revenue by month can be attributed to the festivals going on throughout the year, along with its main attraction of sea beaches.



B. Trends



This plot shows the distribution of the number of monthly searches across time from 2010-2022.

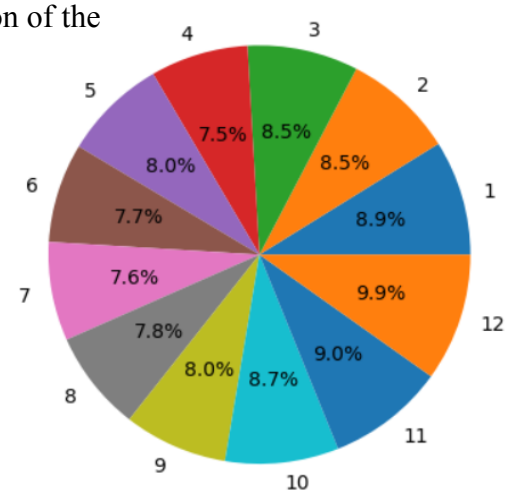
- The searches have generally shown an increase over time, which might be mainly due to incorporation of the Internet into our lives.
- The decrease in searches around 2020-2021 can be attributed to COVID-19.

The steep rise at around 2020 end or the beginning of 2021 might be due to the slight relaxation of COVID related restrictions.

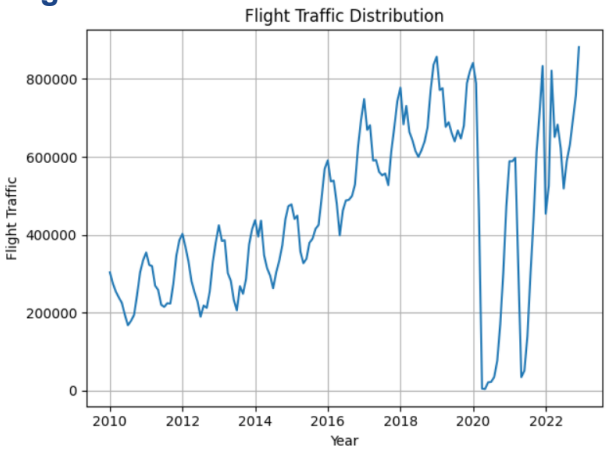
- The immediate steep fall would then be attributed to the 2nd wave of COVID.
- The trend seems to have revived itself in 2022.

We also perform analysis of the mean monthly distribution of the Trends data.

- As is clearly visible, the distributions look nearly uniform, with higher during the holiday seasons.
- Goa being one of the culturally and diversely richest states in India enjoys a fairly distributed search indices over the 12 months.
- The maximum is during the months of November and December, due to Christmas being one of their most important festivals.



C. Flight Traffic

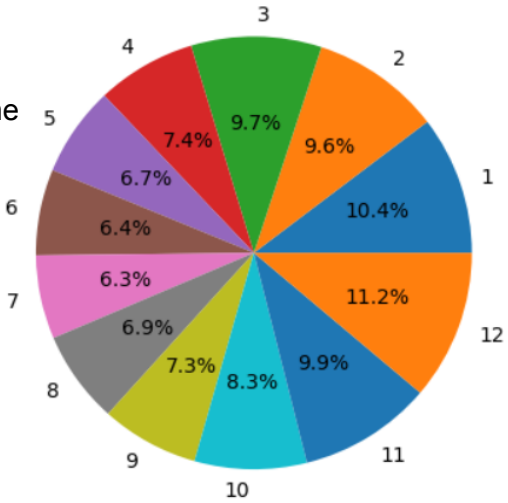


The given plot demonstrates the distribution of Hotel Revenue across time from 2010 - 2022.

Inferences:

- A drop in flight traffic seen during 2020 - 2021 period, probably due to COVID restrictions.
- The upward steep rise during the end of 2020 could be a result of the weakening of the first COVID wave.
- The consequent steep fall can be attributed to the 2nd wave of COVID.

The month-wise average flight traffic has been shown through the following pie-chart:



- Maximum traffic is again observed during Christmas and New Year.
- Nearly 60% of the traffic is between October to December and January to March.
- A level of demand for flight is maintained throughout the year.

<u>From the above three analysis, we observe a few common conclusions which we can arrive at:</u>

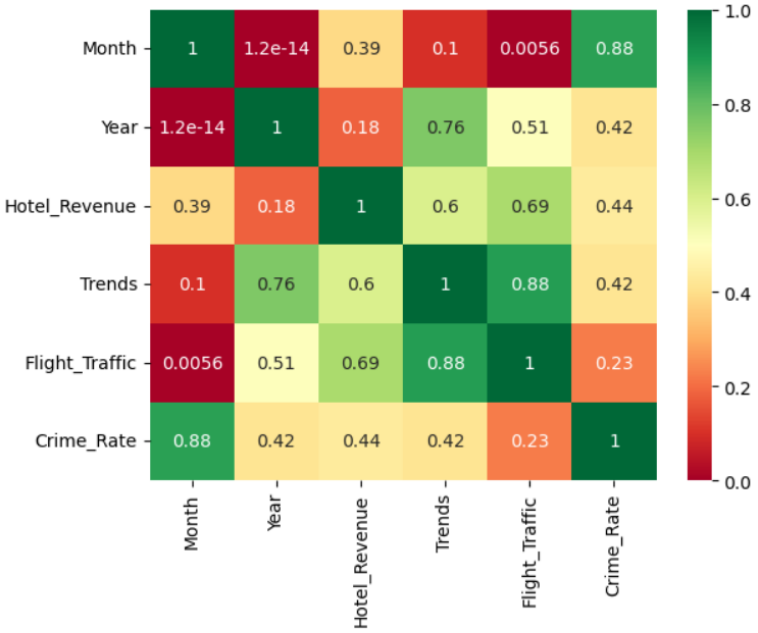
1. The tourism demand at Goa is never too low. The hotels as well as flights can expect a decent revenue throughout the year with peaks during the holiday season.
2. COVID-19 had affected the general trends during 2020-2021, where there was a sudden fall in demand.

3. The time around December, 2020 to February, 2021 showed some recovery in demand due to weakening of the first wave of COVID, which dropped again due to the arrival of the 2nd wave of COVID.

D. Correlation Analysis

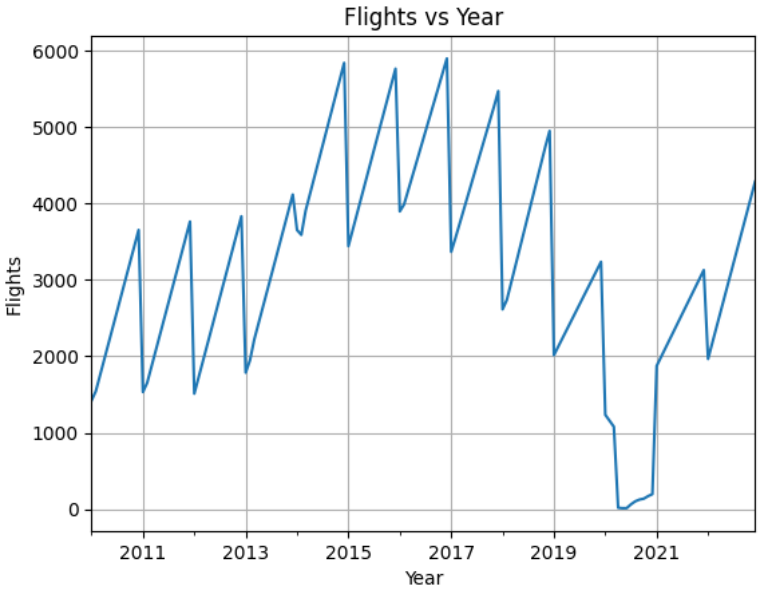
From this Correlation HeatMap plot, we can conclude that in the inner 4x4 square of the plot we see that the major colour we see is green, that means we see a high correlation between pairs of features, which form the inner 4x4 square, whereas if we see the outer layer we observe a low correlation.
Some notable Correlations-

- **Flight traffic** vs **Trends**
  Correlation is **0.88** which is a very high correlation *validating the intuition about the authenticity of the data.*
- **Year** vs **Trends**
  Correlation is **0.76** which is *pretty high.*
- **Hotel Revenue** vs **Year**
  Correlation is **0.18** which is *relatively low.*



Shimla:

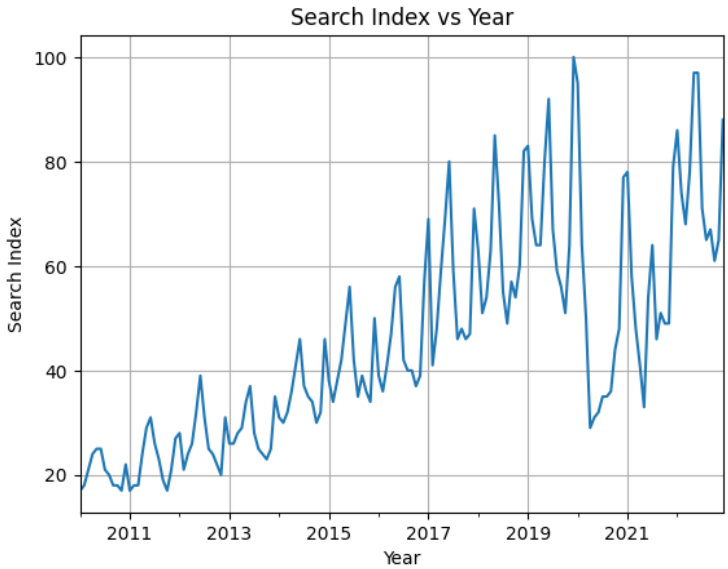
A. Flights



- The upward trend in the number of flights from **2011 to 2019** reflects the growing popularity of Shimla as a tourist destination.
- The sharp decline in the number of flights in **2020** and **2021** is a direct result of the COVID-19 pandemic. Although the number of flights to Shimla has been increasing since **2022**.

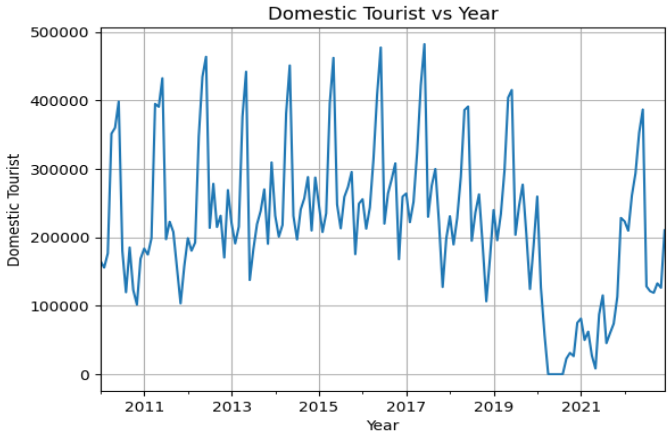
B. Search Index

The Search Index is a measure of the popularity of a search term, and it is calculated by Google based on a number of factors, including the number of searches for the term, the number of clicks on search results, and the quality of the content associated with the term. The following inferences are :



- The Search Index of the term "Shimla" increased steadily from **2011 to 2019,** reaching a peak of over **100** in **2019**. This shows the increasing popularity of Shimla as a tourist destination and also increase in Internet users.
- However the **COVID-19** pandemic imposed restrictions on travel, leading users to search less about tourist destinations.

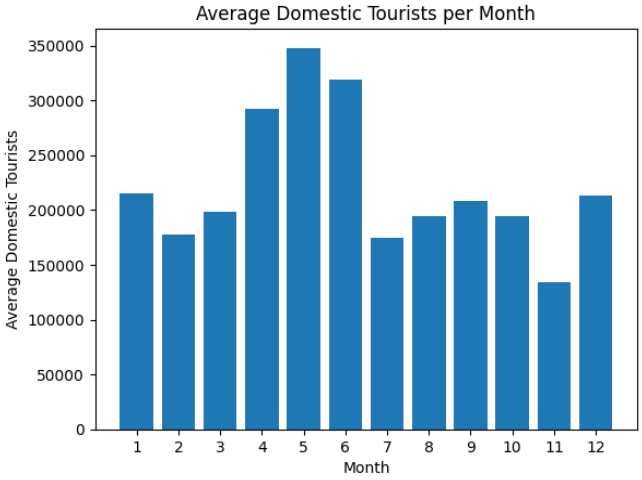
C. Domestic Tourists



- Shimla experienced a consistent surge in domestic tourism. This growth was propelled by the growing preference for domestic travel, the emergence of budget-friendly airlines, and expanded travel routes
- .The COVID-19 pandemic significantly impacted tourism in Shimla. Travel restrictions, lockdown measures resulted in this downfall.

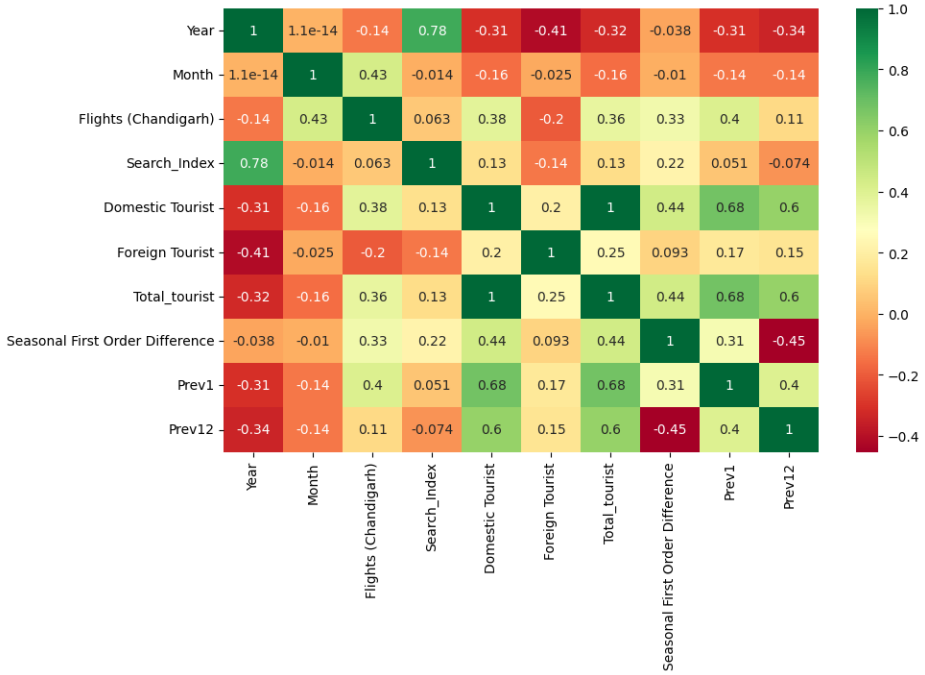
- The most popular months for domestic tourism in Shimla are May and June. This is likely because the **weather is pleasant during these months**, and there are a number of festivals held during this time.

- The least popular months for domestic tourism in Shimla are December and January. This is likely because **the weather is cold during these months**, and there are fewer festivals and events held in Shimla during this time.



D. Correlation Analysis

- The correlation between the Foreign Tourists and Domestic Tourists is 0.2 (which is very low) and goes to show that the two parts of the tourism sector in Shimla are **fairly independent.**
- The correlation between Domestic Tourists and Total Tourists(Domestic+Foreign) is 1, because Domestic tourists are much more in number w.r.t Foreign Tourists so their addition

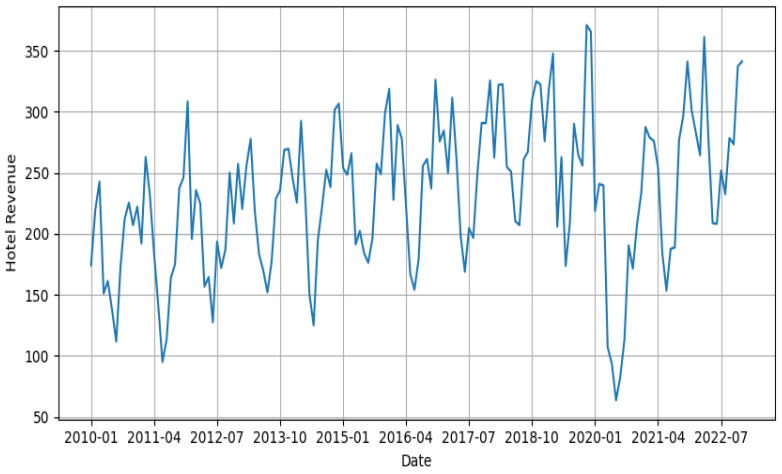


does not make significant contribution to Domestic .

KERALA:

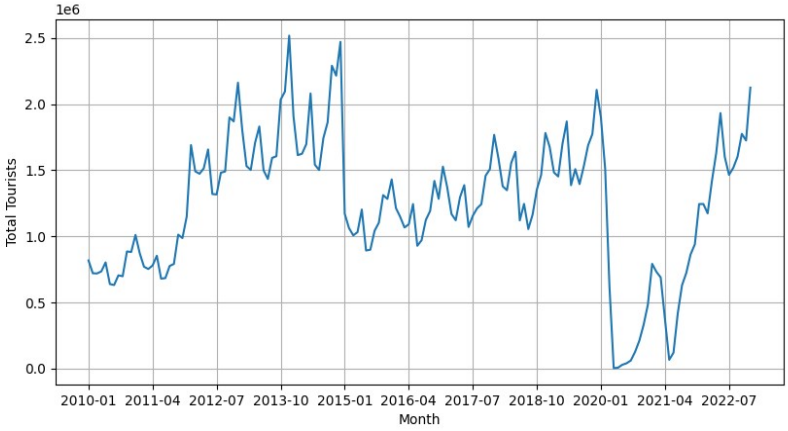
A. Hotel Revenue

From the given graph it can be concluded that every year during a certain time of the year there's a peak in the number of tourist arrivals which results in an increase in hotel revenue for that time of the year. However it can be noticed that during the period 2020-2021 there's a sharp fall in the hotel revenues which can be owed to the covid-pandemic that happened during that time.



B. Tourist Arrivals

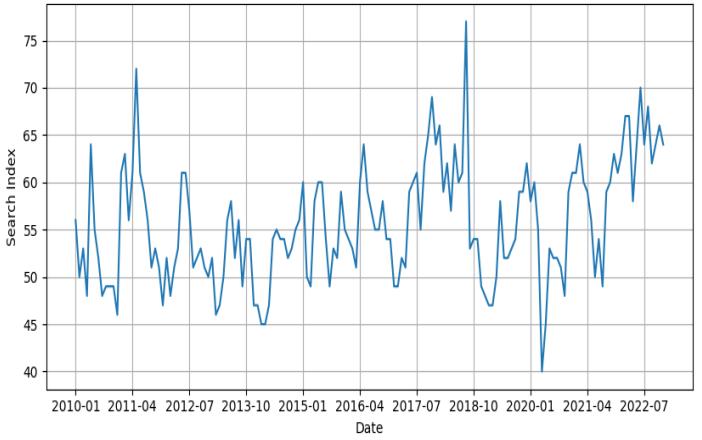
From the given graph it can be seen that during the period 2010 to 2015 Kerala saw a steady increase in Tourism with Yearly peaks, but suddenly Tourism declined in 2015 suggesting some Negative factors/events in Kerala, after that from 2016 to 2020 starting there was a steady increase but again after that Kerala Tourism hit a brick wall due to covid-19 pandemic, but soon started recovering in terms of tourism.



C. Search Index

From the graph the following can be inferred:

- The spike in search results for Kerala in 2018 was due to the Kerala floods, as people sought information about the disaster and ways to help.
- The decline in search results for Kerala in 2020-2021 can be attributed to the COVID-19 pandemic, which reduced interest in travel as people stayed indoors.



D. Correlation Analysis(Heatmap)

It is obvious that if we want to find the correlation between two same graphs we will have the correlation as 1, which is what we observe here . We have three different features which are **Search Index, Tourist Arrivals, Hotel Revenue** (3 features). So we will have 3 different combinations.

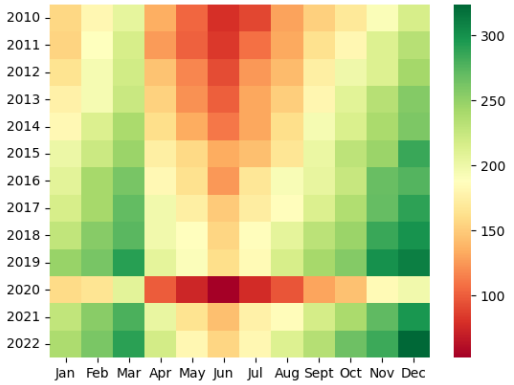
- **Search Index** vs **Tourist Arrivals**
  Correlation is **0.2** which is *relatively low*.

- **Tourist Arrivals** vs **Hotel revenue**
  Correlation is **0.45** which is *good*.

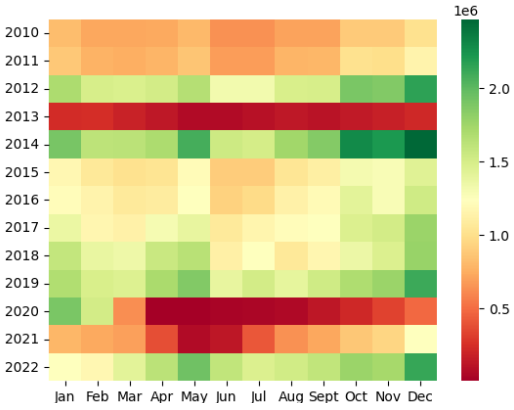
- **Search Index** vs **Hotel Revenue**
  Correlation is **0.12** which is *very low*.

E. Month Wise Analysis

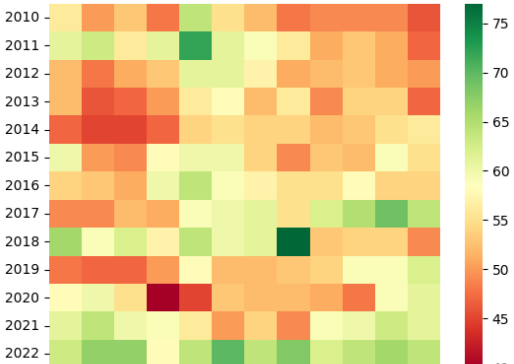
Hotel Revenue-
From the heat map of Hotel Revenue we can clearly infer that more revenue was generated in Shimla during the winter (Jan to Mar) and (Nov to Dec). Also the very prominent red region for the year 2020 clearly suggests a drop in revenue due to Covid-19.



Tourist Arrivals-
Here we clearly observe two distinct red lines for the years 2013 and 2020-2021. In the first part the drop for The year **2013** was probably caused due to **2013 North India Floods** which severely affected Shimla, while in 2020-21 the drop was due to **Covid-19**.



Search Index-
Searches for Shimla were relatively less in the years before 2020. 2020 saw a slight decrease in searches (2020 April and 2020 May) but after Covid-19, it recovered and 2022 saw a good amount of searches.

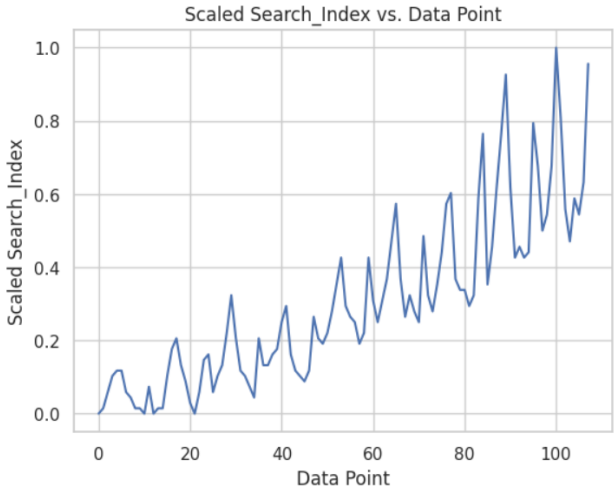


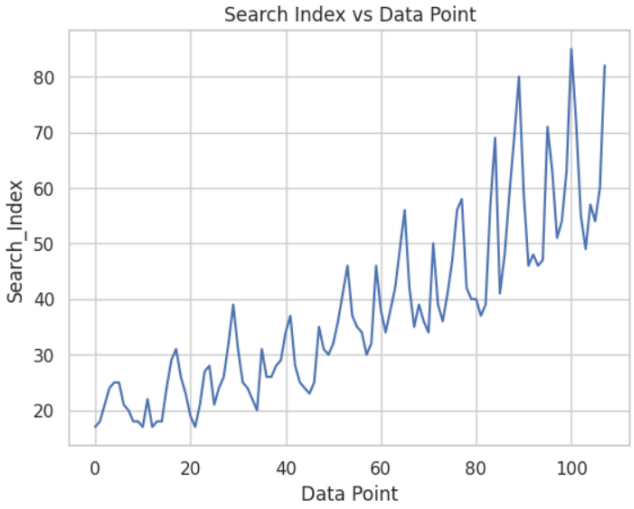
Feature Engineering

Feature Engineering is a fundamental process in data preprocessing and machine learning, where raw data is transformed and manipulated to create informative, predictive features. This artful practice involves scaling, dimensionality reduction, and creating new variables. Effective Feature Engineering is the cornerstone of building accurate, robust machine learning models, allowing them to extract meaningful patterns from data and make informed predictions or classifications.

Scaling: In Feature Engineering, Scaling, specifically Min-Max Scaling), standardizes numerical attributes. It transforms data into a consistent range, ensuring each feature contributes proportionately to the model's performance.

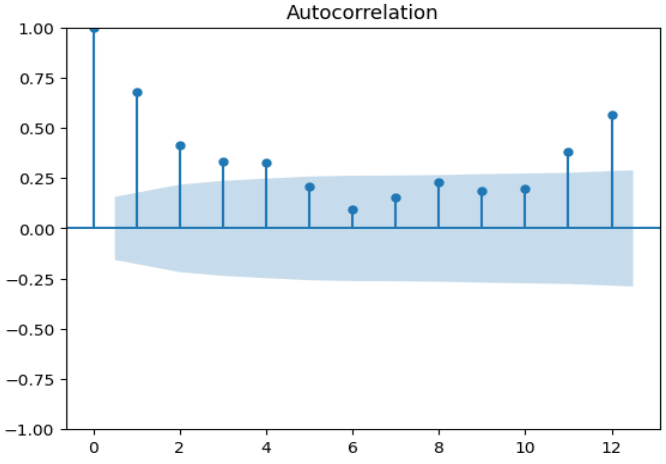
Example:



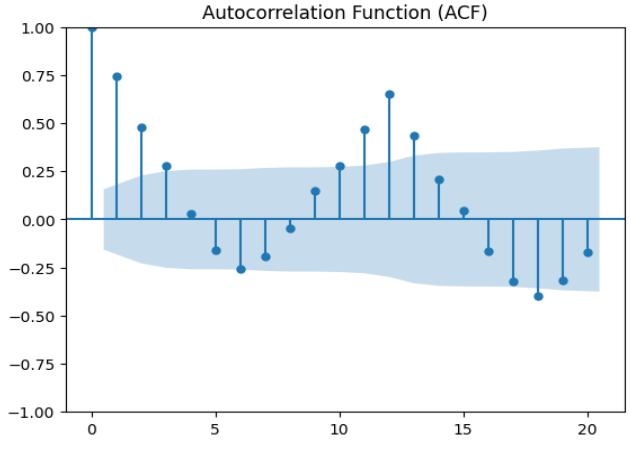


Feature Creation: Leveraging tools like ACF plots, we introduced new attributes by shifting data according to ACF values. This strategic approach improves a model's capacity to capture complex temporal relationships and enhances prediction accuracy.
In our case, we introduced 2 new features according to this shift.

ACF values of **Domestic Tourists** for **Shimla**.



ACF values of **Hotel Revenue** for **Goa**.



NaN Imputation using KNN

k-nearest neighbors imputation (kNN imputation) is a widely used technique for handling missing data in datasets. It operates on the principle of identifying the k data points that are most similar to the one with missing values, usually measured using a distance metric like Euclidean distance. The missing values are then imputed by using a weighted average of the corresponding values from these neighbouring points.
kNN imputation is advantageous for its non-parametric approach, avoiding assumptions about the data distribution. It works well for datasets with randomly scattered missing values. However, choosing an appropriate value for k, the number of nearest neighbors, is crucial for accurate imputation and computational efficiency. Effective handling of both categorical and continuous features is also important to ensure meaningful imputation, especially in datasets with diverse data types.

Distance Metric Used: The distance metric used in this analysis is the weighted Euclidean Distance, which, for two feature vectors $\mathbf{x_1} = (x_1^{(1)}, x_1^{(2)}, \dots, x_1^{(n)})$ and $\mathbf{x_2} = (x_2^{(1)}, x_2^{(2)}, \dots, x_2^{(n)})$, where some $x_1^{(i)}$ and $x_2^{(j)}$ are absent, is given by

$$d_{WE}(\mathbf{x_1}, \mathbf{x_2}) = \sqrt{w\, d_E^{\,2}},$$

where d_E^2 is the squared Euclidean distance between the features that are present in BOTH of the vectors, and

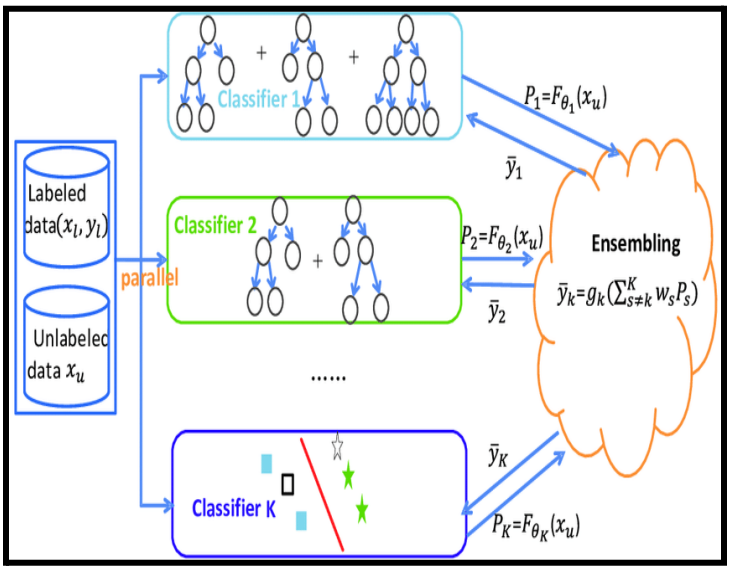
$$w = \frac{Total\ number\ of\ features\ present\ in\ BOTH\ vectors}{Total\ number\ of\ features\ (n)}.$$

Some of the key features used are,

1) **Internet search index (Google):** This illustrates the level of popularity a specific location experiences during a particular season.
2) **Foreign Tourist Arrival Data:** It indicates the extent of popularity of the location among international visitors during a specific time of the year.
3) **Domestic Tourist Arrival Data:** It indicates the extent of popularity of the location among Indian nationals during a specific time of the year.
4) **No. of Flight Booking:** Flight booking data is vital for predicting tourist behavior, offering real-time insights on travel intent, seasonal trends, demographics, and economic indicators, improving the model's accuracy.
5) **Crime Rate:** Crime rate data is essential for estimating tourists as it influences safety, risk assessment, behavioral insights, destination recommendations, and crisis management, enhancing prediction accuracy.

GOA

Forecasting Models:



Extreme Gradient Boosting(XGBoost)

Theory

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements machine learning algorithms under the Gradient Boosting framework. It is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

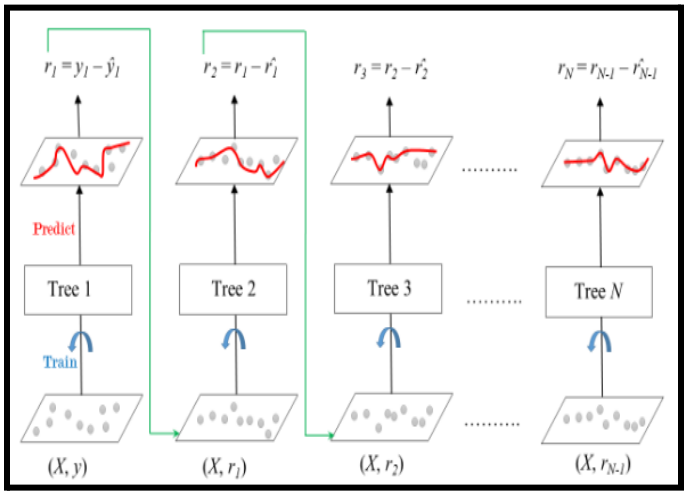
Why XGBoost?

It can capture the nuances of the relationships between features, especially when you have observed seasonality and trends over time. Its ability to handle complex, non-linear patterns in data and missing values. Also, it provides the benefit of capturing much more complex relationships between the data points without having to perform difficult transformations on our own. We have used our kernel as the Radial Basis Function for this purpose. This algorithm is accurate yet less prone to overfitting.

Gradient Boosting

Theory

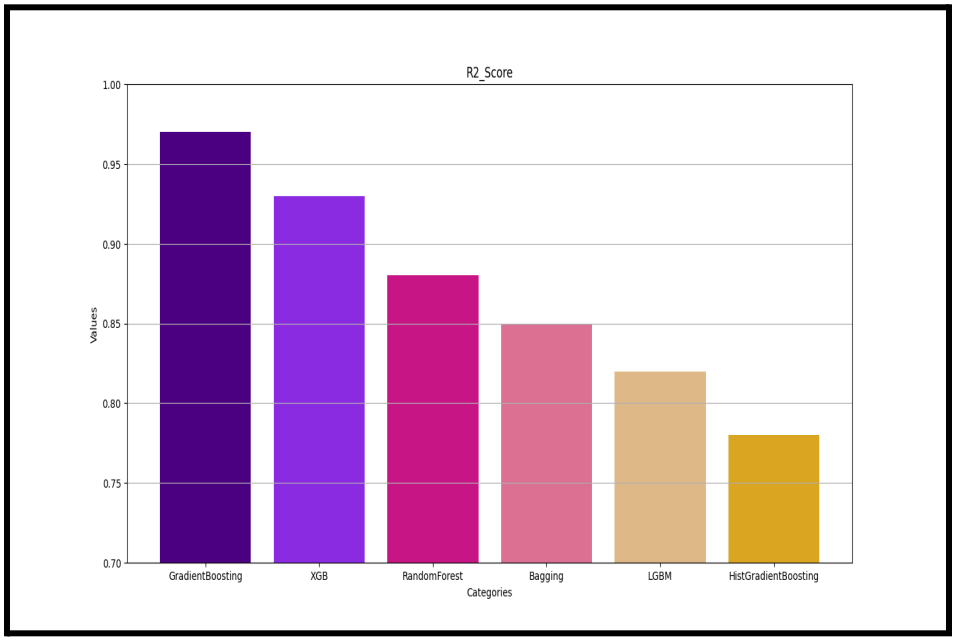
Gradient Boosting is a powerful boosting algorithm that combines several weak learners into strong learners, in which each new model is trained to minimize the loss function using gradient descent. In each iteration, the algorithm computes the gradient of the loss function with respect to the predictions of the current ensemble and then trains a new weak model to minimize this gradient. The predictions of the new model are then added to the ensemble, and the process is repeated until a stopping criterion is met.



Why Gradient Boosting?

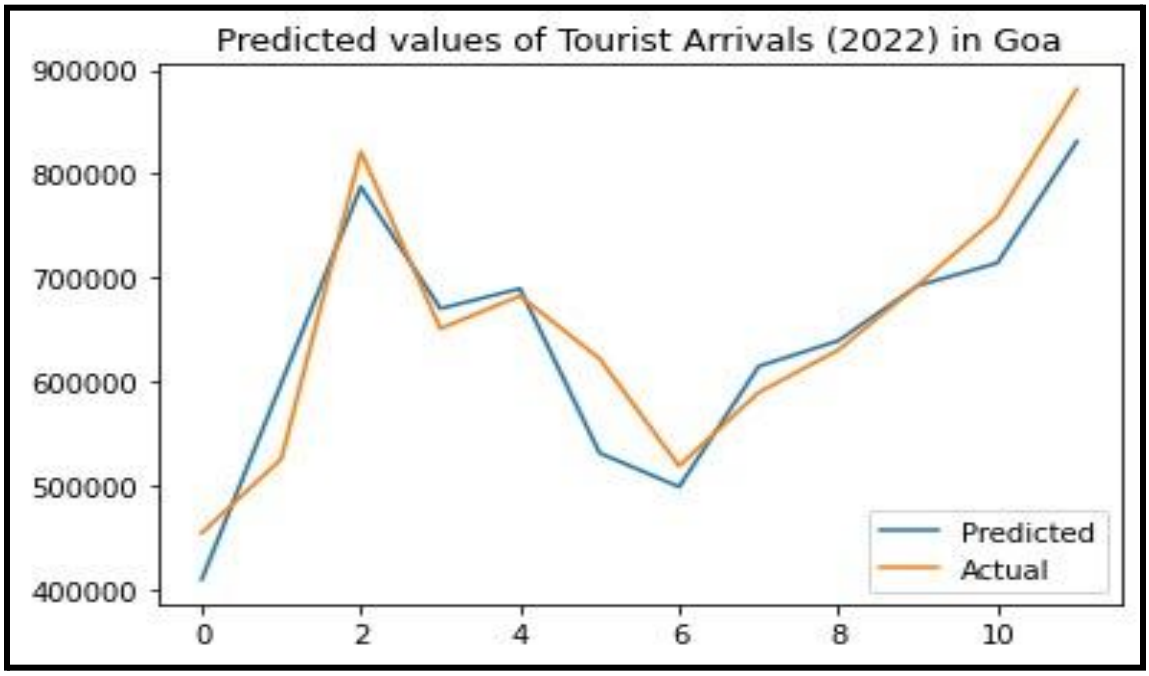
Gradient Boosting performs here better for its technical advantages. It's an ensemble learning method that minimizes loss by iteratively adding decision trees, optimizing through gradient descent. This approach, based on weak learners, provides exceptional predictive accuracy. Its robustness in handling outliers, ability to capture intricate data patterns, and suitability for various data types make it a popular choice.

RESULTS:



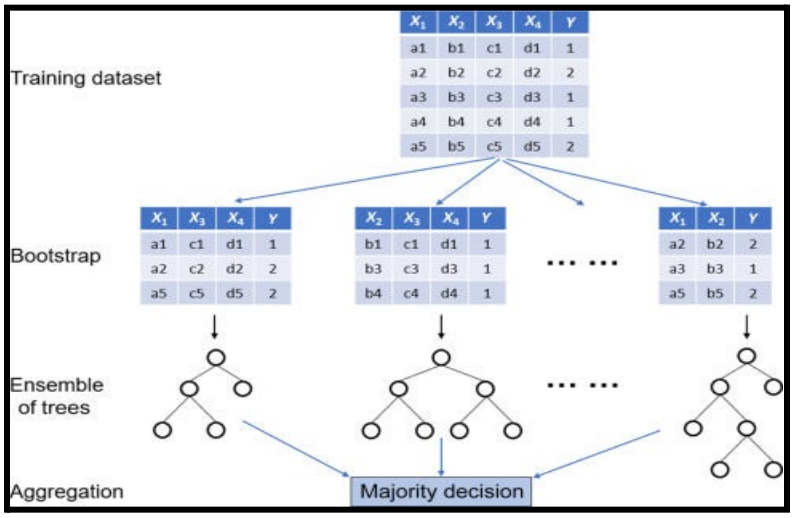
The best performing models for Forecasting Tourist Arrival for Goa are Gradient Boosting Algorithm and XGBoost Regressor with a R2-Score of **0.97** and **0.93** respectively.

COMPARISON PLOT BETWEEN PREDICTED AND ACTUAL VALUES:



SHIMLA

Forecasting Models:



Extra Trees Regressor

Theory

Extra trees regressor is a type of ensemble learning technique that uses randomized decision trees to improve predictive accuracy and control over-fitting. The algorithm works by creating a large number of fitted trees from the training dataset and then averaging the predictions of these decision trees. It is faster than Random Forest since it chooses the splitting node randomly and not the optimal one.

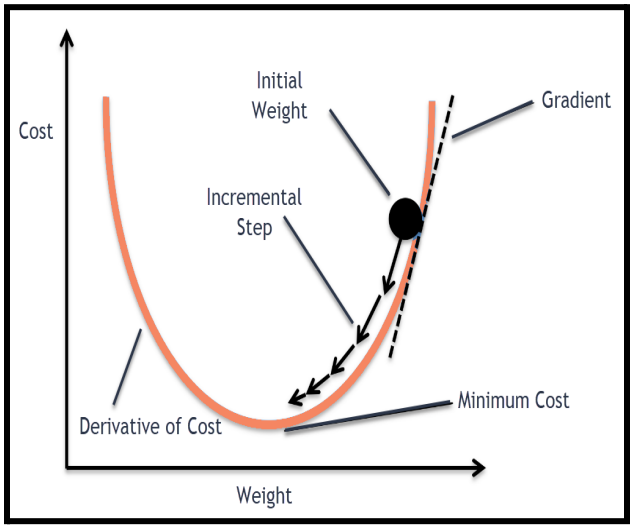
Why Extra Trees Regressor?

Extra Trees Regressor works with a lot of randomization at each step. This makes it robust to noise and outliers in the data. Also extra trees has the ability to capture non-linearity in data due to its ability to create deep and diverse decision trees and ensemble them to get the final prediction.

Stochastic Gradient Descent Regressor

Theory

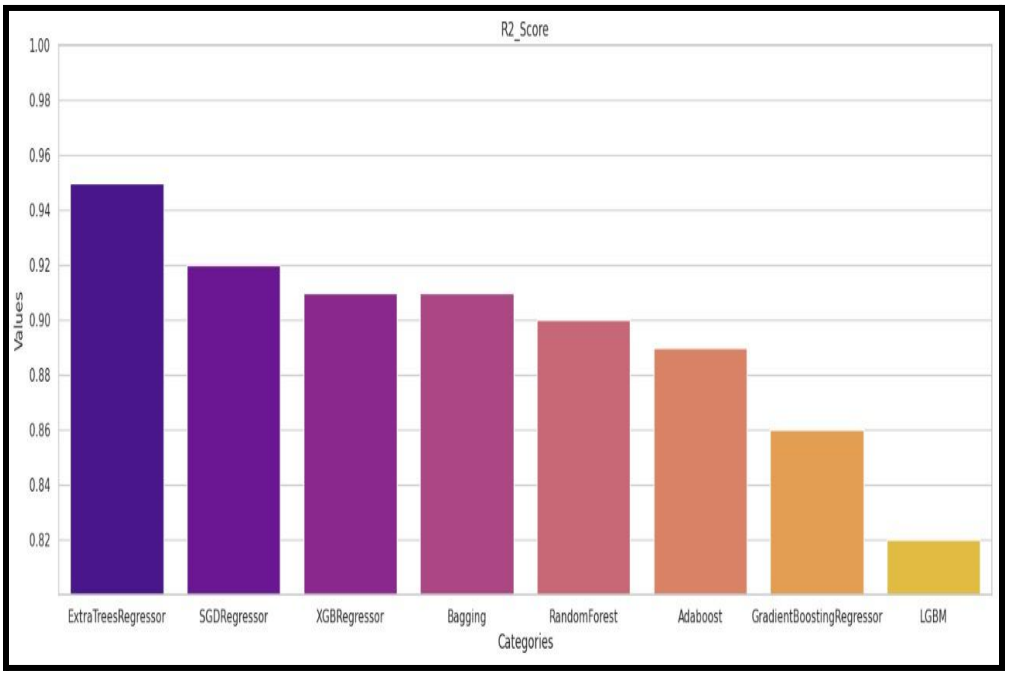
Stochastic Gradient Descent Regression works similar to regularized linear regression except that it uses Stochastic Gradient Descent as its optimization algorithm. With Stochastic Gradient Descent, the values of the different parameters (weights and biases) are updated after going through each training example. This method is comparatively faster than Batch Gradient Descent and is especially suitable to scenarios where the model must learn as soon as new data becomes available to the model.



Why SGD Regressor?

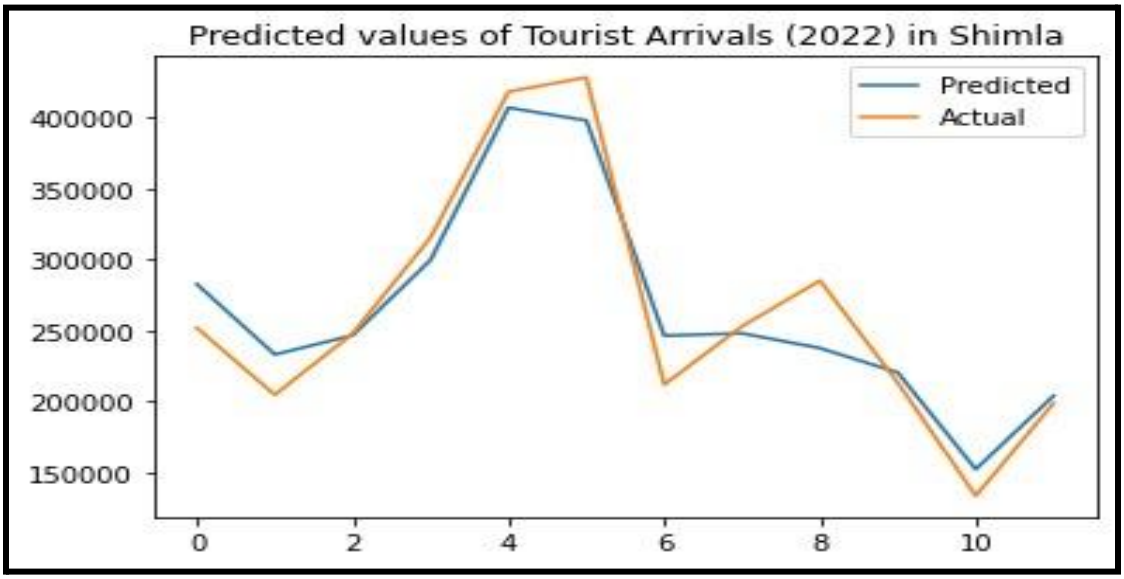
The Stochastic Gradient Descent regressor works well with scaled features. This can be attributed to the fact that the parameters are updated one training example at a time, feature-by-feature, due to which scaling of a particular feature does not affect it much. Another huge positive of the SGD Regressor is that it can work with customized loss functions which are used sometimes to handle time series data.

RESULTS:



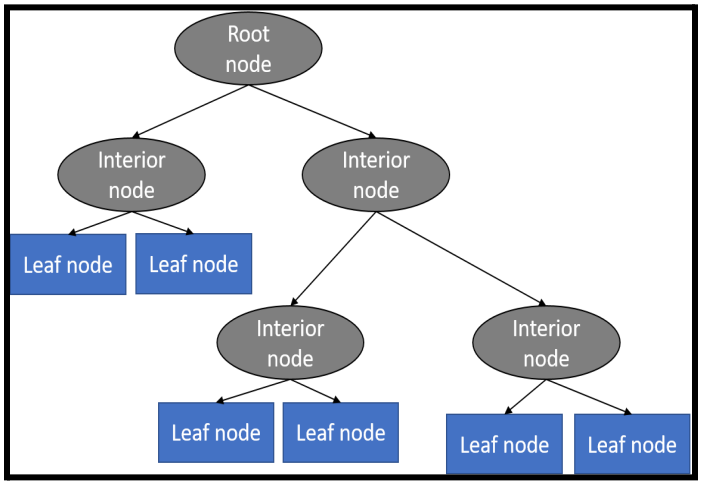
The best performing models for Forecasting Tourist Arrival for Shimla are Gradient Boosting Algorithm and Extra Trees Regressor with a SGD Regressor of **0.95** and **0.92** respectively.

COMPARISON PLOT BETWEEN PREDICTED AND ACTUAL VALUES



KERALA

Forecasting Models:



Decision Tree Regressor

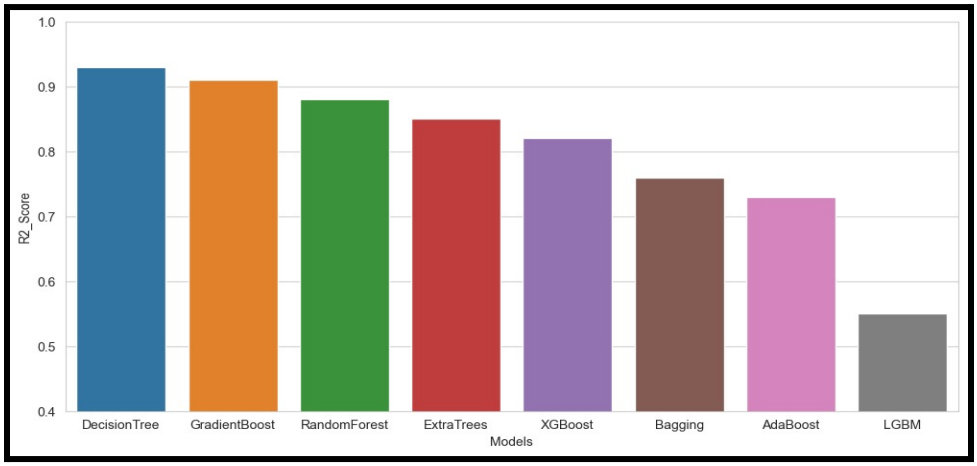
A Decision Tree is a fundamental machine learning algorithm that builds a tree-like structure to make decisions. It partitions data into subsets at each node by selecting the feature that minimizes a splitting criterion. The tree continues to split, recursively refining decision boundaries, until a predefined stopping criterion is met, such as a maximum tree depth or a minimum number of samples in a leaf. Techniques like pruning and limiting the tree depth are employed to mitigate overfitting.

Why Decision Tree Regressor?

Decision Tree Regressor gives the best results for our dataset because it can handle both categorical and numerical data, which are present in our dataset. It can also handle non-linear relationships between the input features and the target variable, which may be present in our dataset. Additionally, Decision Tree Regressor is easy to interpret and can provide insights into the importance of each input feature for predicting the target variable.

RESULTS:

The best performing models for Forecasting Tourist Arrival for Kerala is Decision-Tree Regressor and Gradient Boosting Algorithm with a R2 a score of **0.93** and **0.91** respectively.



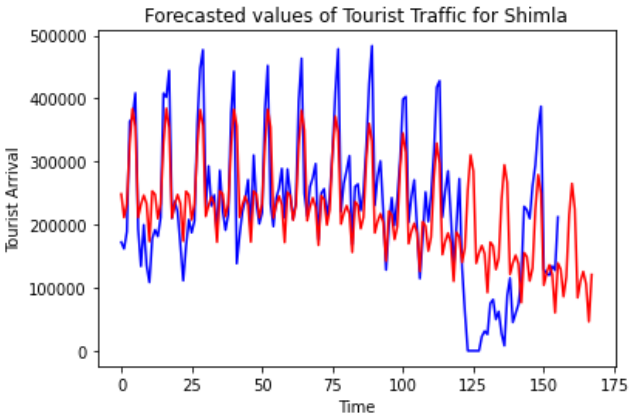
Prophet:

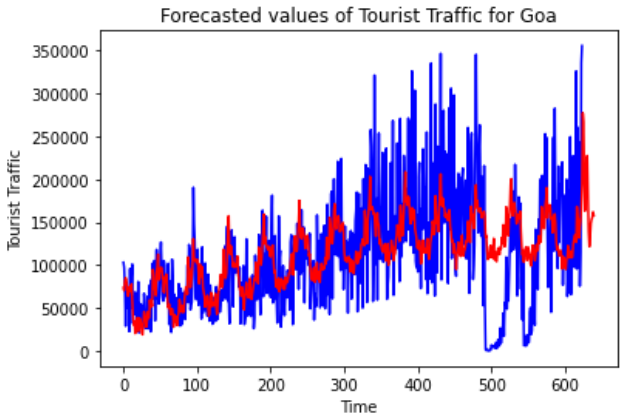
Prophet is a time series forecasting model developed by Meta that offers several compelling reasons for its use in various applications. It excels in handling data with strong seasonal patterns, holiday effects, and missing values. It automatically detects and models changepoints in data, and it estimates uncertainties in forecasts, aiding in decision-making.

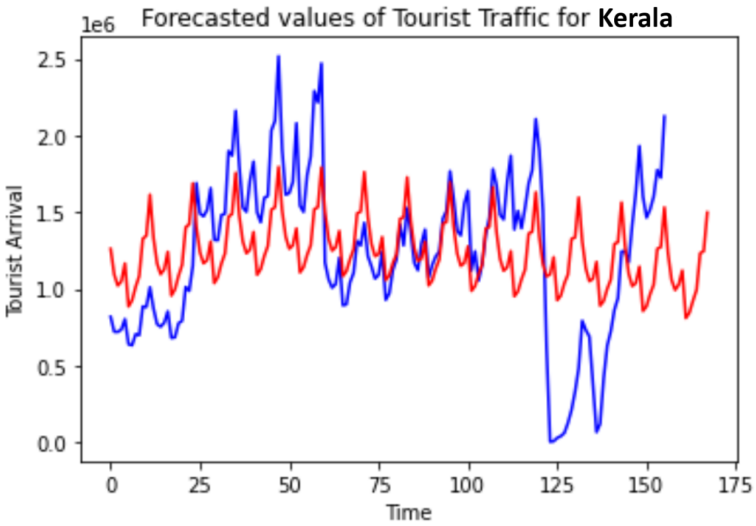
Why Prophet?

Prophet is versatile and can handle a wide range of time series forecasting tasks. It is designed to work well with data that has strong seasonal patterns and multiple sources of uncertainty, which is common in our case. It includes the ability to account for holidays and special events, which is crucial when forecasting tourist arrivals since these destinations often experience fluctuations due to holidays, festivals, and other events.

Results:







ANNEXURE

ANNEXURE 1 (K-Nearest Neighbors[KNN])

Introduction:

K-Nearest Neighbors (KNN) is a simple yet powerful machine learning algorithm used for both classification and regression tasks. In this annexure, we will delve into the key concepts and practical aspects of KNN, providing a comprehensive overview for a better understanding.

How KNN Works?

KNN is an instance-based learning algorithm, meaning it makes predictions based on the proximity of data points in a feature space. The fundamental idea behind KNN can be summarized as follows:

- For a given data point, KNN identifies the K-nearest data points (neighbors) from the training dataset.
- For classification, it assigns the class label that is most common among the K-nearest neighbors.
- For regression, it calculates a weighted average of the target values of the K-nearest neighbors.

Key Parameters:

a. ***K (Number of Neighbors):*** Choosing an appropriate value for K is crucial. A smaller K makes the model sensitive to noise, while a larger K can lead to oversmoothing.

b. ***Distance Metric:*** Euclidean distance is commonly used, but other distance metrics like Manhattan, Minkowski, or custom metrics can be employed based on the problem.

Advantages:

- KNN is easy to understand and implement.
- It can be used for both classification and regression tasks.
- It doesn't require model training; it memorizes the training data.

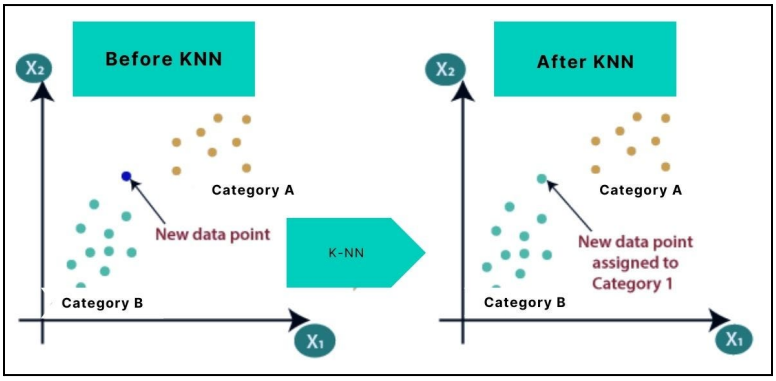
Limitations:

- KNN can be computationally expensive, especially with large datasets.
- It's sensitive to the choice of K and the distance metric.
- Handling imbalanced datasets can be challenging.

Use Cases:

KNN can be applied in various domains, like:

- Recommender systems
- Anomaly detection
- Image classification
- Healthcare for disease diagnosis
- Predictive maintenance in manufacturing



Practical Implementations:

- Preprocessing data: Feature scaling and handling missing values are essential.
- Choosing the right value of K through techniques like cross-validation.
- Evaluating the model's performance using metrics like accuracy, F1-score, or mean squared error for regression.

ANNEXURE 2 (XGBoost Algorithm)

Introduction:

XGBoost, short for Extreme Gradient Boosting, is a high-performance and widely used machine learning algorithm known for its exceptional predictive accuracy. In this annexure, we will explore the core concepts, features, and practical applications of XGBoost.

What is XGBoost?

XGBoost, short for Extreme Gradient Boosting, is a high-performance and widely used machine learning algorithm known for its exceptional predictive accuracy. In this annexure, we will explore the core concepts, features, and practical applications of XGBoost.

Key Features:

a. ***Gradient Boosting:*** XGBoost employs a boosting process where it combines the predictions of multiple weak learners (usually decision trees) to create a strong learner.
b. ***Regularization:*** It incorporates L1 (Lasso) and L2 (Ridge) regularization techniques to prevent overfitting and improve generalization.
c. ***Parallel Processing:*** XGBoost is optimized for efficient parallel and distributed computing, making it suitable for large datasets.

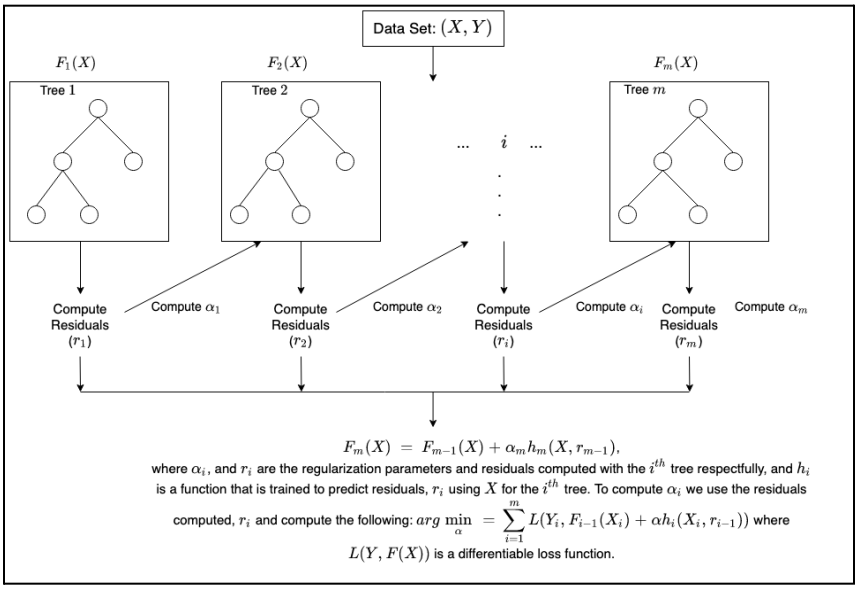
Use Cases:

XGBoost is commonly employed in various domains and applications, including:

- Kaggle competitions and data science challenges.
- Financial forecasting and stock price prediction.
- Healthcare for disease diagnosis and prognosis.
- Anomaly detection in cybersecurity.

Practical Implementations:

- Preprocessing data: Feature engineering, handling missing values, and encoding categorical variables.
- Setting up a robust validation strategy, like k-fold cross-validation.
- Monitoring the model's performance using evaluation metrics such as accuracy, AUC-ROC, or Mean Squared Error (MSE).



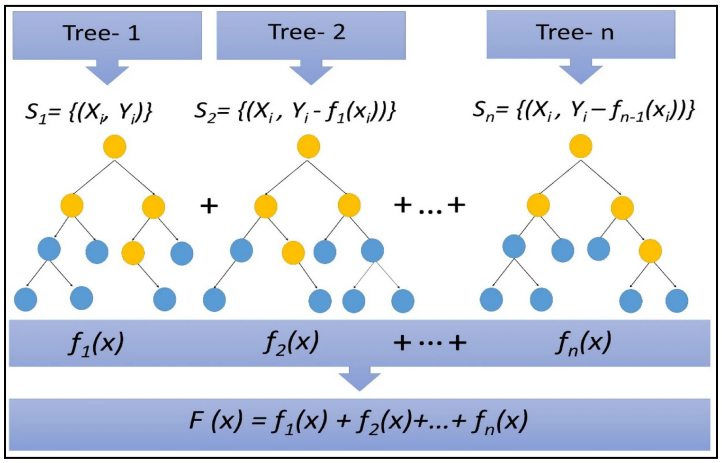
ANNEXURE 3 (Gradient Boosting Algorithm)

Introduction:

Gradient Boosting is a powerful ensemble learning technique that combines the predictions of multiple weak learners, usually decision trees, to create a strong predictive model.

Key Concepts:

a. **Weak Learners:** Decision trees are commonly used as weak learners with limited depth.
b. **Gradient Descent:** Gradient Boosting minimizes a loss function by iteratively adjusting parameters using gradient descent.
c. **Ensemble Learning**: It aggregates predictions to create an accurate model.



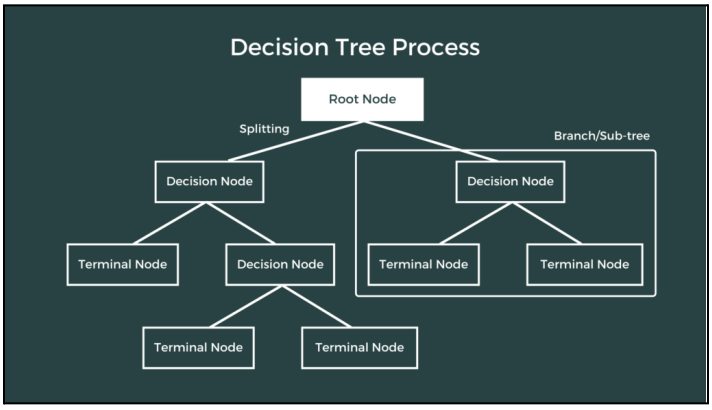
ANNEXURE 4 (Decision Tree)

Introduction:

A Decision Tree Regressor resembles an inverted tree, where each node represents a decision based on a feature, and each branch represents a possible outcome. The tree structure is constructed by recursively partitioning the dataset into subsets based on the values of the features.

Key Concepts:

a. ***Nodes:*** Decision Trees consist of nodes that represent features or attributes. The root node is the initial feature, and internal nodes represent decisions based on features. Leaf nodes are the final outcomes.
b. ***Splitting:*** Decision Trees make decisions by splitting data into subsets based on the values of specific features.
c. ***Criteria for Splitting:*** Various criteria, such as Gini impurity for classification and mean squared error for regression, are used to determine the best feature to split on.



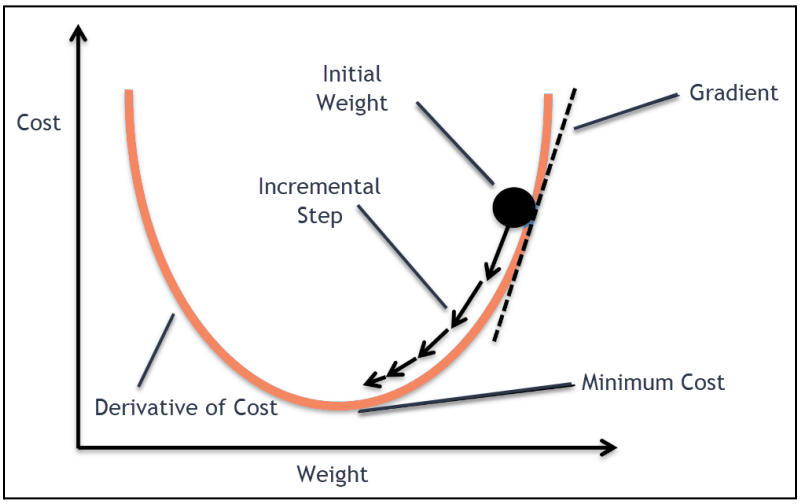
ANNEXURE 5 (Stochastic Gradient Descent[SGD])

Introduction:

Stochastic Gradient Descent (SGD) is a key optimization algorithm in machine learning. It iteratively adjusts model parameters to minimize a cost function. This annexure explores the essential aspects of SGD.

Key Concepts:

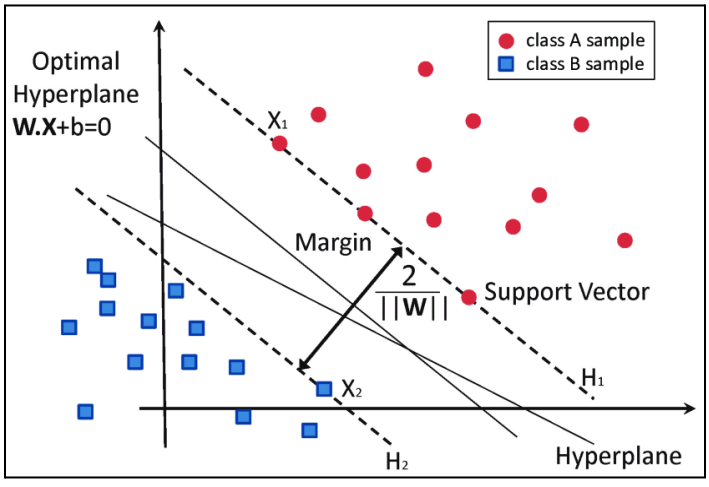
- **Objective Function:** SGD minimizes a loss function.
- **Learning Rate:** Determines step size during parameter updates.
- **Batch Size:** SGD can process data one example at a time (stochastic), in small batches, or the entire dataset.



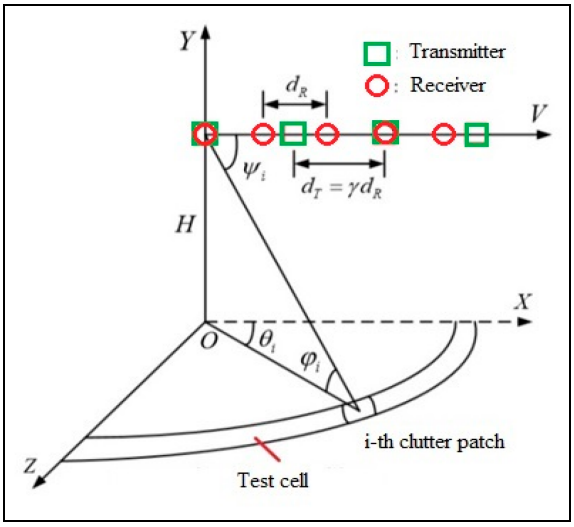
ANNEXURE 6 (Support Vector Regression[SVR])

Introduction:

SVR is a supervised learning algorithm that analyzes data for regression analysis. It works by finding a hyperplane that best represents the data points, and the data points closest to the hyperplane are known as support vectors.



ANNEXURE 7 (Orthogonal Matching Pursuit)

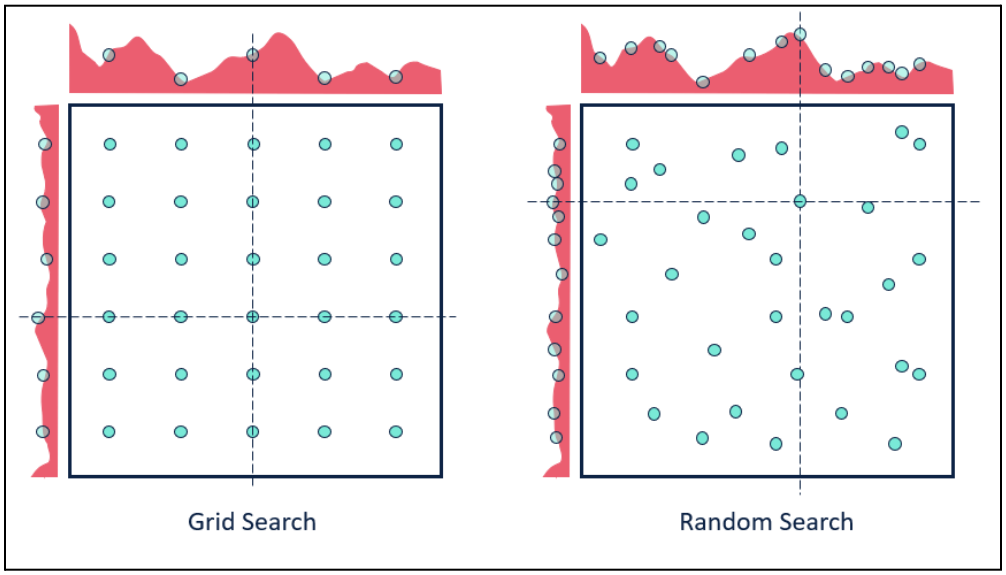


Introduction:

Orthogonal Matching Pursuit (OMP) is a greedy algorithm used in signal processing and machine learning for feature selection and signal reconstruction. It sequentially selects the most informative features (columns) of a matrix to represent a signal or data point while maintaining orthogonality between the selected features. OMP aims to find a sparse representation of data by iteratively approximating the signal as a linear combination of a few chosen features, making it useful for applications like compressive sensing and dimensionality reduction.

ANNEXURE 8 (Grid Search Parameter)

Introduction:



Grid search is a hyperparameter optimization technique used to find the optimal hyperparameters of a model that results in the most accurate predictions. It is an exhaustive search algorithm that searches through a manually specified subset of the hyperparameter space of a learning algorithm.

Grid search builds a model for every combination of hyperparameters specified and evaluates each model. The objective function takes a tuple of hyperparameters and returns the associated loss. Cross-validation is often used to estimate this generalization performance and choose the set of values for hyperparameters that maximize it.

ANNEXURE 9 (Hyperparameter Tuning)

Introduction:

Hyperparameter tuning is the process of selecting the optimal set of hyperparameters for a machine learning model. Hyperparameters are external configuration variables that data scientists use to manage machine learning model training.

The aim of hyperparameter tuning is to find the best combination of hyperparameters that results in the most accurate predictions

There are several methods of hyperparameter tuning :

- Grid-Search
- Random-Search
- Bayesian Optimization
- Hyperband
- Genetic Algorithms