

The background features a light blue gradient with several geometric elements: a dark blue triangle at the top left, a yellow triangle at the top left, a yellow square on the left, a large yellow triangle at the bottom left, and a cluster of overlapping blue and yellow squares at the bottom right. Four blue starburst shapes are scattered across the background.

# OPEN IIT DATA ANALYTICS

D2

# Introduction

“The world is a book and those who do not travel read only one page.”

- by Saint Augustine

India, with its diverse cultures, rich heritage, and stunning natural beauty, holds a remarkable place in the world of tourism. According to a report by India Brand Equity Foundation (IBEF), in 2022, the travel & tourism industry's contribution to the GDP is estimated to be over US\$ 215 billion; this is expected to reach US\$ 488 billion by 2029. In India, the industry's direct contribution to the GDP is expected to record an annual growth rate of 7-9% between 2019 and 2030. Thus, forecasting tourist numbers is gaining significance in predicting future economic growth. The practice of forecasting tourism demand can furnish essential data for future planning and policy making endeavors. With the internet seeping deeper into our lives, internet search queries that reflect user behaviors, in addition to the government's basic visitor statistics, can be used to make tourism forecasting models more robust and scalable.

## Problem Statement

Create a predictive machine learning model using Internet search index data to forecast tourist arrivals at a specific destination. The objective is to provide accurate predictions for tourism authorities and businesses to enhance resource allocation, marketing strategies, and destination management.

## Dataset Description

Dataset for the Problem statement was scrapped from the tourism website of Himachal Pradesh. Search queries were scrapped from Google Search Engine and search volume for the queries was extracted from Google Trends. Dataset for forecasting number of tourist visiting Shimla is broadly classified into 3 parts:

- **Timestamp:** It indicates the dates.
- **Search index data:** It indicates the search volumes of the query's searched on google search index which by a tourist. Search queries were based on all aspects of tourism planning, including travel, traffic, lodging, dining, recreation and shopping.
- **Number of Tourists:** It shows the data of monthly number of tourist both foreign and Indian visiting Shimla from 2010 to 2022.

Root mean squared error(RMSE) and mean absolute error(MAE) were employed as metrics to assess the model's accuracy and correctness.

# Index

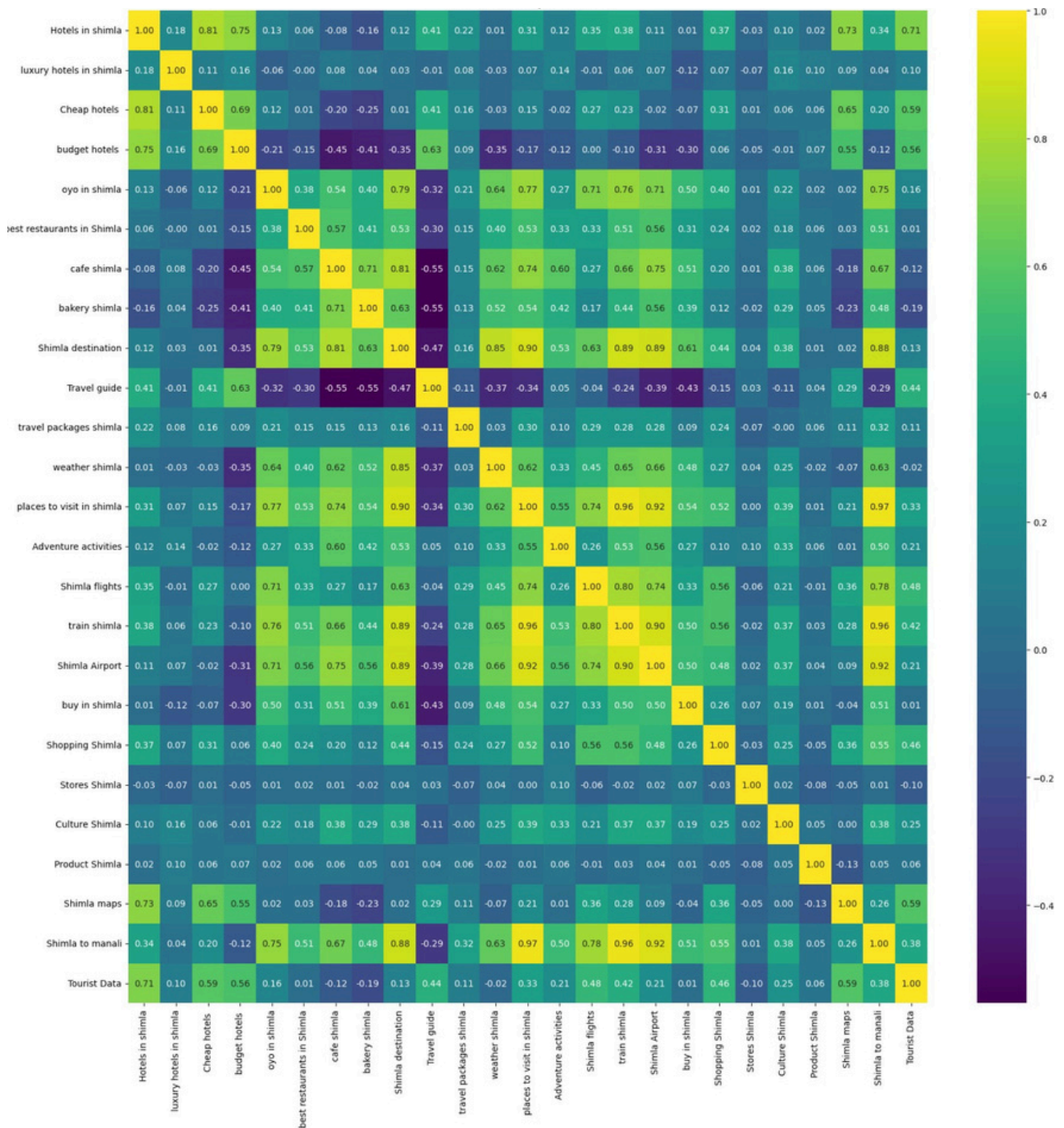
---

1. Introduction	(1)
2. Data Preprocessing	(2)
a. Web Scraping	(2)
b. Creation of training dataset	(2)
3. Feature Engineering	(3)
a. Heatmap	(3)
b. Granger Causality test	(4)
4. Data Visualization	(5)
5. Models and Approach	(8)
a. Artificial Neural Network (ANN)	(8)
b. Long Short Term Network (LSTM)	(9)
c. SARIMA	(10)
d. SARIMAX	(11)
e. CATBOOST	(12)
f. k-Nearest Neighbours (k-NN)	(12)
g. Exponential Smoothing	(13)
h. Weekly Interpolation Method	(14)
6. Results and Conclusion	(15)
7. Annexure	(16)

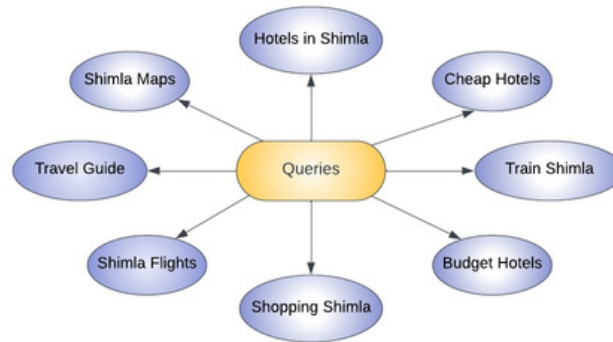


# Feature Engineering

## Heatmap



The above heatmap shows the number of tourists along with the search queries from google trends. Using the heatmap we inferred that the features having PCA greater than 0.5 would have more weightage thus allowing us to identify the prominent features and sort them out. The final queries are shown on the next page.



## Granger Causality Test

Granger causality is a statistical concept used to determine whether one time series can predict the future values of another time series. It measures the extent to which the past values of one variable provide valuable information for forecasting the other variable's future behavior. Granger causality helps analyze potential relationships between variables, aiding in predicting trends.

$$Y_t = \sum_{i=1}^n \alpha_i Y_{t-i} + \sum_{j=1}^n \beta_j X_{t-j} + u_{1t} \quad \sum \beta_j \neq 0$$

$$X_t = \sum_{i=1}^n \lambda_i Y_{t-i} + \sum_{j=1}^n \sigma_j X_{t-j} + u_{2t} \quad \sum \lambda_i \neq 0$$

A variable Y is causal for another variable X if knowledge of the past history of Y is useful for predicting the future state of X over and above knowledge of the past history of X itself. So if the prediction of X is improved by including Y as a predictor, then Y is said to be Granger causal for X.

For the Tourist data we performed Granger causality test of overall search index data from Google Trends with Tourist Volume and we got the following results:

Number of lags(monthly)	F Value	P Value
0	5.7846	0.0174
1	3.0911	0.0484
2	2.4792	0.0635
3	2.2582	0.0658

For lags= 0 and 1 Granger causality test gave positive results, the p value for these lags were less than 0.05 but from lags =2 the p value was greater than 0.05 showing that up-till 1 month lag data of search trends data can be used to forecast number of Tourist.



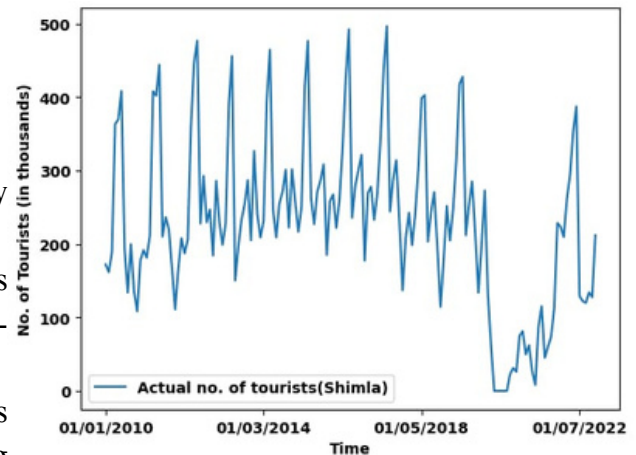
# Data Visualisation

## Actual No. of Tourists vs Timestamp:

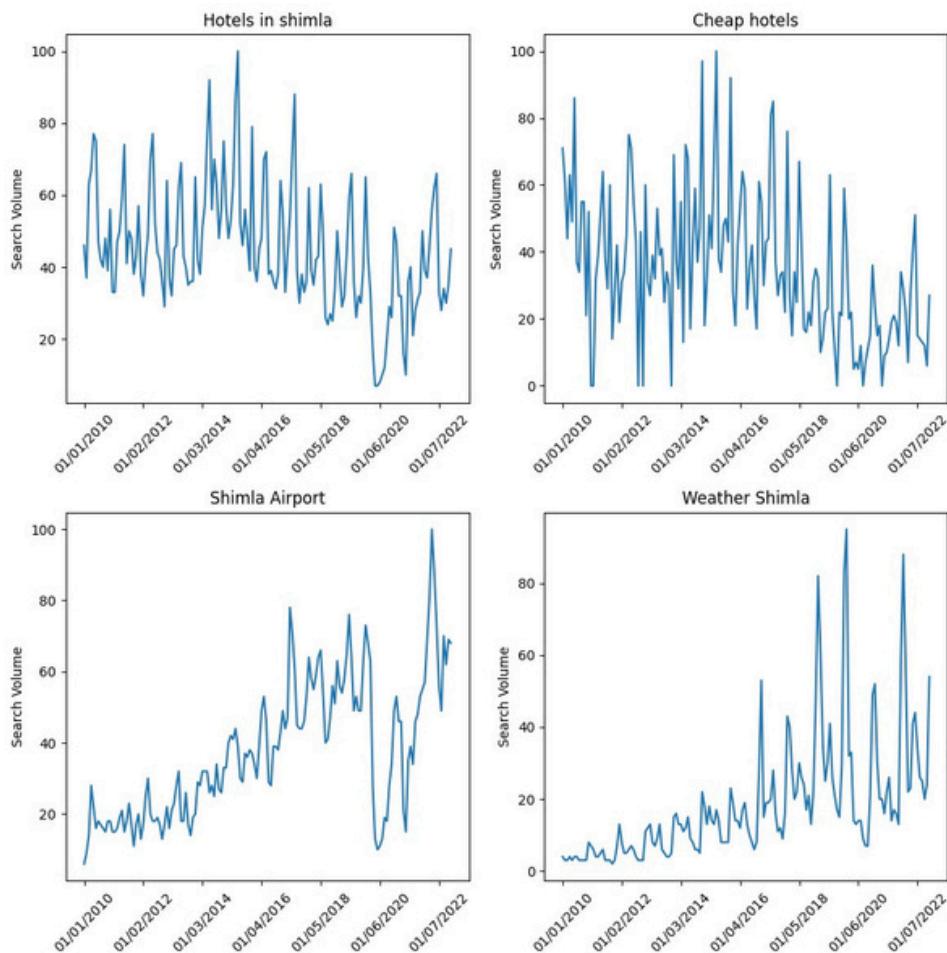
We plot the total actual no. of tourists vs the timestamp.  
The peaks in the plot denotes the rise in no. of tourists.

By observing the plot, we can infer that-

- In general, the no. of tourists is affected by seasonality (denoted by the peaks in the graph).
- There has been a significant dip in the no. of tourists visiting from 2020-2021 due to the effect of COVID-19.
- After 2021, the industry of tourism, although has not reached the pre-Covid trends, is at its growing stage.

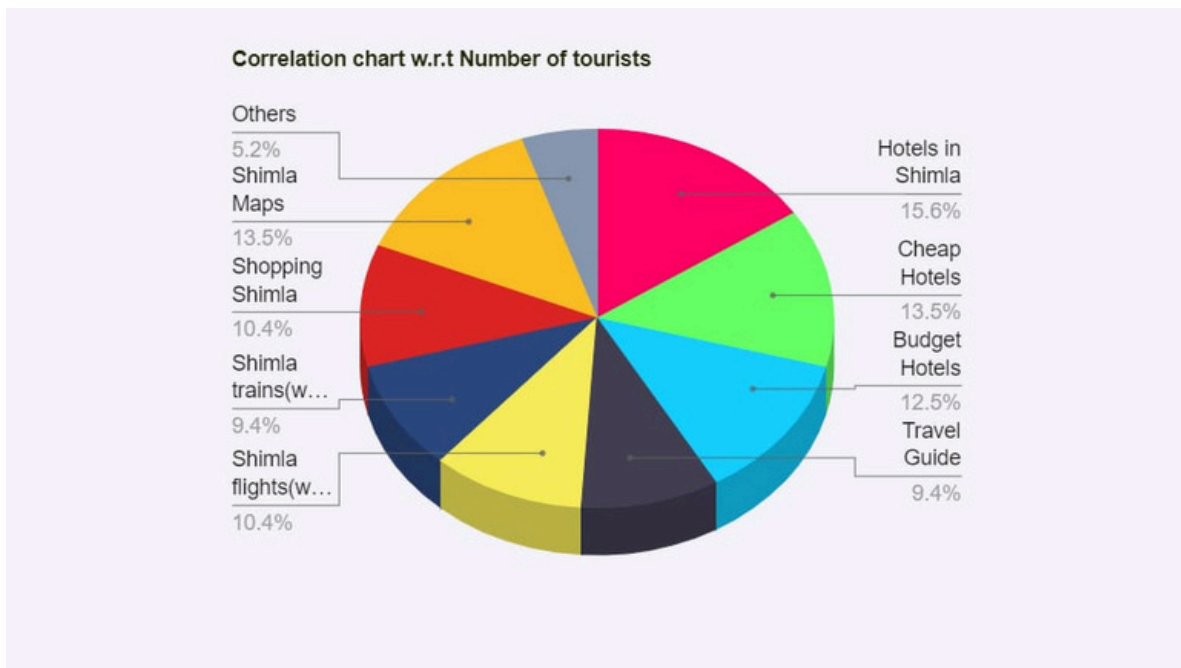


## Search Trends of highly correlated queries(with tourist data):



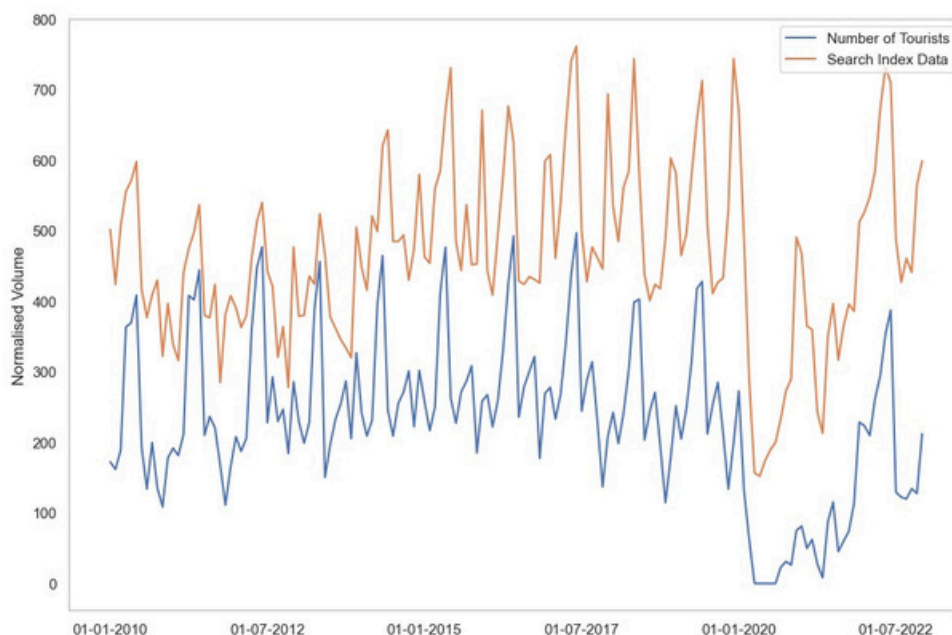
- These are the plots of search trends of best correlated features with the number of tourists .
- From the above plots, we can infer that the trends of these features match with the trends of the number of tourists visiting Shimla.

## Correlation chart w.r.t Number of Tourists



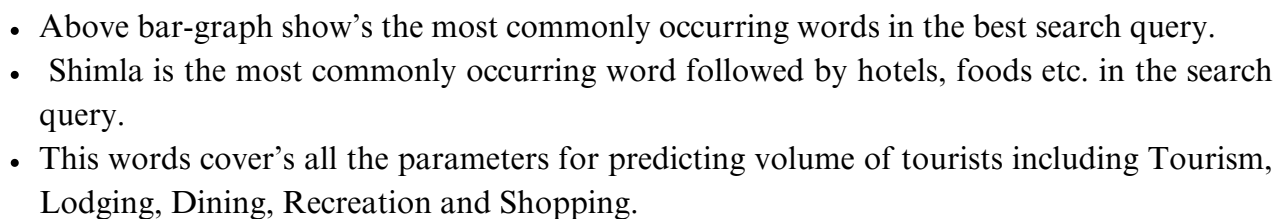
- Above Pie-plot shows how good trends of search query related with the number of tourists visiting Shimla.
- The higher the area of the sector it the more that search trend is converting into volume of tourist.

## Plot of search Index and Tourists volume data





- ## Frequency of words in search queries:



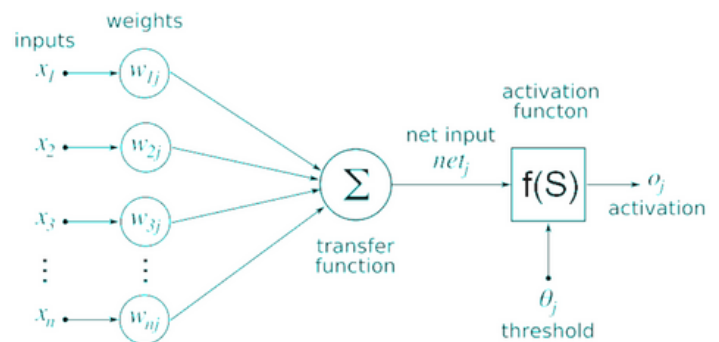
# Models and Approach

## 1. Artificial Neural Network (ANN)

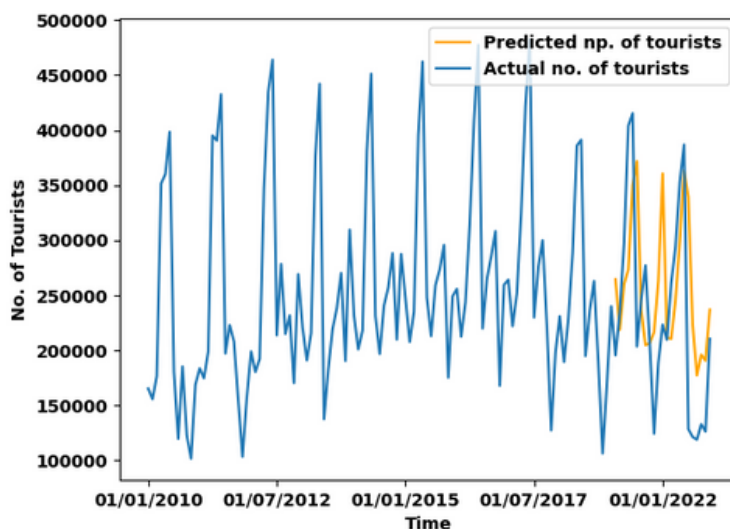
Artificial neural network (ANN) is a deep learning method that arose from the concept of the human brain biological networks. ANNs are built in a way to mimic the network of neurons in a human brain by a computer. Even though they work in an extremely similar manner, they are not identical.

The basic architecture of ANN consists of 3 layers:

- Input layer : which accepts numeric and structured data.
- Hidden layer(s) : which extracts some of the most relevant patterns from the inputs and sends them on to the next layer for further analysis.
- Output layer : which gives the final result that we are looking for.



For our problem the preprocessed vectorized dataset (containing the features including number of hotels in Shimla, flights, trains etc.) was given as an input to the ANN and the number of tourists were provided as the output.



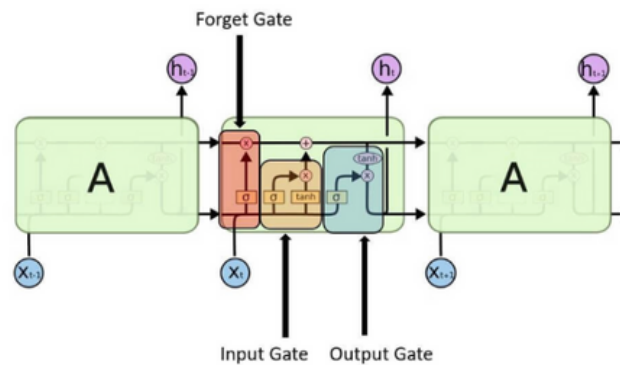
Input	RMSE	MAE
Tourist + Search Index	6.31	4.72

## 2. Long Short Term Memory (LSTM)

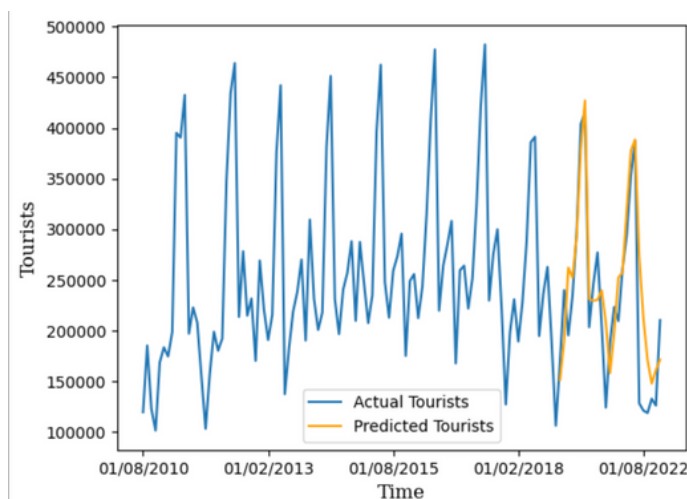
Long Short-Term Memory (LSTM) networks represent a significant advancement in the realm of recurrent neural network (RNN) architectures, purpose-built for the intricacies of sequence and time series data. In contrast to conventional RNNs, LSTMs possess a remarkable capability to capture long-range dependencies within data sequences while dealing with the notorious vanishing gradient problem. They achieve this through deployment of three gates:

- Input gate
- Forget gate
- Output gate

These gates collaboratively exert fine control over information flow within network. The result is an architecture that can selectively remember or discard information, rendering LSTMs exceptionally well-suited for the task of time series forecasting and an array of other sequence-related tasks.



For our problem the preprocessed vectorized dataset (containing the features including number of hotels in Shimla, flights, trains etc.) was given as an input to the ANN and the number of tourists were provided as the output.



Input	RMSE	MAE
Tourist + search Index	4.8	3.7

### 3. SARIMA

The Seasonal Autoregressive Integrated Moving Average (SARIMA) model is a time series forecasting model which extends ARIMA, by incorporating seasonal components to handle seasonality which may be present in our time series dataset.

#### Components of the SARIMA model:

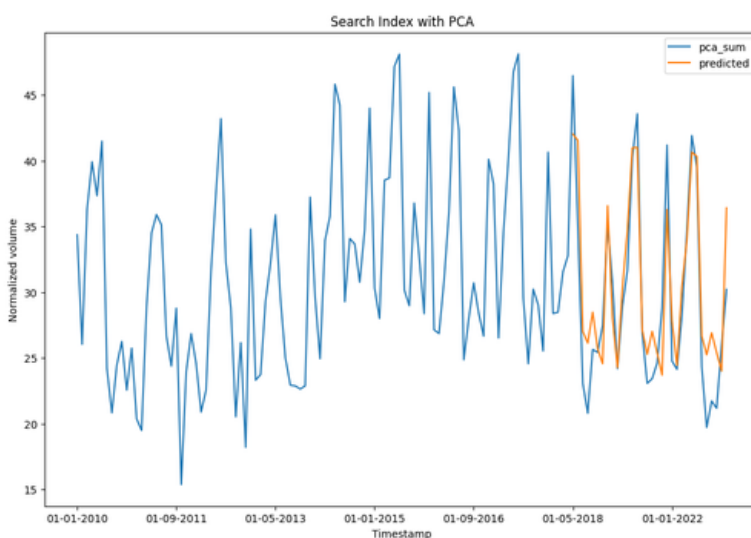
Sarima consists of Seasonal Autoregressive (SAR) and Seasonal Moving Average (SMA) terms, which are denoted as SARIMA(p,d,q)(P,D,Q,s).

Here (p,d,q) represent the non-seasonal Autoregressive (AR), Differencing and Moving Average (MA) parameters respectively, while (P,D,Q) represent the seasonal counterparts. s denotes the seasonality, representing the number of steps in a seasonal cycle which we have chosen as 12. By plotting the ACF and PACF plots we chose the values of p and q as 2 and 3 respectively.

$$SARIMA(\underbrace{p,d,q}_{non-seasonal})(\underbrace{P,D,Q}_m)_m$$

We also observe that if we use PCA to reduce dimensionality of the model replacing the less correlated columns by using column transformation, RMSE and MAE are reduced, thus improving our model.

#### Results:



Input	RMSE	MAE
No. of Tourists	6.31	4.72
Tourist + Search Index	3.98	3.38
Tourists + Search Index with PCA	3.34	2.77

## 4. SARIMAX

SARIMAX extends SARIMA by including external variables for improved forecasting of time series data with additional influences. It regroups AR, MA, differencing, and seasonal effects. On top of that, it adds the X: external variables.

**Equation :**

$$X_t = c + \sum_{i=1}^p \varphi_i X_{t-i} + \varepsilon_t$$

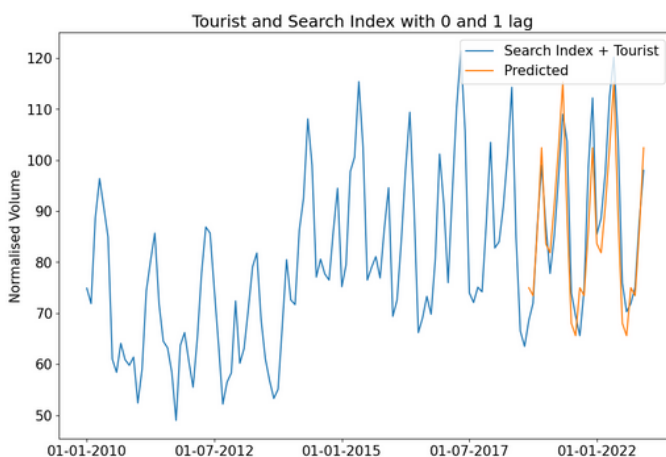
**Model Parameters:**

Sarimax(p,d,q)(P, D, Q, M) is our model where parameters are obtained by acf and pacf plots, where P and Q are the highest significant lag in pacf plot and acf plot respectively . by our observation best-suited values of p,q ranges between 0 to 2 and d and D are the number of steps used in differencing is generally 0 or 1.

**Most Suited Approach:**

We applied the rolling mean function on some of the features to get smoother and better plots. We got the best plot (tourist vs search ) by adding the features (one with the best plot) and their lagged value(lag=1). Below is the plot of Number of Tourist and Summation of the features and their lagged(lag=1) values .

**Results:**

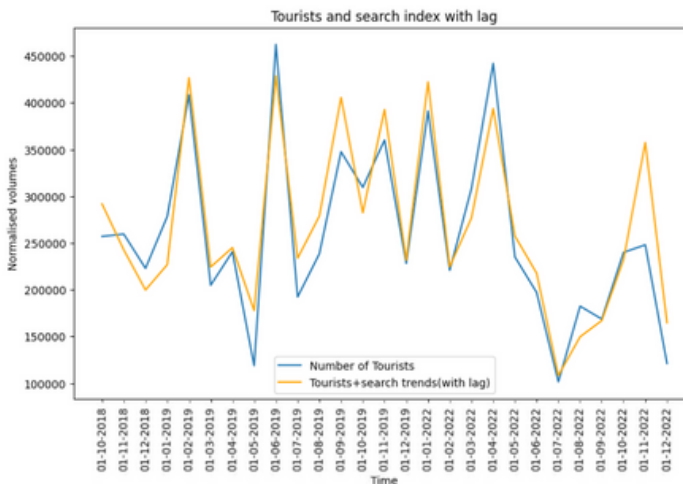


Input	RMSE	MAE
No. of Tourists	6.08	5.31
Tourist + Search Index	7.83	6.48
Search Index with Lag	6.85	5.72

## 5. CATBOOST

CatBoost is a gradient boosting algorithm designed for supervised machine learning tasks, such as classification and regression. It is known for its efficient handling of categorical features and its impressive performance on tabular data. It is designed to be efficient and can train models relatively quickly. It employs various optimization techniques like ordered boosting and data partitioning that help speed up the training process.

### Results:

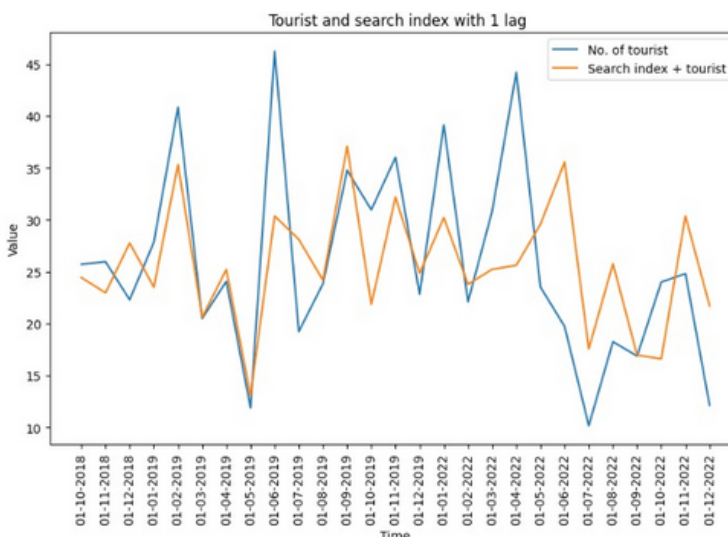


Input	RMSE	MAE
No. of Tourists	3.69	2.87
Tourist + Search Index	3.86	3.10
Search Index with Lag	3.78	3.02

## 6. k-Nearest Neighbours(k-NN)

k-Nearest Neighbors (k-NN) is a simple, non-parametric machine learning algorithm for classification and regression. It makes predictions by finding the majority class or averaging near data points. k represents the number of neighbours to consider.

### Results:



Input	RMSE	MAE
No. of Tourists	7.18	9.10
Tourist + Search Index	6.81	8.851
Search Index with Lag	6.001	7.87



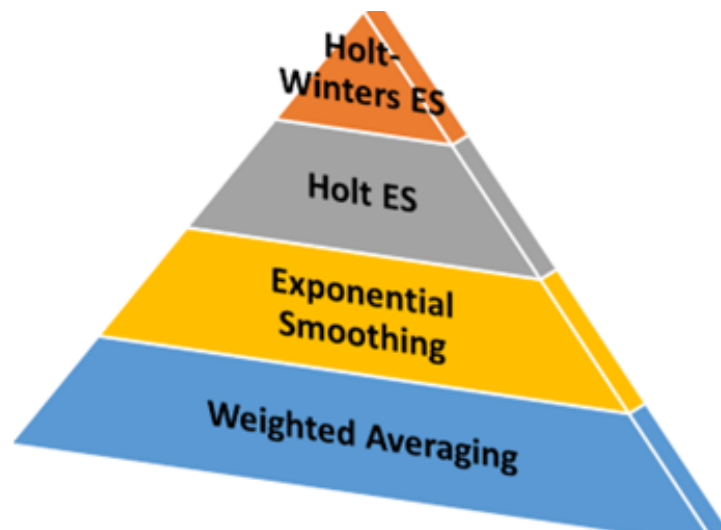
## 7. Exponential Smoothing

Exponential Smoothing is a time series forecasting model that is widely used in various fields. It is designed to capture and forecast time series data with trends and seasonality. Here, new observations receive greater weight than older ones, giving more importance to recent data.

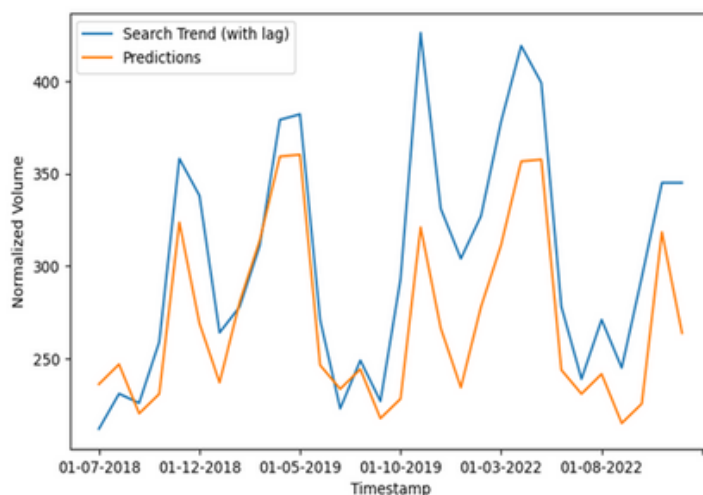
The Exponential Smoothing Model calculates forecast by updating the level, trend and seasonal components based on historical data. The prediction for the next period is generated using the updated components.

### Model Parameters:

Exponential Smoothing models involve parameters like the smoothing level ( $\alpha$ ), smoothing trend ( $\beta$ ), smoothing seasonality ( $\gamma$ ), and seasonal period ( $m$ ). These parameters are estimated from historical data and can affect the forecasting accuracy.



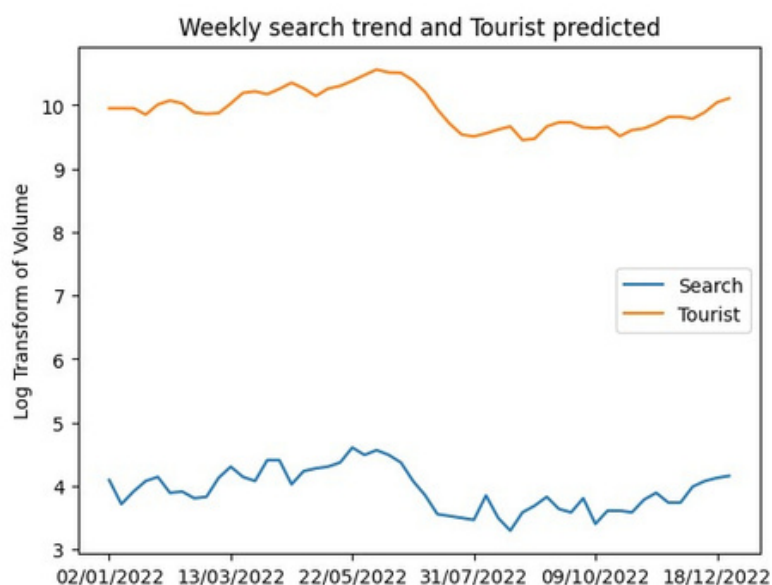
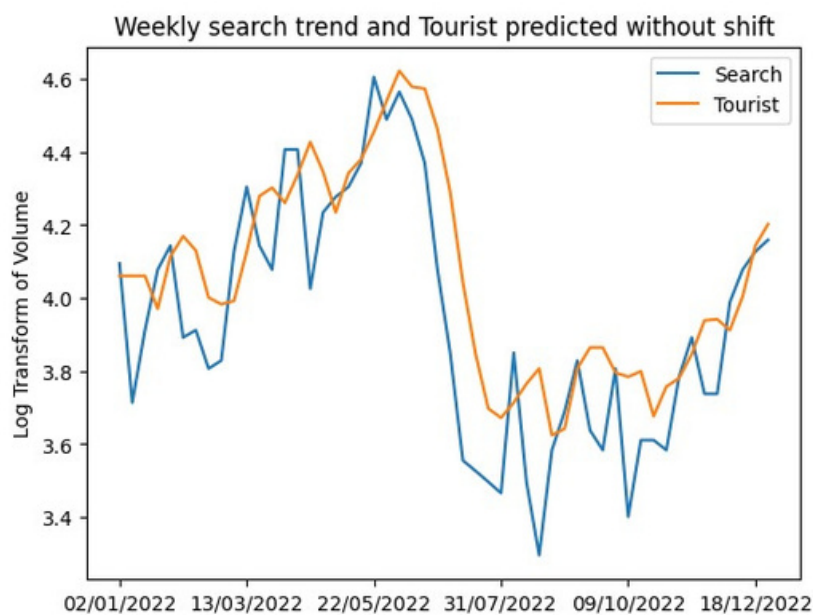
### Results:



Input	RMSE	MAE
No. of Tourists	6.44	7.92
Tourist + Search Index	9.44	11.64
Search Index with Lag	3.60	4.46

# Weekly Interpolation Approximation

- The time series analysis of search trend data and actual tourist numbers in Shimla demonstrates remarkable similarities in their patterns, with identical seasonality and peak occurrences.
- The observed one-month offset between search data and tourist data implies that individuals tend to search for related information both prior to and after their visit to Shimla.
- It can be concluded that the number of tourists is nothing but the summation of search trend and the shift in the graph along Y-axis(due to difference in scaling).
- Granger-causality tests conducted on monthly data have established that a one-month lag significantly influences the estimation of current-month tourist numbers.
- This derived monthly equation can be extended to weekly data by applying a weighted mean of the search trend from the previous three weeks and integrating the shift parameter from the monthly data to obtain precise weekly tourist figures.



# Results and Conclusion

MODELS	RMSE	MAE
ANN	6.31	4.72
Sarima	3.34	2.77
Sarimax	6.85	5.72
Exponential Smoothing	4.46	3.60
XG Boost	6.54	5.02
CatBoost	3.78	3.02
LSTM	4.8	3.7
KNN	6.00	7.87

- After deploying various Time series Forecasting model we found that integrating search trend with the normal statics of tourist helps in increasing the accuracy of the model and facilitate with better forecasting.
- The search trend data for model were selected for the queries whose search trend gave highest correlation with tourist data and granger causality test was used to find the number of lags which can be integrated with tourist volume data.
- Finally models were trained and tested and we found that models gave high accuracy when integrated with search volume data apart from forecasting with only tourist volume.

## Robustness and Scalability

- Our scraping model gives the set of queries related to lodging, food, recreational activities, transport and traffic for the specified place. These search queries are fed to Google trends API which gives the search trend data and out of these queries best queries are selected based on best PCA scores and lags was find out by Granger Causality test. Finally any Forecasting model can be deployed integrating search data with tourist data to predict future tourist. For Kerala, our scraping model gave the following word cloud with best PCA scores.



# Annexure

- <https://www.sciencedirect.com/science/article/pii/S1877050919320423#:~:text=However%2C%20the%20traditional%20statistical%20data,behavioral%20intentions%20in%20real%20time.>
- <https://www.analyticsvidhya.com/blog/2021/07/time-series-forecasting-complete-tutorial-part-1/>
- <https://www.influxdata.com/time-series-forecasting-methods/>
- <https://towardsdatascience.com/time-series-forecasting-with-arima-sarima-and-sarimax-ee61099e78f6>
- <https://www.analyticsvidhya.com/blog/2023/06/sarima-model-for-forecasting-currency-exchange-rates/>
- <https://medium.com/@ritusantra/introduction-to-sarima-model-cbb885ceabe8>
- [https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/#:~:text=LSTM%20\(Long%20Short%20Term%20Memory,ideal%20for%20sequence%20prediction%20tasks.](https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/#:~:text=LSTM%20(Long%20Short%20Term%20Memory,ideal%20for%20sequence%20prediction%20tasks.)
- <https://intellipaat.com/blog/what-is-lstm/>
- <https://www.kaggle.com/code/ritesh7355/develop-lstm-models-for-time-series-forecasting>
- <https://neptune.ai/blog/arima-vs-prophet-vs-lstm>
- <https://www.statsmodels.org/dev/generated/statsmodels.tsa.statespace.sarimax.SARIMAX.html>
- <https://medium.com/swlh/a-brief-introduction-to-arima-and-sarima-modeling-in-python-87a58d375def>
- [https://www.sciencedirect.com/topics/earth-and-planetary-sciences/artificial-neural-network#:~:text=Artificial%20neural%20network%20\(ANN\)%20is,on%20their%20predefined%20activation%20functions.](https://www.sciencedirect.com/topics/earth-and-planetary-sciences/artificial-neural-network#:~:text=Artificial%20neural%20network%20(ANN)%20is,on%20their%20predefined%20activation%20functions.)
- <https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/>
- <https://catboost.ai/>
- <https://www.geeksforgeeks.org/catboost-ml/>
- <https://www.analyticsvidhya.com/blog/2017/08/catboost-automated-categorical-data/>
- <https://www.ibm.com/topics/knn#:~:text=Related%20solutions-,Resources,of%20an%20individual%20data%20point.>
- <https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/>
- <https://www.datacamp.com/tutorial/k-nearest-neighbor-classification-scikit-learn>
- <https://machinelearningmastery.com/exponential-smoothing-for-time-series-forecasting-in-python/>