OPEN IIT
DATA ANALYTICS
2023-24

REPORT

TEAM
D40

Index

Introduction

- Using ML models , prediction of the footfalls in specific regions at specific times and in particular that of “Shimla” as used in our solution.
- The pipeline of the solution begins with the scrapping of data of Shimla tourism available on internet.
- The solution approach is based on an established fact that the actual number of footfalls in a particular region shows direct relation with the google searched keywords related to the similar words to “tourism” , “travelling” , “Dining” etc.
- In an attempt to successfully make our model robust , the trends of the travelers in a specific place could also be relevant for the tourism industry.
- The use of different algorithms which handles seasonality can be implemented to predict the footfalls or the trends.
- The scalability of the model can be taken into consideration by automating the tourism related words that we have implemented using the Py-trends library that automates the words generation and can be used to extract the search indices using the Google Trends.

DATA PRE-PROCESSING

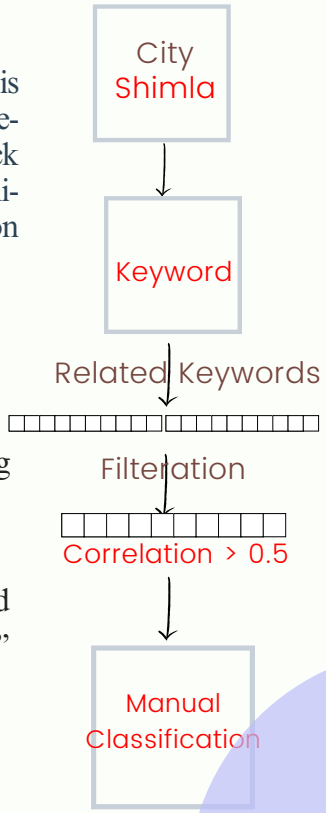
1.Manual Extraction.

" SHIMLA " is the main Keyword Used to get various features.

- We adopt the various steps as follows.
- Visit the Google Trends Website and enter the interesting keywords.
- Set Date ranges: We set the date starting from 01-01-2010 to 01-12-2022.
- Explore the Data: Google Trends will provide us with a graph showing the relative popularity of the keywords over time. We downloaded the data as a CSV file for further analysis.
- As Such, We obtained 100+ CSV Files For 100+ Keywords revolving around "Shimla"
- We combined these 100+ CSV Files into 1 File. Further we find the correlation of these words with Number of tourists using heat maps.
- Using the Key Words with Correlation Coefficient > 0.5 , we narrowed down the no. of keywords to 30.
- Now, Manually, we Classify these 30 Keywords into Categorical bags of Tourism, Dining, Lodging ,Shopping, Traffic and Recreation.
- Further, we applied a heatmap of the correlation matrix using Seaborn and checked that these 6 features are nearly independent.

2. Semi-Automatic Extraction.

Introduction:
In today's information-driven world, the availability of vast and diverse datasets is both a boon and a challenge . Traditional manual extraction processes are time-consuming and prone to errors, while in fully automated methods you may find lack of finesse required for the data. To overcome this dilemma ,we can use semi-automatic data extraction, a methodology that combines the strengths of automation and human oversight.



Steps:
Here "Shimla" is the main keyword used to get various features.
We adopt these steps as follows:

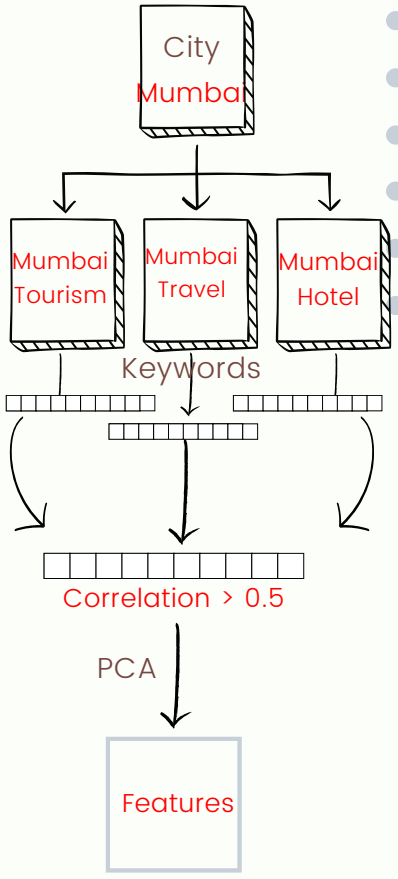
- Instead of using Google Trends directly, we interpret Google Trends data using the "pytrends" Python library.
- We set the date from 01-01-2010 to 01-12-2022.
- In the pytrends library, we use the "relatedQueries" method to retrieve related queries for a given keyword and fetch the number of top queries using the "top" parameter.

- Now, we utilize two key functions: “pytrends.build_payload” and “pytrends.interest_over_time.” These functions allow us to retrieve historical search interest data for specific keywords and time frames (in our case, on a monthly basis).
- We plot the “Spearman correlation” between these keywords and store keywords with a correlation greater than 0.5 with the number of tourists in a list.
- Finally, we manually classify these highly correlated keywords into categorical groups like Tourism, Dining, and Lodging, and create the final CSV file with timestamps, the number of tourists, and these features.

3) Automatic Extraction .

Steps:

- We have the flexibility to enter the name of any city of interest in order to predict the number of tourists visiting.
- Then, we start searching for keywords related to tourism, hotels and travel specific to the chosen city using the function “relatedQueries” present in the pytrends library.
- To ensure that the keywords are trending, we have used the “top” parameter.
- For each individual query we receive 50+ keywords related to it.
- Out of the pool of more than 150 keywords, we select only those keywords which show a correlation of 0.5 or higher with the actual number of tourists in the previous years.
- We've chosen 0.5 as the threshold because going lower would introduce unwanted noise in our predictions, while going higher would reduce the number of features, potentially affecting accuracy.
- We used PCA to categorize keywords into uncorrelated features and analyzed predictions for 1 to 5 features.
- PCA reduces the dimensionality of complex datasets, making them easier to analyze and visualize.
- We found out that our model worked the best when the number of features was 1.



VISUALISATION AND ANALYSIS

Plot of number of tourist visits

- The plot below(Fig 1) shows the number of tourist visits for each month in the years 2019, 2020, 2021, and 2022.
- Seasonal Patterns: In a typical year before 2020 it is observed that the plot is following a certain seasonal pattern. It may be due to many reasons like due to vacations, holidays, or favorable weather conditions.
- Covid Impact (2020 and 2021): In 2020 and 2021, it's expected that the number of tourist visits was significantly lower due to the COVID-19 pandemic. Look for the period when the dip occurred, and note how severe it was. This can help in understanding the pandemic's impact on the tourism industry in the city.
- Recovery in 2022: There was a recovery in tourist visits in 2022. It was back to pre pandemic levels .

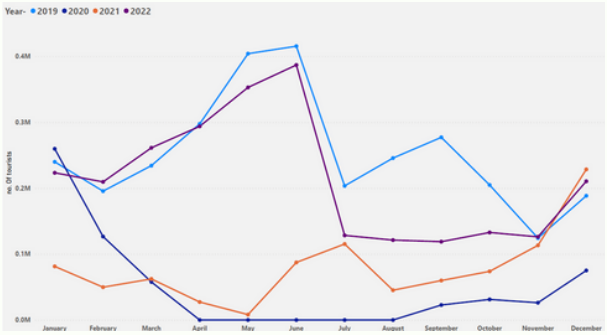


fig 1

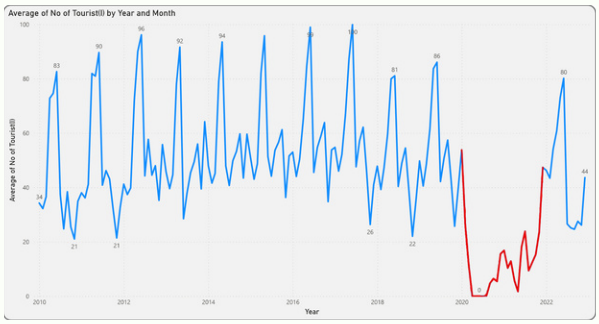
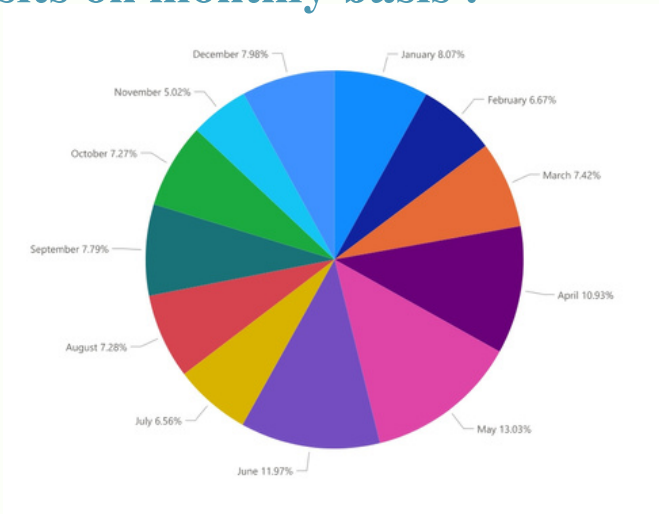


fig 2

The graph above (fig2) shows the average number of tourists who visited Shimla each month from 2010 to 2022. Seasonal Patterns Breakdown : From 2010 to 2019, the plot follows a clear seasonal pattern, with peaks in the summer months (April to June) and lows in the winter months (November to February). This is likely due to factors such as vacations, holidays, and favorable weather conditions. However, the COVID-19 pandemic caused a significant disruption to this seasonal pattern in 2020 and 2021, as tourist visits plummeted across the globe. Recovery in 2022:In 2022, there was a significant recovery in tourist visits to Shimla, with numbers returning to pre-pandemic levels. This suggests that the tourism industry in Shimla is rebounding after the devastating impact of the COVID-19 pandemic.

Plot of number of tourist visits on monthly basis :



- The graph above shows the percentage of tourists who visited Shimla monthly.
- The pie chart above provides a clear overview of the seasonal patterns in tourist visits to Shimla. The peak tourist season is from May to June(>30%), with the summer months of April to June (>35%)also being popular with tourists. The winter months of November to February(are the least popular with tourists.
- Shimla's tourist influx follows a seasonal pattern rooted in three key factors: pleasant summer climate (15-25°C), ideal refuge from the plains' heat; school holidays in May and June, drawing families; and vibrant festivals, such as the Summer Festival, luring travelers from India and beyond.

Spearman Correlation Coefficient

The Spearman correlation coefficient, also called Spearman's Rank Correlation (RHO), assesses the strength and direction of the relationship between two variables. It's suitable for non-normally distributed data, like IQ and test scores, indicating linear or nonlinear associations within a range of -1.0 to +1.0.

Formula to calculate Spearman correlation coefficient :

$$= \rho_{\mathrm{R}(X),\mathrm{R}(Y)} = \frac{\mathrm{cov}(\mathrm{R}(X),\mathrm{R}(Y))}{\sigma_{\mathrm{R}(X)}\sigma_{\mathrm{R}(Y)}}$$

We create a heatmap of the correlation matrix using Seaborn.

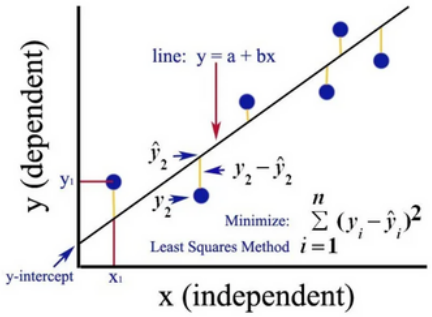
APPROACH AND MODELS

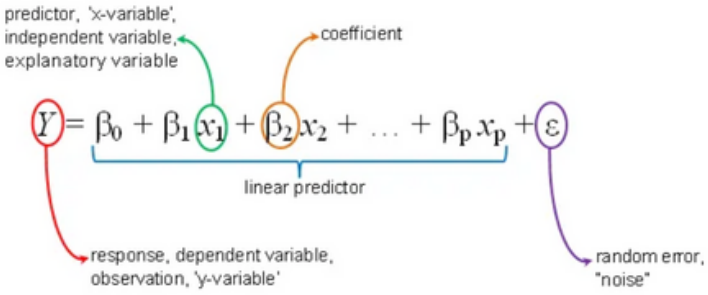
Linear Regresion

Linear Regression estimates the coefficients of the linear equation, involving one or more independent features that best predict the value of the data. Linear regression fits a straight line or surface that minimizes the discrepancies between predicted and actual output values. There are simple linear regression calculators that use a “least squares” method to discover the best-fit line for a set of paired data. You then estimate the value of Data from X (features).

You can use linear regression to provide better insights by uncovering patterns and relationships that your business colleagues might have previously seen and thought they already understood.

For our problem the Tourist head count per month was given as an input to the model. We observed a Root Mean Squared Error of **113425.62**

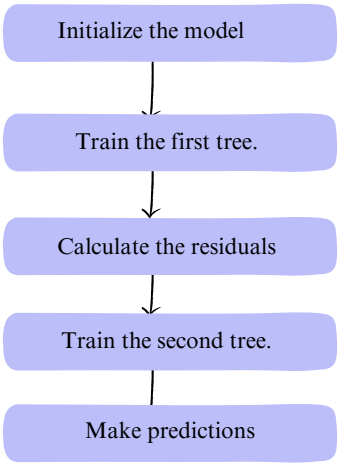


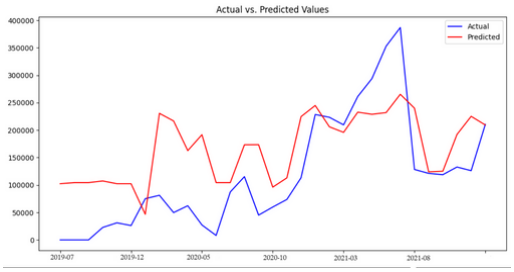


Catboost Regression

CatBoost is a gradient boosting on decision trees algorithm that is known for its high performance and efficiency. It is a popular choice for machine learning competitions and is used by many companies in production.

Flowchart of working of Model.





Graph showing Actual and Predicted values line

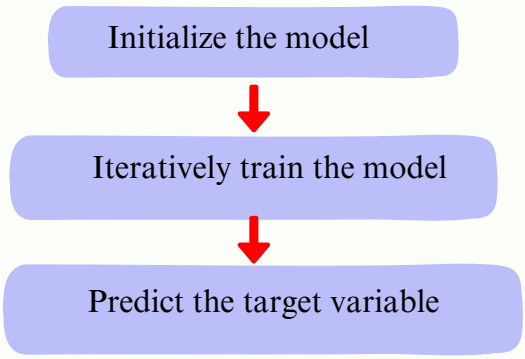
RMSE = 88,722

MAE = 74275

Accuracy = 56.67%

LightGBM Regressor

LightGBM uses a leaf-wise tree growth algorithm, which is more efficient than the level-wise tree growth algorithm used by other GBDT algorithms. Applying LGBM, we get RMSE **1,26,723,** MAE about **1,38,117** and Percentage accuracy is **56.67%**.
Training a LightGBM Model:



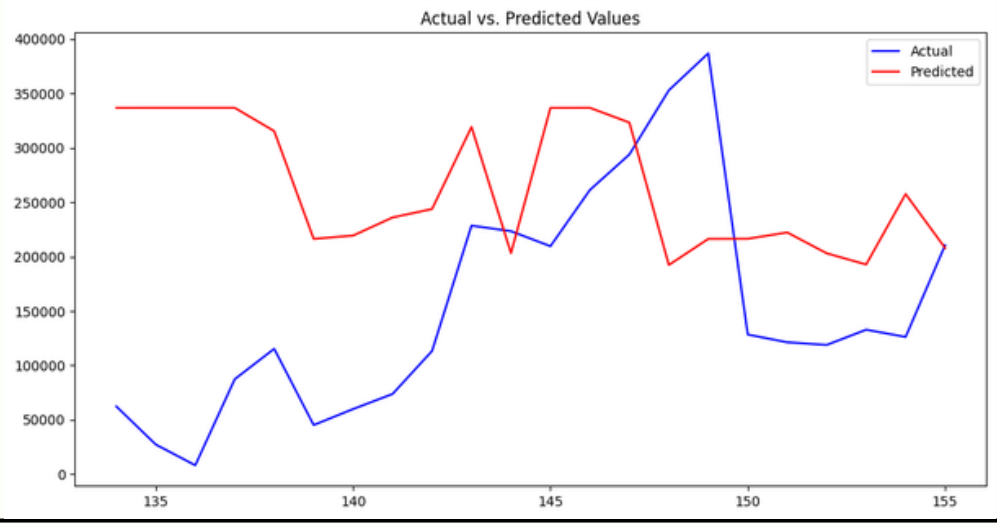
LightGBM Model have advantages over other because it provides Speed, Efficiency and Accuracy.

LightGBM can be used for a variety of machinelearning tasks, including Classification, Regression and Ranking.

LightGBM is a relatively new algorithm, and it has a few disadvantages, including Lack of support for certain features and Complexity.

Gradient Boosting(XGBoost)

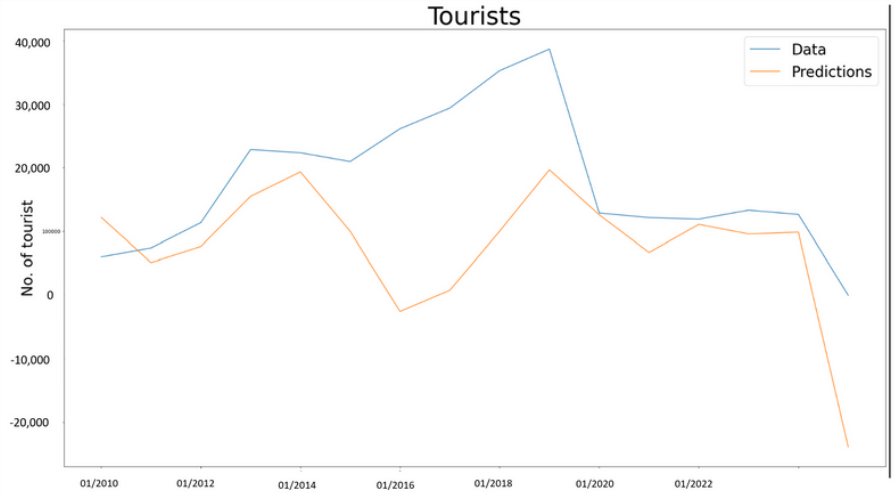
XGBoost, also known as "Extreme Gradient Boosting," is a widely acclaimed and powerful machine learning algorithm. It excels in both regression and classification tasks, making it the top choice for various applications spanning finance to natural language processing. Belonging to the ensemble learning family, specifically boosting algorithms, XGBoost stands out with its exceptional predictive accuracy and unmatched versatility.
XGBoost boasts exceptional efficiency and speed, making it one of its key strengths. By applying gradient boosting techniques, XGBoost enhances the performance of decision trees while effectively reducing overfitting and maintaining remarkable predictive-accuracy. Moreover, it provides robust support for handling missing data and effortlessly handles feature selection .XGBoost's popularity among data scientists and machine learning practitioners stems from its versatility, user-friendly interface, and consistent top-tier performance in various machine learning competitions.



MLPRegressor

MLP (Multi-Layer Perceptron)Regressors are a type of feedforward neural network that works by learning complex mappings between input features and output values. An MLP Regressor consists of multiple layers of interconnected neurons, and it's called "multi-layer" because it has at least three layers: an input layer, one or more hidden layers, and an output layer. This Analytical Model is similar to the ANN Model. However, there is **no activation function in the output layer.**

We utilized an MLP Regressor to evaluate the machine learning analysis of our dataset, which we meticulously collected. This approach allowed us to explore a novel perspective concerning the correlation between actual foot traffic and the associated trends in the data.



We optimized by tuning hyperparameters, notably increasing the maximum iteration count from **100** to **500,000**. Despite the increased runtime, this led to a substantial reduction in **Root Mean Square Error**, dropping from **437,783.42** to **148,045.38**. Here's the MLP-generated graph.

TIME SERIES ANALYSIS

Due to the nature of our data being a time-series data, there are a number of specificities involved. That's why time series modeling is one of the approaches we can take for our data.

Univariate vs multivariate time series models

The first specificity of time series is that the timestamp that identifies the data has intrinsic meaning. Univariate time series models are forecasting models that use only one variable (the target variable) and its temporal variation to forecast the future. Univariate models are specific to time series.

In other situations, you may have additional explanatory data about the future. For example, imagine that you want to factor the weather forecast into your product demand forecast, or that you have some other data that will influence your predictions. In this case, you can use Multivariate time series models. Multivariate Time Series models are the Univariate Time Series models that are adapted to integrate external variables.

If you want to use a temporal variation on your time series data, you will first need to understand the different types of temporal variations that you can expect.

Autocorrelation

Let's move on to the second type of temporal information that can be present in time series data: autocorrelation.

Autocorrelation is the correlation between a time series' current value with past values. If this is the case, you can use present values to better predict future values.

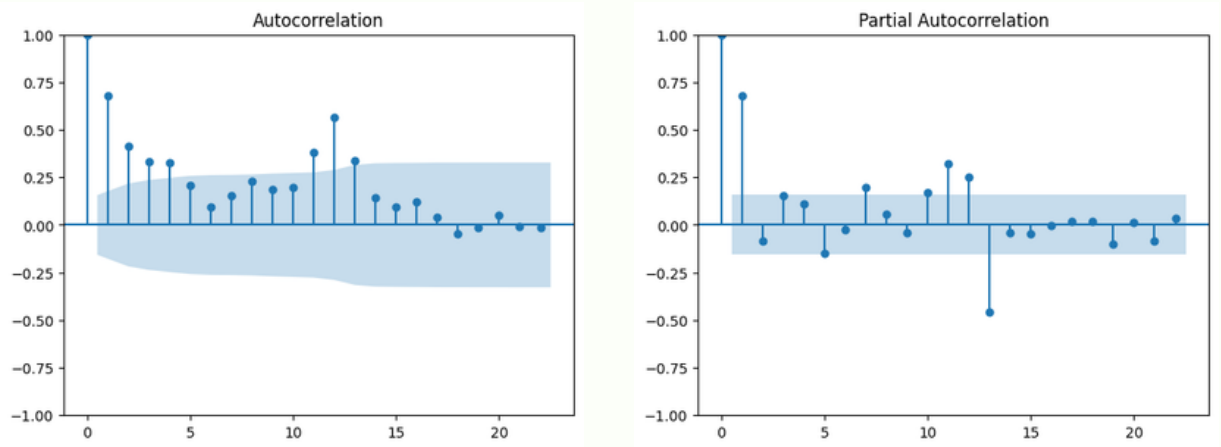
Autocorrelation can be positive or negative:

1. Positive autocorrelation means that a high value now is likely to yield a high value in the future and vice versa. You can think about the stock market: if everybody is buying a stock, then the price goes up. When the price goes up, people think that this is a good stock to buy and they buy it too, thereby driving the price even higher. However, if the price goes down, then everybody is fearful of a crash, sells their stocks and the price becomes lower.
2. Negative autocorrelation is the opposite: a high value today implies a low value tomorrow and a low-value today implies a high-value tomorrow. A common example is rabbit populations in natural environments. If there are a lot of wild rabbits in the summer of one year, they will eat all of the natural resources available. During winter, there will be nothing left to eat, so many of them will die and the surviving rabbit population will be small. During this year with a small rabbit population, the natural resources will grow back and allow the rabbit population to grow in the following year.

Two famous graphs can help you detect autocorrelation in your dataset: the ACF plot and the PACF plot.

The autocorrelation function is a tool that helps identify whether autocorrelation exists in your time series. The PACF is an alternative to the ACF. Rather than giving the autocorrelations, it gives you the partial autocorrelation. This autocorrelation is called partial, because with each step back in the past, only additional autocorrelation is listed. This is different from the ACF, as the ACF contains duplicate correlations when variability can be explained by multiple points in time.

For our data the Autocorrelation graphs are:



One of the major difference between ACF and PACF is that, ACF also measures indirect correlation up to the lag in question, while PCAF does not.

ARIMA MODEL

Before ARIMA lets understand some terminologies.

Lag

Lags are simply delays in time steps within a series. Consider a time index t, the lag 1 time index with respect to t is simply t-1, lag 2 is t-2, and so on.

Stationarity

A stationary time series is one that has its mean, variance and autocorrelation structure unchanging overtime. In other words, it does not have any cycle/trend or seasonality. ARIMA models assume that the data is stationary, meaning that the statistical properties of the data do not change over time. To assess stationarity we use "Augmented Dickey-Fuller (ADF) test". If the data is not stationary, we may need to perform differencing (removing trends and seasonality) to make it stationary.

ARIMA

is class of models. Intuitively, ARIMA models compose 2 parts:

The autoregressive term

AR(p): $y_t = \beta + \epsilon_t + \sum_{i=1}^{p} \theta_i y_{t-i}$

and

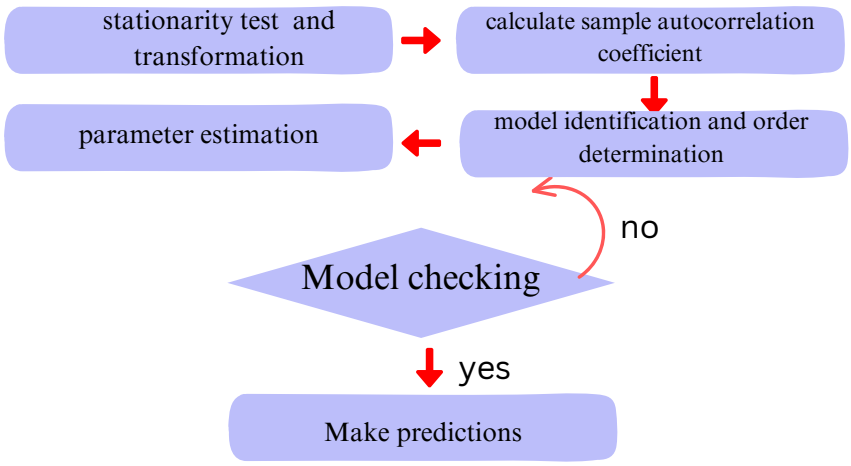
The moving-average term

MA(q): $y_t = \Phi(L)^q \epsilon_t + \epsilon_t$

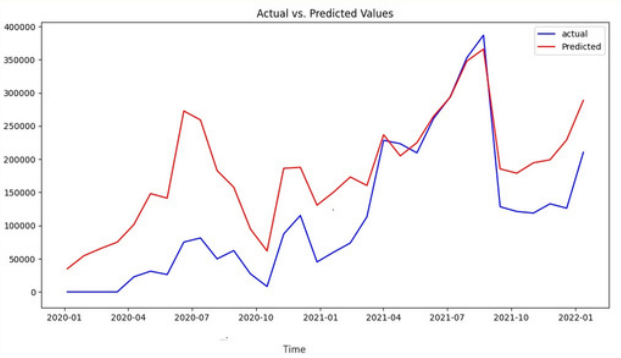
(the first variable represents the series' mean, denoted by μ ; a finite number of MA coefficients are added up to give the second variable, denoted by θ , and the model residuals, denoted by ω; and white noise is represented by ωt .)

The former views the value at one time just as a weighted sum of past values. The latter model that same value also as a weighted sum but of past residuals (confer. *time series decomposition*). There is also an integrated term (I) to difference the time series

Now , we have to identify these parameters(p , d , q) by analyzing the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots.



For our problem the Tourist head count per month was given as an input to the model. We observed an accuracy of 71.875 % on the validation dataset , Root Mean Squared Error of 85079.39140190213.



SARIMAX MODEL

Why SARIMAX over ARIMA

Overall, ARIMA is a very decent type of models. However, the problem with this vanilla version is that it cannot handle seasonality — a big weakness. Comes SARIMA — the predecessor of SARIMAX. One shorthand notation for SARIMA models is:

$$\text{SARIMA}\ (p,d,q) \times (P,D,Q,S)$$

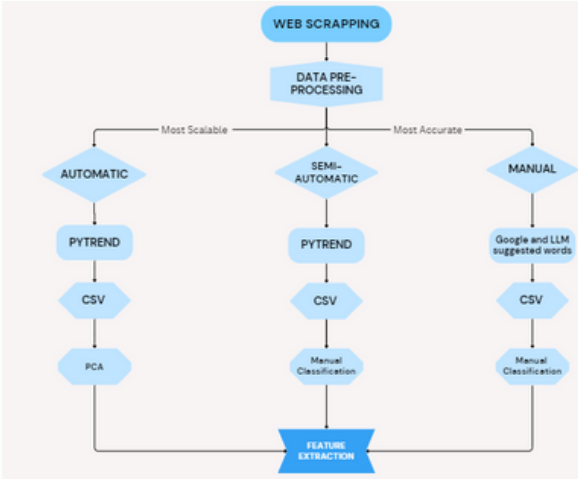
$$\phi_p(B)\,\Phi_P(B^S)\,W_t = \theta_q(B)\,\Theta_Q(B^S)\,\omega_t$$

where, p = non-seasonal autoregressive (AR) order, d = non-seasonal differencing, q= non-seasonal moving average (MA) order, P = seasonal AR order, D = seasonal differencing, Q = seasonal MA order, and S = length of repeating seasonal pattern.

SARIMAX extends on this framework just by adding the capability to handle exogenous variables. Number of Tourist is the go-to option, but you can also get your own domain-specific features if you need to. In our case, we fetched the list of Google Trend Searches people like to do before going to tour destination.

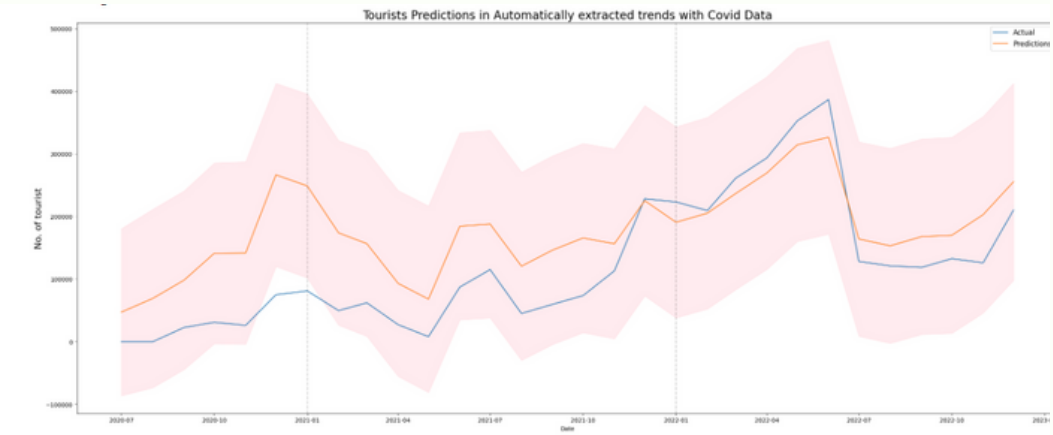
Seasonal Auto-Regressive Integrated Moving Averages with exogenous factors, or SARIMAX, is an extension of the ARIMA class of models. Intuitively, ARIMA models compose 2 parts: the autoregressive term (AR) and the moving-average term (MA). The former views the value at one time just as a weighted sum of past values. The latter model that same value also as a weighted sum but of past residuals (confer. *time series decomposition*). There is also an integrated term (I) to difference the time series

OUR MODEL:

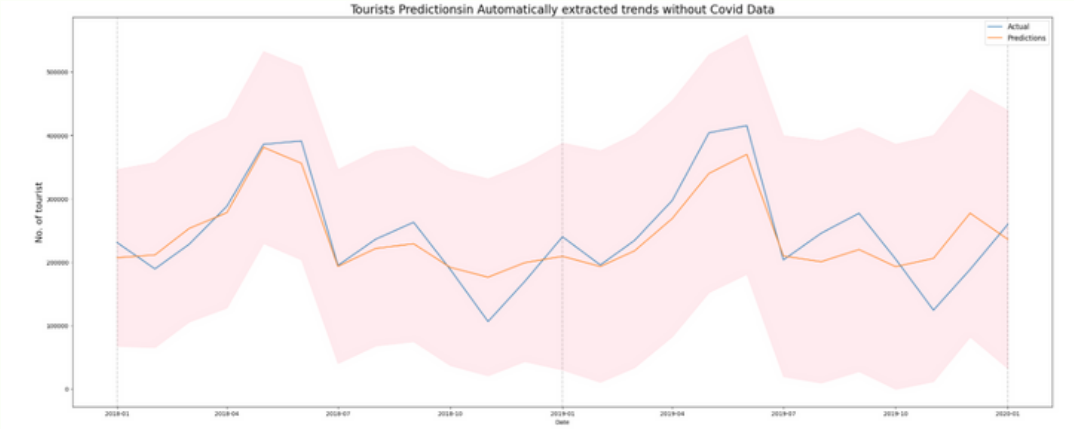


Results for Automatic

For our Data we got and RMSE of **80611** and an accuracy of **70%** when we consider the covid data.

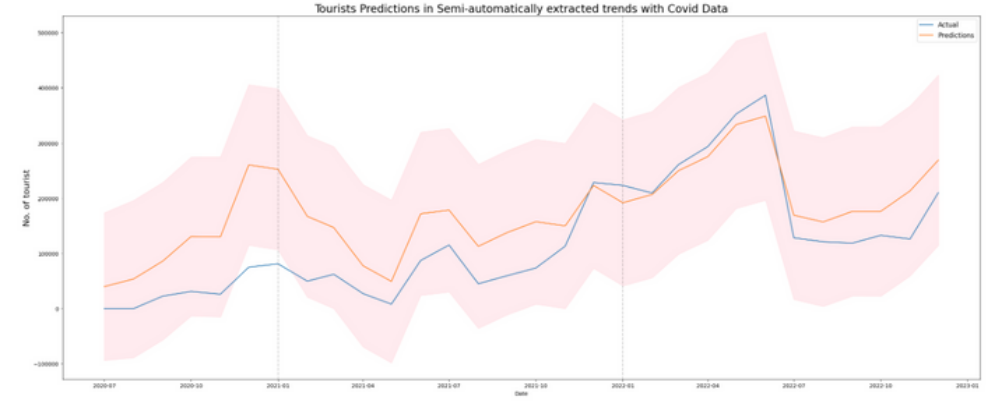


While this is the best trend model can better predict the values of up coming months. Still if we want to consider the covid data RMSE value we get **70677**.

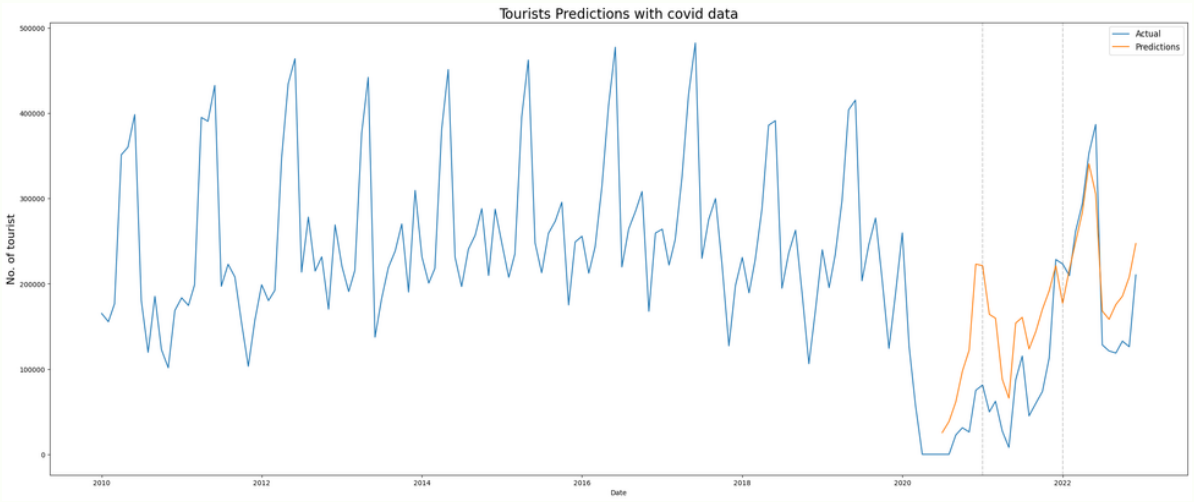


Results for Semi-Automatic

For our Data we got and RMSE of **75488.45** and an accuracy of 7**5%** when we do consider the covid data.

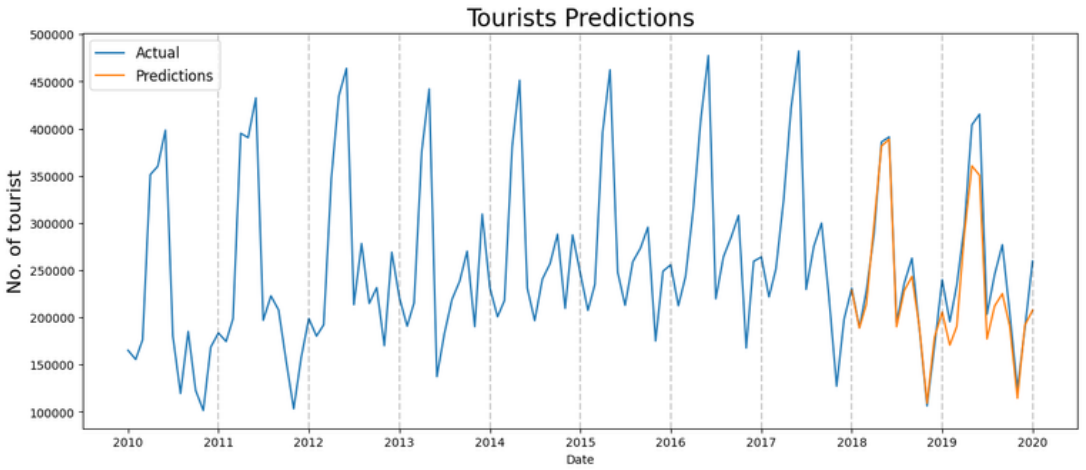


While this is the best trend model can better predict the values of up coming months. Still if we want to consider the covid data RMSE value we get **70677.07345080991**

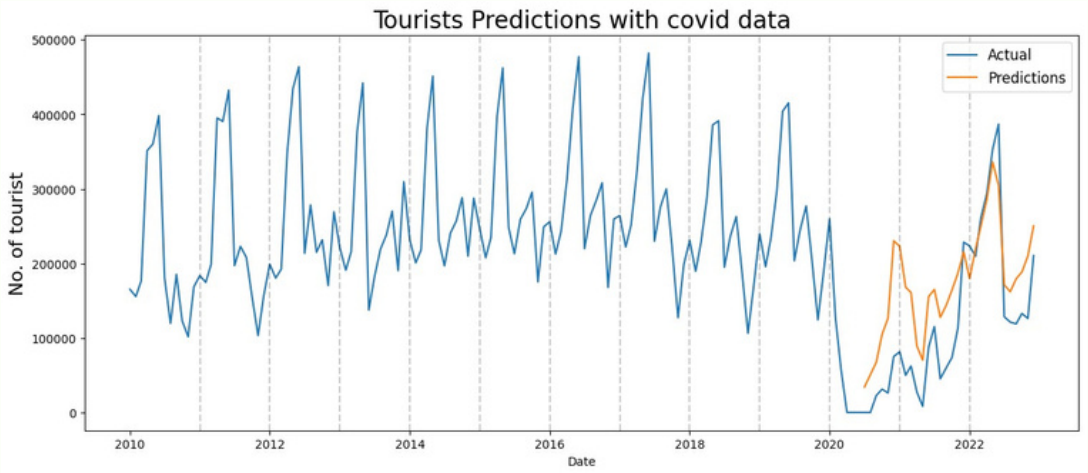


Results for Manual

For our Data we got and RMSE of **24927.41** and an accuracy of **91%** when we do not consider the covid data.



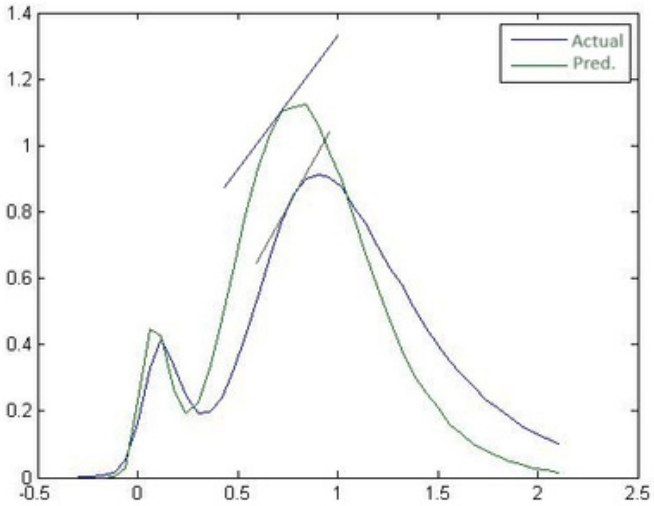
While this is the best trend model can better predict the values of up coming months. Still if we want to consider the covid data RMSE value we get **72899.14273752493**



Hypertuning of model parameters

Analysing Behavior of Model output

A similarity metric **e** was used to compare the behaviour of actual and predicted curves. The metric's definition is as follows:



This method was inspired from Williamson's Dual Primal Method

$$\text{Similarity Metric } \mathbf{e} = \left(\frac{dx}{dt}\right) \times \left(\frac{dy}{dt}\right)$$

$$\mathbf{e} \geq 0 \rightarrow \text{Curves have similar behaviour}$$

$$\text{Overall Behaviour determined by: } E = \sum_t \overline{\text{sgn}}\,(e_t) \text{ where } \overline{\text{sgn}}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

$$= \text{Number of months were trend matched}$$

$$\text{Percentage Matched} = \frac{\text{Number of Months were trend matched}}{\text{Total Number of Months}} \times 100 = \frac{E}{\Delta t} \times 100$$

Conclusion and Final Approach

The machine learning model we built could be scaled to make it fully automated which could not only predict the trends of tourism at particular places but also predict the actual number of footfalls in the place provided we have the historical tourism data of that place.
As far as our model was concerned, we implemented the three approaches namely **automated**, **semiautomated,** and **manual searching,** and had a tradeoff between accuracy and robustness.

After benchmarking multiple models, we opted for the **Sarimax model** with specific hyperparameters (**p=3, q=1, r=1**) and (**P=1, D=1, Q=1, M=12**). Our analysis revealed that the automated model exhibited the highest scalability, while the manual model demonstrated superior accuracy.
Furthermore, after excluding data from the years affected by **COVID-19** as outliers, it became evident that all models showed **improved performance**.

Annexure

- https://www.geeksforgeeks.org/catboost-ml/
- https://towardsdatascience.com/deep-neural-multilayer-perceptron-mlp-with-scikit-learn-2698e77155e
- https://trends.google.com/trends/
- https://www.geeksforgeeks.org/python-arima-model-for-time-series-forecasting/
- https://www.geeksforgeeks.org/lightgbm-light-gradient-boosting-machine/
- https://medium.com/@sandha.iitr/tuning-arima-for-forecasting-an-easy-approach-in-python-5f40d55184c4
- https://towardsdatascience.com/an-introduction-to-support-vector-regression-svr-a3ebc1672c2
- https://math.mit.edu/~goemans/PAPERS/book-ch4.pdf