OPEN IIT DATA ANALYTICS

Team number: D30

Problem statement:

- Machine leaning model to predict tourist arrivals to **Tirupati** using google trends.

No table of contents entries found.

INDEX

1. Why Tirupati?
2. Data Collection
3. Models
   i) Statistical Models
   ii) Machine Learning Models
4. Annexure
   I) Data Description
   ii) Features Used
   iii) Plots
   iv) Model accuracies
5. References

Location: Tirupati, Andhra Pradesh, India

Why Tirupati?

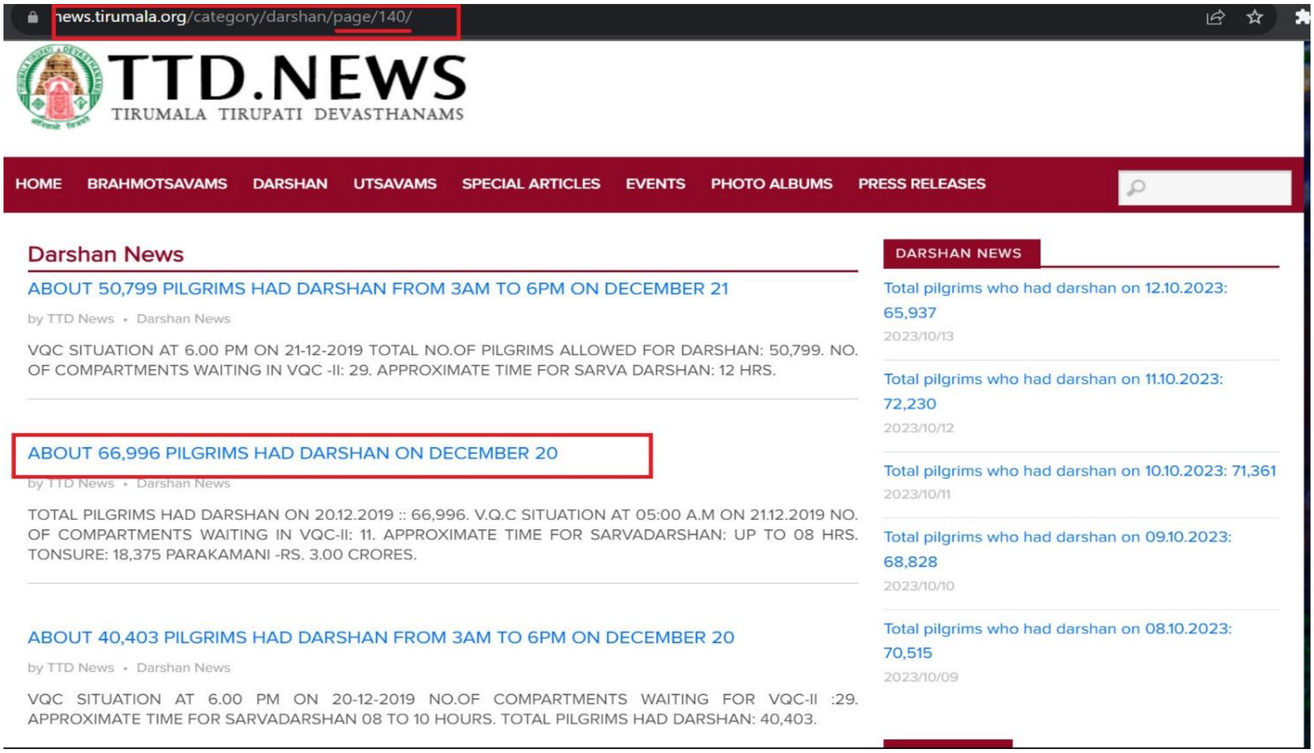
- **Wealth and Donations:** The Tirupati Balaji temple is renowned for its wealth and the donations it receives. Devotees from all over the world contribute to the temple, making it one of the wealthiest Hindu temples globally. This wealth is used for various charitable and religious activities.
- **Devotee Footfall:** The temple receives a massive number of visitors daily, with an average of 50,000 to 100,000 devotees visiting the Tirumala temple at Tirupati. This high footfall showcases the immense religious and cultural significance of the temple.
- **TTD Arrangements:** The Tirumala Tirupati Devasthanams (TTD) has made significant efforts to provide a comfortable and convenient experience for devotees. This includes arrangements for smooth darshan (viewing of the deity) of Lord Venkateswara, accommodation facilities, and other amenities to make the pilgrimage a fulfilling experience.
- **Tourist Arrival Prediction:** The use of machine learning models to predict tourist arrivals to Tirupati using internet search indices can be beneficial for planning and managing resources. It helps authorities prepare for the influx of visitors, improving infrastructure, and ensuring the safety and comfort of pilgrims.
- **Food Service Management:** Managing the free food service called "Anna-Prasadam" efficiently is crucial. With such a high number of devotees visiting daily, there is a need to prevent wastage of food while ensuring that everyone in need is served. Predictive models can help in resource allocation, reducing waste, and making the food service more effective.

DATA COLLECTION:

- From google trends, we have collected data using 13 relevant search indexes:

Tirupati rooms	Tirupati temple history	Ttd
Venkateswara (youtube search)	Venkateswara swamy	Tirupati Darshan timings
Tirumala	Train to Tirupati	Tirupati
Tirupati trains	Kanipakam	Tirupati distance
kalahasti		

- From the archives of http://news.tirumala.org/ we have used web scraping to extract the data of number of piligrims visiting Tirumala temple.



Darshan News

ABOUT 50,799 PILGRIMS HAD DARSHAN FROM 3AM TO 6PM ON DECEMBER 21

by TTD News • Darshan News

VQC SITUATION AT 6.00 PM ON 21-12-2019 TOTAL NO.OF PILGRIMS ALLOWED FOR DARSHAN: 50,799. NO. OF COMPARTMENTS WAITING IN VQC -II: 29. APPROXIMATE TIME FOR SARVA DARSHAN: 12 HRS.

ABOUT 66,996 PILGRIMS HAD DARSHAN ON DECEMBER 20

by TTD News • Darshan News

TOTAL PILGRIMS HAD DARSHAN ON 20.12.2019 :: 66,996. V.Q.C SITUATION AT 05:00 A.M ON 21.12.2019 NO. OF COMPARTMENTS WAITING IN VQC-II: 11. APPROXIMATE TIME FOR SARVADARSHAN: UP TO 08 HRS. TONSURE: 18,375 PARAKAMANI -RS. 3.00 CRORES.

ABOUT 40,403 PILGRIMS HAD DARSHAN FROM 3AM TO 6PM ON DECEMBER 20

by TTD News • Darshan News

VQC SITUATION AT 6.00 PM ON 20-12-2019 NO.OF COMPARTMENTS WAITING FOR VQC-II :29. APPROXIMATE TIME FOR SARVADARSHAN 08 TO 10 HOURS. TOTAL PILGRIMS HAD DARSHAN: 40,403.

DARSHAN NEWS

Total pilgrims who had darshan on 12.10.2023: 65,937
2023/10/13

Total pilgrims who had darshan on 11.10.2023: 72,230
2023/10/12

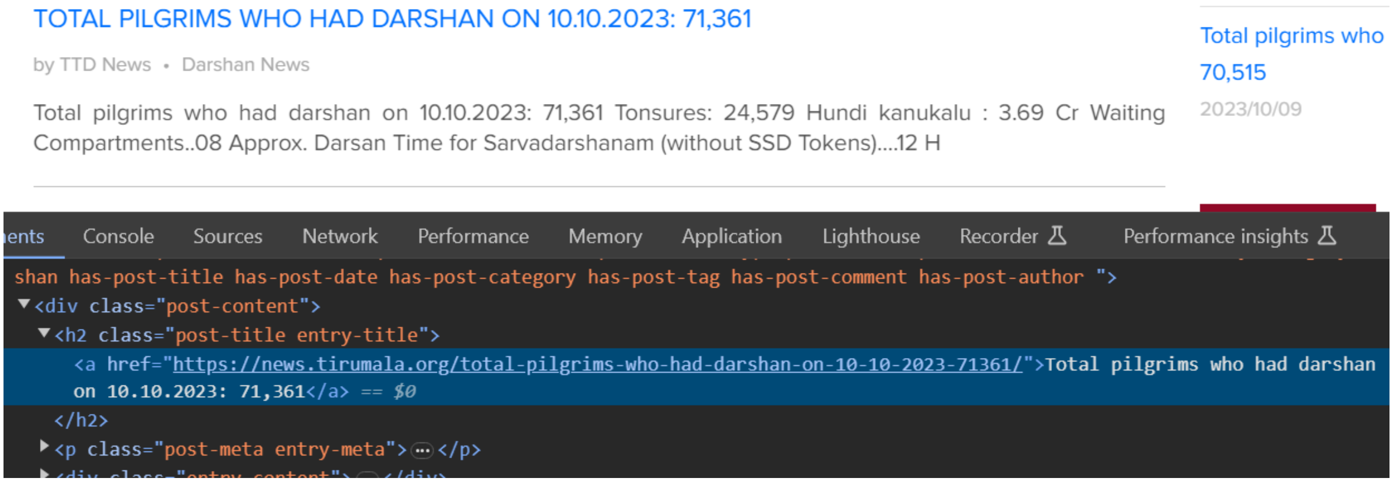
Total pilgrims who had darshan on 10.10.2023: 71,361
2023/10/11

Total pilgrims who had darshan on 09.10.2023: 68,828
2023/10/10

Total pilgrims who had darshan on 08.10.2023: 70,515
2023/10/09

  - The code is tailored to fetch pilgrim data from the "Darshan News" articles on tirumala.org. It captures data specific to the date of each article, providing a snapshot of daily pilgrim statistics

- We employed a combination of tools and libraries for web scraping and data manipulation. These include Selenium for web automation, Firefox web driver, WebDriver Manager for managing the web driver, and the Pandas library for data handling and export.



TOTAL PILGRIMS WHO HAD DARSHAN ON 10.10.2023: 71,361

by TTD News • Darshan News

Total pilgrims who had darshan on 10.10.2023: 71,361 Tonsures: 24,579 Hundi kanukalu : 3.69 Cr Waiting Compartments..08 Approx. Darsan Time for Sarvadarshanam (without SSD Tokens)....12 H

Total pilgrims who
70,515
2023/10/09

```
shan has-post-title has-post-date has-post-category has-post-tag has-post-comment has-post-author ">
▼<div class="post-content">
  ▼<h2 class="post-title entry-title">
      <a href="https://news.tirumala.org/total-pilgrims-who-had-darshan-on-10-10-2023-71361/">Total pilgrims who had darshan
      on 10.10.2023: 71,361</a> == $0
    </h2>
  ▶<p class="post-meta entry-meta">⋯</p>
```

- **Data Extraction Objective**: In the first illustration, our primary goal is to extract specific data, comprising the date and the count of pilgrims. This valuable information can be retrieved from the 'href' attribute or the 'innerHTML' of anchor tags within the page.
- **Selective Targeting**: To ensure precision, we exclusively focus on anchor tags that are nested within 'h2' tags with the class attribute set to "post-title entry-title." Any other anchor tags present on the page are disregarded. This selective process is efficiently accomplished using the 'querySelectorAll' method in JavaScript.
- **Data Refinement with Pandas**: Following the data extraction phase, we employ the powerful Pandas library for post-processing. This step involves cleaning and organizing the extracted data into a structured format.
- **Structured Data Storage**: The resultant {date, count of pilgrims} data is then meticulously stored as a Pandas data frame, facilitating further analysis and utilization.
- **Data Range**: It's noteworthy that the website hosts data spanning from the year 2013 to 2023. However, due to variations in the Document Object Model (DOM) structure of the news articles, a uniform CSS selector cannot be uniformly applied to all cases.
- **Comprehensive Data Acquisition**: This ambitious data extraction project encompasses ten years of data and spans over 500+ pages. To achieve this, we employ the Firefox driver for systematic page scanning.
- **Adaptive Approach**: As the daily text data (in 'href' and 'innerHTML' formats) doesn't maintain a consistent structure, our team remains adaptable and responsive. We meticulously address different patterns and exceptions encountered during the extraction process to ensure data accuracy and completeness.
- Beyond the first 120 pages, some news data lacks the year, presenting only the day and month. To address this, we dynamically inferred the year by tracking back from the most recent 2023 article based on page numbering.

MODELS

In this study, we employed a comprehensive approach to predictive modeling, combining statistical models known as SARIMA n-Hits with an ensemble of ML and DL models like LSTM and xgboost. We used these techniques to make accurate predictions based on the available data.

Statistical Models

SARIMA

The primary steps in our analysis involved ensuring data "stationarity" through differencing and validating stationarity using the "ADF test." Additionally, we conducted parameter tuning through grid search to optimize the model's performance. Finally, we selected the best model based on the AIC criterion, taking into account both endog regressors" and exog regressors.

We employed Principal Component Analysis (PCA) as a powerful technique for dimensionality reduction. The objective was to streamline a dataset with 13 original features into a more concise representation containing only 5 key features.

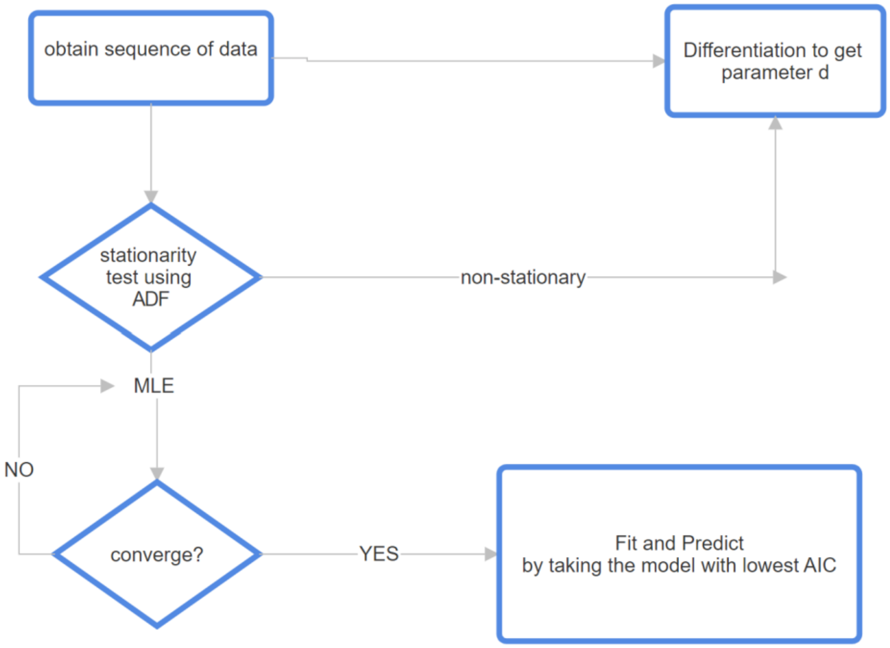
PCA operates by identifying and retaining the most significant components of the data, thereby reducing its dimensionality while minimizing information loss.

We employed the "SARIMA" (Seasonal Autoregressive Integrated Moving Average) model, a time series forecasting method that accounts for seasonality and trends in the data. SARIMA models are widely used for their ability to capture the temporal dependencies in time series data.

- **Data Stationarity:** To ensure the suitability of our data for time series analysis, we utilized "differentiation" to make the data "stationary." Stationarity is a key prerequisite for time series modeling as it allows for more accurate predictions.
- **ADF Test:** The "ADF test" (Augmented Dickey-Fuller test) is a statistical test used to confirm data stationarity. ADF test helps in determining whether differencing has successfully removed any trends or seasonality from the data.
    - The ADF test evaluates whether the null hypothesis can be rejected, which means that the data is stationary. The rejection criteria in the ADF test depend on the test statistic and critical values. The test statistic is compared to a critical value from the Dickey-Fuller distribution to determine if the null hypothesis can be rejected.
    - If the test statistic is less than the critical value, the null hypothesis is rejected, indicating that the data is stationary.
    - If the test statistic is greater than the critical value, the null hypothesis is not rejected, suggesting that the data is non-stationary.
    - The choice of critical values depends on the specific ADF test version being used, and it can vary based on factors like sample size and lag order.
    - The unit test involves estimating the autoregressive model of the time series, calculating the test statistic, and then comparing it to the critical values to make a determination regarding stationarity.
- **Parameter Tuning - Grid Search:** We performed "parameter tuning" through a "grid search" approach. Grid search systematically explores various combinations of model hyperparameters to identify the optimal configuration. This helps in fine-tuning the model to improve its accuracy.

- **AIC Criterion:** The "AIC criterion" (Akaike Information Criterion) is a measure used to evaluate the goodness of fit of a statistical model. In our case, it was used to select the best model from the candidate models generated during parameter tuning. Lower AIC values indicate a better model fit. We prevented over fitting by eliminating the lowest AIC if it is away from 3 standard deviation from AIC.

  We choose AIC over to improve prediction accuracy, model exploration, smaller sample sizes, or when comparing non-nested models. AIC is suited for multi model inference.
- **Endog Regressors:** Endog regressors are endogenous variables used in SARIMA models. These are the variables that are dependent on other variables within the model.
- **Exog Regressors:** Exog regressors are exogenous variables used in SARIMA models. These are external variables that can have an influence on the time series data being analyzed. In your case, "Google search indices" served as exogenous regressors, potentially adding valuable information to the model.

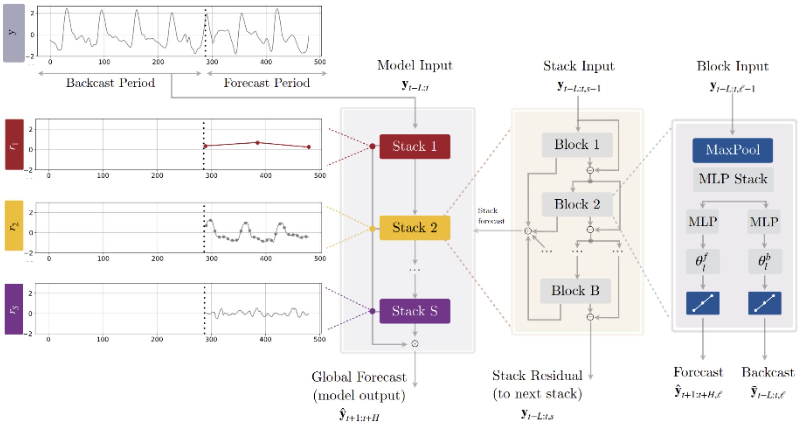

obtain sequence of data
Differentiation to get parameter d
stationarity test using ADF
non-stationary
MLE
NO
converge?
YES
Fit and Predict
by taking the model with lowest AIC

Neural Hierarchical Interpolation for Time Series Forecasting

- N-HiTS stands for Neural Hierarchical Interpolation for Time Series. It is an extension of the N-BEATS model, which is a neural network-based model for time series forecasting.
- The target variable is chosen as 'Pilgrims,' representing the number of pilgrim arrivals.
- Various features, such as 'tirupati rooms,' 'tirupati temple history and others, are selected for use as covariates.

Model Architecture and working:

- The model architecture of N-HiTS is based on the N-BEATS model, which consists of several stacks of blocks. Each block is a multi-layer perceptron (MLP) that predicts the coefficients of a basis expansion for the input time series. The blocks are connected by forward and backward residual links, which allow the model to learn different levels of abstraction and decomposition for the time series.



- N-HiTS extends the N-BEATS model by introducing two novel techniques: hierarchical interpolation and multi-rate data sampling. Hierarchical interpolation means that the model splits the input time series into different frequency components and predicts each component separately, then combines them to form the final forecast. This allows the model to capture the multi-scale structure and dynamics of the time series. Multi-rate data sampling means that the model samples the input time series at different rates depending on the frequency component, which reduces the computational cost and memory usage.
- Each stack has a different sampling rate, which is determined by a max-pooling layer at the beginning of each stack. The output of each stack is interpolated to match the original sampling rate and added to the final forecast. The model also outputs a backcast, which is used to compute the residual error and update the weights.

- Our NHiTS model is employed for time series forecasting with the following parameters:

  Input Chunk Length: 1
  Output Chunk Length: 'k' weeks
  Number of Blocks: 10
  Number of Epochs: 20

  Model performance:
- We evaluate the accuracy of our approach using mean absolute error (MAE) and mean squared error (MSE) metrics, which are well-established in the literature

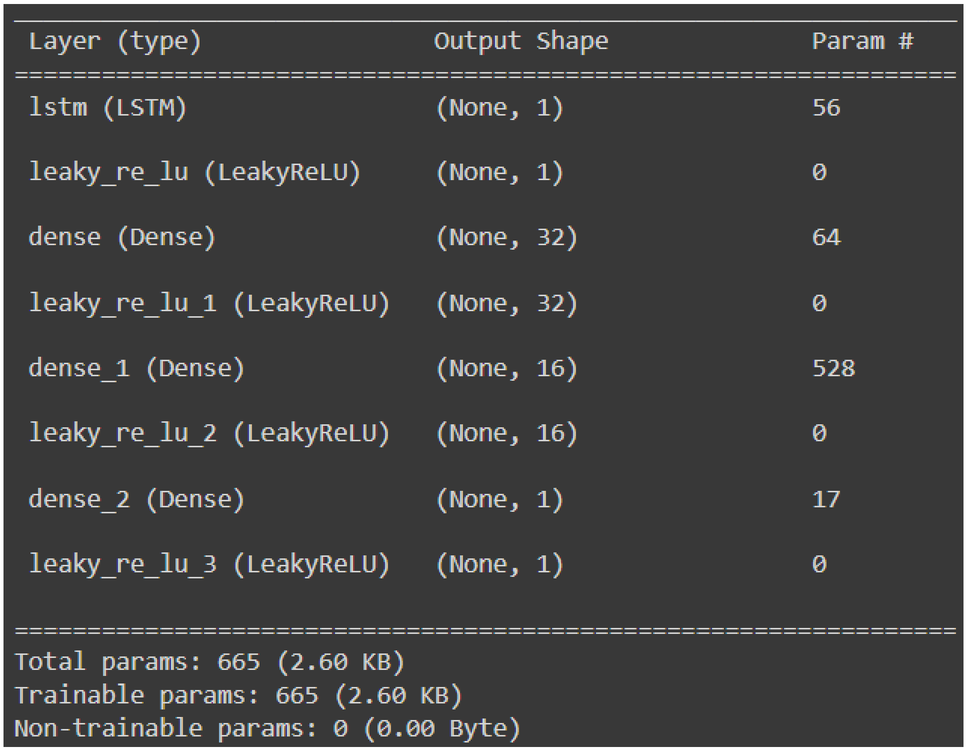
$$\mathrm{MSE} = \frac{1}{H}\sum_{\tau=t}^{t+H}\left(\mathbf{y}_{\tau} - \hat{\mathbf{y}}_{\tau}\right)^{2}, \qquad \mathrm{MAE} = \frac{1}{H}\sum_{\tau=t}^{t+H}\left|\mathbf{y}_{\tau}-\hat{\mathbf{y}}_{\tau}\right|$$

- Model accuracy and plots are mentioned in the annexture.

ML models
LSTM Model :

The LSTM model used in this prediction is designed as a sequential neural network. Below is the architecture of the model

LSTM Layer: The LSTM layer is the primary component for sequence modeling. It is responsible for capturing temporal dependencies in the input data.
Activation Function: Leaky Rectified Linear Unit (LeakyReLU) activation functions are applied after each layer. LeakyReLU is used to introduce non-linearity while preventing the vanishing gradient problem.
Dense Layers : Two dense layers follow the LSTM layer. The first dense layer contains 32 units, and the second has 16 units. These layers are meant to extract higher-level representations from the LSTM output.
Output Layer: The output layer is a single dense unit, which produces the predicted tourist count.
Optimizer: The Adam optimizer with a learning rate of 0.001 is employed for gradient descent. Adam is known for its efficiency and effectiveness in training deep neural networks.



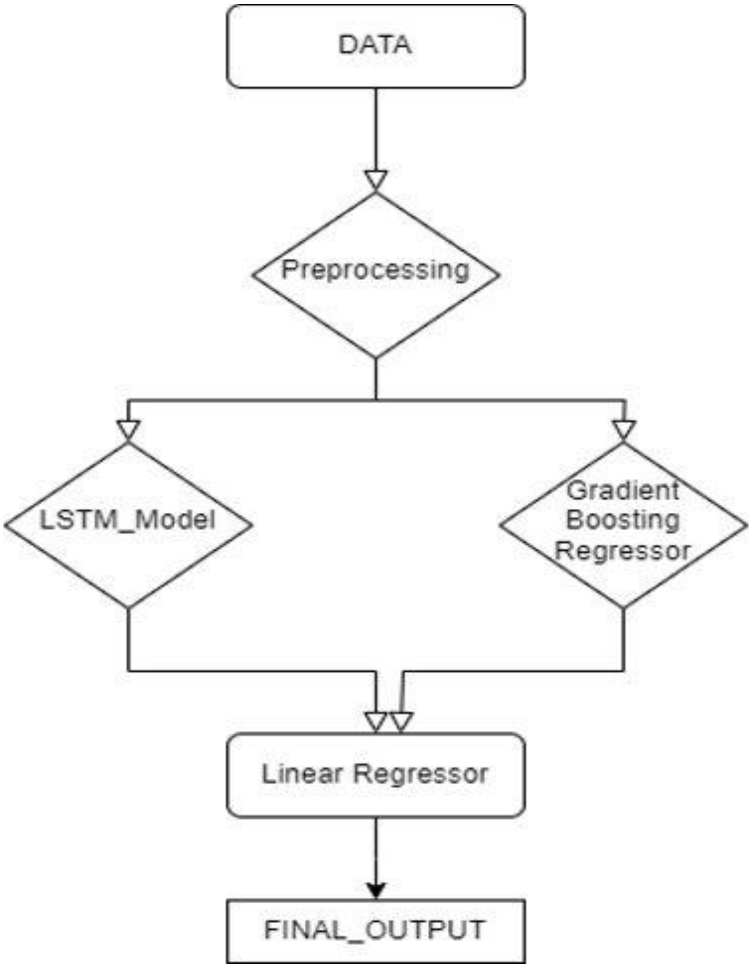
```
 Layer (type)                Output Shape              Param #   
=================================================================
 lstm (LSTM)                 (None, 1)                 56        
                                                                 
 leaky_re_lu (LeakyReLU)     (None, 1)                 0         
                                                                 
 dense (Dense)               (None, 32)                64        
                                                                 
 leaky_re_lu_1 (LeakyReLU)   (None, 32)                0         
                                                                 
 dense_1 (Dense)             (None, 16)                528       
                                                                 
 leaky_re_lu_2 (LeakyReLU)   (None, 16)                0         
                                                                 
 dense_2 (Dense)             (None, 1)                 17        
                                                                 
 leaky_re_lu_3 (LeakyReLU)   (None, 1)                 0         
                                                                 
=================================================================
Total params: 665 (2.60 KB)
Trainable params: 665 (2.60 KB)
Non-trainable params: 0 (0.00 Byte)
```

Gradient Boosting Regressor:

- Gradient Boosting is an ensemble learning technique used for regression and classification tasks.
- It combines multiple weak learners (typically decision trees) to create a strong predictive model.
- Gradient Boosting works in an iterative manner, where each new tree corrects the errors of the previous one.
- Trees are added sequentially, and the process continues until a predefined number of trees is reached or a performance threshold is met.
- Gradient Boosting minimizes the loss function by iteratively optimizing the model's predictions.
- It calculates the gradient of the loss function with respect to the model's predictions and updates the model to minimize the loss.
- Key hyperparameters in Gradient Boosting include the learning rate, tree depth, and the number of tree.
- We used "ensemble.GradientBoostingRegressor(**params)" from Sci-kit learn library.

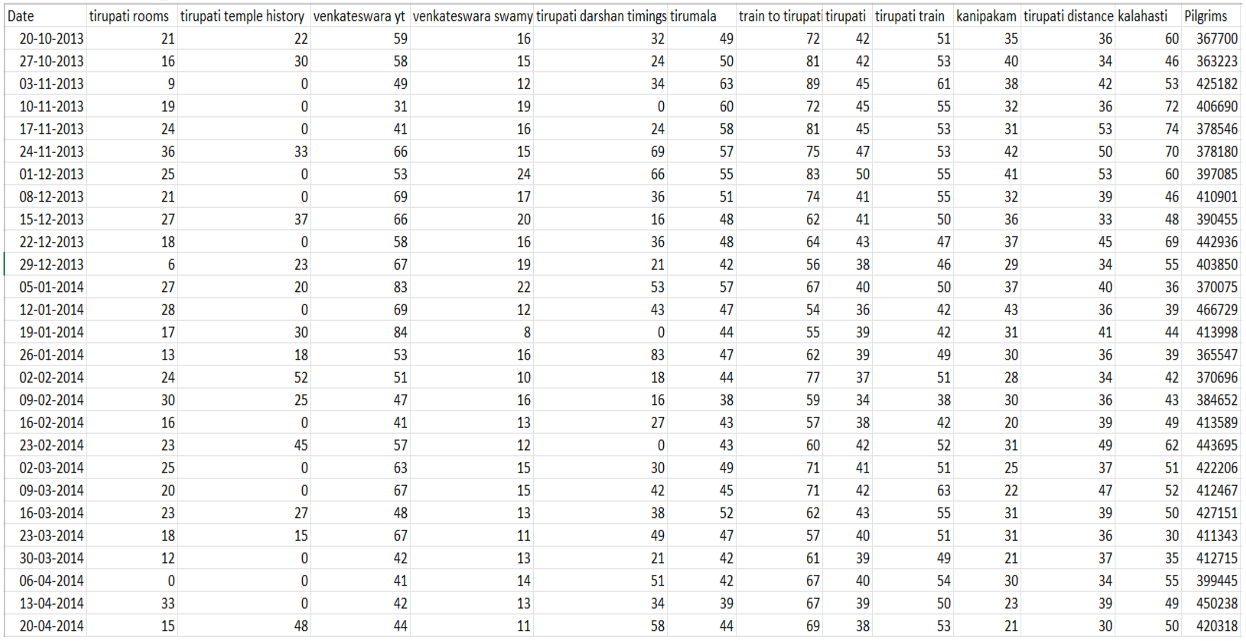
Ensemble Model:

- We ensembled an LSTM Model and a Gradient Boosting regressor to achieve a more robust and generalized model.
- The weights for the LSTM Model and Gradient Boosting Regressor in the final ensemble regression is 0.46 and 0.56 respectively

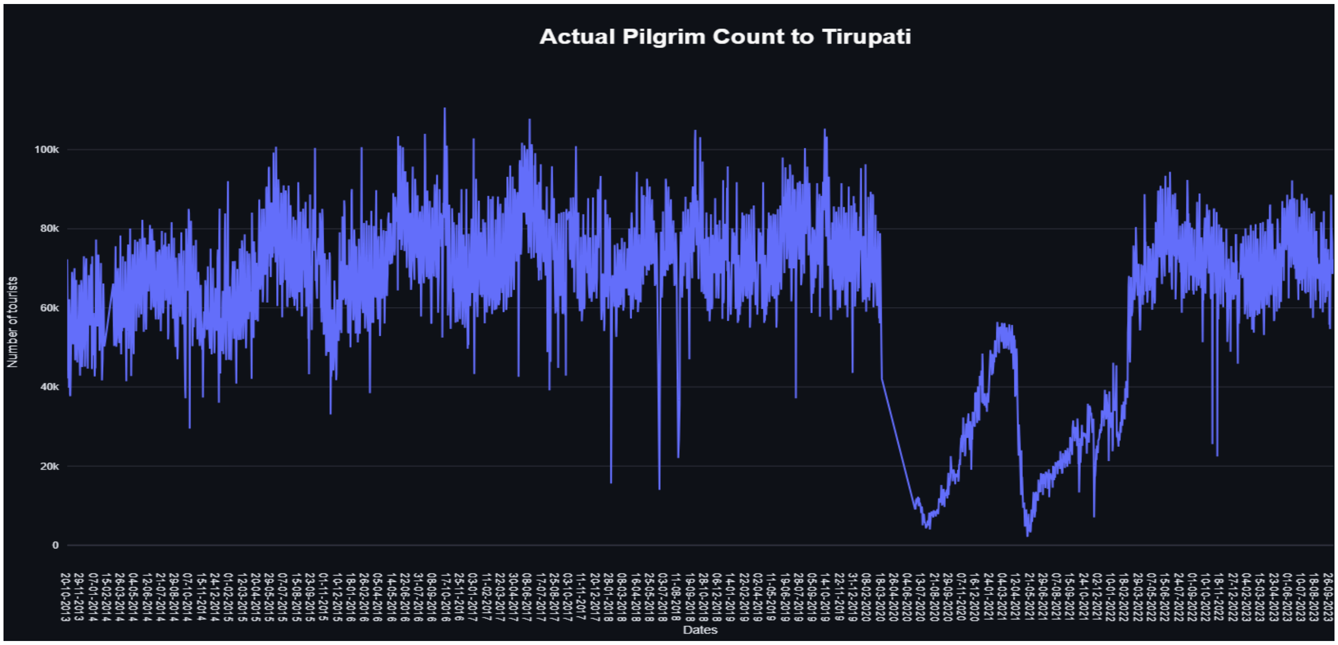

DATA
Preprocessing
LSTM_Model
Gradient Boosting Regressor
Linear Regressor
FINAL_OUTPUT

Annexure

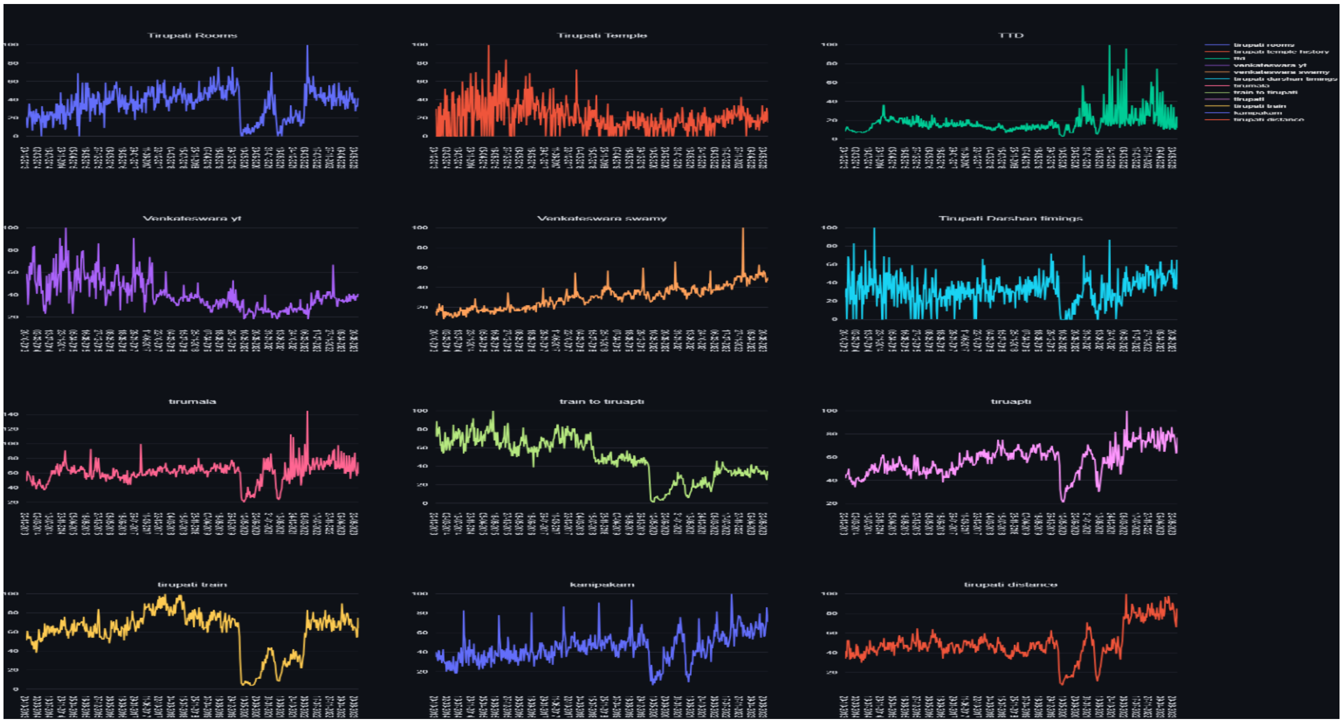
Data description- A brief overview



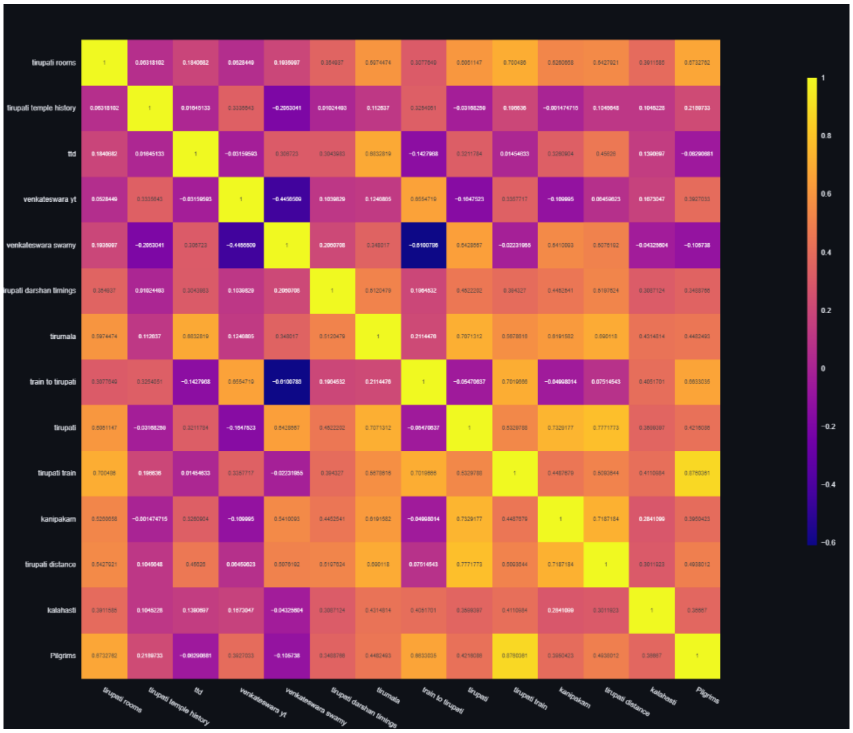
Date	tirupati rooms	tirupati temple history	venkateswara yt	venkateswara swamy	tirupati darshan timings	tirumala	train to tirupati	tirupati	tirupati train	kanipakam	tirupati distance	kalahasti	Pilgrims
20-10-2013	21	22	59	16	32	49	72	42	51	35	36	60	367700
27-10-2013	16	30	58	15	24	50	81	42	53	40	34	46	363223
03-11-2013	9	0	49	12	34	63	89	45	61	38	42	53	425182
10-11-2013	19	0	31	19	0	60	72	45	55	32	36	72	406690
17-11-2013	24	0	41	16	24	58	81	45	53	31	53	74	378546
24-11-2013	36	33	66	15	69	57	75	47	53	42	50	70	378180
01-12-2013	25	0	53	24	66	55	83	50	55	41	53	60	397085
08-12-2013	21	0	69	17	36	51	74	41	55	32	39	46	410901
15-12-2013	27	37	66	20	16	48	62	41	50	36	33	48	390455
22-12-2013	18	0	58	16	36	48	64	43	47	37	45	69	442936
29-12-2013	6	23	67	19	21	42	56	38	46	29	34	55	403850
05-01-2014	27	20	83	22	53	57	67	40	50	37	40	36	370075
12-01-2014	28	0	69	12	43	47	54	36	42	43	36	39	466729
19-01-2014	17	30	84	8	0	44	55	39	42	31	41	44	413998
26-01-2014	13	18	53	16	83	47	62	39	49	30	36	39	365547
02-02-2014	24	52	51	10	18	44	77	37	51	28	34	42	370696
09-02-2014	30	25	47	16	16	38	59	34	38	30	36	43	384652
16-02-2014	16	0	41	13	27	43	57	38	42	20	39	49	413589
23-02-2014	23	45	57	12	0	43	60	42	52	31	49	62	443695
02-03-2014	25	0	63	15	30	49	71	41	51	25	37	51	422206
09-03-2014	20	0	67	15	42	45	71	42	63	22	47	52	412467
16-03-2014	23	27	48	13	38	52	62	43	55	31	39	50	427151
23-03-2014	18	15	67	11	49	47	57	40	51	31	36	30	411343
30-03-2014	12	0	42	13	21	42	61	39	49	21	37	35	412715
06-04-2014	0	0	41	14	51	42	67	40	54	30	34	55	399445
13-04-2014	33	0	42	13	34	39	67	39	50	23	39	49	450238
20-04-2014	15	48	44	11	58	44	69	38	53	21	30	50	420318



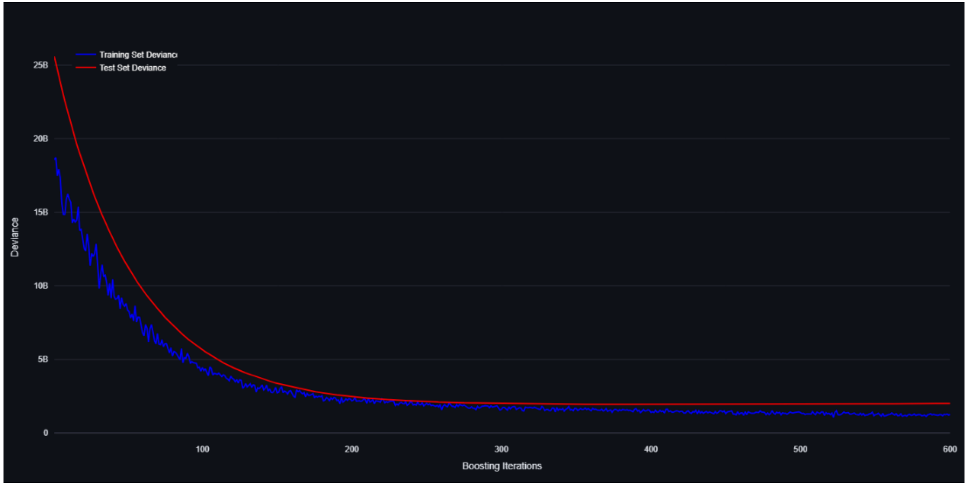
Features Used



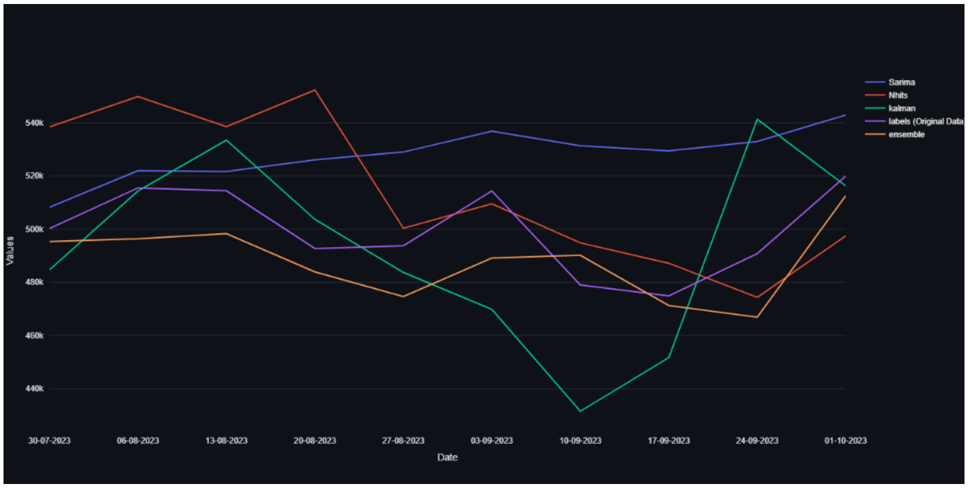
Heat map of correlation among features



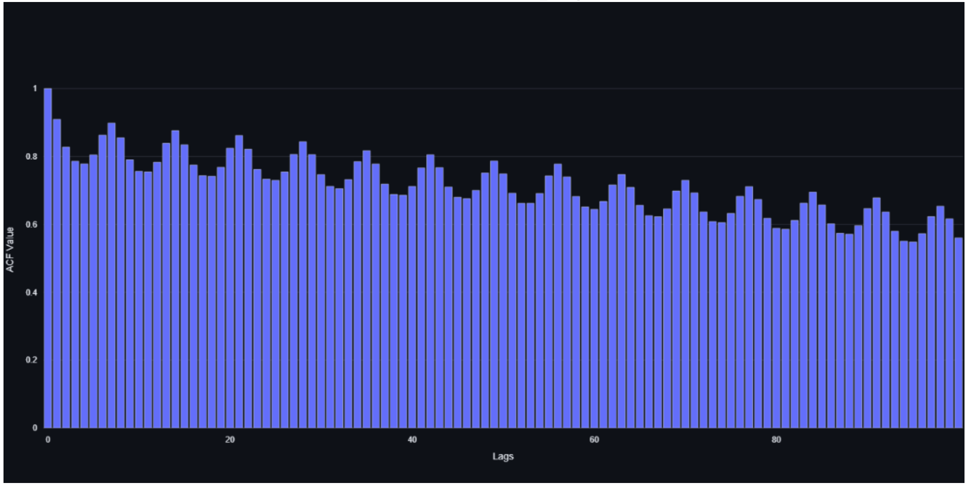
Xgboost deviance



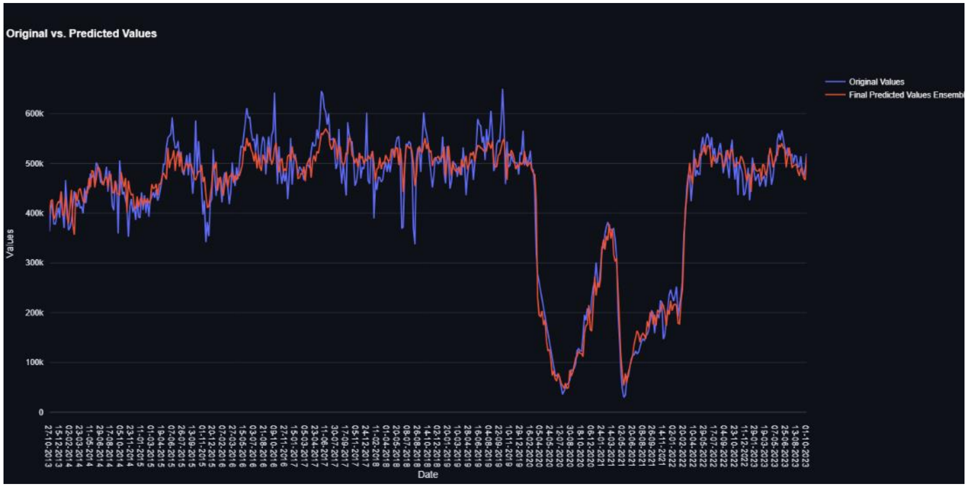
Predictions of various ML models



Auto correlation Plot of pilgrim Data



Ensembled model predictions



Model Accuracies

Model	MAPE
Gradient Boosting Regressor	12.06
LSTM	4.9
Ensemble Model	8.43
SARIMA	5.3
n-Hits	4.6

References:

https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/
https://iopscience.iop.org/article/10.1088/1757-899X/394/5/052024

https://arxiv.org/abs/2201.12886
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8258439