Booking Price Forecasting: Smart Models for Revenue Mangement

Alexandra Gladkova Christina Saju Hao Lun Rong

Kushwanth Sai Kolli Ladan Asempour

I. Introduction

This report contains an in-depth analysis of a hotel booking dataset, where we will employ time series analysis as well as various statistical models and machine learning models, including neural networks, to forecast the behaviour of the booking price. Accurate prediction of price is a crucial statistic when it comes to revenue management and operation planning for the hospitality industry, and this will help create a better understanding of price prediction especially in an industry where seasonality plays a big role. This report will go over the process of the data exploration and key insights obtained about the dataset, insights obtained from time series decomposition, as well as the results obtained from the various models we will be using, namely ARIMA, Holt's Linear Trend, and Prophet. Analysis of the results will be done using common statistical analysis metrics such as RMSE and MAE. We will also be discussing the application of time series modeling in this industry versus others and seasonality may affect it, and what adjustments may need to be made in different industries. Lastly, we will go over the interactive Streamlit web application we will be launching the code with, which will allow users to view the insights from our time series decomposition and price forecasting models in an interactive format.

II. Dataset and key insights from EDA

A. Data Description

Disclaimer*: A large portion of the EDA process was taken from a previous project that was worked on where we were aiming to target hotel booking cancellations as well as prediction price. As discussed with the professor, it was recommended to use the same dataset as previously. Due to this there will be a large overlap between reports, mostly concerning the first page where EDA is focused, proper citation to the previous project will be included.

The dataset we used in for our study was obtained from Kaggle. It contains a total of 119,390 samples and includes 32 columns, with a variety of attributes that offer valuable insights into the booking process. Key columns include the type of hotel, whether the booking was canceled (represented by the "is_canceled" attribute), lead time (the duration between booking and arrival), and detailed arrival date information such as the year, month, week number, and day of the month.

The dataset also includes data about guests like the number of adults, children, and babies, as well as the meal plan selected by guests. Additionally, it captures the country of origin, the market segment, and the distribution channel used for the booking. To understand guest behavior, the dataset provides information on whether the guest is a repeat visitor, the number of previous cancellations, and the number of previous bookings not canceled.

Further details include the reserved and assigned room types, the number of booking changes, and the deposit type chosen by the guest. It also includes agent and company details, the number of days a booking was on the waiting list, and the customer type. Financial data such as the Average Daily Rate (ADR), the number of car parking spaces required, and the total count of special requests are also captured. The dataset concludes with reservation status and the status date.

B. Preliminary Data Exploration

The purpose of data exploration was to get a better grasp of the data, identify what could be useful, and decide which columns should be removed or adjusted due to missing information.

We started by checking for missing values across the columns. We found that the **Country** and **Children** columns had only a small amount of missing data. To avoid these missing values, the **Children** column had the missing values replaced with the median, as it works better with heavily skewed data., and the **Country** column had been replaced with the mode.

However, we noticed that the **Company** and **Agent** columns had a larger amount of missing data. After examining these columns more closely, we realized that the missing values likely meant that there was no agent or company associated with the booking. Instead of removing these columns, we decided to convert them into binary columns. We marked the rows with non-missing values as 1 (indicating there was an agent or company involved) and the rows with missing values as 0 (indicating there was no agent or company).

C. Data Preprocessing and Feature Engineering

For this dataset, a large portion of the data was hard to interpret in a way that could be used for our Price Prediction Models. As a lot of the data was in text-based format, which could be used for the classification models, the data had to be transformed in a way that was usable for both models. To do so, a lot of the features had to be transformed into binary columns. The most notable piece of data was that there were 177 countries in the dataset. We had to change this feature to something that could be more easily represented and decided to classify these into more general groups as continents, from here we were able to create 7 dummy variables instead of 177. Other features such as the agency or company that the booking was done by, as well as the type of meal plan for the booking were also made into numerical data. Lastly, our target variable was created by multiplying the ADR with the number of days

of the stay. While we know that this information would not be completely accurate, it is a good benchmark and estimate for the values which we would be predicting for.

After all columns were in place, we inspected the data to check for outliers, and while we did see a large amount of outlier data which could affect our results, we believe that it is important for our scenario to include outlier data as outlier data in booking datasets often reflect fringe cases such as holiday seasons. In addition, these outliers do affect overall demand, potentially affecting the Average Daily Rate for new bookings in that time period. For this reason, we have kept most of the outliers, however there is an understanding that this would reduce the accuracy of our models.

However, to better ensure the dataset was suitable for our modeling, we applied the Yeo – Johnson transformation to reduce skew, the skewness was tested for every column, and applying the transformation would overall help to reduce the effect of the outliers in the rest of the analysis and prediction modeling.

D. External Factors and Seasonality

Booking prices in the hospitality industry could very well be affected by numerous external factors. Holidays are a frequent contributor to increased business for hotels. In addition, changing seasons may make a vacation more, or less desirable depending on the temperature or other potential weather conditions. Economic or sociopolitical factors could also affect travel to countries, changes in the currency exchange or changes to travel regulations or country relations would heavily impact travel. With all this being mentioned, changes in time do potentially impact travel and hospitality industries, which is something that will be addressed below.

The final step which needed to be taken was to check for stationarity. To do so we applied the Augmented Dickey Fuller (ADF) test on the booking price data. The results for the ADF test indicated non-stationarity, meaning that target variable, Average Booking price, showed a strong enough statistical difference throughout time. Because the ADF test showed non-stationarity to warrant changes. Therefore smoothing will be necessary for some models such as ARIMA which will be conducted later in the project. The results of the stationarity test indicate that there is a time-based influence from either holiday or weather conditions which plays a part in booking price. From this insight, this leads us to believe some factors such as the location of the hotels, are more heavily affected by climate and weather changes. This lines up with the initial information that the hotel data came from a beach resort, and a city hotel respectively.

III. Time Series Decomposition

A. Applying Seasonal Decompose

Time Series Decomposition was used on the booking price data to get a better understanding of the nature of the booking price across time. Both Additive and Multiplicative seasonal decomposition was applied to the data. This allowed us to look at the data in three distinct components, Trend, Seasonal, and the Irregular or Residual component, also known as Noise. By doing so, the long term trend or movement of the data, the trend that coincides with the calendar year, and the irregular components of the data can give us insights into what is really happening to the booking price over a given year.

B.Additive vs. Multiplicative Comparison

The results for both the additive and multiplicative decompositions are shown in the figures below:

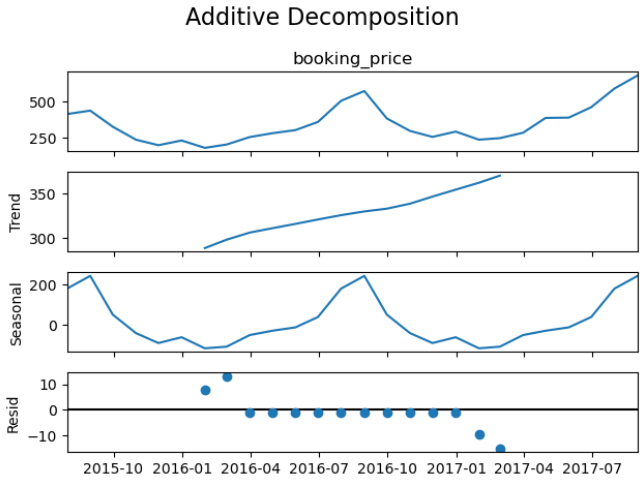


Figure 1 - Additive Decomposition Results

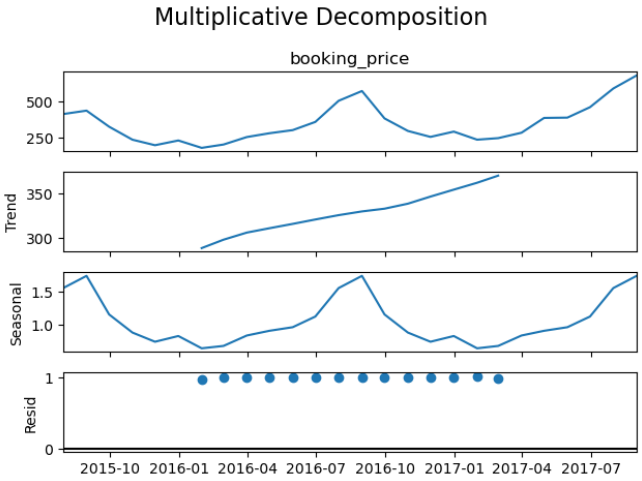


Figure 2 - Multiplicative Decomposition Results

The results of the decomposition from 2015 to 2017 show a easily readable upward trend, the seasonal pattern shows a peak in the summer months, and a dip around the middle of winter, showing fluctuations in season. While both models show small noise inputs, indicating that there are not a lot of random factors shifting the data, due to the nature of the multiplicative model, multiplying the three factors together instead of adding, it is better to go with the multiplicative decomposition as the seasonal trends scale well with the changes in time.

In addition while residuals are small for both models, the multiplicative ones show less overall fluctuation in noise.

C. Business Insights from Decomposition

The results of the seasonal decomposition give a couple insights into the behaviours of patterns from the hotel datasets. The trend component reveals a steady upwards trend in booking price, while the seasonal compontent shows a up and down pattern depending on the season, with the winter holiday season having a lower booking price than the summer months, with the spring and fall months equalizing in between. Being aware of these patterns gives value add to hotels by knowing how to properly plan events for the most customer intake, as well as save budget in employees or other services in downtime of customers and booking prices.

IV. Forecasting Models & Performance Evaluation

A. Data Splitting and Preprocessing

The last steps in the data preprocessing involved differencing the data and making the data stationary, once the data was transformed, the ADF test was done on the dataset again to check for stationarity. The ADF test initially returned a p-value of 0.728, while the new test returned a value of 0.001, much lower than the 0.05 threshold. Indicating that the data was now stationary and allowed to be used in our time series price prediction models.

Due to the nature of the data being categorized in months, the data was not split with the conventional 70% - 30 % split. The data contained 25 months' worth of data and was split with the first 19 months as the train, and last 6 as the test set. This equaled roughly 76% of the data being in the training set and the other 24% in the test set. MinMax Scaler was used on LSTM for modeling capabilities.

B. Forecasting Models

The Forecasting models we will be using are listed and explained below:

- Autoregressive Integrated Moving Average (ARIMA), An autoregressive model that accounts for moving average based on time [3].
- Holt, Also known as Double Exponential Smoothing model.
- Prophet, Meta's hybrid forecasting model [2].
- Long Short Term Memory (LSTM), a neural network,

C. Graphs and Model Evaluation Metrics

The results of each Model are presented with a graph visualizing the forecasted prediction, as well as error metrics in a table. After all results are listed we will be analyzing the results.

ARIMA:

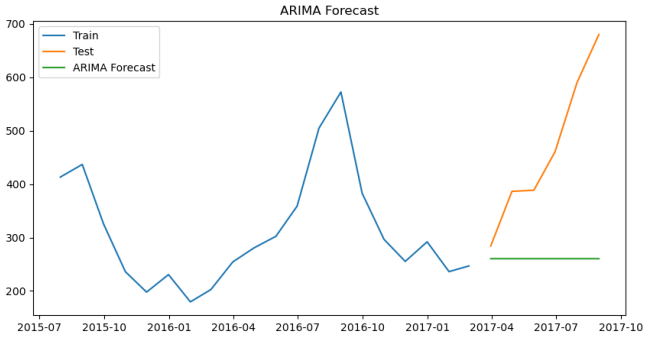


Figure 3 - ARIMA Forecasting Results

ARIMA	
MAE	204.441
RMSE	244.020
MAPE	39.150 %

Table 1 - ARIMA Error Metrics

HOLT:

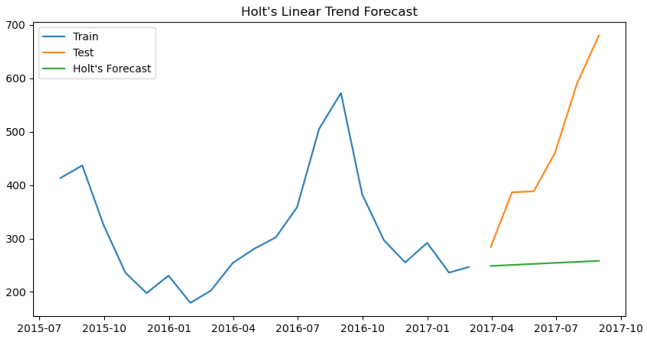


Figure 4 - Holt Forecasting Results

Holt	
MAE	211.437
RMSE	201.843 248.221
MAPE	40.998 %

Table 2 - Holt Error Metrics

Prophet:

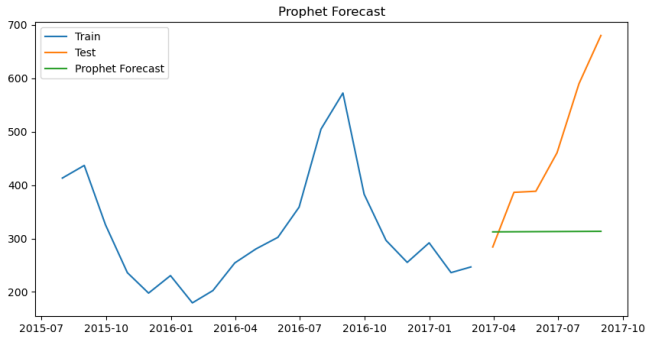


Figure 5 - Prophet Forecasting Results

Prophet	
MAE	161.353
RMSE	201.843
MAPE	30.220 %

Table 3 - Prophet Error Metrics

LSTM:

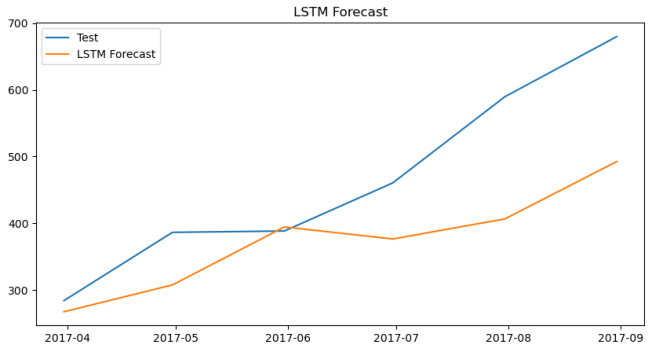


Figure 6 - LSTM Forecasting Results

LSTM	
MAE	92.685
RMSE	117.061
MAPE	17.450 %

Table 4 - LSTM Error Metrics

D. Comparative Analysis

Comparing both the visual insights, as well as the error metrics for all forecast models, it is clear that LSTM exceeds in performance in terms of both accurately matching the test data, as well as performing the lowest in all of the Error metrics. With LSTM only having a Mean Absolute Error (MAE) of 92.685, and an MAPE of 17.45%, the LSTM model performs exceedingly well. Only showing an average of $92 from the actual booking price.

Prophet also performed rather well based on the error metrics, however, visually you can see that it failed to capture any of the trends in seasonality. The green line shows a slight upward trend however does not move according to the time of the months. This may suggest that the model underfit the data, and needs increased sensitivity to the factors in our dataset. Another reason that the model may have failed to capture the seasonal trends is the lack of seasonal data for that time period. Due to the length of time the data was recorded, only totalling 25 months, it has only captured a little over 1 year cycle of data, as the test set starts in April, 2017, only being able to reference the same season the previous year.

ARIMA and Holt both displayed the same visual line as the Prophet model, however the prediction it made was much further away from any of the actual data. The model undershot the predictions and this could potentially be due to the model overfitting due to how the model trends upwards. The results for these two suggest that the models had issues with the data or was not well tuned to accurately forecast the price.

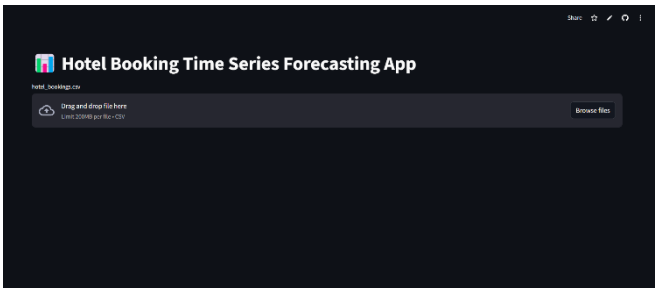
Due to the nature of LSTM and it being better suited at capturing short term changes in data, the results displayed make sense. However, if given data of a longer time period to observe seasonality, it is believed that Prophet would outperform LSTM.

V. Screenshots and Explanation of Streamlit Web App

A Streamlit web app was created to enable users to interact and explore our forecasting models. This app allows users to upload and analyze booking data without prior knowledge of the python code or other technical software, making the project accessible to both technical and non-technical stakeholders.

A. Application Overview

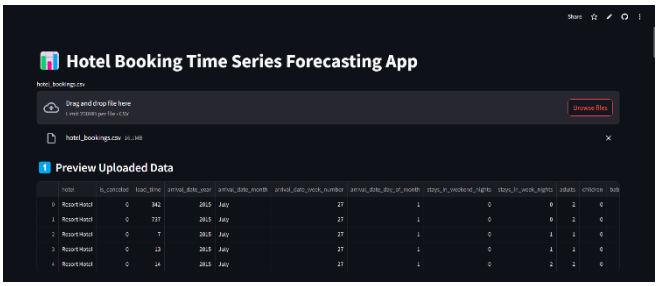
The app, titled Hotel Booking Time Series Forecasting App, follows a sequential process beginning with a data upload of the original dataset, followed by preprocessing, visualization, decomposition, and forecasting. It allows different grouping and filter options of the data for various insights of the data.



B. Data Preprocessing and Transformation

Upon uploading the dataset, the app performs all the preprocessing procedures mentioned in the report, allowing for a clean dataset to analyze. A Yeo-Johnson transformation

is applied to relevant columns to reduce the effect of outliers and improve model performance.



C. Time Series Visualization and Decomposition

Users can review the plotted monthly data, and have the option to choose between additive and multiplicative decomposition. The slider at the top allows for the seasonal period to adjust in length, and the resulting plots are then displayed.

D. Forecasting and Evaluation

Once a decomposition is assessed and selected, the user can select a forecasting model and visualize its predictions against the actual data. The app automatically calculates and displays error metrics including Mean Absolute Error (MAE) and Mean Squared Error (MSE), assisting allowing users to compare performance between different models.

REFERENCES

[1] A. Gladkova, C. Saju, H. L. Rong, L. Asempour, and V. Sekhri, Evaluating Machine Learning Models for Hotel Booking Cancellation and Pricing Prediction, Unpublished class project, 2025.

[2] Laptev, N., Hewamalage, H., & Triebe, O. (2021, November 30). Neuralprophet: The neural evolution of meta's prophet. AI at Meta. https://ai.meta.com/blog/neuralprophet-the-neural-evolution-of-facebooks-prophet/

[3] Noble, J. (2024, May 24). What are Arima Models?. IBM. https://www.ibm.com/think/topics/arima-model