A logo for college computing

Description automatically generated

# HotelRank: Elevating Revenue Performance Through Machine Learning and Deep Learning Techniques.

|  |  |
| --- | --- |
| *Student Name:* ***Giorgio Zoppi*** |  |
| *Student Number:* ***sba2301*** |  |
| *Module Name:* ***Strategic Thinking (HDip in Data Analytics - Feb 2024 - HCI cohort)*** |  |
| *Assignment Title:*  ***Capstone Project*** |  |
| *Assessment Due Date: 7****th November 2024*** |  |
| *Date of Submission 7****th November 2024*** |  |

**Declaration**

By submitting this assessment, I confirm that I have read the CCT policy on academic misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source.

I declare it to be my own work and that all material from third parties has been appropriately referenced.

I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Contents

[HotelRank: Elevating Revenue Performance Through Machine Learning and Deep Learning Techniques. 1](#_Toc180884612)

[*Module Name:* ***Strategic Thinking (HDip in Data Analytics - Feb 2024 - HCI cohort)*** 1](#_Toc180884613)

[Introduction 4](#_Toc180884614)

[Understanding hotel room demand over time. 4](#_Toc180884615)

[Prediction of booking cancellations. 4](#_Toc180884616)

[Online hotel reputation. 4](#_Toc180884617)

[Problem Domain and Objectives 4](#_Toc180884618)

[Demand forecasting. 4](#_Toc180884619)

[Booking Cancellations. 5](#_Toc180884620)

[Online Reputation. 5](#_Toc180884621)

[HotelRank Score. 5](#_Toc180884622)

[Scope. 5](#_Toc180884623)

[Data Sources. 6](#_Toc180884624)

[Business Scenario and SWOT Analysis of Hotel Rank. 6](#_Toc180884625)

[Data Cleaning 7](#_Toc180884626)

[Demand Score Computation: Forecasting the Booking. 8](#_Toc180884627)

[Data Preparation 8](#_Toc180884628)

[Outlier and Anomaly Detection 8](#_Toc180884629)

[Stationarity Check. 8](#_Toc180884630)

[Model time serier forecast: Prophet. 8](#_Toc180884631)

[Baseline model: 8](#_Toc180884632)

[Hyperparameter and crossvalidation tuning with Optuna 8](#_Toc180884633)

[Results 8](#_Toc180884634)

[Model time serie SARIMAX: 8](#_Toc180884635)

[Conversion to stationary series 8](#_Toc180884636)

[Baseline model: 8](#_Toc180884637)

[Hyperpameter tuning in SARIMAX 8](#_Toc180884638)

[Compute Demand Score : algorithm selection. 8](#_Toc180884639)

[Cancellation Score: Why are important? 8](#_Toc180884640)

[Data Cleaning 8](#_Toc180884641)

[Feature selection 8](#_Toc180884642)

[Univariate Analysis 8](#_Toc180884643)

[Anomaly Detection 8](#_Toc180884644)

[Baseline Model 8](#_Toc180884645)

[PCA and Random Forest Model 8](#_Toc180884646)

[Cross Validation 8](#_Toc180884647)

[PCA and SVM 8](#_Toc180884648)

[Cross Validation 8](#_Toc180884649)

[Compute the demand score: algorithm selection. 8](#_Toc180884650)

[Review Score: Taking in account social reputation for Hotels. 9](#_Toc180884651)

[Data Preparation 9](#_Toc180884652)

[TripAdvisor Scraper Design 9](#_Toc180884653)

[BERT and Distillbert 9](#_Toc180884654)

[ReviewScore Computation. 9](#_Toc180884655)

[Data Engineering Architecture. 9](#_Toc180884656)

[Model Deployment Design. 9](#_Toc180884657)

[Accomplishment and future work. 9](#_Toc180884658)

[Ethical Considerations. 9](#_Toc180884659)

[References 10](#_Toc180884660)

# Introduction

Revenue management (Ivanov, 2014) is a very important to make profits in the hotel industry, three main factors play an important role to get it right:

* Hotel room demand over time.
* Prediction of booking and cancellations.
* Online hotel reputation.

### Understanding hotel room demand over time.

Accurate demand forecasts enable hotels and revenue managers to adjust prices dynamically, thus maximizing revenue potential. However, factors such as location, cultural events, weather, seasonal patterns and more, significantly impact on optimal hotel room forecasting (Apostolos, 2021).

### Prediction of booking cancellations.

Booking cancellations (Nuno Antonio, 2019) are an issue for the hotel revenue manager because they make harder to predict the number of booked rooms. A common pattern to address this issue is trying to achieve the overbooking by lowering the price but when overbooking is real, it is a problem because it damages hotel reputation and revenue. People might find themselves without a room and complain online. So, from a revenue manager perspective, it is important having cancellations predictions or at least an tool to assess them.

### Online hotel reputation.

Using the framework developed in (Diana-Jens & Rodríguez Ruibal, 2015) , we define what means online hotel reputation. Online reputation is *“the result of what clients, former clients, future clients, employees, etc. say, write and communicate to another anywhere in the internet social media based on their perceptions and experience in any moment of their relationship, direct or indirect, with the brand”*. As demonstrated in a Chinese study on ten luxury hotels (Wang, et al., 2023), negative online reviews can lead to a low booking rate that takes months to recover, so online reputation is an important factor to consider.

Our focus is therefore to create a hotel ranking algorithm called **HotelRank** that considers all these three factors in a weighted manner.This research project will be executed in strict collaboration with Blastness Spa(Blasteness.com, 2023)**,** the #1 provider in Italy for luxury hotels with a portfolio of over nine hundred hotels.

# Problem Domain and Objectives

Our main goal is to increase the profit for our customers that are mainly hotel revenue managers providing a way to compare with competitors. To achieve our main goal, we want to put our focus on creating models for demand forecasting, booking cancellation and online reputation and then combining them to create a global hotel score to understand hotel performance.

## Demand forecasting.

Our demand challenge here is to validate our dataset, cleaning the data, selecting the model, evaluate his performance and predict results. The hypothesis is that a time series has a trend and use that trend to compute the demand score to differentiate each hotel.

## Booking Cancellations.

Using the same process as we plan to use for the demand, here we want to understand how cancellations affect the demand. We assume that the hospitality market impacts on the hotel performance so we classify the reservation status and then just later we can check the cancellation.

## Online Reputation.

Here our objective is to collect TripAdvisor data for the hotels in our dataset and see how the reviews and comments, providing our own reputation score. The hypothesis to validate here is that higher is the rank, higher is the revenue as some studies state. (Diana-Jens & Rodríguez Ruibal, 2015). To reach our goal of scoring review we will use pretrained language models, specifically Hugging Face DistillBert, to create and train a model on Internet Movie Database and use it later TripAdvisor.

## HotelRank Score.

Once we’ve created the models, we can design an iterative process to compute **HotelRank** ranking score for each hotel. Let’s define **HotelRank** as weighted combination linear between those factors:

+ *w4 \* OptionalScore*

After model training, the correct weights **w1, w2, w3** are set (w1=1.5, w2=0.8, w3=0.5) and **w4** to zero and reserved for future uses. During the design we have decided to set fixed the weight and change only if future observations lead to wrong results. The idea is to apply this ranking just at most interesting Blastness customer and then generalize the process through a data engineering project that it is outside the scope of this study.

# Project Scope.

Project management CRISP-DM methodology will be used most of the project except during deploy where we switch to Scrumban (Alliance, 2017). An important point in CRISP-DM is that it is meant to be an iterative process since the construction of a model requires several cycles. Over the arc of two semesters the scope of the project we’ve tried to answer the following questions:

* Which is most accurate model for demand forecasting?
* How does cancellations affect revenue performance?
* How does hotel online reputation affect revenue performance?
* Can we determine how good we are respect our competitors?

In the table below we summarize key milestones we’ve reached. The deadlines might have some minor deviations due to the project complexity, so we’ve kept one month buffer at the end.

|  |  |  |
| --- | --- | --- |
| **Phase** | **Objectives** | **Milestone Deadline** |
| **Hotel Domain Knowledge Research** | Domain Analysis. Understand how ML techniques are used. Understand how Deep Learning are used in the domain. Report about domain knowledge. | **20th April 2024** |
| **Data Collection** | The dataset consists of the bookings of eight Italian luxury hotels in a two-year period. Data Collection from TripAdvisor. | **21st May 2024** |
| **Data Exploration.** | Explore common proprieties in the datasets. First cleaning the data, remove all Italian references. Visualization of the datasets. Understanding data patterns. | **4th June 2024** |
| **Feature Engineering** | Create a merged datasets and coded TripAdvisor Scraper. Cross reference hotel and user reviews. Select/Add/Remove features. | **15th June 2024** |
| **Cleaning Data** | Handling missing values. Handling Duplicates. Assure Data Consistency. | **20st June 2024** |
| **Integrate Data in Iceberg Tables** | Once the data is clean format in a query able data source to facilitate training. | **1st August 2024** |
| **Modelling: Demand and cancellation forecast models using ML.** | Creating models, training and evaluating their performance using an iterative approach using SARIMAX and Prophet algorithm for demand forecasting. Detect overfitting. K-Fold Cross validation to determine which model performs better. | **1st September 2024** |
| **Hyperparameters tuning.** | Tune ML and deep learning models hyperparameters and test visualization. | **15th October 2024** |
| **Compute Reputation Score** | Top reviews analysis and score computation between the hotel in the dataset using NLP classification algorithms. | **20th October 2024** |
| **Compute HotelRank.** | Select the models and weight for HotelRank and perform the computation on the dataset. | **25th October 2024** |
| **Reporting and Project Close.** | Project report with summary of the results indicating the chosen models and the process.  Project review: Retrospective document to indicate what went well and which are the areas of improvement. | **3th November 2024.** |

# Data Sources.

The data has been provided and released by Blastness.com on Creative Common License in the GitHub repository <https://github.com/CCT-Dublin/capstone-project-feb-2024-pt-giorgiozoppi> . The folder **hoteldataset** that contains bookings on eight luxury hotels, during a period of two years at least.

All work has been developed in Google Colab because the usage of Neural Network and fit of the pretrained language modules is expensive in a laptop environment.

Blasteness.com (Blasteness.com, 2023) has also provided the latitude and longitude of each hotel to cross reference with TripAdvisor API but this will not be disclosed for keeping the hotel name private.

We had to develop a TripAdvisor scraper in Go and released opensource in GitHub (<https://github.com/bcncpp/scraper>). Using the scraper we’re able to collect reviews for each hotel in the dataset, clean them and store in CSV files as shown in the picture below.

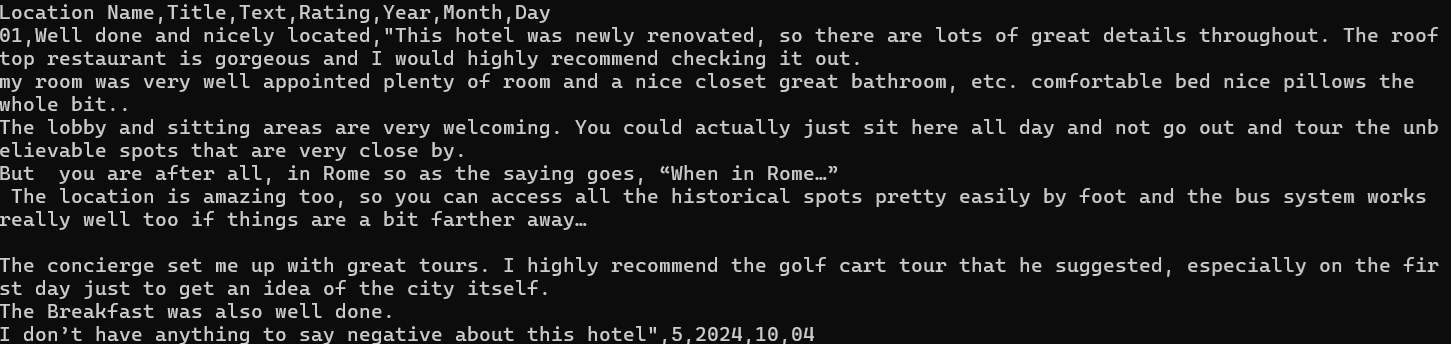


Figure 1 Hotel Review Format: Location (HotelID),Title,Text,Rating,Year,Month,Day

# Business Scenario of Hotel Rank.

The vision of this project is to provide a tool in the Italian market that allow our top Blastness customers increase their annual revenue. For this reason before starting the project we’ve looked at the business scenario of such study and done a SWOT Analysis (Wikipedia, 2024) . The picture above is the result of this investigation.

A close-up of several words

Description automatically generated

Figure 2 HotelRank SWOT Analysis

**HotelRank** and its parts (**DemandScore**, **CancellationScore**, **ReviewScore**) are interesting and challenging because theirs:

* **Strengths**: Our customers want a personalized ranking algorithm to tune better their performance and increase their revenue.
* **Opportunities**: Having a good reputation algorithm will allow the company to expand to new markets and proceed with the internationalization of the company. Furthermore, it will improve the product and increase internal knowledge about NLP and Machine Learning.
* **Weaknesses**: Handling the machine learning operation process and intensive training is costly at scale, market specificity, and the ranking can have a limited visibility to managers.
* **Threats:** Machine learning operational processes are complex and can be done in non a correct way. Due to the CPU/GPU intensive nature of this project all development and testing took place in Google Colab.

During this study and the future put in production we use our strengths to reach the appealing opportunities of this work and take care of the risks.

## Success Criteria.

The success of this project is measured in the ability of craft and use accurate machine learning and NLP models for computing smoothly the parts of HotelRank. Due to time and scope restrictions we’re not able to implement the data engineering part with data pipelines to automate all the flow.

# Data Cleaning and descriptive statistics.

The data has been provided by **Blastness** and it needed to be heavily cleaned. The data initially was more than fifteen features all named in Italian with each file per Hotel, so as first step we had to:

* Merging all files in a unique dataset.
* Rename the columns in English.
* Restrict the time frame from 2019 to 2023 of the booking due to low quality of the data.

A screenshot of a computer

Description automatically generated

Figure 3 Cleaned dataset

The resulting dataset was 11 features and 151857 booking for ten hotels. This process was not enough since all the data in the dataset was text, so we had to:

* Covert all text columns in numeric or date to respective numeric value or timestamp
* Check the feature **BookingDevice** and we’ve realized that most of the time is empty.

## Descriptive statistics.

During descriptive statistics process we’ve executed simple Pandas library (McKinney, 2024) commands and plotting to understand the nature of our cleaning dataset and weather is clean enough to work with it.

Our findings here are:

* We've 39345 booking in the period between 2019 and 2023.
* The average staying is 1.6 days for each booking.
* Most of the booking are confirmed or cancelled, modified bookings are just a minor percentuage.
* The medium booking revenue is 370 euros.
* The most used booking channel is Booking.com, but most of the booking channel were empty, so we must drop BookingDevice feature.

A screen shot of a computer program

Description automatically generated

Figure 4 Descriptive Statistics on Blastness Bookings.

To have a better idea we wanted to know more about the dataset and answer at the following questions:

* How frequent is a booking?
* Which between our customer client how had most revenue?
* Which is the season in which we've most room booked?

### How is frequent is a booking?

Reservation times follows a beta distribution with cycle with peak in summer whereas the purchases follow a binomial distribution. Hotel managers can’t predict the daily number of arrivals, but they learn as soon as they as people are coming, so they might adjust staffing around the day since arrival follow a beta distribution.

A graph of blue bars with red circle

Description automatically generated

Figure 5 Arrival Distribution

A graph with a red line

Description automatically generated

Figure 6 Purchase Distribution

The purchase date follows a binomial distribution, this mean that even some period of years has a higher probability than other, and this behaviour is cyclic. This tells us that the top season is summer, it might also depend on the Hotel locations, but for our customers client, summer is a good season for revenue opportunities.

### Which between our customer client how had most revenue?

A graph of a bar graph

Description automatically generated

There is a strong difference between all hotels, since the type of hotels, we’ve one customer that has more revenues than others. Some data related to Hotel001 is missing and invalid. This initial data is ready now we’ve to focus on our score computations, before that we’d like to share other findings that are evident:

- Rome is the city with higher possible revenue.

- Purchase Date and Arrival Date are two features highly correlated.

Now we focus on **Demand Score** data analysis, we treat the problem as time series forecasting, restricting to number of bookings in a time frame.

# Demand Score Computation: Forecasting the Booking.

## Data Preparation

Our dependent variable will be the **BookingCount**, that’s the number of customers arrived in the Hotel daily, we consider the Arrival as our independent variable.

The goal is to design a specific process, keeping mind the most valuable client of our customer is Hotel 008 that can we generalize later. So, the subproblem is a time series analysis for Hotel 008. Once we’ve solved this challenge we can generalize the methodology to the other hotels.

A screenshot of a graph

Description automatically generated

Figure 7 Density of BookingCount per Hotel

From the density of the target variable, we see that there are anomalies to check.

## Outlier and Anomaly Detection.

For the time series we’ve decided to investigate two algorithms: Facebook Prophet (Taylor & Letham, 2018) and SARIMAX. Whereas SARIMAX requires by design anomaly removal and stationary series, Prophet drops these requirements. To evaluate SARIMAX we’ve implemented an algorithm based on unsupervised learning (Bergia, 2021) for anomaly detection since it is very sensitive to stationarity and anomalies.

### What is an anomaly in a time series?

An anomaly in a time series refers to a data point or sequence of data points that significantly deviates from the expected patterns or trends typically observed in the data. These anomalies can manifest in various ways, such as abrupt changes in values, an increase in NULL values, missing data segments, or other irregular patterns that deviate from normal fluctuations that can make your prediction without any real value. As discussed in *Anomaly Detection in Time Series*: A Comprehensive Evaluation (Schmitd, et al., 2020) such anomalies can indicate significant events or issues within the system being analysed.

In our specific context, an anomaly might suggest that in some days, the demand exceeded the hotel's operational capacity so client can be rejected, or the hotel is quite empty, so we don’t need a lot of staff.

Numerous techniques for anomaly detection have been evaluated in the literature. During our research, we identified an effective algorithm that can be applied to the residual component of a seasonal-trend decomposition, utilizing the Isolation Forest method. We chose this approach because it is easy to automate, and it doesn’t require a lot of visual. The algorithm will be applied for each hotel dataset, and it is essentially based on the work on *Unsupervised Anomaly Detection on Server Metrics* (Bergia, 2021). Figure 6 below describes the algorithmic steps, some high view rationale of the steps:

A flowchart of a forest

Description automatically generated

Figure 8 Time Series Anomaly Removal

1. The normalization/scaling process is a requirement of isolation forest.
2. The decomposition has been done to show the residual, in general anomalies/errors (Schmitd, et al., 2020) are outside the trend of a time series.
3. From the residual we apply Isolation Forest that is adapted to find outliners, later we’ll go more in depth on Isolation Forest.
4. Once we have the outliers, we’re going to see which data points are far from others, if they are far from there is a higher probability that are error data points. So, we use clustering for this. We apply DB-SCAN. DB-SCAN two hyper-parameters, we followed an empirical approach setting eps=0.5 and the dimension to 20. There are better ways to tune eps and sampling dimension as explained in *Fast Density-Based Clustering with R* (Hansler & Piekenbrock, 2019)and in this paper (Schubert , et al., 2017) but we don’t want to add more complexiy
5. We select the data points that are far from the cluster and delete them.

Given the nature of the data and the specific requirements of this problem, we’ve excluded holiday periods from our anomaly detection algorithm. To accomplish this, we've created a Python script to filter out dates that fall within these periods. Our process initially filtered out 213 data points at Isolation Forest stage than later just 1 point. After filtering we can proceed to see the cleaned series behaviour.

## Descriptive Statistics

A graph showing a number of blue and orange lines

Description automatically generated

In the picture we can see the difference between the Booking Count, its 12 months mean, and 12 months standard deviation. We know from literature that time series data can have a well-known set of patterns. (Daniel, et al., 2023)

For this reason, is often helpful to split a time series into several components, each representing an underlying pattern category.

* Trend: A trend exists when there is a long-term increase or decrease in the data.
* A seasonal pattern occurs when a time series is affected by seasonal factors such as the time of the year or the day of the week. Seasonality is always of a fixed and at known frequency.
* Cycles occurs when the data exhibit rises and falls that are not of a fixed frequency.

There are two important indicators here to consider as explained well in (Monigatti, 2022):

* ACF: Autocorrelation function. It is the correlation between the observation at current time and aa future observation (lagged version).
* PACF: Partial Autocorrelation function. It is the correlation between the observation of point A and point B of the series.

### Autocorrelation Function

A graph with blue dots and numbers

Description automatically generated

Figure 9 ACF

Autocorrelation measures the linear relationship between lagged values. In the Autocorrelation plot we have several autocorrelation coefficients plotted together.

* When data have a trend, the autocorrelations for small lags tend to be large and positive because observations nearby in time are also nearby in size. So, the ACF of trended time series tend to have positive values that slowly decrease as the lags increase.
* When data are seasonal, the autocorrelations will be larger for the seasonal lags (at multiples of the seasonal frequency) than for other lags.
* When data are both trended and seasonal, you see a combination of these effects.

Here we’re experiencing a periodic behaviour in the time series as we can see in the autocorrelation, we must understand if this is due to seasonality or if the series is cyclic. There is a well-defined seasonality as you can see in the series decomposition.

A group of blue lines

Description automatically generated

Figure 10 Series decomposition

The decomposition shows us a period behaviour with up and downs and strong repeating seasonality cycle depending on the day and month. The trend is increasing or decreasing following

### Stationarity Check.

To study the series, we must understand if the series is stationary. A time-series is stationary when its statistical proprieties have the tendency to do not change in time. For statistical proprieties we intend the distribution’s shape, mean, variance have the pattern to be similar regardless of when the observation was made. We’ve provided a small function to detect is stationary based on (Brownlee, 2020) article.

A computer screen shot of a computer code

Description automatically generated

Figure 11 Stationarity in Time Series: Our dataset.

So back to the business perspective we know that reservations have a cyclic behaviour that repeat itself. with a trend to decrease in time. The series is stationary so we can use ARIMA family to study the problem and compare with Prophet.

## Model 1. Model time series forecast with Prophet.

Prophet (Taylor & Letham, 2018) is a forecasting algorithm implemented at Facebook, based on additive regression. Additive regression is a technique that (Ian H. Witten, 2017) that combines multiple models into an ensemble to enhance overall prediction performance.

Prophet uses different methods to treat seasonality, trend, holidays and events. It is also providing an easy-to-use interface the end users not assuming that the series is stationary.

### Baseline model.

In building the baseline model we started to divide the time in test and data where 75% of the date were in the train dataset and the other %25 in the test one. After training a model we used the Prophet facilities to generate future date and looked how the forecast was.

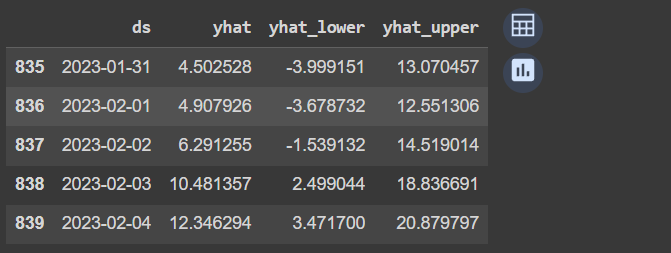


Figure 12 Prophet Forecasting Action

Figure 12 gives an idea the number of customers that Hotel08 will have at the end of February 2024 with a variation from 4 to 13.

A graph with blue and red lines

Description automatically generated

Figure 13 Testing Data

In the Figure 12 we observe the behaviour of our forecast on future dates compared to real data appears that there are meaningful differences, so we must tune better our algorithms. This has been confirmed by the graph below. The red line is the test data, and the blue one is the prediction data. The trend is quite similar, so we can tell our customers when expect more people, but the numbers are slightly different.

A graph showing a graph of a graph

Description automatically generated with medium confidence

Figure 14 Test Data Bookings vs Predicted Bookings

### Hyperparameter and cross validation tuning with Optuna.

We proceed the fine tuning of our model doing hyperparameter and cross validation tuning, and we choose to use Optuna framework (Optuna, 2024) for hyperparameters tuning. The main reason are the speed of execution and flexibility: it adapts to any machine learning model; it is written in Rust with Python binding (so it is fast) and uses a different approach from GridSearchCV. Instead of exploring randomly the parameter space and trying to find the best uses a heuristic to prune the solutions and converge faster. It works in several step:

* define an objective function
* create several trials
* create study that trigger the trials
* find the best parameters as result of that study.

Accordingly, to Prophet documentation, if we want to tune parameters, the first ones to be tunned should be:

1. **changepoint\_prior\_scale**: This is probably the most impactful parameter. It determines the flexibility of the trend and how much the trend changes at the trend changepoints.
2. **seasonality\_prior\_scale**: This parameter controls the flexibility of the seasonality.
3. **holidays**: This controls flexibility to fit holiday effects.
4. **seasonality\_mode**: Options are ['additive', 'multiplicative'].

## A screen shot of a computer program Description automatically generated

Figure 15 Objective Function for Prophet Hyperparameter Search.

The objective in tuning was to optimize the root mean square error that it is needed to compare the algorithm with SARIMAX. It also gives us a qualitative idea on the error that appears on the number of the bookings daily (Kolassa, 2020).

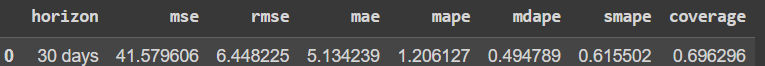


Figure 16 Performance Measures Tuned Prophet

The results after hyperparameter tuning are better. MAPE indicates the error in percentage on forecasting whereas RMSE indicate the error over time of the time series. We can expect to tell our customers to keep a margin of %1.2 in their estimate during the time and expect a variation of 6.4 bookings. In Figure 17 you can see the difference between test and train data forecasting there are errors but the trend is quite similar. We’ll use the trend to compute **DemandScore**.

A graph showing the growth of the stock market

Description automatically generated

Figure 17 Tuned Prophet vs Test Data

## Model 2: Model time series SARIMAX.

SARIMAX is the last arrived in ARIMA models family. The key idea in the ARIMA models is to compute the forecasting using a weighted sum of the lagged values (previous timestamps) and correct these forecasts adding a moving average (Geron, 2023). There are three parts:

* **Autoregressive** (AR) considers the relationship between current timestamps and previous ones.
* **Integrated** (I), involved differencing between timestamps making the series stationary.
* **Moving Average** (MA), represents the error between the current observations and the moving average.

SARIMAX differ from ARIMA in a way that consider the seasonality of the series and allows the inclusion of external factor, i.e. the holiday season, in the prediction (Msac, 2023).

### Baseline model.

A graph showing the value of a stock market

Description automatically generated with medium confidence

Figure 18 SARIMAX Bookings Test Data and Predicted Bookings in the same interval.

We’ve used the same test and train Data Frames used in Prophet and our findings are that the parameters needs to be tuned since there is an important difference between test and predictions as shown in Figure 18. In this case the RSME is around 9 room booked with 2% of error in all bookings.

A black background with white text

Description automatically generated

Figure 19 Performance Measures Base Model.

### Hyperparameter tuning in SARIMAX.

We proceed to fine-tuning the baseline model with Optuna:

* 1. Generate all possible combinations of parameters using Python **itertools.**
  2. Send those parameters to an objective function to minimize RMSE.
  3. Start the Optuna Study.

A screen shot of a computer program

Description automatically generated

Figure 20 Optuna Study to Sarimax Hyperparameters Search

## Compute Demand Score: Algorithm selection.

Due to our results and developer experience, we’ve decided to select Prophet for computing demand score. The reason is that despite on some occasions SARIMAX performs better, Prophet fits most of the cases.

For computing the demand score we apply the forecast and then see the trend and from the trend we get the mean we rescale to 100. With this we have defined a process for each hotel series, the algorithm is scattered in two phases: Data Cleaning, Training and Computing the score, basically the following steps. \*So for each we need to flow the below diagram flow for each hotel.

A screenshot of a graph

Description automatically generated

Figure 21 Demand Score Diagram Flow

# Cancellation Score: Our objectives.

Our custome*r Blastness* (Blasteness.com, 2023) holds all booking inside a Property Management System for its clients. Each booking has three states (classes): *Cancelled, Confirmed, Modified*. Our objective is to find an algorithm to compute the cancellation score, based on cancellation forecasting. Most of the recent studies treated as a classification problem (Nuno , et al., 2019). The cancellation factors can depend on hotel, customers, booking and external factors. Our goal is to predict the next booking trend for our customers and count the number of cancellations in a future temporal period and compute a score based on the number of cancellations forecasted. As first step we want to review again the dataset in this classification context, encode categorical data and see results.

Explora

## Feature selection

## Univariate Analysis

## Anomaly Detection

## Baseline Model

## PCA and Random Forest Model

## 

## Cross Validation

## PCA and SVM

## Cross Validation

## Compute the demand score: algorithm selection.

A diagram of a hotel rank cancellation score

Description automatically generated

# Review Score: Taking in account social reputation for Hotels.

## Data Preparation

## TripAdvisor Scraper Design

## SparkNLP and BERT.

As cited in a medium blog () *Spark NLP comes with 17,800+ pretrained pipelines and models in more than 250+ languages. It supports most of the NLP tasks and provides modules that can be used seamlessly in a cluster* .

A screen shot of a computer

Description automatically generated

## 

## ReviewScore Computation.

# Data Engineering Architecture.

# Model Deployment Design.

# Accomplishment and future work.

# Ethical Considerations.

Hotel revenue management can have a positive influence on the society. A socially responsible hotel management could use this study promote a fair or a festival in the moment of maximum demand, creating a nice environment for locals and tourists or reduce their environmental impact when the demand is at its minimum (i.e. waste less water or electricity, less trash to recycle and so on).

The data on this study has been anonymized to remove all hotel and customer confidential data. Any data and study taken in consideration for learning the field will be cited in the report under Harvard guidelines and avoid plagiarism. All the model developed can be used and published online, also Blastness(Blasteness.com, 2023) will have the right to cite this work as joint effort.

The repository data has been release with Creative Common License at <https://github.com/CCT-Dublin/capstone-project-feb-2024-pt-giorgiozoppi> where you can find this report as well.

# References

Alliance, S., 2017. *What is Scrumban?.* [Online]   
Available at: https://www.agilealliance.org/scrumban  
[Accessed 03 03 2024].

Apostolos, A., 2021. Modeling and Forecasting Daily Hotel Demand: A Comparison Based on SARIMAX, Neural Networks, and GARCH Models. *Forecasting*, 21 August, p. 16.

Bergia, S., 2021. *Unsupervised Anomaly Detection on Server Metircs,* Torino: Polito.

Blasteness.com, 2023. *Blastness - Growing your business.* [Online]   
Available at: https://www.blastness.com/en/index  
[Accessed 26 03 2024].

Brownlee, J., 2020. *How to check if a time series is stationary in Python?.* [Online]   
Available at: https://machinelearningmastery.com/time-series-data-stationary-python/  
[Accessed 10 10 2024].

Daniel, S., Haller, V. & Bellone, B., 2023. *Seasonal Trend And Holiday Decomposition with Loess: A Real-Time Approach to Analyzing High-Frequency Alternative Data,* London: Quant Tecnology.

Diana-Jens, P. & Rodríguez Ruibal, A., 2015. La reputación online y su impacto en la política de precios de los hoteles. *Cuaderno de Turismo*, 15 07, p. 129–155.

Hansler, M. & Piekenbrock, M., 2019. dbscan: Fast Density-Based Clustering with R. *Journal of Statistical Software,* 91(1), pp. 1-30.

Ivanov, S., 2014. *Hotel revenue management: From theory to practice.* 1 ed. Varna, Bulgaria: Zangador Ltd..

McKinney, W., 2024. *Panda data library.* [Online]   
Available at: https://pandas.pydata.org/  
[Accessed 27 10 2024].

Monigatti, L., 2022. *Interpreting ACF and PACF Plots for Time Series Forecasting.* [Online]   
Available at: https://towardsdatascience.com/interpreting-acf-and-pacf-plots-for-time-series-forecasting-af0d6db4061c  
[Accessed 27 10 2024].

Nuno , A., De Almeida , A. & Nunes, L., 2019. Hotel booking demand datasets. *Data in Brief,* 22(February), pp. 41-49.

Nuno Antonio, A. D. A. L. N., 2019. Big data in hotel revenue management: Exploring cancellation drivers to gain insights into booking cancellation behavior. *Cornell Hospitality Quarterly,* 60(4), pp. 298-319.

Optuna, 2024. *Optiuna: An open source hyperparameter optimization framework to automate hyperparameter search.* [Online]   
Available at: https://optuna.org/  
[Accessed 01 10 2024].

Schmitd, S., Wenig, P. & Paperbrock, T., 2020. Anomaly Detection in Time Series: A Comprehensive Evaluation. *Proceedings of the VLDB Endowment, ,* 15(9).

Schubert , E. et al., 2017. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM Transactions on Database Systems,* 42(2), pp. 1-21.

Taylor, S. J. & Letham, B., 2018. Forecasting at Scale. *The American Statistician,* 72(1), pp. 37-45.

Teaml, S., 2024. *Statsmodel.* [Online]   
Available at: https://www.statsmodels.org/stable/index.html  
[Accessed 28 10 2024].

Wang, Z., Lin, X. & Li, H., 2023. Impact of reputation on hospitality profitability: impact of service failure online exposure on revenue performance – evidence from the hotel industry in China. *Tourism Review*, 17 3, pp. 1387-1413.

Wikipedia, 2024. *SWOT Analysus.* [Online]   
Available at: https://en.wikipedia.org/wiki/SWOT\_analysis  
[Accessed 27 10 2024].