A logo for college computing

Description automatically generated

# TrackML/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: 10****th November 2024*** |  |
| *Date of Submission 10****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

[TrackML/HotelRank: Elevating Revenue Performance Through Machine Learning and Deep Learning Techniques. 1](#_Toc182136065)

[*Module Name:* ***Strategic Thinking (HDip in Data Analytics - Feb 2024 - HCI cohort)*** 1](#_Toc182136066)

[Introduction 4](#_Toc182136067)

[Understanding hotel room demand over time. 4](#_Toc182136068)

[Prediction of booking cancellations. 4](#_Toc182136069)

[Online hotel reputation. 4](#_Toc182136070)

[Problem Domain and Objectives 4](#_Toc182136071)

[Demand forecasting. 4](#_Toc182136072)

[Booking Cancellations. 5](#_Toc182136073)

[Online Reputation. 5](#_Toc182136074)

[HotelRank Score. 5](#_Toc182136075)

[Project Scope 5](#_Toc182136076)

[Data Sources. 6](#_Toc182136077)

[Business Scenario of Hotel Rank. 7](#_Toc182136078)

[Success Criteria. 7](#_Toc182136079)

[Data Cleaning and descriptive statistics. 9](#_Toc182136080)

[Descriptive statistics. 9](#_Toc182136081)

[How is frequent is a booking? 10](#_Toc182136082)

[Which between our customer client how had most revenue? 12](#_Toc182136083)

[Demand Score Computation: Forecasting the Booking. 12](#_Toc182136084)

[Data Preparation 12](#_Toc182136085)

[Outlier and Anomaly Detection. 13](#_Toc182136086)

[What is an anomaly in a time series? 13](#_Toc182136087)

[Time Series Descriptive Statistics 16](#_Toc182136088)

[Stationarity Detection. 16](#_Toc182136089)

[Autocorrelation Function and Serie Decomposition. 18](#_Toc182136090)

[Model 1. Model time series forecast with Prophet. 20](#_Toc182136091)

[Baseline model. 20](#_Toc182136092)

[Hyperparameter and cross validation tuning with Optuna. 20](#_Toc182136093)

[Model 2: Model time series SARIMAX. 23](#_Toc182136094)

[Baseline model. 23](#_Toc182136095)

[Hyperparameter tuning in SARIMAX. 24](#_Toc182136096)

[Compute Demand Score: Algorithm selection. 25](#_Toc182136097)

[Cancellation Score: Our objectives. 27](#_Toc182136098)

[Exploratory Data Analysis. 27](#_Toc182136099)

[Encoding Categorical Variables. 31](#_Toc182136100)

[Anomaly Detection and Dimensionality Reduction. 32](#_Toc182136101)

[PCA: Dimensionality Reduction 32](#_Toc182136102)

[SMOTE: Oversampling to rebalance classes 33](#_Toc182136103)

[Correlation between features. 33](#_Toc182136104)

[Process of Model Selection. 36](#_Toc182136105)

[Model 3: Random Forest with PCA. 36](#_Toc182136106)

[Baseline Model 38](#_Toc182136107)

[Model 4: Support Vector Classifier. 40](#_Toc182136108)

[Baseline Model 40](#_Toc182136109)

[Results 41](#_Toc182136110)

[Model 5: Neural Network Classifier. 41](#_Toc182136111)

[Baseline Model 41](#_Toc182136112)

[Neural Network Tuned Model Results 41](#_Toc182136113)

[Computing Cancellation Score: algorithm and model selection. 42](#_Toc182136114)

[Review Score: Online reputation for hotels. 43](#_Toc182136115)

[Data Preparation 43](#_Toc182136116)

[Source and Dataset creation. 43](#_Toc182136117)

[Review Score Computation. 45](#_Toc182136118)

[Hotel Rank Computation. 46](#_Toc182136119)

[Conclusion and future work. 47](#_Toc182136120)

[Ethical Considerations. 47](#_Toc182136121)

[Blastness Data Consent. 48](#_Toc182136122)

[References 48](#_Toc182136123)

# 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 (Blasteness.com, 2023) 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 as main guid wil (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/DL techniques are used. | **20th April 2024** |
| **Data Collection** | The dataset consists of the bookings of ten Italian luxury hotels in a two-year period. Data Collection from TripAdvisor. | **21st May 2024** |
| **Data Exploration.** | 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** |
| **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 and cross-validation. | **1st September 2024** |
| **Hyperparameters tuning.** | Tune ML and DL 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 Hotel Rank.** | Select the models and weight for Hotel Rank 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. | **4th 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 ten luxury hotels, during a period of two years at least.

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

We had to refactor a TripAdvisor Web Scraper in Go (Authors, 2024) 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.

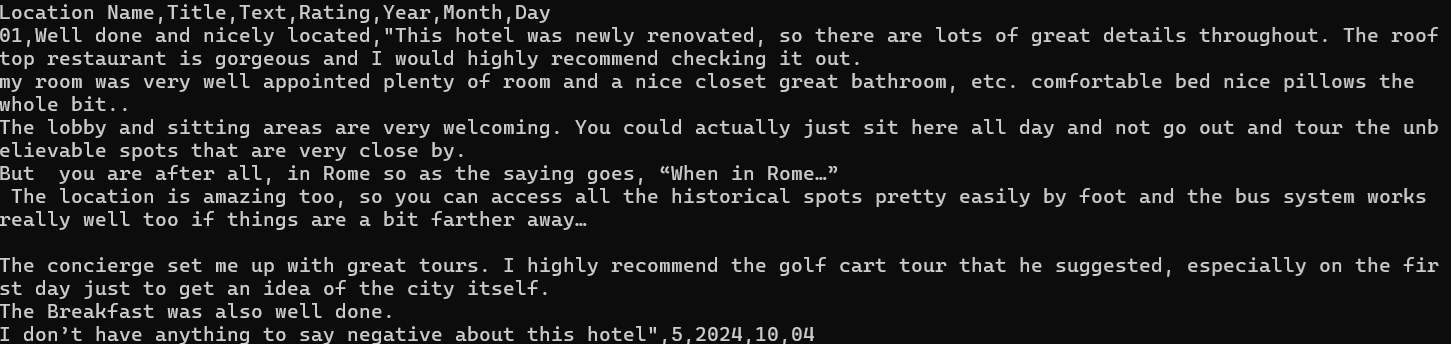


Figure Hotel Review Format: Location (HotelID), Title ,Text ,Rating, Year, Month ,Day

# 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 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.

During this study and the future deploy in production we use our strengths to reach the appealing opportunities presented 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 Hotel Rank. Due to time and scope restrictions, we were 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 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, so proceeded to drop it.

## 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:

* We've nearly forty thousand booking in the period between 2019 and 2023.
* The average staying is 1.6 days for each booking.
* The most used booking channel is Booking.com.
* We had to drop BookingDevice feature due errors.

A screenshot of a computer screen

Description automatically generated

Figure Descriptive Statistics.

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 due to this distribution.

A graph of blue bars with red circle

Description automatically generated

Figure Arrival Distribution

A graph with a red line

Description automatically generated

Figure 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.

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

A graph of a bar graph

Description automatically generated

Figure Blastness customer per revenue.

There is a strong difference between all hotels, since the type of hotels, we’ve one customer that has more revenues than others. Findings that are evident:

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

- Most booked customer is Hotel008 but Hotel009 has bigger room per price.

After this exploration we focused on **Demand Score** data analysis, we treated 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 is **BookingCount**, that’s the number of customers arrived in the Hotel daily. The goal is to design a specific process, keeping mind the most booked client of our customer (**Hotel 008**) that can we generalize later. So, the subproblem is a time series analysis forecasting for Hotel008.

Once we’ve solved this challenge is easy to generalize the methodology to the other hotels.

A screenshot of a graph

Description automatically generated

Figure Density of BookingCount per Hotel

In above figure we can notice 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. To better SARIMAX (Fahad & Denes, 1994) evaluation 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. 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.

During our research, we identified an effective algorithm and applied to our candidate hotel data, and it is essentially based on the work on *Unsupervised Anomaly Detection on Server Metrics* (Bergia, 2021).

The diagram below describes the algorithmic steps.

A flowchart of a forest

Description automatically generated

Figure Time Series Anomaly Removal

Let’s give an explanation:

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 complexity.
5. We select the data points that are far from the cluster and delete them.

A screenshot of a computer

Description automatically generated

Figure Anomalies detected from Isolation Forest

Given the nature of the data and the specific requirements of this problem, we’ve excluded holiday periods from our anomaly detection algorithm. Figure 10 and Figure 11 show us the results.

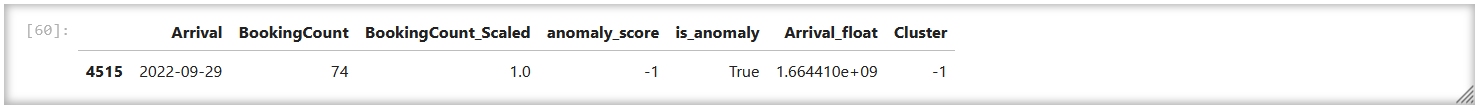


Figure Remaining Anomalies to drop.

## Time Series Descriptive Statistics

A graph showing a number of data

Description automatically generated with medium confidence

Figure Descriptive Statistics

In the picture we can see the difference between the Booking Count, its 12 months mean, and 12 months standard deviation. The data is irregular with a strong variance and possible cycles and 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.

### Stationarity Detection.

To understand better time 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. We’ve provided a small function to detect is stationary based on (Brownlee, 2020) article.

A screenshot of a computer code

Description automatically generated

Figure Stationarity in Time Series: Our dataset.

The series is not stationary, so we had to transform through a time series differencing algorithm for using ARIMA family to study the problem and compare with Prophet.

In the Prophet analysis we decided to use the series as it is without applying the differencing since Prophet supports non-stationary series with a well-defined seasonality.

### Autocorrelation Function and Serie Decomposition.

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.

A graph with blue dots and numbers

Description automatically generated

Figure Autocorrelation factor.

Autocorrelation measures the linear relationship between lagged values, and it is useful to detect when seasonality, random walk and trend are present and how they influence the series. 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 we are presence of random walk.

A graph of blue lines

Description automatically generated with medium confidence

Figure Series decomposition

The decomposition shows us a period behaviour with up and downs and strong repeating seasonality cycle depending on the day and month.

## 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 the forecast.

A graph with blue and red lines

Description automatically generated

Figure Prophet in action

In the Figure 16 we observe the behaviour of our forecast on future dates compared to real data appears that there are meaningful differences, so we had to tune better our algorithm. The red line is the test data, and the blue one is the prediction data. The trend is quite similar but the quality of forecasting is not good.

### 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 reasons are the speed of execution and flexibility: instead of exploring randomly the parameters space and trying to find the best, it uses a heuristic to prune the solutions and converge faster. In the diagram the algorithm we used.

A diagram of a research process

Description automatically generated

Figure How Optuna works.

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. When the hotel is close.
4. **seasonality\_mode**: Options are ['additive', 'multiplicative'].

A screenshot of a computer program

Description automatically generated

Figure Objective Function for Prophet Hyperparameter Search.

The objective, we accomplished in hyper parameter search, was to optimize the root mean square error (RMSE) and cross validate while tuning. RMSE gives us a qualitative idea on the error that appears on the number of the bookings daily (Kolassa, 2020).

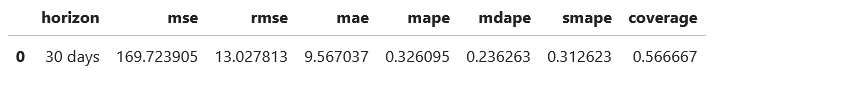
The results after hyperparameter tuning are better. In Figure 17 you can see the difference between test and train data forecasting there are errors, but the trend is quite similar. We’ll use the trend to compute **DemandScore**.

A graph showing the loss of a stock market

Description automatically generated

Figure Tuned Prophet vs Test Data

IMAPE and RMSE are better now, 30% error and a max 14 rooms, however this might be too high for our customers.



## 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 a number of data

Description automatically generated with medium confidence

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

SARIMAX on the differencing series works far better than Prophet. Looks like our series for Hotel008 has a moving average, so we select SARIMAX has an error of half of the rooms.



### Hyperparameter tuning in SARIMAX.

We proceed to fine-tuning the baseline model with Optuna as shown in the diagram below.

A diagram of a program

Description automatically generated

Figure HPS for SARIMAX.

We present the algorithm for the objective function in the Figure 22

A screenshot of a computer code

Description automatically generated

Figure Optuna Study to SARIMAX Hyperparameters Search

Finally, we’ve for the times series a quite satisfying result where the RMSE is lower than Prophet ones.

A screenshot of a computer error

Description automatically generated

Figure Performance Measures in Cross validated Model

The comparison says that for this series the best choice is SARIMAX.

|  |  |
| --- | --- |
| SARIMAX Forecasting Performance Measures | Prophet Performance Measures |
| **RMSE:** 6.4 | **RMSE:** 13.03 |

A graph showing the amount of data

Description automatically generated with medium confidence

Figure SARIMAX tuned and cross validated.

In this case the choice is obligatory for SARIMAX, but we dare to choose both for each hotel and see which performs better in any single case. We’ve defined an algorithm for computing demand score as result of this analysis.

## Compute Demand Score: Algorithm selection.

Due to our results and developer experience, we’ve decided to select both and looking for the minimum RMSE, so for computing demand score we’ve designed the algorithm below.

A diagram of a process

Description automatically generated

Figure Demand Score Algorithm.

# Cancellation Score: Our objectives.

Our custome*r Blastness* (Blasteness.com, 2023) holds all bookings inside its Central Reservation System and Revenue Management System for its clients. We know from studies (Nuno Antonio, 2019) that cancellations affect hotel revenue.

Most of the recent studies treated them as a classification problem (Nuno , et al., 2019). The cancellation factors can depend on hotel, customers, booking and external factors.

Our goal was 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 wanted to review again the dataset in this classification context, encode categorical data and see results.

### Exploratory Data Analysis.

The current dataset in the partially cleaned form has:

* 11 Feature, divided numerical and categorical variables, as we can see in Figure 23:
  + 4 Categorical (BookingChannel, HotelId, City, Season)
  + 7 Numerical (Status, Nights, Total, Arrival\_Timestamp, Departure\_Timestamp, Purchase\_Timestamp, City)
  + Most of the booking are confirmed.

The first actions we’ve done were:

* Set the dependent variable as Status, since as cancellation score, we want to know the total confirmed booking and cancellations.
* Separate numerical and categorical features to study them independently.

A screenshot of a computer

Description automatically generated

Figure Numerical and categorical features

In this analysis we’ve started to discover the nature of the independent variable that models the booking state (**Status**) and its statistic behaviour, then the relationship between the variable dependent and independent variables and their distribution.

A graph of a state

Description automatically generated

Figure Distribution of Target Variable

From figure above that each state has a binomial distribution where each state: 0 means booking cancelled, 1 means booking confirmed, **2** means booking modified.

Most of our client customers tend to confirm the booking and just a minor part modify the booking. The number of cancellations is also high between the guests of our customer clients. Each customer has a fixed probability to book or not, so the revenue managers could increase this probability providing discounts when this fixed probability is lower than expected. From these considerations we derived the dataset is unbalanced, as expected. We decided to do not use oversampling techniques such as SMOTE for rebalance the dataset but just doing any training with cross validation using the Stratified K-Fold method.

SA graph of a distribution of state

Description automatically generated with medium confidence

Figure Dataset unbalanced.

#### Which relationship has Booking Status with Customer Arrival?

A graph of a flight arrival

Description automatically generated with medium confidence

Figure Relationship between Status and Customer Arrival

As expected, most of our customers if the arrive their booking is confirmed.

#### Which relationship has Booking Status with City?

A graph of a number of bars

Description automatically generated with medium confidence

The city with highest number of bookings is Rome while our client custumer in Olbia experiences the highest rate of cancellations. Guests heading to the Mont Blanc tends to not cancel whereas touris travelling to Naples don’t cancel their bookings.

#### Which relationship has Booking Status with Night reserved?

A graph with different colored bars

Description automatically generated

Guests who cancel don’t end up staying at our client hotels, but it is interesting to note that most of our guests are one night stayers. Additionally, the modified status appears to be irrelevan

#### Which relationship has Booking Status with Season?

We’ve previously encoded the categorical variable season with 4 integer *(0=Winter, 1=Summer, 2=Spring, 3=Autumn)*. Most of Italians go on vacation in summer and a interesting portion of them cancels or modify their plans in Summer.

A graph of different colored squares

Description automatically generated

Figure Booking Status per Season

The features Total (total revenue), CancRate, and PurchaseData/Tmestap we’ve found that follow the same pattern. After this initial analysis we’ve discovered the distribution of numerical features.

#### Distibution of Numerical Fatures.

A group of blue and white graphs

Description automatically generated

As we’ve seen at the beginning DepartureTimeStamp, ArrivalTimestamp, PurchaseTimestamp follow the same pattern. The cancellation between client is very low, the max percentage of cancellation rate is lower than 5 percent and people don’t stay in same Hotel more than ten day at max.

## Encoding Categorical Variables.

After this explorative phase and while investigating the nature of the features we’ve proceeded to encoding categorical variables using two approaches:

* Dummy encoding for City (**get\_dummies).**
* Ordinal Encoding or Python direct integer conversion for all the other ones.

In the model selection, our algorithms candidates were RandomForest and Support Vector Machine (SVM) and a Dense Neural Network.

## Anomaly Detection and Dimensionality Reduction.

We faced a classification problem. Given a list of 10 features, we wanted to predict whether a reservation has been booked, cancelled or modified so some steps are interesting before modelling:

* Data cleaning and outliers’ removal.
* Reduce the number of features (dimensionality reduction)
* Decision about facing unbalance classes.

#### Anomaly detection: Outliers.

Outliers are data points that significantly deviate from most of the data. They can be caused by errors, anomalies, or simply rare events. Outliers may lead to overfitting if the model tries to fit them perfectly, thus capturing their noise rather than the actual patterns in the data. We decided to drop outliners, and we noticed that it doesn’t lead to underfitting. For detecting outliners in a classification problem, we use Isolation Forest.

##### Isolation Forest

The algorithm builds a random forest (Geron, 2023) in which each decision tree is grown randomly: at each node, it picks a feature randomly, it picks a random threshold value to split the dataset in two. This process is repeated until all instances are isolated. It assumes that since anomalies are isolated can be picked up quickly through the process of splitting.

We’ve identified around three thousand anomalies, in percentage 9% of the dataset. We have marked and dropped outliners records.

Immagine che contiene schermata

Descrizione generata automaticamente

Figure Outliers distribution.

### PCA: Dimensionality Reduction

The encoding of categorical variables brought the number of features from 11 to 16, we wanted to see if there was an opportunity for dimensionality reduction. We did an initial attempt through univariate feature selection method, but it was not simple and flexible enough lie Principal Component Analysis.

Principal Component Analysis (Geron, 2023) is method that projects the data into in a hyperplane trying to preserve the statistical properties of the dataset and as result of this project we can select the most relevant dimensions/components of the training dataset.

In our case we’re able to reduce the features from 16 to 10. To make this process more robust we decided to delay his application and pack it in **Sklearn** (sklearn, 2024) pipeline during model selection and tuning as you can see in Figure 30.

A screenshot of a computer

Description automatically generated

Figure Pipeline that includes a PCA and Classifier modules

### SMOTE: Oversampling to rebalance classes

In our case we’ve the dataset unbalanced this might lead to accuracy problems. Classification algorithms, such as SVM, tend to be biased toward the most common classes when the dataset is imbalanced. One technique to overcome this problem is oversampling.

**SMOTE** or the Synthetic Minority Over-sampling Technique is an oversampling method that increases the size of the training set by generating many realistic variants of each training instance. In this context, it is used to generate more samples of the less frequent classes.

### Correlation between features.

To investigate the relationship between features we’ve used Pearsons’ correlation coefficient (standard correlation) and Predictive Power Score (PPS), it differs from standard correlation since it can detect both linear and nonlinear patterns. His semantic is 0 no predictive power and 1 full predictive power.

A chart with different colored squares

Description automatically generated with medium confidence

Figure Pearson Correlation Matrix

A chart with red and blue squares

Description automatically generated

Figure PPS Matrix

We have detected as you can see in Figure 31 and Figure 32 that some features are strictly correlated with the Status (BookingStatus). For example, the number of nights and the revenue (Total). Other features are correlated between them, for example **Arrival\_Timestamp, Purchase\_Timestamp and Departure\_Timestamp.**

## Process of Model Selection.

A diagram of a process

Description automatically generated

Figure Process of model selection

In the above figure there is the self-explanatory algorithm that we’ve followed in data cleaning and model selection.

## Model 3: Random Forest with PCA.

Random Forest is an ensemble learning method made up of multiple machine learning algorithms and based on Decision Trees. We present the algorithm for completeness in the diagram in Figure 32.

A flowchart of a tree

Description automatically generated

Figure Random Forest Algorithm

Our objective was to tune and cross validate a random forest classifier that can predict whether a reservation is cancelled, confirmed or modified. We considered modified bookings but at the end of the game we’ve observed that the percentages of reserved bookings are minimal.

### Baseline Model

A close-up of a computer code

Description automatically generated

Figure Random Forest Baseline Classifier

We followed an approach based made up of the following steps:

* Split add the train in 25% Test, 75%
* Feature scaling through Standard Scalar since it is a requirement for PCA.
* Apply Principal Component Analysis and obtain a reduced feature train and test datasets.
* Train the model with a **RandomForestClassifier**.
* Cross validates the model using **cross\_val\_score** with Stratified-K-Fold (Szilvia & Attila, 2024).

For our classification algorithms, we used stratified cross-validation because it effectively handles imbalanced data and is well-suited for classification tasks. This method first sorts of the samples by class and groups them accordingly. Then, for each class, it divides the grouped samples into *K* non-overlapping sets, or "strata." Each “strata” is created to be the same size, ensuring balanced representation across classes. After setting up the strata, we build the folds by selecting one item from each “strata”, which prevents any fold from being imbalanced during cross-validation.

Below we show the results we obtained in our dataset with a tuned and cross validated Random Forest.

A screenshot of a graph

Description automatically generated

Figure Tuner and Cross Validated Random Forest Performance

A chart with numbers and labels

Description automatically generated

Figure Confusion Matrix

## Model 4: Support Vector Classifier.

Support Vector Classifier is classification method based on a minimization problem, each feature in the training set gets represented by a vector that is mapped into a higher (maybe infinite) dimensional space by a kernel function. SVM finds the best separating hyperplane with the maximal margin in this higher dimensional space, that separates the training data in classes. There are different types of kernels in our case we’ve followed the guidelines placed in *A Practical Guide to Support Vector Classificatio*n (Lin, 2023). The algorithm we’ve followed is in Figure.

A diagram of a program

Description automatically generated

Figure Tuning SVM

### Baseline Model

A screenshot of a computer

Description automatically generated

Figure PCA and SVN combined in scitekit-learn pipeline

A computer code with black text

Description automatically generated

Using the suggested algorithm we have proceeded to define a baseline and then do hyperparameter search. The result is good but let’s try to do better following the tuning advice (Lin, 2023).

### Results

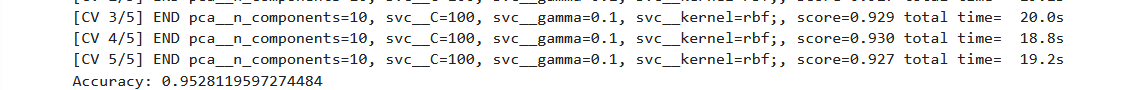


Figure SVC RBF Kernel

We were able to tune well the algorithm.

## Model 5: Neural Network Classifier.

Now we want to explore how in this case we can work with a simple NN.

Baseline Model.

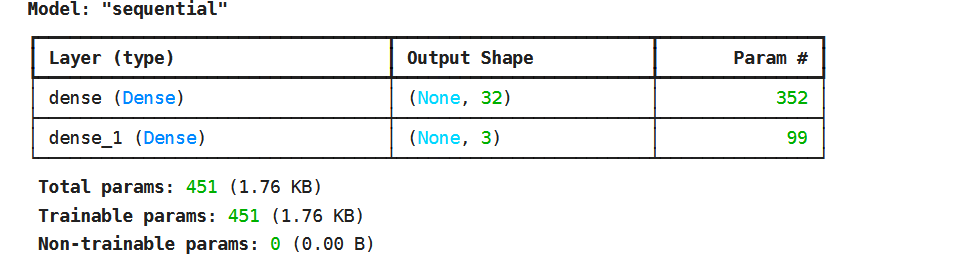


Figure Structure of Neural Network

The baseline neural network model is a sequential model with ten inputs and two layers of 32 and 3 neurons. The accuracy in our dataset of a trained baseline model is around 0.79, lower than other algorithms. In compiling the model with Keras we used sparse **categorical crossentropy** loss because we have sparse labels and the classes are exclusive.

### Neural Network Tuned Model Results

We have used Kera Tuner (O'Malley, 2019) for hyperparameters tuning. During training, the samples are again split into 2 internal subsets. One which is used for actual training and other which is used for validation after each epoch. The ratio of split can be controlled by the parameter 'validation\_split', in our case 20% of the training set. The accuracy is:



Figure Accuracy with KT tuner cross validated

## Computing Cancellation Score: algorithm and model selection.

A blue squares with white text

Description automatically generated

Figure Algorithm Selection

As part of the model selection process, we’ve selected RandomForest since it is faster to tune despite a 0.2 per cent less of performance.

So, using data trained and testing data we can proceed with the algorithm of cancellation score as shown in the diagram below.

A diagram of a hotel rank cancellation score

Description automatically generated

Figure Cancellation Score Algorithm

# Review Score: Online reputation for hotels.

## Data Preparation

Our goal has been to select the top 10 luxury hotel indicated by Blastness and download all TripAdvisor reviews to score them later because in Hotel Revenue Management, online reputation has a significant impact on present and future revenues: attracts new customers, add authority to your brand and boost your search engine ranking.

## Source and Dataset creation.

A screen shot of a computer

Description automatically generated

Figure 47 Example TripAdvisor Scraped Review.

The first challenge that faced Is the lack of a valid dataset review dataset that includes Blastness hotels, so we used an opensource TripAdvisor scraper, that helped use to generate the dataset.

The scraping application does a GraphQL query to TripAdvisor HTTPS endpoint, then it parses the results and store them in a CSV, one for each hotel. We had to extend and fix the application for our purposes since it was sporadically crashing.

A diagram of a company

Description automatically generated

Figure 48 TripAdvisor Go Scraper

The scraped resulted dataset is about 35MB of hotel raw reviews data scattered around 10 small files, this mean the file size on average of 3MB, but the direct inspection reveals that the variance is high. From this file to create a process to be repeated, we selected the reviews of Hotel008. We’ve also performed an exploratory data analysis of the reviews for our selected hotel. Out client performed quite well during the years.

A graph of a graph of different colored bars

Description automatically generated with medium confidence

Figure Hotel08 Reviews per rating

## Review Score Computation.

A diagram of text and words

Description automatically generated with medium confidence

Figure Review score pipeline.

After having all the reviews on disk and the initial exploratory analysis we had the problem on how to do sentiment analysis to detect the number of positive and negative reviews.

We wanted do perform sentiment analysis because performance in terms of revenue depends on online reputation (Diana-Jens & Rodríguez Ruibal, 2015), so our goal in designing **Hotel Rank** algorithm is to detect bad hotel reputation.

State-of-the-art sentiment analysis is achieved by fine-tuning pretrained BERT (Talaat, 2023) models on sentiment datasets.

A diagram of a diagram

Description automatically generated

Figure Overall training in BERT from Bert paper (Delvin, et al., 2019)

We collected reviews using a scraper coded in Go and created a dataset, cleaned the data. Once the dataset is created, we were in a position use BERT and doing sentiment analysis using Apache Spark NLP, following the algorithm in the diagram below. We selected the number of positive and negative review, computed the percentage and added to the average rating previously computed.

A diagram of a computer program

Description automatically generated

Figure Review Score process.

# Hotel Rank Computation.

Once we’ve defined cancellation score, demand score, review score we can compute Hotel Rank algorithm following the diagram.

A diagram of a hotel

Description automatically generated

Figure Hotel Rank Algorithm

# Conclusion and future work.

In this work we’ve successfully provided a score algorithm for revenue managers in doing that we’ve:

1. Defined a process to forecast customer arrival and room demand for our customers and a score algorithm that models the guest arrival.
2. Created a cancellation prediction algorithm based on three years dataset of booking from ten Italian luxury hotels to provide a score that minimize overbooking and under booking in that hotel segment. Our customer’s hotel managers have now a tool to measure the future cancellations.
3. Defined a process to fetch online review and use BERT Large Language models to provide a way to estimate the Hotel reputation.

We’ve also experimented with big data storage for retrieving reviews and Apache Spark for data processing, but more work must be accomplished to make Hotel Rank a product to be sold to luxury hotels. We plan to continue this work and receive funding for exploring more deep learning opportunities such as forecasting with long short-term memory networks and creating a world class data engineering infrastructure for training.

# 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.

# Blastness Data Consent.

Immagine che contiene testo, algebra, schermata

Descrizione generata automaticamente

Figure Blastness Data Consent

# 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.

Authors, G., 2024. *Go Programming Language.* [Online]   
Available at: https://go.dev  
[Accessed 10 10 2024].

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.

Delvin, J., Ming-Wei, C., Kenton, L. & Kristina, T., 2019. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,* s.l.: Cornell University.

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.

Fahad, A. R. & Denes, C., 1994. A Seasonal Autoregressive Integrated Moving Average with Exogenous Factors (SARIMAX) Forecasting Model-Based Time Series Approach. *Inventions,* 7(4), pp. 40-1.

Geron, A., 2023. *Hands-On Machine Learning with Scikit-Learn, Keras and Tensorflow.* 3rd ed. Sebastopols: O'Reilly.

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

Ian H. Witten, E. F. M. A. H. C. J. P., 2017. *Data Mining: Practical Machine Learning Tools and Techniques.* 4 ed. New York: Morgan Kaufman.

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

Kolassa, S., 2020. Why the “best” point forecast depends on the error or accuracy measure. *International Journal of Forecasting,* 1(36), pp. 208-2011.

Lin, C., 2023. *A Pratical Guide to Support Vector Classification,* Taipei: National Taiwan Univesity.

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].

Msac, S., 2023. *ARIMA vs SARIMA vs SARIMAX.* [Online]   
Available at: https://medium.com/@sophiamsac/arima-vs-sarima-vs-sarimax-03dd04fc7c66  
[Accessed 29 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.

O'Malley, T. a. B. E. a. L. J. a. C. F. a. J. H. a. I. L. a. o., 2019. *Keras, Tuner.* [Online]   
Available at: https://github.com/keras-team/keras-tuner  
[Accessed 01 11 2024].

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.

sklearn, 2024. *Sklearn: Machine Learning in Python.* [Online]   
Available at: https://scikit-learn.org/stable/  
[Accessed 30 10 2024].

Szilvia, S. & Attila, F., 2024. A Comparative Study of the Use of Stratified Cross-Validation and Distribution-Balanced Stratified Cross-Validation in Imbalanced Learning. *Sensors,* 23(4), pp. 13-31.

Talaat, A., 2023. Sentiment analysis classification system using hybrid BERT models. *Journal of Big Data,* 110(4), p. 11.

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].