A logo for a university

Description automatically generated

**ASSIGNMENT COVER SHEET**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PROGRAMME | : | Master of Business Analytics | | |
| SUBJECT CODE AND TITLE | : | BAA5053 Machine Learning for Business Decisions | | |
| ASSIGNMENT TITLE | : | Predicting Hotel Booking Cancellations to Optimize Revenue Management | | |
|  |  |  | | |
| LECTURER | : | Dr. Aaron Aw Teik Hong | ASSIGNMENT DUE DATE: | 13/12/2024 |

STUDENT’S DECLARATION

1. I hereby declare that this assignment is based on my own work except where acknowledgement of sources is made.
2. I also declare that this work has not been previously submitted or concurrently submitted for any other courses in Sunway University/College or other institutions.

[ Submit “Turn-it-in” report (please tick √): Yes \_\_\_\_\_ No \_\_√\_\_\_ ]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | NAME | STUDENT ID NO. | SIGNATURE | DATE |
| 1. | Abdul Hakim Bin Kamalur Rahman | 24015257 | Abdul Hakim | 13/12/2024 |
| 2. | Annabel Ching Ke Xin | 24002685 | Annabel | 13/12/2024 |
| 3. | Chia Poh Yang | 23138555 | Chia | 13/12/2024 |
| 4. | Divani A/P Arumugam | 19058908 | Divani | 13/12/2024 |
| 5. | Harresh A/L Ragunathan | 19076090 | Harresh | 13/12/2024 |
| 6. |  |  |  |  |

E-mail Address / Addresses (according to the order of names above):

|  |  |
| --- | --- |
| 1. 24015257 @imail.sunway.edu.my | 4. 19058908@imail.sunway.edu.my |
| 2. 24002685@imail.sunway.edu.my | 5. 19076090 @imail.sunway.edu.my |
| 3. 23138555@imail.sunway.edu.my | 6. |

APPROVAL FOR LATE SUBMISSION OF ASSIGNMENT (If applicable)

IF extension is granted, what is the revised due date? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature of Lecturer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| Marker’s Comments: |

Marks and / or Grade Awarded: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ADDENDUM**

**USE OF ARTIFICAL INTELLIGENCE (A.I.) DECLARATION**

Students are allowed to use AI to support completion of assessments. However, students are reminded to do so ethically and transparently. This is so that (a) submissions can be fairly and accurately marked; and (b) feedback can be provided on the content that reflects student ability, in order to help with future submissions. Students are also reminded that in accordance with the University’s Academic Malpractice Policy, Item 4.11.2, “*… the representation of work: written, visual, practical or otherwise, of any other person, including another student or* ***anonymous web-based material*** *[emphasis added], or any institution, as the candidate’s own*” is considered malpractice.

**Declaration**

[ ] I / We used the following A.I. tools to produce content in this submission:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tool** | **Purpose** | **Prompts** | **Sections where AI output was used / Outcome(s) in the submission** |
| *ChatGPT* | *Generating coding reference* | *“How to manage with missing value in Python?”*  *“Suggest some visualization suitable for Exploratory Data Analysis”* | *The coding part in Python. All the answer provided by ChatGPT are for references.* |
| *Grammarly* | *Correcting grammar and spelling, improving sentence structure* | *N/A* | *Grammarly suggestions were used for all sections of the essay* |
|  |  |  |  |

*Note: Add additional rows if necessary.*

**OR**

[ ] I / We did not use any A.I. tools to produce any of the content in this submission.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | NAME | STUDENT ID NO. | SIGNATURE | DATE |
| 1. |  |  |  |  |
| 2. |  |  |  |  |
| 3. |  |  |  |  |
| 4. |  |  |  |  |
| 5. |  |  |  |  |
| 6. |  |  |  |  |

E-mail Address / Addresses (according to the order of names above):

|  |  |
| --- | --- |
| 1. | 4. |
| 2. | 5. |
| 3. | 6. |

**Contents**

[**Part 1: Project Title and Business Definition 1**](#_Toc184910350)

[**Project Title 1**](#_Toc184910351)

[**Business Area 1**](#_Toc184910352)

[**Problem Statement 1**](#_Toc184910353)

[**Project Objectives 2**](#_Toc184910354)

[**Part 2: Exploratory Data Analysis and Data Preprocessing. 2**](#_Toc184910355)

[**Significance of the Research 2**](#_Toc184910356)

[**Analytical Questions 2**](#_Toc184910357)

[**Data Needs 3**](#_Toc184910358)

[**Dataset 3**](#_Toc184910359)

[**Variables Description 3**](#_Toc184910360)

[**Data Cleaning 5**](#_Toc184910361)

[**Exploratory Data Analysis 6**](#_Toc184910362)

[**Feature Engineering 12**](#_Toc184910363)

[**Part 3: Machine Learning Approach 13**](#_Toc184910364)

[**Machine Learning Model: 13**](#_Toc184910365)

[**Model Evaluation 16**](#_Toc184910366)

[**Part 4: Implementation and Results 18**](#_Toc184910367)

[**Base Models Testing 18**](#_Toc184910368)

[**Soft Voting 30**](#_Toc184910369)

[**Part 5: Business Insights and Recommendations 33**](#_Toc184910370)

[**Business Insights 33**](#_Toc184910371)

[**Actionable Recommendations 35**](#_Toc184910372)

[**Limitations and Future Work 37**](#_Toc184910373)

[**References 38**](#_Toc184910374)

**List of Figure**

[**Figure 2.1: Missing Values in the Dataset** 5](#_Toc184910451)

[**Figure 2.2: Summary Statistics** 6](#_Toc184910452)

[**Figure 2.3: Summary Statistics (Continued)** 6](#_Toc184910453)

[**Figure 2.4: Summary Statistics (Continued)** 7](#_Toc184910454)

[**Figure 2.5: Correlation Matrix** 8](#_Toc184910455)

[**Figure 2.6: Outlier Detected** 10](#_Toc184910456)

[**Figure 2.7: Outlier Cleaning using 1.5 Threshold** 10](#_Toc184910457)

[**Figure 2.8: Outlier Detected using 1.0 Threshold** 11](#_Toc184910458)

[**Figure 4.1: Accuracy for Base Models** 18](#_Toc184910459)

[**Figure 4.2: The Performance for Top Five Models** 19](#_Toc184910460)

[**Figure 4.3: The ROC Curve for Top 3 Models by AUC Score** 19](#_Toc184910461)

[**Figure 4.4: The Precision-Recall Curves for Top 3 Models** 20](#_Toc184910462)

[**Figure 4.5: Confusion Matrix for Random Forest** 21](#_Toc184910463)

[**Figure 4.6: Confusion Matrix for Gradient Boosting Machine** 21](#_Toc184910464)

[**Figure 4.7: Confusion Matrix for XGBoost** 22](#_Toc184910465)

[**Figure 4.8: The Training Set before Feature Selection** 22](#_Toc184910466)

[**Figure 4.9: The Training Set after Feature Selection** 23](#_Toc184910467)

[**Figure 4.10: Performance for Random Forest after Tuning** 24](#_Toc184910468)

[**Figure 4.11: ROC Curve for Random Forest after Tuning** 25](#_Toc184910469)

[**Figure 4.12: Performance for Gradient Boosting Machines after Tuning** 26](#_Toc184910470)

[**Figure 4.13: ROC Curve for Gradient Boosting Machines after Tuning** 26](#_Toc184910471)

[**Figure 4.14: Performance for XGBoost after Tuning** 27](#_Toc184910472)

[**Figure 4.15: ROC Curve for XGBoost after Tuning** 28](#_Toc184910473)

[**Figure 4.16: Confusion Matrix for Random Forest after Tuning** 28](#_Toc184910474)

[**Figure 4.17: Confusion Matrix for Gradient Boosting Machine after Tuning** 29](#_Toc184910475)

[**Figure 4.18: Confusion Matrix for XGBoost after Tuning** 29](#_Toc184910476)

[**Figure 4.19: Performance for Soft Voting** 31](#_Toc184910477)

[**Figure 4.20: ROC Curve for Soft Voting Classifier** 32](#_Toc184910478)

[**Figure 4.21: Confusion Matrix for Soft Voting Classifier** 32](#_Toc184910479)

[**Figure 5.1: Feature Importance based on Random Forest Model** 33](#_Toc184910480)

# Part 1: Project Title and Business Definition

## **Project Title**

Predicting Hotel Booking Cancellations to Optimize Revenue Management

## **Business Area**

Customer retention and revenue optimization in the hospitality industry.

The hospitality industry encounters significant revenue challenges due to last-minute booking cancellations. These cancellations often lead to revenue losses, underutilized resources and inefficient operations. The ability to predict booking cancellation probabilities using machine learning can offer valuable insights for hotels to implement more effective revenue management strategies. By predicting cancellations in advance, hotels can take proactive measures such as overbooking policies or personalized customer retention offers to minimize potential losses and maximize revenue. This project aims to utilize machine learning techniques to forecast booking cancellations which allow better decision-making and optimized resource allocation within the hospitality sector.

## **Problem Statement**

Booking cancellations create a major challenge to the hospitality industry resulting in substantial revenue losses, underutilized resources and operational inefficiencies. The unpredictable nature of cancellations makes it difficult for hotels to effectively manage their occupancy rates, leading to missed opportunities in maximizing their revenue. This project aims to leverage machine learning to predict hotel booking cancellations, enabling proactive management strategies to mitigate these issues. By analyzing key factors that contribute to cancellations, such as customer demographics, booking patterns and other relevant factors, the project seeks to provide accurate predictions of cancellation probabilities. These predictions will allow hotels to adopt targeted strategies, such as overbooking policies or personalized retention offers to improve occupancy rates and optimize revenue management. The fundamental purpose of this project is to provide actionable recommendations that enhance operational efficiency and minimize revenue losses caused by last minute cancellations.

## **Project Objectives**

1. To predict booking cancellations using machine learning models based on historical data.
2. To analyze factors influencing cancellations.
3. To enable dynamic management strategies, such as overbooking or customer re-engagement to mitigate the impact of cancellations.
4. To provide actionable insights through data visualization to assist hotel managers in making informed decisions.

# Part 2: Exploratory Data Analysis and Data Preprocessing.

## **Significance of the Research**

Predicting hotel booking cancellations is crucial for the hospitality industry as it directly influences operational efficiency, revenue management and customer satisfaction. Cancellations often result in lost revenue, underutilized resources and logistical inefficiencies, which can severely impact the financial well-being of hotel businesses. By developing and implementing a predictive model for cancellations, hotels can reduce these issues in several ways. Firstly, it supports revenue optimization by allowing for overbooking strategies or the reallocation of inventory to offset the financial impact of cancellations. Additionally, the model can enhance customer experience by identifying at-risk bookings and offering proactive solutions such as alternative arrangements or incentives to retain guests. Predicting cancellations also allows for more efficient resource allocation, improving planning for staffing, room availability and inventory management. Finally, it helps reduce operational costs by minimizing waste, ensuring resources are not left unutilized due to last-minute cancellations. This research ultimately provides actionable insights that can enhance both the financial and operational performance of the hospitality industry.

## **Analytical Questions**

This project will address the following key questions:

1. What factors most strongly influence booking cancellations?
2. Can machine learning models accurately predict whether a booking will be canceled based on historical data?
3. How do customer demographics and booking characteristics correlate with cancellation probabilities?
4. What insights can be visualized to support decision-making for proactive strategies, such as overbooking or retention offers?

## **Data Needs**

The analysis and model development requires the following data:

* Customer demographics include details about the guests, such as their country of origin.
* Booking characteristics include information like lead time, deposit type, market segment, booking changes, meal type, and the total duration of stay.
* Hotel information which is the type of hotel, room type reserved and assigned.
* Operational data includes information on special requests, previous cancellations and booking changes.
* Outcome or target variable which is whether the booking was canceled.

## **Dataset**

The chosen dataset, “Hotel Booking Demand” is publicly available on Kaggle. This dataset contains detailed booking information for 119,391 hotel stays at city and resort hotels from July 2015 to August 2017.

## **Variables Description**

* 1. **Customer Demographics**
* **Country** (Categorical): Origin of the customer (ISO 3166-3 format).
* **Is\_repeated\_guest** (Binary): Indicates if the booking is from a repeat guest (1 = Yes, 0 = No).
* **Adults** (Integer): Number of adults.
* **Children** (Integer): Number of children.
* **Babies** (Integer): Number of babies.
  1. **Booking Characteristics**
* **Lead\_time** (Integer): Days between booking date and arrival date.
* **Arrival\_date\_year** (Integer): Year of the arrival date.
* **Arrival\_date\_month** (Categorical): Month of the arrival date (January to December).
* **Arrival\_date\_week\_number** (Integer): Week number of the arrival date.
* **Arrival\_date\_day\_of\_month** (Integer): Day of the month of the arrival date.
* **Stays\_in\_weekend\_nights** (Integer): Number of weekend nights booked.
* **Stays\_in\_week\_nights** (Integer): Number of weekday nights booked.
* **Meal** (Categorical): Type of meal package (e.g., No meal, BB, HB, FB).
* **Market\_segment** (Categorical): Market segment (e.g., Travel Agents, Tour Operators).
* **Distribution\_channel** (Categorical): Booking channel (e.g., TA, TO).
* **Deposit\_type** (Categorical): Type of deposit (No deposit, Non-refundable, Refundable).
* **Agent** (Categorical): ID of the travel agency.
* **Company** (Categorical): ID of the company/entity responsible for the booking.
* **Days\_in\_waiting\_list** (Integer): Days the booking spent on the waiting list.
* **Customer\_type** (Categorical): Type of booking (e.g., Contract, Group, Transient).
* **Booking\_changes** (Integer): Number of amendments made to the booking.
* **Adr** (Numerical): Average daily rate (lodging revenue/total nights).
  1. **Hotel Information**
* **Hotel** (Categorical): Type of hotel (City or Resort).
* **Reserved\_room\_type** (Categorical): Reserved room type (anonymized).
* **Assigned\_room\_type** (Categorical): Assigned room type (may differ from reserved type).
  1. **Operational Data**
* **Previous\_cancellations** (Integer): Number of past canceled bookings by the customer.
* **Previous\_bookings\_not\_canceled** (Integer): Number of past bookings not canceled.
* **Required\_car\_parking\_spaces** (Integer): Number of parking spaces requested.
* **Total\_of\_special\_requests** (Integer): Number of special requests made.
* **Reservation\_status** (Categorical): Final status of the booking (Canceled, Check-out, No-show).
* **Reservation\_status\_date** (Date):Date when the final status was recorded.
  1. **Outcome Variable**
* **Is\_canceled** (Binary): Indicates if the booking was canceled (1 = Yes, 0 = No).

## **Data Cleaning**

**Unique values**

Each column contains at least two distinct values, indicating that there are no columns with completely unique values.

**Missing values**

**A screen shot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generated**

**Figure 0.1: Missing Values in the Dataset**

The output above shows that some columns including children, country, agent and company contain missing values. For the children column, missing values were replaced with 0, indicating that no children were included in those bookings. In the country column, entries with null values were removed as they lacked crucial location details. For the agent and company columns, missing values were kept but filled with 0 to represent self-made bookings without the involvement of an agent or a company.

**Datatype adjustment**

The datatype of the children column was converted from float to integer for consistency, as it represents a count of individuals.

**Handle noisy data**

To address noisy data, observations with 0 adults were reviewed and dropped. These 403 records were considered likely data entry errors, as it is highly improbable for bookings to consist of only children or babies without adults.

Additionally, any “undefined” values in the Distribution Channel and Market Segment columns were removed as these values lack interpretive significance. For the Meal column, undefined values were replaced with “SC”, indicating no meals were ordered, aligning with its equivalent meaning in the dataset.

**Data transform into binary indicators**

To improve machine learning outcomes, we transformed the data by replacing all entries in the agent column with 1 if a value is present indicating an agent and 0 if no value is present indicating no agent. A similar transformation was applied to the company column, values were replaced with 1 to indicate the presence of a company and 0 for its absence.

## **Exploratory Data Analysis**

**Summary statistics**

A white background with black and blue text

Description automatically generated

**Figure 0.2: Summary Statistics**

A screenshot of a computer

Description automatically generated

**Figure 0.3: Summary Statistics (Continued)**

A screenshot of a computer

Description automatically generated

**Figure 0.4: Summary Statistics (Continued)**

The dataset presents key insights into various aspects of hotel bookings. The target variable, is\_canceled, indicates that approximately 37.1% of bookings were canceled. The lead\_time variable, which represents the number of days between booking and arrival, has a mean of 104.3 days, with values ranging from 0 to 737 days. This suggests that while most bookings are made well in advance, last-minute bookings also still occur. Bookings primarily occur in 2016, as indicated by the arrival\_date\_year variable, with the median arrival week being 28, suggesting a mid-year peak.

When examining stay duration, the mean for stays\_in\_weekend\_nights is 0.93, and the mean for stays\_in\_week\_nights is 2.5, suggesting that stays are generally short with fewer weekend nights. The dataset also reveals that most bookings involve 1 to 2 adults, with a relatively low occurrence of children or babies. Only about 3% of bookings are made by repeated guests, highlighting the transient nature of majority guests.

In terms of booking modifications, the dataset shows that most bookings have no prior cancellations, with a mean of 0.08 for previous\_cancellations and 0.13 for previous\_bookings\_not\_canceled. Additionally, the booking\_changes variable indicates that most bookings are not modified with a mean of 0.22.

The agent column indicates that a significant portion of bookings (about 86%) are made via agents, while only 5.5% of bookings are associated with a company. In terms of additional features, the days\_in\_waiting\_list variable suggests that most bookings experience minimal waiting time, with a mean of 2.33 days. However, the adr (Average Daily Rate) shows significant variation, with a maximum value of 5400, suggesting the presence of outliers or luxury bookings. Additionally, the required\_car\_parking\_spaces variable has a low mean, indicating that parking requests are uncommon. Finally, the total\_of\_special\_requests variable shows that most guests do not make special requests, with a mean of 0.57.

**Correlation matrix**

A diagram with numbers and a red line

Description automatically generated with medium confidence

**Figure 0.5: Correlation Matrix**

The correlation matrix reveals several important relationships between features in the dataset. The target variable, is\_canceled, exhibits a moderate positive correlation with lead\_time (0.25), indicating that bookings that were made further in advance have a higher chance to be cancelled. On the other hand, previous\_bookings\_not\_canceled shows a strong negative correlation with is\_canceled (-0.53), suggesting that customers with a history of successful bookings are less likely to cancel. Additionally, required\_car\_parking\_spaces has a weak negative correlation (-0.18) with is\_canceled, implying that bookings requiring parking spaces are slightly less likely to be canceled.

Other significant correlations include a moderate positive relationship (0.48) between stays\_in\_week\_nights and stays\_in\_weekend\_nights, indicating that customers who stay longer during the end of the weekdays are more likely to extend their stays into the weekend. Similarly, features like adults and total\_of\_special\_requests show moderate correlations with adr (average daily rate), highlighting that bookings with more adults or special requests tend to be associated with higher rates.

**Outliers**

After addressing the missing values, the outliers are detected using the Interquartile Range (IQR) method which is a technique for identifying outliers in continuous data (Vinutha et al., 2018). The IQR represents the range between the 25th and 75th percentiles (Q1 and Q3) respectively, capturing the middle 50% of the data (Barbato et al., 2011). With the IQR, Q1, and Q3 values, the lower and upper fences can be calculated. These fences are essential for identifying outliers, as any value that falls below the lower fence or above the upper fence is considered an outlier. In this case, we apply a –1/+1 threshold using the following formulas:

Lower Fence = Q1 - 1IQR

Upper Fence = Q3 + 1IQR

Initially, a –1.5/+1.5 threshold was used to handle the outliers. However, it was observed that not all the outliers were eliminated, which lead to the decision of applying a –1/+1 threshold.

A graph of a graph of a number of numbers

Description automatically generated with medium confidence

**Figure 0.6: Outlier Detected**

A comparison of a graph

Description automatically generated

**Figure 0.7: Outlier Cleaning using 1.5 Threshold**

A comparison of a graph

Description automatically generated with medium confidence

**Figure 0.8: Outlier Detected using 1.0 Threshold**

The outliers in the “children” and “babies” columns were also identified. As a result, a decision was made to remove these outliers based on the following conditions, where if the number of children exceed 2 or the number of babies exceed 3.

**Remove directly related features**

Features such as reservation\_status and reservation\_status\_date are directly linked to the target variable, is\_canceled. Reservation\_status is a categorical feature that indicates the current status of a reservation with values like canceled, check-out and no-show. If the reservation status is canceled, the is\_canceled should be 1 automatically. Using this feature in the model would directly reveal the target variable and cause data leakage. This implies that the model would depend only on this column to predict the answer, neglecting the process of learning from other patterns in the data. Therefore, it is important to remove this feature to build a model that can actually predict booking cancellations. Reservation\_status\_date is a date feature that indicates the latest date at which reservation status was updated. Hence, this feature is also considered irrelevant and should be excluded. For the assigned\_room\_type feature, it represents the type of room assigned to the guest. It is assigned after the booking is made, so it cannot be used to predict cancellation before the guests cancel the booking. Thus, this feature should also be omitted.

**Drop irrelevant features**

Since the arrival\_date\_year in this dataset only includes three years, it is recommended to exclude this feature to make sure the model is not biased towards specific years. Therefore, the model would likely fail to generalize effectively to new and unseen data.

## **Feature Engineering**

**Converting categorical data into numerical format**

The arrival\_date\_month variable, which contains categorical values in the form of month names such as “January” and “February”, is transformed into numerical representations corresponding to the months. For example, January is represented as 1, February is represented as 2 and so on. This transformation is achieved by assigning each month to its index in a predefined list and adding 1 to align it with the standard month numbering. Converting the data into a numerical format is an important step for analysis and compatibility with machine learning techniques, as they require numerical inputs. Additionally, this process also standardizes the representation of months which in turn reduces potential inconsistencies that may arise due to string-based categorical values.

**One-hot encoding**

The reason why one-hot encoding is used is that many machine learning algorithms cannot work directly with categorical data, as they require numerical input for calculations. By converting categorical variables into binary indicator variables, one-hot encoding allows the model to interpret and process the data without introducing unintended ordinal relationships, which could occur if arbitrary numerical values were assigned to the categories.

For this dataset, columns such as hotel, meal, market\_segment, distribution\_channel, reserved\_room\_type, deposit\_type, and customer\_type are one-hot encoded. Each unique category within these columns is converted into a separate binary column, where a value of 1 indicates the presence of the category and 0 indicates its absence. In order to prevent multicollinearity, one binary column from each set is excluded, ensuring the encoded data remains free from redundancy. This transformation is essential for algorithms to accurately capture relationships within the data while at the same time avoiding biases associated with categorical variables.

After that, we applied get\_dummies function to create a new binary column for each unique category within the “country” variable, representing whether the observation belongs to that specific category. It drops the first category to prevent multicollinearity, as having all dummy variables would lead to perfect correlation between them. This approach is common when encoding categorical variables for machine learning, as it converts the data into a form that can be easily used by most algorithms that require numerical input.

Feature scaling is another important preprocessing step in machine learning, especially when the features in the dataset have varying ranges or units. Using MinMaxScaler, all the features in the dataset are scaled to a common range, typically between 0 and 1. This ensures that no single feature influences the learning process due to its larger magnitude. The fit\_transform() method of the scaler is used to compute the minimum and maximum values for each feature and then scale them accordingly. After the scaling, the resulting transformed data is converted back into a Pandas DataFrame to maintain the column names and structure of the dataset. This step is essential for models that are sensitive to the scale of the data, such as distance-based algorithms like KNN or neural networks, helping them to learn and perform more effectively.

# Part 3: Machine Learning Approach

## **Machine Learning Model**

Since the target variable in this dataset is booking cancellation, which is represented as a binary outcome where 0 for non-canceled bookings and 1 for canceled bookings, a classification model is the most suitable approach for the analysis. Classification models are designed to predict categorical outcomes, making them ideal for binary situations. By applying these models, we can uncover patterns and factors that influence booking cancellations, providing valuable predictive insights for informed decision-making. The classification models employed in this research include logistic regression, K-Nearest Neighbors (KNN) classifier, decision tree, Naive Bayes, gradient boosting classifier, random forest, and XGBoost classifier. These methods fall under supervised learning, where the model is initially trained on known data, and then, when new, unseen data is introduced, the system is expected to accurately classify it.

**Logistic Regression**

Logistic Regression is a commonly used classification method that examines the relationship between a categorical dependent variable and a set of independent variables. Typically, the dependent variable has two possible outcomes, such as 0/1 or Yes/No, which makes this technique particularly useful for predicting binary results (Verma & Verma, 2019). The model works by estimating the parameters of a logistic equation, where the log-odds of the outcome being “1” are represented as a linear combination of the independent variables (Verma & Verma, 2019). Known for its simplicity and ease of use, Logistic Regression performs well across a variety of tasks, including spam detection, and is a versatile tool for classification problems (Othman & Din, 2019).

**K-Nearest Neighbors (KNN) Classifier**

The K-nearest neighbor (KNN) classifier assigns an object to a class based on the majority vote of its nearest neighbors in the input parameter space. The object is categorized into the class that is most common among its K closest neighbors, with K being an integer value determined by the user. The classification decision is made by a simple majority vote from the nearest neighbors of each data point (Syriopoulos et al., 2023). The advantages of this algorithm include its ease of implementation, robustness to noisy training data, and effectiveness when the training dataset is large (Guo et al., 2003). However, its disadvantages include the need to determine the appropriate value for K and the high computational cost, as it requires calculating the distance between each instance and all training samples (Zhang, 2021).

**Decision Tree**

A decision tree is a type of machine learning algorithm that generates a set of rules to classify data using a tree-like structure, as its name suggests. It divides the sample into two or more homogeneous groups based on the most significant differentiating factors within the input variables (Tangirala, 2020). To identify these differentiators, the algorithm evaluates all features, performing binary splits on them, and selects the one with the lowest cost. This process is repeated recursively until the data is fully divided into leaves or the maximum depth of the tree is reached (Herrera et al., 2024). According to the study by Charbuty and Abdulazeez (2021), decision trees are valued for their simplicity, ease of visualization, minimal data preparation requirements, and its ability to handle both numerical and categorical data. However, decision trees also have disadvantages, such as their tendency to generate overly complex trees that may not generalize well to new data, and their instability, as small changes in the dataset can lead to the creation of entirely different trees (Priyanka & Kumar, 2020).

**Naive Bayes**

The Naive Bayes Classifier is based on Bayes’ theorem and is often used when handling high-dimensional input data. It utilizes the method of maximum likelihood for parameter estimation, making it particularly valuable in complex real-world situations, especially when training data and parameter estimation are limited (Nakra & Duhan, 2019). This classifier operates under the assumption of class conditional independence, meaning the impact of each attribute value on a given class is independent of the other attributes. This simplification, known as “naïve”, helps streamline computational processes. In contrast, Bayesian belief networks are graphical models that represent the dependencies between attribute subsets and can also be applied to classification tasks (Tribhuvan, Tribhuvan & Gade, 2015).

**Gradient Boosting Classifier**

The Gradient Boosting Classifier is a machine learning technique that combines multiple weak models, typically decision trees to create a strong predictive model (Trang et al., 2021). It builds these models sequentially, with each new model correcting the errors of the previous ones. This iterative process allows Gradient Boosting to effectively capture complex, non-linear relationships in the data, making it suitable for both regression and classification tasks (Upadhyay et al., 2020). Based on the study from Bentéjac et al. (2020), this technique offers several advantages, including high accuracy, flexibility to work with various loss functions and the ability to identify important features automatically. However, it has some limitations, such as longer training times due to its sequential nature, a tendency to overfit noisy data if not properly tuned and reduced interpretability compared to simpler models.

**Random Forest**

Random Forest is a popular ensemble technique used for both classification and regression tasks, known for enhancing accuracy and minimizing overfitting. The algorithm constructs multiple decision trees during training, each using a random selection of features, and combines their outputs to generate a final prediction. This method is effective for handling large, high-dimensional datasets and ensures robust performance by averaging the results from several trees to reduce overfitting (Clarin, 2022). However, Random Forest can be computationally intensive and time-consuming, particularly as the number of trees grows, which may slow down the training process. Additionally, due to its complexity, Random Forest is less interpretable than simpler models like individual decision trees (Patil & Burkpalli, 2021).

**XGBoost Classifier**

eXtreme Gradient Boosting (XGBoost) is a powerful and efficient machine learning algorithm widely used for classification and regression tasks. It operates by building multiple decision trees sequentially, where each tree corrects the errors of the previous ones, enabling the model to learn complex patterns in the data (Herrera et al., 2024). Sinha (2020) and Bentéjac et al. (2021) highlight that XGBoost is known for its high speed and scalability, making it suitable for large datasets. It effectively handles both numerical and categorical data while incorporating regularization techniques to reduce overfitting which in turn ensures better generalization. However, XGBoost can be computationally intensive for very large datasets and requires careful parameter tuning to achieve optimal performance. Additionally, it is less interpretable compared to simpler models such as decision trees, which can pose challenges in understanding its predictions.

## **Model Evaluation**

The model performance of predictive models will be evaluated using several key metrics including accuracy, precision, recall, and F1 score. For a comprehensive view, a confusion matrix will summarize the predicted outcomes, while the ROC-AUC curve and precision-recall curve will further illustrate the model’s classification ability.

**Confusion Matrix**

The confusion matrix is a vital tool for evaluating classification algorithms. It provides insights into the accuracy, sensitivity, and specificity of the model by presenting a tabular summary of actual and predicted classifications.

The matrix consists of four key components as below.

True Positive (TP) is where the booking is correctly predicted to be cancelled and was actually cancelled.

False Negative (FN) is where the booking is incorrectly predicted as not cancelled but was actually cancelled.

False Positive (FP) is where the booking is incorrectly predicted to be cancelled but was not actually cancelled.

True Negative (TN) is where the booking is correctly predicted as not cancelled and was not actually cancelled.

**Evaluation Metrics**

Accuracy measures the proportion of correctly predicted instances including both canceled and not canceled out of the total predictions made by the model. Different algorithms may vary in their performance regarding correctly classified instances.

Precision indicates the proportion of true positives among all positive predictions, reflecting prediction accuracy for cancellations.

Recall measures the completeness of predicting proportion of actual cancellations correctly identified by the model.

F-Measure measures the harmony and balance of precision and recall.

The Receiver Operating Characteristic (ROC) curve plots the true positive rate which is sensitivity against the false positive rate (1-specificity) at various threshold levels. AUC (Area Under Curve) represents the model’s overall ability to differentiate between classes. Higher values indicate better classification performance.

The precision-recall curve provides a complementary view that focuses on precision and recall trade-offs across thresholds. It is particularly useful for imbalanced datasets, where the ROC curve might not adequately reflect performance.

# Part 4: Implementation and Results

## **Base Models Testing**

Through the use of Python in Google Colab, the proposed regression-based machine learning models were implemented, and their performances were evaluated. In order for these models to be evaluated, the dataset must be split into training and testing sets. In this case, 75% of the data was split for the training set, while the remaining 25% was used as the testing set. The training set was used to train the data, while the testing set is used to evaluate the model’s performance. With the use of the testing set, the performance metrics that were evaluated include the model’s accuracy, precision, recall, F1 score, ROC curve, precision-recall curve, and their confusion matrix.

A white background with black numbers

Description automatically generated

**Figure 0.1: Accuracy for Base Models**

The preliminary step was base model testing, where seven models using their default settings and parameters were tested to identify and select the top 3 best performers. During this step, the seven models’ accuracies are initially tested using k-fold cross validation, using 10 folds, with the purpose of narrowing down the seven models to only five models. Based on the results, it may be observed that based on the models’ accuracies, the top five models are RF\_Gini100, XGB, CART, KNN7, and GBM respectively.

A screenshot of a graph

Description automatically generated

**Figure 0.2: The Performance for Top Five Models**

After the top five models have been identified, these models have been tested using the training and testing sets, with their performance metrics being displayed in the table above. Based on the table, it may be concluded that from the base model testing, RF\_Gini100 is the best model overall due to it producing the most desirable metrics across the board aside from sensitivity. The next best model is XGB as it produces comparable metrics to RF\_Gini100. Lastly, based on these metrics, CART is the next best model. CART performs relatively well, however RF\_Gini100 and XGB both outperform this model.

A diagram of a curve

Description automatically generated

**Figure 0.3: The ROC Curve for Top 3 Models by AUC Score**

Aside from the evaluation metrics displayed on the table above, the Receiver Operating Characteristic (ROC) curve is a key evaluation metric that needs to be analyzed in order to select the top 3 models. In ideal model would have an Area Under Curve (AUC) value of 1, indicating perfect classification. As seen in the graph above, the ROC curves for the top 3 models with the highest AUCs are plotted. Those models being RF\_Gini100 (AUC = 0.949), XGB (AUC = 0.936), and GBM (AUC = 0.902) respectively. Based on the observations from the ROC-AUC plot, the ROC curves for RF\_Gini100 and XGB further reinforce these models as the best performing models among the rest. In terms of GBM, its ROC curve justifies why it is one of the best performing models despite having weaker performance displayed by the table.

A graph of different colored lines

Description automatically generated

**Figure 0.4: The Precision-Recall Curves for Top 3 Models**

The plot above displays the Precision-Recall curves for the top 3 models. Based on the average precisions, RF\_Gini100 (AP = 0.923) performs the best, followed by XGB (AP = 0.901), and GBM (AP = 0.854). The PR curve for RF\_Gini100 demonstrates a good balance between precision and recall as the curve extends adequately to both the right and top of the plot. This indicates that the trade-off between precision and recall is relatively smooth, which means that this model is able to consistently predict less false positives and false negatives. XGB’s PR curve is similar to RF\_Gini100’s curve, however it demonstrates significant drops in precision when its recall is higher. As a result, this model may predict more false positives when capturing positives. Finally, GBM’s PR curve is the worst among the three due to its significant drop in precision as recall increases, indicating that its predictions are more prone to be false positives when it captures positives.

A blue squares with white text

Description automatically generated

**Figure 0.5: Confusion Matrix for Random Forest**

A blue squares with white text

Description automatically generated

**Figure 0.6: Confusion Matrix for Gradient Boosting Machine**

A blue squares with white text

Description automatically generated

**Figure 0.7: Confusion Matrix for XGBoost**

The confusion matrices above display each model’s predictions, indicating their True Negatives, False Positives, False Negatives, and True Positives. By comparing the three matrices, it may be concluded that RF\_Gini100 has the best accuracy (88.4%) due to having the highest True Negative and True Positive. XGB has the next best accuracy (86.2%). However, among the three models, XGB has the lowest True Positive value, meaning it is the worst at predicting positives correctly among these models. Lastly, GBM has the lowest accuracy (81.7%) among these models. Its True Positive is worse than both models, however its True Negative is better than XGB’s. Based on the evaluation metrics, ROC-AUC plot, PR plot, and confusion matrices, it may be concluded that the top 3 performing models are RF\_Gini100, XGB, and GBM.

**Feature Selection**

A screenshot of a computer

Description automatically generated

**Figure 0.8: The Training Set before Feature Selection**

A screenshot of a computer

Description automatically generated

**Figure 0.9: The Training Set after Feature Selection**

After the top 3 models have been selected, feature selection may be performed by using Recursive Feature Elimination (RFE) from sklearn.feature\_selection. This step is crucial as it removes any features that do not significantly contribute to the model’s decision making. Furthermore, due to having less features, it elevates computational load when training these models, allowing for more efficient training. This is especially crucial when using a relatively large dataset. As RF\_Gini100, XGB, and GBM are the top performing models, RFE was performed on each of these models. Once RFE is performed on each model, the union of features are selected, meaning that all features selected by each model are selected. As a result, the number of features reduced from 215 to 154 as seen above.

**Hyperparameter tuning**

After feature selection, the top 3 models may now be tuned to increase their performance. This was done by finding their best parameters using RandomizedSearchCV. This hyperparameter tuning technique will search for the best parameters for each model by randomly sampling and testing hyperparameter values. After the specified number of iterations of parameter sampling has been conducted, the best parameters will be produced. The model will then be trained and tested using these hyperparameters.

**Random Forest (Tuned)**

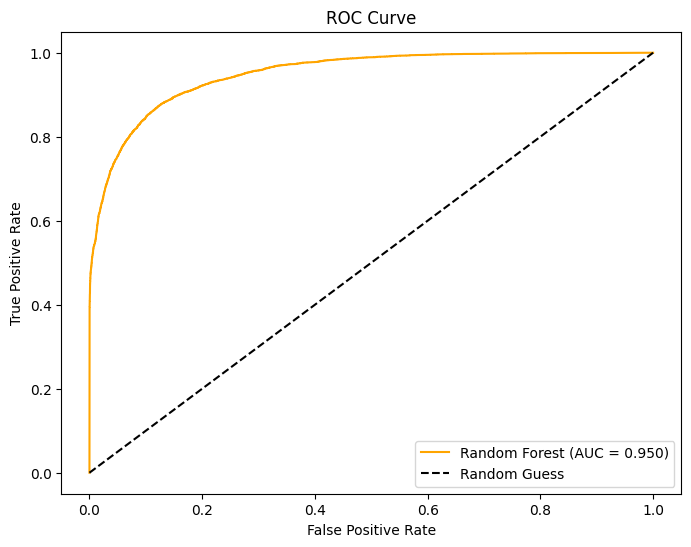
After running RandomizedSearchCV for Random Forest, the best parameters were, ‘n\_estimators’: 300, ‘min\_samples\_split’: 5, ‘min\_samples\_leaf’: 1, ‘max\_features’: ‘sqrt’, ‘max\_depth’: None. It should be noted that, only for RF, the number of iterations of random search was reduced to 10 and the cross-validation (CV) value was reduced to 3. This was done to speed up the process of finding the best parameters. Since the dataset is large enough, reducing these values would not have a huge impact on the results.

A screenshot of a computer

Description automatically generated

**Figure 0.10: Performance for Random Forest after Tuning**

Comparing the metrics from the base model testing with the tuned RF metrics, it may be observed that there are slight decreases in its accuracy, precision, recall, and F1-score. However, the decreases across these metrics are very minimal, which demonstrates the stability of this model, as the drop in performance is not drastic. Although there are slight drops in these metrics, this tune RF model made improvements in the form of a significantly lower log loss (from 4.17314 to 0.282). This drop in log loss may indicate improved confidence in prediction.



**Figure 0.11: ROC Curve for Random Forest after Tuning**

By observing the ROC curve for RF, it may be noted that the AUC has improved compared to the base model, from 0.949 to 0.950. After tuning the model, it brings the AUC value closer to 1, which indicates that this tuned model has the ability to predict positive and negative classes more reliably compared to the base model.

**Gradient Boosting Machines (Tuned)**

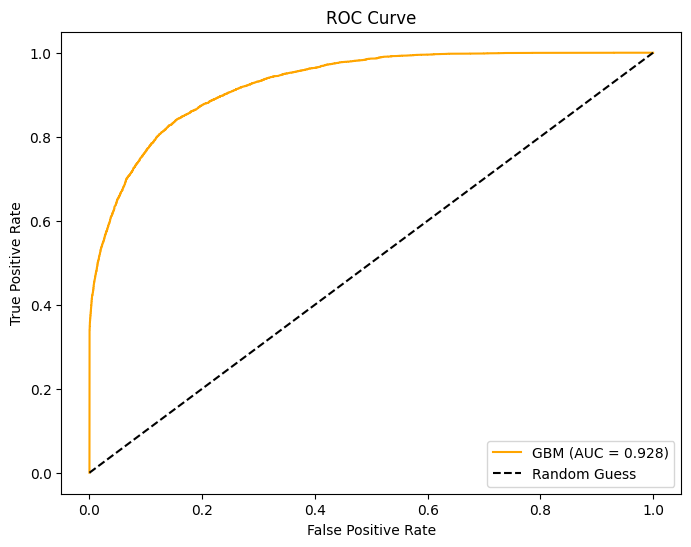
After running RandomizedSearchCV for GBM, the best parameters were ‘n\_estimators’: 300, ‘min\_samples\_split’: 5, ‘min\_samples\_leaf’: 1, ‘max\_features’: ‘log2’, ‘max\_depth’: 5, and ‘learning\_rate’: 0.2. These parameters were selected by running 20 iterations of random search with a CV value of 10, resulting in a total of 100 fits.

A screenshot of a computer

Description automatically generated

**Figure 0.12: Performance for Gradient Boosting Machines after Tuning**

Comparing the metrics from the base model testing with the tuned GBM metrics, it may be observed that this model has made improvements in terms of accuracy, recall, F1-score, and log loss. GBM’s accuracy increased by 3.6%, which indicates improvement in making correct predictions. Its significant improvements in recall (improved by 13.1%) and F1-score (improved by 8%) indicate improvement in handling imbalanced classes. Finally, GBM’s log loss significantly dropped from 6.579 to 0.317, indicating an improvement in its confidence in predictions.



**Figure 0.13: ROC Curve for Gradient Boosting Machines after Tuning**

Based on GBM’s tuned ROC curve, its AUC has made a significant improvement compared to the base model, from 0.902 to 0.928. This significant increase in GBM’s AUC indicates major improvements in the model’s classification capability.

**XGBoost (Tuned)**

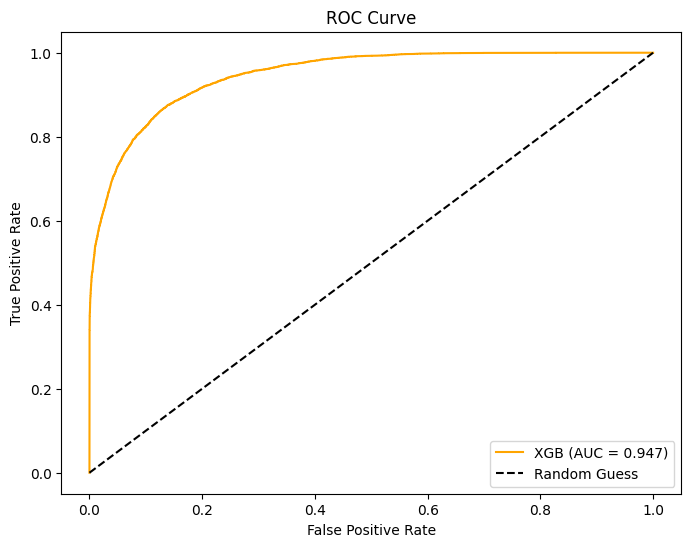
After running RandomizedSearchCV for XGBoost, the best parameters were ‘n\_estimator:’ 200, ‘max\_depth’: 9, ‘learning\_rate’: 0.2, and ‘subsample’: 1.0. These parameters were selected by running 20 iterations of random search with a CV value of 10, resulting in a total of 100 fits.

A screenshot of a computer

Description automatically generated

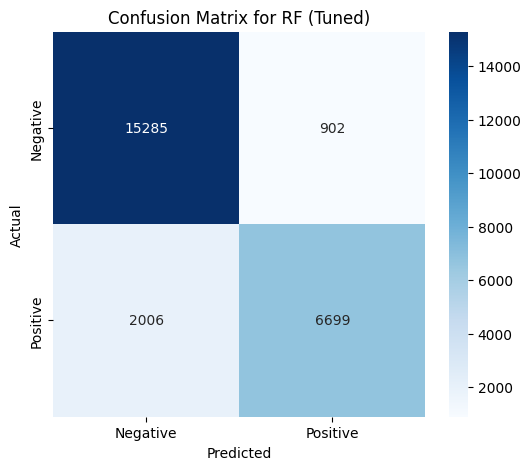
**Figure 0.14: Performance for XGBoost after Tuning**

After tuning the XGB model, it may be observed that there are improvements across all metrics compared to its base model. Its accuracy improved by 1.4%, indicating an improvement in making correct predictions. Its precision increased by 1.9% while the recall increased by 2.3%, indicating that the model has improved in terms of making less false positive predictions, along with making more true positive predictions. In turn, this improvement in both precision and recall increases its F1-score by 2.1%.

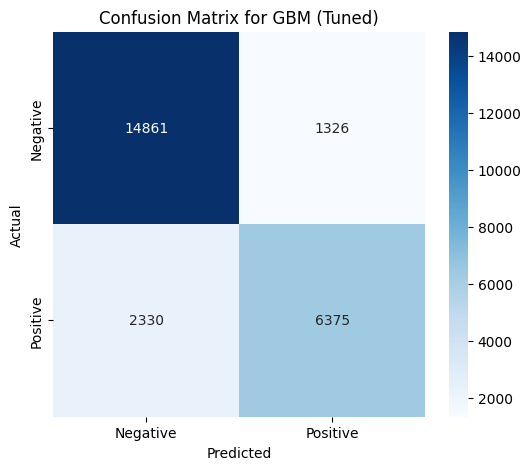


**Figure 0.15: ROC Curve for XGBoost after Tuning**

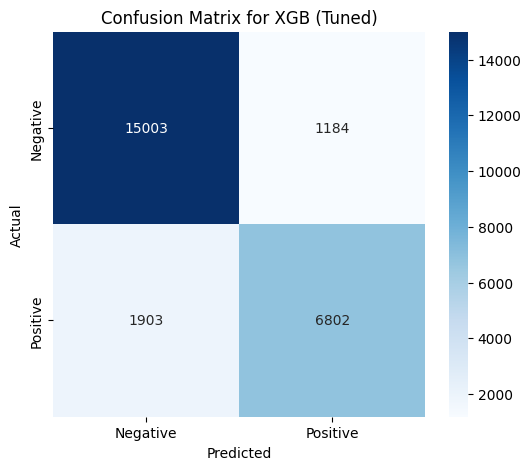
Looking at XGB’s tuned ROC curve, it may be observed that its AUC has improved significantly compared to its base model, from 0.936 to 0.947. Similarly to RF and GBM, this improvement in its AUC indicates an improvement in the reliability of this model’s classification.



**Figure 0.16: Confusion Matrix for Random Forest after Tuning**



**Figure 0.17: Confusion Matrix for Gradient Boosting Machine after Tuning**



**Figure 0.18: Confusion Matrix for XGBoost after Tuning**

The confusion matrices above each model’s predictions after tuning. Observing the tuned RF confusion matrix, the tuned RF model also has the highest True Negative value (15285) compared to the other tuned models. This indicates that this model is the best at predicting non-cancellations. Furthermore, it may be noticed that it is able to predict non-cancellations with high accuracy due to the low number of False Positives. Where the tuned RF model falls off is accurately predicting cancellations, as its False Negative value is relatively high.

Looking at the tuned GBM confusion matrix, it may be observed that the tuned GBM model have the highest False Positive and False Negative values (1326 and 2330 respectively). Having higher values for these aspects of the confusion matrix demonstrates its lower accuracy compared to the other tuned models.

Finally, observing the tuned XGB confusion matrix, it may be observed that it has the highest True Positive value compared to the other tuned models. Furthermore, it also has the least number of False Negatives (1903). These values indicate that the tuned XGB model is the best at predicting cancellations compared to the other tuned models.

Based on these various evaluation metrics, it may be concluded that Random Forest is the best model in terms of overall performance, specifically predicting non-cancellations. XG Boost is a good alternative as it produces comparable metrics with Random Forest. XG Boost thrives in predicting cancellations compared to Random Forest.

## **Soft Voting**

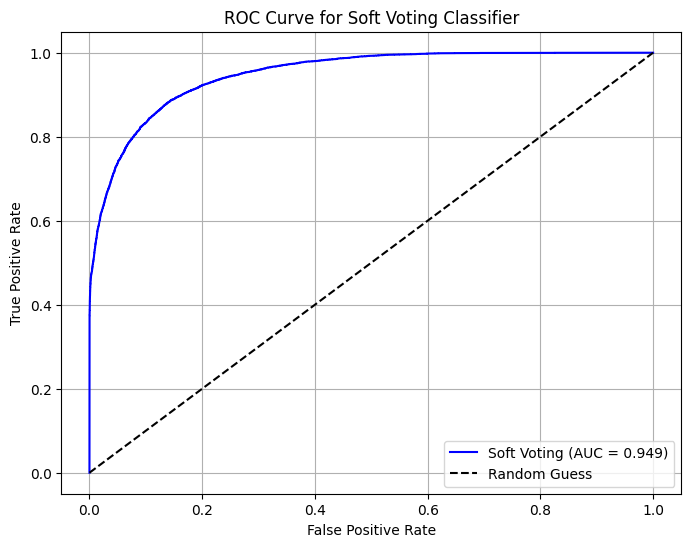
Once the models have undergone hyperparameter tuning and their metrics have been evaluated, these tuned models may be used for a machine learning ensemble technique called soft voting. This technique utilizes the strengths of each model to create an ensemble classifier, the soft voting classifier. This soft voting classifier will take the probabilities of each tuned model, for both positive and negative outcomes, and average them out. The higher average is what the soft voting classifier will predict (eg. if the average probability for 1 is higher than the average probability for 0, the soft voting classifier will predict 1).

A screenshot of a computer

Description automatically generated

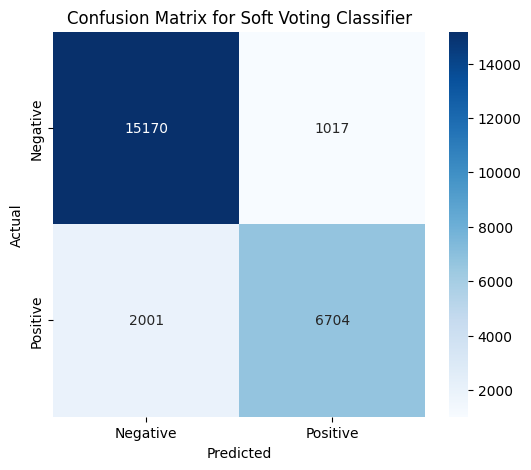
**Figure 0.19: Performance for Soft Voting**

After creating and testing the Soft Voting Classifier, it may be observed that this classifier achieves an accuracy of 0.879, which is slightly lower than RF, but higher than both XGB and GBM. Its precision of 0.868 is comparable to RF, but higher than both XGB and GBM. This indicates that the Soft Voting Classifier is able to make lesser False Positive predictions. Meanwhile, its recall of 0.770 is equal to RF, but lower than both XGB and GBM. This indicates that the Soft Voting Classifier is able to efficiently predict positives. The F1-score of 0.816 is slightly lower than RF and XGB, but higher than GBM. The F1-Score reflects a good balance between its precision and recall. Finally, its log loss of 4.370 is relatively high compared to the tuned models, which indicates that the Soft Voting Classifier is more confident about incorrect predictions. Overall, the Soft Voting Classifier’s metrics portray a good balance of each model’s strengths.



**Figure 0.20: ROC Curve for Soft Voting Classifier**

Observing the Soft Voting Classifier’s ROC curve, its AUC is comparable to RF as it is only worse by 0.01. This high AUC value indicates that this classifier can reliably predict positive and negative classes.



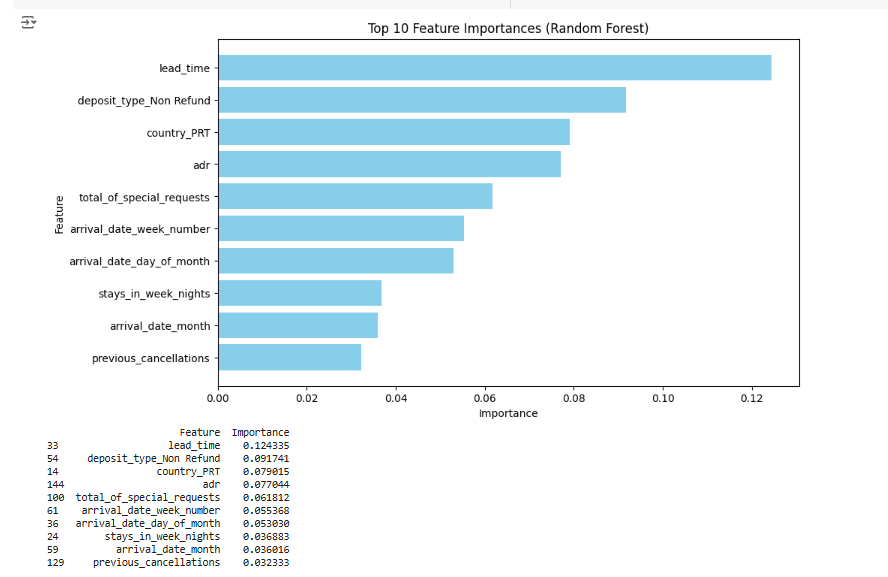
**Figure 0.21: Confusion Matrix for Soft Voting Classifier**

Based on the confusion matrix for the Soft Voting Classifier it may be observed that it reflects a good balance between the tuned RF and XGB models. There is a compromise between predicting True Negatives and True Positives, as the number of predicted True Negatives by the Soft Voting Classifier is 15170, which is slightly lower than RF but higher than XGB. Meanwhile, the number of predicted True Positives by the Soft Voting Classifier is 6704, which is slightly lower than XGB but higher than RF. This compromise makes this classifier more balanced in which class it predicts, meaning it does not specialize in predicting one class only, like how RF specializes in predicting non-cancellation (negative class), while XGB specializes more in predicting cancellations (positive class).

Overall, the Soft Voting Classifier is an effective technique that is able to combine the strengths of Random Forest, XGBoost, and GBM into an ensemble of classifier that acts as a middle ground between these classification models.

# Part 5: Business Insights and Recommendations

## **Business Insights**



**Figure 0.1: Feature Importance based on Random Forest Model**

The graph above shows significant insight into the factors influencing hotel cancellation which are vital for the hotel management to focus on to make better decisions. On the very top of the graph is lead time and it has the highest influence on booking cancellation. Lead time refers to the number of days between customers booked their hotel and checked in. By looking at this finding, it suggests that the length of lead time plays an important role in cancellation. Customers who have already confirmed booking far in advance are susceptible to cancellation due to unforeseen changes. On the other hand, customers who book in shorter lead times often show higher interest in committing the reservation. Thus, hotels management should cope with this issue by sending timely reminders to customers with long lead times, providing more personalized updates, offers or additional services to maintain their commitment to their bookings.

The second important feature is deposit type (non-refundable) which acts as a deterrent to cancellations. Customers with a financial commitment are less prone to making cancellations. This feature, which ranks among the highest shows that it is one of the most influential predictors of cancellations behavior. This might be due to several reasons such as any emergency events prior to the travel or changes in travel plan that eventually leads to cancellation. By looking at this, hotel management should implement flexible deposit options, which are essential for attracting a bigger customer base. However, they also need to have a very clear policy which emphasizes the benefits of non-refundable deposits such as discounts or perks. This can help in reducing the number of cancellations.

Guest country is the third most influential factor, with the data shows the guest in Portugal (PRT). The origin of the customers has a significant impact towards cancellation behaviors, and it can be because of the economic conditions, cultural norms and proximity to the hotels. Hotels management should have tailored strategies based on geographic segmentation as it could prove beneficial for addressing regional trends.

Another key feature is ADR (Average Daily Rate), which reflects the booking rate per day. When the ADR is higher, customers are more likely to cancel their booking potentially due to overestimating their willingness to pay for the rooms. This is also a similar case where ADR is lower where less committed customers are more prone to cancel. By looking at this feature, it is important for the hotel management to look into their pricing strategy as price not only drives demand but also can affect the likelihood of customers cancelling their bookings. Hence, by carefully calibrating ADR, hotels can have healthy revenue growth while increasing the customers retention.

There is also an increasing trend in customers making special requests. They are more likely to follow through with their reservations as they are more well-planned for their stay. This feature indicates a higher level of commitment made by the customers as they are more invested in their thoughts and plans for their stay. This could be a good indicator for the hotels for them to prioritize these customers as special requests are a strong measure of reliability. Hotels can provide more personalized communications with them so that the customers feel more valuable.

The next feature is the arrival date, which can influence the rate of cancellations depending on the seasonality and specific periods. There can be where in certain months, weeks or even days when the probability of cancellations is higher. For example, fewer cancellations can be seen during the holidays or peak travel seasons as customers are more committed, while during off-peak seasons, the cancellation rate is higher due to inconvenience face by the customers to stay at the hotels, which they may have to cancel their bookings. Thus, hotel management should factor in seasonality in their decision making as it can help in making a better pricing strategy and to increase the customers’ retention rate.

In summary, the graph indicates the key factors that influence booking cancellations. Lead time has the highest impact for cancellations, which hotels can cope with by sending a reminder to the customers. ADR has the second highest impact, highlighting the importance of having a good pricing strategy. Non-refundable deposits deter cancellation, stressing the importance of having clear policies and incentives for such bookings. Customer countries and special requests indicate higher customers commitment, which offers opportunities for personalized services. Meanwhile, seasonality and arrival dates focus on the need for tailored pricing and retention strategies. These features can help hotels to minimize cancellations, improve customer satisfaction and optimize revenue.

## **Actionable Recommendations**

Based on the business insight, there are several actions that are recommended for the hotels to reduce cancellations thus improving the revenue and optimizing resource allocation. The first strategy is to use dynamic pricing to increase revenue. One of the methods that can be used in this strategy is implementing seasonally adjusted pricing which helps to account for periods with a higher cancellation rate. For example, for off-peak months or season, hotels can offer discounts or incentives to reduce cancellation while maintaining stable pricing for high-demand months. This method also can be used for the customers with longer lead time where hotels can offer discounts to incentivize early booking. Besides, introducing personalized pricing for customers that make repeated bookings or show lower cancellation tendencies can help to increase the customers retention.

The next step that the hotels can take is promoting non-refundable policies by offering rewards such as discounts or complimentary services. On top of that, hotels should have clear communications with the customers about the benefits of non-refundable bookings during the reservation process to increase the adoption. This can create an awareness in the customers about the benefits thus helping them to understand better about the benefits.

Developing tailored marketing campaigns for guests from high-cancellations countries can help to reduce the cancellation rate. This can be done by providing region-specific discounts, promotions or incentives to encourage commitment. On top of that, hotels should be able to adjust message and policies to align with cultural preferences of different regions. One of the examples is to offer more flexible booking terms for countries which have higher cancellation rates. By doing this, customers will feel more incentivized to commit their stays in the hotels.

For booking with a tendency of having high risk of cancellation, hotels can provide a proactive retention measure. This can help these customers to have options rather than cancelling the bookings. Some of the measures that can be implemented such as flexible rebooking options, personalized communication and incentives like room upgrade. However, for customers with special requests, hotels can develop a priority engagement program for them, offering personalized attention. This is to ensure that the customers have higher satisfaction thus reducing cancellations.

Hotels can reduce the risk of cancellations by using dynamic pricing, offering discounts during off-peak season, tailored strategies by guest countries and rewarding loyal customers with discount rates. Promoting non-refundable bookings with incentives and clear communications can decrease the likelihood of cancellation. Proactive measures for customers with high-risk for cancellation and prioritizing customers with special request can enhance customers satisfactions. These strategies will help to optimize resources and boost revenue.

## **Limitations and Future Work**

While this study provides useful information for predicting hotel booking cancellation, it also has some limitations. One major limitation is the dependency of historical data in a specific period. For this dataset, it only covers the year 2015 to 2017 which may not capture the current booking trends or behaviors, particularly during the major disruptions like the COVID-19 pandemic.

The dataset also does not include external variables such as customer reviews, competitor pricing and macroeconomic factors which has the ability to improve accuracy of the predictions. As an illustration, customer reviews can provide insights into satisfaction and preferences whereas competitor pricing data aids in capturing market dynamics and competitive positioning. Additionally, macroeconomic factors like inflation rates influence customer purchasing power and behaviour which offers further context to refine predictions. Including these factors in future analysis could significantly enhance the robustness and reliability of the results.

For future work, expanding the variables of the datasets to include more recent and diverse data would help enhance the model’s generalizability. Incorporating additional features such as customer reviews or competitor analysis could provide an overall view of cancellation behavior. Future studies can also dive into more advanced machine learning techniques such as deep learning to improve prediction accuracy. These enhancements would allow hotels to have more precise strategies, ensuring better resource management, higher revenue and better customer satisfaction.

# References

Barbato, G., Barini, E. M., Genta, G., & Levi, R. (2011). Features and performance of some outlier detection methods. In *Journal of Applied Statistics* (Vol. 38, Issue 10). https://doi.org/10.1080/02664763.2010.545119

Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2020). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, *54*(3), 1937–1967. https://doi.org/10.1007/s10462-020-09896-5

Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, *54*(3). https://doi.org/10.1007/s10462-020-09896-5

Charbuty, B., & Abdulazeez, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. *Journal of Applied Science and Technology Trends*, *2*(01), 20–28. https://doi.org/10.38094/jastt20165

Clarin, A. (2022). Comparison of the performance of several regression algorithms in predicting the quality of white wine in WEKA. Int. J. Emerg. Technol. Adv. Eng., 12(7), 20-26.

Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2003). KNN Model-Based Approach in Classification. In *Lecture notes in computer science* (pp. 986–996). https://doi.org/10.1007/978-3-540-39964-3\_62

Herrera, A., Arroyo, Á., Jiménez, A., & Herrero, Á. (2024). Forecasting hotel cancellations through machine learning. *Expert Systems*. https://doi.org/10.1111/exsy.13608

Nakra, A., & Duhan, M. (2019). Comparative Analysis of Bayes Net Classifier, Naive Bayes Classifier and Combination of both Classifiers using WEKA. International Journal of Information Technology and https://doi.org/10.5815/ijitcs.2019.03.04

Othman, N. H., & Othman, N. (2019). A systematic review on entrepreneurship education in higher learning institutions in Southeast Asia. Universal Journal of Educational Research, 7(11). <https://doi.org/10.13189/ujer.2019.071118>

Patil, B. M., & Burkpalli, V. (2021). A perspective view of cotton leaf image classification using machine learning algorithms using WEKA. Advances in Human‐Computer Interaction, 2021(1), 9367778.

Priyanka, N., & Kumar, D. (2020). Decision tree classifier: a detailed survey. *International Journal of Information and Decision Sciences*, *12*(3), 246. https://doi.org/10.1504/ijids.2020.108141

Sinha, N. N. K. (2020). Developing A Web based System for Breast Cancer Prediction using XGboost Classifier. *International Journal of Engineering Research And*, *V9*(06). <https://doi.org/10.17577/ijertv9is060612>

Syriopoulos, P. K., Kalampalikis, N. G., Kotsiantis, S. B., & Vrahatis, M. N. (2023). kNN Classification: a review. *Annals of Mathematics and Artificial Intelligence*. https://doi.org/10.1007/s10472-023-09882-x

Tangirala, S. (2020). Evaluating the impact of GINI index and information gain on classification using decision tree classifier algorithm. *International Journal of Advanced Computer Science and Applications*, **11**(2), 612–619.

Trang, L. H., Huy, T. D., & Le, A. N. (2021). Clustering helps to improve price prediction in online booking systems. *International Journal of Web Information Systems*, *17*(1), 45–53. https://doi.org/10.1108/ijwis-11-2020-0065

Tribhuvan AP, Tribhuvan PP, & Gade JG. (2015). Advances in Computational Research. 7(1), 239–242. http://www.bioinfopublication.org/jouarchive.php?opt=&jouid=BPJ0000187

Upadhyay, D., Manero, J., Zaman, M., & Sampalli, S. (2020). Gradient Boosting Feature Selection With Machine Learning Classifiers for Intrusion Detection on Power Grids. *IEEE Transactions on Network and Service Management*, *18*(1), 1104–1116. https://doi.org/10.1109/tnsm.2020.3032618

Verma, G., & Verma, H. (2019). Predicting Bollywood Movies Success Using Machine Learning Technique. Proceedings - 2019 Amity International Conference on Artificial Intelligence, AICAI 2019. https://doi.org/10.1109/AICAI.2019.8701239

Vinutha, H. P., Poornima, B., & Sagar, B. M. (2018). Detection of outliers using interquartile range technique from intrusion dataset. *Advances in Intelligent Systems and Computing*, *701*. https://doi.org/10.1007/978-981-10-7563-6\_53

Zhang, S. (2021). Challenges in KNN Classification. *IEEE Transactions on Knowledge and Data Engineering*, *34*(10), 4663–4675. https://doi.org/10.1109/tkde.2021.3049250