Predicting hotel booking demand analysis and booking cancellations using machine learning with python

The ultimate guide for a hotel to maximize their hotel revenue and optimize business performance

Group members

1) Aden Rajput
2) Muhammad Arslan
3) Jacky Lohano
4) Hamza Afzal
5) Amna Farooq
6) Muhammad Mujtaba
7) Asma Farhan
8) Zartashia Afzal

Date: 9/9/2022

Table of Contents

1 Project Summary

1.1 Project Aim

In this project, we have built an end-end industry oriented and real-world data-driven machinelearning project in the hotel industry (hospitality) using AGILE CRISP-DM Methodology right from understanding the business problem to the model deployment of web application using Python and Stream_lit.

We have also applied advanced machine learning techniques like ensemble learning techniques and evaluation metrics to maximize revenue and optimize business performance like inventory management, dynamic pricing, distribution, and overbooking strategies in the hotel industry.

We have implemented a combination of both supervised and unsupervised machine learning algorithms to solve three business problems faced by the hotel industry.

Supervised Machine Learning

Classification: Predicting whether the hotel booking made by the guest/customer will be cancelled or not?

Regression: Predicting the cost/price of a hotel booking made by the hotelguest/customer.

Unsupervised Machine Learning

Clustering: Identifying profitable customers/guests by customer market segmentation.

1.2 Project Motivation

Our motivation for this project is to solve various business problems faced by the hotel industry by making a scalable and robust data science proof of concept (POC) product that can be accessed by the business user or a customer to generate business value.

Using machine learning techniques, we can predict the guests who are likely to cancel the reservation at the hotels and predict the price of hotel booking which is expected to increase or decrease in the future leading to booking now or waiting for a better offer/discount. Hence this will create a surplus revenue, better forecasts, and reduce uncertainty in business management decisions to both the hotels and hotel technology firms.

We wanted to document the entire machine learning workflow implemented in this project and apply that workflow in the hotel industry.

1.3 Project Skills and Tools

Project Skills	Project Tools
Analytical Skills	Python Programming
Business Understanding	Matplotlib
Data Visualization	Seaborn
Exploratory Data Analysis	Pandas
Machine Learning	NumPy
Problem Solving	Scikit-Learn
Project Documentation	Jupyter Notebook

2 Project Domain

2.1 Artificial Intelligence and Machine Learning

What is Artificial Intelligence?

Artificial Intelligence empowers computers to mimic human intelligence such as decision making, text processing, and visual perception. In simpler terms, Artificial Intelligence is Science that is concerned with building smart and intelligent machines.

AI is a broader field that contains several subfields such as Machine learning, Deep learning, Robotics, Natural Language Processing (NLP) and Computer Vision.

What is Machine Learning?

Machine Learning is a subfield of Artificial Intelligence that enables machines to improve at a given task with experience. In simpler terms, Machine Learning is a technique to implement AI that can learn from data, examples and experiences by themselves without being explicitlyprogrammed.

2.2 Artificial Intelligence and Machine Learning in Hotel Industry (Hospitality)

Artificial intelligence is playing an increasingly important role in Hotel Industry because of itsability to carry out traditionally human functions at any time of the day by saving significant money, eliminating human error, and delivering excellent customer service.

The use of Machine learning and Artificial intelligence (AI) is gradually embraced by the Hotel industry. During the past decade, the digital revolution that took place has completely transformed practices within the industry to adopt innovative strategies.

In the past, bookings and transactions were manually handled by the individual itself or by a travel agent. However, with AI technology, all this data automatically is now available, hoteliers should turn to Big Data, AI, and machine learning to take advantage of such information.

In particular, Customer Service is a vital part of the travel industry, with hotels often living and dying based on the way they treat their customers. With artificial intelligence, the possibilities for improving this aspect are almost endless, ranging from increased personalization to tailored recommendations.

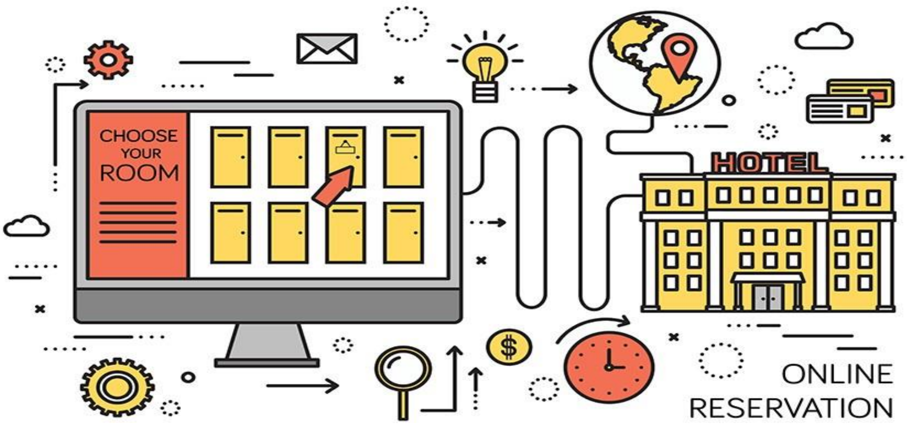
In recent times, Chatbots and Personal Assistants have all captured the imagination. But moving forward, hotels will find the real advantage of AI lies in pricing applications. For example, Apps such as Google Flights already offer “predictive pricing,” which provides consumers with suggestions on when they might find the lowest price.

The future success of hoteliers to attract customers and maintain their loyalty lies in Machine learning, Big Data, and AI. This ultimately frees up the resources of hoteliers and allows them to focus on offering the best possible services to their customers, without constantly worrying about setting the right prices and updating the demand forecasts [3].

2.3 Emerging Applications of AI and Machine Learning in Hotel Industry

Recommendation Engines: By collecting data and synthesizing information about the needs, preferences, and budget of each customer, recommendation engines build powerful algorithms, which they use to suggest deals that would make a good fit for each individual based on their previous choices. By taking advantage of such data, these engines can offer personalized and tailored products including car rental deals, new travel destinations, monuments and local attractions, and even propose alternative dates [3].

Enhanced Customer Experience: While AI in the hotel industry is mostly associated with the pre-arrival hotel reservation stage, the technology can also be used to boost the customer experience throughout a traveller’s stay [3]. Virtual voice assistants such as Amazon’s Alexa, as well as the assistant of Apple and Microsoft, are connected with motion sensors and room control, allowing customers to control devices in their rooms, as well as communicate with the reception of the hotel to order room service or report any issues [7].



Revenue Management Systems: The shift in the competitive landscape has led to hoteliers focusing on profitability per room rather than overall occupancy. Consequently, concentrating on profitable guests with higher cross-selling and upselling potential has become the main objective of the big hotel chains [3]. With thegrowing availability of data, it becomes impossible for hotel staff to make pricing decisions, making AI crucial when implementing dynamic pricing models.

Optimizing Demand Forecasting: The rise of machine learning allows hoteliers to forecast demand more accurately as they provide real-time analysis and automatically update forecasts according to changing external factors and in less time [7].

3 Machine Learning Project Workflow

Machine learning project comprises various steps required to build an end-end project. Sucha process or workflow of drawing insights from data is best described by CRISP-DM methodology.

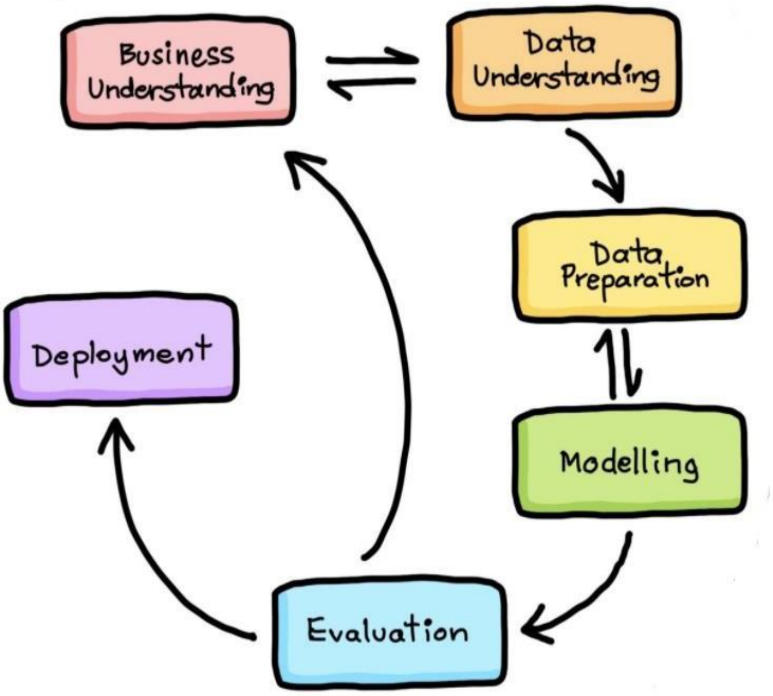
3.1 What is CRISP-DM Methodology?

The acronym CRISP-DM stands for Cross Industry Standard Process for Data Mining (CRISP- DM) is a process model with six phases that naturally describes the data science life cycle.

CRISP- DM was published in 1999 to standardize data mining processes across industries, it has since become the most common methodology for data mining, analytics, and data scienceprojects.

3.2 The life cycle of Machine Learning Project

1. Business Understanding
2. Data Collection and Understanding
3. Data Exploration
4. Data Preparation
5. Modelling
6. Model Evaluation
7. Model Deployment



1. Business Understanding: The Business Understanding phase focuses on understandingthe objectives and requirements of the project. Before deciding which dataset or algorithm we should use to solve a machine learning problem, it is very important to understand what the problem statement is.

This entails the understanding of a project's objectives and requirements from the business viewpoint.

Determine business objectives: we should first thoroughly understand, from a business perspective, what the customer wants to accomplish and then define business success criteria.

Assess the situation: Determine resources availability, project requirements, assess risks and contingencies, and conduct a cost-benefit analysis.

Determine data mining goals: In addition to defining the business objectives, we should also define what success looks like from a technical data mining perspective.

Produce project plan: Select technologies and tools and define detailed plans for each project phase.

2. **Data Collection and Understanding:** After understanding the business problem, the next step is to collect the most appropriate dataset to solve the problem. We can alsouse some free data sets which are present on the internet. Kaggle and UCI Machine learning Repository are the repositories that are used the most for making Machine learning models.

Kaggle is one of the most visited websites that is used for practising machine learning algorithms, they also host competitions in which people can participate and get to test their knowledge of machine learning.

3. **Data Exploration:** Data Exploration involves the use of data visualization tools andstatistical techniques to uncover data set characteristics and patterns. During data exploration, raw data is

reviewed and visually explored data sets, look forsimilarities, patterns and outliers and to identify the relationships between different variables. This process is called Exploratory Data Analysis (EDA).

This phase allows us to become familiar with the data and this involves performing exploratory data analysis. Such initial data exploration may allow us to figure out which subsets of data to use for further modelling as well as aid in the generation of hypotheses to explore.

Now the next step is to explore the data that we are using for solving the problem. Below are tasks to be performed

- Whether data contains any missing values?
- How to treat and handle the missing values?
- Check for Descriptive statistics
- Data visualisation of all the important features
- Check the correlation between features
- Understanding the relationship between the features and labels

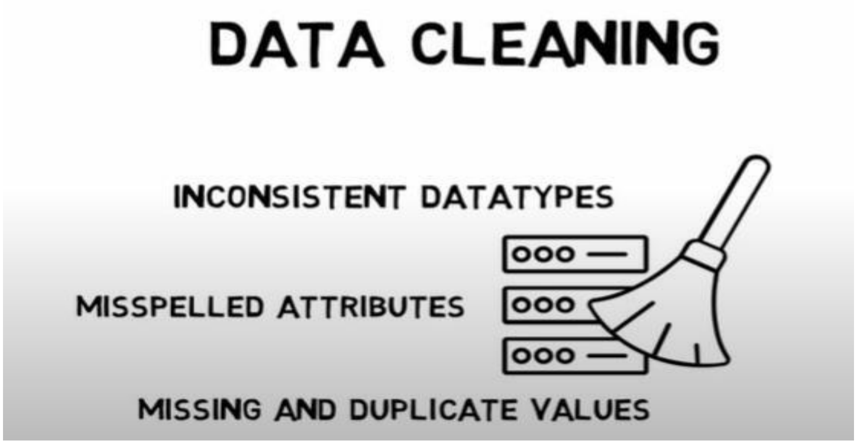
4. **Data Preparation:** This can be considered to be the most time-consuming phase of the data mining process as it involves rigorous data cleaning and pre-processing as well as the handling of missing data. In machine learning, there is an 80/20 rule. Everydata scientist should spend 80% time on data pre-processing and 20% time performingthe analysis.

   Data pre-processing is a process of cleaning the raw data i.e., the data is collected in the real world and is converted to a clean data set. Therefore, certain steps are executed to convert the data into a small clean data set, this part of the process is called data pre-processing. Most of the real-world data is messy, some of these typesof data are:

   Missing data: Missing data can be found when it is not continuously created or due to technical issues in the application (IoT system).

   Noisy data: This type of data is also called outliers and this can occur due to humanerrors (humans manually gathering the data) or some technical problem of the device at the time of collection of data.

Inconsistent data: This type of data might be collected due to human errors (mistakes with the name or values) or duplication of data.



The Data Preparation phase involves the below tasks:

Select data: Determine which data sets will be used and document reasons for inclusion/exclusion.

Clean data: Often this is the lengthiest task. Without it, you'll likely fall victim to garbage-in, garbage-out. A common practice during this task is to correct, impute, or remove erroneous values.

Construct data: Derive new attributes that will be helpful. For example, derive

someone's body mass index from height and weight fields.

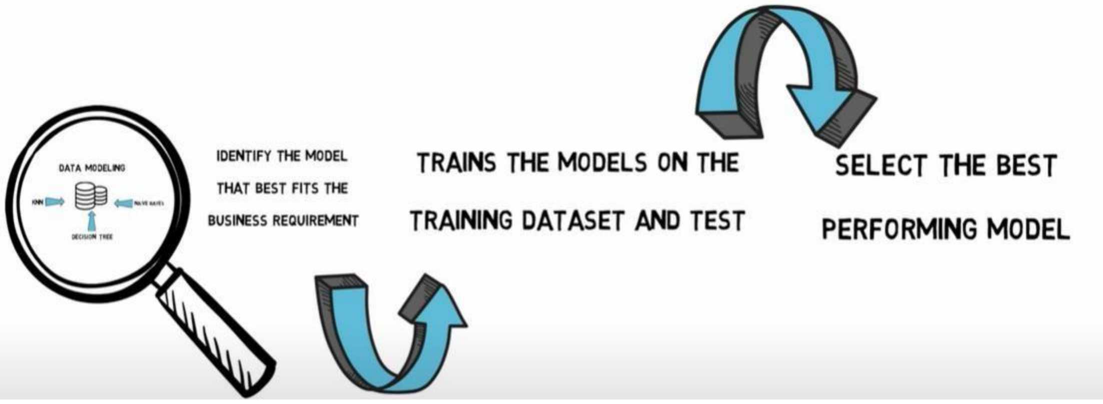
Integrate data: Create new data sets by combining data from multiple sources.

Format data: Re-format data as necessary. For example, you might convert stringvalues that store numbers to numeric values so that you can performmathematical operations.

5. **Modeling:** The next step now is to train a machine learning model using a machine learning algorithm. Here you need to divide the data into training and test sets first, and then train a model on the training set.

For better training of a machine learning model, it is necessary to divide the training data with more numbers (70% to 80%) of samples and the test set with 20% to 30% ofthe dataset depending on the size of the data.

We will build and assess various models based on several different modelling techniques. This


DATA MODELING
KNN
NAIVE BAYES
DECISION TREE
IDENTIFY THE MODEL THAT BEST FITS THE BUSINESS REQUIREMENT
TRAINS THE MODELS ON THE TRAINING DATASET AND TEST
SELECT THE BEST PERFORMING MODEL

phase has four tasks:

Select modelling techniques: Determine which algorithms to try (e.g., regression, classification, clustering).

Generate test design: Pending your modelling approach, you might need to split the data into training, test, and validation sets.

Build model: As glamorous as this might sound, this might just be executing a few lines of code using fit(X, y)

Assess model: Generally, multiple models are competing against each other, andthe data scientist needs to interpret the model results based on domain knowledge, the pre-defined success criteria, and the test design.

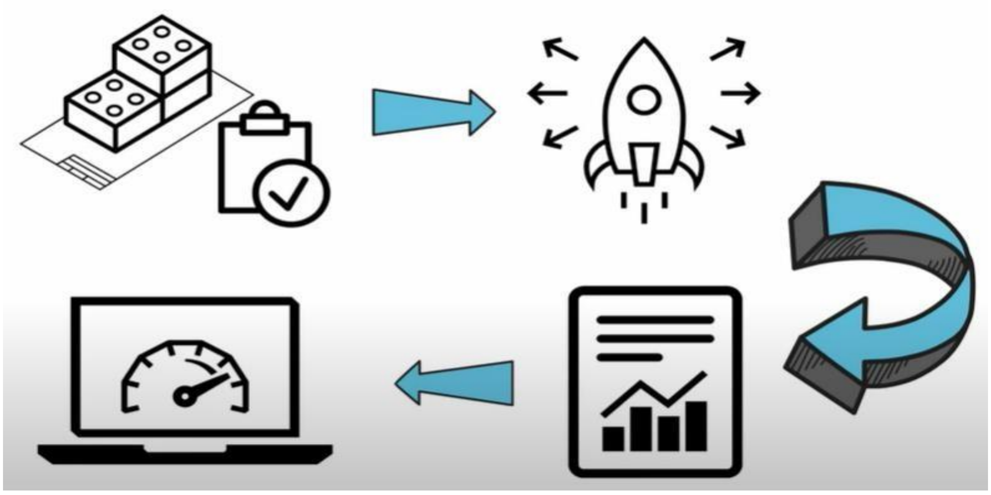
6. **Model Evaluation:** It is important to evaluate the accrued results and review the process performed to determine whether the originally set business objectives are met or not. If deemed appropriate, some steps may need to be performed again. Thisphase has three tasks:

Evaluate results: Do the models meet the business success criteria? Which one(s) should we approve for the business?

Review process: Review the work accomplished. Was anything overlooked? Were all steps properly executed? Summarize findings and correct anything if needed.

Determine next steps: Based on the previous three tasks, determine whether to proceed to deployment, iterate further, or initiate new projects.

7. **Model Deployment:** A model is not particularly useful unless the customer can access its results. This final phase has four tasks:



Plan deployment: Develop and document a plan for deploying the model.

Plan monitoring and maintenance: Develop a thorough monitoring and maintenance plan to avoid issues during the operational phase (or post-project phase) of a model.

Produce final report: The project team documents a summary of the project whichmight include a final presentation of data mining results.

Review project: Conduct a project retrospective about what went well, what couldhave been better, and how to improve in the future.

4 Project Data Set Description

4.1 Project Data Set Content

For this project, I have selected a data set containing hotel booking information that was uploaded to Kaggle, an online community of data scientists, by user Jesse Mostipak.

This dataset is available at Kaggle in the link: Hotel Booking Demand Kaggle Dataset

This data set contains data for a city hotel and a resort hotel in Portugal between the year 2015 and 2017 and includes information such as when the booking was made, length of stay, the number of adults, children, and/or babies, and the number of available parking spaces, among other things. All personally identifying information has been removed from the data.

4.2 Acknowledgements

The data is created and documented by Nuno Antonio, Ana Almeida, and Luis Nunes for the article "**Hotel booking demand datasets**" published in Data in Brief (Volume 22, February 2019).

4.3 Project Data Set Summary

- The data set contains **119,390 Observations** (Rows) representing distinct hotel bookings at one of two hotels in Portugal between the year 2015 and 2017
- The data set contains **32 Attributes** (Columns) corresponding to various details associated with each booking
- One of the hotels is a Resort hotel while the other is a city hotel.
- Numerical Data Type Attributes are 20 and Categorical Data Type Attributes are 12
- 37% of the observations represent cancelled bookings

4.4 Project Data Set Description

Variable	Description
Hotel	Resort Hotel or City Hotel
is_canceled	If the booking was cancelled (1) or not (0)
lead_time	Number of days that elapsed between the entering date of the booking into the PMS and the arrival date
arrival_date_year	Year of arrival date of the guest
arrival_date_month	The month of arrival date of the guest
arrival_date_week_number	Week number of year for the arrival date of the guest
arrival_date_day_of_month	Day of arrival date of the guest
stays_in_weekend_nights	Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel
stays_in_week_nights	Number of weeknights (Monday to Friday) the guest stayed or booked to stay at the hotel
adults	Number of adults
children	Number of children
babies	Number of babies
meal	Type of meal booked. Categories are presented in standard hospitality meal packages are Undefined/SC no meal package, BB – Bed & Breakfast, HB – Halfboard

country	Country of origin. Categories are represented in the ISO 3155–3:2013 format
market segment	Market segment designation type
distribution_channel	Booking distribution channel
is_repeated_guest	A value indicating if the booking name was from a repeated guest (1) or not (0)
previous cancellations	Number of previous bookings that were cancelled by the customer before the current booking
previous_bookings_not_canceled	Number of previous bookings not cancelled by the customer before the current booking
reserved_room_type	Code of room type reserved. Code is presented instead of designation for anonymity reasons
assigned_room_type	Code for the type of room assigned to the booking. Sometimes the assigned room type differs from the reserved room type due to hotel operations
booking changes	Number of changes/amendments made to the booking from the moment the booking was entered on the PMS until the moment of check-in or cancellation
deposit type	Indication on if the customer deposited to guarantee the booking. This variable can assume three categories are No Deposit, Non-Refund, Refundable
agent	The ID of the travel agency that made the booking
company	The ID of the company/entity that made the booking or is responsible for paying the booking. ID is presented instead of designation for anonymityreasons
days_in_waiting_list	Number of days the booking was on the waiting list before it was confirmed to the customer
customer type	Type of booking, assuming one of four categories: Contract, Group, Transient, Transient-party
ADR	Average Daily Rate as defined by dividing the sum of all lodging transactions by the total number of stayingnights
required_car_parking_spaces	Number of car parking spaces required by the customer
total_of_special_requests	Number of special requests made by the customer (e.g., twin bed or high floor)
reservation_status	Reservation the last status, assuming one of three categories are Cancelled, Check-out, No-Show
reservation_status_date	The date at which the last status was set.

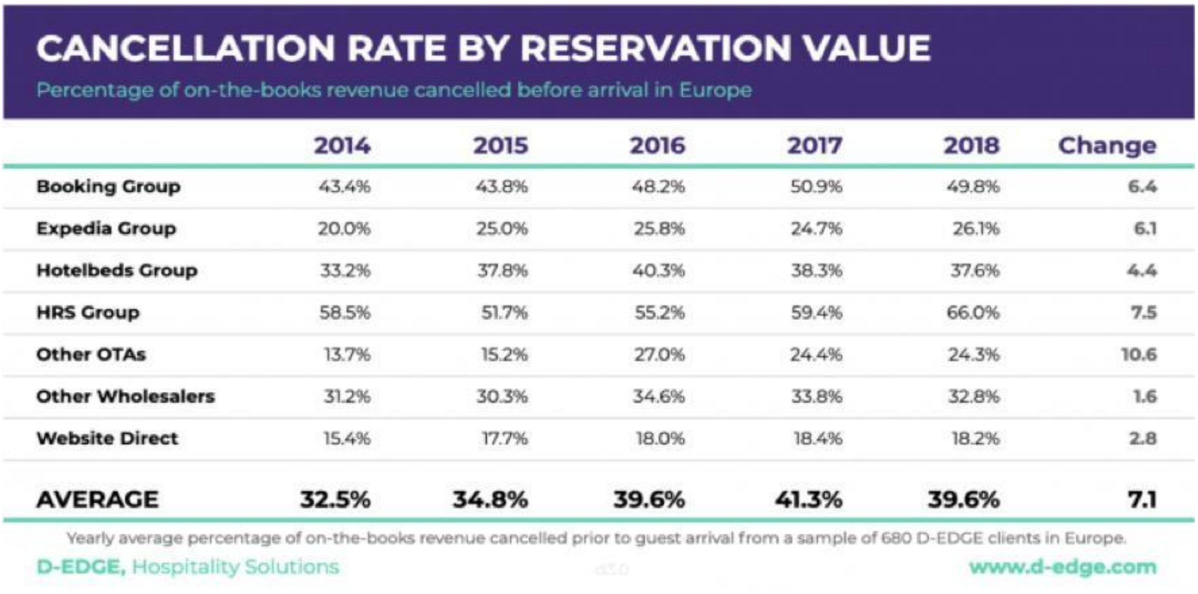
5 Implementation of Machine Learning in the Hotel Industry

5.1 Understanding the Business Problem

Booking cancellations are undoubtedly one of the biggest challenges of any Revenue manager or Hotel manager nowadays. Booking cancellations have a substantial impact on demand management decisions in the hospitality industry.

Every year, more than 140 million bookings were made on the internet and many hotel bookings were booked through top-visited travel websites like Booking.com, Expedia.com, Hotels.com, Airbnb etc.

According to D-Edge Hospitality solutions has found that the global average cancellation rate on bookings has reached almost 40% and this trend produces a very negative impact on hotel revenue. If we look into the data available of the European market, we see that in 2018 49.8%of bookings reserved on the OTA Booking.com were cancelled.



CANCELLATION RATE BY RESERVATION VALUE

Percentage of on-the-books revenue cancelled before arrival in Europe

	2014	2015	2016	2017	2018	Change
Booking Group	43.4%	43.8%	48.2%	50.9%	49.8%	6.4
Expedia Group	20.0%	25.0%	25.8%	24.7%	26.1%	6.1
Hotelbeds Group	33.2%	37.8%	40.3%	38.3%	37.6%	4.4
HRS Group	58.5%	51.7%	55.2%	59.4%	66.0%	7.5
Other OTAs	13.7%	15.2%	27.0%	24.4%	24.3%	10.6
Other Wholesalers	31.2%	30.3%	34.6%	33.8%	32.8%	1.6
Website Direct	15.4%	17.7%	18.0%	18.4%	18.2%	2.8
AVERAGE	32.5%	34.8%	39.6%	41.3%	39.6%	7.1

Yearly average percentage of on-the-books revenue cancelled prior to guest arrival from a sample of 680 D-EDGE clients in Europe.

D-EDGE, Hospitality Solutions www.d-edge.com

5.1.1 Hotel Booking Cancellations, A Growing Problem…

Online Travel Agents (OTA´s) are encouraging customers to cancel when they actively encourage you to book now and cancel later, free of charge, whenever you want. This results in customers booking more than one hotel and deciding later which one they will choose.

Examples of the impact of cancellations on a hotel.

- Loss of revenue when they cannot resell the room
- Additional costs of distribution channels by increasing commissions or paying for publicityto help sell these rooms
- Lowering prices last minute, so they can resell a room, resulting in reducing profit margin. Once the reservation has been cancelled, there is almost nothing to be done and it creates discomfort for many Hotels and Hotel Technology companies.

5.1.2 What can Hotels do to maximize their Revenue and Optimize business performance?

To overcome the problems caused by booking cancellations, hotels implement rigid cancellation policies, inventory management, and overbooking strategies, which can also have a negative influence on revenue and reputation. Therefore, it would seem that we havea complex problem and not a viable solution.

However, thanks to Machine learning so that we can accurately predict which guests and specific reservations are going to cancel, identify profitable guests by customer market segmentation and predict the Average daily rate (ADR) of hotel booking.

This will create a surplus revenue, better forecasts and reduce uncertainty in business management decisions for both Hotels and Hotel Technology companies.

5.1.3 Business Problem Specification

In this project, I have implemented a combination of both Supervised Machine learning algorithms and Unsupervised learning Algorithms to solve three business problems faced by the Hotel Industry.

Supervised Machine Learning

Regression: Predicting the Cost/Price of a Hotel Booking made by the Hotel Guest/Customer

Classification: Predicting whether the Hotel Booking made by the Guest/Customer will be Cancelled or not?

Unsupervised Machine Learning

Clustering: Identifying Profitable Customers/Guests by Customer Market Segmentation

5.2 Data Collection

For this project, I have selected a data set containing hotel booking information that was uploaded to Kaggle, an online community of data scientists, by user Jesse Mostipak.

The data set was created and documented by Nuno Antonio, Ana Almeida, and Luis Nunes for the article "Hotel booking demand datasets" published in Data in Brief (Volume 22, February 2019).

Kaggle is one of the most visited websites that is used for practising machine learning algorithms, they also host competitions in which people can participate and get to test their knowledge of machine learning.

This dataset is available at Kaggle in the link: Hotel Booking Demand Kaggle Dataset

5.3 Data Exploration

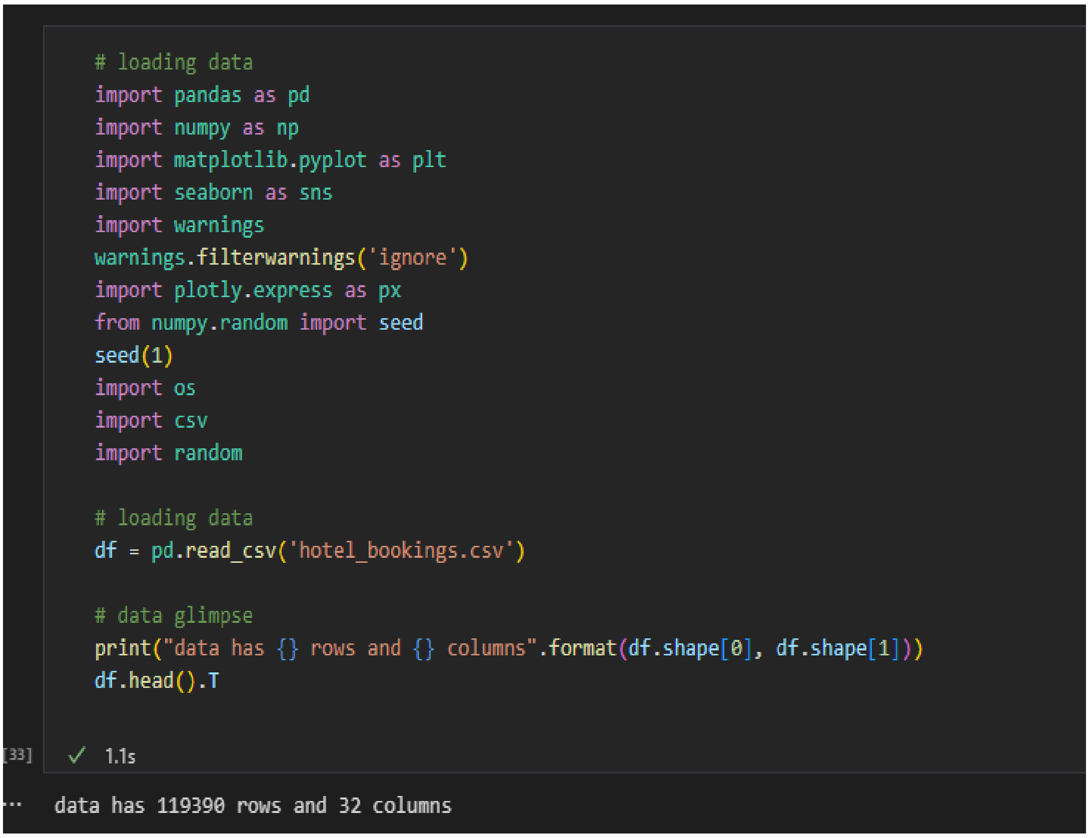
Data Exploration involves the use of data visualization tools and statistical techniques to uncover data set characteristics and patterns.

During data exploration, raw data is reviewed and visually explored data sets, look for similarities, patterns and outliers and to identify the relationships between different variables. This process is called Exploratory Data Analysis (EDA).

Steps to Perform in Data Exploration

I have applied Exploratory Data Analysis (EDA) techniques to extract insights from the data set to know which features have contributed more in predicting target attributes by performing Data Analysis using **Pandas** and **NumPy** and Data visualization using **Matplotlib, Seaborn** and **Plotly**.

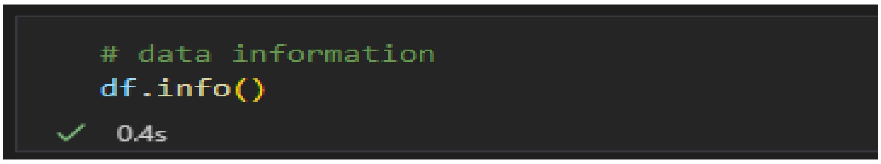
5.3.1 Understanding the Dataset and Shape (Rows & Columns)



```
# loading data
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
import plotly.express as px
from numpy.random import seed
seed(1)
import os
import csv
import random

# loading data
df = pd.read_csv('hotel_bookings.csv')

# data glimpse
print("data has {} rows and {} columns".format(df.shape[0], df.shape[1]))
df.head().T
```

```
[33]  ✓ 1.1s

data has 119390 rows and 32 columns
```

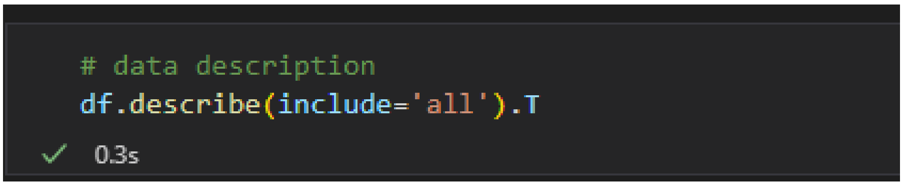
5.3.2 Checking Data Types of the Attributes



```
# data information
df.info()
```

✓ 0.4s

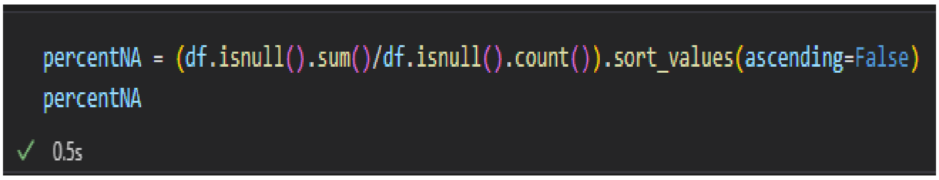
5.3.3 Checking Statistical Summary of the dataset - Descriptive Statistics



```
# data description
df.describe(include='all').T
```

✓ 0.3s

5.3.4 Checking Missing Values (Nan) in Dataset

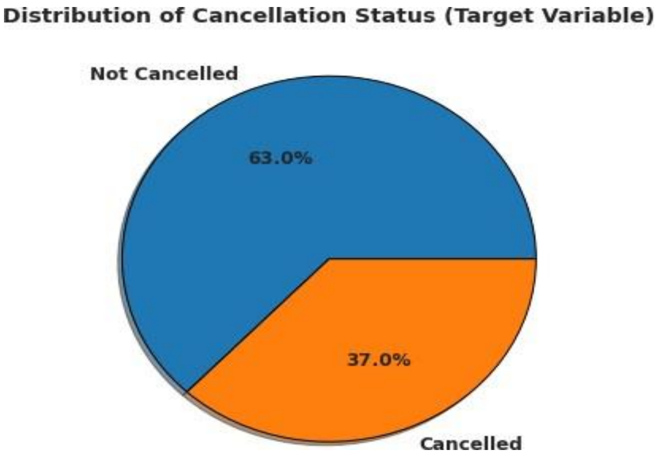


```
percentNA = (df.isnull().sum()/df.isnull().count()).sort_values(ascending=False)
percentNA
```

✓ 0.5s

5.3.5 Checking Distribution of Target Attribute

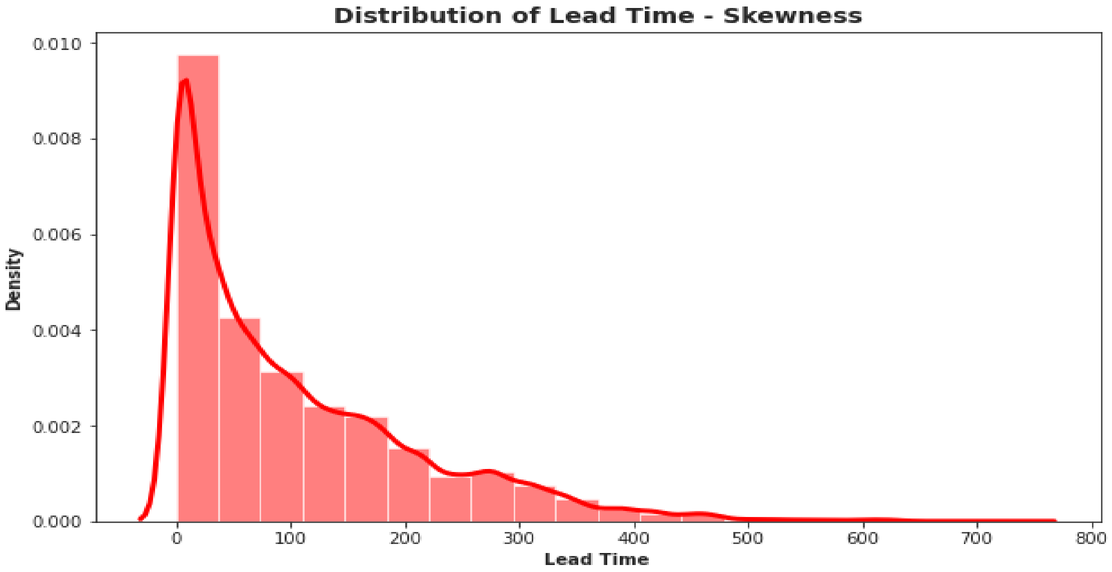
From the pie chart, we can depict from the target attribute (is_canceled) that **63%** of bookings were not cancelled and **37%** of the bookings were cancelled at the Hotel.

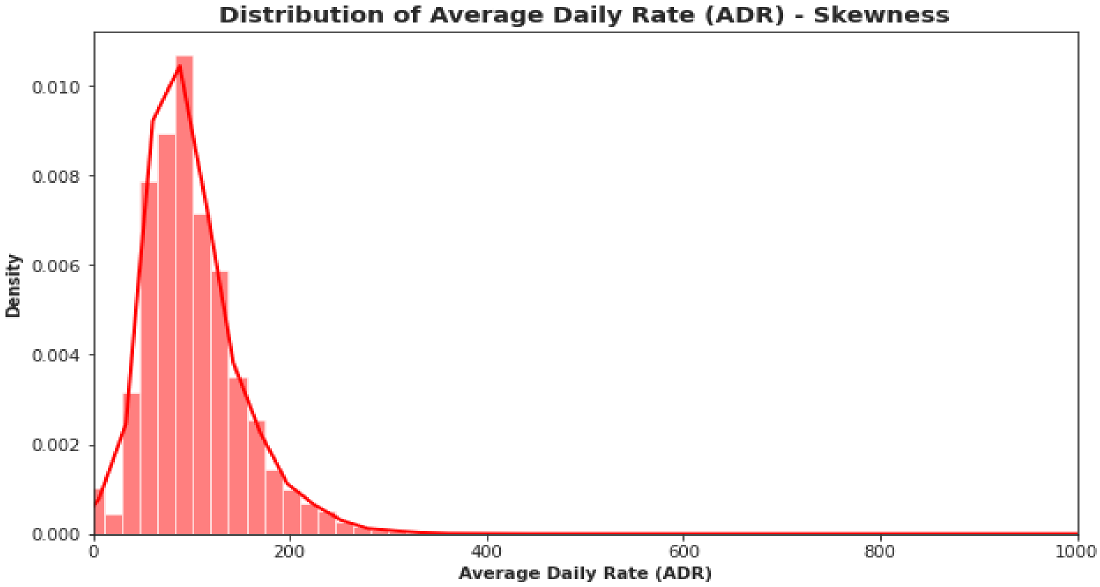


5.3.6 Checking Skewness of Attributes - Inferential Statistics

- Many numerical variables naturally follow a normal distribution, also known as a Gaussian distribution or Normal Distribution.
- The **skewness** of a distribution is defined as the lack of symmetry and tells us about the distribution of our data. In a symmetrical distribution, the Mean, Median and Mode are equal. The normal distribution has a skewness of 0.
- From both the graphs, we can depict that distribution of numerical features "**Lead time"** and

"Average Daily Rate" is heavily right-skewed and should be handled in Data Preparation by handling skewness and making it normally distributed



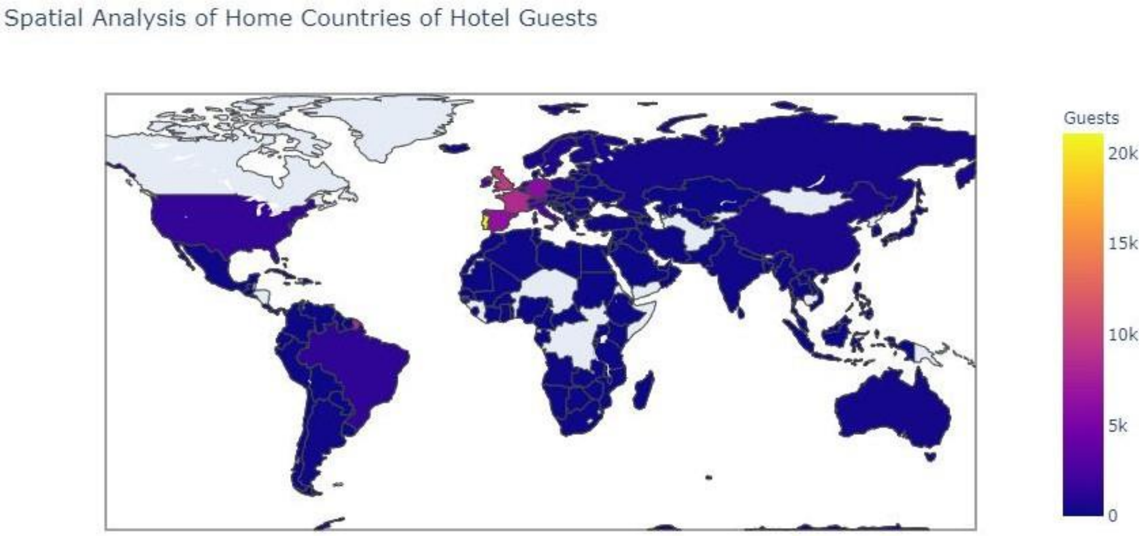


5.3.7 Data Visualization of Important Attributes

1. Geo-Spatial Analysis of the Hotel Guests By their Home Countries

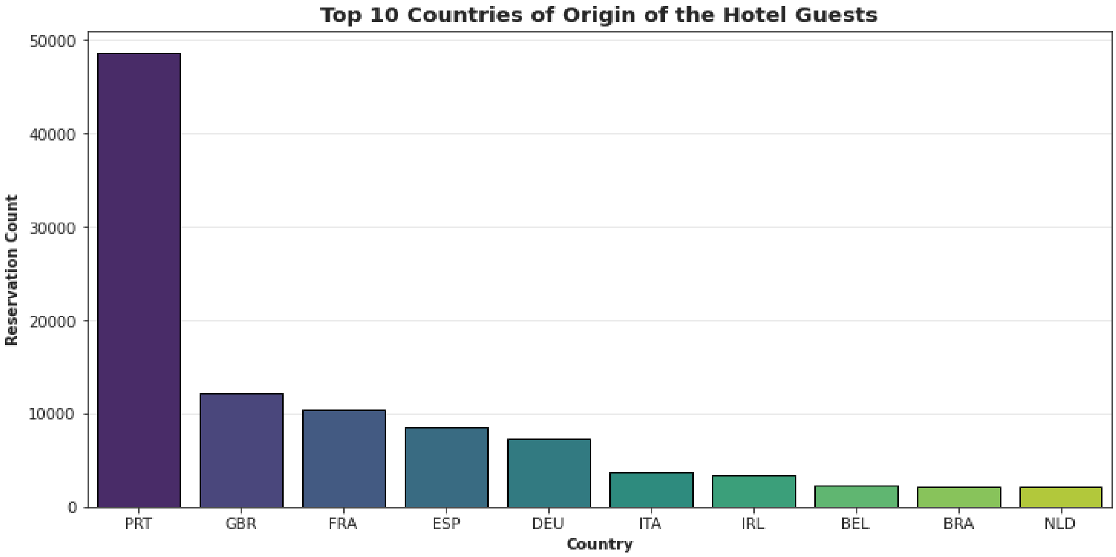
- From the Geo-Spatial Analysis, we can depict the information about the number ofguests who made the booking at hotels from various countries in the world.

- The highest number of guests were arrived from Portugal with around **21.071k** guests.


Spatial Analysis of Home Countries of Hotel Guests

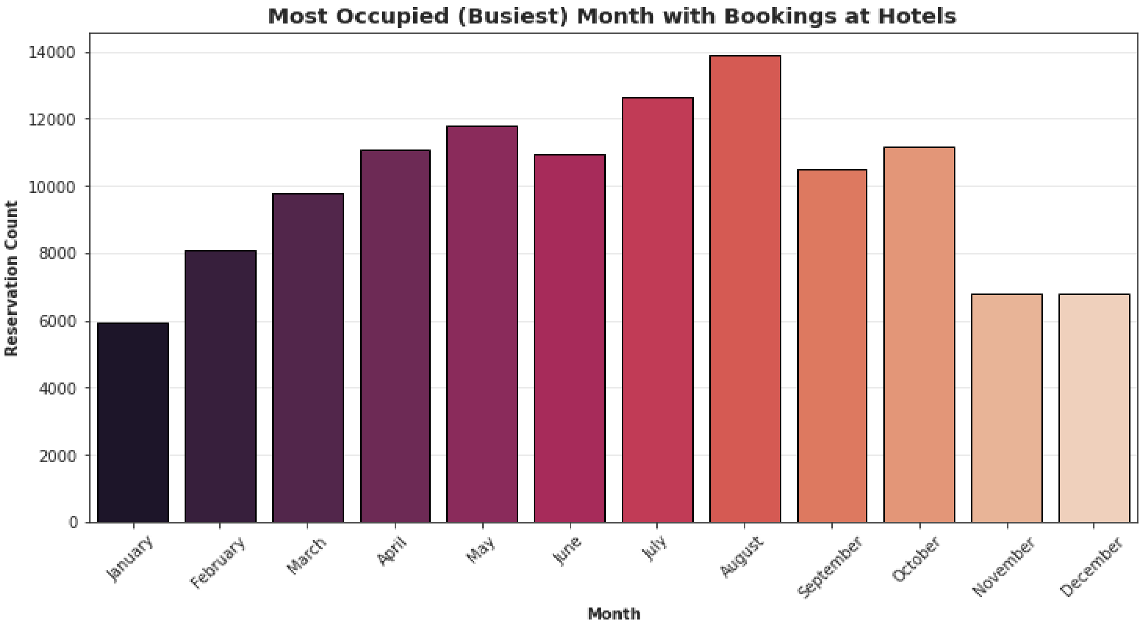
2. Analyzing Top 10 Countries of Origin of Hotel Guests

- About **40%** of all bookings are created from **Portugal** followed by **Great Britain 10%** and **France 8%**


Top 10 Countries of Origin of the Hotel Guests

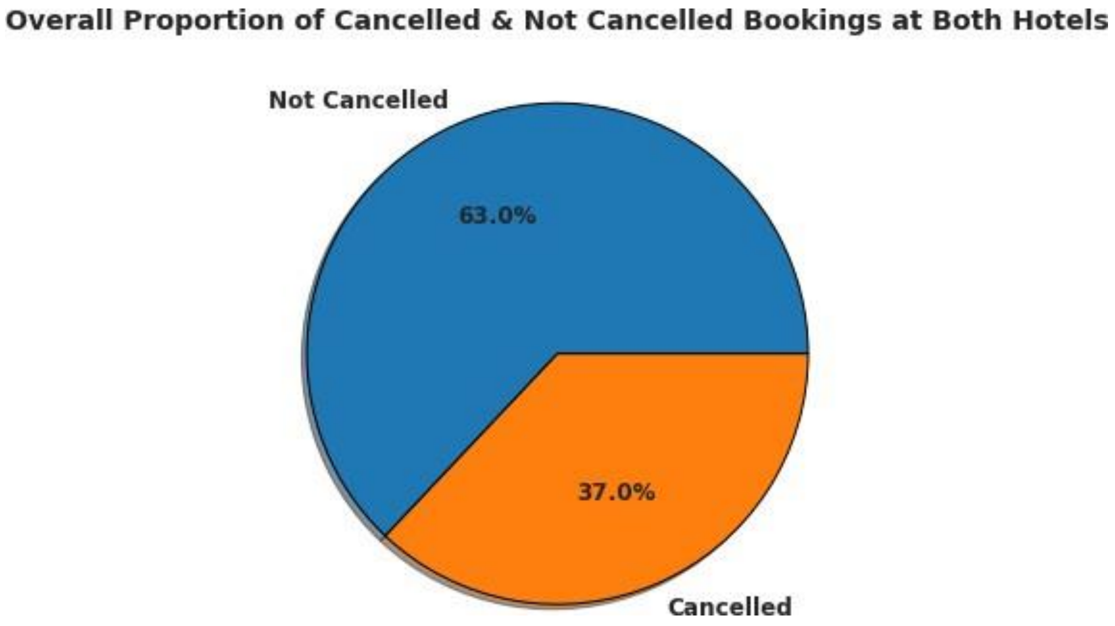
3. Analyzing Most Occupied (Busiest) Month with Bookings at the Hotel

- **August** is the most **occupied** (busiest) month with **11.62%** bookings and **January** is the most **unoccupied** month with **4.96%** bookings.



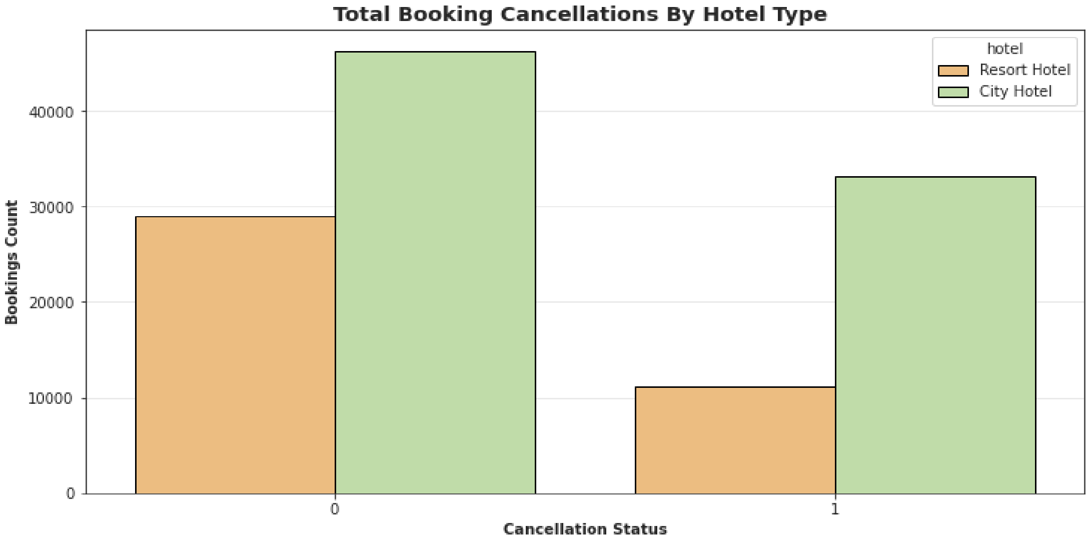
4. Analyzing Overall Bookings Cancelled at Both Hotels

- According to the pie chart, **63%** of bookings were **not cancelled** and **37%** of thebookings were **cancelled** at the Hotel.



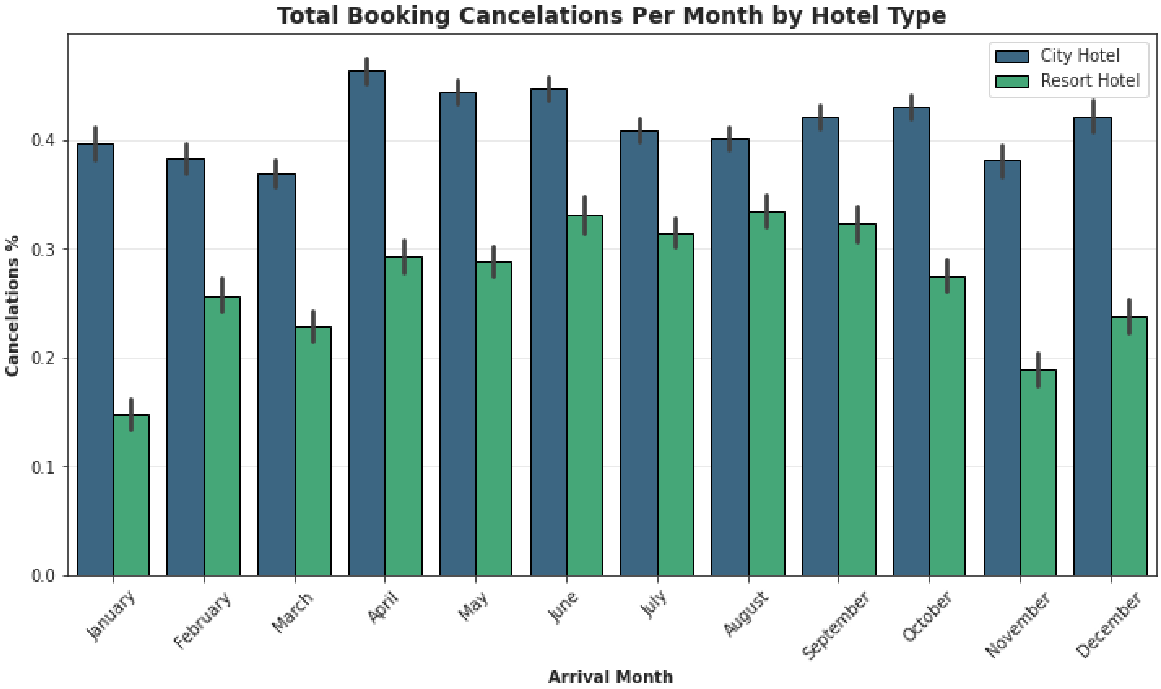
5. Analyzing How many Bookings were Cancelled by Hotel Type

- For the **Resort Hotel**, a total of **25.14%** of Bookings was cancelled and for **City Hotel** total of **74.85%** Bookings were cancelled



6. Analyzing Which Month has Highest Number of Cancellations by Hotel Type

- For the **City hotel**, the number of cancellations for every month is around **40 %**throughout the year.
- For the **Resort hotel**, the cancellations are highest in the **summer** (**June, July, August**) and lowest during the **winter** (**November, December, January**). In short, the possibility of cancellation for resort hotels in winter is very low.

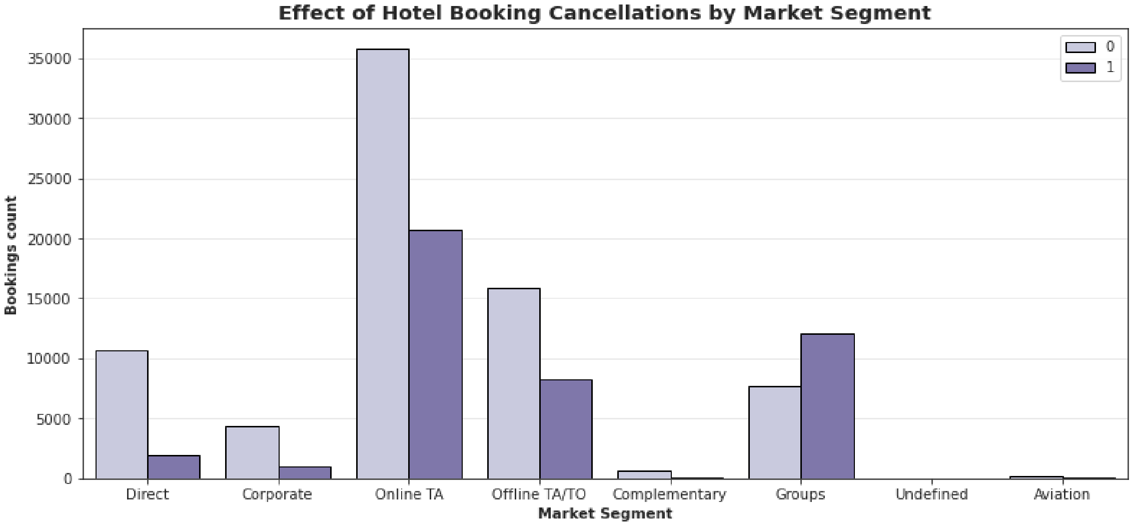


7. Analyzing Effect of Booking Cancellations by Market Segment

From the graph, we can depict the below statistics for cancellations by MarketSegment

- 21% of bookings by Aviation, 13% of bookings by Complementary, 18.7% ofbookings by Corporate, 15.3% of bookings by Direct

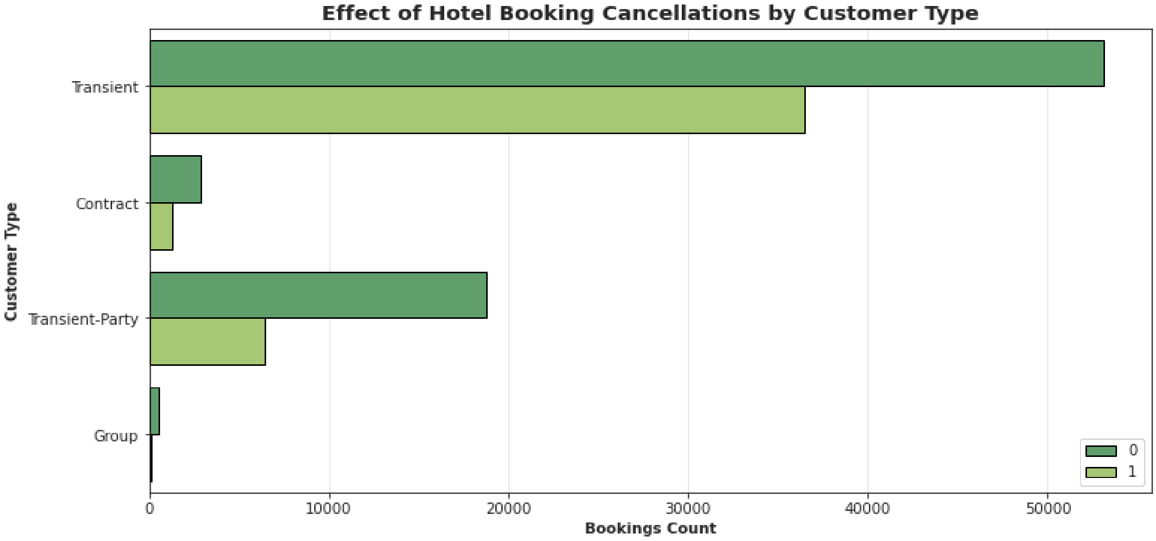
- **38.9%** of bookings by **Groups**, 34.3% of bookings by Offline TA/TO, **36.7%** ofbookings by **Online TA/TO**



8. Analyzing Effect of Hotel Booking Cancellations by Customer Type

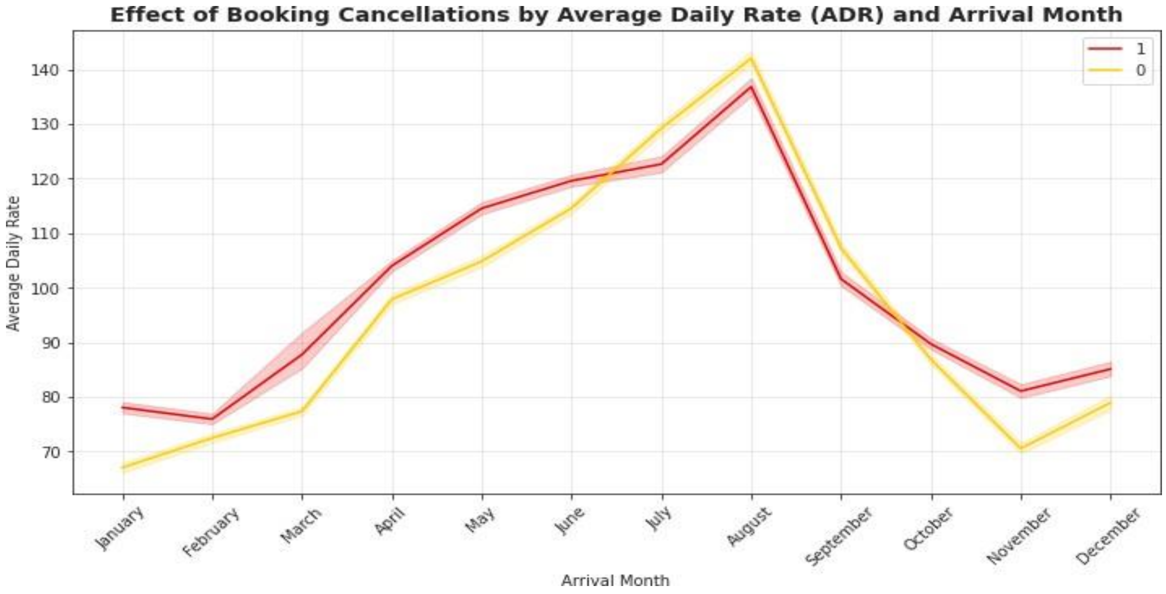
From the graph, we can depict the below statistics for cancellations by Customer type

- 30.9% of bookings by Contract, 10.2% by Group, **40.7% by Transient**, 25.4% by Transient-Party



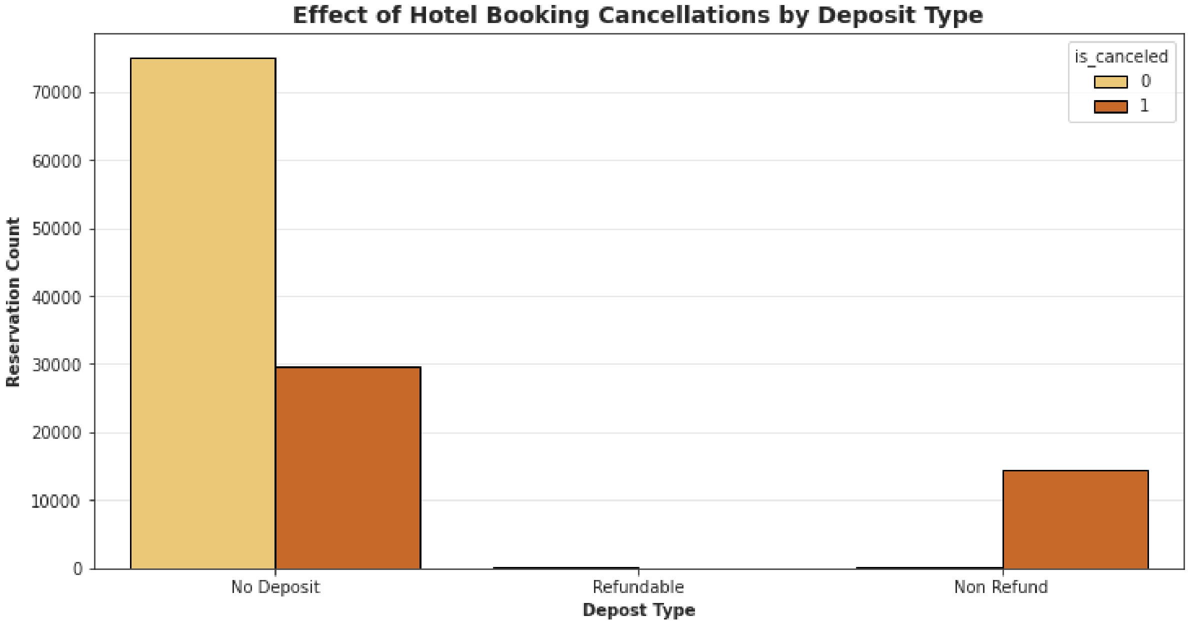
9. Analysing Effect of Hotel Booking Cancellations by Average Daily Rate (ADR) and Arrival Month

- The Average Daily Rate (ADR) is **more expensive** during **July, August & September**. Hence, it could be one of the reasons for more booking cancellations from the guest.



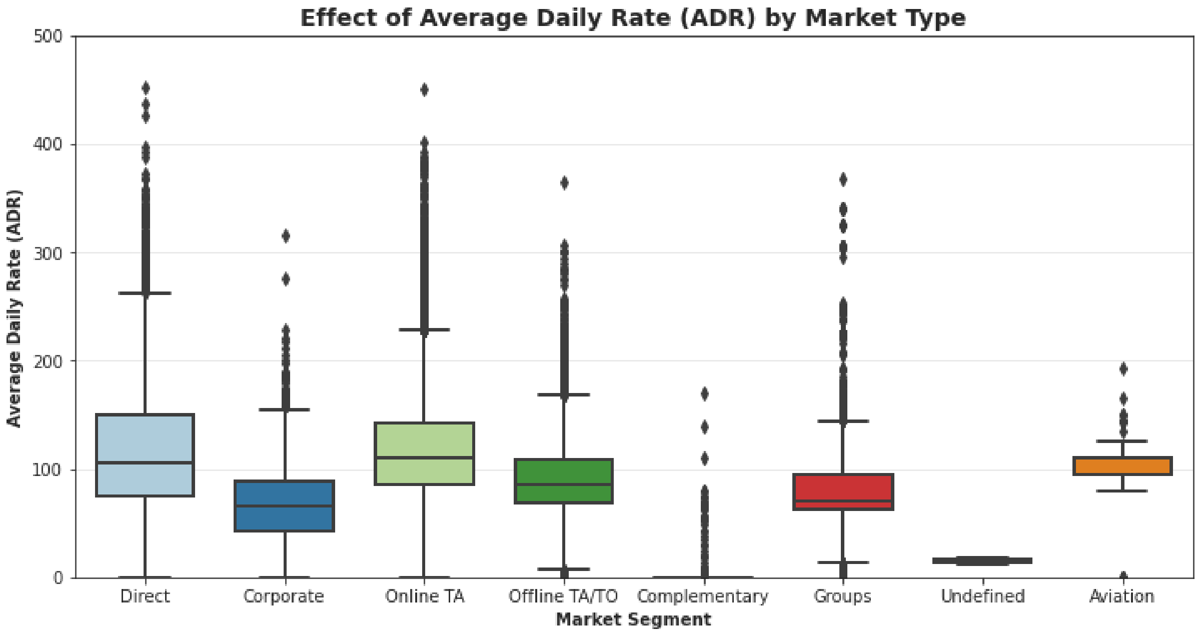
10. Analysing Effect of Hotel Booking Cancellations by Deposit Type

- Around **28% of bookings were cancelled by guests with no deposit**. Hence it is obvious that guests who do not pay any deposit while booking are likely to cancel more reservations.



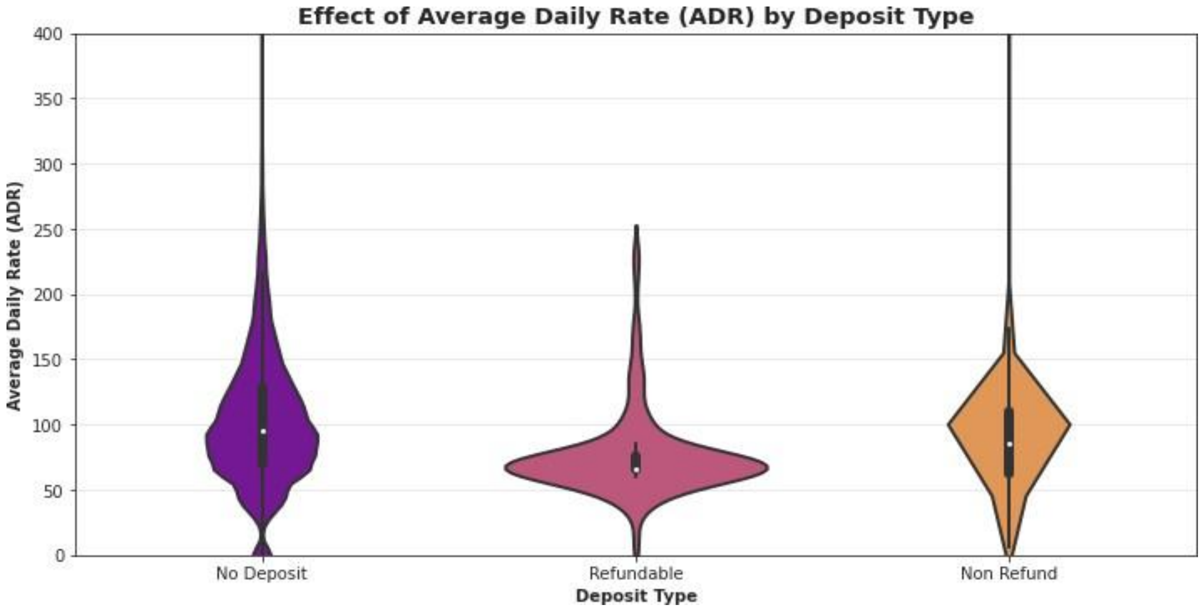
11. Analysing Effect of Average Daily Rate (ADR) by Market Type

- From the graph, we can depict that the Average Daily Rate (ADR) is more with **Direct** and **Online Travel Agents** bookings when compared with other marketsegments.



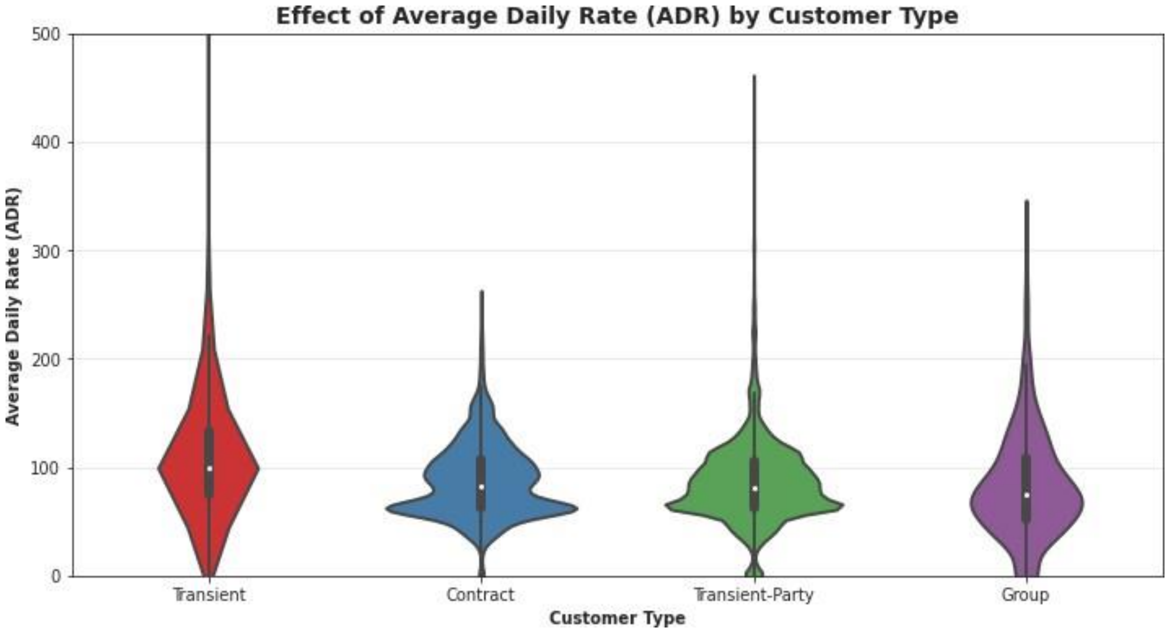
12. Analyzing Effect of Average Daily Rate (ADR) by Deposit Type

- From the graph, we can depict that the Average Daily Rate (ADR) is more with **No Deposit** bookings however, guests who do not pay any deposit while booking are likely to cancel more reservations



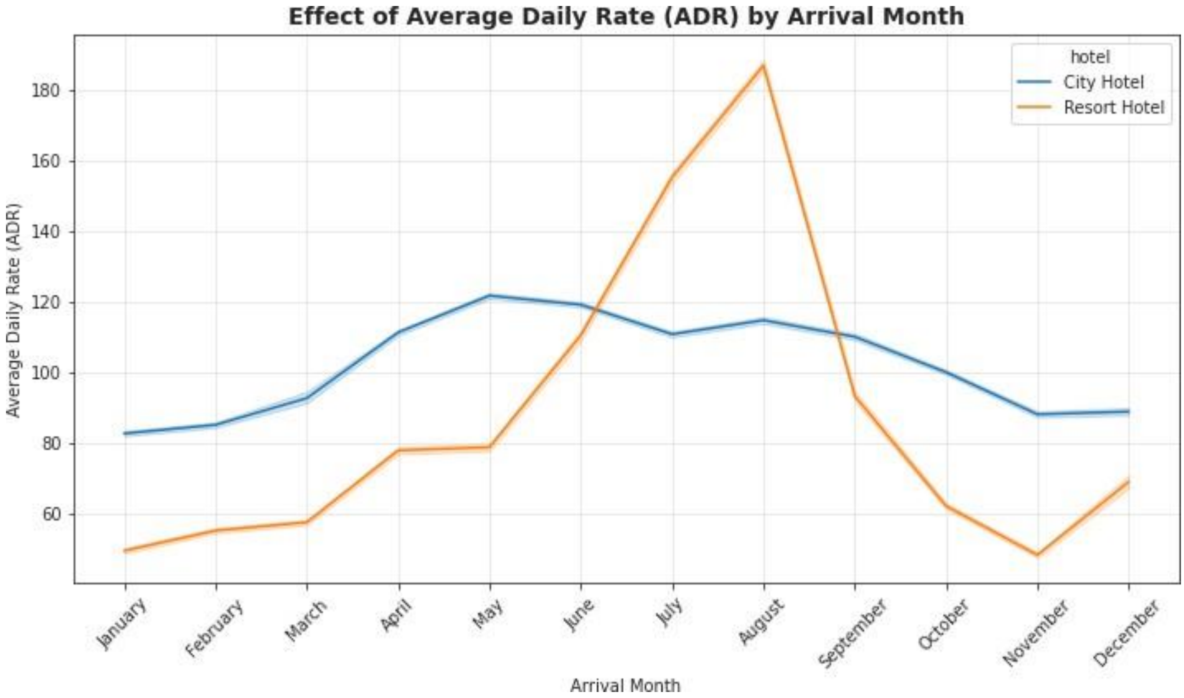
13. Analysing Effect of Average Daily Rate (ADR) by Customer Type

- From the graph, we can depict that the Average Daily Rate (ADR) is more with **Transient Bookings** because **75% of bookings** were actual Transient Bookings



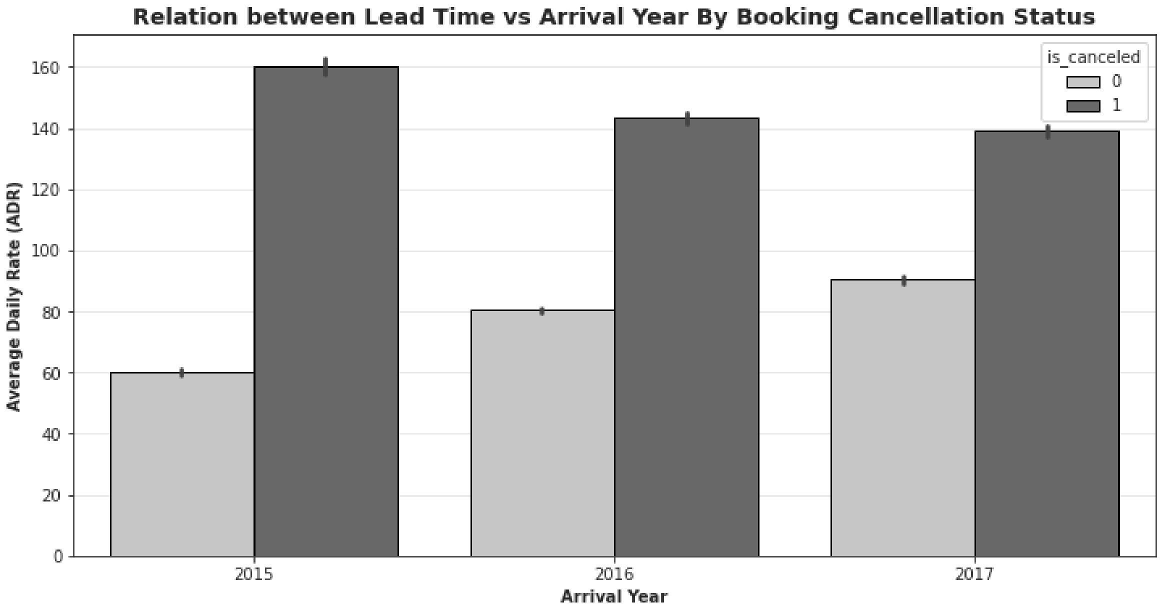
14. Analysing Effect of Average Daily Rate (ADR) by Arrival Month

- For Resort Hotel, ADR is more expensive during **July, August & September**
- For City Hotel, ADR is slightly more during **March, April & May**



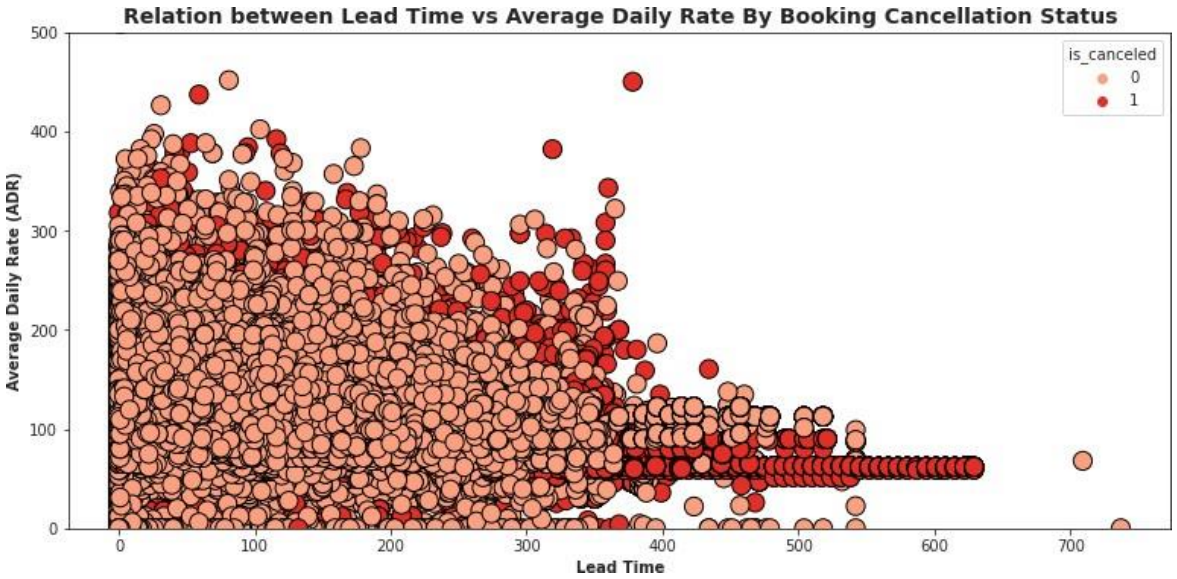
15. Analysing Relation between Lead Time vs Arrival Year by Booking Cancellation Status

- For all the 3 years, bookings with a lead time of fewer than 100 days have fewer chances of getting cancelled and with a lead time of more than 100 days have more chances of getting cancelled



16. **Analysing Relation between Lead Time vs Average Daily Rate by Booking Cancellation Status**

- From the Scatter plot, we can depict that the guests with the lowest lead time and the highest ADR are deemed to be the most profitable**.**



6 Statistical Test

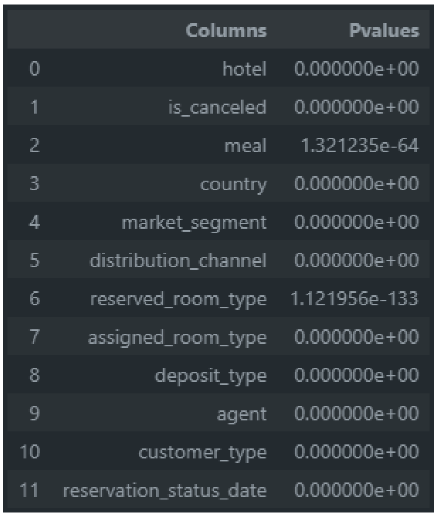
6.1 Chi square test

Categorical columns – For categorical columns we performed chi-square test to check for the significance of the categorical column with respect to is_canceled(target variable) column.

Hypothesis of Chi-square test

H0: Attributes are independent

H1: Attributes are dependent



	Columns	Pvalues
0	hotel	0.000000e+00
1	is_canceled	0.000000e+00
2	meal	1.321235e-64
3	country	0.000000e+00
4	market_segment	0.000000e+00
5	distribution_channel	0.000000e+00
6	reserved_room_type	1.121956e-133
7	assigned_room_type	0.000000e+00
8	deposit_type	0.000000e+00
9	agent	0.000000e+00
10	customer_type	0.000000e+00
11	reservation_status_date	0.000000e+00

We observe that the p_values of all the columns are less than 0.05. Hence, we reject the null hypothesis. Therefore, we conclude that all the categorical features are significant.

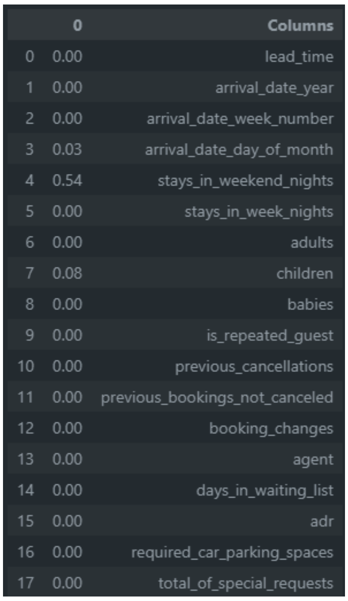
6.2 ANOVA test

Numerical columns – We performed parametric test for the numerical columns using ANOVA test.

Hypothesis for numerical tests

H0: Two samples have the same mean (i.e insignificant)

H1: Two samples have different mean (i.e significant)



	0	Columns
0	0.00	lead_time
1	0.00	arrival_date_year
2	0.00	arrival_date_week_number
3	0.03	arrival_date_day_of_month
4	0.54	stays_in_weekend_nights
5	0.00	stays_in_week_nights
6	0.00	adults
7	0.08	children
8	0.00	babies
9	0.00	is_repeated_guest
10	0.00	previous_cancellations
11	0.00	previous_bookings_not_canceled
12	0.00	booking_changes
13	0.00	agent
14	0.00	days_in_waiting_list
15	0.00	adr
16	0.00	required_car_parking_spaces
17	0.00	total_of_special_requests

We observe that the p_values for some variables is greater than 0.05. Hence these variables are insignificant.

7 Data Preparation

7.1 Data Cleaning

Data Cleaning is a very important step in the Machine Learning project pipeline. Knowing howto clean your data is advantageous for many reasons.

- Data Cleaning makes the ETL process run faster.
- Properly cleaned and formatted data speed up computation in advanced algorithms
- It prevents you from wasting time on wobbly or even faulty analysis
- It prevents you from making the wrong conclusions, which affect business performance

7.2 Handling Missing Values in the dataset

To further work with the data set, I have examined the dataset to see if any data cleaning actions needed to be performed. Four attributes that contain missing values in the dataset

- agent
- company
- children
- country

According to the documentation for the data set, the "agent" attribute represents the hotel booking agent used to make the booking and the "company" attribute represents the company that paid for the booking.

- **Company** feature has almost 94% missing values. Therefore, we do not have enough values to fill the rows or impute the company column by mean, median etc. Hence, I will drop the "Company" feature.
- **Agent** feature has 13.69% missing values. "Agent" feature is a travel agency Id and these values are unique and we cannot impute Id by mean, median or mode. Since missing values are 13% of all data, we can't drop them. Therefore, missing data for "Agent" can be filled by 0.
- **Children** feature represents the number of children included in the booking and the Children feature has only 4 missing values and I will fill these missing values by 0 considering guests have no children.
- **Country** feature represents the country in which the booking originated and the Country feature has 0.4% missing values. Since missing data of Country is less than 1%, I will impute with most frequent value (Mode).

7.3 Supervised Machine Learning - Classification Problem

6.3.1 Classification Problem Statement

Predicting whether the Hotel Booking made by the Guest/Customer will be Cancelled or not?

6.3.2 Feature Engineering

Feature engineering is the process of enriching a data set by creating additional independent variables based on existing attributes.

- New features can help improve the accuracy and predictive power of ML models
- Feature engineering is often used to convert fields into “model-friendly” formats. For example, one-hot encoding transforms categorical variables into binary (1/0) fields.

6.3.3 Selecting the Target Feature

The most suitable target feature for classification in the data set is the "**is_canceled**" feature, which indicates whether or not a booking was ultimately cancelled.

As "is_canceled" can only assume one of two values, this is a binary classification problem. Solving this classification problem allows us to predict whether or not a customer will cancel their hotel booking based on the details associated with the booking.

Predicting booking cancellations is of great businesses interest to hotels. An accurate forecast of room cancellations can allow a hotel to efficiently manage its room inventory, optimize its overbooking strategy, and determine its cancellation policy.

6.3.4 Removing Irrelevant Attributes

The goal for this classification problem is to predict whether or not a booking will be cancelled based solely on the information available at the time the booking is placed.

Below attributes are irrelevant for this classification problem as their values can only be known after a booking has been placed. These irrelevant attributes were dropped from the data set

- **arrival_date_week_number, arrival_date_month**
- **arrival_date_year, stays_in_week_nights**
- **stays_in_weekend_nights, reservation_status_date.**
- **reservation_status** attribute seems to be the most impactful feature and because of its negative correlation with the "**is_canceled**" feature it can cause a wrong prediction or overfitting and there is a chance of data leakage. Hence, I will drop this irrelevant feature.

6.3.5 Feature Encoding Categorial Attributes

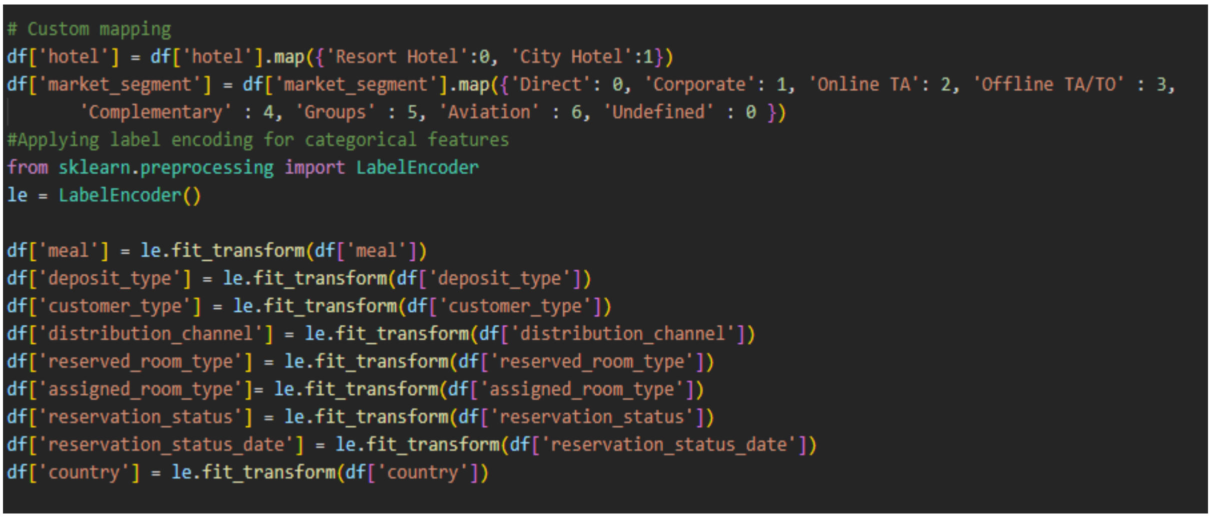
In Machine Learning, we have many types of data types like String, Integer, Datetime etc. We know Computers & our Machine Learning Model can only understand and interpret numericdata.

We cannot pass in categorical features in Machine Learning models. So, we need to convert them into numeric features. This is where Feature Encoding comes into the picture.

Feature Encoding is converting and representing the String data in a numeric format or representing them in such a way that our model can interpret that data.

In this Classification problem, I have applied various Feature Encoding techniques like **Label Encoding, Custom Mapping, Replacing** etc.

6.3.6 Implementation of Feature Encoding techniques using Python applied for categoricalfeatures in the dataset

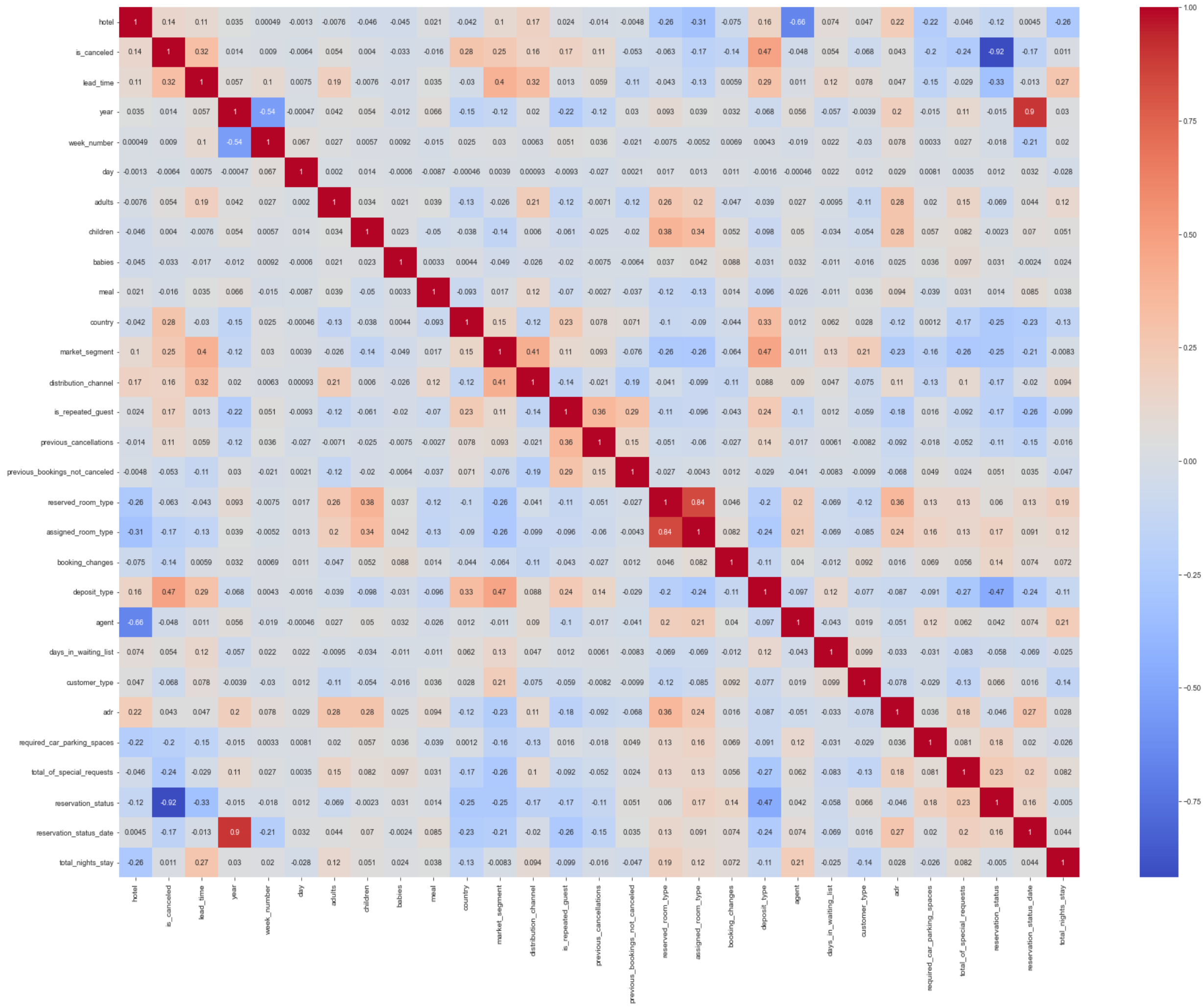


```
# Custom mapping
df['hotel'] = df['hotel'].map({'Resort Hotel':0, 'City Hotel':1})
df['market_segment'] = df['market_segment'].map({'Direct': 0, 'Corporate': 1, 'Online TA': 2, 'Offline TA/TO' : 3,
       'Complementary' : 4, 'Groups' : 5, 'Aviation' : 6, 'Undefined' : 0 })
#Applying label encoding for categorical features
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

df['meal'] = le.fit_transform(df['meal'])
df['deposit_type'] = le.fit_transform(df['deposit_type'])
df['customer_type'] = le.fit_transform(df['customer_type'])
df['distribution_channel'] = le.fit_transform(df['distribution_channel'])
df['reserved_room_type'] = le.fit_transform(df['reserved_room_type'])
df['assigned_room_type']= le.fit_transform(df['assigned_room_type'])
df['reservation_status'] = le.fit_transform(df['reservation_status'])
df['reservation_status_date'] = le.fit_transform(df['reservation_status_date'])
df['country'] = le.fit_transform(df['country'])
```

6.3.7 Correlation

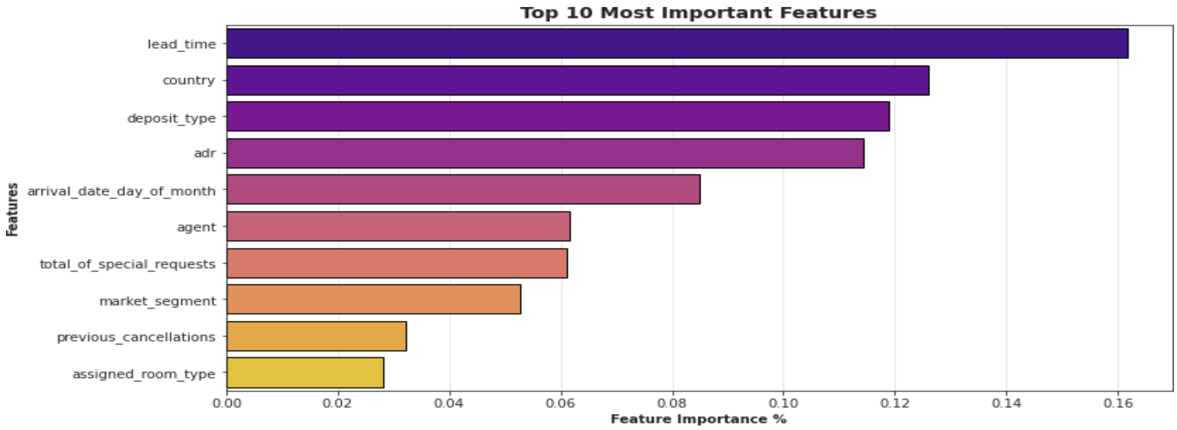
Correlation is the most common multivariate profiling metric and is used to describe how a pair of variables are linearly related. In simpler words, for a given row, when one variable's observation goes above its mean, does the other variable's observation also go above its mean.

Relationship between Independent and dependent feature (Correlation Heat map



6.3.8 Feature Selection

Feature selection is the process of reducing the number of input variables when developing apredictive model and relationship between each input variable and the target variable using statistics and selecting those input variables that have the strongest relationship with the target variable.



8 Modeling – Classification

8.1 Algorithm Selection

I have selected four Machine Learning algorithms like **Logistic Regression, KNN, SVM, Naïve Bayes Classification, Decision Trees** and advanced Ensemble Algorithms like **Random Forest** and **XGBoost** for the classification problem.

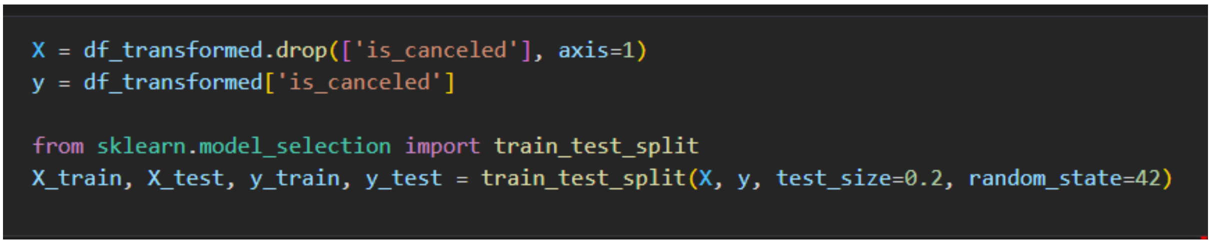
- **Logistic Regression** utilizes the logistic function, also known as the sigmoid function, to output the probability that an observation belongs to the positive class in a binary classification problem. As the output of the logistic function is bounded between 0 and 1 for all input values, it is well suited for modelling probabilities.
- **Decision Tree** is a Supervised algorithm that uses a set of rules to make decisions, similar to how humans make decisions. It has several advantages like visualizing the decision tree and no pre-processing required
- **Random Forest** is an ensemble machine learning algorithm that follows the bagging technique. The base estimators in random forests are decision trees. Random forest randomly selects a set of features that are used to decide the best split at each node of the decision tree. Random forest uses a collection of multiple decision trees known as a random forest to maximize accuracy.
- **XGBoost (Extreme Gradient Boosting)** is an advanced implementation of the gradient boosting algorithm. XGBoost has an immensely high predictive power ensemble algorithm which makes it the best choice for accuracy making the algorithm almost 10times faster than existing gradient booster techniques. XGBoost is comparatively better than other algorithms because it

    1. Reduces Overfitting
    2. Parallel processing
    3. Handles missing values
    4. Built-in Cross-Validation

8.2 Model Training

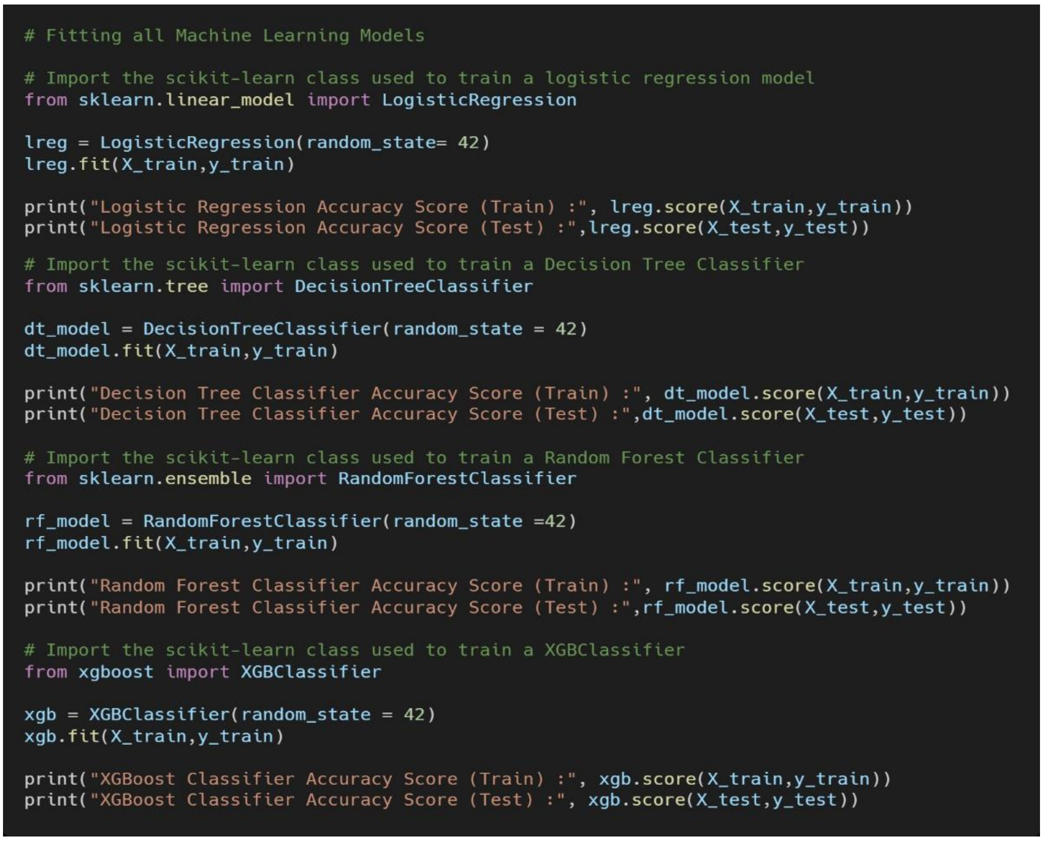
7.2.1 Train - Test Split

I have reserved 75% of the observations for the train set and 25% of the observations for the testset.



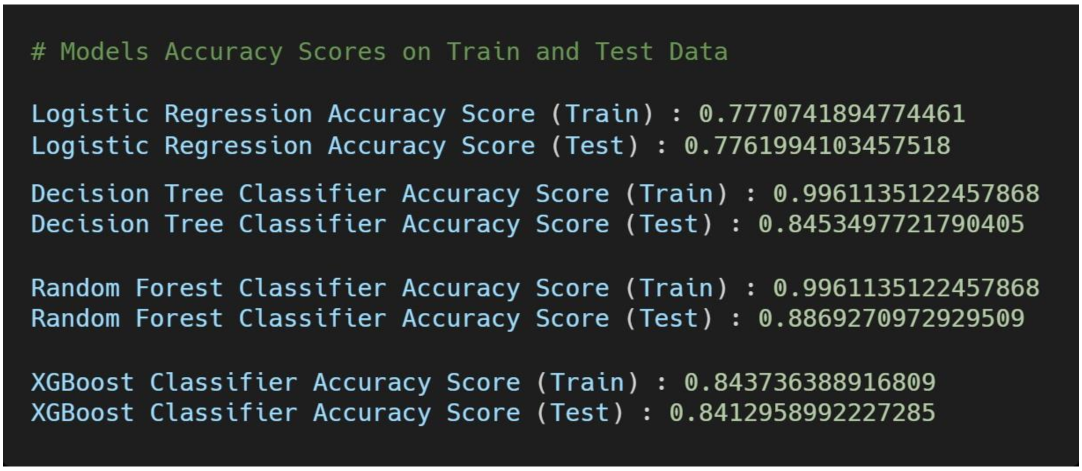
```
X = df_transformed.drop(['is_canceled'], axis=1)
y = df_transformed['is_canceled']

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

7.2.2 Fitting Machine Learning Models using Scikit – Learn Library



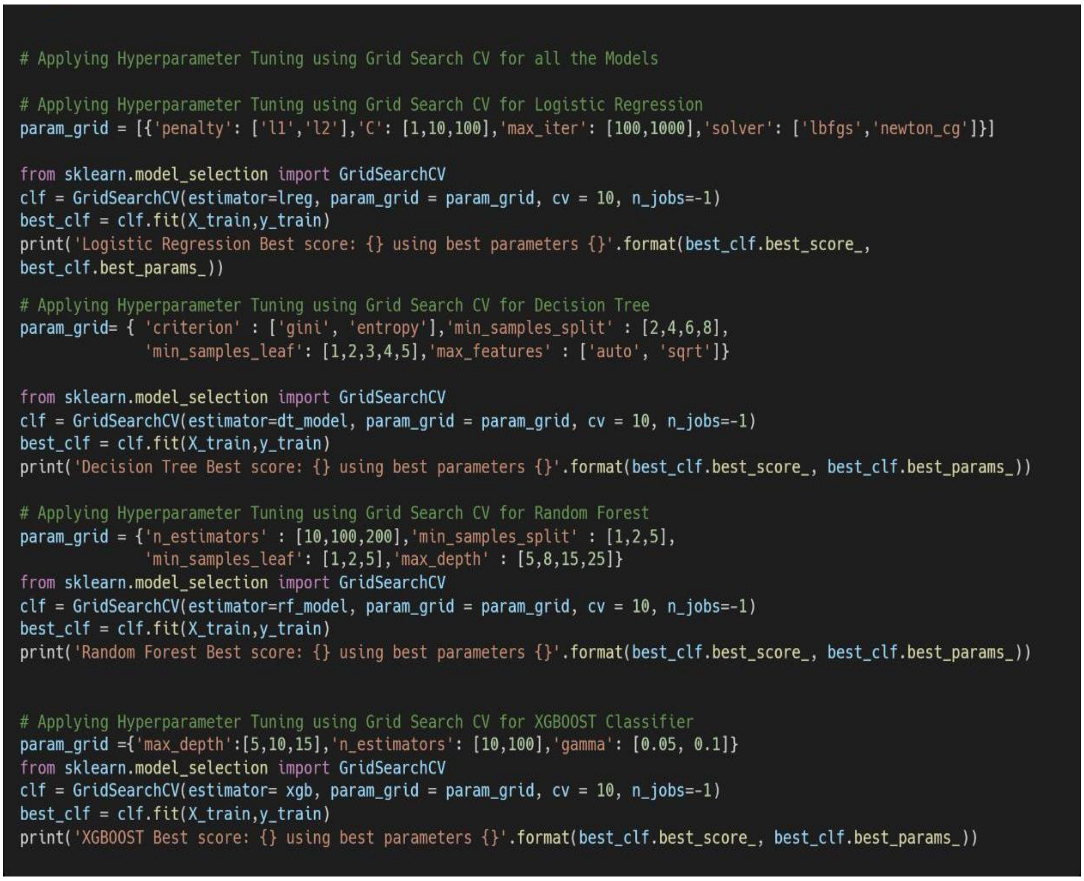
```
# Fitting all Machine Learning Models

# Import the scikit-learn class used to train a logistic regression model
from sklearn.linear_model import LogisticRegression

lreg = LogisticRegression(random_state= 42)
lreg.fit(X_train,y_train)

print("Logistic Regression Accuracy Score (Train) :", lreg.score(X_train,y_train))
print("Logistic Regression Accuracy Score (Test) :",lreg.score(X_test,y_test))

# Import the scikit-learn class used to train a Decision Tree Classifier
from sklearn.tree import DecisionTreeClassifier

dt_model = DecisionTreeClassifier(random_state = 42)
dt_model.fit(X_train,y_train)

print("Decision Tree Classifier Accuracy Score (Train) :", dt_model.score(X_train,y_train))
print("Decision Tree Classifier Accuracy Score (Test) :",dt_model.score(X_test,y_test))

# Import the scikit-learn class used to train a Random Forest Classifier
from sklearn.ensemble import RandomForestClassifier

rf_model = RandomForestClassifier(random_state =42)
rf_model.fit(X_train,y_train)

print("Random Forest Classifier Accuracy Score (Train) :", rf_model.score(X_train,y_train))
print("Random Forest Classifier Accuracy Score (Test) :",rf_model.score(X_test,y_test))

# Import the scikit-learn class used to train a XGBClassifier
from xgboost import XGBClassifier

xgb = XGBClassifier(random_state = 42)
xgb.fit(X_train,y_train)

print("XGBoost Classifier Accuracy Score (Train) :", xgb.score(X_train,y_train))
print("XGBoost Classifier Accuracy Score (Test) :", xgb.score(X_test,y_test))
```

7.2.3 Models Accuracy Scores on Train and Test Data



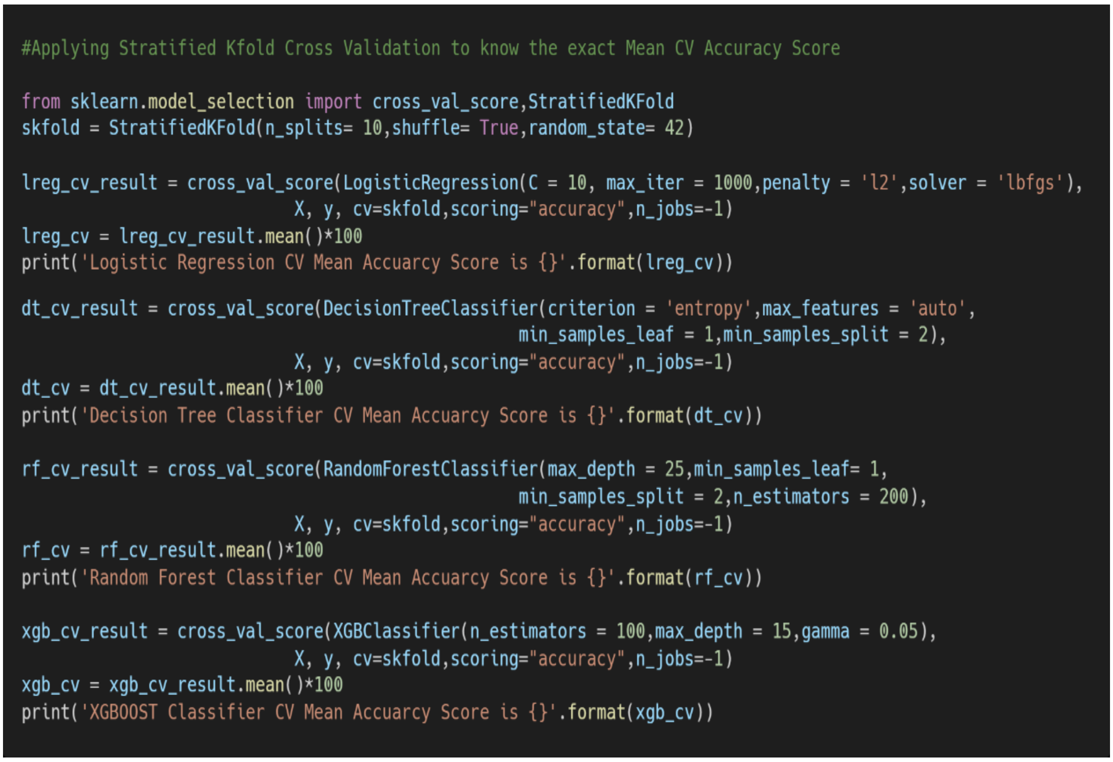
```
# Models Accuracy Scores on Train and Test Data

Logistic Regression Accuracy Score (Train) : 0.7770741894774461
Logistic Regression Accuracy Score (Test) : 0.7761994103457518

Decision Tree Classifier Accuracy Score (Train) : 0.9961135122457868
Decision Tree Classifier Accuracy Score (Test) : 0.8453497721790405

Random Forest Classifier Accuracy Score (Train) : 0.9961135122457868
Random Forest Classifier Accuracy Score (Test) : 0.8869270972929509

XGBoost Classifier Accuracy Score (Train) : 0.843736388916809
XGBoost Classifier Accuracy Score (Test) : 0.8412958992227285
```

7.2.4 Applying Hyperparameter Tuning using Grid Search CV for all the Models



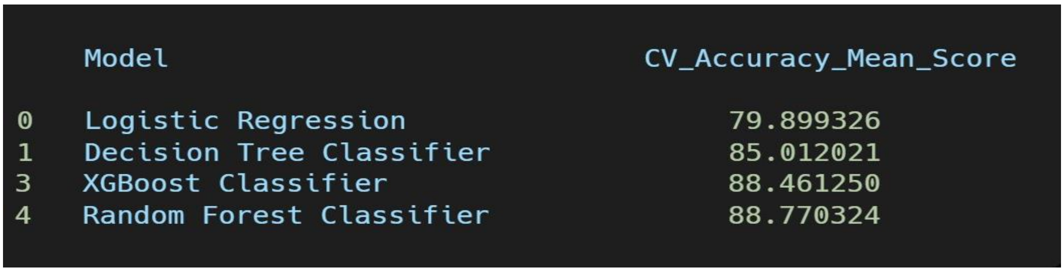
```
# Applying Hyperparameter Tuning using Grid Search CV for all the Models

# Applying Hyperparameter Tuning using Grid Search CV for Logistic Regression
param_grid = [{'penalty': ['l1','l2'],'C': [1,10,100],'max_iter': [100,1000],'solver': ['lbfgs','newton_cg']}]

from sklearn.model_selection import GridSearchCV
clf = GridSearchCV(estimator=lreg, param_grid = param_grid, cv = 10, n_jobs=-1)
best_clf = clf.fit(X_train,y_train)
print('Logistic Regression Best score: {} using best parameters {}'.format(best_clf.best_score_,
best_clf.best_params_))

# Applying Hyperparameter Tuning using Grid Search CV for Decision Tree
param_grid= { 'criterion' : ['gini', 'entropy'],'min_samples_split' : [2,4,6,8],
              'min_samples_leaf': [1,2,3,4,5],'max_features' : ['auto', 'sqrt']}

from sklearn.model_selection import GridSearchCV
clf = GridSearchCV(estimator=dt_model, param_grid = param_grid, cv = 10, n_jobs=-1)
best_clf = clf.fit(X_train,y_train)
print('Decision Tree Best score: {} using best parameters {}'.format(best_clf.best_score_, best_clf.best_params_))

# Applying Hyperparameter Tuning using Grid Search CV for Random Forest
param_grid = {'n_estimators' : [10,100,200],'min_samples_split' : [1,2,5],
              'min_samples_leaf': [1,2,5],'max_depth' : [5,8,15,25]}
from sklearn.model_selection import GridSearchCV
clf = GridSearchCV(estimator=rf_model, param_grid = param_grid, cv = 10, n_jobs=-1)
best_clf = clf.fit(X_train,y_train)
print('Random Forest Best score: {} using best parameters {}'.format(best_clf.best_score_, best_clf.best_params_))


# Applying Hyperparameter Tuning using Grid Search CV for XGBOOST Classifier
param_grid ={'max_depth':[5,10,15],'n_estimators': [10,100],'gamma': [0.05, 0.1]}
from sklearn.model_selection import GridSearchCV
clf = GridSearchCV(estimator= xgb, param_grid = param_grid, cv = 10, n_jobs=-1)
best_clf = clf.fit(X_train,y_train)
print('XGBOOST Best score: {} using best parameters {}'.format(best_clf.best_score_, best_clf.best_params_))
```

7.2.5 Applying Stratified Kfold Cross-Validation to know the exact Mean CV Accuracy Score



```
#Applying Stratified Kfold Cross Validation to know the exact Mean CV Accuracy Score

from sklearn.model_selection import cross_val_score,StratifiedKFold
skfold = StratifiedKFold(n_splits= 10,shuffle= True,random_state= 42)

lreg_cv_result = cross_val_score(LogisticRegression(C = 10, max_iter = 1000,penalty = 'l2',solver = 'lbfgs'),
                            X, y, cv=skfold,scoring="accuracy",n_jobs=-1)
lreg_cv = lreg_cv_result.mean()*100
print('Logistic Regression CV Mean Accuarcy Score is {}'.format(lreg_cv))

dt_cv_result = cross_val_score(DecisionTreeClassifier(criterion = 'entropy',max_features = 'auto',
                                                   min_samples_leaf = 1,min_samples_split = 2),
                            X, y, cv=skfold,scoring="accuracy",n_jobs=-1)
dt_cv = dt_cv_result.mean()*100
print('Decision Tree Classifier CV Mean Accuarcy Score is {}'.format(dt_cv))

rf_cv_result = cross_val_score(RandomForestClassifier(max_depth = 25,min_samples_leaf= 1,
                                                   min_samples_split = 2,n_estimators = 200),
                            X, y, cv=skfold,scoring="accuracy",n_jobs=-1)
rf_cv = rf_cv_result.mean()*100
print('Random Forest Classifier CV Mean Accuarcy Score is {}'.format(rf_cv))

xgb_cv_result = cross_val_score(XGBClassifier(n_estimators = 100,max_depth = 15,gamma = 0.05),
                            X, y, cv=skfold,scoring="accuracy",n_jobs=-1)
xgb_cv = xgb_cv_result.mean()*100
print('XGBOOST Classifier CV Mean Accuarcy Score is {}'.format(xgb_cv))
```

7.2.6 Cross-Validation Mean Accuracy Scores of the Models



```
   Model                          CV_Accuracy_Mean_Score

0  Logistic Regression                  79.899326
1  Decision Tree Classifier             85.012021
3  XGBoost Classifier                   88.461250
4  Random Forest Classifier             88.770324
```

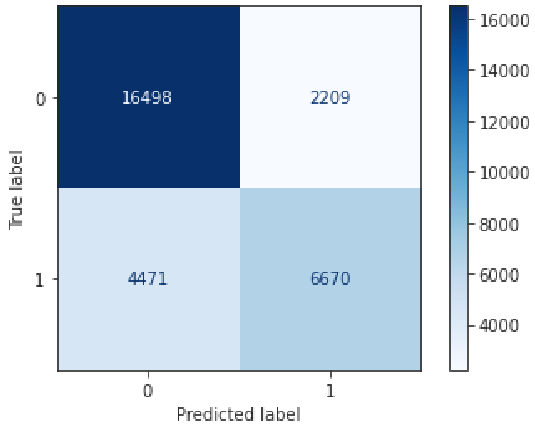
9 Model Evaluation – Classification

The metrics used to evaluate the performance of the models on the cross-validation set are F1 score and accuracy. While accuracy is likely the most common metric used for evaluating classifiers, it can misrepresent the performance of classifiers with imbalanced classes in the target feature.

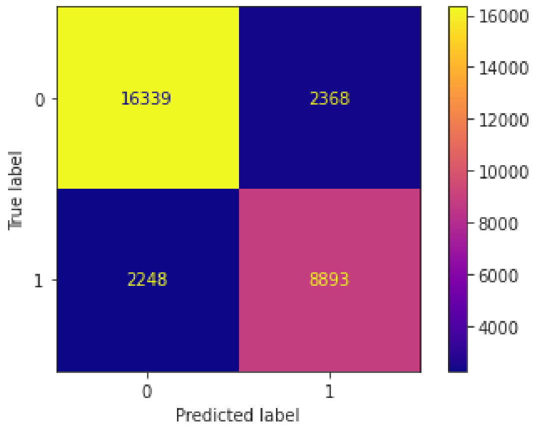
The primary evaluation metric, F1 score, ranges from 0 to 1 and considers both precision and recall, making it well suited for evaluating classifiers with imbalanced classes in the target feature.

9.1 Performance on Train and Test Sets Using Confusion Matrix

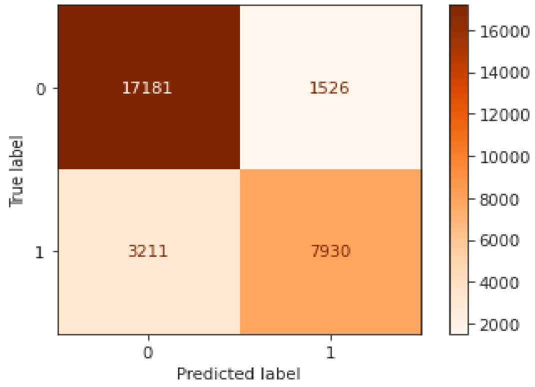
Logistic Regression Confusion Matrix



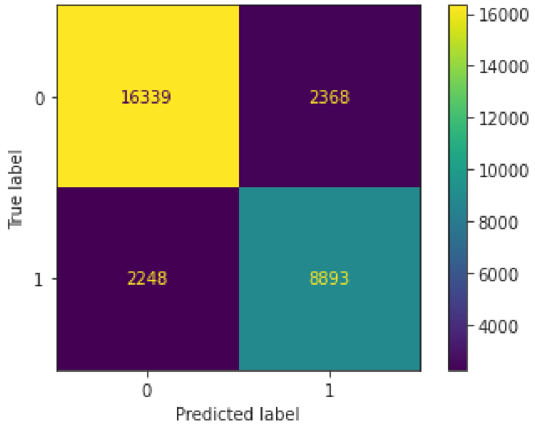
Decision Tree Confusion Matrix



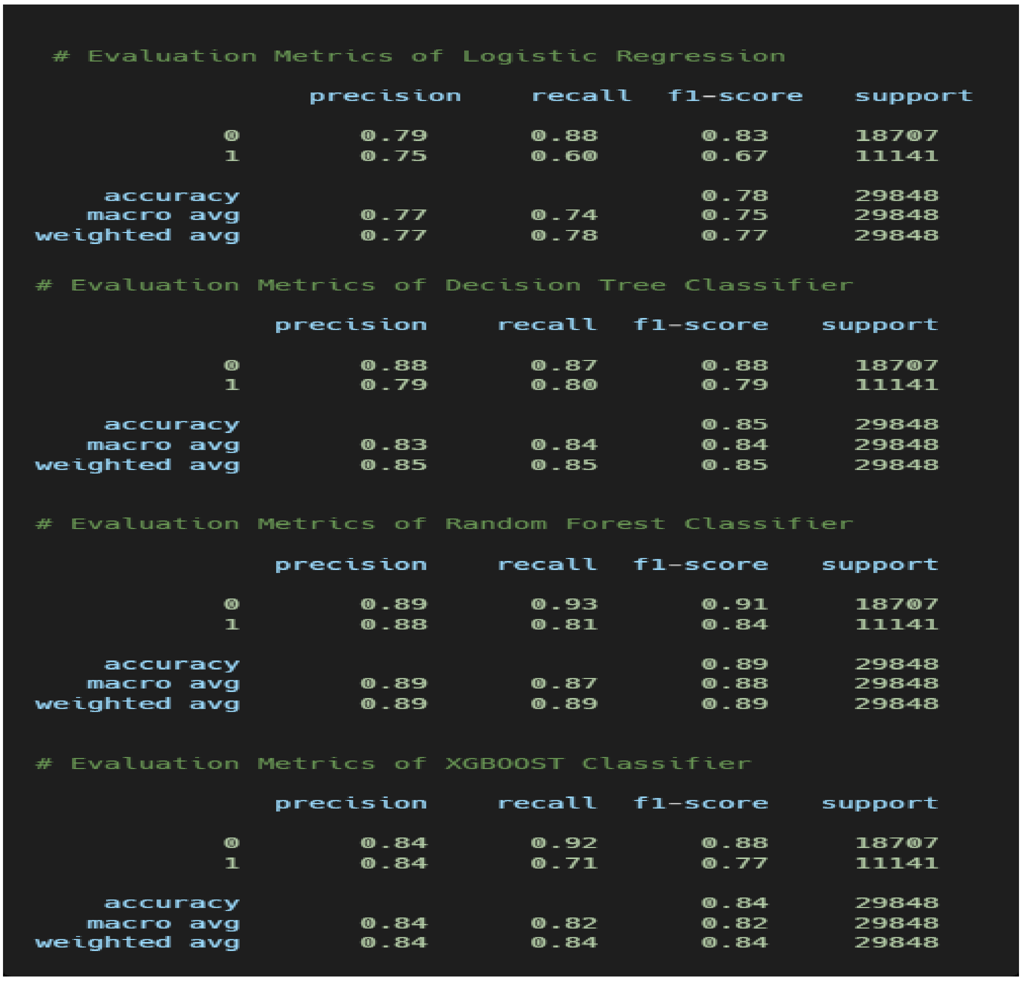
XGBOOST Confusion Matrix



Random Forest Confusion Matrix



9.2 Classification Model Evaluation Metrics Report



```
  # Evaluation Metrics of Logistic Regression

               precision    recall  f1-score   support

           0       0.79      0.88      0.83     18707
           1       0.75      0.60      0.67     11141

    accuracy                           0.78     29848
   macro avg       0.77      0.74      0.75     29848
weighted avg       0.77      0.78      0.77     29848

# Evaluation Metrics of Decision Tree Classifier

              precision    recall  f1-score   support

           0       0.88      0.87      0.88     18707
           1       0.79      0.80      0.79     11141

    accuracy                           0.85     29848
   macro avg       0.83      0.84      0.84     29848
weighted avg       0.85      0.85      0.85     29848

# Evaluation Metrics of Random Forest Classifier

              precision    recall  f1-score   support

           0       0.89      0.93      0.91     18707
           1       0.88      0.81      0.84     11141

    accuracy                           0.89     29848
   macro avg       0.89      0.87      0.88     29848
weighted avg       0.89      0.89      0.89     29848

# Evaluation Metrics of XGBOOST Classifier

              precision    recall  f1-score   support

           0       0.84      0.92      0.88     18707
           1       0.84      0.71      0.77     11141

    accuracy                           0.84     29848
   macro avg       0.84      0.82      0.82     29848
weighted avg       0.84      0.84      0.84     29848
```

9.3 Performance Summary and Model Selection - Classification

Model	Dataset	F1 Score	Accuracy
Logistic Regression	Train	0.83	0.78
	Test	0.67	
Decision Tree Classifier	Train	0.88	0.85
	Test	0.79	
Random Forest Classifier	Train	0.91	0.89
	Test	0.89	
XGBOOST Classifier	Train	0.88	0.84
	Test	0.77	

While neither of these models can be considered skilled at predicting whether or not a hotel booking will be cancelled, the classification performance of the Random Forest Algorithm was better than that of the XGBoost model. This result suggests that a non-linear model is more capable of capturing the relationships between the features in this data set.

10 Supervised Machine Learning – Regression Problem

10.1 Regression Problem Statement

Predicting the Cost/Price of a Hotel Booking made by the Hotel Guest/Customer.

10.2 Feature Engineering

Having cleaned the data set in the first Supervised ML classification problem, I will beginthe process of feature engineering to solve the second ML problem addressed in this project.

10.3 Selecting Target Feature

Firstly, I will select the target feature for the Regression problem, i.e., the numeric feature that will be predicted. The most suitable feature for regression in the data set is the "ADR" feature. "adr" represents the Average Daily Rate or Average Daily Cost for the hotel room associated with the booking. Hence solving this Regression problem allows us to predict how much a hotel room might cost based on its booking details, such as the arrival date and the room type.

10.4 Removing Irrelevant Attributes

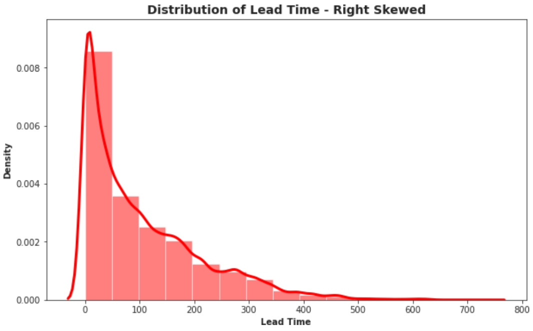
The data set contains several attributes that are irrelevant to the task of predicting the Average Daily Rate (ADR) for a hotel booking. In particular, these attributes correspond to booking details that cannot be known at the time a booking is placed by a customer. These attributes can only be known after the initial booking, such as when the customer arrives at the hotel.

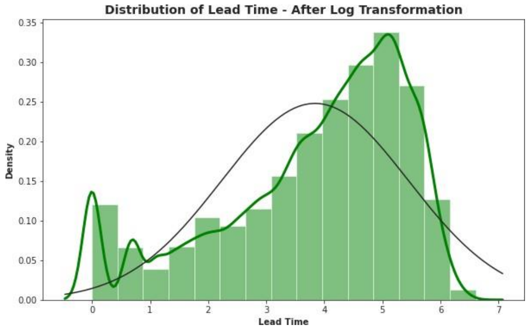
The irrelevant attributes for this regression problem should be dropped from the dataset

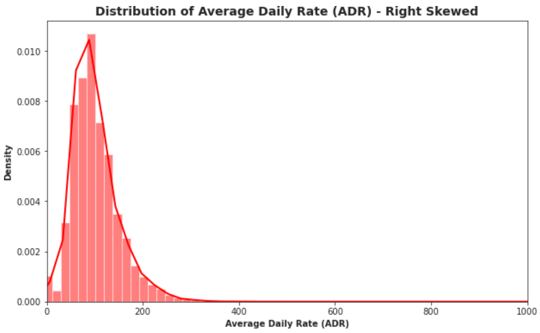
- **assigned_room_type**
- **booking_change**
- **is_canceled**
- **reservation_status**
- **reservation_status_date**

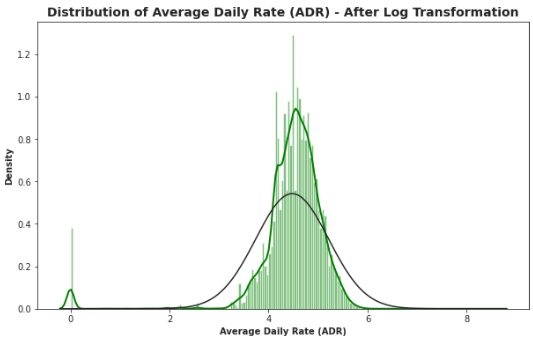
10.5 Handling Skewness in the dataset

- Many numerical attributes follow a normal distribution, also known as a **Gaussian distribution or Bell Curve**. The **skewness** of a distribution is defined as the lack of symmetry and tells us about the distribution of our data. In a symmetrical distribution,the Mean, Median and Mode are equal. The normal distribution has a skewness of 0.
- From both the graphs, we can depict that distribution of numerical features "**Lead time"** and **"Average Daily Rate"** is heavily right-skewed. So, I will apply a logarithmic transformation using NumPy’s log1p method to make it Normal Distribution.









11 Modeling – Regression Problem

11.1 Algorithm Selection

Linear models are preferable as they are more interpretable, less computationally expensive and they are unable to model some of the more complex relationships that non-linear modelscan handle.

If the performance of a linear model and a non-linear model is comparable, I would use and optimize the linear model because of its simplicity. However, if the performance of the linear model was considerably worse than that of the non-linear model, I would use and optimize the non-linear model despite its complexity.

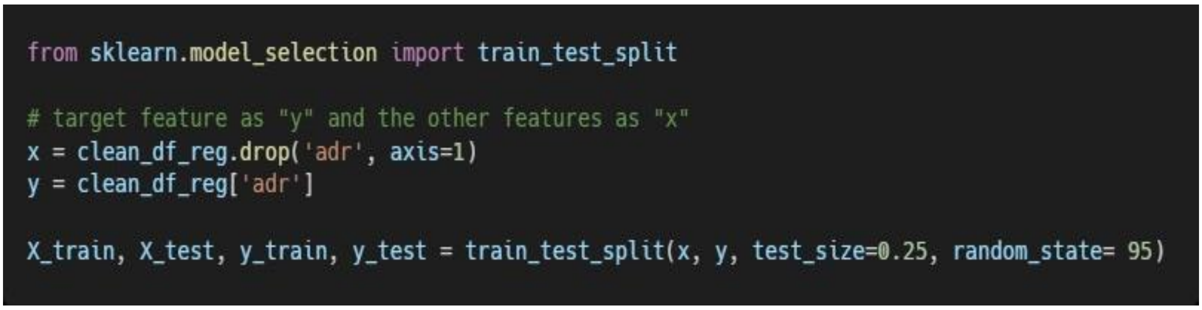
I have selected **Elastic-Net** as the linear regression model for this problem. Elastic-Net is a compromise between Lasso Regression and Ridge Regression that has built-in feature selection and intuitive handling of collinear features. Through the appropriate selection of hyperparameters, Elastic-Net reverts to either Lasso Regression or Ridge Regression, which gives Elastic-Net a great deal of flexibility as a linear regression model.

I have selected **Random Forest** as the non-linear regression model. A random forest is an ensemble model that consists of many decision trees that have been combined via bagging. Random forests generally exhibit strong out-of-the-box performance, don't require extensive hyperparameter tuning, and work well on a wide variety of data sets.

11.2 Model Training

10.2.1 Train - Test Split

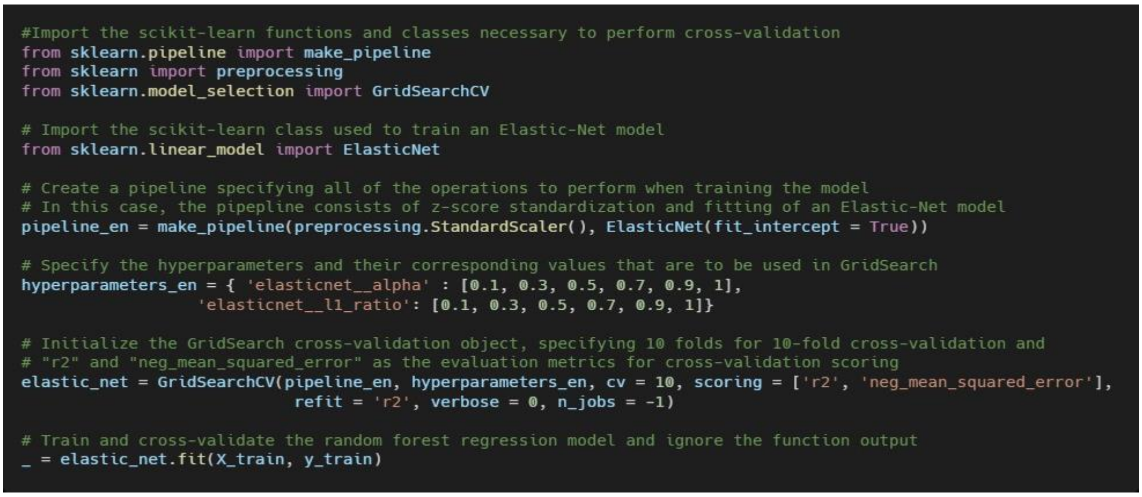
First, I split the data set to reserve 75% of the observations for model training and cross-validationand 25% of the observations for model testing.



```
from sklearn.model_selection import train_test_split

# target feature as "y" and the other features as "x"
x = clean_df_reg.drop('adr', axis=1)
y = clean_df_reg['adr']

X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state= 95)
```

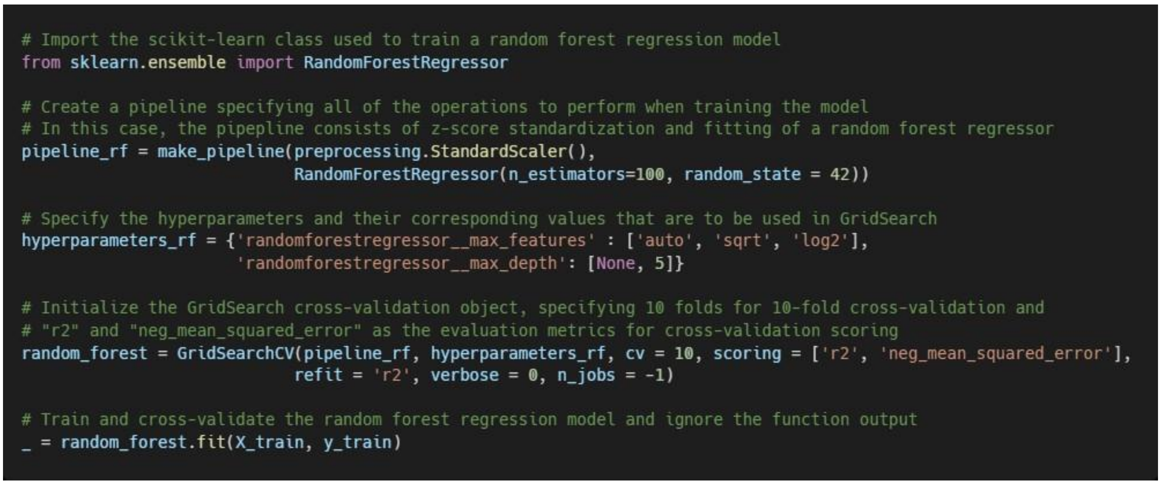
10.2.2 Fitting ML Models using Scikit – Learn Library to perform Grid Search CV Cross-Validation

Grid Search CV Cross-Validation for Elastic Net



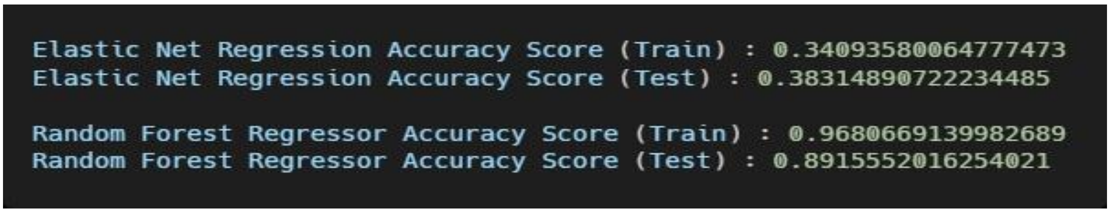
```
#Import the scikit-learn functions and classes necessary to perform cross-validation
from sklearn.pipeline import make_pipeline
from sklearn import preprocessing
from sklearn.model_selection import GridSearchCV

# Import the scikit-learn class used to train an Elastic-Net model
from sklearn.linear_model import ElasticNet

# Create a pipeline specifying all of the operations to perform when training the model
# In this case, the pipepline consists of z-score standardization and fitting of an Elastic-Net model
pipeline_en = make_pipeline(preprocessing.StandardScaler(), ElasticNet(fit_intercept = True))

# Specify the hyperparameters and their corresponding values that are to be used in GridSearch
hyperparameters_en = { 'elasticnet__alpha' : [0.1, 0.3, 0.5, 0.7, 0.9, 1],
                  'elasticnet__l1_ratio': [0.1, 0.3, 0.5, 0.7, 0.9, 1]}

# Initialize the GridSearch cross-validation object, specifying 10 folds for 10-fold cross-validation and
# "r2" and "neg_mean_squared_error" as the evaluation metrics for cross-validation scoring
elastic_net = GridSearchCV(pipeline_en, hyperparameters_en, cv = 10, scoring = ['r2', 'neg_mean_squared_error'],
                            refit = 'r2', verbose = 0, n_jobs = -1)

# Train and cross-validate the random forest regression model and ignore the function output
_ = elastic_net.fit(X_train, y_train)
```

Grid Search CV Cross-Validation for Random Forest Regressor



```
# Import the scikit-learn class used to train a random forest regression model
from sklearn.ensemble import RandomForestRegressor

# Create a pipeline specifying all of the operations to perform when training the model
# In this case, the pipepline consists of z-score standardization and fitting of a random forest regressor
pipeline_rf = make_pipeline(preprocessing.StandardScaler(),
                            RandomForestRegressor(n_estimators=100, random_state = 42))

# Specify the hyperparameters and their corresponding values that are to be used in GridSearch
hyperparameters_rf = {'randomforestregressor__max_features' : ['auto', 'sqrt', 'log2'],
                      'randomforestregressor__max_depth': [None, 5]}

# Initialize the GridSearch cross-validation object, specifying 10 folds for 10-fold cross-validation and
# "r2" and "neg_mean_squared_error" as the evaluation metrics for cross-validation scoring
random_forest = GridSearchCV(pipeline_rf, hyperparameters_rf, cv = 10, scoring = ['r2', 'neg_mean_squared_error'],
                            refit = 'r2', verbose = 0, n_jobs = -1)

# Train and cross-validate the random forest regression model and ignore the function output
_ = random_forest.fit(X_train, y_train)
```

10.2.3 Cross-Validation Mean Accuracy Scores on Train and Test Data

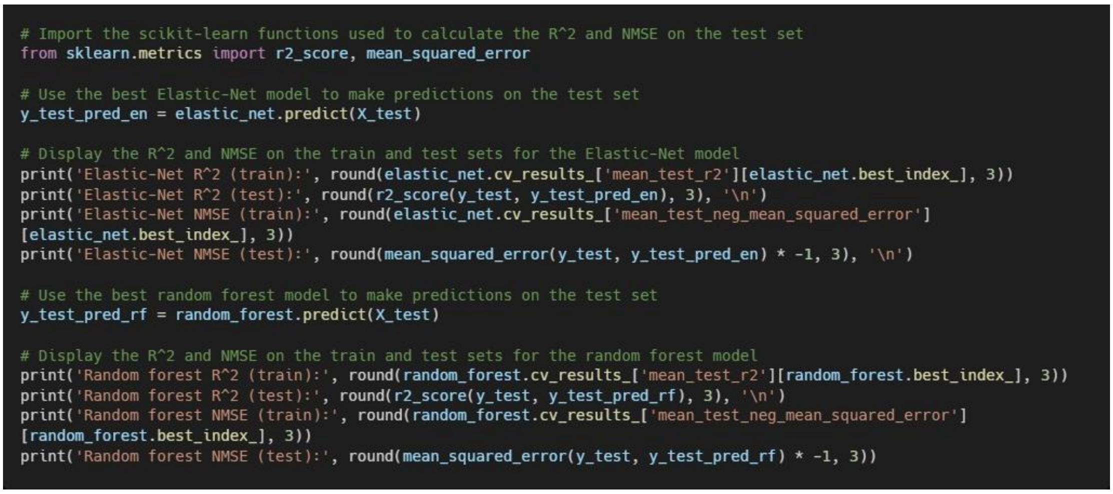


```
Elastic Net Regression Accuracy Score (Train) : 0.34093580064777473
Elastic Net Regression Accuracy Score (Test) : 0.38314890722234485

Random Forest Regressor Accuracy Score (Train) : 0.9680669139982689
Random Forest Regressor Accuracy Score (Test) : 0.8915552016254021
```

12 Model Evaluation – Regression

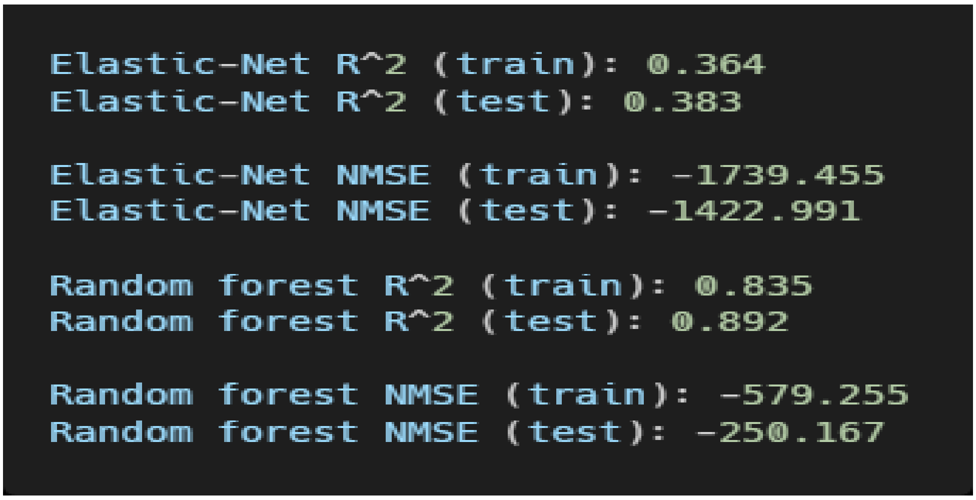
12.1 Performance on Train and Test Sets

Having trained and cross-validated the models, I then used the models to make predictions onthe test set. I evaluated the performance of the models on the test set using the same R^2 andNMSE metrics used to evaluate the models during cross-validation.



```
# Import the scikit-learn functions used to calculate the R^2 and NMSE on the test set
from sklearn.metrics import r2_score, mean_squared_error

# Use the best Elastic-Net model to make predictions on the test set
y_test_pred_en = elastic_net.predict(X_test)

# Display the R^2 and NMSE on the train and test sets for the Elastic-Net model
print('Elastic-Net R^2 (train):', round(elastic_net.cv_results_['mean_test_r2'][elastic_net.best_index_], 3))
print('Elastic-Net R^2 (test):', round(r2_score(y_test, y_test_pred_en), 3), '\n')
print('Elastic-Net NMSE (train):', round(elastic_net.cv_results_['mean_test_neg_mean_squared_error']
[elastic_net.best_index_], 3))
print('Elastic-Net NMSE (test):', round(mean_squared_error(y_test, y_test_pred_en) * -1, 3), '\n')

# Use the best random forest model to make predictions on the test set
y_test_pred_rf = random_forest.predict(X_test)

# Display the R^2 and NMSE on the train and test sets for the random forest model
print('Random forest R^2 (train):', round(random_forest.cv_results_['mean_test_r2'][random_forest.best_index_], 3))
print('Random forest R^2 (test):', round(r2_score(y_test, y_test_pred_rf), 3), '\n')
print('Random forest NMSE (train):', round(random_forest.cv_results_['mean_test_neg_mean_squared_error']
[random_forest.best_index_], 3))
print('Random forest NMSE (test):', round(mean_squared_error(y_test, y_test_pred_rf) * -1, 3))
```

The performance of the models as indicated by these metrics is displayed below.



```
Elastic-Net R^2 (train): 0.364
Elastic-Net R^2 (test): 0.383

Elastic-Net NMSE (train): -1739.455
Elastic-Net NMSE (test): -1422.991

Random forest R^2 (train): 0.835
Random forest R^2 (test): 0.892

Random forest NMSE (train): -579.255
Random forest NMSE (test): -250.167
```

12.2 Performance Summary and Model Selection

Model	Dataset	R2	NMSE	Accuracy
Elastic Net	Train	0.36	-1739.45	0.34
	Test	0.38	-1422.29	0.38
Random Forest	Train	0.83	-579.25	0.96
	Test	0.89	-250.16	0.89

Based on these findings, it is clear that the Random Forest model is better at predicting the Average Daily Rate for hotel bookings. The poor performance of the Elastic-Net model indicates that a linear model is not capable of accurately representing this data and that the non-linear random forest model should be used and optimized despite its increased complexity and computation cost.

13 Unsupervised Machine Learning – Clustering Problem

13.1 Clustering Problem Statement

The final problem addressed in this project is clustering via unsupervised learning. The objective of clustering is to group similar observations in a data set when the groups, referredto as clusters, are not predefined.

Clustering is utilized in a variety of fields to illuminate patterns in data sets. In business, clustering is a powerful tool for performing customer segmentation.

The purpose of the cluster analysis carried out in this project is to find groups of similar hotel customers based on the details of their hotel bookings.

Among many other use cases, the two hotels represented in this data set could utilize these clusters to **identify profitable hotel customers/guests by customer market segmentation**.

13.2 Feature Engineering

12.2.1 Selecting the Target Feature

As clustering is an unsupervised ML problem in which the clusters are not predefined, a targetfeature does not need to be specified.

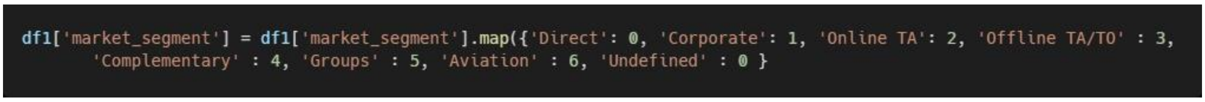
12.2.2 Removing Irrelevant attributes

The goal of this cluster analysis is to segment profitable customers who have already stayed at the hotel or have cancelled their bookings. As such, there is no need to limit the data set to attributes that were determined at the time the initial booking was placed.

12.2.3 Feature Encoding

The "**Market Segment**" attribute entries in the dataset are categorical and these values should be encoded to apply cluster analysis on the dataset to cluster or group by market segment type.

To encode the "market segment" feature, I will apply a custom mapping technique to convertit into a numeric feature.



```
df1['market_segment'] = df1['market_segment'].map({'Direct': 0, 'Corporate': 1, 'Online TA': 2, 'Offline TA/TO' : 3,
      'Complementary' : 4, 'Groups' : 5, 'Aviation' : 6, 'Undefined' : 0 }
```

14 Modeling – Clustering Problem

14.1 Algorithm Selection

I have decided to select K-Means as the clustering algorithm for this project. One of the most common algorithms used for clustering is the K-Means algorithm, which utilizes an iterative process to group the observations into k clusters.

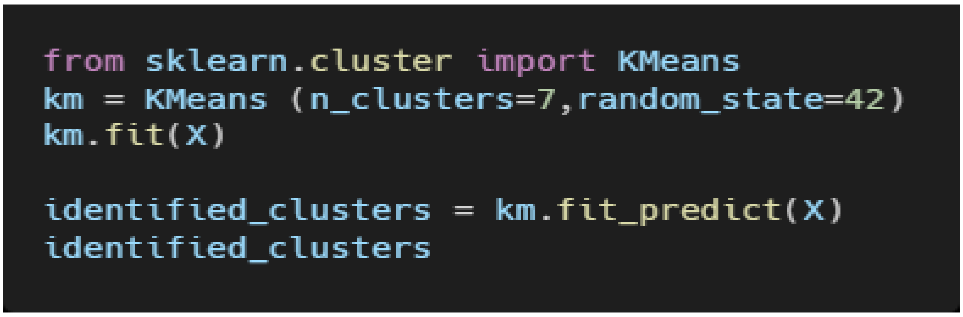
- First, the algorithm randomly initializes k cluster centroids, or feature vectors, within the feature space.
- Next, the algorithm performs a cluster assignment step by assigning each observationto its nearest cluster centroid, as measured by Euclidean distance. A move centroid step is then performed by moving each cluster centroid to the new mean feature vector of its corresponding cluster.
- The cluster assignment and move centroid steps are then repeated until the algorithm converges on a final location for the cluster centroids.

The k-means algorithm is interpretable and robust but its handling of categorical features can be unintuitive due to its reliance on Euclidean distance and mean, metrics that are more useful for working with numeric features than categorical features.

14.2 Model Fitting

When it comes to choosing the number of clusters, one possible solution is to use what is called the Elbow method. However, this technique is not necessarily suitable for smaller clusters.

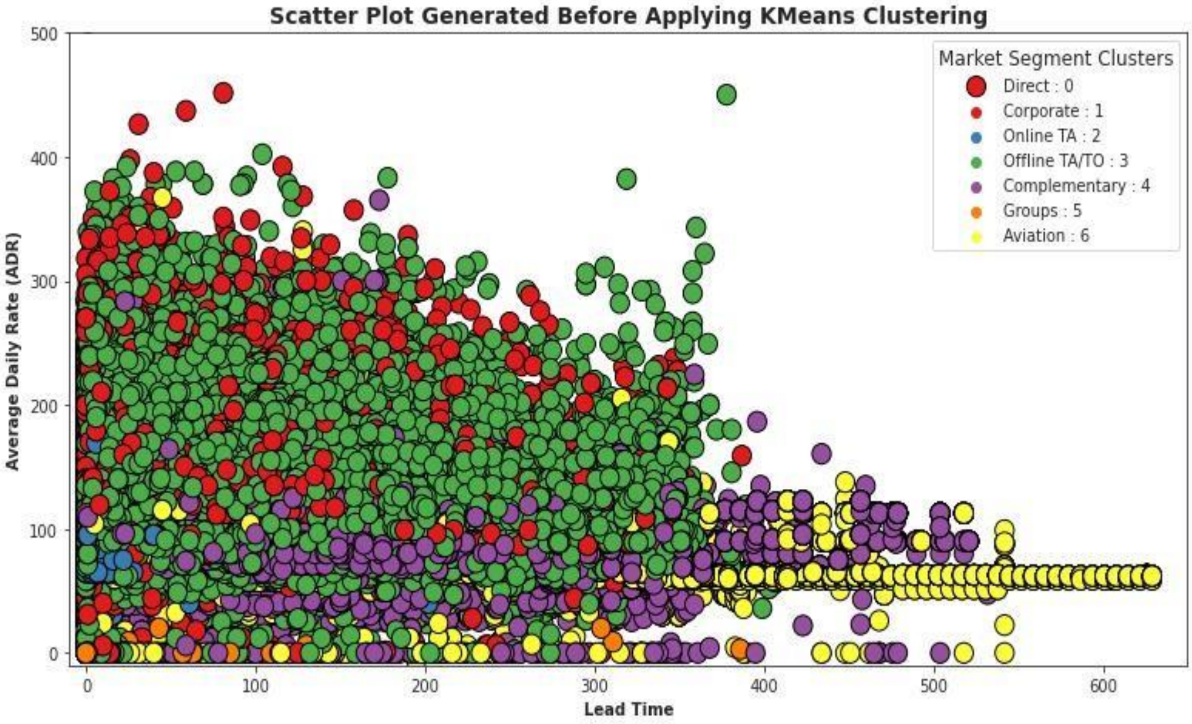
Moreover, we already know the number of clusters (k=7) that we wish to define, as we already know the number of market segments that we wish to analyse.



```
from sklearn.cluster import KMeans
km = KMeans (n_clusters=7,random_state=42)
km.fit(X)

identified_clusters = km.fit_predict(X)
identified_clusters
```

14.3 Market Segment Clustering Before Applying K-Means Algorithm

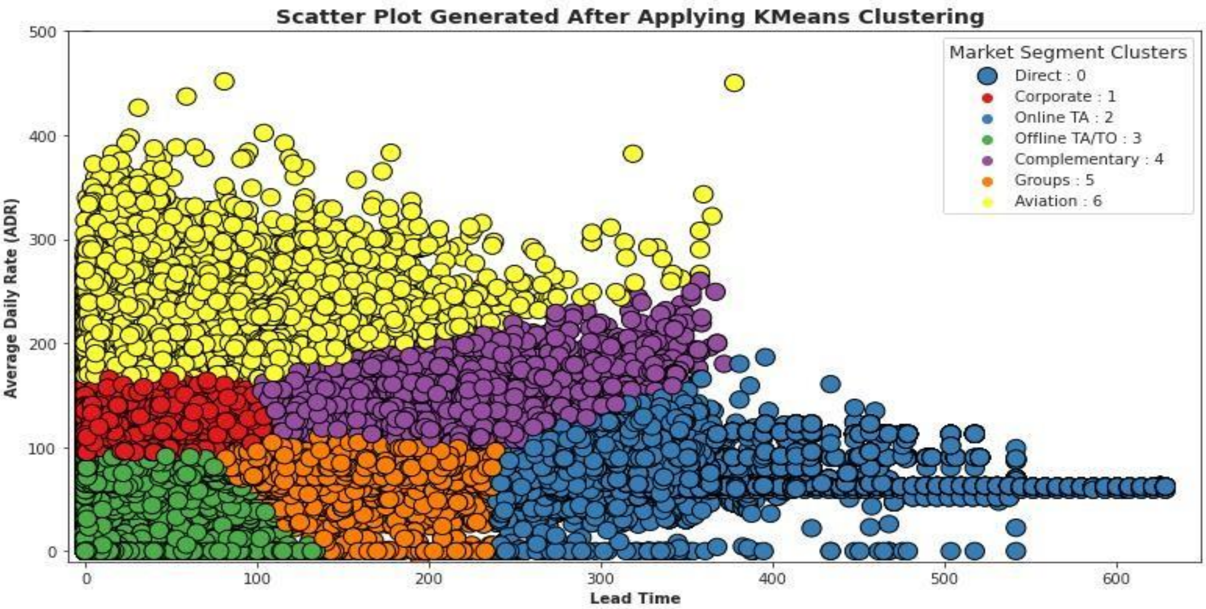
Scatter plot of the Actual labels



14.4 Market Segment Clustering After Applying K-Means Algorithm

From the Cluster Analysis, it is observed that customers with the lowest lead time and the highest ADR are deemed to be the most profitable.

A customer with a high ADR and a low lead time are ideal, as it means that the customer is paying a high daily rate which means a greater profit for the hotel, while a low lead time means that the customer pays for their booking quicker which increases cash flow for the hotel.

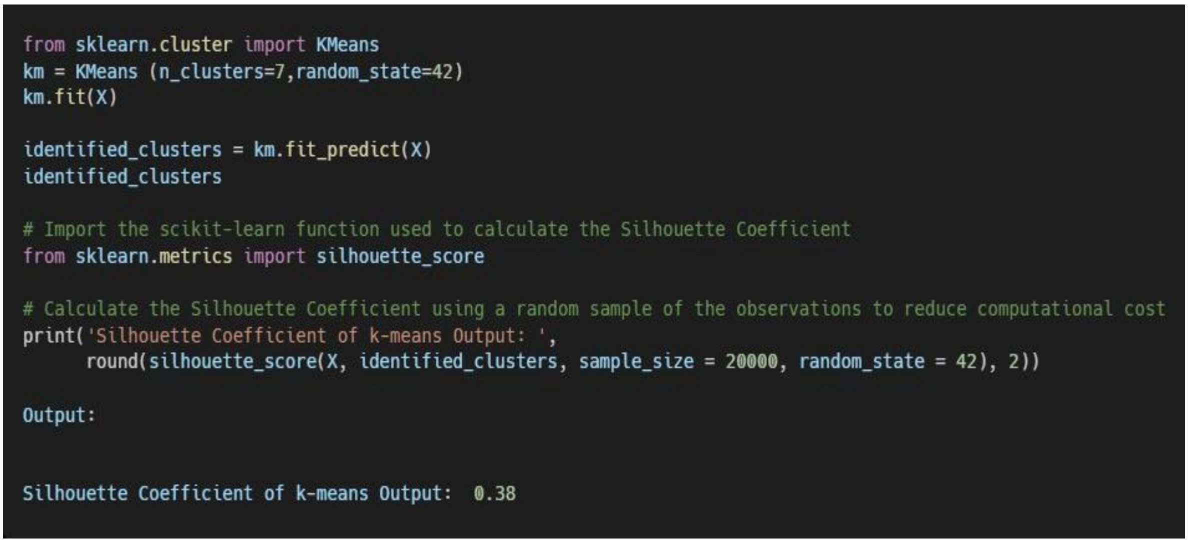


For instance, we already know which customers belong to which market segment. In this regard, generating a k-means clustering algorithm to predict does not serve much use. Rather, the point of running this algorithm is to get a quick visual of what types of customers are mostprofitable.

15 Model Evaluation – Clustering

As the clusters were not predefined, i.e., none of the observations was initially labelled as belonging to a specific cluster, there is no straightforward way to test or cross-validate the k- means output. Furthermore, the usefulness of the clusters determined by k-means is largely dependent on the use case for the algorithm. However, the k-means output can still be quantitatively evaluated by rating the degree of intra-cluster density and inter-cluster separation with a metric such as the Silhouette Coefficient.

The Silhouette Coefficient is a measure of how similar the observation is to the other observations in its cluster and how dissimilar it is to the observations in other clusters. The coefficient ranges from -1 to 1 with values near -1 indicating highly overlapping clusters and values near 1 indicating highly distinct clusters.



```
from sklearn.cluster import KMeans
km = KMeans (n_clusters=7,random_state=42)
km.fit(X)

identified_clusters = km.fit_predict(X)
identified_clusters

# Import the scikit-learn function used to calculate the Silhouette Coefficient
from sklearn.metrics import silhouette_score

# Calculate the Silhouette Coefficient using a random sample of the observations to reduce computational cost
print('Silhouette Coefficient of k-means Output: ',
      round(silhouette_score(X, identified_clusters, sample_size = 20000, random_state = 42), 2))

Output:


Silhouette Coefficient of k-means Output:  0.38
```

The Silhouette Coefficient of **0.38** indicates that the k-means algorithm was successful at grouping the hotel bookings into 7 clusters (K= 7).

Depending on how a hotel would utilize these clusters, the k-means output could still providebusiness value despite the lack of clear separation between the clusters.

16 Model Deployment

After building a model and evaluating it, the final stage of the machine learning lifecycle is todeploy the ML model. A Machine Learning model is not particularly useful unless the customer can access its results.

ML model should be scalable and accessible to the business users/Customers by creating Web Applications and deploying the model on the cloud using Streamlit, AWS, Google Cloud, Heroku cloud platforms.

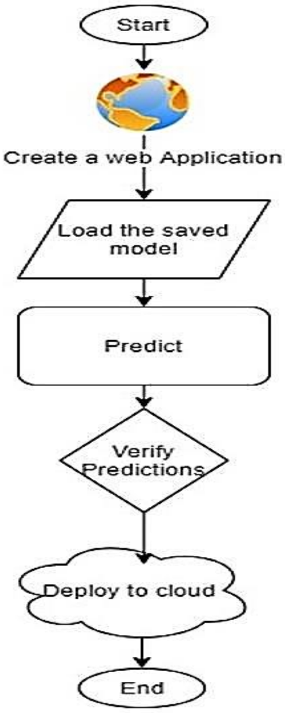
One of the most prevalent misunderstandings and mistakes for a failed ML project is spending a significant amount of time optimizing the ML model. Instead, teams that have completed asuccessful machine learning project devote time to gathering data, developing efficient datapipelines to reduce training and constructing dependable model serving infrastructure.

16.1 Overview of Model Deployment Using Stream_lit

To build the Web Application, I will use the 10 most important features that help predict theHotel Booking Cancelation from the guests since it would be a pain for a front-end user to fillall 23 features on the web app.

Once the training is completed, we need to expose the trained model as an API for the user to consume it. For prediction, the saved pickle file and model is loaded first and then the predictions are made using it. If the web app works fine, the same app is deployed to the cloud platform.

ML Web Application Deployment Flow chart

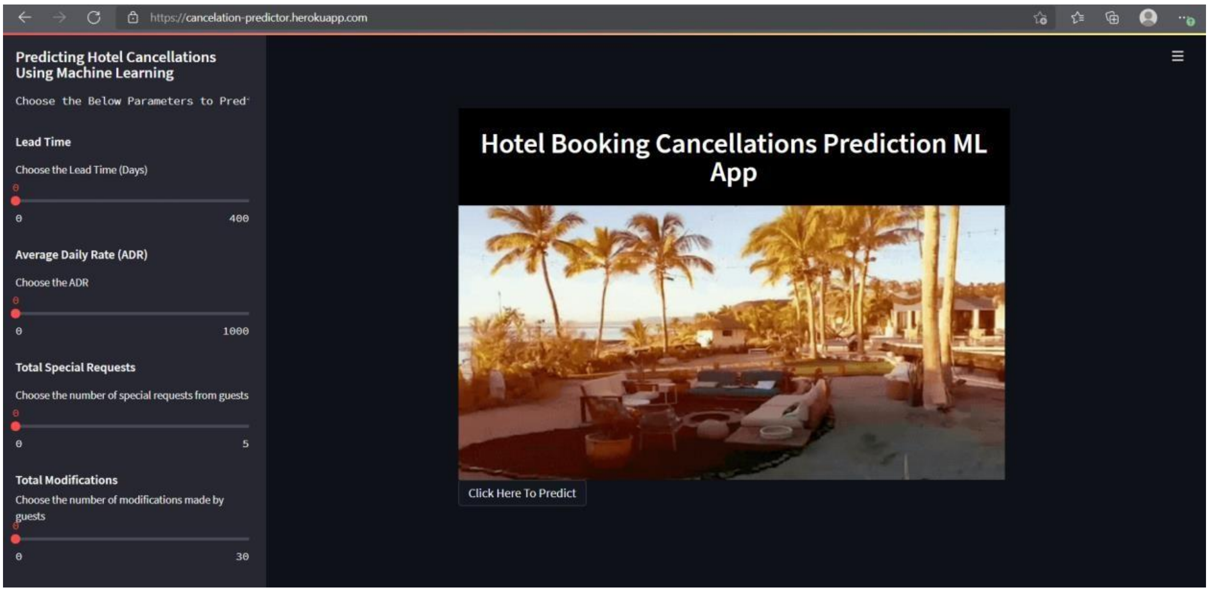


16.2 ML Web Application Hosted on streamlit cloud

I have deployed Machine Learning Model on streamlit cloud using Python and Streamlit. Once we have deployed it on the cloud, we have successfully built a Data Science product that can be used by the Business User/ Customer.

Link to the ML Web App:

16.3 Snapshot of the ML Web Application



Predicting Hotel Cancellations Using Machine Learning

Choose the Below Parameters to Pred

Lead Time

Choose the Lead Time (Days)

0

0 400

Average Daily Rate (ADR)

Choose the ADR

0

0 1000

Total Special Requests

Choose the number of special requests from guests

0

0 5

Total Modifications

Choose the number of modifications made by guests

0

0 30

Hotel Booking Cancellations Prediction ML App

17 Recommendations

- Set Non-refundable Rates, collect deposits, and implement more rigid cancellationpolicies.
- Using Advanced Purchase Rates with varying Lead Time windows
- Encourage Direct bookings by offering special discounts
- Hotels should consider the total number of special requests from guests to reduce thepossibility of cancellations by improving customer service.
- Monitor where the cancellations are coming from such as Market Segment, distribution channels.

18 Conclusion

Due to the COVID-19 pandemic, we are currently seeing rapid changes in customer behavior, however is too early to make predictions, we already saw that the hotels are lowering the window of days that a customer can cancel, as a way of encouraging them to book more.

From Hotel Booking Demand Dataset, we have observed the importance of Machine Learning which helps in maximizing hotel revenue and optimizing business performance. So now is theperfect time for the industry to start investing in this area and particularly in AI and Machinelearning.

19 References

[1] Information Age (2016) “Machine learning and artificial intelligence technology: revolutionising how hotel rooms are priced”

http://www.information-age.com/machine-learning-hospitality-123462758/ October 2016

[2] Altexsoft (2017) “Data Science and AI in the Travel Industry: 9 Real-Life Use Cases”

https://www.altexsoft.com/blog/datascience/data-science-andai-in-the-travel-industry-9-real-life-use-cases/ October 2017

[3] Quanovo (2017) “Machine Learning Technologies for The Hospitality Industry”

http://www.quanovo.com/whitepapers/machine-learningtechnologies-for-the-hospitality-industry.pdf 2017

[4] Forbes (2017) “The Value of Real-Time Data Analytics”

https://www.forbes.com/sites/forbestechcouncil/2017/08/08/the-value-of-real-time-dataanalytics/#64c971151220 August 2017

[5] Pegasus https://www.pegs.com/

https://www.pegs.com/blog/why-and-how-you-should-apply-the-netflix-model-to-your-hotel/

[6] Data Science Process Alliance https://www.datascience-pm.com/crisp-dm-2/

[7] D-edge Hospitality Solutions https://www.d-edge.com/how-online-hotel-distribution-is-changing-in-europe/