HOTEL RESERVATIONS PREDICTION

Muhammad Randa Yandika

Data Science Portofolio

PROJECT OUTLINE

EDA

DATA VISUALIZATION

DATA PREPROCESSING

DATA MODELLING

ABOUT DATASET

The online hotel reservation channels have dramatically changed booking possibilities and customers' behavior. A significant number of hotel reservations are called-off due to cancellations or no-shows. The typical reasons for cancellations include change of plans, scheduling conflicts, etc. This is often made easier by the option to do so free of charge or preferably at a low cost which is beneficial to hotel guests but it is a less desirable and possibly revenue-diminishing factor for hotels to deal with.

DATASET FEARUE DETAILS

Booking_ID	unique identifier of each booking
no_of_adults	Number of adults
no_of_children	Number of Children
no_of_weekend_nights	Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel
no_of_week_nights	Number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel
type_of_meal_plan	Type of meal plan booked by the customer
required_car_parking_space	Does the customer require a car parking space? (0 - No, 1- Yes)
room_type_reserved	Type of room reserved by the customer. The values are ciphered (encoded) by INN Hotels.
lead_time	Number of days between the date of booking and the arrival date

arrival_year	Year of arrival date
arrival_month	Month of arrival date
arrival_date	Date of the month
market_segment_type	Market segment designation
repeated_guest	Is the customer a repeated guest? (0 - No, 1- Yes)
no_of_previous_cancellations	Number of previous bookings that were canceled by the customer prior to the current booking
no_of_previous_bookings_not_canceled	Number of previous bookings not canceled by the customer prior to the current booking
avg_price_per_room	Average price per day of the reservation; prices of the rooms are dynamic. (in euros)
no_of_special_requests	Total number of special requests made by the customer (e.g. high floor, view from the room, etc)
booking_status	Flag indicating if the booking was canceled or not.

EXPLORATORY DATA ANALYSIS (EDA)

FIND DETAIL INFORMATION ABOUT THIS DATASET AND DATA DESCRIBE

```
[4]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36275 entries, 0 to 36274
Data columns (total 19 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Booking_ID                            36275 non-null  object
 1   no_of_adults                          36275 non-null  int64
 2   no_of_children                        36275 non-null  int64
 3   no_of_weekend_nights                  36275 non-null  int64
 4   no_of_week_nights                     36275 non-null  int64
 5   type_of_meal_plan                     36275 non-null  object
 6   required_car_parking_space            36275 non-null  int64
 7   room_type_reserved                    36275 non-null  object
 8   lead_time                             36275 non-null  int64
 9   arrival_year                          36275 non-null  int64
 10  arrival_month                         36275 non-null  int64
 11  arrival_date                          36275 non-null  int64
 12  market_segment_type                   36275 non-null  object
 13  repeated_guest                        36275 non-null  int64
 14  no_of_previous_cancellations          36275 non-null  int64
 15  no_of_previous_bookings_not_canceled  36275 non-null  int64
 16  avg_price_per_room                    36275 non-null  float64
 17  no_of_special_requests                36275 non-null  int64
 18  booking_status                        36275 non-null  object
dtypes: float64(1), int64(13), object(5)
memory usage: 5.3+ MB
```

```
df.describe()
```

	no_of_adults	no_of_children	no_of_weekend_nights	no_of_week_nights	required_car_parking_space	lead_time	arr
count	36275.000000	36275.000000	36275.000000	36275.000000	36275.000000	36275.000000	3627
mean	1.844962	0.105279	0.810724	2.204300	0.030986	85.232557	201
std	0.518715	0.402648	0.870644	1.410905	0.173281	85.930817	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	201
25%	2.000000	0.000000	0.000000	1.000000	0.000000	17.000000	201
50%	2.000000	0.000000	1.000000	2.000000	0.000000	57.000000	201
75%	2.000000	0.000000	2.000000	3.000000	0.000000	126.000000	201
max	4.000000	10.000000	7.000000	17.000000	1.000000	443.000000	201

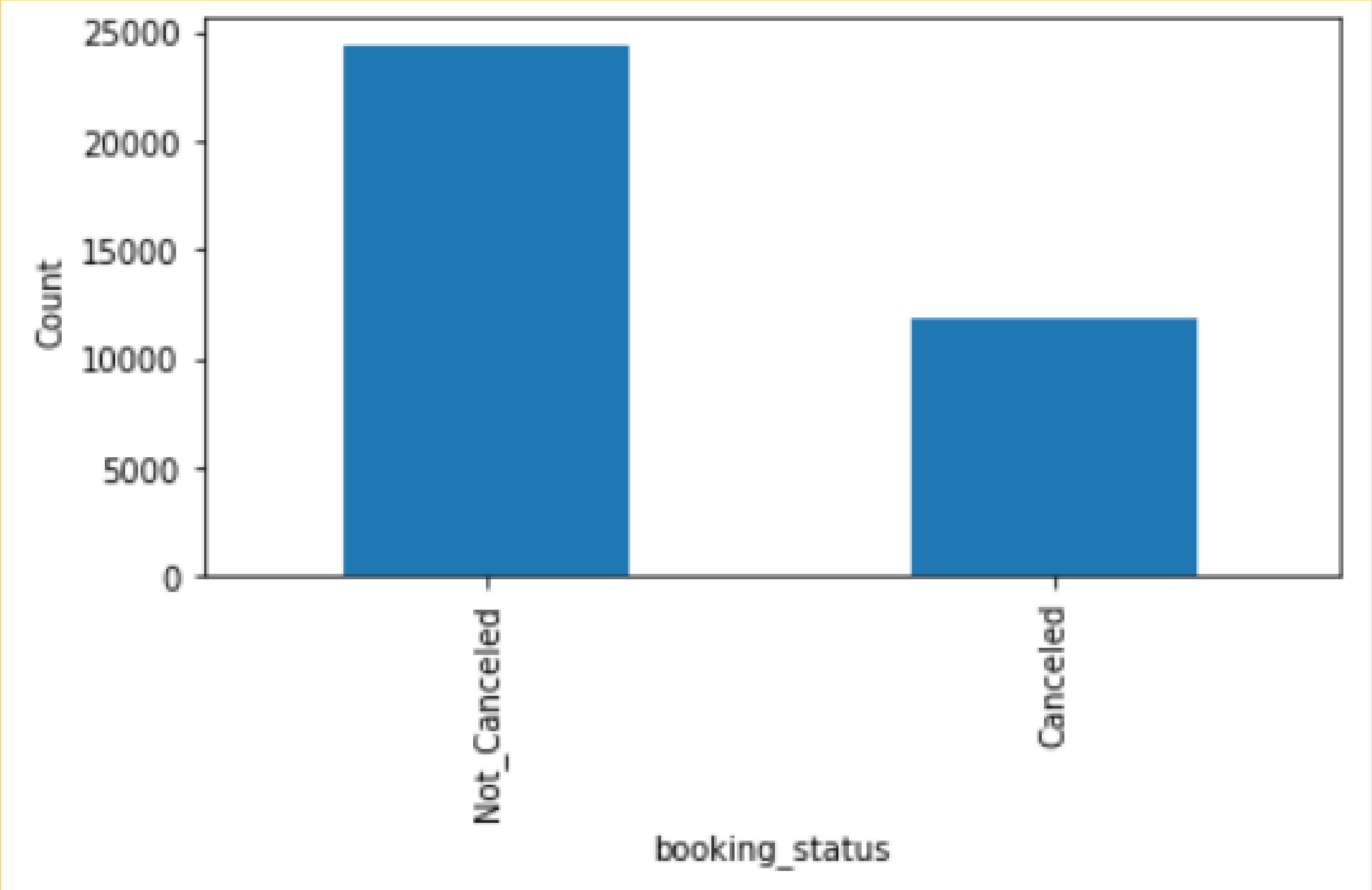
SPLIT DATA BETWEEN NUMERICAL AND CATEGORICAL COLUMN

```
[11]:  numerical =  df.select_dtypes(include=[np.number])
       numerical.columns

[11]: Index(['no_of_adults', 'no_of_children', 'no_of_weekend_nights',
             'no_of_week_nights', 'required_car_parking_space', 'lead_time',
             'arrival_year', 'arrival_month', 'arrival_date', 'repeated_guest',
             'no_of_previous_cancellations', 'no_of_previous_bookings_not_canceled',
             'avg_price_per_room', 'no_of_special_requests'],
            dtype='object')
```

```
[12]:  categorical = df.select_dtypes(exclude=[np.number])
       categorical.columns

[12]: Index(['Booking_ID', 'type_of_meal_plan', 'room_type_reserved',
             'market_segment_type', 'booking_status'],
            dtype='object')
```

DATA VISUALIZATION

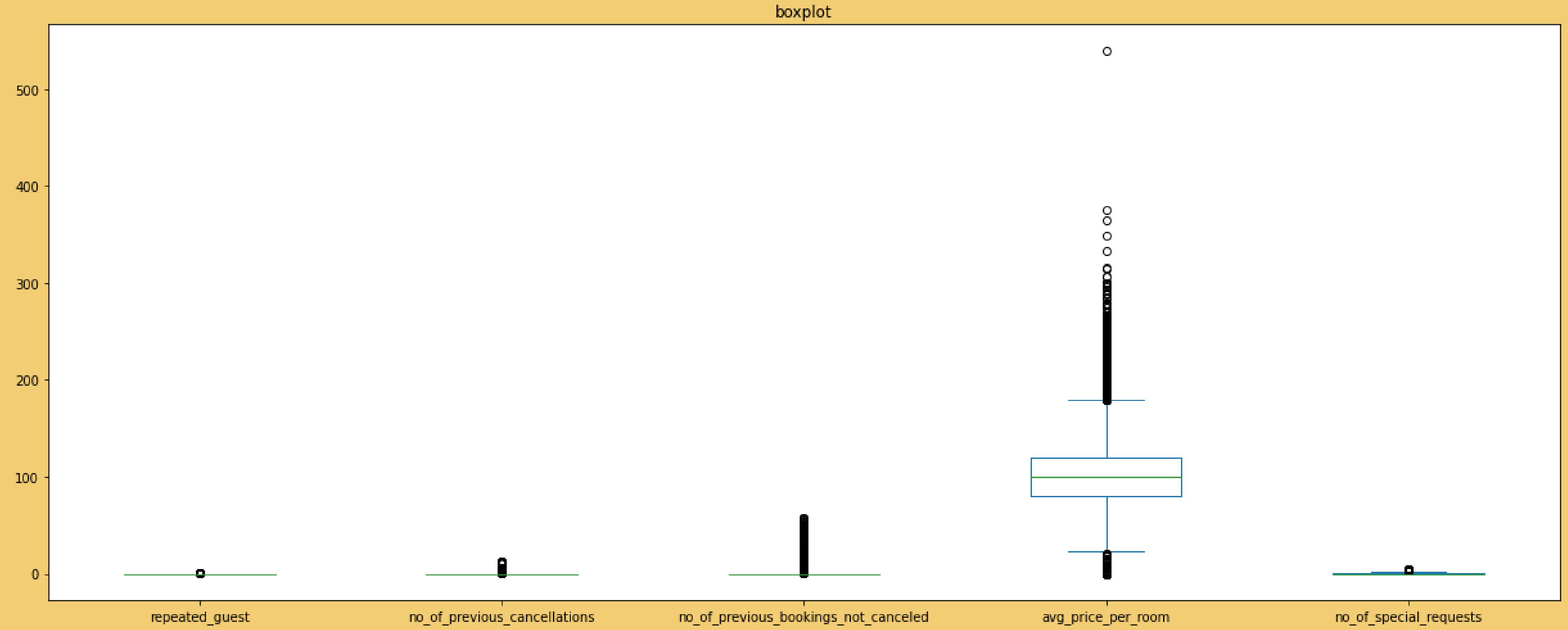
Find the number of hotel visitors who cancel bookings and not by using a bar chart, 24390 customers didn't cancel it and 11885 customer decide to cancel it

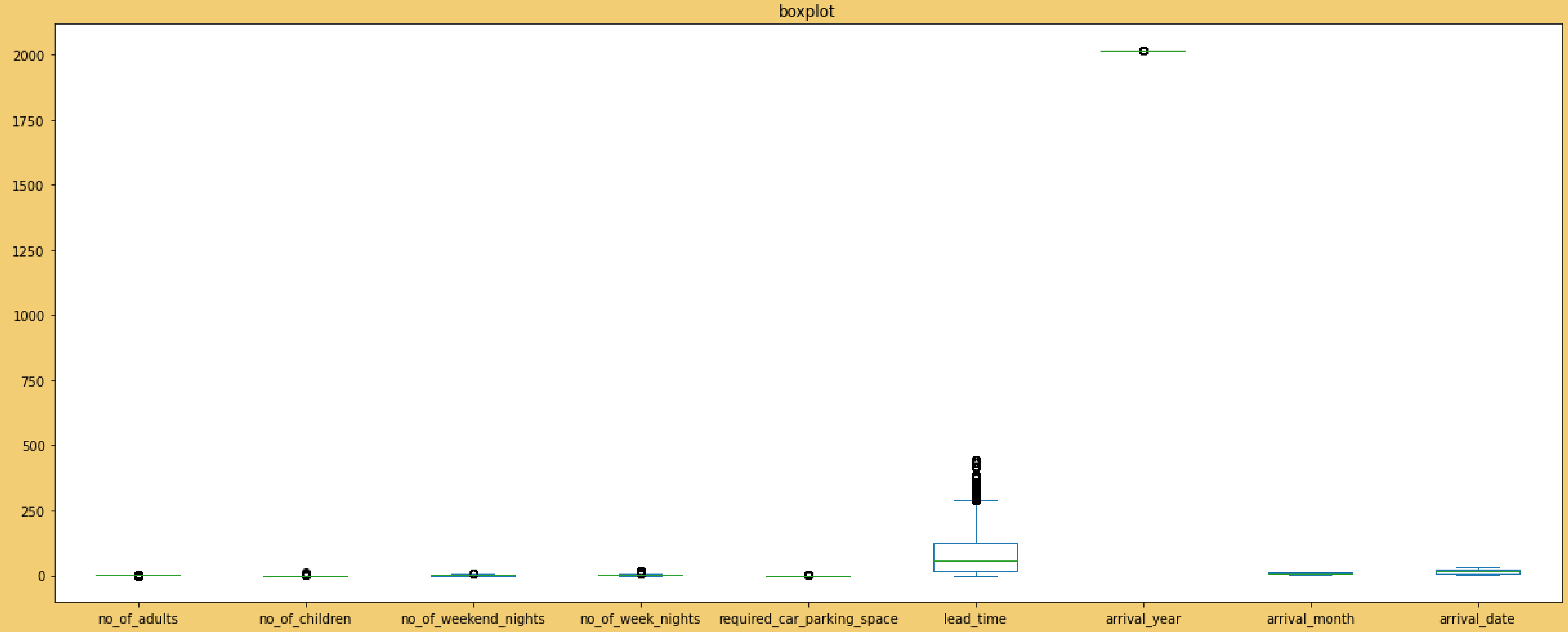



```
Not_Canceled    24390
Canceled        11885
Name: booking_status, dtype: int64
```

DATA VISUALIZATION

Find the outliers in the dataset using boxplot



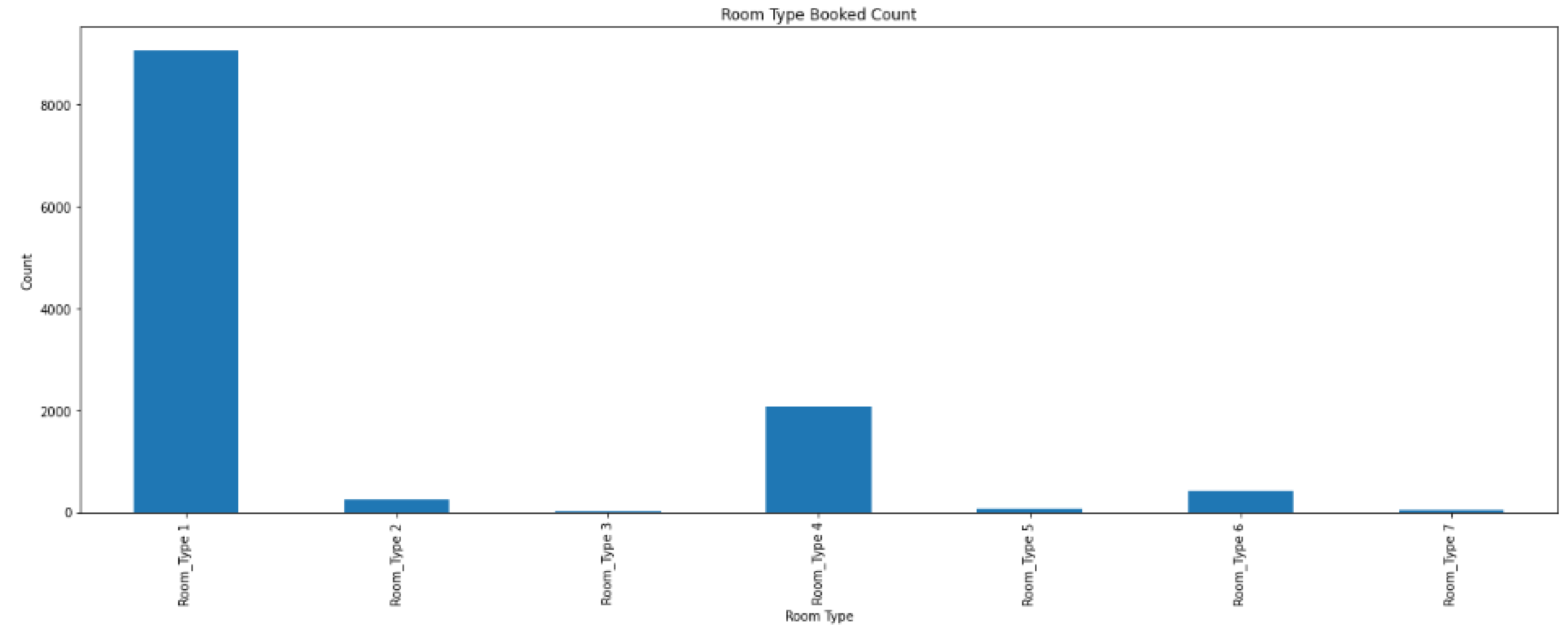


DATA VISUALIZATION

Customer Booking count by month in 2017 and 2018 using bar chart

DATA VISUALIZATION

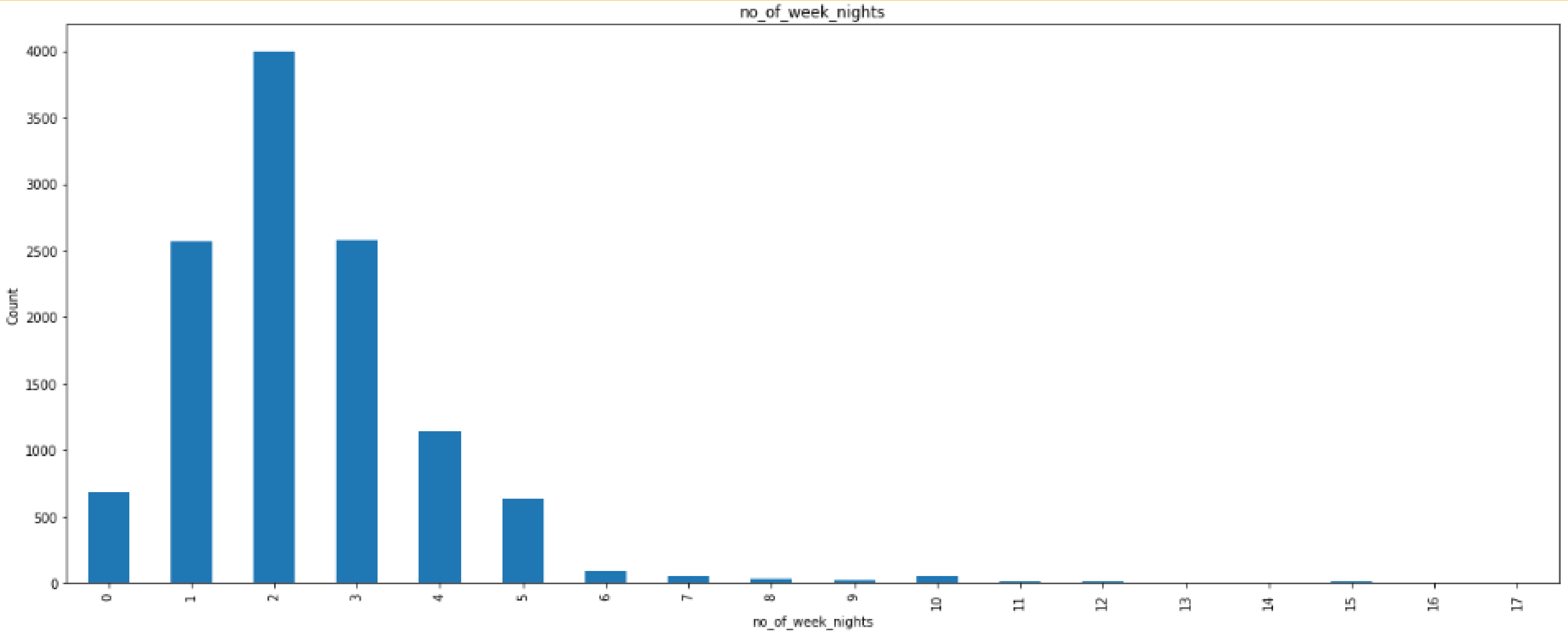
This bar chart displays the amount booked for each type of hotel room




```
Room_Type 1    28130
Room_Type 4     6057
Room_Type 6      966
Room_Type 2      692
Room_Type 5      265
Room_Type 7      158
Room_Type 3        7
Name: room_type_reserved, dtype: int64
```

DATA VISUALIZATION

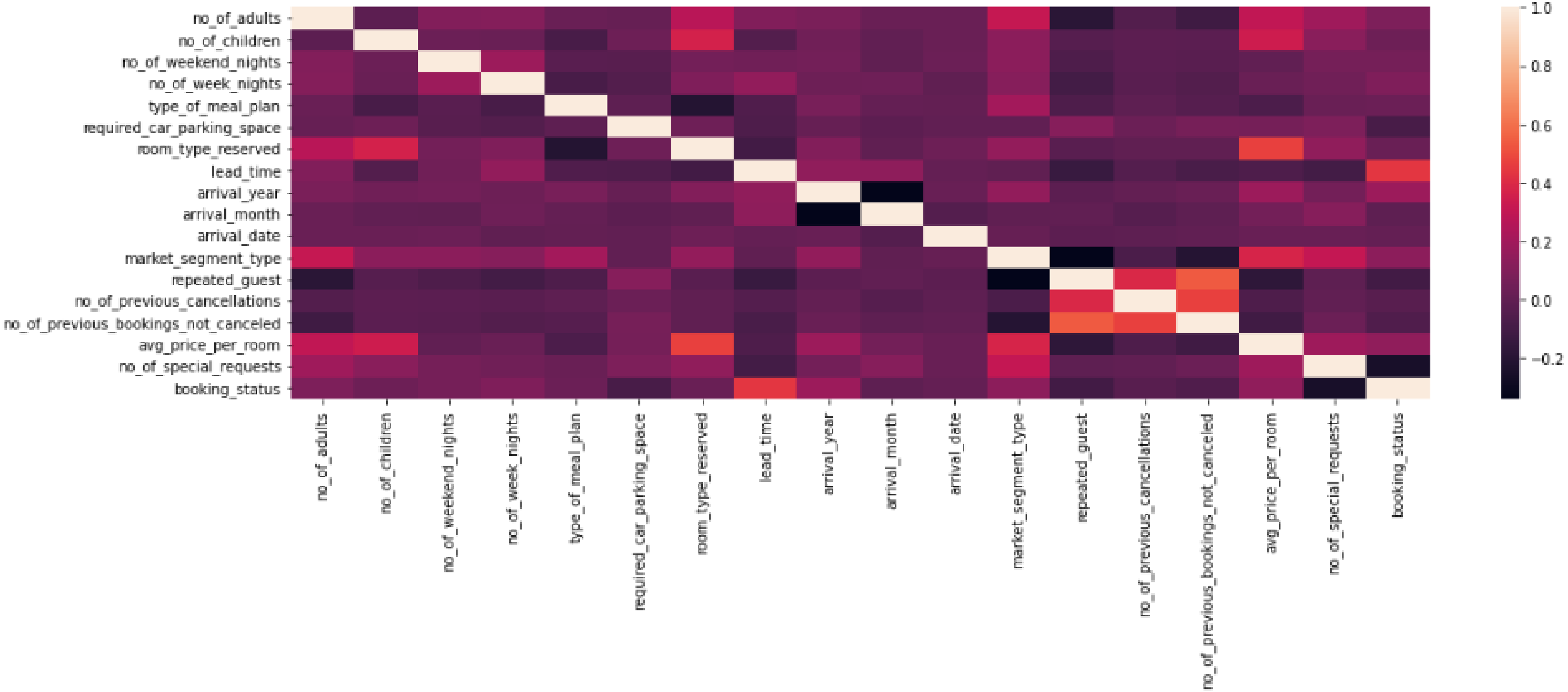
This bar chart displays the amount Number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel



2	11444
1	9488
3	7839
4	2990
0	2387
5	1614
6	189
7	113
10	62
8	62
9	34
11	17
15	10
12	9
14	7
13	5
17	3
16	2

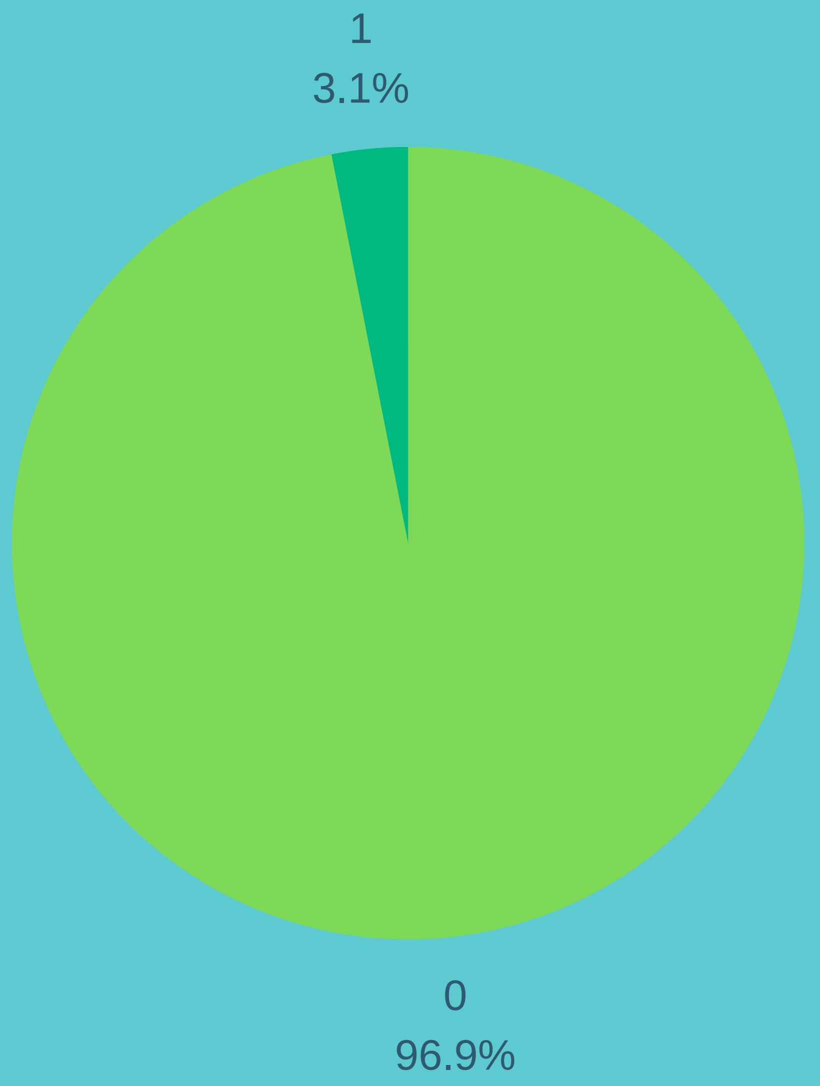
Correlation heatmaps are a type of plot that visualize the strength of relationships between variables.

<AxesSubplot:>



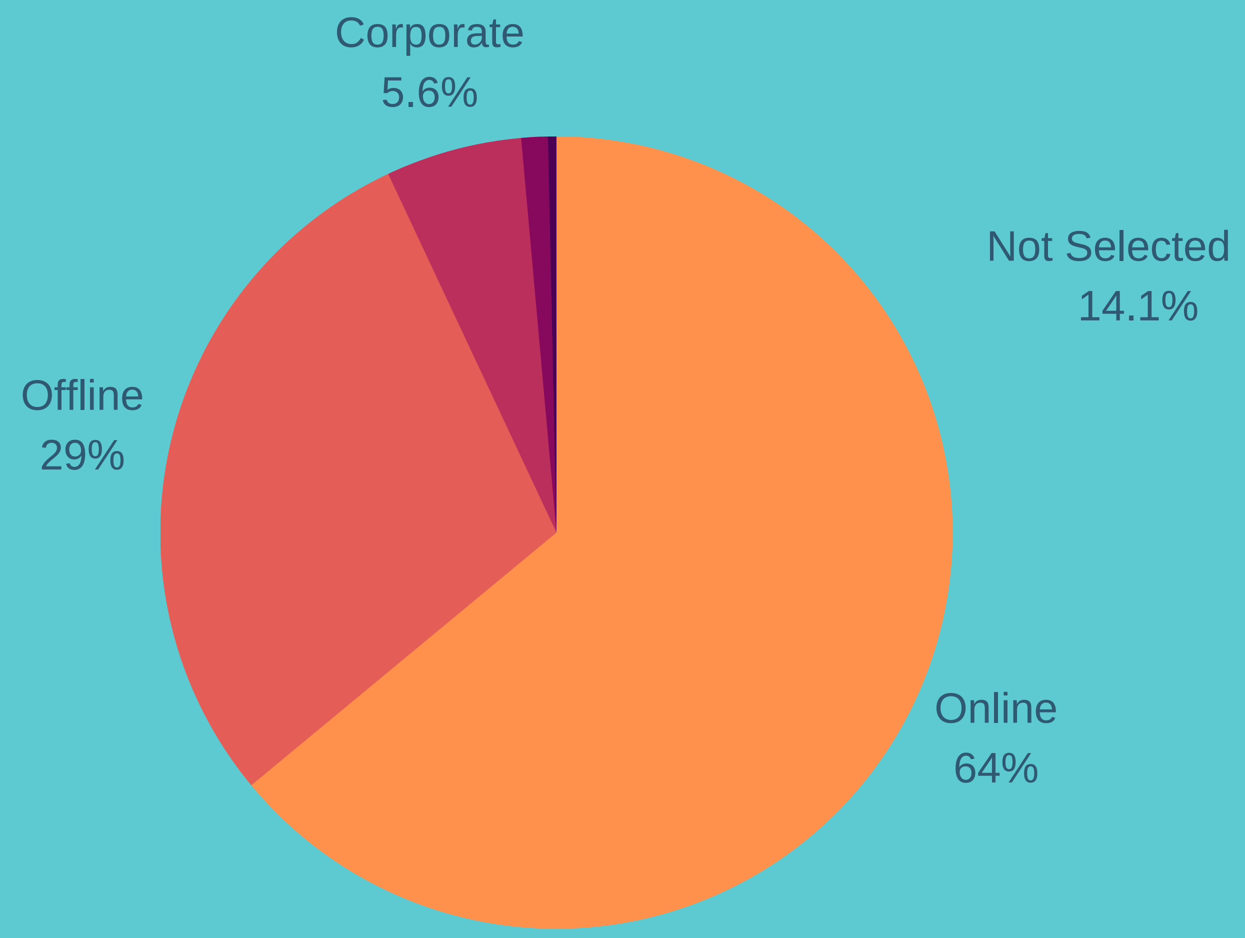
PIE CHART

REQUIRED CAR PARKING SPACE



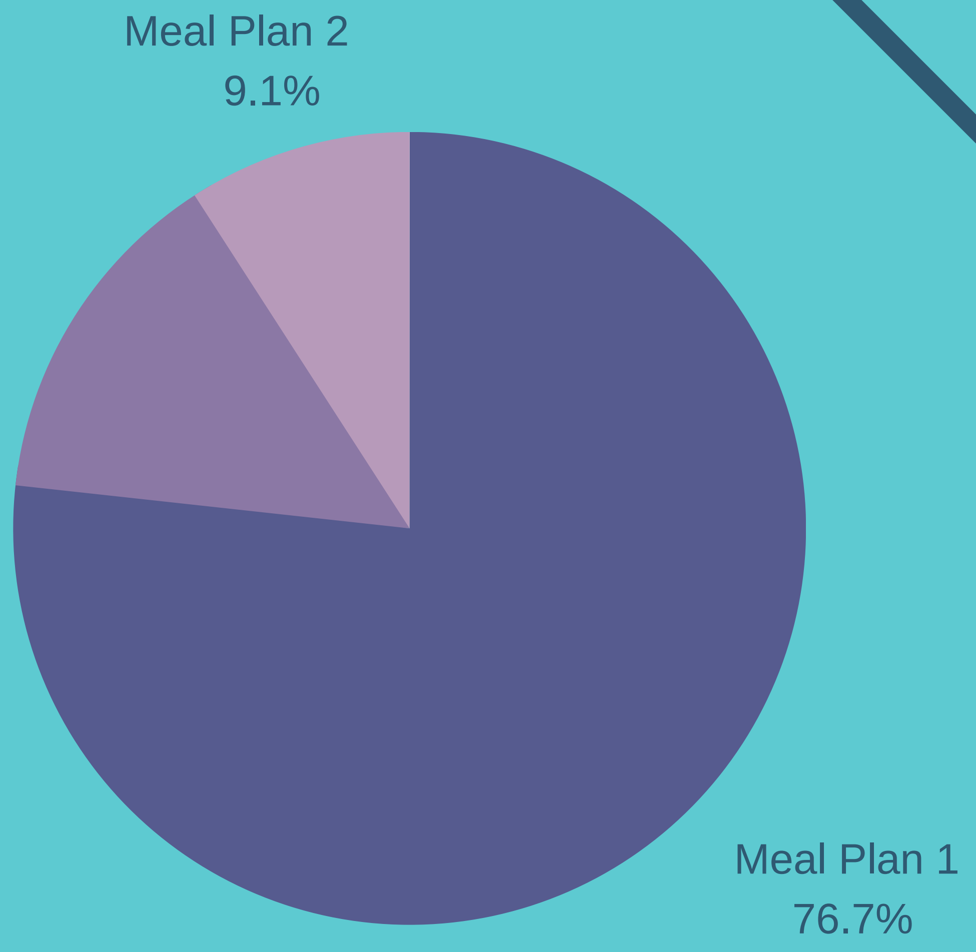
No	35151
Yes	1124

MARKET SEGMENT TYPE



Online	23214
Offline	10528
Corporate	2017
Complementary	391
Aviation	125

TYPE OF MEAL PLAN



Meal Plan 1	27835
Not Selected	5130
Meal Plan 2	3305
Meal Plan 3	5

DATA PREPROCESSING

In this step,

1. Check null in dataset.
2. Check count of unique value.
3. Convert booking_status as target into numerical value.
4. Drop Booking_ID because it's not needed in the next step.

```
df.drop('Booking_ID', axis=1, inplace=True)
```

```
# converting target variable into numerical value
df['booking_status'] = np.where((df['booking_status'] == 'Canceled'),1,0)
```

```
# Checking if any rows are missing any data.
df.isnull().sum()
```

```
Booking_ID                              0
no_of_adults                            0
no_of_children                          0
no_of_weekend_nights                    0
no_of_week_nights                       0
type_of_meal_plan                       0
required_car_parking_space              0
room_type_reserved                      0
lead_time                               0
arrival_year                            0
arrival_month                           0
arrival_date                            0
market_segment_type                     0
repeated_guest                          0
no_of_previous_cancellations            0
no_of_previous_bookings_not_canceled    0
avg_price_per_room                      0
no_of_special_requests                  0
booking_status                          0
dtype: int64
```

```
# Determine count of unique values for each
df.nunique()
```

```
Booking_ID                              36275
no_of_adults                                5
no_of_children                              6
no_of_weekend_nights                        8
no_of_week_nights                          18
type_of_meal_plan                           4
required_car_parking_space                  2
room_type_reserved                          7
lead_time                                 352
arrival_year                                2
arrival_month                              12
arrival_date                               31
market_segment_type                         5
repeated_guest                              2
no_of_previous_cancellations                9
no_of_previous_bookings_not_canceled       59
avg_price_per_room                       3930
no_of_special_requests                      6
booking_status                              2
dtype: int64
```

DATA PREPROCESSING

Then, deleting outliers from the dataset. The column that is filtered has remaining 32675 from 36275 using IQR

and,

```
print(f'Jumlah Baris Sebelum Outlier Dihapus: {len(df)}')
filtered_entries = np.array([True] * len(df))
for col in['lead_time','no_of_previous_bookings_not_canceled',
           'no_of_previous_cancellations','avg_price_per_room']:

    q1=df[col].quantile(0.25)
    q3=df[col].quantile(0.75)
    iqr=q3-q1

    min_IQR = q1 - (1.5 * iqr)
    max_IQR = q3 + (1.5 * iqr)

    filtered_entries=((df[col]>=min_IQR) & (df[col]<=max_IQR)) & filtered_entries
    df=df[filtered_entries]

print(f'Jumlah Baris Sebelum Outlier Dihapus: {len(df)}')
```

```
Jumlah Baris Sebelum Outlier Dihapus: 36275
Jumlah Baris Sebelum Outlier Dihapus: 32675
```

Encoding process to change the categorical feature to numerical feature using One Hot Encoding

```
categorical1 = ['type_of_meal_plan', 'room_type_reserved','market_segment_type']
```

```
for cat in categorical1:
    onehots = pd.get_dummies(df[cat], prefix=cat)
    df = df.join(onehots)
```

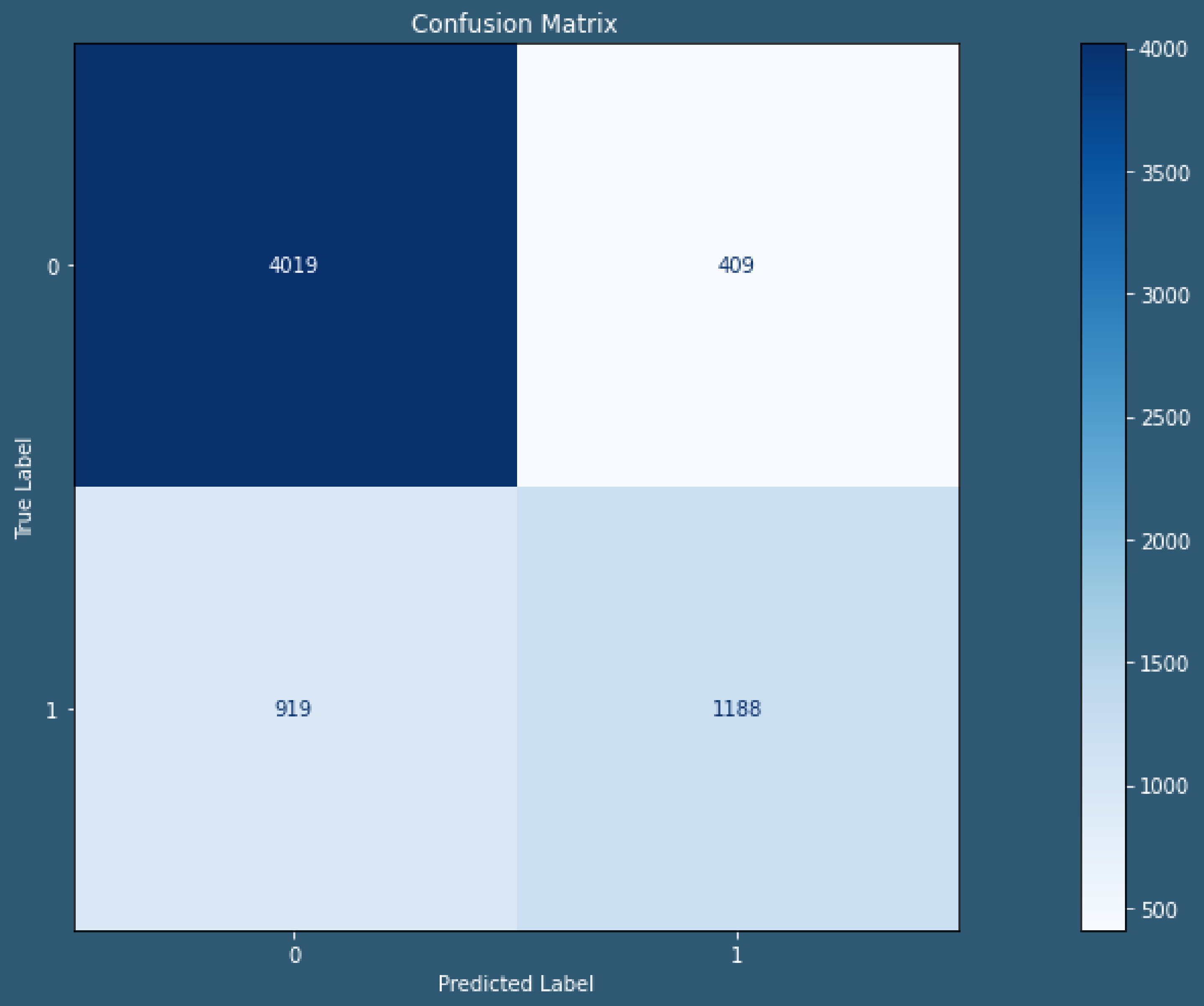
DATA MODELLING

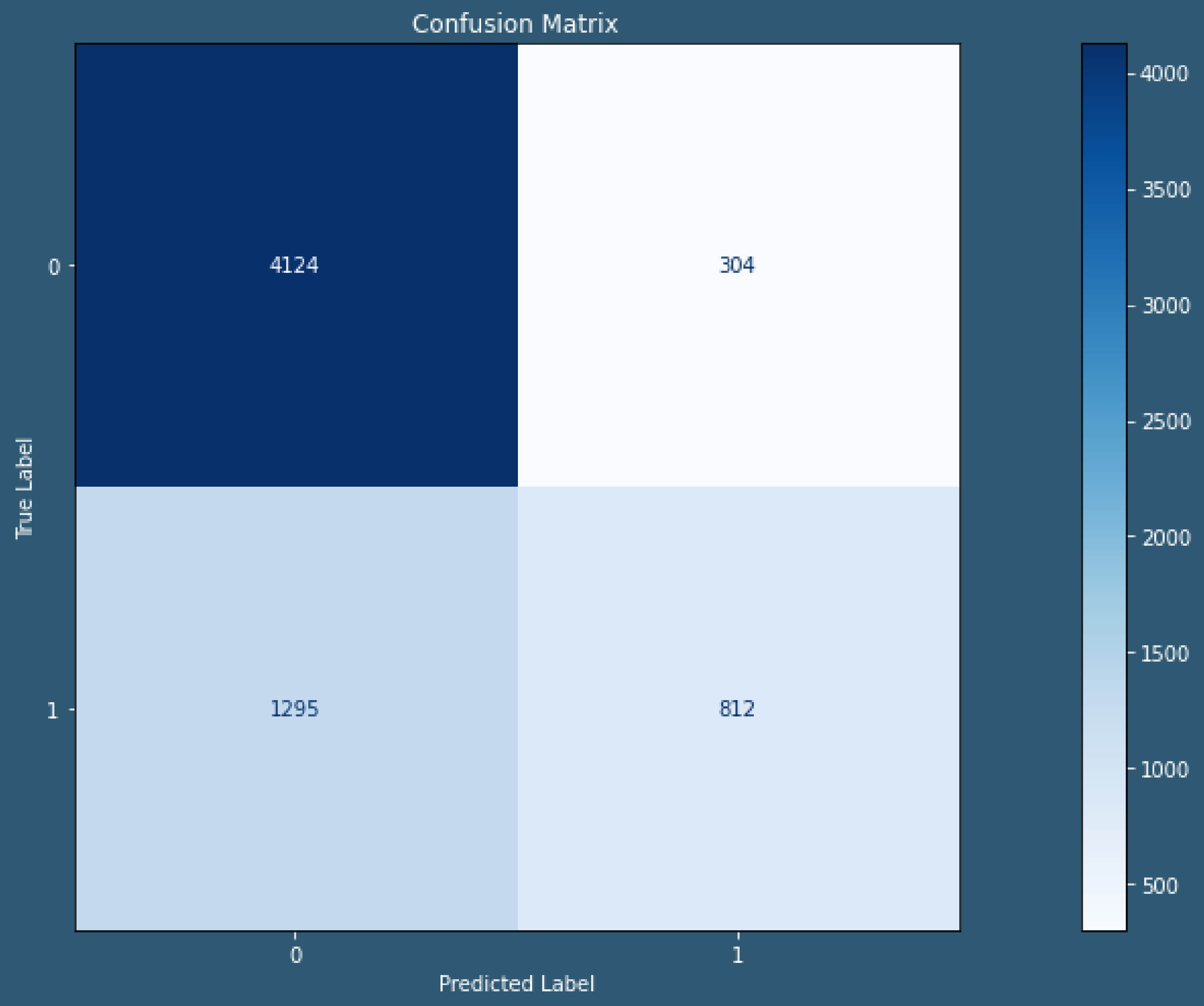
This is result of modelling process using 4 different method with with a data train and data test ratio is 80:20. XGBOOST is model with highest performance compared with other model

KNN	SVM
Accuracy: 79,68%	Accuracy: 75,53%
Recall: 56,4%	Recall: 38,6%
Precision: 74,4%	Precision: 72,8%

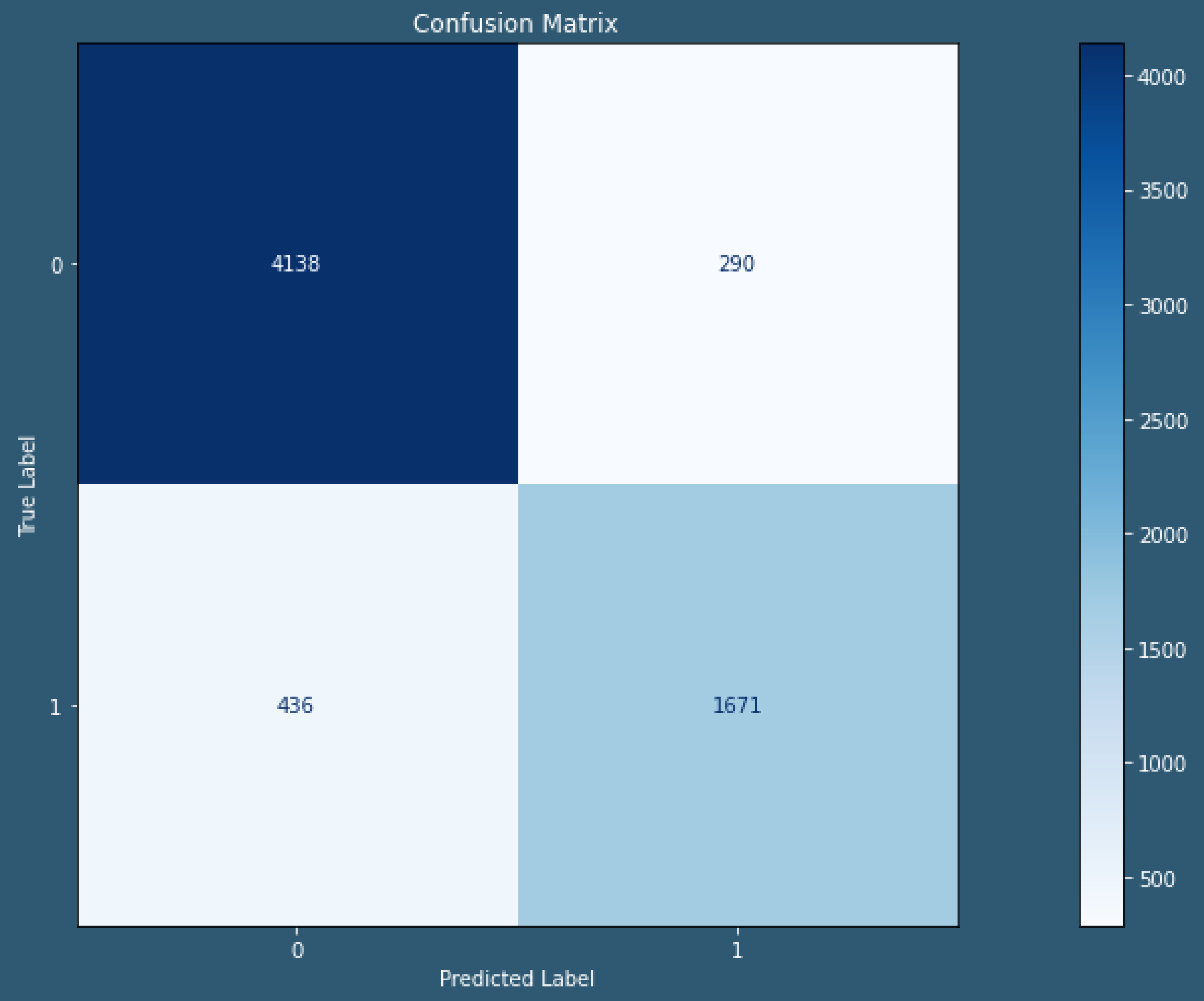
CATBOOST	XGBOOST
Accuracy: 83,85%	Accuracy: 88,89%
Recall: 64,9%	Recall: 79,3%
Precision: 81,1%	Precision: 85,2%

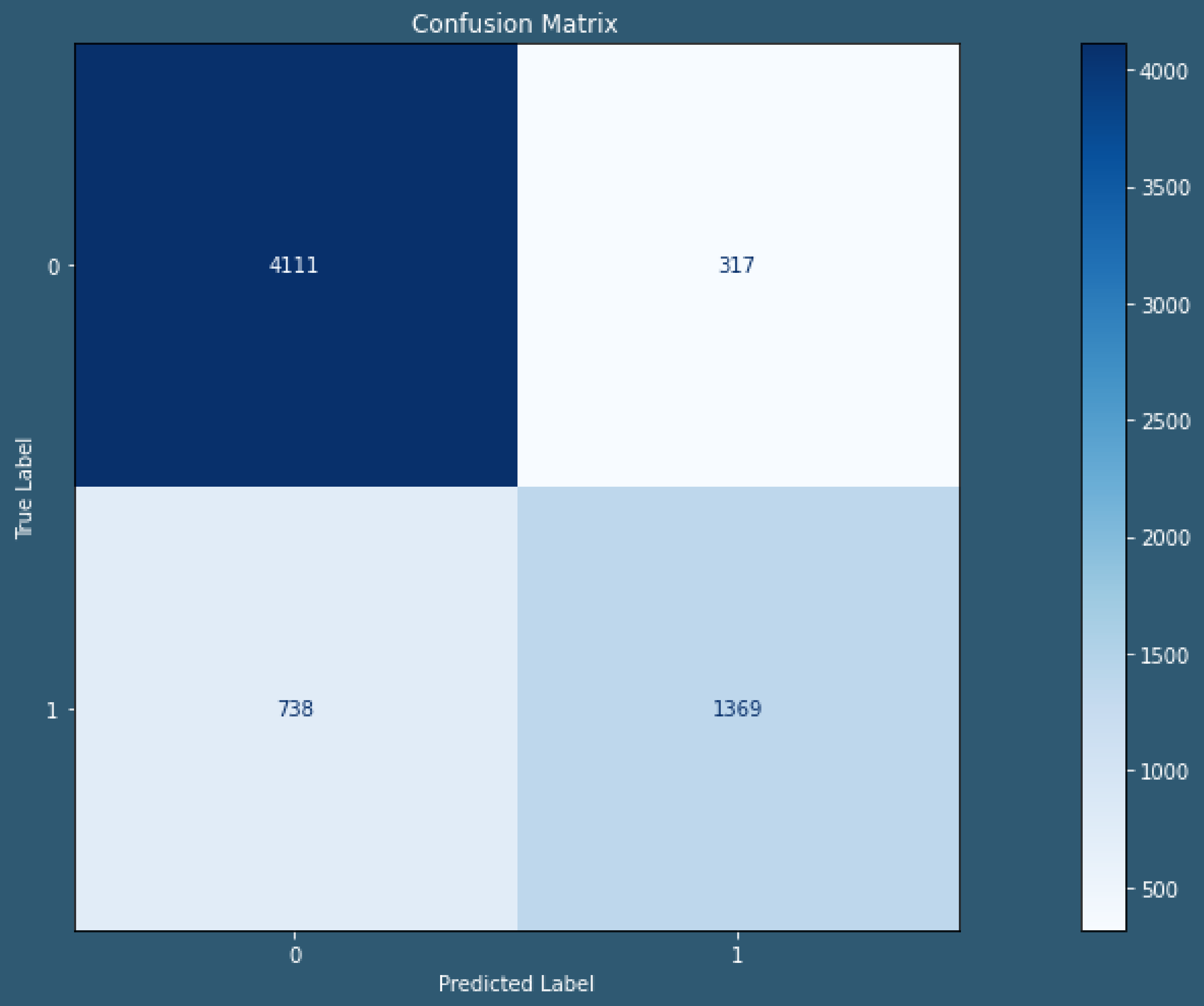
KNN & SVM MODEL CONFUSION MATRIX





XGBOOST & CATBOOST MODEL CONFUSION MATRIX





THANK YOU

GITHUB

https://bit.ly/Randa_Portofolio

LINKEDIN

inkedin.com/in/muhammad-randa-yandika/