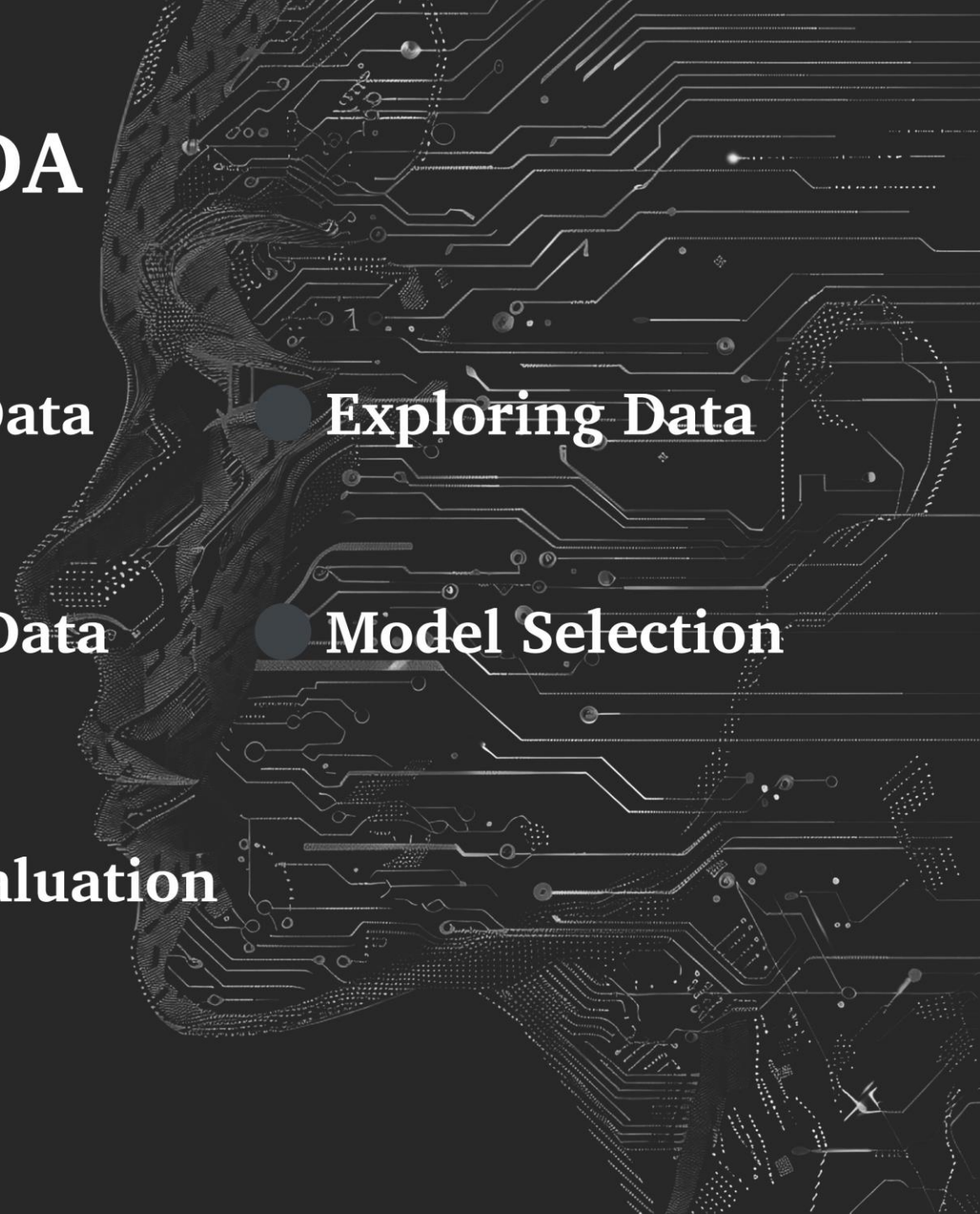
HOTEL RESERVATION

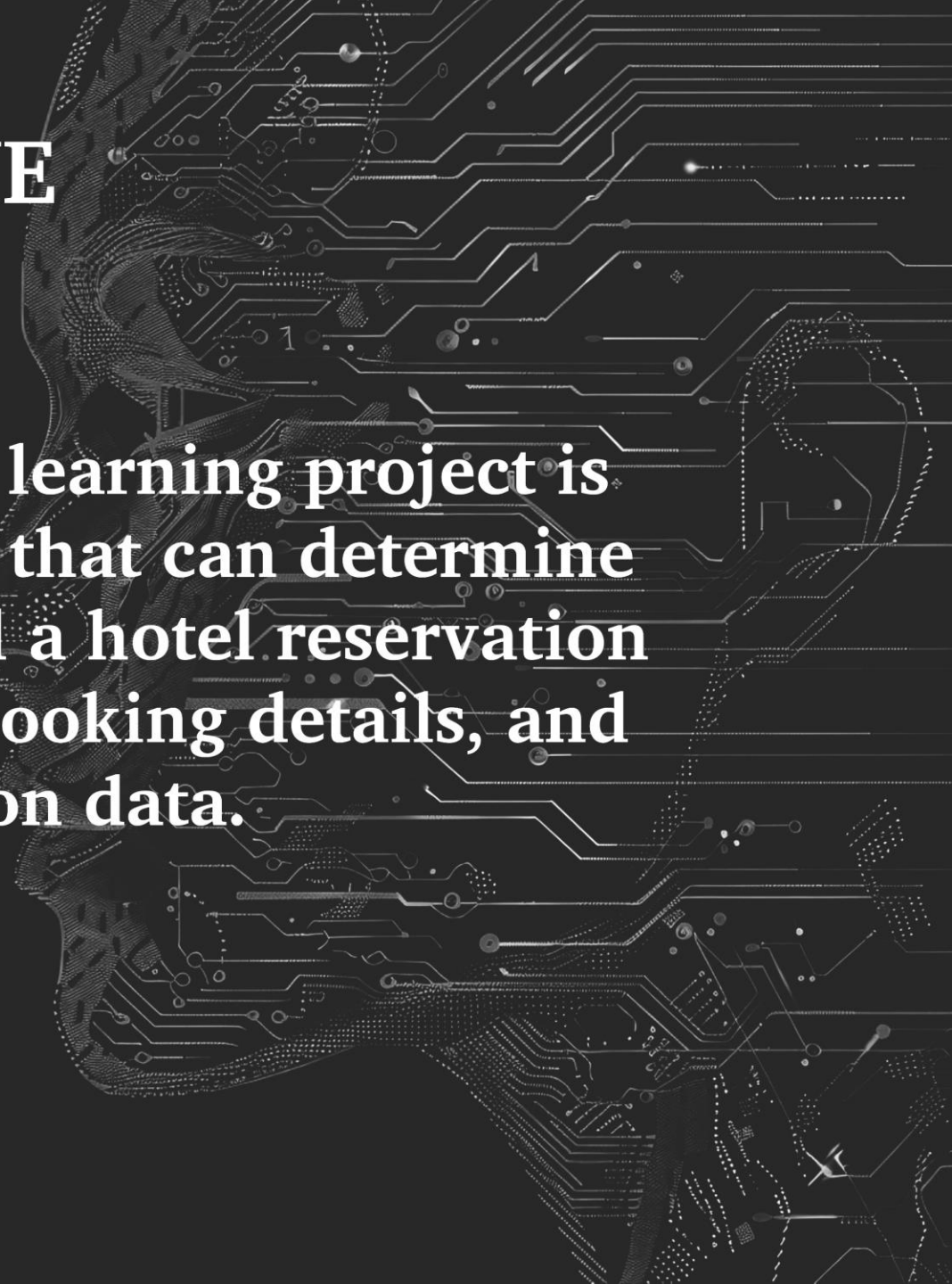
BY - JITHIN JAYACHANDRAN

AGENDA

- Objectives
- Loading Data
- Exploring Data
- Encoding
- Splitting Data
- Model Selection
- Model Training
- Model Evaluation



OBJECTIVE



The objective of this machine learning project is to develop a predictive model that can determine whether a customer will cancel a hotel reservation based on their demographic, booking details, and previous interaction data.

LOADING DATA

```
IMPORT PANDAS AS PD

# LOAD THE DATASET
DF = PD.READ_CSV('/CONTENT/HOTEL RESERVATIONS.CSV')

# CHANGING THE INDEX TO BOOKING ID COLUMN
DF.SET_INDEX('BOOKING_ID', INPLACE=TRUE)
```

Booking_ID	no_of_adults	no_of_children	no_of_weekend_nights	no_of_week_nights	type_of_meal_plan	required_car_parking_space	room_type_reserved	lead_time	arrival_year
INN00001	2	0	1	2	Meal Plan 1	0	Room_Type 1	224	2017
INN00002	2	0	2	3	Not Selected	0	Room_Type 1	5	2018
INN00003	1	0	2	1	Meal Plan 1	0	Room_Type 1	1	2018
INN00004	2	0	0	2	Meal Plan 1	0	Room_Type 1	211	2018
INN00005	2	0	1	1	Not Selected	0	Room_Type 1	48	2018

arrival_month	arrival_date	market_segment_type	repeated_guest	no_of_previous_cancellations	no_of_previous_bookings_not_canceled	avg_price_per_room	no_of_special_requests	booking_status
10	2	Offline	0	0	0	65.00	0	Not_Canceled
11	6	Online	0	0	0	106.68	1	Not_Canceled
2	28	Online	0	0	0	60.00	0	Canceled
5	20	Online	0	0	0	100.00	0	Canceled
4	11	Online	0	0	0	94.50	0	Canceled

EXPLORING DATA

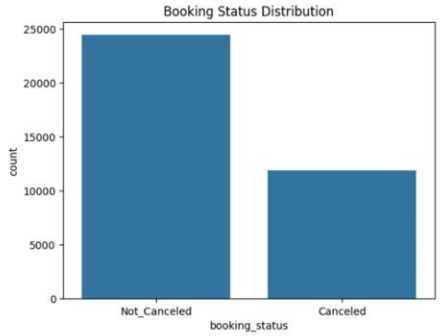
```
# Basic information about the dataset
df.info()
```

```
<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
```

```
# Checking the distribution of the target variable

import seaborn as sns
import matplotlib.pyplot as plt

sns.countplot(x='booking_status', data=df)
plt.title('Booking Status Distribution')
plt.show()
```

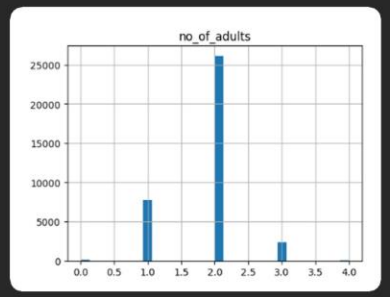


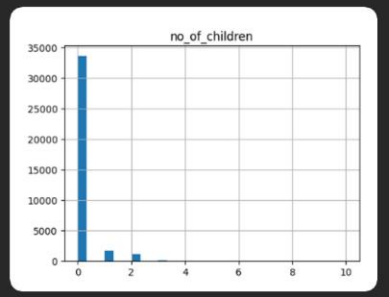
[3]
```
# Statistical summary of the dataset
df.describe()
```

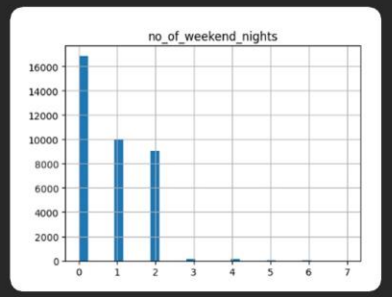
	no_of_adults	no_of_children	no_of_weekend_nights	no_of_week_nights	required_car_parking_space	lead_time	arrival_year	arrival_month	arrival_date	repeated_guest	no_of_previous_cancellations	no_of_previous_bookings_not_canceled	avg_price_per_room	no_of_special_requests
count	36275.000000	36275.000000	36275.000000	36275.000000	36275.000000	36275.000000	36275.000000	36275.000000	36275.000000	36275.000000	36275.000000	36275.000000	36275.000000	36275.000000
mean	1.844962	0.105279	0.810724	2.204300	0.030986	85.232557	2017.820427	7.423653	15.596995	0.025637	0.023349	0.153411	103.423539	0.619655
std	0.518715	0.402648	0.870644	1.410905	0.173281	85.930817	0.383836	3.069894	8.740447	0.158053	0.368331	1.754171	35.089424	0.786236
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	2017.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	2.000000	0.000000	0.000000	1.000000	0.000000	17.000000	2018.000000	5.000000	8.000000	0.000000	0.000000	0.000000	80.300000	0.000000
50%	2.000000	0.000000	1.000000	2.000000	0.000000	57.000000	2018.000000	8.000000	16.000000	0.000000	0.000000	0.000000	99.450000	0.000000
75%	2.000000	0.000000	2.000000	3.000000	0.000000	126.000000	2018.000000	10.000000	23.000000	0.000000	0.000000	0.000000	120.000000	1.000000
max	4.000000	10.000000	7.000000	17.000000	1.000000	443.000000	2018.000000	12.000000	31.000000	1.000000	13.000000	58.000000	540.000000	5.000000

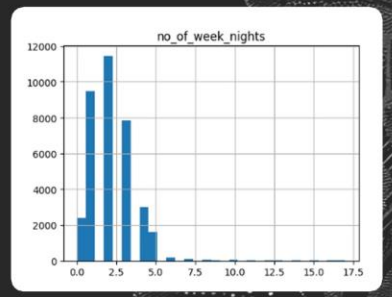
EXPLORING DATA

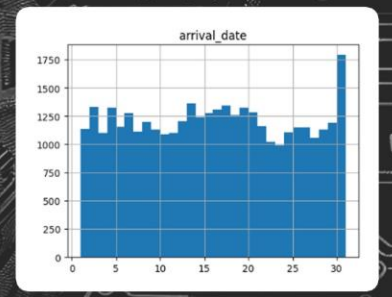
 # Visualizing the distribution of numerical features

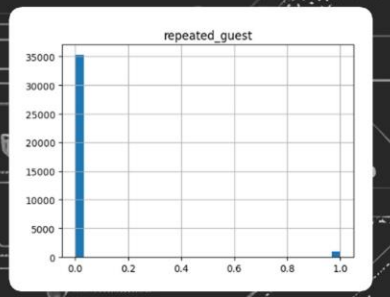


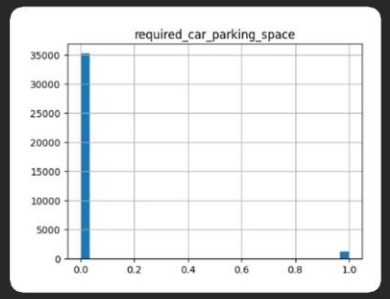


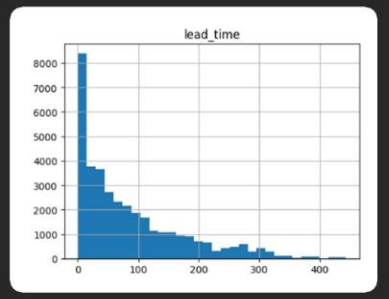


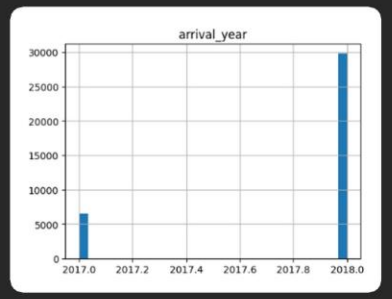


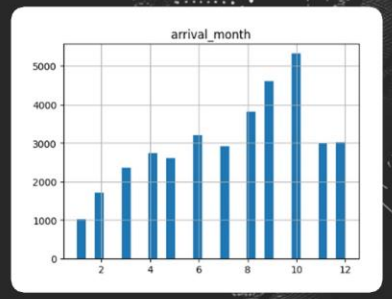


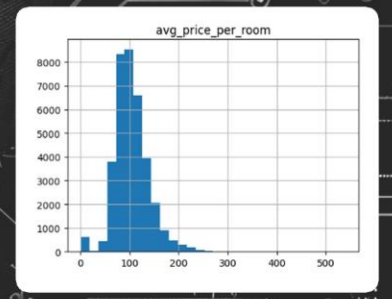


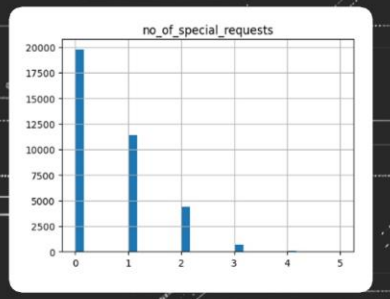


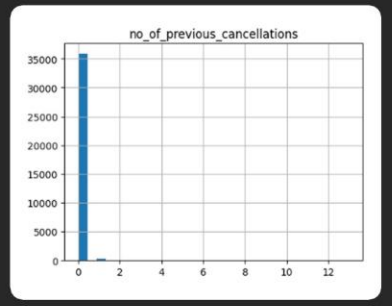


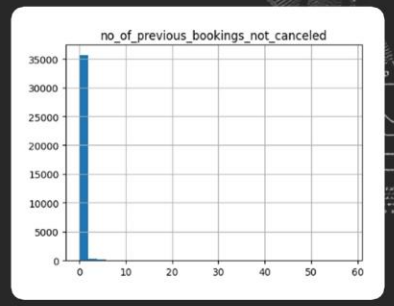






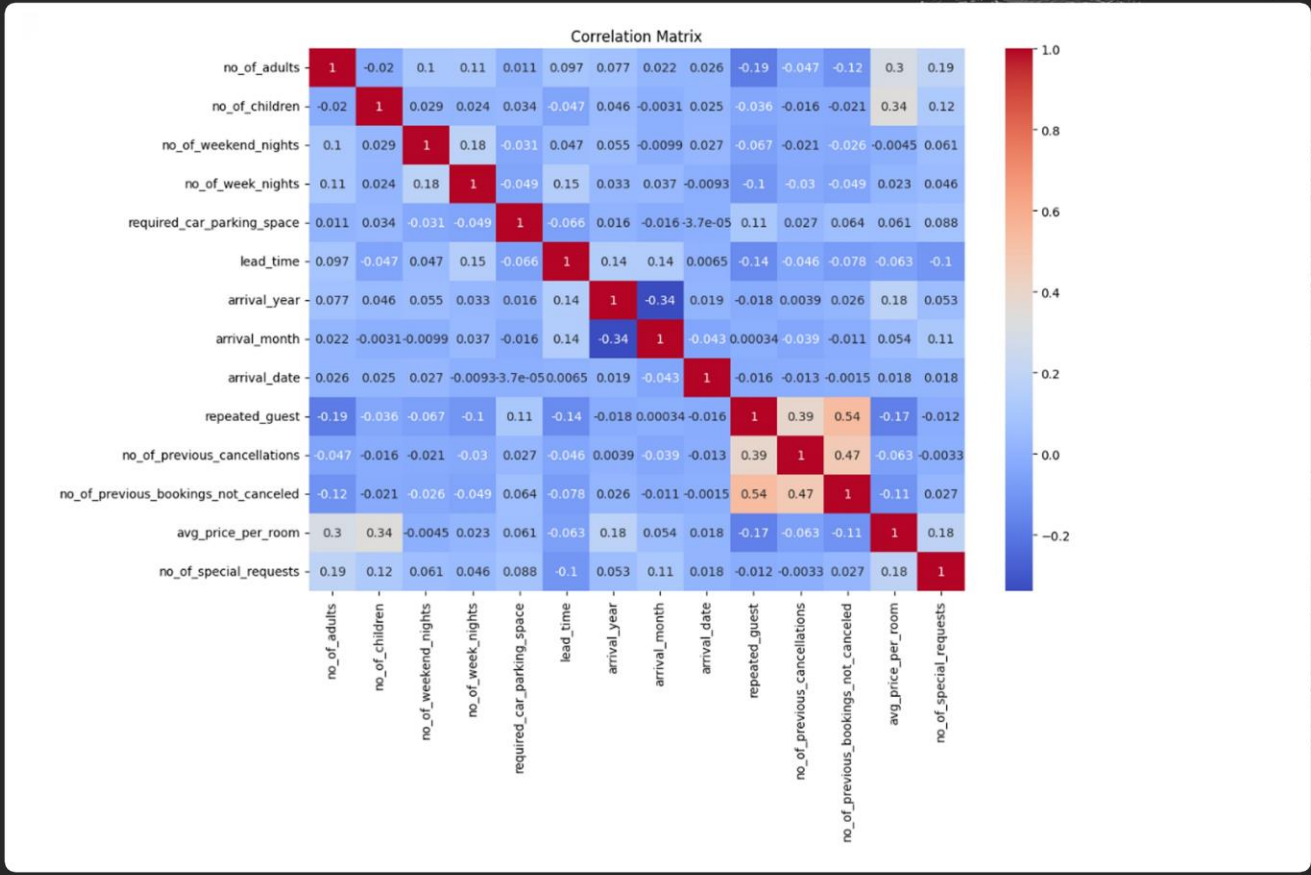






EXPLORING DATA

Correlation between numerical features



EXPLORING DATA

MISSING & DUPLICATE VALUES

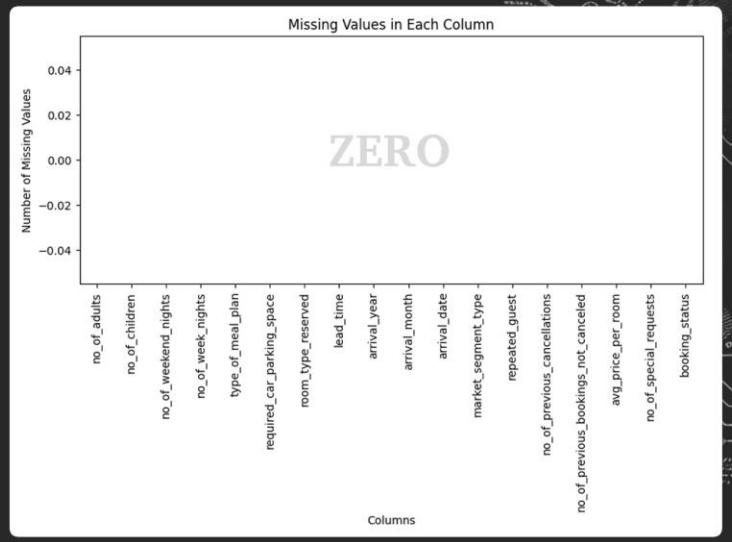
```
# Check for missing values
missing_values = df.isnull().sum()
missing_values
```

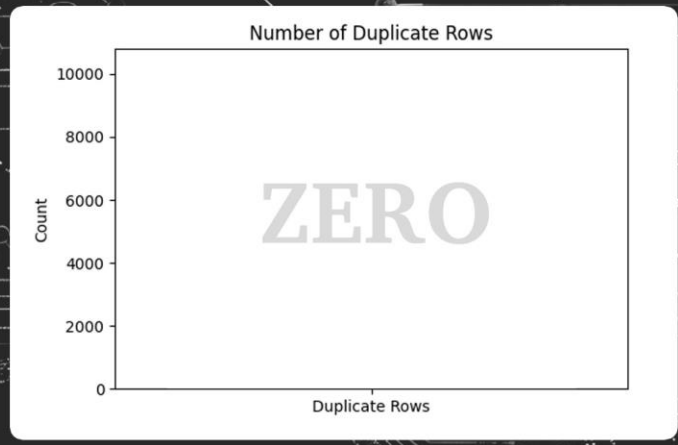
```
no_of_adults                            0
no_of_children                          0
no_of_weekend_nights                    0
no_of_week_nights                       0
type_of_meal_plan                       0
required_car_parking_space              0
room_type_reserved                      0
lead_time                               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
```

```
# Check for duplicates
duplicate_count = df.duplicated().sum()
print("Number of duplicate rows:", duplicate_count)
```

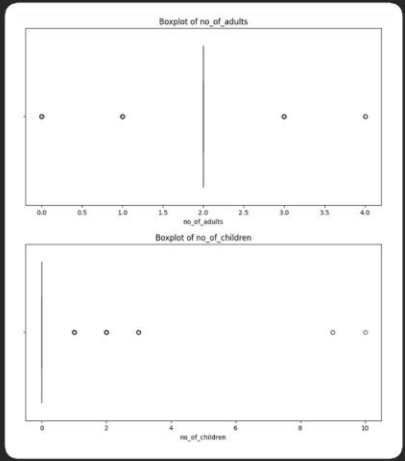
```
Number of duplicate rows: 0
```

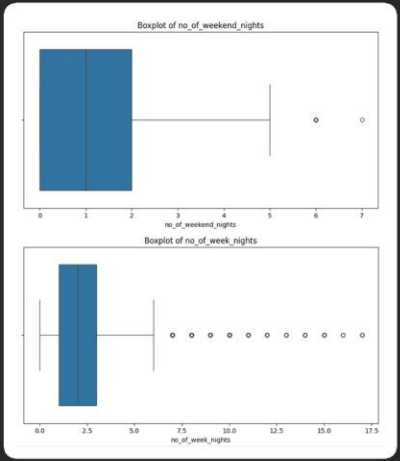
```
# Drop duplicate rows
df = df.drop_duplicates()
```

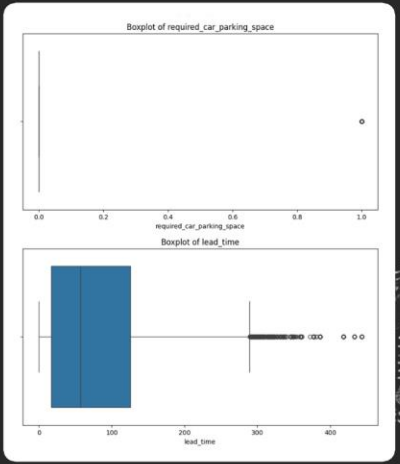


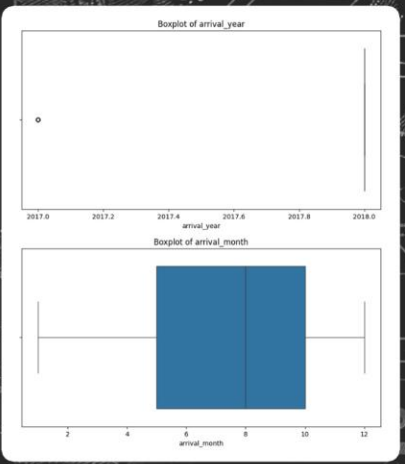


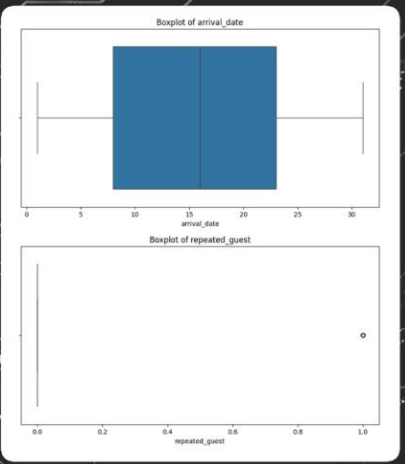
HANDLING OUTLIERS

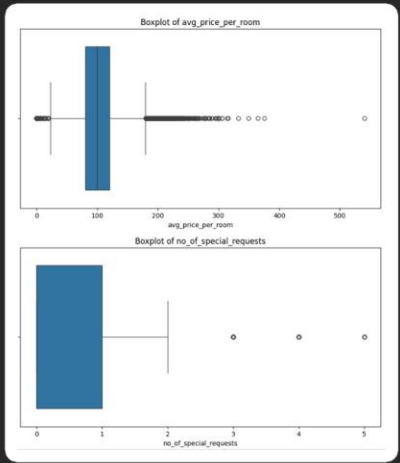


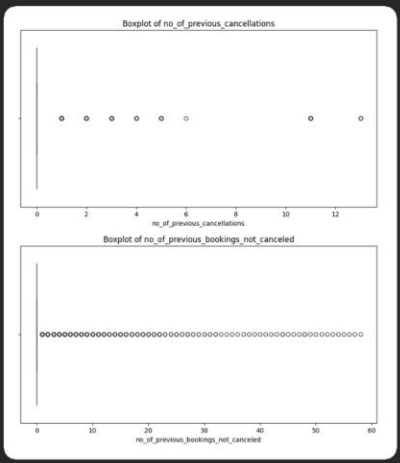













```
[12] # Handling Outliers
     def remove_outliers(df, col):
         Q1 = df[col].quantile(0.25)
         Q3 = df[col].quantile(0.75)
         IQR = Q3 - Q1
         df = df[~((df[col] < (Q1 - 1.5 * IQR)) | (df[col] > (Q3 + 1.5 * IQR)))]
         return df

     numerical_cols = ['no_of_adults', 'no_of_children', 'no_of_weekend_nights', 'no_of_week_nights', 'lead_time', 'avg_price_per_room', 'no_of_special_requests']
     for col in numerical_cols:
         df = remove_outliers(df, col)
```

SCALING & NORMALIZATION

```python
from sklearn.preprocessing import StandardScaler

# Scaling and Normalization
scaler = StandardScaler()
df[numerical_cols] = scaler.fit_transform(df[numerical_cols])
```

ENCODING CATEGORICAL VARIABLES

```python
# Encoding Categorical Variables
df = pd.get_dummies(df, columns=['type_of_meal_plan', 'room_type_reserved', 'market_segment_type'], drop_first=True)
```

SPLITTING THE DATA

FEATURE-TARGET SPLIT:

```
# Split the data into training and testing sets
x = df.drop(columns = ['booking_status'])
y = df['booking_status']
```

TRAIN-TEST SPLIT:

```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2, random_state=42)
```

MODEL SELECTION

OBJECTIVE:

TO IDENTIFY THE BEST-PERFORMING MODEL FOR PREDICTING CUSTOMER BEHAVIOR USING DIFFERENT MACHINE LEARNING ALGORITHMS.

MODELS TESTED:

LOGISTIC REGRESSION:
PLAIN
BALANCED

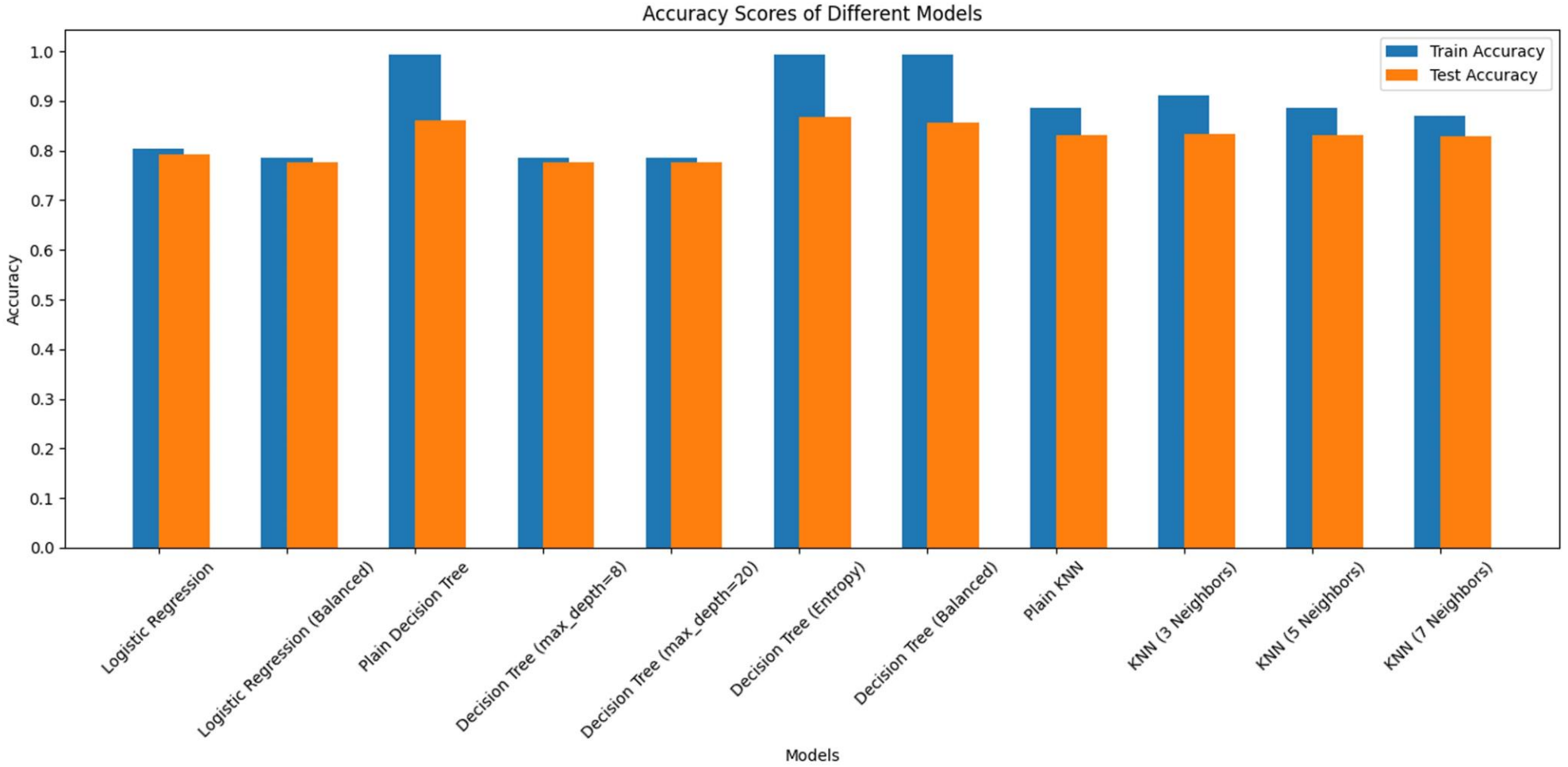
DECISION TREE:
PLAIN
MAX_DEPTH=8
MAX_DEPTH=20
ENTROPY
BALANCED

K-NEAREST NEIGHBORS (KNN):
PLAIN
3 NEIGHBORS
5 NEIGHBORS
7 NEIGHBORS

OUTCOME:
THE DECISION TREE WITH ENTROPY OUTPERFORMED ALL OTHER MODELS IN TERMS OF ACCURACY AND RELIABILITY.
HEREFORE, I DECIDED TO PROCEED WITH THIS MODEL FOR FURTHER ANALYSIS.

NEXT STEPS:

FINALIZE THE DECISION TREE (ENTROPY) MODEL.
IMPLEMENT FURTHER EVALUATION AND TUNING.
DEPLOY THE MODEL FOR PREDICTION TASKS.


Accuracy Scores of Different Models
Train Accuracy
Test Accuracy
Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Logistic Regression
Logistic Regression (Balanced)
Plain Decision Tree
Decision Tree (max_depth=8)
Decision Tree (max_depth=20)
Decision Tree (Entropy)
Decision Tree (Balanced)
Plain KNN
KNN (3 Neighbors)
KNN (5 Neighbors)
KNN (7 Neighbors)
Models

EVALUATING THE MODEL

```python
# Evaluation on the training set
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

dt_entropy_train_acc = accuracy_score(ytrain, train_pred_entropy)
print(f'Training Accuracy (Decision Tree with Entropy): {dt_entropy_train_acc}')
print('Classification Report (Training):')
print(classification_report(ytrain, train_pred_entropy))
print('Confusion Matrix (Training):')
print(confusion_matrix(ytrain, train_pred_entropy))

# Evaluation on the testing set
dt_entropy_test_acc = accuracy_score(ytest, test_pred_entropy)
print(f'Testing Accuracy (Decision Tree with Entropy): {dt_entropy_test_acc}')
print('Classification Report (Testing):')
print(classification_report(ytest, test_pred_entropy))
print('Confusion Matrix (Testing):')
print(confusion_matrix(ytest, test_pred_entropy))
```

```
Training Accuracy (Decision Tree with Entropy): 0.9932062114638045
Classification Report (Training):
              precision    recall  f1-score   support

           0       0.99      1.00      0.99     11484
           1       0.99      0.99      0.99      6032

    accuracy                           0.99     17516
   macro avg       0.99      0.99      0.99     17516
weighted avg       0.99      0.99      0.99     17516

Confusion Matrix (Training):
[[11446    38]
 [   81  5951]]
Testing Accuracy (Decision Tree with Entropy): 0.8668949771689498
Classification Report (Testing):
              precision    recall  f1-score   support

           0       0.91      0.89      0.90      2932
           1       0.79      0.81      0.80      1448

    accuracy                           0.87      4380
   macro avg       0.85      0.85      0.85      4380
weighted avg       0.87      0.87      0.87      4380

Confusion Matrix (Testing):
[[2620  312]
 [ 271 1177]]
```

TRAINING ACCURACY - 0.99
TESTING ACCURACY - 0.86

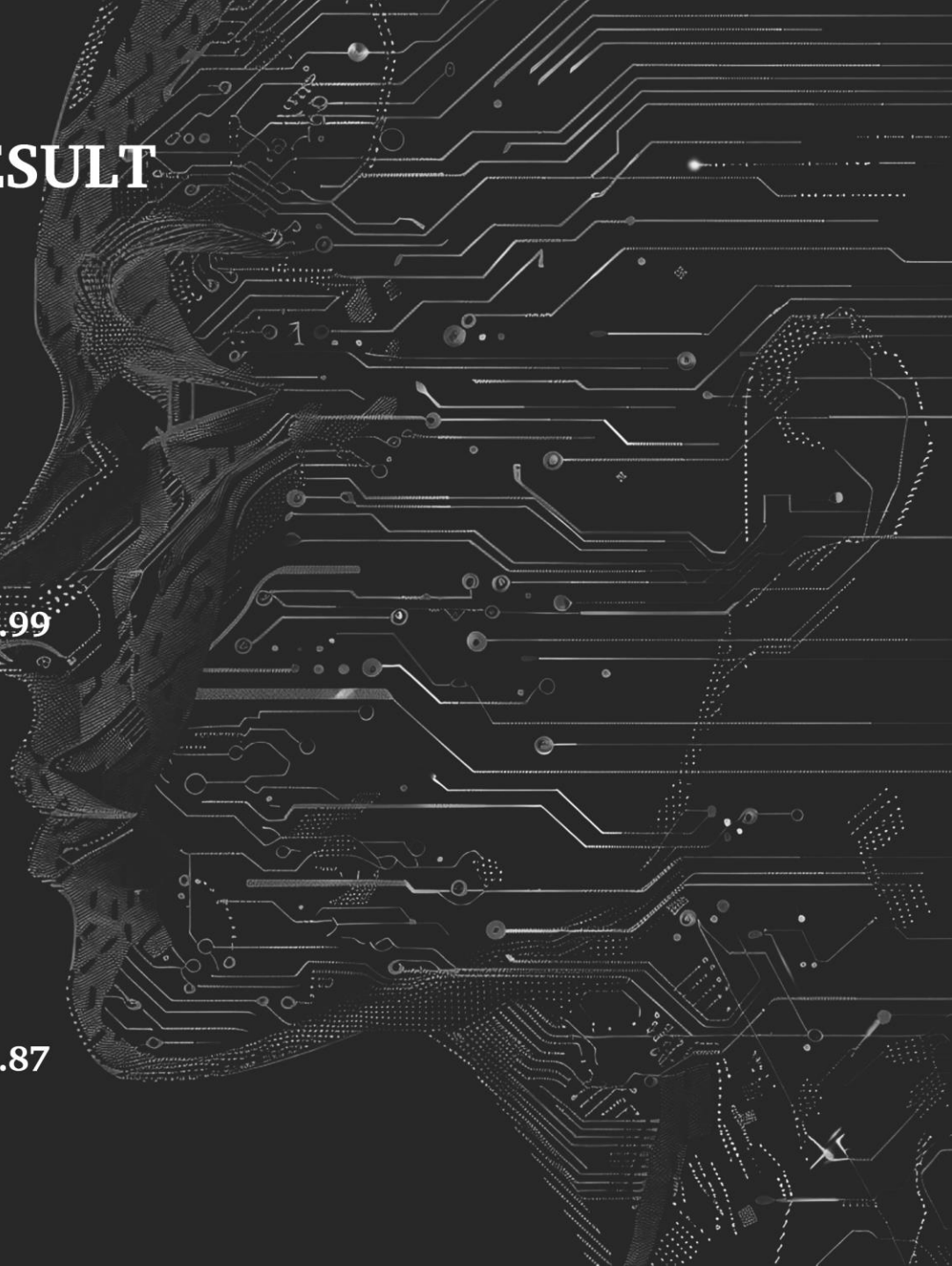
EVALUATION RESULT

TRAINING PERFORMANCE (DECISION TREE WITH ENTROPY):

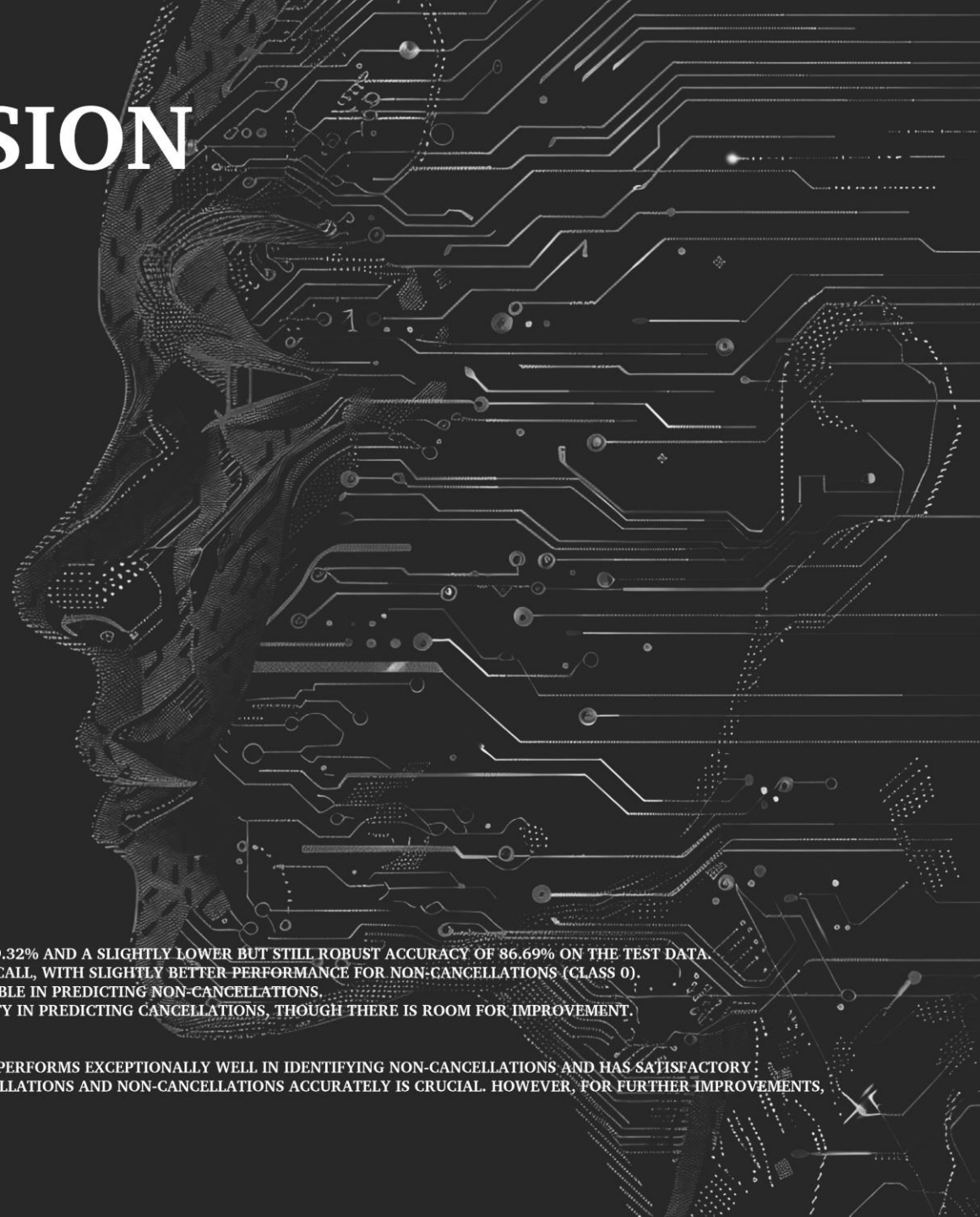
ACCURACY: 99.32%
CLASS 0: PRECISION 99%, RECALL 100%, F1-SCORE 0.99
CLASS 1: PRECISION 99%, RECALL 99%, F1-SCORE 0.99
OVERALL ACCURACY: 99%
MACRO AVERAGE: PRECISION 0.99, RECALL 0.99, F1-SCORE 0.99
WEIGHTED AVERAGE: PRECISION 0.99, RECALL 0.99, F1-SCORE 0.99
CONFUSION MATRIX: 11,446 TN, 38 FP, 81 FN, 5,951 TP

TESTING PERFORMANCE (DECISION TREE WITH ENTROPY):

ACCURACY: 86.69%
CLASS 0: PRECISION 91%, RECALL 89%, F1-SCORE 0.90
CLASS 1: PRECISION 79%, RECALL 81%, F1-SCORE 0.80
OVERALL ACCURACY: 87%
MACRO AVERAGE: PRECISION 0.85, RECALL 0.85, F1-SCORE 0.85
WEIGHTED AVERAGE: PRECISION 0.87, RECALL 0.87, F1-SCORE 0.87
CONFUSION MATRIX: 2,620 TN, 312 FP, 271 FN, 1,177 TP



CONCLUSION



GOAL: CAN YOU PREDICT IF A CUSTOMER IS GOING TO CANCEL THE RESERVATION?

TRAINING PERFORMANCE (DECISION TREE WITH ENTROPY):
ACCURACY: 99.32%
THE MODEL CORRECTLY CLASSIFIED 99.32% OF THE TRAINING DATA.
CLASS 0 (NO CANCELLATION):
PRECISION: 99% — WHEN THE MODEL PREDICTS NO CANCELLATION, IT'S CORRECT 99% OF THE TIME.
RECALL: 100% — THE MODEL CORRECTLY IDENTIFIES 100% OF ALL ACTUAL NON-CANCELLATIONS.
F1-SCORE: 0.99 — BALANCING PRECISION AND RECALL.
CLASS 1 (CANCELLATION):
PRECISION: 99% — WHEN THE MODEL PREDICTS A CANCELLATION, IT'S CORRECT 99% OF THE TIME.
RECALL: 99% — THE MODEL CORRECTLY IDENTIFIES 99% OF ALL ACTUAL CANCELLATIONS.
F1-SCORE: 0.99 — BALANCING PRECISION AND RECALL.
CONFUSION MATRIX:
TRUE NEGATIVES: 11,446
FALSE POSITIVES: 38
FALSE NEGATIVES: 81
TRUE POSITIVES: 5,951

TESTING PERFORMANCE (DECISION TREE WITH ENTROPY):
ACCURACY: 86.69%
THE MODEL CORRECTLY CLASSIFIED 86.69% OF THE TESTING DATA.
CLASS 0 (NO CANCELLATION):
PRECISION: 91% — WHEN THE MODEL PREDICTS NO CANCELLATION, IT'S CORRECT 91% OF THE TIME.
RECALL: 89% — THE MODEL CORRECTLY IDENTIFIES 89% OF ALL ACTUAL NON-CANCELLATIONS.
F1-SCORE: 0.90 — BALANCING PRECISION AND RECALL.
CLASS 1 (CANCELLATION):
PRECISION: 79% — WHEN THE MODEL PREDICTS A CANCELLATION, IT'S CORRECT 79% OF THE TIME.
RECALL: 81% — THE MODEL CORRECTLY IDENTIFIES 81% OF ALL ACTUAL CANCELLATIONS.
F1-SCORE: 0.80 — BALANCING PRECISION AND RECALL.
CONFUSION MATRIX:
TRUE NEGATIVES: 2,620
FALSE POSITIVES: 312
FALSE NEGATIVES: 271
TRUE POSITIVES: 1,177

OVERALL PERFORMANCE:
HIGH ACCURACY: THE MODEL DEMONSTRATES HIGH ACCURACY, PARTICULARLY IN THE TRAINING PHASE WITH AN ACCURACY OF 99.32% AND A SLIGHTLY LOWER BUT STILL ROBUST ACCURACY OF 86.69% ON THE TEST DATA.
BALANCED PERFORMANCE: THE F1-SCORES FOR BOTH CLASSES INDICATE A BALANCED PERFORMANCE BETWEEN PRECISION AND RECALL, WITH SLIGHTLY BETTER PERFORMANCE FOR NON-CANCELLATIONS (CLASS 0).
ROBUST PREDICTION FOR NON-CANCELLATIONS: HIGHER PRECISION AND RECALL FOR CLASS 0 SUGGEST THE MODEL IS MORE RELIABLE IN PREDICTING NON-CANCELLATIONS.
SATISFACTORY PREDICTION FOR CANCELLATIONS: REASONABLE PRECISION AND RECALL FOR CLASS 1 INDICATE THE MODEL'S UTILITY IN PREDICTING CANCELLATIONS, THOUGH THERE IS ROOM FOR IMPROVEMENT.

CONCLUSION:
THE DECISION TREE MODEL WITH ENTROPY IS EFFECTIVE IN PREDICTING WHETHER A CUSTOMER WILL CANCEL A RESERVATION. IT PERFORMS EXCEPTIONALLY WELL IN IDENTIFYING NON-CANCELLATIONS AND HAS SATISFACTORY PERFORMANCE IN PREDICTING CANCELLATIONS. THIS MAKES THE MODEL VALUABLE FOR APPLICATIONS WHERE PREDICTING CANCELLATIONS AND NON-CANCELLATIONS ACCURATELY IS CRUCIAL. HOWEVER, FOR FURTHER IMPROVEMENTS, ENHANCING THE MODEL'S ABILITY TO PREDICT CANCELLATIONS CAN BE CONSIDERED.

THANK YOU