Predicting Hotel Booking Cancellations Using Machine Learning

Name: Lavanya Kondisetty
ID: 23094273

Table of Contents

1. Introduction

Hotel booking cancellations create major obstacles within the hospitality industry because they disrupt revenue management systems as well as operational planning and affect customer service quality. Advance prediction of cancellations lets hotels establish tactics that protect their revenue while adjusting their resource use. The field of cancellation prediction has historically used historical patterns plus manual evaluation techniques but these methods struggle to analyze the intricate network between multiple variables spanning demographic data and terms of booking to rates and outside environmental elements.

The implementation of classification models helps hotels to predict cancellations which provides them with data-based capabilities to decide about their overbooking approaches price adjustments and customized promotional initiatives. The researchers use Decision Trees and Random Forest classifiers from machine learning to forecast hotel booking cancellations by analyzing important determining elements. Predictive performance assessment occurs through a three-step process where data is prepared before exploratory visualization and testing evaluation of models take place. The main goal of this research support hotel managers in diminishing cancellation possibilities to promote better business sustainability.

2. Business Problem

Current booking management within the hospitality sector proves difficult because hotels experience a high rate of reservation drops. Hotels experience major financial damages along with operational setbacks which decrease their profit margins when guests cancel their bookings. There is a financial impact on hotels because they fail to refill empty rooms after customers cancel reservations at the last minute (Chen *et al.* 2023). Resource planning becomes complex when cancellations occur frequently because these affect the staff allocation and inventory requirements and overall service delivery quality. Hotels attempt to find predictive solutions because they need to foresee future cancellations and develop strategies to reduce their impact.

Current methods used for cancellation management base themselves on historical records and manual analyses but demonstrate both inefficiency and error vulnerabilities. Hotels practice overbooking to handle cancelled bookings but their operational capacity gets stretched thin when more guests show up than room capacity allows thus leading to diminished guest satisfaction. Such an issue demands an improved automated solution that helps decision-making processes (Bhardwaj *et al.* 2024). Through the application of machine learning technology, the prediction of booking cancellations becomes possible through data analysis which includes lead time characteristics with customer demographics and specific requests and selected room choices together with pricing data patterns.

3. Methods

Structured data science procedures are used for pre-processing and engineering features while selecting and evaluating machine learning models for booking cancellation prediction purposes. The dataset undergoes initial checks for inconsistency and then shifts to required preprocessing operations before implementing categorical encoding techniques with feature scaling methods. A comparison is made between the Decision Tree Classifier and Random Forest Classifier through implementation followed by different performance metric evaluations.

3.1 Dataset Overview and Preprocessing

A dataset in this research contains multiple hotel booking elements and information about customers and their reservations. The booking status variable functions as the target measure to reveal whether reservations were approved or rejected. A dataset analysis should start with an investigation of distribution patterns to uncover data types and assess the extent of missing values (Chen et al 2022). Both the median value and mode value serve as substitution methods that replace missing points for numerical and categorical features respectively. The model maintains stable predictions through its approach to data gaps no matter which of its analysis inputs contains missing information. The system converts the date of reservation data into datetime format for possible extraction needs but omits it from training since its predictive value is minimal.

3.2 Feature Engineering and Encoding

Multiple categorical variables get transformed through the application of Label Encoding to produce numerical quantities for machine learning models in "type of meal," "room type," "market segment type," and "booking status." The encoding technique makes categorical data numerically interpretable which prevents the model from becoming too complex.

The dataset undergoes division into training and testing subsets by using an 80:20 ratio which protects the evaluation stage from biased assessment with a separate portion of data (Herrera *et al.* 2024). The structure of the data separation mechanism stops models from becoming too specialized while enabling them to correctly handle previously unseen cases.

3.3 Model Selection and Training

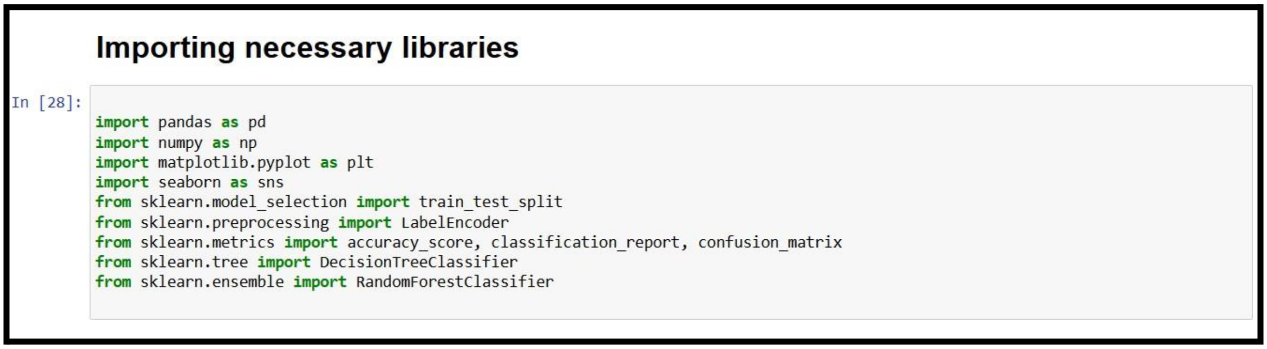
Two machine learning systems remain in use:

Decision Tree Classifier: The decision Tree Classifier operates as a tree-based model where it separates data through feature conditions to identify reservations as either confirmed or cancelled.

Random Forest Classifier: Multiple decision trees merge their predictions through Random Forest Classifier which enhances accuracy performance as well as minimizing overfitting issues. The models receive training from the collected dataset through which accuracy precision and recall F1-score assessments and confusion matrix evaluation occur (Adil *et al.* 2021). The evaluation analyzes feature importance so researchers can identify what factors play the most significant role in booking cancellations.

4. Analysis of the Result

4.1 Loading and Pre-processing



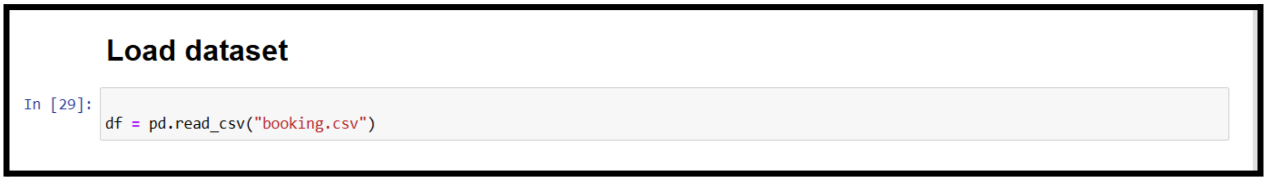
Importing necessary libraries

```
In [28]:
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

Figure 4.1.1: Importing Libraries

(Source: Jupyter)

The figure 4.1.1 shows the procedure for bringing in vital libraries which support data handling alongside visualization as well as machine learning execution (Febrian *et al.* 2024). The required libraries Pandas along with NumPy Matplotlib and Seaborn together with Scikit-learn help process and develop the dataset effectively.



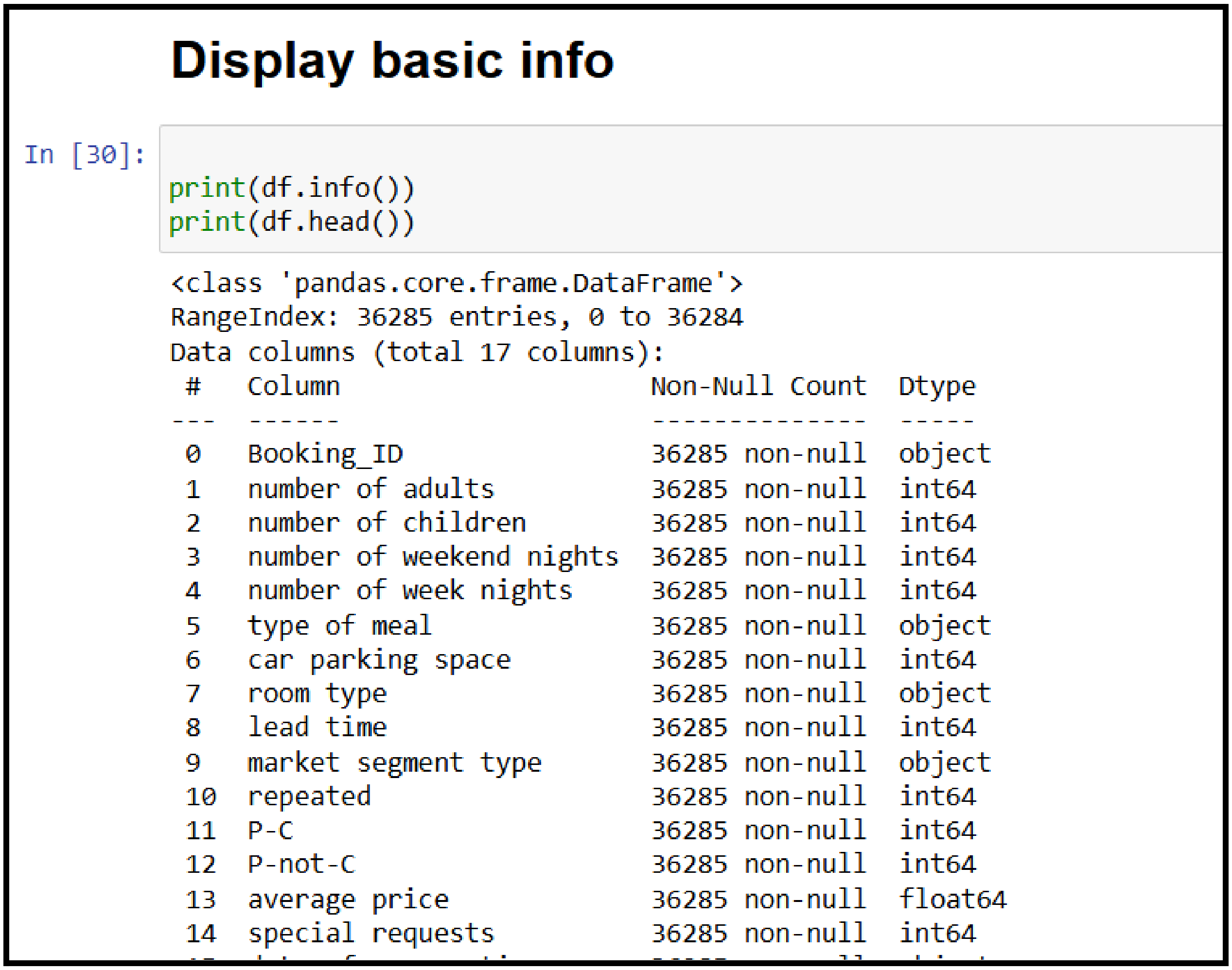
Load dataset

```
In [29]:
df = pd.read_csv("booking.csv")
```

Figure 4.1.2: Loading the Dataset

(Source: Jupyter)

The figure 4.1.2 presents the dataset in its Pandas DataFrame format (Saputro *et al.* 2021). The proper structure of the dataset becomes achievable after this step because Python provides efficient data retrieval and manipulation to enable analysis preprocessing and modeling.



Display basic info

In [30]:

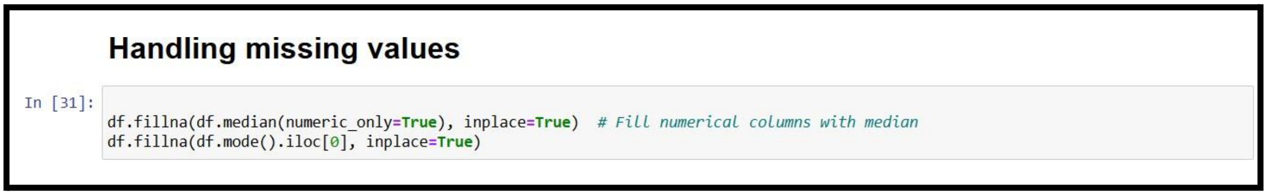
```
print(df.info())
print(df.head())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36285 entries, 0 to 36284
Data columns (total 17 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Booking_ID                 36285 non-null  object
 1   number of adults           36285 non-null  int64
 2   number of children         36285 non-null  int64
 3   number of weekend nights   36285 non-null  int64
 4   number of week nights      36285 non-null  int64
 5   type of meal               36285 non-null  object
 6   car parking space          36285 non-null  int64
 7   room type                  36285 non-null  object
 8   lead time                  36285 non-null  int64
 9   market segment type        36285 non-null  object
 10  repeated                   36285 non-null  int64
 11  P-C                        36285 non-null  int64
 12  P-not-C                    36285 non-null  int64
 13  average price              36285 non-null  float64
 14  special requests           36285 non-null  int64
```

Figure 4.1.3: Display basic info

(Source: Jupyter)

The figure 4.1.3 shows an overview of the dataset structure which reveals the number of rows and columns and depicts data types per feature together with null value details (Hermawan *et al.* 2025). The analysis unveils fundamental structural and quality aspects regarding the dataset.



Handling missing values

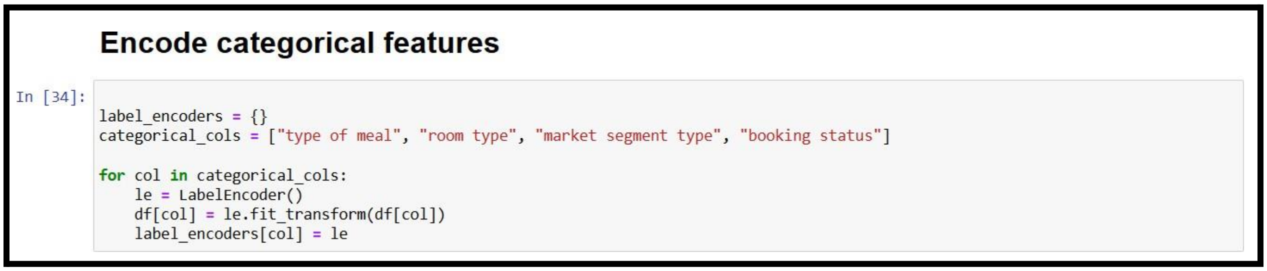
In [31]:

```
df.fillna(df.median(numeric_only=True), inplace=True)  # Fill numerical columns with median
df.fillna(df.mode().iloc[0], inplace=True)
```

Figure 4.1.4: Handling missing values

(Source: Jupyter)

The figure 4.1.4 shows the workflow for handling null values across the data set (Shirisha *et al.* 2023). The median value replaces missing numerical data and categorical values get filled with their modal distribution for accurate model prediction.



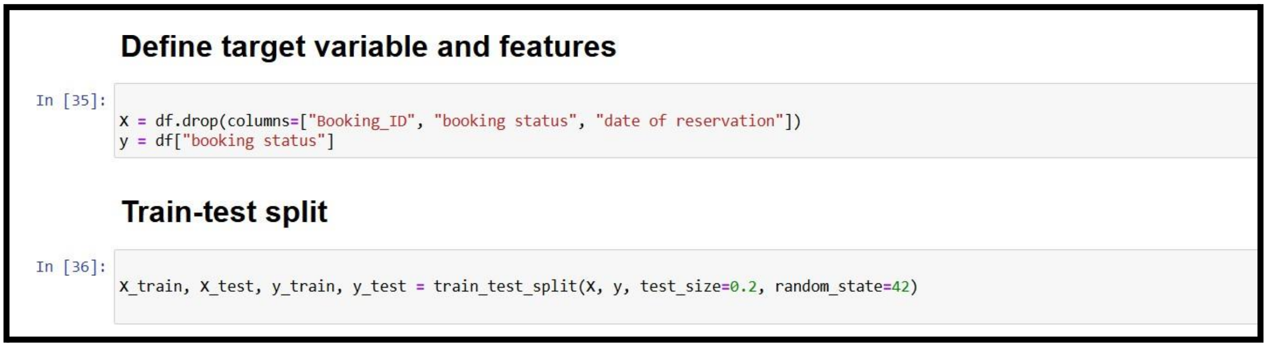
Encode categorical features

In [34]:
```
label_encoders = {}
categorical_cols = ["type of meal", "room type", "market segment type", "booking status"]

for col in categorical_cols:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le
```

Figure 4.1.5: Encode categorical features

(Source: Jupyter)

The figure 4.1.5 shows the conversion process for turning categorical variables into numerical data that appear in the provided visualization through label encoding (Rahmawati *et al.* 2024). A transformation of categorical data into numeric values becomes essential for linking machine learning models since these systems only accept numbers as input for their training and predictive tasks.



Define target variable and features

In [35]:
```
X = df.drop(columns=["Booking_ID", "booking status", "date of reservation"])
y = df["booking status"]
```

Train-test split

In [36]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Figure 4.1.6: Splitting the dataset into train and test

(Source: Jupyter)

The figure 4.2.6 shows the division process of the dataset into training and testing parts (Putro *et al.* 2021). The division of data into training and testing sets allows the model to accept most of the available data for training while a separate segment remains dedicated exclusively to performance evaluation.

4.2 Visualizations

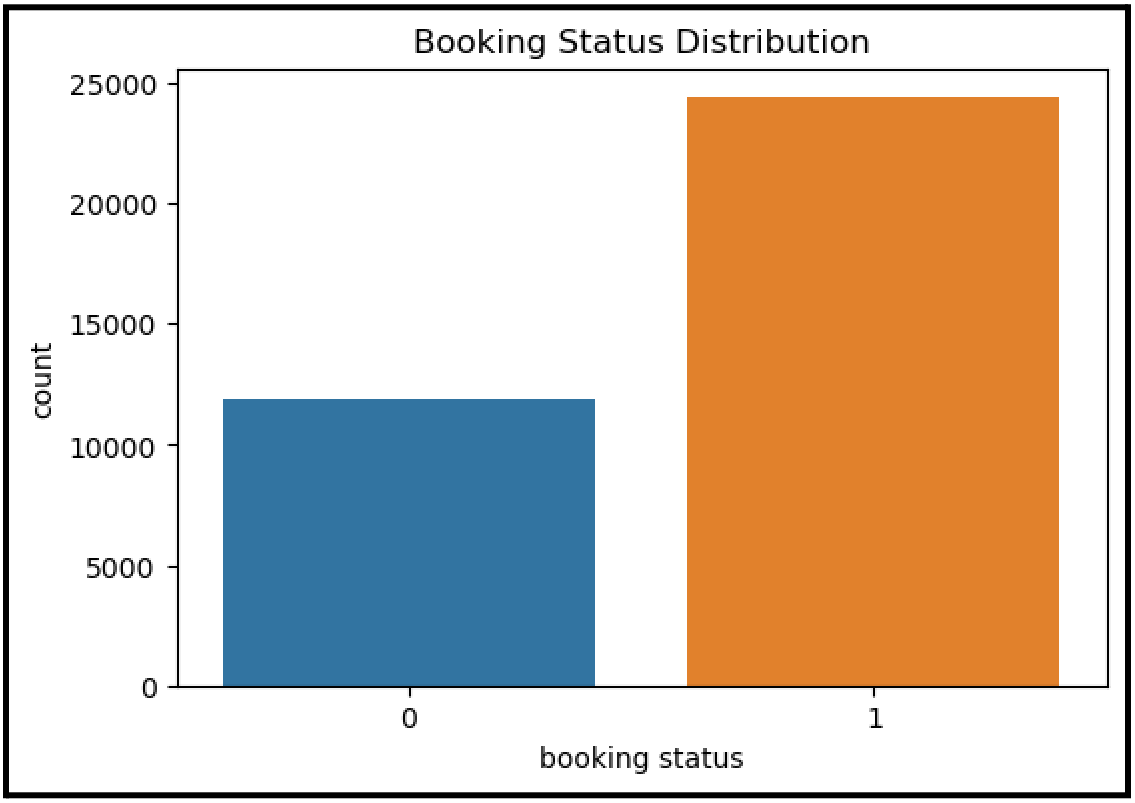


Figure 4.2.1: Booking Status Distribution

(Source: Jupyter)

The figure 4.2.1 shows a distribution of booking statuses which displays both confirmed and cancelled reservation proportions. The displayed information about total cancellation percentage reveals structural booking retention factors that help data analysis.

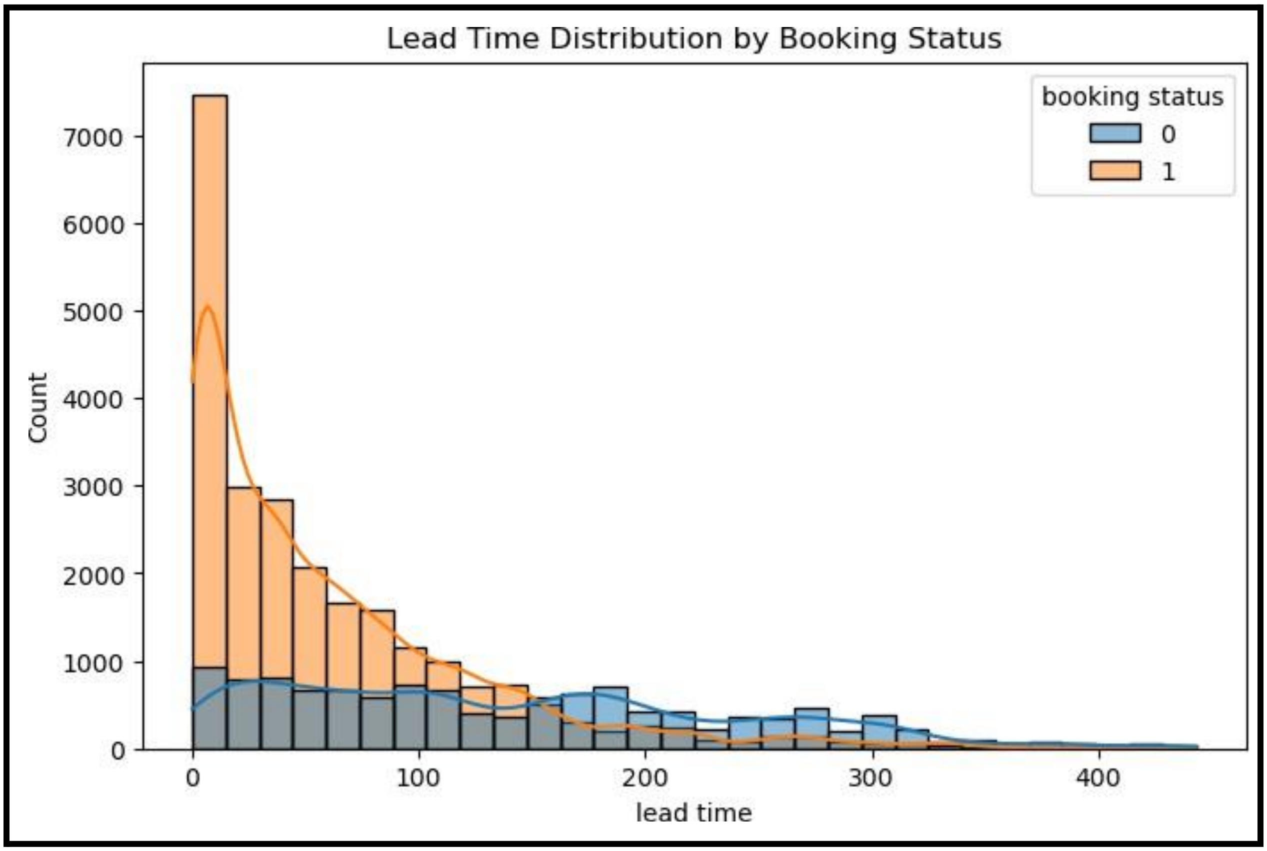


Figure 4.2.2: Lead Time Distribution by Booking Status

(Source: Jupyter)

The figure 4.2.2 shows how reservation timing influences cancellation patterns (Tian et al. 2023). Analysis shows booking timing before arrival impacts cancellation patterns and this influences business methods used to forecast upcoming patterns of demand.

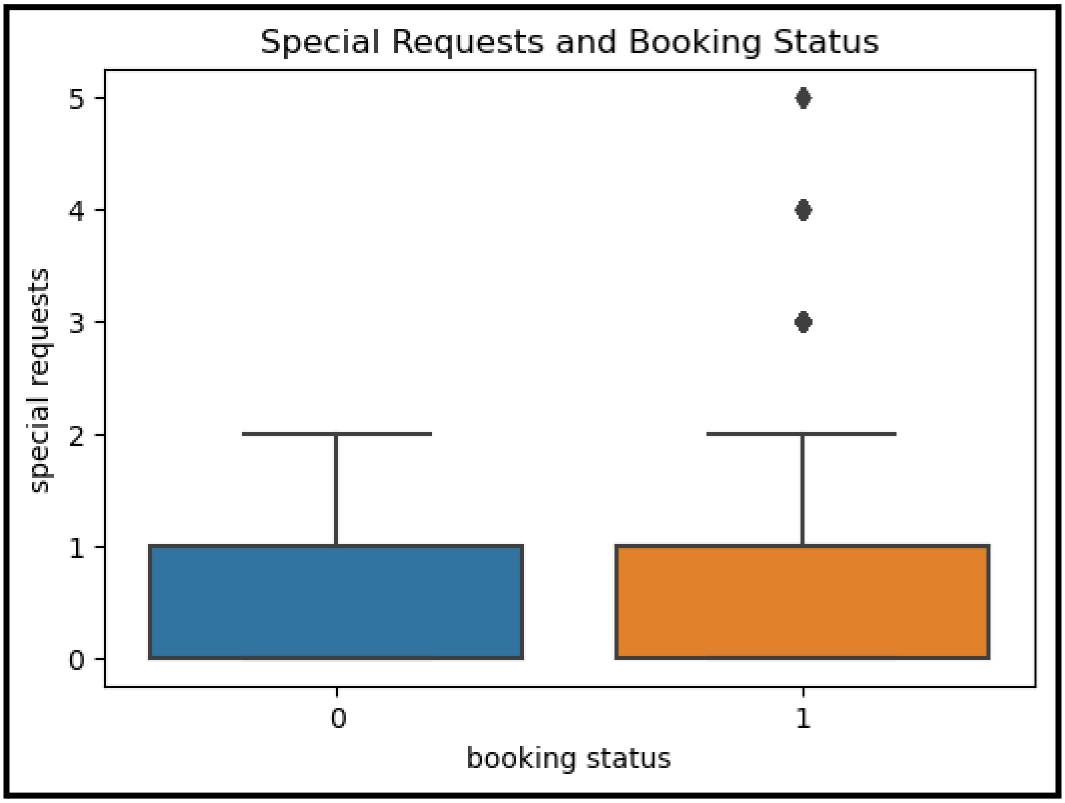


Figure 4.2.3: Special Requests and Booking Status

(Source: Jupyter)

The figure 4.2.3 shows how service requirements affect booking status changes about guest requests during the guest stay period. Through this knowledge, we can determine both the role of customer preferences in cancellation decisions and their effect on guest commitment.

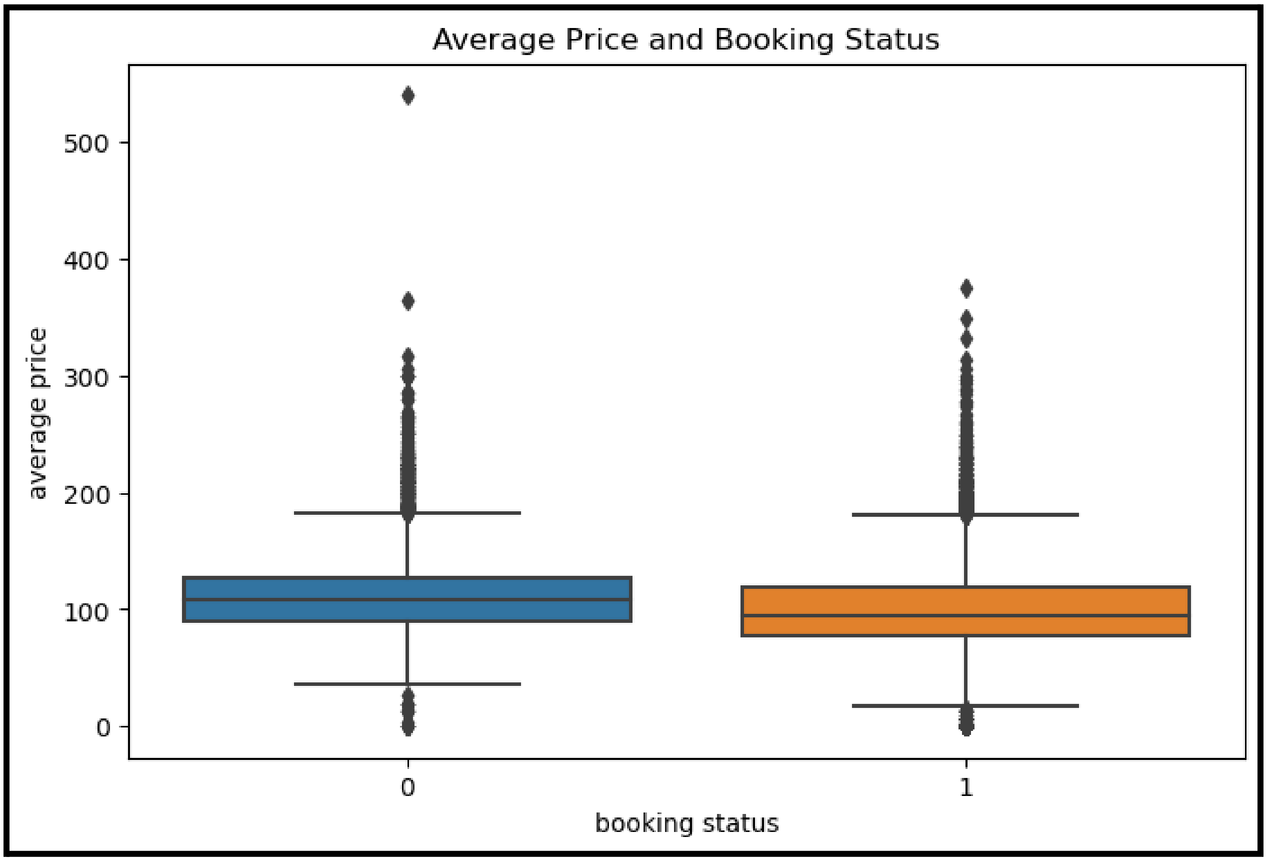


Figure 4.2.4: Average Price and Booking Status

(Source: Jupyter)

The figure 4.2.4 shows the price points that cause customers to make reservations yet result in reservation cancellations (Ahmed et al. 2024). The chart shows the level to which price sensitivity impacts reservation decisions for customers.

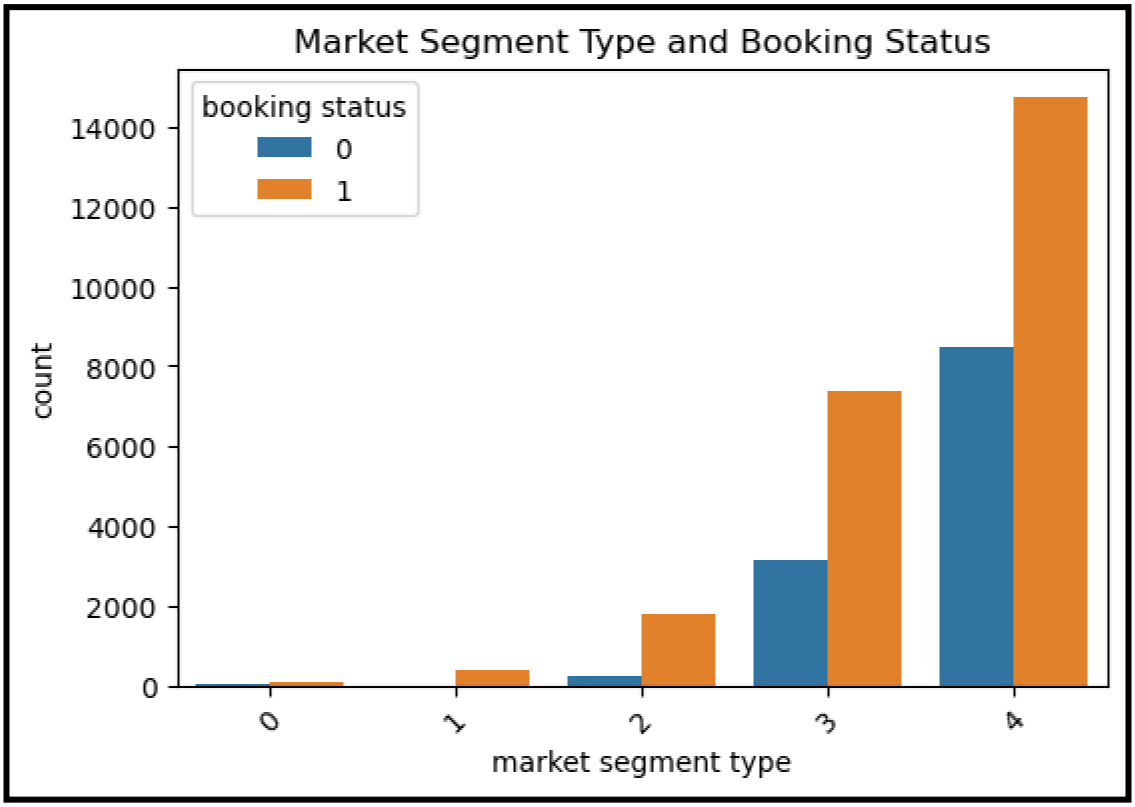


Figure 4.2.5: Market Segment Type and Booking Status

(Source: Jupyter)

The figure 4.2.5 shows the booking status distribution across multiple market segments which exhibits various ways customer groups manage cancellations. The analysis helps businesses detect the segments which show elevated cancellation rates so they can create specialized retention policies.

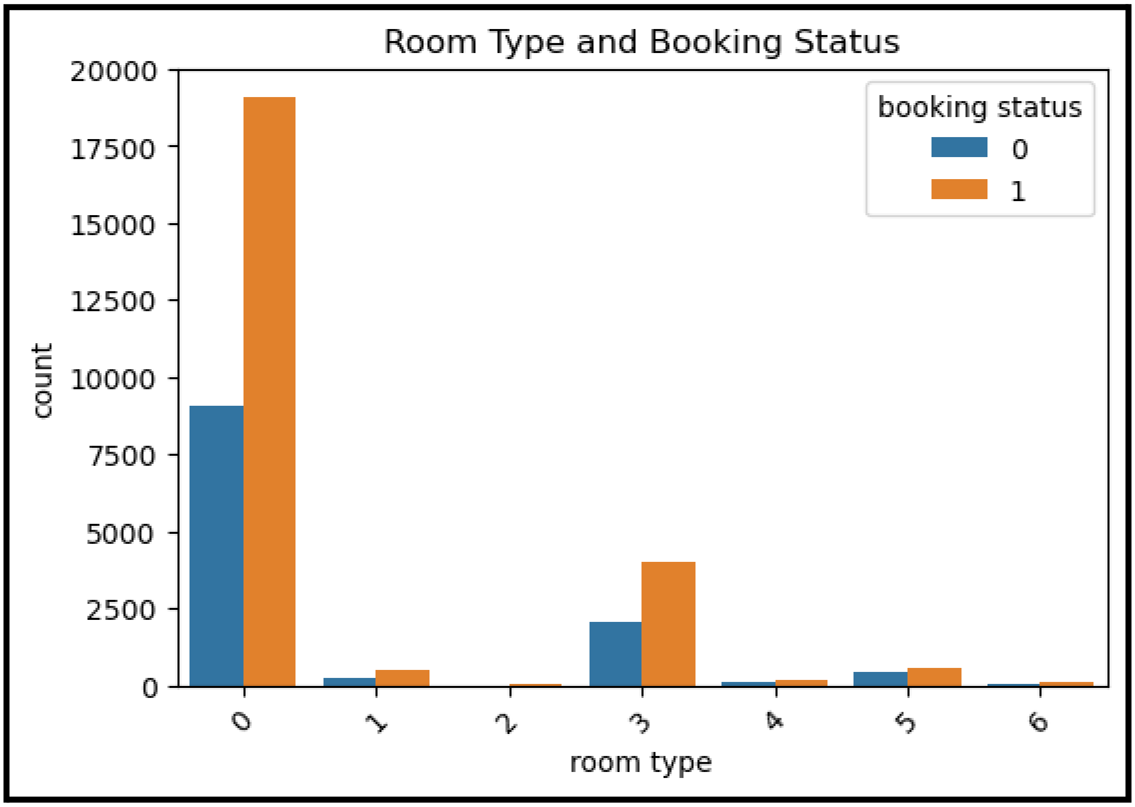
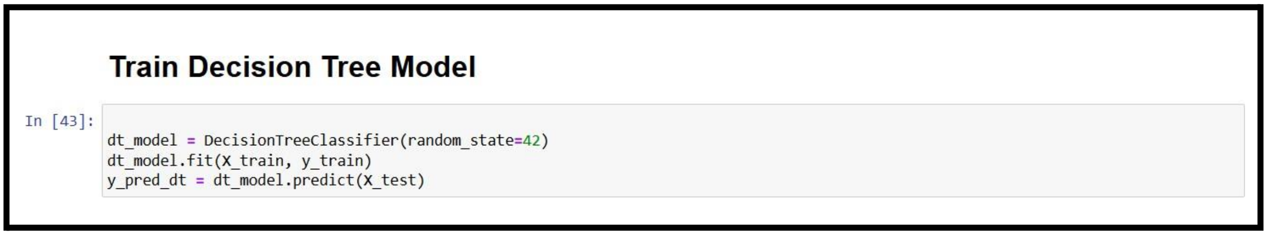


Figure 4.2.6: Room Type and Booking Status

(Source: Jupyter)

The figure 4.2.6 shows the cancellation patterns among different hotel categories as they relate to accommodation types (Rakesh *et al.* 2022). Evaluation of hotel cancellations reveals which types of rooms experience higher rates thus helping hotels develop their pricing and allocation systems.

4.3 Model Evaluation



Train Decision Tree Model

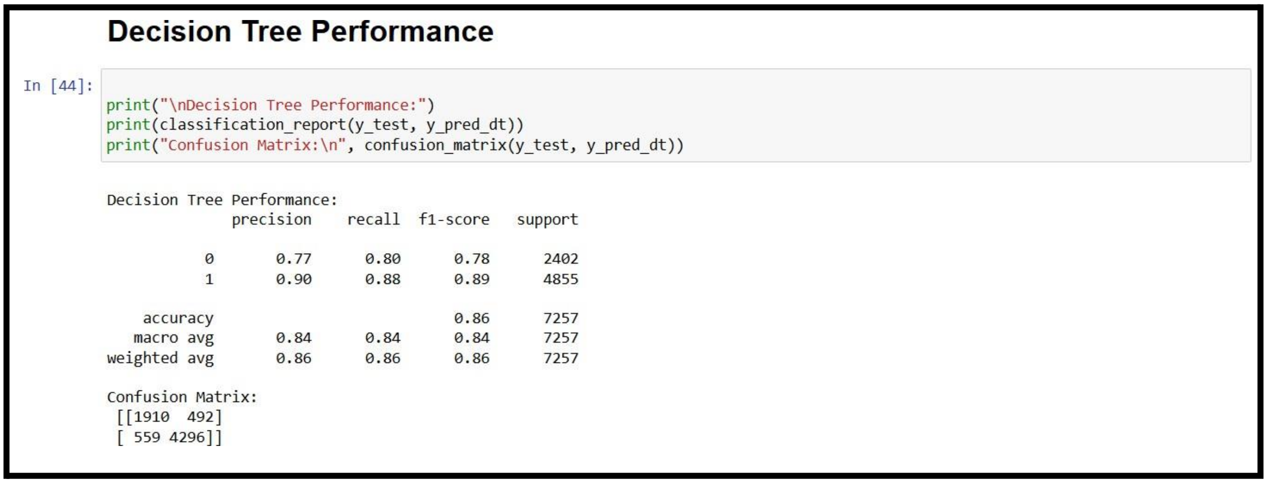
In [43]:

```
dt_model = DecisionTreeClassifier(random_state=42)
dt_model.fit(X_train, y_train)
y_pred_dt = dt_model.predict(X_test)
```

Figure 4.3.1: Decision Tree Model Training

(Source: Jupyter)

The figure 4.3.1 shows the Decision Tree model development through the training process for the dataset (Liu *et al.* 2023). Through training the model understands various patterns from the provided data which enables it to perform booking classification operations based on the provided features.



Decision Tree Performance

In [44]:

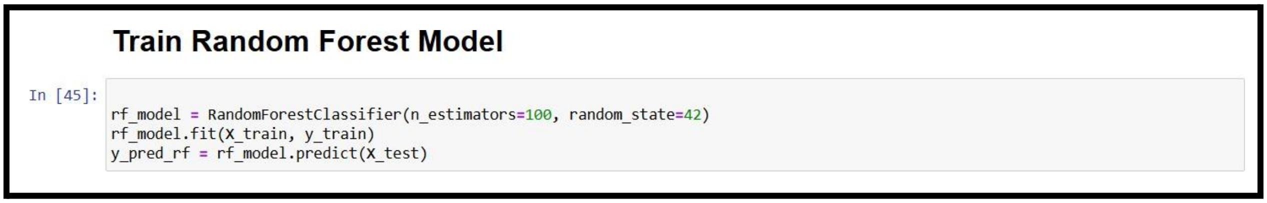
```
print("\nDecision Tree Performance:")
print(classification_report(y_test, y_pred_dt))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_dt))
```

```
Decision Tree Performance:
              precision    recall  f1-score   support

           0       0.77      0.80      0.78      2402
           1       0.90      0.88      0.89      4855

    accuracy                           0.86      7257
   macro avg       0.84      0.84      0.84      7257
weighted avg       0.86      0.86      0.86      7257

Confusion Matrix:
 [[1910  492]
 [ 559 4296]]
```

Figure 4.3.2: Decision Tree Performance

(Source: Jupyter)

The figure 4.3.2 shows the decision tree model evaluation features accuracy, precision, and recall together with the F1-score (Prabha *et al.* 2022). The performance evaluation of the prediction model reveals its capacity to detect booking cancellations while revealing productive spaces to enhance its operational efficiency.



Train Random Forest Model

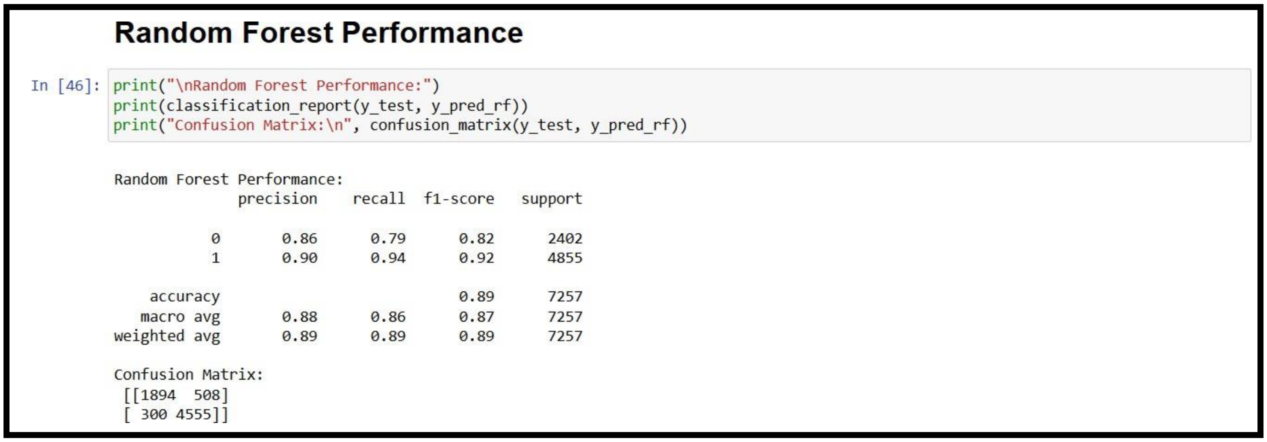
In [45]:

```
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
y_pred_rf = rf_model.predict(X_test)
```

Figure 4.3.3: Random Forest Model Training

(Source: Jupyter)

The figure 4.3.3 shows the Random Forest model trains through a procedure that includes multiple decision trees (Yoo *et al.* 2023). Through ensemble techniques, the prediction accuracy improves because multiple tree outputs combine to create stable and dependable booking cancellation estimates.



Random Forest Performance

```
In [46]: print("\nRandom Forest Performance:")
         print(classification_report(y_test, y_pred_rf))
         print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_rf))
```

```
Random Forest Performance:
              precision    recall  f1-score   support

           0       0.86      0.79      0.82      2402
           1       0.90      0.94      0.92      4855

    accuracy                           0.89      7257
   macro avg       0.88      0.86      0.87      7257
weighted avg       0.89      0.89      0.89      7257

Confusion Matrix:
 [[1894  508]
 [ 300 4555]]
```

Figure 4.3.4: Random Forest Performance

(Source: Jupyter)

The figure 4.3.4 shows the performance evaluations of the Random Forest model through its application of accuracy and precision at the same time as recall and confusion matrix assessment (Prasetya *et al.* 2024). The model demonstrates its ability to minimize overfitting conditions by improving the classification results above Decision Trees.

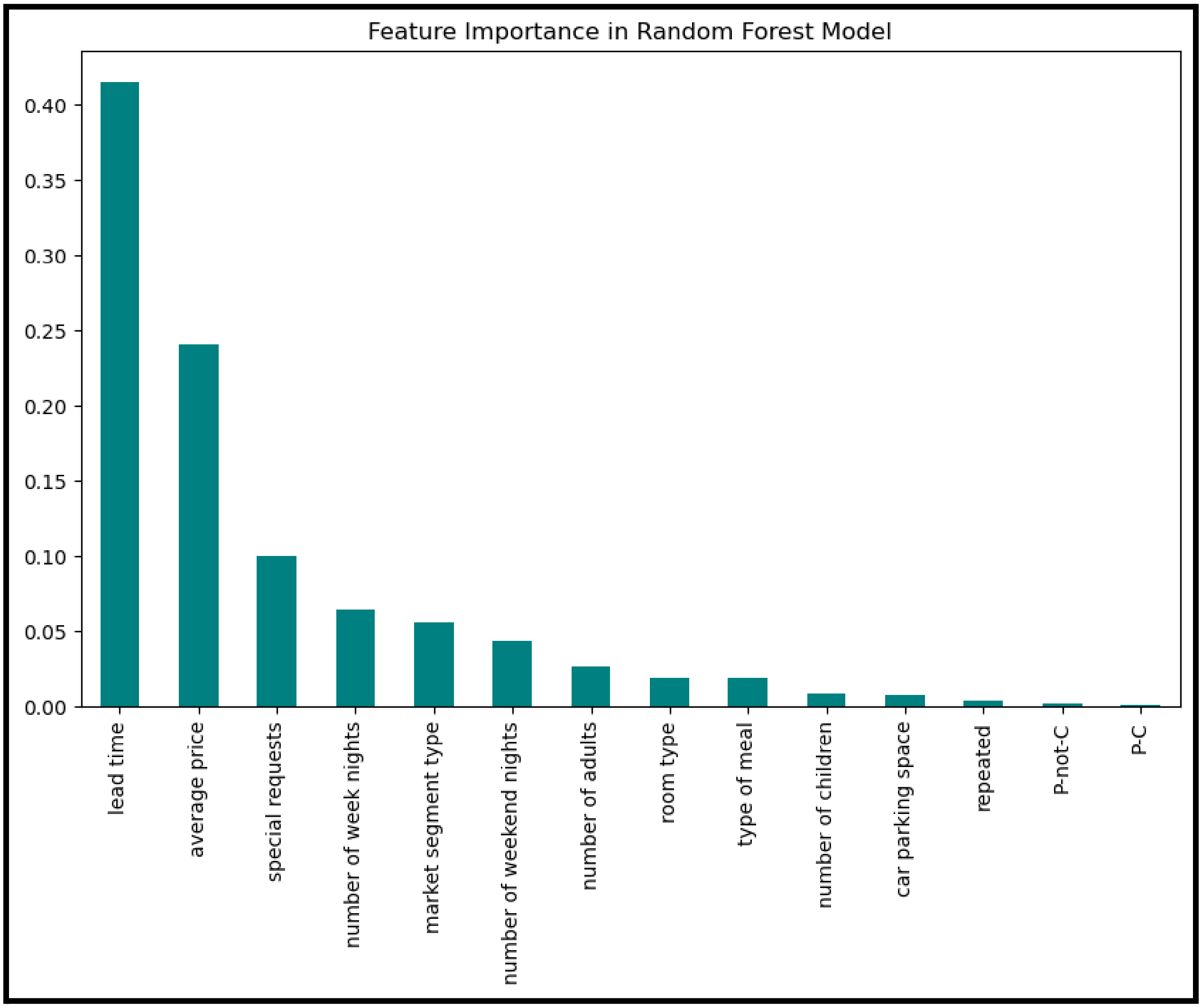


Figure 4.3.5: Feature Importance in Random Forest Model

(Source: Jupyter)

The figure 4.3.5 shows the feature importance scores stemming from the Random Forest analysis in this visual representation (Nababan *et al.* 2022). An analysis based on this chart identifies key factors which influence booking cancellations since these elements prove essential in determining a reservation's likelihood of cancellation.

5. Conclusion

The research indicates how machine learning systems successfully identify upcoming hotel reservation cancellations. The models evaluate different features including lead time room type market segment and special requests to understand why customers cancel bookings. Research on the data reveals strong connections between characteristics and booking decisions which strengthens the necessity of data-based decision processes in hotel operations.

The assessment of prediction models shows that Decision Tree along with Random Forest classifiers achieve high accuracy rates for cancellation prediction but Random Forest proves more robust in this task. The lead time together with average price and market segment emerge as the three critical factors which influence prediction results according to feature importance analysis.

The research indications show that machine learning technology can support hotel revenue management operations by letting operators take anticipatory steps to lower cancellation rates. Organizations should implement predictive models inside their hotel booking systems which enhances resource distribution develop more effective customer retention plans and reduce monetary losses stemming from booking cancellations.

6. Recommendation

The research results enable multiple recommendations to enhance hotel booking systems and decrease booking cancellations. Hotels should use machine learning models particularly Random Forest along with Decision Tree classifiers to make predictions about upcoming cancellations. Through booking system integration hotels gain the ability to create strategic overbooking plans for reducing revenue loss and offer cancellation incentives to vulnerable customers through predictive models.

Data collection optimization strategies will enhance prediction accuracy levels for hotels. Hotels need to maintain continual records of essential influences from customer demographics and both past booking patterns and seasonal market conditions which affect cancellations (Kumar *et al.* 2023). The model's performance can be boosted through improved methods of encoding categorical data and a better selection of features.

Adding an ongoing system for tracking model developments alongside booking behaviour changes should become part of the hotel management strategy. The model performance accuracy increases when new data sets are incorporated for maintenance alongside hyperparameter optimization. In order to achieve additional optimization of the prediction models hoteliers should experiment with advanced modeling approaches which comprise ensemble methods together with deep learning models.

References

Chen, S., Ngai, E.W., Ku, Y., Xu, Z., Gou, X. and Zhang, C., 2023. Prediction of hotel booking cancellations: Integration of machine learning and probability model based on interpretable feature interaction. *Decision Support Systems*, *170*, p.113959.

Herrera, A., Arroyo, A., Jiménez, A. and Herrero, Á., 2024. Forecasting hotel cancellations through machine learning. *Expert Systems*, *41*(9), p.e13608.

Adil, M., Ansari, M.F., Alahmadi, A., Wu, J.Z. and Chakrabortty, R.K., 2021. Solving the problem of class imbalance in the prediction of hotel cancelations: A hybridized machine learning approach. *Processes*, *9*(10), p.1713.

Saputro, P.H. and Nanang, H., 2021. Exploratory data analysis & booking cancelation prediction on hotel booking demands datasets. *Journal of Applied Data Sciences*, *2*(1), pp.40-56.

Hermawan, A., Saputra, A., Lailinajma, N., Julianti, R., Hartanto, T. and Daniel, T.K., 2025. Predicting Hotel Booking Cancellations Using Machine Learning for Revenue Optimization. *Router: Jurnal Teknik Informatika dan Terapan*, *3*(1), pp.37-48.

Shirisha, N., Anusha, K., Kiran, A. and Buavani, Y.T.S., 2023, January. Prediction of hotel booking & cancellation using machine learning algorithms. In *2023 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-4). IEEE.

Rahmawati, E., Nurohim, G.S., Agustina, C., Irawan, D. and Muttaqin, Z., 2024. Develompent of Machine Learning Model to Predict Hotel Room Reservation Cancellations. *Jurnal Teknologi Informasi dan Terapan (J-TIT)*, *11*(2).

Putro, N.A., Septian, R., Widiastuti, W., Maulidah, M. and Pardede, H.F., 2021. Prediction of hotel booking cancellation using deep neural network and logistic regression algorithm. *Jurnal Techno Nusa Mandiri*, *18*(1), pp.1-8.

Tian, X., Pan, B., Bai, L. and Mo, D., 2023. Md-pred: A multidimensional hybrid prediction model based on machine learning for hotel booking cancellation prediction. *International Journal of Pattern Recognition and Artificial Intelligence*, *37*(05), p.2351009.

Liu, Z., Jiang, P., Wang, J., Du, Z., Niu, X. and Zhang, L., 2023. Hospitality order cancellation prediction from a profit-driven perspective. *International Journal of Contemporary Hospitality Management*, *35*(6), pp.2084-2112.

Prabha, R., Senthil, G.A., Nisha, A.S.A., Snega, S., Keerthana, L. and Sharmitha, S., 2022, December. Comparison of machine learning algorithms for hotel booking cancellation in automated method. In *2022 International Conference on Computer, Power and Communications (ICCPC)* (pp. 413-418). IEEE.

Yoo, M., Singh, A.K. and Loewy, N., 2023. Predicting hotel booking cancelation with machine learning techniques. *Journal of Hospitality and Tourism Technology*, *15*(1), pp.54-69.

Prasetya, J., Fallo, S.I. and Aprihartha, M.A., 2024. Stacking Machine Learning Model for Predict Hotel Booking Cancellations. *Jurnal Matematika, Statistika dan Komputasi*, *20*(3), pp.525-537.

Nababan, A.A., Jannah, M. and Nababan, A.H., 2022. Prediction Of Hotel Booking Cancellation Using K-Nearest Neighbors (K-Nn) Algorithm And Synthetic Minority Over-Sampling Technique (Smote). *INFOKUM*, *10*(03), pp.50-56.

Rakesh, M.V., Kumar, S.P. and Aishwarya, R., 2022, February. Hotel Booking Cancelation Prediction using ML algorithms. In *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)* (pp. 466-471). IEEE.

Chen, Y., Ding, C., Ye, H. and Zhou, Y., 2022, March. Comparison and analysis of machine learning models to predict hotel booking cancellation. In *2022 7th International Conference on Financial Innovation and Economic Development (ICFIED 2022)* (pp. 1363-1370). Atlantis Press.

Kumar[1], A., Prasad, U., Tiwari, R.K. and Pandey[1], V., 2023, July. Check for updates Research Study: Data Preprocessing Using Machine Learning for Prediction of Booking Cancellations. In *Recent Trends in Artificial Intelligence and IoT: First International Conference, ICAII 2022, Jamshedpur, India, April 4-5, 2023, Revised Selected Papers* (p. 164). Springer Nature.

Febrian, Y.Y., Wijaya, D.R. and Ervina, E., 2024, February. Hotel Reservation Cancellation Prediction using Boosting Model. In *2024 2nd International Conference on Software Engineering and Information Technology (ICoSEIT)* (pp. 138-143). IEEE.

Bhardwaj, A., Yadav, T. and Chaudhary, R., 2024, June. Predicting Hotel Booking Cancellations using Machine Learning Techniques. In *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (pp. 1-6). IEEE.

Ahmed, S., Chowdhury, S. and Rahman, R.M., 2024, August. Hotel Booking Cancellation with Visual Analytics. In *2024 IEEE 12th International Conference on Intelligent Systems (IS)* (pp. 1-8). IEEE.

Appendix

GitHub Link: **https://github.com/Lucky122002/Machine_Learning.git**

```
“# # Importing necessary libraries


# In[28]:



import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier



# # Load dataset


# In[29]:



df = pd.read_csv("booking.csv")



# # Display basic info


# In[30]:
```

```
print(df.info())
print(df.head())


# # Handling missing values

# In[31]:


df.fillna(df.median(numeric_only=True), inplace=True)    # Fill numerical columns with median
df.fillna(df.mode().iloc[0], inplace=True)


# # Convert date column to datetime

# In[33]:


df["date of reservation"] = pd.to_datetime(df["date of reservation"], dayfirst=True, errors='coerce')


# # Encode categorical features

# In[34]:


label_encoders = {}
categorical_cols = ["type of meal", "room type", "market segment type", "booking status"]
```

```
for col in categorical_cols:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le



# # Define target variable and features

# In[35]:



X = df.drop(columns=["Booking_ID", "booking status", "date of reservation"])
y = df["booking status"]



# # Train-test split

# In[36]:



X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)



# # Booking Status Distribution

# In[37]:



plt.figure(figsize=(6, 4))
sns.countplot(x=df["booking status"])
plt.title("Booking Status Distribution")
plt.show()



# # Lead Time Distribution b Bookin Status


# In[38]:



plt.figure(figsize=(8, 5))
sns.histplot(df, x="lead time", hue="booking status", kde=True, bins=30)
plt.title("Lead Time Distribution by Booking Status")
plt.show()



# # Special Requests and Booking Status


# In[39]:



plt.figure(figsize=(6, 4))
sns.boxplot(x="booking status", y="special requests", data=df)
plt.title("Special Requests and Booking Status")
plt.show()



# # Average Price and Booking Status


# In[40]:



plt.figure(figsize=(8, 5))
```

```
sns.boxplot(x="booking status", y="average price", data=df)
plt.title("Average Price and Booking Status")
plt.show()


# # Market Segment Type and Booking Status

# In[41]:


plt.figure(figsize=(6, 4))
sns.countplot(x="market segment type", hue="booking status", data=df)
plt.title("Market Segment Type and Booking Status")
plt.xticks(rotation=45)
plt.show()


# # Room Type and Booking Status

# In[42]:


plt.figure(figsize=(6, 4))
sns.countplot(x="room type", hue="booking status", data=df)
plt.title("Room Type and Booking Status")
plt.xticks(rotation=45)
plt.show()


# # Train Decision Tree Model
# In[43]:



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



# # Decision Tree Performance

# In[44]:



print("\nDecision Tree Performance:")
print(classification_report(y_test, y_pred_dt))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_dt))



# # Train Random Forest Model

# In[45]:



rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
y_pred_rf = rf_model.predict(X_test)



# # Random Forest Performance

# In[46]:
```

```
print("\nRandom Forest Performance:")
print(classification_report(y_test, y_pred_rf))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_rf))


# In[49]:


# Feature Importance in Random Forest Model


# In[47]:


feature_importances = pd.Series(rf_model.feature_importances_, index=X.columns)
feature_importances.sort_values(ascending=False).plot(kind='bar', figsize=(10, 6), color='teal')
plt.title("Feature Importance in Random Forest Model")
plt.show()”
```