Lab 8: Define and Solve an ML Problem of Your Choosing

In [18]:
```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
```

In this lab assignment, you will follow the machine learning life cycle and implement a model to solve a machine learning problem of your choosing. You will select a data set and choose a predictive problem that the data set supports. You will then inspect the data with your problem in mind and begin to formulate a project plan. You will then implement the machine learning project plan.

You will complete the following tasks:

1. Build Your DataFrame
2. Define Your ML Problem
3. Perform exploratory data analysis to understand your data.
4. Define Your Project Plan
5. Implement Your Project Plan:
    - Prepare your data for your model.
    - Fit your model to the training data and evaluate your model.
    - Improve your model's performance.

Part 1: Build Your DataFrame

You will have the option to choose one of four data sets that you have worked with in this program:

- The "census" data set that contains Census information from 1994: `censusData.csv`
- Airbnb NYC "listings" data set: `airbnbListingsData.csv`
- World Happiness Report (WHR) data set: `WHR2018Chapter2OnlineData.csv`
- Book Review data set: `bookReviewsData.csv`

Note that these are variations of the data sets that you have worked with in this program. For example, some do not include some of the preprocessing necessary for specific models.

Load a Data Set and Save it as a Pandas DataFrame

The code cell below contains filenames (path + filename) for each of the four data sets available to you.

Task: In the code cell below, use the same method you have been using to load the data using `pd.read_csv()` and save it to DataFrame `df`.

You can load each file as a new DataFrame to inspect the data before choosing your data set.

In [19]:
```
# File names of the four data sets
adultDataSet_filename = os.path.join(os.getcwd(), "data", "censusData.csv")
airbnbDataSet_filename = os.path.join(os.getcwd(), "data", "airbnbListingsDa
WHRDataSet_filename = os.path.join(os.getcwd(), "data", "WHR2018Chapter2Onli
bookReviewDataSet_filename = os.path.join(os.getcwd(), "data", "bookReviewsD


df = pd.read_csv(airbnbDataSet_filename)# YOUR CODE HERE

df
```

Out[19]:

	name	description	neighborhood_overview	host_name	host_location
0	Skylit Midtown Castle	Beautiful, spacious skylit studio in the heart...	Centrally located in the heart of Manhattan ju...	Jennifer	New York, New York, United States
1	Whole flr w/private bdrm, bath & kitchen(pls r...	Enjoy 500 s.f. top floor in 1899 brownstone, w...	Just the right mix of urban center and local n...	LisaRoxanne	New York, New York, United States
2	Spacious Brooklyn Duplex, Patio + Garden	We welcome you to stay in our lovely 2 br dupl...	NaN	Rebecca	Brooklyn, New York, United States
3	Large Furnished Room Near B'way	Please don't expect the luxury here just a bas...	Theater district, many restaurants around here.	Shunichi	New York, New York, United States
4	Cozy Clean Guest Room - Family Apt	Our best guests are seeking a safe, clean, spa...	Our neighborhood is full of restaurants and ca...	MaryEllen	New York, New York, United States
...	...	...	...	...	...
28017	Astoria Luxury suite 2A	THIS LOVELY HOME IS THE SPACIOUS SUITE WITH PR...	NaN	Vicky	Queens, New York, United States
28018	Newly renovated suite in the heart of Williams...	Just fully renovated from head to toe. On the ...	NaN	Samuel	New York, New York, United States
28019	Perfect Room to Stay in Brooklyn! Near Metro!	Amazing and comfortable space in Brooklyn, sam...	NaN	Carlos	US
28020	New Beautiful Modern One Bedroom	This stylish place to stay is perfect for a gr...	NaN	Lexia	New York, New York, United States

	name	description	neighborhood_overview	host_name	host_location
	in Brooklyn				
28021	Large, modern, private 1 bedroom in beach condo	Private bedroom on its own floor with very lar...	Beach, surf shop, stop and shop, Dunkin' Donut...	Justine	US

28022 rows × 50 columns

Part 2: Define Your ML Problem

Next you will formulate your ML Problem. In the markdown cell below, answer the following questions:

1. List the data set you have chosen.
2. What will you be predicting? What is the label?
3. Is this a supervised or unsupervised learning problem? Is this a clustering, classification or regression problem? Is it a binary classificaiton or multi-class classifiction problem?
4. What are your features? (note: this list may change after your explore your data)
5. Explain why this is an important problem. In other words, how would a company create value with a model that predicts this label?

Dataset Chosen

Airbnb Listings Data Set

Prediction Task

Prediction: Whether an Airbnb listing has availability. **Label**: `has_availability`

Type of Learning Problem

Learning Type: Supervised learning
Problem Type: Classification
Classification Type: Binary classification (True or False)

Features

Initially considered features:

- `host_response_rate`

- `host_acceptance_rate`
- `host_is_superhost`
- `host_listings_count`
- `review_scores_communication`
- `review_scores_location`
- `review_scores_value`
- `instant_bookable`
- `calculated_host_listings_count`
- `calculated_host_listings_count_entire_homes`
- `calculated_host_listings_count_private_rooms`
- `calculated_host_listings_count_shared_rooms`
- `reviews_per_month`
- `n_host_verifications`

Note: This list may change after data exploration and preprocessing.

Importance of the Problem

Value Creation:

- **Improved Guest Experience**: Ensures guests see available listings, reducing frustration and enhancing satisfaction.
- **Optimized Host Management**: Helps hosts manage listings better, adjusting strategies to maximize occupancy.
- **Platform Efficiency**: Optimizes search results, leading to higher conversion rates and better resource utilization.
- **Revenue Maximization**: Higher occupancy rates mean more revenue for both the platform and hosts.
- **Strategic Insights**: Provides valuable insights into factors affecting availability, helping hosts improve listing performance.

Part 3: Understand Your Data

The next step is to perform exploratory data analysis. Inspect and analyze your data set with your machine learning problem in mind. Consider the following as you inspect your data:

1. What data preparation techniques would you like to use? These data preparation techniques may include:

    - addressing missingness, such as replacing missing values with means
    - finding and replacing outliers
    - renaming features and labels
    - finding and replacing outliers

- performing feature engineering techniques such as one-hot encoding on categorical features
- selecting appropriate features and removing irrelevant features
- performing specific data cleaning and preprocessing techniques for an NLP problem
- addressing class imbalance in your data sample to promote fair AI

2. What machine learning model (or models) you would like to use that is suitable for your predictive problem and data?

   - Are there other data preparation techniques that you will need to apply to build a balanced modeling data set for your problem and model? For example, will you need to scale your data?

3. How will you evaluate and improve the model's performance?

   - Are there specific evaluation metrics and methods that are appropriate for your model?

Think of the different techniques you have used to inspect and analyze your data in this course. These include using Pandas to apply data filters, using the Pandas `describe()` method to get insight into key statistics for each column, using the Pandas `dtypes` property to inspect the data type of each column, and using Matplotlib and Seaborn to detect outliers and visualize relationships between features and labels. If you are working on a classification problem, use techniques you have learned to determine if there is class imbalance.

Task: Use the techniques you have learned in this course to inspect and analyze your data. You can import additional packages that you have used in this course that you will need to perform this task.

Note: You can add code cells if needed by going to the **Insert** menu and clicking on **Insert Cell Below** in the drop-drown menu.

Part 4: Define Your Project Plan

Now that you understand your data, in the markdown cell below, define your plan to implement the remaining phases of the machine learning life cycle (data preparation, modeling, evaluation) to solve your ML problem. Answer the following questions:

- Do you have a new feature list? If so, what are the features that you chose to keep and remove after inspecting the data?
- Explain different data preparation techniques that you will use to prepare your data for modeling.
- What is your model (or models)?

- Describe your plan to train your model, analyze its performance and then improve the model. That is, describe your model building, validation and selection plan to produce a model that generalizes well to new data.

Plan for Implementing Remaining Phases of the Machine Learning Life Cycle

Feature List

After inspecting the data, I decided to keep the following features:

- Numerical: `host_response_rate`, `host_acceptance_rate`, `host_listings_count`, `host_total_listings_count`, `accommodates`, `bathrooms`, `bedrooms`, `beds`, `price`, `minimum_nights`, `maximum_nights`, `availability_30`, `availability_60`, `availability_90`, `availability_365`, `number_of_reviews`, `review_scores_rating`, `review_scores_cleanliness`, `review_scores_checkin`, `review_scores_communication`, `review_scores_location`, `review_scores_value`, `calculated_host_listings_count`, `reviews_per_month`
- One-hot encoded categorical: `host_is_superhost`, `instant_bookable`

Removed features include columns with high missing values, irrelevant columns, and those dropped due to preprocessing.

Data Preparation Techniques

1. **Handling Missing Values**: Fill missing numeric values with the mean and categorical values with the mode.
2. **Convert Percentages**: Convert percentage values to float.
3. **Binary Conversion**: Convert boolean columns to integers.
4. **Standardization**: Scale numerical features using StandardScaler.
5. **Remove Constant Features**: Use `VarianceThreshold` to remove features with zero variance.
6. **One-hot Encoding**: One-hot encode binary categorical variables.

Model(s)

1. **Logistic Regression**: For initial model building and evaluation.
2. **Random Forest Classifier**: For improved performance with hyperparameter tuning.

Plan for Model Training, Analysis, and Improvement

1. **Model Building**:
    - **Initial Model**: Train a Logistic Regression model to set a baseline.
    - **Feature Selection**: Use `SelectKBest` to identify the top 10 features.

2. **Model Validation**:
   - **Cross-Validation**: Use 5-fold cross-validation to evaluate model performance.
   - **Hyperparameter Tuning**: Use GridSearchCV to find the best hyperparameters for the Random Forest model.
3. **Model Analysis**:
   - **Performance Metrics**: Evaluate using accuracy, precision, recall, F1 score, and ROC AUC score.
4. **Model Improvement**:
   - **Feature Engineering**: Explore additional feature creation or transformation if needed.
   - **Ensemble Methods**: Consider combining multiple models if individual model performance is insufficient.
5. **Model Selection**:
   - **Best Model Selection**: Select the model with the best cross-validation performance.
   - **Final Evaluation**: Evaluate the final model on a separate test set to ensure generalizability.

This plan will guide the preparation, modeling, and iterative improvement to build a robust model that generalizes well to new data.

Part 5: Implement Your Project Plan

Task: In the code cell below, import additional packages that you have used in this course that you will need to implement your project plan.

In [20]:
```
import pandas as pd
import os
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f
from sklearn.impute import SimpleImputer
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.feature_selection import SelectKBest, f_classif
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import OneHotEncoder
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import roc_curve, auc, precision_recall_curve
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix, roc_curve, auc
```

Task: Use the rest of this notebook to carry out your project plan.

You will:

1. Prepare your data for your model.
2. Fit your model to the training data and evaluate your model.
3. Improve your model's performance by performing model selection and/or feature selection techniques to find best model for your problem.

Add code cells below and populate the notebook with commentary, code, analyses, results, and figures as you see fit.

In [21]:
```
# Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.feature_selection import SelectKBest, f_classif, VarianceThresh
from sklearn.metrics import accuracy_score, precision_score, recall_score, f

# Load the Airbnb dataset into a pandas DataFrame
airbnb_df = df # Use the correct path to the dataset

# Drop unnecessary columns
airbnb_df.drop(columns=[
    'name', 'room_type', 'neighbourhood_group_cleansed', 'amenities', 'host_
    'description', 'neighborhood_overview', 'host_name', 'host_about'
], inplace=True)

# Display the first few rows of the DataFrame to confirm successful loading
print("Airbnb Listings Data Set:")
print(airbnb_df.head())

# Check for missing values
print(airbnb_df.isnull().sum())

# Get a summary of the dataset
print(airbnb_df.info())
print(airbnb_df.describe())

# Fill missing numeric values with the mean
for col in ['host_response_rate', 'host_acceptance_rate', 'reviews_per_month
    airbnb_df[col].fillna(airbnb_df[col].mean(), inplace=True)

# Drop rows with missing target values
airbnb_df.dropna(subset=['has_availability'], inplace=True)
```

```
# Fill other missing categorical values with the mode
for col in ['host_is_superhost', 'instant_bookable']:
    airbnb_df[col].fillna(airbnb_df[col].mode()[0], inplace=True)

def convert_percentage_to_float(x):
    return float(x.strip('%')) / 100 if isinstance(x, str) else x

airbnb_df['host_response_rate'] = airbnb_df['host_response_rate'].apply(conv
airbnb_df['host_acceptance_rate'] = airbnb_df['host_acceptance_rate'].apply(

# Convert 'has_availability' to binary (True -> 1, False -> 0)
airbnb_df['has_availability'] = airbnb_df['has_availability'].apply(lambda x

# Convert boolean columns to integers
bool_cols = ['host_has_profile_pic', 'host_identity_verified', 'instant_book
for col in bool_cols:
    airbnb_df[col] = airbnb_df[col].astype(int)

# One-hot encode binary categorical variables
airbnb_df = pd.get_dummies(airbnb_df, columns=['host_is_superhost', 'instant

# List of numerical features to scale
num_features = [
    'host_response_rate', 'host_acceptance_rate', 'host_listings_count',
    'host_total_listings_count', 'host_has_profile_pic',
    'host_identity_verified', 'accommodates', 'bathrooms', 'bedrooms',
    'beds', 'price', 'minimum_nights', 'maximum_nights',
    'minimum_minimum_nights', 'maximum_minimum_nights',
    'minimum_maximum_nights', 'maximum_maximum_nights',
    'minimum_nights_avg_ntm', 'maximum_nights_avg_ntm',
    'availability_30', 'availability_60', 'availability_90',
    'availability_365', 'number_of_reviews', 'number_of_reviews_ltm',
    'number_of_reviews_l30d', 'review_scores_rating',
    'review_scores_cleanliness', 'review_scores_checkin',
    'review_scores_communication', 'review_scores_location',
    'review_scores_value', 'calculated_host_listings_count',
    'calculated_host_listings_count_entire_homes',
    'calculated_host_listings_count_private_rooms',
    'calculated_host_listings_count_shared_rooms', 'reviews_per_month',
    'n_host_verifications'
]

# Remove the target column from the numerical features list
if 'has_availability' in num_features:
    num_features.remove('has_availability')

# Verify numerical features exist in DataFrame
for feature in num_features:
    if feature not in airbnb_df.columns:
        print(f"Warning: Feature '{feature}' not found in DataFrame")

# Standardize the numerical features
scaler = StandardScaler()
airbnb_df[num_features] = scaler.fit_transform(airbnb_df[num_features])

# Check if all columns are numeric
```

```
print(airbnb_df.dtypes)

# Check the columns in the DataFrame
print("DataFrame columns:", airbnb_df.columns)

# Define numerical and categorical columns
numerical_cols = airbnb_df.select_dtypes(include=['int64', 'float64']).colum
categorical_cols = airbnb_df.select_dtypes(include=['object', 'bool', 'uint8

# Define the target column
target_column = 'has_availability'

# Remove the target column from numerical and categorical features list if i
if target_column in numerical_cols:
    numerical_cols.remove(target_column)
if target_column in categorical_cols:
    categorical_cols.remove(target_column)

# Verify columns before preprocessing
print("Numerical columns before preprocessing:", numerical_cols)
print("Categorical columns before preprocessing:", categorical_cols)

# Define preprocessing steps
preprocessor = ColumnTransformer(
    transformers=[
        ('num', Pipeline(steps=[
            ('imputer', SimpleImputer(strategy='mean')),
            ('scaler', StandardScaler())
        ]), numerical_cols),
        ('cat', Pipeline(steps=[
            ('imputer', SimpleImputer(strategy='most_frequent')),
            ('onehot', OneHotEncoder(handle_unknown='ignore'))
        ]), categorical_cols)
    ]
)

# Define features and target variable
X = airbnb_df.drop('has_availability', axis=1)
y = airbnb_df['has_availability']

# Verify columns before transformation
print("Columns in X before transformation:", X.columns)

try:
    X_transformed = preprocessor.fit_transform(X)
except ValueError as e:
    missing_columns = set(numerical_cols + categorical_cols) - set(X.columns
    raise ValueError(f"The following columns are missing in the DataFrame: {

# Verify there are no NaN values in the transformed data
if pd.DataFrame(X_transformed).isnull().sum().sum() > 0:
    raise ValueError("There are still missing values in the transformed feat

# Verify transformation output
print("Initial X_transformed shape:", X_transformed.shape)
```

```
# Remove constant features
constant_filter = VarianceThreshold(threshold=0)
X_transformed = constant_filter.fit_transform(X_transformed)

# Verify shape after removing constant features
print("Shape after VarianceThreshold:", X_transformed.shape)

# Select top k features
k = 10
selector = SelectKBest(score_func=f_classif, k=k)

if pd.DataFrame(X_transformed).isnull().sum().sum() > 0:
    raise ValueError("There are still missing values in the transformed feat

X_new = selector.fit_transform(X_transformed, y)

# Verify shape after feature selection
print("Shape after SelectKBest:", X_new.shape)

# Split the data into training and testing sets with selected features
X_train_new, X_test_new, y_train_new, y_test_new = train_test_split(X_new, y

# Verify shapes after train-test split
print("Shapes after train-test split - X_train_new:", X_train_new.shape, "X_

# Define the model
rf = RandomForestClassifier(random_state=42)

# Define the parameter grid
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

# Debug statement to confirm we reached here
print("Starting GridSearchCV")

# Perform grid search
try:
    grid_search = GridSearchCV(estimator=rf, param_grid=param_grid, cv=5, sc
    grid_search.fit(X_train_new, y_train_new)
    print("GridSearchCV completed")
except Exception as e:
    print("Error during GridSearchCV:", e)
    raise

# Get the best model
best_rf = grid_search.best_estimator_

# Make predictions with the best model
y_pred_new = best_rf.predict(X_test_new)

# Evaluate the best model
print("Random Forest Classifier with Grid Search:")
```

```
print("Accuracy:", accuracy_score(y_test_new, y_pred_new))
print("Precision:", precision_score(y_test_new, y_pred_new))
print("Recall:", recall_score(y_test_new, y_pred_new))
print("F1 Score:", f1_score(y_test_new, y_pred_new))
print("ROC AUC Score:", roc_auc_score(y_test_new, y_pred_new))
```

```
Airbnb Listings Data Set:
   host_response_rate  host_acceptance_rate  host_is_superhost  \
0                0.80                  0.17               True   
1                0.09                  0.69               True   
2                1.00                  0.25               True   
3                1.00                  1.00               True   
4                 NaN                   NaN               True   

   host_listings_count  host_total_listings_count  host_has_profile_pic  \
0                  8.0                        8.0                  True   
1                  1.0                        1.0                  True   
2                  1.0                        1.0                  True   
3                  1.0                        1.0                  True   
4                  1.0                        1.0                  True   

   host_identity_verified  accommodates  bathrooms  bedrooms  ...  \
0                    True             1        1.0       NaN  ...   
1                    True             3        1.0       1.0  ...   
2                    True             4        1.5       2.0  ...   
3                    True             2        1.0       1.0  ...   
4                    True             1        1.0       1.0  ...   

   review_scores_communication  review_scores_location  review_scores_value  
\
0                         4.79                    4.86                 4.41   
1                         4.80                    4.71                 4.64   
2                         5.00                    4.50                 5.00   
3                         4.42                    4.87                 4.36   
4                         4.95                    4.94                 4.92   

   instant_bookable  calculated_host_listings_count  \
0             False                               3   
1             False                               1   
2             False                               1   
3             False                               1   
4             False                               1   

   calculated_host_listings_count_entire_homes  \
0                                            3   
1                                            1   
2                                            1   
3                                            0   
4                                            0   

   calculated_host_listings_count_private_rooms  \
0                                             0   
1                                             0   
2                                             0   
3                                             1   
4                                             1   

   calculated_host_listings_count_shared_rooms  reviews_per_month  \
0                                            0               0.33   
1                                            0               4.86   
2                                            0               0.02   
3                                            0               3.68   
```

```
4                                            0                0.87

   n_host_verifications
0                     9
1                     6
2                     3
3                     4
4                     7

[5 rows x 41 columns]
host_response_rate                              11843
host_acceptance_rate                            11113
host_is_superhost                                   0
host_listings_count                                 0
host_total_listings_count                           0
host_has_profile_pic                                0
host_identity_verified                              0
accommodates                                        0
bathrooms                                           0
bedrooms                                         2918
beds                                             1354
price                                               0
minimum_nights                                      0
maximum_nights                                      0
minimum_minimum_nights                              0
maximum_minimum_nights                              0
minimum_maximum_nights                              0
maximum_maximum_nights                              0
minimum_nights_avg_ntm                              0
maximum_nights_avg_ntm                              0
has_availability                                    0
availability_30                                     0
availability_60                                     0
availability_90                                     0
availability_365                                    0
number_of_reviews                                   0
number_of_reviews_ltm                               0
number_of_reviews_l30d                              0
review_scores_rating                                0
review_scores_cleanliness                           0
review_scores_checkin                               0
review_scores_communication                         0
review_scores_location                              0
review_scores_value                                 0
instant_bookable                                    0
calculated_host_listings_count                      0
calculated_host_listings_count_entire_homes         0
calculated_host_listings_count_private_rooms        0
calculated_host_listings_count_shared_rooms         0
reviews_per_month                                   0
n_host_verifications                                0
dtype: int64
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28022 entries, 0 to 28021
Data columns (total 41 columns):
 #   Column                                        Non-Null Count  Dtype
```

```
 ---  ------                                       --------------  -----
 0   host_response_rate                           16179 non-null  float64
 1   host_acceptance_rate                         16909 non-null  float64
 2   host_is_superhost                            28022 non-null  bool
 3   host_listings_count                          28022 non-null  float64
 4   host_total_listings_count                    28022 non-null  float64
 5   host_has_profile_pic                         28022 non-null  bool
 6   host_identity_verified                       28022 non-null  bool
 7   accommodates                                 28022 non-null  int64
 8   bathrooms                                    28022 non-null  float64
 9   bedrooms                                     25104 non-null  float64
 10  beds                                         26668 non-null  float64
 11  price                                        28022 non-null  float64
 12  minimum_nights                               28022 non-null  int64
 13  maximum_nights                               28022 non-null  int64
 14  minimum_minimum_nights                       28022 non-null  float64
 15  maximum_minimum_nights                       28022 non-null  float64
 16  minimum_maximum_nights                       28022 non-null  float64
 17  maximum_maximum_nights                       28022 non-null  float64
 18  minimum_nights_avg_ntm                       28022 non-null  float64
 19  maximum_nights_avg_ntm                       28022 non-null  float64
 20  has_availability                             28022 non-null  bool
 21  availability_30                              28022 non-null  int64
 22  availability_60                              28022 non-null  int64
 23  availability_90                              28022 non-null  int64
 24  availability_365                             28022 non-null  int64
 25  number_of_reviews                            28022 non-null  int64
 26  number_of_reviews_ltm                        28022 non-null  int64
 27  number_of_reviews_l30d                       28022 non-null  int64
 28  review_scores_rating                         28022 non-null  float64
 29  review_scores_cleanliness                    28022 non-null  float64
 30  review_scores_checkin                        28022 non-null  float64
 31  review_scores_communication                  28022 non-null  float64
 32  review_scores_location                       28022 non-null  float64
 33  review_scores_value                          28022 non-null  float64
 34  instant_bookable                             28022 non-null  bool
 35  calculated_host_listings_count               28022 non-null  int64
 36  calculated_host_listings_count_entire_homes  28022 non-null  int64
 37  calculated_host_listings_count_private_rooms 28022 non-null  int64
 38  calculated_host_listings_count_shared_rooms  28022 non-null  int64
 39  reviews_per_month                            28022 non-null  float64
 40  n_host_verifications                         28022 non-null  int64
dtypes: bool(5), float64(21), int64(15)
memory usage: 7.8 MB
None
       host_response_rate  host_acceptance_rate  host_listings_count  \
count        16179.000000          16909.000000         28022.000000
mean             0.906901              0.791953            14.554778
std              0.227282              0.276732           120.721287
min              0.000000              0.000000             0.000000
25%              0.940000              0.680000             1.000000
50%              1.000000              0.910000             1.000000
75%              1.000000              1.000000             3.000000
max              1.000000              1.000000          3387.000000

       host_total_listings_count  accommodates     bathrooms      bedrooms
```

```
\
count            28022.000000  28022.000000  28022.000000  25104.000000
mean                14.554778      2.874491      1.142174      1.329708
std                120.721287      1.860251      0.421132      0.700726
min                  0.000000      1.000000      0.000000      1.000000
25%                  1.000000      2.000000      1.000000      1.000000
50%                  1.000000      2.000000      1.000000      1.000000
75%                  3.000000      4.000000      1.000000      1.000000
max               3387.000000     16.000000      8.000000     12.000000

               beds         price  minimum_nights  ...  review_scores_checki
n  \
count  26668.000000  28022.000000    28022.000000  ...           28022.00000
0
mean       1.629556    154.228749       18.689387  ...               4.81430
0
std        1.097104    140.816605       25.569151  ...               0.43860
3
min        1.000000     29.000000        1.000000  ...               0.00000
0
25%        1.000000     70.000000        2.000000  ...               4.81000
0
50%        1.000000    115.000000       30.000000  ...               4.96000
0
75%        2.000000    180.000000       30.000000  ...               5.00000
0
max       21.000000   1000.000000     1250.000000  ...               5.00000
0

       review_scores_communication  review_scores_location  \
count                 28022.000000            28022.000000
mean                      4.808041                4.750393
std                       0.464585                0.415717
min                       0.000000                0.000000
25%                       4.810000                4.670000
50%                       4.970000                4.880000
75%                       5.000000                5.000000
max                       5.000000                5.000000

       review_scores_value  calculated_host_listings_count  \
count         28022.000000                    28022.000000
mean              4.647670                        9.581900
std               0.518023                       32.227523
min               0.000000                        1.000000
25%               4.550000                        1.000000
50%               4.780000                        1.000000
75%               5.000000                        3.000000
max               5.000000                      421.000000

       calculated_host_listings_count_entire_homes  \
count                                 28022.000000
mean                                      5.562986
std                                      26.121426
min                                       0.000000
25%                                       0.000000
50%                                       1.000000
```

```
75%                                            1.000000
max                                          308.000000

       calculated_host_listings_count_private_rooms  \
count                                      28022.000000
mean                                           3.902077
std                                           17.972386
min                                            0.000000
25%                                            0.000000
50%                                            0.000000
75%                                            1.000000
max                                          359.000000

       calculated_host_listings_count_shared_rooms  reviews_per_month  \
count                                     28022.000000       28022.000000
mean                                          0.048283           1.758325
std                                           0.442459           4.446143
min                                           0.000000           0.010000
25%                                           0.000000           0.130000
50%                                           0.000000           0.510000
75%                                           0.000000           1.830000
max                                           8.000000         141.000000

       n_host_verifications
count          28022.000000
mean               5.169510
std                2.028497
min                1.000000
25%                4.000000
50%                5.000000
75%                7.000000
max               13.000000

[8 rows x 36 columns]
host_response_rate                              float64
host_acceptance_rate                            float64
host_listings_count                             float64
host_total_listings_count                       float64
host_has_profile_pic                            float64
host_identity_verified                          float64
accommodates                                    float64
bathrooms                                       float64
bedrooms                                        float64
beds                                            float64
price                                           float64
minimum_nights                                  float64
maximum_nights                                  float64
minimum_minimum_nights                          float64
maximum_minimum_nights                          float64
minimum_maximum_nights                          float64
maximum_maximum_nights                          float64
minimum_nights_avg_ntm                          float64
maximum_nights_avg_ntm                          float64
has_availability                                  int64
availability_30                                 float64
availability_60                                 float64
```

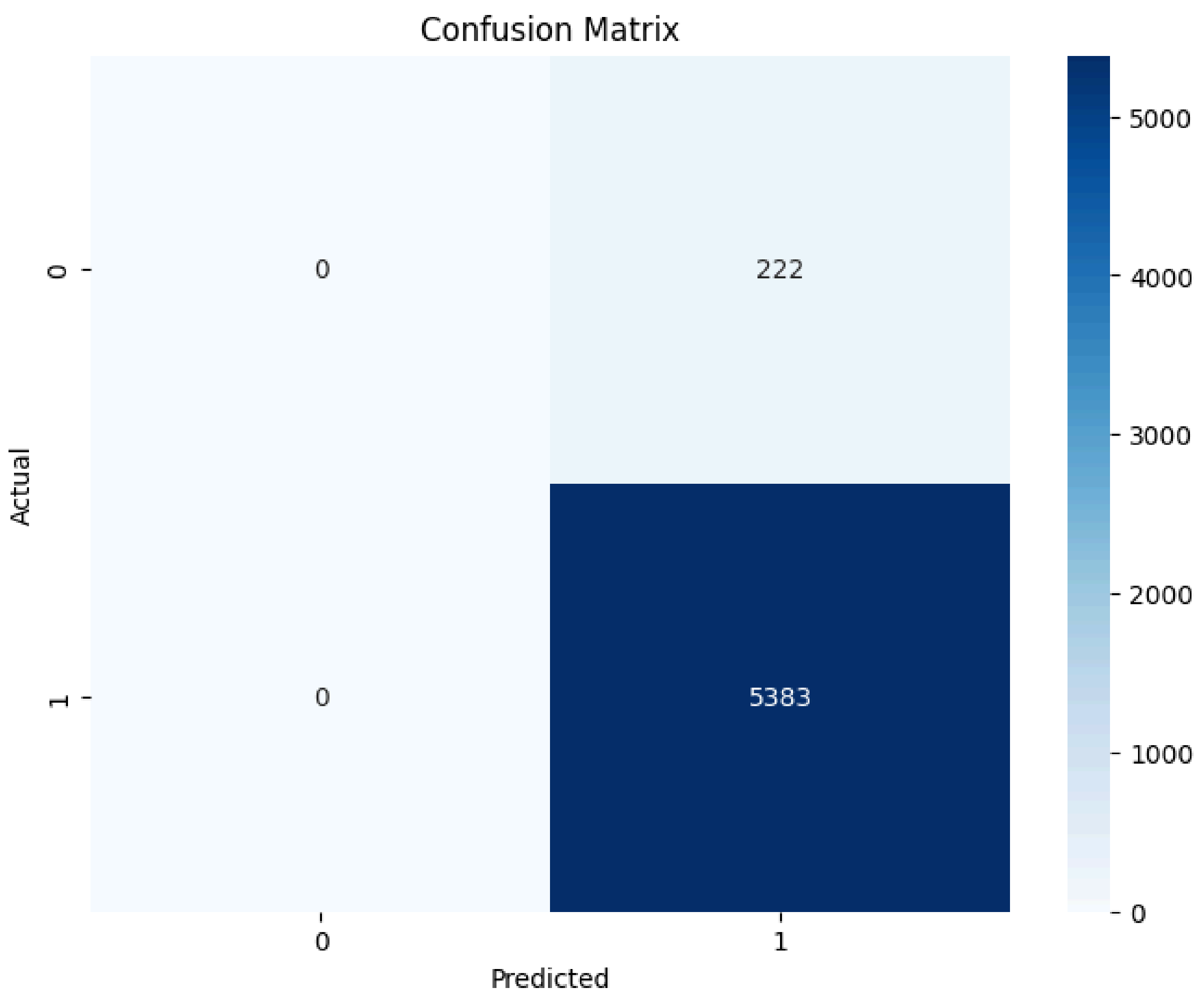
```
availability_90                                  float64
availability_365                                 float64
number_of_reviews                                float64
number_of_reviews_ltm                            float64
number_of_reviews_l30d                           float64
review_scores_rating                             float64
review_scores_cleanliness                        float64
review_scores_checkin                            float64
review_scores_communication                      float64
review_scores_location                           float64
review_scores_value                              float64
calculated_host_listings_count                   float64
calculated_host_listings_count_entire_homes     float64
calculated_host_listings_count_private_rooms    float64
calculated_host_listings_count_shared_rooms     float64
reviews_per_month                                float64
n_host_verifications                             float64
host_is_superhost_True                             uint8
instant_bookable_0                                 uint8
instant_bookable_1                                 uint8
dtype: object
DataFrame columns: Index(['host_response_rate', 'host_acceptance_rate', 'hos
t_listings_count',
       'host_total_listings_count', 'host_has_profile_pic',
       'host_identity_verified', 'accommodates', 'bathrooms', 'bedrooms',
       'beds', 'price', 'minimum_nights', 'maximum_nights',
       'minimum_minimum_nights', 'maximum_minimum_nights',
       'minimum_maximum_nights', 'maximum_maximum_nights',
       'minimum_nights_avg_ntm', 'maximum_nights_avg_ntm', 'has_availabilit
y',
       'availability_30', 'availability_60', 'availability_90',
       'availability_365', 'number_of_reviews', 'number_of_reviews_ltm',
       'number_of_reviews_l30d', 'review_scores_rating',
       'review_scores_cleanliness', 'review_scores_checkin',
       'review_scores_communication', 'review_scores_location',
       'review_scores_value', 'calculated_host_listings_count',
       'calculated_host_listings_count_entire_homes',
       'calculated_host_listings_count_private_rooms',
       'calculated_host_listings_count_shared_rooms', 'reviews_per_month',
       'n_host_verifications', 'host_is_superhost_True', 'instant_bookable_
0',
       'instant_bookable_1'],
      dtype='object')
Numerical columns before preprocessing: ['host_response_rate', 'host_accepta
nce_rate', 'host_listings_count', 'host_total_listings_count', 'host_has_pro
file_pic', 'host_identity_verified', 'accommodates', 'bathrooms', 'bedroom
s', 'beds', 'price', 'minimum_nights', 'maximum_nights', 'minimum_minimum_ni
ghts', 'maximum_minimum_nights', 'minimum_maximum_nights', 'maximum_maximum_
nights', 'minimum_nights_avg_ntm', 'maximum_nights_avg_ntm', 'availability_3
0', 'availability_60', 'availability_90', 'availability_365', 'number_of_rev
iews', 'number_of_reviews_ltm', 'number_of_reviews_l30d', 'review_scores_rat
ing', 'review_scores_cleanliness', 'review_scores_checkin', 'review_scores_c
ommunication', 'review_scores_location', 'review_scores_value', 'calculated_
host_listings_count', 'calculated_host_listings_count_entire_homes', 'calcul
ated_host_listings_count_private_rooms', 'calculated_host_listings_count_sha
red_rooms', 'reviews_per_month', 'n_host_verifications']
```

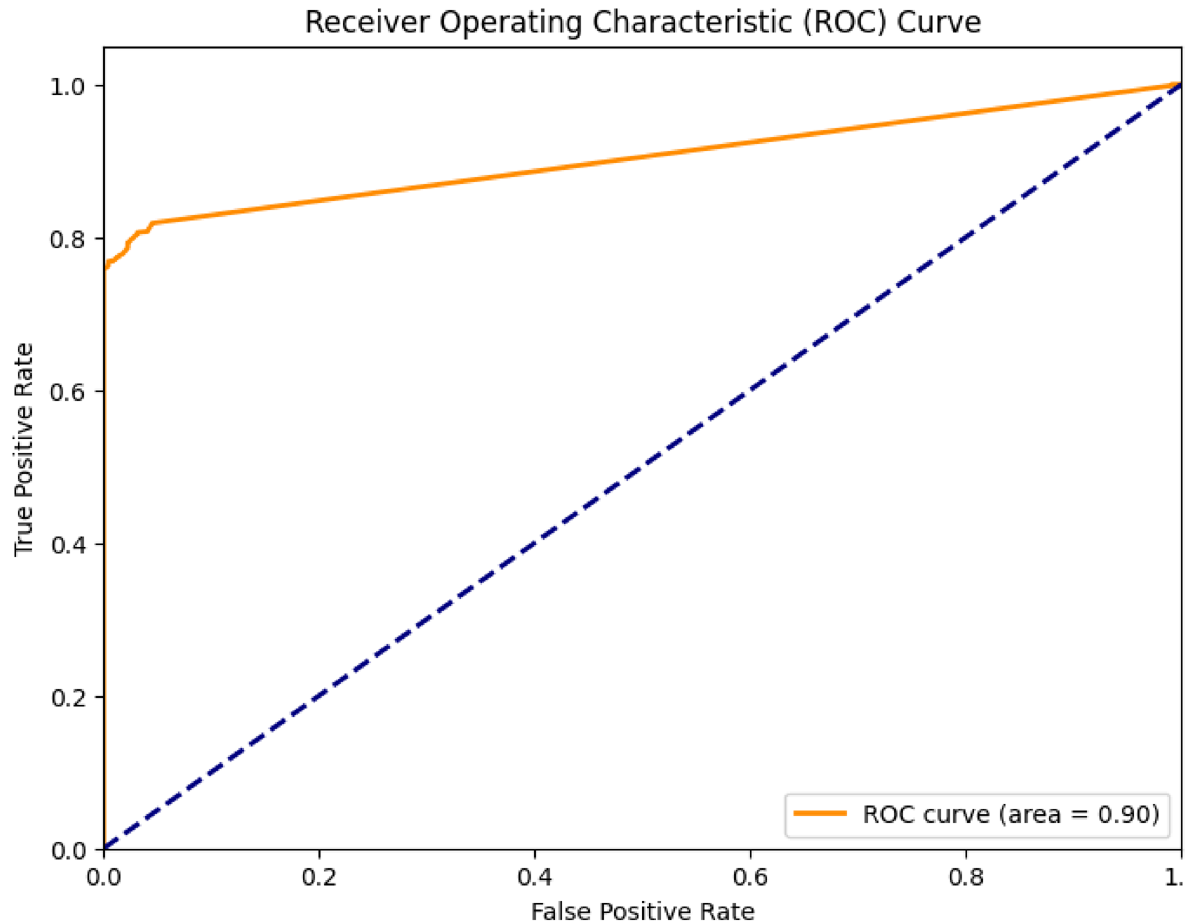
```
Categorical columns before preprocessing: ['host_is_superhost_True', 'instan
t_bookable_0', 'instant_bookable_1']
Columns in X before transformation: Index(['host_response_rate', 'host_accep
tance_rate', 'host_listings_count',
       'host_total_listings_count', 'host_has_profile_pic',
       'host_identity_verified', 'accommodates', 'bathrooms', 'bedrooms',
       'beds', 'price', 'minimum_nights', 'maximum_nights',
       'minimum_minimum_nights', 'maximum_minimum_nights',
       'minimum_maximum_nights', 'maximum_maximum_nights',
       'minimum_nights_avg_ntm', 'maximum_nights_avg_ntm', 'availability_3
0',
       'availability_60', 'availability_90', 'availability_365',
       'number_of_reviews', 'number_of_reviews_ltm', 'number_of_reviews_l30
d',
       'review_scores_rating', 'review_scores_cleanliness',
       'review_scores_checkin', 'review_scores_communication',
       'review_scores_location', 'review_scores_value',
       'calculated_host_listings_count',
       'calculated_host_listings_count_entire_homes',
       'calculated_host_listings_count_private_rooms',
       'calculated_host_listings_count_shared_rooms', 'reviews_per_month',
       'n_host_verifications', 'host_is_superhost_True', 'instant_bookable_
0',
       'instant_bookable_1'],
      dtype='object')
Initial X_transformed shape: (28022, 43)
Shape after VarianceThreshold: (28022, 40)
Shape after SelectKBest: (28022, 10)
Shapes after train-test split - X_train_new: (22417, 10) X_test_new: (5605,
10) y_train_new: (22417,) y_test_new: (5605,)
Starting GridSearchCV
GridSearchCV completed
Random Forest Classifier with Grid Search:
Accuracy: 0.960392506690455
Precision: 0.960392506690455
Recall: 1.0
F1 Score: 0.9797961412449946
ROC AUC Score: 0.5
```

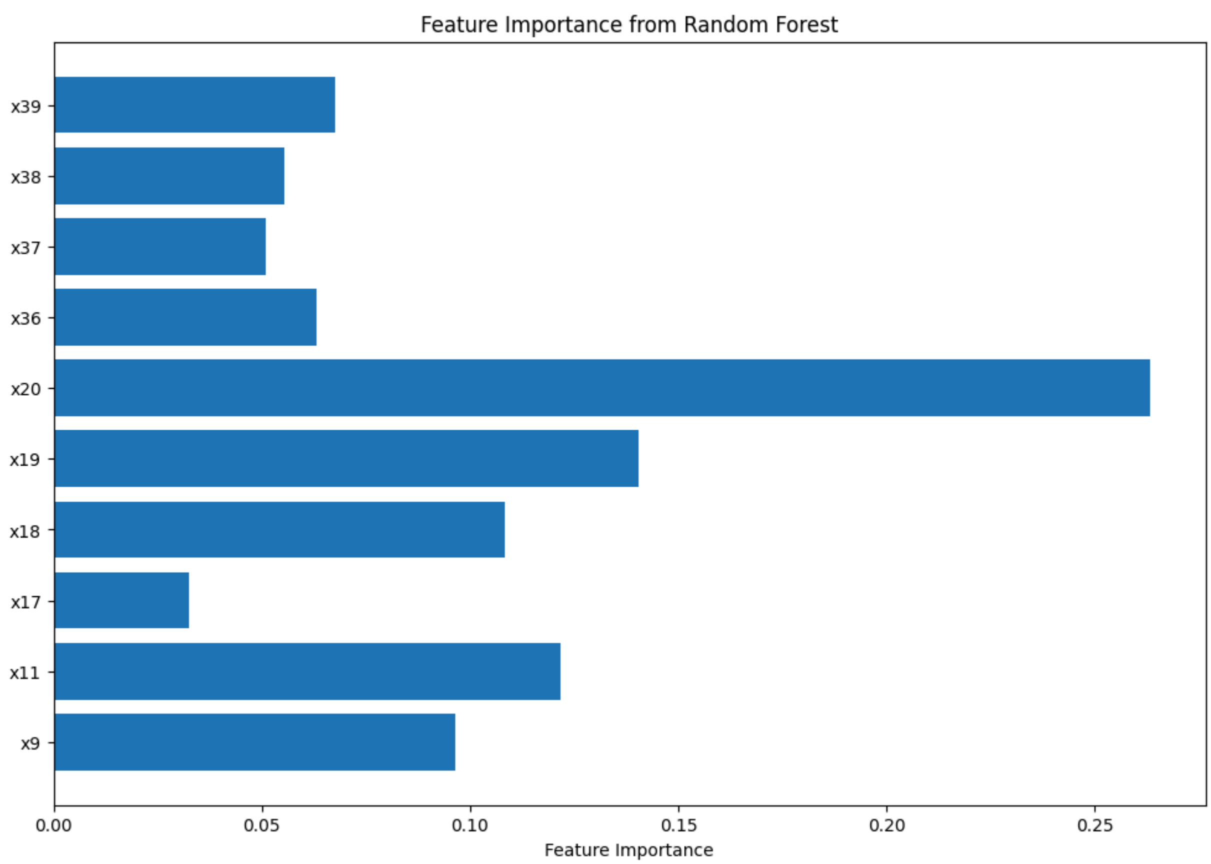
In [22]:

```python
# Plot confusion matrix
conf_matrix = confusion_matrix(y_test_new, y_pred_new)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

# Plot ROC curve
fpr, tpr, thresholds = roc_curve(y_test_new, grid_search.predict_proba(X_tes
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (area = {roc_
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
```

```
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc='lower right')
plt.show()

# Plot feature importance
feature_importances = best_rf.feature_importances_
features = selector.get_feature_names_out()

plt.figure(figsize=(12, 8))
plt.barh(range(len(feature_importances)), feature_importances, align='center
plt.yticks(range(len(feature_importances)), features)
plt.xlabel('Feature Importance')
plt.title('Feature Importance from Random Forest')
plt.show()
```




Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 0.90)


Feature Importance from Random Forest
x39
x38
x37
x36
x20
x19
x18
x17
x11
x9
0.00
0.05
0.10
0.15
0.20
0.25
Feature Importance

In [ ]: