Data Collection and Preprocessing Phase

Date	18 JULY 2024
Team ID	739977
Project Title	Unveiling Airbnb Price Patterns: Machine Learning For Forecasting
Maximum Marks	6 Marks

Data Exploration and Preprocessing Report

Dataset variables will be statistically analyzed to identify patterns and outliers, with Python employed for preprocessing tasks like normalization and feature engineering. Data cleaning will address missing values and outliers, ensuring quality for subsequent analysis and modeling, and forming a strong foundation for insights and predictions.

Section	Description
Data Overview	Dimension: 614 rows × 13 columns Descriptive statistics:

price distribution

Entire home/apt 55.7%

Shared room 2.9%

Private room 41.3%

NYC 43.65%

4.68% Boston

5.02% Chicago

7.67% LA

SF 30.30%

8.68% DC

cancellation_policy

cleaning_fee

bed type

Most popular Neighbourhood
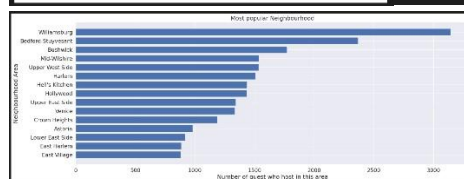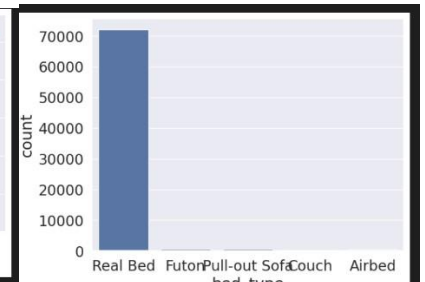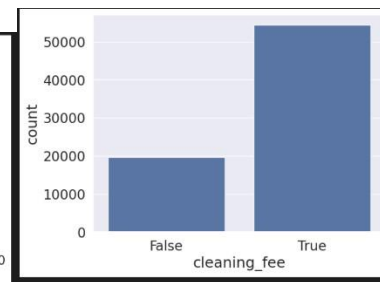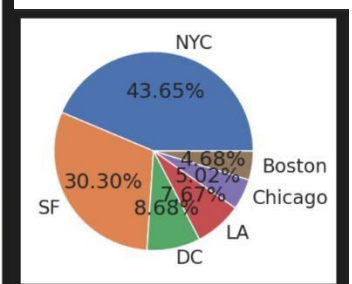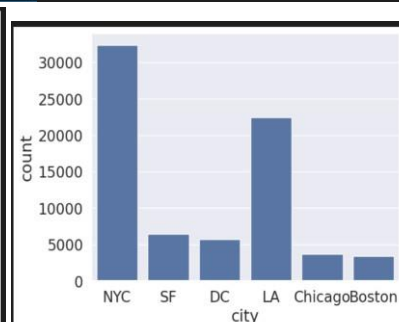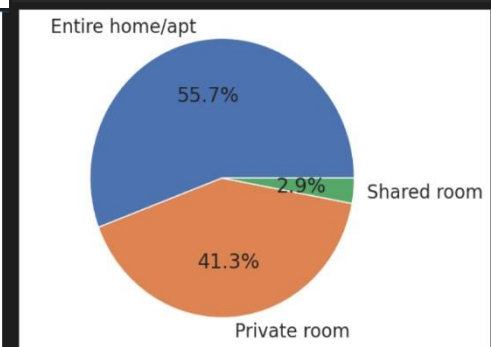
| Univariate Analysis | |

Bivariate Analysis

Density and distribution of prices

Density and distribution of prices

Density and Distribution of prices
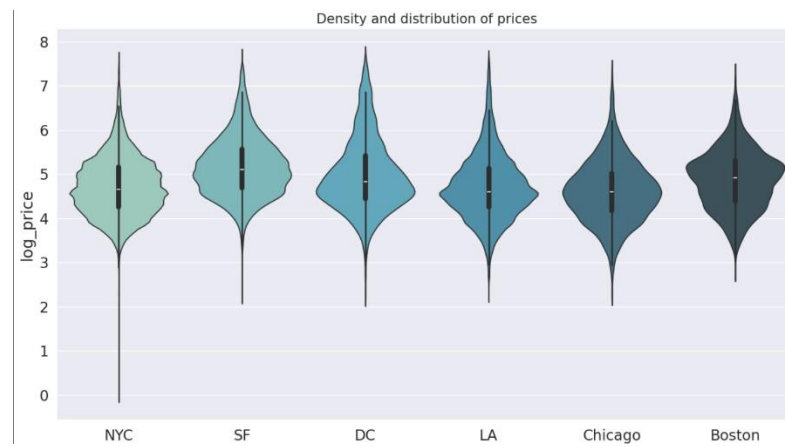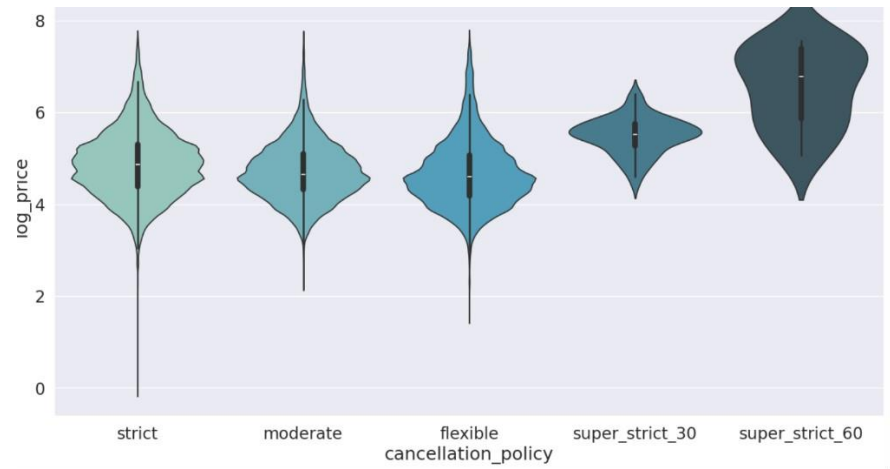
Multivariate
Analysis
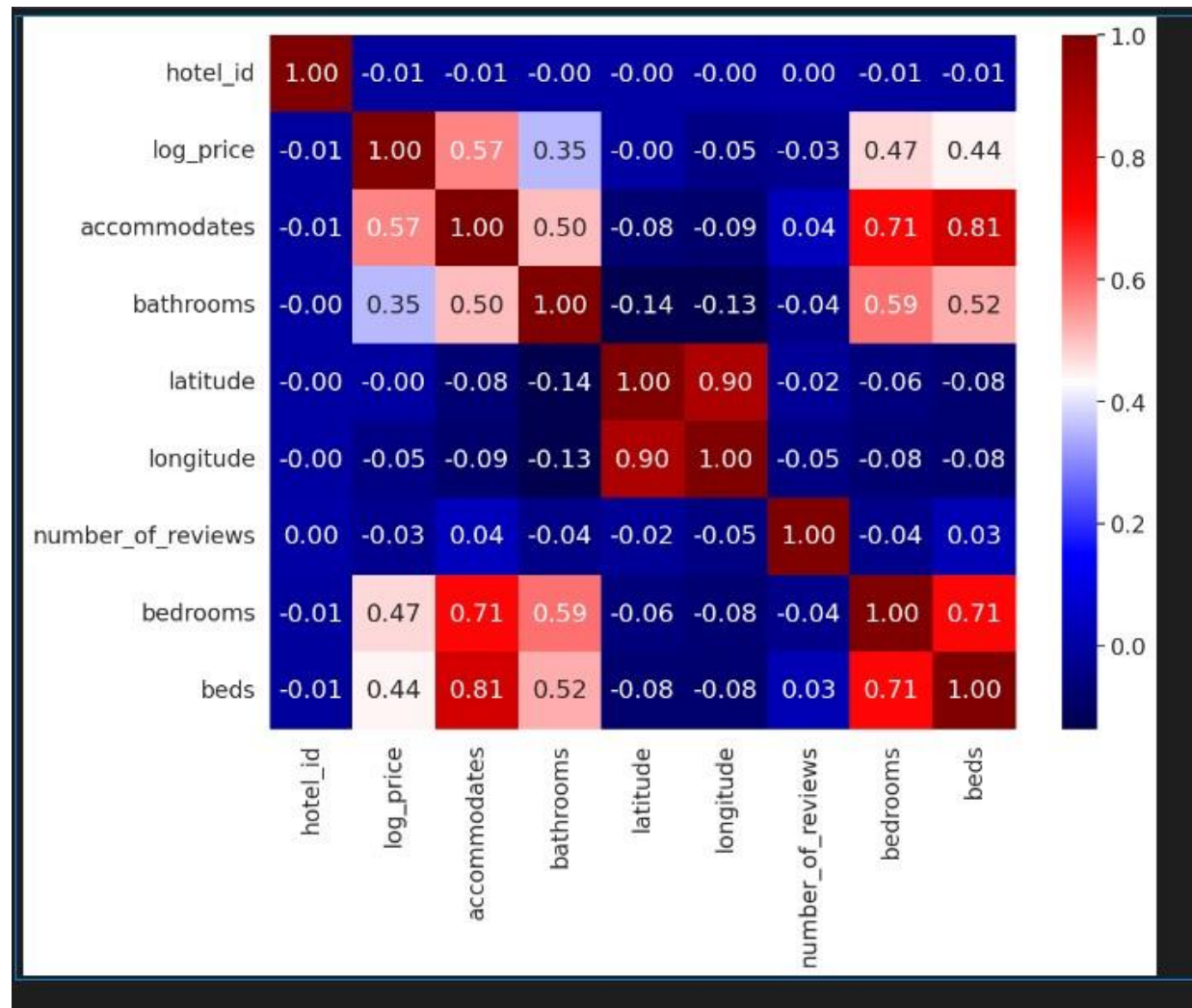
Outliers and Anomalies	-
Data Preproces **sing Code Screenshots**	

Loading Data	
| Handling Missing Data | ```
hotel_id                   0
log_price                  0
property_type              0
room_type                  0
accommodates               0
bathrooms                200
bed_type                   0
cancellation_policy        0
cleaning_fee               0
city                       0
description                0
host_identity_verified   188
host_since               188
instant_bookable           0
latitude                   0
longitude                  0
name                       0
neighbourhood           6872
number_of_reviews          0
bedrooms                  91
beds                     131
dtype: int64
``` |
| Data Transformation | ```python
from sklearn.preprocessing import StandardScaler
import numpy as np

# Generate random data with matching dimensions
x_train = np.random.rand(55583, 13)
y_train = np.random.rand(55583)
x_test = np.random.rand(18528, 13)
y_test = np.random.rand(18528)

# Scale the data
scaler = StandardScaler()
x_train_scaled = scaler.fit_transform(x_train)
x_test_scaled = scaler.transform(x_test)
``` |
| Feature Engineering | Attached the codes in final submission. |
| Save Processed Data | - |