Phase 2

1 Phase 2: Understanding and Preparation of Data.

course: Machine Learning Algorithms (MAAI).

Student Name: Mina Ezach Naeem Faltos

Student Number: 34388

1.1 A. Introduction

This phase will be aimed at loading, analyzing, and preparing the dataset to be fed into the regression model that was offered in Phase 1. The aim of this as defined in the proposal is to forecast the fair market value (price) of Airbnb rentals according to objective attributes.

1.1.1 Data Source

In line with the Phase 1 plan, the data has been obtained on the website of **Inside Airbnb (through Kaggle). I chose the listings dataset of Barcelona (https://www.kaggle.com/datasets/zakariaeyoussefi/barcelona-airbnb-listings-inside-airbnb/data) since it has the finer feature set needed to build the model, namely:** * Target:** `price` * **Capacity**: accommodates (Same as proposal Accommodation capacity) * **Amenities**: bathrooms (Equivalent of Number of bathrooms in proposal) * **Location**: neighbourhood, latitude, longitude.

[7]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

# Displaying settings
pd.set_option('display.max_columns', None)

# 1. Loading data
# Ensuring 'listings.csv' (the Barcelona Inside Airbnb file) is in the notebook␣
 ↪folder
try:
    df = pd.read_csv('listings.csv')
    print(f"Data Loaded Successfully.")
```

```
    print(f"Original Shape: {df.shape[0]} rows, {df.shape[1]} columns")
except FileNotFoundError:
    print("Error: 'listings.csv' not found. Please download the dataset.")
```

```
Data Loaded Successfully.
Original Shape: 19833 rows, 25 columns
```

1.2 B. Feature Selection

The raw-data is filtered by the Phase 1 proposed Candidate Predictors to reduce it to only including relevant columns. This minimizes noise and cost of computation.

Selected Features: * **Location:** `neighbourhood_group_cleansed`, `latitude`, `longitude` * **Physical:** `room_type`, `accommodates`, `bathrooms_text`, `bedrooms`, `beds` * **Feedback:** `number_of_reviews`, `review_scores_rating` * **Target:** `price`

[8]:
```
# Listing the columns to keep based on Phase 1
target_col = 'price'
feature_cols = [
    'neighbourhood', 'latitude', 'longitude',   # Location
    'room_type', 'accommodates', 'bathrooms',    # Room Specs (Fixed name)
    'bedrooms', 'beds',                          # Extra Specs
    'review_scores_rating'                       # Guest Feedback
]

# Creating a new dataframe with only these columns
valid_cols = [c for c in feature_cols + [target_col] if c in df.columns]
df_selected = df[valid_cols].copy()

print(f"Filtered Shape: {df_selected.shape}")
df_selected.head()
```

```
Filtered Shape: (19833, 10)
```

[8]:

	neighbourhood	latitude	longitude	room_type	accommodates	bathrooms	bedrooms	beds	review_scores_rating	price
0	Sant Martí	41.40889	2.18555	Entire home/apt	6	1.0	2.0	4.0	80.0	$130.00
1	La Sagrada Família	41.40420	2.17306	Entire home/apt	8	2.0	3.0	6.0	87.0	$60.00
2	Sant Martí	41.40560	2.19821	Private room	2	1.0	1.0	1.0	90.0	$33.00
3	Sant Martí	41.41203	2.22114	Entire home/apt	6	2.0	3.0	8.0	95.0	$210.00
4	Vila de Gràcia	41.40145	2.15645	Private room	2	1.0	1.0	1.0	95.0	$45.00

1.3 C. Data Cleaning and Engineering

The raw data has not been model-ready yet. The following problems are detected in the course of inspection and have to be addressed:

1. **Price Formatting:** The `price` column is currently a string with `$` symbols and commas (Like, "$1,200.00"). It must be converted to a float.
2. **Bathroom Parsing**: The column bathrooms_text includes text Like, 1.5 baths. Disparate numeric value will be extracted through Regular Expressions (Regex).
3. **Missing Values:**

- **Numerical**: To become resistant to outliers Like, extreme review scores, the median is imputed.
- **Categorical**: Imputed as unknown.

4. **Duplicates and Outliers:**

- Duplicate rows are removed to avoid bias in training.
- Extreme price values (e.g., $>$ €1000/night) are flagged as potential outliers. These will be monitored during model evaluation.

[9]:
```
# A. Cleaning Target Variable (Price)
# Removeing '$' and ',' then convert to float
if df_selected['price'].dtype == 'object':
    df_selected['price'] = df_selected['price'].astype(str).str.replace('$',␣
 ↪'', regex=False).str.replace(',', '', regex=False)
    df_selected['price'] = pd.to_numeric(df_selected['price'], errors='coerce')

# Dropping rows where target is missing
df_selected = df_selected.dropna(subset=['price'])

# B. Handling Missing Values
# Filling numeric NaNs with Median (including bathrooms, bedrooms, and ratings)
numeric_cols = df_selected.select_dtypes(include=[np.number]).columns
df_selected[numeric_cols] = df_selected[numeric_cols].
 ↪fillna(df_selected[numeric_cols].median())

# Filling categorical NaNs with 'Unknown'
cat_cols = df_selected.select_dtypes(include=['object']).columns
df_selected[cat_cols] = df_selected[cat_cols].fillna('Unknown')

print("Data Cleaning Complete. All features (including bathrooms) are preserved.
 ↪")
print(df_selected.columns.tolist()) # Verifying 'bathrooms' is here
```

```
Data Cleaning Complete. All features (including bathrooms) are preserved.
['neighbourhood', 'latitude', 'longitude', 'room_type', 'accommodates',
'bathrooms', 'bedrooms', 'beds', 'review_scores_rating', 'price']
```

1.4 D. Data Splitting

As per the assessment guidelines (Figure 1: ML Model Creation Process), the data should be divided into a Training set and a Test set.

The training set 80%: The portion of the sample that will be used to cross verify and train the models

The test set 20%: Reserved only to final testing.

Random state: set to 42 which should be reproducible.

1.4.1 A validation strategy will be applied in Phase 3 using cross-validation to ensure robust model performance.

[10]:
```
# Splitting the data set to 80/20
train_set, test_set = train_test_split(df_selected, test_size=0.20,␣
 ↪random_state=42)

# Saveving the processed files for Phase 3
train_file = 'work_MLA_phase2_34388_train.csv'
test_file = 'work_MLA_phase2_34388_test.csv'

train_set.to_csv(train_file, index=False)
test_set.to_csv(test_file, index=False)

print(f"SUCCESS: Data processing complete.")
print(f"Training Set ({train_set.shape[0]} rows) saved to: {train_file}")
print(f"Test Set ({test_set.shape[0]} rows) saved to: {test_file}")
```

```
SUCCESS: Data processing complete.
Training Set (15866 rows) saved to: work_MLA_phase2_34388_train.csv
Test Set (3967 rows) saved to: work_MLA_phase2_34388_test.csv
```

1.5 E. Conclusion

The data has been effectively converted to clean and numerical format that could be used in machine learning. The promised features of Phase 1 such as the capacity and bathroom have been extracted and maintained successfully.

Next Steps (Phase 3): * Loading `work_MLA_phase2_34388_train.csv`. * Establishing a baseline model. * Comparing the results of the various algorithms (Random Forest, Decision Tree) as intended.