Digital Lighthouse Assessment Report

Jingyuan Dong

jingyuan_dong@qq.com

Introduction & Background

This project is based on a dataset of hotel reviews which contains customer feedback and ratings for hundreds of hotels. These data originate from public review platforms and encompass various types of accommodations, ranging from budget to luxury hotels. By analysing this data, we aim to gain insights into customers' satisfaction levels and the overall perception of these hotels.

I. Project Scope

The main objective of this project is to utilise data visualisation and text analysis techniques to conduct an in-depth analysis of the hotel customer feedback. Through these analyses, the goal is to identify which hotels receive the most positive reviews, which ones exhibit significant variability in their review scores, and the most frequent words used in the positive comments. These analyses will help hotel managers understand specific customer needs, thereby improving service quality and customer satisfaction.

II. Analytical Goals

The project is divided into two parts, each with specific objectives:

Part 1: Data Visualization and Text Analysis

Data Visualization:

- Identify and visualise the top 10 hotels by number of reviews received.
- Analyse and visualise the 10 hotels with the highest variability in their review scores.
- Explore the relationship between the number of words in positive reviews and the scores.

Text Analysis:

- Extract and analyse the most frequently used words in positive reviews.

Part 2: Advanced Data Exploration

- Develop a classification model to predict whether a review score is greater than nine based on the data provided.
- Clearly state any assumptions made in the model and discuss feature engineering strategies to enhance model performance.

Data Loading and Preprocessing

The data for this analysis is sourced from a comprehensive dataset of hotel reviews. This dataset includes several key pieces of information for each review, such as the hotel name, the reviewer's nationality, number of reviews, positive and negative textual reviews, and the reviewer's score ranging from 0 to 10. The dataset is structured in a CSV format.

I. Loading Process

The dataset is loaded into a Python environment using Pandas, a powerful data manipulation library that provides straightforward methods for handling large datasets. Below is the Python code used to load the dataset:

```python
import pandas as pd

# Define the path to the dataset
file_path = 'hotels.csv'

# Load the dataset into a Pandas DataFrame
hotels_data = pd.read_csv(file_path)
```

Fig 1. Python code of loading the data.

II. Preprocessing Steps

Before conducting any analysis, it is crucial to preprocess the data to ensure its quality and usefulness. To avoid biassed or incorrect analysis results, it is important to check the integrity of the data:

```
# Check for missing values in each column
missing_values = hotels_data.isnull().sum()

print(missing_values)
```

Fig 2. Python code of checking missing values in each column.

The output is:

```
hotel_address                                  0
review_date                                    0
hotel_name                                     0
negative_review                                0
positive_review                                0
reviewer_score                                 0
tags                                           0
days_since_review                              0
reviewer_nationality                           0
total_number_of_reviews_reviewer_has_given     0
lat                                         3268
lng                                         3268
dtype: int64
```

Fig 3. The output of checking missing values.

Although null values are present in the longitude and latitude columns, all other key data columns are free from null values, which essentially confirms the completeness of the dataset.

Data Analysis

I. Part One

A. Top 10 Hotels by Review Count

To identify the hotels with the highest number of reviews, the dataset was grouped by hotel name, and the count of reviews for each hotel was calculated. The top ten hotels were then selected based on this count. Initially, I considered using '*matplotlib.pyplot*' to create a vertical bar chart for the visualisation. However, due to the lengthy nature of most hotel names, displaying the names

horizontally beneath each bar proved challenging, necessitating an angled or rotated text for clarity. To achieve a more harmonious visual representation, I opted for a horizontal bar chart, which accommodates longer hotel names more comfortably. Furthermore, I chose to utilise '*plotly*' instead of '*matplotlib.pyplot*' for plotting, as Plotly's charts are more aesthetically appealing and offer a sense of design. The following Python code snippet demonstrates how this analysis and visualisation were executed:

```
import plotly.express as px

# Calculate the top 10 hotels by review count
top_hotels = hotels_data['hotel_name'].value_counts().head(10).reset_index()
top_hotels.columns = ['Hotel Name', 'Reviews Num']

# Visualization using Plotly
fig = px.bar(top_hotels, y='Hotel Name', x='Reviews Num', orientation='h',
             title='Top 10 hotels with the most reviews', text='Reviews Num')

# Update layout for better spacing and ordering
fig.update_layout(
    xaxis_title="Reviews Num",
    yaxis_title="Hotel Name",
    yaxis=dict(autorange="reversed")
)
fig.show()
```

Fig 4. Python code of visualising top 10 hotels by review count

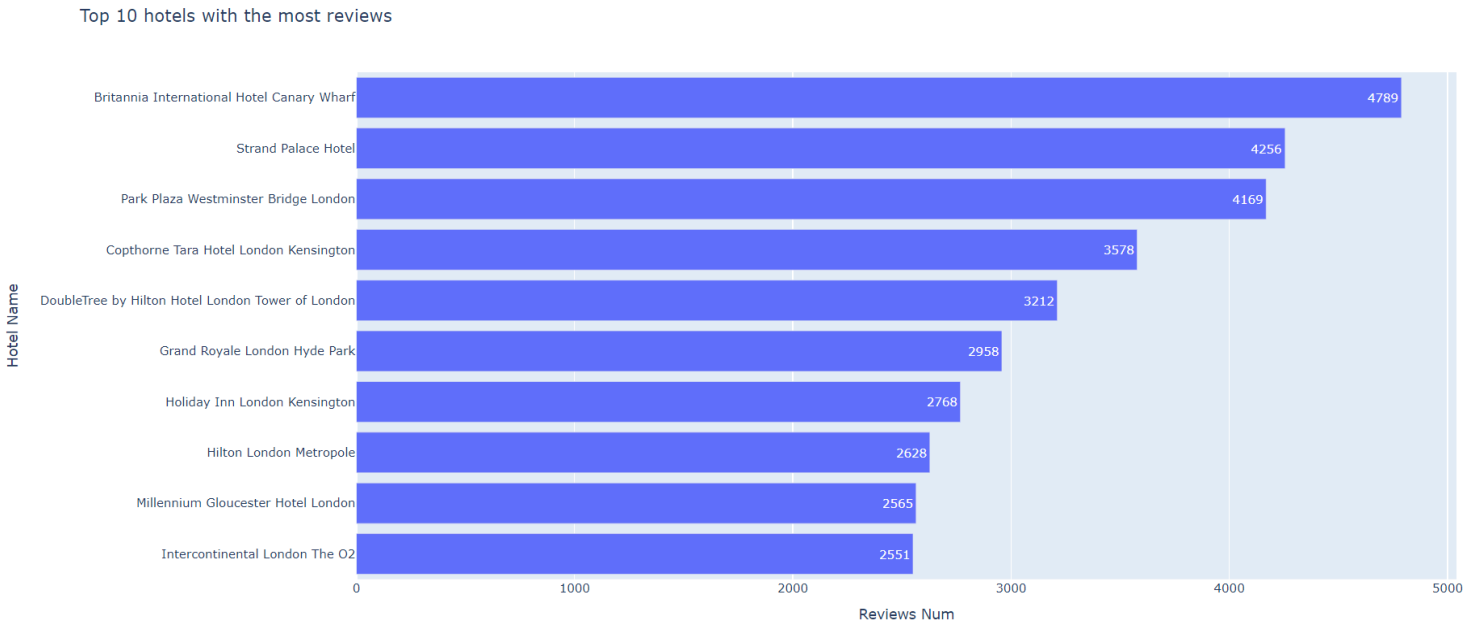

Fig 5. The plot of top 10 hotels with the most reviews

B. The 10 Hotels with Least Reliable Scores

To analyse the reliability of review scores, we calculate the Interquartile Range (IQR) for each hotel's scores. A higher IQR indicates a broader distribution of data, signifying greater variability in the hotel's ratings and, consequently, less score reliability. Therefore, our aim is to identify the ten hotels with the highest IQRs, as these are considered to have the least reliable scores. Similar to the previous task, the visualisation requires a horizontal bar chart due to the length of the hotel names. For this task, I continue to utilise *'plotly'* for its aesthetic and interactive chart capabilities. The following Python code snippet demonstrates how to carry out this analysis and visualisation:

```
import plotly.express as px
import pandas as pd

# Calculate IQR for each hotel
iqr_data = hotels_data.groupby('hotel_name')['reviewer_score'].agg(lambda x: x.quantile(0.75) - x.quantile(0.25))
# Get the top 10 hotels with the highest IQR
top_hotels_least_reliable = iqr_data.nlargest(10).reset_index()
top_hotels_least_reliable.columns = ['Hotel Name', 'IQR']

# Create a horizontal bar chart using Plotly
fig = px.bar(top_hotels_least_reliable, y='Hotel Name', x='IQR', orientation='h',
             title='Top 10 Hotels with Least Reliable Review Scores',
             text='IQR')
# Because the value of each bar is float format, we need to limit it with only 2 digit
fig.update_traces(texttemplate='%{text:.2s}')
fig.update_layout(
    xaxis_title="Interquartile Range (Score)",
    yaxis_title="Hotel Name",
    yaxis=dict(autorange="reversed")
)
fig.show()
```

Fig 6. Python code of visualising top 10 hotels by review count

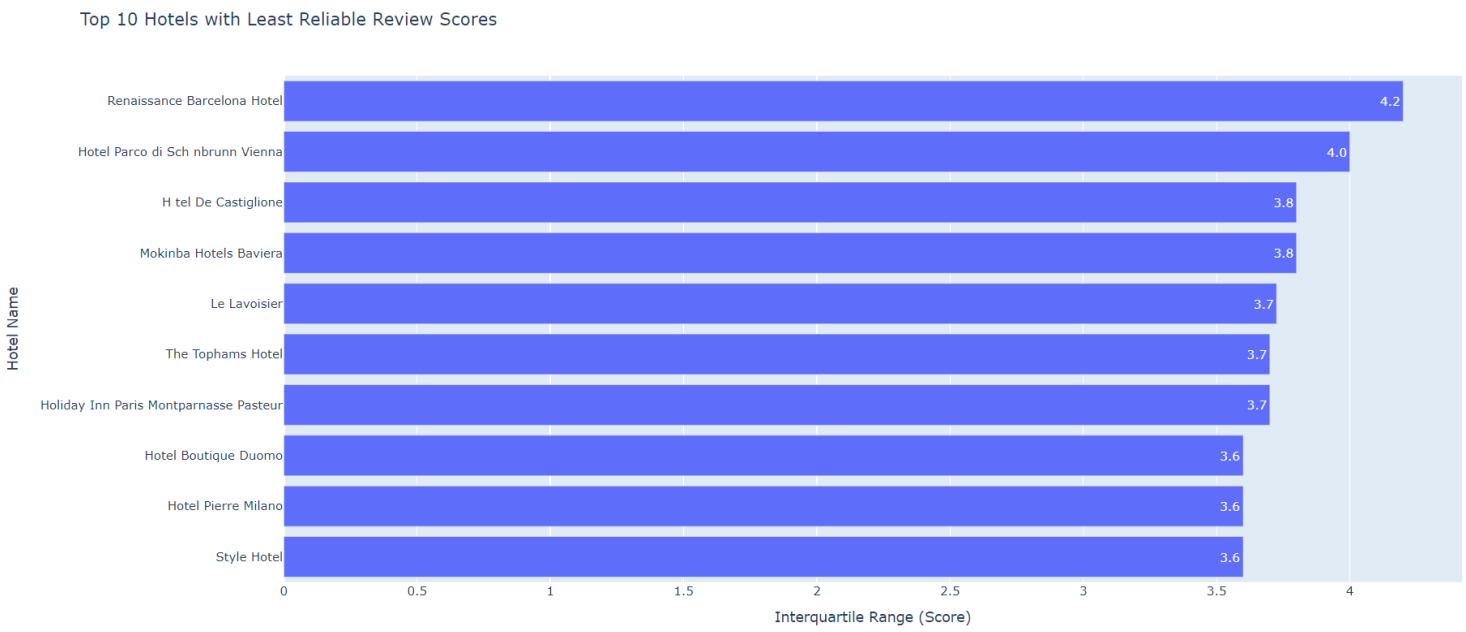

Fig 7. The plot of top 10 hotels with least reliable review scores

C. Create a new column as the word count of positive review

The creation of the new column can be easily accomplished with the following Python code:

```
# create a new column 'positive_review_wc'
hotels_data['positive_review_wc'] = hotels_data['positive_review'].apply(lambda x: len(x.split()))
```

Fig 8. Create a new column shows the word count of the positive review column

D. Scatter Plot Comparing Positive Review Word Count and Score

We can still use 'plotly' to plot the relationship between positive review word count and reviewer score with the following Python code:

```
import plotly.express as px

# Create a scatter plot to visualize the relationship between word count and reviewer scores
fig = px.scatter(hotels_data, x='positive_review_wc', y='reviewer_score',
                 title='Relationship between Positive Review Word Count and Reviewer Scores',
                 labels={'positive_review_wc': 'Word Count in Positive Reviews', 'reviewer_score': 'Reviewer Score'})

# Show the plot
fig.show()
```

Fig 9. Python code of visualising the relationship between positive review word count and reviewer score

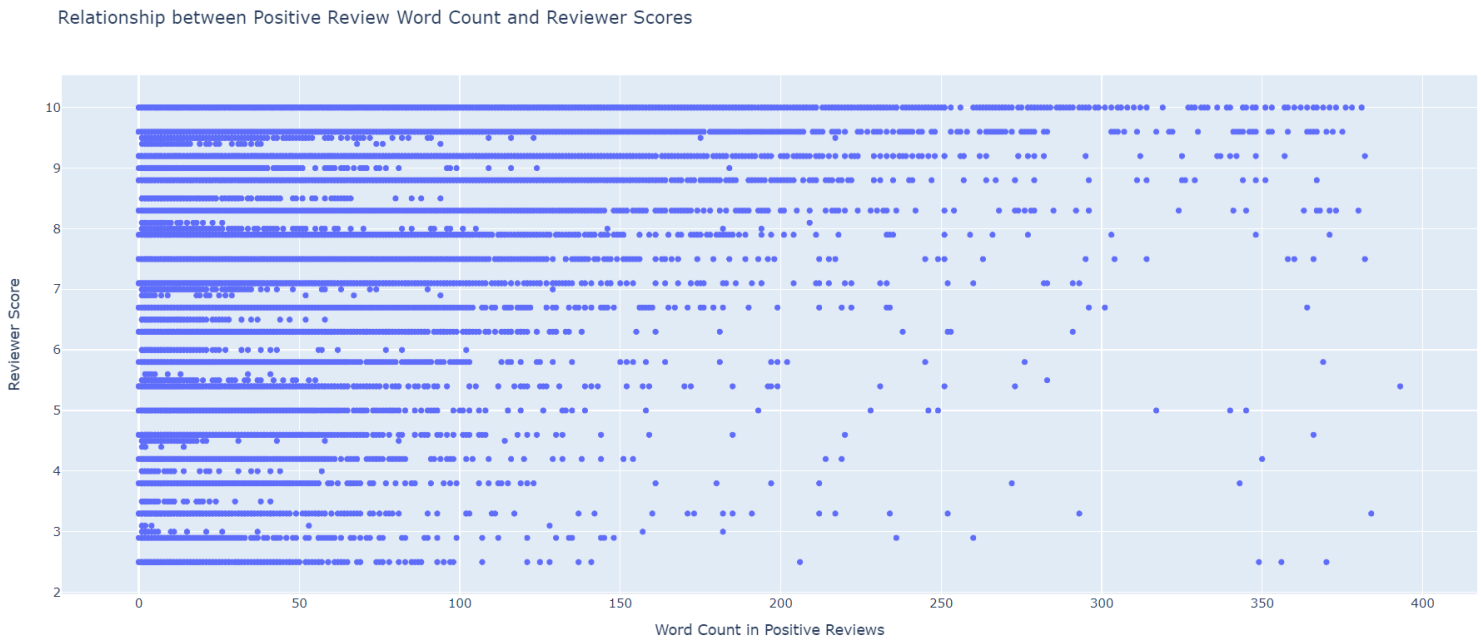

Fig 10. The scatter plot of the relationship between positive review word count and reviewer score

The scatter plot produced from the dataset presents the correlation between the length of positive reviews and the corresponding scores given by reviewers. As visualised, each point represents an individual review with its word count plotted against its score.

Comments:

Firstly, it is noticeable that longer positive reviews tend to correlate with higher scores, as we can observe that reviews exceeding 300 words are rarely found below 7 score. Conversely, as scores rise above 8, the frequency of reviews with more than 300 words noticeably increases. On the other hand, shorter reviews do not necessarily correspond to lower ratings; for instance, reviews with fewer than 50 words appear more densely distributed in the range above 7 score than in the range below 5 score. Additionally, aside from exact scores of ten and five, there is a marked scarcity of reviews on the whole number horizontal lines, suggesting that reviewers tend to avoid round number ratings.

Explanations:

We can make some assumptions to explain the observations mentioned above. First, when guests have an extremely positive experience, they are likely to provide more detailed feedback, resulting in longer positive reviews and higher ratings. Conversely, when guests have a negative experience, their dissatisfaction may lead them to write fewer words in positive reviews. Additionally, since ratings are not uniformly distributed, it may simply be that there are more reviews above a score of 5 than below, which would explain why reviews with fewer words do not appear more densely concentrated in the lower scoring regions. In most cases, a score of 10 represents perfection, while a score of 5 indicates complete neutrality, leading people to round their scores to these whole numbers to express total satisfaction or neutrality. On the other hand, for instance, scores like exactly 4 are less common than scores of 4-point-something. This analysis indicates that while longer reviews may slightly tend towards higher scores, word count is not a reliable predictor of reviewer ratings. A more detailed approach is necessary to consider other variables and factors that might influence customer review scores.

E. Word Frequency in Positive Hotel Reviews

Due to the need to remove stopwords, I utilised the Natural Language Toolkit (NLTK) library, which offers tools for filtering out stopwords. I employed regular expressions to remove non-alphabetic characters and converted the text to lowercase. Then, I used the Counter class to count the word frequencies. And still use '*plotly*' for visualisation. Below is the Python code:

```python
import plotly.express as px
from collections import Counter
from nltk.corpus import stopwords
import re

# Load stopwords
stop_words_set = set(stopwords.words('english'))

# Simply clean text and extract vocabulary
def process_text_simple(texts):
    all_words = []
    for text in texts:
        # Use regular expression to remove non-alphabetic characters
        # and convert text to lowercase
        words = re.findall(r'\b[a-z]+\b', text.lower())
        # filter stopwords
        filtered_words = [word for word in words if word not in stop_words_set]
        all_words.extend(filtered_words)
    return all_words

# get all text
all_reviews = hotels_data['positive_review'].dropna().tolist()

# Perform text processing and word frequency statistics
all_filtered_words = process_text_simple(all_reviews)
word_counts = Counter(all_filtered_words)

# Get the 10 most frequently used words
most_common_words_simple = word_counts.most_common(10)

# Visualisation
df_words = pd.DataFrame(most_common_words_simple, columns=['Word', 'Frequency'])

fig = px.bar(df_words, x='Word', y='Frequency',
             title='Top 10 Most Common Words in Positive Reviews',
             labels={'Word':'Words', 'Frequency':'Frequency of Words'},
             text='Frequency')
fig.update_layout(xaxis_title="Words",yaxis_title="Frequency")
fig.show()
```

Fig 11. Python code of visualising the 10 most frequently used words from positive reviews

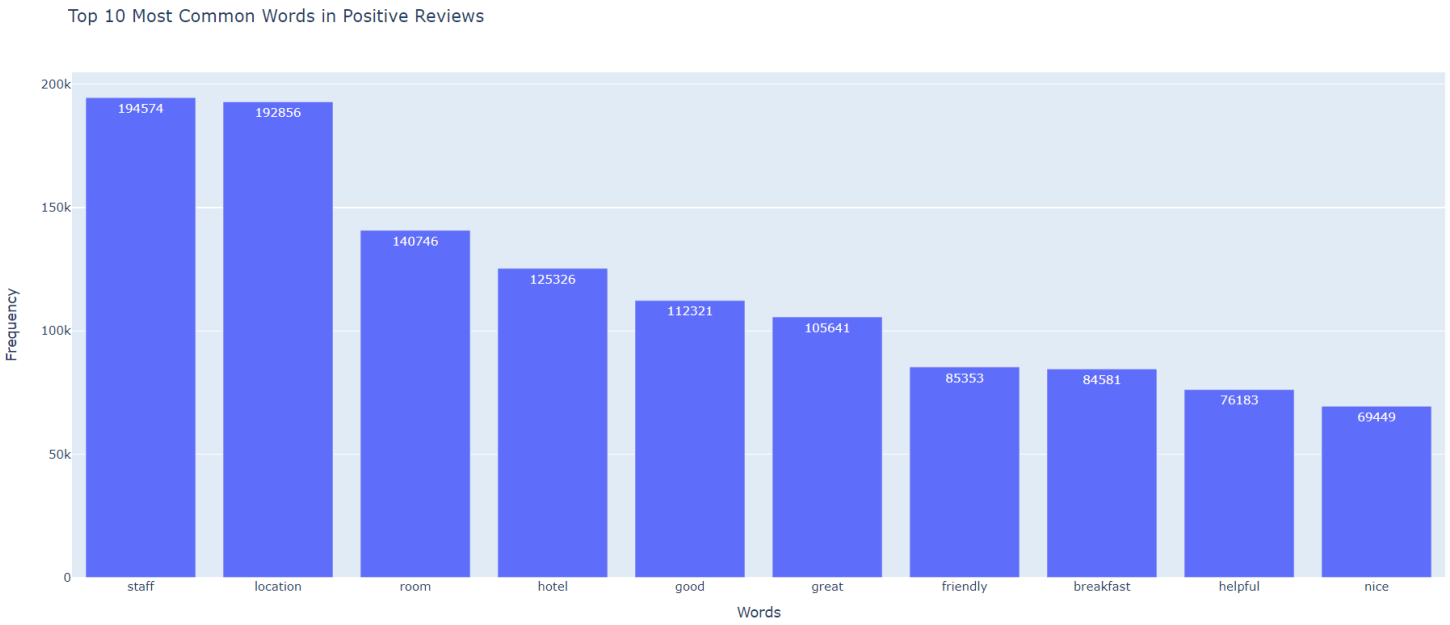

Fig 12. The plot of 10 most frequently used words from positive reviews

The ten most frequent words in the positive reviews were: "staff" highlighted in 194,574 reviews, "location" in 192,856, "room" in 140,746, "hotel" in 125,326, "good" in 112,321, "great" in 105,641, "friendly" in 85,353, "breakfast" in 84,581, "helpful" in 76,183, and "nice" in 69,449. These terms suggest that guests often value the friendliness and helpfulness of staff, the quality of breakfast, and the hotel's location.

The frequent mention of words like "staff," "friendly," and "helpful" indicates that interpersonal interactions have a significant impact on guest satisfaction. Additionally, the emphasis on "location" supports the well-known hospitality principle that location is crucial. This analysis not only highlights what guests consider important but also guides hotels on where to focus their service improvement and marketing efforts.

II. Part Two

A. Comparison of Review Scores by Trip Type in 'tags' Column

For guests categorised by different travel types, in this case leisure trip and business trip, scoring tendencies may diverge. Therefore, I conducted a comparative analysis of the percentage distribution across various score ranges for these two categories of travellers. The approach entailed filtering reviews by each category, calculating the total number of reviews per category, and then determining the percentage of reviews falling within each score range relative

to the total. To visualise the data, I utilised 'plotly' to create a grouped bar chart, which illustrates the proportion of scores within specific intervals for leisure versus business travel reviews. Python code and output plot are illustrated below:

```python
import plotly.express as px
from collections import OrderedDict

# Function to calculate score ranges and percentages for a given trip type
def calculate_scores(trip_type):
    # Filter reviews based on trip type
    filtered_data = hotels_data[hotels_data['tags'].str.contains(trip_type)]

    # Define score ranges
    score_ranges = [(9, 10), (8, 9), (7, 8), (6, 7), (5, 6), (4, 5), (3, 4), (2, 3)]

    # Total number of reviews for this type
    total_reviews = len(filtered_data)

    # Calculate total count for all score ranges
    total_count = sum(filtered_data['reviewer_score'].apply(lambda x: any(low <= x < high for low, high in score_ranges)))

    # OrderedDict to hold score percentages
    score_percentages = OrderedDict()

    for low, high in score_ranges:
        # Count reviews within each score range
        count_in_range = filtered_data[(filtered_data['reviewer_score'] >= low) & (filtered_data['reviewer_score'] < high)].shape[0]
        # Calculate percentage
        percentage = (count_in_range / total_count) * 100 if total_count > 0 else 0
        score_percentages[f'{low}-{high}'] = percentage

    return score_percentages

# Calculate for Leisure trip and Business trip
leisure_scores = calculate_scores('Leisure trip')
business_scores = calculate_scores('Business trip')

# Prepare DataFrame for visualization
df_scores = pd.DataFrame({
    'Score Range': list(leisure_scores.keys()),
    'Leisure Trip (%)': list(leisure_scores.values()),
    'Business Trip (%)': list(business_scores.values())
})

# create group bar plot
fig = px.bar(df_scores, x='Score Range', y=['Leisure Trip (%)', 'Business Trip (%)'],
             title='Comparison of Review Scores by Trip Type',
             labels={'value': 'Percentage (%)', 'variable': 'Trip Type'},
             barmode='group')

fig.update_traces(texttemplate='%{y:.2f}%')

# update layout
fig.update_layout(
    xaxis_title='Score Ranges',
    yaxis_title='Percentage of Reviews (%)',
    xaxis={'categoryorder':'array', 'categoryarray': list(leisure_scores.keys())},
    legend_title='Trip Type'
)

fig.show()
```

Fig 13. Python code of visualising the comparison of review scores by trip type

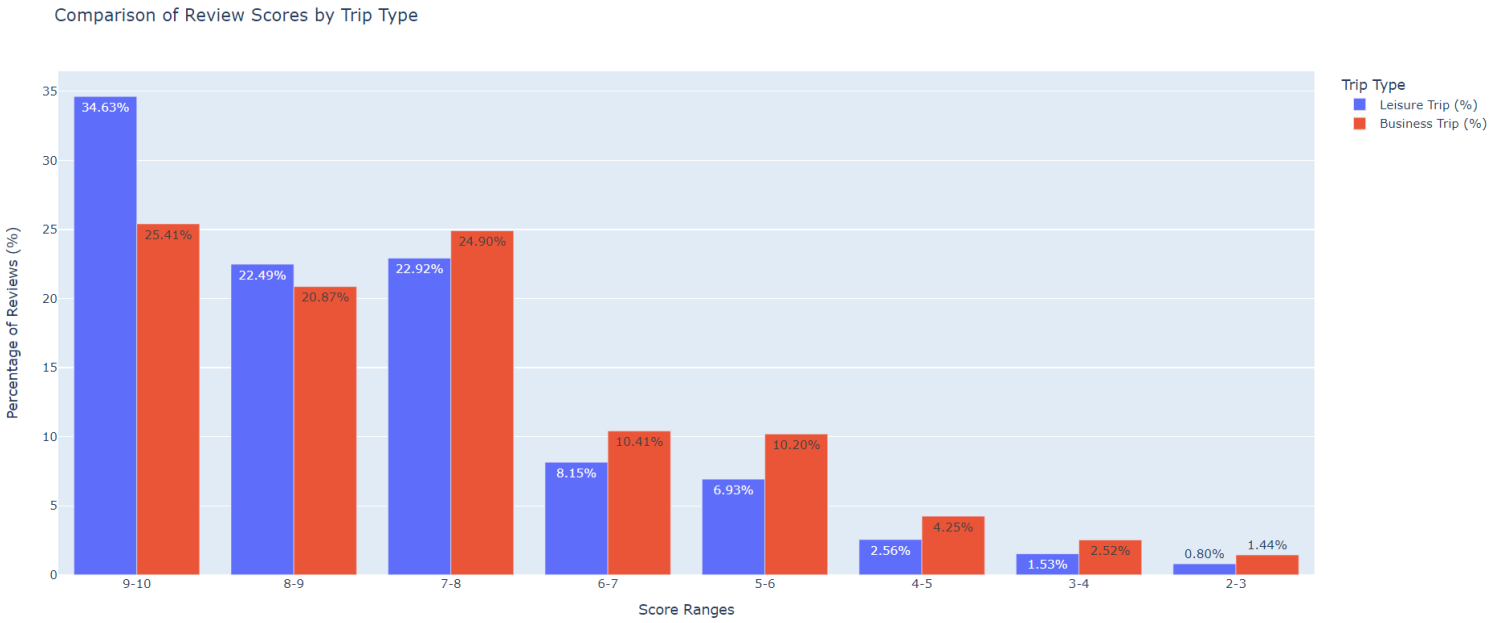

Fig 14. The group bar plot of the comparison of review scores by trip type

Comments:

The grouped bar chart allows us to observe that for both leisure and business travel, the highest proportion of reviews falls within the top score range (9 to 10), with leisure travel reviews representing a notably larger share than business travel within this bracket. Moreover, this high-score bracket displays the greatest disparity in review proportions between the two types of travel. Beginning with the slightly lower high-score range (7 to 8), there is a greater percentage of reviews for business travel as compared to leisure. Reviews in the 7 to 8 score range significantly outnumber those in lower brackets, and the proportion of reviews with scores above 7 is almost double that of scores below 7.

Explanations:

To interpret these observations, we can posit some logical assumptions. Firstly, for both leisure and business trips, individuals seem disinclined to leave low scores, generally preferring to award mid to high scores (above 7). Additionally, given a satisfactory experience, there appears to be a tendency to provide scores close to perfect (9 to 10). Leisure travellers, possibly due to being in a state of relaxation and enjoyment, often have a better overall experience and thus tend to give higher ratings. They might have less specific and stringent expectations of their accommodations than business travellers, focusing more

on the overall experience. Particularly when leisure travellers are highly satisfied, they are more likely to award near-perfect scores (9 to 10) compared to business travellers. On the other hand, business travellers may have higher standards for their lodgings and, compared to leisure travellers, are more prone to issue moderate scores (4 to 6), indicating that it may be more challenging for accommodations to meet their expectations.

B. Analysis of Reviewer Levels and Positive Review Word Count

In this analysis, we explore the relationship between the experience level of reviewers and the length of their positive reviews. Reviewers were categorised into four levels based on the quartiles of the total number of reviews they have provided: Novice, Intermediate, Advanced, and Expert. Similarly, the word count of positive reviews was also divided into four categories: Short, Medium, Long, and Very Long, based on quartiles. The dataset was segmented based on the total number of reviews each reviewer has given, assigning them into quartiles. These quartiles help in distinguishing between reviewers based on their experience. Positive reviews were analysed for word count, which was also categorised into quartiles to observe trends across varying lengths of feedback. A heatmap was then generated to visualise the density and distribution of word counts across different reviewer levels. Python code and output plot are illustrated below:

```python
import plotly.express as px

# Calculate the quartiles for the number of reviews given by each reviewer
reviewer_quartiles = hotels_data['total_number_of_reviews_reviewer_has_given'].quantile([0.25, 0.5, 0.75])

#  Assign reviewer levels based on the calculated quartiles
def assign_reviewer_level(reviews_count):
    if reviews_count <= reviewer_quartiles[0.25]:
        return 'Novice'
    elif reviews_count <= reviewer_quartiles[0.5]:
        return 'Intermediate'
    elif reviews_count <= reviewer_quartiles[0.75]:
        return 'Advanced'
    else:
        return 'Expert'

# Assign reviewer levels to each reviewer in the dataset
hotels_data['reviewer_level'] = hotels_data['total_number_of_reviews_reviewer_has_given'].apply(assign_reviewer_level)

# Calculate the word count for each positive review
hotels_data['positive_review_word_count'] = hotels_data['positive_review'].apply(lambda x: len(str(x).split()))

# Calculate the quartiles for the word count of positive reviews
word_count_quartiles = hotels_data['positive_review_word_count'].quantile([0.25, 0.5, 0.75])

# Assign word count levels based on the calculated quartiles
def assign_word_count_level(word_count):
    if word_count <= word_count_quartiles[0.25]:
        return 'Short'
    elif word_count <= word_count_quartiles[0.5]:
        return 'Medium'
    elif word_count <= word_count_quartiles[0.75]:
        return 'Long'
    else:
        return 'Very Long'

# Assign word count levels to each positive review in the dataset
hotels_data['word_count_level'] = hotels_data['positive_review_word_count'].apply(assign_word_count_level)

# Count the occurrences of each word count level within each reviewer level
grouped_counts = hotels_data.groupby(['reviewer_level', 'word_count_level']).size().reset_index(name='count')

# Calculate the total number of reviews for each reviewer level
total_counts_by_level = hotels_data.groupby('reviewer_level').size().reset_index(name='total_count')

# Merge the counts with the total counts to calculate the percentage of each word count level within each reviewer level
grouped_counts = grouped_counts.merge(total_counts_by_level, on='reviewer_level')
grouped_counts['percentage'] = (grouped_counts['count'] / grouped_counts['total_count']) * 100

# Pivot the data to create a heatmap-ready format
pivot_data = grouped_counts.pivot(index='reviewer_level', columns='word_count_level', values='percentage').fillna(0)

# Create a heatmap for visualisation
fig = px.imshow(
    x=['Short', 'Medium', 'Long', 'Very Long'],
    y=['Novice', 'Intermediate', 'Advanced', 'Expert'],
    title="Heatmap of Reviewer Levels and Word Count Levels by Percentage"
)

fig.update_xaxes(side="bottom")
fig.update_layout(
    xaxis_title='Word Count Level',
    yaxis_title='Reviewer Level',
    coloraxis_colorbar_title='Percentage %'
)

fig.show()
```

Fig 15. Python code of visualising the reviewer levels and positive review word count

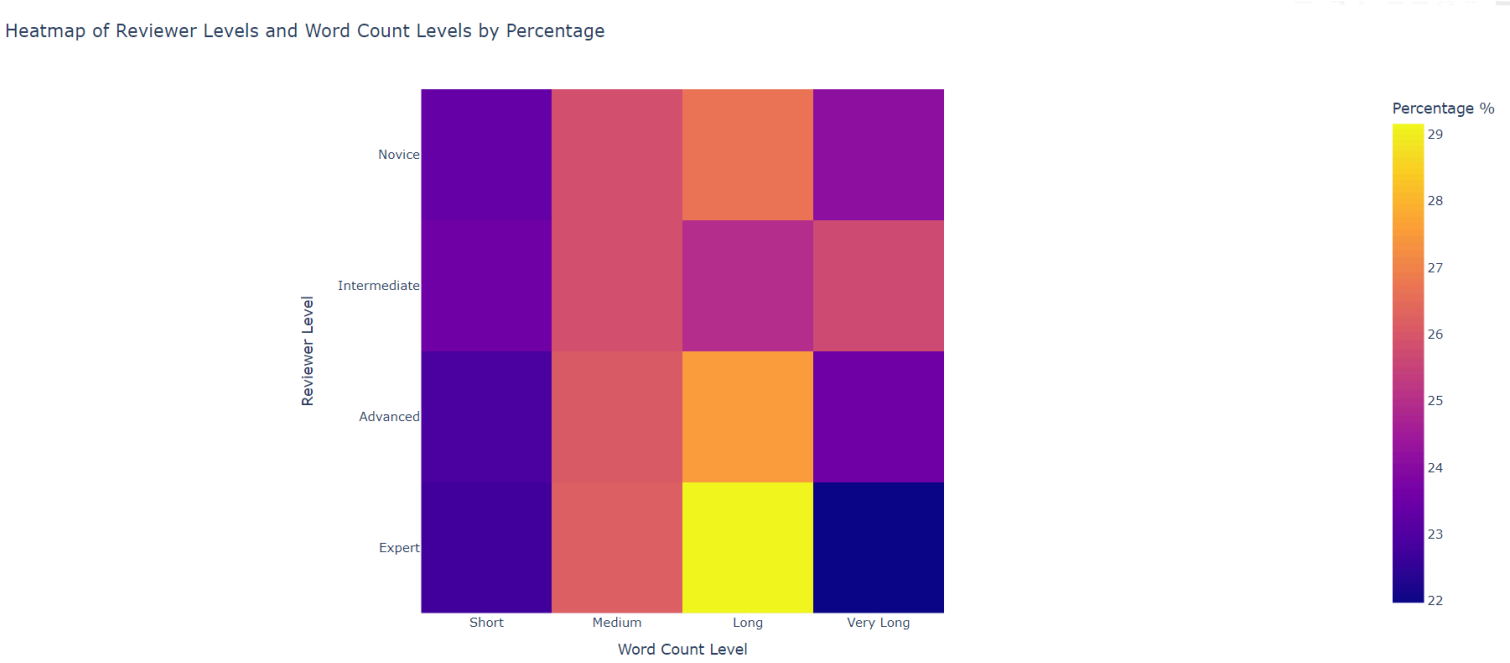

Fig 16. The heatmap plot of the reviewer levels and positive review word count

The heatmap provides a visual representation of how different reviewer levels correlate with the length of their positive reviews. This visualisation helps in understanding whether more experienced reviewers tend to write longer reviews or if the review length is consistent across all levels. The heatmap vividly demonstrates the distribution of word count levels among different reviewer experience categories. Expert reviewers show a pronounced tendency to write longer reviews, as evidenced by the conspicuous yellow segment, signifying that seasoned reviewers tend to offer more detailed and extensive feedback. Novice and Advanced levels exhibit a more similar spread of all word count levels. The review lengths of Intermediate reviewers are relatively evenly distributed.

Comments:

The propensity for Novice reviewers to write shorter reviews may stem from their relative unfamiliarity with platform norms or initial hesitation to compose lengthy narratives. As reviewers accrue more experience, they seem to experiment with a broader range of review lengths, possibly feeling a duty to provide comprehensive feedback. Advanced reviewers, similar to Novices, predominantly write medium and long reviews rather than short or very long ones. The inclination among Expert reviewers to produce lengthy reviews could also arise from a desire to be thorough and beneficial to the community, drawing from their wealth of experience.

Explanations:

The progression from 'Short' to 'Long' reviews with increasing reviewer experience can be attributed to several factors. Novices may lack the confidence or experience needed to craft in-depth reviews. In contrast, Expert reviewers, likely seasoned travellers with a plethora of experiences to share, may find it necessary to include more detail in their reviews. They may also have developed a more discerning eye for details, leading to longer reviews. Furthermore, platforms often encourage detailed reviews through community recognition and rewards, which might motivate more experienced reviewers to write expansively.