TEAM 22 : Similar Places Recommendation System for Sustainable Tourism

Hari Chandan Gooda UBIT : 50614165

Pramila Yadav UBIT : 50613803

Tharunnesh Ramamoorthy UBIT : 50611344

Keshav Narayan Srinivasan UBIT : 50610509

TASK 1: PROBLEM STATEMENT

Tourism has grown tremendously over the past two decades, majorly because of affordable transport fares and rise in middle class population. But being mostly aware of popular places, a large proportion of tourists choose to visit those places, causing overcrowding. As a result, we see reports of huge demand over resources and a surge in pollution in such places. By studying the data, we aim to understand the patterns in tourism and suggest similar yet less popular places that will promote sustainable tourism. In other words, we want to reduce the repercussions caused by overcrowding in certain places and also, let people know about new and exciting places.

TASK 3: DATA RETRIEVAL

We have planned to use the dataset from Google maps, hence we have used few scraping techniques such as seleniuma nd scrapy with the help of certain scraping guides to create our dataset for 4000 tourism places in the United States of America. The dataset contains data pertaining to popularity controls, Working hours and location that can give us some meaningful data that we can work with. It is completely unstructured and we have to clean it accordingly to get the data and modeling we want to do in the future for this project.

Loading our Scraped Dataset which needs to be cleaned

In [1]:
```
import pandas as pd
import numpy as np
import re
import seaborn as sns
import matplotlib.pyplot as plt
```

In [ ]:
```
file_path = "Data.csv"
df = pd.read_csv(file_path)
```

DATA CLEANING

1. Handling missing values in important columns

In [3]:
```
df_cleaned = df.dropna(subset=['latitude', 'popular_times', 'longitude', 'rating
```

2. Converting Cities and States to a unified format

In [4]:
```
df_cleaned['city'] = df_cleaned['city'].str.lower()
df_cleaned['us_state'] = df_cleaned['us_state'].str.lower()
df_cleaned['city'] = df_cleaned['city'].str.capitalize()
df_cleaned['us_state'] = df_cleaned['us_state'].str.capitalize()
```

3. Normalizing the working hours for time consistent format

In [5]:
```
df_cleaned['working_hours'] = df_cleaned['working_hours'].astype(str)
df_cleaned['working_hours'] = df_cleaned['working_hours'].apply(lambda x: re.sub
df_cleaned['working_hours'] = df_cleaned['working_hours'].fillna('Unknown')

def get_weekday_hours(working_hours):
    try:
        parts = working_hours.split(':')
        weekday_parts = parts[:5]
        weekday_hours = sum(1 for part in weekday_parts if part.strip())
        return weekday_hours
    except Exception as e:
        return 0

def get_weekend_hours(working_hours):
    try:
        parts = working_hours.split(':')
        weekend_parts = parts[5:7]
        weekend_hours = sum(1 for part in weekend_parts if part.strip())
        return weekend_hours
    except Exception as e:
        return 0

df_cleaned['weekday_hours'] = df_cleaned['working_hours'].apply(get_weekday_hour
df_cleaned['weekend_hours'] = df_cleaned['working_hours'].apply(get_weekend_hour
```

4. Making the names uniform by applying title case

In [6]:
```
df_cleaned['name'] = df_cleaned['name'].str.title()
```

5. Removing invalid latitude and longitude range and bringing precision to them

In [7]:
```
df_cleaned = df_cleaned[(df_cleaned['latitude'].between(-90, 90)) & (df_cleaned[
df_cleaned['latitude'] = df_cleaned['latitude'].round(6)
df_cleaned['longitude'] = df_cleaned['longitude'].round(6)
```

6. Creating a new column 'rating_category' based on 'rating' (with tiers as low, medium, high)

In [8]:
```
df_cleaned['rating_category'] = pd.cut(df_cleaned['rating'], bins=[0, 3, 4.5, 5]
```

7. Removing the whitespace from following columns

In [9]:
```
df_cleaned['name'] = df_cleaned['name'].str.strip()
df_cleaned['city'] = df_cleaned['city'].str.strip()
```

8. Introducing a new column 'is_weekend_open' based on 'working_hours' to check if locations are open on weekends

In [10]:
```
def has_values_after_colon(working_hours):
    try:
        parts = working_hours.split(':')
        if len(parts) > 6 and parts[6].strip():
            return True
        if len(parts) > 7 and parts[7].strip():
            return True
        return False
    except Exception as e:
        return False

df_cleaned['is_weekend_open'] = df_cleaned['working_hours'].apply(has_values_aft
```

9. Convertings 'rating' column to numeric data if any inconsitency

In [11]:
```
df_cleaned['rating'] = pd.to_numeric(df_cleaned['rating'], errors='coerce')
df_cleaned = df_cleaned.dropna(subset=['rating'])
```

10. Creating columns 'weekday_hours'and 'weekend_hours'

In [12]:
```
df_cleaned['working_hours'] = df_cleaned['working_hours'].apply(lambda x: re.sub
df_cleaned['weekday_hours'] = df_cleaned['working_hours'].apply(get_weekday_hour
df_cleaned['weekend_hours'] = df_cleaned['working_hours'].apply(get_weekend_hour
```

11. Standardize the 'rating' column by scaling it to a range of 0 to 5

In [13]:
```
df_cleaned['rating_scaled'] = (df_cleaned['rating'] / df_cleaned['rating'].max()
df_cleaned.to_csv("C:\\Users\\kesha\\OneDrive\\UB\\Fall 2024\\Data Intensive Com
```

Task 2

Name: Tharunnesh Ramamoorthy Ub person id:50611344

Question1 : Does the availability of weekend hours impact the customer ratings?

How it leads to our objective: So this helps our model to consider the tourism during weekends and helps in modeling a better recommendation system.

Why it is significant: The availability of these popular places attracts local tourism we need to observe if it has a capacity to accommodate tourists outside local regions.

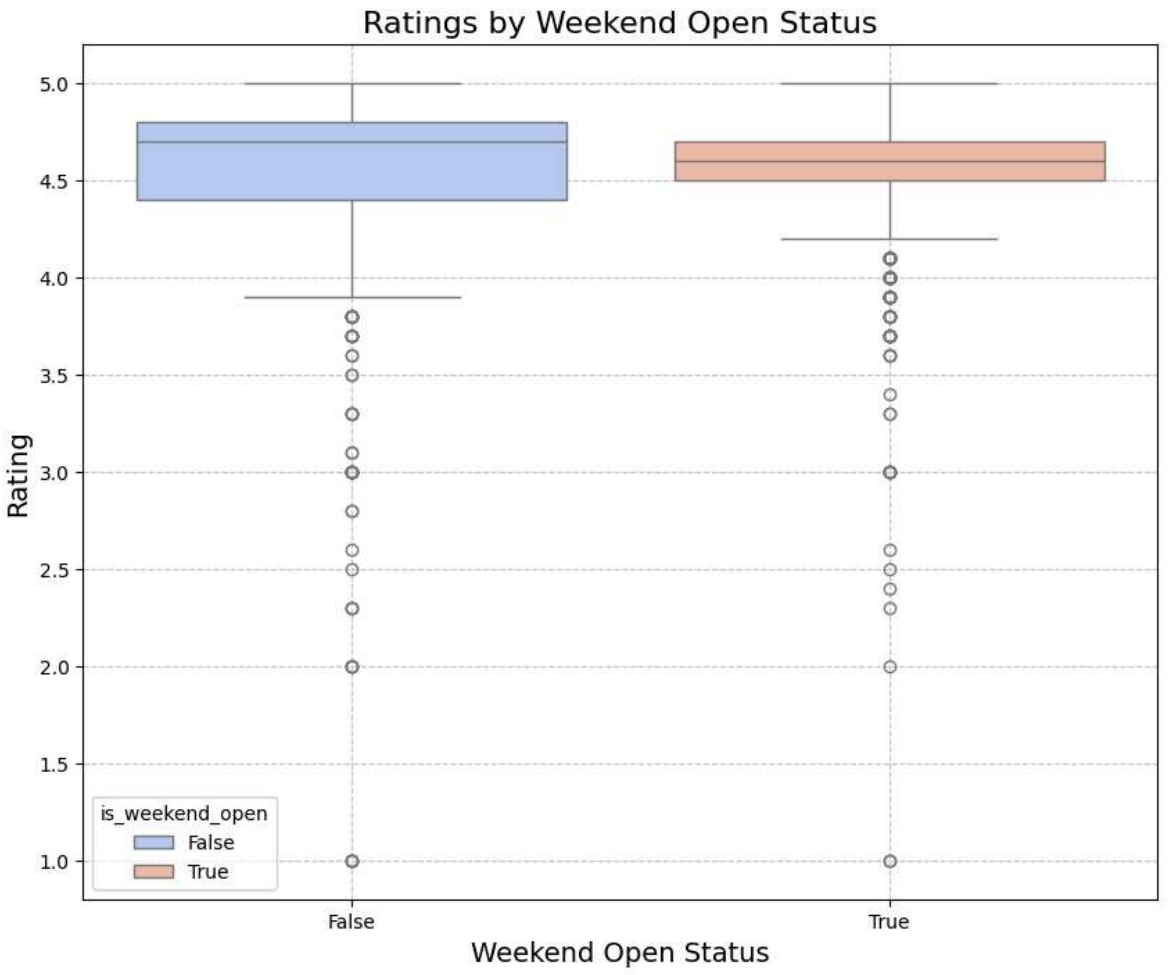
Question 2: Is there a geographical pattern in the popularity of locations based on longitude and latitude?

How it leads to our objective: This is important for developing the objective recommendations in a feasible tourism that showcasing the destination that align with geographical preferences.

Why it is significant: It will identifies whether there are particular geographical region like longitudes and latitudes that are more popular will provide a awareness to the tourists. It helps to determine whether it is environmental-friendly destination in particular areas attracts more tourists.

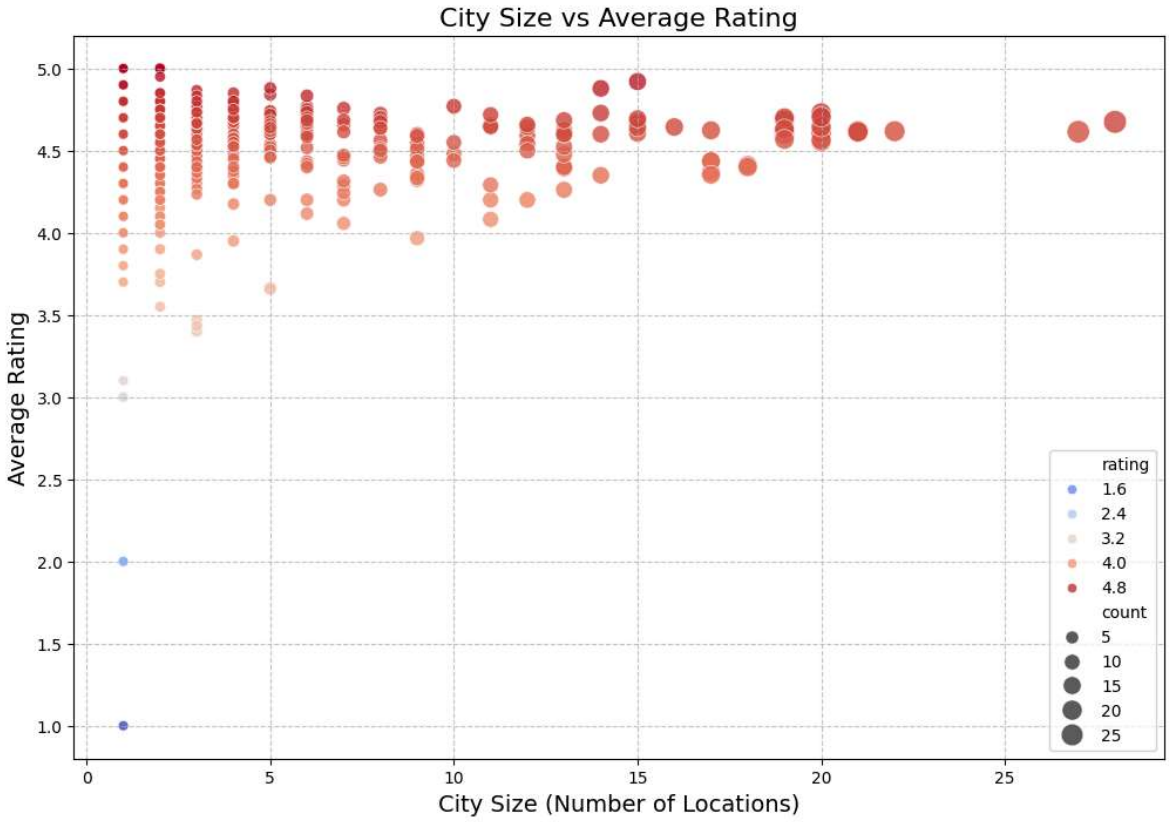
TASK 5: EDA for the Questions

In [14]:
```
# Hypothesis 1: Difference in ratings between weekend open vs not
plt.figure(figsize=(10, 8))
sns.boxplot(x='is_weekend_open', y='rating', data=df_cleaned, hue='is_weekend_op
plt.title('Ratings by Weekend Open Status', fontsize=16)
plt.xlabel('Weekend Open Status', fontsize=14)
plt.ylabel('Rating', fontsize=14)
plt.grid(True, linestyle='--', alpha=0.7)
plt.show()
```



In [15]:
```
# Hypothesis 2: Average ratings in large vs small cities
city_rating_count = df_cleaned.groupby('city')['rating'].mean().reset_index()
city_size = df_cleaned['city'].value_counts().reset_index()
city_size.columns = ['city', 'count']
city_data = pd.merge(city_rating_count, city_size, on='city')

plt.figure(figsize=(12, 8))
sns.scatterplot(x='count', y='rating', data=city_data, hue='rating', palette='cc
plt.title('City Size vs Average Rating', fontsize=16)
plt.xlabel('City Size (Number of Locations)', fontsize=14)
plt.ylabel('Average Rating', fontsize=14)
plt.grid(True, linestyle='--', alpha=0.7)
plt.show()
```



TASK 2

Name: Pramila Yadav UBIT: 50613803

Question 1: How does the geographical location (latitude/longitude) relate to customer ratings?

How it leads to our objective: This question explores whether geographical regions have an impact on customer ratings. It contributes to the objective of promoting sustainable tourism by identifying lesser-known but high-potential regions where tourists may have better experiences due to fewer crowds.

Why it is significant: Overcrowded tourist spots often result in lower satisfaction, while less visited areas may offer better services and experiences. Recognizing these patterns

allows for better promotion of alternative destinations which ensures more balanced tourist traffic.

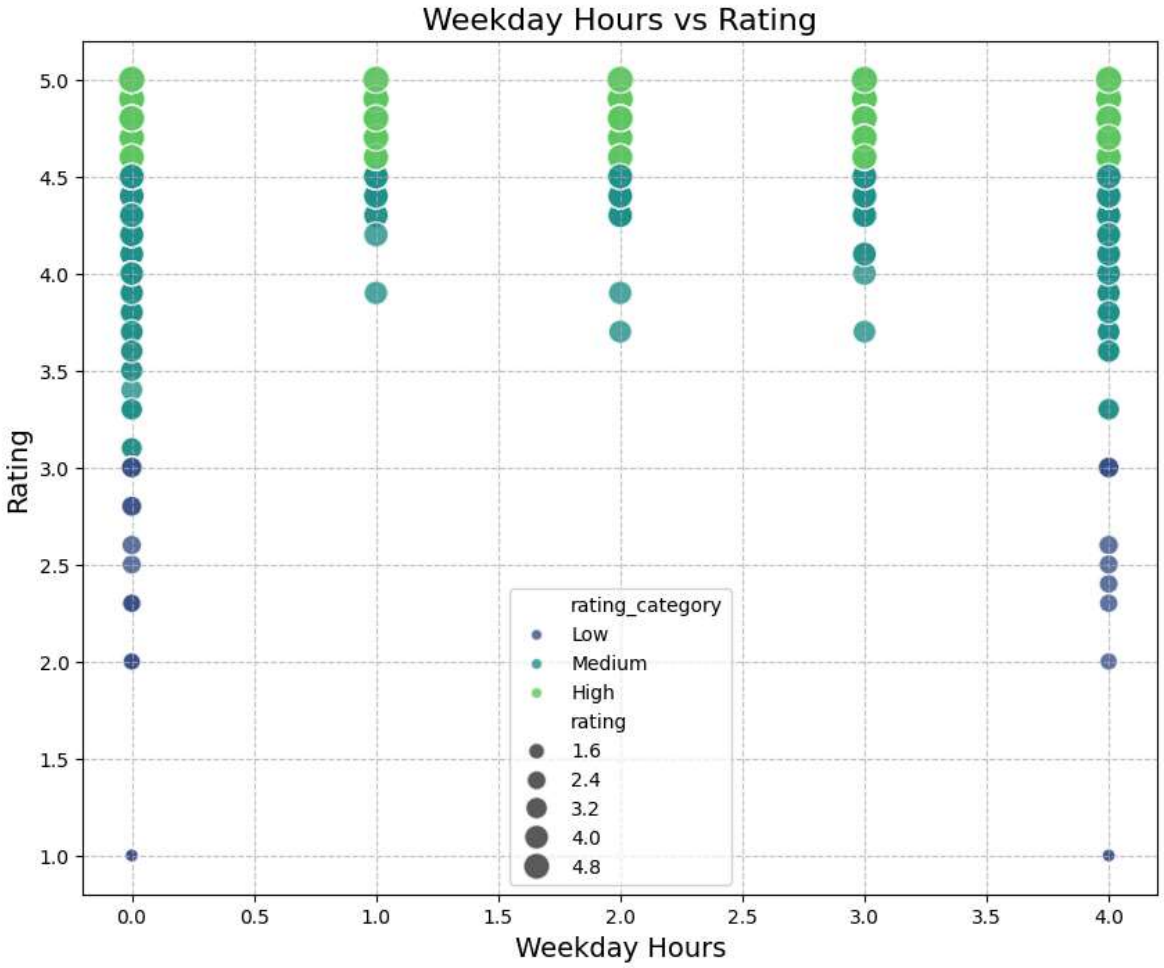
Question 2: Do locations with longer weekday operational hours tend to have higher popularity?

How it leads to our objective: This question helps in optimizing resource usage in sustainable tourism by balancing visiting hours during weekdays. By analyzing the correlation between weekday operational hours and popularity, we can identify the optimal hours of operation that attract more visitors during the week which will help in reducing the demand on weekends.

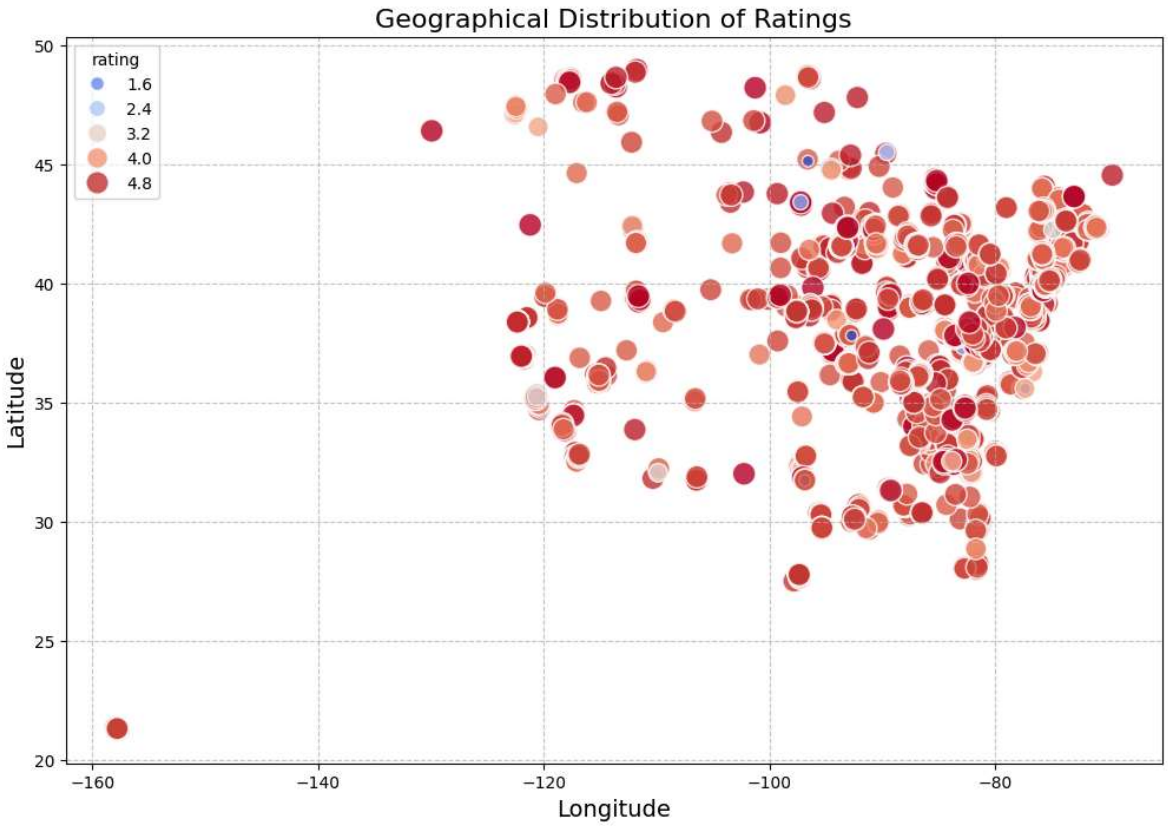
Why it is significant: The relationship between operational hours and popularity is crucial for identifying locations that attract tourists during weekdays. This hypothesis helps in encouraging weekday visits, which in turn reduces the weekend rush and promotes a more consistent flow of visitors throughout the week. This will enhance the overall tourist experience.

TASK 5: EDA for the Questions

In [16]:
```
# Hypothesis 1: Correlation between weekday hours and rating
plt.figure(figsize=(10, 8))
sns.scatterplot(x='weekday_hours', y='rating', data=df_cleaned, hue='rating_cate
plt.title('Weekday Hours vs Rating', fontsize=16)
plt.xlabel('Weekday Hours', fontsize=14)
plt.ylabel('Rating', fontsize=14)
plt.grid(True, linestyle='--', alpha=0.7)
plt.show()
```



In [17]:
```
# Hypothesis 2: Geographical relationship with rating (latitude and longitude)
plt.figure(figsize=(12, 8))
sns.scatterplot(x='longitude', y='latitude', hue='rating', size='rating', data=d
plt.title('Geographical Distribution of Ratings', fontsize=16)
plt.xlabel('Longitude', fontsize=14)
plt.ylabel('Latitude', fontsize=14)
plt.grid(True, linestyle='--', alpha=0.7)
plt.show()
```



TASK 2

Name: Keshav Narayan Srinivasan UBIT: 50610509

Question 1: Do states with higher average ratings offer a better overall tourism experience?

How it is significant: As we Analyze the average ratings by state, we can be able to provide valuable insights into the overall tourism quality that is being in different regions especially in States this case which has consistently high ratings might offer better services and tourism attractions, while those with lower ratings may have issues that need to be addressed.

How it leads to objective: We can do this by identifying high-potential states and understanding what makes them successful in terms of the tourist's satisfaction

Question 2: How does the variance between weekend working hours impact customer ratings?

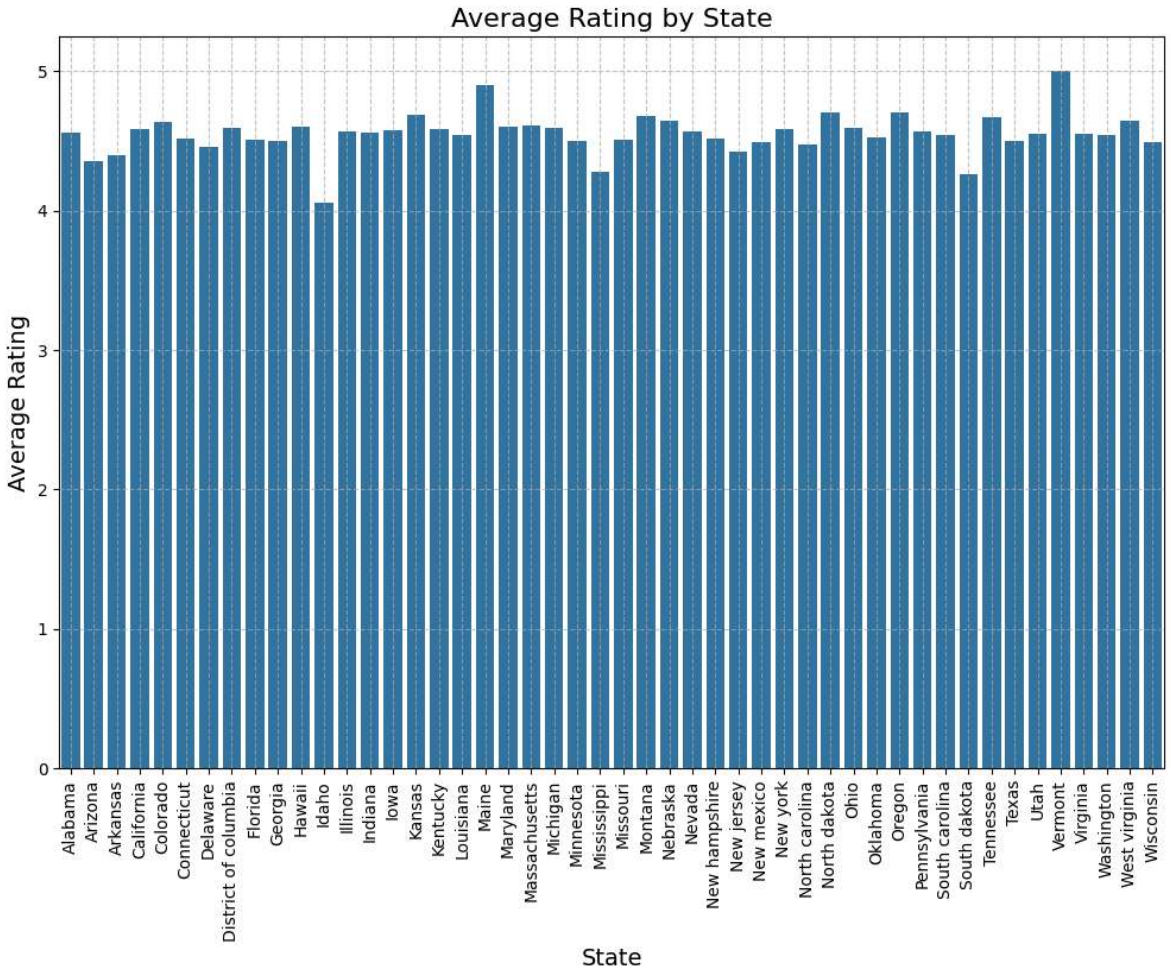
How it is significant: Large amount of variance in working hours might affect customer's enjoyment and their satisfaction. Their comfortableness will be ruined with lower ratings. For example, if a location offers less hours on the weekend, it may lead to customer dissatisfaction, especially for office goers who are only free in weekend.

How it leads to objective: By understanding this relationship, Tourism companies and places can change and optimize their hours, leading to higher satisfaction by balancing the demand to adjust both the weekend and weekday population.

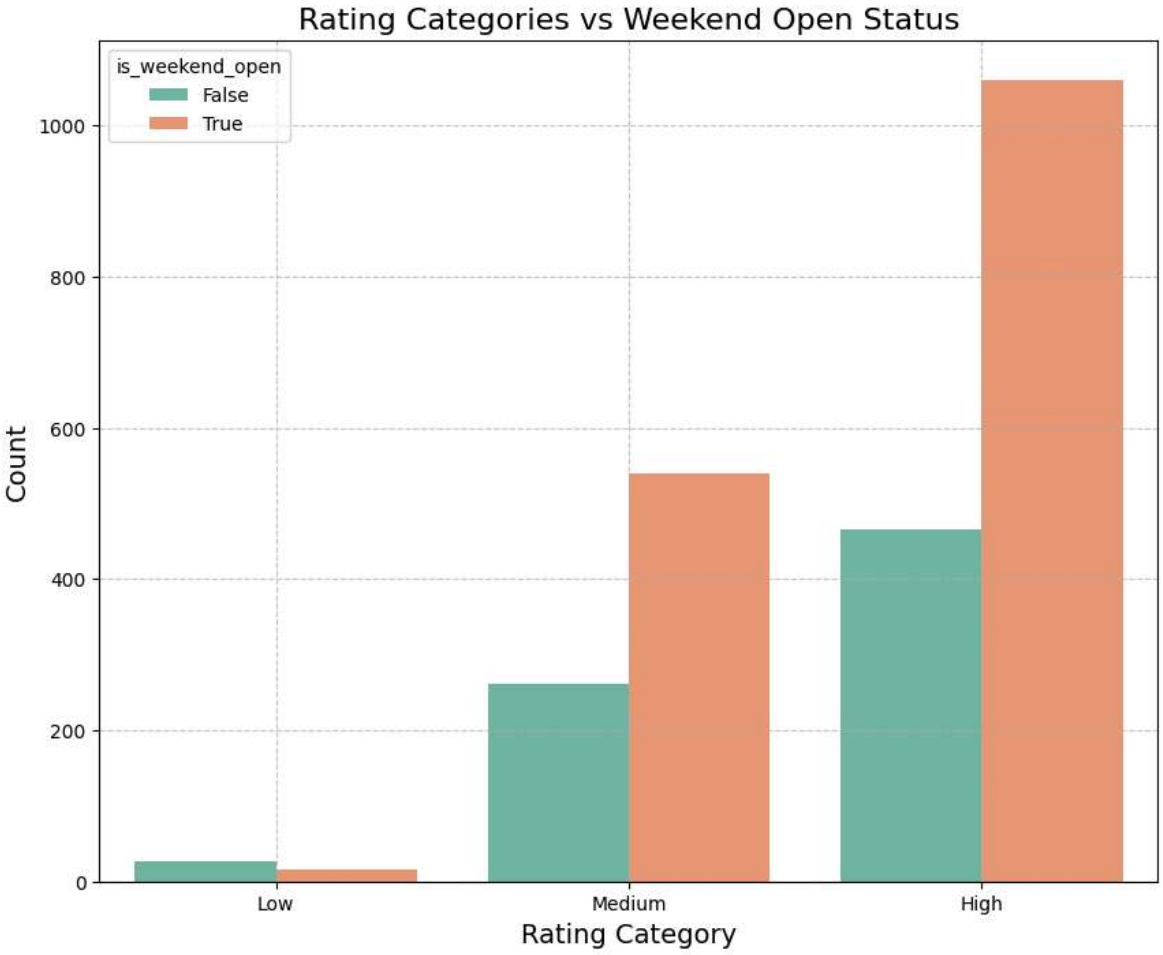
TASK 5: EDA for the Questions

In [18]:
```
# Hypothesis 1: Average rating by state
state_avg_rating = df_cleaned.groupby('us_state')['rating'].mean().reset_index()

plt.figure(figsize=(12, 8))
sns.barplot(x='us_state', y='rating', data=state_avg_rating)
plt.title('Average Rating by State', fontsize=16)
plt.xlabel('State', fontsize=14)
plt.ylabel('Average Rating', fontsize=14)
plt.xticks(rotation=90)
plt.grid(True, linestyle='--', alpha=0.7)
plt.show()
```



In [19]:
```
# Hypothesis 2: Rating categories and weekend open status
plt.figure(figsize=(10, 8))
sns.countplot(x='rating_category', hue='is_weekend_open', data=df_cleaned, palet
plt.title('Rating Categories vs Weekend Open Status', fontsize=16)
plt.xlabel('Rating Category', fontsize=14)
plt.ylabel('Count', fontsize=14)
plt.grid(True, linestyle='--', alpha=0.7)
plt.show()
```



Task 2

Hari Chandan Gooda, 50614165

Question 1

Does geographical pattern affect weekend availability?

How it is significant: Understanding whether certain regions are more likely available during weekends can give us insights about the regional crowd. This understanding will help us first prioritize local tourism, then filter out remaining places to recommend to other tourists. This way it not only promotes economic boost but also sustainable tourism.

How it leads to the objective: This is helpful for our objective since it meticulously handles tourism, especially during weekends, which is considered to be critical time by many tourists.

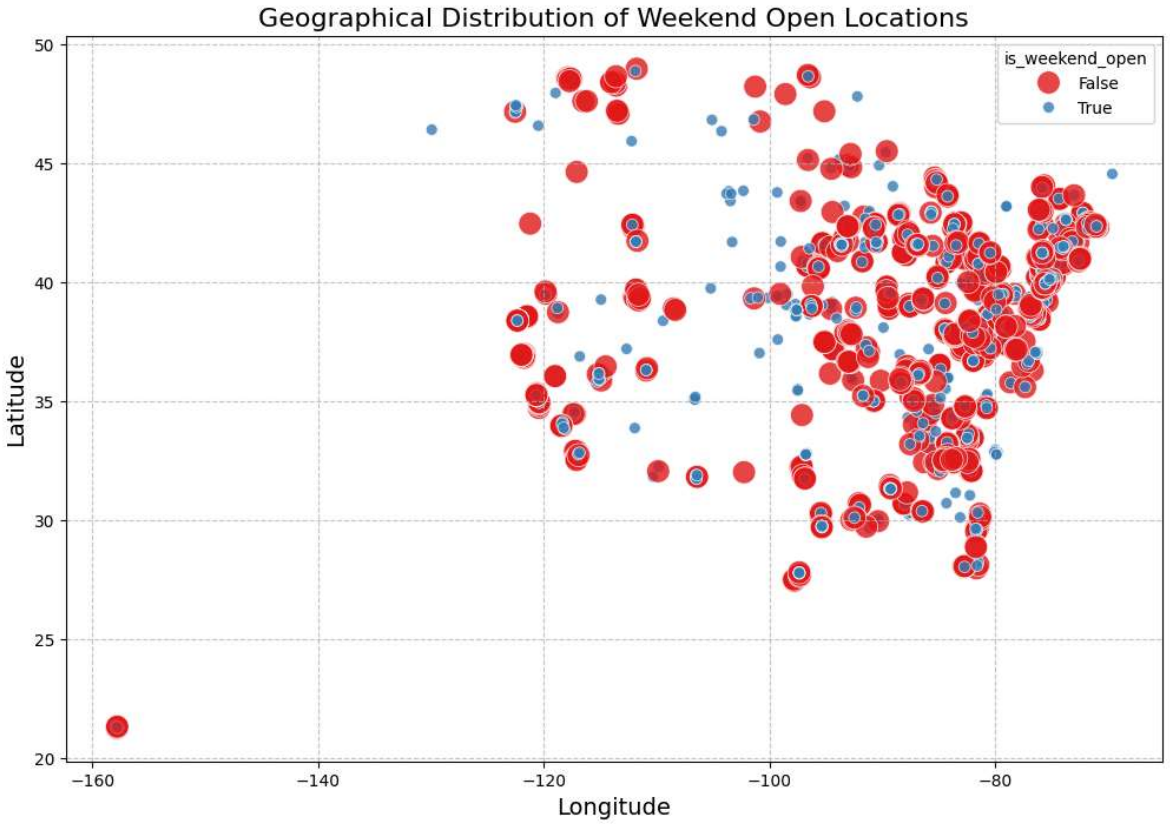
Question 2: How does the operational time (working hours) of a location affect its popularity?

How it leads to our objective: By identifying this, it will reveal the optimistic visiting hours for the tourists and it will recommend the time slots for feasible tourism, mainly for places which prioritize the low environmental impact on off-peak hours.

Why it is significant: It will recognize the relationship between how long is the location hours (Working hours) and its popularity can able to predict the best hours for tourists. Location which has longer hours may attract more visitors.

TASK 5: EDA for the Questions

In [20]:
```
# Hypothesis 1: Geographical distribution of weekend open status
plt.figure(figsize=(12, 8))
sns.scatterplot(x='longitude', y='latitude', hue='is_weekend_open', data=df_clea
plt.title('Geographical Distribution of Weekend Open Locations', fontsize=16)
plt.xlabel('Longitude', fontsize=14)
plt.ylabel('Latitude', fontsize=14)
plt.grid(True, linestyle='--', alpha=0.7)
plt.show()
```



In [21]:
```
# Hypothesis 2: Regular working hours and rating
import datetime

def calculate_hours_variance(working_hours):
    try:
        hours = re.findall(r'(\d{2}):(\d{2})-(\d{2}):(\d{2})', working_hours)
        if hours:
            start_hour, start_min, end_hour, end_min = map(int, hours[0])
            start_time = datetime.time(start_hour, start_min)
            end_time = datetime.time(end_hour, end_min)
            start_dt = datetime.datetime.combine(datetime.date.today(), start_ti
            end_dt = datetime.datetime.combine(datetime.date.today(), end_time)
            return (end_dt - start_dt).seconds / 3600.0
    except Exception as e:
        return 0
```

```
df_cleaned['hours_variance'] = df_cleaned['working_hours'].apply(calculate_hours

plt.figure(figsize=(10, 8))
sns.scatterplot(x='hours_variance', y='rating', data=df_cleaned, hue='rating_cat
plt.title('Working Hours Variance vs Rating', fontsize=16)
plt.xlabel('Hours Variance (Weekday - Weekend)', fontsize=14)
plt.ylabel('Rating', fontsize=14)
plt.yscale('log')
plt.grid(True, linestyle='--', alpha=0.7)
plt.show()
```