Travel Satisfaction Predictor

Computational Analysis of Big Data B

John Iwenofu, Beste Kuruefe, Abdi Kusata

GitHub Repository: https://github.com/jiwenofu/Big-Data-Project

Introduction

Travel can be rewarding and pleasurable for most people. However, planning it can be tiring and stressful. As study abroad students, we want to take full advantage of the opportunity to explore new places and immerse ourselves in different cultures during our time abroad. There are various destinations offered, but we prefer to go to one that best satisfies our tastes and preferences.

This project involves simulating the best travel destination based on predicted satisfaction levels across various categories. Using a travel dataset from Kaggle, we trained machine learning models to predict a user's travel satisfaction score (0 for not satisfied, 1 for satisfied) based on the parameters they value—such as travel type, total cost, hotel rating, and more. We hope that by achieving strong prediction accuracy, users will be able to compare expected satisfaction scores for different destinations and choose the one they are most likely to enjoy.

Our first research question was how we can use feature engineering to better understand the relationship between seasons and travel satisfaction. Initially, we only had travel dates and no column showing the season, but we believed this information could provide valuable insights. Therefore, we extracted the season from the travel dates, which will be discussed in more detail in the Data section.

Our second research question was how we can use data visualization techniques to reveal the relationship between destination, travel type, and travel satisfaction patterns—such as by hotel rating, season and destination, total cost range, and travel duration. To explore this, we used line plots, heatmaps, bar plots, and histograms to identify patterns across various travel features, which will be discussed in more detail in the Methodology section.

Our third research question was how well machine learning models like Logistic Regression and Random Forest can predict travel satisfaction, and which model performs better. To answer this, we trained both models on the cleaned dataset and compared their accuracies and classification reports, which are detailed in the Results section.

To ensure we answer all the research questions above, this report will first provide an overview of the dataset and the preprocessing steps we took to best represent the data over a one-year period. We will then present the findings from our data visualizations to address the satisfaction patterns. Finally, we will evaluate how well machine learning models can predict travel satisfaction and determine which model performs best.

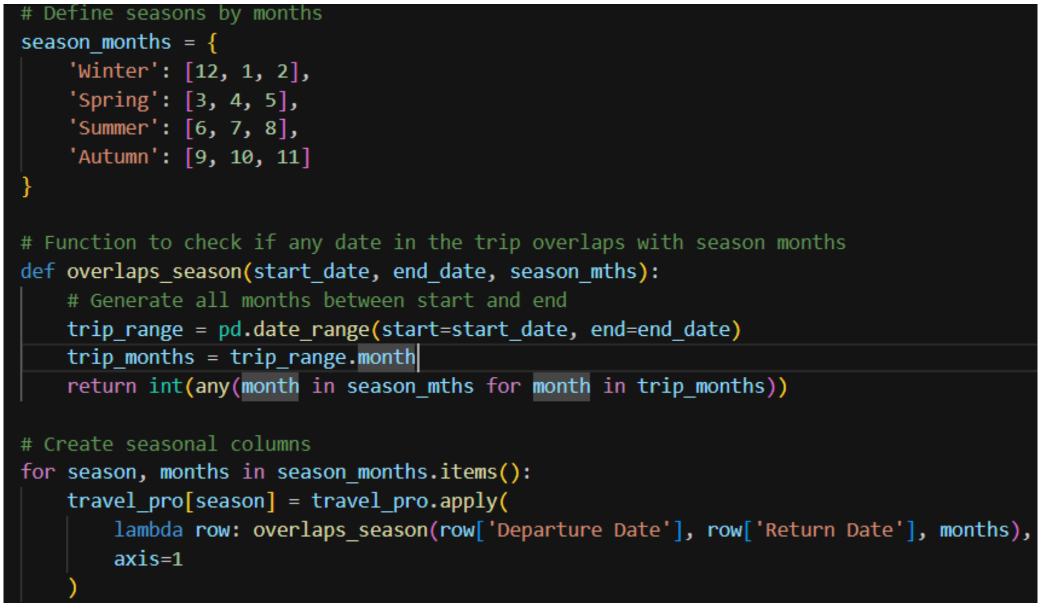
Data

a. The Dataset

The dataset that was used in the project is from the *Kaggle* database, called Predicting User Travel Satisfaction: https://www.kaggle.com/datasets/gauravduttakiit/predicting-user-travel-satisfaction/data. This dataset includes 10 columns, such as the user trip, destination, departure/return date, travel type, transportation mode, hotel rating, total cost, and customer satisfaction. The raw dataset includes 19,800 data points.

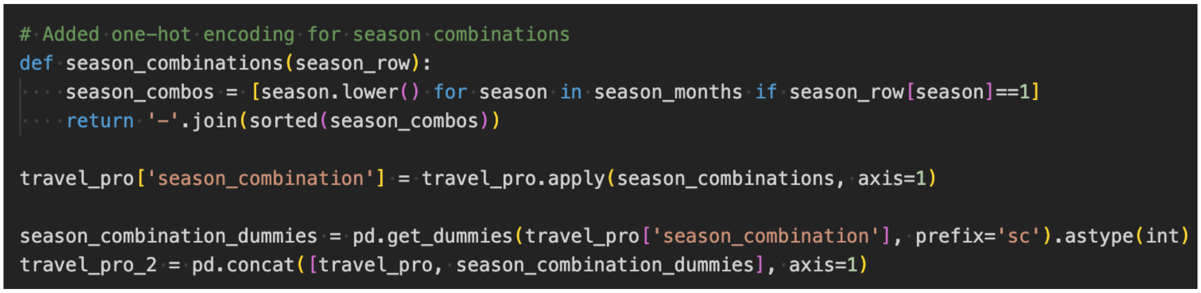
b. Adding Season Columns

Our first research question was how we could use feature engineering to better understand the relationship between seasons and travel satisfaction. To achieve this, we added a season column based on each user's departure and return dates. This was implemented using one-hot encoding, meaning each season was represented by its own column. However, since travel durations can span multiple seasons, overlaps may occur. For example, a user might depart in November and return in December. To address this, we introduced additional columns to represent combined seasonal durations—such as a column for an Autumn–Winter span.



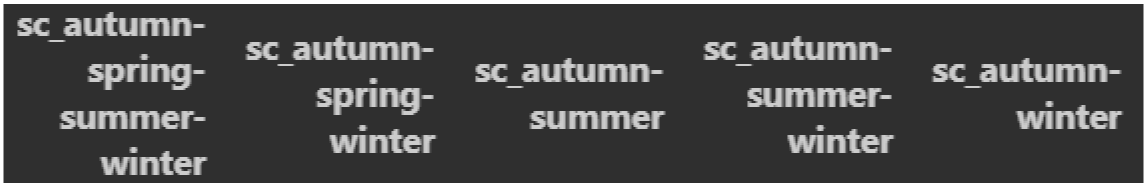
```python
# Define seasons by months
season_months = {
    'Winter': [12, 1, 2],
    'Spring': [3, 4, 5],
    'Summer': [6, 7, 8],
    'Autumn': [9, 10, 11]
}

# Function to check if any date in the trip overlaps with season months
def overlaps_season(start_date, end_date, season_mths):
    # Generate all months between start and end
    trip_range = pd.date_range(start=start_date, end=end_date)
    trip_months = trip_range.month
    return int(any(month in season_mths for month in trip_months))

# Create seasonal columns
for season, months in season_months.items():
    travel_pro[season] = travel_pro.apply(
        lambda row: overlaps_season(row['Departure Date'], row['Return Date'], months),
        axis=1
    )
```

Figure 1. The screenshot of the Jupyter notebook where we created columns for the seasons (Cell 64 in TravelProject.ipynb)



```
# Added one-hot encoding for season combinations
def season_combinations(season_row):
    season_combos = [season.lower() for season in season_months if season_row[season]==1]
    return '-'.join(sorted(season_combos))

travel_pro['season_combination'] = travel_pro.apply(season_combinations, axis=1)

season_combination_dummies = pd.get_dummies(travel_pro['season_combination'], prefix='sc').astype(int)
travel_pro_2 = pd.concat([travel_pro, season_combination_dummies], axis=1)
```

Figure 2. The screenshot of the Jupyter notebook where we created columns for the season combinations (Cell 65 in TravelProject.ipynb)



sc_autumn-spring-summer-winter	sc_autumn-spring-winter	sc_autumn-summer	sc_autumn-summer-winter	sc_autumn-winter

Figure 3. The screenshot of the one-hot encodings for season combinations

c. *Adding a Travel Satisfaction Column*

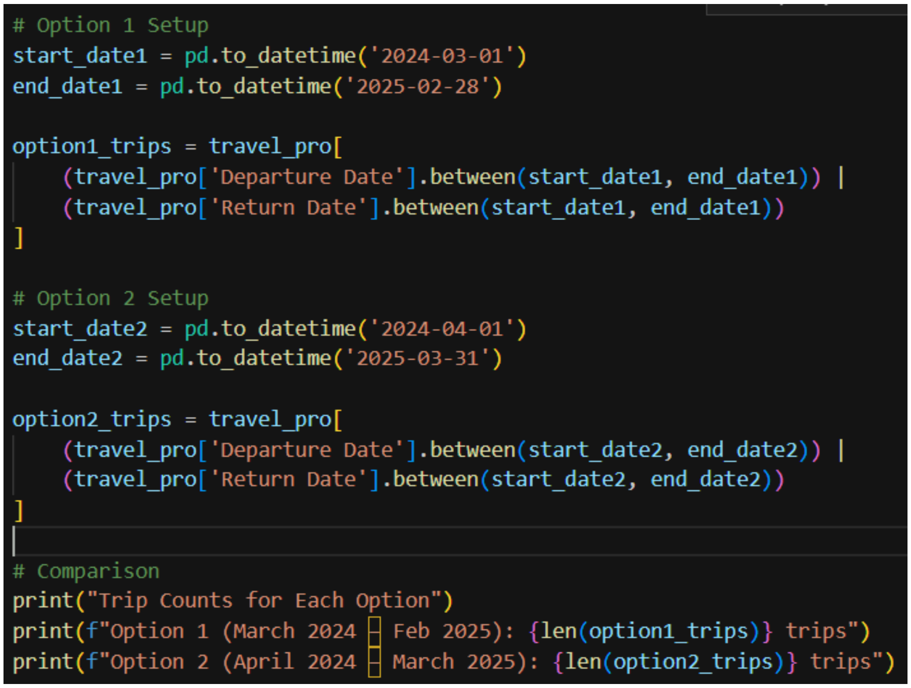
Next, we added a travel satisfaction column in the dataset, called cust_sat_summary, which outputs a 0 if its satisfaction is from 1-5 and a 1 if it is from 6-10. 0 means that the user is not fully satisfied, while 1 means that the user is satisfied.

d. *Adding One-Hot Encodings for Travel Type, Transportation Modes, and Destination and a Travel Duration Column*

Then, one-hot encoding was applied to the following characteristics: travel type, transportation modes, and destination. This was done so that it can be easier to train the dataset using the machine learning models. We also added the travel duration in days, which calculates the difference between the return date and the departure date in datetime format.

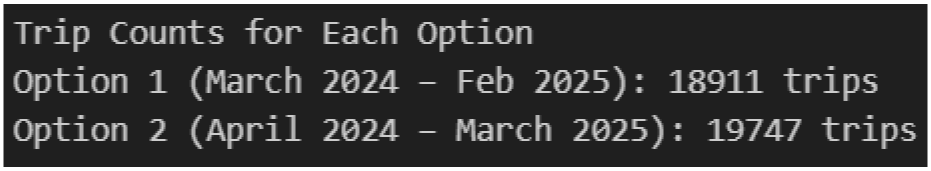
e. *Deciding on Which of the 12-Month Period to Use*

Looking at the dataset, the travel ranges from 2024 to 2025. Due to the dataset spanning over the two years, we decided to focus on a 12-month period to use as our true dataset. Our goal was to find which 12-month period captures the most trips, considering trips where either the departure or return date falls within the period. We had two options. Option 1 was March 1st, 2024 - February 28th, 2025, while Option 2 was April 1st, 2024 - March 31st, 2025.



```
# Option 1 Setup
start_date1 = pd.to_datetime('2024-03-01')
end_date1 = pd.to_datetime('2025-02-28')

option1_trips = travel_pro[
    (travel_pro['Departure Date'].between(start_date1, end_date1)) |
    (travel_pro['Return Date'].between(start_date1, end_date1))
]

# Option 2 Setup
start_date2 = pd.to_datetime('2024-04-01')
end_date2 = pd.to_datetime('2025-03-31')

option2_trips = travel_pro[
    (travel_pro['Departure Date'].between(start_date2, end_date2)) |
    (travel_pro['Return Date'].between(start_date2, end_date2))
]

# Comparison
print("Trip Counts for Each Option")
print(f"Option 1 (March 2024 □ Feb 2025): {len(option1_trips)} trips")
print(f"Option 2 (April 2024 □ March 2025): {len(option2_trips)} trips")
```

Figure 4. The screenshot of the Jupyter notebook where we explored which 12-month period to use (Cell 73 in TravelProject.ipynb)



```
Trip Counts for Each Option
Option 1 (March 2024 – Feb 2025): 18911 trips
Option 2 (April 2024 – March 2025): 19747 trips
```

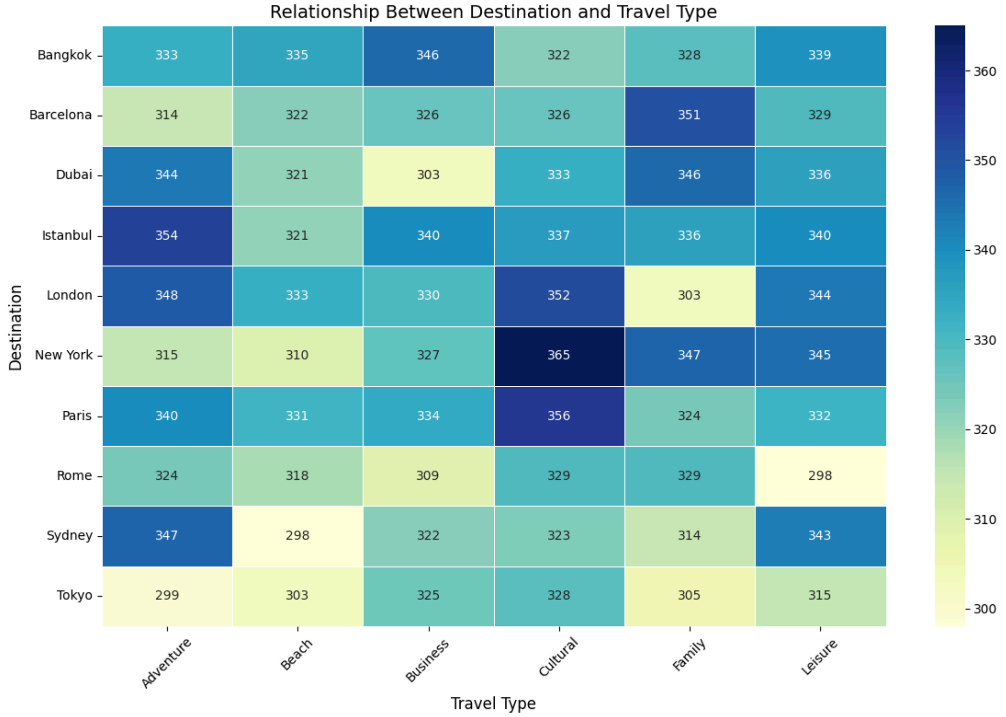
Figure 5. The screenshot of the code output after we ran cell 73, which is shown above in Figure 4.

It was determined that Option 2 would be the favorite, considering that there are more trips that happened within that period than Option 1, as shown in Figure 5. Therefore, we cut the dataset to only consider the travel between April 2024 - March 2025.

Methodology

After completing the preprocessing and feature engineering steps (as described in the Data section), we explored patterns in travel satisfaction to address the second research question and to better understand our dataset before training machine learning models. We created several visualizations to examine how satisfaction varied by destination, season, travel type, cost, and duration. These visualizations helped us identify features that might be most useful for predicting satisfaction and provided insights into user preferences and trends.

a. Destination and Travel Type



Destination	Adventure	Beach	Business	Cultural	Family	Leisure
Bangkok	333	335	346	322	328	339
Barcelona	314	322	326	326	351	329
Dubai	344	321	303	333	346	336
Istanbul	354	321	340	337	336	340
London	348	333	330	352	303	344
New York	315	310	327	365	347	345
Paris	340	331	334	356	324	332
Rome	324	318	309	329	329	298
Sydney	347	298	322	323	314	343
Tokyo	299	303	325	328	305	315

Figure 6. This heat map shows the relationship between travel destinations and travel types, with the intensity of the color representing the number of trips for each combination. For example, 'Cultural' trips are significant in destinations like New York and Paris.

b. Satisfaction Level by Hotel Rating

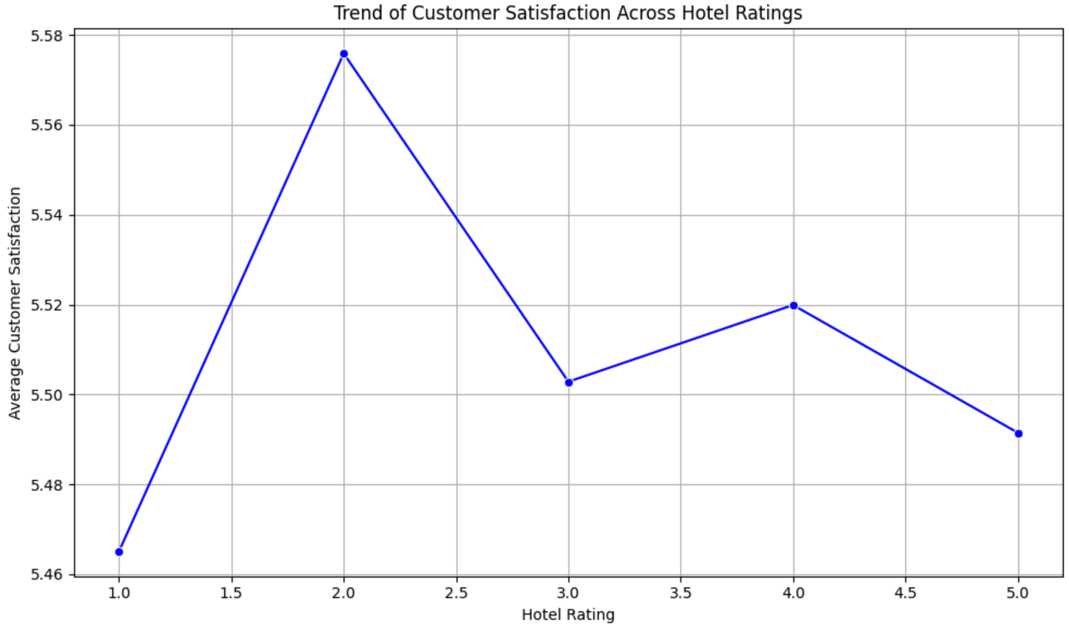
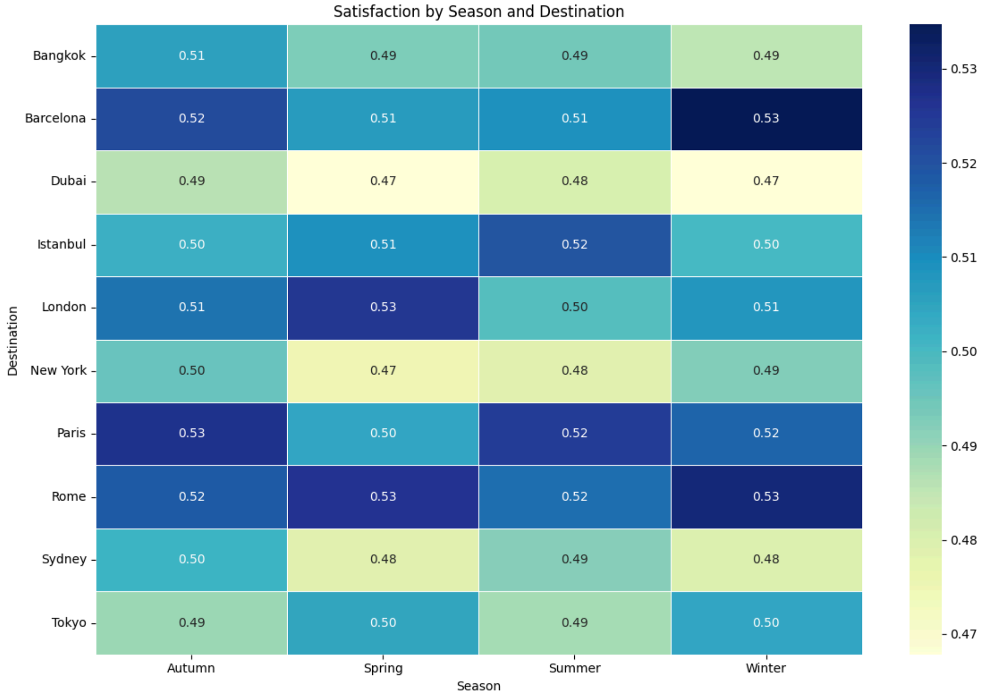


Figure 7. This graph shows the trend of average customer satisfaction across different hotel ratings. It reveals that satisfaction peaks at a rating of 2.0 but declines as the ratings increase.

c. Satisfaction Level by Season and Destination



Destination	Autumn	Spring	Summer	Winter
Bangkok	0.51	0.49	0.49	0.49
Barcelona	0.52	0.51	0.51	0.53
Dubai	0.49	0.47	0.48	0.47
Istanbul	0.50	0.51	0.52	0.50
London	0.51	0.53	0.50	0.51
New York	0.50	0.47	0.48	0.49
Paris	0.53	0.50	0.52	0.52
Rome	0.52	0.53	0.52	0.53
Sydney	0.50	0.48	0.49	0.48
Tokyo	0.49	0.50	0.49	0.50

Figure 8. This heat map illustrates customer satisfaction across different destinations and seasons, with darker colors indicating higher satisfaction levels. For example, satisfaction is highest in Barcelona and Rome during the winter season.

d. Satisfaction Level by Total Cost Range

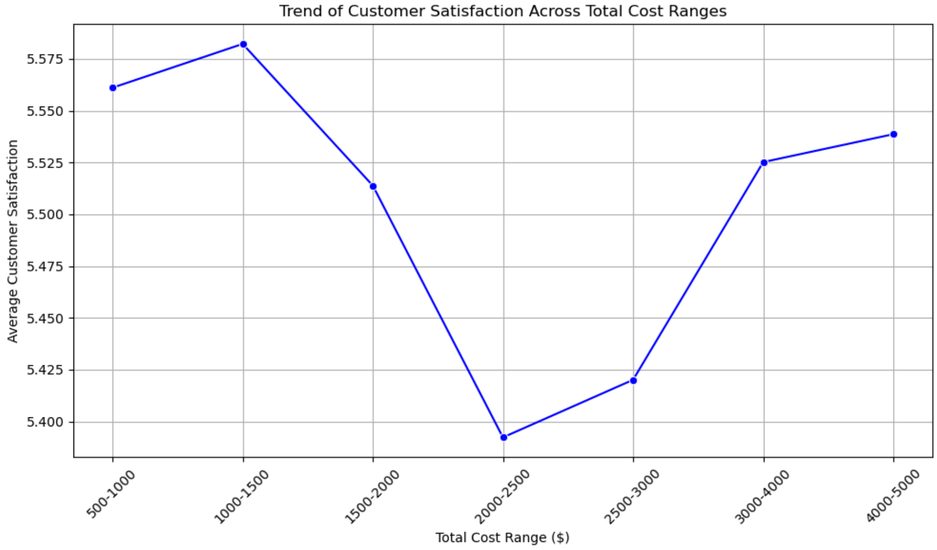


Figure 9. This graph shows the trend of average customer satisfaction across different total cost ranges for trips. It highlights that satisfaction is highest in the lower cost ranges (e.g., $1000–1500) and decreases as costs increase, with a slight recovery in higher cost ranges (e.g., $3000–5000).

e. *Satisfaction Level by Travel Duration*

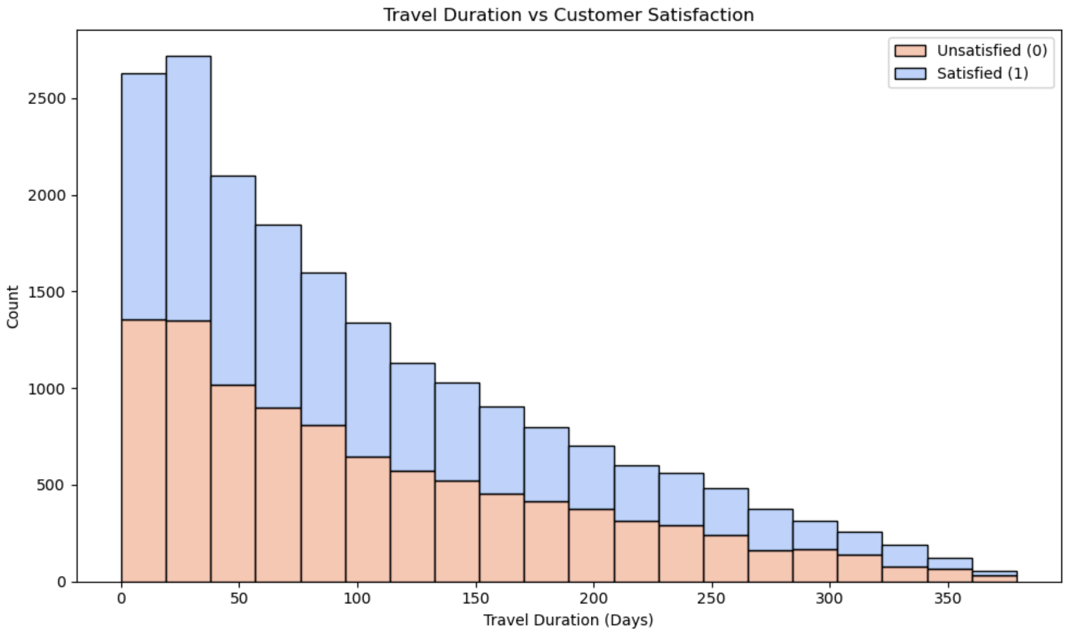


Figure 10. The figure represents the distribution of travel durations (in days) and their relationship to customer satisfaction, with the histogram stacked to show the counts of satisfied (1) and unsatisfied (0) customers. It highlights how satisfaction levels vary across different trip lengths, with shorter trips being more frequent.

Next, we trained and evaluated two classification models—Logistic Regression and Random Forest—using the engineered dataset. We chose these two models to compare a basic model that works well with straightforward data with a more advanced model that can handle more complex patterns. Their performance was assessed using standard classification metrics such as accuracy, precision, recall, and F1-score, which are discussed in the Results section.

Results

Brief Overview of How Logistic Regression and Random Forest Models Work

To address our final research question on the performance of machine learning models in predicting travel satisfaction, we compared two widely used models: Logistic Regression and Random Forest.

Logistic Regression is used for classification problems where the outcome is binary. In our case, the model predicts whether a traveler is satisfied (1) or not satisfied (0). It calculates the probability of satisfaction based on the input features and classifies the outcome accordingly (“Logistic Regression”). It works best when there is a linear relationship between the input variables and the outcome (“Logistic Regression vs Random Forest Classifier”).

Random Forest is also used for classification but works differently. It builds multiple decision trees using different parts of the data and combines their predictions to make a final decision (“Logistic Regression vs Random Forest Classifier”). This approach makes it more flexible and better at capturing complex patterns than Logistic Regression.

Since Random Forest can handle more complex patterns in the data, our hypothesis was that it would achieve better accuracy and F1 score than Logistic Regression.

Comparison of the Two Models in Our Project (Table Format):

Metric	**Logistic Regression**	**Random Forest**
Accuracy	0.4947	0.4944
Precision (Class 0)	0.50	0.50
Recall (Class 0)	0.48	0.48
F1-score (Class 0)	0.49	0.49
Precision (Class 1)	0.49	0.49
Recall (Class 1)	0.51	0.51
F1-score (Class 1)	0.50	0.50
Macro Avg F1	0.49	0.49
Weighted Avg F1	0.49	0.49

Figure 11. The comparison of the classifications report results for Logistic Regression and Random Forest

Definitions of Key Terms Highlighted in Figure 11 (According to Statology.org):

- Precision: Percentage of correct positive predictions relative to total positive predictions.
- Recall: Percentage of correct positive predictions relative to total actual positives.
- F1 Score: A weighted harmonic mean of precision and recall. The closer to 1, the better the model.

What These Terms Mean in the Context of Our Project for Both Logistic Regression and Random Forest:

- Precision (Class 1 – Satisfied): Out of all the travelers the model predicted would be satisfied, only 49% actually were.
- Recall (Class 1 – Satisfied): Out of all the travelers who were actually satisfied, the model correctly predicted 51% of them.
- Precision (Class 0 – Unsatisfied): Out of all the travelers the model predicted would be unsatisfied, only 50% actually were.
- Recall (Class 0 – Unsatisfied): Out of all the travelers who were actually unsatisfied, the model correctly predicted 48% of them.

Confusion Matrices of the Two Models:

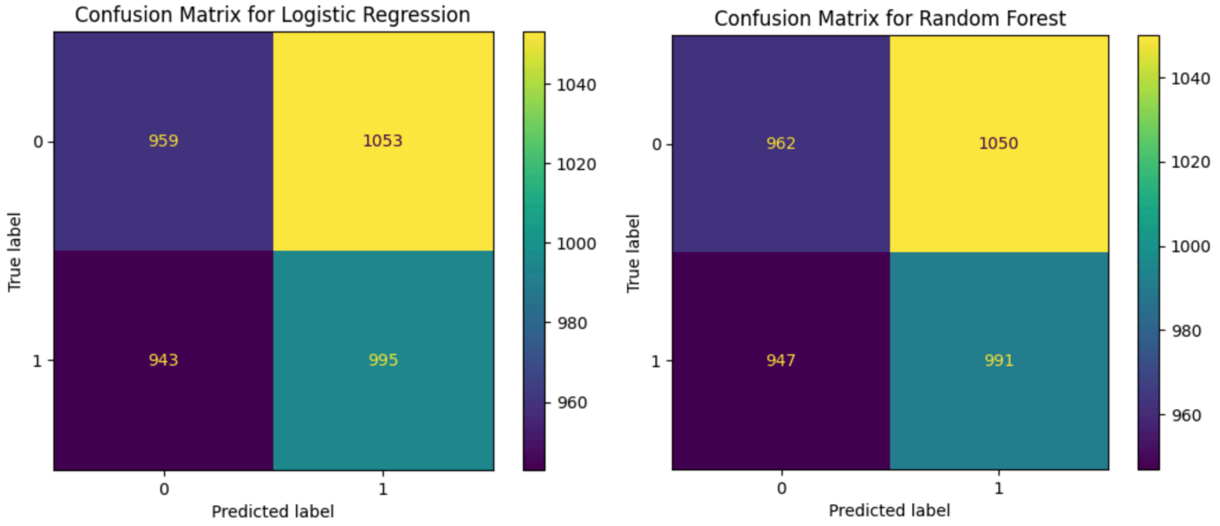


Figure 12. Confusion matrices for Logistic Regression (left) and Random Forest (right). The diagonal from the bottom-left to the top-right corner shows the number of incorrectly predicted travel satisfaction cases.

Comparison Summary for the Models:

Both the Logistic Regression and Random Forest models showed similar performance when we compared their accuracies, classification reports, and confusion matrices. Logistic Regression achieved a slightly higher accuracy (0.4947) than Random Forest (0.4944), which disproved our initial hypothesis that Random Forest would perform better. One possible reason is that the patterns in the data may have been more straightforward than we expected, so the added complexity and robustness of Random Forest were not necessary.

Both models performed slightly better at identifying satisfied travelers (recall ≈ 51%) than unsatisfied ones (recall ≈ 48%). However, the precision, recall, and F1-scores for both classes remained around 49–50%, indicating that neither model is particularly effective at predicting traveler satisfaction in their current form. This may be due to limited data or insufficiently informative features. More data, additional feature engineering, or experimenting with more advanced models may be needed to improve prediction performance.

Conclusion and Discussion

Looking at the model training, there is a nearly 50% split between correct and incorrect classifications using Logistic Regression and Random Forest models. Several factors may contribute to this result, such as varying travel preferences across trips. It is difficult to categorize overall characteristics into a single satisfaction level, as the model must consider transportation mode, hotel rating, the season(s) of travel, and total cost. Due to the differences in values, labels, and categorizations unique to each trip, it is challenging to identify a clear relationship within the dataset. This may help explain the 50% split in predicted satisfaction levels, as a clearer relationship would likely have led to stronger model performance.

Through the data visualization process, we learned that travelers choose destinations based on several personal factors. It's difficult to determine the overall popularity of a destination without considering what each traveler hopes to gain from the trip. For example, someone visiting Paris for a cultural experience may have a different level of satisfaction than someone visiting for leisure. Season also plays a role, as some cities are more appealing during certain times of the year. This is further influenced by geography—cities in the Southern Hemisphere experience opposite seasons compared to those in the Northern Hemisphere.

Some of our research findings were surprising, particularly the statistic that more people prefer to travel in the winter than in the summer. Initially, we expected summer to be more popular, given that many people are out of school and may have more time off work. However, winter's popularity is understandable, as it includes major holidays such as Christmas, Hanukkah, and New Year's. Additionally, as shown in Figure 6, some destinations in the Southern Hemisphere, like Sydney, experience summer during the Northern Hemisphere's winter months.

Possible future directions for this project can include having the user input their travel preferences, such as the season they are considering traveling, their travel type, and their budget. That way, our model can recommend a travel destination based on their information, increasing the likelihood of a satisfying future trip.

Works Cited

Bobbitt, Zach. “How to Interpret the Classification Report in Sklearn (with Example).” *Statology*, 9 May 2022, www.statology.org/sklearn-classification-report/.

Dutta, Gaurav. “Predicting User Travel Satisfaction.” *Kaggle*, 19 Mar. 2025, www.kaggle.com/datasets/gauravduttakiit/predicting-user-travel-satisfaction/data.

“Logistic Regression.” *Datatab*, datatab.net/tutorial/logistic-regression. Accessed 3 May 2025.

“Logistic Regression vs Random Forest Classifier.” *GeeksforGeeks*, GeeksforGeeks, 28 Apr. 2025, www.geeksforgeeks.org/logistic-regression-vs-random-forest-classifier/.