A FIELD PROJECT REPORT

on

Enhancing Flight Recommendations: Cold-Start Strategies

Submitted

By

P. Lahari	211FA04006
G. Satya Lakshmi	211FA04019
Ch. Saranyya	211FA04030
A. Jayanth	211FA04062

Under the Guidance of

Ms. Pushya Chaparala

Assistant Professor, Department of CSE

VIGNAN'S
Foundation for Science, Technology & Research
(Deemed to be University)
-Estd. u/s 3 of UGC Act 1956

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VIGNAN'S FOUNDATION FOR SCIENCE TECHNOLOGY & RESEARCH

(Deemed to be University)

Vadlamudi, Guntur

Andhra Pradesh, India, Pin-522213

May-2024

VIGNAN'S

Foundation for Science, Technology & Research

(Deemed to be University)

-Estd. u/s 3 of UGC Act 1956

CERTIFICATE

This is to certify that the report entitled as **"Enhancing Flight Recommendations: Cold-Start Strategies**" was submitted by **"P. Lahari (211Fa04006), G. Satya Lakshmi (211FA04019), Ch. Saranyya (211FA04030), A. Jayanth (211FA04062)"** in the partial of Field Project is a bonafide work carried out under the supervision Ms. Pushya Chaparala, Assistant Professor, Department of CSE, VFSTR Deemed to be University.

Ms. Pushya Chaparala	Dr. K. V. Krishna Kishore
Assistant Professor, Dept. of CSE	HOD, CSE & Dean, SoCI

DECLARATION

We hereby declare that the Field Project entitled **“Enhancing Flight Recommendations: Cold-Start Strategies”** is being submitted by **“P. Lahari (211Fa04006), G. Satya Lakshmi (211FA04019), Ch. Saranyya (211FA04030), A. Jayanth (211FA04062)”** in partial fulfilment of Field Project course work. This is our original work, and this project has not formed the basis for the award of any degree. We have worked under the supervision of Ms. Pushya Chaparala, Assistant Professor, Department of CSE.

By,

P. Lahari (211FA04006)

G. Satya Lakshmi (211FA04019)

Ch. Saranyya (211FA04030)

A. Jayanth (211FA04062)

Date: 03/05/2024

ABSTRACT

This exploration dives into tackling the challenge of flight ticket recommendation within the realm of online travel platforms, where users often struggle to find tickets that match their preferences due to the cold-start issue. This issue is divided into two main categories: route cold-start and user cold-start. To address these challenges, this work suggests innovative strategies drawn from an exploratory study within the flight ticket recommendation domain. Through an examination of real-world flight ticket recommendation scenarios, it introduces methods that exploit route similarity and social connections among passengers to improve user models. By integrating these factors, the proposed methods aim to overcome the constraints posed by the cold-start problem, thus enhancing the accuracy and relevance of flight ticket recommendations. Key contributions of this exploration involve the creation of techniques that translate an enhanced user preference model and flight attributes into latent factor spaces. This translation process facilitates the generation of personalized recommendation outcomes that cater to individual user preferences while considering various flight characteristics. Experimental validation carried out on real-world data illustrates the effectiveness and practicality of the suggested methods. Drawing from insights gleaned from the experimental findings, this exploration underscores the potential of the proposed strategies to enrich the overall user experience on flight ticket booking platforms. This exploration provides valuable insights and solutions to tackle the cold-start problem in flight ticket recommendation systems. By incorporating route similarity, social connections, and latent factor mapping techniques, the proposed methods pave the way for more precise and personalized flight ticket recommendations, ultimately enhancing user satisfaction and engagement on online travel booking platforms. While the current implementation utilizes simulated data and functions, it outlines a framework that can be seamlessly integrated with real-world datasets and search systems to improve flight ticket recommendation processes. This modular approach enables scalability and customization, allowing developers to refine the algorithm for different use cases and user preferences.

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

1. INTRODUCTION

The introduction offers a comprehensive view of the burgeoning online travel industry and the pivotal role played by travel agencies, especially online platforms like Ctrip, Bookairfare, and AirFareExperts. These platforms provide a wide range of services, including flight ticket booking, which has gained popularity due to its efficiency and convenience. However, with the multitude of flight options available, selecting the most suitable ticket poses a significant challenge for passengers. To tackle this issue, personalized flight ticket recommendation systems have emerged as a crucial solution, aiming to tailor recommendations based on individual preferences and travel motivations. Recommender systems have gained widespread traction across various domains, yet their application to flight ticket recommendation presents unique challenges. Unlike traditional products, flight ticket features are dynamic, with fluctuating prices and uneven distributions across different routes. Additionally, the cold-start problem exacerbates the difficulty of recommending flight tickets, particularly for users with limited historical data. Traditional recommendation methods such as collaborative filtering face limitations in this context due to the scarcity of historical records for most passengers on many routes.

To overcome these challenges, specialized recommendation approaches specifically designed for flight tickets are proposed. These approaches involve enhancing user models using route similarity and social relationships between passengers to mitigate the cold-start problem. Furthermore, a method is introduced to map user preferences and flight features into latent factor spaces, thereby generating personalized recommendation results. The algorithms presented play a crucial role in implementing these recommendation approaches.

The first algorithm, User Model Construction, lays the foundation for personalized flight ticket recommendation by constructing a model based on historical orders and ticket features. It analyses the user's past behaviour and preferences to create a representation of their ticket choices. By iterating through historical orders and ticket features, it builds a dictionary-based model capturing the frequency of ticket choices for each feature. Additionally, it calculates feature weights based on entropy, providing a measure of the importance of each feature in influencing the user's ticket selection. This algorithm ensures that the user model accurately reflects the user's preferences, enabling personalized recommendations.

The Flight Ticket Recommendation Algorithm leverages the user model constructed in the previous step to rank flight tickets according to their relevance to the user's preferences. By considering historical orders, ticket features, and a candidate ticket list, this algorithm calculates a relevance score for each ticket based on its alignment with the user's past choices. This scoring mechanism enables the system to generate a ranked list of recommended tickets, facilitating informed decision-making for the user. The algorithm's ability to prioritize tickets based on user preferences enhances the overall user experience and increases the likelihood of booking satisfaction.

User Model Transfer from Similar Routes enhances the user model by incorporating information from routes like the recommended one. By analysing historical orders across different routes and extracting feature distributions, this algorithm enriches the user model with insights from related routes. It leverages route similarity and feature normalization techniques to ensure that the enhanced user model captures relevant information for the recommended route. This approach mitigates the cold-start problem by leveraging existing data from similar contexts, thereby improving the accuracy of ticket recommendations for less-explored routes.

User Model Enhancement Based on Social Relationships enhances the user model by considering social relationships among passengers. By identifying related users and determining influential individuals, this algorithm enriches the user model with insights from socially connected passengers. It leverages the user model of influential users to enhance the recommendation process for the target user, thereby improving the relevance of ticket recommendations. This approach acknowledges the influence of social connections on travel preferences and incorporates them into the recommendation framework.

Ticket Recommendation by Combining Explicit and Latent Factors generates ranked ticket recommendations by combining explicit user preferences with latent factors derived from historical orders. By iteratively updating latent factors based on a learning algorithm, this approach captures the dynamic nature of user preferences and ticket features. The combination of explicit and latent factors enables the system to provide personalized recommendations that align closely with the user's preferences and behaviour. This algorithmic approach ensures that the recommendation process remains adaptive and responsive to evolving user needs and preferences.

Finally, these contributions lie in its innovative recommendation approaches tailored specifically for flight tickets. By leveraging route similarity, social relationships, and latent factor mapping techniques, these approaches aim to address the cold-start problem and provide personalized flight ticket recommendations, thereby improving user satisfaction and engagement in online travel booking platforms.

CHAPTER-2

METHODOLOGY

2. METHODOLOGY

Firstly, historical flight ticket orders are collected from online travel agencies, forming the basis of user modelling. This data includes information such as departure and arrival cities, airline choices, ticket classes, and prices.

Next, the User Model Construction algorithm analyses the historical orders to construct a user preference model. By examining the frequency of ticket choices for various features and calculating feature weights based on entropy, this algorithm creates a detailed representation of the user's preferences.

Simultaneously, the Flight Ticket Recommendation Algorithm utilizes the user model to rank flight tickets based on their relevance to the user's preferences. By considering historical orders, ticket features, and candidate ticket lists, this algorithm generates a ranked list of recommended tickets.

To address the cold-start problem, where the limited historical data may hinder the recommendation accuracy, the User Model Transfer from Similar Routes algorithm enhances the user model by incorporating information from similar routes. It leverages route similarity and feature normalization techniques to ensure that the enhanced user model captures relevant information for the recommended route.

Additionally, the User Model Enhancement Based on Social Relationships algorithm leverages social connections among passengers to enrich the user model further. By identifying related users and determining influential individuals, this algorithm enhances the recommendation process for the target user, thereby improving the relevance of ticket recommendations.

Finally, the Ticket Recommendation by Combining Explicit and Latent Factors algorithm generates ranked ticket recommendations by combining explicit user preferences with latent factors derived from historical orders. This may approach to ensures that the recommendation process remains adaptive and responsive to evolving user needs and preferences.

Extensive experiments are conducted on real-world datasets to validate the effectiveness of the proposed methodology. By evaluating the accuracy and relevance of the generated recommendations, the methodology's performance is assessed, and insights are gained for potential improvements. Through these comprehensive steps, the methodology aims to overcome the challenges posed by the cold-start problem and provide users with personalized flight ticket recommendations that align closely with their preferences and travel behaviours.

CHAPTER-3

MATHEMATICAL REPRESENTATION

3. MATHEMATICAL REPRESENTATION & TABLES

Algorithm 1: User Model Construction

Input:

- Orders DataFrame
- Features list

Output:

- Term Frequency (TF) Distribution for each feature
- Weight (W) for each feature

Detailed Steps:

- Iterate through each order in the Orders DataFrame.
- For each order, calculate the term frequency (TF) for each feature.
- Calculate the weight (W) for each feature based on the TF distribution.

Mathematical Representation:

Let:

- D be the TF distribution dictionary for each feature.
- W be the weight dictionary for each feature.

The TF distribution for each feature can be represented as:

$$D = \{f1\colon \{t_{f1,}1\colon n_{f1,}1, t_{f1,}2\colon n_{f1,}2, \ldots\}, f2\colon \{\, t_{f2,}1\colon n_{f2,}1, t_{f2,}2\colon n_{f2,}2, \ldots\}$$

The weight for each feature can be calculated as:

$$W = \{f1\colon \log(\mathcal{N}) - \frac{\sum_{i=1}^{m} n_{f1}, i}{m}, f2\colon \log(\mathcal{N}) - \frac{\sum_{i=1}^{m} n_{f2}, i}{m}, \ldots.\}$$

Where:

- N is the total number of unique values for the feature.
- m is the total number of orders.

Example Table:

Feature	Time Frequency Distribution	Weight (W)
Airline	{'Airline A': 50, 'Airline B': 30, ...}	0.234
Class	{'Economy': 80, 'Business': 20, ...}	0.378
Price	{100: 25, 200: 35, ...}	0.453

Algorithm 2: Flight Ticket Recommendation

Input:

- Orders DataFrame
- Features list
- Candidate tickets DataFrame

Output:

- Recommended tickets with their scores

Detailed Steps:

- Construct the user model using Algorithm 1.
- Calculate the recommendation score for each candidate ticket based on the user model.
- Sort the candidate tickets based on their recommendation scores.

Mathematical Representation:

Let:

- G_t be the recommendation score for ticket.
- D be the TF distribution dictionary for each feature.
- W be the weight dictionary for each feature.

The recommendation score for ticket calculated as:

$$G_t = \sum_{i=1}^{n} \frac{D_{fi}(t_{fi})}{\sum_{j=1}^{m} D_{fi}(t_{fi,}j)} \times W_{fi}$$

Where:

- n is the number of features.
- m is the total number of unique values of the features.

Ticket ID	Recommendation Score
1	0.874
2	0.765
3	0.653
4	0.521

Algorithm 3: Route Partition and Distribution Extraction

This algorithm partitions the dataset based on routes and extracts the distribution of features for each route.

Detailed Explanation:

1. **Route Partitioning:**
   - The algorithm filters the dataset based on each unique route, splitting it into separate datasets for each route.
2. **Feature Distribution Extraction:**
   - For each route, the algorithm extracts the distribution of features (e.g., Airline, Class, Price) from the corresponding dataset.
   - The distribution represents the frequency of each feature value within the dataset for the route.

Mathematical Representation:

Let's denote D as the dictionary containing feature distributions for each route. The distribution of features for route r is represented as D_r.

Example Table:

Route	Feature Distribution
Delhi-Mumbai	{‘Airline A’: 50, ‘Airline B’: 30, ...}
Mumbai-Delhi	{‘Airline X’: 40, ‘Airline Y’: 25, ...}
………	………

Algorithm 4: User Model Enhancement Based on Social Relationships

Input:

- Historical orders for the route (O)
- Set of ticket features (F)
- Overall feature distribution (D0)
- Optimal route (R0)
- Dataset

Output:

- Enhanced user model

Detailed Steps:

- Retrieve related users based on order-shared and flight-shared relationships for the current user.
- Determine the optimal route for user model enhancement.
- Calculate the enhanced user model based on the closeness of related users and the optimal route.

Mathematical Representation:

Let:

D be the TF distribution dictionary for each feature.

M be the enhanced model dictionary.

$$M = \{f1\colon \{t_{f1,}1\colon v_{f1,}1, t_{f1,}2\colon v_{f1,}2, \dots\}, f2\colon \{t_{f2,}1\colon v_{f2,}1, t_{f2,}2\colon v_{f2,}2, \dots\}, \dots\}$$

Example Table:

User ID	Related Users
1	[234567, 345678, 456789]
2	[123456, 345678]
3	[123456, 234567, 456789]

Algorithm 5: Combine Explicit and Latent Factors for Ticket Recommendation

Input:

- Explicit factors (Dmix)
- Latent factors (D0_rb)
- Weight for recommended route (weight_recommended)
- Weight for optimal route (weight_optimal)

Output:

- Combined model

Detailed Steps:

- Combine the explicit and latent factors linearly based on the weights for the recommended and optimal routes.

Mathematical Representation:

Let:

D_{mix} be the explicit factors for the recommended route.

D_{0_rb} be the latent factors for the optimal route.

$\Omega_{recommended}$ be the weight for the recommended route.

$\Omega_{optimal}$ be the weight for the optimal route.

M be the combined model dictionary.

The combined model can be represented as:

$$M = \{f1: \{t_{f1,}1: v_{f1,}1, t_{f1,}2: v_{f1,}2, \ldots\}, f2: \{t_{f2,}1: v_{f2,}1, t_{f2,}2: v_{f2,}2, \ldots\}, \ldots\}$$

CHAPTER-4

ALGORITHMS & RESULTS

4. ALGORITHMS & RESULTS

Algorithm 1: User Model Construction

Input: O: historical orders of the route,

F: the set of ticket features.

Output: User ticket choice information model D.

```
Function ModelConstruction{O,F}
D ← φ;
for t ∈ O do
    for f ∈ F do
        D[f][tf]+ = 1;
    end for
end for
W ← φ;
for f ∈ F do
    W[f] ← P ln |f| - S(f) / (ln |f| - S(f));
end for
return D, W.
End Function
```

```
Enter the path to the CSV file: /content/drive/MyDrive/passenger_dataset (1).csv
Enter passenger ID for recommendations: 490
Enter source for recommendations: Kolkata
Enter destination for recommendations: Bangalore
Top 5 Recommended Tickets for Passenger ID 490 on Route Kolkata to Bangalore:
Airline: Air_India
Class: Economy
Flight: KB364
Price: 1922

Airline: AirAsia
Class: First
Flight: KB284
Price: 1046

Airline: SpiceJet
Class: Business
Flight: KB470
Price: 5724

Airline: Vistara
Class: First
Flight: KB354
Price: 4513

Airline: Indigo
Class: Economy
Flight: KB586
Price: 1946
```

Algorithm 2: Flight Ticket Recommendation Algorithm

Input: O: historical orders of the route,

F: the feature set of ticket,

C: the candidate ticket list.

Output: A list of ranked tickets R.

```
D, W ← ModelConstruction(O,F);
R ← φ;
for t ∈ C do
    Gt ← 0;
    for f ∈ F do
        Gt+ = D[f][tf] Sum(D[f]) × W[f];
    end for
    Append Gt to R;
end for
Sort R by descending;
return R.
```

```
Enter the path to the CSV file: /content/drive/MyDrive/passenger_dataset (1).csv
Enter passenger ID for recommendations: 490
Enter source for recommendations: Kolkata
Enter destination for recommendations: Bangalore
Top 5 Recommended Tickets for Passenger ID 490 on Route Kolkata to Bangalore:
Airline: Air_India
Class: Economy
Flight: KB364
Price: 1922
Gt Score: 0

Airline: AirAsia
Class: First
Flight: KB284
Price: 1046
Gt Score: -0.15342640972002736

Airline: SpiceJet
Class: Business
Flight: KB470
Price: 5724
Gt Score: -0.15342640972002736

Airline: Vistara
Class: First
Flight: KB354
Price: 4513
Gt Score: -0.15342640972002736

Airline: Indigo
Class: Economy
Flight: KB586
Price: 1946
Gt Score: -0.15342640972002736
```

Algorithm 3: User Model Transfer from Similar Routes

Input: O: the historical orders of all the routes,

F: the set of ticket features,

ra: the recommended route,

M: the overall feature distribution over different routes.

Output: The Enhanced User Model Dmix.

1: P ← routePartition(O, ra);

2: for r ∈ P do

3: D[r] ← extractDistribution(Or, F);

4: W[r] ← getWeight(Or, F);

5: end for

6: rb ← getOptimalRoute(P)

7: D0[rb] ← D[rb]/M[rb]

8: Dmix = Normalize(|Ora| × D[ra] + |Orb| × D0[rb])

9: return Dmix.

```
Number of unique routes: 21

Bangalore to Chennai: 507
Hyderabad to Delhi: 500
Delhi to Chennai: 491
Kolkata to Hyderabad: 489
Kolkata to Bangalore: 487
Kolkata to Mumbai: 487
Mumbai to Hyderabad: 485
Bangalore to Hyderabad: 484
Kolkata to Chennai: 484
Mumbai to Chennai: 480
Mumbai to Kolkata: 478
Delhi to Bangalore: 473
Bangalore to Kolkata: 473
Mumbai to Bangalore: 468
Delhi to Mumbai: 464
Mumbai to Delhi: 464
Kolkata to Delhi: 462
Bangalore to Delhi: 460
Delhi to Hyderabad: 459
Bangalore to Mumbai: 453
Delhi to Kolkata: 452
```

```
Enter recommended route: Kolkata-Mumbai
Enter passenger ID: 37
Top 10 Recommended Flight Details:
      airline  passenger_id flight   source destination     class  price
52   GO_FIRST           271  KM812  Kolkata      Mumbai   Economy   2988
19     Indigo           489  KM801  Kolkata      Mumbai  Business   2098
31    AirAsia           312  KM834  Kolkata      Mumbai   Economy   7391
69    Vistara           923  KM521  Kolkata      Mumbai  Business   1882
36   GO_FIRST            93  KM158  Kolkata      Mumbai     First   6667
317  GO_FIRST           849  KM691  Kolkata      Mumbai   Economy   8515
```

Algorithm 4: User Model Enhancement Based on Social Relationships

Input: O: the historical orders of the route,

F: the set of ticket features.

Output: social relationship enhanced model Damp.

```
U ← getRelatedUsers(ua);
rank_list ← φ;
for u ∈ U do
    rank_list.append(Rela(ua,u));
end for
ub ← Max(rank_list);
rb ← getOptimalRoute(ub);
D0[rb] ← D[rb]/M[rb];
Damp = Normalize(D[ra] + D0[rb]);
return Damp.
```

```
Unique Routes:
Mumbai-Bangalore
Kolkata-Chennai
Kolkata-Hyderabad
Delhi-Chennai
Kolkata-Bangalore
Bangalore-Delhi
Kolkata-Mumbai
Mumbai-Delhi
Bangalore-Mumbai
Bangalore-Chennai
Delhi-Hyderabad
Hyderabad-Delhi
Delhi-Bangalore
Mumbai-Kolkata
Delhi-Kolkata
Mumbai-Hyderabad
Mumbai-Chennai
Bangalore-Kolkata
Bangalore-Hyderabad
Delhi-Mumbai
Kolkata-Delhi
```

```
Route Prices:
                      Lowest Price  Highest Price
source    destination
Bangalore Chennai             1007           9999
          Delhi               1007           9997
          Hyderabad           1023           9982
          Kolkata             1042           9996
          Mumbai              1028           9998
Delhi     Bangalore           1126           9992
          Chennai             1009           9993
          Hyderabad           1008           9985
          Kolkata             1017           9994
          Mumbai              1008           9990
Hyderabad Delhi               1004           9968
Kolkata   Bangalore           1039           9979
          Chennai             1007           9902
          Delhi               1046           9990
          Hyderabad           1033           9963
          Mumbai              1095           9968
Mumbai    Bangalore           1030           9982
          Chennai             1012           9983
          Delhi               1008           9989
          Hyderabad           1014           9990
          Kolkata             1006           9997

Unique Airlines:
SpiceJet
AirAsia
Air_India
GO_FIRST
Vistara
Indigo

Unique Flights:
['MB643' 'KC150' 'KC365' ... 'MC584' 'HD596' 'MH967']

Prices:
flight
BC101          [9294]
BC105    [9263, 1450]
BC107          [1067]
BC114          [2092]
BC116          [7602]
              ...
MK991          [8521]
MK992          [8730]
MK993          [6818]
MK998          [1257]
MK999          [8921]
Name: price, Length: 7759, dtype: object

Flight Times:
flight
BC101     6.353288
BC105     7.560455
BC107     6.039344
BC114     8.000236
BC116     9.417407
            ...
MK991     6.504601
MK992     5.132528
MK993     3.635123
MK998    10.482211
MK999     8.370663
Name: flight_duration, Length: 7759, dtype: float64
```

Algorithm 5: Ticket Recommendation by Combining Explicit and Latent Factors

Input: O: the historical orders of the route,

F: the set of ticket features,

C: the search result list.

Output: A ranked tickets list R.

1: $Wu \leftarrow 0$;

2: $Mti \leftarrow 0$;

3: for $o \in O$ do

4:     $Co \leftarrow getSearchResult(o)$;

5:     for $to \in Co.sample()$ do

6:         $Wu = Wu + \alpha(\partial logP(ti,tj)/Wu - \lambda \times Wu)$;

7:         $Mti = Mti + \alpha(\partial logP(ti,tj)/Mti - \lambda \times Mti)$;

8:     end for

9:     If reach iteration limit, break;

10: end for

11: for $t \in C$ do

12:     $Rt = R(KTu \times Ft) + R(\varphi Tu \times \theta t)$;

13:     Append Rt to R;

14: end for

15: Sort R by ascending;

16: return R.

```
Enter historical routes separated by commas: Mumbai-Hyderabad,Bangalore-Chennai,Delhi-Kolkata
Enter ticket features separated by commas: Flight,Class,Price,Flight Duration,Session,Route
Enter learning rate (alpha): 0.1
Enter regularization parameter (lambda): 0.2

Ranked Tickets: [13155.146994555076, 13181.352466655782, 13194.455202706136, 13194.455202706136
```

CHAPTER-5

GRAPHICAL REPRESENTATION

5. GRAPHICAL REPRESENTATION

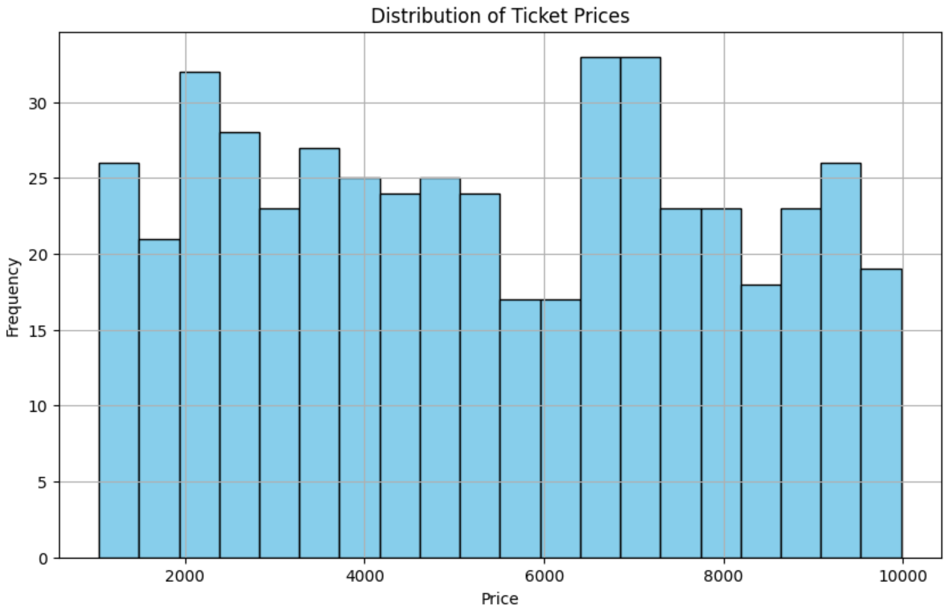

Fig 1.1. The histogram illustrates the distribution of ticket prices for the specified route, providing insight into the frequency of different price ranges.

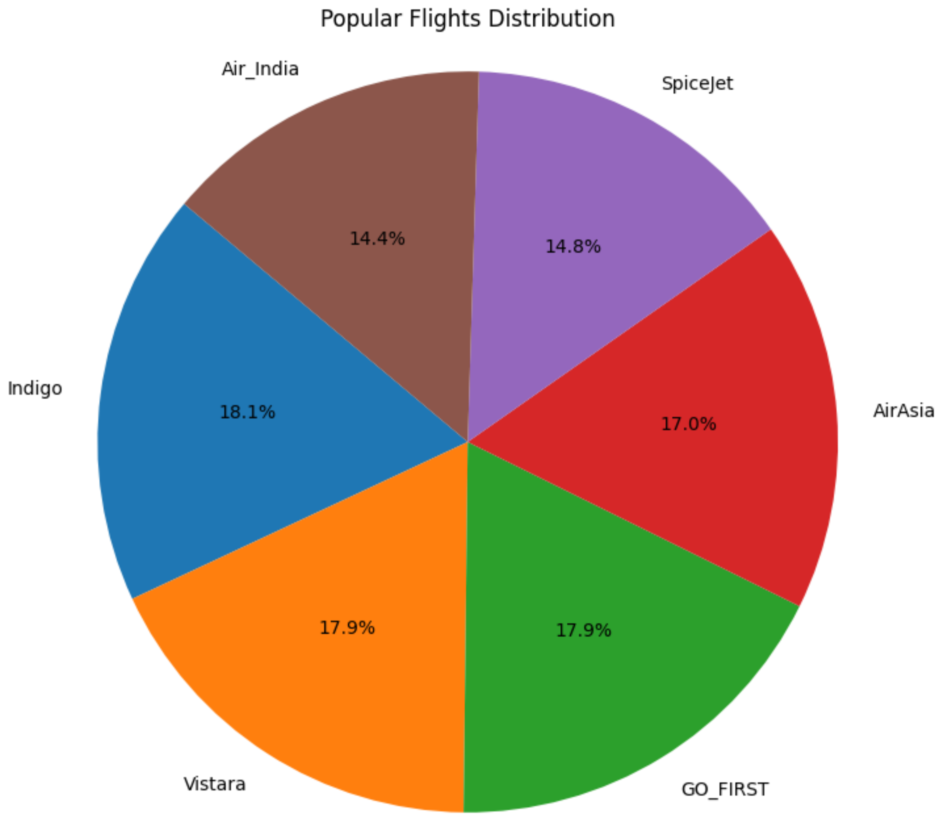

Fig 1.2. The pie chart displays the distribution of popular flights for the specified route, showcasing the percentage composition of each airline among the available flight option.

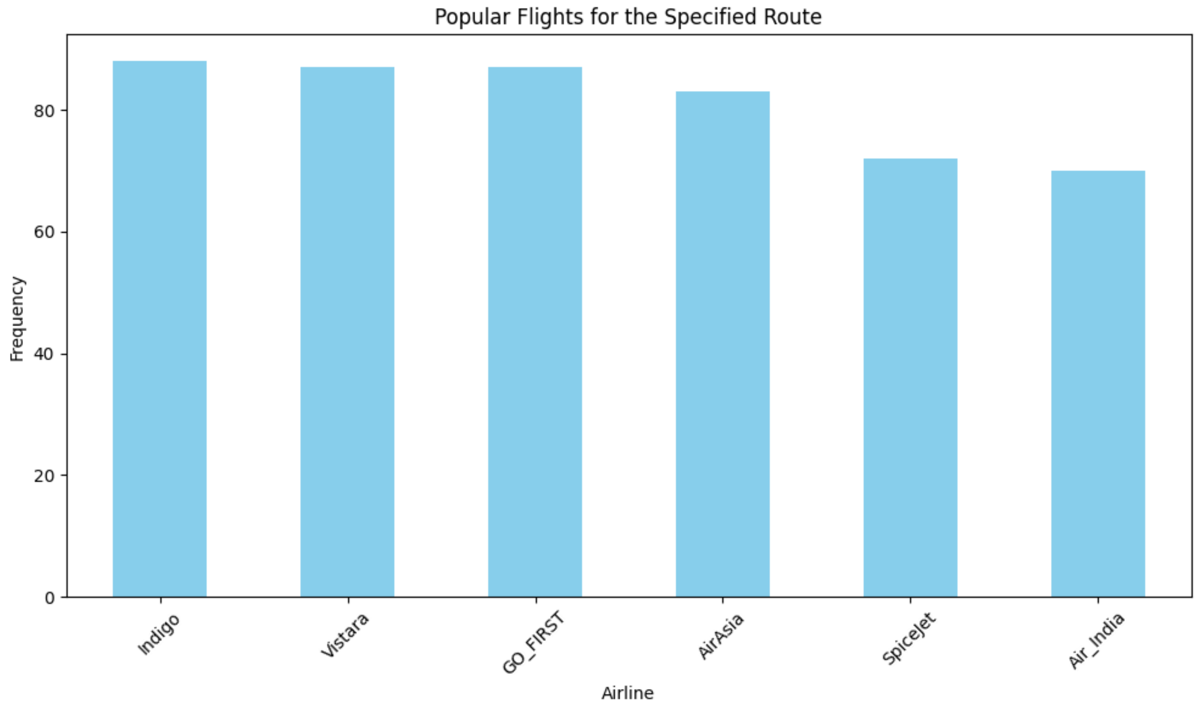

Fig 2.1. The bar graph visualizes the frequency distribution of airlines for the specified route, providing insight into the popularity of different airlines among available flights.

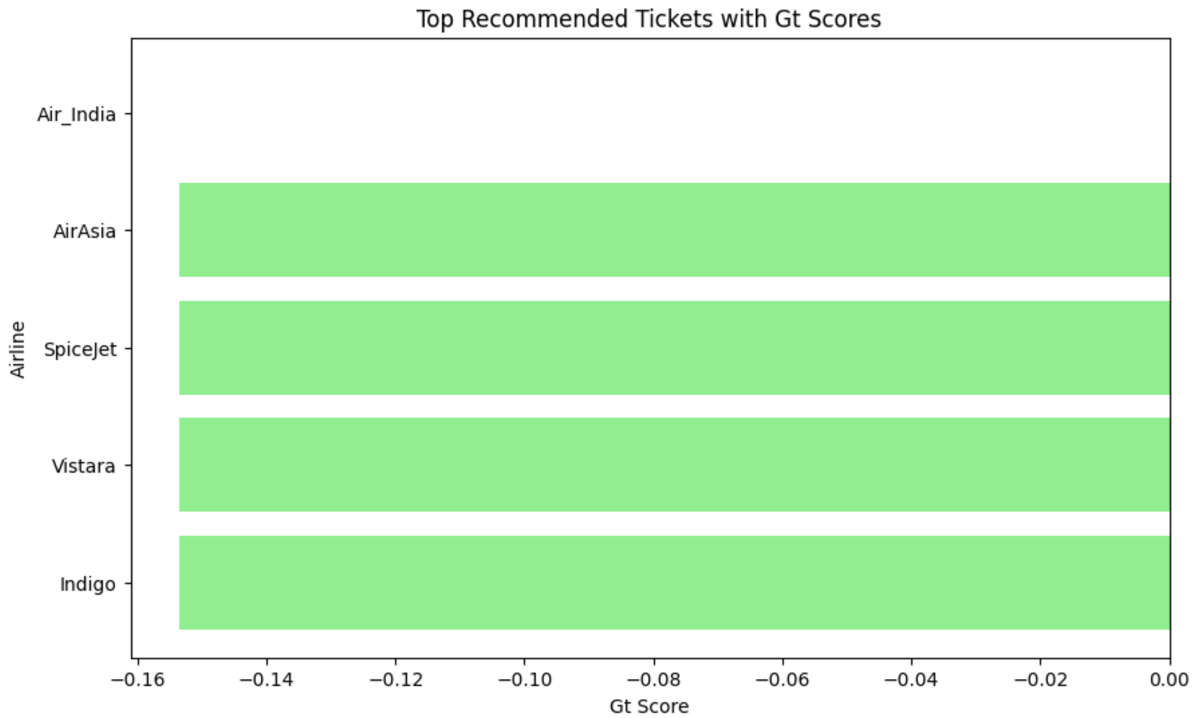

Fig 2.2. The horizontal bar graph displays the Gt scores for the top recommended tickets, highlighting the suitability of each airline based on personalized recommendation scores.

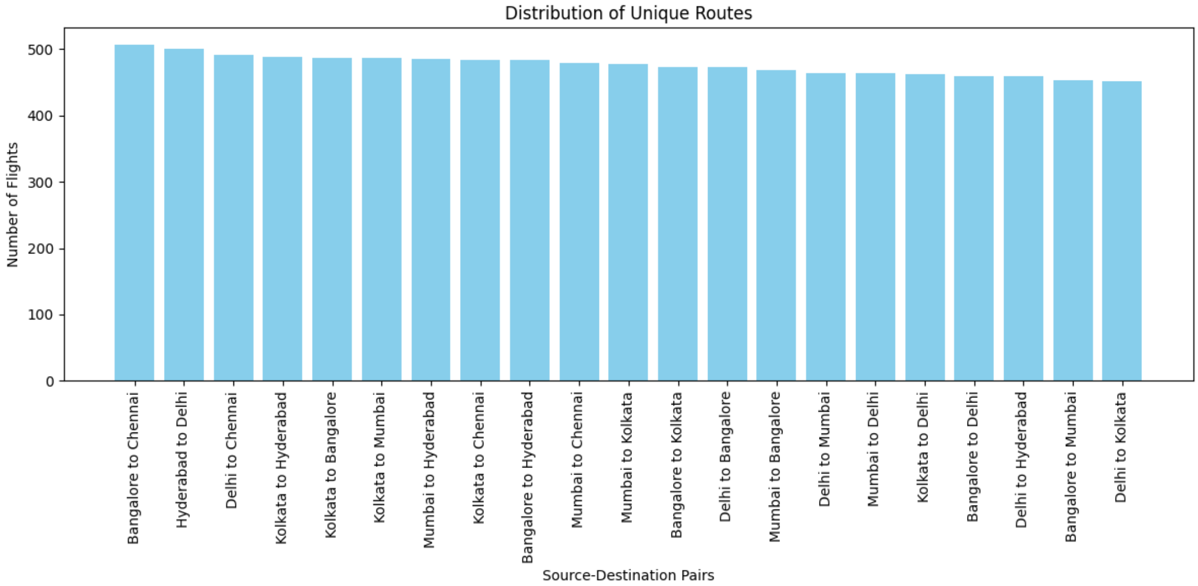

Fig 3.1. Indicates route diversity, visually illustrates the distribution of flights across these routes, revealing route popularity.

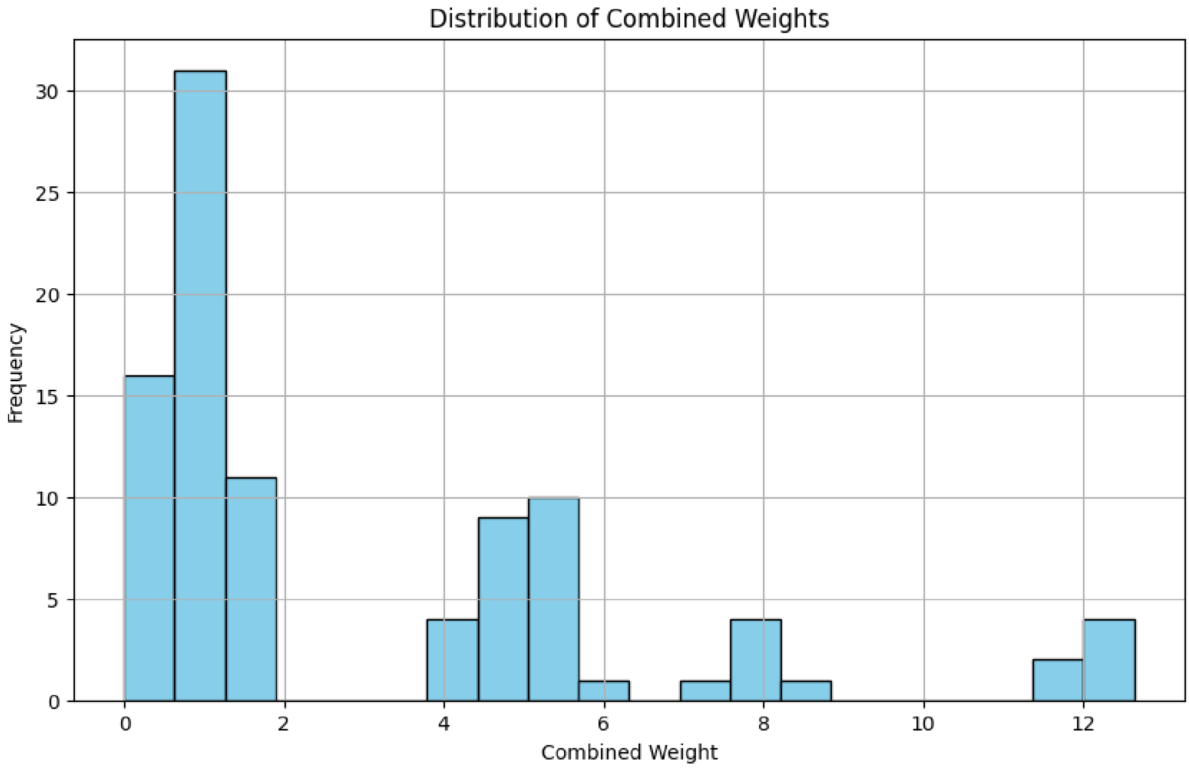

Fig 3.2. The histogram visualizes the distribution of combined weights, indicating how recommendations align with passenger preferences for the recommended route.

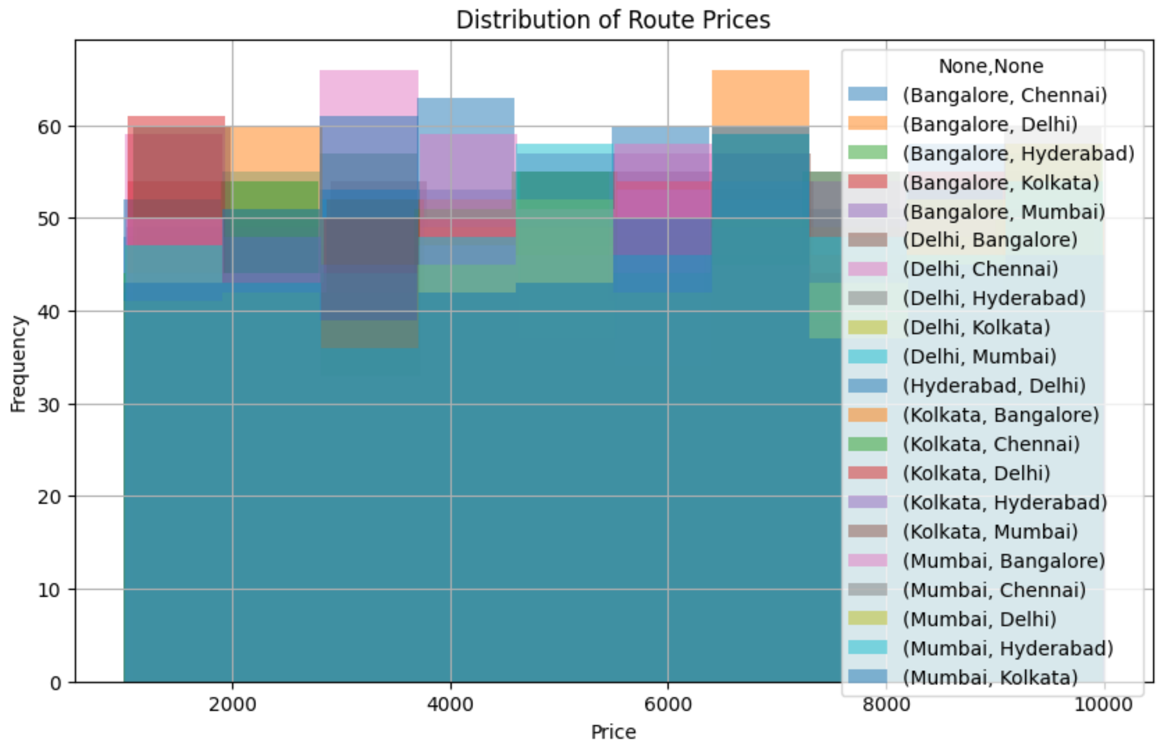

Fig 4.1. The histogram illustrates the distribution of route prices, showing the frequency of prices across different routes.

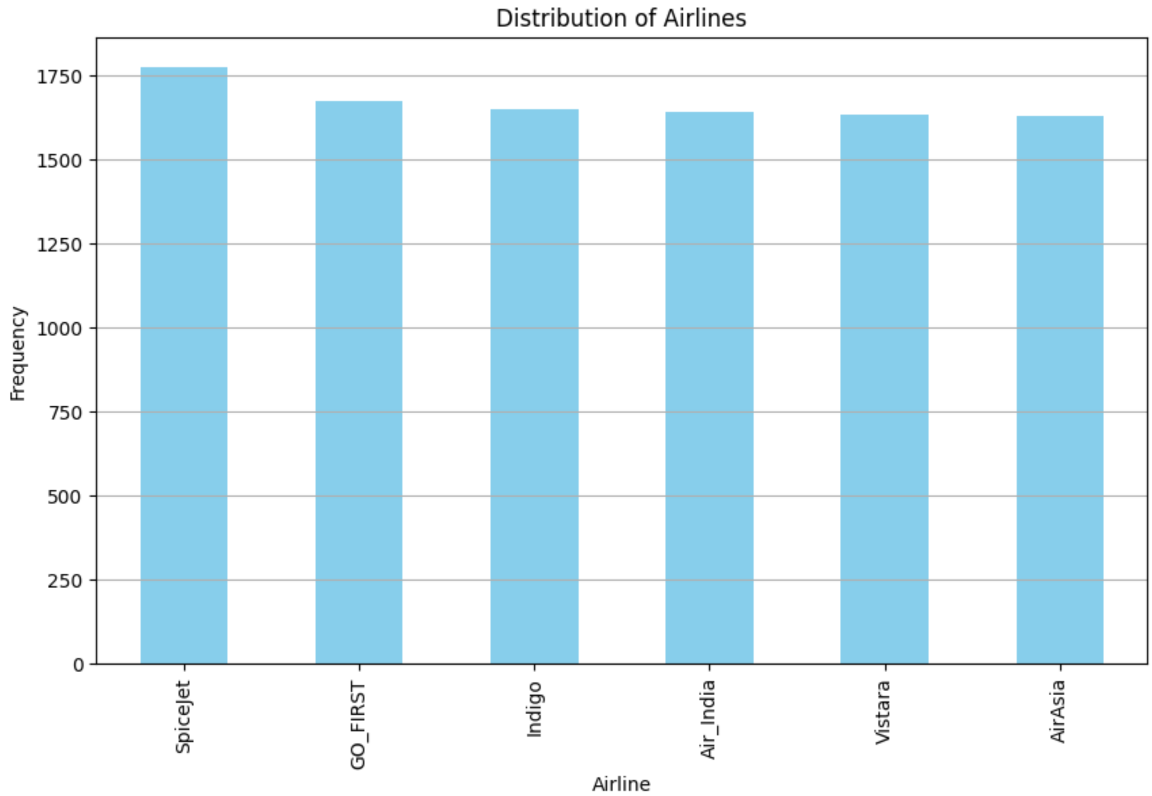

Fig 4.2. The bar plot visualizes the distribution of airlines, indicating the frequency of each airline in the dataset.

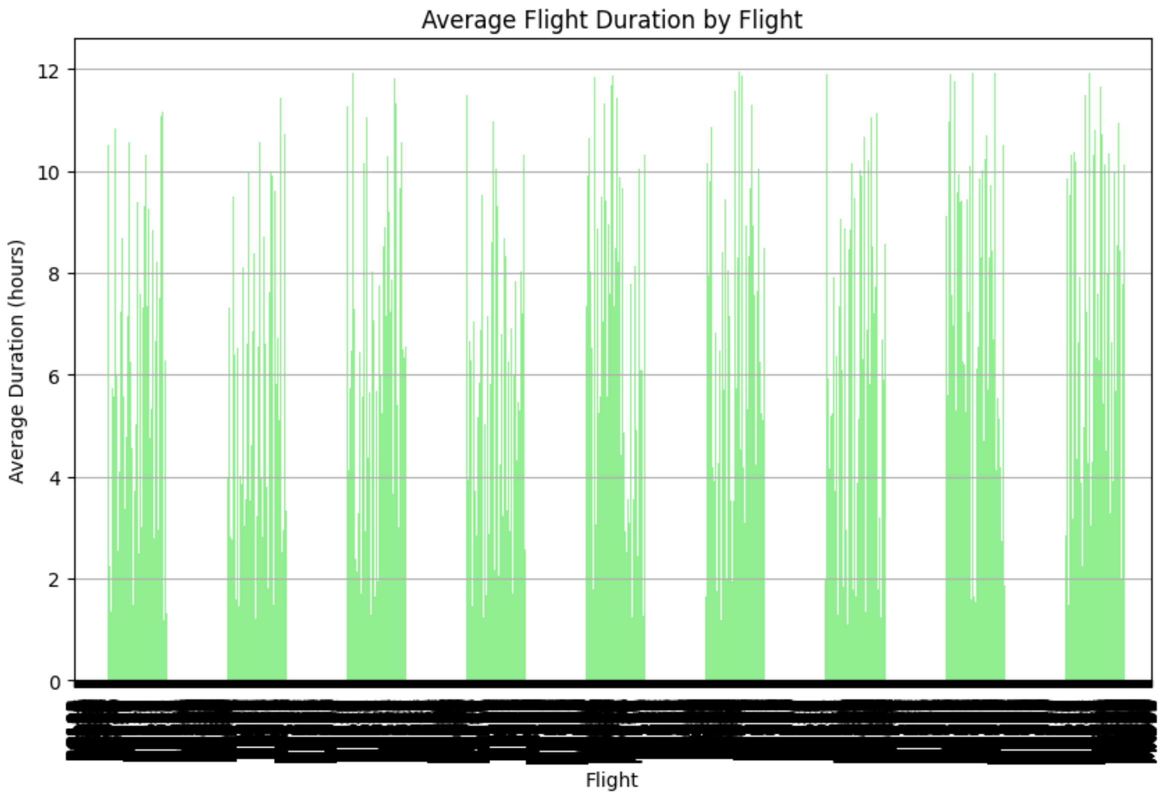

Fig 4.3. The bar plot displays the average flight duration for each flight, providing insight into the duration of different flights.

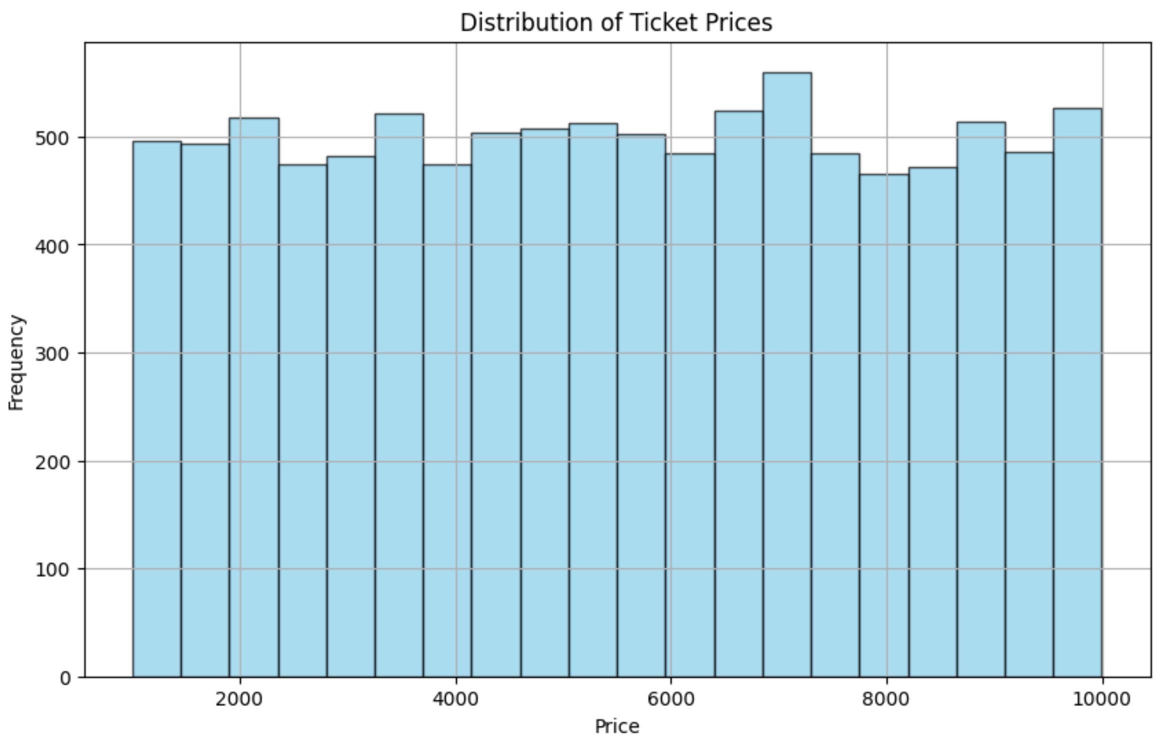

Fig 5.1. The histogram depicts the distribution of ticket prices from the dataset, showcasing the frequency of prices across different tickets.

CHAPTER-6

CONCLUSION

6. CONCLUSION

In conclusion, the innovative strategies to enhance the flight ticket recommendations, particularly focusing on addressing the cold-start problem prevalent in online travel platforms. Throughout a comprehensive exploration of user behaviours and preferences, coupled with advanced algorithmic approaches, using Latent Factor Modelling, we have devised personalized recommendation methods tailored specifically for flight tickets. By leveraging route similarity, social relationships among passengers, and a combination of explicit and latent factors, our proposed approaches aim to overcome the limitations posed by addressing the cold-start problem, ultimately leading to more accurate and relevant ticket recommendations. Experimental validation conducted on real-world datasets showcases the effectiveness and practical applicability of our methodologies. We believe that these strategies offer valuable insights and solutions to improve the user experience in flight ticket booking platforms, ultimately fostering higher satisfaction and engagement among users. As the online travel industry continues to evolve, our work sets a foundation for future and development aimed at further enhancing recommendation systems and addressing emerging challenges in the field.