AIRLINE REVIEW CLASSIFICATION

1.INTRODUCTION

1.1 Project Overview

First, we will develop a machine learning model to classify airline reviews as positive, negative, or neutral and then we will utilize a dataset of airline reviews containing text and sentiment labels. We will do employ preprocessing techniques, feature extraction methods, and machine learning algorithms. At last, we will evaluate the model's performance using accuracy, precision, recall, and F1-score metrics and achieve high classification accuracy and gain insights into factors influencing airline review sentiment.

1.2 Purpose

Airline review classification is the process of automatically classifying airline reviews as positive, negative, or neutral using machine learning algorithms. This process has several important purposes:

1. Understanding customer sentiment: Airline reviews provide a valuable source of feedback for airlines, allowing them to understand customer satisfaction levels and identify areas for improvement. By classifying reviews, airlines can quickly gauge the overall sentiment of their customers and gain insights into specific aspects of their service, such as check-in, boarding, in-flight experience, and baggage handling.

2. Improving customer service: Airline review classification can be used to identify common customer complaints and areas where the airline can improve its service. For instance, if a significant number of reviews mention delays or cancellations, the airline can investigate the root causes of these issues and implement measures to reduce their occurrence. Similarly, if reviews consistently highlight issues with in-flight amenities or customer service interactions, the airline can take steps to enhance these aspects of the travel experience.

3. Enhancing reputation management: Positive airline reviews can significantly boost an airline's reputation and attract new customers. By analyzing and addressing negative reviews, airlines can demonstrate their commitment to customer satisfaction and proactively manage their online reputation. Positive sentiment analysis can also be used to highlight areas where the airline is excelling and share these positive reviews with potential customers to reinforce their brand image.

4.Identifying trends and patterns: Airline review classification can help airlines identify trends and patterns in customer sentiment over time. This information can be used to track the impact of specific initiatives or changes in service offerings and gain valuable insights into customer preferences and expectations.

2.LITERATURE SURVEY

2.1 Existing problem

- Fine-grained Sentiment Analysis:
    - Problem: Traditional sentiment analysis may provide a binary classification of positive or negative sentiments, but airline reviews often contain more nuanced opinions and sentiments.
- Aspect-based Sentiment Analysis:
    - Problem: Airline reviews cover various aspects such as customer service, in-flight experience, punctuality, and more. Existing models may struggle to identify sentiments specific to each aspect.
- Handling Multimodal Data:
    - Problem: Reviews can include not only textual content but also images or videos, each contributing to the overall sentiment.
- Cross-domain Generalization:
    - Problem: Models trained on reviews from one airline may not generalize well to other airlines due to differences in service quality, customer expectations, or review writing styles.
- Temporal Dynamics:
    - Problem: Sentiments towards airlines can change over time due to factors like seasonal variations, marketing campaigns, or industry-wide events.

2.2 References

https://ieeexplore.ieee.org/document/9667818

https://www.sciencedirect.com/science/article/pii/S1877050923002211

https://www.mdpi.com/2078-2489/12/2/78

https://globalilluminators.org/wp-content/uploads/2015/12/MISG-15-198.pdf

2.3 Problem Statement Definition

The enormous growth of internet evaluations in today's travel environment has made them an invaluable tool for travellers and airlines alike. Nevertheless, there are inherent problems with the sentiment analysis approaches now used in the aviation sector, which restricts their ability to yield insightful results. The primary problem is the simple binary categorization of feelings, which is unable to adequately represent the complex and varied character of passenger input. This shortcoming makes it more difficult for airlines to understand the range of opinions that travellers have to offer and prevents prospective passengers from getting a clear picture of the level of service that airlines provide.

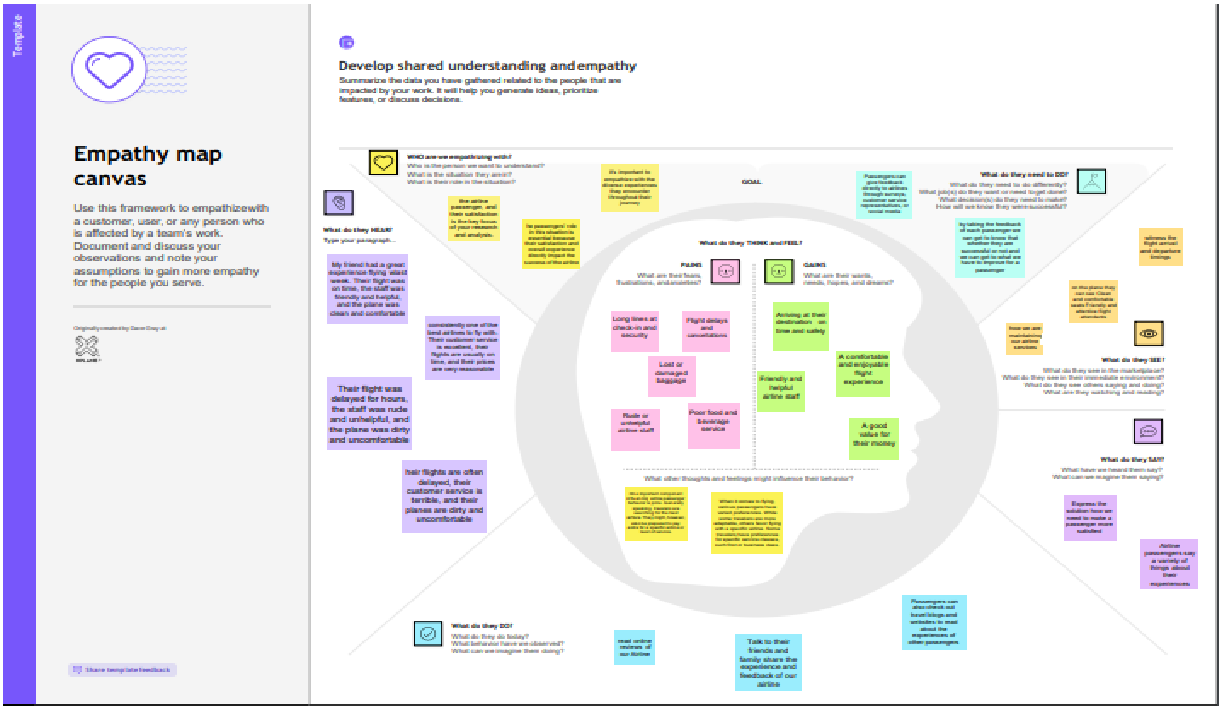
Moreover, current sentiment analysis methods are not detailed enough to analyse evaluations according to certain features of flying, such customer service, in-flight amenities, and timeliness. The lack of aspect-based sentiment analysis leads to a superficial portrayal of passenger feelings, which reduces the usefulness of the acquired insights for airlines looking to make specific service changes.

In the modern day, there is a trend towards multimodal content, and users are adding visual components to their evaluations, such photos and videos. Nevertheless, most sentiment analysis algorithms in use today concentrate on textual content, ignoring the abundance of information that may be expressed through visual media. This restriction makes it difficult to fully understand feelings, especially in a culture where visual communication is becoming more and more common.

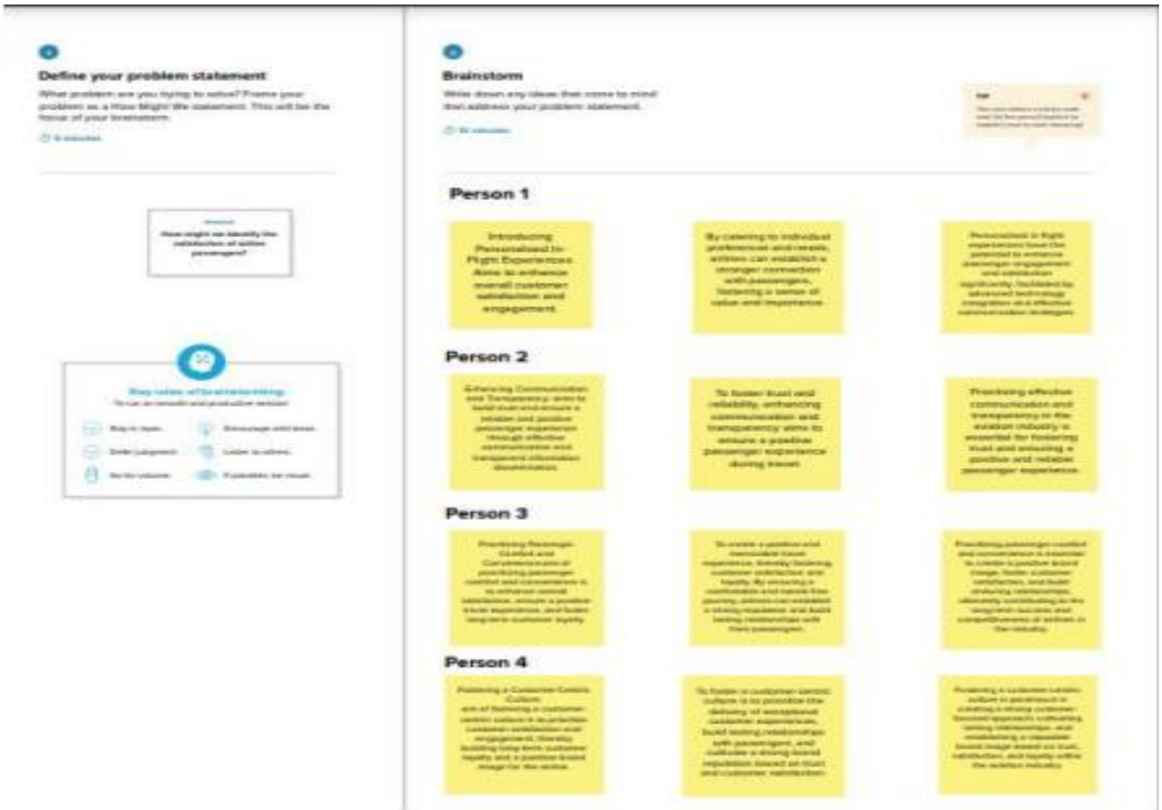
A paradigm change in the way airline review classification is approached is necessary to address these issues. It is imperative to have a more complex and nuanced sentiment analysis framework that can handle the diverse nature of passenger attitudes, integrate multimodal data, adjust to temporal variations, and take individual subjectivity into consideration. We can only fully realise airline reviews' potential as a useful tool for travellers and the aviation sector by overcoming these obstacles.

3.IDEATION PHASE AND PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation and Brainstorming



Group Ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

Introducing Personalized In-Flight Experiences: Aims to enhance overall customer satisfaction and engagement.

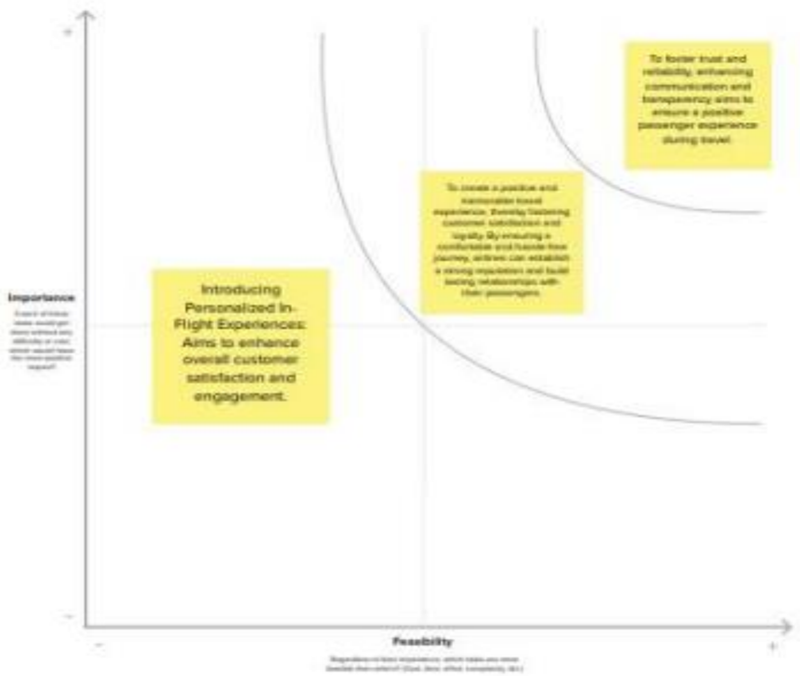
To foster trust and reliability, enhancing communication and transparency aims to ensure a positive passenger experience during travel.

To create a positive and memorable travel experience, thereby fostering customer satisfaction and loyalty. By ensuring a comfortable and hassle-free journey, airlines can establish a strong reputation and build lasting relationships with their passengers.

Fostering a customer-centric culture is paramount in creating a strong customer-focused approach, cultivating lasting relationships, and establishing a reputable brand image based on trust, satisfaction, and loyalty within the aviation industry.

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.



4.REQUIREMENT ANALYSIS

4.1 Functional requirements

Authorization and Authentication of Users:

It should be necessary for users to log in before they can submit or categorise reviews.

To manage the system, admin roles need to be established.

Examine the submission:

Reviews from users should be able to include information about the airline, the date of the flight, and the content of the review.

Emotional Analysis:

Reviewers can be categorised as neutral, negative, or positive by using sentiment analysis.

Assessment Method:

Permit consumers to rate various characteristics such as timeliness, cleanliness, and service with numbers.

Examine the classification:

Reviews can be automatically categorised according to pre-established standards, such general satisfaction or service areas.

Looking for and Sorting:

Give consumers the ability to search for reviews using criteria like airline, date, rating, or other pertinent information.

Analytics and Reporting:

For every airline, create reports on general sentiment trends.

Provide statistics on the good and negative elements that are frequently brought up in evaluations.

Feedback System:

Permit people to comment on how accurate the sentiment analysis and classification are.

4.2 Non-Functional requirements

Operation:

Even during periods of high utilisation, the system ought to react quickly to user interactions.

Scalability:

It should be possible for the system to grow in size to accommodate more users and reviews.

Dependability:

Make sure there is little downtime for updates or maintenance on the system.

Protection:

Put strong security measures in place to safeguard user information and guarantee safe system access.

Utilisation:

It should be simple for users to post reviews and use the system thanks to an intuitive user interface.

Adaptability:

To improve user accessibility, make sure the system is interoperable with a range of devices and browsers.

Data Security:

Respect user privacy when managing personal data and follow data protection laws.

Reliability:

The system must be simple to maintain, enabling upgrades and enhancements without posing any problems.

5.PROJECT DESIGN

5.1 Data Flow Diagrams and User Stories

Data Collection:

- Collect airline reviews from online websites or datasets.
- Clean and label the data.
- Split the data into training and test sets.

Data Pre-processing:

- Clean the text by removing extra spaces, punctuation, and special characters.
- Convert all letters to lowercase for consistency.
- Split the text into individual words (tokenization) for analysis.

Model Evaluation:

- Split the data into training and test sets. The training set is used to train the model and the test set is used to evaluate the model's performance.
- Train the model on the training set.
- Make predictions on the test set.
- Calculate the evaluation metrics.
- Analyze the results and make necessary adjustments to the model.

Model Deployment:

- Choose a deployment platform, such as a web server or cloud service.
- Prepare your model for deployment, including packaging it into a deployable format.

• Set up an interface for users to input reviews, and use the model to classify them in real-time.

User Interaction:

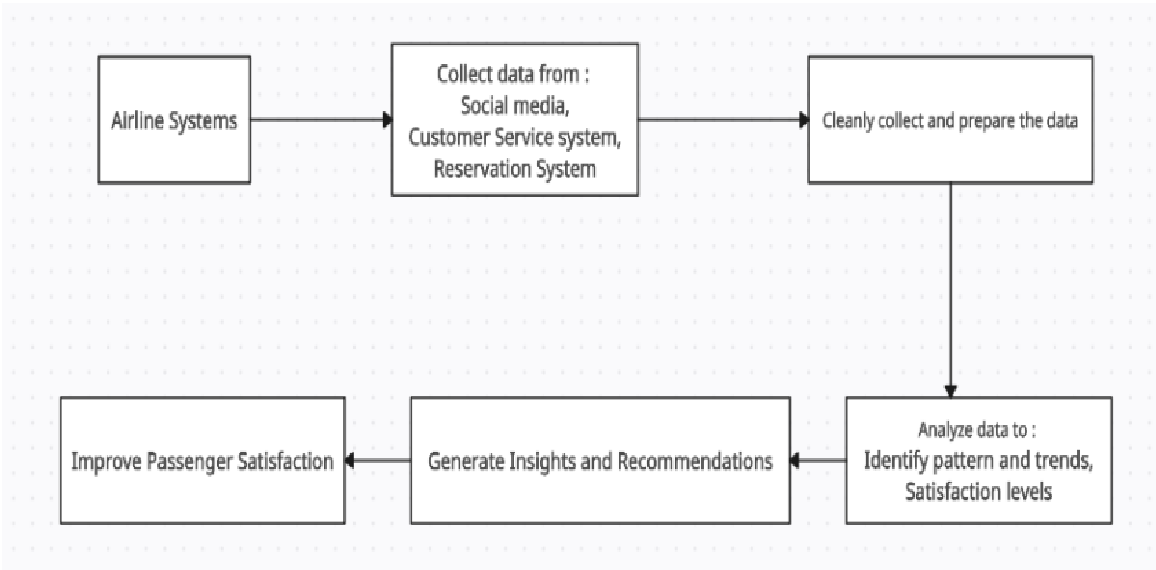
• Allow users to enter their airline reviews.

• Trigger the review classification process.

• Display the classification (positive/negative/neutral).

• Include messages for invalid inputs or errors.

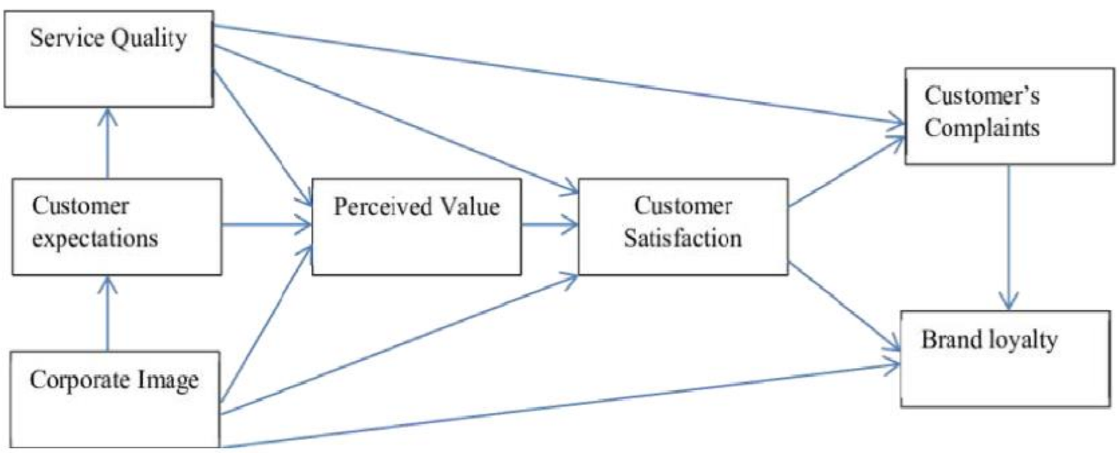
Data Flow Diagram



User Stories

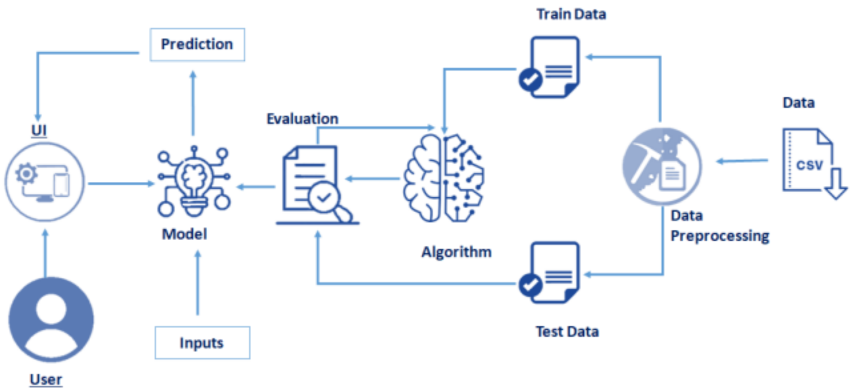
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Traveler	Classify airline reviews as positive, negative, or neutral	USN - 1	As a traveler, I want to be able to classify airline reviews as positive, negative, or neutral so that I can make informed decisions about which airlines to choose.	The system shall be able to classify airline reviews as positive, negative, or neutral with an accuracy of at least 90%.	High	Sprint-1
Traveler	View overall sentiment of airline reviews	USN - 2	As a traveler, I want to be able to see the overall sentiment of airline reviews so that I can get a quick overview of how other travelers feel about a particular airline.	The system shall be able to calculate the overall sentiment of airline reviews for a given airline, with a breakdown by sentiment category	High	Sprint-1
Transportation provider	Identify satisfied and dissatisfied passengers	USN - 3	As a transportation provider, I want to be able to identify satisfied and dissatisfied passengers so that I can improve my service and customer satisfaction.	The system shall be able to identify satisfied and dissatisfied passengers with an accuracy of at least 80%.	High	Sprint-2
Airline	Identify areas where customer satisfaction is below average	USN - 4	As an airline, I want to be able to identify areas where customer satisfaction is below average so that I can take steps to improve them.	The system shall provide airlines with insights into the areas where their customer satisfaction is below average, with a breakdown by satisfaction category	High	Sprint-1
Researcher	Study the relationship between passenger satisfaction and other factors	USN - 5	As a researcher, I want to be able to study the relationship between passenger satisfaction and other factors (e.g., price, travel time, amenities) so that I can understand the drivers of passenger satisfaction and identify opportunities for improvement.	The system shall provide researchers with access to passenger satisfaction data and data on other relevant factors (e.g., price, travel time, amenities), with the ability to perform statistical analysis to study the relationship between these factors.	Medium	Sprint - 1
Transportation provider	Identify satisfied and dissatisfied passengers by travel route	USN - 6	As a transportation provider, I want to be able to identify satisfied and dissatisfied passengers by travel route so that I can identify specific areas where I need to improve my service.	The system shall allow transportation providers to identify satisfied and dissatisfied passengers by travel route, with an accuracy of at least 80%.	Medium	Sprint - 1
Transportaion provider	Identify satisfied and dissatisfied passengers by demographics (e.g., age, gender, location)	USN - 7	As a transportation provider, I want to be able to identify satisfied and dissatisfied passengers by demographics so that I can target my marketing and service improvement efforts accordingly.	The system shall allow transportation providers to identify satisfied and dissatisfied passengers by demographics (e.g., age, gender, location), with an accuracy of at least 80%.	Medium	Sprint - 2

5.2 Solution Architecture:



6.PROJECT PLANNING AND SCHEDULING

6.1 Technical Architecture



6.2 Sprint planning and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Project Setup & Infrastructure	USN-1	Set up the development environment with the required tools and frameworks to start the Airline Review Satisfaction	1	High	Karthik
Sprint-1	Development Environment	USN-2	Gather a dataset Containing the several number of reviews about a airlines for the Machine Learning Model	3	High	Karthik
Sprint-2	Data collection	USN-3	Collect the Data from the Dataset by using pandas and the collect the data info of the dataset	1	medium	Deepak
Sprint-2	Data Pre-Processing	USN-4	Pre-process the collected dataset by removing the unwanted columns and Handling the Null values and feature scaling the data and at last we have to separate the dependent and independent Variables and we have to separate the data as training data and splitting data	2	High	Lathvik
Sprint-3	Model Development	USN-5	We have to explore for classification Machine learning Algorithms and we have to select the best machine learning algorithm based on our pre-processed data	4	High	Chethan
Sprint-3	Training	USN-6	We have to fit the training data to the Classification Machine Learning Model which we have developed then our Model will get Trained	3	High	Lathvik
Sprint-4	model deployment & Integration	USN-7	deploy the trained machine learning model as an API or web service to make it accessible for airline review classification. integrate the model's API into a user- friendly web interface for users to search for the airline and receive airline review satisfaction classification results.	3	medium	Deepak
Sprint-5	Testing & quality assurance	USN-8	Now we have to use the testing data and we have to search for bugs or any failures in our trained model and we have to give best parameters and optimize the model on the user feedback. And testing results like accuracy precision score and f1 score	3	High	Chethan

6.3 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	4	5 Days	21 Oct 2023	25 Oct 2023	20	21 Oct 2023
Sprint-2	3	5 Days	26 Oct 2023	30 Oct 2023		
Sprint-3	7	5 Days	31 Oct 2023	04 Nov 2023		
Sprint-4	3	5 Days	05 Nov 2023	09 Nov 2023		
Sprint-5	3	5 Days	10 Nov 2023	14 Nov 2023		

7.CODING AND SOLUTIONS

IMPORTING THE LIBRARIES

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        from sklearn.model_selection import train_test_split
        from sklearn.preprocessing import LabelEncoder
        from sklearn.preprocessing import MinMaxScaler
        from imblearn.over_sampling import SMOTE
        from sklearn.neighbors import KNeighborsClassifier
        from sklearn.metrics import accuracy_score,classification_report
```

Here we are importing the pandas library as pd, numpy as np, matplotlib.pyplot as plt, seaborn as sns, and various modules from scikit-learn and imbalanced learn libraries for the tasks such as model selection, preprocessing, and machine learning.

Activity 1.1: Reading the dataset:

```
In [2]: data=pd.read_csv("Airline_Reviews.csv") #Reading the dataset

In [3]: data.shape #getting the no.of rows and columns
Out[3]: (23171, 20)
```

i) The data.shape command in Pandas is used to dispaly the number of rows and columns.
ii)The data.head() command is used to display the first few rows of the Pandas DataFrame named data.

```
In [4]: data.head() #reading the dataset
```

Out[4]:

	Unnamed: 0	Airline Name	Overall_Rating	Review_Title	Review Date	Verified	Review	Aircraft	Type Of Traveller	Seat Type	Route	Date Flown	Seat Comfort	Cabin Staff Service	F Beve
0	0	AB Aviation	9	"pretty decent airline"	11th November 2019	True	Moroni to Moheli. Turned out to be a pretty ...	NaN	Solo Leisure	Economy Class	Moroni to Moheli	November 2019	4.0	5.0	
1	1	AB Aviation	1	"Not a good airline"	25th June 2019	True	Moroni to Anjouan. It is a very small airline...	E120	Solo Leisure	Economy Class	Moroni to Anjouan	June 2019	2.0	2.0	
2	2	AB Aviation	1	"flight was fortunately short"	25th June 2019	True	Anjouan to Dzaoudzi. A very small airline an...	Embraer E120	Solo Leisure	Economy Class	Anjouan to Dzaoudzi	June 2019	2.0	1.0	
3	3	Adria Airways	1	"I will never fly again with Adria"	28th September 2019	False	Please do a favor yourself and do not fly wi...	NaN	Solo Leisure	Economy Class	Frankfurt to Pristina	September 2019	1.0	1.0	
4	4	Adria Airways	1	"it ruined our last days of holidays"	24th September 2019	True	Do not book a flight with this airline! My fr...	NaN	Couple Leisure	Economy Class	Sofia to Amsterdam via Ljubljana	September 2019	1.0	1.0	

iii) The data.info() command in Pandas provides a concise summary of the DataFrame.

```
In [5]: data.info() #getting the info of data set

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23171 entries, 0 to 23170
Data columns (total 20 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Unnamed: 0              23171 non-null  int64
 1   Airline Name            23171 non-null  object
 2   Overall_Rating          23171 non-null  object
 3   Review_Title            23171 non-null  object
 4   Review Date             23171 non-null  object
 5   Verified                23171 non-null  bool
 6   Review                  23171 non-null  object
 7   Aircraft                7129 non-null   object
 8   Type Of Traveller       19433 non-null  object
 9   Seat Type               22075 non-null  object
 10  Route                   19343 non-null  object
 11  Date Flown              19417 non-null  object
 12  Seat Comfort            19016 non-null  float64
 13  Cabin Staff Service     18911 non-null  float64
 14  Food & Beverages        14500 non-null  float64
 15  Ground Service          18378 non-null  float64
 16  Inflight Entertainment  10829 non-null  float64
 17  Wifi & Connectivity     5920 non-null   float64
 18  Value For Money         22105 non-null  float64
 19  Recommended             23171 non-null  object
dtypes: bool(1), float64(7), int64(1), object(11)
memory usage: 3.4+ MB
```

iv) droping the columns to get efficient model. In: df = data.drop(['Inflight Entertainment', 'Wifi & Connectivity', 'Aircraft', 'Value For Money', 'Cabin Staff Service', 'Unnamed: 0', 'Review Date', 'Review_Title', 'Review'], axis=1) The above mentioned columns are dropped from the dataset. df.head() of the modified dataset:

```
In [7]: df.head() #modified dataset
```

Out[7]:

	Airline Name	Overall_Rating	Verified	Type Of Traveller	Seat Type	Route	Date Flown	Seat Comfort	Food & Beverages	Ground Service	Recommended
0	AB Aviation	9	True	Solo Leisure	Economy Class	Moroni to Moheli	November 2019	4.0	4.0	4.0	yes
1	AB Aviation	1	True	Solo Leisure	Economy Class	Moroni to Anjouan	June 2019	2.0	1.0	1.0	no
2	AB Aviation	1	True	Solo Leisure	Economy Class	Anjouan to Dzaoudzi	June 2019	2.0	1.0	1.0	no
3	Adria Airways	1	False	Solo Leisure	Economy Class	Frankfurt to Pristina	September 2019	1.0	NaN	1.0	no
4	Adria Airways	1	True	Couple Leisure	Economy Class	Sofia to Amsterdam via Ljubljana	September 2019	1.0	1.0	1.0	no

v) df.info() command in Pandas provides a concise summary of the DataFrame.[to get info of updated dataset]

```
In [8]: df.info() #to get info of updated dataset

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23171 entries, 0 to 23170
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Airline Name       23171 non-null  object
 1   Overall_Rating     23171 non-null  object
 2   Verified           23171 non-null  bool
 3   Type Of Traveller  19433 non-null  object
 4   Seat Type          22075 non-null  object
 5   Route              19343 non-null  object
 6   Date Flown         19417 non-null  object
 7   Seat Comfort       19016 non-null  float64
 8   Food & Beverages   14500 non-null  float64
 9   Ground Service     18378 non-null  float64
 10  Recommended        23171 non-null  object
dtypes: bool(1), float64(3), object(7)
memory usage: 1.8+ MB
```

ACTIVITY 2: DATA PREPERATION

Activity 2.1: CHECKING FOR NULL VALUES AND HANDLING NULL VALUES:

CHECKING FOR NULL VALUES:

i)df.isnull().any():

The df.isnull().any() expression is used to check if there are any null (missing) values in each column of the DataFrame df.

```
In [9]: df.isnull().any() #checking for null values

Out[9]: Airline Name         False
        Overall_Rating       False
        Verified             False
        Type Of Traveller     True
        Seat Type             True
        Route                 True
        Date Flown            True
        Seat Comfort          True
        Food & Beverages      True
        Ground Service        True
        Recommended          False
        dtype: bool
```

ii) df.isnull().sum(): The df.isnull().sum() expression is used to calculate the total number of null (missing) values in each column of the DataFrame df.

```
In [10]: df.isnull().sum()  #checking for the sum of null values

Out[10]: Airline Name            0
         Overall_Rating          0
         Verified                0
         Type Of Traveller    3738
         Seat Type            1096
         Route                3828
         Date Flown           3754
         Seat Comfort         4155
         Food & Beverages     8671
         Ground Service       4793
         Recommended             0
         dtype: int64
```

By observation, looks like there are some null values, so to ensure accuracy and reliability we handle those null values.

HANDLING NULL VALUES:

iii) There are null values in some columns so we have to modify the dataset categorical values are replace with mode and numerical are replace with mean.

e.g; df["Type Of Traveller"].fillna(df["Type Of Traveller"].mode()[0], inplace=True)

```
In [11]: #There is null values in some columns so we have to modify the dataset categorical values are replace with mode and numerical
         df["Type Of Traveller"].fillna(df["Type Of Traveller"].mode()[0],inplace=True)
         df["Seat Type"].fillna(df["Seat Type"].mode()[0],inplace=True)
         df["Route"].fillna(df["Route"].mode()[0],inplace=True)
         df["Date Flown"].fillna(df["Date Flown"].mode()[0],inplace=True)
         df["Seat Comfort"].fillna(df["Seat Comfort"].mean(),inplace=True)
         df["Food & Beverages"].fillna(df["Food & Beverages"].mean(),inplace=True)
         df["Ground Service"].fillna(df["Ground Service"].mean(),inplace=True)
```

CHECKING FOR NULL VALUES:

We have to check whether the null values are present or not in the modified data. iv) df.isnull().any():

```
In [12]: df.isnull().any() #checking again there is any null values

Out[12]: Airline Name         False
         Overall_Rating       False
         Verified             False
         Type Of Traveller    False
         Seat Type            False
         Route                False
         Date Flown           False
         Seat Comfort         False
         Food & Beverages     False
         Ground Service       False
         Recommended          False
         dtype: bool
```

Activity 2.2: CONVERTING THE CATEGORICAL COLUMN TO NUMERIC USING LABEL ENCODING:

i)le=LabelEncoder(), introducing a new object for label encoding:

```
In [13]: le=LabelEncoder() #creating an object for label encoding
```

ii)df.head():

The df.head() command in Pandas is used to display the first few rows of the DataFrame df.

```
In [14]: df.head()
Out[14]:
```

	Airline Name	Overall_Rating	Verified	Type Of Traveller	Seat Type	Route	Date Flown	Seat Comfort	Food & Beverages	Ground Service	Recommended
0	AB Aviation	9	True	Solo Leisure	Economy Class	Moroni to Moheli	November 2019	4.0	4.000000	4.0	yes
1	AB Aviation	1	True	Solo Leisure	Economy Class	Moroni to Anjouan	June 2019	2.0	1.000000	1.0	no
2	AB Aviation	1	True	Solo Leisure	Economy Class	Anjouan to Dzaoudzi	June 2019	2.0	1.000000	1.0	no
3	Adria Airways	1	False	Solo Leisure	Economy Class	Frankfurt to Pristina	September 2019	1.0	2.553586	1.0	no
4	Adria Airways	1	True	Couple Leisure	Economy Class	Sofia to Amsterdam via Ljubljana	September 2019	1.0	1.000000	1.0	no

iii) df[['Month flown','Year flown']]=df['Date Flown'].str.split(expand=True):

To split the values in the 'Date Flown' column of the DataFrame df and then assigning the resulting substrings to new columns 'Month flown' and 'Year flown'.

```
In [15]: df[['Month flown','Year flown']]=df['Date Flown'].str.split(expand=True) #splitting the string
```

iv) df=df.drop(["Date Flown"],axis=1):

To remove the "Date Flown" column from the DataFrame df.

```
In [16]: df=df.drop(["Date Flown"],axis=1) #dropping the data flown column
```

v) order=['Airline Name','Overall_Rating','Verified','Type Of Traveller','Seat Type','Route','Month flown','Year flown','Seat Comfort' ,'Food & Beverages' ,'Ground Service','Recommended']

df=df.reindex(columns=order):

To reorder the columns of the DataFrame df based on the specified list of column names in the order list.

```
In [18]: #we are converting the columns by using label Encoder
df['Airline Name']=le.fit_transform(df['Airline Name'])
df['Overall_Rating']=le.fit_transform(df['Overall_Rating'])
df['Verified']=le.fit_transform(df['Verified'])
df['Type Of Traveller']=le.fit_transform(df['Type Of Traveller'])
df['Seat Type']=le.fit_transform(df['Seat Type'])
df['Route']=le.fit_transform(df['Route'])
df['Month flown']=le.fit_transform(df['Month flown'])
df['Recommended']=le.fit_transform(df['Recommended'])
```

vi) df.head():

The df.head() command in Pandas is used to display the first few rows of the DataFrame df.

```
In [19]: df.head()
Out[19]:
```

	Airline Name	Overall_Rating	Verified	Type Of Traveller	Seat Type	Route	Month flown	Year flown	Seat Comfort	Food & Beverages	Ground Service	Recommended
0	0	8	1	3	1	8585	9	2019	4.0	4.000000	4.0	1
1	0	0	1	3	1	8584	6	2019	2.0	1.000000	1.0	0
2	0	0	1	3	1	482	6	2019	2.0	1.000000	1.0	0
3	4	0	0	3	1	4256	11	2019	1.0	2.553586	1.0	0
4	4	0	1	1	1	11778	11	2019	1.0	1.000000	1.0	0

Exploratory Data Analysis

Activity 1: VISUALIZATION OF DATA:

i)df.describe():

The df.describe() is a Pandas DataFrame method that provides statistical summaries of the numerical columns in the DataFrame.

```
In [20]: df.describe() #it gives the statical calculation of data
Out[20]:
```

	Airline Name	Overall_Rating	Verified	Type Of Traveller	Seat Type	Route	Month flown	Seat Comfort	Food & Beverages	Ground Service	Recomr
count	23171.000000	23171.000000	23171.000000	23171.000000	23171.000000	23171.000000	23171.000000	23171.000000	23171.000000	23171.000000	23171
mean	245.027880	2.393423	0.531785	2.008675	0.973242	7016.380130	5.495188	2.618321	2.553586	2.353738	0
std	144.313766	3.104117	0.498999	1.077595	0.457584	3612.341934	3.019861	1.327017	1.207396	1.421145	0
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0
25%	124.000000	0.000000	0.000000	1.000000	1.000000	4101.500000	3.000000	1.000000	2.000000	1.000000	0
50%	237.000000	0.000000	1.000000	2.000000	1.000000	8093.000000	6.000000	2.618321	2.553586	2.353738	0
75%	373.000000	5.000000	1.000000	3.000000	1.000000	9482.000000	7.000000	4.000000	3.000000	3.000000	1
max	496.000000	9.000000	1.000000	3.000000	3.000000	13606.000000	11.000000	5.000000	5.000000	5.000000	1

ii) df.corr(numeric_only=True):

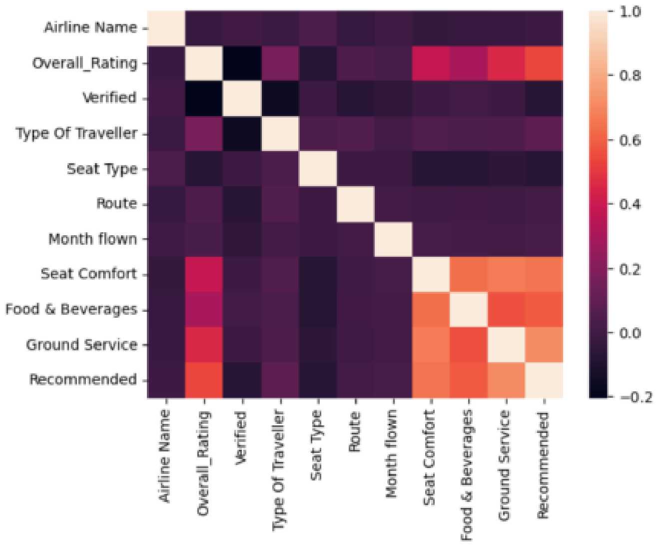
The df.corr() method in Pandas is used to calculate the correlation coefficients between numerical columns in a DataFrame.

```
In [21]: df.corr(numeric_only=True) #corretion between the columns
Out[21]:
```

	Airline Name	Overall_Rating	Verified	Type Of Traveller	Seat Type	Route	Month flown	Seat Comfort	Food & Beverages	Ground Service	Recommended
Airline Name	1.000000	-0.022610	0.006762	-0.018686	0.032295	-0.030981	-0.006680	-0.035857	-0.026203	-0.023924	-0.015614
Overall_Rating	-0.022610	1.000000	-0.204288	0.163941	-0.076057	0.041676	0.027643	0.382214	0.295038	0.450949	0.534467
Verified	0.006762	-0.204288	1.000000	-0.159073	-0.008561	-0.076355	-0.053120	-0.009543	0.009813	-0.009349	-0.073313
Type Of Traveller	-0.018686	0.163941	-0.159073	1.000000	0.036095	0.049901	0.009462	0.048675	0.033289	0.041037	0.087046
Seat Type	0.032295	-0.076057	-0.008561	0.036095	1.000000	-0.008572	-0.007745	-0.078122	-0.080341	-0.061260	-0.076048
Route	-0.030981	0.041676	-0.076355	0.049901	-0.008572	1.000000	0.012067	-0.000635	0.006327	-0.003002	0.010862
Month flown	-0.006680	0.027643	-0.053120	0.009462	-0.007745	0.012067	1.000000	0.021298	0.015861	0.011897	0.025007
Seat Comfort	-0.035857	0.382214	-0.009543	0.048675	-0.078122	-0.000635	0.021298	1.000000	0.635971	0.672536	0.651380
Food & Beverages	-0.026203	0.295038	0.009813	0.033289	-0.080341	0.006327	0.015861	0.635971	1.000000	0.558308	0.581473
Ground Service	-0.023924	0.450949	-0.009349	0.041037	-0.061260	-0.003002	0.011897	0.672536	0.558308	1.000000	0.709446
Recommended	-0.015614	0.534467	-0.073313	0.087046	-0.076048	0.010862	0.025007	0.651380	0.581473	0.709446	1.000000

iii) sns.heatmap(df.corr(numeric_only=True)): The sns.heatmap() function from the Seaborn library is used to visualize the correlation matrix as a heatmap. The heatmap visually represents the strength and direction of the relationships between numerical columns in the DataFrame.

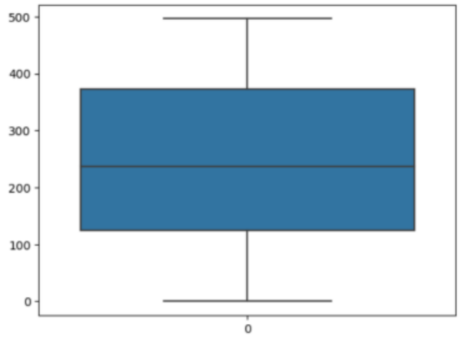
```
In [22]: sns.heatmap(df.corr(numeric_only=True)) #it will gives relation between the columns
Out[22]: <Axes: >
```



iv) sns.boxplot(df["Airline Name"]):

The sns.boxplot() function from the Seaborn library is used to create a boxplot for the distribution of the "Airline Name" column variable.

```
In [23]: sns.boxplot(df["Airline Name"])
Out[23]: <Axes: >
```



Activity 2:

SPLITTING THE INDEPENDENT AND DEPENDENT VARIABLE:

i)x=df.iloc[:,:11] #independent variable

y=df["Recommended"] #dependent variable

```
In [30]: x=df.iloc[:,:11] #independent variable
         y=df["Recommended"]  #dependent variable
```

ii) x.head():

The x.head() is used to display the first few rows of the DataFrame x.

```
In [31]: x.head()
Out[31]:
```

	Airline Name	Overall_Rating	Verified	Type Of Traveller	Seat Type	Route	Month flown	Year flown	Seat Comfort	Food & Beverages	Ground Service
0	0	8	1	3	1	8585	9	2019	4.0	4.000000	4.0
1	0	0	1	3	1	8584	6	2019	2.0	1.000000	1.0
2	0	0	1	3	1	482	6	2019	2.0	1.000000	1.0
3	4	0	0	3	1	4256	11	2019	1.0	2.553586	1.0
4	4	0	1	1	1	11778	11	2019	1.0	1.000000	1.0

iii) y.head()

The y.head() is used to display the first few rows of the DataFrame y.

```
In [32]: y.head()
Out[32]: 0    1
         1    0
         2    0
         3    0
         4    0
         Name: Recommended, dtype: int32
```

iv)x.shape()

The x.shape is used to determine the number of rows and columns of the DataFrame x.

```
In [33]: x.shape
Out[33]: (23171, 11)
```

v)y.shape()

The y.shape is used to determine the number of rows and columns of the DataFrame y.

```
In [34]: y.shape
Out[34]: (23171,)
```

vi) y.value_counts():

The y.value_counts() is used to get a count of unique values in the DataFrame column represented by the variable y.

```
In [35]: y.value_counts()
Out[35]: 0    15364
         1     7807
         Name: Recommended, dtype: int64
```

vii) ms=MinMaxScaler()

x_scaled=pd.DataFrame(ms.fit_transform(x),columns=x.columns) #feature scaling

x_scaled:

To standardize the range of independent variables.

```
In [36]: ms=MinMaxScaler()
         x_scaled=pd.DataFrame(ms.fit_transform(x),columns=x.columns)  #feature scaling
         x_scaled
Out[36]:
```

	Airline Name	Overall_Rating	Verified	Type Of Traveller	Seat Type	Route	Month flown	Year flown	Seat Comfort	Food & Beverages	Ground Service
0	0.000000	0.888889	1.0	1.000000	0.333333	0.630972	0.818182	0.636364	0.800000	0.800000	0.750000
1	0.000000	0.000000	1.0	1.000000	0.333333	0.630898	0.545455	0.636364	0.400000	0.200000	0.000000
2	0.000000	0.000000	1.0	1.000000	0.333333	0.035426	0.545455	0.636364	0.400000	0.200000	0.000000
3	0.008065	0.000000	0.0	1.000000	0.333333	0.312803	1.000000	0.636364	0.200000	0.510717	0.000000
4	0.008065	0.000000	1.0	0.333333	0.333333	0.865648	1.000000	0.636364	0.200000	0.200000	0.000000
...	...	...	...	...	...	...	...	...	...	...	...
23166	0.981855	0.000000	0.0	0.333333	0.333333	0.094517	0.545455	0.909091	0.400000	0.510717	0.000000
23167	0.981855	0.000000	1.0	1.000000	0.333333	0.863296	0.545455	0.909091	0.523664	0.510717	0.338435
23168	0.981855	0.222222	1.0	0.000000	0.333333	0.094517	0.727273	0.909091	0.400000	0.400000	0.000000
23169	0.981855	0.555556	1.0	0.000000	0.000000	0.927826	0.727273	0.909091	0.600000	0.600000	0.000000
23170	0.981855	0.000000	1.0	1.000000	0.333333	0.863296	0.727273	0.909091	0.523664	0.510717	0.000000

23171 rows × 11 columns

Activity 3:

Splitting data into train and test and validation sets.

Now let's split the Dataset into train, test and validation sets. First split the dataset into train and test sets.

i)x_train,x_test,y_train,y_test=train_test_split(x_scaled,y,test_size=0.2, random_state=None):

The train_test_split function from scikit-learn to split the scaled feature matrix (x_scaled) and the target variable (y) into training and testing sets.

```
In [37]: #splitting the training data and testing data
         x_train,x_test,y_train,y_test=train_test_split(x_scaled,y,test_size=0.2,random_state=None)
```

ii)smote=SMOTE()

x_train_smote,y_train_smote=smote.fit_resample(x_train,y_train):

'''we are dealing with imbalanced data

so to balance that data we use smote technique'''

SMOTE technique is a common approach when dealing with imbalanced datasets, where one class is underrepresented compared to the other.

```
In [38]: '''we are dealing with imbalanced data
         so to balance that data we use smote technique'''
         smote=SMOTE()
         x_train_smote,y_train_smote=smote.fit_resample(x_train,y_train)
```

iii) x_train_smote.shape:

The x_train_smote.shape is used to check thevnumber of rows and columns of the feature matrix x_train_smote after applying the SMOTE.

```
In [39]: x_train_smote.shape
Out[39]: (24572, 11)
```

iv) y_train_smote.shape:

The y_train_smote.shape is used to check thevnumber of rows and columns of the feature matrix y_train_smote after applying the SMOTE.

```
In [40]: y_train_smote.shape
Out[40]: (24572,)
```

v) y_train_smote.value_counts():

Now the data is balanced.

```
In [41]: y_train_smote.value_counts() #now the data is balanced
Out[41]: 0    12286
         1    12286
         Name: Recommended, dtype: int64
```

Model Building:

i) knn=KNeighborsClassifier()

knn.fit(x_train_smote,y_train_smote):

The k-Nearest Neighbors (KNN) classifier from scikit-learn to train the model on the resampled training data.

```
In [42]: knn=KNeighborsClassifier()
         knn.fit(x_train_smote,y_train_smote)
Out[42]: KNeighborsClassifier()
         In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
         On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.
```

ii) pred=knn.predict(x_test):

To make predictions on the test set (x_test).

```
In [43]: pred=knn.predict(x_test)
```

iii)pred:

```
In [44]: pred
Out[44]: array([1, 1, 0, ..., 1, 1, 0])
```

iv)y_test:

```
In [45]: y_test
Out[45]: 12130    1
         5371     1
         7832     0
         15735    0
         17239    1
                 ..
         14642    0
         2328     1
         12056    1
         20284    1
         18881    0
         Name: Recommended, Length: 4635, dtype: int32
```

Model Deployment:

i)accuracy_score(pred,y_test):

To compute the accuracy of the machine learning model's predictions.

```
In [46]: accuracy_score(pred,y_test)
Out[46]: 0.9376483279395901
```

ii) print(classification_report(pred,y_test)):

To generate a detailed classification report for your machine learning model's predictions.

```
In [47]: print(classification_report(pred,y_test))
              precision    recall  f1-score   support

           0       0.94      0.97      0.95      2975
           1       0.94      0.88      0.91      1660

    accuracy                           0.94      4635
   macro avg       0.94      0.93      0.93      4635
weighted avg       0.94      0.94      0.94      4635
```

import pickle

```
In [48]: import pickle
```

pickle.dump(knn,open('ar_knn.pkl','wb'))

```
In [50]: pickle.dump(knn,open('ar_knn.pkl','wb'))
```

8.PERFORMANCE TESTING

8.1 Performace Metrics

Activity 2:

Integrate with Web Framework In this section,

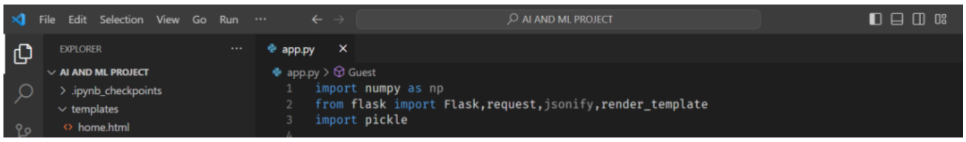
we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI. This section has the following tasks

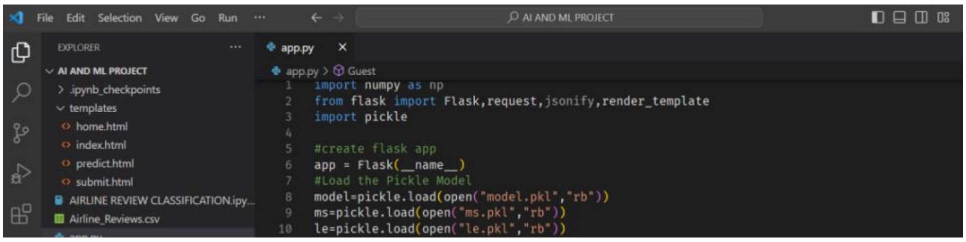
- Building HTML Pages
- Building server-side script
- Run the web application

Activity 2.2:

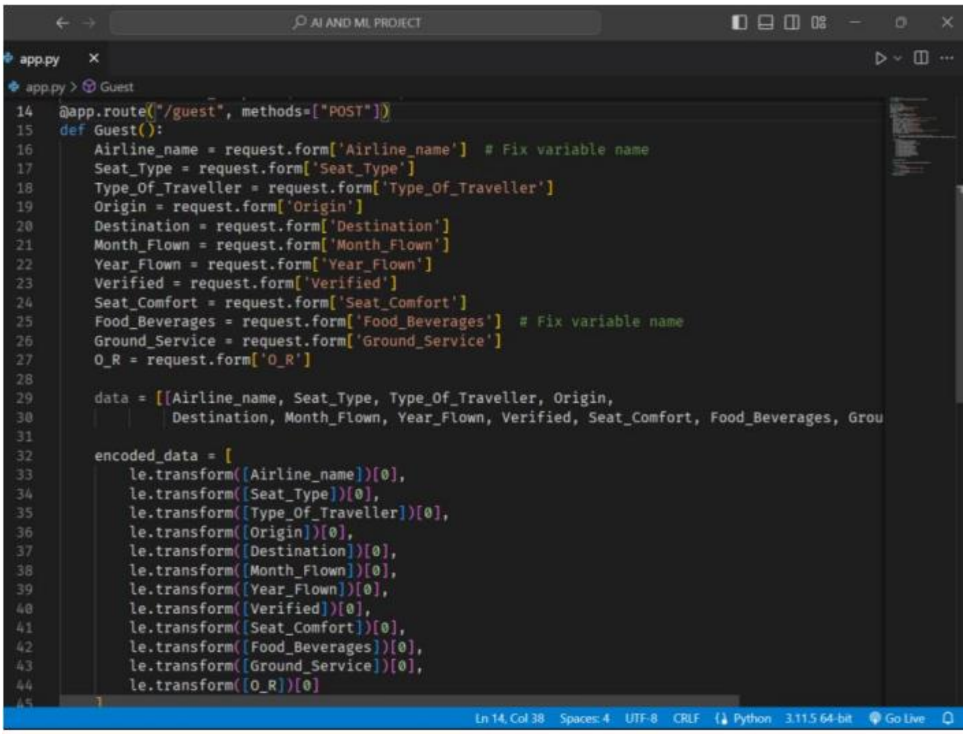
Build Python code:

Importing the flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (name) as argument.





Here we will be using a declared constructor to route to the HTML page which we have created earlier. In the above example, '/' URL is bound with the index.html function. Hence, when the home page of the web server is opened in the browser, the html page will be rendered. Whenever you enter the values from the html page the values can be retrieved using POST Method. Retrieves the value from UI:

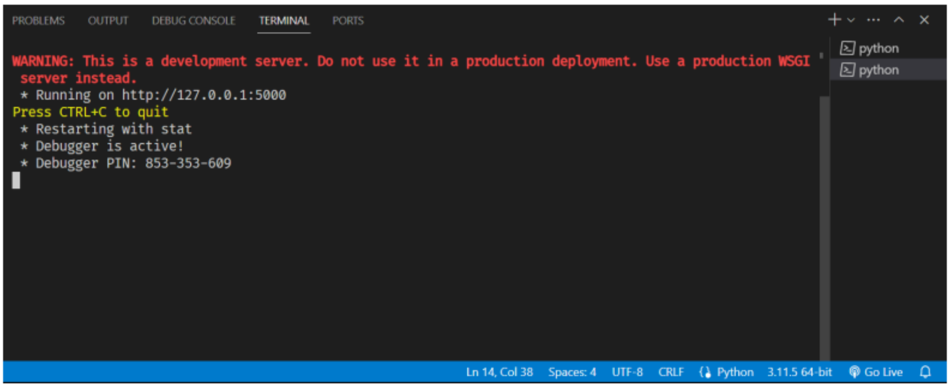


Here we are routing our app to Guest() function. This function retrieves all the values from the HTML page using Post request. That is stored in an array. This array is passed to the model.predict() function. This function returns the prediction. And this prediction value will be rendered to the text that we have mentioned in the index.html page. Set app.run(debug=True) so that we can edit.

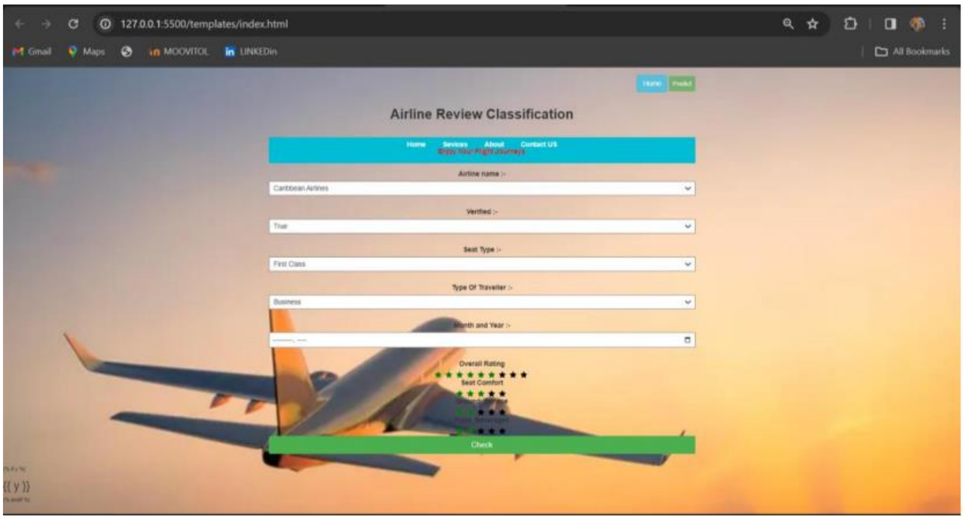
9.RESULTS

9.1 Output Screenshots

Running the Application:



Now, Go the web browser and write the localhost URL (http://127.0.0.1:5000) to get the below result



10.ADVANTAGES AND DISADVANTAGES

Advantages:

Customer Insights:

Classification of airline reviews provides valuable insights into customer opinions and preferences, helping airlines understand areas of strength and weakness.

Quality Improvement:

Identifying specific aspects mentioned in reviews allows airlines to focus on improving services or addressing issues raised by customers.

Competitive Analysis:

Airline review classification enables competitive analysis by comparing sentiment trends and service ratings with those of other airlines.

Marketing and Branding:

Positive reviews can be used for marketing and branding purposes, showcasing the strengths of the airline and attracting potential customers.

Operational Efficiency:

Reviews can highlight operational inefficiencies or areas where processes can be streamlined, leading to improved overall efficiency.

Customer Engagement:

Responding to reviews, especially negative ones, demonstrates a commitment to customer satisfaction and engagement.

Product Development:

Insights gained from reviews can inform product development strategies, helping airlines tailor their services to meet customer expectations.

Disadvantages:

Bias in Sentiment Analysis:

Sentiment analysis algorithms may have biases, leading to inaccurate classification of reviews and potential misinterpretation of customer sentiments.

Limited Context Understanding:

Automated systems might struggle to understand the nuanced context of reviews, potentially misclassifying sentiments due to sarcasm or complex language.

Overemphasis on Extreme Views:

Classification systems may give disproportionate weight to extreme positive or negative reviews, potentially skewing the overall perception of the airline.

Inconsistent Review Criteria:

Different users may have varying criteria for rating airlines, leading to inconsistencies in the classification process.

Privacy Concerns:

Analysing and categorizing reviews raises privacy concerns, especially if personal details or specific incidents are mentioned in the reviews.

Dependency on User Participation:

The system's effectiveness relies on users actively participating in submitting reviews, which may not capture the opinions of all customers.

Resource Intensive:

Developing and maintaining a robust review classification system can be resource-intensive, requiring ongoing efforts in algorithm refinement and system updates.

Resistance to Change:

Airlines may face resistance to change if internal processes or policies need adjustment based on customer feedback obtained through the classification system.

11.CONCLUSION

In summary, the way that passengers and airlines interact with each other is greatly influenced by the categorization of airline reviews. Airlines may obtain important insights into customer feelings, pinpoint areas for development, and increase operational efficiency by methodically classifying and evaluating evaluations. This procedure not only makes it easier to make strategic decisions, but it also gives airlines the ability to proactively address customer problems and promote a continuous improvement culture. But issues like sentiment analysis's possible biases and the requirement for uniform standards draw attention to how crucial it is to combine automated processes with human control. In the end, airlines may find that a well-executed review classification system is a useful tool for meeting consumer expectations, streamlining their offerings, and preserving their competitive advantage in the ever-changing aviation sector.

12.FUTURE SCOPE

Advanced Sentiment Analysis Algorithms:

Continued advancements in natural language processing (NLP) and machine learning algorithms will likely lead to more sophisticated sentiment analysis. This includes improved context understanding, sentiment nuance recognition, and reduced biases.

Integration of Emerging Technologies:

Integration with emerging technologies such as artificial intelligence (AI), machine learning, and blockchain may further enhance the accuracy and security of the review classification process.

Real-time Feedback Systems:

The development of real-time feedback systems could allow airlines to address customer concerns immediately, leading to quicker problem resolution and improved overall customer satisfaction.

13. APPENDIX

The github link for the ipynb file:

https://github.com/smartinternz02/SI-GuidedProject-609020-1698402277/blob/main/PHASE%204/AIRLINE%20REVIEW%20CLASSIFICATION.ipynb

PROJECT DEMO VIDEO LINK:

https://drive.google.com/file/d/15ELtri5BmB2UM3QtcFdfpXK41BqUjsv3/view?usp=sharing

THE GITHUB LINK FOR THE FLASK CODE:

https://github.com/smartinternz02/SI-GuidedProject-609020-1698402277/blob/main/PHASE%204/FLASK/app.py

GITHUB REPOSITORY LINK:

https://github.com/smartinternz02/SI-GuidedProject-609020-1698402277