**Project Report Format**

**1.INTRODUCTION**

* 1. Project Overview

The project's goal is to use machine learning techniques to determine airline passenger satisfaction. The goal is to forecast whether a passenger will be satisfied or dissatisfied with their flying experience based on factors such as airline, flight duration, cabin class, seat comfort, in-flight entertainment, and onboard service.

* 1. Purpose

Identifying airline passengers' satisfaction using Machine Learning (ML) serves a multifaceted purpose that significantly enhances the overall travel experience for both passengers and airlines. ML algorithms examine a variety of data sources, including customer feedback and historical data, to identify patterns that influence satisfaction. This allows airlines to tailor services, predict passenger behaviour, and optimize operational efficiency, all of which contribute to more comfortable journeys. In essence, artificial intelligence (AI) applications in the aviation industry improve the passenger experience by making it more efficient, personalized, and satisfying.

**2.LITERATURE SURVEY**

* 1. Existing problem

The literature review for the project on determining airline passenger satisfaction using Machine Learning reveals a number of common challenges and gaps. The scarcity of high-quality, standardized datasets, potential biases in data, and the need for models to generalize across diverse contexts are all issues.

* 1. References

Gao, K., Yang, Y., & Qu, X. (2021). Examining nonlinear and interaction effects of multiple determinants on airline travel satisfaction. *Transportation Research Part D: Transport and Environment*, *97*, 102957.

Baydogan, C., & Alatas, B. (2019, November). Detection of customer satisfaction on unbalanced and multi-class data using machine learning algorithms. In *2019 1st International Informatics and Software Engineering Conference (UBMYK)* (pp. 1-5). IEEE.

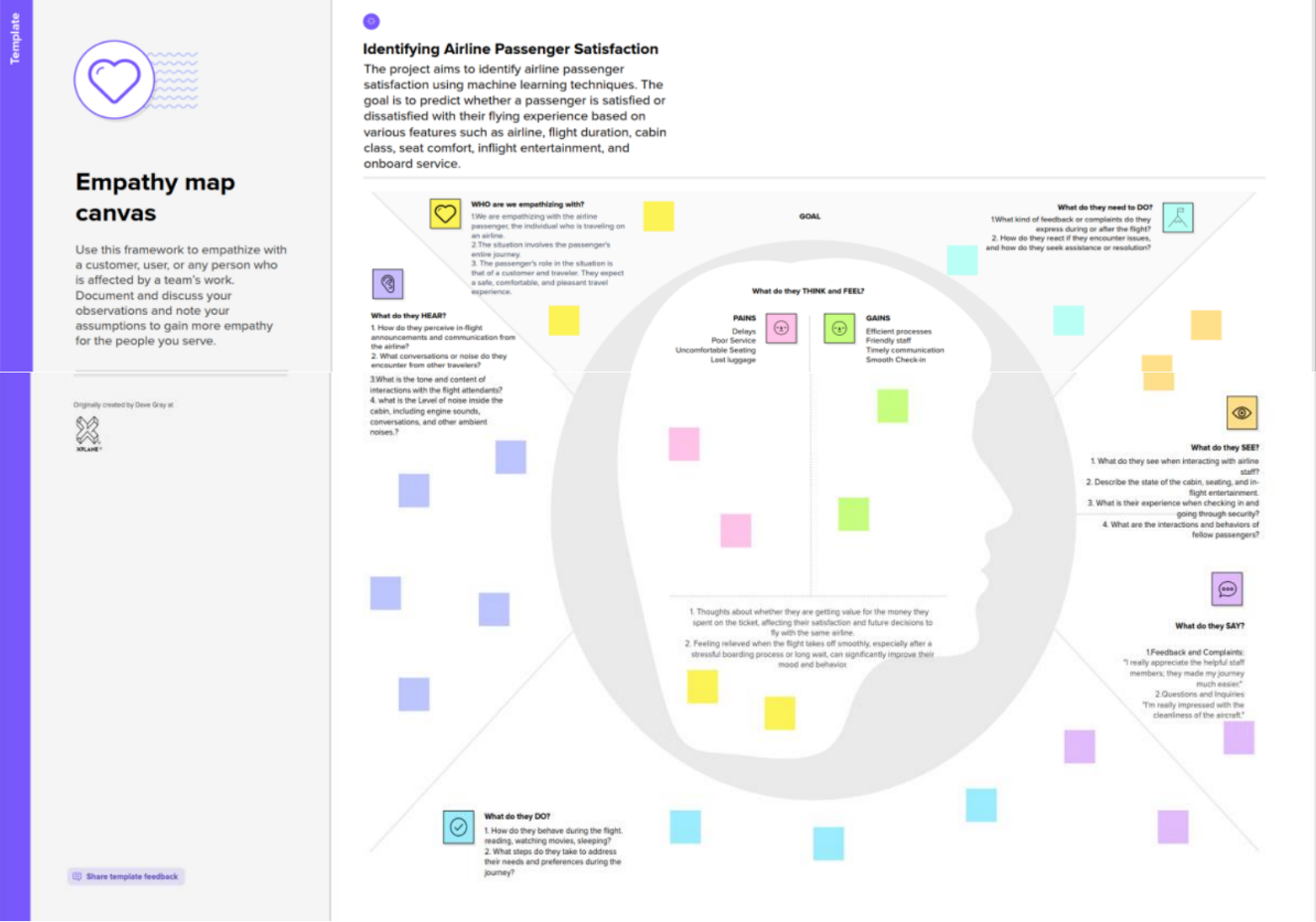
* 1. Problem Statement Definition

Create an ML solution to analyse extensive airline passenger data, identifying patterns and factors influencing satisfaction, enabling airlines to make data-driven decisions, enhance services, and elevate overall customer satisfaction levels.

**3.IDEATION & PROPOSED SOLUTION**

* 1. Empathy Map Canvas

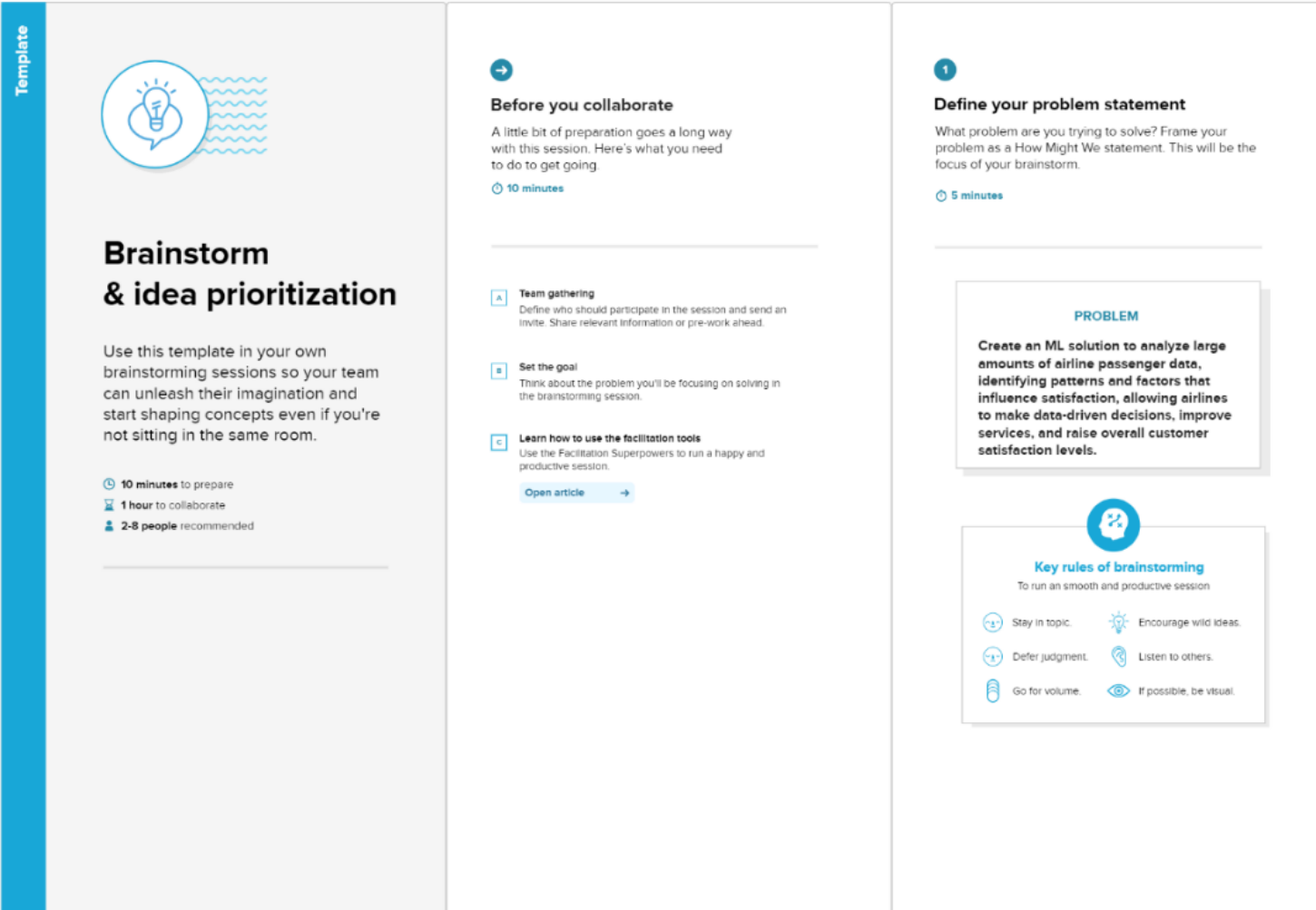
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.



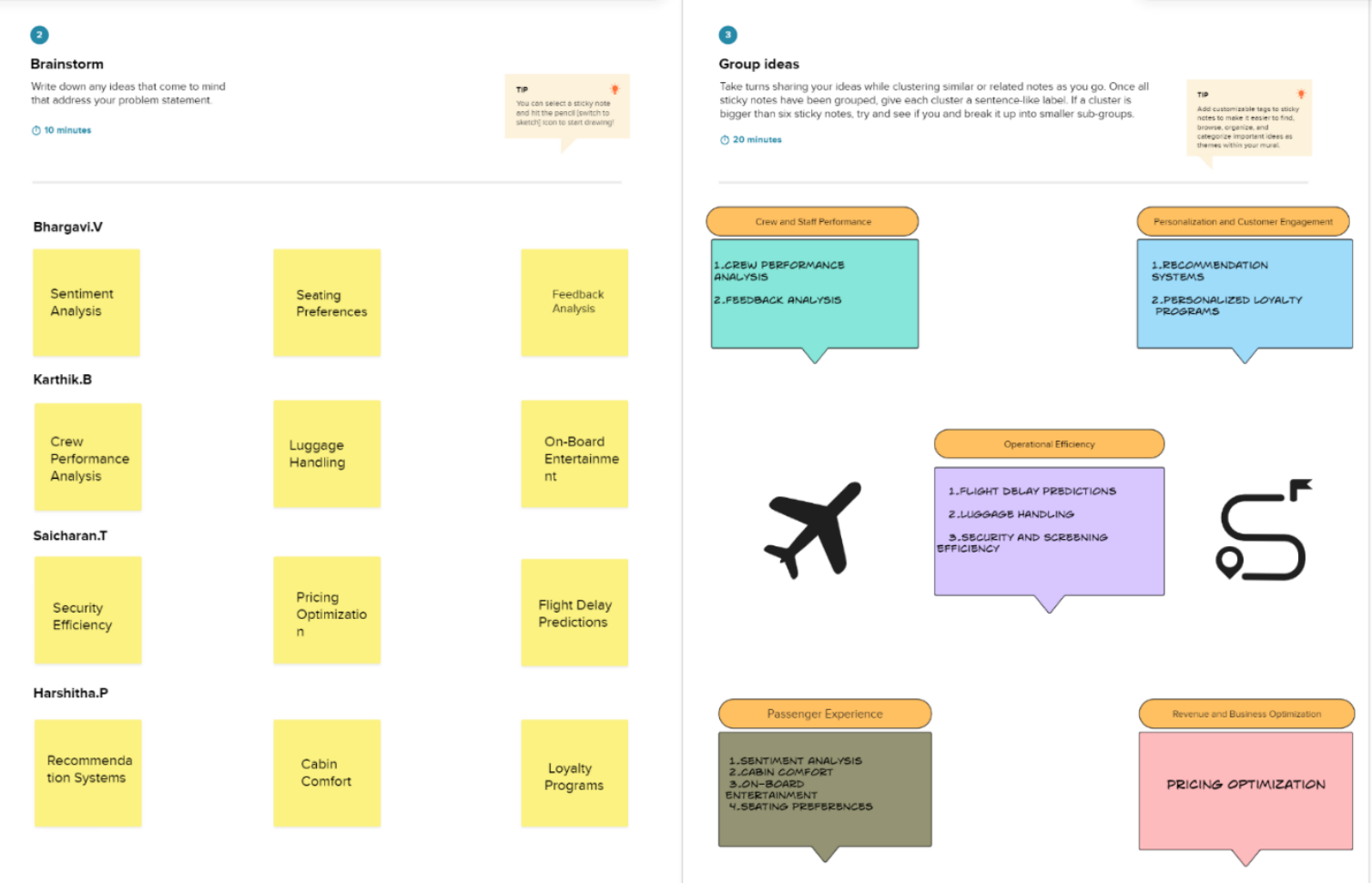
* 1. Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich number of creative solutions.

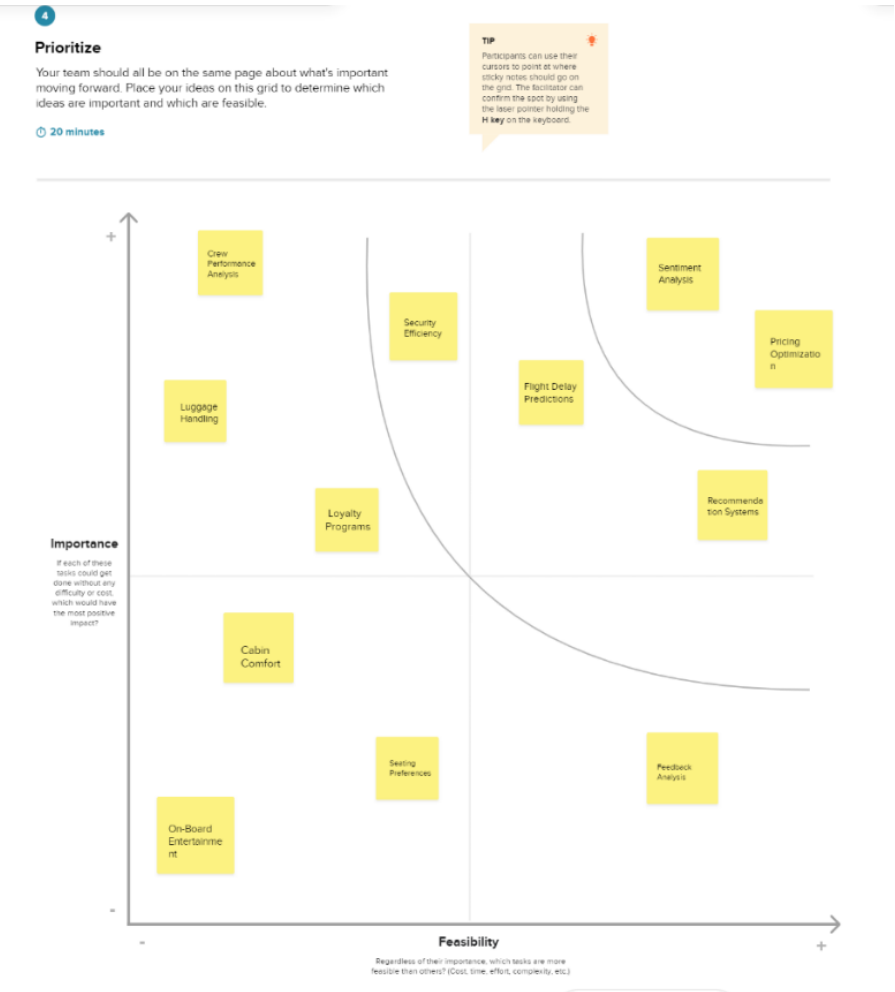
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



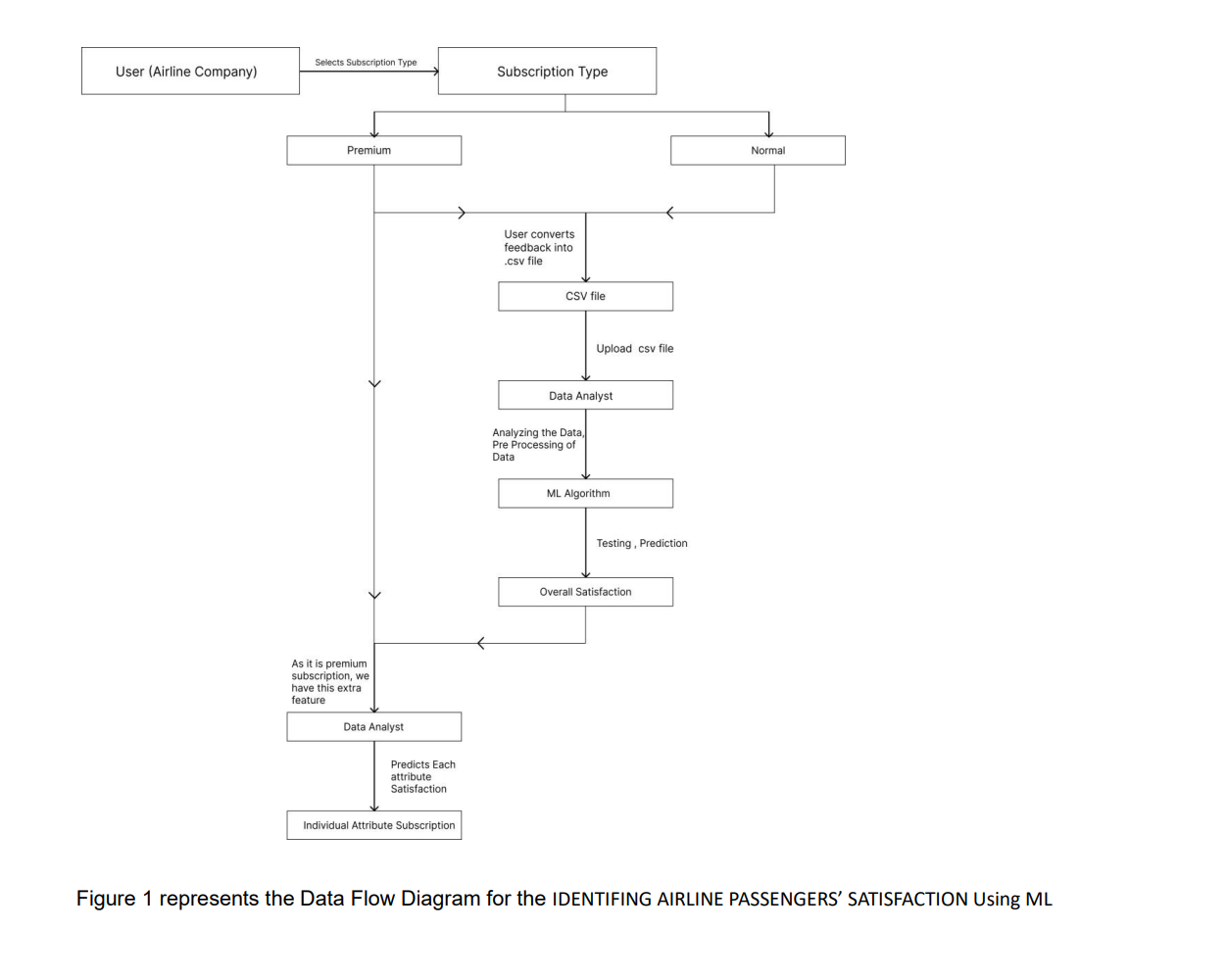
**4.REQUIREMENT ANALYSIS**

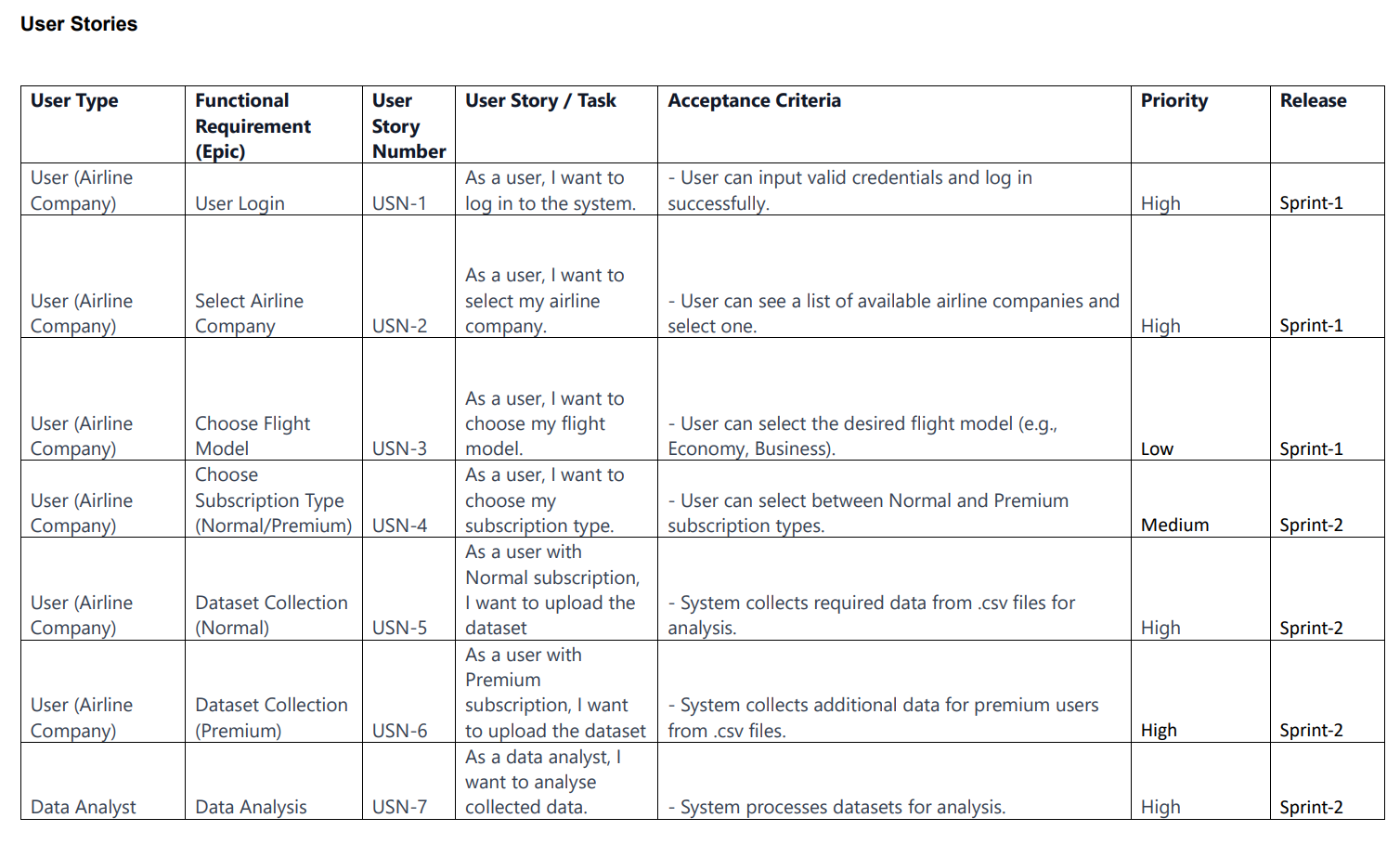
* 1. Functional requirement
* Data Collection and Integration: The system should be able to collect and integrate diverse data sources, including passenger feedback, flight information, and operational data.
* Predictive Modelling: Develop ML models capable of predicting passenger satisfaction levels based on historical data and real-time inputs.
* Feedback Processing: Implement automated systems for processing and categorizing feedback, identifying specific areas of concern or praise.
  1. Non-Functional requirements
* Performance: The system should respond to user queries and requests within a defined timeframe, ensuring optimal performance even with a growing dataset.
* Data Quality: Implement measures to ensure the quality and accuracy of data used for training ML models, minimizing biases and errors.
* Cost Efficiency: Optimize system architecture and operations to ensure cost efficiency in terms of infrastructure, maintenance, and data storage.

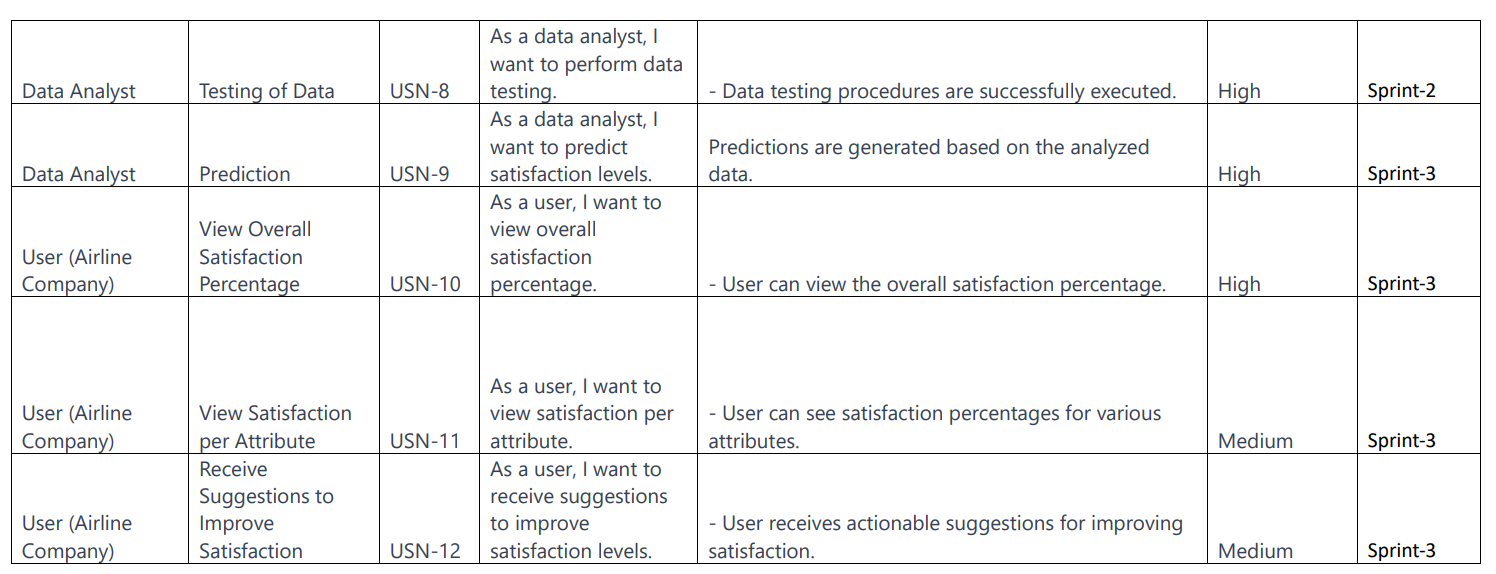
**5.PROJECT DESIGN**

5.1 Data Flow Diagrams & User Stories

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.







* 1. Solution Architecture

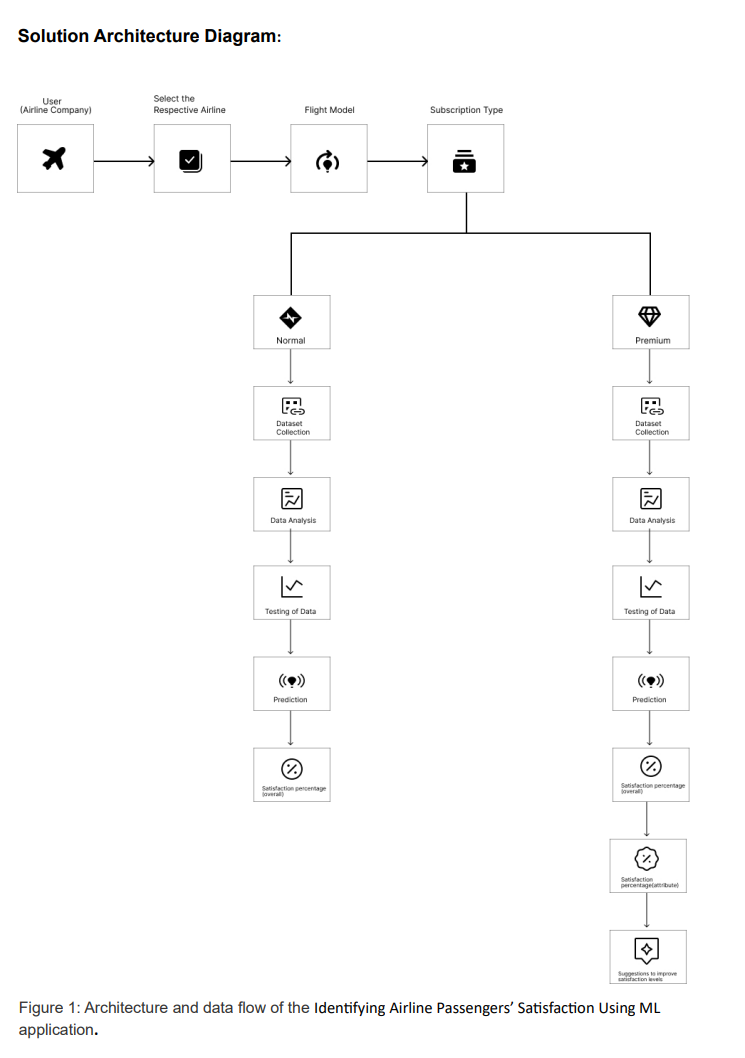
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

• Find the best tech solution to solve existing business problems.

• Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.

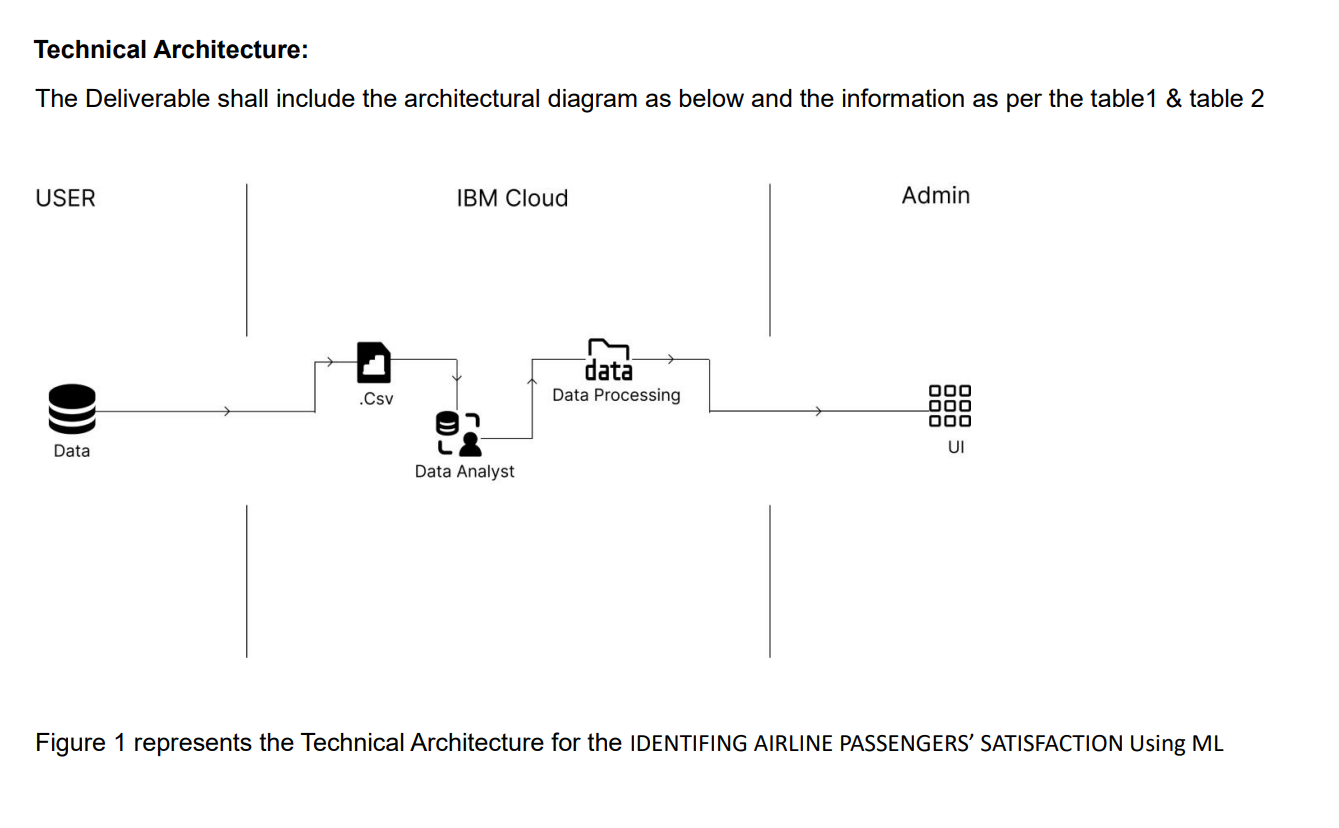
• Define features, development phases, and solution requirements.

• Provide specifications according to which the solution is defined, managed, and delivered.

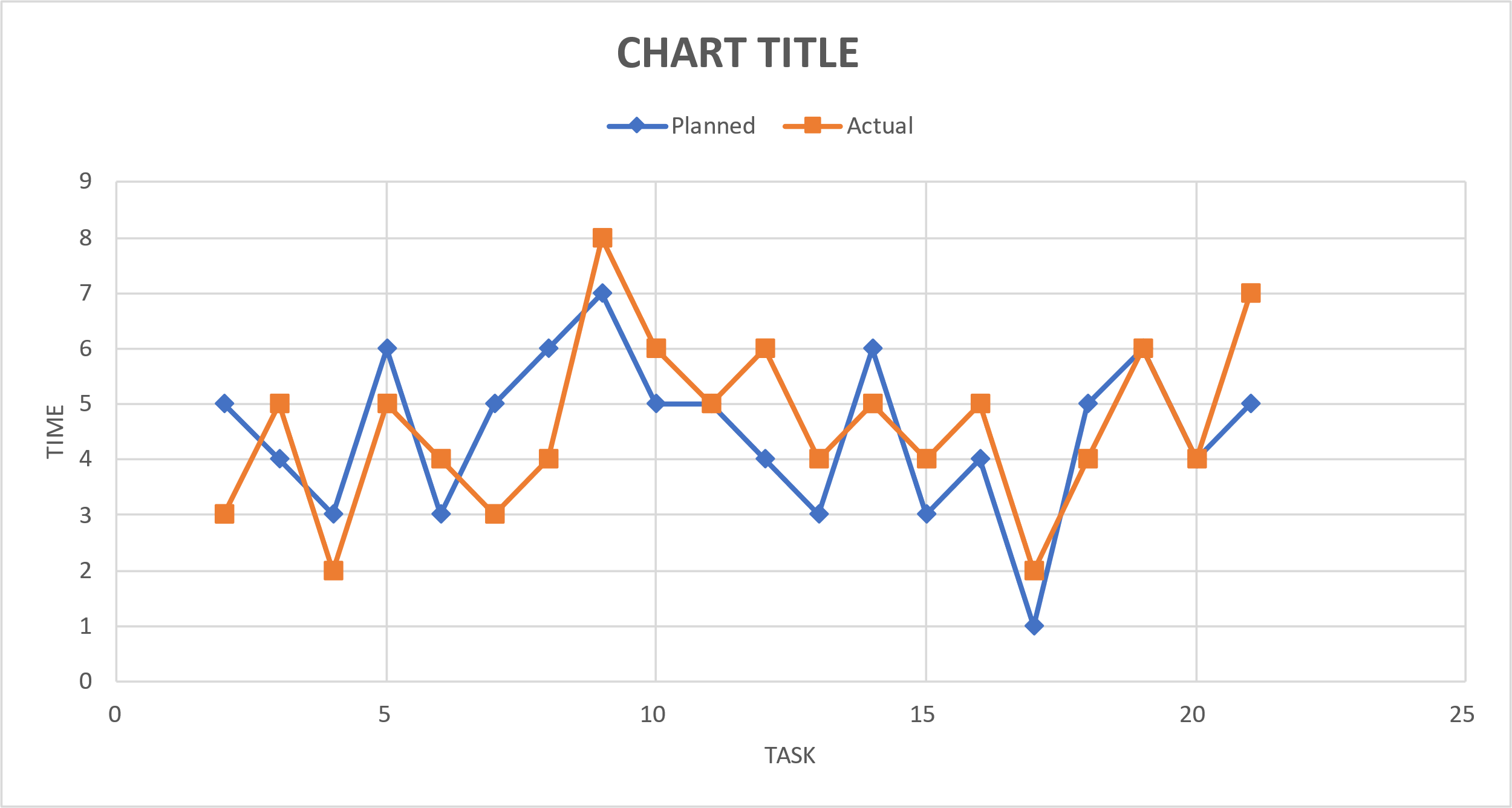


**6.PROJECT PLANNING & SCHEDULING**

6.1 Technical Architecture



* 1. Sprint Planning & Estimation



* 1. Sprint Delivery Schedule

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sprint** | **Total Story Points** | **Sprint Start Date** | **Sprint End Date**  **(Planned)** | **Story Points**  **Completed (as on**  **Planned End Date)** | **Sprint End date (Actual)** |
| 1 | 26 | 30-10-23 | 05-11-23 | 21 | 06-11-23 |
| 2 | 21 | 06-11-23 | 09-11-23 | 23 | 09-11-23 |
| 3 | 21 | 10-11-23 | 15-11-23 | 23 | 15-11-23 |
| 4 | 20 | 16-11-23 | 18-11-23 | 21 | 18-11-23 |

|  |  |  |
| --- | --- | --- |
| **Dates** | **Planned** | **Actual** |
|  |  |  |
| 30-10-2023 | 5 | 3 |
| 31-10-2023 | 4 | 5 |
| 01-11-2023 | 3 | 2 |
| 02-11-2023 | 6 | 5 |
| 03-11-2023 | 3 | 4 |
| 04-11-2023 | 5 | 3 |
| 05-11-2023 | 6 | 4 |
| 06-11-2023 | 7 | 8 |
| 07-11-2023 | 5 | 6 |
| 08-11-2023 | 5 | 5 |
| 09-11-2023 | 4 | 6 |
| 10-11-2023 | 3 | 4 |
| 11-11-2023 | 6 | 5 |
| 12-11-2023 | 3 | 4 |
| 13-11-2023 | 4 | 5 |
| 14-11-2023 | 1 | 2 |
| 15-11-2023 | 5 | 4 |
| 16-11-2023 | 6 | 6 |
| 17-11-2023 | 4 | 4 |
| 18-11-2023 | 5 | 7 |

**7.CODING & SOLUTIONING (Explain the features added in the project along with code)**

7.1 Feature 1:

Using Machine Learning to identify airline passenger satisfaction, our project provides satisfaction levels based on feedback.

App.py

from flask import Flask, render\_template, request , jsonify

import numpy as np

import pickle

model = pickle.load(open('Airline Passenger.pkl', 'rb'))

app = Flask(\_name\_)

gender\_mapping = {'male': 1, 'female': 0}

travel\_mapping = {'Business Travel': 0, 'Personal Travel': 1}

customer\_mapping={'Loyal Customer':0, 'Disloyal Customer':1}

class\_mapping={'Business':0, 'Eco':1, 'Eco Plus':2}

satisfy\_mapping={'Satisfied':0, 'Neutral or Dissatisfied':1}

@app.route("/")

def about():

return render\_template('home.html')

@app.route("/home")

def home():

return render\_template('home.html')

@app.route("/predict")

def home1():

return render\_template('predict.html')

@app.route("/submit")

def home2():

return render\_template('submit.html')

@app.route("/predict",methods = ["POST"])

def predict():

if request.method=="POST":

ID=request.form['id']

Gender = gender\_mapping[request.form["gender"]]

Customer\_Type=customer\_mapping[request.form['customerType']]

Age = int(request.form['age'])

Class =class\_mapping[request.form['Class']]

Type\_of\_Travel\_Class= travel\_mapping[request.form['travelClass']]

Flight\_Distance=int(request.form['flightDistance'])

Inflight\_wifi\_service= request.form['wifiService']

Departure\_Arrival\_time\_convenient = request.form['timeConvenience']

Ease\_of\_Online\_booking =request.form['onlineBooking']

Gate\_location =request.form['gateLocation']

Food\_and\_drink =request.form['foodAndDrink']

Online\_boarding=request.form['onlineBoarding']

Seat\_comfort =request.form["seatComfort"]

Inflight\_entertainment= request.form['inflightEntertainment']

On\_board\_service= request.form['onboardService']

Leg\_room\_service= request.form['legRoomService']

Baggage\_handling= request.form['baggageHandling']

Checkin\_service=request.form['checkinService']

Inflight\_service=request.form['inflightService']

Cleanliness =request.form['cleanliness']

Departure\_Delay\_in\_Minutes= int(request.form['departureDelay'])

Arrival\_Delay\_in\_Minutes=int(request.form['arrivalDelay'])

Satisfaction=satisfy\_mapping[request.form['satisfaction']]

t = [[int(ID),float(Gender),str(Customer\_Type),int(Age),str(Class),str(Type\_of\_Travel\_Class),int(Flight\_Distance),int(Inflight\_wifi\_service),

int(Departure\_Arrival\_time\_convenient),int(Ease\_of\_Online\_booking),int(Gate\_location),int(Food\_and\_drink),

int(Online\_boarding),int(Seat\_comfort),int(Inflight\_entertainment),int(On\_board\_service),int(Leg\_room\_service),

int(Baggage\_handling),int(Checkin\_service),int(Inflight\_service),int(Cleanliness),int(Departure\_Delay\_in\_Minutes),int(Arrival\_Delay\_in\_Minutes),int(Satisfaction)]]

#x=airline\_model.transform(t)

output =model.predict(t)

print(output)

return render\_template("submit.html", result = "The predicted profit is "+str(np.round(output[0])))

"""i = [x for x in request.form.values()]

f = [np.array(i)]

print(f)

output = model.predict(f)"""

"""@app.route('/predicts',methods =['GET','POST'])

def predicts():

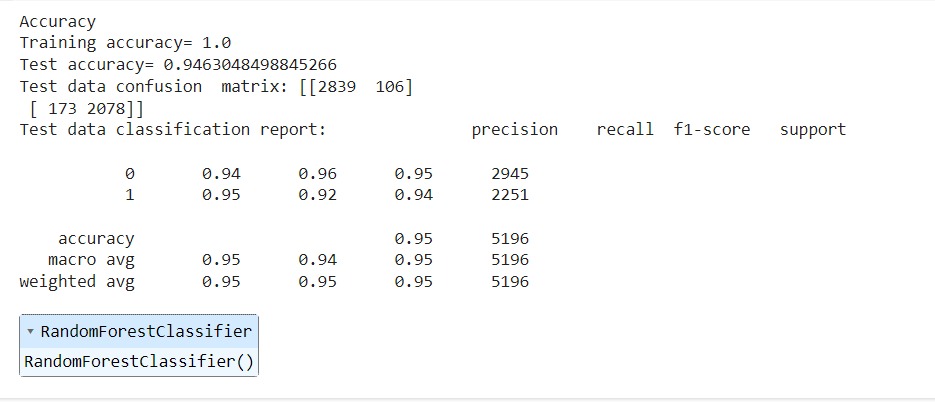
return render\_template('index.html', prediction\_text = 'Suitable drug type is {}'.format(prediction))"""

if \_name\_ == "\_main\_":

app.run(debug=True)

**8.PERFORMANCE TESTING**

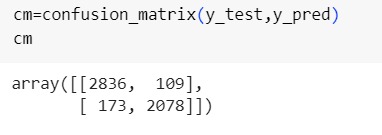
* 1. Performance Metrics



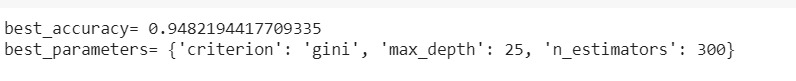
9.**RESULTS**

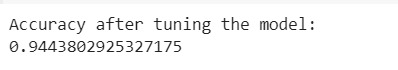
* 1. Output Screenshots

IPYNB File:



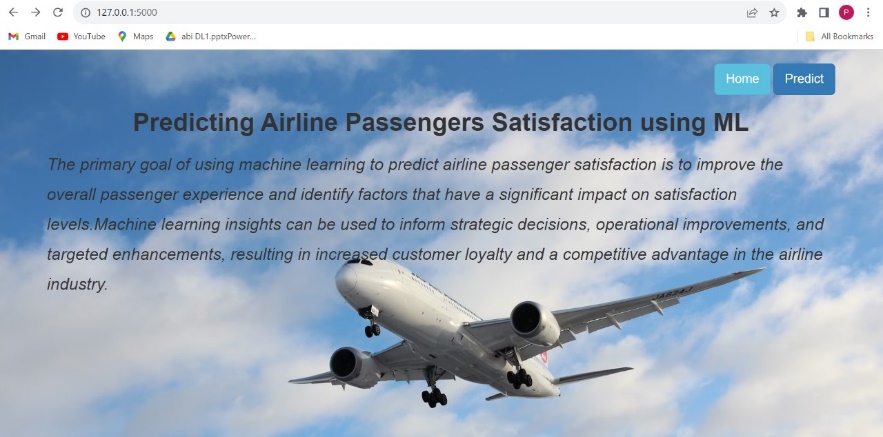




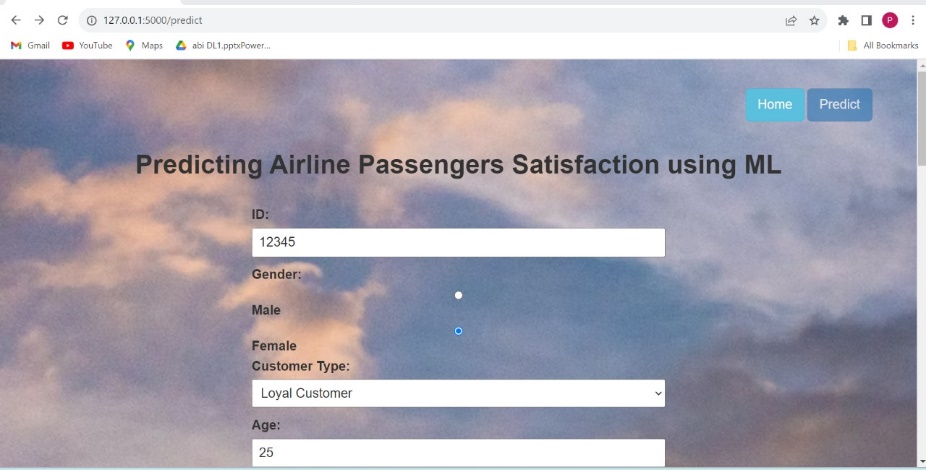


Flask:

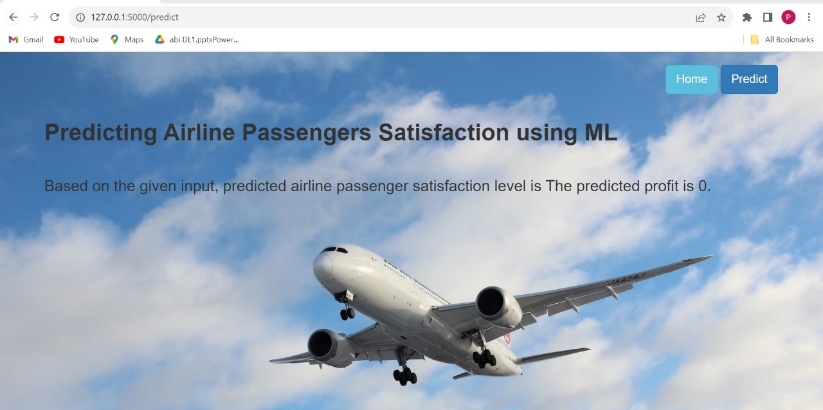
Home page



Predict page:



Submit/Output page:



**10.ADVANTAGES & DISADVANTAGES**

Advantages:

* Enhanced Customer Experience: models can analyse passenger preferences and behaviours, allowing airlines to personalize services and improve overall satisfaction.
* Continuous Improvement: Automated feedback analysis enables continuous improvement, as airlines can identify trends and areas for enhancement in real-time.
* Competitive Advantage: Airlines leveraging ML for passenger satisfaction gain a competitive edge by offering personalized services and efficient operations.

Disadvantages:

* Initial Implementation Costs: Implementing ML systems requires significant upfront investment in technology, training, and infrastructure.
* Complexity and Integration Challenges: Integrating ML solutions with existing airline systems may be complex, requiring careful planning and execution.
* Overreliance on Technology: Overreliance on ML predictions without human oversight may lead to misguided decisions or overlook nuances in passenger satisfaction.

**11.CONCLUSION**

Finally, incorporating Machine Learning (ML) to identify and improve airline passenger satisfaction represents a significant step toward a more efficient and personalized aviation experience. The potential benefits of ML in the aviation industry, such as operational optimizations, predictive insights, and improved customer service, highlight the transformative power of the technology. As the aviation industry navigates this technological evolution, careful consideration of both opportunities and challenges will be critical to ML's long-term success and positive impact on passenger satisfaction.

**12.FUTURE SCOPE**

The future potential of using Machine Learning (ML) to improve airline passenger satisfaction holds enormous promise for transformative advancements in the aviation sector. As technology advances, the emphasis is shifting toward more sophisticated personalization, in which ML algorithms can delve deeper into individual preferences and behaviours to provide tailored services throughout the travel journey. Predictive maintenance is set to play a critical role in optimizing aircraft operations by anticipating maintenance requirements and minimizing downtime.

**13. APPENDIX**

Source Code:

Source code for the IPYNB Notebook:

The steps in the IPYNV Notebook includes the

Importing the statements, Reading the data, Label Encoding, Sub plotting, Model Building, Saving the Model.

import numpy as npimport pandas as pdimport matplotlib.pyplot as pltimport seaborn as snsfrom sklearn.model\_selection import train\_test\_splitfrom sklearn.ensemble import RandomForestClassifier,GradientBoostingClassifierfrom sklearn.tree import DecisionTreeClassifierfrom sklearn.neighbors import KNeighborsClassifierfrom sklearn.metrics import f1\_scorefrom sklearn.metrics import classification\_report,confusion\_matriximport warningsimport picklefrom scipy import statswarnings.filterwarnings('ignore')plt.style.use('fivethirtyeight')df=pd.read\_csv('./test.csv')df.head()df.info()df.isnull().sum()df["Arrival Delay in Minutes"].fillna(df["Arrival Delay in Minutes"].mean(),inplace=True)df["Arrival Delay in Minutes"]df.isnull().sum()df['Gender'].unique()df['Customer Type'].unique()df['Type of Travel'].unique()df['Class'].unique()df['satisfaction'].unique()from sklearn.preprocessing import LabelEncoderle=LabelEncoder()df['Gender']=le.fit\_transform(df['Gender'])df['Customer Type']=le.fit\_transform(df['Customer Type'])df['Type of Travel']=le.fit\_transform(df['Type of Travel'])df['Class']=le.fit\_transform(df['Class'])df['satisfaction']=le.fit\_transform(df['satisfaction'])quant=df.quantile(q=[0.25,0.75],axis=0)quantdf['Customer Type']=np.where(df['Customer Type']>0.1,0,df['Customer Type'])df['Customer Type']=np.where(df['Customer Type']<0,0,df['Customer Type'])df['Flight Distance']=np.where(df['Flight Distance']>3700,1744,df['Flight Distance'])df['Flight Distance']=np.where(df['Flight Distance']<0,414,df['Flight Distance'])df['Checkin service']=np.where(df['Checkin service']>4.9,4.0,df['Checkin service'])df['Checkin service']=np.where(df['Checkin service']<2,3.0,df['Checkin service'])df['Departure Delay in Minutes']=np.where(df['Departure Delay in Minutes']>10,12,df['Departure Delay in Minutes'])df['Departure Delay in Minutes']=np.where(df['Departure Delay in Minutes']<0,3.0,df['Departure Delay in Minutes'])df['Arrival Delay in Minutes']=np.where(df['Arrival Delay in Minutes']>10,13,df['Arrival Delay in Minutes'])df['Arrival Delay in Minutes']=np.where(df['Arrival Delay in Minutes']<0,3.0,df['Arrival Delay in Minutes'])df.describe()plt.figure(figsize=(20,5))plt.subplot(121)sns.distplot(df['Age'],color='r')plt.subplot(122)sns.distplot(df['Cleanliness'])plt.show()sns.countplot(df['satisfaction'])plt.figure(figsize=(12, 5))plt.subplot(121)sns.countplot(x='Gender', hue='satisfaction', data=df)plt.title('Count of Passenger Satisfaction by Gender')plt.show()x=df.iloc[:,:-1]y=df.iloc[:,-1]from sklearn.model\_selection import train\_test\_splitx\_train,x\_test,y\_train,y\_test=train\_test\_split(x,df['satisfaction'],test\_size=0.2,random\_state=42)x\_train.shape,y\_train.shapex\_test.shape,y\_test.shape# !pip install joblibimport joblibdef RF(x\_train,y\_train,x\_test,y\_test, save\_model=False): reg4=RandomForestClassifier() reg4.fit(x\_train,y\_train) print('Accuracy') print('Training accuracy=',reg4.score(x\_train,y\_train)) print('Test accuracy=',reg4.score(x\_test,y\_test)) y\_test\_pred2=reg4.predict(x\_test) print('Test data confusion matrix:',confusion\_matrix(y\_test,y\_test\_pred2)) print('Test data classification report:',classification\_report(y\_test,y\_test\_pred2)) if save\_model: joblib.dump(reg4, 'airline\_model') # Save the model to a file return reg4#Random ForestAirline2 = RF(x\_train, y\_train, x\_test, y\_test,save\_model=True)Airline2.predict([[1,2,1,3,1,0,7,5,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]])RF(x\_train,y\_train,x\_test,y\_test)from sklearn.model\_selection import GridSearchCVfrom sklearn.ensemble import RandomForestClassifierforest=RandomForestClassifier()parameters={ 'n\_estimators':[25,50,200,300], 'criterion':['gini','entropy'], 'max\_depth':[14,20,25,30]}df.head()grid\_search=GridSearchCV(estimator=forest, param\_grid=parameters, scoring='accuracy', cv=10, n\_jobs=-1)grid\_search=grid\_search.fit(x\_train,y\_train)print('best\_accuracy=',grid\_search.best\_score\_)print('best\_parameters=',grid\_search.best\_params\_)random=RandomForestClassifier(criterion='gini',max\_depth=30,n\_estimators=200)random.fit(x\_train,y\_train)y\_pred=random.predict(x\_test)cm=confusion\_matrix(y\_test,y\_pred)cmfrom sklearn.metrics import accuracy\_scoreacc=accuracy\_score(y\_test,y\_pred)print('Accuracy after tuning the model:')accimport picklewith open("./AirlinePassengers.pkl", "wb") as f: pickle.dump(random, f)Airline = pickle.load(open("./AirlinePassengers.pkl", "rb"))GitHub & Project Demo Link:

<https://github.com/smartinternz02/SI-GuidedProject-612235-1698851888>