MACHINE LEARNING

PREDICTING AIRLINE PASSENGER SATISFACTION

By
Wissal Talbi
Soulayma Mansouri
Wejdene Bedoui
Emna Hamzaoui

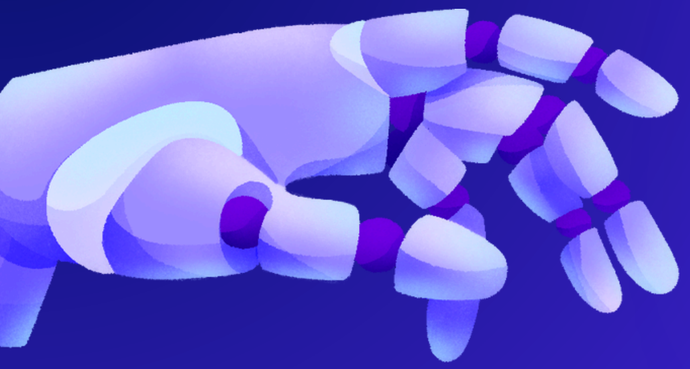
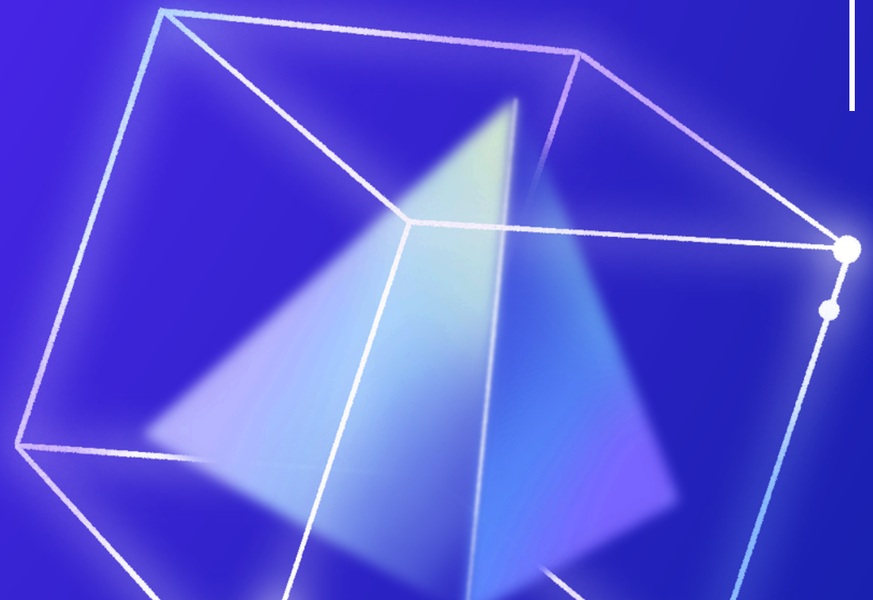


TABLE OF CONTENTS



INTRODUCTION

In today's highly competitive aviation industry, customer satisfaction is a critical factor in maintaining loyalty and driving business growth. Airlines must identify and address key drivers of passenger satisfaction to enhance the overall travel experience. The challenge lies in leveraging data analytics to accurately predict passenger satisfaction and uncover actionable insights.

Significance of the Project:

- Helps airlines prioritize improvements in service and operations.
- Supports data-driven decision-making to enhance customer retention.
- Contributes to long-term competitiveness in the aviation industry.

DATASET DESCRIPTION

Source

Dataset: Airline Passenger Satisfaction

Key Features/Variables

Demographics: Gender, Age, Travel Class, Customer

Type Flight Information: Flight Distance, Type of Travel, In-flight Service Service

Ratings: Seat Comfort, Food and Beverage, Cleanliness, Online Check-in, Gate Location, In-flight Entertainment, ...

Delays: Arrival Delay in Minutes, Departure Delay in Minutes

Dataset Size

Total Records: 129,880 passengers

Features: 25 columns (24 predictors + 1 target variable)

Target Variable

Satisfaction: Indicates whether the passenger was satisfied (Satisfied) or not (Neutral or Dissatisfied).

DATA CLEANING AND PREPROCESSING

- Checking For Null Values

Checking For Null Values

```
df.isnull().sum()
```

✓ 2.3s

```
Unnamed: 0                          0
id                                  0
Gender                              0
Customer Type                       0
Age                                 0
Type of Travel                      0
Class                               0
Flight Distance                     0
Inflight wifi service               0
Departure/Arrival time convenient   0
Ease of Online booking              0
Gate location                       0
Food and drink                      0
Online boarding                     0
Seat comfort                        0
Inflight entertainment              0
On-board service                    0
Leg room service                    0
```

DATA CLEANING AND PREPROCESSING

- Feature Engineering

Getting Rid of the object-type columns with Label Encoder

Generate | + Code | + Markdown

```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
```

[9] ✓ 36.2s Python

Extracting of features of type object

```
encoded_need=[]
for i in data_cleaned.columns:
    if data_cleaned[i].dtype=='object':
        encoded_need.append(i)
```

[10] ✓ 0.0s Python

```
for i in encoded_need:
    data_cleaned[i]=label_encoder.fit_transform(data_cleaned[i])
```

[11] ✓ 0.2s Python

```
data_cleaned.head()
```

[12] ✓ 0.1s Python

DATA CLEANING AND PREPROCESSING

- Data Splitting

Splitting The Data

```
[20]  from sklearn.model_selection import train_test_split
      ✓ 2.1s
```

```
[21]  target=data_cleaned.iloc[:,-1]
      X_train,X_test,y_train,y_test=train_test_split(features,target,test_size=0.2,random_state=42)
      print(f"Shape of training set is : {X_train.shape} and test set is :{X_test.shape}" )
      ✓ 0.0s
```

```
...   Shape of training set is : (83123, 22) and test set is :(20781, 22)
```

MACHINE LEARNING APPROACH

Machine Learning Models Used

01

Logistic Regression

Establishes a baseline; explains linear relationships clearly.

02

K-Nearest Neighbors (KNN)

Captures patterns in similar passenger groups; models non-linear trends.

03

Decision Tree

Simple and interpretable; identifies key factors driving satisfaction.

04

XGBoost

High accuracy; handles complex patterns and prevents overfitting.

MACHINE LEARNING APPROACH

Implementing XGBOOST

```
import xgboost as xgb
from sklearn.metrics import accuracy_score


# Create DMatrix for XGBoost
dtrain = xgb.DMatrix(X_train, label=y_train)
dtest = xgb.DMatrix(X_test, label=y_test)

# Define XGBoost parameters
params = {
    "objective": "binary:logistic",
    "eval_metric": "logloss",
    "eta": 0.1,
    "max_depth": 6,
    "seed": 42
}

# Train the model
num_round = 100
bst = xgb.train(params, dtrain, num_round)

# Predict on test data
y_pred_prob = bst.predict(dtest)
y_pred = [1 if prob > 0.5 else 0 for prob in y_pred_prob]

# Evaluate the model
```

[29]

RESULTS(ACCURACY)

01

Logistic Regression Algorithm

```
from sklearn.metrics import ConfusionMatrixDisplay,confusion_matrix
from sklearn.metrics import accuracy_score
y_pred_lr=model_lr.predict(X_test)
accuracy=accuracy_score(y_pred_lr,y_test)
print(f"Accuracy of Logistic Regression is : {round(accuracy*100,2)} % .")
```

23] ✓ 0.0s

```
Accuracy of Logistic Regression is : 86.72 % .
```

RESULTS(ACCURACY)

02

K-Nearest Neighbors (KNN)

```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=5)
#fit the model with trained data
knn.fit(X_train,y_train)

#Predict the accuracy of the model
y_pred_knn=knn.predict(X_test)

accuracy=accuracy_score(y_pred_knn,y_test)
print(f"Accuracy of KNN is : {round(accuracy*100,2)} % .")
```

✓ 4.8s

```
Accuracy of KNN is : 74.76 % .
```

RESULTS(ACCURACY)

03

Decision Tree

```
from sklearn.tree import DecisionTreeClassifier

DecisionTree=DecisionTreeClassifier(criterion='gini', splitter='best',max_features=6,random_state=1234)
#fit the model with trained data
DecisionTree.fit(X_train,y_train)

#Predict the accuracy of the model
y_pred_dt=DecisionTree.predict(X_test)

accuracy=accuracy_score(y_pred_dt,y_test)
print(f"Accuracy of Decision Tree is : {round(accuracy*100,2)} % .")
```

[27] ✓ 0.6s

```
Accuracy of Decision Tree is : 93.75 % .
```

RESULTS(ACCURACY)

04 XGBoost

```
dtest = xgb.DMatrix(X_test, label=y_test)

# Define XGBoost parameters
params = {
    "objective": "binary:logistic",
    "eval_metric": "logloss",
    "eta": 0.1,
    "max_depth": 6,
    "seed": 42
}

# Train the model
num_round = 100
bst = xgb.train(params, dtrain, num_round)

# Predict on test data
y_pred_prob = bst.predict(dtest)
y_pred = [1 if prob > 0.5 else 0 for prob in y_pred_prob]

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
```

✓ 1.3s

```
Accuracy: 0.96
```

DEPLOYMENT

REFERENCES

ML-COSTUMER-SATISFACTIONPRIVATE

kaggle

AIRLINE PASSENGER SATISFACTION DATASET

WE USED FASTAPI TO DEPLOY OUR MODEL

CONCLUSION

XGBoost outperformed other models with the highest accuracy.

Decision Tree provided interpretable insights into key factors influencing satisfaction.

Logistic Regression and KNN served as effective benchmarks for comparison.

THANK YOU!