In-depth Analysis of Airline Customer Satisfaction

Invistico Airlines

Hector Enriquez, Ifa Afariogun

Introduction:

This project's dataset is used to discuss and go into detail regarding Invistico Airline customer satisfaction. With the use of this dataset, we were able to gain more insight about different amenities offered and how satisfied their customers are with them. Given the specifics of the other parameter values, the primary goal of this dataset is to forecast if a prospective customer will be satisfied with their service. In order to increase customer satisfaction, the airline must determine which aspects of their services need to be prioritized.

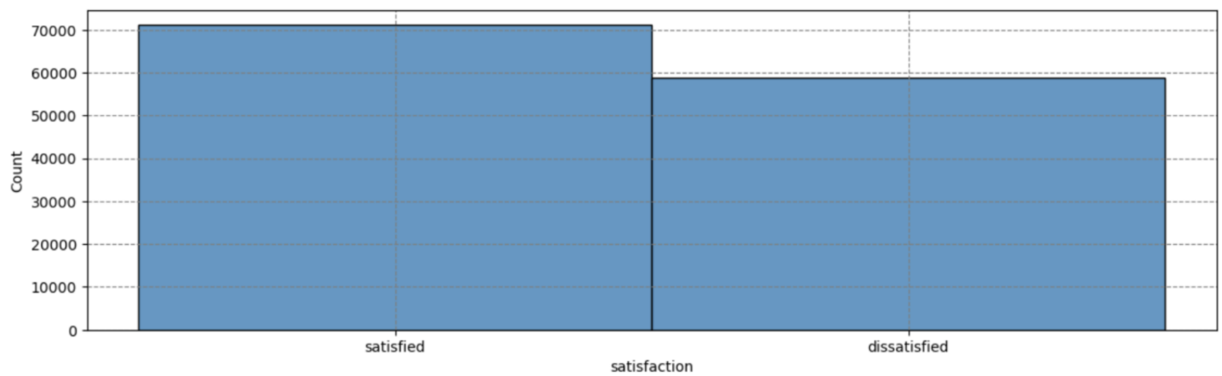
	satisfaction	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Seat comfort	Departure/Arrival time convenient	Food and drink	...	Online support	Ease of Online booking	bo serv
0	satisfied	Female	Loyal Customer	65	Personal Travel	Eco	265	0	0	0	...	2	3	
1	satisfied	Male	Loyal Customer	47	Personal Travel	Business	2464	0	0	0	...	2	3	
2	satisfied	Female	Loyal Customer	15	Personal Travel	Eco	2138	0	0	0	...	2	2	
3	satisfied	Female	Loyal Customer	60	Personal Travel	Eco	623	0	0	0	...	3	1	
4	satisfied	Female	Loyal Customer	70	Personal Travel	Eco	354	0	0	0	...	4	2	

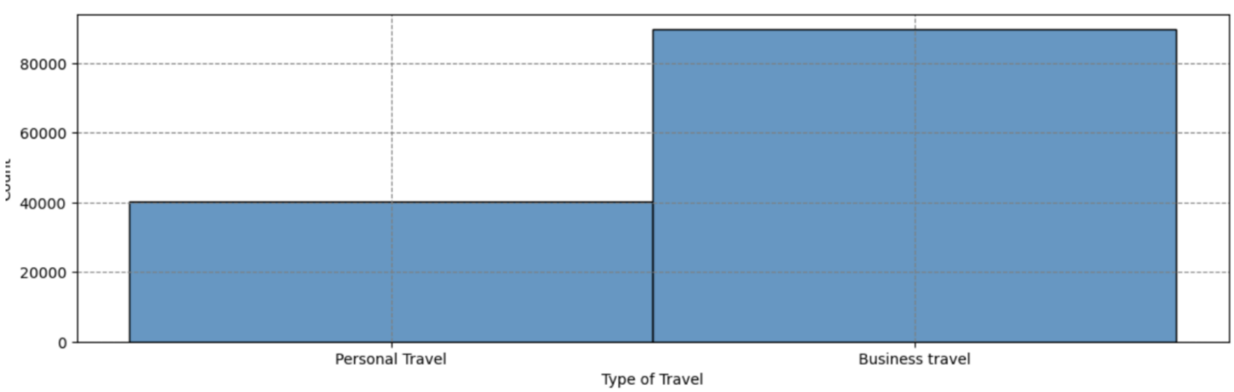
5 rows × 23 columns

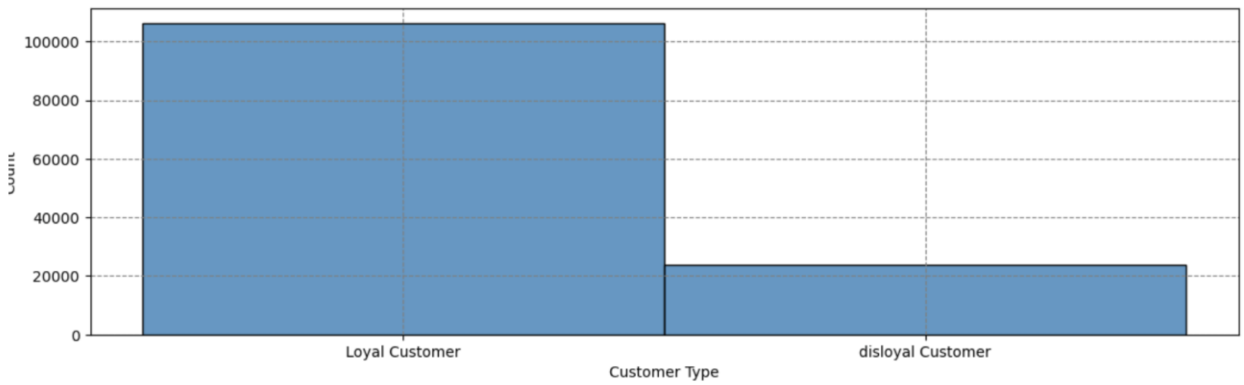
We first found the dataset contains 129,880 rows and 23 columns. The features include: satisfaction levels, gender, age, flight details, and seat comfort.

Exploratory Data Analysis (EDA):

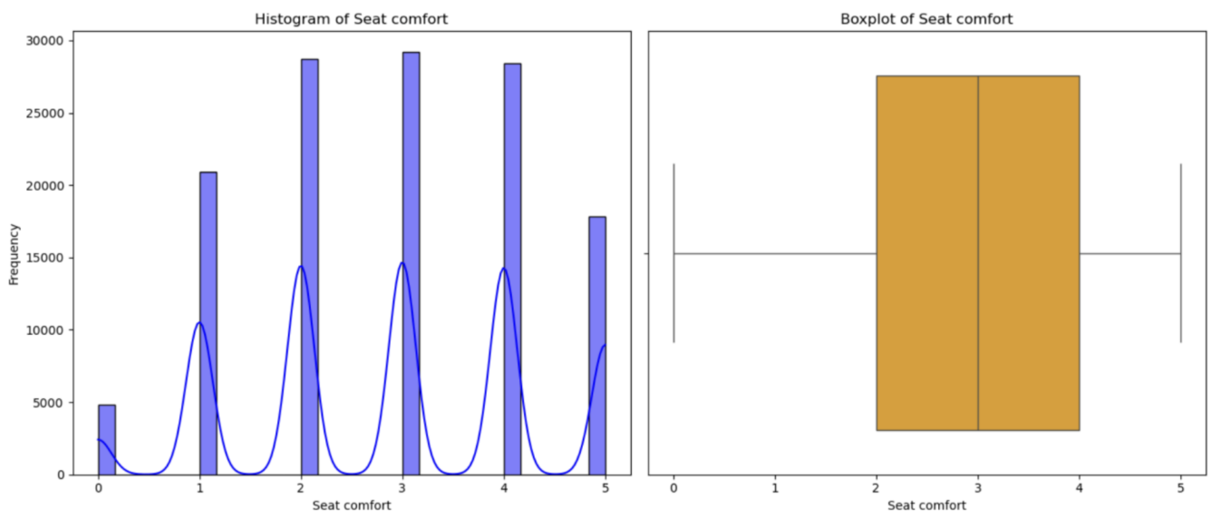
From the information that we discovered, we then started with the univariate analysis. Differentiating and separating the data allowed us to get a clearer insight on each feature in the set.

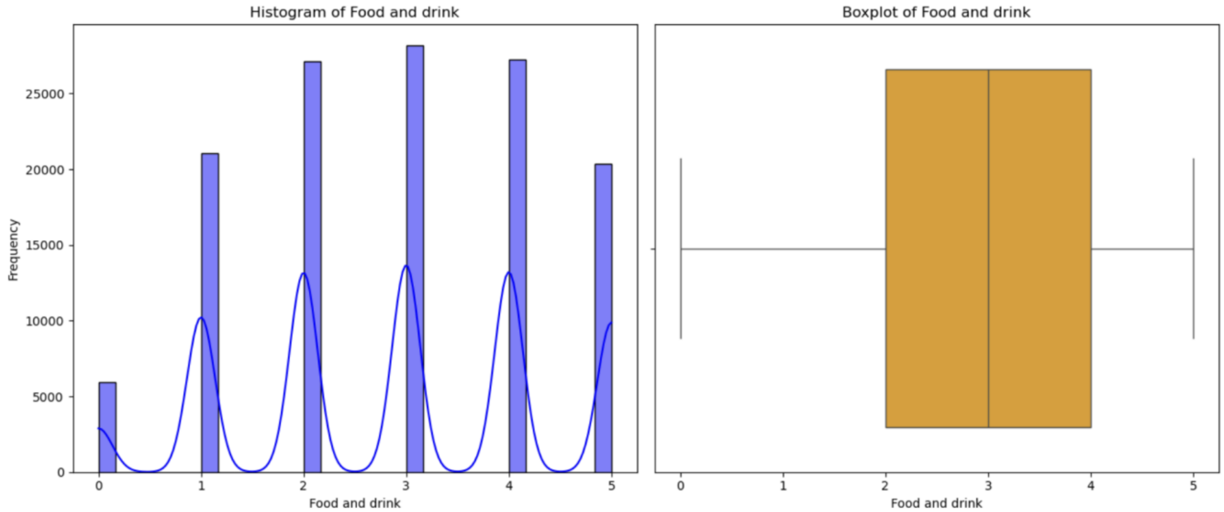


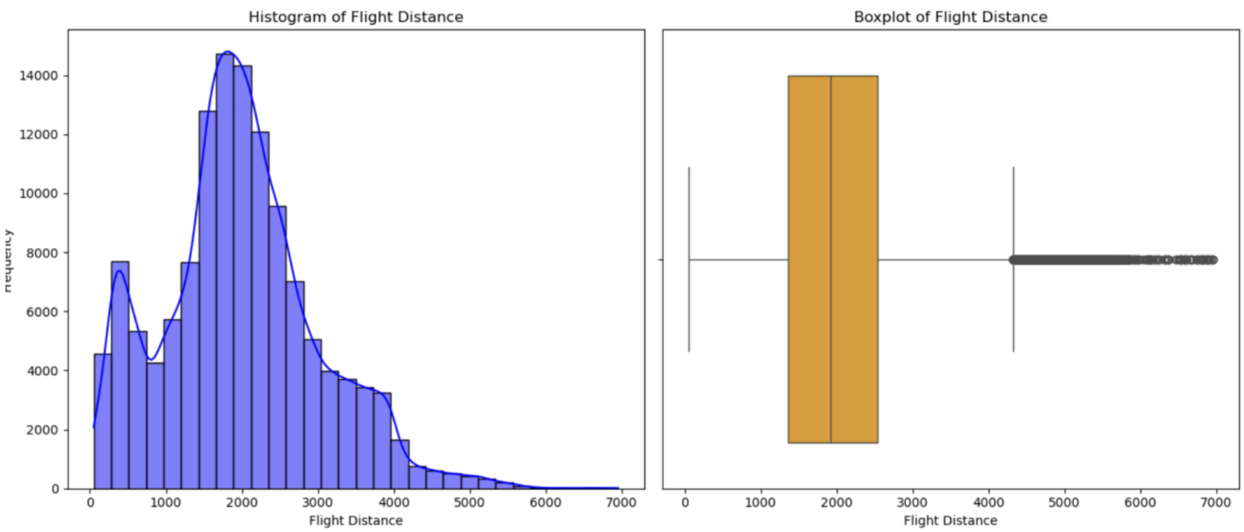




An important observation is that the airline has almost 50/50 satisfaction and dissatisfaction. A majority of their customers are loyal customers. From the charts we gathered, most of the individuals flying on the airline are going for business purposes.

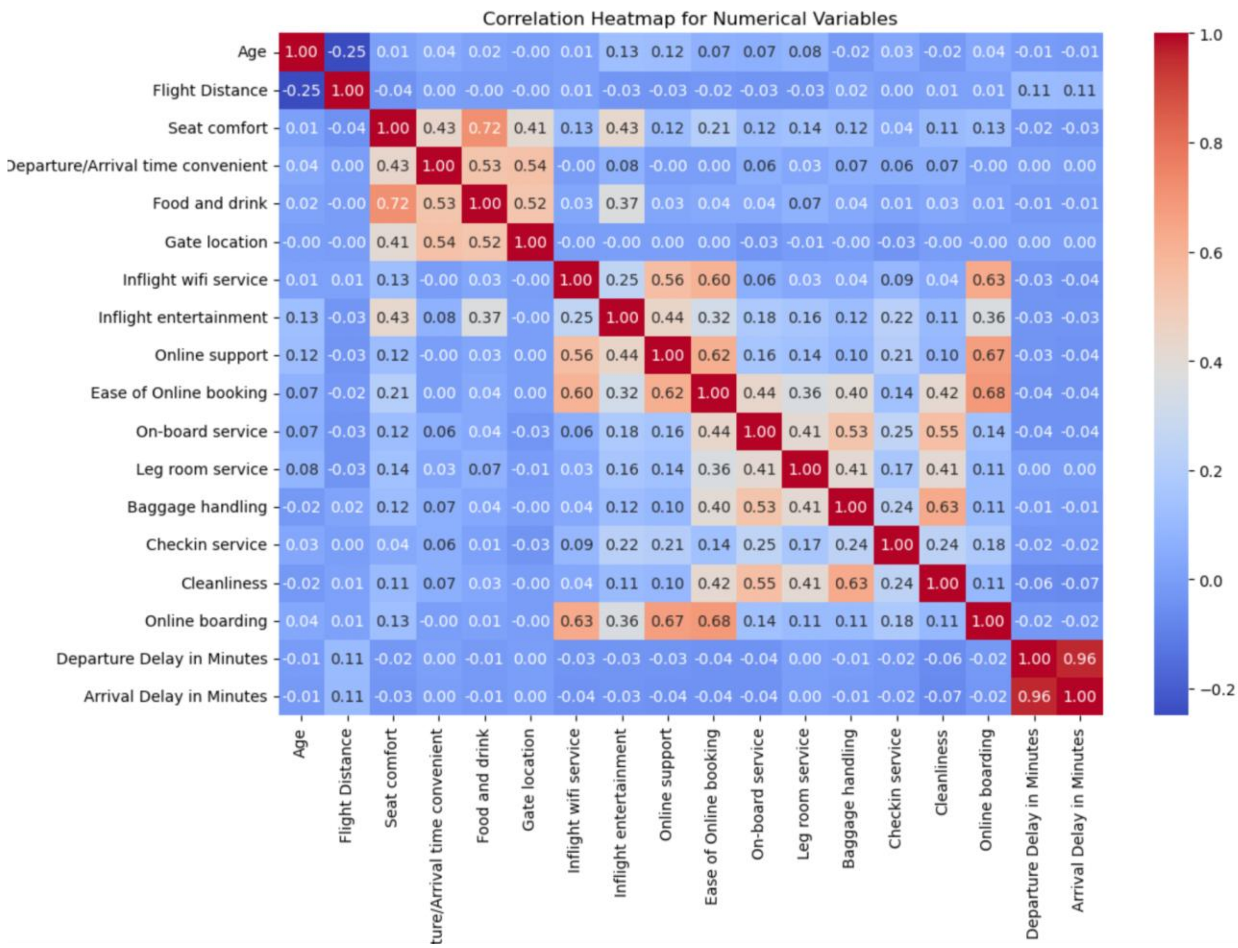




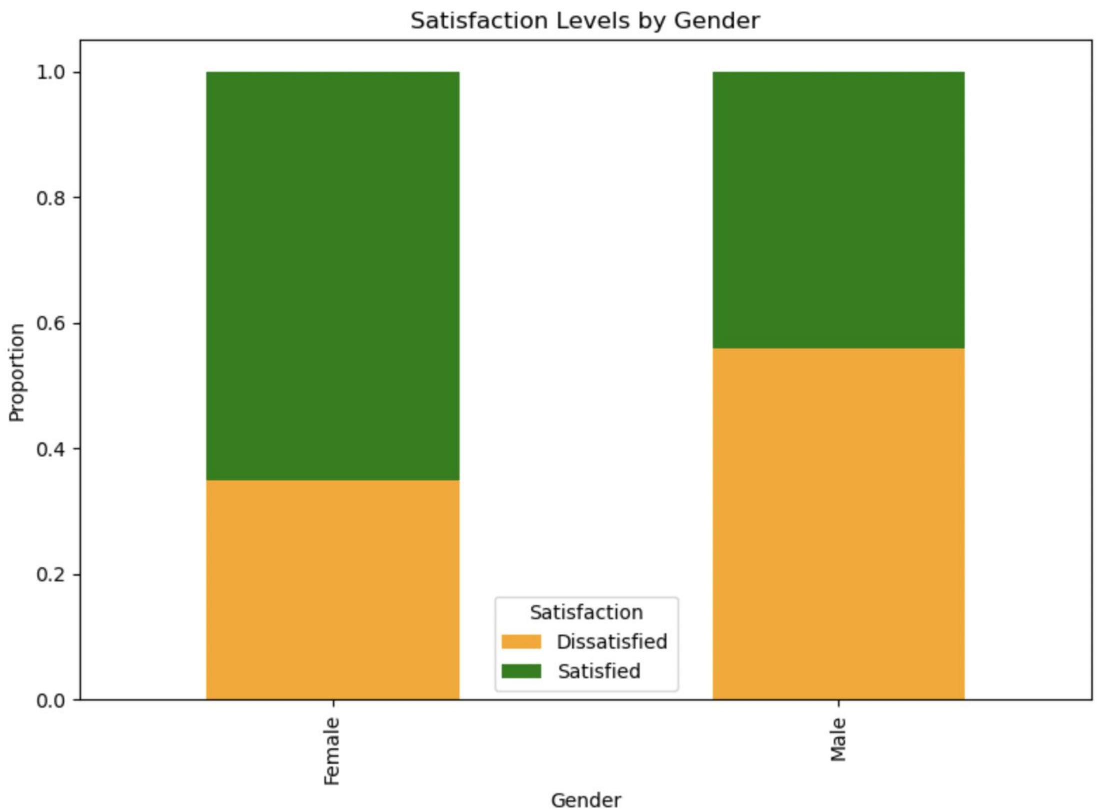


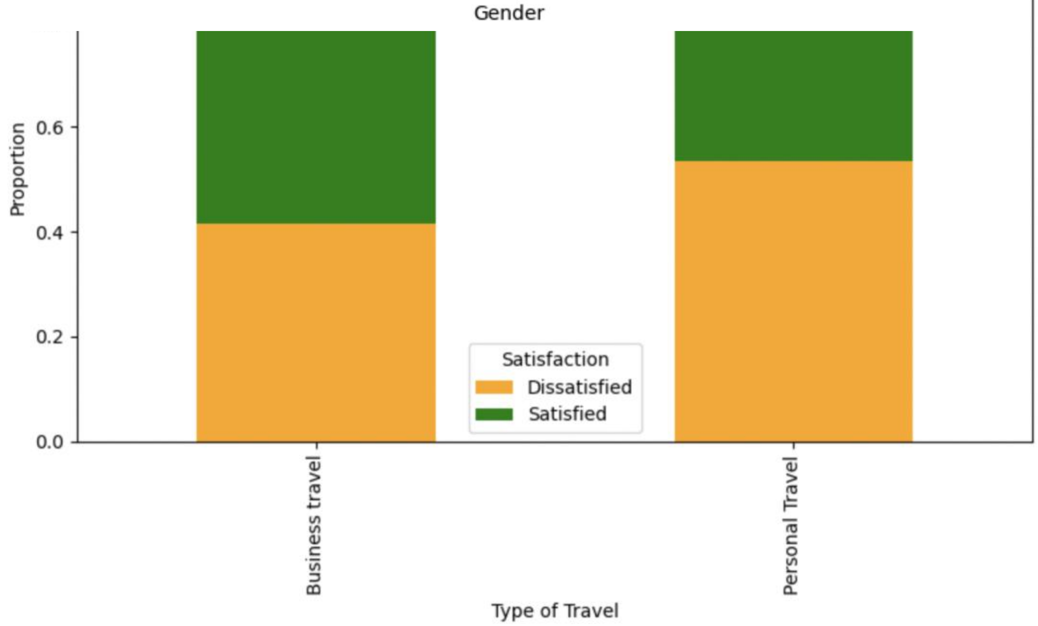
The satisfaction ratings for seat comfort, food, and drink range between scores 3-5, meaning the services are meeting or exceeding expectation levels. The distance traveled on these flights are not typically far. This is very important because shorter flights have simpler amenities than longer flights. For example, traveling 10 hours can result in having 2 meals served and more leg room as opposed to flying for 4 hours.

Heatmap Observations:



Ease of Online Booking and Online Boarding (0.68) show a relationship between customer satisfaction with the boarding procedure and the ease of booking. The moderate correlations between seat comfort, food and drink, and in-flight entertainment (0.4–0.7) imply that there is a relationship between in-flight service pleasure. Departure/Arrival Time Convenient and other service attributes are moderately correlated with gate location.


Satisfaction Levels by Gender
Proportion
1.0
0.8
0.6
0.4
0.2
0.0
Satisfaction
Dissatisfied
Satisfied
Female
Male
Gender


Proportion
0.6
0.4
0.2
0.0
Satisfaction
Dissatisfied
Satisfied
Business travel
Personal Travel
Type of Travel

Compared to female passengers, male passengers are more likely to be dissatisfied. It might be necessary for airlines to evaluate the service experiences and expectations of male clients. In contrast to personal travelers, business travelers are more satisfied. Personal travelers could be less satisfied because of factors like comfort or cost.

Data Preparation for Logistic Regression:

```
# Convert satisfaction to binary values
data_encoded['satisfaction'] = data_encoded['satisfaction'].apply(lambda x: 1 if x == 'satisfied' else 0)

# We ran  running into issues regarding the conversion of the categorical variables into 0 and 1s  so I de
# and this was able to Ensure that all dummy variables are converted to numeric (0/1)
data_encoded = data_encoded.apply(lambda col: col.astype(int) if col.dtype == 'bool' else col)
```

```
# Verifying that the first few rows of the predictors (X) after corrections
X = data_encoded.drop('satisfaction', axis=1)
y = data_encoded['satisfaction']

X.head()
```

In order to train and assess a logistic regression model that predicts customer satisfaction (represented by the target variable satisfaction), the code provided is a component of a machine learning pipeline. To prepare them for modeling, categorical variables (such as "Gender," "Customer Type," etc.) are first transformed into dummy variables by one-hot encoding. After that, the dataset is divided into training and testing sets, with the stratify parameter being used to make sure that the target variable (satisfaction) is equal in both groups. The training data is used to train a logistic regression model. The following functions (model_performance_classification_LR and make_confusion_matrix_LR) compute metrics such as accuracy, recall, precision, and F1 score in order to evaluate the model's performance at different classification thresholds. Understanding false positives, false negatives, and other misclassifications is made easier by the confusion matrix function, which visualizes the model's classification results. These functions help identify the ideal threshold for forecasting customer satisfaction and enable a thorough assessment of the model's performance across various thresholds.

```
# fitting the Logistic regression model
log_reg = LogisticRegression()
log_reg.fit(X_train, y_train)
```

Using the supplied training data (log_reg.fit(X_train, y_train)), the code trains a Logistic Regression model after initializing it (log_reg = LogisticRegression()). In order to generate predictions on fresh data, this procedure enables the model to understand the connection between the input features (X_train) and the target variable (y_train).

Observations:

Training performance:

	Threshold	Accuracy	Recall	Precision	F1
0	0.5	0.786	0.85352	0.77732	0.81364

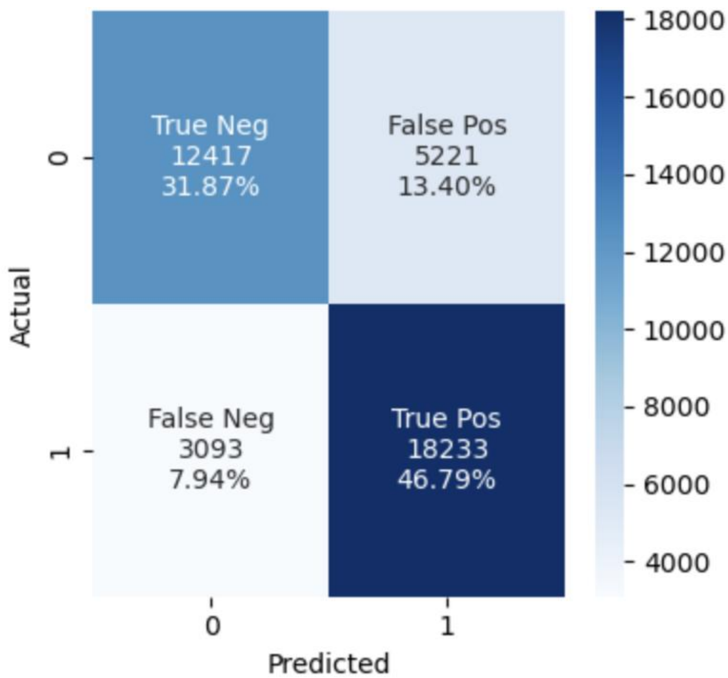
Testing performance:

	Threshold	Accuracy	Recall	Precision	F1
0	0.5	0.786624	0.854966	0.777394	0.814337

For the training performance, the accuracy was 78.6% with a recall percentage of 85.3%, a 77.7% precision, and F-1 Score of 81.4%. The testing performance had the same result except with a recall percentage of 85.6%.

A large percentage of happy consumers are accurately identified by the model. This is important since it might be expensive to misclassify satisfied clients. Furthermore, the precision

(77.7%) shows that the model still does an adequate job of detecting true positives even though there are some false positives. Lastly, the model generalizes well without overfitting, as evidenced by the accuracy and F1-scores staying constant during training and testing.



True Positives (TP) – 46.79%: Customers who were actually satisfied and correctly classified as satisfied.

True Negatives (TN) – 31.87%: Customers who were actually dissatisfied and correctly identified as dissatisfied.

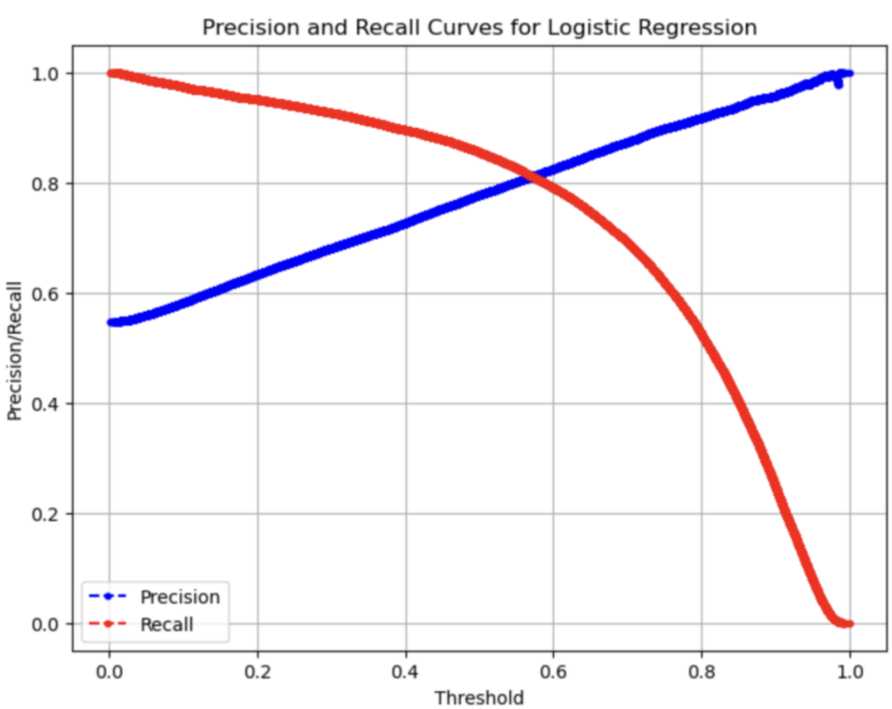
False Positives (FP) – 13.40%: Customers who were actually dissatisfied but were misclassified as satisfied.

False Negatives (FN) – 7.94%: Customers who were actually satisfied but were misclassified as dissatisfied.

Overall, the model performed well in identifying the true positives and negatives. The percentages of false negatives and false positives were extremely low, meaning a majority of the individuals were classified correctly.

In terms of business risk, false positives are more significant. This may conceal deeper problems in the services. Customers who are unhappy may not use the airline again or refer it to others. Customer turnover may result from management's failure to identify issue areas. Because it guarantees that unhappy consumers are appropriately recognized and targeted for improvement initiatives, lowering False Positives is essential. However, even if it's less dangerous, it could lead to resources being misdirected toward clients who are already happy. However, as it has no direct effect on dissatisfaction rates, this danger is smaller. Our goal would be to reduce false positives to ensure a lower chance of a business risk.

<u>Tuning Model:</u>



Precision (blue curve): As the threshold rises, so does precision. This is fair since greater tolerance renders the model more stringent, resulting in fewer customers being classified as "satisfied" yet guaranteeing more accurate predictions. Recall (red curve): As the threshold rises, recall falls. This happens when the model misses more real satisfied consumers when the threshold is raised since it produces less "satisfied" forecasts. Trade-off: Because the model captures the majority of satisfied customers but contains a large number of false positives, recall is high but precision is low at lower thresholds. Because the model becomes too conservative at higher thresholds, recall decreases while precision increases. Since our goal is to minimize false positives, it would make sense to apply a threshold around 0.6-0.7, in order to keep precision high and recall tolerable, as shown below.

```
Training performance:
    Threshold  Accuracy    Recall  Precision        F1
0        0.57   0.79403  0.813428   0.810854  0.812139
Testing performance:
    Threshold  Accuracy   Recall  Precision        F1
0        0.57  0.793296  0.81192   0.810703  0.811311
```

```
Training performance:
    Threshold  Accuracy   Recall  Precision        F1
0        0.65  0.788508  0.74685    0.84859  0.794476
Testing performance:
    Threshold  Accuracy    Recall  Precision        F1
0        0.65  0.789267  0.746366   0.850313  0.794956
```

Based on these results, we believed a threshold of 0.65 would be most appropriate. A threshold of 0.65 gives us an 85% precision, which targets our goal. Recall, however, would decrease to 74.6%, meaning a big portion of satisfied customers are still categorized correctly.

Avoiding false positives has a significant effect than a higher recall percentage. If customers were to be under an inaccurate category, the business may not have the proper abilities to address certain problems and improve on them.

Decision Tree Model:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report

# Libraries to build decision tree classifier
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree


# To tune different models
from sklearn.model_selection import GridSearchCV

dTree = DecisionTreeClassifier(criterion="gini", random_state=1)
dTree.fit(X_train, y_train)
```

We started off using this code to train a decision tree classifier using (X_train, Y_train). The decision tree model will identify the target variable (in this case, y_train) using the x_train features. "Gini" is used to assess the splits at each node. The results are reproducible since random_state=1. The significance of this step is to allow the decision tree to be compared with other models for performance.

```
Decision Tree - Training Performance:
    Accuracy  Recall  Precision   F1
0        1.0     1.0        1.0  1.0

Decision Tree - Test Performance:
    Accuracy    Recall  Precision        F1
0  0.937481  0.945559   0.940532  0.943039
```

True Negatives (41,155 - 45.27%): The model accurately predicted "Not Satisfied" for 45.27% of the data. This shows that the model is performing well when classifying dissatisfied customers.

True Positives (49,761 - 54.73%): The model accurately predicted "Satisfied" for 54.73% of the data, meaning the majority of satisfied customers were classified correctly.

As our goal is to reduce false positives, we want to ensure that the model mirrors that. We will do that by:

1. The decision tree's maximum depth restricts how deep it can go, and we want to guarantee sure the tree can overfit the data without becoming too shallow to miss significant patterns.
2. We want to establish that there are at least five samples, as the minimum sample leaf specifies the minimum amount of samples.
3. Since the maximum leaf node is the total number of leaf nodes in the tree, we will set numbers like 10, 15, and 30 to balance the tree's complexity.

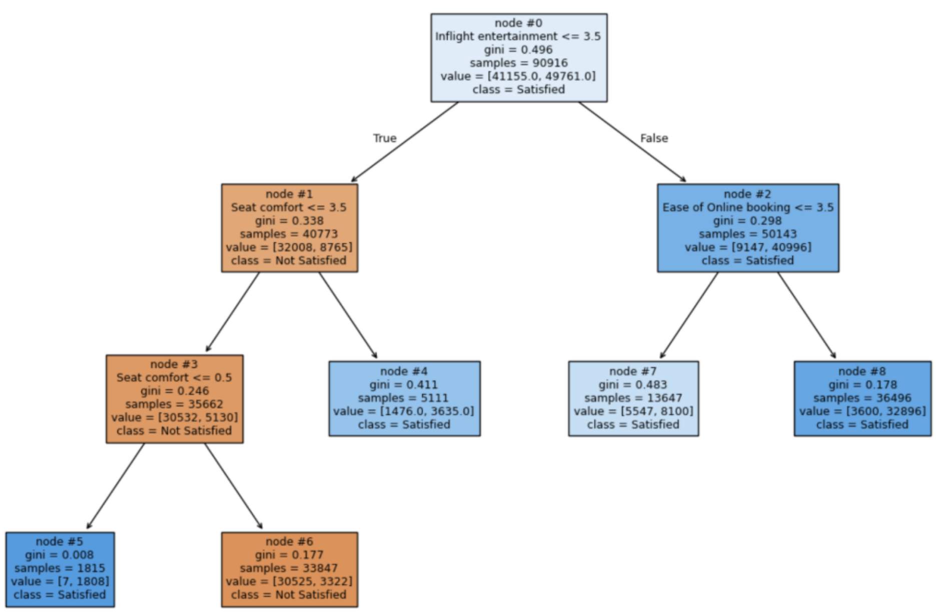
4. Only meaningful splits are produced due to the minimum impurity drop.

```
# Choose the type of classifier.
dTree_tuned = DecisionTreeClassifier(criterion='gini', random_state=1)

# Grid of parameters to choose from

parameters = {
    "max_depth": np.arange(3, 15, 2),
    "min_samples_leaf": [2, 5, 10, 15, 20, 25],
    "max_leaf_nodes": [5, 10, 15, 20, 30, 50],
    "min_impurity_decrease": [0.0001, 0.001, 0.01, 0.1],
}

# Scoring metric - F1 Score (Balanced Precision & Recall)
f1_scorer = make_scorer(f1_score)

# Run the grid search
grid_obj = GridSearchCV(dTree_tuned, parameters, scoring=acc_scorer, cv=5)
grid_obj = grid_obj.fit(X_train, y_train)

# Set the clf to the best combination of parameters
dTree_tuned = grid_obj.best_estimator_

# Fit the best algorithm to the data.
dTree_tuned.fit(X_train, y_train)

# Output the best parameters
print("Best Parameters:", grid_obj.best_params_)
```

```
# Calculating different metrics
dTree_tuned_model_train_perf = model_performance_classification(
    dTree_tuned, X_train, y_train
)
print("Training performance:\n", dTree_tuned_model_train_perf)
dTree_tuned_model_test_perf = model_performance_classification(dTree_tuned, X_test, y_test)
print("Testing performance:\n", dTree_tuned_model_test_perf)
# Creating confusion matrix
create_confusion_matrix(dTree_tuned, X_test, y_test, figsize=(4, 3))
```

The goal of the code is to find the best combination of parameters that would give us the best performance for the model. The function GridSearchCV tests the possible combinations using the specific parameters, and studies the model based on F-1 score. Next, the tuned DecisionTreeClassifier is evaluated in both the training and the testing datasets. This is executed by using model_performace_classification. It helps us determine how well the model generalizes to new data and fits the training set. In order to graphically depict the model's classification performance, a confusion matrix is also generated using the create_confusion_matrix function. This matrix displays the counts of true positives, false positives, true negatives, and false negatives. The confusion matrix is a useful technique for deciding how the model differentiates between various categories.

Observations:



Key Takeaways

1. Inflight Entertainment is the most important factor:
    a. If inflight entertainment is poor, customers tend to be dissatisfied.
2. Seat Comfort matters for customers already unhappy with entertainment:
    a. Very low ratings for seat comfort lead to almost universal dissatisfaction.
3. Ease of Online Booking plays a big role when inflight entertainment is good:
    a. If booking is easy, customers are much more likely to be satisfied.

Business Recommendations and Conclusion:

The Decision Tree model is the best option for forecasting customer happiness since it offers a better level of precision, ensuring that we are accurately identifying unhappy clients. This will enable the organization to adjust more quickly and absorb valuable input. Furthermore, its visual interpretability provides precise, useful guidelines for commercial decision-making. Since seat comfort and inflight entertainment do not meet expectations, the best idea is to prioritize and improve on them. This can be achieved by upgrading seating arrangements and improving the quality of in flight entertainment (especially on longer flights).

Most of the flights taken have been short distances for business purposes, meaning they have different amenities. To increase their satisfaction, offering priority seating, lounge access, and better meal options. Bundling them into a premium package could also increase loyalty from business flyers. Additionally, the split between satisfied and dissatisfied customers is relatively even. Enabling robust feedback mechanisms can also help improve the airline by using real time feedback systems and customer surveys.

In conclusion, boosting important areas of Invistico Airlines' service—like seat comfort, in-flight entertainment, and the simplicity of online booking—is essential to raising consumer happiness. These elements are essential for client fulfilment, according to data-driven insights from the investigation, and giving them top priority will probably have the most effect on the total customer experience. Additionally, the airline can lower turnover, foster loyalty, and maximize its resources by concentrating on lowering false positives in satisfaction predictions and customizing services for particular client segments (such as male and business travelers). Invistico Airlines can boost customer satisfaction, operational effectiveness, and spur long-term growth by strategically enhancing the aspects that matter most to their clients

Resources:

1. https://www.sciencedirect.com/science/article/abs/pii/S0952197624007553
2. https://www.mdpi.com/2071-1050/15/2/1320
3. https://www.kaggle.com/code/iamsouravbanerjee/shopping-trends-unveiled-eda-for-beginners
4. https://www.ibm.com/topics/exploratory-data-analysis
5. https://www.geeksforgeeks.org/what-is-exploratory-data-analysis/
6. https://www2.deloitte.com/us/en/pages/consumer-business/articles/rising-above-the-clouds-aviation-transportation-loyalty.html