FINAL PROJECT

AIRLINE PASSENGER SATISFACTION

CWID: A20539793

Name: Saurabh Sonawale

Institution: ILLINOIS INSTITUTE OF TECHNOLOGY

CS 584 : MACHINE LEARNING

Instructor Name: Steve Avsec

Due Date: 04/29/24

INDEX

1. ABSTRACT

1.1 ABSTRACT

The primary business challenge addressed by the dataset under investigation is to effectively classify passengers based on their satisfaction with their flight experience. This classification is crucial as it can help airlines tailor services and improvements to enhance passenger satisfaction. The dataset comprises 24 predictive features, excluding the target feature, encompassing a variety of aspects related to the flight, such as seating comfort, service quality, food quality, and overall experience.

The training set consists of 103,903 records, providing a robust sample size for training our models, while the test set includes 25,975 records, which allows for a thorough evaluation of the model's performance in unseen scenarios.

To address this classification problem, I plan to employ machine learning algorithms. These include decision tree, Random Forest and KNN which are ensemble methods known for their effectiveness in handling diverse datasets by reducing variance and bias.

These algorithms will be tuned and evaluated to determine their accuracy and effectiveness. The goal is to generate practical insights that could inform enhancements in service provision and overall passenger experience management.

2. INTRODUCTION

2.1 INTRODUTION

In the rapidly evolving airline industry, customer satisfaction remains a pivotal benchmark that significantly influences business success. As airlines strive to enhance their competitive edge, understanding the multifaceted dimensions of passenger experience becomes crucial. This project leverages a rich dataset derived from survey responses of airline passengers, encompassing both business and leisure travellers.

The dataset reflects a wide array of variables that capture the essence of the air travel experience, including but not limited to inflight wifi service, seat comfort, cleanliness, food quality, and the efficiency of check-in and baggage services. These factors are analysed to evaluate their impact on overall passenger satisfaction, categorized into two distinct outcomes: satisfied and neutral or dissatisfied.

The aviation sector is characterized by its high operational costs and the significant value placed on customer retention. It is well-documented that enhancing passenger satisfaction is not just about addressing the negatives but also about reinforcing the positive aspects of the travel experience. Therefore, this study does not only identify the pain points and areas requiring improvement but also highlights the attributes that are most appreciated by passengers.

This project employs a methodical approach to analyse the data, utilizing statistical tools to uncover underlying patterns and correlations. By doing so, it aims to provide a comprehensive overview of the current state of passenger satisfaction within the industry. The insights derived from this analysis will not only aid airlines in refining their customer service strategies but also contribute to the academic literature on service quality and customer satisfaction in air travel.

Ultimately, the findings of this study are expected to assist airline management teams in making informed decisions that enhance customer loyalty, improve operational efficiency, and increase profitability. The overarching goal is to bridge the gap between passenger expectations and the services offered, thereby enriching the travel experience and fostering a stronger relationship between airlines and their customers.

3. PROBLEM DEFINITION AND ALGORITHM

3.1 PROBLEM DEFINITION

The primary objective of this research is to classify airline passengers based on their satisfaction levels regarding their flight experience. This classification is pivotal as it informs service modifications tailored to enhancing passenger satisfaction. Such targeted improvements can lead to increased customer retention and profitability in the highly competitive airline industry.

Airline industries operate in a highly competitive environment where customer satisfaction is directly linked to market success. A dissatisfied passenger not only refrains from using the same airline again but is also likely to influence others through negative reviews and feedback. Thus, understanding and predicting passenger satisfaction through data-driven insights is critical for strategic decision-making in service management and operational improvements.

Problem Statement is that despite the vast amount of data collected on passenger experiences, airlines often struggle to effectively use this information to improve service quality and passenger satisfaction. The challenge lies in accurately predicting passenger satisfaction based on various features related to their flight experience.

First, the data is heterogeneous, containing both ordinal and categorical variables that describe subjective passenger experiences. Second, the relationships between these variables and overall satisfaction are not straightforward due to the hidden interactions and dependencies. Therefore, the main question this project addresses is: "How can machine learning be employed to predict and classify the satisfaction levels of airline passengers based on their reported flight experience?"

Approach: To tackle this problem, I will employ several machine learning algorithms that are adept at handling classification tasks with complex, multifaceted data. The algorithms chosen for this project include:

A. Decision Trees
B. Random Forest
C. K-Nearest Neighbors (KNN)

3.2 ALGORTHMS

3.2.1 Decision Trees:

Decision Trees are a type of supervised learning algorithm predominantly used for classification and regression tasks. I will be using classification in this dataset. They model decisions and their possible consequences as a tree-like structure, where each internal node represents a "test" on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.

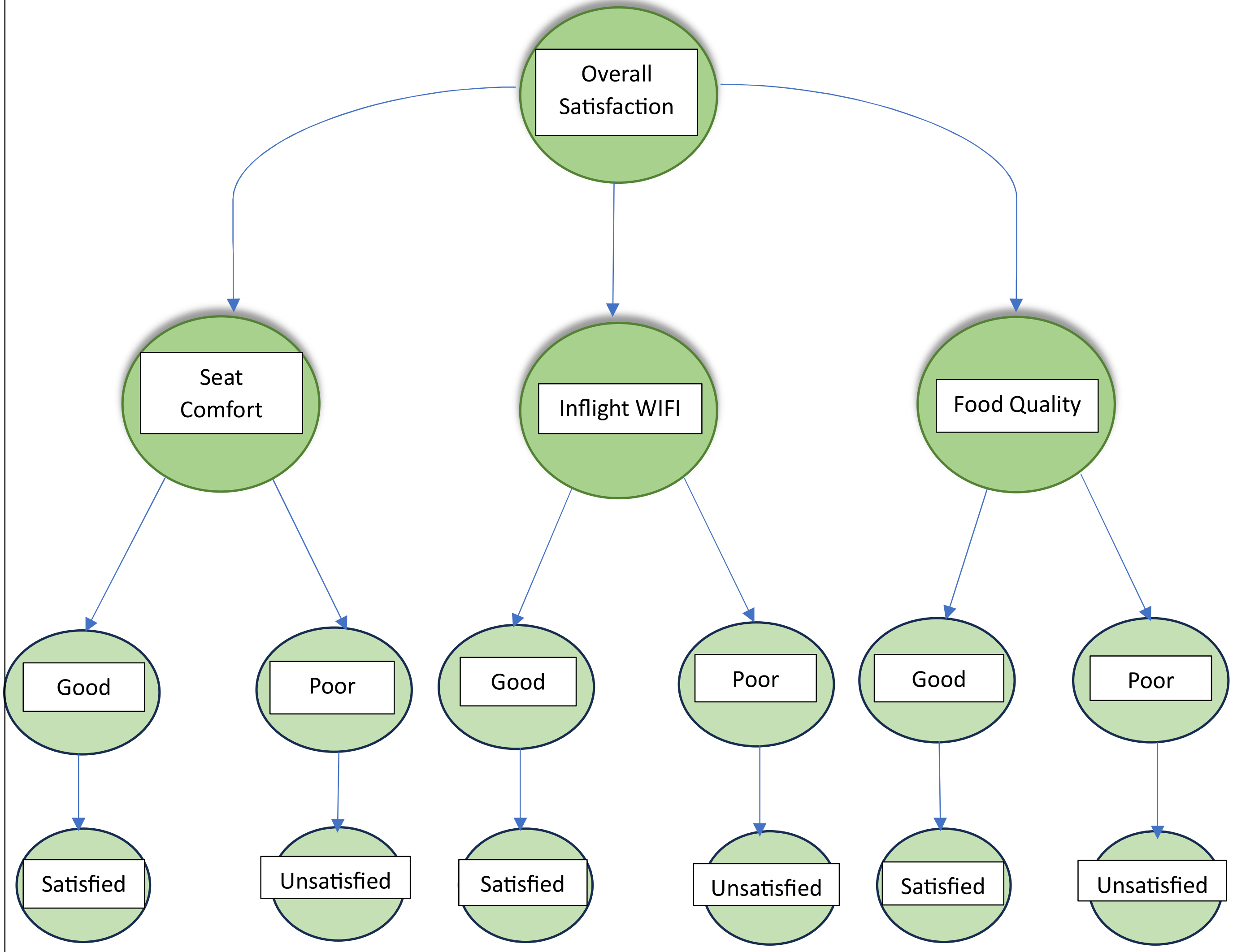


Fig- 1

Above is a simple visual representation of a decision tree involves illustrating how the decision nodes and branches might be structured based when applied to an airline passenger satisfaction features in my dataset.

a. Root Node: The topmost decision is based on the overall satisfaction, which is a general assessment of the passenger's experience.
b. Decision Nodes: Each node (e.g., Seat Comfort, Inflight Wifi, Food Quality) represents a feature in the dataset that significantly impacts the passenger's satisfaction. These features are hypothetical and should be selected based on actual data analysis.
c. Branches: The branches represent the outcomes of each decision node. For example, seat comfort can be judged as 'Good' or 'Poor'. For each feature, depict possible outcomes that lead to further decisions or final judgments.
d. Leaf Nodes: These are the final nodes that represent the classification outcome, in this case, 'Satisfied' or 'Unsatisfied'. These are your final decisions based on the paths taken through the tree.

Key features in selection of Decision tree:

a. Simplicity and Visual Appeal: Decision trees are simple to understand and interpret visually, making them a popular choice in situations where explaining the model to stakeholders is crucial.
b. Non-linear Data Handling: They can handle both numerical and categorical data and can capture non-linear relationships between features without requiring transformation.
c. Automatic Feature Selection: They inherently perform feature selection by choosing the most discriminative questions to ask at each node.

Effectiveness for This Dataset:

The airline passenger satisfaction dataset comprises various features, both categorical and continuous, that affect a passenger's overall travel experience.

a. Handling Mixed Data Types: Decision Trees effortlessly manage the mixed data types (e.g., ratings for comfort, categorical data like class of service) present in the dataset without requiring extensive preprocessing.
b. Interpretability for Business Insights: They provide clear insights into which factors most heavily influence passenger satisfaction, crucial for strategic business decisions. For instance, a decision tree can reveal if seat comfort more significantly impacts satisfaction compared to other features like inflight meals or legroom.
c. Capability to Model Complex Decision-Making Processes: The decision-making process of a passenger's satisfaction is complex and involves multiple factors. Decision Trees can model such processes by examining a series of decisions that lead to a final satisfaction rating.

(in model building add the notes and pseducode)

3.2.2 Random Forest

Random Forest is an ensemble learning method that combines the predictions from multiple decision tree models to produce a more accurate and robust output than any single tree could achieve on its own. It starts by creating multiple subsets of the original dataset through a process called bootstrap sampling. This involves randomly selecting data points from the dataset with replacement, meaning the same data point can appear multiple times in a single subset.

A Random Forest consists of multiple decision trees, each generated from a subset of the data and using a subset of the features at each split. A single decision tree in a Random Forest would look like the one above in decision tree, with some variations due to the random selection of data points and features. However, visualizing the entire Random Forest model would not involve just one tree, but rather an ensemble of many such trees.

Key features in selection of Random Forest:

a. Robustness to Overfitting: Unlike individual decision trees which can overfit on noisy data, Random Forest mitigates this by averaging multiple trees, thereby reducing variance without a substantial increase in bias.
b. Handling of Large Datasets with High Dimensionality: It can handle thousands of input variables without variable deletion, making it highly scalable.
c. Feature Importance: It provides a straightforward indication of the importance of each feature on prediction, aiding in understanding which factors most influence the outcome.

Effectiveness for This Dataset:

a. Improved Accuracy: By using multiple decision trees, Random Forest can achieve higher accuracy than a single decision tree. It reduces the risk of an erroneous conclusion from a single tree by averaging multiple decisions.
b. Handling Diverse Data Types: Like decision trees, Random Forest can handle categorical and continuous variables, which are prevalent in your dataset (e.g., age, gender, customer type, service ratings).
c. Avoidance of Overfitting: Your dataset, with its multiple predictors, is susceptible to overfitting if modeled with too complex a model. Random Forest inherently avoids this while still being able to capture complex interactions between variables.

3.2.3 K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a simple, non-parametric, and lazy learning algorithm used for classification and regression. It is called non-parametric because it doesn’t make any assumptions about the underlying data distribution.

The term 'lazy' refers to the fact that it does not use the training data points to do any generalization, meaning there is no explicit training phase or it is very minimal. The KNN algorithm is well-equipped to handle the heterogeneous mix of categorical and numerical data typically found in customer satisfaction surveys. For instance, passenger age is a numerical variable, while seat comfort may be categorical (ratings such as "good", "average", "poor"). KNN can accommodate these different types of data by using an appropriate distance metric that can measure the similarity between such varied data points effectively.

Key features in selection of KNN

a. Versatility: it predicts the label of a new sample based on the majority vote of its nearest neighbors.
b. Simplicity: KNN's straightforward implementation is an asset, allowing for quick deployment and ease of maintenance.
c. No Training Phase: The airline industry's data is frequently updated with new customer feedback, and KNN can incorporate this new data without downtime or the need to retrain models.
d. Non-parametric: Customer satisfaction data is often nuanced and doesn't always fit a standard pattern. KNN's ability to work without assumptions about the data distribution means it can more accurately reflect the diverse experiences and opinions of airline passengers.

Effectiveness for This Dataset:

a. Diverse Feature Types: my selected dataset includes a mix of categorical and continuous features. KNN can handle this diversity effectively by using appropriate distance metrics. Using Euclidean distance.
b. Complex and Subtle Patterns: KNN is particularly well-suited to datasets where satisfaction may be influenced by complex, subtle patterns rather than clear-cut, easily modelled trends.
c. No Assumptions About Data: KNN makes no assumptions about the distribution of data, which is helpful if the satisfaction factors are complex or if there is no clear hypothesis about what distribution the data follows.

I will be performing implementation of the K-Nearest Neighbors (KNN) algorithm using TensorFlow, which is highly efficient for handling large datasets and can leverage hardware acceleration (like GPUs). KNN is a simple, yet powerful, non-parametric algorithm used in both classification and regression. It's non-parametric because it makes no assumptions about the underlying data distribution.

4. DATA PREPROCESSING

4.1 DATA PREPROCESSING

The initial phase of the project involved preparing the data for analysis, a critical step in ensuring the quality and reliability of the results. The following subsections outline the systematic approach taken to preprocess the airline passenger satisfaction dataset.

A. Library Importation:
The data preprocessing began with the importation of essential Python libraries:
- pandas for data manipulation and analysis,
- numpy for numerical operations,
- seaborn and matplotlib.pyplot for data visualization

B. Dataset loading:
Two datasets were loaded into the Python environment using pandas: the training dataset (train.csv) and the testing dataset (test.csv).

C. Initial Data exploration:
A preliminary examination of the training dataset was conducted using the head() method to obtain an overview of the data structure, including the initial rows and the columns present.

D. Irrelevant Feature Removal:
Columns that do not contribute to the analysis, specifically 'Unnamed: 0' and 'id', were identified as unnecessary and subsequently dropped from both the training and testing datasets. These columns likely represent indices and unique identifiers that have no predictive power for passenger satisfaction.

E. Response Variable Encoding:
The response variable satisfaction in both datasets was converted from a categorical variable (with values 'neutral or dissatisfied' and 'satisfied') into a numerical format (0 and 1, respectively). This encoding facilitates computational handling during model training as machine learning algorithms operate on numerical data.

F. Class balance check:
The balance of the classes in the 'satisfaction' variable was checked through a value count. An imbalanced dataset can bias the predictive model, leading to a skew in favor of the majority class.

G. Missing Value Assessment:
Both the training and testing datasets were checked for null values using the isnull().sum() method. Handling missing data is a crucial step in data preprocessing to prevent inaccuracies in model predictions.

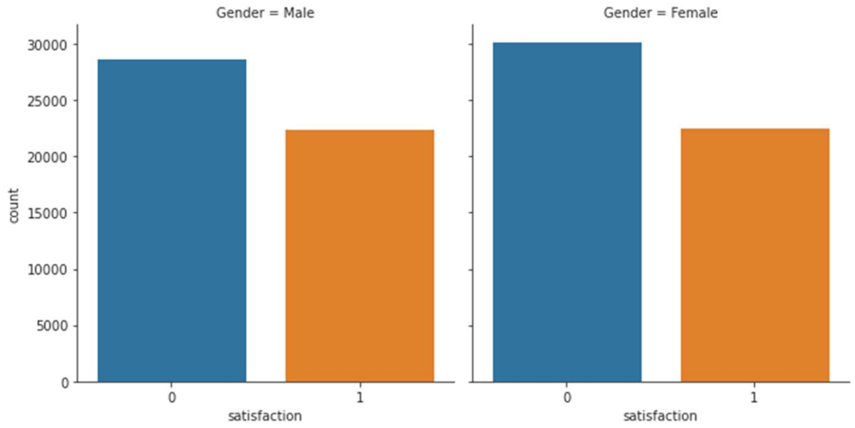
H. Null Value Imputation:
The null values in the 'Arrival Delay in Minutes' feature were replaced with the mean of the respective feature in both datasets. This imputation method assumes that the absence of data can be reasonably estimated by the average observed delay, maintaining the distribution of the feature.

5. EXPLORATORY DATA ANALYSIS

5.1 Exploratory data analysis

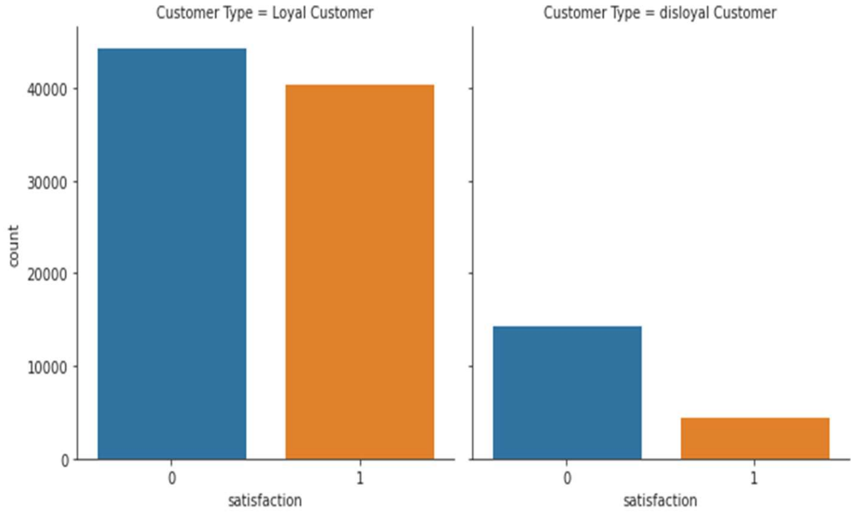
A. Gender:

From the graphs above we can see that passenger dissatisfaction is quite similar across the genders, with female passengers a bit dissatisfied than male passengers.



B. Customer type:

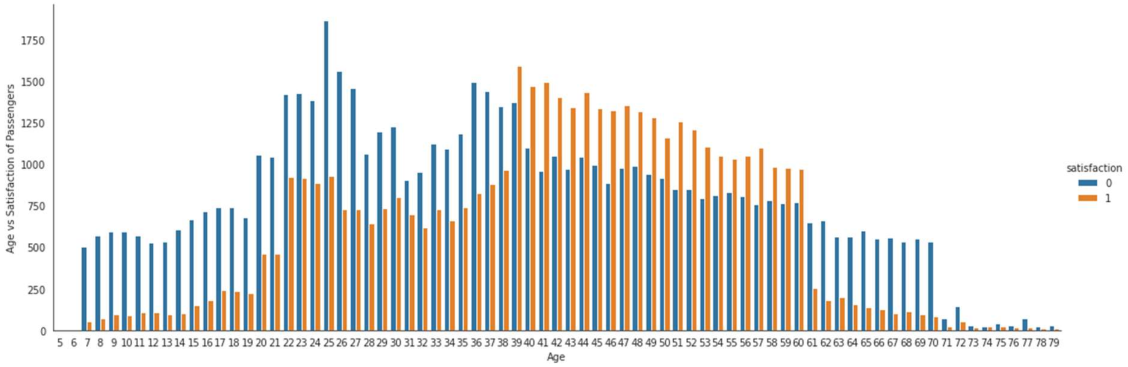
There are a lot more Loyal Customers for the airline than the disloyal ones, but even though that is the case, there seem to be a lot more dissatisfied customers across both customer types.



C. **Age:**

It is observed that from age 7 to 38 and from age 61 to 79 the number of dissatisfied passengers is comparatively higher, which gives us an insight as to which target group should the airline focus to improve the passenger satisfaction ratings.

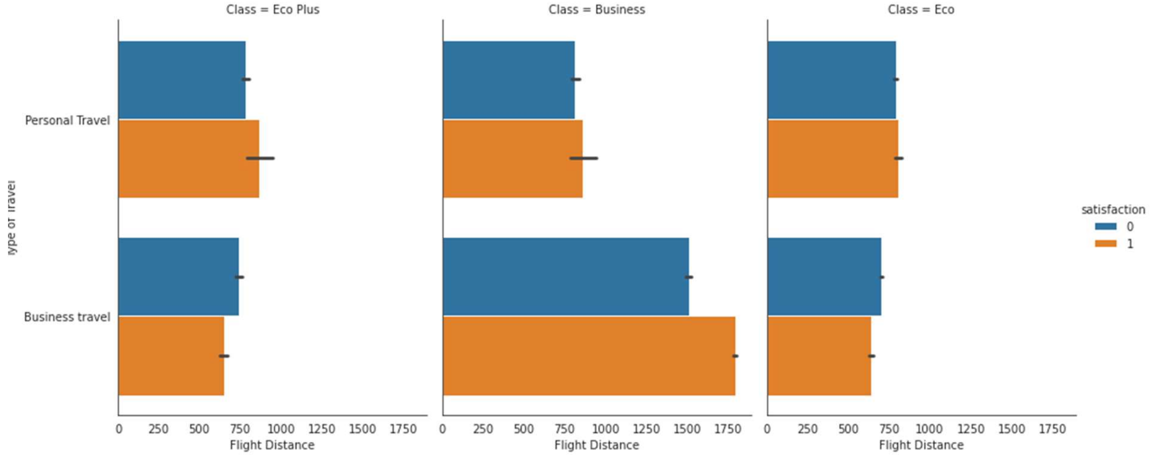
On the contrary, we can see that passenger in the age 39 - 60 are satisfied with their experience.



D. **Class, Flight Distance and Type of Travel:**

We can see that for Eco, Eco Plus and Business class passengers who are travelling for Personal Reasons the number of Satisfied customers are just a bit more than disssatisfied passengers.

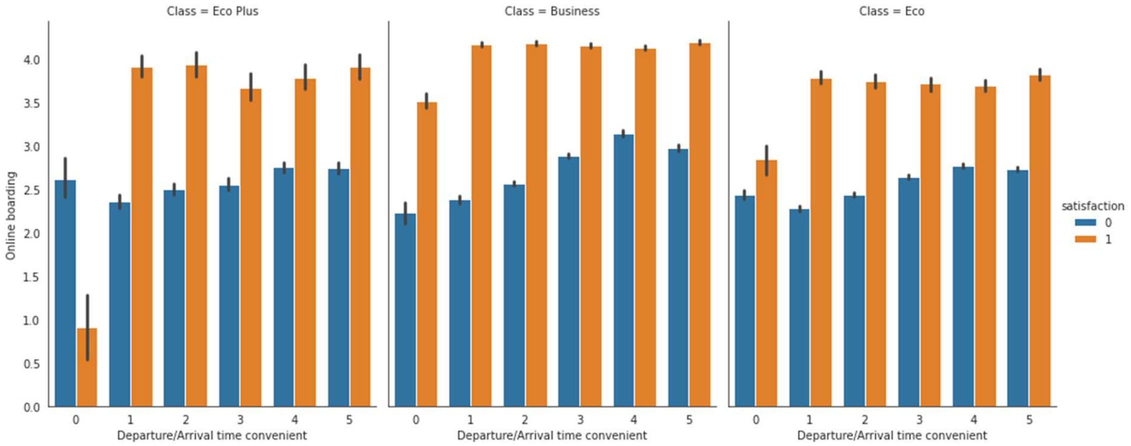
Also we can observe an interesting comparision here, the passengers who are traveling for Business Purpose, but are travelling through Eco and Eco Plus class are more disssatisfied, on the contrary the passengers travelling by Business class for Business Purpose are more satisfied.



E. **Departure/Arrival Time, Online Boarding grouped by Class:**

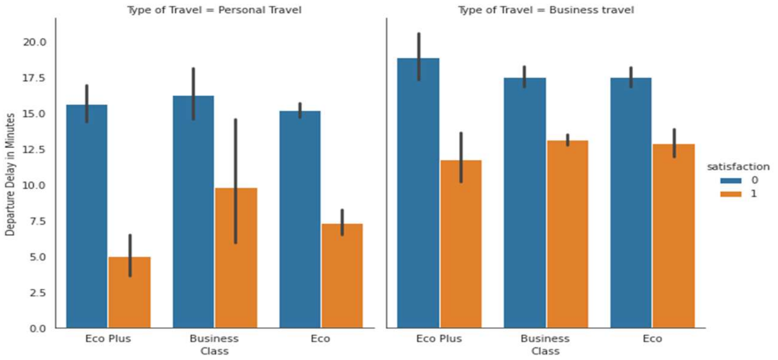
Except for the Eco Plus class which has higher number of dissatisfied passengers, where they have provided 0 rating, there seems to be a greater number of satisfied passengers across classes.

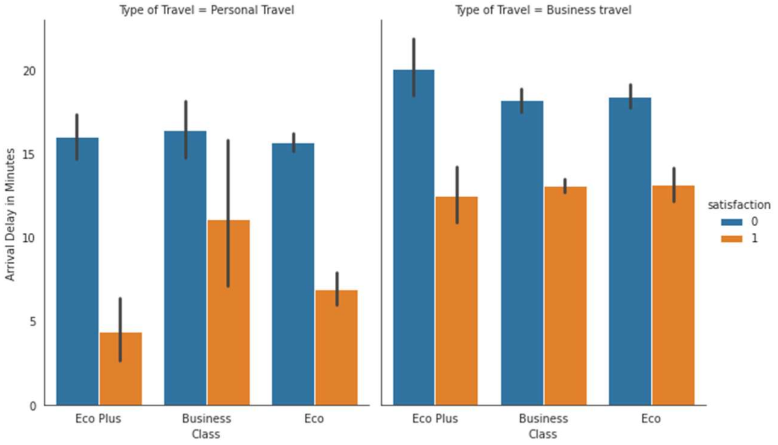
This analysis proves that passengers need convenient features like Online Boarding to make their flight experience pleasing.



F. **Arrival and Departure Delay grouped by Type of Travel:**

From the graphs above it is evident that no passenger likes delays. The number of dissatisfied passengers travelling for Business Purpose are greater than those travelling for Personal Reasons.



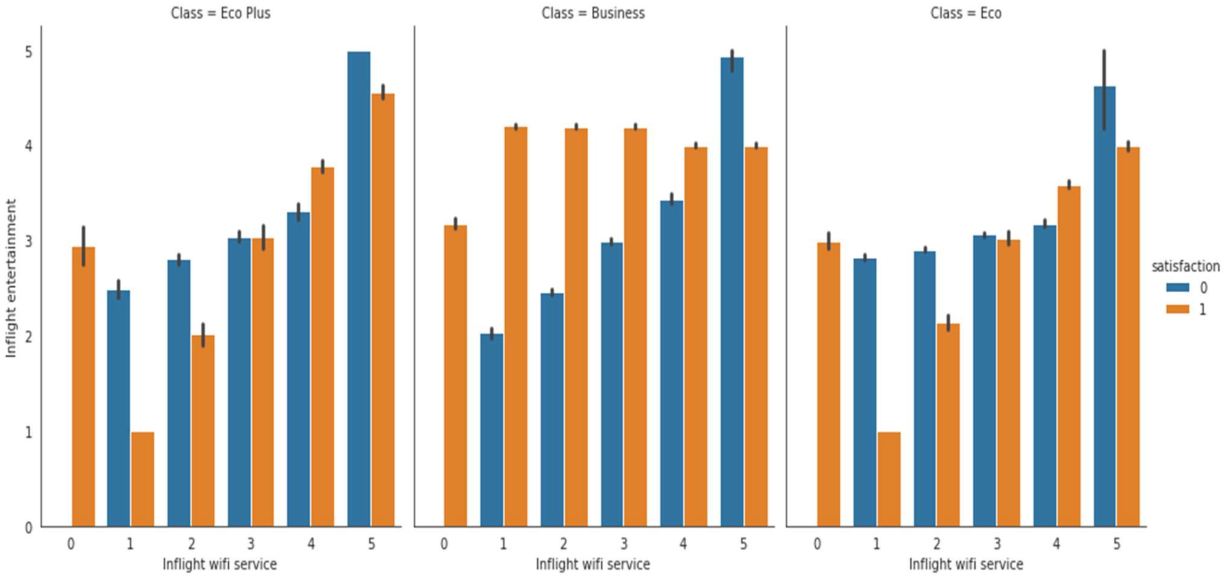


G. Inflight WiFi and Entertainment grouped by Class:

We have a very unusual stat here, where we can see that Eco Plus passengers are satisfied even if they do not have any WiFi services or just Mid Level of Entertainment.

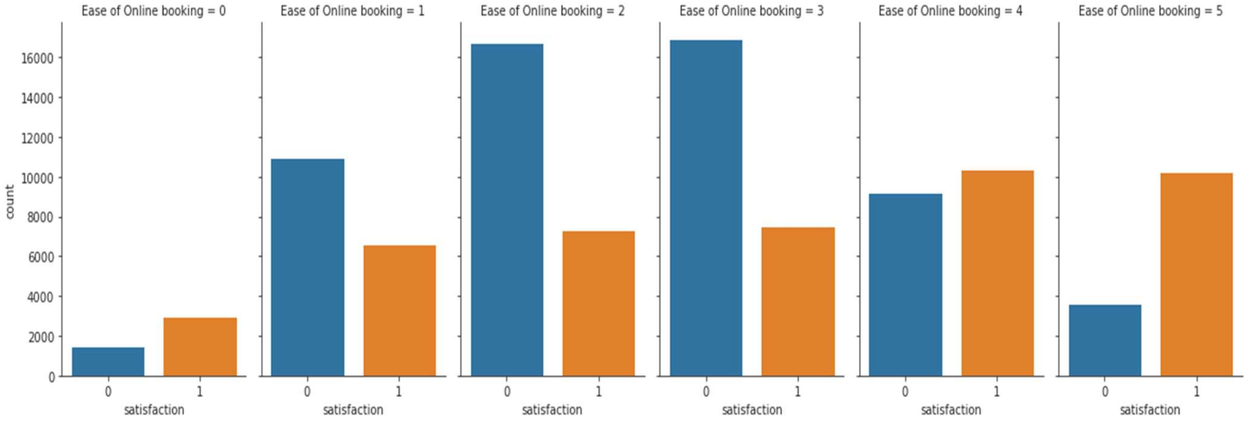
For Business class passengers it is evident that they need the highest levels of WiFi and Entertainment services as they have paid a significantly higher number of charges per seat.

For Eco passengers, they need high level of Entertainment and WiFi services to be satisfied.



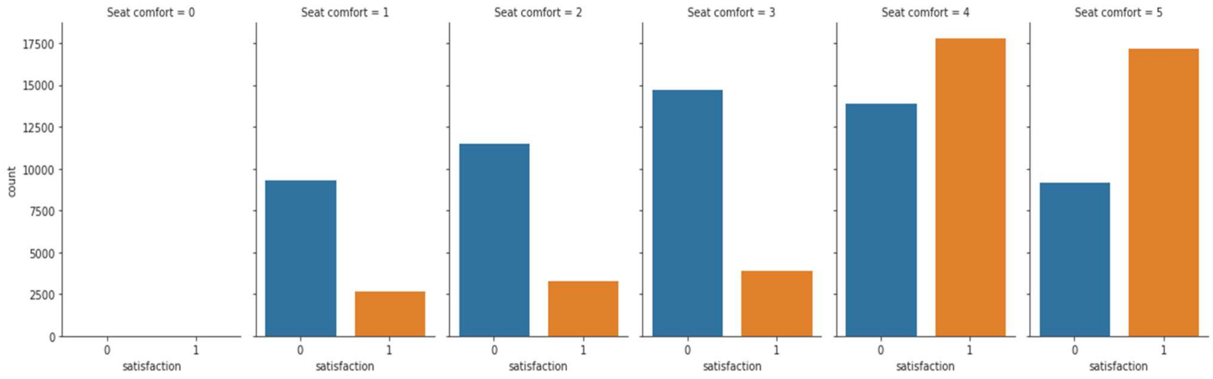
H. Ease of Online Booking:

We can see that passengers are only satisfied with the highest level of convenience of ratings 4 and 5 to be satisfied.



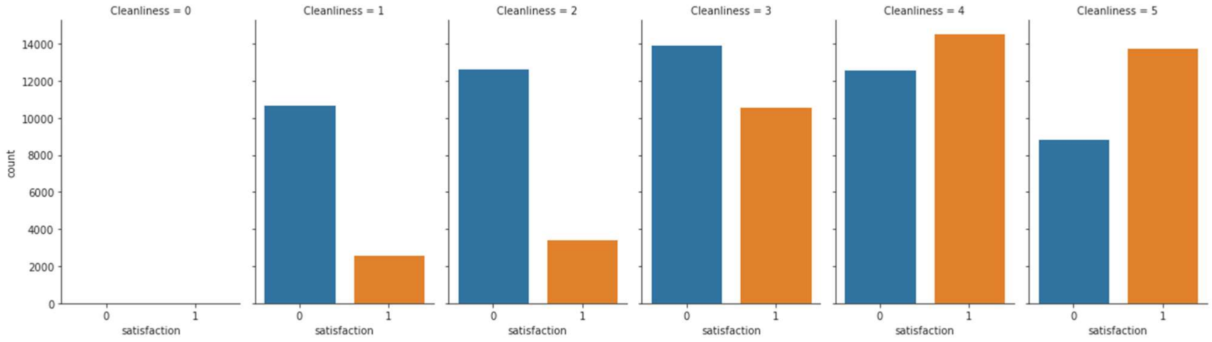
I. Seat Comfort:

It can be observed that passengers are only satisfied with the highest level of seat comforts, where they are probably getting more leg space or window seats which has ratings 4 and 5 to be satisfied.



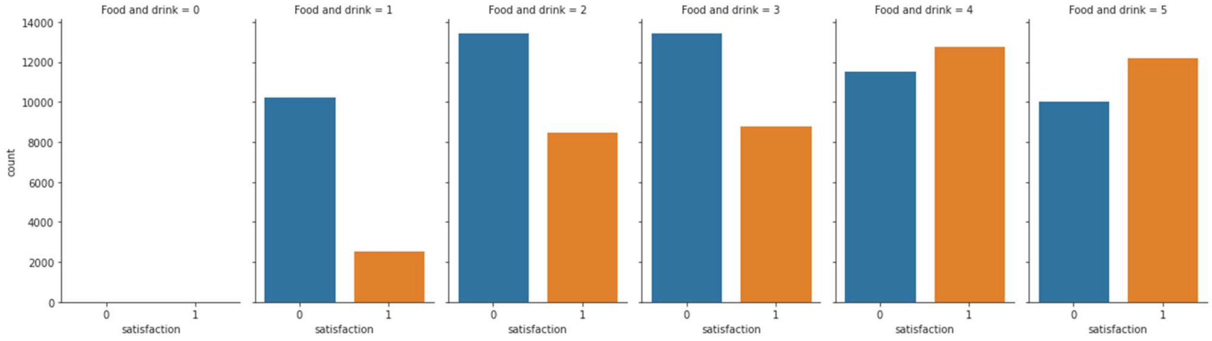
J. Cleanliness:

It can be observed that passengers are only satisfied with the highest level of cleanliness of ratings 4 and 5 to be satisfied.

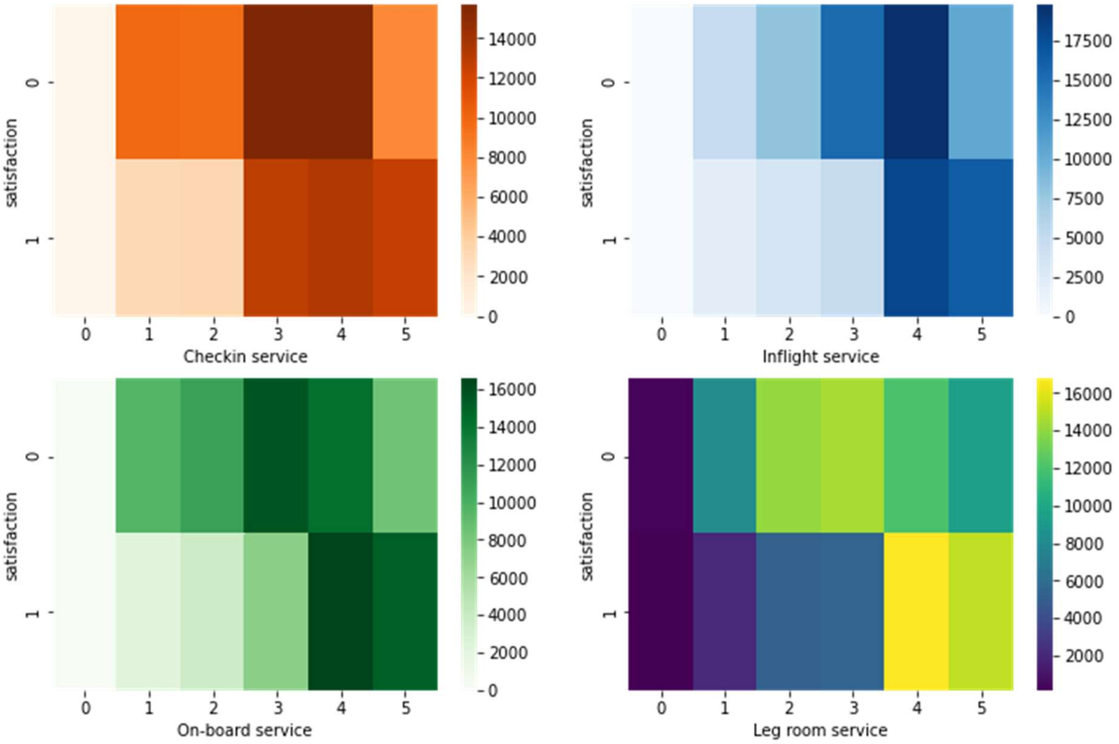


K. Food and Drinks:

It can be observed that passengers are only satisfied with the highest level of Food and Drink where they are probably getting all of the requested food and drinks which has ratings 4 and 5 to be satisfied.



Except for the check in service, which has 0 to 2 ratings provided by passengers who look to be the most dissatisfied, for rest of services, passengers who provided 4 and 5 ratings seem to be satisfied.



6. MODEL BUILDING

6.1 DECISION TREE

Provided each section and clarify how the decision tree is build, trained and used to make prediction for my dataset, building custom decision tree classifier.
(Logic building in written notes)

A. Class Initialization:
   - max_features: Determines how many features are considered for each split. The default 'sqrt' means the square root of the number of features.
   - max_depth: The maximum depth of the tree. Deeper trees can model more complex patterns but may overfit.
   - min_samples_split: The minimum number of samples required to split a node. Smaller numbers could lead to overfitting.
   - default_class: The fallback class used when a prediction is requested from an empty tree.

B. Entropy Calculation:
   Calculates the entropy of a set y, which measures the impurity of labels. Lower entropy means more homogeneous labels.

C. Information Gain calculation:
   Computes the reduction in entropy from a potential split, serving as a criterion to choose the best split.

D. Best split determination:
   Examines possible splits on all features (or a subset, based on max_features), calculating information gain for each and choosing the split with the highest gain.

E. Recursive Tree Building:
   Recursively builds the decision tree. Stops recursion if the maximum depth is reached, if there are fewer samples in the node than min_samples_split, or if all labels are the same.
   If no valid split is found (or further splitting is not possible due to the constraints), it returns the most common label (mode).

F. Prediction Function:
   - Predict: Makes predictions for a batch of instances.
   - predict_one: Makes a prediction for a single instance by traversing the tree from the root to a leaf, following the decision rules established by the nodes.

G. Model fitting:
   Initializes the tree by determining the number of features to consider (max_features) based on the input setting and starts the recursive tree building process with the entire training dataset.

6.2 RANDOM FOREST

Provided each section and clarify how the Random Forest is build, trained and used to make prediction for my dataset, building custom random forest classifier. Most of the things were done in decision tree for building of random forest.

A. Class Initialization:
   - n_estimators: Specifies the number of trees in the forest. A larger number generally improves the model's performance but increases computational cost.
   - max_features: Determines the number of features to consider when looking for the best split at each node. 'sqrt' is a common default, meaning the square root of the total number of features.
   - max_depth: Maximum depth of each tree. Deeper trees can learn more detailed patterns but may lead to overfitting.
   - min_samples_split: Minimum number of samples required to split a node. Helps control overfitting.
   - default_class: Default prediction class to use if no decision can be made based on the trees.

B. Bootstrap Sampling:
   Randomly samples the dataset with replacement to create a new dataset that will be used to train each decision tree. This introduces randomness into the dataset, which helps in building a more robust model by reducing variance.

C. Model Fitting:
   - self.trees = []: Initializes an empty list to store the individual trees.
   - Loops through a specified number of estimators, building a tree for each iteration by first creating a bootstrap sample, then fitting a decision tree to this sample.

D. Prediction:
   - tree_preds = np.array(...): Collects predictions from each tree in the forest for the input data X.
   - np.swapaxes(tree_preds, 0, 1): Changes the shape of the predictions array for aggregation.
   - mode(tree_preds, axis=1, nan_policy='omit').mode: Computes the mode (most common value) across predictions of all trees for each sample, which constitutes the final prediction for each sample.

6.3 KNN

Provided each section and clarify how the KNN is build, trained and used to make prediction for my dataset. I used TensorFlow-based implementation of the K-Nearest Neighbors (KNN) algorithm for classification.

A. Euclidean Distance Function:

- tf.expand_dims(data1, 1): Modifies data1 to have an additional dimension, making it compatible for subtraction with data2 using broadcasting. This is crucial for computing pairwise distances between sets of vectors without explicit loops.
- tf.square(data1 - data2): Calculates the square of the difference between every pair of points (from data1 and data2), element-wise. This operation utilizes TensorFlow's broadcasting capability.
- tf.reduce_sum(..., axis=2): Sums up the squared differences along the last axis to get the total squared Euclidean distance for each pair.
- tf.sqrt(distances): Takes the square root of the summed squares to finalize the Euclidean distance calculation for each pair.

B. Functions:

Following converts training data, test instances, and training labels into TensorFlow tensors, which are optimized for fast computation and can utilize GPU acceleration.

- def predict_classification_tf(training_set, training_labels, test_instance, k):
- train_tensor = tf.convert_to_tensor(training_set, dtype=tf.float32)
- test_tensor = tf.convert_to_tensor(test_instance, dtype=tf.float32)
- labels_tensor = tf.convert_to_tensor(training_labels, dtype=tf.int32)
- euclidean_distance_tf function to compute the distances between the test instance and all training instances.
- tf.nn.top_k(tf.negative(distances), k=k): Finds the indices of the k smallest distances
- tf.reshape(indices, [-1]): Flattens the indices array to ensure it is one-dimensional.
- tf.gather(labels_tensor, indices): Retrieves the labels of the k nearest neighbors from the training labels.
- tf.unique_with_counts(nearest_labels): Identifies unique labels and their counts among the nearest labels.
- tf.argmax(count): Finds the index of the label with the maximum count (i.e., the most frequent label).
- majority_label: Retrieves the label corresponding to the maximum index, which is the final prediction by majority vote.

C. Prediction Function:

Loops through each test instance, calls the prediction function, and collects the predictions.

- predictions = [predict_classification_tf(X_train, y_train, np.array([test_instance]), k) for test_instance in X_test[:]]

7. RESULTS

7.1 Results

For airline passenger satisfaction, selecting appropriate metrics to evaluate the performance of machine learning models is of paramount importance. Using confusion matrices, the performance of three different models Decision Tree, Random Forest, and K-Nearest Neighbors (KNN) was assessed through various metrics including accuracy, precision, recall, and F1 score. These metrics are essential for understanding how well each model performs in predicting both satisfied and dissatisfied passengers, which directly influences service improvement strategies and overall passenger experience management.

A confusion matrix is typically structured as follows:

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

A. Accuracy: Measures the overall correctness of the model, calculated as the ratio of correctly predicted instances to the total instances. High accuracy is essential for ensuring general reliability across diverse passenger data.

$$\textbf{Accuracy} = \mathbf{TP+TN/\ TP+TN+FP+FN}$$

B. Precision: Indicates the reliability of positive predictions. The ratio of correctly predicted positive observations to the total predicted positives. It shows how precise the model is when predicting positive classes.

$$\textbf{Precision} = \mathbf{TP/\ TP+FP}$$

C. Recall (Sensitivity): Reflects the model's ability to detect all positive instances (correctly identifying dissatisfied passengers). The ratio of correctly predicted positive observations to all observations in actual class. It measures the model's ability to capture actual positives.

$$\textbf{Recall} = \mathbf{TP/\ TP+FN}$$

D. F1 Score: Harmonic mean of precision and recall. It is a useful measure when seeking a balance between precision and recall, particularly when classes are imbalanced. An excellent F1 score indicates a robust model that balances the detection of dissatisfied passengers with the precision necessary to avoid overestimating dissatisfaction.

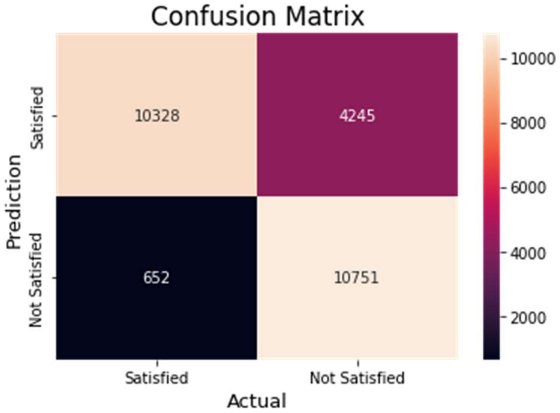
$$\textbf{F1 Score} =2\times(\ \textbf{Precision}*\textbf{Recall}/\ \textbf{Precision}+\textbf{Recall})$$

7.2 Model Evaluation

The results of the model evaluations are as follows:

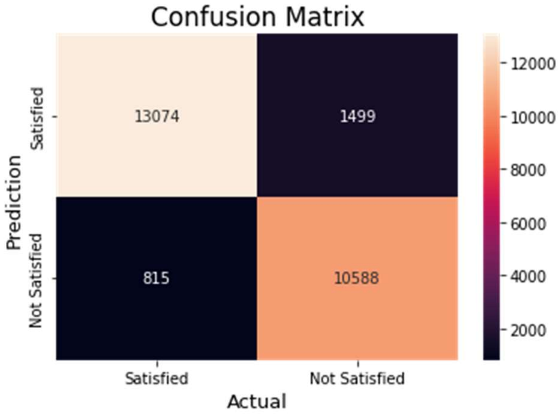
A. Decision Tree

- Accuracy: 81.15%
- Precision: 0.94 (Class 0), 0.72 (Class 1)
- Recall: 0.71 (Class 0), 0.94 (Class 1)
- F1 Score: 0.81 (Class 0), 0.81 (Class 1)
- Analysis: The Decision Tree model shows a strong ability to identify dissatisfied passengers (high recall for Class 1) but at the cost of misclassifying a significant number of satisfied passengers (lower recall for Class 0). This could lead to over-prioritizing resources towards appeasing already satisfied customers.



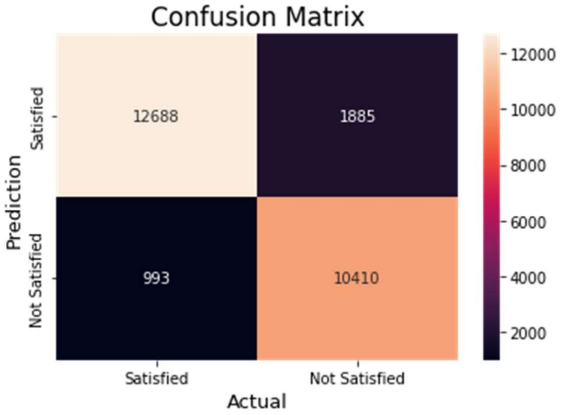
B. Random Forest

- Accuracy: 91.09%
- Precision: 0.94 (Class 0), 0.88 (Class 1)
- Recall: 0.90 (Class 0), 0.93 (Class 1)
- F1 Score: 0.92 (Class 0), 0.90 (Class 1)
- Analysis: Exhibiting the highest accuracy and balanced performance across all metrics, the Random Forest model efficiently identifies both satisfied and dissatisfied passengers. Its higher precision and recall for both classes indicate a reliable prediction capability, making it highly suitable for operational improvements and targeted service enhancements.



C. KNN (K-Nearest Neighbors)

- Accuracy: 88.92%
- Precision: 0.93 (Class 0), 0.85 (Class 1)
- Recall: 0.87 (Class 0), 0.91 (Class 1)
- F1 Score: 0.90 (Class 0), 0.88 (Class 1)
- Implication: KNN shows a good balance between identifying satisfied and dissatisfied passengers. While it has a slightly lower performance compared to the Random Forest, it still presents a viable option for scenarios where model interpretability and execution speed are critical.



8. CONCLUSION

8.1 CONCLUSION

This project aimed to classify airline passengers based on their satisfaction levels, leveraging a dataset with 24 predictive features related to various aspects of the flight experience. Given the complexity of the airline passenger satisfaction dataset, which encompasses a diverse array of predictive features, using a multifaceted approach to evaluate model performance is essential.

Through rigorous testing and evaluation using a confusion matrix, each model was assessed based on accuracy, precision, recall, and F1 score. These metrics provided a comprehensive view of each model's performance in distinguishing between satisfied and dissatisfied passengers.

The choice of metrics was driven by the need to understand both the overall effectiveness of the models (accuracy) and their ability to correctly identify dissatisfied passengers without misclassifying satisfied ones (precision and recall). The F1 score was particularly critical, providing a balanced measure of precision and recall, essential for dealing with the potentially skewed distributions of satisfied versus dissatisfied passengers.

A. **Decision Tree** showed a tendency to favor recall over precision, particularly effective in identifying dissatisfied passengers but at the risk of false positives.
B. **K-Nearest Neighbors** offered good balance and reasonable computational efficiency but did not achieve the highest marks in any particular metric.
C. **Random Forest** emerged as the superior model, demonstrating the highest accuracy (91.09%) and an impressive balance between precision and recall. This model effectively minimized false positives while accurately identifying dissatisfied passengers, crucial for allocating resources wisely and enhancing passenger experiences.

The Random Forest model is recommended for operational deployment, given its robust performance across all key metrics. It has proven especially capable of handling the complex, multifaceted data presented by the airline passenger satisfaction survey, effectively reducing variance and bias, and providing trustworthy predictions. This model not only meets the analytical needs of the project but also aligns with the business objectives of maximizing passenger satisfaction and operational efficiency.

9. REFERENCES

1. Alsabti K., Ranka S. and Singh V., CLOUDS: A Decision Tree Classifierfor Large Datasets, Conference on Knowledge Discovery and Data Mining(KDD-98), August 1998.
2. Breiman L., Friedman J., Olshen R., and Stone C.. Classification and Regres-sion Trees. Wadsworth Int. Group, 1984.
3. Murthy S. K., Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey. Data Mining and Knowledge Discovery, 2(4):345-389,1998.
4. Biau G, Devroye L, Lugosi G (2008) Consistency of random forests and other averaging classifiers. J Mach Learn Res 9:2015–2033
5. S. Zhang, X. Li, M. Zong, X. Zhu and R. Wang, "Efficient kNN Classification With Different Numbers of Nearest Neighbors," in IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 5, pp. 1774-1785, May 2018, doi: 10.1109/TNNLS.2017.2673241.
6. Breiman L (2001) Random forests. Mach Learn 45:5–32
7. Boyu Li, Yun Wen Chen, and Yan Qiu Chen. 2008. The nearest neighbor algorithm of local probability centers. IEEE Trans. Syst. Man Cybernet. B 38, 1 (2008), 141--154