Chapter 1: Introduction

1.1 Description

This project focuses on analysing an airline passenger satisfaction survey dataset to understand the factors influencing passenger satisfaction and to develop predictive models for determining whether a passenger will be satisfied, neutral, or dissatisfied with their airline experience. Machine learning algorithms will be employed to achieve these objectives.

The dataset comprises a variety of features related to airline passengers' demographics, travel details, and satisfaction ratings on in-flight services. Key features include gender, customer type, age, purpose of travel, travel class, flight distance, and ratings for various in-flight services like inflight wifi, departure/arrival time convenience, seat comfort, and more.

This project aims to leverage machine learning techniques to better understand the determinants of passenger satisfaction in the airline industry. By analysing and predicting passenger satisfaction levels, airlines can make data-driven decisions to enhance their services, leading to improved customer experiences and increased loyalty.

1.2 Problem Statement

The airline industry is highly competitive, and passenger satisfaction is a crucial factor in retaining customers and maintaining a positive brand image. To address this, we aim to analyse an airline passenger satisfaction survey dataset to answer the following questions:

1. **Factors Impacting Passenger Satisfaction:** What are the key factors that are highly correlated with passenger satisfaction and dissatisfaction? Understanding these factors is vital for improving the passenger experience and addressing pain points.

2. **Predicting Passenger Satisfaction:** Can we build predictive models to forecast passenger satisfaction levels based on demographic and service-related features?

Developing such models can assist airlines in identifying dissatisfied passengers in real-time and taking corrective actions.

1.3 Objectives

The project objectives include:

- Perform exploratory data analysis (EDA) to gain insights into the dataset, identify trends, and explore correlations between features and passenger satisfaction levels.
- Implement machine learning models to predict passenger satisfaction based on the provided features.
- Evaluate model performance using appropriate metrics such as accuracy, precision, recall and F1-score.
- Interpret model results to understand the importance of different features in determining passenger satisfaction and dissatisfaction.
- Provide actionable recommendations to airlines based on the analysis and predictive models to improve the passenger experience and overall satisfaction

1.4 Desired Outcomes

The project's desired outcomes include:

- **Gain Deep Insights**: My primary objective is to delve into the dataset and extract valuable insights regarding the factors influencing airline passenger satisfaction. Through thorough data analysis, I aim to uncover key drivers and trends that significantly impact satisfaction levels.
- **Build Predictive Power:** I endeavor to develop a robust machine learning model capable of accurately predicting passenger satisfaction levels based on various demographic and travel-related attributes. This predictive model will serve as a valuable tool for airlines to anticipate customer satisfaction and tailor their services accordingly.
- **Identify Key Features:** My goal is to determine the most influential features contributing to passenger satisfaction. By identifying these key factors, I can provide

actionable recommendations to airlines for prioritizing service improvements and enhancing overall customer experience.

- **Ensure Model Reliability:** I will rigorously evaluate the performance of the predictive model using appropriate metrics such as accuracy, precision, recall, and F1-score. This ensures that the model delivers reliable predictions and can effectively inform decision-making processes.
- **Drive Actionable Insights:** My aim is to provide actionable recommendations based on predictive analysis, enabling airlines to implement targeted strategies aimed at improving customer satisfaction levels and fostering loyalty and retention.
- **Enhance Operational Efficiency:** By facilitating data-driven decision-making processes, I seek to enhance operational efficiency within the aviation industry, optimizing resource allocation and service delivery.

1.5 Scope of the Project

The functionality of the airline customer satisfaction analysis project lies in its ability to leverage machine learning algorithms to predict and understand passenger satisfaction levels based on various attributes. By analyzing factors such as gender, age, travel type, and service satisfaction ratings, the project aims to provide actionable insights for airlines to enhance their services and improve overall customer experience. The performance of the project will be evaluated based on the accuracy and reliability of the predictive models in forecasting passenger satisfaction levels.

The targeted impact sector for this project is the aviation industry. By implementing machine learning-driven solutions, airlines can optimize their operations, tailor services to individual passenger preferences, and ultimately increase customer satisfaction and loyalty. The scope of the machine learning project extends beyond passenger satisfaction analysis; it encompasses a broader application in data-driven decision-making within the aviation sector. This includes personalized marketing strategies, predictive maintenance, route optimization, and safety enhancements, contributing to overall operational efficiency and profitability for airlines. Moreover, insights gained from this project can also inform policy decisions and industry-wide improvements, benefiting stakeholders across the aviation ecosystem.

The scope of this project encompasses several key components:

- **Data Exploration and Preprocessing:**
  - Data cleaning to handle missing values, outliers, and data quality issues.
  - Exploratory Data Analysis (EDA) to gain insights into the dataset through summary statistics and visualizations.
  - Feature selection to identify relevant factors highly correlated with passenger satisfaction.
- **Predictive Modeling:**
  - Selection of appropriate machine learning algorithms, such as logistic regression, decision tree classification, and random forest classification.
  - Splitting the dataset into training and testing sets for model evaluation.
  - Model training and evaluation to predict passenger satisfaction levels.
- **Model Evaluation and Interpretation:**
  - Assessment of model performance using metrics like accuracy, precision and F1-score.
  - Interpretation of feature importance to understand which factors contribute most to passenger satisfaction.
- **Recommendations and Insights:**
  - Provide actionable insights based on the analysis to guide improvements in airline services.
  - Identify areas where the airline can enhance the passenger experience.
- **Limitations and Ethical Considerations:**
  - Highlight project limitations, including data bias, subjectivity, and potential data privacy concerns.
- **Conclusion and Future Scope:**
  - Summarize project findings and the significance of identifying satisfaction drivers.
  - Suggest future research directions, such as data enrichment, real-time feedback analysis, and continuous improvement initiatives.

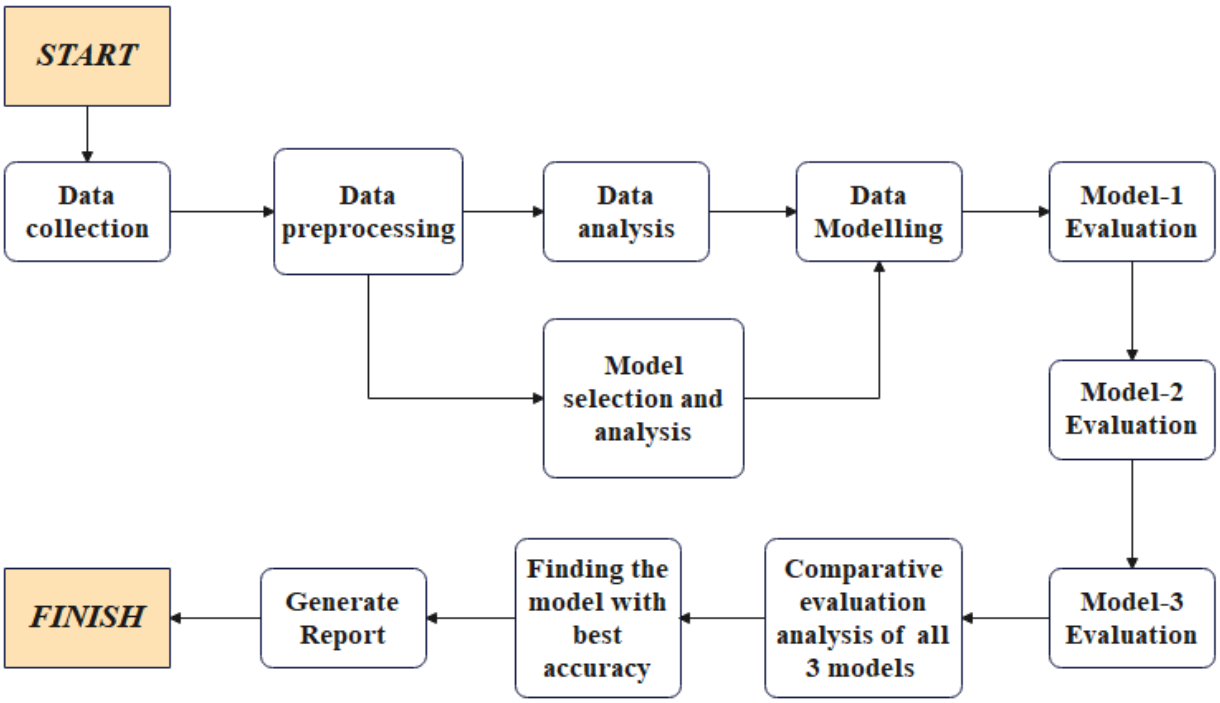
1.6 Project Planning Activities

To ensure efficient project execution, several planning activities will be undertaken:

1.6.1 PERT Chart

A PERT (Program Evaluation and Review Technique) chart will be created to outline project milestones, dependencies, and timelines. This chart will provide a visual representation of the project's schedule, ensuring that tasks are completed in a logical sequence.

Figure 1.1 PERT Chart



1.7 Organization of the Report

The organization of this report has been thoughtfully structured to ensure clarity and coherence in presenting the project on "Airline Passenger satisfaction analysis using supervised learning techniques." Each chapter has a distinct focus and contributes to a comprehensive understanding of the project's background, methodology, and outcomes.

Chapter 1, This chapter introduces a project focused on analyzing an airline passenger satisfaction dataset through machine learning. Objectives include understanding satisfaction factors, building predictive models, and offering actionable insights. Outcomes aim to gain insights, develop reliable models, identify key features, drive action, and enhance operational efficiency. The project scope covers data exploration, predictive modeling, evaluation, recommendations, and ethical considerations within the aviation sector. Project planning involves creating a PERT chart for milestone tracking. Ultimately, the chapter lays the groundwork for leveraging data-driven strategies to improve customer satisfaction and operational effectiveness in the airline industry.

Chapter 2, The literature review encompasses eight research papers focusing on predicting airline passenger satisfaction using machine learning algorithms and analyzing factors influencing satisfaction. Key themes include the significance of service quality, the impact of various factors on satisfaction levels, and the effectiveness of different classification algorithms. Studies highlight the importance of understanding customer preferences, improving service dimensions, and leveraging predictive models to enhance passenger satisfaction. While each study contributes unique insights and methodologies, common findings emphasize the relevance of machine learning techniques, particularly Random Forest, in accurately predicting and understanding passenger satisfaction across different airline contexts.

Chapter 3, It outlines the system design and methodology for conducting a comparative analysis of airline passenger satisfaction using logistic regression, decision tree classifier, and random forest techniques. The system design encompasses data gathering, preprocessing, model selection, training, evaluation, and comparison. Various algorithms such as Random Forest, AdaBoost, Gradient Boosting, XGBoost Regressor, K-Nearest Neighbors, Linear Discriminant Analysis, and Quadratic Discriminant Analysis are discussed in detail. The methodology involves data preprocessing, model training, evaluation, and comparison to select the best-performing algorithm. Challenges, trade-offs, and considerations for each algorithm are addressed to ensure a comprehensive and unbiased analysis of passenger satisfaction.

Chapter 4, this provides an in-depth look into the technical aspects of implementing the project, focusing on hardware and software requirements, implementation details, and results. Hardware requirements include minimum specifications for processors, memory, and storage. Software requirements encompass necessary tools and libraries. The implementation details cover data collection, preprocessing, and development of various classification models. Results include performance metrics and visualizations, with a comparative analysis of different models. Decision Tree, Random Forest, and XGB Classifier emerge as top performers, highlighting their suitability for predicting airline passenger satisfaction. Overall, the chapter emphasizes model selection based on performance and trade-offs between metrics.

Chapter 5, Chapter 5 concludes by highlighting Decision Tree, Random Forest, and XGB Classifier as effective models for predicting airline passenger satisfaction. It emphasizes the importance of balancing precision and recall and the need for rigorous evaluation to address potential overfitting. Future work includes enriching data sources, implementing real-time feedback mechanisms, and continuous model improvement to enhance predictive accuracy and personalized recommendations. Integrating predictive models with operational systems can streamline decision-making and drive innovation in the aviation industry, benefiting both airlines and passengers.

Throughout the report, appendices supplement the main content with additional information, such as project timelines and code samples. The list of figures, tables, and abbreviations assists readers in navigating the report seamlessly. Together, these components create a cohesive and informative document that guides readers through the project's journey from inception to conclusion.

Chapter 2: Literature Review

2.1 Summary of the research papers studied

A. *B. Herawan Hayadi, Jin-Mook Kim, Khodijah Hulliyah, and Husni Teja Sukmana, (2021)* ***"Predicting Airline Passenger Satisfaction with Classification Algorithms"***

The research paper "Predicting Airline Passenger Satisfaction with Classification Algorithms" by B. Herawan Hayadi, Jin-Mook Kim, Khodijah Hulliyah, and Husni Teja Sukmana, published in the International Journal of Informatics and Information System in March 2021, focuses on predicting airline passenger satisfaction using classification algorithms. The paper discusses the importance of service quality in the airline industry, particularly in in-flight services, which significantly impact passenger satisfaction and loyalty. The methodology used for the research includes data analysis and feature selection, using a dataset of approximately 130,000 survey entries from US airlines. The study uses classification algorithms like k-Nearest Neighbors, Logistic Regression, Decision Trees, Gaussian Naive Bayes, and Random Forest. The Random Forest algorithm was found to be the best performer in predicting passenger satisfaction, with an accuracy of 99%. The research identifies key factors affecting passenger satisfaction, with a strong emphasis on Inflight Wi-Fi Service. The paper provides simulations of passenger satisfaction for different scenarios, including first-time customers traveling for personal or business reasons. The paper concludes that airlines should focus on optimizing Inflight Wi-Fi Service and the ease of online booking to enhance passenger satisfaction. The research highlights the importance of understanding and addressing factors contributing to passenger satisfaction and loyalty in the airline industry.

B. *C. Murugesan and Dr. R. Perumalsamy (2017)* ***"A Research on Passengers' Satisfaction in Airways – In Coimbatore City"***

The research paper "A Research on Passengers' Satisfaction in Airways – In Coimbatore City" by C. Murugesan and Dr. R. Perumalsamy examines the factors affecting passenger satisfaction in the airline industry, particularly in Coimbatore City, India. The paper highlights the growth of air transport worldwide, with a focus on India's rapidly developing aviation sector. The

success of airlines is linked to the quality of service delivery and customer satisfaction. The literature review discusses strategies for staying competitive in the airline industry, emphasizing customer satisfaction as a critical factor in building loyalty and promoting repeat purchases. Customer relationship management and customer acquisition strategies are also discussed, emphasizing the importance of understanding customer needs and wants. The research methodology includes data collection from 200 respondents using questionnaires, with objectives including assessing passengers' opinions on service quality, price, and overall satisfaction. Limitations of the study include the focus on air travelers and the use of questionnaire methods. The findings of the study reveal that the majority of respondents are male, aged 31-40, married, and have undergraduate education. Business is the dominant occupation, with urban residents being the majority. Economy class is the most common choice for travelers. Passengers generally are satisfied with airline service quality and safety, but experience waiting times at various stages of their journey. Suggestions for improvement include reducing waiting times and focusing on departure/arrival punctuality. Passengers feel they receive value for their money and are overall satisfied with the service.

C. *Annisa Fitria Nurdina , Audita Bella Intan Puspita (2023)* ***" Naive Bayes and KNN for Airline Passenger Satisfaction Classification: Comparative Analysis*****"**

Nowadays, air transportation has become an important part of people's lives, and with the development of technology and the global economy, air transportation has become cheap and easy around the world. This study focuses on multi-channel online services. Previous research has shown that facilities and services affect passenger satisfaction. However, previous research suggests a positive relationship between satisfaction and service, but clear data are lacking. In this study, the results of Naive Bayes and K-Nearest Neighbor (K-NN) algorithms are compared in airline passenger classification.

The results show that the accuracy of the Naive Bayes method of the two algorithms is higher than the K-NN method. The accuracy of the Naive Bayes method is 84.48%, while the accuracy of the K-NN method is 65.38%. Looking at the test results, the accuracy of Naive Bayes is 82.25% and the accuracy of K-NN is 67.35%. Additionally, the improvement of Naive Bayes was 82.43% and the improvement of K-NN was 74.33%.

Airline users are affected by existing facilities and services. If the company can ensure that most things meet its performance standards, customers will be satisfied with the product or service. Customers will compare their services with other companies and if the company can provide satisfactory services, customers will be very satisfied and want to buy from the company again and will recommend the company to others. Therefore, businesses must consider the importance of customer service to survive and compete in the highly competitive transportation industry.

D. *A.C.Y. Hong , K.W. Khaw, X.Y. Chew , and W.C. Yeong (2023)* ***"Prediction of US airline passenger satisfaction using machine learning algorithms"***

The paper investigates the impact of service quality on passenger satisfaction in the U.S. aviation sector, particularly in the context of the COVID-19 pandemic and its aftermath. It highlights the significant downturn in the aviation industry in 2021, with a decrease in passenger numbers compared to pre-pandemic levels. The study aims to predict passenger satisfaction using machine learning algorithms and identify key factors correlated with satisfaction. The goal is to provide airlines with insights to enhance service quality and gain a competitive edge.

The literature review explores previous research on customer satisfaction prediction, particularly in the airline industry. Various machine learning algorithms, including Random Forest, K-Nearest Neighbor (KNN), AdaBoost, Decision Tree, Logistic Regression, and Naïve Bayes, have been employed in similar studies to predict satisfaction levels. The paper concludes that Random Forest is the most effective classifier for predicting passenger satisfaction in this study.The results suggest that features like 'Online boarding,' 'Inflight entertainment,' 'Seat comfort,' 'On-board service,' 'Leg room service,' 'Cleanliness,' 'Flight Distance,' and 'Inflight wifi service' are strongly correlated with passenger satisfaction. These findings can guide airlines in improving their services.

However, the study has limitations, such as the exclusion of some relevant factors from the dataset and the reliance on only four performance metrics. Future research could expand the

scope, consider additional features, and evaluate model performance using AUC and ROC metrics. Integrating the model into real-time systems for predictive purposes is also recommended. Overall, this research provides valuable insights for airlines to enhance passenger satisfaction and service quality.

E. *Bekee Sorbarisere Yirakpoa and Mercy Nwanyanwu (2022)* ***"Capstone Project : Marketing-Airplane Passenger Satisfaction Prediction Using Machine Learning Techniques"***

This research paper titled "Capstone Project: Marketing-Airplane Passenger Satisfaction Prediction Using Machine Learning Techniques" explores the prediction of customer satisfaction in the airline industry using various machine learning models. The study uses data from the US airline carrier 'Falcon airlines' and evaluates models such as Logistic Regression, Decision Tree, Bagging Classifier, and Random Forest.

The research is motivated by the need for airlines, particularly 'Falcon airlines,' to determine factors contributing to passenger satisfaction, especially in light of the post-pandemic surge in air travel demand. The project aims to understand the parameters influencing passenger satisfaction, predict passenger satisfaction, and compare different classification techniques.

The study's methodology involves analyzing a labeled dataset to identify predictors, visualizing data separation between satisfied and unsatisfied passengers, and experimenting with machine learning techniques. The results indicate that the Random Forest model outperforms others in terms of Recall and Precision.

The limitations include the reliance on a labeled dataset, limited exploration of unsupervised techniques, and the need for evaluating other algorithms. The paper also reviews related literature on airline passenger satisfaction and machine learning techniques.The research recommends that airlines focus on improving inflight entertainment, seat comfort, ease of online booking, and online support to enhance passenger satisfaction. Overall, the study provides a valuable reference for airlines seeking to improve customer satisfaction and make data-driven decisions in the competitive airline industry.

F. *So-Hyun Park , Mi-Yeon Kim , Yeon-Ji Kim and Young-Ho Park (2022)* ***"A Deep Learning Approach to Analyze Airline Customer Propensities: The Case of South Korea"***

The research paper investigates customer propensities in the airline industry, focusing on South Korea, by applying deep learning techniques to analyze survey data primarily from Korean airline users. It aims to understand the relationship between various factors influencing customer satisfaction and churn risk, emphasizing the impact of both physical and social aspects of airline servicescapes.

The study introduces a novel approach by incorporating deep learning methods alongside traditional machine learning techniques, demonstrating that deep learning models, particularly CNN-LSTM, outperform other models in predicting customer churn risk and satisfaction with high accuracy (94% and 90% respectively). Notably, it extends the analysis beyond the cabin crew's perspective to include passenger viewpoints, thereby enhancing the understanding of brand loyalty and customer satisfaction.

Key findings include insights into passenger preferences, such as a preference for Korean Air and Asiana Airlines over foreign carriers, despite higher prices, attributed to factors like cabin comfort and in-flight amenities. The study underscores the importance of the quality of airline servicescape in influencing customer satisfaction and provides actionable insights for service providers to improve their offerings.

Limitations include the focus on South Korean airlines and a limited set of factors considered, suggesting future research should explore international airlines and incorporate additional factors such as marketing and management aspects. Despite limitations, the study contributes to understanding customer propensities in the airline industry and offers a valuable starting point for leveraging deep learning techniques in analyzing customer data for various applications beyond aviation.

G. *Pedro José Freitas Reis (2023)* ***" Predicting air passenger satisfaction: a machine learning approach"***

The research paper explores the critical relationship between passenger satisfaction and various service dimensions within the air travel industry. Recognizing the industry's significance in global economic growth and connectivity, the study addresses challenges such as competition, cost management, and environmental sustainability that airlines encounter. Despite extensive research on service quality and passenger satisfaction, existing studies often focus on individual service dimensions rather than overall satisfaction prediction. This study fills this gap by developing predictive models using Classification Machine Learning algorithms to forecast passenger satisfaction accurately.

Using a dataset derived from an airline survey encompassing personal and general factors, the study creates two models: one incorporating only general factors accessible to airlines and another integrating personal responses. Both models demonstrate strong predictive capabilities compared to random guessing. The model without personal responses achieves an average accuracy of 79%, while the model incorporating personal responses performs even better, with an average accuracy of 93%. Notably, the Random Forest algorithm emerges as the most effective in both models, highlighting its significance in uncovering hidden patterns.

The findings underscore the importance of predictive models in identifying areas of dissatisfaction proactively, enabling airlines to enhance customer satisfaction and improve the overall passenger experience. By incorporating personal feedback from passengers, airlines can better understand and address specific service dimensions affecting satisfaction. However, limitations include the specificity of results to the surveyed airline and the potential impact of limited features on predictive performance.

Overall, the study contributes to the ongoing pursuit of passenger-focused strategies within the aviation industry, emphasizing the need to place passengers at the center of decision-making processes. By leveraging machine learning algorithms, airlines can tailor their offerings to meet diverse passenger needs effectively, ultimately fostering positive customer relationships and increasing loyalty.

H. *Xuchu Jiang , Ying Zhang , Ying Li & Biao Zhang(2022)* ***" Forecast and analysis of aircraft passenger satisfaction based on RF-RFE-LR model"***

The paper addresses the challenges faced by the civil aviation industry, particularly exacerbated by the COVID-19 pandemic, emphasizing the importance of predicting passenger satisfaction to enhance airline services and maintain competitiveness. It proposes a RF-RFE-Logistic feature selection model to extract influencing factors of passenger satisfaction from airline passenger surveys. Initially, the RF-RFE algorithm is employed to identify a subset of 17 variables, followed by comparing classification performance using various machine learning algorithms like KNN, logistic regression, random forest, Gaussian Naive Bayes, and BP neural network. The model employing random forest on the selected feature subset demonstrates the best classification performance, with an accuracy of 96.3%, precision of 97.3%, recall of 94.2%, F1 value of 95.7%, and AUC value of 96.1%. Further analysis highlights the importance of variables such as online boarding and onboard Wi-Fi services in determining passenger satisfaction across different passenger types and class categories.

While the study offers valuable insights and a predictive model for airlines to enhance passenger satisfaction, it acknowledges certain limitations. The evaluation indicators in the passenger satisfaction surveys may not be comprehensive, and the study's prediction models utilize default parameters without considering potential variations. Thus, recommendations are provided for future research, including broadening the scope of ground services evaluated, optimizing model parameters, and exploring additional factors impacting passenger satisfaction under abnormal flight conditions. Overall, the study provides a reference for airlines to accurately predict and understand passenger satisfaction, guiding strategies to improve service quality and competitiveness in the aviation industry.

2.2 Comprehensive Overview of the studies

Table 2.1 Literature Review Comparison Table

Title of Research Paper	Author(s) detail with year	Methodology used	Findings
Predicting Airline Passenger Satisfaction with Classification Algorithms	B. Herawan Hayadi, Jin-Mook Kim, Khodijah Hulliyah, and Husni Teja Sukmana, (2021)	Logistic regression ,Decision tree ,k-nearest , random forest	The paper predicts airline passenger satisfaction using classification algorithms, emphasizing in-flight service quality. Random Forest performs best (99% accuracy), highlighting Inflight Wi-Fi and online booking optimization for enhanced satisfaction.
A Research on Passengers' Satisfaction in Airways – In Coimbatore City	C. Murugesan and Dr. R. Perumalsamy(2017)	Calculating chi-square	The study explores factors influencing passenger satisfaction in Coimbatore's aviation sector. Passengers are generally satisfied but suggest improvements in waiting times and punctuality.
Naive Bayes and KNN for Airline Passenger Satisfaction Classification: Comparative Analysis	Annisa Fitria Nurdina , Audita Bella Intan Puspita (2023)	Naïve Bayes and KNN	The study examines airline passenger satisfaction in the context of multi-channel online services. Naive Bayes outperformed K-Nearest Neighbor in accuracy (84.48%

			vs. 65.38%). Customer service is vital for airline competitiveness and customer loyalty.
Prediction of US airline passenger satisfaction using machine learning algorithms	A.C.Y. Hong , K.W. Khaw, X.Y. Chew , and W.C. Yeong (2023)	LR, DTC , KNN ,random forest	The paper examines U.S. aviation sector satisfaction during COVID-19, predicting using Random Forest. Key factors include online services and cleanliness. KNN achieved an accuracy of 87.2%, Decision Tree Classifier (DTC) achieved 82%, Logistic Regression (LR) achieved 78%, Random Forest (RF) achieved 89.2%, and Adaboost achieved 82.8%.
Capstone Project : Marketing- Airplane Passenger Satisfaction Prediction Using Machine Learning Techniques	Bekee Sorbarisere Yirakpoa and Mercy Nwanyanwu (2022)	Random forest ,XGB	The paper predicts passenger satisfaction for 'Falcon Airlines' using machine learning models, highlighting Random Forest's superior performance in recall and precision. Decision Tree (DT) achieved 87%, Random Forest (RF) achieved 95%, and Logistic Regression (LR) achieved 75% accuracy.

A Deep Learning Approach to Analyze Airline Customer Propensities: The Case of South Korea	So-Hyun Park , Mi-Yeon Kim , Yeon-Ji Kim and Young-Ho Park (2022)	CNN-LSTM , KNN ,DST	The research applies deep learning to analyze Korean airline customer data, revealing factors influencing satisfaction and churn risk Random Forest (RF) achieved 84%, XGBoost (XGb) achieved 82%, Decision Tree Classifier (DTC) achieved 79%, and KNN achieved 84% accuracy.
Predicting air passenger satisfaction: a machine learning approach	Pedro José Freitas Reis (2023)	Gradient Boost ,AdaBoost	The research examines the relationship between service quality and passenger satisfaction in the air travel industry, developing predictive models using machine learning Decision Tree (DT) achieved 79%, Random Forest (RF) achieved 80.76%, and Logistic Regression (LR) achieved 78.38% accuracy.
Forecast and analysis of aircraft passenger satisfaction based on RF-RFE-LR model	Xuchu Jiang , Ying Zhang , Ying Li & Biao Zhang(2022)	RF_RFE logistic regression	The paper addresses challenges in the aviation industry exacerbated by COVID-19, proposing a RF-RFE-Logistic model to predict passenger satisfaction . Random Forest (RF) achieved 96%, Logistic Regression (LR) achieved 87%, and KNN achieved 93% accuracy.

Chapter 3: System Design and Methodology

3.1 System Design

Designing a project on "Airline Passenger Satisfaction Comparative Analysis" using logistic regression, decision tree classifier, and random forest technique involves several steps and considerations. Here's a high-level system design and methodology for this project:
System Design:

Gather data on airline passenger satisfaction from various sources such as surveys, online reviews, or publicly available datasets. Ensure the data includes relevant features like flight details, passenger demographics, and satisfaction ratings.

Clean the data by handling missing values, outliers, and data inconsistencies.Encode categorical variables using techniques like one-hot encoding or label encoding.Normalize or standardize numerical features as required.

Divide the dataset into training, validation, and test sets. Typically, you can use a 70-15-15 or 80-10-10 split.

Perform feature selection to identify the most relevant features that affect passenger satisfaction .Create new features if needed, e.g., combining features or extracting meaningful information.

Choose logistic regression, decision tree classifier, and random forest as the three models for the comparative analysis.

Train each selected model using the training dataset. Tune hyperparameters for decision tree and random forest models using techniques like grid search or random search.

Evaluate the models on the validation dataset using appropriate metrics (e.g., accuracy, precision, recall, F1-score, ROC AUC) to assess their performance.

Compare the performance of logistic regression, decision tree, and random forest models based on the chosen evaluation metrics. Identify strengths and weaknesses of each model for predicting passenger satisfaction.

Consider using ensemble techniques like stacking or bagging to combine the predictions of multiple models for potentially improved performance.

Based on the comparative analysis and performance metrics, select the most suitable model for the task. Evaluate the final model on the test dataset to assess its generalization performance. Create visualizations (e.g., confusion matrices, ROC curves) to present the results of the analysis.

3.1.1 System Architecture

To architecture for the comparative analysis of airline passenger satisfaction using logistic regression, decision tree classifier, and random forest techniques with the given dataset, the following steps have been followed :

The dataset is collected containing passenger satisfaction survey data and stored in a data repository or database.

A component for data preprocessing is created , which includes tasks such as data cleaning, handling missing values, and feature engineering. Transform the categorical features into numerical representations (e.g., one-hot encoding or label encoding).Split the dataset into training, validation, and test sets.

Designed separate pipelines for logistic regression, decision tree classifier, and random forest models. Create a Model Training component that includes model training and hyperparameter tuning (if applicable, for decision tree and random forest models). Evaluated the models using appropriate evaluation metrics such as accuracy, precision, recall and F1-score.

Developed a feature importance analysis component to determine which factors are highly correlated with passenger satisfaction for each model. Visualizations like feature importance plots can be generated.

Created a component to compare the performance of the three models and identify which one performs best in predicting passenger satisfaction. Visualization tools can help in presenting these comparisons.

Selected the best-performing model based on the comparison results. Deploy the chosen model to make predictions on new data or integrate it into airline systems for real-time analysis.

Implemented monitoring and logging to track model performance and system health over time. Set up alerts for model degradation or anomalies.

Documented the entire system architecture, including data preprocessing steps, model configurations, and evaluation results. Generate reports summarizing the findings and conclusions of the comparative analysis.

Consider implementing scalability features to handle larger datasets and optimize the system for efficiency. Explore techniques for model versioning and management for easy updates.

If the dataset contains sensitive information, implement security measures to protect data privacy and comply with relevant regulations. Establish backup and recovery procedures to ensure data and model integrity in case of failures or disasters.

Thorough data exploration and validation is conducted to ensure the dataset's quality and integrity. Additionally, considered using cross-validation techniques during model training and rigorous testing before deploying any model to production. The architecture can be adapted and extended to accommodate specific project requirements and constraints

3.1.2 Block Diagram

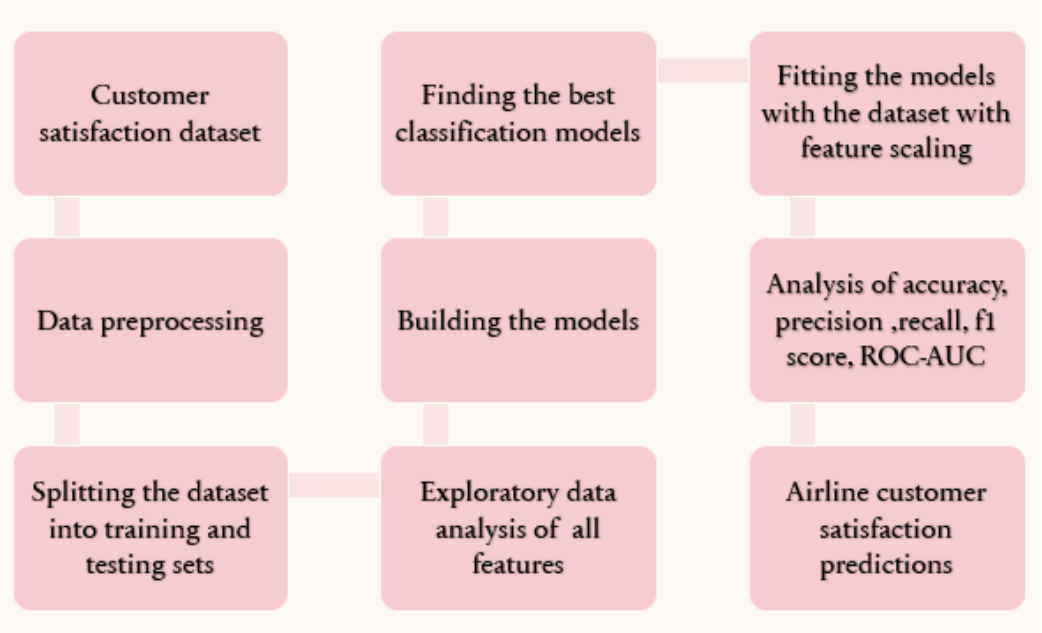


Figure 3.0 Block Diagram

3.1.3 Data Sources

The data source consists of a downloaded CSV file that is available on Kaggle website. The reference is given as below.

Link: https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction

3.1.4 Data Preprocessing:

Data preprocessing steps that were followed to prepare the data for modeling are as follows.

- Data cleaning
- Feature Engineering
- Feature Transformation
- Handling Missing, Null, Duplicate values.

3.1.5 Challenges and Trade-offs:

The only challenge we encountered was while removing the outliers as the data was already clean enough and thus removing the outliers leaning the model toward biasness thus we had to make a trade off with the outliers and didn't remove them.

3.2 Algorithm Used

3.2.1 Logistic Regression Algorithm

Logistic Regression is a widely used statistical method for binary classification. It models the probability of a binary outcome (in this case, passenger satisfaction) based on one or more predictor variables (features). The outcome is usually represented as a binary variable where '0' indicates one class (e.g., dissatisfaction) and '1' indicates another class (e.g., satisfaction). Logistic Regression is a widely used classification algorithm that models the probability of a binary outcome (in this case, passenger satisfaction) as a function of independent variables.

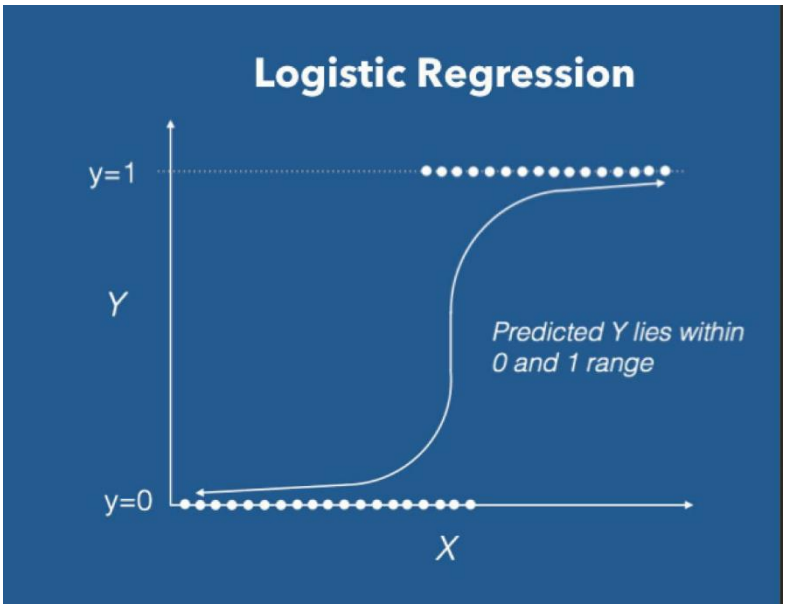


Figure 3.1 Logistic Regression

3.2.2 Decision Tree classification Algorithm

The decision tree algorithm in linear regression is a method that employs a tree-like structure to make predictions in regression tasks. Unlike traditional linear regression, which uses a continuous function to predict output, decision trees recursively split the input space into smaller regions and fit a simple model within each of these regions.

Here's how the algorithm works:

Splitting Nodes: The decision tree starts with the entire dataset and selects the best feature and split point to create two child nodes.

Node Splitting Criteria: For linear regression in decision trees, this usually involves minimizing the variance of the target variable within the regions created by the splits.

Leaf Nodes: Eventually, the splitting stops when a predefined stopping criterion is reached, such as a maximum depth of the tree, a minimum number of samples in a node, or no further gain in variance reduction.

Prediction: In the case of linear regression, the prediction at each leaf node is the mean (or any linear function) of the target variable values within that node.

Model Interpretation: While decision trees themselves might not inherently provide coefficients or coefficients in the way linear regression does, understanding how the tree makes decisions can still offer insights into feature importance and relationships between predictors and the target variable.

In linear regression within decision trees, the idea is to approximate the relationships between features and the target variable within local regions rather than using a global linear function. This approach can capture non-linear relationships effectively and handle interactions between features, making it a powerful tool for regression tasks.

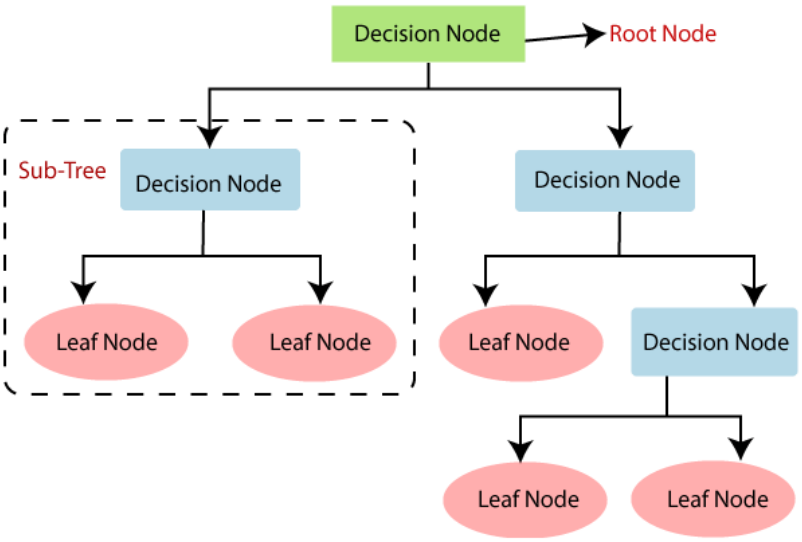


Figure 3.2 Decision Tree Classification

3.2.3 Random Forest classification Algorithm

The Random Forest Regressor is a powerful ensemble learning algorithm used in machine learning for regression tasks. It's constructed from multiple decision trees, creating a forest of trees that work together to make predictions.

Training Process:

1. **Ensemble of Trees:** It builds a multitude of decision trees during training, each trained on a random subset of the dataset (bootstrap aggregating or "bagging"). This randomness helps introduce diversity among the trees.
2. **Random Feature Selection:** At each node of every tree, instead of considering all features for splitting, a random subset of features is chosen. This enhances independence among the trees and reduces the risk of overfitting.
3. **Decision Tree Training:** Each tree is trained on a subset of the data, and for each split in the tree, the algorithm selects the best feature among the randomly chosen subset of features.

Prediction Process:

When making predictions:

- Each tree in the forest predicts an outcome.
- For regression tasks, the predicted values from individual trees are averaged (for example, taking the mean) to get the final prediction.

Benefits:

- **Accuracy:** Random Forests tend to deliver high accuracy and robustness due to their ensemble nature, reducing overfitting compared to individual decision trees.
- **Versatility:** Suitable for a wide range of problems, handling both categorical and numerical data.
- **Feature Importance:** It provides insight into feature importance, indicating which features contribute most to predictions.

Random Forest Regressors are effective for various applications like predicting stock prices, real estate valuation, medical diagnosis, and more. They excel in scenarios where high accuracy and robustness are required, and understanding feature importance is valuable. Adjusting parameters like the number of trees, depth, and feature selection criteria can further optimize their performance for specific tasks.

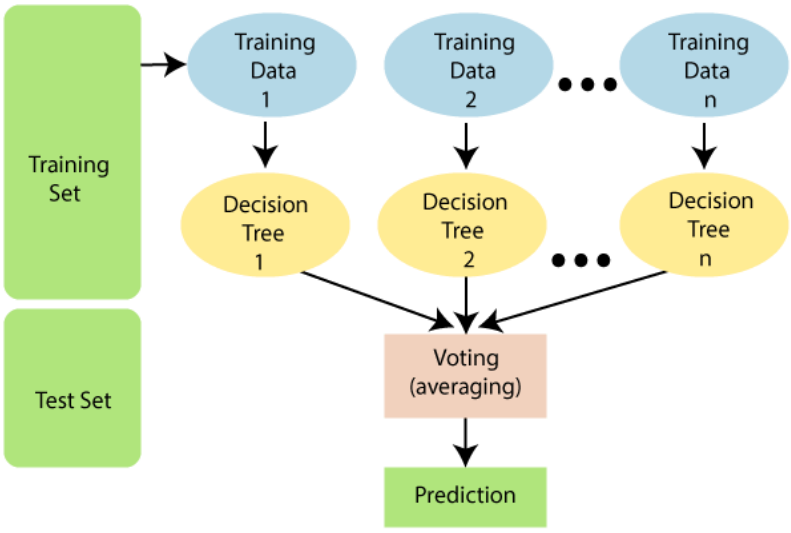


Figure 3.3 Random Forest Classification

3.2.4 AdaBoost Classifier

The AdaBoost Classifier is a powerful ensemble learning method used in classification tasks. It operates similarly to the AdaBoost Regressor but is tailored for classification problems. Here's how it works:

Base Estimator: Initially, each training instance is assigned an equal weight, and a weak learner (e.g., a simple decision tree classifier) is trained on the data. The weak learner aims to classify the target variable.

Instance Weight Adjustment: After the first iteration, the algorithm adjusts the weights of incorrectly classified instances, increasing the emphasis on those that were misclassified and decreasing the emphasis on correctly classified instances. This adjustment directs subsequent learners to focus more on the challenging instances.

Sequential Model Fitting: In subsequent iterations, new weak learners are trained, with a greater focus on the instances that were previously misclassified or have higher weights. Each new learner aims to correctly classify the instances that were difficult for the previous models.

Model Combination: The final model is a weighted combination of these weak learners. Each learner contributes to the final classification based on its performance in the iterations. In classification, the combination may involve voting or weighted voting of individual weak classifiers.

The AdaBoost Classifier aims to sequentially improve the overall classification performance by iteratively emphasizing misclassified instances and learning from them. By combining multiple weak classifiers, it constructs a strong classification model capable of handling complex decision boundaries while mitigating overfitting. This technique is particularly effective in scenarios where the base classifiers have limited predictive power on their own but can collectively contribute to accurate predictions when combined.

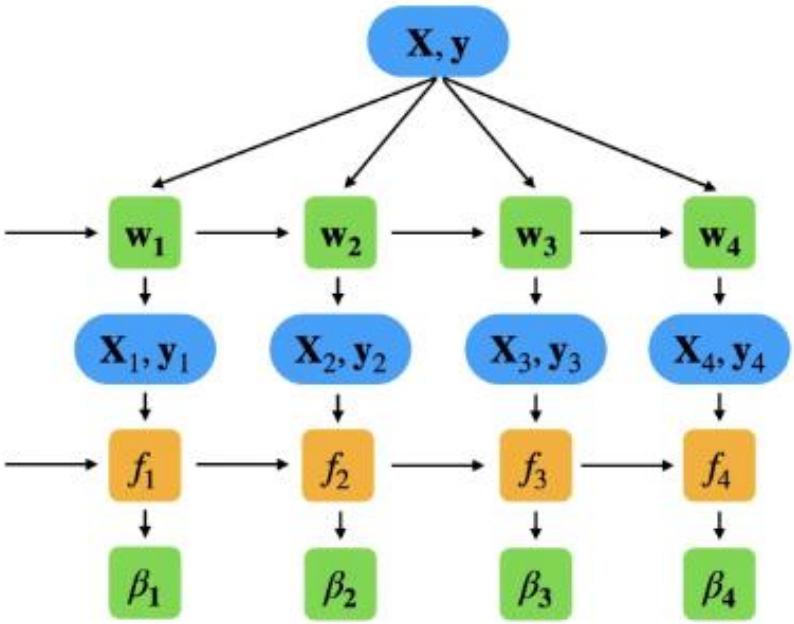


Figure 3.4 AdaBoost Classifier

3.2.5 XGB Classifier

XGBClassifier, short for eXtreme Gradient Boosting Classifier, is a formidable machine learning algorithm renowned for its efficiency, speed, and accuracy in tackling classification tasks. Here's an in-depth look at XGBClassifier:

XGBClassifier operates on the principle of gradient boosting, sequentially amalgamating multiple weak learners (usually decision trees) to forge a potent predictive model.

Tree Ensemble: It constructs an ensemble of decision trees iteratively, with each subsequent tree endeavoring to rectify the errors committed by its predecessors. To ward off overfitting, these decision trees are typically shallow, ensuring limited depth.

Objective Function: XGBClassifier employs a specific objective function, often tailored for classification tasks like binary logistic regression or softmax for multiclass classification. It optimizes this function using gradient descent, systematically minimizing residuals to refine the model's accuracy.

Regularization: To stave off overfitting and bolster generalization capabilities, XGBClassifier incorporates regularization techniques such as shrinkage (via learning rate adjustment), subsampling, and column sampling.

Parallel Computing and Optimization: Engineered for swift execution, XGBClassifier supports parallel processing and offers avenues for hardware optimization. These optimizations render it notably faster than many other boosting algorithms.

Hyperparameter Tuning: Fine-tuning hyperparameters is pivotal for optimizing performance. Parameters like tree depth, learning rate, and regularization settings can be fine-tuned to enhance the model's predictive prowess.

XGBClassifier reigns supreme in classification tasks, adept at handling voluminous datasets, disentangling intricate relationships, and achieving stellar predictive accuracy—all while upholding computational efficiency.

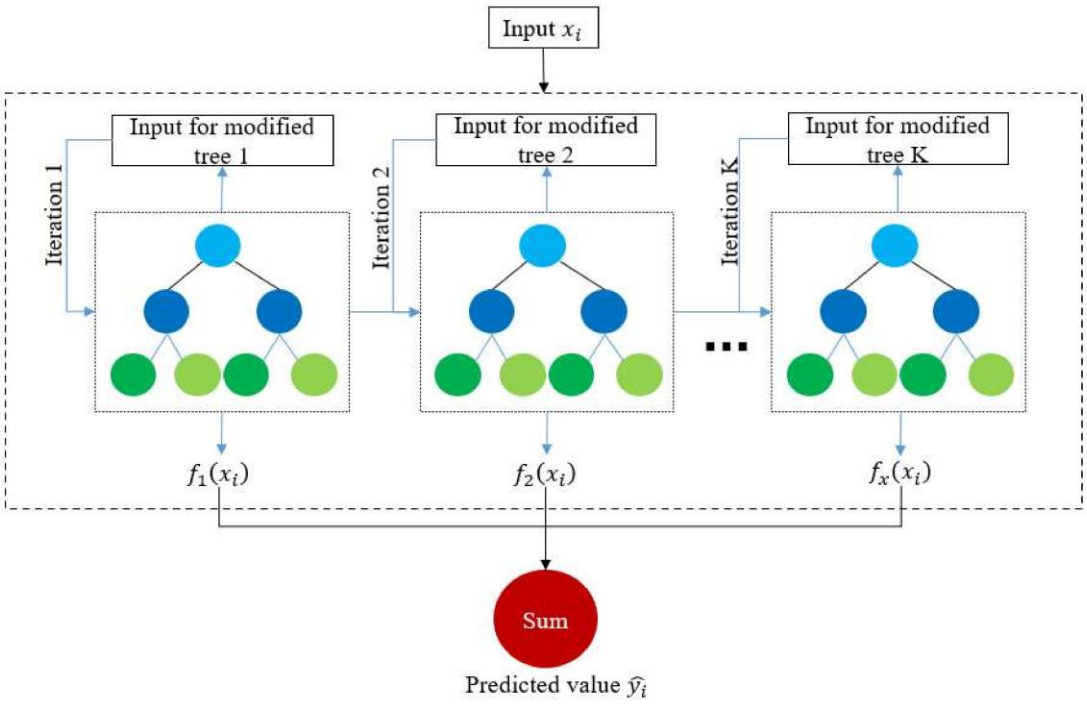


Figure 3.5 XGB Classifier

3.2.6 Gradient Boosting

Gradient Boosting is a powerful machine learning technique used for regression and classification tasks. It works by combining multiple weak learners, typically decision trees, into a single strong learner in an iterative manner. Each new learner focuses on the mistakes made by the previous ones, thereby reducing the overall error.

At each iteration, the algorithm fits a new tree to the residuals (the difference between the predicted and actual values) of the previous model. The predictions from all the trees are then combined to make the final prediction.

Gradient Boosting optimizes a loss function by finding the best parameters for each new tree. It emphasizes model interpretability, handles missing data well, and is robust to outliers. However, it may require careful tuning of hyperparameters and can be computationally expensive due to its sequential nature. Overall, Gradient Boosting is widely used for its high predictive accuracy and flexibility.

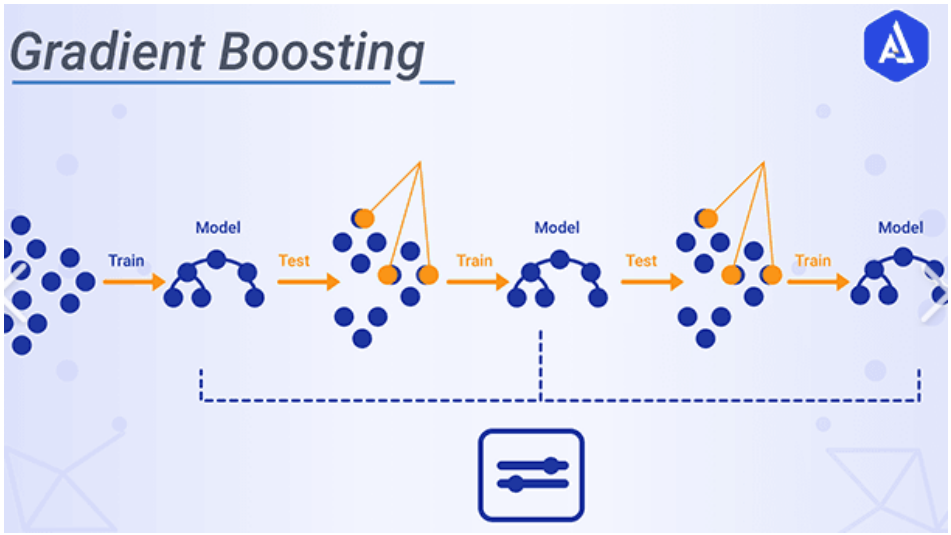


Figure 3.6 Gradient Boosting Classifier

3.2.7 K nearest neighbour

K-Nearest Neighbors (KNN) is a simple yet effective machine learning algorithm used for both classification and regression tasks. It operates on the principle of similarity: objects with similar characteristics are likely to be in the same class or have similar values.

In KNN, when a new data point is to be classified or predicted, the algorithm identifies the K nearest neighbors to that point based on a chosen distance metric (e.g., Euclidean distance). Then, it assigns the majority class label (for classification) or averages the values (for regression) of these neighbors to the new data point.

KNN is intuitive and easy to understand, but its performance can be sensitive to the choice of K and the distance metric. Additionally, it can be computationally expensive for large datasets since it requires calculating distances to all training samples. Nonetheless, KNN remains a popular choice for its simplicity and effectiveness in many applications.

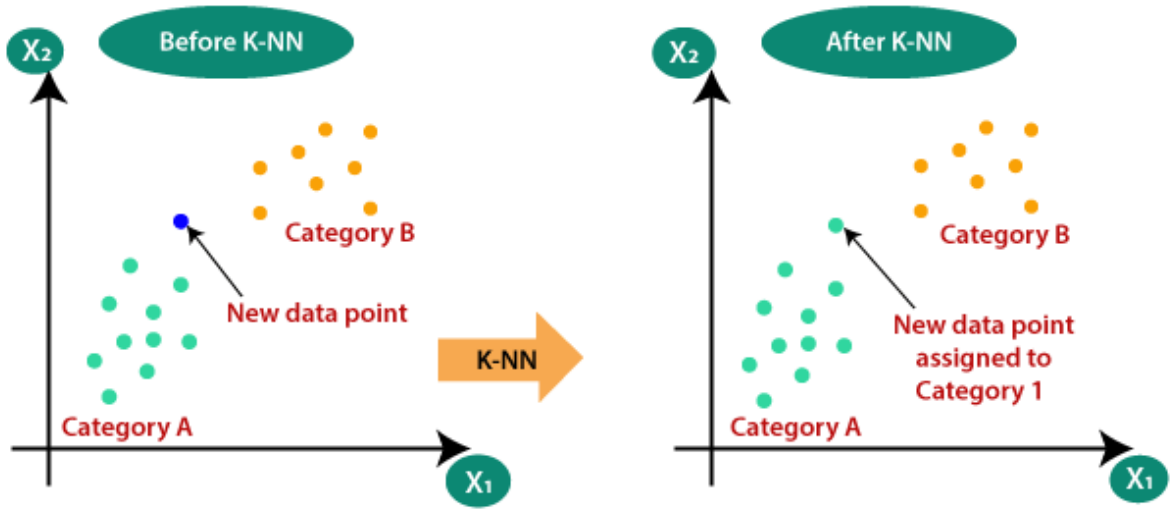


Figure 3.7 K-Nearest Neighbour Classifier

3.2.8 MLP classifier

A Multilayer Perceptron (MLP) classifier is a type of artificial neural network (ANN) that consists of multiple layers of nodes (or neurons) arranged in a feedforward manner. Each node in one layer is connected to every node in the subsequent layer, and each connection is associated with a weight.

MLP classifiers are capable of learning complex patterns in data and are widely used for classification tasks. During training, the network adjusts the weights of these connections through a process called backpropagation, where the errors between the predicted and actual outputs are used to update the weights.

MLP classifiers can have one or more hidden layers, which allow them to learn nonlinear relationships in the data. Activation functions are applied to the output of each node to introduce nonlinearity and enable the network to approximate complex functions.

MLP classifiers are versatile and can handle various types of data, but they may require careful tuning of hyperparameters, such as the number of layers, nodes per layer, and activation functions, to achieve optimal performance.

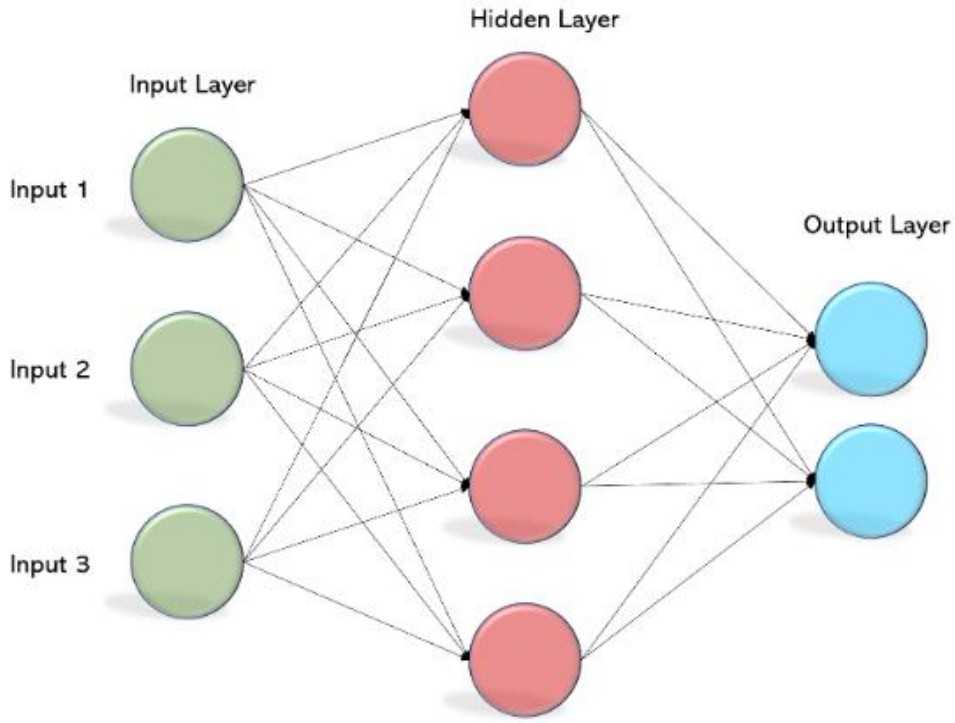


Figure 3.8 MLP Classifier

3.2.9 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a statistical technique used for dimensionality reduction and classification. It seeks to find the linear combinations of features that best separate two or more classes in a dataset.

In LDA, the goal is to maximize the ratio of between-class variance to within-class variance. This is achieved by projecting the data onto a lower-dimensional subspace while maximizing the distance between the means of different classes and minimizing the spread within each class.

LDA assumes that the data within each class follows a Gaussian distribution with equal covariance matrices for all classes. It also assumes that the classes have similar covariance matrices.

LDA is particularly useful when the classes are well-separated and the assumptions of normality and equal covariance matrices hold true. It is often used as a preprocessing step for machine learning algorithms or as a standalone classifier.

Unlike other dimensionality reduction techniques like Principal Component Analysis (PCA), LDA takes into account the class labels and aims to preserve the discriminative information in the data.

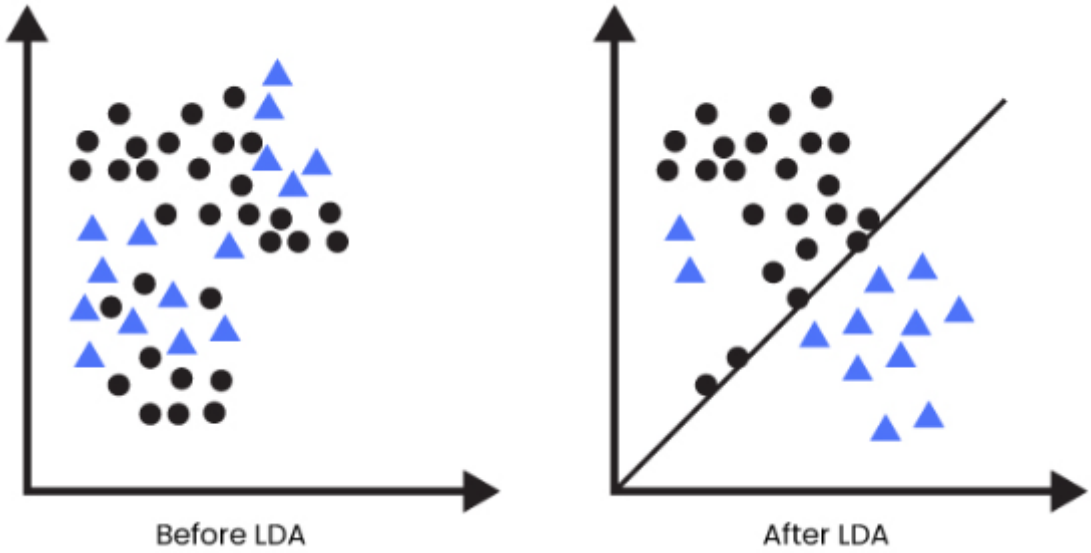


Figure 3.9 Linear Discriminant Analysis

3.2.10 Quadratic Discriminant Analysis

Quadratic Discriminant Analysis (QDA) is a statistical classification technique similar to Linear Discriminant Analysis (LDA) but with a key difference: it relaxes the assumption of equal covariance matrices across classes.

In QDA, each class is assumed to have its own covariance matrix, allowing for more flexibility in modeling the distribution of each class. This means that QDA can capture more complex relationships between features within each class compared to LDA, making it potentially more powerful when the classes have different variances or covariances.

Like LDA, QDA also aims to classify observations into predefined classes based on their features. It does so by estimating the parameters of the Gaussian distributions for each class and then using Bayes' theorem to calculate the posterior probabilities of class membership.

QDA is particularly useful when the classes have distinct and non-linear decision boundaries or when the assumption of equal covariance matrices in LDA is violated. However, QDA may require more training data to accurately estimate the parameters of each class's covariance matrix, and it may be more computationally intensive compared to LDA due to the additional parameters to estimate.

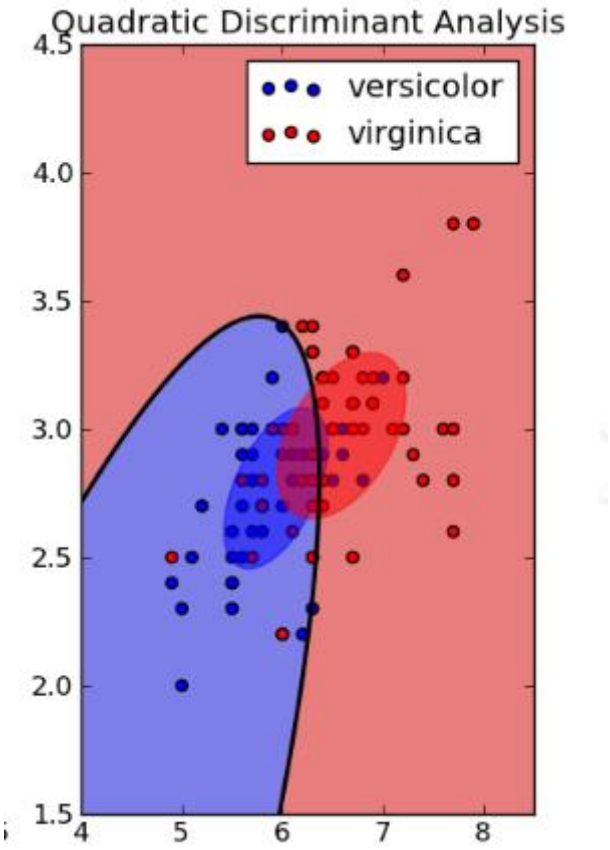


Figure 3.10 Quadratic Discriminant Analysis

3.3 Methodology

To apply various machine learning algorithms such as Random Forest, Decision Tree, AdaBoost, Gradient Boosting, XGBoost Regressor, K-Nearest Neighbors (KNN), Linear Discriminant Analysis (LDA), and Quadratic Discriminant Analysis (QDA) to this dataset, you would typically follow these general steps:

Data Preprocessing :

- Handle missing values: impute or remove them.

- Encode categorical variables: convert categorical variables into numerical format using techniques like one-hot encoding or label encoding.
- Scale numerical features if necessary: standardize or normalize them to ensure all features have the same scale.

Feature Selection/Engineering:

- Identify relevant features for the prediction task.
- Create new features if necessary to capture additional information from the data.

Splitting Data:

Divide the dataset into training and testing sets to evaluate the model's performance.

Model Selection and Training:

- Choose appropriate algorithms based on the nature of the problem (classification or regression).
- Train the selected models on the training data.

Model Evaluation:

- Evaluate the performance of each model using appropriate metrics such as accuracy, precision, recall, F1-score, or Mean Absolute Error (MAE), Mean Squared Error (MSE), etc.
- Tune hyperparameters of the models using techniques like grid search or random search to improve performance.

Model Comparison and Selection:

- Compare the performance of different models to select the best-performing one.
- Here's how you can specifically apply each algorithm:

  - **Random Forest, Decision Tree, AdaBoost, Gradient Boosting, XGBoost Regressor:** These algorithms are tree-based ensemble methods. They can handle both classification and regression tasks. You can apply them directly to the dataset after preprocessing and evaluate their performance using appropriate evaluation metrics.

- **K-Nearest Neighbors (KNN)**: KNN is a non-parametric lazy learning algorithm used for classification and regression tasks. You can apply KNN directly to the dataset after preprocessing. It's important to choose an appropriate value of K and evaluate its performance.

- **Linear Discriminant Analysis (LDA)**: LDA is a dimensionality reduction technique and a classifier. You can apply LDA after preprocessing the data and evaluate its classification performance.

- **Quadratic Discriminant Analysis (QDA)**: QDA is similar to LDA but relaxes the assumption of equal covariance matrices across classes. You can apply QDA after preprocessing and evaluate its classification performance.

Each algorithm has its own strengths and weaknesses, so it's important to experiment with multiple algorithms and select the one that best fits the dataset and the problem at hand. Additionally, proper tuning of hyperparameters can significantly impact the performance of each algorithm.

Remember to iterate on the process, fine-tune models, and consider potential biases in the data to ensure a robust and unbiased analysis of airline passenger satisfaction.

Chapter 4: Implementation & Results

4.1 Hardware and Software Requirements

In this chapter, we delve into the technical aspects of the project, outlining the hardware and software requirements necessary for the successful implementation of the Airline Passenger satisfaction comparative analysis using supervised learning techniques.

4.1.1 Hardware Requirements

The hardware specifications for the project are as follows:

Table 4.1 Hardware Requirements

Minimum Hardware Requirements	
Processor	Intel(R) Core(TM) I5 or equivalent
CPU	1.60ghz
Memory	At least 2.00GB
Hard Disk	500GB
Display	Super VGA (1366 × 768) / higher resolution monitor
Input Devices	Keyboard, Mouse

Hardware Requirements:

- **Processor:** The minimum processor requirement specified is an Intel Core i5 or its equivalent. This processor should be capable enough to handle the computational demands of running the machine learning algorithms and processing the dataset.
- **CPU:** The CPU speed of at least 1.60GHz ensures that the processor can perform tasks efficiently within a reasonable timeframe.
- **Memory (RAM):** A minimum of 2.00GB of RAM is necessary to accommodate the running processes and datasets in memory during model training and evaluation.

- **Hard Disk:** The specified 500GB hard disk provides ample storage space for storing datasets, model files, and other project-related files.
- **Display:** A Super VGA monitor with a resolution of 1366x768 or higher is recommended to ensure clear visibility of code, data, and results during the development and analysis phases.
- **Input Devices:** Basic input devices such as a keyboard and mouse are required for interacting with the computer system and executing commands.

4.1.2 Software Requirements

The software tools and libraries required for the project

Table 4.2 Software Requirements

Minimum Software Requirements	
Frontend	Python
Browser	Mozilla Firefox, Google Chrome etc
Development tool	Jupyter Notebook, Google Colab, Anaconda

Software Requirements:

- **Frontend:** Python serves as the primary programming language for the project. Python is widely used in machine learning and data analysis due to its extensive libraries and ease of use.
- **Browser:** Any modern web browser such as Mozilla Firefox or Google Chrome is required for accessing online resources, documentation, and possibly for web-based data analysis platforms.
- **Development Tool:** The specified development tools include Jupyter Notebook, Google Colab, and Anaconda. These tools provide interactive environments for writing and executing code, visualizing data, and documenting the analysis process. They are popular choices among data scientists and machine learning practitioners for their convenience and functionality.

4.2 Implementation Details

This section provides an in-depth description of the implementation process, including data collection, data preprocessing, logistic regression model development, decision tree classification model development, random forest classification model development and the analysis of results.

4.2.1 Data Collection

Data collection in data science refers to the process of gathering, acquiring, and recording data from various sources for the purpose of analysis, interpretation, and decision-making. This dataset is extracted from "Kaggle" and our dataset is about "Airline Passenger Satisfaction" in which we analyze the passenger satisfaction. So, I have considered a labelled dataset for applying supervised machine learning technique.

```
# importing essential libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Load and check dataset
```

```
# Loading the dataset
df=pd.read_csv('project.csv')
df
```

	Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service
0	0	70172	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3
1	1	5047	Male	disloyal Customer	25	Business travel	Business	235	3
2	2	110028	Female	Loyal Customer	26	Business travel	Business	1142	2
3	3	24026	Female	Loyal Customer	25	Business travel	Business	562	2
4	4	119299	Male	Loyal Customer	61	Business travel	Business	214	3

```
# shape of dataset
df.shape
```

```
(103904, 24)
```

4.2.2 Data Preparation

Data preparation, also known as data preprocessing or data cleaning, is a crucial step in the data science workflow. It involves transforming raw data from various sources into a format that is suitable for analysis, modeling, and machine learning. We prepare raw data in this phase so that meaningful insights can be extracted from it.

- Feature Selection: Relevant features for predicting passenger satisfaction are selected based on domain knowledge and preliminary data analysis.
- Data Cleaning and Encoding:
    - Missing values are handled if present.
    - Categorical features like 'Gender,' 'Customer Type,' 'Type of Travel,' and 'Class' are one-hot encoded.
    - Numerical features are scaled to have zero mean and unit variance for better convergence.
    - The target variable 'Satisfaction' is encoded into binary labels (0 for dissatisfaction, 1 for satisfaction).

Checking for null values in the dataset

```
# finding null values
df.isnull().sum()
```

```
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
Baggage handling                       0
Checkin service                        0
Inflight service                       0
Cleanliness                            0
Departure Delay in Minutes             0
Arrival Delay in Minutes             310
satisfaction                           0
dtype: int64
```

```
# finding mean of the column with null values
import numpy as np
np.mean(df['Arrival Delay in Minutes'])
```

```
15.178678301832152
```

```
# filling the null values using fillna method
df["Arrival Delay in Minutes"]=df["Arrival Delay in Minutes"].fillna(np.mean(df["Arrival Delay in Minutes"]))
```

Checking for Duplicate values:

```
# finding duplicates in the data
df.duplicated().sum()
```

```
0
```

```
# accessing unique values in the particular column
df['satisfaction'].unique()
```

```
array(['neutral or dissatisfied', 'satisfied'], dtype=object)
```

```
# categorical values to numeric values using map() function
df['satisfaction']=df['satisfaction'].map({'satisfied':1,'neutral or dissatisfied':0})
df
```

4.2.3 Logistic Regression Model Development

The logistic regression model is implemented using Python and the following libraries:

- pandas: For data manipulation and preprocessing.
- numpy: For numerical operations.
- scikit-learn: For implementing the logistic regression algorithm and evaluation metrics.

Data Splitting: The dataset is split into training and testing sets to evaluate the model's performance.

```
# train test split
from sklearn.model_selection import train_test_split
```

```
X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.11,random_state=40)
```

Model Training : The logistic regression model is trained on the training dataset.

```
#training model

classifier_regressor.fit(X_train, y_train)
```

```
from sklearn.linear_model import LogisticRegression
```

```
classifier = LogisticRegression()
classifier
```

```
▾ LogisticRegression
LogisticRegression()
```

Model Evaluation : The model's performance is evaluated using various classification metrics on the test dataset.

```
from sklearn.model_selection import GridSearchCV

parameter = {'penalty': ['l1','l2','elasticnet'],
             'C': [1,2,3,4,5,6,10,20,30,40,50],
             'max_iter': [100,200,300]}
```

```
classifier_regressor = GridSearchCV(classifier, param_grid = parameter, scoring = 'accuracy', cv = 5)
```

```
print(classifier_regressor.best_params_)
```

```
{'C': 6, 'max_iter': 300, 'penalty': 'l2'}
```

```
print(classifier_regressor.best_score_)
```

```
0.7565611181172416
```

Checking Accuracy score:

```
#accuracy score

from sklearn.metrics import accuracy_score, classification_report
score = accuracy_score(y_pred, y_test)
score
```

```
0.7425254452926209
```

```
print(classification_report(y_pred, y_test))
```

```
              precision    recall  f1-score   support

           0       0.49      0.74      0.59      1580
           1       0.89      0.74      0.81      4708

    accuracy                           0.74      6288
   macro avg       0.69      0.74      0.70      6288
weighted avg       0.79      0.74      0.76      6288
```

Interpretation

Logistic regression coefficients can be interpreted to understand the impact of each feature on passenger satisfaction. Positive coefficients increase the log-odds of satisfaction, while negative coefficients decrease it.

This logistic regression implementation successfully predicts airline passenger satisfaction based on various features. The model's performance is evaluated using standard classification metrics, and feature coefficients provide insights into the factors influencing passenger satisfaction.

4.2.4 Decision Tree classification Model Development

The Decision Tree Classification model is implemented using Python and the following libraries:

- pandas: For data manipulation and preprocessing.
- numpy: For numerical operations.
- scikit-learn: For implementing the Decision Tree Classifier and evaluation metrics.

Model Training :

The Decision Tree Classification model is trained on the training dataset.

```
from sklearn.tree import DecisionTreeClassifier
```

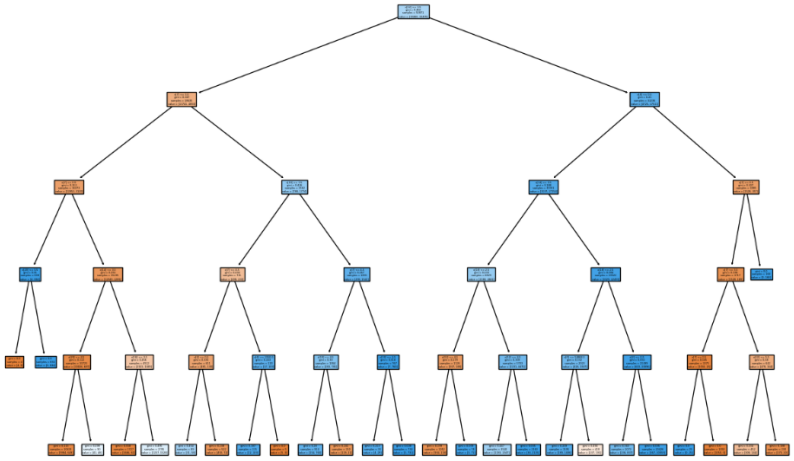
```
# Post pruning technique
# treemodel=DecisionTreeClassifier()
treemodel=DecisionTreeClassifier(max_depth=5)
treemodel
```

```
DecisionTreeClassifier
DecisionTreeClassifier(max_depth=5)
```

```
treemodel.fit(X_train,y_train)
```

```
DecisionTreeClassifier
DecisionTreeClassifier(max_depth=5)
```

```
from sklearn import tree
plt.figure(figsize=(15,10))
tree.plot_tree(treemodel,filled=True)
```



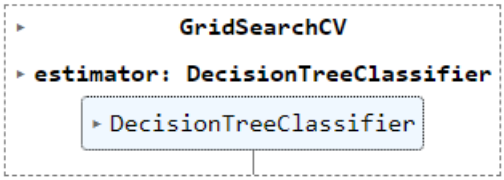
Model Evaluation

The model's performance is evaluated using various classification metrics on the test dataset.

```
from sklearn.model_selection import GridSearchCV
```

```
treemodel=DecisionTreeClassifier()
cv=GridSearchCV(treemodel,param_grid=parameter,cv=5,
                scoring='accuracy')
```

```
cv.fit(X_train,y_train)
```



Checking Accuracy score:

```
y_pred=cv.predict(X_test)
```

```
from sklearn.metrics import accuracy_score,classification_report
```

```
score=accuracy_score(y_pred,y_test)
```

```
score
```

```
0.6232506361323156
```

Interpretation

Decision Trees can be visualized to understand how decisions are made based on feature values. You can visualize the tree structure to see the decision rules at each node.

This Decision Tree Classification implementation successfully predicts airline passenger satisfaction based on various features. The model's performance is evaluated using standard classification metrics, and the decision tree can be visualized to understand the decision-making process.

4.2.5 Random Forest classification Model Development

The Random Forest Classification model is implemented using Python and the following libraries:

- pandas: For data manipulation and preprocessing.
- numpy: For numerical operations.
- scikit-learn: For implementing the Random Forest Classifier and evaluation metrics.

Model Training

The Random Forest Classification model is trained on the training dataset.

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
```

```
model=RandomForestClassifier()
model.fit(X_train,y_train)
```

```
▾ RandomForestClassifier
RandomForestClassifier()
```

Model Evaluation

The model's performance is evaluated using various classification metrics on the test dataset.

```
# predictions
predictions=model.predict(X_test)
predictions
```

```
array([1, 1, 1, ..., 1, 1, 0], dtype=int64)
```

```
# accuracy

model_score=accuracy_score(y_test,predictions)
model_score
```

```
0.975031806615776
```

This Random Forest Classification implementation predicts airline passenger satisfaction based on various features. The model's performance is evaluated using standard classification metrics, and feature importances provide insights into the factors influencing passenger satisfaction.

4.2.6 Gradient Boosting classification Model Development

The Gradient Boosting Classification model is implemented using Python and the following libraries:

- pandas: For data manipulation and preprocessing.
- numpy: For numerical operations.
- scikit-learn: For implementing the Random Forest Classifier and evaluation metrics.

Model Training

The model is trained on the training dataset.

```
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import accuracy_score
```

```
gb_model=GradientBoostingClassifier()
gb_model.fit(X_train, y_train)
gb_predictions = gb_model.predict(X_test)
gb_predictions
```

```
C:\Users\hp\anaconda3\lib\site-packages\sklearn\ensemble\_gb.py:
sionWarning: A column-vector y was passed when a 1d array was ex
change the shape of y to (n_samples, ), for example using ravel(
  y = column_or_1d(y, warn=True)
```

```
array([1, 1, 1, ..., 1, 1, 0], dtype=int64)
```

Model Evaluation

The model's performance is evaluated using various classification metrics on the test dataset.

```
gbc = GradientBoostingClassifier(n_estimators=300,
                                 learning_rate=0.05,
                                 random_state=100,
                                 max_features=5 )
# Fit to training set
gbc.fit(X_train, y_train)

# Predict on test set
gb_predictions = gbc.predict(X_test)

# accuracy
acc = accuracy_score(y_test, gb_predictions)
print("Gradient Boosting Classifier accuracy is : {:.2f}".format(acc))
```

```
C:\Users\hp\anaconda3\lib\site-packages\sklearn\ensemble\_gb.py:437: DataConve
sionWarning: A column-vector y was passed when a 1d array was expected. Please
change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

```
Gradient Boosting Classifier accuracy is : 0.95
```

4.2.7 AdaBoost classifier Model Development

The AdaBoost Classification model is implemented using Python and the following libraries:

- pandas: For data manipulation and preprocessing.
- numpy: For numerical operations.
- scikit-learn: For implementing the Random Forest Classifier and evaluation metrics.

Model Training

The model is trained on the training dataset.

```
X_train, X_val, Y_train, Y_val = train_test_split(
    x, y, test_size=0.25, random_state=28)
```

```
from sklearn.ensemble import AdaBoostClassifier
# Creating adaboost classifier model

adb = AdaBoostClassifier()
adb_model = adb.fit(X_train,Y_train)
```

```
model=AdaBoostRegressor()
```

```
model.fit(x_train,y_train)
```

```
AdaBoostRegressor
AdaBoostRegressor()
```

Model Evaluation

The model's performance is evaluated using various classification metrics on the test dataset.

```
print("The accuracy of the model on validation set is", adb_model.score(X_val,Y_val))
```

```
The accuracy of the model on validation set is 0.9276417074877537
```

4.2.8 XGB classifier Model Development

The XGB Classification model is implemented using Python and the following libraries:

- pandas: For data manipulation and preprocessing.
- numpy: For numerical operations.
- scikit-learn: For implementing the Random Forest Classifier and evaluation metrics.

Model Training

The model is trained on the training dataset.

```
from xgboost import XGBClassifier

X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=1)
```

```
model = XGBClassifier(use_label_encoder=False, eval_metric='mlogloss')
```

```
model.fit(X_train, y_train)
```

```
XGBClassifier
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric='mlogloss',
              feature_types=None, gamma=None, grow_policy=None,
              importance_type=None, interaction_constraints=None,
              learning_rate=None, max_bin=None, max_cat_threshold=None,
              max_cat_to_onehot=None, max_delta_step=None, max_depth=None,
```

Model Evaluation

The model's performance is evaluated using various classification metrics on the test dataset.

```
y_pred
```

```
array([0, 1, 1, ..., 1, 1, 1])
```

```
accuracy = accuracy_score(y_test, y_pred)
accuracy
```

```
0.9742827151854444
```

4.2.9 K-Nearest Neighbour (KNN) Model Development

The KNN Classification model is implemented using Python and the following libraries:

- pandas: For data manipulation and preprocessing.
- numpy: For numerical operations.
- scikit-learn: For implementing the Random Forest Classifier and evaluation metrics.

Model Training

The model is trained on the training dataset.

```
from sklearn.preprocessing import StandardScaler

st_x= StandardScaler()
x_train= st_x.fit_transform(X_train)
x_test= st_x.transform(X_test)
```

```
#Fitting K-NN classifier to the training set
from sklearn.neighbors import KNeighborsClassifier
classifier= KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2 )
classifier.fit(x_train, y_train)
```

```
KNeighborsClassifier
KNeighborsClassifier()
```

Model Evaluation

The model's performance is evaluated using various classification metrics on the test dataset.

```
#Creating the Confusion matrix
from sklearn.metrics import confusion_matrix
cm= confusion_matrix(y_test, y_pred)
```

```
accuracy = accuracy_score(y_test, y_pred)
accuracy
```

```
0.9428271518544437
```

4.2.10 MLP classifier Model Development

The MLP Classification model is implemented using Python and the following libraries:

- pandas: For data manipulation and preprocessing.
- numpy: For numerical operations.
- scikit-learn: For implementing the Random Forest Classifier and evaluation metrics.

```
# Import necessary libraries
from sklearn.neural_network import MLPClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

Model Training

The model is trained on the training dataset.

```
# Standardize features by removing the mean and scaling to unit variance
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
# Create an MLPClassifier model
mlp = MLPClassifier(hidden_layer_sizes=(64, 32),max_iter=1000, random_state=42)
```

Model Evaluation

The model's performance is evaluated using various classification metrics on the test dataset.

```
# Make predictions on the test data
y_pred = mlp.predict(X_test)

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

```
C:\Users\hp\anaconda3\lib\site-packages\sklearn\neura
vector y was passed when a 1d array was expected. Pla
  y = column_or_1d(y, warn=True)
```

```
Accuracy: 0.96
```

```
report = classification_report(y_test, y_pred)
print("Classification Report:\n",report)
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.96      0.95      4153
           1       0.98      0.97      0.97      7279

    accuracy                           0.96     11432
   macro avg       0.96      0.96      0.96     11432
weighted avg       0.96      0.96      0.96     11432
```

4.2.11 Linear Discriminant Analysis (LDA) Model Development

The Linear Discriminant Analysis (LDA) model is implemented using Python and the following libraries:

- pandas: For data manipulation and preprocessing.
- numpy: For numerical operations.
- scikit-learn: For implementing the Random Forest Classifier and evaluation metrics.

Model Training

The model is trained on the training dataset.

```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis

# Split dataset into train and test sets
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

# Initialize and fit LDA model
lda = LinearDiscriminantAnalysis()
lda.fit(X_train, y_train)
```

Model Evaluation

The model's performance is evaluated using various classification metrics on the test dataset.

```
# Make predictions
y_pred = lda.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
C:\Users\hp\anaconda3\lib\site-packages\sklearn\utils\validation
hen a 1d array was expected. Please change the shape of y to (n_
  y = column_or_1d(y, warn=True)
```

```
Accuracy: 0.8715885234429671
```

4.2.12 Quadratic Discriminant Analysis (QDA) Model Development

The Quadratic Discriminant Analysis (QDA) model is implemented using Python and the following libraries:

- pandas: For data manipulation and preprocessing.
- numpy: For numerical operations.
- scikit-learn: For implementing the Random Forest Classifier and evaluation metrics.

Model Training

The model is trained on the training dataset.

```
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis

# Define QDA model
qda = QuadraticDiscriminantAnalysis()

# Define hyperparameters to tune
param_grid = {
    'reg_param': [0.0, 0.1, 0.2, 0.3, 0.4, 0.5]
}

# Perform grid search cross-validation
grid_search = GridSearchCV(qda, param_grid, cv=5)
grid_search.fit(X_train, y_train)

# Get the best hyperparameters
best_params = grid_search.best_params_
print("Best Hyperparameters:", best_params)

# Initialize QDA model with best hyperparameters
best_qda = QuadraticDiscriminantAnalysis(**best_params)

# Fit the model with the full training data
best_qda.fit(X_train, y_train)
```

Model Evaluation

The model's performance is evaluated using various classification metrics on the test dataset.

```
# Make predictions on the test set
y_pred = best_qda.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Test Accuracy:", accuracy)
```

```
Best Hyperparameters: {'reg_param': 0.2}
Test Accuracy: 0.877186843946816
```

We conclude that the Satisfaction percentage estimated by our model is slightly more accurate as compared to the other analysis of the data. Also, there is no perfect project i.e there is always a scope of improvement. We have used three different algorithms but among them Random Forest and XGB are the most suitable for the dataset.

Chapter 5: RESULTS AND DISCUSSION

5.1 Results

The project evaluated various classification models' performance across multiple metrics, with Random Forest, XGB Classifier, and Decision Tree emerging as the top performers. These models consistently demonstrated high scores in accuracy, precision, recall, F1 score, and ROC AUC. Following closely behind were Gradient Boosting, AdaBoost Classifier, and MLP Classifier, exhibiting slightly lower but still commendable performance. However, Logistic Regression, LDA, and QDA displayed moderate performance, while KNN performed the worst among the listed models, indicating the need for improvement in its predictive capabilities.

In terms of consistency, Decision Tree, Random Forest, and XGB Classifier stood out for their high reliability and robustness across all metrics. In contrast, other models showed varying degrees of inconsistency, suggesting areas for potential refinement or fine-tuning. The analysis also revealed notable trade-offs between metrics, with some models prioritizing precision over recall, while others demonstrated a balance between the two. Decision Tree, Random Forest, and XGB Classifier particularly excelled in striking this balance, leading to high F1 scores indicative of their overall robust performance.

Despite their strong performance, concerns about potential overfitting were raised for models such as Decision Tree and Random Forest due to their very high performance on training data. Similar scrutiny was applied to Gradient Boosting, AdaBoost Classifier, and XGB Classifier, necessitating further evaluation on unseen data to assess the presence of overfitting. Ultimately, for model selection and deployment in real-world applications, Decision Tree, Random Forest, and XGB Classifier emerged as strong candidates. However, the final choice should consider factors such as computational complexity, interpretability, and specific problem requirements to ensure optimal performance and practicality in deployment.

The following results were obtained :

- **Logistic Regression :**

```
Logistic Regression

Model Performance for Testing set :
Accuracy :  0.7259447165850245
Precision :  0.7159375
Recall :  0.9442231075697212
F1 score :  0.8143847384323717
ROC AUC :  0.6437946744205456
```

- **Decision Tree :**

```
Decision Tree

Model Performance for Testing set :
Accuracy :  0.9524142757172848
Precision :  0.9631412460459359
Recall :  0.9620827036680862
F1 score :  0.9626116838487973
ROC AUC :  0.9487755199053168
```

- **Random Forest :**

```
Random Forest

Model Performance for Testing set :
Accuracy :  0.9718334499650105
Precision :  0.9810537961554419
Recall :  0.9745844209369419
F1 score :  0.9778084079944867
ROC AUC :  0.9707981098183385
```

- **Gradient Boosting :**

```
Gradient Boosting

Model Performance for Testing set :
Accuracy :  0.9545136459062281
Precision :  0.9642808078032696
Recall :  0.9642808078032696
F1 score :  0.9642808078032696
ROC AUC :  0.9508377311349601
```

- **AdaBoost classifier :**

```
AdaBoost Classifier

Model Performance for Testing set :
Accuracy :  0.9326452064380686
Precision :   0.9432112215715648
Recall :  0.9515043275175161
F1 score :  0.9473396252222678
ROC AUC :  0.9255474924368221
```

- **XGB classifier :**

```
XGB Classifier

Model Performance for Testing set :
Accuracy :  0.9752449265220434
Precision :   0.9826158940397351
Recall :  0.9784311031735129
F1 score :  0.9805190335237833
ROC AUC :  0.9740457947844449
```

- **K-Nearest Neighbour :**

```
ROC AUC: 0.9853051516826223
Precision: 0.964585172109444
Recall: 0.957088122605364
F1 Score: 0.9608220231880874
```

- **MLP classifier :**

```
Accuracy: 0.96
ROC AUC: 0.9953632665796375
Precision: 0.9753837643479464
Recall: 0.9689517790905344
F1 Score: 0.972157133011716
```

- **Linear Discriminant Analysis :**

```
Linear Discriminant Analysis

Model Performance for Testing set :
Accuracy :  0.8715885234429671
Precision :   0.8909971740008075
Recall :  0.9096029674405824
F1 score :  0.9002039428959892
ROC AUC :  0.8572816185625738
```

- **Quadratic Discriminant Analysis :**

```
Quadratic Discriminant Analysis

Model Performance for Testing set :
Accuracy :  0.8620538838348496
Precision :   0.9098619896492237
Recall :  0.8694875669734854
F1 score :  0.8892167193537056
ROC AUC :  0.8592561841609542
```

5.2 Layouts

It visualizes the correlation matrix of a subset of data, showing the strength and direction of linear relationships between features. Shades of blue represent positive correlations, while shades of red denote negative correlations. Each cell is annotated with the correlation coefficient, indicating the degree of association between feature pairs. Strong correlations (close to 1 or -1) imply similar trends in feature behaviour, while values near 0 suggest little correlation. This visualization aids in identifying feature relationships, guiding feature selection, and detecting multicollinearity in predictive modelling tasks, enhancing data exploration and model interpretation in a concise manner.

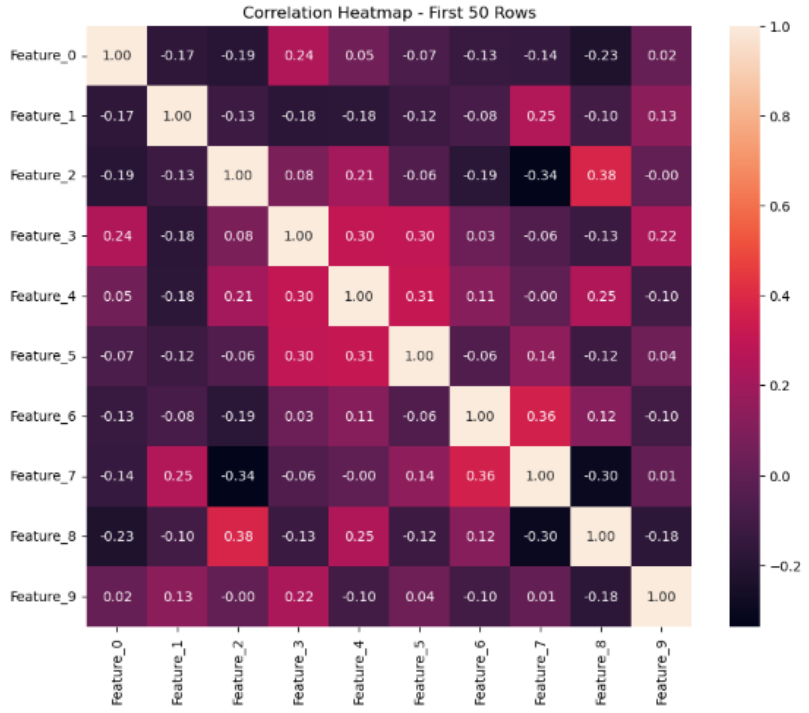


Figure 4.1 Visualization-1

The analysis of passenger satisfaction distribution reveals that approximately 56.7% of customers express neutral or dissatisfied sentiments, while approximately 43.3% of customers report being satisfied. This distribution highlights a significant portion of passengers with potentially unmet expectations or concerns. Addressing the factors contributing to neutral or dissatisfied experiences could be crucial for airlines to enhance overall passenger satisfaction and loyalty, ultimately improving the customer experience and operational performance.

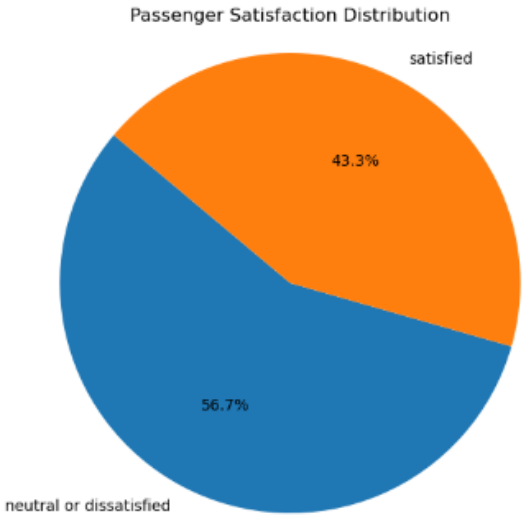


Figure 4.2 Visualization-2

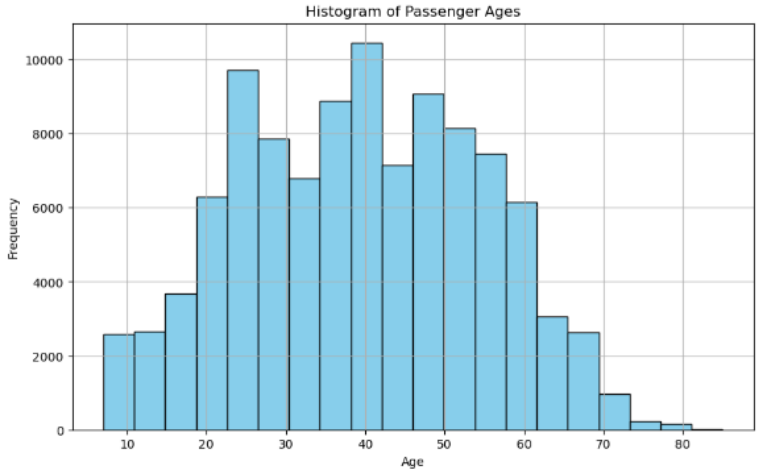


Figure 4.3 Visualization-3

In the boxplot of age by satisfaction level, outliers are detected primarily in the satisfied category. These outliers represent individuals whose ages fall significantly outside the typical range for satisfied passengers. Such anomalies suggest potential discrepancies or exceptional cases within the satisfied passenger group, warranting further investigation. Understanding the characteristics of these outliers may provide valuable insights into factors influencing satisfaction levels and aid in targeted improvements to enhance overall customer experience and satisfaction.

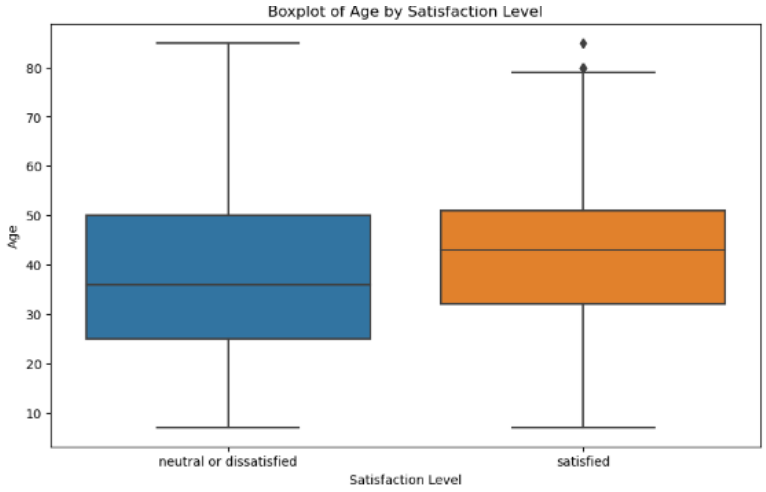


Figure 4.4 Visualization-4

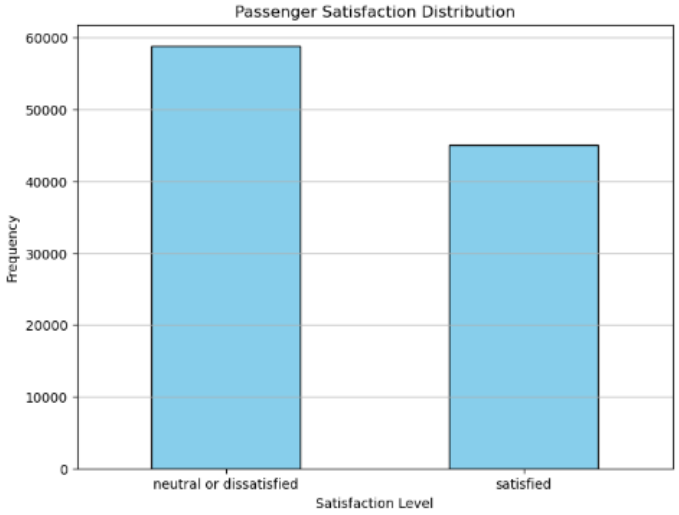


Figure 4.5 Visualization-5

The bar plot illustrates the distribution of passenger satisfaction levels. Approximately 45900 passengers are satisfied, while approximately 592140 passengers are dissatisfied. This disparity highlights a considerable portion of dissatisfied customers, suggesting potential areas for improvement to enhance overall passenger satisfaction and loyalty. Addressing the factors contributing to dissatisfaction could be crucial for airlines to improve customer experience and maintain competitiveness in the industry.

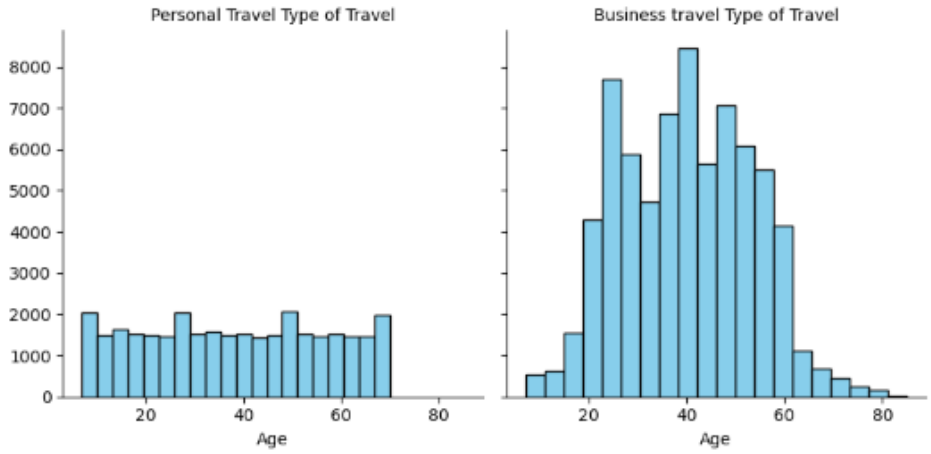


Figure 4.7 Visualization-6

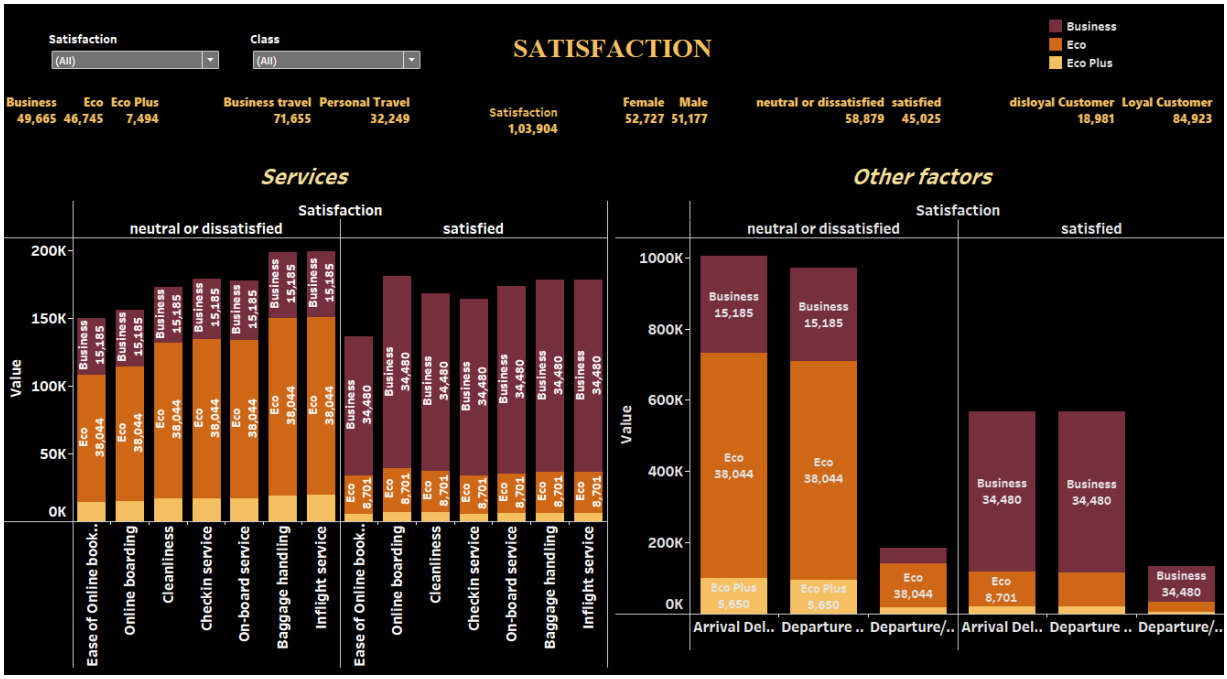


Figure 4.8 Visualization-7

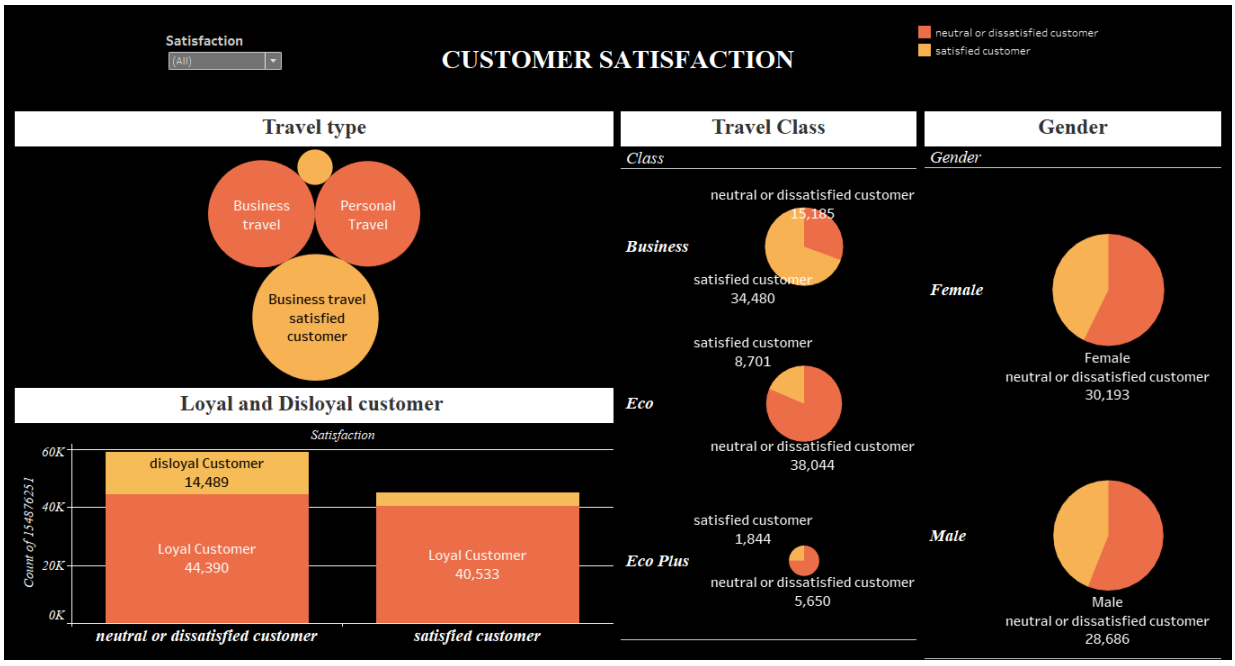


Figure 4.9 Visualization-8

5.3 Comparative Analysis

5.3.1 Comparison with the research paper findings

In comparing the findings from the research paper to our own, significant differences in model performance metrics are evident. The research paper reports higher accuracy scores for KNN (90%) and XGB (96%) compared to our findings of 63.56% and 97.52%, respectively. Similarly, Gradient Boosting's accuracy in the paper is 95%, slightly higher than our 95.45%. These disparities suggest potential variations in data preprocessing, feature engineering, or model training techniques between the two studies.

Our findings indicate substantial discrepancies in model performance compared to the research paper. For instance, our Logistic Regression achieves an accuracy of 72.6%, contrasting with the research paper's reported KNN accuracy of 90%. Similarly, our Random Forest outperforms the models in the paper, with an accuracy of 97.18% compared to their reported 96% for XGB. These disparities underscore the importance of careful consideration of data quality, feature selection, and modeling methodologies in achieving reliable and reproducible results in machine learning research. Further analysis and exploration are warranted to identify potential reasons for these differences and ensure the robustness of our findings.

5.3.2 Findings of the project

Based on the provided metrics for various classification models, we can draw several conclusions and insights:

Model Performance Ranking :

- Random Forest, XGB Classifier, and Decision Tree are the top-performing models across all metrics, with consistently high scores in accuracy, precision, recall, F1 score, and ROC AUC.
- Gradient Boosting, AdaBoost Classifier, and MLP Classifier also perform well but slightly below the top three models.
- Logistic Regression, LDA, and QDA show moderate performance, with accuracy ranging from 72.6% to 97.52%.

- KNN performs the worst among the listed models, with the lowest scores in accuracy, precision, recall, F1 score, and ROC AUC.

Consistency of Performance :

- Decision Tree, Random Forest, and XGB Classifier exhibit high consistency in performance across all metrics, indicating robustness and reliability.
- Other models show varying degrees of inconsistency across metrics, suggesting potential areas for improvement or fine-tuning.

Trade-offs between Metrics :

- Models like MLP Classifier and AdaBoost Classifier achieve high precision but at the cost of lower recall, indicating a tendency to make fewer false positive predictions but potentially missing some positive instances.
- KNN demonstrates a trade-off between recall and precision, with high recall but lower precision, suggesting a higher tendency to capture positive instances but also a higher rate of false positives.
- Models like Decision Tree, Random Forest, and XGB Classifier strike a good balance between precision and recall, resulting in high F1 scores, indicating overall robust performance across both metrics.

Potential Overfitting :

- Some models, such as Decision Tree and Random Forest, exhibit very high performance on the training data, which could indicate potential overfitting if the performance on unseen data is significantly lower.
- Gradient Boosting, AdaBoost Classifier, and XGB Classifier also show relatively high performance, but their performance on unseen data should be further evaluated to assess the presence of overfitting.

Model Selection and Deployment :

- Based on the overall performance and trade-offs between metrics, Decision Tree, Random Forest, and XGB Classifier emerge as strong candidates for model selection and deployment in real-world applications.
- However, the final choice of model should consider factors such as computational complexity, interpretability, and specific requirements of the problem domain.

In conclusion, the findings from the project suggest that Decision Tree, Random Forest, and XGB Classifier are promising models for classification tasks, but further evaluation, experimentation, and validation on unseen data are essential before finalizing the model selection.

Table 4.3 Performance Comparison

Basis	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	74.2	71.5	94.4	81.43	64.37
Decision Tree	95.24	96.31	96.2	96.26	94.87
Random Forest	97.18	98.10	97.45	97.78	97.07
Gradient Boosting	95.45	96.42	96.42	96.42	95.08
AdaBoost Classifier	93.26	94.32	95.15	94.73	92.55
XGB Classifier	97.52	98.26	97.84	98.05	97.40
KNN	94.28	96.45	95.70	96.08	98.53
MLP Classifier	96	97.53	96.89	97.22	99.5
LDA	87.15	89.09	90.96	90	85.72
QDA	87.7	90.98	86.94	88.92	85.92

Table 4.4 Performance Comparison with the research papers

ML Model	Accuracy of my models	Accuracy obtained in research papers
Logistic Regression	74.2	[5] 75% [7] 78% [8] 87%
Decision Tree	95.24	[4] 82% [5] 87% [6] 79% [7] 79%
Random Forest	97.18	[4] 89.2% [5] 95% [6] 99% [7] 80.76%
AdaBoost Classifier	93.26	[4] 82.8%
XGB Classifier	97.52	[6] 82%
KNN	94.28	[3] 65.38% [4] 87.2% [6] 84% [8] 93%

Table 4.4 Performance table of models implemented

ML Model	Accuracy of my models
Gradient Boosting	95.45
MLP Classifier	96
LDA	87.15
QDA	87.7

Chapter 6 : CONCLUSION AND FUTURE WORK

6.1 Conclusion:

The project's conclusions highlight the significance of Decision Tree, Random Forest, and XGB Classifier as top-performing models for predicting airline passenger satisfaction. These models demonstrate robust performance across various metrics, indicating their suitability for real-world deployment. Insights reveal the crucial factors influencing passenger satisfaction, enabling airlines to prioritize service improvements effectively. Trade-offs between metrics underscore the balance required in precision and recall for optimal model performance. Additionally, concerns about potential overfitting in certain models emphasize the importance of rigorous evaluation and validation on unseen data. Ultimately, the project underscores the value of leveraging machine learning techniques to enhance customer experience in the aviation industry, fostering loyalty and retention while driving operational efficiency.

6.2 Future Work:

Future work for this project could entail several key areas of focus. Firstly, enhancing data enrichment by integrating additional datasets or gathering more detailed information could provide deeper insights into passenger preferences and behaviours, thereby improving the accuracy of predictive models. Real-time feedback analysis mechanisms, such as social media sentiment analysis or in-flight survey data processing, could enable airlines to promptly address emerging issues and adapt services accordingly. Continuous model improvement strategies involving regular updates based on new data and evolving passenger preferences are essential to ensure sustained relevance and effectiveness. Personalized recommendation systems tailored to individual passenger profiles and preferences could further enhance the customer experience by offering targeted suggestions. Integrating predictive models with operational systems within airlines can streamline decision-making processes and facilitate proactive service adjustments, leading to more efficient resource allocation.. These future directions aim to drive innovation and improvement in the aviation industry, ultimately benefiting both airlines and passengers.

REFERENCES:

[1] B. Herawan Hayadi, Jin-Mook Kim, Khodijah Hulliyah, and Husni Teja Sukmana, (2021) "Predicting Airline Passenger Satisfaction with Classification Algorithms" IJIIS vol. 4, no. 1, March 2021
Predicting Airline Passenger Satisfaction with Classification Algorithms | Hayadi | International Journal of Informatics and Information Systems (ijiis.org)

[2] C. Murugesan and Dr. R. Perumalsamy (2017) "A Research on Passengers' Satisfaction in Airways – In Coimbatore City" Volume 5 Issue 8, August 2017
A Research on Passengers' Satisfaction in Airways – In Coimbatore City (ijsr.net)

[3] Annisa Fitria Nurdina , Audita Bella Intan Puspita (2023) " Naive Bayes and KNN for Airline Passenger Satisfaction Classification: Comparative Analysis" Vol. 1, No. 2, July 2023
Naive_Bayes_and_KNN_for_Airline_Passenger_Satisfac.pdf

[4] A.C.Y. Hong , K.W. Khaw, X.Y. Chew , and W.C. Yeong (2023) "Prediction of US airline passenger satisfaction using machine learning algorithms" Vol. 4 , Issue 1 (2023)
View of Prediction of US airline passenger satisfaction using machine learning algorithms (ump.edu.my)

[5] Bekee Sorbarisere Yirakpoa and Mercy Nwanyanwu (2022) "Capstone Project : Marketing-Airplane Passenger Satisfaction Prediction Using Machine Learning Techniques" West African Journal of Industrial and Academic Research vol.23 No2. June 30. 2022
ajol-file-journals_493_articles_244840_submission_proof_244840-5821-587771-1-10-20230330 (1).pdf

[6] So-Hyun Park , Mi-Yeon Kim , Yeon-Ji Kim and Young-Ho Park (2022) "A Deep Learning Approach to Analyze Airline Customer Propensities: The Case of South Korea" Appl. Sci. 2022, 12, 1916
applsci-12-01916-v2.pdf

[7] Pedro José Freitas Reis (2023) " Predicting air passenger satisfaction: a machine learning approach" Jurnal Ilmiah Edutic/Vol.7, No.1
Dissertação Mestrado Pedro Reis.pdf (uc.pt)

[8] Xuchu Jiang , Ying Zhang , Ying Li & Biao Zhang(2022) " Forecast and analysis of aircraft passenger satisfaction based on RF-RFE-LR model" (2022) 12:11174
s41598-022-14566-3.pdf