Vietnam National University Ho Chi Minh City

University of Information Technology

Faculty of Information Systems

A REPORT ON

"Predicting Airline Passenger Satisfaction"

SUBMITTED BY

TRAN THANH PHONG – 20521750

NGUYEN XUAN TUAN KIET – 20521502

NGO QUOC HUY – 21522148

MAI TRAN PHUONG NHI – 21522428

UNDER THE GUIDANCE OF

CAO THI NHAN

NGUYEN THI VIET HUONG

SUBJECT

DATA MINING

(Academic year: 2023 – 2024)

Ho Chi Minh City, 12/2023

Instructor's Comments

....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....

……., ……/…………/ 2023

Signature

List of Figures

List of Tables

Abstract

In the rapidly evolving era of technological advancement, the aviation industry has emerged as a pivotal player. With millions of passengers traveling daily, aviation has become one of the largest and most competitive sectors. Ensuring high passenger satisfaction is a key factor for the success of airlines. Satisfied passengers are likely to become loyal customers. However, understanding the factors influencing passenger satisfaction can be challenging. The research team applied classification algorithms to predict customer satisfaction based on survey data. The employed algorithms include K-Nearest Neighbors, Support Vector Machines, Decision Tree, and Naïve Bayes. The results yielded promising predictions. If applied to airlines, these algorithms could significantly enhance the ability to forecast passenger satisfaction, thereby improving the overall service quality of airlines.

Keywords: Predicting Airline Passenger Satisfaction, Machine Learning, Machine Learning In Aviation, Data Mining In Airline Industry.

Chapter 1

1. Introduction

1.1. Introduction

In the dynamic landscape of the aviation industry, ensuring high levels of passenger satisfaction is paramount for airlines to thrive in an intensely competitive environment. However, accurately predicting and comprehending the factors influencing passenger contentment poses significant challenges for the industry. Leveraging advancements in data analysis techniques, this study focuses on the application of K-Nearest Neighbors (KNN), Decision Tree, Naïve Bayes, and Support Vector Machines (SVM) algorithms to forecast and enhance airline passenger satisfaction.

The complexities inherent in predicting passenger contentment make this endeavor crucial for airlines seeking to optimize their service quality. By adopting data mining, a research direction grounded in data sets and processing methods, coupled with machine learning algorithms, this study aims to provide a predictive framework. The proposed decision support system, incorporating sophisticated algorithms, strives to empower airlines to make informed decisions that lead to heightened passenger satisfaction.

As the aviation industry continues to evolve, the utilization of these advanced predictive algorithms becomes instrumental in improving customer experience. The application of KNN, Decision Tree, Naïve Bayes, and SVM algorithms is poised to offer valuable insights that facilitate strategic decision-making, ultimately contributing to the overarching goal of maximizing passenger contentment and refining the overall quality of airline services.

1.2. Related Documents

In addition, in the last three years, there have been some typical articles on predictive research related to this topic, such as:

Forecast and analysis of aircraft passenger satisfaction based on RF-RFE-LR model The study utilizes the RF-RFE-LR model to forecast and analyze aircraft passenger satisfaction [1]. It employs the RF-RFE algorithm to extract a feature subset of 17 variables, with RF demonstrating optimal performance (accuracy: 0.963, precision: 0.973, recall: 0.942, F1 value: 0.957, AUC value: 0.961). A logistic model is then trained on the RF-RFE-selected features to identify key variables affecting passenger satisfaction. The analysis includes comparisons among different passenger and class types, offering recommendations for online boarding and onboard Wi-Fi services. Limitations are acknowledged, such as insufficient evaluation indicators and the use of default parameters. Proposed improvements encompass expanding ground service evaluation, optimizing model parameters, and considering additional variables influencing passenger satisfaction.

Naive Bayes and KNN for Airline Passenger Satisfaction Classification: Comparative Analysis [2]. The study compares Naive Bayes and K-NN algorithms for classifying airline customer satisfaction using RapidMiner Studio version 10.1. Data is sourced from Kaggle.com's Airline Passenger Satisfaction dataset. Results indicate Naive Bayes outperforms K-NN, with an accuracy of 84.48% for Naive Bayes and 65.38% for K-NN. Precision values are 82.25% (Naive Bayes) and 67.35% (K-NN), while recall values are 82.43% (Naive Bayes) and 74.33% (K-NN). Caution is advised due to the diverse attributes influencing passenger satisfaction. Future assessments could employ various techniques for pre-flight, in-flight, and post-flight services to enhance the accuracy of evaluating airline services and passenger satisfaction.

Machine Learning Approach Using MLP and SVM Algorithms for the Fault Prediction of a Centrifugal Pump in the Oil and Gas Industry [3]. This paper proposes a machine learning-based algorithm for fault diagnosis in rotating machinery within the oil and gas industry, emphasizing simplicity and practical implementation. Using data from a real centrifugal pump at the SARLUX refinery, eight sensors measure various parameters. The Support Vector Machine (SVM) and Multilayer Perceptron (MLP) algorithms are compared, with both showing effective classification performance. While SVM demonstrates higher precision, MLP excels in predicting failures.

Classification of Airline Customer Sentiment Expressed in Twitter Tweets using Lexicons, Decision Tree, and Naïve Bayes [4]. This research aims to analyze consumer perceptions of airlines expressed in Tweets on Twitter through sentiment analysis. Utilizing supervised machine learning and lexical-based methods, the study demonstrates that valuable insights can be derived from freely available social media data. The results serve as proof of concept, showcasing the feasibility of monitoring customer sentiments and identifying service criteria that elicit positive or negative responses.

Weighted p-norm distance t kernel SVM classification algorithm based on improved polarization [5]. This study introduces a novel p-norm distance t kernel for the classical SVM algorithm, enhancing classification performance. The kernel, based on the t probability density function, is combined with other kernel functions using an optimized model for weight coefficients and parameters. Experimental results on six datasets demonstrate improved SVM classification compared to traditional single kernel functions. The study also analyzes the influence of p-norm distance on SVM performance, revealing dataset-dependent effects. While promising, challenges remain in theoretical basis, kernel function selection, and optimization convergence. The proposed method is versatile, applicable to tasks like dimensionality reduction, kernel clustering, and medical drug screening, with ongoing efforts to refine and extend its application in future research directions.

Chapter 2

2. Overview

2.1. Data mining

Data mining is a step in doing Knowledge Discovery in Databases (KDD) [6]. Many benefits can be obtained through data mining processing, which helps get valuable information and increase understanding of various data that can be analyzed using multiple algorithms [7]. Data mining is finding patterns contained in datasets with certain methods. This process is essential in creating new discoveries or knowledge from a dataset.

One of the main roles of data mining is classification in classification utilizing data train to improve the model's quality and the analysis result [8]. Data mining places a strong emphasis on the precision of model predictions. A crucial indicator of effectiveness in data mining models is their capacity to make precise forecasts in practical scenarios. This focus on accuracy stems from the origins of data mining within the realm of Artificial Intelligence, which has always been concerned with developing applicable predictive models. These models have found utility in various real-world applications, including predicting insurance fraud, diagnosing illnesses, recognizing patterns, and more [9].

2.2. Classification

One of the goals that many generate in data mining is classification. Classification is a classification or grouping function that explains or distinguishes concepts or data classes to estimate the class of an object whose label is unknown or dividing something according to its classes. Classification is the process of finding a data class so that it can estimate the class of an object whose label is unknown [2].

2.3. K-Nearest Neighbors model

K-Nearest-Neighbors or KNN is a simple classification algorithm used for classification and prediction tasks. This algorithm stores all the input data with its corresponding labels and classifies a new observation based on similarity [10].

In KNN, each data point is represented as a point in a multi-dimensional space. To classify a new data point, the algorithm identifies the K nearest data points (neighbors) based on distance in the feature space. Then, the prediction for the new point is determined based on the majority or average of the labels of these neighboring points.

It's crucial to note that the value of K, i.e., the number of neighbors considered, can impact the performance of the algorithm. A larger K may make the model more robust but could increase computational complexity, while a smaller K might make the model more sensitive to noise in the data. K-NN regression uses the following distance measures for continuous variables. KNN uses a number of measures as follows:

Euclidean [11] :

$$d(x, y) = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2} \tag{2.3.1}$$

In (2.3.1):

d: Distance

k: Data Dimension

x: Test data

y: Train data

Manhattan [11]:

$$d(x, y) = \sum_{i=1}^{k} |x_i - y_i| \tag{2.3.2}$$

In (2.3.2):

d: Distance

k: Data Dimension

x: Test data

y: Train data

Minkowski [12]:

$$d_p(x, y) = (\sum_{i=1}^{k} |x_i - y_i|^q)^{\frac{1}{q}} \tag{2.3.3}$$

In (2.3.3):

d: Distance

k: Data Dimension

x: Test data

y: Train data

q: (q = 2 for Euclidean distance, q = 1 for Manhattan distance)

Chebyshev [12] :

$$d_\infty(x, y) = max_i(|x_i - y_i|) \tag{2.3.4}$$

In (2.3.4):

d: Distance

x: Test data

y: Train data

2.4. Support Vector Machines model

SVM is a powerful method for building a classifier. It aims to create a decision boundary between two classes that enables the prediction of labels from one or more feature vectors [13]. It is a supervised machine learning problem where we try to find a hyperplane that best separates the two classes. SVM does this by finding the maximum margin between the hyperplanes that means maximum distances between the two classes.

Key concepts associated with SVM [14] as below (2.4.1):

$$\min \left(\frac{1}{2}\left|\left|w\right|\right|^2 + C\sum_{1=1}^{n} \xi i + \zeta * i\right) \qquad (2.4.1)$$

In (2.4.1):

$$\begin{cases} yi - < w, xi > - b \le \varepsilon + \xi i \\ b + < w, xi > - yi \le \varepsilon + \zeta * \\ \xi i, \zeta * i \ge 0 \\ i = 1, \ldots, n \end{cases}$$

- **Hyperplane:** In an N-dimensional space, a hyperplane is an (N-1)-dimensional flat affine subspace. For a two-dimensional space (2D), the hyperplane is a line, and for a three-dimensional space (3D), it is a plane.
- **Margin:** The margin is the distance between the hyperplane and the nearest data point from either class. SVM aims to maximize this margin, which helps improve the model's generalization performance.
- **Support Vectors:** These are the data points that are closest to the hyperplane and have the most influence on the position and orientation of the hyperplane. These points support the optimal separation of classes.
- **Kernel Trick**: SVM can efficiently handle non-linear decision boundaries by using a kernel function. The kernel function transforms the input features into a higher-dimensional space, making it easier to find a hyperplane that separates the data. In this project, we use the default kernel which is ‘rbf’ (Radial basis function kernel)
- **C parameter:** This parameter controls the trade-off between achieving a smooth decision boundary and classifying training points correctly. A small C encourages a larger margin but may misclassify some points, while a large C classifies all training points correctly but may result in a smaller margin. In this project, we use the default C = 1.0

2.5. Decision Tree model

Decision tree, one of the most widely used algorithms in machine learning, is an explainable and white box algorithm that shows classification results using an if-then rule format [15]. It is used in both classification and regression problems. In a decision tree, every node symbolizes a feature, each branch indicates a rule, and every leaf represents a result, either a specific value or the continuation of further branching. There are several algorithms for constructing a decision tree. A tree is trained to make predictions for a new instance by traversing from the root node to the leaves, considering the attributes along the path. This report will focus on two of them: CART (Classification and Regression Trees) and ID3 (Iterative Dichotomiser 3).

ID3:

Entropy : A measure of uncertainty associated with a random variable as below

$$E(S) = -\sum_{j=1}^{n} f_s(A_j)\, log_2\, f_s(A_j) \qquad (2.5.1)$$

In (2.5.1):

S: sample set

N: number of different values of all samples in S

Aj: number of sample corresponding to each j

Fs(Aj): ratio of Aj to S

Information Gain:

Information Gain of set of sample S based on attribute A as below:

$$G(S,A) = E(S) - \sum_{i=1}^{m} f_s(A_i) E(S_{Ai}) \qquad (2.5.2)$$

In (2.5.2):

G(S,A): information gain of set S based on attribute A

E(S): entropy of S m: number of different values of attribute A

Ai: number of sample corresponding to each I of attribute A

Fs(Ai): ratio of Ai to S

S_{Ai}: subset of S including all samples having value Ai

Step by step:

Step 1: Calculate the entropy of the current dataset.

Step 2: For all features:

- For each value of the feature, calculate the entropy of the dataset when it is partitioned by that value.
- Calculate the average of entropies computed for each value of the feature.

Step 3: Compare the entropy before and after splitting the dataset with each value of the feature to calculate Information Gain. Choose the feature with the highest Information Gain.

Step 4: Continue the above process for the newly created child nodes using the selected feature. Repeat the process until the decision tree reaches a desired state (e.g., maximum number of leaves, maximum depth).

CART: Using Gini index

$$Gini(D) = 1 - \sum_{i=1}^{k} p(j|D)^2 \qquad (2.5.3)$$

In (2.5.3): p(j|D) the relative frequency of class j in D

If a data set D is split on an into k subsets D_1 , D_2 ,…, D_k the Gini index $Gini_A$ (D) is defined as:

$$Gini_A(D) = \sum_{i=1}^{k} \frac{{n_i}^2}{n} Gini(i) \qquad (2.5.4)$$

In (2.5.4):

ni: #samples of node i

N: #samples of no A

Step by step:

Step 1: Calculate the Gini index of the current dataset.

Step 2: For all features:

- For each value of the feature, calculate the Gini index of the dataset when it is split based on that value.
- Calculate the average Gini index considering each value of the feature.

Step 3: Compare the Gini index before and after splitting the dataset with each value of the feature. Choose the feature that results in the lowest Gini index, indicating the highest Gini Gain.

Step 4: Continue the process for the newly created child nodes using the selected feature. Repeat the process until the decision tree reaches a desired state (e.g., maximum number of leaves, maximum depth).

2.6. Naive Bayes model

Naive Bayes is a popular and straightforward machine learning algorithm used for classification and prediction tasks. It is based on Bayes' theorem and makes a "naive" assumption that all input features are independent of each other [16]. Despite this strong assumption, often not holding true in reality, it simplifies the model's complexity and enhances computational efficiency.

The algorithm is primarily employed for classification tasks where the goal is to assign a label to a data sample based on its features. Naive Bayes leverages Bayes' theorem, expressing how we can update the probability of a hypothesis given new data. The "naive" assumption significantly simplifies computations, especially when dealing with large datasets.

Despite its simplicity, Naive Bayes often delivers competitive performance and is particularly effective when the naive assumption holds or the data is preprocessed to approximate it.

Bayes’ Theorem basics [16]:

$$P(H|X) = \frac{P(X|H)\, P(H)}{P(X)} \qquad (2.6.1)$$

In (2.6.1):

- Let X be a data sample (“evidence”) : class label is unknown.
- Let H be a hypothesis that X belongs to class C.
- Classification is to determine P(H|X) : the probability that the hypothesis holds given the observed data sample X.
- P(H) (prior probability) : the initial probability.
- P(X) (prior probability) : probability that sample data is observed.
- P(X|H) (likelihood) : the probability of observing the sample X, given that the hypothesis holds.

Chapter 3

3. Method

3.1. Method overview

The input for this data mining project consists of surveyed values related to flight experiences, with the target variable being categorized as either "satisfied" or "neutral or dissatisfied." Following data preprocessing, where essential columns are selected, data is encoded and transformed, the dataset is prepared for training machine learning models. The selected algorithms for this task include K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes, and Decision Tree. The target variable is binary, represented as 1 for "satisfied" and 0 for "neutral or dissatisfied." After fitting the data into the training process for each model, the resulting models are then exported for application in a simple web interface. The following diagram (Fig 3.1.1) will show the steps of the project to predict airline passenger satisfaction.

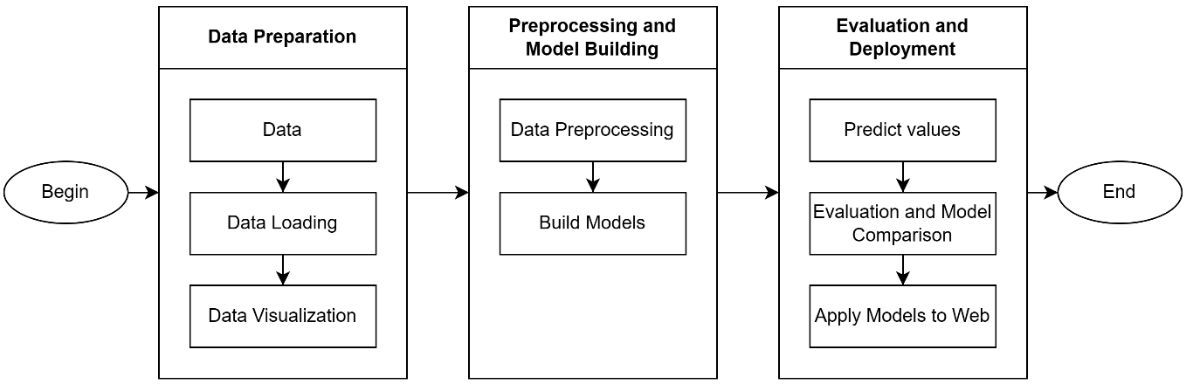

Figure 3.1.1: Implementation diagram using classifier models to predict airline passenger satisfaction

This approach allows for predicting passenger satisfaction on a website based on the trained machine learning models. To carry out a data mining project, the team has divided the process into several steps as follows:

Step 1. Data Preparation: Involves gathering and organizing the dataset.

Step 2. Data Loading: Importing the dataset into the chosen environment for analysis.

Step3. Data Visualization: Creating visual representations of the data for better understanding.

Step 4. Data Preprocessing: This step encompasses various tasks such as handling missing values, removing unnecessary columns and attributes, eliminating duplicate values, addressing outliers, encoding data, normalizing data, and splitting the dataset.

Step5. Building Models and Training: Developing and training machine learning models based on the preprocessed data.

Step 6. Prediction: Applying the trained models to make predictions on new data.

Step 7. Evaluation and Model Comparison: Assessing the performance of the models and comparing their effectiveness.

Strep 8. Application to Reality: Implementing the models in a practical setting by creating a simple website designed to predict passenger satisfaction.

This structured approach ensures a systematic progression from data preparation to real-world application, with a focus on optimizing the performance of the developed models.

3.2. Data overview

This dataset contains a survey of passenger satisfaction on flights. These survey factors are strongly correlated with passenger satisfaction (or dissatisfaction). The data set named "Airline Passenger Satisfaction" includes 2 files "test.csv" and "train.csv" including 25 attributes for each file. The properties of the dataset used in the report are illustrated in Table 3.2.1 below.

Column	Explain
#	Numerical order
id	Flight ID code
Gender	Customer's gender (Male, Female)
Customer Type	Customer type (Loyal customer, disloyal customer)
Age	Customer's age
Type of Travel	Purpose of the customer's flight (Personal Travel, Business Travel)
Class	Customer's ticket class (Business, Eco, Eco Plus)
Flight distance	Distance of the flight journey
Inflight wifi service	Satisfaction level with in-flight wifi service (0:Not Applicable;1-5)
Departure/Arrival time convenient	Level of satisfaction with convenient departure/arrival time

Ease of Online booking	Level of satisfaction when booking tickets online
Gate location	Level of satisfaction with Gate location
Food and drink	Level of satisfaction with food and drinks
Online boarding	Satisfaction level with online check-in
Seat comfort	Level of satisfaction with seat comfort
Inflight entertainment	Level of satisfaction with in-flight entertainment
On-board service	Level of satisfaction with on-board service
Leg room service	Level of satisfaction with seats with wide legroom
Baggage handling	Level of satisfaction with baggage handling
Check-in service	Level of satisfaction with Check-in service
Inflight service	Level of satisfaction with in-flight service
Cleanliness	Level of satisfaction with cleanliness
Departure Delay in Minutes	Departure minutes
Arrival Delay in Minutes	Number of minutes delayed upon arrival
Satisfaction	Customer satisfaction level with the airline (Satisfaction, neutral or dissatisfaction)

Table 3.2.1: Dataset overview

The data set is divided into 2 files with the file "train.csv" used to train the data and file "test.csv" is used to predict results for models. The data set is collected on the Kaggle site, original name is "Airline Passenger Satisfaction[1]".

3.3. Data loading

There are 2 parts of the dataset, proceed to add data. One part for training data and the other for testing. These 2 datasets are presented as a train set named 'train.csv' and a test set named 'test.csv'

3.4. Data visualization

The train data set includes 25 columns and 103904 rows of data, the test data set includes 25 columns and 25976 rows of data displayed as follows (Fig 3.4.1).

[1] https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction/code

```
RangeIndex: 103904 entries, 0 to 103903
Data columns (total 25 columns):
 #   Column                             Non-Null Count   Dtype
---  ------                             --------------   -----
 0   Unnamed: 0                         103904 non-null  int64
 1   id                                 103904 non-null  int64
 2   Gender                             103904 non-null  object
 3   Customer Type                      103904 non-null  object
 4   Age                                103904 non-null  int64
 5   Type of Travel                     103904 non-null  object
 6   Class                              103904 non-null  object
 7   Flight Distance                    103904 non-null  int64
 8   Inflight wifi service              103904 non-null  int64
 9   Departure/Arrival time convenient  103904 non-null  int64
 10  Ease of Online booking             103904 non-null  int64
 11  Gate location                      103904 non-null  int64
 12  Food and drink                     103904 non-null  int64
 13  Online boarding                    103904 non-null  int64
 14  Seat comfort                       103904 non-null  int64
 15  Inflight entertainment             103904 non-null  int64
 16  On-board service                   103904 non-null  int64
 17  Leg room service                   103904 non-null  int64
 18  Baggage handling                   103904 non-null  int64
 19  Checkin service                    103904 non-null  int64
 20  Inflight service                   103904 non-null  int64
 21  Cleanliness                        103904 non-null  int64
 22  Departure Delay in Minutes         103904 non-null  int64
 23  Arrival Delay in Minutes           103594 non-null  float64
 24  satisfaction                       103904 non-null  object
dtypes: float64(1), int64(19), object(5)
memory usage: 19.8+ MB
```

```
RangeIndex: 25976 entries, 0 to 25975
Data columns (total 25 columns):
 #   Column                             Non-Null Count  Dtype
---  ------                             --------------  -----
 0   Unnamed: 0                         25976 non-null  int64
 1   id                                 25976 non-null  int64
 2   Gender                             25976 non-null  object
 3   Customer Type                      25976 non-null  object
 4   Age                                25976 non-null  int64
 5   Type of Travel                     25976 non-null  object
 6   Class                              25976 non-null  object
 7   Flight Distance                    25976 non-null  int64
 8   Inflight wifi service              25976 non-null  int64
 9   Departure/Arrival time convenient  25976 non-null  int64
 10  Ease of Online booking             25976 non-null  int64
 11  Gate location                      25976 non-null  int64
 12  Food and drink                     25976 non-null  int64
 13  Online boarding                    25976 non-null  int64
 14  Seat comfort                       25976 non-null  int64
 15  Inflight entertainment             25976 non-null  int64
 16  On-board service                   25976 non-null  int64
 17  Leg room service                   25976 non-null  int64
 18  Baggage handling                   25976 non-null  int64
 19  Checkin service                    25976 non-null  int64
 20  Inflight service                   25976 non-null  int64
 21  Cleanliness                        25976 non-null  int64
 22  Departure Delay in Minutes         25976 non-null  int64
 23  Arrival Delay in Minutes           25893 non-null  float64
 24  satisfaction                       25976 non-null  object
dtypes: float64(1), int64(19), object(5)
memory usage: 5.0+ MB
```

Figure 3.4.1: The information of datasets

Comment:

About the training data (train):

- Number of rows and columns: 103,904 rows, 25 columns

Data Types:

- Object (Categorical): Including 5 columns >> ‘Gender’, ‘Customer Type’, ‘Type of Travel’, ‘Class’, ‘satisfaction’
- Float: Including 1 column >> ‘Arrival Delay in Minutes’
- 19 columns left are Integer
- Column ‘Arrival Delay in Minutes’ has 310 missing values

About the test data (test):

- Number of rows and columns: 25976 rows, 25 columns

Data Types:

- Object (Categorical): Including 5 columns >> ‘Gender’, ‘Customer Type’, ‘Type of Travel’, ‘Class, satisfaction’
- Float: Including 1 column >> ‘Arrival Delay in Minutes’
- 19 columns left are Integer
- Column ‘Arrival Delay in Minutes’ has 83 missing values

Remarks:

- The training set is imbalanced (There are more dissatisfied passengers than satisfied passengers)
- The distribution of ‘Customer Type’, ‘Type of Travel’, and ‘Class’ is imbalanced

3.5. Data preprocessing

3.5.1. Preprocessing Steps

Depending on the specific characteristics of the data, additional preprocessing steps may be undertaken. These could include handling missing values, scaling features, or encoding categorical variables. To preprocess the data, we follow the steps outlined below (Fig 3.5.1).

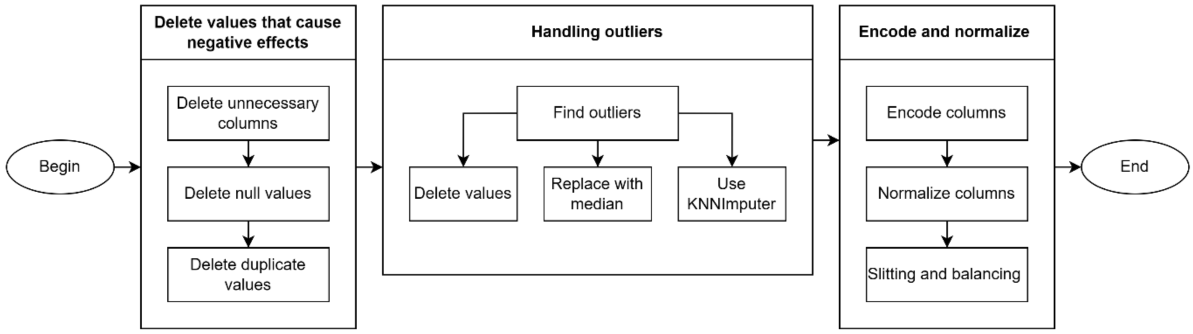

Figure 3.5.1: Preprocessing process

Step 1: Remove unnecessary columns such as the serial number column or id column.

Step 2: Eliminate null values.

Step 3: Eliminate duplicate values.

Step 4: Addressing outliers involves three methods: deleting values, replacing values with the median, and replacing values using KNNImputer.

Step 5: Encode columns.

Step 6: Normalize columns.

Step 7: Split the dataset.

Step 8: Balance the target columns.

3.5.2. Delete unnecessary columns

In datasets, colums ‘Id’ and ‘#’ are two columns that are not necessary in data classification and can be deleted.

3.5.3. Delete null values

Check and delete lines with null values, the missing data of the dataset is shown as follows (Fig 3.5.2).

	Total Missing	Percent Missing
Arrival Delay in Minutes	310	0.298
Gender	0	0.000
Seat comfort	0	0.000
Departure Delay in Minutes	0	0.000
Cleanliness	0	0.000
Inflight service	0	0.000
Checkin service	0	0.000
Baggage handling	0	0.000
Leg room service	0	0.000
On-board service	0	0.000
Inflight entertainment	0	0.000
Online boarding	0	0.000
Customer Type	0	0.000
Food and drink	0	0.000
Gate location	0	0.000
Ease of Online booking	0	0.000
Departure/Arrival time convenient	0	0.000
Inflight wifi service	0	0.000
Flight Distance	0	0.000
Class	0	0.000
Type of Travel	0	0.000
Age	0	0.000
satisfaction	0	0.000

	Total Missing	Percent Missing
Arrival Delay in Minutes	83	0.32
Gender	0	0.00
Seat comfort	0	0.00
Departure Delay in Minutes	0	0.00
Cleanliness	0	0.00
Inflight service	0	0.00
Checkin service	0	0.00
Baggage handling	0	0.00
Leg room service	0	0.00
On-board service	0	0.00
Inflight entertainment	0	0.00
Online boarding	0	0.00
Customer Type	0	0.00
Food and drink	0	0.00
Gate location	0	0.00
Ease of Online booking	0	0.00
Departure/Arrival time convenient	0	0.00
Inflight wifi service	0	0.00
Flight Distance	0	0.00
Class	0	0.00
Type of Travel	0	0.00
Age	0	0.00
satisfaction	0	0.00

Figure 3.5.2: Illustration of null value

We see that the Arrival Delay property has an empty value.

- The training set has 310 empty values, accounting for 0.298%
- The test set has 83 empty values, accounting for 0.32%.

3.5.4. Delete duplicate values

First we check whether overlapping values exist.

There is no duplicate values in both test and train set.

3.5.5. Handling outliers

Calculate the percentage of outliers using quantile

The quartiles are a statistical measure describing the distribution and dispersion of a dataset. There are three quartile values: the first quartile (Q1), the second quartile (Q2), and the third quartile (Q3). These three values divide a dataset (sorted in ascending order) into four parts with an equal number of observations.

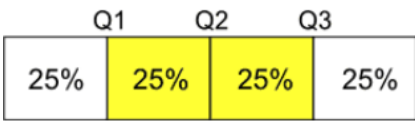

Figure 3.5.3: Illustration of Quantile

Calculates the first quartile (Q1), third quartile (Q3), and calculate interquartile range (IQR)(The interquartile range is a measure of statistical dispersion, representing the range between the first quartile (Q1) and the third quartile (Q3))

Using this formula.

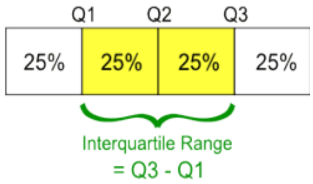

Figure 3.5.4: Illustration of Interquartile range

Lower Bound of a Quantile is the minimum value within that quantile. It represents the threshold below which a certain percentage of the data falls.

Upper Bound of a Quantile is the maximum value within that quantile. It represents the threshold above which a certain percentage of the data falls.

Calculate lower and upper bound

- Lower Bound:

$$Q1 - 1.5 * IQR \tag{3.5.1}$$

- Upper Bound:

$$Q3 + 1.5 * IQR \tag{3.5.2}$$

Outliers of our data show by picture under (Fig 3.5.5).

	Column Name	% Outliers
0	Age	0.0000
1	Arrival Delay in Minutes	13.4699
2	Baggage handling	0.0000
3	Checkin service	12.4071
4	Class	0.0000
5	Cleanliness	0.0000
6	Customer Type	0.0000
7	Departure Delay in Minutes	13.9274
8	Departure/Arrival time convenient	0.0000
9	Ease of Online booking	0.0000
10	Flight Distance	2.2077
11	Food and drink	0.0000
12	Gate location	0.0000
13	Gender	0.0000
14	Inflight entertainment	0.0000
15	Inflight service	0.0000
16	Inflight wifi service	0.0000
17	Leg room service	0.0000
18	On-board service	0.0000
19	Online boarding	0.0000
20	Seat comfort	0.0000
21	Type of Travel	0.0000
22	satisfaction	0.0000

	Column Name	% Outliers
0	Age	0.0000
1	Arrival Delay in Minutes	13.6639
2	Baggage handling	0.0000
3	Checkin service	12.3817
4	Class	0.0000
5	Cleanliness	0.0000
6	Customer Type	0.0000
7	Departure Delay in Minutes	13.6794
8	Departure/Arrival time convenient	0.0000
9	Ease of Online booking	0.0000
10	Flight Distance	2.2400
11	Food and drink	0.0000
12	Gate location	0.0000
13	Gender	0.0000
14	Inflight entertainment	0.0000
15	Inflight service	0.0000
16	Inflight wifi service	0.0000
17	Leg room service	0.0000
18	On-board service	0.0000
19	Online boarding	0.0000
20	Seat comfort	0.0000
21	Type of Travel	0.0000
22	satisfaction	0.0000

Figure 3.5.5: Illustration of showing outliers of datasets

Outliers of Train data

- Arrival Delay in Minutes' ~ 13.42%
- 'Checkin service' ~ 12.40%
- 'Departure Delay in Minutes'~ 13.92%
- 'Flight Distance' ~ 2.20%.

Outliers of Test data

- 'Arrival Delay in Minutes' ~ 13.62%.
- 'Checkin service' ~ 12.38%.
- 'Departure Delay in Minutes' ~ 13.73%.
- 'Flight Distance' ~ 2.24%.

In the ‘Flight Distance’ column, we can drop outliers because the percentage is very small ~2% .

In the ‘Checkin Service’ column, we use median to replace outliers. The median is a statistical measure that represents the middle value of a dataset when it is arranged in numerical order. In other words, it is the middle value that separates the higher half from the lower half of a dataset. We replace outliers as follow steps

- Step 1: Find median of column
- Step 2: Find lower bound and upper bound using quantile
- Step 3: Replace outliers with median

In the ‘Departure delay in Minutes’ column we use KNN to replace new value for the outlier using n_neighbors=5

3.5.6. Encoding target column

This project using LabelEncoder, LabelEncoder is a tool in the Python scikit-learn library. Using LabelEncoder helps us conveniently convert labels into integer values.

3.5.7. Unnecessary features

The heatmap visualizes the correlation between different numerical features in dataset. Heat map of the data as follow (Fig 3.5.6).

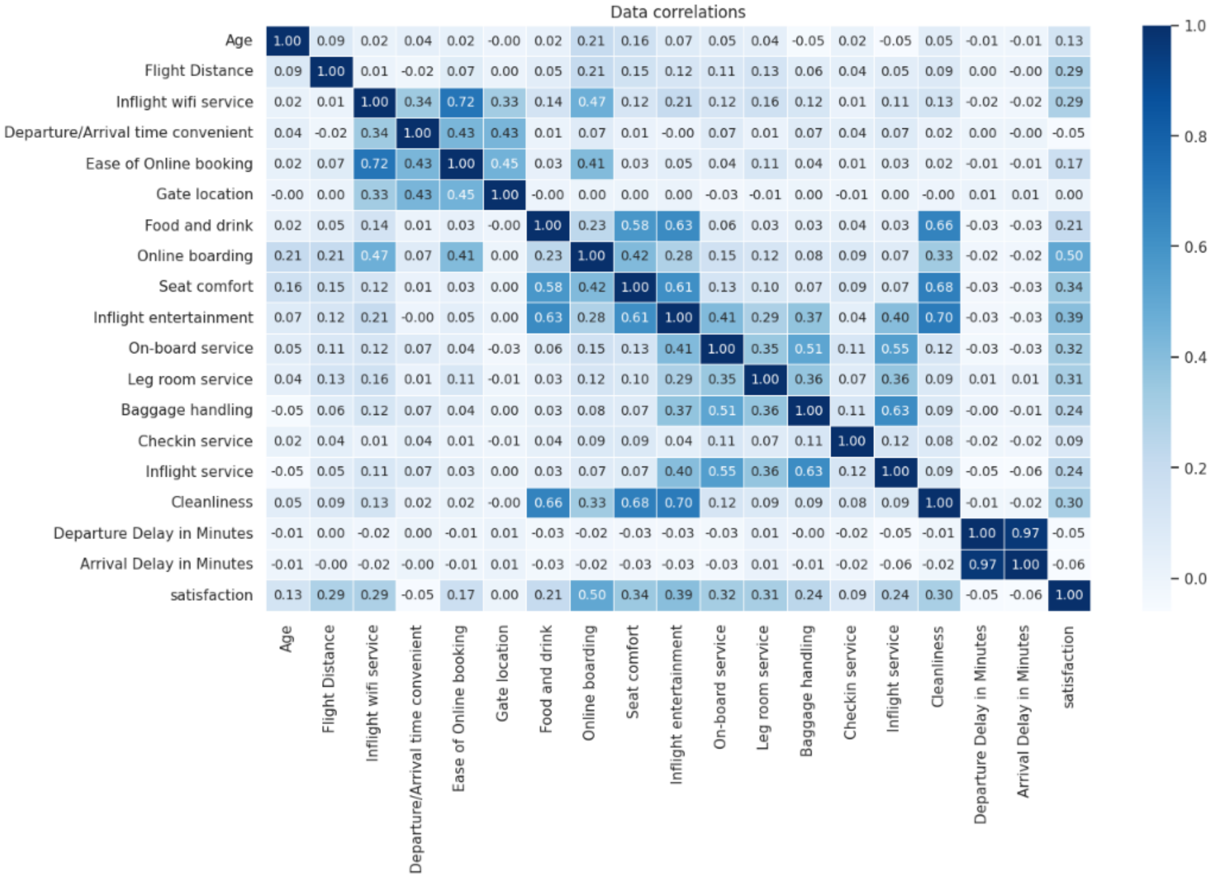

Figure 3.5.6: Heatmap represent data correlation

It can be seen that columns like “Gender”, “Arrival Delay in Minutes”, “Gate location”, “Departure/Arrival time convenient” have low data correlation so are not necessary in the algorithm.

3.5.8. Data encoding

Because ['Customer Type', 'Type of Travel', 'Class'] columns don’t follow the order like 1,2,3 we can use onehot encoding for this columns. We choose OneHotEncoder to encrypt the data.

Encoded dataset as below (Fig 3.5.7).

Customer Type_Loyal Customer	Customer Type_disloyal Customer	Type of Travel_Business travel	Type of Travel_Personal Travel	Class_Business	Class_Eco	Class_Eco Plus
1.0	0.0	0.0	1.0	0.0	0.0	1.0
0.0	1.0	1.0	0.0	1.0	0.0	0.0
1.0	0.0	1.0	0.0	1.0	0.0	0.0
1.0	0.0	1.0	0.0	1.0	0.0	0.0
1.0	0.0	1.0	0.0	1.0	0.0	0.0

Figure 3.5.7: Endcoded data

3.5.9. Standardized data

Standardized data to have a mean of 0 and a standard deviation of 1 for the algorithm running smoothly. We choose StandardScaler to normalize the data

Dataset after normalize as below (Fig 3.5.8 and Fig 3.5.9)

	Age	Flight Distance	Inflight wifi service	Ease of Online booking	Food and drink	Online boarding	Seat comfort
mean	0.000000	-0.000000	-0.000000	0.000000	0.000000	0.000000	-0.000000
min	-2.129774	-1.191352	-2.059601	-1.971745	-2.403238	-2.394896	-2.594202
max	3.015036	2.830740	1.713246	1.608366	1.353866	1.305166	1.188032
median	0.046876	-0.342425	0.204107	0.176322	-0.148976	-0.174859	0.431585
std	1.000005	1.000005	1.000005	1.000005	1.000005	1.000005	1.000005

Figure 3.5.8: Illustration of normalized data (1)

Inflight entertainment	On-board service	Leg room service	Baggage handling	Checkin service	Inflight service	Cleanliness	Departure Delay in Minutes
0.000000	-0.000000	-0.000000	-0.000000	0.000000	-0.000000	-0.000000	-0.000000
-2.507508	-2.617614	-2.538760	-2.223627	-1.871823	-3.091588	-2.496414	-0.387004
1.236425	1.259642	1.258473	1.160950	1.371635	1.158239	1.308499	41.436622
0.487638	0.484191	0.499026	0.314806	-0.250094	0.308274	-0.213466	-0.387004
1.000005	1.000005	1.000005	1.000005	1.000005	1.000005	1.000005	1.000005

Figure 3.5.9: Illustration of normalized data (2)

3.5.10. Slitting

As the data has already been divided into training and testing sets, further splitting is unnecessary. Our focus now is to designate the training set as 'X_train' and the test set as 'X_test', both excluding the 'satisfaction' column. The target columns will be labeled as 'y_train' and 'y_test', with the 'satisfaction' column retained in their respective datasets. This approach streamlines the dataset preparation for model training and testing in our analysis..

3.5.11. Data balancing

The data exhibits an imbalance between values 0 and 1. To address this issue, we have opted to implement Random Over-sampling as a strategy to balance the dataset. Figure 3.5.10 will illustrate before and after balance data(Fig 3.5.10).

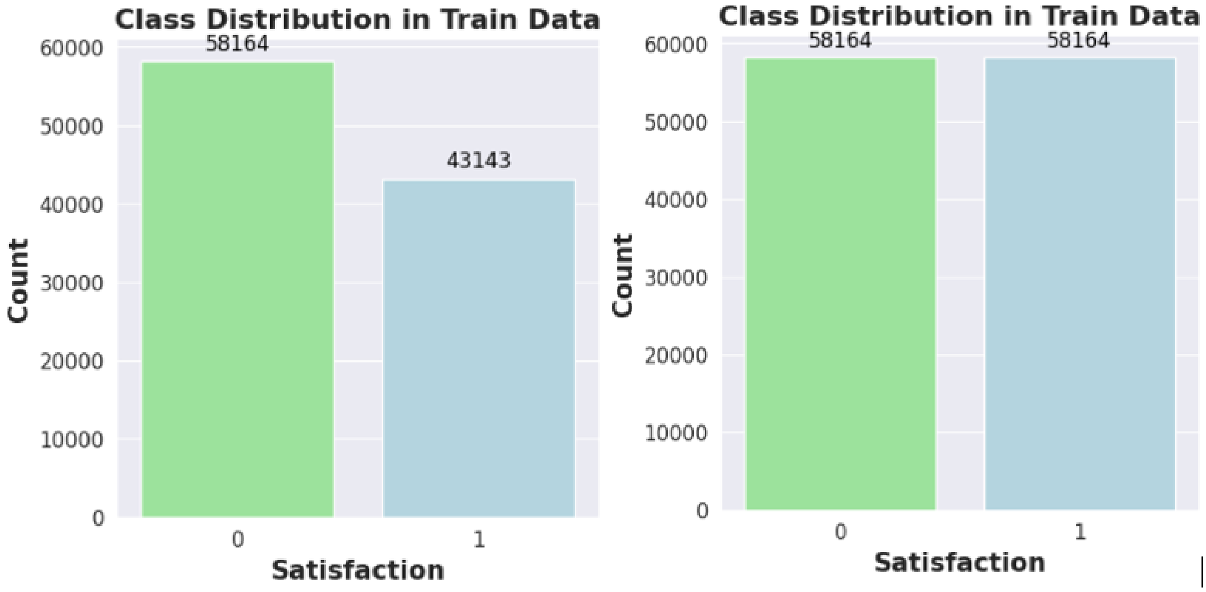

Figure 3.5.10: Compare distribution of target columns

3.6. Model building

3.6.1. Model K-Nearest Neighbors (KNN)

In this study, we focus on employing the K-Nearest Neighbors (KNN) model with a specified parameter k=7, chosen after careful screening for optimal performance. Simultaneously, we implement K-folds Cross Validation to assess the accuracy and stability of the model on the test data. The anticipated results aim to provide valuable insights for deploying the KNN model in real-world applications.

3.6.2. Model Support Vector Machine (SVM)

In this object, we use SVM by default the SVC class uses the Radial Basis Function (RBF) kernel, which is commonly referred to as the Gaussian kernel. The RBF kernel is flexible and works well in a variety of scenarios.

The default value for the regularization parameter C is 1.0. The parameter C controls the trade-off between having a smooth decision boundary and classifying the training points correctly. Higher values of C result in a more strict classification of the training data, possibly leading to overfitting, while lower values may allow for a smoother decision boundary, possibly leading to underfitting.

3.6.3. Model Naïve Bayes (NB)

In this investigation, our focus shifts towards the implementation of the Naive Bayes algorithm. Specifically, we employ the Gaussian Naive Bayes (GaussianNB) model, which assumes feature independence within the dataset. Unlike K-Nearest Neighbors, the Gaussian Naive Bayes model does not involve the parameter k. Our attention is directed towards fine-tuning parameters related to probability estimation, such as Laplace smoothing, to optimize the model's performance on the test data.

Simultaneously, we integrate K-folds Cross Validation to rigorously evaluate the accuracy and stability of the Gaussian Naive Bayes model on the test dataset. The envisioned outcomes of this analysis aspire to provide meaningful insights, aiding in the effective deployment of the Gaussian Naive Bayes algorithm in practical, real-world applications.

3.6.4. Model Decision Tree (DT)

Deploying the Decision Tree model with the criteria set to 'gini' and splitter set to 'best'. Concurrently, we are implementing K-folds Cross Validation to evaluate the accuracy and stability of the model on the train data. The anticipated outcomes aim to furnish precise accuracy metrics, contributing valuable insights for the deployment of the Decision Tree model in this problem.

3.7. Deploy into practice

To deploy to the web, we need to perform some steps as follows:

Step 1. Use “Joblib” to export file model trained.

Step 2. Use HTML to create a Front-end file.

Step 3. Use Flask to run the server environment.

Step 4. Add model files and scaler files to the Back-end environment.

Step 5. Deploy and demonstrate results.

The website interface is shown as follows (Fig 3.7.1, Fig 3.7.2, Fig 3.7.3, Fig 3.7.4):

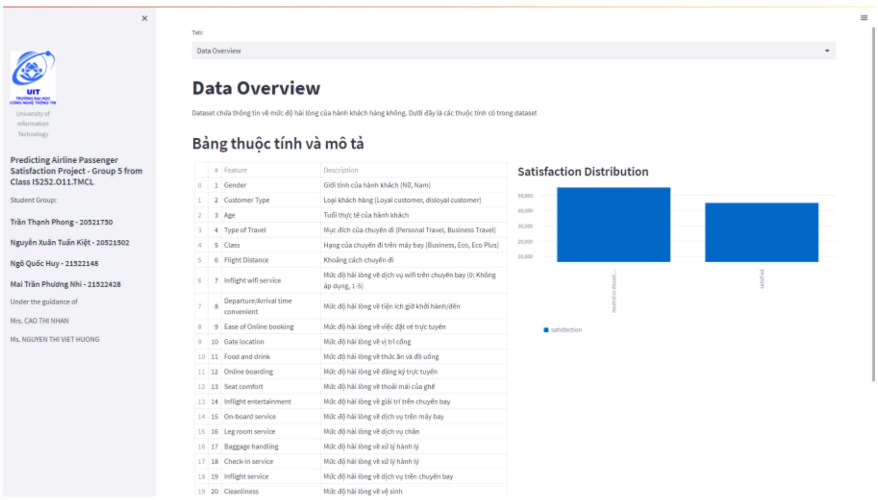

Figure 3.7.1: Data overview in web

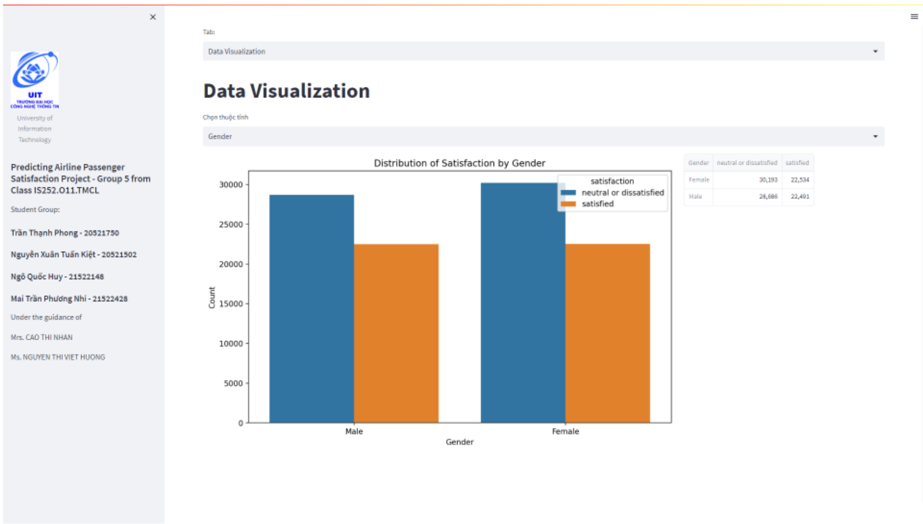

Figure 3.7.2: Data visualization in web

Figure 3.7.3: Satisfied prediction in web

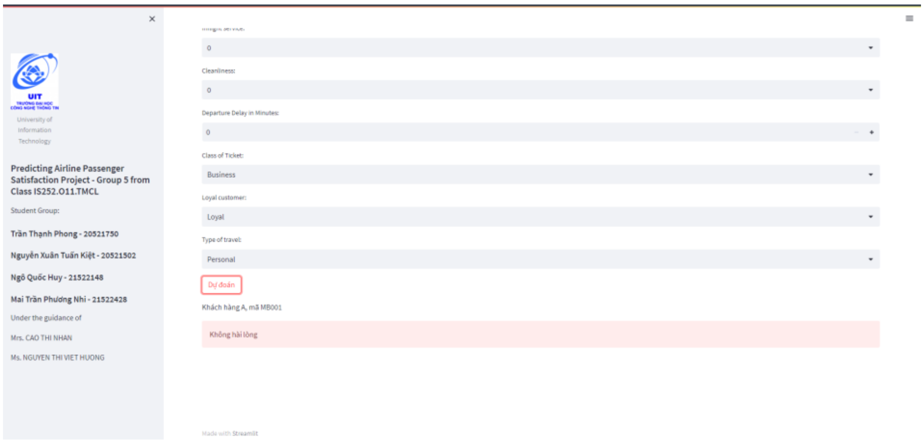

Figure 3.7.4: Result of prediction in web

Chapter 4

4. Experiment

4.1. Tools used

In this project, our primary programming language is Python, and we use Google Colab as our development tool. Several libraries support our project. We use the pandas library to process data in the form of data frames. Seaborn, Matplotlib, and Waffle are employed to create visualizations for data representation. Numpy and SciPy libraries are used for computations and statistical tasks. Scikit-learn provides tools and various machine learning classifiers along with evaluation metrics. Imbalanced Learn is use to address imbalances in datasets.

4.2. K-fold cross-validation method

Cross-validation is a statistical method used to estimate the skill of machine learning models. It is commonly used in applied machine learning to compare and select a model for a given predictive modeling problem because it is easy to understand, easy to implement, and results in skill estimates that generally have a lower bias than other methods.

The dataset is divided into k subsets or folds. The model is trained and evaluated k times, using a different fold as the validation set each time. Performance metrics from each fold are averaged to estimate the model's generalization performance. This method aids in model assessment, selection, and hyperparameter tuning, providing a more reliable measure of a model's effectiveness.

In each set (fold) training and the test would be performed precisely once during this entire process. It helps us to avoid overfitting. As we know when a model is trained using all of the data in a single short and give the best performance accuracy. To resist this k-fold cross-validation helps us to build the model is a generalized one. The following model will demonstrate the implementation steps of K-Folds CV (Fig 4.2.1)

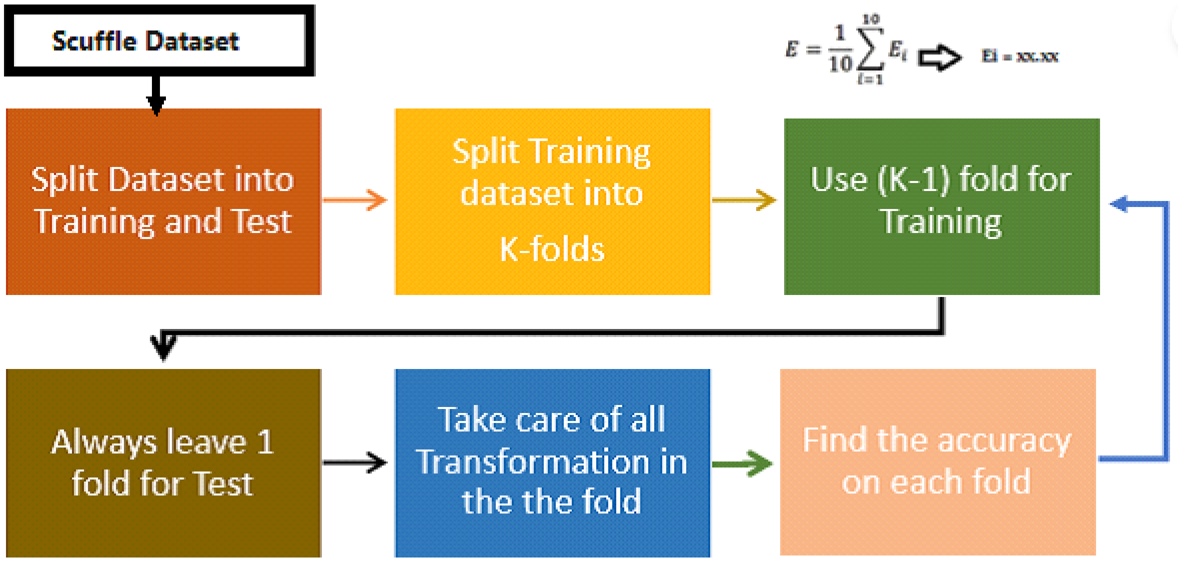

Figure 4.2.1: K-fold cross validation process model

The biggest advantage of using the K-Fold CV technique is that it does not care about how the data is divided [17].

In the test set, every data point appears exactly once, but in the training set, it appears 'k-1′ times. This K-Fold CV technique follows some basic steps (Fig 4.2.2):

Step 1: Choose a number of K folds.

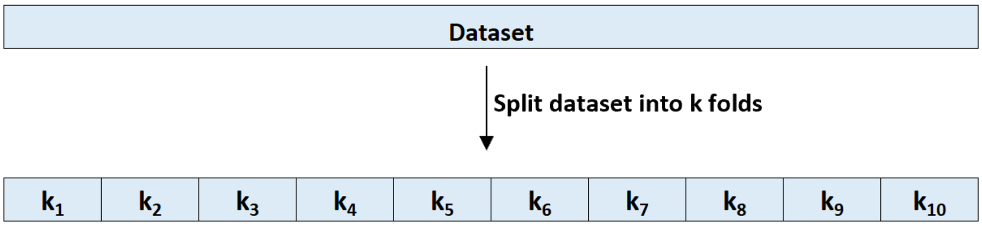

Figure 4.2.2: Dividing a data set into folds

Step 2: Split the dataset into k equal parts.

Step 3: Assign k – 1 folds for the training set and the last fold will be for the test set.

Step 4: Train the model on training set.

Step 5: Verify the hypothesis at the test set.

Step 6: Save the validation outcome.

Step 7: Steps 3 through 6 should be repeated 'k' times total. Each time, use the last fold as a test set. Finally, validate the model on each fold (Fig 4.2.3).

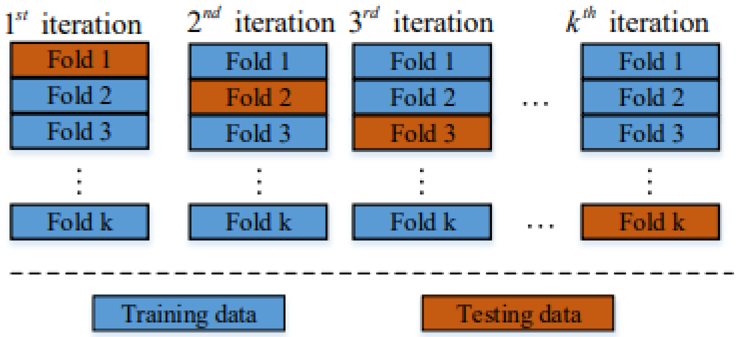

Figure 4.2.3: Looping from step 3 to step 6

Step 8: To have the final score, average the results got from step 6.

The choice of the K value will affect the number of iterations, with a larger K resulting in smaller data splits. It is important to choose a suitable K value, ideally the smallest K that still provides a representative average [18].

4.3. Evaluation forecasting models

4.3.1. Confusion Matrix

To evaluate the accuracy of the classification model, our team uses 4 parameters: Accuracy, Precision, Recall and F1-score. These parameters are calculated through the confusion matrix.

Confusion matrix [19] is shown in Table 4.3.1 as follows.

Predicted label			
True label		Positive	Negative
	Positive	TP	FN
	Negative	FP	TN

Table 4.3.1: Confusion matrix

Observing the confusion matrix, we have the following information:

- TP (true positive) – Values are actually Positive and predicted to be Positive.
- FN (false negative) – The values are actually Positive but are incorrectly predicted to be Negative. Also known as Type II Error.
- FP (false positive) – The values are actually Negative but are incorrectly predicted as Positive. Also known as Type I Error.
- TN (true negative) – The values are actually Negative and are predicted to be Negative.

4.3.2. Evaluation Metrics

Accuracy (4.3.1):

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (4.3.1)$$

Recall (4.3.2):

$$Recall = \frac{TP}{TP+FN} \quad (4.3.2)$$

Precision (4.3.3):

$$Precision = \frac{TP}{TP+FP} \quad (4.3.3)$$

F1-score (4.3.4):

$$F1 - Score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (4.3.4)$$

4.4. Predicting Airline Passenger Satisfaction

Our team utilized four classification algorithms: SVM, KNN, Decision Tree, and Naïve Bayes. We executed the models on the test datasets and obtained the following results.

The confusion matrix of the four models is as follows:

- Confusion matrix of K-Nearest Neighbors model

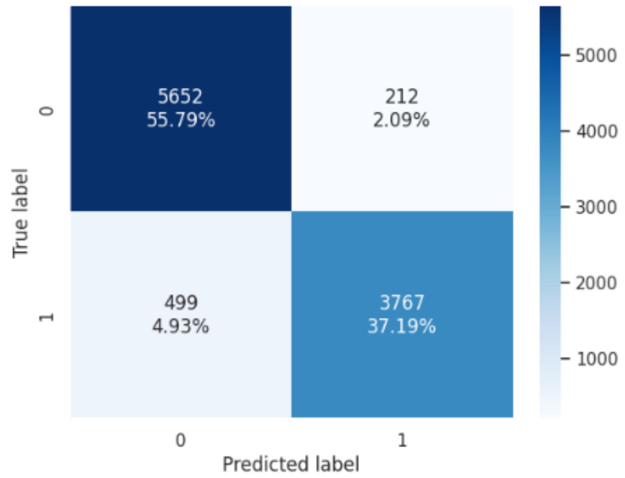

Figure 4.4.1: Confusion matrix of KNN model

- Confusion matrix of Support Vector Machine

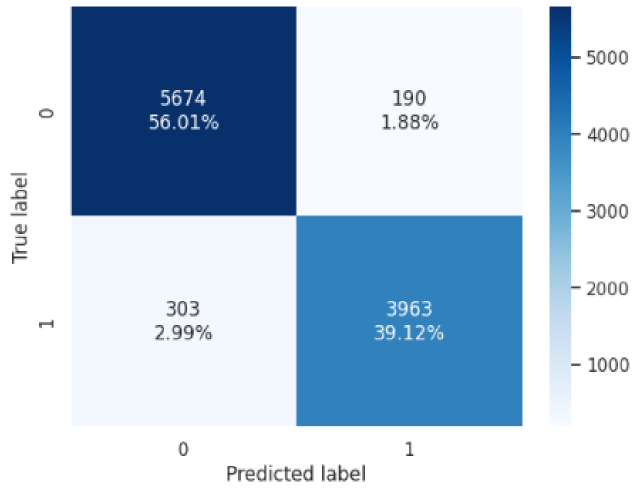

Figure 4.4.2: Confusion matrix of SVM model

- Confusion matrix of Naïve Bayes model

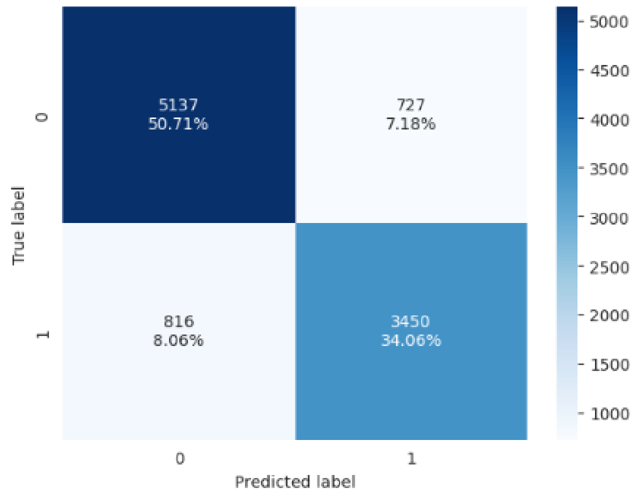

Figure 4.4.3: Confusion matrix of Naïve Bayes model

- Confusion matrix of Decision Tree model

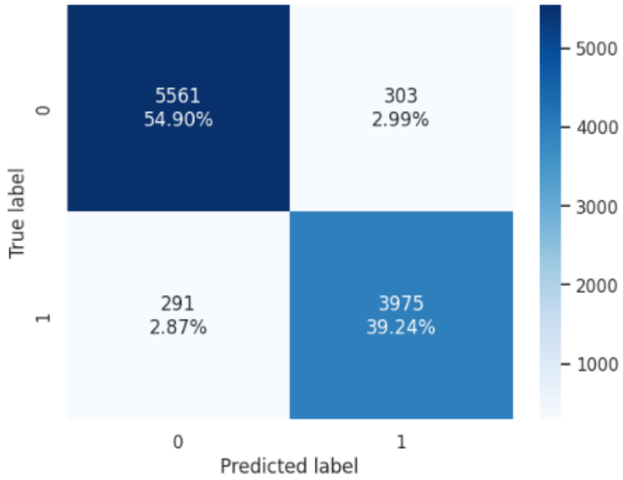

Figure 4.4.4: Confusion matrix of Decision Tree model

Evaluation Metrics of models

- For neutral or dissatisfied cases is shown in Table 4.4.1 as follows :

	Accuracy	Precision	Recall	F1-score
SVM	93%	92%	96%	94%
KNN	93%	92%	96%	94%
Decision Tree	94%	95%	95%	95%
Naïve Bayes	84%	86%	87%	86%

Table 4.4.1: Results when passengers are dissatisfied or neutral

- For satisfied cases is shown in Table 4.4.2 as follows:

	Accuracy	Precision	Recall	F1-score
SVM	93%	95%	89%	92%
KNN	93%	95%	89%	92%
Decision Tree	94%	94%	94%	94%
Naïve Bayes	84%	83%	81%	82%

Table 4.4.2: Results when passengers are satisfied

Following the evaluation of the four algorithmic models on the test dataset, notable levels of accuracy were observed for each algorithm. The primary goal of our team is to ascertain the most effective algorithm in predicting customer satisfaction within the airline industry. This undertaking is designed to empower airlines in a proactive approach towards enhancing service quality, pinpointing areas for process refinement, and ultimately elevating customer satisfaction.

4.5. Discussion

Upon completing the analysis of the four algorithms—KNN, SVM, Naïve Bayes, and Decision Tree—we observed that achieving a result above 0.8 is indicative of excellent performance. This outcome is attributed to the quality of our data and the efficacy of our processing methods. Consequently, we are now equipped to predict customer satisfaction with their flight.

The SVM (Support Vector Machine) emerged as the most promising model, boasting the highest score among the algorithms evaluated. Notably, SVM, being a classification algorithm, excels in high-dimensional spaces, rendering it particularly suitable for datasets characterized by a substantial number of features, as is the case with our dataset.

Conversely, the Naïve Bayes model exhibited the least favorable performance. This can be elucidated by the inherent simplicity of the Naïve Bayes model, which assumes independence among all features in the dataset. However, in the realm of customer satisfaction, factors are intricately interrelated, thereby challenging the oversimplified assumptions of the Naïve Bayes model.

In conclusion, our comprehensive exploration leveraging machine learning models has fulfilled the objectives set for this endeavor. The SVM model emerges as the optimal choice for predicting customer satisfaction in the realm of air travel, considering its robust performance in handling complex, high-dimensional datasets.

Chapter 5

5. Conclusion

In summary, our research successfully applied data mining and machine learning techniques to predict airline passenger satisfaction. The use of K-Nearest Neighbors, Support Vector Machines, Decision Tree, and Naïve Bayes algorithms showed promising results. The study contributes valuable insights for airlines aiming to enhance customer service and satisfaction. While acknowledging some limitations, the research highlights the potethatntial of predictive models in adapting to changing customer preferences, emphasizing the importance of technology in the aviation industry's continual improvement. In future research, exploring advanced machine learning models and incorporating real-time data streams could further enhance the accuracy and adaptability of passenger satisfaction predictions. Additionally, investigating the integration of customer feedback loops and sentiment analysis tools would provide airlines with actionable insights for immediate service improvements. This ongoing commitment to leveraging cutting-edge technologies will be crucial in meeting the ever-evolving expectations of airline passengers and ensuring sustained customer loyalty.

Chapter 6

6. Reference

[1] X. Jiang, Y. Zhang, Y. Li, and B. Zhang, "Forecast and analysis of aircraft passenger satisfaction based on RF-RFE-LR model," *Sci. Rep.*, vol. 12, no. 1, Art. no. 1, Jul. 2022, doi: 10.1038/s41598-022-14566-3.

[2] A. Nurdina and A. B. I. Puspita, "Naive Bayes and KNN for Airline Passenger Satisfaction Classification: Comparative Analysis," *J. Inf. Syst. Explor. Res.*, vol. 1, no. 2, Art. no. 2, Jul. 2023, doi: 10.52465/joiser.v1i2.167.

[3] P. F. Orrù, A. Zoccheddu, L. Sassu, C. Mattia, R. Cozza, and S. Arena, "Machine Learning Approach Using MLP and SVM Algorithms for the Fault Prediction of a Centrifugal Pump in the Oil and Gas Industry," *Sustainability*, vol. 12, no. 11, Art. no. 11, Jan. 2020, doi: 10.3390/su12114776.

[4] L. Higgins, "Classification of Airline Customer Sentiment Expressed in Twitter Tweets using Lexicons, Decision Tree, and Naïve Bayes," masters, Dublin, National College of Ireland, 2022. Accessed: Dec. 16, 2023. [Online]. Available: https://norma.ncirl.ie/6131/

[5] W. Liu, S. Liang, and X. Qin, "Weighted p-norm distance t kernel SVM classification algorithm based on improved polarization," *Sci. Rep.*, vol. 12, no. 1, Art. no. 1, Apr. 2022, doi: 10.1038/s41598-022-09766-w.

[6] S. Sharma, K.-M. Osei-Bryson, and G. M. Kasper, "Evaluation of an integrated Knowledge Discovery and Data Mining process model," *Expert Syst. Appl.*, vol. 39, no. 13, pp. 11335–11348, Oct. 2012, doi: 10.1016/j.eswa.2012.02.044.

[7] H. H. P. Nucci *et al.*, "Use of computer vision to verify the viability of guavira seeds treated with tetrazolium salt," *Smart Agric. Technol.*, vol. 5, p. 100239, Oct. 2023, doi: 10.1016/j.atech.2023.100239.

[8] T. T. Nguyen *et al.*, "Scalable maximal subgraph mining with backbone-preserving graph convolutions," *Inf. Sci.*, vol. 644, p. 119287, Oct. 2023, doi: 10.1016/j.ins.2023.119287.

[9] X. Shu and Y. Ye, "Knowledge Discovery: Methods from data mining and machine learning," *Soc. Sci. Res.*, vol. 110, p. 102817, Feb. 2023, doi: 10.1016/j.ssresearch.2022.102817.

[10] D. Lopez-Bernal, D. Balderas, P. Ponce, and A. Molina, "Education 4.0: Teaching the Basics of KNN, LDA and Simple Perceptron Algorithms for Binary Classification Problems," *Future Internet*, vol. 13, no. 8, Art. no. 8, Aug. 2021, doi: 10.3390/fi13080193.

[11] R. Suwanda, Z. Syahputra, and E. M. Zamzami, "Analysis of Euclidean Distance and Manhattan Distance in the K-Means Algorithm for Variations Number of Centroid K," *J. Phys. Conf. Ser.*, vol. 1566, no. 1, p. 012058, Jun. 2020, doi: 10.1088/1742-6596/1566/1/012058.

[12] É. O. Rodrigues, "Combining Minkowski and Chebyshev: New distance proposal and survey of distance metrics using k-nearest neighbours classifier," *Pattern Recognit. Lett.*, vol. 110, pp. 66–71, Jul. 2018, doi: 10.1016/j.patrec.2018.03.021.

[13] S. Huang, N. Cai, P. P. Pacheco, S. Narrandes, Y. Wang, and W. Xu, "Applications of Support Vector Machine (SVM) Learning in Cancer Genomics," *Cancer Genomics Proteomics*, vol. 15, no. 1, pp. 41–51, Jan. 2018.

[14] Y. Peng, P. H. M. Albuquerque, J. M. Camboim de Sá, A. J. A. Padula, and M. R. Montenegro, “The best of two worlds: Forecasting high frequency volatility for cryptocurrencies and traditional currencies with Support Vector Regression,” *Expert Syst. Appl.*, vol. 97, pp. 177–192, May 2018, doi: 10.1016/j.eswa.2017.12.004.

[15] S. Lee, K. Choi, and D. Yoo, “Building a core rule-based decision tree to explain the causes of insolvency in small and medium-sized enterprises more easily,” *Humanit. Soc. Sci. Commun.*, vol. 10, no. 1, Art. no. 1, Dec. 2023, doi: 10.1057/s41599-023-02382-7.

[16] L. Jiang, H. Zhang, and Z. Cai, “A Novel Bayes Model: Hidden Naive Bayes,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 10, pp. 1361–1371, Oct. 2009, doi: 10.1109/TKDE.2008.234.

[17] A. R. Bhatt, A. Ganatra, and K. Kotecha, “Cervical cancer detection in pap smear whole slide images using convNet with transfer learning and progressive resizing,” *PeerJ Comput. Sci.*, vol. 7, p. e348, Feb. 2021, doi: 10.7717/peerj-cs.348.

[18] I. K. Nti, O. Nyarko-Boateng, and J. Aning, “Performance of Machine Learning Algorithms with Different K Values in K-fold Cross-Validation,” *Int. J. Inf. Technol. Comput. Sci.*, vol. 13, no. 6, p. 61.

[19] G. Phillips *et al.*, “Setting nutrient boundaries to protect aquatic communities: The importance of comparing observed and predicted classifications using measures derived from a confusion matrix,” *Sci. Total Environ.*, vol. 912, p. 168872, Feb. 2024, doi: 10.1016/j.scitotenv.2023.168872.

Chapter 7

7. Assign tasks

The contributions and tasks of the members are shown in Table 7.14.5.1 as follows:

Member	Task	Assessment
Trần Thạnh Phong 20521750	Learn the KNN algorithm Preprocess data Embed the model in a website Learn K-fold cross validation Write reports	Well Done 10/10
Nguyễn Xuân Tuân Kiệt 20521502	Learn SVM algorithm Data encryption Exception data handling Support report writing	Well Done 10/10
Ngô Quốc Huy 21522148	Learn the Naïve Bayes algorithm Learn model performance evaluation indicators Implement actual Web pages Support report writing	Well Done 10/10
Mai Trần Phương Nhi 21522428	Learn the Decision Tree algorithm Visualize data Support Web site implementation Summary of used tools Support report writing	Well Done 10/10

Table 7.14.5.1: Assign tasks