

# **Airline Passenger Satisfaction**

*A report submitted in partial fulfillment of the requirements for the Award of Degree of*

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**By**

**Gode Sri Chandra Shekhar**

**Regd. No.: 20B91A0592**

**Under Supervision of Mr. Gundala Nagaraju Henotic  
Technology Pvt Ltd, Hyderabad (Duration: 7th July,  
2022 to 6th September, 2022)**



**DEPARTMENT OF COMPUTER SCIENCE AND  
ENGINEERING SAGI RAMA KRISHNAM RAJU ENGINEERING  
COLLEGE**

**(An Autonomous Institution)**

**Approved by AICTE, NEW DELHI and Affiliated to JNTUK, Kakinada**

**CHINNA AMIRAM, BHIMAVARAM,**

**ANDHRA PRADESH**

SAGI RAMA KRISHNAM RAJU ENGINEERING COLLEGE  
(Autonomous)  
Chinna Amiram, Bhimavaram

DEPARTMENT OF  
COMPUTER SCIENCE ENGINEERING



**CERTIFICATE**

This is to certify that the “**Summer Internship Report**” submitted by **GODE SRI CHANDRA SHEKHAR, 20B91A0592** is work done by him/her and submitted during 2021- 2022 academic year, in partial fulfillment of the requirements for the award of the Summer Internship Program for **Bachelor of Technology in Computer Science Engineering**, at **HENOTIC TECHNOLOGIES** from 07.07.2022 to 06.09.2022 for AIML

**Department Internship  
Coordinator**

**Dean -T & P Cell**

**Head of the Department**

## Table of Contents

<b>1.0</b>	<b>Introduction.....</b>	<b>1</b>
1.1.	What are the different types of Machine Learning? .....	2
1.2.	Benefits of Using Machine Learning in Airline Passenger Satisfaction .....	4
1.3.	About Airline Passenger Satisfaction.....	5
1.3.1	AI / ML Role in Airline Passenger Satisfaction .....	6
<b>2.0</b>	<b>Airline Passenger Satisfaction .....</b>	<b>7</b>
2.1.	Main Drivers for Airline Passenger Satisfaction .....	7
2.2.	Airline Passenger Satisfaction Project - Data Link .....	8
<b>3.0</b>	<b>AI / ML Modelling and Results.....</b>	<b>9</b>

3.1.	Airline Passenger Satisfaction Problem Statement.....	9
3.2.	Data Science Project Life Cycle .....	9
3.2.1	Data Exploratory Analysis .....	10
3.2.2	Data Pre-processing .....	10
3.2.2.1.	Check the Duplicate and low variation data .....	10
3.2.2.2.	Identify and address the missing variables .....	Error!
	Bookmark not defined.	
3.2.2.3.	Handling of Outliers .....	Error!
	Bookmark not defined.	
3.2.2.4.	Categorical data and Encoding Techniques.....	Error!
	Bookmark not defined.	
3.2.2.5.	Feature Scaling.....	Error!
	Bookmark not defined.	
3.2.3	Selection of Dependent and Independent variables.....	Error!
	Bookmark not defined.	
3.2.4	Data Sampling Methods .....	Error!
	Bookmark not defined.	
3.2.4.1.	Stratified sampling.....	Error!
	Bookmark not defined.	
3.2.4.2.	Simple random sampling .....	Error!
	Bookmark not defined.	
3.2.5	Models Used for Development.....	Error!
	Bookmark not defined.	
3.2.5.1.	Model 01 .....	Error!
	Bookmark not defined.	
3.2.5.2.	Model 02 .....	Error!
	Bookmark not defined.	
3.2.5.3.	Model 03 .....	Error!
	Bookmark not defined.	
3.2.5.4.	Model 04 .....	Error!
	Bookmark not defined.	
3.2.5.5.	Model 05 .....	Error!
	Bookmark not defined.	
3.2.5.6.	Model 06 .....	Error!
	Bookmark not defined.	
3.2.5.7.	Model 07 .....	Error!
	Bookmark not defined.	
3.2.5.8.	Model 08 .....	Error!
	Bookmark not defined.	
3.2.5.9.	Model 09 .....	Error!
	Bookmark not defined.	
3.2.5.10.	Model 10 .....	Error!
	Bookmark not defined.	
3.3.	AI / ML Models Analysis and Final Results.....	Error!
	Bookmark not defined.	
3.3.1	Different Model codes .....	Error!
	Bookmark not defined.	
4.0	Conclusions and Future work .....	24
5.0	References.....	27
6.0	Appendices.....	28

6.1.	Python code Results.....	28
6.2.	List of Charts.....	30
6.2.1	Visualization of Outliers.....	30
6.2.2	Histograms .....	31
6.2.3	Airline Satisfaction Graph.....	31
6.2.4	Flight Satisfaction Pie-Chart.....	32

## **Abstract**

This study was conducted to understand customer experience and satisfaction through airline passengers' online review. To achieve the purpose of this study, the semantic network analysis was conducted qualitatively by collecting reviews in top 10 airlines selected by Skytrax (airlinequality.com). In addition, this study quantitatively identified the relationship among six evaluation factors (seat comfort, staff, food and beverage (F&B), entertainment, ground service, and value for money), customer satisfaction and recommendation. This study collected 9632 reviews from the Skytrax. Through a CONCOR (CONvergence of iterated CORrelation) analysis, keywords were grouped into six clusters (seat comfort, staff, entertainment, ground service, value for money, and airline brand). Through the linear regression analysis, all evaluation factors except 'entertainment' factor significantly had impact on customer satisfaction and recommendation. These results showed that understanding online review can provide both academic implication and practical implication to develop sustainable strategy in the airline industry.

The aim of this research is to conduct a comparative study of passenger satisfaction and service quality in Nigeria among the considered airlines (Virgin Atlantic, British airways and Med-view airline) on Lagos to London route/flights. The research was based on assessing the level of passenger satisfaction on the airline service quality attributes/dimensions. The study shows and ranked Virgin Atlantic's services first, British airways services second and Med-view airline services third. The common and important services that the airlines respondent's indicated to be more important are 'safety and security, 'Appealing appearance, attitude and uniform of employees', 'flight attendant's courteousness', 'Behaviour of Staffs instils confidence in customers', and 'Staffs good communication skills'. A new National carrier should be established by the government and structures and frameworks need to be put in place to ensure good quality services were offered to passengers and services ranked as importance in this research should be giving optimum consideration for better service delivery and inturn excellent passenger satisfaction.

## 1.0 Introduction

With the increasing power of computer technology, companies and institutions can now store large amounts of data at reduced cost. The amount of available data is increasing exponentially and cheap disk storage makes it easy to store data that previously was thrown away. There is a huge amount of information locked up in databases that is potentially important but has not yet been explored. The growing size and complexity of the databases makes it hard to analyse the data manually, so it is important to have automated systems to support the process. Hence there is the need of computational tools able to treat these large amounts of data and extract valuable information.

Customer's experience is one of the important concern for airline industries. Twitter is one of the popular social media platform where flight travelers share their feedbacks in the form of tweets. This study presents a machine learning approach to analyze the tweets to improve the customer's experience. Features were extracted from the tweets using word embedding with Glove dictionary approach and n-gram approach. Further, SVM (support vector machine) and several ANN (artificial neural network) architectures were considered to develop classification model that maps the tweet into positive and negative category. Additionally, convolutional neural network (CNN) were developed to classify the tweets and the results were compared with the most accurate model among SVM and several ANN architectures. It was found that CNN outperformed SVM and ANN models. In the end, association rule mining have been performed on different categories of tweets to map the relationship with sentiment categories. The results show that interesting associations were identified that certainly helps the airline industries to improve their customer's experience.

Many studies have employed survey methods to measure service quality in the airline industry. However, a few recent studies have highlighted the advantages of analyzing online review data for studying customers' satisfaction or their experience of the airline. Online reviews are critical since it is a significant source for business growth, performance and improvement of customer experience, and allow airline companies to conduct two-way communication with airline passengers. Moreover, electronic word of mouth (eWOM) shared by other airline passengers are considered trustworthy, fast and widespread. In the previous study, the service quality of airline passenger has been measured in various ways. They found that

service expectations differed between the two groups. Significant differences were found among passengers from different ethnic groups and among passengers who travel for different purposes. However, there were limited studies on the understanding experience and satisfaction of airline passengers using both qualitative and quantitative methods to analyze over 9000 online reviews.

The main contribution of this study is the understanding of customer experience and satisfaction through the airline passengers' online review. In order to reach the purpose, large amounts of customer reviews were collected from Skytrax (airlinequality.com). The analysis can be divided into two parts. One was to analyze the meaning of words extracted from the review data using the semantic network analysis by qualitative analysis. The other was conducted using the quantitative analysis method to understand relationships among six evaluation factors, customer satisfaction and recommendation.

### **1.1. What are the different types of Machine Learning?**

Machine learning is a subset of AI, which enables the machine to automatically learn these algorithms to train them, and on the basis of training, they build the model & perform a specific task.

These ML algorithms help to solve different business problems like Regression, Classification, Forecasting, Clustering, and Associations, etc.

Based on the methods and way of learning, machine learning is divided into mainly four types, which are:

1. Supervised Machine Learning
2. Unsupervised Machine Learning
3. Reinforcement Learning

#### **Supervised learning**

The algorithm makes predictions and is corrected by the operator – and this process continues until the algorithm achieves a high level of accuracy/performance.

Under the umbrella of supervised learning fall:

Classification, Regression and Forecasting.

1. **Classification:** In classification tasks, the machine learning program must draw a conclusion from observed values and determine to what category new observations belong. For example, when filtering emails as 'spam' or 'not spam', the program must look at existing observational data and filter the emails accordingly.



2. **Regression:** In regression tasks, the machine learning program must estimate – and understand – the relationships among variables. Regression analysis focuses on one dependent variable and a series of other changing variables – making it particularly useful for prediction and forecasting.
3. **Forecasting:** Forecasting is the process of making predictions about the future based on the past and present data, and is commonly used to analyse trends.

### **Semi-supervised learning**

Semi-supervised learning is similar to supervised learning, but instead uses both labelled and unlabelled data. Labelled data is essentially information that has meaningful tags so that the algorithm can understand the data, whilst unlabelled data lacks that information. By using this combination, machine learning algorithms can learn to label unlabelled data.

### **Unsupervised learning**

Here, the machine learning algorithm studies data to identify patterns. There is no answer key or human operator to provide instruction. Instead, the machine determines the correlations and relationships by analysing available data. In an unsupervised learning process, the machine learning algorithm is left to interpret large data sets and address that data accordingly. The algorithm tries to organise that data in some way to describe its structure. This might mean grouping the data into clusters or arranging it in a way that looks more organised. As it assesses more data, its ability to make decisions on that data gradually improves and becomes more refined.

Under the umbrella of unsupervised learning, fall:

1. **Clustering:** Clustering involves grouping sets of similar data (based on defined criteria). It's useful for segmenting data into several groups and performing analysis on each data set to find patterns.
2. **Dimension reduction:** Dimension reduction reduces the number of variables being considered to find the exact information required.

### **Reinforcement learning**

Reinforcement learning focuses on regimented learning processes, where a machine learning algorithm is provided with a set of actions, parameters and end values. By defining the rules, the machine learning algorithm then tries to explore different options and possibilities, monitoring and evaluating each result to determine which one is optimal. Reinforcement learning teaches the machine trial and error. It learns

from past experiences and begins to adapt its approach in response to the situation to achieve the best possible result. The Reinforcement Learning process is similar to a human being; for example, a child learns various things by experiences in his day-to-day life. An example of reinforcement learning is to play a game, where the Game is the environment, moves of an agent at each step define states, and the goal of the agent is to get a high score. Agent receives feedback in terms of punishment and rewards. Due to its way of working, reinforcement learning is employed in different fields such as Game theory, Operation Research, Information theory, multi-agent systems. A reinforcement learning problem can be formalized using Markov Decision Process (MDP). In MDP, the agent constantly interacts with the environment and performs actions; at each action, the environment responds and generates a new state

## **1.2. Benefits of Using Machine Learning in Airline Passenger Satisfaction**

### **1. Improved decision making:**

One of the benefits of machine learning in banking is improved decision making. As compared to traditional methods, artificial intelligence helps banks to calculate credit scores accurately. The main reason ML can do this is that it can provide an objective evaluation without any bias. The huge amount of data collected from the potential borrower assists banks in making better decisions.

### **2. Better risk management:**

AI and ML reduce risks for both customers and banks through accurate reporting. Artificial intelligence can also make predictions based on transaction history after giving credit to customers. Employees have more insights into credit risk testing. Early detection of errors and the availability of potential future risks helps the banking industry to prepare in advance.

### **3. Prevention of fraud:**

Credit card fraud is a huge problem in the banking industry. Machine learning for banking can significantly lower the number of fraudulent activities. The majority of fraud occurs when customers pay for products, whether online or offline. Machine learning in banking prevents this from happening in several ways. For example, facial recognition can be used to confirm the person using a credit card is the owner

#### **4. Improved customer experience:**

With technology changing almost every aspect of life, consumers are looking for better services and eager to get the same from banking institutions. At the same time, banks that can provide more security and a personalized experience would attract more clients. Customers want digital banking products that are easy to use. One way in which ML improves the overall experience and services is by reducing the time it takes to make credit decisions and banking operations. Loan application, which used to take weeks, can now be made within days. Machine learning can make an unbiased analysis based on several credit factors

#### **5. Internal operational solutions:**

Machine learning in the banking sector has greatly changed internal operations for the better. Automation reduces the time staff spends on redundant tasks. Therefore, resources can be allocated towards improving the overall experience. Robots perform routine tasks with minimal risk of errors. So a bank can provide efficient solutions while automation gives employees the chance to pay more attention to the most important tasks. Using ML has so many advantages, with the most important one being internal operational solutions today. Robots can go through a customer database at record time, thus reducing the need for employees to do this manually.

#### **6. Marketing and lending solutions:**

ML and AI in fintech collect data and also search for specific patterns that help banks make better marketing predictions. Examples of predictions that ML can make include:

1. Changes in currencies,
2. The best investment ideas,
3. Credit risks,
4. The optimum loan agreement for a client

#### **1.3. About Airline Passenger Satisfaction**

The study aims will find out the relationship between customer experiences on airline services and customer satisfaction and also define the operational characteristic that influences customer choice in airline. In order to obtain the research objectives, the result of this study will answer the research question, which is what are the factors affecting customer choice in airline? The previous research reveals that customer satisfaction in airline industry can be measure by using SERQVUAL model to

evaluate customer experience based on five different elements which are tangibles, reliability, responsiveness, assurance and empathy. Each element has different level of impact on customer satisfaction. However, most of the services provided by airline are intangible. These kinds of services are engaging with customer emotion and are difficult to obtain. Furthermore, it requires higher cost to maintain high quality of both products and services. The result of the studies are important to help airline companies in creating attractive marketing strategies by an understanding of the customer expectation and market need.

### **1.3.1 AI / ML Role in Airline Passenger Satisfaction**

Machine Learning is a sub-set of artificial intelligence where computer algorithms are used to autonomously learn from data. Machine learning (ML) is getting more and more attention and is becoming increasingly popular in many other industries. Within the airline industry, there is more application of ML regarding the satisfaction.

## 2.0 Airline Passenger Satisfaction

Auto insurance premiums have historically been priced on underwriting and rating. Underwriting is a process where the insurer assesses the applicant's risk. They do this by incorporating personal information and internal claims data into weighted algorithms. Insurers then look at rating factors to predict the likelihood of a claim's submission. The rating assigns a price based on the projected cost to the insurer of assuming financial responsibility of potential claims. Auto insurance premiums fluctuate with the projected risk to the insurer. A policyholder can lower their premium by taking on more risk.

The main factors for auto insurance BI claims are Location, Age, Gender, Marital status, driving experience, driving record, Claims history, Credit history, Previous insurance coverage, Vehicle type, Vehicle use, Miles driven, Coverages and deductibles.

### 2.1. Main Drivers for AI Airline Passenger Satisfaction

Predictive modelling allows for simultaneous consideration of many variables and quantification of their overall effect. When a large number of claims are analysed, patterns regarding the characteristics of the claims that drive loss development begin to emerge.

The following are the main drivers which influencing the Claims Analytics:

<ul style="list-style-type: none"><li>• <b>Customer Data</b><ul style="list-style-type: none"><li>✓ Gender</li><li>✓ Customer Type</li><li>✓ Age</li></ul></li><li>• <b>Travel Details</b><ul style="list-style-type: none"><li>✓ Type of Travel</li><li>✓ Class</li><li>✓ Flight Distance</li></ul></li><li>• <b>Flight Services</b><ul style="list-style-type: none"><li>✓ Inflight Wifi Service</li><li>✓ Departure/Arrival</li></ul></li></ul>	<ul style="list-style-type: none"><li>• <b>Satisfaction Survey</b><ul style="list-style-type: none"><li>✓ Food and Drink</li><li>✓ Seat Comfort</li><li>✓ Inflight Entertainment</li><li>✓ On-board Service</li><li>✓ Leg room Service</li><li>✓ Baggage Handling</li><li>✓ Check-in Service</li><li>✓ Inflight Service</li><li>✓ Cleanliness</li></ul></li><li>• <b>Flight Times</b></li></ul>
--	---

Time Convinement <ul style="list-style-type: none"> <li>• <b>Booking details</b> <ul style="list-style-type: none"> <li>✓ Ease of Online Booking</li> <li>✓ Gate Location</li> <li>✓ Online Boarding</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>✓ Departure delay in Minutes</li> <li>✓ Arrival Delay in Minutes</li> </ul>
---	--

## 2.1. A

## 2.2. Airline Passenger Satisfaction Project - Data Link

The internship project data has taken from Kaggle and the link is: -

[www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction](https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction)

- Size:15.23 MB
  - Train.csv
  - Test.csv
- Number of Rows:25976
- Number of Columns:25

### 3.0 AI / ML Modelling and Results

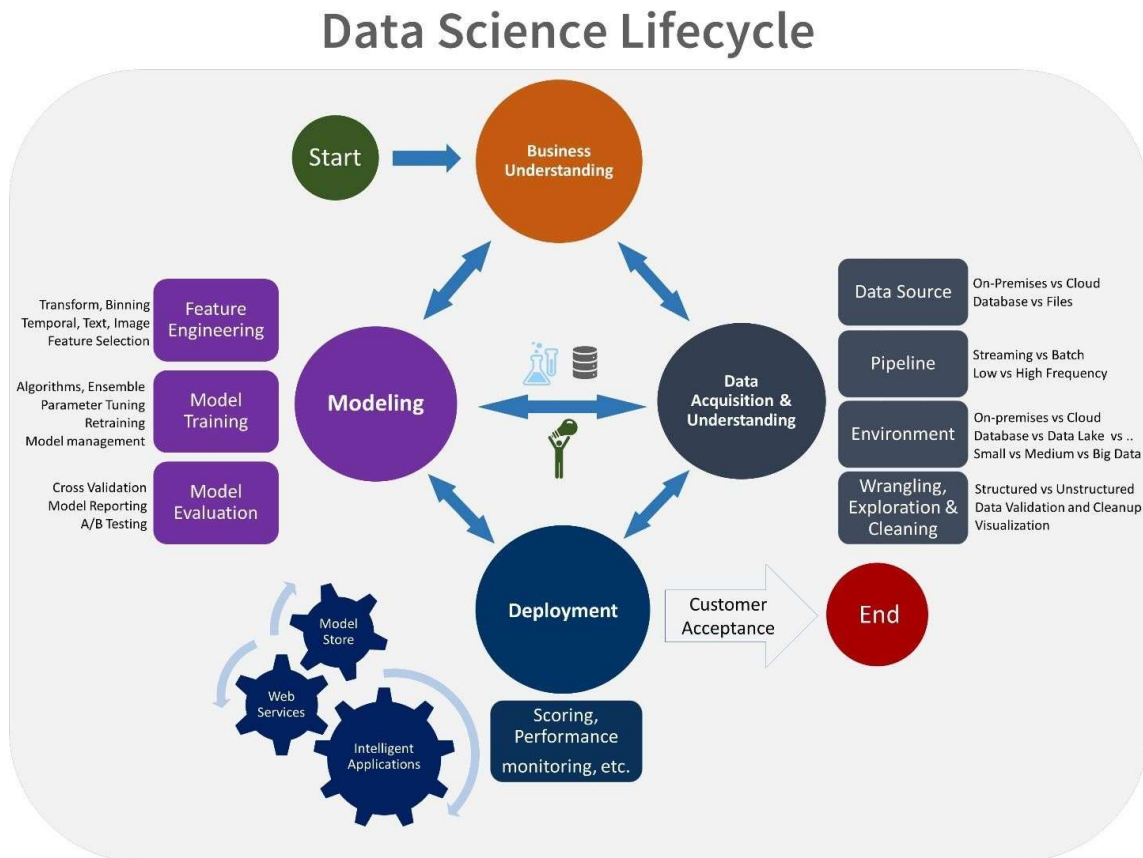
#### 3.1. Airline Passenger Satisfaction Problem Statement

In this study the aim is to identify the level of satisfaction of the passengers to know the quality of services provided by the airline companies, the key factors that derive customer satisfaction and to identify the ways how the airline industry can improve the service quality.

Focus on the Study of

- passengers' satisfaction in airline transport.
- The study examines which factors have a positive influence and which factors have
- negative influence on service quality of airline industry.

#### 3.2. Data Science Project Life Cycle



Data Science is a multidisciplinary field of study that combines programming skills, domain expertise and knowledge of statistics and mathematics to extract useful insights and knowledge from data.

### **3.2.1 Data Exploratory Analysis**

Exploratory data analysis has been done on the data to look for relationship and correlation between different variables and to understand how they impact or target variable.

The exploratory analysis is done for Auto Quote / Policy Conversion with different parameters and all the charts are presented in **Appendices 6.2 - List of charts (6.2.1 to 6.2.9)**

### **3.2.2 Data Pre-processing**

We removed variables which does not affect our target variable (Term Deposit) as they may add noise and also increase our computation time, we checked the data for anomalous data points and outliers. We did principal component analysis on the data set to filter out unnecessary variables and to select only the important variables which have greater correlation with our target variable.

#### **3.2.2.1. Check the Duplicate and low variation data**

Feature Selection is the process of reducing the number of input variables when developing a predictive model. It reduces the computational cost of model training and also improves the performance of the model.

These can be of two types:

1. Duplicate Values: When two features have the same set of values
  2. Duplicate Index: When the value of two features are different, but they occur at the same index
- ☐ Duplicate values -> Same Value for each record.
  - ☐ Duplicate Index -> Value of two Features are different but they occur at the same index.

#### **Variation data:**

In predictive analytics, we build machine learning models to make predictions on new, previously unseen samples. The whole purpose is to be able to predict the unknown. But the models cannot just make predictions out of the blue. We show



some samples to the model and train it. Then we expect the model to make predictions on samples from the same distribution.

There is no such thing as a perfect model so the model we build and train will have errors. There will be differences between the predictions and the actual values. The performance of a model is inversely proportional to the difference between the actual values and the predictions. The smaller the difference, the better the model. Our goal is to try to minimize the error. We cannot eliminate the error but we can reduce it. The part of the error that can be reduced has two components: Bias and Variance

The performance of a model depends on the balance between bias and variance.

Variance occurs when the model is highly sensitive to the changes in the independent variables (features). The model tries to pick every detail about the relationship between features and target. It even learns the noise in the data which might randomly occur. A very small change in a feature might change the prediction of the model. Thus, we end up with a model that captures each and every detail on the training set so the accuracy on the training set will be very high.

**Low variance:** tells you that the smallest change in the data set causes the results to change in the target function.

Examples of low variance in machine learning include linear regression, linear analysis, linear logic regression, and logistic regression.

### **3.2.2.2. Identify and address the missing variables**

The real-world data often has a lot of missing values. The cause of missing values can be data corruption or failure to record data. The handling of missing data is very important during the preprocessing of the dataset as many machine learning algorithms do not support missing values.

This article covers 7 ways to handle missing values in the dataset:

- Deleting Rows with missing values
- Impute missing values for continuous variable
- Impute missing values for categorical variable
- Other Imputation Methods

- Using Algorithms that support missing values
- Prediction of missing values
- Imputation using Deep Learning Library

### **3.2.2.3. Handling of Outliers**

A data point that varies greatly from other results is referred to as an outlier. An outlier may also be described as an observation in our data that is incorrect or abnormal as compared to other observations.

Outliers can be caused by measurement uncertainty or due to experimental error. Outliers in data can spoil and deceive the training process of machine learning models, resulting in less accurate models and eventually bad performance.

We can measure the boundary for outliers once we've decided whether outliers are present in the data using the box plot. To measure the boundary for outliers, we can use the two methods below, both based on data distribution:

#### I) If the Data is Normally Distributed:

We can use the empirical formula of Normal Distribution to determine the boundary for outliers if the data is normally distributed.

Lower Boundary = Mean — 3\* (Standard Deviation) Upper

Boundary= Mean + 3 \* (Standard Deviation)

#### II) If the Data is Either Right Skewed or Left Skewed:

We will use the Interquartile Range to measure the limits of Outliers if the data doesn't follow a Normal Distribution or is either right-skewed or left-skewed

### **3.2.2.4. Categorical data and Encoding Techniques**

Categorical variables are usually represented as 'strings' or 'categories' and are finite in number.

Ex: The city where a person lives: Delhi, Mumbai, Ahmedabad, Bangalore, etc LabelEncoder :

Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

```
# Import label encoder
```

```
from sklearn import preprocessing
```

```
# label_encoder object knows how to understand word labels.
```

```
label_encoder = preprocessing.LabelEncoder() LabelBinarizer :
```

Label Binarizer is an SciKit Learn class that accepts Categorical data as input and returns an Numpy array. Unlike Label Encoder, it encodes the data into dummy variables indicating the presence of a particular label or not. Encoding make column data using Label Binarizer.

```
# Enoding make column using LabelBinarizer from
```

```
sklearn.preprocessing import LabelBinarizer
```

```
labelbinarizer = LabelBinarizer()
```

### **3.2.2.5. Feature Scaling**

Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing to handle highly varying magnitudes or values or units. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.

#### **Working:**

Given a data-set with features- Age, Salary, BHK Apartment with the data size of 5000 people, each having these independent data features.

Each data point is labeled as:

Class1- YES (means with the given Age, Salary, BHK Apartment feature value one can buy the property)

Class2- NO (means with the given Age, Salary, BHK Apartment feature value one can't buy the property).

Techniques to perform Feature Scaling

Consider the two most important ones:

Min-Max Normalization: This technique re-scales a feature or observation value with distribution value between 0 and 1.

Standardization: It is a very effective technique which re-scales a feature value so that it has distribution with 0 mean value and variance equals to 1.

### **3.2.3 Selection of Dependent and Independent variables**

The dependent or target variable here is Claimed Target which tells us a particular policy holder has filed a claim or not the target variable is selected based on our business problem and what we are trying to predict.

The independent variables are selected after doing exploratory data analysis and we used Boruta to select which variables are most affecting our target variable.

### **3.2.4 Data Sampling Methods**

The data we have is highly unbalanced data so we used some sampling methods which are used to balance the target variable so our model will be developed with good accuracy and precision. We used three Sampling methods

#### **3.2.4.1. Stratified sampling**

Stratified sampling randomly selects data points from majority class so they will be equal to the data points in the minority class. So, after the sampling both the class will have same no of observations.

It can be performed using strata function from the library sampling.

#### **3.2.4.2. Simple random sampling**

Simple random sampling is a sampling technique where a set percentage of the data is selected randomly. It is generally done to reduce bias in the dataset which can occur if data is selected manually without randomizing the dataset.

We used this method to split the dataset into train dataset which contains 70% of the total data and test dataset with the remaining 30% of the data.

### **3.2.5 Models Used for Development**

We built our predictive models by using the following ten algorithms

#### **3.2.5.1. Model 01**

**Logistic** uses logit link function to convert the likelihood values to probabilities so we can get a good estimate on the probability of a particular observation to be positive class or negative class. It also gives us p-value of the variables which tells us about significance of each independent variable.

#### **3.2.5.2. Model 02**

**Random forest** is an algorithm that consists of many decision trees. It was first developed by Leo Breiman and Adele Cutler. The idea behind it is to build several trees, to have the instance classified by each tree, and to give a "vote" at each class. The model uses a "bagging" approach and the random selection of features to build a collection of decision trees with controlled variance. The instance's class is to the class with the highest number of votes, the class that occurs the most within the leaf in which the instance is placed.

The error of the forest depends on:

- Trees correlation: the higher the correlation, the higher the forest error rate.
- The strength of each tree in the forest. A strong tree is a tree with low error. By using trees that classify the instances with low error the error rate of the forest decreases.

#### **3.2.5.3. Model 03**

**Artificial neural networks** can theoretically solve any problem. ANNs can identify hidden patterns between the variables and can find how different combinations of variables can affect the target variable. The error correction is done by gradient descent algorithm which can reduce the error rate as much as possible for the given data.

#### **3.2.5.4. Model 04**

**Extra Trees** is an ensemble machine learning algorithm that combines the predictions from many decision tree. It is related to the widely used random forest algorithm. It can often achieve as-good or better performance than the random forest algorithm, although it uses a simpler algorithm to construct the decision trees used as members of the ensemble. It is also easy to use given that it has few key hyperparameters and sensible heuristics for configuring these hyperparameters. In this tutorial, you will discover how to develop Extra Trees ensembles for classification and regression.

#### **3.2.5.5. Model 05**

**K-Nearest Neighbour** is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.

#### **3.2.5.6. Model 06**

**Naïve Bayes** is a probabilistic machine learning algorithm used for many classification functions and is based on the Bayes theorem. Gaussian Naïve Bayes is the extension of naïve Bayes. While other functions are used to estimate data distribution, Gaussian or normal distribution is the simplest to implement as you will need to calculate the mean and standard deviation for the training data. Naïve Bayes is a probabilistic machine learning algorithm that can be used in several classification tasks. Typical applications of Naïve Bayes are classification of documents, filtering spam, prediction and so on. This algorithm is based on the discoveries of Thomas Bayes and hence its name.

The name “Naïve” is used because the algorithm incorporates features in its model that are independent of each other. Any modifications in the value of one feature do not directly impact the value of any other feature of the algorithm. The main advantage of the Naïve Bayes algorithm is that it is a simple yet powerful algorithm.

It is based on the probabilistic model where the algorithm can be coded easily, and predictions did quickly in real-time. Hence this algorithm is the typical choice to solve real-world problems as it can be tuned to respond to user requests instantly. But before we dive deep into Naïve Bayes and Gaussian Naïve Bayes, we must know what is meant by conditional probability.

### **3.2.5.7. Model 07**

**Support vector machines (SVMs)** are powerful yet flexible supervised machine learning methods used for classification, regression, and, outliers' detection. SVMs are very efficient in high dimensional spaces and generally are used in classification problems. SVMs are popular and memory efficient because they use a subset of training points in the decision function. The main goal of SVMs is to divide the datasets into number of classes in order to find a maximum marginal hyperplane (MMH) which can be done in the following two steps

Support Vector Machines will first generate hyperplanes iteratively that separates the classes in the best way.

After that it will choose the hyperplane that segregate the classes correctly.

The objective of a Linear SVC (Support Vector Classifier) is to fit to the data you provide, returning a "best fit" hyperplane that divides, or categorizes, your data. From there, after getting the hyperplane, you can then feed some features to your classifier to see what the "predicted" class is. This makes this specific algorithm rather suitable for our uses, though you can use this for many situations

### **3.2.5.8. Model 08**

**Bagging classifier** is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. Such a meta-estimator can typically be used as a way to reduce the variance of a black-box estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble out of it.

Each base classifier is trained in parallel with a training set which is generated by randomly drawing, with replacement, N examples(or data) from the original training dataset — where N is the size of the original training set. Training set for each of the

base classifiers is independent of each other. Many of the original data may be repeated in the resulting training set while others may be left out.

### **3.2.5.9. Model 09**

**Gradient Boosting** is a popular boosting algorithm. In gradient boosting, each predictor corrects its predecessor's error. In contrast to Adaboost, the weights of the training instances are not tweaked, instead, each predictor is trained using the residual errors of predecessor as labels. There is a technique called the Gradient Boosted Trees whose base learner is CART (Classification and Regression Trees).

The below diagram explains how gradient boosted trees are trained for regression problems.

The ensemble consists of  $N$  trees. Tree1 is trained using the feature matrix  $X$  and residual errors  $r_1$ . Tree2 is then trained using the feature matrix  $X$  and the residual errors  $r_1$  of Tree1 as labels. The predicted results  $\hat{r}_1$  are then used to determine the residual  $r_2$ . The process is repeated until all the  $N$  trees forming the ensemble are trained.

### **3.2.5.10. Model 10**

**LightGBM** is a gradient boosting framework based on decision trees to increase the efficiency of the model and reduce memory usage. It uses two novel techniques: Gradient-based One Side Sampling and Exclusive Feature Bundling (EFB) which fulfill the limitations of histogram-based algorithm that is primarily used in all GBDT (Gradient Boosting Decision Tree) frameworks. The two techniques of GOSS and EFB described below form the characteristics of LightGBM Algorithm. They comprise together to make the model work efficiently and provide it a cutting edge over other GBDT frameworks

Gradient-based One Side Sampling Technique for LightGBM:

Different data instances have varied roles in the computation of information gain. The instances with larger gradients (i.e., under-trained instances) will contribute more to the information gain. GOSS keeps those instances with large gradients (e.g., larger than a predefined threshold, or among the top percentiles), and only randomly drop those instances with small gradients to retain the accuracy of information gain estimation. This treatment can lead to a more accurate gain



estimation than uniformly random sampling, with the same target sampling rate, especially when the value of information gain has a large range.

### **3.3. AI / ML Models Analysis and Final Results**

We used our train dataset to build the above models and used our test data to check the accuracy and performance of our models.

We used confusion matrix to check accuracy, Precision, Recall and F1 score of our models and compare and select the best model for given auto dataset of size  $\sim 272252$  policies.

#### **3.3.1 Different Model codes**

```
# Build the Classification models and compare the results from

sklearn.linear_model import LogisticRegression from
sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier from
sklearn.ensemble import ExtraTreesClassifier from
sklearn.neighbors import KNeighborsClassifier from
sklearn.naive_bayes import GaussianNB

from sklearn.svm import SVC

from sklearn.ensemble import BaggingClassifier

from sklearn.ensemble import GradientBoostingClassifier import
lightgbm as lgb

# Create objects of classification algorithm with default hyper-parameters ModelLR =
LogisticRegression()

ModelDC = DecisionTreeClassifier()

ModelRF = RandomForestClassifier()

ModelET = ExtraTreesClassifier() ModelSVM
= SVC(probability=True)
```

```
modelBAG=BaggingClassifier(base_estimator=None,n_estimators=100, max_samples=1.0,
max_features=1.0,bootstrap=True, bootstrap_features=False, oob_score=False,
warm_start=False,n_jobs=None, random_state=None, verbose=0)
```

```
ModelGB = GradientBoostingClassifier(loss='deviance', learning_rate=0.1,
n_estimators=100, subsample=1.0, criterion='friedman_mse', min_samples_split=2,
min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_depth=3,
min_impurity_decrease=0.0, init=None, random_state=None, max_features=None,
verbose=0, max_leaf_nodes=None, warm_start=False, validation_fraction=0.1,
n_iter_no_change=None, tol=0.0001, ccp_alpha=0.0)
```

```
ModelLGB = lgb.LGBMClassifier()
```

```
ModelGNB = GaussianNB()
```

```
ModelKNN = KNeighborsClassifier(n_neighbors=5) #
```

Evaluation matrix for all the algorithms

```
MM = [ModelLR, ModelDC, ModelRF, ModelET, ModelKNN, modelBAG, ModelGB,
ModelLGB, ModelGNB]
```

for models in MM: #

Fit the model

```
models.fit(x_train, y_train) #
```

Prediction

```
y_pred = models.predict(x_test) y_pred_prob =
```

```
models.predict_proba(x_test) # Print the model
```

name

```
print('Model Name: ', models) #
```

confusion matrix in sklearn

```
from sklearn.metrics import confusion_matrix from
```

```
sklearn.metrics import classification_report # actual
```

values

```

actual = y_test

# predicted values

predicted = y_pred #

confusion matrix

matrix = confusion_matrix(actual,predicted, labels=[1,0],sample_weight=None,
normalize=None)

print('Confusion matrix : \n', matrix) #

outcome values order in sklearn

tp, fn, fp, tn = confusion_matrix(actual,predicted,labels=[1,0]).reshape(-1) print('Outcome
values : \n', tp, fn, fp, tn)

# classification report for precision, recall f1-score and accuracy

C_Report = classification_report(actual,predicted,labels=[1,0])

print('Classification report : \n', C_Report)

# calculating the metrics sensitivity =

round(tp/(tp+fn), 3); specificity =

round(tn/(tn+fp), 3);

accuracy = round((tp+tn)/(tp+fp+tn+fn), 3); balanced_accuracy =

round((sensitivity+specificity)/2, 3); precision =

round(tp/(tp+fp), 3);

f1Score = round((2*tp/(2*tp + fp + fn)), 3);

# Matthews Correlation Coefficient (MCC). Range of values of MCC lie between
-1 to +1.

# A model with a score of +1 is a perfect model and -1 is a poor model from math

import sqrt

mx = (tp+fp) * (tp+fn) * (tn+fp) * (tn+fn)

MCC = round((((tp * tn) - (fp * fn)) / sqrt(mx), 3)

```

```

print('Accuracy :', round(accuracy*100, 2),'%')
print('Precision :', round(precision*100, 2),'%')
print('Recall :', round(sensitivity*100,2), '%')
print('F1 Score :', f1Score)
print('Specificity or True Negative Rate :', round(specificity*100,2), '%')
print('Balanced Accuracy :', round(balanced_accuracy*100, 2),'%') print('MCC :',
MCC)
# Area under ROC curve
from sklearn.metrics import roc_curve, roc_auc_score
print('roc_auc_score:', round(roc_auc_score(actual, predicted), 3)) #
ROC Curve
from sklearn.metrics import roc_auc_score from
sklearn.metrics import roc_curve
logit_roc_auc = roc_auc_score(actual, predicted)
fpr, tpr, thresholds = roc_curve(actual, models.predict_proba(x_test)[:,-1]) plt.figure()
# plt.plot(fpr, tpr, label='Logistic Regression (area = %0.2f)' % logit_roc_auc) plt.plot(fpr, tpr,
label= 'Classification Model' % logit_roc_auc)
plt.plot([0, 1], [0, 1], 'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05]) plt.xlabel('False
Positive Rate')plt.ylabel('True
Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.savefig('Log_ROC')

```

```

plt.show()

print('_____')

#_____ -

new_row = {'Model Name' : models, 'True
          Positive' : tp,
          'False Negative' : fn,
          'False Positive' : fp,
          'True Negative' : tn,
          'Accuracy' : accuracy,
          'Precision' : precision,
          'Recall' : sensitivity, 'F1
          Score' : f1Score,
          'Specificity' : specificity,
          'MCC':MCC,
          'ROC_AUC_Score':roc_auc_score(actual, predicted),
          'Balanced Accuracy':balanced_accuracy}

EMResults = EMResults.append(new_row, ignore_index=True)

#_____

```

### 3.3.2 Random Forest Python Code

# To build the 'Random Forest' model with random sampling from

```
sklearn.ensemble import RandomForestClassifier
```

```

ModelRF = RandomForestClassifier(n_estimators=100, criterion='gini', max_depth=None,
min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0,
max_features='sqrt', max_leaf_nodes=None, min_impurity_decrease=0.0, bootstrap=True,
oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False,
class_weight=None, ccp_alpha=0.0, max_samples=None

```

### 3.3.3 Extra Trees Python code

```
from sklearn.ensemble import ExtraTreesClassifier

ModelET = ExtraTreesClassifier(n_estimators=100, criterion='gini', max_depth=None,
min_samples_split=2,min_samples_leaf=1,min_weight_fraction_leaf=0.0,
max_features='sqrt',max_leaf_nodes=None,min_impurity_decrease=0.0, bootstrap=False,
oob_score=False,n_jobs=None,random_state=None, verbose=0, warm_start=False,
class_weight=None,ccp_alpha=0.0, max_samples=None)

# Train the model with train data

ModelET.fit(x_train,y_train)

# Predict the model with test data set y_pred =

ModelET.predict(x_test)

y_pred_prob = ModelET.predict_proba(x_test)
```

**Stratified Sampling:** Random Forest model performance is good, by considering the confused matrix, highest accuracy (1.0) & good F1 score (1.0). This is because random forest uses bootstrap aggregation which can reduce bias and variance in the data and can lead to good predictions with claims dataset.

**Simple Random Sampling:** Artificial Neural Networks / Random Forest are outperformed by the Logistic Regression model, by considering the confused matrix, highest accuracy (1.0) & good F1 score (1.0). This is because Artificial Neural Networks have hidden and complex patterns between different variables and can lead to good predictions with claims dataset.

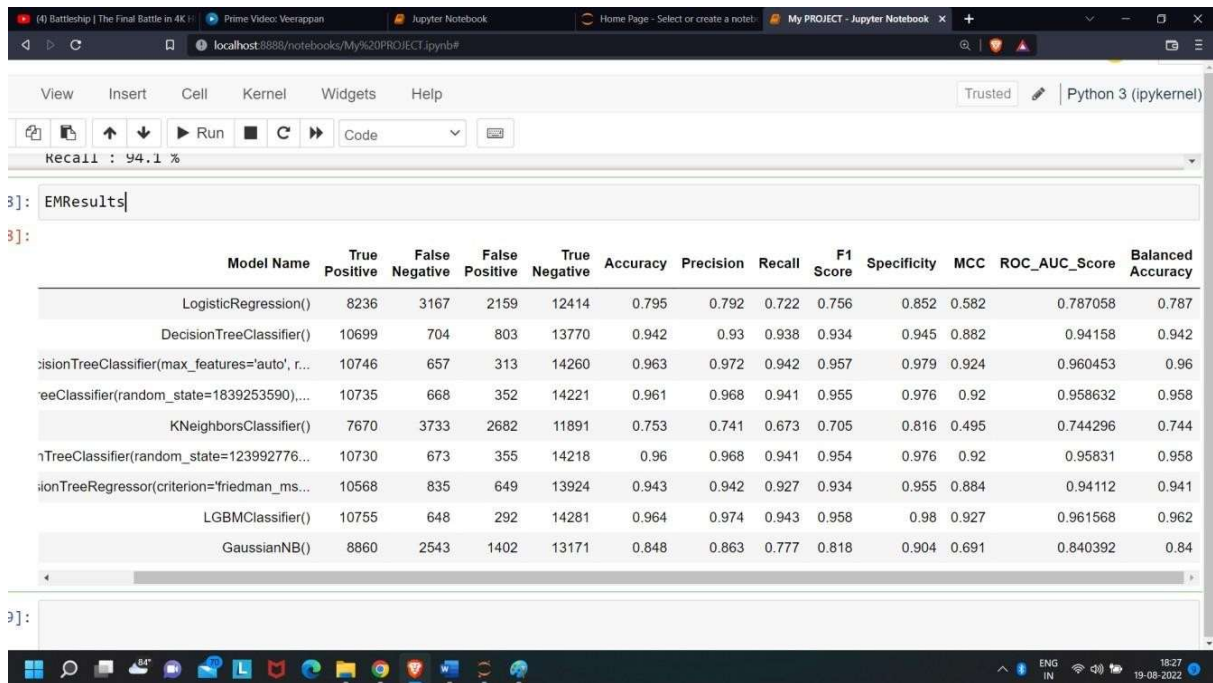
## 4.0 Conclusions and Future work

The model results in the following order by considering the model accuracy, F1 score and RoC AUC score.

- 1) **LGBM Classifier** with Stratified and Random Sampling
- 2) **Decision Tree Classifier** with Simple Random Sampling
- 3) **Logistic Regression** with Simple Random Sampling

We recommend model - **Random Forest** with Stratified and Random Sampling technique as a best fit for the give n BI claims dataset. We considered Random Forest because it uses bootstrap aggregation which can reduce bias and variance in the data and can leads to good predictions with claims dataset.

The future work to evaluate the “Other Airline Passenger Satisfaction factors” in Airline Passenger Satisfaction by using classification methods.



Recall : 94.1 %

```
3]: EMResults
```

```
3]:
```

Model Name	True Positive	False Negative	False Positive	True Negative	Accuracy	Precision	Recall	F1 Score	Specificity	MCC	ROC_AUC_Score	Balanced Accuracy
LogisticRegression()	8236	3167	2159	12414	0.795	0.792	0.722	0.756	0.852	0.582	0.787058	0.787
DecisionTreeClassifier()	10699	704	803	13770	0.942	0.93	0.938	0.934	0.945	0.882	0.94158	0.942
DecisionTreeClassifier(max_features='auto', r...	10746	657	313	14260	0.963	0.972	0.942	0.957	0.979	0.924	0.960453	0.96
DecisionTreeClassifier(random_state=1839253590),...	10735	668	352	14221	0.961	0.968	0.941	0.955	0.976	0.92	0.958632	0.958
KNeighborsClassifier()	7670	3733	2682	11891	0.753	0.741	0.673	0.705	0.816	0.495	0.744296	0.744
DecisionTreeClassifier(random_state=123992776...)	10730	673	355	14218	0.96	0.968	0.941	0.954	0.976	0.92	0.95831	0.958
DecisionTreeRegressor(criterion='friedman_ms...)	10568	835	649	13924	0.943	0.942	0.927	0.934	0.955	0.884	0.94112	0.941
LGBMClassifier()	10755	648	292	14281	0.964	0.974	0.943	0.958	0.98	0.927	0.961568	0.962
GaussianNB()	8860	2543	1402	13171	0.848	0.863	0.777	0.818	0.904	0.691	0.840392	0.84

```
3]:
```

## LGBM Classifier()

Model Name: LGBMClassifier()

Confusion matrix :

```
[[10755  648]
 [ 292 14281]]
```

Outcome values :

10755 648 292 14281

Classification report :

	precision	recall	f1-score	support
1	0.97	0.94	0.96	11403
0	0.96	0.98	0.97	14573
accuracy			0.96	25976
macro avg	0.97	0.96	0.96	25976
weighted avg	0.96	0.96	0.96	25976

Accuracy : 96.4 %

Precision : 97.4 %

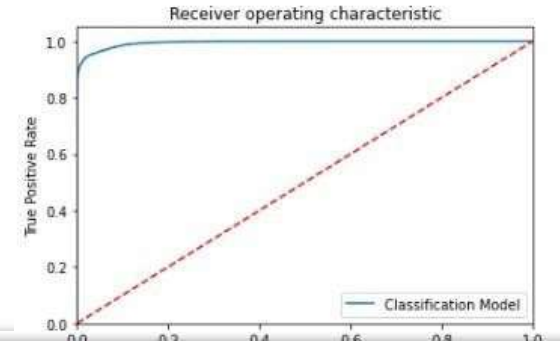
Recall : 94.3 %

Specificity or True Negative Rate : 98.0 %

Balanced Accuracy : 96.2 %

MCC : 0.927

roc\_auc\_score: 0.962



## Random Forest Classifier()

Model Name: RandomForestClassifier()

Confusion matrix :

```
[[10746  657]
 [ 313 14260]]
```

Outcome values :

10746 657 313 14260

Classification report :

	precision	recall	f1-score	support
1	0.97	0.94	0.96	11403
0	0.96	0.98	0.97	14573
accuracy			0.96	25976
macro avg	0.96	0.96	0.96	25976
weighted avg	0.96	0.96	0.96	25976

Accuracy : 96.3 %

Precision : 97.2 %

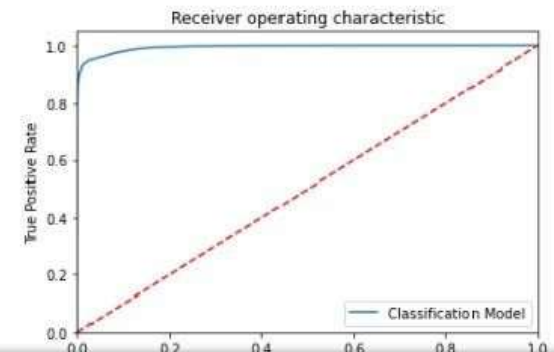
Recall : 94.2 %

Specificity or True Negative Rate : 97.9 %

Balanced Accuracy : 96.0 %

MCC : 0.924

roc\_auc\_score: 0.96



## Extra Tree Classifier()

Model Name: ExtraTreesClassifier()

Confusion matrix :

```
[[10735  668]
 [ 352 14221]]
```

Outcome values :

10735 668 352 14221

Classification report :

	precision	recall	f1-score	support
1	0.97	0.94	0.95	11403
0	0.96	0.98	0.97	14573
accuracy			0.96	25976
macro avg	0.96	0.96	0.96	25976
weighted avg	0.96	0.96	0.96	25976

Accuracy : 96.1 %

Precision : 96.8 %

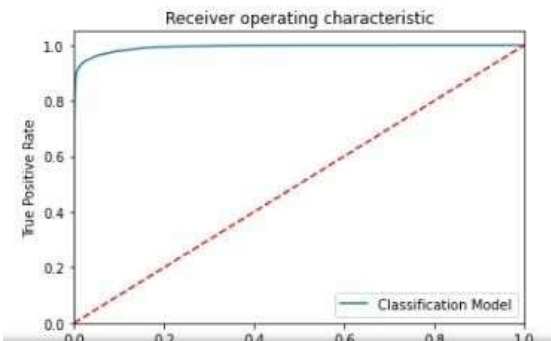
Recall : 94.1 %

Specificity or True Negative Rate : 97.6 %

Balanced Accuracy : 95.8 %

MCC : 0.92

roc\_auc\_score: 0.959





## 5.0 References

5.0.1 [www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction](https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction)

5.0.2 [www.github.com/Maggie0927/AirlinePassengerSatisfaction](https://www.github.com/Maggie0927/AirlinePassengerSatisfaction)

5.0.3 [www.geeksforgeeks.org](https://www.geeksforgeeks.org)

5.0.4 [www.tutorialpoints.com](https://www.tutorialpoints.com)

5.0.5 [www.en.wikipedia.org/wiki/Airline](https://www.en.wikipedia.org/wiki/Airline)

5.0.6 [www.business-essay.com/](https://www.business-essay.com/)

## 6.0 Appendices

### 6.1. Python code Results

#### 1. Train dataset

Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking	Gate location	Food and drink	Online boarding	Seat comfort	
0	0	70172	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	4	3	1	5	3	5
1	1	5047	Male	disloyal Customer	25	Business travel	Business	235	3	2	3	3	1	3	1
2	2	110028	Female	Loyal Customer	26	Business travel	Business	1142	2	2	2	2	5	5	5
3	3	24026	Female	Loyal Customer	25	Business travel	Business	562	2	5	5	5	2	2	2
4	4	119299	Male	Loyal Customer	61	Business travel	Business	214	3	3	3	3	4	5	5

#### 2. Test dataset

Inflight entertainment	On-board service	Leg room service	Baggage handling	Checkin service	Inflight service	Cleanliness	Departure Delay in Minutes	Arrival Delay in Minutes	satisfaction
5	4	3	4	4	5	5	25	18.0	neutral or dissatisfied
1	1	5	3	1	4	1	1	6.0	neutral or dissatisfied
5	4	3	4	4	4	5	0	0.0	satisfied
2	2	5	3	1	4	2	11	9.0	neutral or dissatisfied
3	3	4	4	3	3	3	0	0.0	satisfied

### 3. Train info

```
<class 'pandas.core.frame.DataFrame'>
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
```

### 4. Train isnull()

```
Unnamed: 0      0
id              0
Gender          0
Customer Type   0
Age            0
Type of Travel  0
Class          0
Flight Distance 0
Inflight wifi service
Departure/Arrival time convenient
Ease of Online booking
Gate location   0
Food and drink  0
Online boarding 0
Seat comfort    0
Inflight entertainment
On-board service
Leg room service
Baggage handling
Checkin service
Inflight service
Cleanliness     0
Departure Delay in Minutes
Arrival Delay in Minutes
satisfaction     0
dtype: int64
```

### 5. Value Counts

```
0    52727
1    51177
Name: Gender, dtype: int64    84923
1    18981
Name: Customer Type, dtype: int64    71655
1    32249
Name: Type of Travel, dtype: int64    49665
1    46745
2     7494
Name: Class, dtype: int64
58879
1    45025
Name: satisfaction, dtype: int64
class 0: 14573
class 1: 11403
class 0: class 1= 1.27799701832851

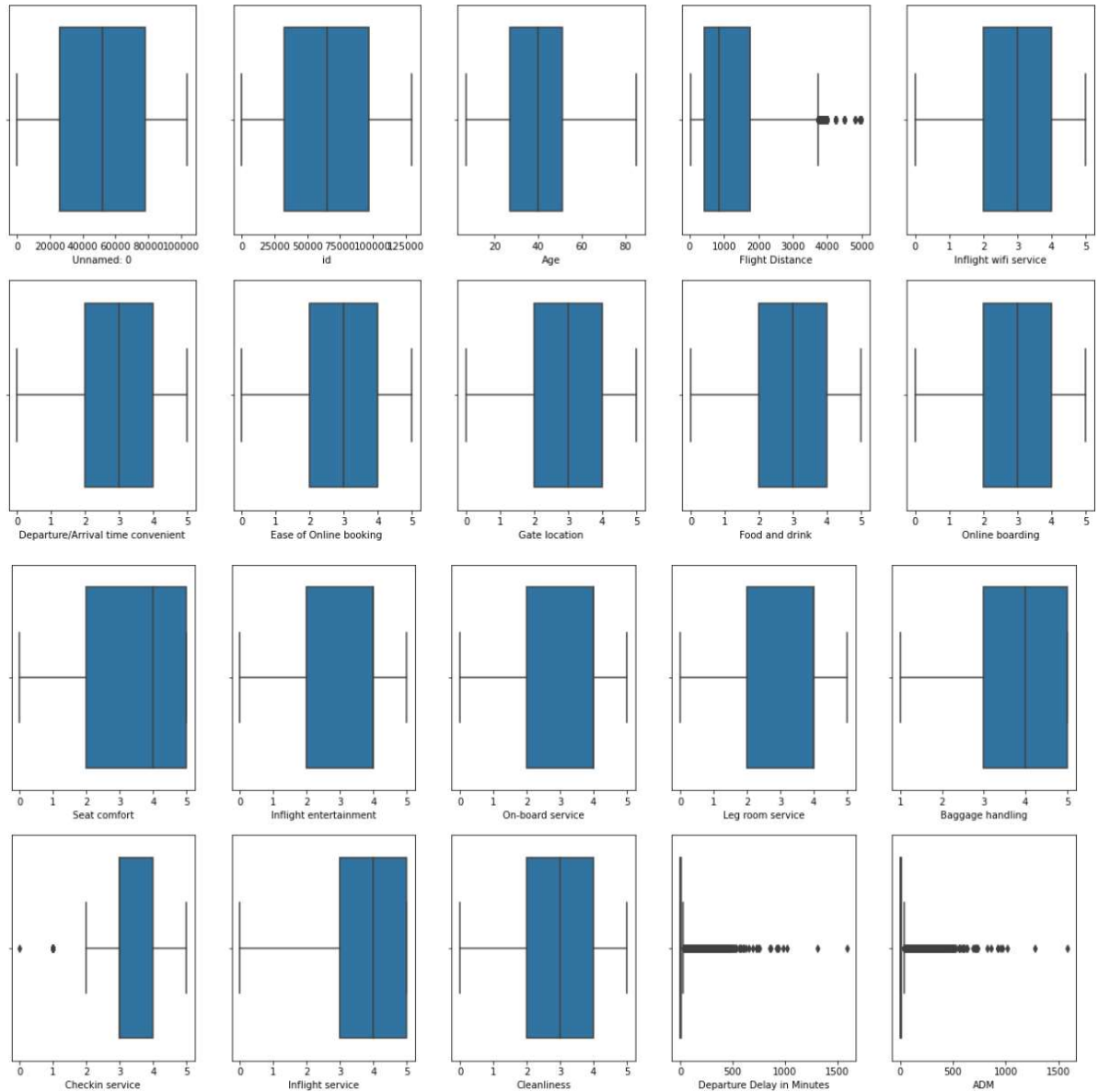
array([0, 0, 1, ..., 0, 0, 0])
```

### 6. Final Predicted Values

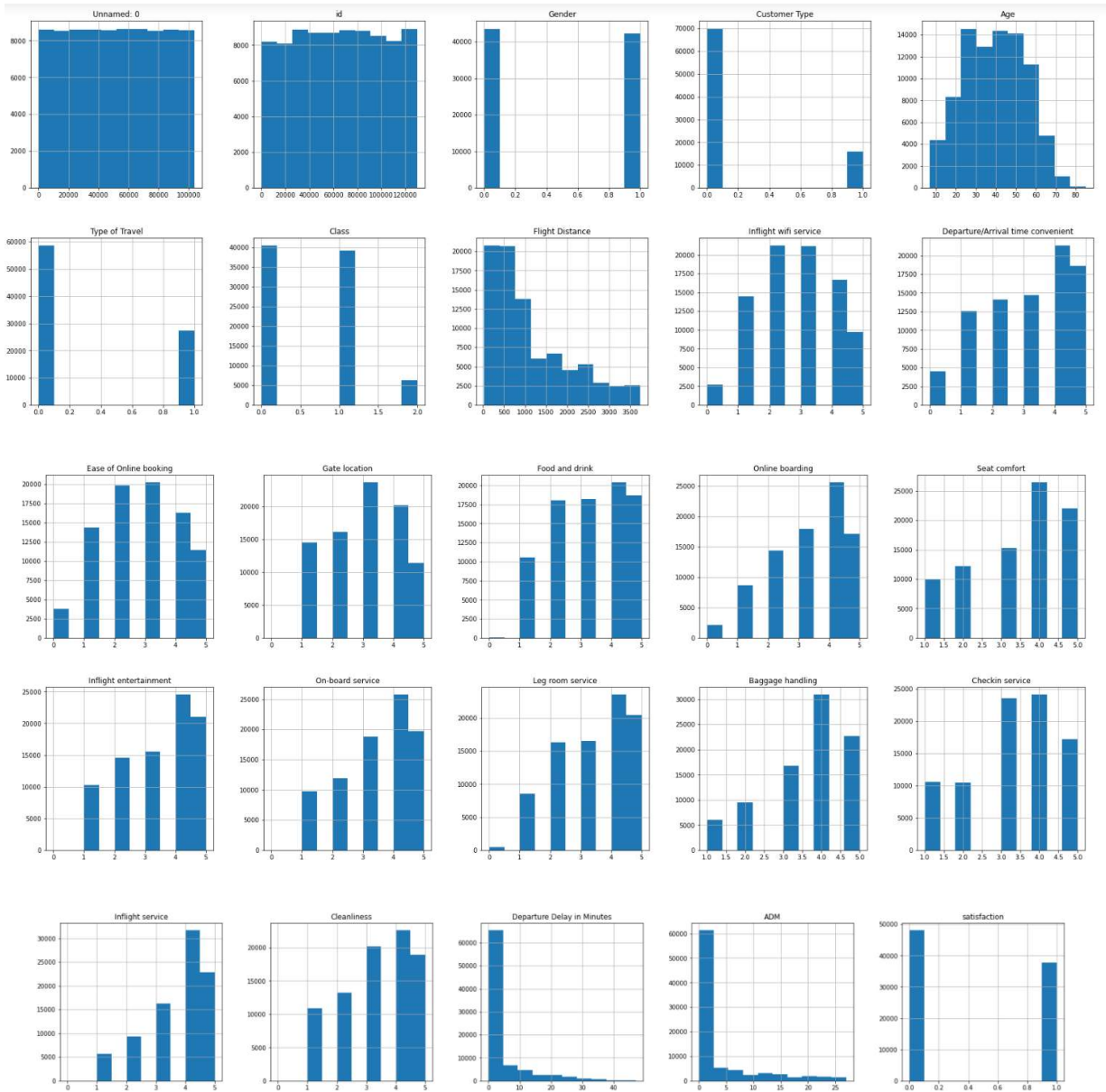
	id	Actual_Satisfaction	Pred_Satisfaction
0	70172	0	0
1	5047	0	0
2	110028	1	1
3	24026	0	0
4	119299	1	1
5	111157	0	0
6	82113	0	0
7	96462	1	1
8	79485	0	0
9	65725	0	0

## 6.2. List of Charts

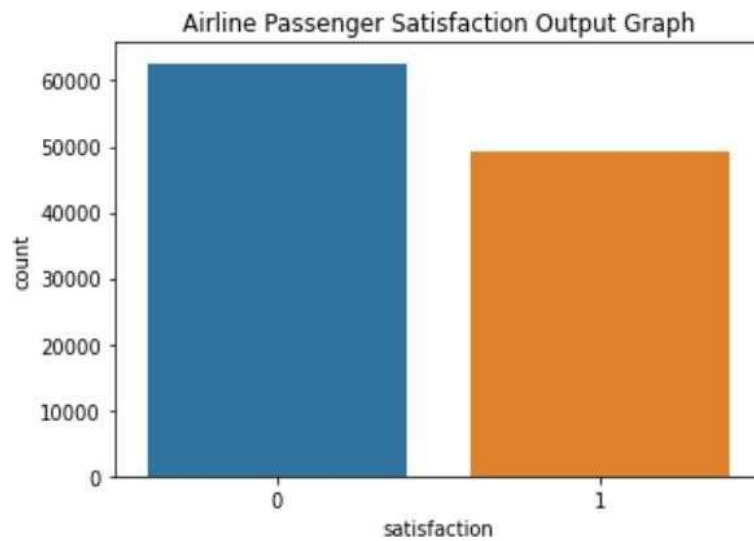
### 6.2.1 Chart 01: Visualization of Outliers



## 6.2.2 Chart 02: Histograms



### 6.2.3 Chart 03: Airline Passenger Satisfaction Graph



### 6.2.4 Chart 04: Flight Satisfaction Pie-Chart

