Table of Contents

[Contribution 3](#_Toc154226762)

[Special terms in this report 4](#_Toc154226763)

[Abstract 5](#_Toc154226764)

[1. Introduction 6](#_Toc154226765)

[2. Understand the task 7](#_Toc154226766)

[3. Describe the “experience” to perform the task 8](#_Toc154226767)

[4. Implement machine learning algorithm 9](#_Toc154226768)

[5. Evaluate the performance of the trained model 10](#_Toc154226769)

[6. Conclusion 11](#_Toc154226770)

[References 12](#_Toc154226771)

# Contribution

|  |  |  |
| --- | --- | --- |
| Name | Mission | Contribution Percentage |
| Nguyen Van A | Conduct research, experimentation, and implement the Multinomial Naive Bayes model | 33.33% |
| Le Van B | Conduct research, experimentation, and implement the Support Vector Machine model | 33.33% |
| Tran Van C | Data preprocessing, extracting relevant features that significantly influence the predictions of models, and standardizing the data, research accuracy metrics to evaluate models’ results | 33.33% |

# Special terms in this report

|  |  |
| --- | --- |
| SVM | Support Vector Machine |
| MultinomialNB | Multinomial Naive Bayes |
| VNA | Vietnam Airlines |

# Abstract

The “VNA Customers Satisfaction” project employs machine learning to classify Vietnam Airlines (VNA) customers based on satisfaction, transcending traditional metrics. Its primary goal is to offer actionable insights into the nuanced factors shaping passenger experiences, Utilizing advanced algorithms and data-driven approach, the project equips VNA with a dynamic tool for optimizing services and enhancing customer engagement. This intiniative not only offer VNA’s commitment to customer-centric stragetries but also extends its influence to contribute vlauable insights to the broader aviation industry. By leveraging technological innovation, the project aligns VNA with the evolving landscape of customer satisfaction strategies, affirming its position as a proactive player in the aviation sector. As VNA strives for operational excellence, the “VNA Customers Satisfaction” project stands as a testatement to the transformative potential of machine learning in shaping the future of customer experiences within the airline industry.

# Introduction

In the dynamic landscape of the aviation industry, customer satisfaction stands as a pivotal factor influencing the success and longevity of airline companies. In light of this, the “VNA Customers Satisfaction” project endeavors to harness the capabilities of machine leanrning to cultivate a deeper understanding of customer sentiments and, ultimately, build a robust model for classifying the satisfaction levels of Vietnam Airlines customers.

The overarching objective of this project is to leverage the wealth of customer data available to VNA, employing advanced machine learning techniques to develop a predictive model that accurately classifies customer satisfaction. By systematically analyzing historical data, indentifying patterns, and discerning influential factors, our aim is to empower VNA with a powerful tool that not only categorizes customers based on their satisfaction but also provides actionable insights for enhancing service quality and addressing specific customer needs.

Through the stages of data collection, preprocessing, feature engineering, and model development, this project seeks to unravel the complexities inherent in customer satisfaction within the context of the airline industry. As we delve into the intricacies of VNA’s customer interactions, the machine learning model will serve as a strategic asset, aiding the airline in tailoring services, anticipating customer expectations, and fostering an environment of continuous improvement.

By embarking on the “VNA Customers Satisfaction” project, we align with VNA’s commitment to delivering an exceptional customer experience. The insights garnered from the machine learning model are posed not only to optimize internal operations but also to solidify VNA’s reputation as customer-centric airline. This project encapsulates the synergy between data science and customer-centric strategies, highlighting the transformative potential of machine-learning in shaping the future of customer satisfaction within the airline industry.

# Understand the task

Source of the dataset: <https://www.kaggle.com/competitions/cityu9e-funai-finalproject/overview>

Supervised learning: Use labeled datasets to train or “supervise” algorithms in classifying data or predicting outcomes.

* Type of problems: Classification.
* Purpose: The task is to build a machine learning model that based on the VNA customer’s features, which will classify customer’s satisfactions as “neutral or dissatisfied” or “satisfied”.
* Input: The input dataset consists of 23 variables. Within that, we have selected inputs data for analysis, which include: Class, On-board service, Ease of Online booking, Gender, Customer Type, Age, Type of Travel, Inflight wifi service, Departure/Arrival time convenient, Food and drink, Online boarding, Seat comfort, Inflight entertainment, Leg room service, Baggage handling, Checkin service, Inflight service, Cleanliness, Gate location.
* Output: Classification of customers’ satisfactions as “neutral or dissatisfied” (0) or satisfied (1).
* Reason: This problem belong to the type of binnary classification problem because the features of the input data are divided into a finite number of groups, and the output required the data type as boolean, with 1 means “satisfied” and 0 means “neutral or dissatisfied”.
* Possible classes:
  + 1 means “satisfied”
  + 0 means “neutral or dissatisfied”

# Analyze the data

The data contains total 85,000 survey data records, splitted into 2 datasets: the train dataset contains 70,000 records, the test dataset contains 15,000 records. Each record consists of 24 attributes:

* **id:** A unique identifier for each entry, typically used for indexing and referencing specific records.
* **Gender:** Indicates the gender of the passenger, with options such as “Male” or “Female”.
* **Customer Type:** Describes whether the customer is a “Loyal Customer” or a “disloyal Customer”, providing insights into customer loyalty.
* **Age:** Represents the age of the passengers, providing demographic information.
* **Type of Travel:** Specifies the purpose of the travel, such as “Personal Travel” or “Business travel”.
* **Class:** Indicates the class of travel, with options like “Eco Plus”, “Business” and others.
* **Flight Distance:** Represents the distance of the flights in miles, providing information about the journey’s scale.
* **Inflight wifi service:** Rates the satisfaction with the inflight wifi service on a scale, where higher values indicate greater satisfaction, scaled from 0 to 5.
* **Departure/Arrival time convenient:** Rates the satisfaction with the convenience of departure and arrival times, scaled from 0 to 5.
* **Ease of Online booking:** Measures the ease of the online booking process, indicating user-friendliness, scaled from 0 to 5.
* **Gate location:** Evaluates the satisfaction with the gate location at the airport, scaled from 0 to 5.
* **Food and drink:** Rates the satisfaction with the onboard food and drink services, scaled from 0 to 5.
* **Online boarding:** Measures the satisfaction with the online boarding process, scaled from 0 to 5.
* **Seat comfort:** Rates the satisfaction with the comfort of the seats on the flight, scaled from 0 to 5.
* **Inflight entertainment:** Evaluates the satisfaction with the inflight entertainment options, rated from 0 to 5.
* **On-board service:** Rates the satisfaction with the inflight entertainment options, scaled from 0 to 5.
* **Leg room service:** Measures the satisfaction with the legroom space available, rated from 0 to 5.
* **Baggage handling:** Rates the satisfaction with the handling, scaled from 0 to 5.
* **Checkin service:** Measures the satisfaction with the check-in process, rated from 0 to 5.
* **Inflight service:** Rates the satisfaction with the overall inflight service, scaled from 0 to 5.
* **Cleanliness:** Evaluates the satisfaction with the cleanliness of the aircraft, rated from 0 to 5.
* **Departure delay in Minutes:** Specifies the duration of departure delay in minutes.
* **Arrival Delay in Minutes:** Specifies the duration of arrival delay in minutes.
* **satisfaction:** Indicates the overall satisfactions level of the passenger, with values like “satisfied” or “neutral or dissatisfied”.

# Implement machine learning algorithm

In our project, various machine learning algorithms, including Multinomial Naive Bayes and Support Vector Machine, were employed to address tasks on the dataset. The specific reasons and key considerations guiding the selection of these algorithms will be provided below.

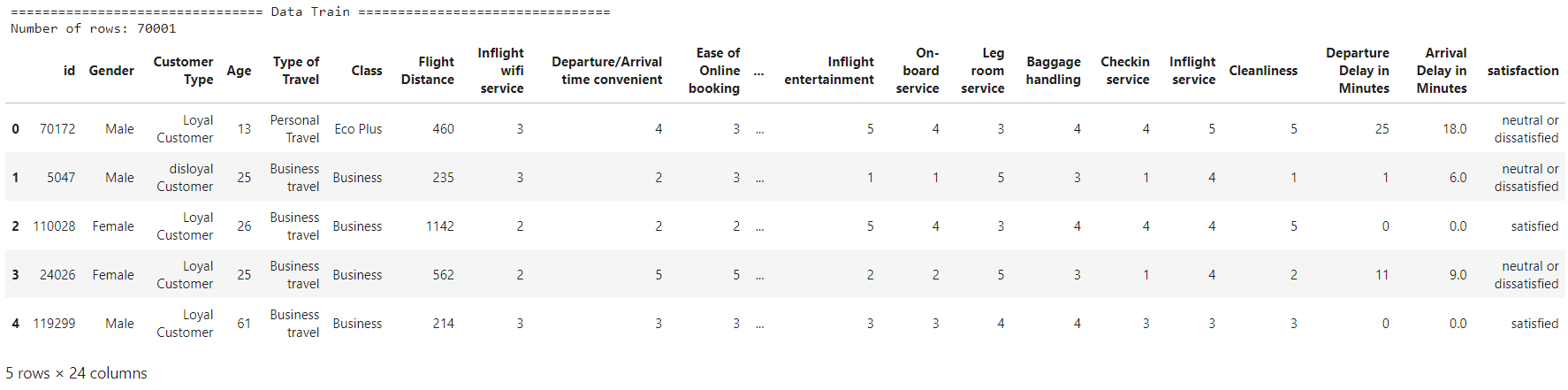
Firstly, we import all nescessary libraries used for this project:

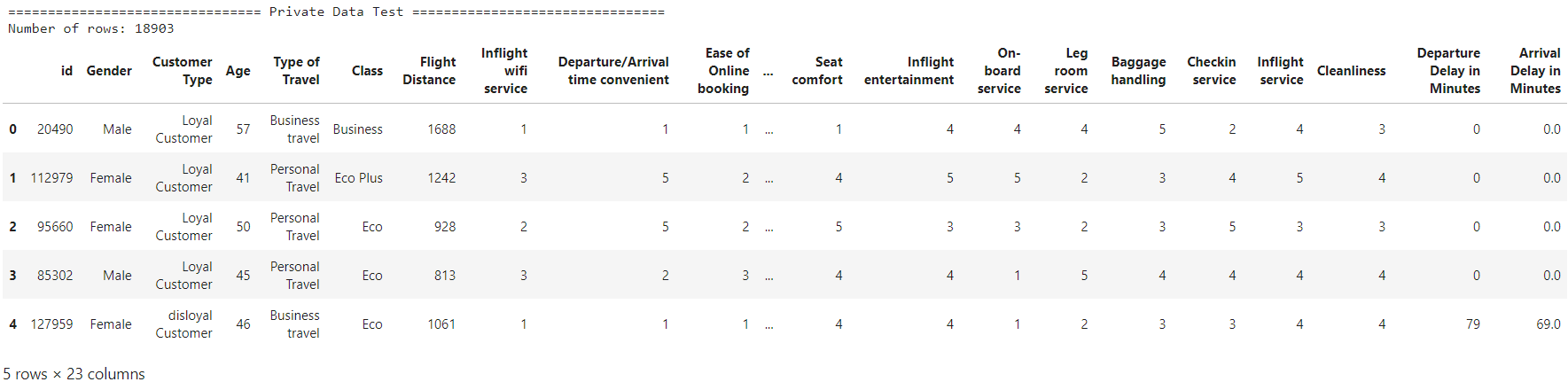


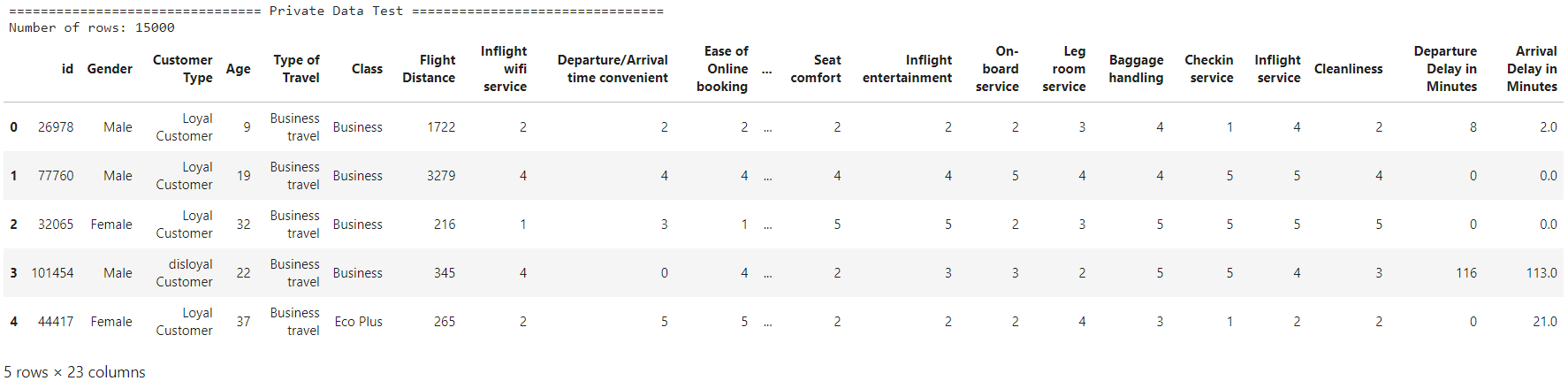
Some information about the libraries used in this project:

* NumPy: is a powerful open-source library in Python used for numerical and mathematical operations. It provides support for large, multi-dimensional arrays and matrices, along with a collection of high-level mathematical functions to operate on these arrays. NumPy is a fundamental library for scientific computing in Python and serves as the foundation for many other libraries and frameworks.
* Pandas: is a popular open-source data manipulation and analysis library for the Python programming language. It provides high-performance, easy-to-use data structures such as Series (1D labeled arrays) and DataFrame (2D labeled tables with rows and columns). Pandas is design to handle and analyze structured data seamlessly, making it a fundamental tool for data scientists, analysts, and researchers.
* Scikit-learn (sklearn): is a powerful and widely used open-source machine learning library for Python. It provides simple and efficient tools for data analysis and modeling, making it a go-to choice for practitioners and researchers in the field of machine learning and data science. scikit-learn is built on NumPy, SciPy, and Matplotlib, and it integrates well with other popular Python libraries.

Using Pandas, we readed the datasets contain in CSV files:







Concatinate the two data frames which are public\_test\_dataset and private\_test\_dataset, we receive the data frame test\_dataset:



Before creating and training the models, we first need to do some preprocessing steps, so that the data can become suitable for machine learning. The original data contains 24 features, but we just use 19 of them for the model to learn and predict our target feature – satifaction. In a different phrasing, we extract 19 columns from the dataset as X and separately take the column "satisfaction" as y.



After conducting a series of experiments and analyses, we have arrived at the conclusion that the incorporation of these specific features collectively has the potential to significantly enhance the overall accuracy of the model.

We conducted data encoding phase carefully since it is one of the most important step in the preprocessing pipeline, which can affect significantly to the model’s prediction. Data encoding typically refers to the process of converting data from one format or representation to another. However, when it comes to working with categorical data, especially in the context of machine learning, data encoding often specifically involves transforming categorical variables into a numerical format that can be used as input for machine learning algorithms. Two common techniques for this purpose are Label Encoding and One-Hot Encoding. In the context of this project, we contemplated employing Label Encoding as it aligns better with the nature of the categorical data under consideration. Label Encoding is a technique where each unique category or label in a categorical variable is assigned an integer. This method is suitable when the categorical variable has an inherent ordinal relationship. The LabelEncoder class from the scikit-learn library is commonly used for label encoding.



It is essential to divide the labeled data into training and testing sets to assess the learning outcomes of our models:



The first model we implemented is Multinomial Naive Bayes (MultinomialNB). Multinomial Naive Bayes is a probabilistic classification algorithm that is an extension of the Naive Bayes algorithm. It is particularly well-suited for classification tasks involving discrete features, especially when dealing with text data or data with word frequency counts. The "multinomial" part of its name comes from the fact that it models the likelihood of observing multiple discrete outcomes.

Bayes' theorem provides a way to update our beliefs about an event based on new evidence. In the context of classification, it helps us estimate the probability of a particular class given the observed features.

* : Posterior probability of class given features
* : Likelihood of observing features given class .
* : Prior probability of class .
* : Evidence or marginal likelihood.

Multinomial Naive Bayes makes the "naive" assumption that the features are conditionally independent given the class label. This simplifying assumption allows for tractable computations and is particularly useful in text classification tasks.

Multinomial Naive Bayes assumes that the likelihood of observing features follows a multinomial distribution. In text classification, these features are often word frequencies or counts.

* : Number of times feature appears in documents of class .
* : Total count of all features in class
* : Number of distinct features
* : Laplace smoothing parameter (a small positive value to handle unseen features

Here is the code implemented and train the Multinomial Naive Bayes Model:



The second model we conducted experiment is Support Vector Machine (SVM). Support Vector Machine (SVM) is a supervised machine learning algorithm designed for classification and regression tasks. Developed by Vapnik and his colleagues in the 1990s, SVM has become a popular choice for a wide range of applications due to its effectiveness in both linear and non-linear decision boundaries. SVM work with labeled training data, where each data point belongs to a specific class. If the data is not linearly separable, SVM may apply a kernel trick to transform the input features into a higher-dimensional space. This transformation makes it easier to find a hyperplane that separates the classes. SVM aims to find a hyperplane that best separates the data into different classes. The hyperplane is the decision boundary that maximizes the margin between the classes. The margin is the distance between the hyperplane and the nearest data point from each class. SVM seeks to maximize this margin as it improves the model's ability to generalize to unseen data.

To work with Support Vector Machine, we need to add one more phase in the preprocessing pipeline, which is standardization. Standardization is a crucial preprocessing step in machine learning that involves transforming the numerical features of a dataset to have a mean of 0 and a standard deviation of 1. This process is also known as z-score normalization or zero-mean normalization. Standardization ensures that all features have a consistent scale, preventing certain features from dominating the learning process simply because they have larger magnitudes. Some machine learning algorithms, like Support Vector Machines (SVM) and K-Nearest Neighbors (KNN), are sensitive to the scale of input features. Standardization helps these algorithms perform better. Standardization can accelerate the convergence of gradient-based optimization algorithms, particularly in iterative optimization processes. The standardized value for a feature is given by:

* : The standardized value for the feature .
* : The mean of the feature .
* : The standard deviation of the feature .

Here is the code for data standardization used in the project:



Then we implemented and train the SVM model:



# Evaluate the performance of the trained model

In assessing the effectiveness of the two models employed in this project, we will utilize distinct performance metrics, including precision, recall, and F1-score. These metrics offer insights into various aspects of the models' performance, and we will employ the provided method to generate a comprehensive classification report using the scikit-learn library. Let's delve into the explanation of these metrics:

* Precision: Precision measures the accuracy of positive predictions made by the model. It is calculated as the ratio of true positives to the sum of true positives and false positives. A higher precision indicates fewer false positives.
* Recall (Sensitivity): Recall assesses the model's ability to correctly identify positive instances among all actual positive instances. It is calculated as the ratio of true positives to the sum of true positives and false negatives. A higher recall signifies fewer false negatives.
* F1-score: The F1-score is the harmonic mean of precision and recall. It provides a balanced measure that considers both false positives and false negatives. It ranges between 0 and 1, with higher values indicating better model performance.

By employing the scikit-learn library's classification report, we can obtain a detailed and informative summary of these metrics, aiding in a thorough evaluation of the models' performance.

Table 1: Classification report of the Multinomial Naive Bayes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-score | Support |
| neutral or dissatisfied | 0.85 | 0.86 | 0.85 | 7883 |
| satisfied | 0.81 | 0.8 | 0.81 | 6118 |
| Accuracy |  |  | 0.83 | 14001 |
| Macro accuracy | 0.83 | 0.83 | 0.83 | 14001 |
| Weight accuracy | 0.83 | 0.83 | 0.83 | 14001 |

Based on the information presented in the table above, we can evaluate the performance of the Multinomial Naive Bayes model:

* For the class 'neutral or dissatisfied,' the precision is 0.85, indicating that 85% of instances predicted as 'neutral or dissatisfied' were correct. For the class 'satisfied,' the precision is 0.81, meaning that 81% of instances predicted as 'satisfied' were correct.
* The recall for 'neutral or dissatisfied' is 0.86, denoting that the model correctly identified 86% of actual 'neutral or dissatisfied' instances. The recall for 'satisfied' is 0.80, indicating that the model captured 80% of actual 'satisfied' instances.
* The F1 score is a balanced metric that considers both precision and recall. For 'neutral or dissatisfied,' the F1 score is 0.85, and for 'satisfied,' it is 0.81.
* 'neutral or dissatisfied' has 7,883 instances, and 'satisfied' has 6,118 instances in the test set.
* The overall accuracy of the model is 0.83, meaning that the model correctly predicted the class for 83% of instances in the test set.
* The macro average calculates the average of precision, recall, and F1 score across all classes, giving each class equal weight. In this case, the macro average for precision, recall, and F1 score is 0.83.
* The weighted average calculates the average of metrics, taking into account the support (number of instances) for each class. Classes with more instances have a higher impact on the average. In this case, the weighted average for precision, recall, and F1 score is 0.83.

Table 2: Classification report of the Support Vector Machine model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-score | Support |
| neutral or dissatisfied | 0.95 | 0.97 | 0.96 | 7883 |
| satisfied | 0.96 | 0.93 | 0.94 | 6118 |
| Accuracy |  |  | 0.95 | 14001 |
| Macro accuracy | 0.95 | 0.95 | 0.95 | 14001 |
| Weight accuracy | 0.95 | 0.95 | 0.95 | 14001 |

Based on the information presented in the table above, we can evaluate the performance of the Support Vector Machine model:

* For the class 'neutral or dissatisfied,' the precision is 0.95, indicating that 95% of instances predicted as 'neutral or dissatisfied' were correct. For the class 'satisfied,' the precision is 0.96, meaning that 96% of instances predicted as 'satisfied' were correct.
* The recall for 'neutral or dissatisfied' is 0.97, denoting that the model correctly identified 97% of actual 'neutral or dissatisfied' instances. The recall for 'satisfied' is 0.93, indicating that the model captured 93% of actual 'satisfied' instances.
* The F1 score is a balanced metric that considers both precision and recall. For 'neutral or dissatisfied,' the F1 score is 0.96, and for 'satisfied,' it is 0.94.
* 'neutral or dissatisfied' has 7,883 instances, and 'satisfied' has 6,118 instances in the test set.
* The overall accuracy of the model is 0.95, meaning that the model correctly predicted the class for 95% of instances in the test set.
* The macro average calculates the average of precision, recall, and F1 score across all classes, giving each class equal weight. In this case, the macro average for precision, recall, and F1 score is 0.95.
* The weighted average calculates the average of metrics, taking into account the support (number of instances) for each class. Classes with more instances have a higher impact on the average. In this case, the weighted average for precision, recall, and F1 score is 0.95.

Based on the tables presented earlier, it is evident that the Support Vector Machine (SVM) model consistently outperforms the Multinomial Naive Bayes model across key metrics, including precision, recall, and F1-score. SVM exhibits higher precision and recall for both classes, indicative of its superior overall performance. The SVM model achieves an accuracy of 0.95, surpassing the Naive Bayes model's accuracy of 0.83. Furthermore, the F1-score, which strikes a balance between precision and recall, is higher for SVM, implying a more effective trade-off between false positives and false negatives.

In summary, the Support Vector Machine model, as assessed by the provided metrics, exhibits superior performance compared to the Multinomial Naive Bayes model in the task of classifying customer satisfaction.

# Conclusion

In conclusion, the "VNA Customer Satisfaction" project embarked on the mission of constructing a machine learning model to effectively classify the satisfaction levels of Vietnam Airlines customers based on their features and ratings. The journey involved a systematic process that encapsulated the extraction of essential columns as features (X) and the target variable (y), followed by dataset encoding to ensure compatibility for model training. Subsequently, the data was meticulously split into training and test sets, laying the foundation for a robust evaluation.

The exploration commenced with experiments utilizing the Multinomial Naive Bayes algorithm, shedding light on its performance in the given context. Following this, a pivotal step involved standardizing the data, paving the way for the subsequent experimentation with the Support Vector Machine (SVM) algorithm. The application of SVM introduced a new dimension to the model, enabling a comprehensive comparison with the Naive Bayes approach.

Throughout this process, the efficacy of the models was rigorously assessed using performance metrics such as Precision, Recall, and F1-score. These metrics provided a nuanced understanding of the models' abilities to make accurate predictions, striking a balance between false positives and false negatives. The performance evaluation served as a crucial benchmark in determining the model that excelled in classifying customer satisfaction.

In essence, this project not only achieved its primary goal of building a machine learning model for VNA customer satisfaction classification but also laid the groundwork for future endeavors in customer experience analysis. The systematic workflow, from data extraction to model evaluation, reflects a commitment to precision and effectiveness in leveraging machine learning for enhancing customer satisfaction insights in the aviation domain.

# References

[1] Anshul Saini (October 27th, 2023) Guide on Support Vector Machine (SVM) Algorithm <https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/>

[2] Everton Gomede (November 11th, 2023) Understanding Multinomial Naive Bayes Classifier

<https://medium.com/@evertongomede/understanding-multinomial-naive-bayes-classifier-fdbd41b405bf>

[3] Aniruddha Bhandari (October 27th, 2023) Feature Engineering: Scaling, Normalization, and Standardization

<https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/>