

project report

comparison learning vector quantization and naïve bayes algorithm in airline passenger satisfaction

NIKODEMUS GALIH CANDRA WICAKSONO

18.K1.0082

Faculty of Computer Science

Soegijapranata Catholic University

2020

APPROVAL AND RATIFICATION PAGE (Heading plain)

COMPARISON LEARNING VECTOR QUANTIZATION AND NAÏVE BAYES ALGORITHM IN AIRLINE PASSENGER SATISFACTION

by

NIKODEMUS GALIH CANDRA WICAKSONO – 18.K1.0082

This project report has been approved and ratified

By the Faculty of Computer Science on January, DD. YYYY

With approval,

|  |  |
| --- | --- |
|  | Supervisor  Hironimus Leong, S.Kom., M.Kom.  NPP:05812007273 ­ |
| Examiners,  1.)  [NAMA PENGUJI 1]  NPP :  2.)  [NAMA PENGUJI 2]  NPP :  3.)  [NAMA PENGUJI 3]  NPP : |  |
|  | Dean of Faculty of Computer Science  Robertus Setiawan AJi Nugroho, Ph.D  NPP: 058.1.2004.264 |

DECLARATION OF AUTHORSHIP

I, the undersigned:

Name : NIKODEMUS GALIH CANDRA WICAKSONO

ID : 18.K1.0082

declare that this work, titled “COMPARISON LEARNING VECTOR QUANTIZATION AND NAÏVE BAYES ALGORITHM IN AIRLINE PASSENGER SATISFACTION", and the work presented in it is my own. I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at Soegijapranata Catholic University
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
3. Where I have consulted the published work of others, this is always clearly attributed.
4. Where I have quoted from the work of others, the source is always given.
5. Except for such quotations, this work is entirely my own work.
6. I have acknowledged all main sources of help.
7. Where the work is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Semarang, MONTH, DD, YYYY



NIKODEMUS GALIH CANDRA WICAKSONO

18.K1.0082

ACKNOWLEDGMENT

I thank all lecturers of Informatics Engineering at Soegijapranata Unika Semarang. Because they all the knowledge that I have gained during the study period is very useful in writing this thesis report. Especially to Mr. Hironimus Leong as my supervisor who always helped in the making of this thesis from beginning to end. And also for always providing useful motivation in the preparation of this thesis and for my future.

I am also grateful and thank all my close friends. They have always accompanied, helped, and motivated since the beginning of the study period until the completion of this thesis. Because of them I was able to develop myself together during my studies.

I also thank my family who always give support and encouragement so that this thesis can be completed quickly because they are always asked "How is the thesis going?". They are also the ones who have helped in various ways to date.

And finally I say to myself. Thank you for fighting so far. And also, spirit and good luck always after graduating from the study period.

ABSTRACT

Data is something that is very easy to obtain nowadays. Because data is easily obtained and stored digitally, the data becomes very large. For example, data from airline customer satisfaction which reaches around 130000 data. However, to process that much data, an algorithm is needed. Therefore, here I use the Naive Bayes algorithm and learning vector quantization (LVQ). Both algorithms are algorithms for classifying. This study was conducted to determine whether the two algorithms can be used in classifying airline passenger satisfaction data. I also compared the two algorithms to find out which one is better at classifying the airline satisfaction data. The level of accuracy becomes a parameter to determine a better algorithm.

In completing this research, several steps were carried out. The first step is to get and process the data. In processing the data, data preprocessing is carried out, namely deleting empty data and changing the data so that the program can process it. After that, the classification process is carried out using both algorithms. Each algorithm test was carried out 5 tests with different amounts of training data, namely 90%, 75%, 50%, 25% and 10%. After the test is complete, the accuracy of all tests is calculated. The average accuracy of each test with different training data will be the accuracy value of the algorithm.

From the results of the testing that has been done, there are 5 tests for naive Bayes and also 5 tests for learning vector quantization. The accuracy results obtained are the average Naive Bayes accuracy of 89.076% while the average accuracy of learning vector quantization is 79.39% for the airline's passenger satisfaction data. So it can be concluded that the two algorithms can be used to classify airline passenger satisfaction data and the Naive Bayes algorithm is better than learning vector quantization.

Keyword: naïve\_bayes, bayesian, learning\_vector\_quantization, lvq

TABLE OF CONTENTS

COVER i

[APPROVAL AND RATIFICATION PAGE (Heading plain) ii](#_Toc92109985)

[DECLARATION OF AUTHORSHIP iii](#_Toc92109986)

[ACKNOWLEDGMENT iv](#_Toc92109987)

[ABSTRACT (Abstract Title) v](#_Toc92109988)

[TABLE OF CONTENTS vi](#_Toc92109989)

[LIST OF FIGURE viii](#_Toc92109990)

[LIST OF TABLE ix](#_Toc92109991)

[CHAPTER 1 INTRODUCTION 10](#_Toc92109992)

[1.1. Background 10](#_Toc92109993)

[1.2. Problem Formulation 11](#_Toc92109994)

[1.3. Scope 11](#_Toc92109995)

[1.4. Objective 11](#_Toc92109996)

[CHAPTER 2 LITERATURE STUDY 12](#_Toc92109997)

[CHAPTER 3 RESEARCH METHODOLOGY 15](#_Toc92109998)

[3.1. Data Collection 15](#_Toc92109999)

[3.2. Algorithm 15](#_Toc92110000)

[3.3. Coding and Design 15](#_Toc92110001)

[3.4. Analysis 16](#_Toc92110002)

[3.5. Make a Report 16](#_Toc92110003)

[CHAPTER 4 ANALYSIS AND DESIGN 17](#_Toc92110004)

[CHAPTER 5 IMPLEMENTATION AND RESULTS 32](#_Toc92110005)

[5.1. Implementation 32](#_Toc92110006)

[5.2. Results 44](#_Toc92110007)

[CHAPTER 6 CONCLUSION 50](#_Toc92110008)

[REFERENCES 51](#_Toc92110009)

[APPENDIX a](#_Toc92110010)

LIST OF FIGURE

[**Figure 4.1** Workflow 17](#_Toc92107559)

[**Figure 4.2** LVQ Architecture 25](#_Toc92107560)

[**Figure 5.1** Graph of accuracy LVQ alpha=0.9 47](#_Toc92107561)

[**Figure 5.2** Graph of accuracy LVQ alpha=0.1 and eps=0.0000001 47](#_Toc92107562)

[**Figure 5.3** Graph of accuracy LVQ alpha=0.1 and eps=0.0001 48](#_Toc92107563)

[**Figure 5.4** Graph of accuracy alpha=0.1 and eps=0.01 48](#_Toc92107564)

[**Figure 5.5** Graph of accuracy alpha=0.01 and eps=0.0000001 48](#_Toc92107566)

[**Figure 5.6** Graph of accuracy alpha=0.01 and eps=0.0001 48](#_Toc92107567)

[**Figure 5.7** Graph of accuracy alpha=0.05 and eps=0.0000001 49](#_Toc92107568)

[**Figure 5.8** Graph of accuracy alpha=0.05 and eps=0.0001 49](#_Toc92107569)

LIST OF TABLE

[**Table 3.1.** Analysis 16](#_Toc92107570)

[**Table 4.1.** Data Table 17](#_Toc92107571)

[**Table 4.2.** Modified Attribute Data Table 19](#_Toc92107572)

[**Table 4.3.** Distribution of Training and Testing Data 19](#_Toc92107573)

[**Table 4.4.** Data Sample Naïve Bayes (20 data) 21](#_Toc92107574)

[**Table 4.5.** Training Dataset Naïve Bayes 21](#_Toc92107575)

[**Table 4.6.** Testing Dataset Naïve Bayes 22](#_Toc92107576)

[**Table 4.7.** Data Sample LVQ (20 data) 26](#_Toc92107577)

[**Table 4.8.** Training Dataset LVQ 26](#_Toc92107578)

[**Table 4.9.** Testing Dataset LVQ 27](#_Toc92107579)

[**Table 4.10.** Initial Weight 28](#_Toc92107580)

[**Table 5.1.** Naive Bayes Results 44](#_Toc92107581)

[**Table 5.2.** Results LVQ Alpha 0.9 and Eps 0.0000001, 0.0001, 0.01 45](#_Toc92107582)

[**Table 5.3.** Results LVQ Alpha 0.1 and Eps 0.0000001 45](#_Toc92107583)

[**Table 5.4.** Results LVQ Alpha 0.1 and Eps 0.0001 45](#_Toc92107584)

[**Table 5.5.** Results LVQ Alpha 0.1 and Eps 0.01 46](#_Toc92107585)

[**Table 5.6.** Results LVQ Alpha 0.01 and Eps 0.0000001 46](#_Toc92107586)

[**Table 5.7.** Results LVQ Alpha 0.01 and Eps 0.0001 46](#_Toc92107587)

[**Table 5.8.** Results LVQ Alpha 0.05 and Eps 0.0000001 46](#_Toc92107588)

[**Table 5.9.** Results LVQ Alpha 0.05 and Eps 0.0001 46](#_Toc92107589)

# CHAPTER 1 INTRODUCTION

## Background

Passenger satisfaction is one of the important factors for the improvement of an airline. The airline can find out what things need to be improved. With the hope that more and more airplane passengers use the airline, of course this increase must be done so that income also increases. To improve service, of course, you must know what things make passengers satisfied. This can be done from the data of passengers who have traveled by plane.

In this digital era, data is very easy to store and obtain. Not like in the past, which used paper to record data, but used the help of computers. One of the advantages is that it is easy to store large amounts of data, including passenger satisfaction data. If there are about 130,000 airline passenger satisfaction data, of course it is very difficult to process manually. This will make it difficult for airlines to improve services.

Because data storage uses a computer, we can also use a computer to process it. However, to process the existing data in order to get the results we want, an algorithm is needed. With the algorithm implemented on passenger satisfaction data, we can classify things that can make passengers satisfied with airline flight services. Of course, this is better than processing thousands of data manually.

Therefore, this time I implemented the Learning Vector Quantization (LVQ) and Naïve Bayes algorithms on the airline passenger satisfaction data that I got through Kaggle. It is hoped that this algorithm can process thousands of existing data and classify them. I am using 2 different algorithms so that I can compare the results of each implemented algorithm. And also, to find out which algorithm is better for classifying airline passenger satisfaction data by comparing the accuracy of the two algorithms. The results of this classification algorithm are expected to help airlines know what to do in the future.

## Problem Formulation

From the background above, we can formulate the existing problems.

1. Can the Naïve Bayes algorithm classify airline passenger satisfaction data?
2. Can the Learning Vector Quantization algorithm classify airline passenger satisfaction data?
3. Based on the level of accuracy, which algorithm is better in classifying passenger satisfaction data?

## Scope

In this project, I applied Learning vector quantization and Naive Bayes algorithm only for the data I used from https://www.kaggle.com/binaryjoker/airline-passenger-satisfaction with 129,880 data. The data consists of 23 measuring columns and 1 response column. To find out a better algorithm, I use the accuracy parameters of each algorithm. There will be 5 tests for each algorithm with a percentage of training data of 90%, 75%, 50%, 25%, and finally 10%.

## Objective

The purpose of this project is to find out whether the Learning Vector Quantization and Naive Bayes algorithms can classify aircraft passenger satisfaction from existing data. In addition, to find out from the two implemented algorithms, which algorithm is better based on the level of accuracy.

# LITERATURE STUDY

Gorzalczany et al. [1] explain that a lot of data mining does not provide deeper explanations and justifications than decisions. Therefore, they apply their knowledge discovery technique based on fuzzy rules to the problem of airline passenger satisfaction. They used a dataset from Kaggle of 259,760 records. With 23 variable columns, the dataset is almost the same as the dataset that I will use. The results obtained are that the most significant attribute is Inflight Entertainment with an accuracy of 75.2%. Followed by the attributes of Seat comfort and Inflight Wi-Fi Service. They do not classify, but can determine which classification variables affect airline passenger satisfaction more.

With the US Airlines dataset which is almost the same as before, Hayadi et al. [2] uses several classification algorithms. The algorithms used are KNN, Logistic regression, Gaussian NB, Decision Trees and Random Forest. The author runs using the GridSearchCV algorithm from Scikit-Learn. Of all the algorithms that have been run, Random forest has the best performance with 99% accuracy, 97% precision and 94% recall. From the many simulations carried out, the authors suggest optimizing the in-flight wi-fi service. After that also simplicity about online booking. Unlike before, this time with around 130,000 data that becomes 70,000 after deleting the NaN (Not a Number) value, it doesn't include inflight entertainment as an attribute that needs to be improved.

Different from the previous ones, but still about airline customer satisfaction. Hanif et al. [3] uses a dataset of 152 respondents who have used one of the Indonesian airlines, namely Lion Air. The data is taken and grouped by occupation so that it becomes 100 data and 5 classes of work. The author uses the SPSS tool to get the conclusions. By looking for multiple regression, validity, reliability, T test, F value test and the coefficient of determination and correlation, it is found that there is a positive and significant influence between service quality, passenger satisfaction and passenger behavioral intentions. The disadvantage of this research is that the data used is too little so that it can get different results if there are more datasets.

In the journal written by Wijayanto et al. [4], the Naive Bayes algorithm is also used for the passenger satisfaction dataset taken from Kaggle. The dataset used is most likely the same as that which will be used from this journal. With 129,880 data, the author uses the help of the KNime application for classification with Naive Bayes. The distribution of training data and data testing consists of 4 experiments. The first is training data: testing data is 90:10, the second is 85:15, the third is 80:20 and the last is 75:25. The results obtained that 90% of training data and 10% of testing data have an accuracy of 81.466%.

Religia and Amali [5] also uses Naive Bayes to classify airline passenger satisfaction. The dataset used is also from Kaggle but is different, as many as 25,976 data. In their research, they used Naive Bayes, Naive Bayes optimized particle swarm Optimization (PSO) and finally Naive Bayes optimized Genetic Algorithm (GA). To measure the performance used accuracy, precision and recall. The results obtained are that Naive Bayes optimized by PSO has the best results, namely the accuracy value is 86.13%, the precision value is 87.9% and the recall value is 87.29%.

Similar to this journal, Nugraha et al. [6] compare Naive Bayes with Learning Vector Quantization (LVQ) to classify. But here it is used to classify uterine diseases. In using Naive Bayes, the author uses 2 methods, Naive Bayes by using Laplacian Smoothing and without using it. The data used are 125 data from the medical records of patients at RSUD Dr. Moewardi Solo. The data here is divided into 4 experiments/simulations with the first experiment being training: the data is 20:80, the second is 40:60, the third is 60:40 and the last is 80:20. The results of 4 trials with training 20%, 40%, 60%, 80% got Naive Bayes without Laplacian Smoothing had 32%, 67.8%, 79%, 88.8% accuracy. These results are less good than if Naive Bayes using Laplician smoothing has an accuracy of 88%, 92.4%, 92.8%, 92.4%. The accuracy is said to be stable even though the training data is changed. Compared to LVQ the accuracy is 82.4%, 88.8%, 89.4%, 95.2%. However, the highest accuracy is obtained from LVQ with 80% training.

In another journal, for LVQ signature pattern recognition compared by Prabowo et al. [7] and combined by Ginting et al. [8]. Prabowo et al. compared with the Kohonen Neural Network (KNN), while Ginting et al. combined with Self Organizing Kohonen (SOK).

In the journal Prabowo et al. did 3 tests. Each test with a different number of classes, resolutions and patterns. In the first test with 25 patterns and a resolution of 30x20 Kohonen had 96% success for 1 second while LVQ was 100% for 2 seconds. Second with 40 patterns and 30x20 resolution with 95% Kohonen less than 1 second while LVQ 92.5% less than 1 second. The last test was 9 patterns with 100x100 resolution with 77.78% Kohonen for 2 seconds and LVQ 88.89% for 7 seconds. LVQ does have better accuracy than Kohonen, but it takes longer. While in the journal Ginting et al. can speed up the computational process. The combination of LVQ with SOK increases the processing speed of computing during training or during signature pattern recognition.

Unlike previous comparisons or combinations, Meliawati et al. [9] implement LVQ to predict majors at SMA PGRI 1 Banjarbaru. The data used is obtained from the value of report cards in 2010, 2011 and 2013. The data is used as training data, while the value of report cards in 2014 is used as testing data. It is not known how much of the exact amount of data was used. Researchers get 79.31% accuracy for iterations 60 and 90.

Samsir [10] also implements LVQ. LVQ is used to classify Throat Nose and Ear (ENT) disease at Rantauprapat Hospital Labuhanbatu. The input variable consists of 10 disease symptoms. The dataset used is small, which is only 57 data. Of the 57 data divided into 4 training. With the comparison of training data: Testing data is 60:40, second 70:30, third 80:20 and 90:10. In the results of testing accuracy, it is not found that the more testing data, the accuracy will improve. Maybe it's because there are too few datasets, so you might get different results if you get more datasets.

From the journal Gorzalczany et al. [1] and Hayadi et al. [2], the dataset used is almost the same. But both use different algorithms in classifying them. While Hanif et al. [3] using very different datasets and different algorithms, but it's still about passenger satisfaction. However, Wijayanto et al. [4] using the same dataset and algorithm, namely Naive Bayes only, but not compared to LVQ. Likewise, Religia and Amali [5] use only Naive Bayes to classify airline passenger satisfaction, but the datasets used are different. In the journal Nugraha et al. [6] The algorithms both compare LVQ and Naive Bayes, but they use it to classify obstetrical diseases. Prabowo et al. [7] also compared LVQ but with KNN for the case of signature pattern recognition. While Ginting et al. [8] combines LVQ with SOK for signature pattern recognition cases as well. For Meliawati et al. [9] and Samsir [10], they only implement LVQ with different datasets without comparing them or combining them.

# RESEARCH METHODOLOGY

## Data Collection

In collecting datasets, I use websites that provide various kinds of datasets. For this research I used data from https://www.kaggle.com/binaryjoker/airline-passenger-satisfaction. Data with the file name airline\_passengeer\_satisfaction.csv has a file size of 14.34MB. I downloaded this data on September 20, 2021. To download it you are required to Sign In first (Register if you don't have an account). The downloaded file will be a zip file, so it must be extracted to get the csv file. The total data obtained were 129,880 with 23 measuring columns and 1 response column.

## Algorithm

In choosing the algorithm, I consulted my supervisor. During the consultation, my lecturer informed and suggested the Learning Vector Quantization (LVQ) algorithm. This algorithm has not been used very often. Therefore, I use this LVQ algorithm. After using LVQ I looked for another algorithm to use as a comparison. Then I chose Naive Bayes because this algorithm is an algorithm that is often used, easy and has good accuracy. I use these two algorithms to classify supervised learning data about airline passenger satisfaction that has been obtained previously. In addition to knowing which algorithm is better in accuracy.

## Coding and Design

In this step, the MySql tools will be used. MySql is used because the existing dataset is in the form of 2-dimensional data (columns and rows) the same as the MySql database table. In addition, the installation of Mysql is very easy. By downloading xampp through the website https://www.apachefriends.org/download.html. Xampp already provides several versions for Windows, Linux and OS X operating systems. Here I use Linux. After MySql is installed, the data will be preprocessed. Continuous data such as age and distance will be changed first to make it easier to classify.

## Analysis

In analyzing, I will do 5 tests as follows :

Analysis

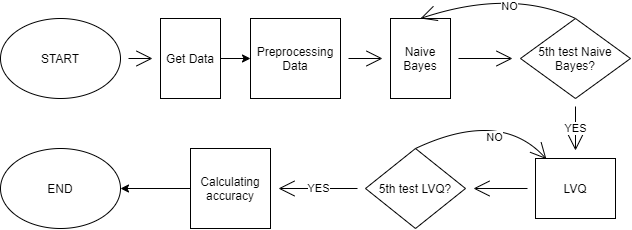
|  |  |  |
| --- | --- | --- |
|  | Training Data | Testing Data |
| I | 90% | 10% |
| II | 75% | 25% |
| III | 50% | 50% |
| IV | 25% | 75% |
| V | 10% | 90% |

In this analysis, it is divided into 5 stages to determine whether the amount of training has an effect. Influence on Naive Bayes accuracy and on LVQ accuracy.

## Make a Report

In making the report, I wrote chapters 1-4 first. After chapter 4 finished, I started the coding stage for program development. Then the results that have been carried out during the coding stage will be recorded in the chapter 5 report. And finally, conclusions will be drawn from the results of the coding stage which will be written in chapter 6.

# ANALYSIS AND DESIGN

In this research, there are several steps in outline. The first to get the data. The second is data preprocessing. Continued implementation of Naive Bayes and Learning Vector (LVQ) and the last is calculating accuracy. The flow is as in the following workflow:

Workflow

The first workflow is getting data. The data I use is data taken through Kaggle on September 20, 2021. The file can be downloaded with the file name airline\_passenger\_satisfaction.csv via the link https://www.kaggle.com/binaryjoker/airline-passenger-satisfaction. The data has 129,880 records in all. Has 24 attributes consisting of id, 22 input attributes and 1 label attribute. These attributes are as in table 4.1.

Data Table

|  |  |  |
| --- | --- | --- |
| No. | Attribute Name | Attribute Description |
| 1. | Id | Is the id of the data |
| 2. | Gender | “Female” and “Male” |
| 3. | Customer Type | “Loyal Customer” and “Disloyal Customer” |
| 4. | Age | Numbers from 7 to 85 |
| 5. | Type of Travel | “Business travel” and “Personal Travel” |
| 6. | Customer Class | “Business”, “Eco” and “Eco Plus” |
| 7. | Flight Distance | Numbers from 31 to 4983 |
| 8. | Inflight Wi-fi Service | “0”, “1”, “2”, “3”, “4”, “5” |
| 9. | Departure Arrival Time Convenient | “0”, “1”, “2”, “3”, “4”, “5” |
| 10. | Ease of Online Booking | “0”, “1”, “2”, “3”, “4”, “5” |
| 11. | Gate Location | “0”, “1”, “2”, “3”, “4”, “5” |
| 12. | Food and Drink | “0”, “1”, “2”, “3”, “4”, “5” |
| 13. | Online Boarding | “0”, “1”, “2”, “3”, “4”, “5” |
| 14. | Seat Comfort | “0”, “1”, “2”, “3”, “4”, “5” |
| 15. | Inflight Entertainment | “0”, “1”, “2”, “3”, “4”, “5” |
| 16. | Onboard Service | “0”, “1”, “2”, “3”, “4”, “5” |
| 17. | Leg Room Service | “0”, “1”, “2”, “3”, “4”, “5” |
| 18. | Baggage Handling | “0”, “1”, “2”, “3”, “4”, “5” |
| 19. | Check in Service | “0”, “1”, “2”, “3”, “4”, “5” |
| 20. | Inflight Service | “0”, “1”, “2”, “3”, “4”, “5” |
| 21. | Cleanliness | “0”, “1”, “2”, “3”, “4”, “5” |
| 22. | Departure Delay in Minutes | Numbers from 0 to 1592 |
| 23. | Arrival Delay in Minutes | Numbers from 0 to 1584 |
| 24. | Satisfaction | “Neutral or dissatisfied” and “Satisfied” |

The id attribute is only used as the line numbering of each record. Meanwhile, the gender attribute to the delay in minutes attribute will be used as input variables for both algorithms. The input variable is the value of the attribute. For example, the variables of gender are female and male. And lastly, the satisfaction attribute is a label attribute. The label attribute is an attribute that already contains the class of each record because the algorithm that will be used is supervised learning, which is an algorithm where the class has been determined. The class consists of 2 classes, namely the "satisfied" and "neutral or dissatisfied" classes.

After the data is obtained, the next step is to enter the data into the database. The data is entered into the database so that it can be processed by the program. From the existing data as shown above, there is data that cannot be processed by the program. Therefore, according to the workflow, the next step after getting the data is "data preprocessing". In this step the data that has a null value will be deleted first. This is so that the data processed is quality data. In preprocessing there are also attribute records that will be changed. Notes will be converted to numbers at small intervals. For example, there are too many age categories, which will then be changed to "0" where the age is <28, then "1" where the age is between 28 and 52, and finally "2" which is over 52. The value of 28 and 52 is based on quantile values ​​that can be seen on the data link is downloaded. There is also data that is not in the form of numbers will be converted to numbers. This is because the LVQ algorithm will be calculated based on the value of the attribute. So that it is converted into a number so that it can be calculated. For example, the original gender "Female" and "Male" will be changed to "0" and "1". The attributes that are changed in the preprocessing stage are as follows:

Modified Attribute Data Table

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Attribute Name | Before | After |
| 1. | Gender | “Female” and “Male” | “0” and “1” |
| 2. | Customer Type | “Loyal Customer” and “Disloyal Customer” | “0” and “1” |
| 3. | Age | Numbers from 7 to 85 | “0” (<28), ”1” (<52) and “2” (>=52) |
| 4. | Type of Travel | “Business travel” and “Personal Travel” | “0” and “1” |
| 5. | Customer Class | “Business”, “Eco” and “Eco Plus” | “0”, “1”, and “2” |
| 6. | Flight Distance | Numbers from 31 to 4983 | “0” (<=414), “1” (<=1744), “2” (>1744) |
| 7. | Departure Delay | Numbers from 0 to 1592 | “0” (<=12) and “1” (>12) |
| 8. | Arrival Delay | Numbers from 0 to 1584 | “0” (<=13) and “1” (>13) |
| 9. | Satisfaction | “Neutral or dissatisfied” and “Satisfied” | “0” and “1” |

In addition to changing the data, in the preprocessing, deletion of data will be carried out. Deleted data are records that have attributes with null or empty values. This is done so that the data can be processed by the program. I did not change the blank data with 0 or 1 to maintain the quality of the existing data. After deleting the data, the preprocessing step has been completed. The next step is to implement an algorithm for airline passenger satisfaction data.

In implementing the two algorithms, 5 tests will be carried out on each algorithm. In each test, the amount of training data and testing data will be different. The difference in the amount of data is later to see whether the amount of different data will affect the final result. Comparison of the amount of data as shown in the following table.

Distribution of Training and Testing Data

|  |  |  |
| --- | --- | --- |
| Test | Training Data | Testing Data |
| I | 90 % | 10% |
| II | 75% | 25% |
| III | 50% | 50% |
| IV | 25% | 75% |
| V | 10% | 90% |

As in the workflow, after preprocessing it will implement Naive Bayes. Naive Bayes will be tested up to 5 times. Each test will use a different number of datasets as shown in table 4.3. And at the end of the Naive Bayes implementation, the accuracy value will be calculated. Likewise with LVQ, which will test 5 times and look for accuracy. In finding the value of accuracy will use the formula. The formula used is like the following function.

True Positive (TP) = Total class 1 (satisfied) and classified as class 1

True Negative (TN) = Total class 0 (neutral or dissatisfied) and classified as class 0

False Negative (FN) = Total class 1 (satisfied) and classified as class 0

False Positive (FP) = Total class 0 (neutral or dissatisfied) and classified as class 1

Accuracy = The result of dividing the number of correct classifications with the total data and multiplied by 100%

The formula above will be used to find the accuracy value of each test from the two algorithms. Therefore in each test will be calculated the number of TP, TN, FP and FN. After all the tests are complete, the accuracy value of all the tests will be obtained.

For the first, testing will be carried out using the Naive Bayes algorithm. This algorithm is a supervised learning classification algorithm. Which means the class of data has been defined or labeled. In this study, there is the attribute 'satisfaction'. Naive Bayes itself is a good algorithm. Because the formula used is easy and also has a high accuracy value. Broadly speaking, the Bayes theorem formula used is like the following function.

x = attribute class/label

y = attribute input

In the Naive Bayes algorithm there are steps in implementing it. As an example of implementation, in this report I use 20 sample data from data that has been preprocessed. This data will also be used as an example implementation in this report for the LVQ algorithm. The data is as shown in the table below.

Data Sample Naïve Bayes (20 data)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 1 | 1 | 1 | 0 | 1 | 2 | 1 | 3 | 4 | 3 | 1 | 5 | 3 | 5 | 5 | 4 | 3 | 4 | 4 | 5 | 5 | 1 | 1 | 0 |
| 2 | 1 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 3 | 3 | 1 | 3 | 1 | 1 | 1 | 5 | 3 | 1 | 4 | 1 | 0 | 0 | 0 |
| 3 | 0 | 1 | 0 | 0 | 0 | 1 | 2 | 2 | 2 | 2 | 5 | 5 | 5 | 5 | 4 | 3 | 4 | 4 | 4 | 5 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 2 | 5 | 5 | 5 | 2 | 2 | 2 | 2 | 2 | 5 | 3 | 1 | 4 | 2 | 0 | 0 | 0 |
| 5 | 1 | 1 | 2 | 0 | 0 | 0 | 3 | 3 | 3 | 3 | 4 | 5 | 5 | 3 | 3 | 4 | 4 | 3 | 3 | 3 | 0 | 0 | 1 |
| 6 | 0 | 1 | 0 | 1 | 1 | 1 | 3 | 4 | 2 | 1 | 1 | 2 | 1 | 1 | 3 | 4 | 4 | 4 | 4 | 1 | 0 | 0 | 0 |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 4 | 2 | 3 | 2 | 2 | 2 | 2 | 3 | 3 | 4 | 3 | 5 | 2 | 0 | 1 | 0 |
| 8 | 0 | 1 | 2 | 0 | 0 | 2 | 4 | 3 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 4 | 5 | 4 | 0 | 0 | 1 |
| 9 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 4 | 3 | 3 | 1 | 1 | 2 | 1 | 4 | 1 | 2 | 0 | 0 | 0 |
| 10 | 1 | 0 | 0 | 0 | 1 | 1 | 3 | 3 | 3 | 4 | 2 | 3 | 3 | 2 | 2 | 3 | 4 | 4 | 3 | 2 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 | 1 | 1 | 4 | 5 | 5 | 4 | 2 | 5 | 2 | 2 | 3 | 3 | 5 | 3 | 5 | 2 | 0 | 0 | 0 |
| 12 | 0 | 1 | 0 | 1 | 2 | 0 | 2 | 4 | 2 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 5 | 5 | 5 | 1 | 0 | 0 | 0 |
| 13 | 1 | 1 | 2 | 0 | 1 | 1 | 1 | 4 | 4 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 3 | 4 | 4 | 1 | 1 | 0 | 0 |
| 14 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 2 | 4 | 3 | 4 | 4 | 4 | 4 | 4 | 5 | 2 | 2 | 2 | 4 | 0 | 0 | 1 |
| 15 | 0 | 1 | 0 | 1 | 1 | 1 | 3 | 2 | 3 | 2 | 2 | 3 | 2 | 2 | 4 | 3 | 2 | 2 | 1 | 2 | 1 | 1 | 0 |
| 16 | 1 | 0 | 0 | 0 | 1 | 1 | 2 | 1 | 2 | 3 | 4 | 2 | 1 | 4 | 2 | 1 | 4 | 1 | 3 | 4 | 0 | 0 | 0 |
| 17 | 0 | 1 | 0 | 0 | 0 | 2 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 5 | 3 | 4 | 5 | 4 | 4 | 1 | 1 | 1 |
| 18 | 1 | 1 | 1 | 0 | 0 | 2 | 4 | 4 | 2 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 3 | 5 | 5 | 0 | 0 | 1 |
| 19 | 0 | 1 | 1 | 0 | 0 | 2 | 4 | 4 | 4 | 4 | 3 | 4 | 5 | 5 | 5 | 5 | 5 | 3 | 5 | 4 | 0 | 0 | 1 |
| 20 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 3 | 2 | 5 | 3 | 5 | 5 | 1 | 2 | 4 | 3 | 2 | 5 | 1 | 1 | 0 |

The table data above will be used as an example of implementation in this chapter 4 report. Column 1 is gender, 2 is customer type and so on as in table 4.1. The Naive Bayes steps used in this study are:

1. Divide the dataset into training datasets and testing datasets. The distribution of the dataset is as shown in table 4.3. For example, I will take 20 sample data that has been preprocessed. Because this is the first test, 18 data are used as training and 2 data as testing. I separate manually by id.

Training Dataset Naïve Bayes

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 1 | 1 | 1 | 0 | 1 | 2 | 1 | 3 | 4 | 3 | 1 | 5 | 3 | 5 | 5 | 4 | 3 | 4 | 4 | 5 | 5 | 1 | 1 | 0 |
| 2 | 1 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 3 | 3 | 1 | 3 | 1 | 1 | 1 | 5 | 3 | 1 | 4 | 1 | 0 | 0 | 0 |
| 3 | 0 | 1 | 0 | 0 | 0 | 1 | 2 | 2 | 2 | 2 | 5 | 5 | 5 | 5 | 4 | 3 | 4 | 4 | 4 | 5 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 2 | 5 | 5 | 5 | 2 | 2 | 2 | 2 | 2 | 5 | 3 | 1 | 4 | 2 | 0 | 0 | 0 |
| 5 | 1 | 1 | 2 | 0 | 0 | 0 | 3 | 3 | 3 | 3 | 4 | 5 | 5 | 3 | 3 | 4 | 4 | 3 | 3 | 3 | 0 | 0 | 1 |
| 6 | 0 | 1 | 0 | 1 | 1 | 1 | 3 | 4 | 2 | 1 | 1 | 2 | 1 | 1 | 3 | 4 | 4 | 4 | 4 | 1 | 0 | 0 | 0 |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 4 | 2 | 3 | 2 | 2 | 2 | 2 | 3 | 3 | 4 | 3 | 5 | 2 | 0 | 1 | 0 |
| 8 | 0 | 1 | 2 | 0 | 0 | 2 | 4 | 3 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 4 | 5 | 4 | 0 | 0 | 1 |
| 9 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 4 | 3 | 3 | 1 | 1 | 2 | 1 | 4 | 1 | 2 | 0 | 0 | 0 |
| 10 | 1 | 0 | 0 | 0 | 1 | 1 | 3 | 3 | 3 | 4 | 2 | 3 | 3 | 2 | 2 | 3 | 4 | 4 | 3 | 2 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 | 1 | 1 | 4 | 5 | 5 | 4 | 2 | 5 | 2 | 2 | 3 | 3 | 5 | 3 | 5 | 2 | 0 | 0 | 0 |
| 12 | 0 | 1 | 0 | 1 | 2 | 0 | 2 | 4 | 2 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 5 | 5 | 5 | 1 | 0 | 0 | 0 |
| 13 | 1 | 1 | 2 | 0 | 1 | 1 | 1 | 4 | 4 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 3 | 4 | 4 | 1 | 1 | 0 | 0 |
| 14 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 2 | 4 | 3 | 4 | 4 | 4 | 4 | 4 | 5 | 2 | 2 | 2 | 4 | 0 | 0 | 1 |
| 15 | 0 | 1 | 0 | 1 | 1 | 1 | 3 | 2 | 3 | 2 | 2 | 3 | 2 | 2 | 4 | 3 | 2 | 2 | 1 | 2 | 1 | 1 | 0 |
| 16 | 1 | 0 | 0 | 0 | 1 | 1 | 2 | 1 | 2 | 3 | 4 | 2 | 1 | 4 | 2 | 1 | 4 | 1 | 3 | 4 | 0 | 0 | 0 |
| 17 | 0 | 1 | 0 | 0 | 0 | 2 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 5 | 3 | 4 | 5 | 4 | 4 | 1 | 1 | 1 |
| 18 | 1 | 1 | 1 | 0 | 0 | 2 | 4 | 4 | 2 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 3 | 5 | 5 | 0 | 0 | 1 |

Testing Dataset Naïve Bayes

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 1 | 0 | 1 | 1 | 0 | 0 | 2 | 4 | 4 | 4 | 4 | 3 | 4 | 5 | 5 | 5 | 5 | 5 | 3 | 5 | 4 | 0 | 0 | 1 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 3 | 2 | 5 | 3 | 5 | 5 | 1 | 2 | 4 | 3 | 2 | 5 | 1 | 1 | 0 |

1. Calculate P(x) for each class/label attribute variable (‘satisfied’ and ‘neutral or dissatisfied’). The formula used to find it is as follows.

x = Total data class/label from training data

Total data = Total data from training data

P(x) = Probability of variable class/label

The class/label consists of data, there are 2 labels, namely 0 instead of "neutral or dissatisfied" and 1 substitute for "satisfied". Therefore, P (0) and P (1) will be calculated from the training data. Because the number of data that has the label 0 is 12 then:

After that we also look for the value of P (1). Because the number of data that has the label 1 is 6 then:

1. Calculate the probability of the input variable from each class/label or P(a|x). Calculate P(a|x) for all input attributes(gender, customer type, age, etc). The formula used is as follows for each attribute.

a = class input from testing data

x = class/label

Total data ax = Total data where class/label is x and input is a from training data

Total data x = Total data where class/label is x from training data

P(a|x) = Probability of a against x

In the first testing data (id=1), the value of a is 0 for the gender class. Since the number of data with gender 0 and also label 0 is 6 and the number of data with label 0 is 12, then p(gender=0|label=0) is:

Next we also look for the probability value for gender 0 as well but with the label 1, the probability is:

Because the customer type value in the first testing data is 1, then look for P (customer type = 1 | label=0). After that also calculate P (customer type =1 | label=1). Do the same for the age, type of travel and other attributes. Then we will get P(a|x) a number of input attributes, which is 22 P(a|x).

1. Calculate the result of multiplying P(a|x) all attributes and P(x).

Because the number of attributes is too many, the result of the multiplication of P(a|x) that I show as an example is only the gender and customer type attributes. Then the result of each class/label is:

1. The biggest results are prediction results

From the results of label 0 and label 1, it can be seen that label 0 has a greater distance value with a value of 0.22445. Therefore, the prediction result is 0.

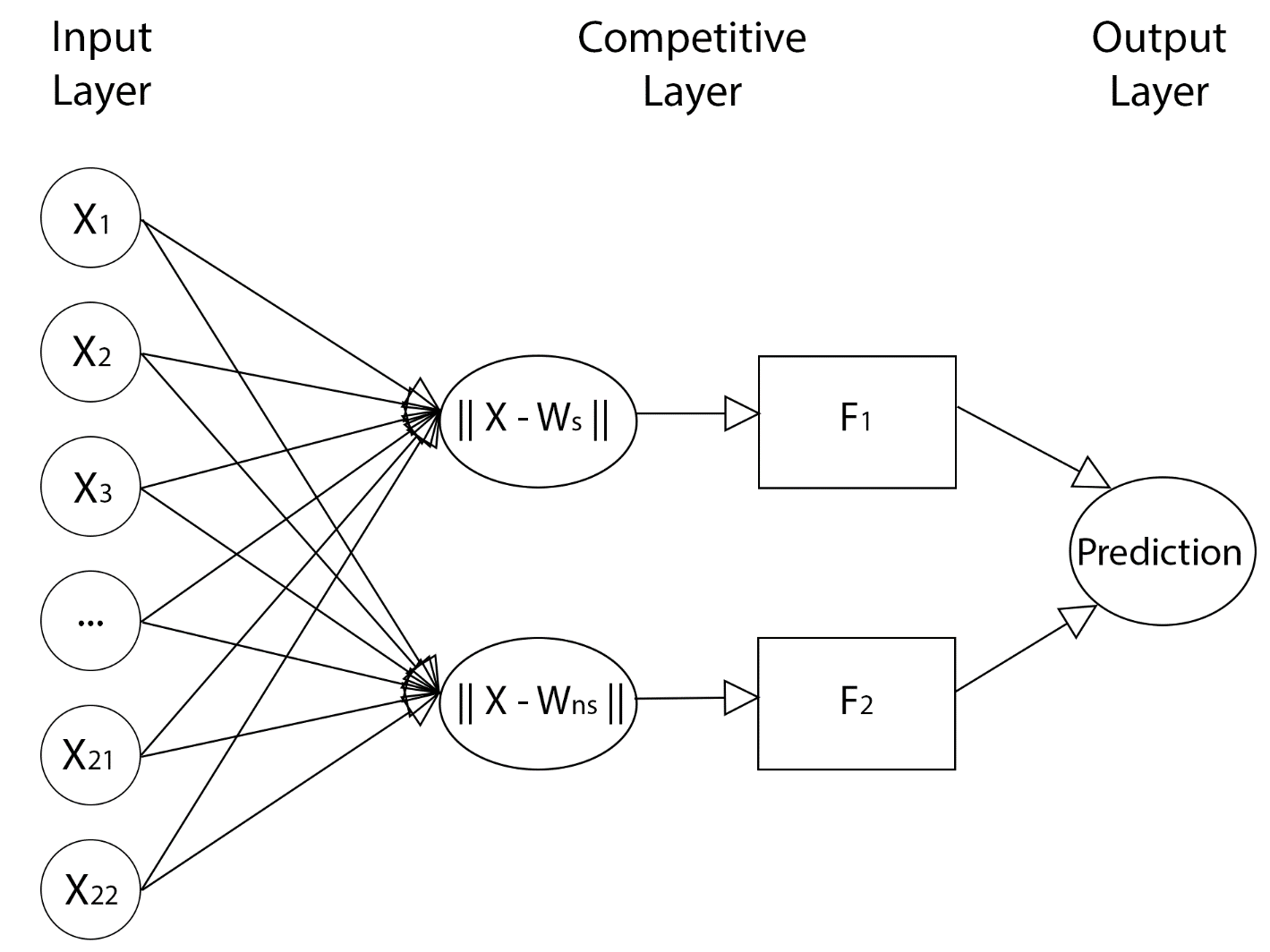
1. To find the accuracy results later then if:
   1. Label class “1” and prediction results is “1”, TP added 1
   2. Label class “0” and prediction result is “0”, TN added 1
   3. Label class “1” and prediction results is “0”, FN added 1
   4. Label class “0” and prediction results is “1”, FP added 1

In the testing data table, it can be seen that the class/label from the first test (id=1) is 1. However, the prediction result from the calculation that has been done is 0. Therefore, we add 1 number of FN for this first Naive Bayes test.

1. Repeat steps 3-6 for the second test(id=2) and so on until the last id of the testing data.

After step 7 is complete then we calculate the accuracy. To calculate accuracy like Function 1 with input in step 6. Then the first test is done. Repeat steps 1-7 for the second to fifth Naïve Bayes test with the number of training data and datasets as specified. If it has been tested 5 times, then Naive Bayes has been completed in this study.

After 5 times of testing Naive Bayes, next is the Learning Vector Quantization (LVQ) algorithm. LVQ is a classification algorithm like Naive Bayes which is supervised learning. The architecture of LVQ in this study looks like the following design.

LVQ Architecture

In the LVQ architecture there are layers, namely input, process or competitive and finally output. In the input layer, there are 22 inputs, namely X1 to X22. Xn is the value of the input attribute, namely gender as the first attribute, customer type as the second attribute to the 22nd attribute. From 22 inputs it will be 2 in the competitive layer. This is because there are 2 class/labels, namely 'satisfied' and 'neutral or dissatisfied'. To make these two results, calculations are carried out using the Euclidean distance. The calculation is to find the input distance to each class/label. Euclidean distance formula like the following function.

||X-W|| = Euclidean distance

Xn = Value from attribute n

Wcn = Weight of class/label c and attribute n

After calculating the input to the weight of each class, we can get the prediction results. Prediction results on the output layer can be obtained by looking for a smaller value. However, if the values are the same, it can be determined which class will be entered. Here I specify enter the class "1" which is satisfied. As an example of implementation of LVQ, I use sample for dataset like Naïve Bayes. I use 20 sample datasets. The data is as shown in the table below.

Data Sample LVQ (20 data)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 1 | 1 | 1 | 0 | 1 | 2 | 1 | 3 | 4 | 3 | 1 | 5 | 3 | 5 | 5 | 4 | 3 | 4 | 4 | 5 | 5 | 1 | 1 | 0 |
| 2 | 1 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 3 | 3 | 1 | 3 | 1 | 1 | 1 | 5 | 3 | 1 | 4 | 1 | 0 | 0 | 0 |
| 3 | 0 | 1 | 0 | 0 | 0 | 1 | 2 | 2 | 2 | 2 | 5 | 5 | 5 | 5 | 4 | 3 | 4 | 4 | 4 | 5 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 2 | 5 | 5 | 5 | 2 | 2 | 2 | 2 | 2 | 5 | 3 | 1 | 4 | 2 | 0 | 0 | 0 |
| 5 | 1 | 1 | 2 | 0 | 0 | 0 | 3 | 3 | 3 | 3 | 4 | 5 | 5 | 3 | 3 | 4 | 4 | 3 | 3 | 3 | 0 | 0 | 1 |
| 6 | 0 | 1 | 0 | 1 | 1 | 1 | 3 | 4 | 2 | 1 | 1 | 2 | 1 | 1 | 3 | 4 | 4 | 4 | 4 | 1 | 0 | 0 | 0 |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 4 | 2 | 3 | 2 | 2 | 2 | 2 | 3 | 3 | 4 | 3 | 5 | 2 | 0 | 1 | 0 |
| 8 | 0 | 1 | 2 | 0 | 0 | 2 | 4 | 3 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 4 | 5 | 4 | 0 | 0 | 1 |
| 9 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 4 | 3 | 3 | 1 | 1 | 2 | 1 | 4 | 1 | 2 | 0 | 0 | 0 |
| 10 | 1 | 0 | 0 | 0 | 1 | 1 | 3 | 3 | 3 | 4 | 2 | 3 | 3 | 2 | 2 | 3 | 4 | 4 | 3 | 2 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 | 1 | 1 | 4 | 5 | 5 | 4 | 2 | 5 | 2 | 2 | 3 | 3 | 5 | 3 | 5 | 2 | 0 | 0 | 0 |
| 12 | 0 | 1 | 0 | 1 | 2 | 0 | 2 | 4 | 2 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 5 | 5 | 5 | 1 | 0 | 0 | 0 |
| 13 | 1 | 1 | 2 | 0 | 1 | 1 | 1 | 4 | 4 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 3 | 4 | 4 | 1 | 1 | 0 | 0 |
| 14 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 2 | 4 | 3 | 4 | 4 | 4 | 4 | 4 | 5 | 2 | 2 | 2 | 4 | 0 | 0 | 1 |
| 15 | 0 | 1 | 0 | 1 | 1 | 1 | 3 | 2 | 3 | 2 | 2 | 3 | 2 | 2 | 4 | 3 | 2 | 2 | 1 | 2 | 1 | 1 | 0 |
| 16 | 1 | 0 | 0 | 0 | 1 | 1 | 2 | 1 | 2 | 3 | 4 | 2 | 1 | 4 | 2 | 1 | 4 | 1 | 3 | 4 | 0 | 0 | 0 |
| 17 | 0 | 1 | 0 | 0 | 0 | 2 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 5 | 3 | 4 | 5 | 4 | 4 | 1 | 1 | 1 |
| 18 | 1 | 1 | 1 | 0 | 0 | 2 | 4 | 4 | 2 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 3 | 5 | 5 | 0 | 0 | 1 |
| 19 | 0 | 1 | 1 | 0 | 0 | 2 | 4 | 4 | 4 | 4 | 3 | 4 | 5 | 5 | 5 | 5 | 5 | 3 | 5 | 4 | 0 | 0 | 1 |
| 20 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 3 | 2 | 5 | 3 | 5 | 5 | 1 | 2 | 4 | 3 | 2 | 5 | 1 | 1 | 0 |

In doing this LVQ, the steps taken are as follows:

1. Divide the dataset into training datasets and testing datasets. This step is similar to step 1 of Naïve Bayes. Then the training and testing data will look like below.

Training Dataset LVQ

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 1 | 1 | 1 | 0 | 1 | 2 | 1 | 3 | 4 | 3 | 1 | 5 | 3 | 5 | 5 | 4 | 3 | 4 | 4 | 5 | 5 | 1 | 1 | 0 |
| 2 | 1 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 3 | 3 | 1 | 3 | 1 | 1 | 1 | 5 | 3 | 1 | 4 | 1 | 0 | 0 | 0 |
| 3 | 0 | 1 | 0 | 0 | 0 | 1 | 2 | 2 | 2 | 2 | 5 | 5 | 5 | 5 | 4 | 3 | 4 | 4 | 4 | 5 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 2 | 5 | 5 | 5 | 2 | 2 | 2 | 2 | 2 | 5 | 3 | 1 | 4 | 2 | 0 | 0 | 0 |
| 5 | 1 | 1 | 2 | 0 | 0 | 0 | 3 | 3 | 3 | 3 | 4 | 5 | 5 | 3 | 3 | 4 | 4 | 3 | 3 | 3 | 0 | 0 | 1 |
| 6 | 0 | 1 | 0 | 1 | 1 | 1 | 3 | 4 | 2 | 1 | 1 | 2 | 1 | 1 | 3 | 4 | 4 | 4 | 4 | 1 | 0 | 0 | 0 |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 4 | 2 | 3 | 2 | 2 | 2 | 2 | 3 | 3 | 4 | 3 | 5 | 2 | 0 | 1 | 0 |
| 8 | 0 | 1 | 2 | 0 | 0 | 2 | 4 | 3 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 4 | 5 | 4 | 0 | 0 | 1 |
| 9 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 4 | 3 | 3 | 1 | 1 | 2 | 1 | 4 | 1 | 2 | 0 | 0 | 0 |
| 10 | 1 | 0 | 0 | 0 | 1 | 1 | 3 | 3 | 3 | 4 | 2 | 3 | 3 | 2 | 2 | 3 | 4 | 4 | 3 | 2 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 | 1 | 1 | 4 | 5 | 5 | 4 | 2 | 5 | 2 | 2 | 3 | 3 | 5 | 3 | 5 | 2 | 0 | 0 | 0 |
| 12 | 0 | 1 | 0 | 1 | 2 | 0 | 2 | 4 | 2 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 5 | 5 | 5 | 1 | 0 | 0 | 0 |
| 13 | 1 | 1 | 2 | 0 | 1 | 1 | 1 | 4 | 4 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 3 | 4 | 4 | 1 | 1 | 0 | 0 |
| 14 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 2 | 4 | 3 | 4 | 4 | 4 | 4 | 4 | 5 | 2 | 2 | 2 | 4 | 0 | 0 | 1 |
| 15 | 0 | 1 | 0 | 1 | 1 | 1 | 3 | 2 | 3 | 2 | 2 | 3 | 2 | 2 | 4 | 3 | 2 | 2 | 1 | 2 | 1 | 1 | 0 |
| 16 | 1 | 0 | 0 | 0 | 1 | 1 | 2 | 1 | 2 | 3 | 4 | 2 | 1 | 4 | 2 | 1 | 4 | 1 | 3 | 4 | 0 | 0 | 0 |
| 17 | 0 | 1 | 0 | 0 | 0 | 2 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 5 | 3 | 4 | 5 | 4 | 4 | 1 | 1 | 1 |
| 18 | 1 | 1 | 1 | 0 | 0 | 2 | 4 | 4 | 2 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 3 | 5 | 5 | 0 | 0 | 1 |

Testing Dataset LVQ

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 1 | 0 | 1 | 1 | 0 | 0 | 2 | 4 | 4 | 4 | 4 | 3 | 4 | 5 | 5 | 5 | 5 | 5 | 3 | 5 | 4 | 0 | 0 | 1 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 3 | 2 | 5 | 3 | 5 | 5 | 1 | 2 | 4 | 3 | 2 | 5 | 1 | 1 | 0 |

1. Initialization
   1. The initial weight (W) is randomly or manual selected 1 input data training from each class. Because in this dataset there are 2 class/labels, namely "satisfied" and "neutral or dissatisfied", then there are 2 initial weights. Weight for satisfied (Ws) and Weight for neutral or dissatisfied (Wns).

For example, I manually select data from the training data with id 1 for Wns, because data where id 1 has class/label 0 or “neutral or dissatisfied”. Wns1 is the value of the first attribute, namely the gender attribute, so Wns1 is 1. Wns2 is from the second attribute, so Wns2 is 1 and then on to the 22nd attribute of the training data with id 1.

In addition to Wns initialization, it also needs Ws initialization. For example, I manually select data from the training data with id 3 for Ws, because data with id 3 has class/label 1 or “satisfied”. Same as Wns initialization, Ws1 is 0. Ws2 is 1 and so on from training data with id 3. So that the initialization value of W is like the table below.

Initial Weight

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| W | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| s | 0 | 1 | 0 | 0 | 0 | 1 | 2 | 2 | 2 | 2 | 5 | 5 | 5 | 5 | 4 | 3 | 4 | 4 | 4 | 5 | 0 | 0 |
| ns | 1 | 1 | 0 | 1 | 2 | 1 | 3 | 4 | 3 | 1 | 5 | 3 | 5 | 5 | 4 | 3 | 4 | 4 | 5 | 5 | 1 | 1 |

The data used as weight initialization is not reused during the training process later. Therefore, training data with id 1 and 3 are not reused. To facilitate programming, the data will be removed from the training data so that it is no longer possible to use it.

* 1. Maximum Iterations (MaxEpoch). The maximum iteration that I set is 16. I set it that way because the amount of data from the training data is 18 data minus 2 for the initialization weight.
  2. Epoch. Epoch initialization is 1.
  3. Parameters learning rate/alpha (α). Alpha initialization is 0,9
  4. Minimum error (Eps). Eps initialization is 0,0000001

1. Input
   1. Input Xn
      * X = input value
      * n = attribute input to n

The input value is taken from the input attribute values, namely gender, customer type, age and so on. So for the first iteration, the value of X is taken from the first training data. The first training data is data with id 2 because data with id 1 and 3 have been used as initial weights. So X1 is 1, X2 is 0, X3 is 0 and so on.

* 1. Target = Class/label of data from testing data.

From Input Xn above, the data used is training data with id 2. Therefore, Target is the class/label value of training data with id 2. So the value of target is 0.

1. If Epoch < MaxEpoch or α > eps:

Because the epoch with a value of 1 is less than the max epoch with a value of 16 and an alpha value of 0.9 more than the eps value of 0.0000001 then this provision is true. So it will run the steps below.

* 1. Find the input distance to each weight using ||X-W||. Then determine the minimum value as the prediction class (J). However, if the distance between the two weights is the same, the prediction class can be determined, whether it is "satisfied" or "neutral or dissatisfied". I specify here as class satisfied.

In this step we will look for weight satisfied (Ws) and weight neutral or not satisfied (Wns). For example, I only use the initial 3 attributes.

Here it can be seen that the results ||X-Ws|| is 1,4142 and ||X-Wns|| is 1. Because the minimum value is 1 that is the result ||X-Wns|| then the prediction result (J) is 0 (neutral or dissatisfied).

* 1. Update Wj for each Wn.
     + If J= T then Wj`= Wj + α (X - Wj)
     + If J ≠ T then Wj`= Wj – α (X - Wj)

T=Target

Wj = Weight class j

α = Learning ratio

j = prediction class

X = data value

Wn = Weigh index n

The target (T) of the data with id 2 is 0 and J is also 0. So we will change W from prediction class to Wj` = Wj + (X - Wj). So Wns1` = Wns1 + (X1-Wns1). Then Wns1 will change to 1, Wns2 to 0.1 and so on until Wns22.

* 1. Update the value of α.

In updating the alpha value, I use the formula as in the function below

α` = new learning ratio

α = learning ratio

MaxEpoch = Maximum Iteration

α` = new learning ratio

Then the value of the new alpha is 0.9 - (0.9 \* 0.0000001) which is 0.89999991. This new alpha value will be used as the alpha value for the next iteration.

* 1. If all training data has been processed, then epoch = epoch +1. Then the epoch changes to 2.

1. Repeat step 3 and 4 until condition 4 is false

This step will repeat the steps until the condition Epoch < MaxEpoch or alpha > eps is false. The data used is training data. In this loop, the Ws and Wns values ​​will continue to be updated until the condition is false. If the condition is false (stopped) then the last Ws and Wns values ​​will be used for the weights on the testing data.

1. After step 5 is complete, do step 3 but from testing data. After that looking for J like 4b. To find the accuracy results then if:
   1. T class “1” and J class results is “1”, TP added 1
   2. T class “0” and J class results is “0”, TN added 1
   3. T class “1” and J class results is “0”, FN added 1
   4. T class “0” and J class results is “1”, FP added 1

In this step 6, we repeat step 3 which is to determine the value of X and the target. This value is obtained from the first testing data (id = 1). So X1=0, X2=1, X3=1 and so on and T=1. After that we calculate ||X-Ws|| and ||X-Wns|| where Ws and Wns are the final results of step 5. Then we will get the predicted value of class(J) by finding the minimum value between ||X-Ws|| and ||X-Wns||. After that we add the value of TP, TN, FN or FP according to the conditions. The addition of this value is the same as when the Naive Bayes algorithm.

1. Repeat step 6 for all testing datasets

In this step we repeat where the X and T data are testing data also for id = 2, id = 3 and so on until the last data from the testing data.

After step 8 is complete, do steps 1-8 with the amount of training data and testing data as shown in table 4.3. Then find the accuracy value of all LVQ tests that have been carried out using function 1. By getting the accuracy value of each LVQ test, the LVQ algorithm is complete.

Then the whole workflow process has also been completed. The accuracy results of the five Naive Bayes tests and the five LVQ tests were then compared. The accuracy of the Naive Bayes 1 test is compared to the accuracy of the LVQ 1 test, the accuracy of the 2 Naive Bayes test is compared to the 2 LVQ test and so on. The result of a better comparison is the sum of the better accuracy of each comparison.

The results of each test will also be seen. Are the 1,2,3,4 and 5 Naive Bayes tests the accuracy results much different or almost the same. Similarly, the results of the 1,2,3,4 and 5 LVQ tests are the accuracy results much different or almost the same.

# IMPLEMENTATION AND RESULTS

## Implementation

1. LOAD DATA LOCAL INFILE 'airline\_passenger\_satisfaction.csv'
2. INTO TABLE tbldata
3. FIELDS TERMINATED BY ','
4. ENCLOSED BY ''
5. LINES TERMINATED BY '\n'
6. IGNORE 1 LINES
7. (id,
8. Gender,
9. customer\_type,
10. …
11. arrival\_delay\_in\_minutes,
12. satisfaction)
13. SET
14. Gender = IF(Gender = '', null, Gender)
15. , customer\_type = IF(customer\_type = '', null, customer\_type)
16. , arrival\_delay\_in\_minutes = IF(arrival\_delay\_in\_minutes = '', null, arrival\_delay\_in\_minutes)
17. , satisfaction = IF(satisfaction = '', null, satisfaction);

Lines 1-2 to load the downloaded file into the 'tbldata' table with the following conditions. The provisions are as in lines 3-5 based on the csv file format. Row 6 to ignore the first row because the first row is the column heading. Lines 7-17 so that the data in the empty csv (null), when entered into the database remains empty (null).

1. INSERT INTO tbldataprocess
2. SELECT \* FROM tbldata;

Lines 18-19 are used to copy tbldata into tbldataprocess. This is so that tbldata has the exact same data as csv data. The program that will run later is taken from the tbldataprocess data.

1. DELIMITER ##
2. CREATE PROCEDURE preprocessing()
3. BEGIN
4. DECLARE i, iwhile, spinformation\_int INT DEFAULT 0;
5. DECLARE nama, spinformation, spinformation2 VARCHAR(255);
6. DELETE FROM tbldataprocess
7. WHERE
8. Gender IS NULL or
9. customer\_type IS NULL or
10. …
11. satisfaction IS NULL;
12. SET @num := 0;
13. UPDATE tbldataprocess SET id = @num := (@num+1);
14. ALTER TABLE tbldataprocess AUTO\_INCREMENT =1;
15. -- GENDER
16. SELECT count(DISTINCT gender) into i from tbldataprocess;
17. SET iwhile = 0;
18. WHILE iwhile<>i DO
19. SELECT DISTINCT gender INTO spinformation FROM tbldataprocess order by gender ASC limit iwhile, 1;
20. UPDATE tbldataprocess set gender=iwhile where gender=spinformation;
21. set iwhile= iwhile +1;
22. END WHILE ;
23. -- Customer Type
24. SELECT count(DISTINCT customer\_type) into i from tbldataprocess;
25. SET iwhile = 0;
26. WHILE iwhile <> i DO
27. SELECT DISTINCT customer\_type INTO spinformation FROM tbldataprocess order by customer\_type ASC limit iwhile, 1;
28. UPDATE tbldataprocess set customer\_type=iwhile where customer\_type = spinformation;
29. set iwhile= iwhile +1;
30. END WHILE ;
31. -- AGE
32. UPDATE tbldataprocess set age=0 where age <= 27;
33. UPDATE tbldataprocess set age=1 where age > 27 and age <= 51;
34. UPDATE tbldataprocess set age=2 where age > 51;
35. -- Type Of Travel
36. SELECT count(DISTINCT type\_of\_travel) into i from tbldataprocess;
37. SET iwhile = 0;
38. WHILE iwhile <> i DO
39. SELECT DISTINCT type\_of\_travel INTO spinformation FROM tbldataprocess order by type\_of\_travel ASC limit iwhile, 1;
40. UPDATE tbldataprocess set type\_of\_travel=iwhile where type\_of\_travel=spinformation;
41. set iwhile= iwhile +1;
42. END WHILE ;
43. -- Customer Class
44. SELECT count(DISTINCT customer\_class) into i from tbldataprocess;
45. SET iwhile = 0;
46. WHILE iwhile <> i DO
47. SELECT DISTINCT customer\_class into spinformation FROM tbldataprocess order by customer\_class ASC limit iwhile, 1;
48. UPDATE tbldataprocess set customer\_class=iwhile where customer\_class=spinformation;
49. set iwhile= iwhile +1;
50. END WHILE ;
51. -- FLIGHT DISTANCE
52. UPDATE tbldataprocess set flight\_distance=0 where flight\_distance <= 414;
53. UPDATE tbldataprocess set flight\_distance=1 where flight\_distance > 414 && flight\_distance <= 1744;
54. UPDATE tbldataprocess set flight\_distance=2 where flight\_distance > 1744;
55. -- Departure Delay In Minutes
56. UPDATE tbldataprocess set departure\_delay\_in\_minutes=0 where departure\_delay\_in\_minutes <= 12;
57. UPDATE tbldataprocess set departure\_delay\_in\_minutes=1 where departure\_delay\_in\_minutes > 12;
58. -- Arrival Delay In Minutes
59. UPDATE tbldataprocess set arrival\_delay\_in\_minutes=0 where arrival\_delay\_in\_minutes <= 13;
60. UPDATE tbldataprocess set arrival\_delay\_in\_minutes=1 where arrival\_delay\_in\_minutes > 13;
61. -- Satisfaction
62. SELECT count(DISTINCT satisfaction) into i from tbldataprocess;
63. SET iwhile = 0;
64. WHILE iwhile <> i DO
65. SELECT DISTINCT satisfaction into spinformation FROM tbldataprocess order by satisfaction ASC limit iwhile, 1;
66. UPDATE tbldataprocess set satisfaction=iwhile where satisfaction=spinformation;
67. set iwhile= iwhile +1;
68. END WHILE ;
69. END ##
70. DELIMITER ;

Lines 20-89 is a procedure that contains commands to perform preprocessing, namely removing null data and changing data. On lines 20-22 and 88-89 is the program code to create a procedure with the name of the procedure is preprocessing. Lines 23 and 24 to declare the variables that will be used in the procedure. Lines 26-31 are used to delete data that has a null value. Lines 31-33 so that the id attribute returns to order because there is an id that jumps after data is deleted. Lines 35-87 to change all data from tbldataprocess with the conditions as in table 4.2.

1. DELIMITER ##
2. CREATE PROCEDURE bayesian(number\_of\_testing INT)
3. BEGIN
4. DECLARE prob\_satisfied, prob\_gender\_s, prob\_customer\_type\_s,
5. prob\_age\_s, prob\_type\_of\_travel\_s, prob\_customer\_class\_s,
6. prob\_flight\_distance\_s, prob\_inflight\_wifi\_service\_s,
7. prob\_departure\_arrival\_time\_convenient\_s, prob\_ease\_of\_online\_booking\_s
8. , prob\_gate\_location\_s, prob\_food\_and\_drink\_s, prob\_online\_boarding\_s,
9. prob\_seat\_comfort\_s, prob\_inflight\_entertainment\_s,
10. prob\_onboard\_service\_s, prob\_leg\_room\_service\_s,
11. prob\_baggage\_handling\_s, prob\_checkin\_service\_s,
12. prob\_inflight\_service\_s, prob\_cleanliness\_s,
13. prob\_departure\_delay\_in\_minutes\_s, prob\_arrival\_delay\_in\_minutes\_s
14. FLOAT(30,30) DEFAULT 0;
15. DECLARE total\_satisfied, total\_notsatisfied FLOAT(30,20);
16. DECLARE prob\_notsatisfied, prob\_gender\_ns, prob\_customer\_type\_ns,
17. prob\_age\_ns, prob\_type\_of\_travel\_ns, prob\_customer\_class\_ns,
18. prob\_flight\_distance\_ns, prob\_inflight\_wifi\_service\_ns,
19. prob\_departure\_arrival\_time\_convenient\_ns,
20. prob\_ease\_of\_online\_booking\_ns, prob\_gate\_location\_ns,
21. prob\_food\_and\_drink\_ns, prob\_online\_boarding\_ns, prob\_seat\_comfort\_ns,
22. prob\_inflight\_entertainment\_ns, prob\_onboard\_service\_ns,
23. prob\_leg\_room\_service\_ns, prob\_baggage\_handling\_ns,
24. prob\_checkin\_service\_ns, prob\_inflight\_service\_ns, prob\_cleanliness\_ns,
25. prob\_departure\_delay\_in\_minutes\_ns, prob\_arrival\_delay\_in\_minutes\_ns
26. FLOAT(30,30) DEFAULT 0;
27. DECLARE prediksi\_s, prediksi\_ns FLOAT(30,30);
28. DECLARE i, testing\_ke, total\_training, total\_testing, i\_testing,
29. total\_data INT DEFAULT 0;
30. DECLARE info\_satisfaction VARCHAR(2);
31. SELECT COUNT(\*) INTO total\_data FROM tbldataprocess;
32. SET testing\_ke = number\_of\_testing;
33. IF testing\_ke = 1 THEN SET total\_training = 0.9 \* total\_data;
34. ELSEIF testing\_ke = 2 THEN SET total\_training = 0.75 \* total\_data;
35. ELSEIF testing\_ke = 3 THEN SET total\_training = 0.5 \* total\_data;
36. ELSEIF testing\_ke = 4 THEN SET total\_training = 0.25 \* total\_data;
37. ELSEIF testing\_ke = 5 THEN SET total\_training = 0.1 \* total\_data;
38. END IF;
39. SET total\_testing = total\_data-total\_training;
40. SET i\_testing = 1;
41. UPDATE tblaccuracy SET total\_data=0, tp=0, tn=0, fp=0, fn=0,tnull=0,fnull=0 WHERE algoritma = 'Bayesian' AND testing = testing\_ke;
42. UPDATE tblaccuracy SET total\_data\_training=total\_training, total\_data = total\_testing WHERE testing = testing\_ke AND algoritma = "Bayesian";
43. TRUNCATE tbldatatesting;
44. TRUNCATE tbldatatraining;
45. INSERT INTO tbldatatraining ( Gender, customer\_type, age, type\_of\_travel, customer\_class, flight\_distance, inflight\_wifi\_service, departure\_arrival\_time\_convenient, ease\_of\_online\_booking, gate\_location, food\_and\_drink, online\_boarding, seat\_comfort, inflight\_entertainment, onboard\_service, leg\_room\_service, baggage\_handling, checkin\_service, inflight\_service, cleanliness, departure\_delay\_in\_minutes, arrival\_delay\_in\_minutes, satisfaction)
46. SELECT Gender, customer\_type, age, type\_of\_travel, customer\_class, flight\_distance, inflight\_wifi\_service, departure\_arrival\_time\_convenient, ease\_of\_online\_booking, gate\_location, food\_and\_drink, online\_boarding, seat\_comfort, inflight\_entertainment, onboard\_service, leg\_room\_service, baggage\_handling, checkin\_service, inflight\_service, cleanliness, departure\_delay\_in\_minutes, arrival\_delay\_in\_minutes, satisfaction
47. FROM tbldataprocess where id<= total\_training;
48. INSERT INTO tbldatatesting ( Gender, customer\_type, age, type\_of\_travel, customer\_class, flight\_distance, inflight\_wifi\_service, departure\_arrival\_time\_convenient, ease\_of\_online\_booking, gate\_location, food\_and\_drink, online\_boarding, seat\_comfort, inflight\_entertainment, onboard\_service, leg\_room\_service, baggage\_handling, checkin\_service, inflight\_service, cleanliness, departure\_delay\_in\_minutes, arrival\_delay\_in\_minutes, satisfaction)
49. SELECT Gender, customer\_type, age, type\_of\_travel, customer\_class, flight\_distance, inflight\_wifi\_service, departure\_arrival\_time\_convenient, ease\_of\_online\_booking, gate\_location, food\_and\_drink, online\_boarding, seat\_comfort, inflight\_entertainment, onboard\_service, leg\_room\_service, baggage\_handling, checkin\_service, inflight\_service, cleanliness, departure\_delay\_in\_minutes, arrival\_delay\_in\_minutes, satisfaction
50. FROM tbldataprocess WHERE
51. id > total\_training
52. AND id <= ( total\_training + total\_testing);
53. SET total\_satisfied = (SELECT count(satisfaction) FROM tbldatatraining WHERE satisfaction = 1);
54. SET total\_notsatisfied = (SELECT count(satisfaction) FROM tbldatatraining WHERE satisfaction = 0);
55. SET prob\_satisfied = total\_satisfied / total\_training;
56. SET prob\_notsatisfied = total\_notsatisfied / total\_training;
57. -- WHILE per row tbldatatesting
58. WHILE i\_testing <= total\_testing DO
59. -- GENDER
60. SET prob\_gender\_s = (SELECT count(gender) FROM tbldatatraining WHERE gender=(SELECT gender FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
61. SET prob\_gender\_ns = (SELECT count(gender) FROM tbldatatraining WHERE gender=(SELECT gender FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
62. -- Customer Type
63. SET prob\_customer\_type\_s = (SELECT count(customer\_type) FROM tbldatatraining WHERE customer\_type=(SELECT customer\_type FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
64. SET prob\_customer\_type\_ns = (SELECT count(customer\_type) FROM tbldatatraining WHERE customer\_type=(SELECT customer\_type FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
65. …
66. -- Arrival Delay In Minutes
67. SET prob\_arrival\_delay\_in\_minutes\_s = (SELECT count(arrival\_delay\_in\_minutes) FROM tbldatatraining WHERE arrival\_delay\_in\_minutes=(SELECT arrival\_delay\_in\_minutes FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
68. SET prob\_arrival\_delay\_in\_minutes\_ns = (SELECT count(arrival\_delay\_in\_minutes) FROM tbldatatraining WHERE arrival\_delay\_in\_minutes=(SELECT arrival\_delay\_in\_minutes FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
69. SET prediksi\_s = prob\_satisfied\* prob\_gender\_s\* prob\_customer\_type\_s\* prob\_age\_s\* prob\_type\_of\_travel\_s\* prob\_customer\_class\_s\* prob\_flight\_distance\_s\* prob\_inflight\_wifi\_service\_s\* prob\_departure\_arrival\_time\_convenient\_s\* prob\_ease\_of\_online\_booking\_s\* prob\_gate\_location\_s\* prob\_food\_and\_drink\_s\* prob\_online\_boarding\_s\* prob\_seat\_comfort\_s\* prob\_inflight\_entertainment\_s\* prob\_onboard\_service\_s\* prob\_leg\_room\_service\_s\* prob\_baggage\_handling\_s\* prob\_checkin\_service\_s\* prob\_inflight\_service\_s\* prob\_cleanliness\_s\* prob\_departure\_delay\_in\_minutes\_s\* prob\_arrival\_delay\_in\_minutes\_s;
70. SET prediksi\_ns =prob\_notsatisfied\* prob\_gender\_ns\* prob\_customer\_type\_ns\* prob\_age\_ns\* prob\_type\_of\_travel\_ns\* prob\_customer\_class\_ns\* prob\_flight\_distance\_ns\* prob\_inflight\_wifi\_service\_ns\* prob\_departure\_arrival\_time\_convenient\_ns\* prob\_ease\_of\_online\_booking\_ns\* prob\_gate\_location\_ns\* prob\_food\_and\_drink\_ns\* prob\_online\_boarding\_ns\* prob\_seat\_comfort\_ns\* prob\_inflight\_entertainment\_ns\* prob\_onboard\_service\_ns\* prob\_leg\_room\_service\_ns\* prob\_baggage\_handling\_ns\* prob\_checkin\_service\_ns\* prob\_inflight\_service\_ns\* prob\_cleanliness\_ns\* prob\_departure\_delay\_in\_minutes\_ns\* prob\_arrival\_delay\_in\_minutes\_ns;
71. SELECT satisfaction INTO info\_satisfaction from tbldatatesting where id = i\_testing;
72. IF info\_satisfaction = 0 THEN -- actual not satisfied
73. IF prediksi\_s < prediksi\_ns THEN
74. UPDATE tblaccuracy SET tn=tn+1 WHERE algoritma="Bayesian" AND
75. testing=testing\_ke;
76. ELSEIF prediksi\_s > prediksi\_ns THEN
77. UPDATE tblaccuracy SET fp=fp+1 WHERE algoritma="Bayesian" AND
78. testing=testing\_ke;
79. ELSEIF prediksi\_s = 0 AND prediksi\_ns = 0 THEN
80. UPDATE tblaccuracy SET fnull=fnull+1 WHERE algoritma="Bayesian" AND
81. testing=testing\_ke;
82. END IF;
83. ELSEIF info\_satisfaction = 1 THEN -- actual satisfied
84. IF prediksi\_s < prediksi\_ns THEN
85. UPDATE tblaccuracy SET fn=fn+1 WHERE algoritma="Bayesian" AND
86. testing=testing\_ke;
87. ELSEIF prediksi\_s > prediksi\_ns THEN
88. UPDATE tblaccuracy SET tp=tp+1 WHERE algoritma="Bayesian" AND
89. testing=testing\_ke;
90. ELSEIF prediksi\_s = 0 AND prediksi\_ns = 0 THEN
91. UPDATE tblaccuracy SET tnull=tnull+1 WHERE algoritma="Bayesian" AND
92. testing=testing\_ke;
93. END IF;
94. END IF;
95. SET i\_testing = i\_testing+1;
96. END WHILE;
97. END ##
98. DELIMITER ;

Lines 90-92 and 201-202 are creating a procedure with a bayesian name that will perform the Naive Bayes algorithm. This procedure has parameters to determine how many tests. Lines 93-103 to declare the variable used to store the result P(gender=1|label=a). Line 104 to store the values ​​P(gender=1) and P(gender=0). While lines 106-116 for the results P(gender=0|label=a).

On lines 122-131 to determine the amount of distribution of training and testing data. Then it will be entered into tbldatatraining and tbldatatesting as much as the previous amount. The data entered comes from tbldataprocess.

In lines 156 and 200-201 are repetitions for tbldatatesting which include steps 3-6 Naive Bayes. This iteration is the 7th step of Naive Bayes. In lines 158-169 calculate the probability of P(a|x) for both class/label x that is "satisfied" and "neutral or dissatisfied" from all input attributes.

Lines 171-172 to get the result of multiplying P(a|x) all attributes and P(x) as in step 4 of Naive Bayes. Then 174-201 to get the prediction result and also add tp, tn, fn or fp to tblaccuracy. Here I also add tnull and fnull to see if there are any unpredictable tests.

1. DELIMITER ##
2. CREATE PROCEDURE processb()
3. BEGIN
4. CALL bayesian(1);
5. CALL bayesian(2);
6. CALL bayesian(3);
7. CALL bayesian(4);
8. CALL bayesian(5);
9. UPDATE tblaccuracy SET accuracy = ((tp+tn)/(tp+tn+fp+fn+tnull+fnull))\*100 where algoritma='Bayesian';
10. END ##
11. DELIMITER ;

On lines 205-215 this is a procedure used to call a Bayesian procedure with parameters 1-5. These parameters indicate how many tests to determine the amount of training and testing data. Then after running the Naive Bayes algorithm 5 times, the accuracy value will be searched on line 213.

1. DELIMITER ##
2. CREATE FUNCTION ed(
3. w1t FLOAT(30,20)
4. , w1 FLOAT(30,20)
5. ……
6. , w22t FLOAT(30,20)
7. , w22 FLOAT(30,20)
8. )
9. RETURNS FLOAT(30,20)
10. BEGIN
11. DECLARE hasil FLOAT(30,20) DEFAULT 0;
12. SET hasil = SQRT((
13. POWER((w1t - w1),2)
14. + POWER((w2t - w2),2)
15. + POWER((w3t - w3),2)
16. ……
17. + POWER((w22t - w22),2)));
18. RETURN(hasil);
19. END; ##
20. DELIMITER ;

This line 216-235 is the code for creating the ed function. This function returns a Euclidian distance value from the given parameter. This function will be used during the LVQ algorithm.

1. DELIMITER ##
2. CREATE PROCEDURE lvq(number\_of\_testing INT, pAlpha FLOAT(30,20), pEps FLOAT(30,20))
3. BEGIN
4. -- Weight of class satisfied
5. DECLARE w1s, w2s, w3s, w4s, w5s, w6s, w7s, w8s, w9s, w10s, w11s, w12s, w13s, w14s, w15s, w16s, w17s, w18s, w19s, w20s, w21s, w22s, w23s FLOAT(30,20) DEFAULT 0;
6. -- Weight of class neutral or dissatisfied
7. DECLARE w1ns, w2ns, w3ns, w4ns, w5ns, w6ns, w7ns, w8ns, w9ns, w10ns, w11ns, w12ns, w13ns, w14ns, w15ns, w16ns, w17ns, w18ns, w19ns, w20ns, w21ns, w22ns ,w23ns FLOAT(30,20) DEFAULT 0;
8. -- Weight of training
9. DECLARE w1t, w2t, w3t, w4t, w5t, w6t ,w7t, w8t, w9t, w10t, w11t, w12t, w13t, w14t, w15t, w16t, w17t, w18t, w19t, w20t, w21t, w22t, w23t FLOAT(30,20) DEFAULT 0;
10. -- Get id for initial class satisfied and not
11. DECLARE ids, idns, cj, t, epoch, maxepoch,pepoch INT DEFAULT 0;
12. DECLARE tn\_lvq, fp\_lvq, fn\_lvq, tp\_lvq INT DEFAULT 0;
13. DECLARE ws, wns, wt FLOAT(30,20) DEFAULT 0;
14. DECLARE alpha, eps, err, temp\_alpha FLOAT(30,20) DEFAULT 0;
15. DECLARE info\_satisfaction VARCHAR(2);
16. DECLARE prediction INT DEFAULT 0;
17. DECLARE i, testing\_ke, total\_training, total\_testing, i\_testing, i\_training, total\_data INT DEFAULT 0;
18. SELECT COUNT(\*) INTO total\_data FROM tbldataprocess;
19. SET testing\_ke = number\_of\_testing;
20. SET maxepoch=5;
21. IF testing\_ke = 1 THEN SET total\_training = 0.9 \* total\_data;
22. ELSEIF testing\_ke = 2 THEN SET total\_training = 0.75 \* total\_data;
23. ELSEIF testing\_ke = 3 THEN SET total\_training = 0.5 \* total\_data;
24. ELSEIF testing\_ke = 4 THEN SET total\_training = 0.25 \* total\_data;
25. ELSEIF testing\_ke = 5 THEN SET total\_training = 0.1 \* total\_data;
26. END IF;
27. TRUNCATE tbldatatesting;
28. TRUNCATE tbldatatraining;
29. SET total\_testing = total\_data-total\_training;
30. SET i\_testing = 1;
31. SET i\_training = 1;
32. INSERT INTO tbldatatraining ( Gender, customer\_type, age, type\_of\_travel, customer\_class, flight\_distance, inflight\_wifi\_service, departure\_arrival\_time\_convenient, ease\_of\_online\_booking, gate\_location, food\_and\_drink, online\_boarding, seat\_comfort, inflight\_entertainment, onboard\_service, leg\_room\_service, baggage\_handling, checkin\_service, inflight\_service, cleanliness, departure\_delay\_in\_minutes, arrival\_delay\_in\_minutes, satisfaction)
33. SELECT Gender, customer\_type, age, type\_of\_travel, customer\_class, flight\_distance, inflight\_wifi\_service, departure\_arrival\_time\_convenient, ease\_of\_online\_booking, gate\_location, food\_and\_drink, online\_boarding, seat\_comfort, inflight\_entertainment, onboard\_service, leg\_room\_service, baggage\_handling, checkin\_service, inflight\_service, cleanliness, departure\_delay\_in\_minutes, arrival\_delay\_in\_minutes, satisfaction
34. FROM tbldataprocess where id<= total\_training;
35. INSERT INTO tbldatatesting ( Gender, customer\_type, age, type\_of\_travel, customer\_class, flight\_distance, inflight\_wifi\_service, departure\_arrival\_time\_convenient, ease\_of\_online\_booking, gate\_location, food\_and\_drink, online\_boarding, seat\_comfort, inflight\_entertainment, onboard\_service, leg\_room\_service, baggage\_handling, checkin\_service, inflight\_service, cleanliness, departure\_delay\_in\_minutes, arrival\_delay\_in\_minutes, satisfaction)
36. SELECT Gender, customer\_type, age, type\_of\_travel, customer\_class, flight\_distance, inflight\_wifi\_service, departure\_arrival\_time\_convenient, ease\_of\_online\_booking, gate\_location, food\_and\_drink, online\_boarding, seat\_comfort, inflight\_entertainment, onboard\_service, leg\_room\_service, baggage\_handling, checkin\_service, inflight\_service, cleanliness, departure\_delay\_in\_minutes, arrival\_delay\_in\_minutes, satisfaction
37. FROM tbldataprocess WHERE
38. id > total\_training
39. AND id <= ( total\_training + total\_testing);
40. INSERT INTO tblaccuracy(algoritma, testing, total\_data\_training, total\_data, total\_training,tp, tn, fp,fn,tnull, fnull, accuracy) VALUES('LVQ', testing\_ke,0,0,0,0,0,0,0,0,0,0);
41. -- INITIALITATION
42. SET alpha = pAlpha;
43. SET eps = pEps;
44. SELECT id INTO ids FROM tbldatatraining WHERE satisfaction = 1 ORDER BY RAND() LIMIT 1;
45. SELECT id INTO idns FROM tbldatatraining WHERE satisfaction = 0 ORDER BY RAND() LIMIT 1;
46. SELECT gender INTO w1s FROM tbldatatraining WHERE id=ids;
47. SELECT gender INTO w1ns FROM tbldatatraining WHERE id=idns;
48. ……
49. SELECT arrival\_delay\_in\_minutes INTO w22s FROM tbldatatraining WHERE id=ids;
50. SELECT arrival\_delay\_in\_minutes INTO w22ns FROM tbldatatraining WHERE id=idns;
51. -- row used to initialitation is not use again
52. DELETE FROM tbldatatraining WHERE id=ids;
53. DELETE FROM tbldatatraining WHERE id=idns;
54. SET @num := 0;
55. UPDATE tbldatatraining SET id = @num := (@num+1);
56. ALTER TABLE tbldatatraining AUTO\_INCREMENT =1;
57. -- END INITIALITATION
58. -- TRAINING
59. SET temp\_alpha = alpha;
60. WHILE epoch < maxepoch DO
61. SET i\_training = 0;
62. SET temp\_i\_training = 0;
63. SET alpha=temp\_alpha;
64. algolvq: WHILE (i\_training <= total\_training) or (alpha >= eps) DO
65. IF(alpha>=eps)THEN
66. SELECT gender INTO w1t FROM tbldatatraining WHERE id=i\_training;
67. ……
68. SELECT satisfaction INTO w23t FROM tbldatatraining WHERE id=i\_training;
69. SET ws = ed(w1t, w1s, w2t, w2s, w3t, w3s, w4t, w4s, w5t, w5s, w6t, w6s, w7t, w7s, w8t, w8s, w9t, w9s, w10t, w10s, w11t, w11s, w12t, w12s, w13t, w13s, w14t, w14s, w15t, w15s, w16t, w16s, w17t, w17s, w18t, w18s, w19t, w19s, w20t, w20s, w21t, w21s, w22t, w22s);
70. SET wns = ed(w1t, w1ns, w2t, w2ns, w3t, w3ns, w4t, w4ns, w5t, w5ns, w6t, w6ns, w7t, w7ns, w8t, w8ns, w9t, w9ns, w10t, w10ns, w11t, w11ns, w12t, w12ns, w13t, w13ns, w14t, w14ns, w15t, w15ns, w16t, w16ns, w17t, w17ns, w18t, w18ns, w19t, w19ns, w20t, w20ns, w21t, w21ns, w22t, w22ns);
71. IF ws < wns THEN
72. SET cj = 1;
73. ELSEIF ws > wns THEN
74. SET cj = 0;
75. ELSE
76. SET cj = 1;
77. END IF;
78. SELECT satisfaction INTO t FROM tbldatatraining WHERE id=i\_training;
79. IF cj = 1 AND t = 1 THEN
80. SET w1s = w1s + (alpha \* (w1t - w1s));
81. ……
82. SET w22s = w22s + (alpha \* (w22t - w22s));
83. ELSEIF cj = 0 AND t = 0 THEN
84. SET w1ns = w1ns + (alpha \* (w1t - w1ns));
85. ……
86. SET w22ns = w22ns + (alpha \* (w22t - w22ns));
87. ELSEIF cj = 0 AND t = 1 THEN
88. SET w1ns = w1ns - (alpha \* (w1t - w1ns));
89. ……
90. SET w22ns = w22ns - (alpha \* (w22t - w22ns));
91. ELSEIF cj = 1 AND t = 0 THEN
92. SET w1s = w1s - (alpha \* (w1t - w1s));
93. ……
94. SET w22s = w22s - (alpha \* (w22t - w22s));
95. END IF;
96. SET alpha = alpha - (alpha \* eps);
97. UPDATE tblaccuracy SET total\_data\_training = i\_training+1 WHERE id=(SELECT COUNT(\*) FROM tblaccuracy);
98. SET i\_training = i\_training + 1;
99. SET temp\_i\_training = i\_training;
100. IF (i\_training = total\_training) THEN
101. SET temp\_alpha = alpha;
102. SET alpha = eps;
103. ELSEIF (alpha <= eps) THEN
104. SET temp\_alpha = alpha;
105. SELECT alpha as 'alphaa', i\_training as 't';
106. SET i\_training = total\_training +1;
107. END IF;
108. ELSE
109. LEAVE algolvq;
110. END IF;
111. END WHILE;
112. SET epoch = epoch + 1;
113. IF (temp\_i\_training <> 0) THEN
114. SET ptotal\_data\_training=ptotal\_data\_training + temp\_i\_training;
115. SET pepoch=epoch;
116. END IF;
117. END WHILE;
118. WHILE i\_testing <= total\_testing DO
119. SELECT gender INTO w1t FROM tbldatatesting WHERE id=i\_testing;
120. ……
121. SELECT arrival\_delay\_in\_minutes INTO w22t FROM tbldatatesting WHERE id=i\_testing;
122. SELECT satisfaction INTO w23t FROM tbldatatesting WHERE id=i\_testing;
123. SET ws = ed(w1t, w1s, w2t, w2s, w3t, w3s, w4t, w4s, w5t, w5s, w6t, w6s, w7t, w7s, w8t, w8s, w9t, w9s, w10t, w10s, w11t, w11s, w12t, w12s, w13t, w13s, w14t, w14s, w15t, w15s, w16t, w16s, w17t, w17s, w18t, w18s, w19t, w19s, w20t, w20s, w21t, w21s, w22t, w22s);
124. SET wns = ed(w1t, w1ns, w2t, w2ns, w3t, w3ns, w4t, w4ns, w5t, w5ns, w6t, w6ns, w7t, w7ns, w8t, w8ns, w9t, w9ns, w10t, w10ns, w11t, w11ns, w12t, w12ns, w13t, w13ns, w14t, w14ns, w15t, w15ns, w16t, w16ns, w17t, w17ns, w18t, w18ns, w19t, w19ns, w20t, w20ns, w21t, w21ns, w22t, w22ns);
125. SELECT satisfaction INTO info\_satisfaction FROM tbldatatesting WHERE id=i\_testing;
126. IF ws < wns THEN SET prediction = 1;
127. ELSEIF ws > wns THEN SET prediction = 0;
128. ELSE SET prediction = 1;
129. END IF;
130. IF info\_satisfaction = 0 AND prediction=0 THEN
131. SET tn\_lvq=tn\_lvq+1;
132. ELSEIF info\_satisfaction = 0 AND prediction=1 THEN
133. SET fp\_lvq=fp\_lvq+1;
134. ELSEIF info\_satisfaction = 1 AND prediction=0 THEN
135. SET fn\_lvq=fn\_lvq+1;
136. ELSEIF info\_satisfaction = 1 AND prediction=1 THEN
137. SET tp\_lvq=tp\_lvq+1;
138. END IF;
139. SET i\_testing = i\_testing + 1;
140. END WHILE;
141. UPDATE tblaccuracy SET total\_data=total\_testing, tn=tn\_lvq, fp=fp\_lvq, fn=fn\_lvq, tp=tp\_lvq WHERE id = (SELECT count(\*) FROM tblaccuracy);
142. END ##
143. DELIMITER ;

Lines 236-398 are the procedures in which the LVQ algorithm is executed. This procedure has 3 parameters, namely testing to how much, alpha and eps. The first parameter is used to determine the distribution of the amount of training and testing data. While alpha and eps are used to simplify the analysis by replacing the two values.

Line 240 is a variable declaration for label weight 1 while 242 is for label weight 2. In line 244 is a variable for the value of the input weight. Next 245-255 is the variable declaration used in the lvq procedure.

Lines 257-265 are for dividing the amount of training data and testing data. Furthermore, on lines 266-279 will enter the data from tbldataprocess into tbldatatraining and tbldatatesting the amount that has been obtained earlier. After dividing the data, then entering the data into tblaccuracy for testing that is being carried out on line 281.

Then do the initialization step as in line 283-302. Initial weights for label 0 and label 1 are chosen randomly as in lines 287-288. Then enter into the variables for the 22 input attributes of the two classes as in lines 290-294. After being stored in the data variable that has been used as the initial weight, it is deleted on lines 296-297. the id from tbldatatraining is updated again so that no id jumps because it has been deleted as in lines 299-301.

In lines 305-351 do repetitions for the training process. At each repetition of the training process, the Euclidian distance value for labels 0 and 1 is searched using the ed function as in lines 306-311. After that, the prediction result (J) is determined by looking for the minimum value in lines 313-319. After that the weight value will be updated on lines 321-339. After updating the weights, the alpha value is also updated on line 341. In lines 343-349 it is used to make the loop condition false and exit the loop.

After completing the training repetition, the final weight of the training is obtained. These weights will be used in the iteration of lines 352-377 which is the iteration for all testing data from tbldatatesting. In each of these iterations get the input values ​​as in lines 353-356 for all input attributes. After that, look for the value of the Euclidian distance for the two labels as in lines 358-360. The minimum value of the two Euclidian distances is the prediction result. The value of tn, fp, fn or tp will be added by 1 if it is in accordance with the provisions. After adding this value, it will then repeat for the next testing data until all the data is tested. The final results of tn, fp, fn and tp will be updated to tblaccuracy.

1. DELIMITER ##
2. CREATE PROCEDURE processl(alpha FLOAT(30,20), eps FLOAT(30,20))
3. BEGIN
4. CALL lvq(1,alpha,eps);
5. CALL lvq(2,alpha,eps);
6. CALL lvq(3,alpha,eps);
7. CALL lvq(4,alpha,eps);
8. CALL lvq(5,alpha,eps);
9. UPDATE tblaccuracy SET accuracy = ((tp+tn)/(tp+tn+fp+fn))\*100 WHERE algoritma="LVQ";
10. END ##
11. DELIMITER ;

In lines 382-392 this is a procedure to run the lvq procedure 5 times. After 5 tests with different amounts of training and testing data, the accuracy value will be calculated as in line 390.

1. DELIMITER ##
2. CREATE PROCEDURE process()
3. BEGIN
4. CALL processb();
5. CALL processl(0.9,0.0000001);
6. CALL processl(0.9,0.0001);
7. CALL processl(0.9,0.01);
8. CALL processl(0.1,0.0000001);
9. CALL processl(0.1,0.0001);
10. CALL processl(0.1,0.01);
11. CALL processl(0.01,0.0000001);
12. CALL processl(0.01,0.0001);
13. CALL processl(0.05,0.0000001);
14. CALL processl(0.05,0.0001);
15. END ##
16. DELIMITER ;

On lines 393-408 these are Naive Bayes and LVQ procedures. On line 396 run the procedure processb which will perform all 5 Naive Bayes tests. Next it will run the 5 LVQ tests by calling the processl procedure. In running the process, it is done several times with different alpha and eps values.

## Results

From the results of the trials that have been carried out, Naive Bayes and LVQ can be used to determine airline passenger satisfaction. The results of the trial run for almost 4 days. The longest test run when running Naive Bayes is around 3 days and 3 hours. The results of the program that runs the Naive Bayes algorithm are as follows:

Naive Bayes Results

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test | TP | TN | FP | FN | Tnull | Fnull | Accuracy |
| 1 | 4992 | 6485 | 756 | 716 | 0 | 0 | 88.63% |
| 2 | 12438 | 1822 | 1822 | 1735 | 0 | 0 | 89.01% |
| 3 | 24728 | 3565 | 3565 | 3433 | 0 | 0 | 89.18% |
| 4 | 37089 | 5427 | 5427 | 5039 | 1 | 0 | 89.22% |
| 5 | 44629 | 6444 | 6444 | 5965 | 1 | 10 | 89.34% |

From the table above, Naive Bayes can determine quite a lot of predictions that match the original class. This can be seen from the accuracy obtained between the range of 88-89%. The average of this test is 89.076%. However, in tests 4 and 5 there are some testing data whose prediction results are unknown. This is probably due to the large number of input attributes, namely 22 attributes, when multiplied all the results cannot be saved by the computer. Because the program can only store a maximum of 30 digits behind the comma. So it is possible that the result is 0.0 for both predictions and no conclusions can be drawn from the prediction results.

Furthermore, the LVQ algorithm can also be implemented for this airline's passenger satisfaction data. Because there are alpha and eps parameters that do not have an exact value, this study tries to use several kinds of values. For the alpha value, try 4 variations of the value, namely 0.9; 0.1; 0.01; and 0.05. While for eps there are only 3 variations, namely 0.0000001; 0.0001; and 0.01. The results of the LVQ implementation of these variations are shown in the tables below.

Results LVQ Alpha 0.9 and Eps 0.0000001, 0.0001, 0.01

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | TP | TN | FP | FN | Accuracy |
| 1 | 0 | 7421 | 0 | 5708 | 55.92% |
| 2 | 0 | 18198 | 0 | 14173 | 56.22% |
| 3 | 0 | 36572 | 0 | 28171 | 56.49% |
| 4 | 0 | 54985 | 0 | 42129 | 56.62% |
| 5 | 0 | 65942 | 0 | 50595 | 56.58% |

Results LVQ Alpha 0.1 and Eps 0.0000001

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | TP | TN | FP | FN | Accuracy |
| 1 | 3802 | 6406 | 835 | 1906 | 78.83% |
| 2 | 10893 | 14777 | 3421 | 3280 | 79.30% |
| 3 | 22117 | 28842 | 7730 | 6054 | 79.71% |
| 4 | 34469 | 40371 | 14614 | 7660 | 77.06% |
| 5 | 42421 | 47930 | 18012 | 8174 | 77.53% |

Results LVQ Alpha 0.1 and Eps 0.0001

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | TP | TN | FP | FN | Accuracy |
| 1 | 4405 | 5841 | 1400 | 1303 | 79.13% |
| 2 | 10963 | 14652 | 3546 | 3210 | 79.13% |
| 3 | 21824 | 29496 | 7076 | 6347 | 79.27% |
| 4 | 33133 | 43881 | 11104 | 8996 | 79.30% |
| 5 | 39977 | 52072 | 13870 | 10618 | 78.99% |

Results LVQ Alpha 0.1 and Eps 0.01

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | TP | TN | FP | FN | Accuracy |
| 1 | 4412 | 5904 | 1337 | 1296 | 79.67% |
| 2 | 9998 | 15383 | 2815 | 4175 | 78.41% |
| 3 | 21823 | 29752 | 6820 | 6348 | 79.66% |
| 4 | 31784 | 45028 | 9957 | 10345 | 79.09% |
| 5 | 38662 | 54703 | 11239 | 11933 | 80.12% |

Results LVQ Alpha 0.01 and Eps 0.0000001

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | TP | TN | FP | FN | Accuracy |
| 1 | 4252 | 6041 | 1200 | 1456 | 79.49% |
| 2 | 10910 | 14788 | 3410 | 3263 | 79.39% |
| 3 | 21900 | 29310 | 7262 | 6271 | 79.10% |
| 4 | 33299 | 43552 | 11433 | 8830 | 79.13% |
| 5 | 39884 | 51750 | 14192 | 10711 | 78.63% |

Results LVQ Alpha 0.01 and Eps 0.0001

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | TP | TN | FP | FN | Accuracy |
| 1 | 4414 | 5825 | 1416 | 1294 | 79.07% |
| 2 | 10984 | 14616 | 3582 | 3189 | 79.08% |
| 3 | 21869 | 29452 | 7120 | 6302 | 79.27% |
| 4 | 32772 | 44270 | 10715 | 9357 | 79.33% |
| 5 | 39688 | 52322 | 13620 | 10907 | 78.95% |

Results LVQ Alpha 0.05 and Eps 0.0000001

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | TP | TN | FP | FN | Accuracy |
| 1 | 3797 | 6522 | 719 | 1911 | 79.69% |
| 2 | 10809 | 14977 | 3221 | 3364 | 79.66% |
| 3 | 21713 | 29818 | 6754 | 6458 | 79.59% |
| 4 | 33943 | 41848 | 13137 | 8186 | 78.04% |
| 5 | 40709 | 51609 | 14333 | 9886 | 79.22% |

Results LVQ Alpha 0.05 and Eps 0.0001

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | TP | TN | FP | FN | Accuracy |
| 1 | 4407 | 5843 | 1398 | 1301 | 79.16% |
| 2 | 10964 | 14657 | 3541 | 3209 | 79.15% |
| 3 | 21824 | 29522 | 7050 | 6347 | 79.31% |
| 4 | 32969 | 44054 | 10931 | 9160 | 79.31% |
| 5 | 39887 | 51730 | 14212 | 10708 | 78.62% |

Of the 10 variations and each variation remains 5 times the results of the accuracy test also vary. In table 5.2 where alpha 0.9 and different eps the prediction results and accuracy have the same value even though the eps is changed. The average accuracy is 56.366%. But if you look at the Tp values, all are 0. This means that testing with an alpha of 0.9 will produce all "neutral or dissatisfied" predictions. Of course it cannot be said to be predicting the outcome because 100% of the predicted results are labeled "neutral or dissatisfied". However, when alpha 0.1 results are quite good, this result can be said to be predicting the label. From the experiments that have been done, the best results are when alpha 0.1 and eps 0.01. The average accuracy is 79.39%. Because the average accuracy is the best, it will be used when compared to the Naive Bayes algorithm.

So, if based on the accuracy value, Naive Bayes is better than LVQ where Naive Bayes has an average accuracy of 89.076% while LVQ is 79.39%. However, from the experimental results, the drawback of Naive Bayes is the processing time for the 5 tests, which is more than 3 days. Meanwhile, the LVQ for 5 tests with training and testing data differs by an average of 30 minutes. However, it is necessary to find the optimal value for LVQ through alpha and eps values.

If the results of the accuracy of the LVQ algorithm in graphical form will be as below.

Graph of accuracy LVQ alpha=0.9

Graph of accuracy LVQ alpha=0.1 and eps=0.0000001

Graph of accuracy LVQ alpha=0.1 and eps=0.0001

Graph of accuracy alpha=0.1 and eps=0.01

Graph of accuracy alpha=0.01 and eps=0.0000001

Graph of accuracy alpha=0.01 and eps=0.0001

Graph of accuracy alpha=0.05 and eps=0.0000001

Graph of accuracy alpha=0.05 and eps=0.0001

When viewed from each graph of the accuracy of the LVQ algorithm, it has a different curve. This is of course because of the influence of the alpha and eps values. But it is possible because of the quality of the data itself.

# CONCLUSION

Based on the results of the tests that have been carried out, the following conclusions can be drawn:

1. Naive Bayes can be used to classify airline passenger satisfaction. It is proven that the Naive Bayes algorithm can be implemented for customer satisfaction data obtained through Kaggle.

2. Learning Vector Quantization can also classify airline passenger satisfaction. This is because this algorithm can be implemented on the same data to implement Naive Bayes.

3. Of the two algorithms, Naive Bayes is better at classifying airline passenger satisfaction than Learning Vector Quantization. This is based on the average accuracy of the two algorithms. Naive Bayes has an average accuracy of 89.076% while the LVQ is 79.39%.

Suggestions for further research is to focus on one algorithm, namely Learning Vector Quantization (LVQ). With the same data, LVQ is implemented but with the aim of finding the optimal value. What is the most optimal combination of alpha and eps so that it can get the best accuracy.

REFERENCES

[1] M. B. Gorzałczany, F. Rudziński, and J. Piekoszewski, “Business Intelligence in Airline Passenger Satisfaction Study—A Fuzzy-Genetic Approach with Optimized Interpretability-Accuracy Trade-Off,” *Applied Sciences*, vol. 11, no. 11, p. 5098, May 2021, doi: 10.3390/app11115098.

[2] B. H. Hayadi, J.-M. Kim, H. T. Sukmana, and K. Hulliyah, “Predicting Airline Passenger Satisfaction with Classification Algorithms,” *IJIIS : Int. J. Inform. Inform. Systems.*, vol. 4, no. 1, pp. 82–94, Mar. 2021, doi: 10.47738/ijiis.v4i1.80.

[3] M. R. Hanif, R. Chrizara, E. Saribanon, and I. Ozali, “THE EFFECT OF SERVICE QUALITY AND PASSENGER SATISFACTION ON PASSENGER BEHAVIORAL INTENTIONS ON LION AIR,” p. 9.

[4] A. Wijayanto, J. F. A. Bernardo, and S. Pamungkas, “Analisis Klasifikasi Kepuasan Penumpang Maskapai Penerbangan Menggunakan Algoritma Naïve Bayes,” *J-SAKTI*, vol. 3, no. 2, pp. 97–103, May 2021, doi: 10.33084/jsakti.v3i2.2041.

[5] Yoga Religia and A. Amali, “Perbandingan Optimasi Feature Selection pada Naïve Bayes untuk Klasifikasi Kepuasan Airline Passenger,” *RESTI*, vol. 5, no. 3, pp. 527–533, Jun. 2021, doi: 10.29207/resti.v5i3.3086.

[6] P. Aditya Nugroho, R. Saptono, and M. Eko Sulistyo, “Perbandingan Metode Probabilistik Naive Bayesian Classifier dan Jaringan Syaraf Tiruan Learning Vector Quantization dalam Kasus Klasifikasi Penyakit Kandungan,” *ITSmart*, vol. 2, no. 2, p. 21, Mar. 2016, doi: 10.20961/its.v2i2.628.

[7] A. Prabowo and E. A. Sarwoko, “Perbandingan Antara Metode Kohonen Neural Network dengan Metode Learning Vector Quantization Pada Pengenalan Pola Tandatangan,” vol. 14, p. 7.

[8] E. B. Ginting, Prof. Dr. M. Zarlis, and Dr. Z. Situmorang, “KOMBINASI ALGORITMA JARINGAN SYARAF TIRUAN LEARNING VECTOR QUANTIZATION (LVQ) DAN SELF ORGANIZING KOHONEN PADA KECEPATAN PENGENALAN POLA TANDA TANGAN.” 2014.

[9] R. Meliawati, O. Soesanto, and D. Kartini, “PENERAPAN METODE LEARNING VECTOR QUANTIZATION (LVQ) PADA PREDIKSI JURUSAN DI SMA PGRI 1 BANJARBARU,” vol. 04, p. 10, 2016.

[10] Samsir, “Klasifikasi Penyakit Tenggorokan Hidung Telinga (THT) Menggunakan Jaringan Syaraf Tiruan Dengan Metode Learning Vektor Quantization (THT) Di RSUD Rantauprapat Labuhanbatu.”

APPENDIX

**CODING TABLE DATA**

1. DROP DATABASE IF EXISTS dbaps;
2. CREATE DATABASE dbaps;
3. USE dbaps;
4. DROP TABLE IF EXISTS tbldata;
5. CREATE TABLE tbldata(
6. id INT PRIMARY KEY AUTO\_INCREMENT
7. , Gender VARCHAR(10)
8. , customer\_type varchar(50)
9. , age varchar(10)
10. , type\_of\_travel VARCHAR(50)
11. , customer\_class VARCHAR(50)
12. , flight\_distance varchar(10)
13. , inflight\_wifi\_service varchar(10)
14. , departure\_arrival\_time\_convenient varchar(10)
15. , ease\_of\_online\_booking varchar(10)
16. , gate\_location varchar(10)
17. , food\_and\_drink varchar(10)
18. , online\_boarding varchar(10)
19. , seat\_comfort varchar(10)
20. , inflight\_entertainment varchar(10)
21. , onboard\_service varchar(10)
22. , leg\_room\_service varchar(10)
23. , baggage\_handling varchar(10)
24. , checkin\_service varchar(10)
25. , inflight\_service varchar(10)
26. , cleanliness varchar(10)
27. , departure\_delay\_in\_minutes varchar(10)
28. , arrival\_delay\_in\_minutes varchar(10)
29. , satisfaction VARCHAR(50)
30. ) engine=InnoDB;
31. LOAD DATA LOCAL INFILE 'airline\_passenger\_satisfaction.csv'
32. INTO TABLE tbldata
33. FIELDS TERMINATED BY ','
34. ENCLOSED BY ''
35. LINES TERMINATED BY '\n'
36. IGNORE 1 LINES
37. (id, Gender, customer\_type, age, type\_of\_travel, customer\_class,
38. flight\_distance, inflight\_wifi\_service, departure\_arrival\_time\_convenient,
39. ease\_of\_online\_booking, gate\_location, food\_and\_drink, online\_boarding,
40. seat\_comfort, inflight\_entertainment, onboard\_service, leg\_room\_service,
41. baggage\_handling, checkin\_service, inflight\_service, cleanliness,
42. departure\_delay\_in\_minutes, arrival\_delay\_in\_minutes, satisfaction)
43. SET
44. Gender = IF(Gender = '', null, Gender)
45. , customer\_type = IF(customer\_type = '', null, customer\_type)
46. , age = IF(age = '', null, age)
47. , type\_of\_travel = IF(type\_of\_travel = '', null, type\_of\_travel)
48. , customer\_class = IF(customer\_class = '', null, customer\_class)
49. , flight\_distance = IF(flight\_distance = '', null, flight\_distance)
50. , inflight\_wifi\_service = IF(inflight\_wifi\_service = '', null, inflight\_wifi\_service)
51. , departure\_arrival\_time\_convenient = IF(departure\_arrival\_time\_convenient = '', null, departure\_arrival\_time\_convenient)
52. , ease\_of\_online\_booking = IF(ease\_of\_online\_booking = '', null, ease\_of\_online\_booking)
53. , gate\_location = IF(gate\_location = '', null, gate\_location)
54. , food\_and\_drink = IF(food\_and\_drink = '', null, food\_and\_drink)
55. , online\_boarding = IF(online\_boarding = '', null, online\_boarding)
56. , seat\_comfort = IF(seat\_comfort = '', null, seat\_comfort)
57. , inflight\_entertainment = IF(inflight\_entertainment = '', null, inflight\_entertainment)
58. , onboard\_service = IF(onboard\_service = '', null, onboard\_service)
59. , leg\_room\_service = IF(leg\_room\_service = '', null, leg\_room\_service)
60. , baggage\_handling = IF(baggage\_handling = '', null, baggage\_handling)
61. , checkin\_service = IF(checkin\_service = '', null, checkin\_service)
62. , inflight\_service = IF(inflight\_service = '', null, inflight\_service)
63. , cleanliness = IF(cleanliness = '', null, cleanliness)
64. , departure\_delay\_in\_minutes = IF(departure\_delay\_in\_minutes = '', null, departure\_delay\_in\_minutes)
65. , arrival\_delay\_in\_minutes = IF(arrival\_delay\_in\_minutes = '', null, arrival\_delay\_in\_minutes)
66. , satisfaction = IF(satisfaction = '', null, satisfaction)
67. ;
68. INSERT INTO tbldataprocess
69. SELECT \* FROM tbldata

;

**CODING TABLE DATA PROCESS**

1. DROP TABLE IF EXISTS tbldataprocess;
2. CREATE TABLE tbldataprocess(
3. id INT PRIMARY KEY
4. , Gender VARCHAR(10)
5. , customer\_type varchar(50)
6. , age INT
7. , type\_of\_travel VARCHAR(50)
8. , customer\_class VARCHAR(50)
9. , flight\_distance INT
10. , inflight\_wifi\_service INT
11. , departure\_arrival\_time\_convenient INT
12. , ease\_of\_online\_booking INT
13. , gate\_location INT
14. , food\_and\_drink INT
15. , online\_boarding INT
16. , seat\_comfort INT
17. , inflight\_entertainment INT
18. , onboard\_service INT
19. , leg\_room\_service INT
20. , baggage\_handling INT
21. , checkin\_service INT
22. , inflight\_service INT
23. , cleanliness INT
24. , departure\_delay\_in\_minutes INT
25. , arrival\_delay\_in\_minutes INT
26. , satisfaction VARCHAR(50)
27. ) engine=InnoDB;
28. INSERT INTO tbldataprocess
29. SELECT \* FROM tbldata
30. ;

**PROCEDURE PREPROCESSING**

1. DROP PROCEDURE IF EXISTS preprocessing;
2. DELIMITER ##
3. CREATE PROCEDURE preprocessing()
4. BEGIN
5. DECLARE i, iwhile, spinformation\_int INT DEFAULT 0;
6. DECLARE nama, spinformation, spinformation2 VARCHAR(255);
7. DELETE FROM tbldataprocess
8. WHERE
9. Gender IS NULL or
10. customer\_type IS NULL or
11. age IS NULL or
12. type\_of\_travel IS NULL or
13. customer\_class IS NULL or
14. flight\_distance IS NULL or
15. inflight\_wifi\_service IS NULL or
16. departure\_arrival\_time\_convenient IS NULL or
17. ease\_of\_online\_booking IS NULL or
18. gate\_location IS NULL or
19. food\_and\_drink IS NULL or
20. online\_boarding IS NULL or
21. seat\_comfort IS NULL or
22. inflight\_entertainment IS NULL or
23. onboard\_service IS NULL or
24. leg\_room\_service IS NULL or
25. baggage\_handling IS NULL or
26. checkin\_service IS NULL or
27. inflight\_service IS NULL or
28. cleanliness IS NULL or
29. departure\_delay\_in\_minutes IS NULL or
30. arrival\_delay\_in\_minutes IS NULL or
31. satisfaction IS NULL
32. ;
33. SET @num := 0;
34. UPDATE tbldataprocess SET id = @num := (@num+1);
35. ALTER TABLE tbldataprocess AUTO\_INCREMENT =1;
36. -- GENDER
37. SELECT count(DISTINCT gender) into i from tbldataprocess;
38. SET iwhile = 0;
39. WHILE iwhile<>i DO
40. SELECT DISTINCT gender INTO spinformation FROM tbldataprocess order by gender ASC limit iwhile, 1;
41. UPDATE tbldataprocess set gender=iwhile where gender=spinformation;
42. set iwhile= iwhile +1;
43. END WHILE ;
44. -- Customer Type
45. SELECT count(DISTINCT customer\_type) into i from tbldataprocess;
46. SET iwhile = 0;
47. WHILE iwhile <> i DO
48. SELECT DISTINCT customer\_type INTO spinformation FROM tbldataprocess order by customer\_type ASC limit iwhile, 1;
49. UPDATE tbldataprocess set customer\_type=iwhile where customer\_type=spinformation;
50. set iwhile= iwhile +1;
51. END WHILE ;
52. -- AGE
53. UPDATE tbldataprocess set age=0 where age <= 27;
54. UPDATE tbldataprocess set age=1 where age > 27 and age <= 51;
55. UPDATE tbldataprocess set age=2 where age > 51;
56. -- Type Of Travel
57. SELECT count(DISTINCT type\_of\_travel) into i from tbldataprocess;
58. SET iwhile = 0;
59. WHILE iwhile <> i DO
60. SELECT DISTINCT type\_of\_travel INTO spinformation FROM tbldataprocess order by type\_of\_travel ASC limit iwhile, 1;
61. UPDATE tbldataprocess set type\_of\_travel=iwhile where type\_of\_travel=spinformation;
62. set iwhile= iwhile +1;
63. END WHILE ;
64. -- Customer Class
65. SELECT count(DISTINCT customer\_class) into i from tbldataprocess;
66. SET iwhile = 0;
67. WHILE iwhile <> i DO
68. SELECT DISTINCT customer\_class into spinformation FROM tbldataprocess order by customer\_class ASC limit iwhile, 1;
69. UPDATE tbldataprocess set customer\_class=iwhile where customer\_class=spinformation;
70. set iwhile= iwhile +1;
71. END WHILE ;
72. -- FLIGHT DISTANCE
73. UPDATE tbldataprocess set flight\_distance=0 where flight\_distance <= 414;
74. UPDATE tbldataprocess set flight\_distance=1 where flight\_distance > 414 && flight\_distance <= 1744;
75. UPDATE tbldataprocess set flight\_distance=2 where flight\_distance > 1744;
76. -- Inflight Wifi Service
77. SELECT count(DISTINCT inflight\_wifi\_service) into i from tbldataprocess;
78. SET iwhile = 0;
79. WHILE iwhile <> i DO
80. SELECT DISTINCT inflight\_wifi\_service into spinformation\_int FROM tbldataprocess order by inflight\_wifi\_service ASC limit iwhile, 1;
81. set iwhile= iwhile +1;
82. END WHILE;
83. -- Departure Arrival Time Convenient
84. SELECT count(DISTINCT departure\_arrival\_time\_convenient) into i from tbldataprocess;
85. SET iwhile = 0;
86. WHILE iwhile <> i DO
87. SELECT DISTINCT departure\_arrival\_time\_convenient into spinformation\_int FROM tbldataprocess order by departure\_arrival\_time\_convenient ASC limit iwhile, 1;
88. set iwhile= iwhile +1;
89. END WHILE;
90. -- Ease of Online Booking
91. SELECT count(DISTINCT ease\_of\_online\_booking) into i from tbldataprocess;
92. SET iwhile = 0;
93. WHILE iwhile <> i DO
94. SELECT DISTINCT ease\_of\_online\_booking into spinformation\_int FROM tbldataprocess order by ease\_of\_online\_booking ASC limit iwhile, 1;
95. set iwhile= iwhile +1;
96. END WHILE;
97. -- Gate Location
98. SELECT count(DISTINCT gate\_location) into i from tbldataprocess;
99. SET iwhile = 0;
100. WHILE iwhile <> i DO
101. SELECT DISTINCT gate\_location into spinformation\_int FROM tbldataprocess order by gate\_location ASC limit iwhile, 1;
102. set iwhile= iwhile +1;
103. END WHILE;
104. -- Food and Drink
105. SELECT count(DISTINCT food\_and\_drink) into i from tbldataprocess;
106. SET iwhile = 0;
107. WHILE iwhile <> i DO
108. SELECT DISTINCT food\_and\_drink into spinformation\_int FROM tbldataprocess order by food\_and\_drink ASC limit iwhile, 1;
109. set iwhile= iwhile +1;
110. END WHILE;
111. -- Online Boarding
112. SELECT count(DISTINCT online\_boarding) into i from tbldataprocess;
113. SET iwhile = 0;
114. WHILE iwhile <> i DO
115. SELECT DISTINCT online\_boarding into spinformation\_int FROM tbldataprocess order by online\_boarding ASC limit iwhile, 1;
116. set iwhile= iwhile +1;
117. END WHILE;
118. -- Seat Comfort
119. SELECT count(DISTINCT seat\_comfort) into i from tbldataprocess;
120. SET iwhile = 0;
121. WHILE iwhile <> i DO
122. SELECT DISTINCT seat\_comfort into spinformation\_int FROM tbldataprocess order by seat\_comfort ASC limit iwhile, 1;
123. set iwhile= iwhile +1;
124. END WHILE;
125. -- Inflight Enterteinment
126. SELECT count(DISTINCT inflight\_entertainment) into i from tbldataprocess;
127. SET iwhile = 0;
128. WHILE iwhile <> i DO
129. SELECT DISTINCT inflight\_entertainment into spinformation\_int FROM tbldataprocess order by inflight\_entertainment ASC limit iwhile, 1;
130. set iwhile= iwhile +1;
131. END WHILE;
132. -- Onboard Service
133. SELECT count(DISTINCT onboard\_service) into i from tbldataprocess;
134. SET iwhile = 0;
135. WHILE iwhile <> i DO
136. SELECT DISTINCT onboard\_service into spinformation\_int FROM tbldataprocess order by onboard\_service ASC limit iwhile, 1;
137. set iwhile= iwhile +1;
138. END WHILE;
139. -- Leg Room Service
140. SELECT count(DISTINCT leg\_room\_service) into i from tbldataprocess;
141. SET iwhile = 0;
142. WHILE iwhile <> i DO
143. SELECT DISTINCT leg\_room\_service into spinformation\_int FROM tbldataprocess order by leg\_room\_service ASC limit iwhile, 1;
144. set iwhile= iwhile +1;
145. END WHILE;
146. -- Baggage Handling
147. SELECT count(DISTINCT baggage\_handling) into i from tbldataprocess;
148. SET iwhile = 0;
149. WHILE iwhile <> i DO
150. SELECT DISTINCT baggage\_handling into spinformation\_int FROM tbldataprocess order by baggage\_handling ASC limit iwhile, 1;
151. set iwhile= iwhile +1;
152. END WHILE;
153. -- Checkin Service
154. SELECT count(DISTINCT checkin\_service) into i from tbldataprocess;
155. SET iwhile = 0;
156. WHILE iwhile <> i DO
157. SELECT DISTINCT checkin\_service into spinformation\_int FROM tbldataprocess order by checkin\_service ASC limit iwhile, 1;
158. set iwhile= iwhile +1;
159. END WHILE;
160. -- Inflight Service
161. SELECT count(DISTINCT inflight\_service) into i from tbldataprocess;
162. SET iwhile = 0;
163. WHILE iwhile <> i DO
164. SELECT DISTINCT inflight\_service into spinformation\_int FROM tbldataprocess order by inflight\_service ASC limit iwhile, 1;
165. set iwhile= iwhile +1;
166. END WHILE;
167. -- Cleanliness
168. SELECT count(DISTINCT cleanliness) into i from tbldataprocess;
169. SET iwhile = 0;
170. WHILE iwhile <> i DO
171. SELECT DISTINCT cleanliness into spinformation\_int FROM tbldataprocess order by cleanliness ASC limit iwhile, 1;
172. set iwhile= iwhile +1;
173. END WHILE;
174. -- Departure Delay In Minutes
175. UPDATE tbldataprocess set departure\_delay\_in\_minutes=0 where departure\_delay\_in\_minutes <= 12;
176. UPDATE tbldataprocess set departure\_delay\_in\_minutes=1 where departure\_delay\_in\_minutes > 12;
178. -- Arrival Delay In Minutes
179. UPDATE tbldataprocess set arrival\_delay\_in\_minutes=0 where arrival\_delay\_in\_minutes <= 13;
180. UPDATE tbldataprocess set arrival\_delay\_in\_minutes=1 where arrival\_delay\_in\_minutes > 13;
181. -- Satisfaction
182. SELECT count(DISTINCT satisfaction) into i from tbldataprocess;
183. SET iwhile = 0;
184. WHILE iwhile <> i DO
185. SELECT DISTINCT satisfaction into spinformation FROM tbldataprocess order by satisfaction ASC limit iwhile, 1;
186. UPDATE tbldataprocess set satisfaction=iwhile where satisfaction=spinformation;
187. set iwhile= iwhile +1;
188. END WHILE ;
189. END ##
190. DELIMITER ;
191. CALL preprocessing();

**CODING TABLE TO TESTING**

1. DROP TABLE IF EXISTS tbldatatraining;
2. CREATE TABLE tbldatatraining(
3. id INT PRIMARY KEY AUTO\_INCREMENT
4. , Gender VARCHAR(10)
5. , customer\_type varchar(50)
6. , age INT
7. , type\_of\_travel VARCHAR(50)
8. , customer\_class VARCHAR(50)
9. , flight\_distance INT
10. , inflight\_wifi\_service INT
11. , departure\_arrival\_time\_convenient INT
12. , ease\_of\_online\_booking INT
13. , gate\_location INT
14. , food\_and\_drink INT
15. , online\_boarding INT
16. , seat\_comfort INT
17. , inflight\_entertainment INT
18. , onboard\_service INT
19. , leg\_room\_service INT
20. , baggage\_handling INT
21. , checkin\_service INT
22. , inflight\_service INT
23. , cleanliness INT
24. , departure\_delay\_in\_minutes INT
25. , arrival\_delay\_in\_minutes INT
26. , satisfaction VARCHAR(50)
27. ) engine=InnoDB;
28. DROP TABLE IF EXISTS tbldatatesting;
29. CREATE TABLE tbldatatesting(
30. id INT PRIMARY KEY AUTO\_INCREMENT
31. , Gender VARCHAR(10)
32. , customer\_type varchar(50)
33. , age INT
34. , type\_of\_travel VARCHAR(50)
35. , customer\_class VARCHAR(50)
36. , flight\_distance INT
37. , inflight\_wifi\_service INT
38. , departure\_arrival\_time\_convenient INT
39. , ease\_of\_online\_booking INT
40. , gate\_location INT
41. , food\_and\_drink INT
42. , online\_boarding INT
43. , seat\_comfort INT
44. , inflight\_entertainment INT
45. , onboard\_service INT
46. , leg\_room\_service INT
47. , baggage\_handling INT
48. , checkin\_service INT
49. , inflight\_service INT
50. , cleanliness INT
51. , departure\_delay\_in\_minutes INT
52. , arrival\_delay\_in\_minutes INT
53. , satisfaction VARCHAR(50)
54. , prediksi VARCHAR(255)
55. ) engine=InnoDB;
56. CREATE TABLE tblaccuracy (
57. id INT PRIMARY KEY AUTO\_INCREMENT
58. , algoritma VARCHAR(15)
59. , testing INT
60. , total\_data\_training INT
61. , total\_data INT
62. , total\_training INT
63. , tp INT
64. , tn INT
65. , fp INT
66. , fn INT
67. , tnull INT
68. , fnull INT
69. , accuracy FLOAT(4,2)
70. );
71. CREATE TABLE tblW(
72. id INT PRIMARY KEY AUTO\_INCREMENT
73. ,testing INT
74. ,alpha FLOAT(30,20)
75. ,eps FLOAT(30,20)
76. ,w1 FLOAT(30,20)
77. ,w2 FLOAT(30,20)
78. ,w3 FLOAT(30,20)
79. ,w4 FLOAT(30,20)
80. ,w5 FLOAT(30,20)
81. ,w6 FLOAT(30,20)
82. ,w7 FLOAT(30,20)
83. ,w8 FLOAT(30,20)
84. ,w9 FLOAT(30,20)
85. ,w10 FLOAT(30,20)
86. ,w11 FLOAT(30,20)
87. ,w12 FLOAT(30,20)
88. ,w13 FLOAT(30,20)
89. ,w14 FLOAT(30,20)
90. ,w15 FLOAT(30,20)
91. ,w16 FLOAT(30,20)
92. ,w17 FLOAT(30,20)
93. ,w18 FLOAT(30,20)
94. ,w19 FLOAT(30,20)
95. ,w20 FLOAT(30,20)
96. ,w21 FLOAT(30,20)
97. ,w22 FLOAT(30,20)
98. ,w23 FLOAT(30,20)
99. );

**PROCEDURE NAÏVE BAYES**

1. DROP PROCEDURE IF EXISTS bayesian;
2. DELIMITER ##
3. CREATE PROCEDURE bayesian(number\_of\_testing INT)
4. -- CREATE PROCEDURE bayesian()
5. BEGIN
6. -- prob = probability
7. DECLARE
8. prob\_satisfied, prob\_gender\_s, prob\_customer\_type\_s, prob\_age\_s, prob\_type\_of\_travel\_s, prob\_customer\_class\_s
9. , prob\_flight\_distance\_s, prob\_inflight\_wifi\_service\_s, prob\_departure\_arrival\_time\_convenient\_s, prob\_ease\_of\_online\_booking\_s
10. , prob\_gate\_location\_s, prob\_food\_and\_drink\_s, prob\_online\_boarding\_s, prob\_seat\_comfort\_s, prob\_inflight\_entertainment\_s
11. , prob\_onboard\_service\_s, prob\_leg\_room\_service\_s, prob\_baggage\_handling\_s, prob\_checkin\_service\_s, prob\_inflight\_service\_s
12. , prob\_cleanliness\_s, prob\_departure\_delay\_in\_minutes\_s, prob\_arrival\_delay\_in\_minutes\_s
13. FLOAT(30,30) DEFAULT 0;
14. DECLARE total\_satisfied, total\_notsatisfied FLOAT(30,20);
15. DECLARE
16. prob\_notsatisfied, prob\_gender\_ns, prob\_customer\_type\_ns, prob\_age\_ns, prob\_type\_of\_travel\_ns, prob\_customer\_class\_ns
17. , prob\_flight\_distance\_ns, prob\_inflight\_wifi\_service\_ns, prob\_departure\_arrival\_time\_convenient\_ns, prob\_ease\_of\_online\_booking\_ns
18. , prob\_gate\_location\_ns, prob\_food\_and\_drink\_ns, prob\_online\_boarding\_ns, prob\_seat\_comfort\_ns, prob\_inflight\_entertainment\_ns
19. , prob\_onboard\_service\_ns, prob\_leg\_room\_service\_ns, prob\_baggage\_handling\_ns, prob\_checkin\_service\_ns, prob\_inflight\_service\_ns
20. , prob\_cleanliness\_ns, prob\_departure\_delay\_in\_minutes\_ns, prob\_arrival\_delay\_in\_minutes\_ns
22. FLOAT(30,30) DEFAULT 0;
23. DECLARE prediksi\_s, prediksi\_ns FLOAT(30,30);
24. DECLARE i, testing\_ke, total\_training, total\_testing, i\_testing, total\_data INT DEFAULT 0;
25. DECLARE info\_satisfaction VARCHAR(2);
26. SELECT COUNT(\*) INTO total\_data FROM tbldataprocess;
28. SET testing\_ke = number\_of\_testing;
29. IF testing\_ke = 1 THEN SET total\_training = 0.9 \* total\_data;
30. ELSEIF testing\_ke = 2 THEN SET total\_training = 0.75 \* total\_data;
31. ELSEIF testing\_ke = 3 THEN SET total\_training = 0.5 \* total\_data;
32. ELSEIF testing\_ke = 4 THEN SET total\_training = 0.25 \* total\_data;
33. ELSEIF testing\_ke = 5 THEN SET total\_training = 0.1 \* total\_data;
34. END IF;
35. -- SET total\_training = 0.9 \* total\_data;
36. SET total\_testing = total\_data-total\_training;
37. SET i\_testing = 1;
38. UPDATE tblaccuracy SET total\_data=0, tp=0, tn=0, fp=0, fn=0,tnull=0,fnull=0 WHERE algoritma = 'Bayesian' AND testing = testing\_ke;
39. UPDATE tblaccuracy SET total\_data\_training=total\_training, total\_data = total\_testing WHERE testing = testing\_ke AND algoritma = "Bayesian";
40. TRUNCATE tbldatatesting;
41. TRUNCATE tbldatatraining;
42. INSERT INTO tbldatatraining ( Gender, customer\_type, age, type\_of\_travel, customer\_class, flight\_distance, inflight\_wifi\_service, departure\_arrival\_time\_convenient, ease\_of\_online\_booking, gate\_location, food\_and\_drink, online\_boarding, seat\_comfort, inflight\_entertainment, onboard\_service, leg\_room\_service, baggage\_handling, checkin\_service, inflight\_service, cleanliness, departure\_delay\_in\_minutes, arrival\_delay\_in\_minutes, satisfaction)
43. SELECT Gender, customer\_type, age, type\_of\_travel, customer\_class, flight\_distance, inflight\_wifi\_service, departure\_arrival\_time\_convenient, ease\_of\_online\_booking, gate\_location, food\_and\_drink, online\_boarding, seat\_comfort, inflight\_entertainment, onboard\_service, leg\_room\_service, baggage\_handling, checkin\_service, inflight\_service, cleanliness, departure\_delay\_in\_minutes, arrival\_delay\_in\_minutes, satisfaction
44. FROM tbldataprocess where id<= total\_training;
45. INSERT INTO tbldatatesting ( Gender, customer\_type, age, type\_of\_travel, customer\_class, flight\_distance, inflight\_wifi\_service, departure\_arrival\_time\_convenient, ease\_of\_online\_booking, gate\_location, food\_and\_drink, online\_boarding, seat\_comfort, inflight\_entertainment, onboard\_service, leg\_room\_service, baggage\_handling, checkin\_service, inflight\_service, cleanliness, departure\_delay\_in\_minutes, arrival\_delay\_in\_minutes, satisfaction)
46. SELECT Gender, customer\_type, age, type\_of\_travel, customer\_class, flight\_distance, inflight\_wifi\_service, departure\_arrival\_time\_convenient, ease\_of\_online\_booking, gate\_location, food\_and\_drink, online\_boarding, seat\_comfort, inflight\_entertainment, onboard\_service, leg\_room\_service, baggage\_handling, checkin\_service, inflight\_service, cleanliness, departure\_delay\_in\_minutes, arrival\_delay\_in\_minutes, satisfaction
47. FROM tbldataprocess WHERE
48. id > total\_training
49. AND id <= ( total\_training + total\_testing);
50. SET total\_satisfied = (SELECT count(satisfaction) FROM tbldatatraining WHERE satisfaction = 1);
51. SET total\_notsatisfied = (SELECT count(satisfaction) FROM tbldatatraining WHERE satisfaction = 0);
52. SET prob\_satisfied = total\_satisfied / total\_training;
53. SET prob\_notsatisfied = total\_notsatisfied / total\_training;
54. -- WHILE per baris
55. WHILE i\_testing <= total\_testing DO
56. -- GENDER
57. SET prob\_gender\_s = (SELECT count(gender) FROM tbldatatraining WHERE gender=(SELECT gender FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
58. SET prob\_gender\_ns = (SELECT count(gender) FROM tbldatatraining WHERE gender=(SELECT gender FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
59. -- Customer Type
60. SET prob\_customer\_type\_s = (SELECT count(customer\_type) FROM tbldatatraining WHERE customer\_type=(SELECT customer\_type FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
61. SET prob\_customer\_type\_ns = (SELECT count(customer\_type) FROM tbldatatraining WHERE customer\_type=(SELECT customer\_type FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
62. -- AGE
63. SET prob\_age\_s = (SELECT count(age) FROM tbldatatraining WHERE age=(SELECT age FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
64. SET prob\_age\_ns = (SELECT count(age) FROM tbldatatraining WHERE age=(SELECT age FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
66. -- Type Of Travel
67. SET prob\_type\_of\_travel\_s = (SELECT count(type\_of\_travel) FROM tbldatatraining WHERE type\_of\_travel=(SELECT type\_of\_travel FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
68. SET prob\_type\_of\_travel\_ns = (SELECT count(type\_of\_travel) FROM tbldatatraining WHERE type\_of\_travel=(SELECT type\_of\_travel FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
70. -- Customer Class
71. SET prob\_customer\_class\_s = (SELECT count(customer\_class) FROM tbldatatraining WHERE customer\_class=(SELECT customer\_class FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
72. SET prob\_customer\_class\_ns = (SELECT count(customer\_class) FROM tbldatatraining WHERE customer\_class=(SELECT customer\_class FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
74. -- FLIGHT DISTANCE
75. SET prob\_flight\_distance\_s = (SELECT count(flight\_distance) FROM tbldatatraining WHERE flight\_distance=(SELECT flight\_distance FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
76. SET prob\_flight\_distance\_ns = (SELECT count(flight\_distance) FROM tbldatatraining WHERE flight\_distance=(SELECT flight\_distance FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
78. -- Inflight Wifi Service
79. SET prob\_inflight\_wifi\_service\_s = (SELECT count(inflight\_wifi\_service) FROM tbldatatraining WHERE inflight\_wifi\_service=(SELECT inflight\_wifi\_service FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
80. SET prob\_inflight\_wifi\_service\_ns = (SELECT count(inflight\_wifi\_service) FROM tbldatatraining WHERE inflight\_wifi\_service=(SELECT inflight\_wifi\_service FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
82. -- Departure Arrival Time Convenient
83. SET prob\_departure\_arrival\_time\_convenient\_s = (SELECT count(departure\_arrival\_time\_convenient) FROM tbldatatraining WHERE departure\_arrival\_time\_convenient=(SELECT departure\_arrival\_time\_convenient FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
84. SET prob\_departure\_arrival\_time\_convenient\_ns = (SELECT count(departure\_arrival\_time\_convenient) FROM tbldatatraining WHERE departure\_arrival\_time\_convenient=(SELECT departure\_arrival\_time\_convenient FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
86. -- Ease of Online Booking
87. SET prob\_ease\_of\_online\_booking\_s = (SELECT count(ease\_of\_online\_booking) FROM tbldatatraining WHERE ease\_of\_online\_booking=(SELECT ease\_of\_online\_booking FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
88. SET prob\_ease\_of\_online\_booking\_ns = (SELECT count(ease\_of\_online\_booking) FROM tbldatatraining WHERE ease\_of\_online\_booking=(SELECT ease\_of\_online\_booking FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
89. -- Gate Location
90. SET prob\_gate\_location\_s = (SELECT count(gate\_location) FROM tbldatatraining WHERE gate\_location=(SELECT gate\_location FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
91. SET prob\_gate\_location\_ns = (SELECT count(gate\_location) FROM tbldatatraining WHERE gate\_location=(SELECT gate\_location FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
92. -- Food and Drink
93. SET prob\_food\_and\_drink\_s = (SELECT count(food\_and\_drink) FROM tbldatatraining WHERE food\_and\_drink=(SELECT food\_and\_drink FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
94. SET prob\_food\_and\_drink\_ns = (SELECT count(food\_and\_drink) FROM tbldatatraining WHERE food\_and\_drink=(SELECT food\_and\_drink FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
95. -- Online Boarding
96. SET prob\_online\_boarding\_s = (SELECT count(online\_boarding) FROM tbldatatraining WHERE online\_boarding=(SELECT online\_boarding FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
97. SET prob\_online\_boarding\_ns = (SELECT count(online\_boarding) FROM tbldatatraining WHERE online\_boarding=(SELECT online\_boarding FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
99. -- Seat Comfort
100. SET prob\_seat\_comfort\_s = (SELECT count(seat\_comfort) FROM tbldatatraining WHERE seat\_comfort=(SELECT seat\_comfort FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
101. SET prob\_seat\_comfort\_ns = (SELECT count(seat\_comfort) FROM tbldatatraining WHERE seat\_comfort=(SELECT seat\_comfort FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
103. -- Inflight Enterteinment
104. SET prob\_inflight\_entertainment\_s = (SELECT count(inflight\_entertainment) FROM tbldatatraining WHERE inflight\_entertainment=(SELECT inflight\_entertainment FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
105. SET prob\_inflight\_entertainment\_ns = (SELECT count(inflight\_entertainment) FROM tbldatatraining WHERE inflight\_entertainment=(SELECT inflight\_entertainment FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
106. -- Onboard Service
107. SET prob\_onboard\_service\_s = (SELECT count(onboard\_service) FROM tbldatatraining WHERE onboard\_service=(SELECT onboard\_service FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
108. SET prob\_onboard\_service\_ns = (SELECT count(onboard\_service) FROM tbldatatraining WHERE onboard\_service=(SELECT onboard\_service FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
110. -- Leg Room Service
111. SET prob\_leg\_room\_service\_s = (SELECT count(leg\_room\_service) FROM tbldatatraining WHERE leg\_room\_service=(SELECT leg\_room\_service FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
112. SET prob\_leg\_room\_service\_ns = (SELECT count(leg\_room\_service) FROM tbldatatraining WHERE leg\_room\_service=(SELECT leg\_room\_service FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
113. -- Baggage Handling
114. SET prob\_baggage\_handling\_s = (SELECT count(baggage\_handling) FROM tbldatatraining WHERE baggage\_handling=(SELECT baggage\_handling FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
115. SET prob\_baggage\_handling\_ns = (SELECT count(baggage\_handling) FROM tbldatatraining WHERE baggage\_handling=(SELECT baggage\_handling FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
116. -- Checkin Service
117. SET prob\_checkin\_service\_s = (SELECT count(checkin\_service) FROM tbldatatraining WHERE checkin\_service=(SELECT checkin\_service FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
118. SET prob\_checkin\_service\_ns = (SELECT count(checkin\_service) FROM tbldatatraining WHERE checkin\_service=(SELECT checkin\_service FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
119. -- Inflight Service
120. SET prob\_inflight\_service\_s = (SELECT count(inflight\_service) FROM tbldatatraining WHERE inflight\_service=(SELECT inflight\_service FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
121. SET prob\_inflight\_service\_ns = (SELECT count(inflight\_service) FROM tbldatatraining WHERE inflight\_service=(SELECT inflight\_service FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
122. -- Cleanliness
123. SET prob\_cleanliness\_s = (SELECT count(cleanliness) FROM tbldatatraining WHERE cleanliness=(SELECT cleanliness FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
124. SET prob\_cleanliness\_ns = (SELECT count(cleanliness) FROM tbldatatraining WHERE cleanliness=(SELECT cleanliness FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
126. -- Departure Delay In Minutes
127. SET prob\_departure\_delay\_in\_minutes\_s = (SELECT count(departure\_delay\_in\_minutes) FROM tbldatatraining WHERE departure\_delay\_in\_minutes=(SELECT departure\_delay\_in\_minutes FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
128. SET prob\_departure\_delay\_in\_minutes\_ns = (SELECT count(departure\_delay\_in\_minutes) FROM tbldatatraining WHERE departure\_delay\_in\_minutes=(SELECT departure\_delay\_in\_minutes FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
130. -- Arrival Delay In Minutes
131. SET prob\_arrival\_delay\_in\_minutes\_s = (SELECT count(arrival\_delay\_in\_minutes) FROM tbldatatraining WHERE arrival\_delay\_in\_minutes=(SELECT arrival\_delay\_in\_minutes FROM tbldatatesting where id=i\_testing) AND satisfaction =1) / total\_satisfied;
132. SET prob\_arrival\_delay\_in\_minutes\_ns = (SELECT count(arrival\_delay\_in\_minutes) FROM tbldatatraining WHERE arrival\_delay\_in\_minutes=(SELECT arrival\_delay\_in\_minutes FROM tbldatatesting where id=i\_testing) AND satisfaction =0) / total\_notsatisfied;
134. SET prediksi\_s = prob\_satisfied\* prob\_gender\_s\* prob\_customer\_type\_s\* prob\_age\_s\* prob\_type\_of\_travel\_s\* prob\_customer\_class\_s\* prob\_flight\_distance\_s\* prob\_inflight\_wifi\_service\_s\* prob\_departure\_arrival\_time\_convenient\_s\* prob\_ease\_of\_online\_booking\_s\* prob\_gate\_location\_s\* prob\_food\_and\_drink\_s\* prob\_online\_boarding\_s\* prob\_seat\_comfort\_s\* prob\_inflight\_entertainment\_s\* prob\_onboard\_service\_s\* prob\_leg\_room\_service\_s\* prob\_baggage\_handling\_s\* prob\_checkin\_service\_s\* prob\_inflight\_service\_s\* prob\_cleanliness\_s\* prob\_departure\_delay\_in\_minutes\_s\* prob\_arrival\_delay\_in\_minutes\_s;
135. SET prediksi\_ns =prob\_notsatisfied\* prob\_gender\_ns\* prob\_customer\_type\_ns\* prob\_age\_ns\* prob\_type\_of\_travel\_ns\* prob\_customer\_class\_ns\* prob\_flight\_distance\_ns\* prob\_inflight\_wifi\_service\_ns\* prob\_departure\_arrival\_time\_convenient\_ns\* prob\_ease\_of\_online\_booking\_ns\* prob\_gate\_location\_ns\* prob\_food\_and\_drink\_ns\* prob\_online\_boarding\_ns\* prob\_seat\_comfort\_ns\* prob\_inflight\_entertainment\_ns\* prob\_onboard\_service\_ns\* prob\_leg\_room\_service\_ns\* prob\_baggage\_handling\_ns\* prob\_checkin\_service\_ns\* prob\_inflight\_service\_ns\* prob\_cleanliness\_ns\* prob\_departure\_delay\_in\_minutes\_ns\* prob\_arrival\_delay\_in\_minutes\_ns;
136. SELECT satisfaction INTO info\_satisfaction from tbldatatesting where id = i\_testing;

139. IF info\_satisfaction = 0 THEN -- actual not satisfied
140. IF prediksi\_s < prediksi\_ns THEN
141. UPDATE tblaccuracy SET tn=tn+1 WHERE algoritma="Bayesian" AND testing=testing\_ke;
142. ELSEIF prediksi\_s > prediksi\_ns THEN
143. UPDATE tblaccuracy SET fp=fp+1 WHERE algoritma="Bayesian" AND testing=testing\_ke;
144. ELSEIF prediksi\_s = 0 AND prediksi\_ns = 0 THEN
145. UPDATE tblaccuracy SET fnull=fnull+1 WHERE algoritma="Bayesian" AND testing=testing\_ke;
146. END IF;
147. ELSEIF info\_satisfaction = 1 THEN -- actual satisfied
148. IF prediksi\_s < prediksi\_ns THEN
149. UPDATE tblaccuracy SET fn=fn+1 WHERE algoritma="Bayesian" AND testing=testing\_ke;
150. ELSEIF prediksi\_s > prediksi\_ns THEN
151. UPDATE tblaccuracy SET tp=tp+1 WHERE algoritma="Bayesian" AND testing=testing\_ke;
152. ELSEIF prediksi\_s = 0 AND prediksi\_ns = 0 THEN
153. UPDATE tblaccuracy SET tnull=tnull+1 WHERE algoritma="Bayesian" AND testing=testing\_ke;
154. END IF;
155. END IF;
156. SET i\_testing = i\_testing+1;
157. END WHILE;
159. SELECT testing\_ke as 'selesai uji ke';
160. END ##
161. DELIMITER ;
162. DROP PROCEDURE IF EXISTS processb;
163. DELIMITER ##
164. CREATE PROCEDURE processb()
165. BEGIN
166. CALL bayesian(1);
167. CALL bayesian(2);
168. CALL bayesian(3);
169. CALL bayesian(4);
170. CALL bayesian(5);
171. UPDATE tblaccuracy SET accuracy=((tp+tn)/(tp+tn+fp+fn+tnull+fnull))\*100 where algoritma='Bayesian';
172. END ##
173. DELIMITER ;

**FUNCTION EUCLIDIAN DISTANCE**

1. DROP FUNCTION IF EXISTS ed;
2. DELIMITER ##
3. CREATE FUNCTION ed(
4. w1t FLOAT(30,20)
5. , w1 FLOAT(30,20)
6. , w2t FLOAT(30,20)
7. , w2 FLOAT(30,20)
8. , w3t FLOAT(30,20)
9. , w3 FLOAT(30,20)
10. , w4t FLOAT(30,20)
11. , w4 FLOAT(30,20)
12. , w5t FLOAT(30,20)
13. , w5 FLOAT(30,20)
14. , w6t FLOAT(30,20)
15. , w6 FLOAT(30,20)
16. , w7t FLOAT(30,20)
17. , w7 FLOAT(30,20)
18. , w8t FLOAT(30,20)
19. , w8 FLOAT(30,20)
20. , w9t FLOAT(30,20)
21. , w9 FLOAT(30,20)
22. , w10t FLOAT(30,20)
23. , w10 FLOAT(30,20)
24. , w11t FLOAT(30,20)
25. , w11 FLOAT(30,20)
26. , w12t FLOAT(30,20)
27. , w12 FLOAT(30,20)
28. , w13t FLOAT(30,20)
29. , w13 FLOAT(30,20)
30. , w14t FLOAT(30,20)
31. , w14 FLOAT(30,20)
32. , w15t FLOAT(30,20)
33. , w15 FLOAT(30,20)
34. , w16t FLOAT(30,20)
35. , w16 FLOAT(30,20)
36. , w17t FLOAT(30,20)
37. , w17 FLOAT(30,20)
38. , w18t FLOAT(30,20)
39. , w18 FLOAT(30,20)
40. , w19t FLOAT(30,20)
41. , w19 FLOAT(30,20)
42. , w20t FLOAT(30,20)
43. , w20 FLOAT(30,20)
44. , w21t FLOAT(30,20)
45. , w21 FLOAT(30,20)
46. , w22t FLOAT(30,20)
47. , w22 FLOAT(30,20)
48. )
49. RETURNS FLOAT(30,20)
50. BEGIN
51. DECLARE hasil FLOAT(30,20) DEFAULT 0;
52. SET hasil = SQRT((
53. POWER((w1t - w1),2)
54. + POWER((w2t - w2),2)
55. + POWER((w3t - w3),2)
56. + POWER((w4t - w4),2)
57. + POWER((w5t - w5),2)
58. + POWER((w6t - w6),2)
59. + POWER((w7t - w7),2)
60. + POWER((w8t - w8),2)
61. + POWER((w9t - w9),2)
62. + POWER((w10t - w10),2)
63. + POWER((w11t - w11),2)
64. + POWER((w12t - w12),2)
65. + POWER((w13t - w13),2)
66. + POWER((w14t - w14),2)
67. + POWER((w15t - w15),2)
68. + POWER((w16t - w16),2)
69. + POWER((w17t - w17),2)
70. + POWER((w18t - w18),2)
71. + POWER((w19t - w19),2)
72. + POWER((w20t - w20),2)
73. + POWER((w21t - w21),2)
74. + POWER((w22t - w22),2))
75. )
76. ;
77. RETURN(hasil);
78. END; ##
79. DELIMITER ;

**PROCEDURE LEARNING VECTOR QUANTIZATION**

1. DROP PROCEDURE IF EXISTS lvq;
2. DELIMITER ##
3. CREATE PROCEDURE lvq(number\_of\_testing INT, pAlpha FLOAT(30,20), pEps FLOAT(30,20))
4. BEGIN
5. -- Weight of class satisfied
6. DECLARE w1s,w2s,w3s,w4s,w5s,w6s,w7s,w8s,w9s,w10s,w11s,w12s,w13s,w14s,w15s,w16s,w17s,w18s,w19s,w20s,w21s,w22s,w23s FLOAT(30,20) DEFAULT 0;
7. -- Weight of class neutral or dissatisfied
8. DECLARE w1ns,w2ns,w3ns,w4ns,w5ns,w6ns,w7ns,w8ns,w9ns,w10ns,w11ns,w12ns,w13ns,w14ns,w15ns,w16ns,w17ns,w18ns,w19ns,w20ns,w21ns,w22ns,w23ns FLOAT(30,20) DEFAULT 0;
9. -- Weight of training
10. DECLARE w1t,w2t,w3t,w4t,w5t,w6t,w7t,w8t,w9t,w10t,w11t,w12t,w13t,w14t,w15t,w16t,w17t,w18t,w19t,w20t,w21t,w22t,w23t FLOAT(30,20) DEFAULT 0;
11. -- Get id for initial class satisfied and not
12. DECLARE ids, idns, cj, t, epoch, maxepoch,pepoch INT DEFAULT 0;
13. DECLARE tn\_lvq, fp\_lvq, fn\_lvq, tp\_lvq INT DEFAULT 0;
14. DECLARE ws, wns, wt FLOAT(30,20) DEFAULT 0;
15. DECLARE alpha, eps, err, temp\_alpha FLOAT(30,20) DEFAULT 0;
16. DECLARE info\_satisfaction VARCHAR(2);
18. DECLARE prediction INT DEFAULT 0;
19. DECLARE i, testing\_ke, total\_training, total\_testing, i\_testing, i\_training, temp\_i\_training,
20. total\_data, ptotal\_data\_training INT DEFAULT 0;
22. SELECT COUNT(\*) INTO total\_data FROM tbldataprocess;
23. SET testing\_ke = number\_of\_testing;
24. SET maxepoch=5;
25. IF testing\_ke = 1 THEN SET total\_training = 0.9 \* total\_data;
26. ELSEIF testing\_ke = 2 THEN SET total\_training = 0.75 \* total\_data;
27. ELSEIF testing\_ke = 3 THEN SET total\_training = 0.5 \* total\_data;
28. ELSEIF testing\_ke = 4 THEN SET total\_training = 0.25 \* total\_data;
29. ELSEIF testing\_ke = 5 THEN SET total\_training = 0.1 \* total\_data;
30. END IF;
31. TRUNCATE tbldatatesting;
32. TRUNCATE tbldatatraining;
33. SET total\_testing = total\_data-total\_training;
34. SET i\_testing = 1;
35. SET i\_training = 1;
36. INSERT INTO tbldatatraining ( Gender, customer\_type, age, type\_of\_travel, customer\_class, flight\_distance, inflight\_wifi\_service, departure\_arrival\_time\_convenient, ease\_of\_online\_booking, gate\_location, food\_and\_drink, online\_boarding, seat\_comfort, inflight\_entertainment, onboard\_service, leg\_room\_service, baggage\_handling, checkin\_service, inflight\_service, cleanliness, departure\_delay\_in\_minutes, arrival\_delay\_in\_minutes, satisfaction)
37. SELECT Gender, customer\_type, age, type\_of\_travel, customer\_class, flight\_distance, inflight\_wifi\_service, departure\_arrival\_time\_convenient, ease\_of\_online\_booking, gate\_location, food\_and\_drink, online\_boarding, seat\_comfort, inflight\_entertainment, onboard\_service, leg\_room\_service, baggage\_handling, checkin\_service, inflight\_service, cleanliness, departure\_delay\_in\_minutes, arrival\_delay\_in\_minutes, satisfaction
38. FROM tbldataprocess where id<= total\_training;
39. INSERT INTO tbldatatesting ( Gender, customer\_type, age, type\_of\_travel, customer\_class, flight\_distance, inflight\_wifi\_service, departure\_arrival\_time\_convenient, ease\_of\_online\_booking, gate\_location, food\_and\_drink, online\_boarding, seat\_comfort, inflight\_entertainment, onboard\_service, leg\_room\_service, baggage\_handling, checkin\_service, inflight\_service, cleanliness, departure\_delay\_in\_minutes, arrival\_delay\_in\_minutes, satisfaction)
40. SELECT Gender, customer\_type, age, type\_of\_travel, customer\_class, flight\_distance, inflight\_wifi\_service, departure\_arrival\_time\_convenient, ease\_of\_online\_booking, gate\_location, food\_and\_drink, online\_boarding, seat\_comfort, inflight\_entertainment, onboard\_service, leg\_room\_service, baggage\_handling, checkin\_service, inflight\_service, cleanliness, departure\_delay\_in\_minutes, arrival\_delay\_in\_minutes, satisfaction
41. FROM tbldataprocess WHERE
42. id > total\_training
43. AND id <= ( total\_training + total\_testing);
45. INSERT INTO tblaccuracy(algoritma, testing, total\_data\_training, total\_data,total\_training, tp, tn, fp,fn,tnull, fnull, accuracy) VALUES
46. ('LVQ', testing\_ke,0,0,0,0,0,0,0,0,0,0);
47. -- INITIALITATION
48. IF pAlpha=0 THEN
49. SET alpha = 0.9;
50. ELSE
51. SET alpha = pAlpha;
52. END IF;
53. IF pEps=0 THEN
54. SET eps = 0.0000001;
55. ELSE
56. SET eps = pEps;
57. END IF;
58. -- get id from each class
59. SELECT id INTO ids FROM tbldatatraining WHERE satisfaction = 1 ORDER BY RAND() LIMIT 1;
60. SELECT id INTO idns FROM tbldatatraining WHERE satisfaction = 0 ORDER BY RAND() LIMIT 1;
62. SELECT gender INTO w1s FROM tbldatatraining WHERE id=ids;
63. SELECT gender INTO w1ns FROM tbldatatraining WHERE id=idns;
64. SELECT customer\_type INTO w2s FROM tbldatatraining WHERE id=ids;
65. SELECT customer\_type INTO w2ns FROM tbldatatraining WHERE id=idns;
66. SELECT age INTO w3s FROM tbldatatraining WHERE id=ids;
67. SELECT age INTO w3ns FROM tbldatatraining WHERE id=idns;
68. SELECT type\_of\_travel INTO w4s FROM tbldatatraining WHERE id=ids;
69. SELECT type\_of\_travel INTO w4ns FROM tbldatatraining WHERE id=idns;
70. SELECT customer\_class INTO w5s FROM tbldatatraining WHERE id=ids;
71. SELECT customer\_class INTO w5ns FROM tbldatatraining WHERE id=idns;
72. SELECT flight\_distance INTO w6s FROM tbldatatraining WHERE id=ids;
73. SELECT flight\_distance INTO w6ns FROM tbldatatraining WHERE id=idns;
74. SELECT inflight\_wifi\_service INTO w7s FROM tbldatatraining WHERE id=ids;
75. SELECT inflight\_wifi\_service INTO w7ns FROM tbldatatraining WHERE id=idns;
76. SELECT departure\_arrival\_time\_convenient INTO w8s FROM tbldatatraining WHERE id=ids;
77. SELECT departure\_arrival\_time\_convenient INTO w8ns FROM tbldatatraining WHERE id=idns;
78. SELECT ease\_of\_online\_booking INTO w9s FROM tbldatatraining WHERE id=ids;
79. SELECT ease\_of\_online\_booking INTO w9ns FROM tbldatatraining WHERE id=idns;
80. SELECT gate\_location INTO w10s FROM tbldatatraining WHERE id=ids;
81. SELECT gate\_location INTO w10ns FROM tbldatatraining WHERE id=idns;
82. SELECT food\_and\_drink INTO w11s FROM tbldatatraining WHERE id=ids;
83. SELECT food\_and\_drink INTO w11ns FROM tbldatatraining WHERE id=idns;
84. SELECT online\_boarding INTO w12s FROM tbldatatraining WHERE id=ids;
85. SELECT online\_boarding INTO w12ns FROM tbldatatraining WHERE id=idns;
86. SELECT seat\_comfort INTO w13s FROM tbldatatraining WHERE id=ids;
87. SELECT seat\_comfort INTO w13ns FROM tbldatatraining WHERE id=idns;
88. SELECT inflight\_entertainment INTO w14s FROM tbldatatraining WHERE id=ids;
89. SELECT inflight\_entertainment INTO w14ns FROM tbldatatraining WHERE id=idns;
90. SELECT onboard\_service INTO w15s FROM tbldatatraining WHERE id=ids;
91. SELECT onboard\_service INTO w15ns FROM tbldatatraining WHERE id=idns;
92. SELECT leg\_room\_service INTO w16s FROM tbldatatraining WHERE id=ids;
93. SELECT leg\_room\_service INTO w16ns FROM tbldatatraining WHERE id=idns;
94. SELECT baggage\_handling INTO w17s FROM tbldatatraining WHERE id=ids;
95. SELECT baggage\_handling INTO w17ns FROM tbldatatraining WHERE id=idns;
96. SELECT checkin\_service INTO w18s FROM tbldatatraining WHERE id=ids;
97. SELECT checkin\_service INTO w18ns FROM tbldatatraining WHERE id=idns;
98. SELECT inflight\_service INTO w19s FROM tbldatatraining WHERE id=ids;
99. SELECT inflight\_service INTO w19ns FROM tbldatatraining WHERE id=idns;
100. SELECT cleanliness INTO w20s FROM tbldatatraining WHERE id=ids;
101. SELECT cleanliness INTO w20ns FROM tbldatatraining WHERE id=idns;
102. SELECT departure\_delay\_in\_minutes INTO w21s FROM tbldatatraining WHERE id=ids;
103. SELECT departure\_delay\_in\_minutes INTO w21ns FROM tbldatatraining WHERE id=idns;
104. SELECT arrival\_delay\_in\_minutes INTO w22s FROM tbldatatraining WHERE id=ids;
105. SELECT arrival\_delay\_in\_minutes INTO w22ns FROM tbldatatraining WHERE id=idns;
106. -- row used to initialitation is not use again
107. DELETE FROM tbldatatraining WHERE id=ids;
108. DELETE FROM tbldatatraining WHERE id=idns;
109. SET @num := 0;
110. UPDATE tbldatatraining SET id = @num := (@num+1);
111. ALTER TABLE tbldatatraining AUTO\_INCREMENT =1;
112. -- END INITIALITATION
113. -- TRAINING
114. SET temp\_alpha = alpha;
115. WHILE epoch < maxepoch DO
116. SET i\_training = 0;
117. SET temp\_i\_training = 0;
118. SET alpha=temp\_alpha;
119. algolvq: WHILE (i\_training <= total\_training) or (alpha >= eps) DO
120. IF (alpha >= eps) THEN
121. SELECT gender INTO w1t FROM tbldatatraining WHERE id=i\_training;
122. SELECT customer\_type INTO w2t FROM tbldatatraining WHERE id=i\_training;
123. SELECT age INTO w3t FROM tbldatatraining WHERE id=i\_training;
124. SELECT type\_of\_travel INTO w4t FROM tbldatatraining WHERE id=i\_training;
125. SELECT customer\_class INTO w5t FROM tbldatatraining WHERE id=i\_training;
126. SELECT flight\_distance INTO w6t FROM tbldatatraining WHERE id=i\_training;
127. SELECT inflight\_wifi\_service INTO w7t FROM tbldatatraining WHERE id=i\_training;
128. SELECT departure\_arrival\_time\_convenient INTO w8t FROM tbldatatraining WHERE id=i\_training;
129. SELECT ease\_of\_online\_booking INTO w9t FROM tbldatatraining WHERE id=i\_training;
130. SELECT gate\_location INTO w10t FROM tbldatatraining WHERE id=i\_training;
131. SELECT food\_and\_drink INTO w11t FROM tbldatatraining WHERE id=i\_training;
132. SELECT online\_boarding INTO w12t FROM tbldatatraining WHERE id=i\_training;
133. SELECT seat\_comfort INTO w13t FROM tbldatatraining WHERE id=i\_training;
134. SELECT inflight\_entertainment INTO w14t FROM tbldatatraining WHERE id=i\_training;
135. SELECT onboard\_service INTO w15t FROM tbldatatraining WHERE id=i\_training;
136. SELECT leg\_room\_service INTO w16t FROM tbldatatraining WHERE id=i\_training;
137. SELECT baggage\_handling INTO w17t FROM tbldatatraining WHERE id=i\_training;
138. SELECT checkin\_service INTO w18t FROM tbldatatraining WHERE id=i\_training;
139. SELECT inflight\_service INTO w19t FROM tbldatatraining WHERE id=i\_training;
140. SELECT cleanliness INTO w20t FROM tbldatatraining WHERE id=i\_training;
141. SELECT departure\_delay\_in\_minutes INTO w21t FROM tbldatatraining WHERE id=i\_training;
142. SELECT arrival\_delay\_in\_minutes INTO w22t FROM tbldatatraining WHERE id=i\_training;
143. SELECT satisfaction INTO w23t FROM tbldatatraining WHERE id=i\_training;
144. SET ws = ed(w1t, w1s, w2t, w2s, w3t, w3s, w4t, w4s, w5t, w5s, w6t, w6s, w7t, w7s, w8t, w8s, w9t, w9s, w10t, w10s, w11t, w11s, w12t, w12s, w13t, w13s, w14t, w14s, w15t, w15s, w16t, w16s, w17t, w17s, w18t, w18s, w19t, w19s, w20t, w20s, w21t, w21s, w22t, w22s);
145. SET wns = ed(w1t, w1ns, w2t, w2ns, w3t, w3ns, w4t, w4ns, w5t, w5ns, w6t, w6ns, w7t, w7ns, w8t, w8ns, w9t, w9ns, w10t, w10ns, w11t, w11ns, w12t, w12ns, w13t, w13ns, w14t, w14ns, w15t, w15ns, w16t, w16ns, w17t, w17ns, w18t, w18ns, w19t, w19ns, w20t, w20ns, w21t, w21ns, w22t, w22ns);
147. IF ws < wns THEN SET cj = 1;
148. ELSEIF ws > wns THEN SET cj = 0;
149. ELSE SET cj = 1;
150. END IF;
151. SELECT satisfaction INTO t FROM tbldatatraining WHERE id=i\_training;
152. -- cj 1 satisfied, 0 dissatisfied
153. IF cj = 1 AND t = 1 THEN
154. SET w1s = w1s + (alpha \* (w1t - w1s));
155. SET w2s = w2s + (alpha \* (w2t - w2s));
156. SET w3s = w3s + (alpha \* (w3t - w3s));
157. SET w4s = w4s + (alpha \* (w4t - w4s));
158. SET w5s = w5s + (alpha \* (w5t - w5s));
159. SET w6s = w6s + (alpha \* (w6t - w6s));
160. SET w7s = w7s + (alpha \* (w7t - w7s));
161. SET w8s = w8s + (alpha \* (w8t - w8s));
162. SET w9s = w9s + (alpha \* (w9t - w9s));
163. SET w10s = w10s + (alpha \* (w10t - w10s));
164. SET w11s = w11s + (alpha \* (w11t - w11s));
165. SET w12s = w12s + (alpha \* (w12t - w12s));
166. SET w13s = w13s + (alpha \* (w13t - w13s));
167. SET w14s = w14s + (alpha \* (w14t - w14s));
168. SET w15s = w15s + (alpha \* (w15t - w15s));
169. SET w16s = w16s + (alpha \* (w16t - w16s));
170. SET w17s = w17s + (alpha \* (w17t - w17s));
171. SET w18s = w18s + (alpha \* (w18t - w18s));
172. SET w19s = w19s + (alpha \* (w19t - w19s));
173. SET w20s = w20s + (alpha \* (w20t - w20s));
174. SET w21s = w21s + (alpha \* (w21t - w21s));
175. SET w22s = w22s + (alpha \* (w22t - w22s));
176. ELSEIF cj = 0 AND t = 0 THEN
177. SET w1ns = w1ns + (alpha \* (w1t - w1ns));
178. SET w2ns = w2ns + (alpha \* (w2t - w2ns));
179. SET w3ns = w3ns + (alpha \* (w3t - w3ns));
180. SET w4ns = w4ns + (alpha \* (w4t - w4ns));
181. SET w5ns = w5ns + (alpha \* (w5t - w5ns));
182. SET w6ns = w6ns + (alpha \* (w6t - w6ns));
183. SET w7ns = w7ns + (alpha \* (w7t - w7ns));
184. SET w8ns = w8ns + (alpha \* (w8t - w8ns));
185. SET w9ns = w9ns + (alpha \* (w9t - w9ns));
186. SET w10ns = w10ns + (alpha \* (w10t - w10ns));
187. SET w11ns = w11ns + (alpha \* (w11t - w11ns));
188. SET w12ns = w12ns + (alpha \* (w12t - w12ns));
189. SET w13ns = w13ns + (alpha \* (w13t - w13ns));
190. SET w14ns = w14ns + (alpha \* (w14t - w14ns));
191. SET w15ns = w15ns + (alpha \* (w15t - w15ns));
192. SET w16ns = w16ns + (alpha \* (w16t - w16ns));
193. SET w17ns = w17ns + (alpha \* (w17t - w17ns));
194. SET w18ns = w18ns + (alpha \* (w18t - w18ns));
195. SET w19ns = w19ns + (alpha \* (w19t - w19ns));
196. SET w20ns = w20ns + (alpha \* (w20t - w20ns));
197. SET w21ns = w21ns + (alpha \* (w21t - w21ns));
198. SET w22ns = w22ns + (alpha \* (w22t - w22ns));
199. ELSEIF cj = 0 AND t = 1 THEN
200. SET w1ns = w1ns - (alpha \* (w1t - w1ns));
201. SET w2ns = w2ns - (alpha \* (w2t - w2ns));
202. SET w3ns = w3ns - (alpha \* (w3t - w3ns));
203. SET w4ns = w4ns - (alpha \* (w4t - w4ns));
204. SET w5ns = w5ns - (alpha \* (w5t - w5ns));
205. SET w6ns = w6ns - (alpha \* (w6t - w6ns));
206. SET w7ns = w7ns - (alpha \* (w7t - w7ns));
207. SET w8ns = w8ns - (alpha \* (w8t - w8ns));
208. SET w9ns = w9ns - (alpha \* (w9t - w9ns));
209. SET w10ns = w10ns - (alpha \* (w10t - w10ns));
210. SET w11ns = w11ns - (alpha \* (w11t - w11ns));
211. SET w12ns = w12ns - (alpha \* (w12t - w12ns));
212. SET w13ns = w13ns - (alpha \* (w13t - w13ns));
213. SET w14ns = w14ns - (alpha \* (w14t - w14ns));
214. SET w15ns = w15ns - (alpha \* (w15t - w15ns));
215. SET w16ns = w16ns - (alpha \* (w16t - w16ns));
216. SET w17ns = w17ns - (alpha \* (w17t - w17ns));
217. SET w18ns = w18ns - (alpha \* (w18t - w18ns));
218. SET w19ns = w19ns - (alpha \* (w19t - w19ns));
219. SET w20ns = w20ns - (alpha \* (w20t - w20ns));
220. SET w21ns = w21ns - (alpha \* (w21t - w21ns));
221. SET w22ns = w22ns - (alpha \* (w22t - w22ns));
222. ELSEIF cj = 1 AND t = 0 THEN
223. SET w1s = w1s - (alpha \* (w1t - w1s));
224. SET w2s = w2s - (alpha \* (w2t - w2s));
225. SET w3s = w3s - (alpha \* (w3t - w3s));
226. SET w4s = w4s - (alpha \* (w4t - w4s));
227. SET w5s = w5s - (alpha \* (w5t - w5s));
228. SET w6s = w6s - (alpha \* (w6t - w6s));
229. SET w7s = w7s - (alpha \* (w7t - w7s));
230. SET w8s = w8s - (alpha \* (w8t - w8s));
231. SET w9s = w9s - (alpha \* (w9t - w9s));
232. SET w10s = w10s - (alpha \* (w10t - w10s));
233. SET w11s = w11s - (alpha \* (w11t - w11s));
234. SET w12s = w12s - (alpha \* (w12t - w12s));
235. SET w13s = w13s - (alpha \* (w13t - w13s));
236. SET w14s = w14s - (alpha \* (w14t - w14s));
237. SET w15s = w15s - (alpha \* (w15t - w15s));
238. SET w16s = w16s - (alpha \* (w16t - w16s));
239. SET w17s = w17s - (alpha \* (w17t - w17s));
240. SET w18s = w18s - (alpha \* (w18t - w18s));
241. SET w19s = w19s - (alpha \* (w19t - w19s));
242. SET w20s = w20s - (alpha \* (w20t - w20s));
243. SET w21s = w21s - (alpha \* (w21t - w21s));
244. SET w22s = w22s - (alpha \* (w22t - w22s));
245. END IF;
246. SET alpha = alpha - (alpha \* eps);
247. UPDATE tblaccuracy SET total\_data\_training = i\_training+1 WHERE id=(SELECT COUNT(\*) FROM tblaccuracy);
248. SET i\_training = i\_training + 1;
249. SET temp\_i\_training = i\_training;
250. IF (i\_training = total\_training) THEN
251. SET temp\_alpha = alpha;
252. SELECT alpha as 'alphat', i\_training as 't';
253. SET alpha = eps;
254. ELSEIF (alpha <= eps) THEN
255. SET temp\_alpha = alpha;
256. SELECT alpha as 'alphaa', i\_training as 't';
257. SET i\_training = total\_training +1;
258. END IF;
259. ELSE
260. LEAVE algolvq;
261. END IF;
262. END WHILE;
263. SET epoch = epoch + 1;
264. IF (temp\_i\_training <> 0) THEN
265. SET ptotal\_data\_training=ptotal\_data\_training + temp\_i\_training;
266. SET pepoch=epoch;
267. SELECT epoch as "epoch";
268. END IF;
269. END WHILE;
270. UPDATE tblaccuracy SET total\_data\_training = ptotal\_data\_training WHERE id=(SELECT COUNT(\*) FROM tblaccuracy);
271. UPDATE tblaccuracy SET total\_training = pepoch WHERE id=(SELECT COUNT(\*) FROM tblaccuracy);
272. INSERT INTO tblW (testing,alpha, eps, w1,w2,w3,w4,w5,w6,w7,w8,w9,w10,w11,w12,w13,w14,w15,w16,w17,w18,w19,w20,w21,w22,w23) values
273. (
274. testing\_ke
275. ,pAlpha
276. , pEps
277. , w1s
278. , w2s
279. , w3s
280. , w4s
281. , w5s
282. , w6s
283. , w7s
284. , w8s
285. , w9s
286. , w10s
287. , w11s
288. , w12s
289. , w13s
290. , w14s
291. , w15s
292. , w16s
293. , w17s
294. , w18s
295. , w19s
296. , w20s
297. , w21s
298. , w22s
299. , w23s
300. );
301. INSERT INTO tblW (testing,alpha, eps, w1,w2,w3,w4,w5,w6,w7,w8,w9,w10,w11,w12,w13,w14,w15,w16,w17,w18,w19,w20,w21,w22,w23) values
302. (
303. testing\_ke
304. ,pAlpha
305. , pEps
306. , w1ns
307. , w2ns
308. , w3ns
309. , w4ns
310. , w5ns
311. , w6ns
312. , w7ns
313. , w8ns
314. , w9ns
315. , w10ns
316. , w11ns
317. , w12ns
318. , w13ns
319. , w14ns
320. , w15ns
321. , w16ns
322. , w17ns
323. , w18ns
324. , w19ns
325. , w20ns
326. , w21ns
327. , w22ns
328. , w23ns
329. );
330. -- END TRAINING
331. -- SELECT i\_training;
332. -- SELECT alpha;
333. WHILE i\_testing <= total\_testing DO
334. SELECT gender INTO w1t FROM tbldatatesting WHERE id=i\_testing;
335. SELECT customer\_type INTO w2t FROM tbldatatesting WHERE id=i\_testing;
336. SELECT age INTO w3t FROM tbldatatesting WHERE id=i\_testing;
337. SELECT type\_of\_travel INTO w4t FROM tbldatatesting WHERE id=i\_testing;
338. SELECT customer\_class INTO w5t FROM tbldatatesting WHERE id=i\_testing;
339. SELECT flight\_distance INTO w6t FROM tbldatatesting WHERE id=i\_testing;
340. SELECT inflight\_wifi\_service INTO w7t FROM tbldatatesting WHERE id=i\_testing;
341. SELECT departure\_arrival\_time\_convenient INTO w8t FROM tbldatatesting WHERE id=i\_testing;
342. SELECT ease\_of\_online\_booking INTO w9t FROM tbldatatesting WHERE id=i\_testing;
343. SELECT gate\_location INTO w10t FROM tbldatatesting WHERE id=i\_testing;
344. SELECT food\_and\_drink INTO w11t FROM tbldatatesting WHERE id=i\_testing;
345. SELECT online\_boarding INTO w12t FROM tbldatatesting WHERE id=i\_testing;
346. SELECT seat\_comfort INTO w13t FROM tbldatatesting WHERE id=i\_testing;
347. SELECT inflight\_entertainment INTO w14t FROM tbldatatesting WHERE id=i\_testing;
348. SELECT onboard\_service INTO w15t FROM tbldatatesting WHERE id=i\_testing;
349. SELECT leg\_room\_service INTO w16t FROM tbldatatesting WHERE id=i\_testing;
350. SELECT baggage\_handling INTO w17t FROM tbldatatesting WHERE id=i\_testing;
351. SELECT checkin\_service INTO w18t FROM tbldatatesting WHERE id=i\_testing;
352. SELECT inflight\_service INTO w19t FROM tbldatatesting WHERE id=i\_testing;
353. SELECT cleanliness INTO w20t FROM tbldatatesting WHERE id=i\_testing;
354. SELECT departure\_delay\_in\_minutes INTO w21t FROM tbldatatesting WHERE id=i\_testing;
355. SELECT arrival\_delay\_in\_minutes INTO w22t FROM tbldatatesting WHERE id=i\_testing;
356. SELECT satisfaction INTO w23t FROM tbldatatesting WHERE id=i\_testing;
357. SET ws = ed(
358. w1t, w1s
359. , w2t, w2s
360. , w3t, w3s
361. , w4t, w4s
362. , w5t, w5s
363. , w6t, w6s
364. , w7t, w7s
365. , w8t, w8s
366. , w9t, w9s
367. , w10t, w10s
368. , w11t, w11s
369. , w12t, w12s
370. , w13t, w13s
371. , w14t, w14s
372. , w15t, w15s
373. , w16t, w16s
374. , w17t, w17s
375. , w18t, w18s
376. , w19t, w19s
377. , w20t, w20s
378. , w21t, w21s
379. , w22t, w22s
380. );
381. SET wns = ed(
382. w1t, w1ns
383. , w2t, w2ns
384. , w3t, w3ns
385. , w4t, w4ns
386. , w5t, w5ns
387. , w6t, w6ns
388. , w7t, w7ns
389. , w8t, w8ns
390. , w9t, w9ns
391. , w10t, w10ns
392. , w11t, w11ns
393. , w12t, w12ns
394. , w13t, w13ns
395. , w14t, w14ns
396. , w15t, w15ns
397. , w16t, w16ns
398. , w17t, w17ns
399. , w18t, w18ns
400. , w19t, w19ns
401. , w20t, w20ns
402. , w21t, w21ns
403. , w22t, w22ns
404. );
405. SELECT satisfaction INTO info\_satisfaction FROM tbldatatesting WHERE id=i\_testing;
406. IF ws < wns THEN SET prediction = 1;
407. ELSEIF ws > wns THEN SET prediction = 0;
408. ELSE SET prediction = 1;
409. END IF;
410. IF info\_satisfaction = 0 AND prediction=0 THEN
411. SET tn\_lvq=tn\_lvq+1;
412. ELSEIF info\_satisfaction = 0 AND prediction=1 THEN
413. SET fp\_lvq=fp\_lvq+1;
414. ELSEIF info\_satisfaction = 1 AND prediction=0 THEN
415. SET fn\_lvq=fn\_lvq+1;
416. ELSEIF info\_satisfaction = 1 AND prediction=1 THEN
417. SET tp\_lvq=tp\_lvq+1;
418. END IF;
419. SET i\_testing = i\_testing + 1;
420. END WHILE;
422. UPDATE tblaccuracy SET total\_data=total\_testing
423. , tn=tn\_lvq
424. , fp=fp\_lvq
425. , fn=fn\_lvq
426. , tp=tp\_lvq
427. WHERE
428. id = (SELECT count(\*) FROM tblaccuracy)
429. ;
430. SELECT testing\_ke as 'selesai uji ke', pAlpha as 'a', pEps as 'eps';
431. END ##
432. DELIMITER ;
433. DROP PROCEDURE IF EXISTS processl;
434. DELIMITER ##
435. CREATE PROCEDURE processl(alpha FLOAT(30,20), eps FLOAT(30,20))
436. BEGIN
437. CALL lvq(1,alpha,eps);
438. CALL lvq(2,alpha,eps);
439. CALL lvq(3,alpha,eps);
440. CALL lvq(4,alpha,eps);
441. CALL lvq(5,alpha,eps);
442. UPDATE tblaccuracy SET accuracy = ((tp+tn)/(tp+tn+fp+fn))\*100 WHERE algoritma="LVQ";
443. SELECT \* FROM tblaccuracy WHERE id>((SELECT COUNT(\*) FROM tblaccuracy)-5);
444. END ##
445. DELIMITER ;
446. DROP PROCEDURE IF EXISTS processlvq;
447. DELIMITER ##
448. CREATE PROCEDURE processlvq()
449. BEGIN
450. CALL processl(0.9,0.0000001);
451. CALL processl(0.9,0.0001);
452. CALL processl(0.9,0.01);
453. CALL processl(0.1,0.0000001);
454. CALL processl(0.1,0.0001);
455. CALL processl(0.1,0.01);
456. CALL processl(0.01,0.0000001);
457. CALL processl(0.01,0.0001);
458. CALL processl(0.05,0.0000001);
459. CALL processl(0.05,0.0001);
461. END ##
462. DELIMITER ;