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
2022

HALAMAN PENGESAHAN

Judul Tugas Akhir:	:	COMPARISON LEARNING VECTOR QUANTIZATION AND NAIVE BAYES ALGORITHM IN AIRLINE PASSENGER SATISFACTION
Diajukan oleh	:	Nikodemus Galih Candra Wicakso
NIM	:	18.K1.0082
Tanggal disetujui	:	14 Januari 2022
Telah setujui oleh		
Pembimbing	:	Hironimus Leong S.Kom., M.Kom.
Penguji 1	:	Hironimus Leong S.Kom., M.Kom.
Penguji 2	:	R. Setiawan Aji Nugroho S.T., MCompIT., Ph.D
Penguji 3	:	Rosita Herawati S.T., M.I.T.
Penguji 4	:	Y.b. Dwi Setianto S.T., M.Cs.
Penguji 5	:	Yonathan Purbo Santosa S.Kom., M.Sc
Penguji 6	:	Yulianto Tejo Putranto S.T., M.T.
Ketua Program Studi	:	Rosita Herawati S.T., M.I.T.
Dekan	:	Dr. Bernardinus Harnadi S.T., M.T.

Halaman ini merupakan halaman yang sah dan dapat diverifikasi melalui alamat di bawah ini.

sintak.unika.ac.id/skripsi/verifikasi/?id=18.K1.0082

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, January, 20, 2022

NIKODEMUS GALIH CANDRA WICAKSONO

18.K1.0082

STATEMENT PAGE OF SCIENTIFIC PUBLICATIONS FOR ACADEMIC INTEREST

I, the undersigned:

Name : NIKODEMUS GALIH CANDRA WICAKSONO

Study Program : Informatics Engineering

Faculty : Computer Science

Type of Work : Thesis

Approve to grant Soegijapranata Catholic University Semarang Non-exclusive Royalty-Free Rights for the scientific works entitled "COMPARISON LEARNING VECTOR QUANTIZATION AND NAIVE BAYES ALGORITHM IN AIRLINE PASSENGER SATISFACTION" along with existing tools (if needed). With these rights, database form, maintain, and publish this final project as long as it includes my name as the writer/creator and as the copyright owner.

This statement was made truthfully.

Semarang, January, 20, 2022

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.

Semarang, January, 20, 2022

NIKODEMUS GALIH CANDRA WICAKSONO
18.K1.0082

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: naive_bayes, bayesian, learning_vector_quantization, lvq

TABLE OF CONTENTS

LIST OF FIGURE

LIST OF TABLE

CHAPTER 1 INTRODUCTION

1.1. 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.

1.2. 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?

1.3. 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%.

1.4. 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.

CHAPTER 2
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.

CHAPTER 3
RESEARCH METHODOLOGY

3.1. 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.

3.2. 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.

3.3. 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.

3.4. Analysis

In analyzing, I will do 5 tests as follows :

Table 3.1. 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.

3.5. 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.

CHAPTER 4
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:

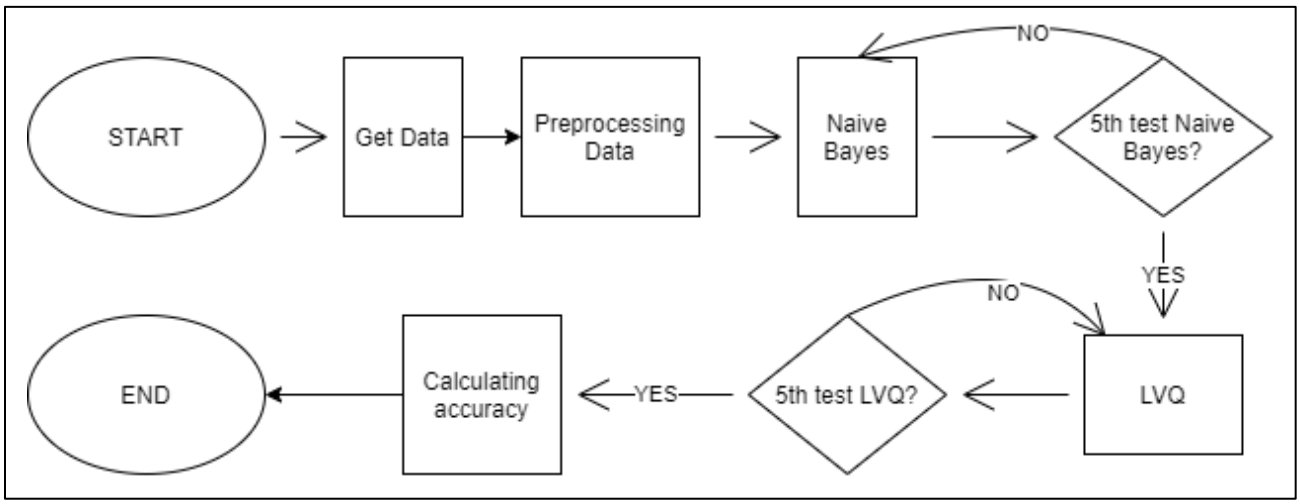

Figure 4.1 Workflow

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.

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:

Table 4.2. 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.

Table 4.3. 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.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP} + \mathrm{TN}} \text{ x } 100\% \quad (1)$$

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.

$$P(y|\boldsymbol{x}) = \frac{\mathrm{P(y)}\ \mathrm{P}(\mathbf{x}|\mathrm{y})}{P(\boldsymbol{x})} \quad (2)$$

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.

Table 4.4. 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.

Table 4.5. 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

Table 4.6. 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

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

$$P(x) = \frac{x}{Total\ data} \quad (3)$$

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:

$$P(0) = \frac{12}{18} = 0.67 \quad (4)$$

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

$$P(1) = \frac{6}{18} = 0.33 \quad (5)$$

3. 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.

$$P(a|x) = \frac{Total\ data\ ax}{Total\ data\ x} \quad (6)$$

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:

$$P(gender = 0|label = 0) = \frac{6}{12} = 0.5 \quad (7)$$

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

$$P(gender = 0|label = 1) = \frac{3}{6} = 0.5 \quad (8)$$

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).

4. 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:

$$label\ 0 = \mathrm{P}(\text{gender} = 0|\text{label} = 0)\ \text{x}\ \mathrm{P}(\text{customer type} = 1|\text{label} = 0)\ \text{x}\ \ldots\ldots\ \text{x}\ \mathrm{P}(\text{label} = 0)$$
$$= 0.5\ \text{x}\ 0.67\ x\ \ldots\ \text{x}\ 0.67$$
$$= 0.22445 \quad (9)$$

$$label\ 1 = \mathrm{P}(\text{gender} = 0|\text{label} = 1)\ \text{x}\ \mathrm{P}(\text{customer type} = 1|\text{label} = 1)\ \text{x}\ \ldots\ldots\ \text{x}\ \mathrm{P}(\text{label} = 1)$$
$$= 0.5\ \text{x}\ 1\ x\ \ldots\ \text{x}\ 0.33$$
$$= 0.165 \quad (10)$$

5. 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.

6. To find the accuracy results later then if:
    a. Label class “1” and prediction results is “1”, TP added 1
    b. Label class “0” and prediction result is “0”, TN added 1
    c. Label class “1” and prediction results is “0”, FN added 1
    d. 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.

7. 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.

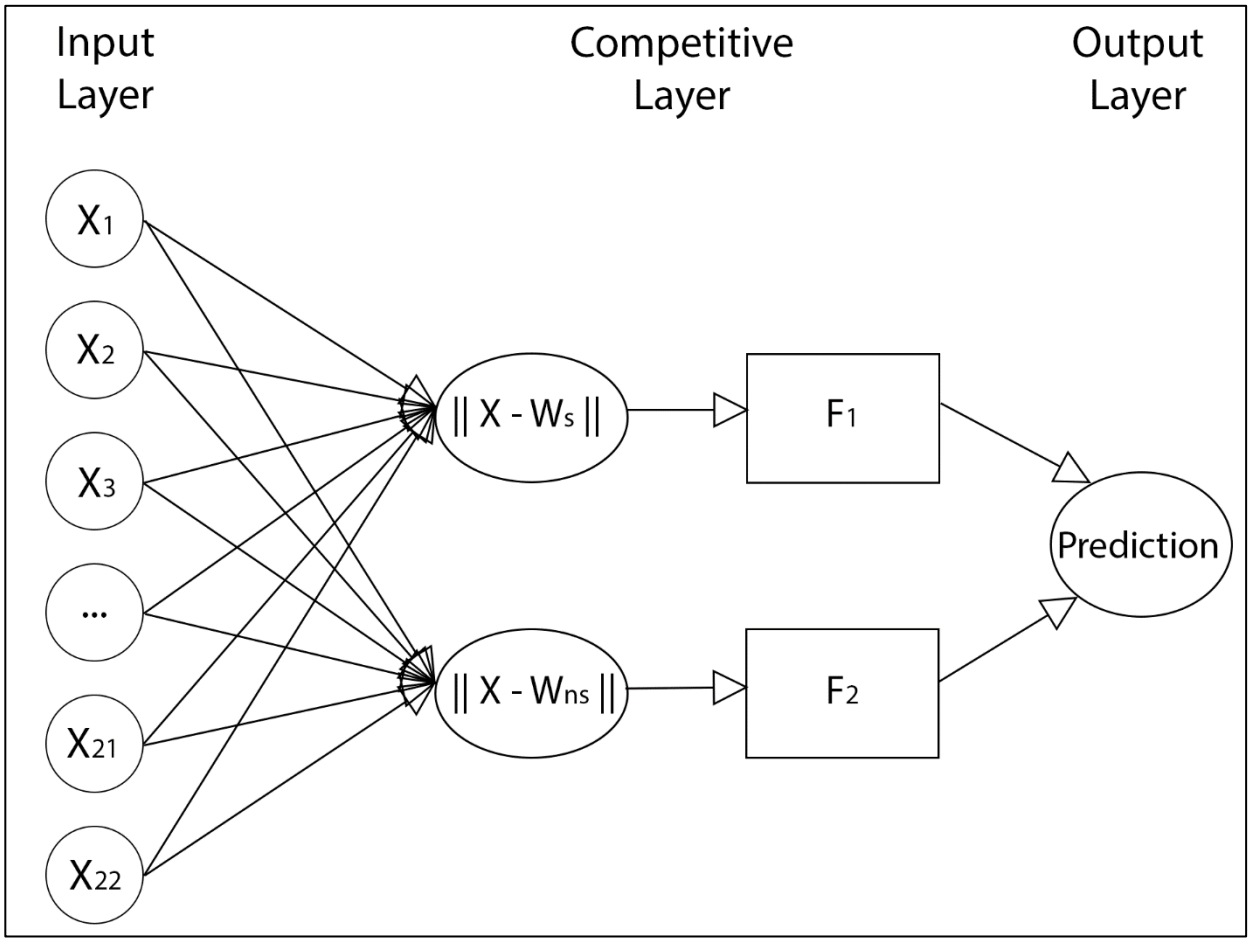

Figure 4.2 LVQ Architecture

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.

$$||X - Wc|| = \sqrt{\sum(Xn - Wcn)^2} \quad (11)$$

||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.

Table 4.7. 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.

Table 4.8. 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

Table 4.9. 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

2. Initialization

   a. 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.

Table 4.10. 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.

b. 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.

c. Epoch. Epoch initialization is 1.

d. Parameters learning rate/alpha (α). Alpha initialization is 0,9

e. Minimum error (Eps). Eps initialization is 0,0000001

3. Input

a. 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.

b. 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.

4. 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.

a. 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.

$$\left||X - Ws|\right| = \sqrt{(X1 - Ws1)^2 + (X2 - Ws2)^2 + \ (X3 - Ws3)^2 + \cdots}$$

$$\left||X - Ws|\right| = \sqrt{(1 - 0)^2 + (0 - 1)^2 + \ (0 - 0)^2 + \cdots}$$

$$\left||X - Ws|\right| = \sqrt{1 + 1 + 0} = 1{,}4142 \quad (12)$$

$$\left||X - Wns|\right| = \sqrt{(X1 - Wns1)^2 + (X2 - Wns2)^2 + \ (X3 - Wns3)^2 + \cdots}$$

$$\left||X - Wns|\right| = \sqrt{(1 - 1)^2 + (0 - 1)^2 + \ (0 - 0)^2 + \cdots}$$

$$\left||X - Wns|\right| = \sqrt{0 + 1 + 0} = 1 \quad (13)$$

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).

b. Update W_j for each W_n.

- If J= T then $W_j` = W_j + \alpha\ (X - W_j)$
- If J ≠ T then $W_j` = W_j - \alpha\ (X - W_j)$

T=Target

W_j = Weight class j

α = Learning ratio

j = prediction class

X = data value

W_n = 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.

c. Update the value of α.

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

$$\alpha` = \alpha - (\alpha * eps) \quad (14)$$

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.

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

5. 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.

6. 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:
   a. T class “1” and J class results is “1”, TP added 1
   b. T class “0” and J class results is “0”, TN added 1
   c. T class “1” and J class results is “0”, FN added 1
   d. 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.

7. 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.

CHAPTER 5
IMPLEMENTATION AND RESULTS

5.1. 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).

```
18.   INSERT INTO tbldataprocess
19.   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.

```
20. DELIMITER ##
21. CREATE PROCEDURE preprocessing()
22. BEGIN
23.  DECLARE i, iwhile, spinformation_int INT DEFAULT 0;
24.  DECLARE nama, spinformation, spinformation2 VARCHAR(255);
25.  DELETE FROM tbldataprocess
26.   WHERE
27.   Gender IS NULL or
28.   customer_type IS NULL or
29.   …
30.   satisfaction IS NULL;
31.  SET @num := 0;
32.  UPDATE tbldataprocess SET id = @num := (@num+1);
```

```
33.  ALTER TABLE tbldataprocess AUTO_INCREMENT =1;
34.  -- GENDER
35.  SELECT count(DISTINCT gender) into i from tbldataprocess;
36.  SET iwhile = 0;
37.  WHILE iwhile<>i DO
38.  SELECT DISTINCT gender INTO spinformation FROM tbldataprocess order by
   gender ASC limit iwhile, 1;
39.  UPDATE tbldataprocess set gender=iwhile where gender=spinformation;
40.  set iwhile= iwhile +1;
41.  END WHILE ;
42.  -- Customer Type
43.  SELECT count(DISTINCT customer_type) into i from tbldataprocess;
44.  SET iwhile = 0;
45.  WHILE iwhile <> i DO
46.  SELECT DISTINCT customer_type INTO spinformation FROM tbldataprocess
   order by customer_type ASC limit iwhile, 1;
47.  UPDATE tbldataprocess set customer_type=iwhile where customer_type =
   spinformation;
48.  set iwhile= iwhile +1;
49.  END WHILE ;
50.  -- AGE
51.  UPDATE tbldataprocess set age=0 where age <= 27;
52.  UPDATE tbldataprocess set age=1 where age > 27 and age <= 51;
53.  UPDATE tbldataprocess set age=2 where age > 51;
54.  -- Type Of Travel
55.  SELECT count(DISTINCT type_of_travel) into i from tbldataprocess;
56.  SET iwhile = 0;
57.  WHILE iwhile <> i DO
58.  SELECT DISTINCT type_of_travel INTO spinformation FROM tbldataprocess
   order by type_of_travel ASC limit iwhile, 1;
59.  UPDATE     tbldataprocess     set     type_of_travel=iwhile     where
   type_of_travel=spinformation;
60.  set iwhile= iwhile +1;
61.  END WHILE ;
62.  -- Customer Class
63.  SELECT count(DISTINCT customer_class) into i from tbldataprocess;
64.  SET iwhile = 0;
65.  WHILE iwhile <> i DO
66.  SELECT DISTINCT customer_class into spinformation FROM tbldataprocess
   order by customer_class ASC limit iwhile, 1;
67.  UPDATE     tbldataprocess     set     customer_class=iwhile     where
   customer_class=spinformation;
68.  set iwhile= iwhile +1;
69.  END WHILE ;
70.  -- FLIGHT DISTANCE
71.  UPDATE tbldataprocess set flight_distance=0 where flight_distance <=
   414;
72.  UPDATE tbldataprocess set flight_distance=1 where flight_distance > 414
   && flight_distance <= 1744;
73.  UPDATE tbldataprocess set flight_distance=2 where flight_distance >
   1744;
74.  -- Departure Delay In Minutes
75.  UPDATE    tbldataprocess    set    departure_delay_in_minutes=0    where
   departure_delay_in_minutes <= 12;
76.  UPDATE    tbldataprocess    set    departure_delay_in_minutes=1    where
   departure_delay_in_minutes > 12;
```

```
77.  -- Arrival Delay In Minutes
78.  UPDATE    tbldataprocess    set    arrival_delay_in_minutes=0    where
   arrival_delay_in_minutes <= 13;
79.  UPDATE    tbldataprocess    set    arrival_delay_in_minutes=1    where
   arrival_delay_in_minutes > 13;
80.  -- Satisfaction
81.  SELECT count(DISTINCT satisfaction) into i from tbldataprocess;
82.  SET iwhile = 0;
83.  WHILE iwhile <> i DO
84.  SELECT DISTINCT satisfaction into spinformation FROM tbldataprocess
   order by satisfaction ASC limit iwhile, 1;
85.  UPDATE      tbldataprocess      set      satisfaction=iwhile      where
   satisfaction=spinformation;
86.  set iwhile= iwhile +1;
87.  END WHILE ;
88. END ##
89. 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.

```
90. DELIMITER ##
91. CREATE PROCEDURE bayesian(number_of_testing INT)
92. BEGIN
93. DECLARE prob_satisfied, prob_gender_s, prob_customer_type_s,
94. prob_age_s, prob_type_of_travel_s, prob_customer_class_s,
95. prob_flight_distance_s, prob_inflight_wifi_service_s,
96. prob_departure_arrival_time_convenient_s, prob_ease_of_online_booking_s
97. , prob_gate_location_s, prob_food_and_drink_s, prob_online_boarding_s,
98. prob_seat_comfort_s, prob_inflight_entertainment_s,
99. prob_onboard_service_s, prob_leg_room_service_s,
100.prob_baggage_handling_s, prob_checkin_service_s,
101.prob_inflight_service_s, prob_cleanliness_s,
102.prob_departure_delay_in_minutes_s, prob_arrival_delay_in_minutes_s
103.FLOAT(30,30) DEFAULT 0;
104.DECLARE total_satisfied, total_notsatisfied FLOAT(30,20);
105.
106.DECLARE prob_notsatisfied, prob_gender_ns, prob_customer_type_ns,
107.prob_age_ns, prob_type_of_travel_ns, prob_customer_class_ns,
108.prob_flight_distance_ns, prob_inflight_wifi_service_ns,
109.prob_departure_arrival_time_convenient_ns,
110.prob_ease_of_online_booking_ns, prob_gate_location_ns,
111.prob_food_and_drink_ns, prob_online_boarding_ns, prob_seat_comfort_ns,
112.prob_inflight_entertainment_ns, prob_onboard_service_ns,
113.prob_leg_room_service_ns, prob_baggage_handling_ns,
114.prob_checkin_service_ns, prob_inflight_service_ns, prob_cleanliness_ns,
115.prob_departure_delay_in_minutes_ns, prob_arrival_delay_in_minutes_ns
```

```
FLOAT(30,30) DEFAULT 0;
DECLARE prediksi_s, prediksi_ns FLOAT(30,30);
DECLARE i, testing_ke, total_training, total_testing, i_testing,
total_data INT DEFAULT 0;
DECLARE info_satisfaction VARCHAR(2);

SELECT COUNT(*) INTO total_data FROM tbldataprocess;

SET testing_ke = number_of_testing;
IF testing_ke = 1 THEN SET total_training = 0.9 * total_data;
 ELSEIF testing_ke = 2 THEN SET total_training = 0.75 * total_data;
 ELSEIF testing_ke = 3 THEN SET total_training = 0.5 * total_data;
 ELSEIF testing_ke = 4 THEN SET total_training = 0.25 * total_data;
 ELSEIF testing_ke = 5 THEN SET total_training = 0.1 * total_data;
END IF;
SET total_testing = total_data-total_training;
SET i_testing = 1;
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;
UPDATE tblaccuracy SET total_data_training=total_training, total_data =
total_testing WHERE testing = testing_ke AND algoritma = "Bayesian";

TRUNCATE tbldatatesting;
TRUNCATE tbldatatraining;

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)
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
FROM tbldataprocess where id<= total_training;

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)
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
```

```
FROM tbldataprocess WHERE
id > total_training
AND id <= ( total_training + total_testing);

SET total_satisfied = (SELECT count(satisfaction) FROM tbldatatraining
   WHERE satisfaction = 1);
SET    total_notsatisfied    =    (SELECT    count(satisfaction)    FROM
   tbldatatraining WHERE satisfaction = 0);

SET prob_satisfied =  total_satisfied / total_training;
SET prob_notsatisfied =  total_notsatisfied / total_training;

-- WHILE per row tbldatatesting
WHILE i_testing <= total_testing DO
-- GENDER
SET prob_gender_s  = (SELECT count(gender) FROM tbldatatraining WHERE
   gender=(SELECT  gender  FROM  tbldatatesting  where  id=i_testing)  AND
   satisfaction =1) / total_satisfied;
SET prob_gender_ns = (SELECT count(gender) FROM tbldatatraining WHERE
   gender=(SELECT  gender  FROM  tbldatatesting  where  id=i_testing)  AND
   satisfaction =0) / total_notsatisfied;

-- Customer Type
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;
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;

…

-- Arrival Delay In Minutes
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;
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;

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;
```

```
172.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;
173.
174.SELECT satisfaction INTO info_satisfaction from tbldatatesting where id
   = i_testing;
175.
176.
177.IF info_satisfaction = 0 THEN -- actual not satisfied
178.  IF prediksi_s < prediksi_ns THEN
179.    UPDATE tblaccuracy SET tn=tn+1 WHERE algoritma="Bayesian" AND
180.    testing=testing_ke;
181.  ELSEIF prediksi_s > prediksi_ns THEN
182.    UPDATE tblaccuracy SET fp=fp+1 WHERE algoritma="Bayesian" AND
183.    testing=testing_ke;
184.  ELSEIF prediksi_s = 0 AND prediksi_ns = 0 THEN
185.    UPDATE tblaccuracy SET fnull=fnull+1 WHERE algoritma="Bayesian" AND
186.    testing=testing_ke;
187.  END IF;
188.ELSEIF info_satisfaction = 1 THEN -- actual satisfied
189.  IF prediksi_s < prediksi_ns THEN
190.    UPDATE tblaccuracy SET fn=fn+1 WHERE algoritma="Bayesian" AND
191.    testing=testing_ke;
192.  ELSEIF prediksi_s > prediksi_ns THEN
193.    UPDATE tblaccuracy SET tp=tp+1 WHERE algoritma="Bayesian" AND
194.    testing=testing_ke;
195.  ELSEIF prediksi_s = 0 AND prediksi_ns = 0 THEN
196.    UPDATE tblaccuracy SET tnull=tnull+1 WHERE algoritma="Bayesian" AND
197.    testing=testing_ke;
198.  END IF;
199.END IF;
200.SET i_testing = i_testing+1;
201.END WHILE;
202.
203.END ##
204.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.

```
205.DELIMITER ##
206.CREATE PROCEDURE processb()
207.BEGIN
208. CALL bayesian(1);
209. CALL bayesian(2);
210. CALL bayesian(3);
211. CALL bayesian(4);
212. CALL bayesian(5);
213. UPDATE          tblaccuracy          SET          accuracy          =
   ((tp+tn)/(tp+tn+fp+fn+tnull+fnull))*100 where algoritma='Bayesian';
214.END ##
215.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.

```
216.DELIMITER ##
217.CREATE FUNCTION ed(
218.  w1t FLOAT(30,20)
219.  , w1 FLOAT(30,20)
220.  ……
221.  , w22t FLOAT(30,20)
222.  , w22 FLOAT(30,20)
223.  )
224.RETURNS FLOAT(30,20)
225.BEGIN
226.  DECLARE hasil FLOAT(30,20) DEFAULT 0;
227.  SET hasil = SQRT((
228.    POWER((w1t - w1),2)
229.    + POWER((w2t - w2),2)
230.    + POWER((w3t - w3),2)
231.    ……
232.    + POWER((w22t - w22),2)));
233.RETURN(hasil);
```

```
234.END; ##
235.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.

```
236.DELIMITER ##
237.CREATE PROCEDURE lvq(number_of_testing INT, pAlpha FLOAT(30,20), pEps
   FLOAT(30,20))
238.BEGIN
239.-- Weight of class satisfied
240.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;
241.-- Weight of class neutral or dissatisfied
242.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;
243.-- Weight of training
244.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;
245.-- Get id for initial class satisfied and not
246.DECLARE ids, idns, cj, t, epoch, maxepoch,pepoch INT DEFAULT 0;
247.
248.DECLARE tn_lvq, fp_lvq, fn_lvq, tp_lvq INT DEFAULT 0;
249.
250.DECLARE ws, wns, wt FLOAT(30,20) DEFAULT 0;
251.DECLARE alpha, eps, err, temp_alpha FLOAT(30,20) DEFAULT 0;
252.DECLARE info_satisfaction VARCHAR(2);
253.
254.DECLARE prediction INT DEFAULT 0;
255.DECLARE i, testing_ke, total_training, total_testing, i_testing,
   i_training, total_data INT DEFAULT 0;
256.
257.SELECT COUNT(*) INTO total_data FROM tbldataprocess;
258.SET testing_ke = number_of_testing;
259.SET maxepoch=5;
260.IF testing_ke = 1 THEN SET total_training = 0.9 * total_data;
261.  ELSEIF testing_ke = 2 THEN SET total_training = 0.75 * total_data;
262.  ELSEIF testing_ke = 3 THEN SET total_training = 0.5 * total_data;
263.  ELSEIF testing_ke = 4 THEN SET total_training = 0.25 * total_data;
264.  ELSEIF testing_ke = 5 THEN SET total_training = 0.1 * total_data;
265.END IF;
266.TRUNCATE tbldatatesting;
267.TRUNCATE tbldatatraining;
268.SET total_testing = total_data-total_training;
269.SET i_testing = 1;
270.SET i_training = 1;
271.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)
```

```
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
FROM tbldataprocess where id<= total_training;

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)
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
FROM tbldataprocess WHERE
id > total_training
AND id <= ( total_training + total_testing);

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);

-- INITIALITATION
SET alpha = pAlpha;
SET eps = pEps;

SELECT id INTO ids FROM tbldatatraining WHERE satisfaction = 1 ORDER BY
   RAND() LIMIT 1;
SELECT id INTO idns FROM tbldatatraining WHERE satisfaction = 0 ORDER
   BY RAND() LIMIT 1;

SELECT gender INTO w1s FROM tbldatatraining WHERE id=ids;
SELECT gender INTO w1ns FROM tbldatatraining WHERE id=idns;
……
SELECT arrival_delay_in_minutes INTO w22s FROM tbldatatraining WHERE
   id=ids;
SELECT arrival_delay_in_minutes INTO w22ns FROM tbldatatraining WHERE
   id=idns;
-- row used to initialitation is not use again
DELETE FROM tbldatatraining WHERE id=ids;
DELETE FROM tbldatatraining WHERE id=idns;

SET  @num := 0;
UPDATE tbldatatraining SET id = @num := (@num+1);
ALTER TABLE tbldatatraining AUTO_INCREMENT =1;
-- END INITIALITATION
```

```
304.-- TRAINING
305.SET temp_alpha = alpha;
306.WHILE epoch < maxepoch DO
307.SET i_training = 0;
308.SET temp_i_training = 0;
309.SET alpha=temp_alpha;
310. algolvq: WHILE (i_training <= total_training) or (alpha >= eps) DO
311.  IF(alpha>=eps)THEN
312.  SELECT gender INTO w1t FROM tbldatatraining WHERE id=i_training;
313.  ……
314.  SELECT satisfaction INTO w23t FROM tbldatatraining WHERE
   id=i_training;
315.
316.  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);
317.  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);
318.
319.  IF ws < wns THEN
320.    SET cj = 1;
321.  ELSEIF ws > wns THEN
322.    SET cj = 0;
323.  ELSE
324.    SET cj = 1;
325.  END IF;
326.
327.  SELECT satisfaction INTO t FROM tbldatatraining WHERE id=i_training;
328.
329.  IF cj = 1 AND t = 1 THEN
330.    SET w1s = w1s + (alpha * (w1t - w1s));
331.    ……
332.    SET w22s = w22s + (alpha * (w22t - w22s));
333.  ELSEIF cj = 0 AND t = 0 THEN
334.    SET w1ns = w1ns + (alpha * (w1t - w1ns));
335.    ……
336.    SET w22ns = w22ns + (alpha * (w22t - w22ns));
337.  ELSEIF cj = 0 AND t = 1 THEN
338.    SET w1ns = w1ns - (alpha * (w1t - w1ns));
339.    ……
340.    SET w22ns = w22ns - (alpha * (w22t - w22ns));
341.  ELSEIF cj = 1 AND t = 0 THEN
342.    SET w1s = w1s - (alpha * (w1t - w1s));
343.    ……
344.    SET w22s = w22s - (alpha * (w22t - w22s));
345.  END IF;
346.
347.  SET alpha = alpha - (alpha * eps);
348.  UPDATE tblaccuracy SET total_data_training = i_training+1 WHERE
   id=(SELECT COUNT(*) FROM tblaccuracy);
349.  SET i_training = i_training + 1;
350.  SET temp_i_training = i_training;
351.  IF (i_training = total_training) THEN
```

```
352.    SET temp_alpha = alpha;
353.    SET alpha = eps;
354.  ELSEIF (alpha <= eps) THEN
355.    SET temp_alpha = alpha;
356.    SELECT alpha as 'alphaa', i_training as 't';
357.    SET i_training = total_training +1;
358.  END IF;
359.  ELSE
360.    LEAVE algolvq;
361.  END IF;
362.  END WHILE;
363.    SET epoch = epoch + 1;
364.  IF (temp_i_training <> 0) THEN
365.    SET ptotal_data_training=ptotal_data_training + temp_i_training;
366.    SET pepoch=epoch;
367.  END IF;
368.  END WHILE;
369.WHILE i_testing <= total_testing DO
370.  SELECT gender INTO w1t FROM tbldatatesting WHERE id=i_testing;
371.  ……
372.  SELECT arrival_delay_in_minutes INTO w22t FROM tbldatatesting WHERE
   id=i_testing;
373.  SELECT satisfaction INTO w23t FROM tbldatatesting WHERE id=i_testing;
374.
375.  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);
376.
377.  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);
378.
379.  SELECT satisfaction INTO info_satisfaction FROM tbldatatesting WHERE
   id=i_testing;
380.  IF ws < wns THEN SET prediction = 1;
381.    ELSEIF ws > wns THEN SET prediction = 0;
382.    ELSE SET prediction = 1;
383.  END IF;
384.  IF info_satisfaction = 0 AND prediction=0 THEN
385.    SET tn_lvq=tn_lvq+1;
386.  ELSEIF info_satisfaction = 0 AND prediction=1  THEN
387.    SET fp_lvq=fp_lvq+1;
388.  ELSEIF info_satisfaction = 1 AND prediction=0 THEN
389.    SET fn_lvq=fn_lvq+1;
390.  ELSEIF info_satisfaction = 1 AND prediction=1 THEN
391.    SET tp_lvq=tp_lvq+1;
392.  END IF;
393.  SET i_testing = i_testing + 1;
394.END WHILE;
395.
396.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);
397.END ##
398.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.

```
399.DELIMITER ##
400.CREATE PROCEDURE processl(alpha FLOAT(30,20), eps FLOAT(30,20))
401.BEGIN
402.  CALL lvq(1,alpha,eps);
403.  CALL lvq(2,alpha,eps);
404.  CALL lvq(3,alpha,eps);
405.  CALL lvq(4,alpha,eps);
406.  CALL lvq(5,alpha,eps);
407.   UPDATE tblaccuracy SET accuracy = ((tp+tn)/(tp+tn+fp+fn))*100 WHERE
   algoritma="LVQ";
408.END ##
409.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.

```
410.DELIMITER ##
411.CREATE PROCEDURE process()
412.BEGIN
413.  CALL processb();
414.  CALL processl(0.9,0.0000001);
415.  CALL processl(0.9,0.0001);
416.  CALL processl(0.9,0.01);
417.  CALL processl(0.1,0.0000001);
418.  CALL processl(0.1,0.0001);
419.  CALL processl(0.1,0.01);
420.  CALL processl(0.01,0.0000001);
421.  CALL processl(0.01,0.0001);
422.  CALL processl(0.05,0.0000001);
423.  CALL processl(0.05,0.0001);
424.END ##
425.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.

5.2. 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:

Table 5.1. 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.

Table 5.2. 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%

Table 5.3. 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%

Table 5.4. 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%

Table 5.5. 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%

Table 5.6. 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%

Table 5.7. 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%

Table 5.8. 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%

Table 5.9. 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.

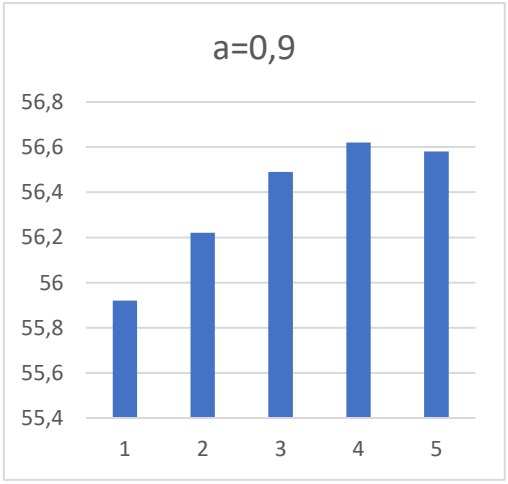

Figure 5.1 Graph of accuracy LVQ alpha=0.9

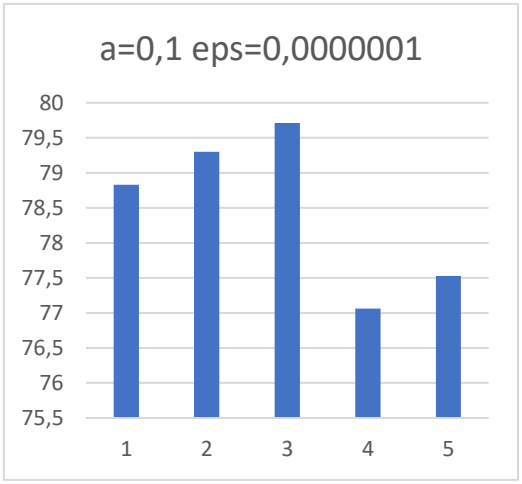

Figure 5.2 Graph of accuracy LVQ alpha=0.1 and eps=0.0000001

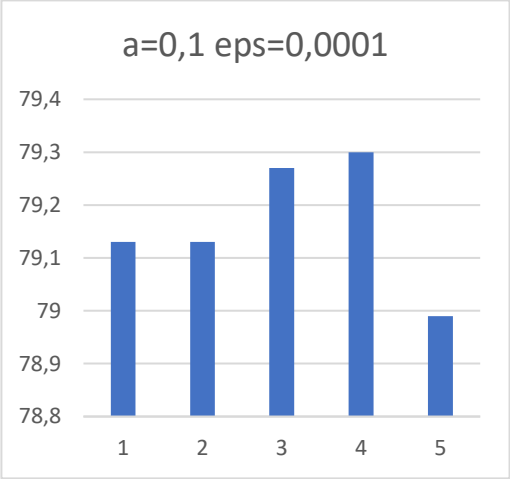

Figure 5.3 Graph of accuracy LVQ alpha=0.1 and eps=0.0001

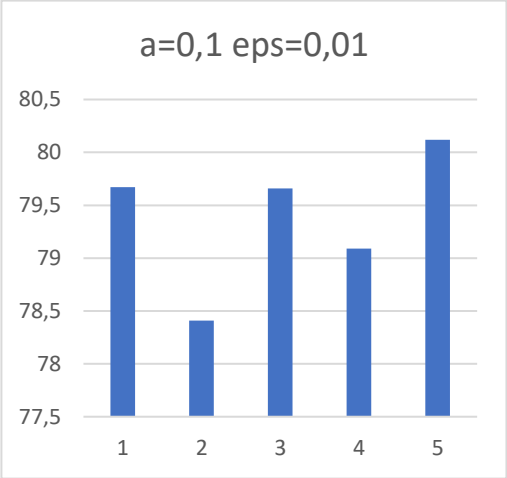

Figure 5.4 Graph of accuracy alpha=0.1 and eps=0.01

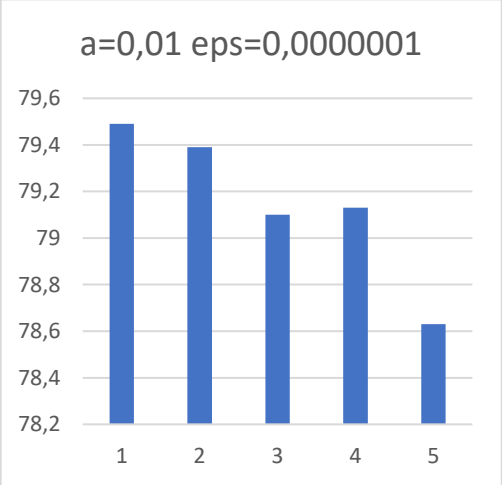

Figure 5.5 Graph of accuracy alpha=0.01 and eps=0.0000001

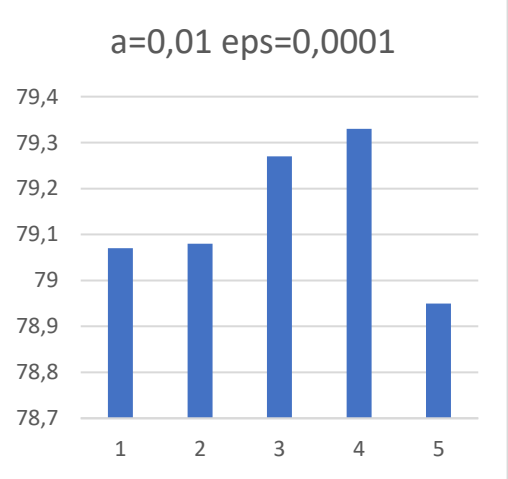

Figure 5.6 Graph of accuracy alpha=0.01 and eps=0.0001

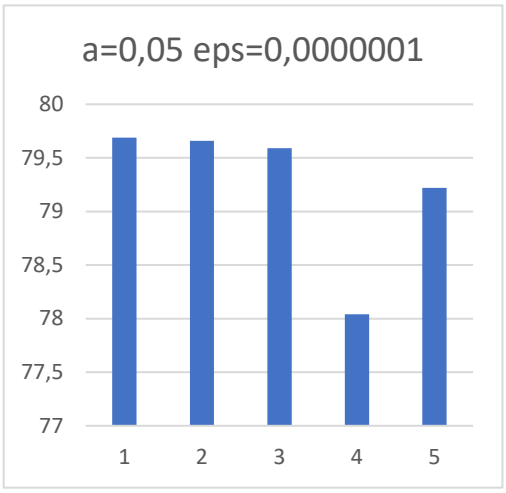

Figure 5.7 Graph of accuracy alpha=0.05 and eps=0.0000001

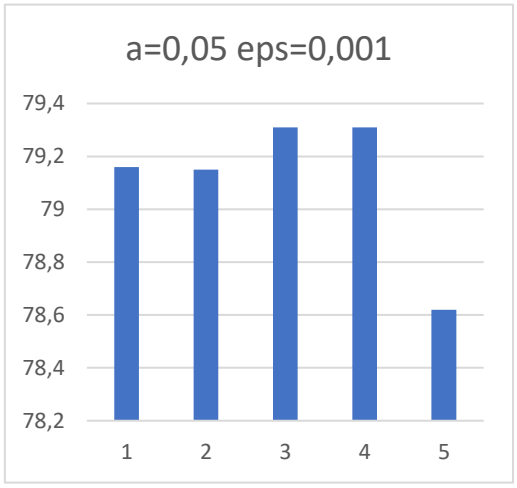

Figure 5.8 Graph of accuracy alpha=0.05 and eps=0.0001

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.

CHAPTER 6
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). From the results of the study, with an alpha of 0.9 and different eps produce the same result. However, some tests with other alphas, namely 0.1, 0.01, and 0.05 produced different results when the eps were changed. With the same data, LVQ is implemented but with the aim of finding the most optimal value. The combination of alpha and eps at the most optimal so that it can get the best accuracy for this airline's flight satisfaction data.

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

```
426.  DROP DATABASE IF EXISTS dbaps;
427.  CREATE DATABASE dbaps;
428.  USE dbaps;
429.
430.  DROP TABLE IF EXISTS tbldata;
431.  CREATE TABLE tbldata(
432.      id INT PRIMARY KEY AUTO_INCREMENT
433.      , Gender VARCHAR(10)
434.      , customer_type varchar(50)
435.      , age varchar(10)
436.      , type_of_travel VARCHAR(50)
437.      , customer_class VARCHAR(50)
438.      , flight_distance varchar(10)
439.      , inflight_wifi_service varchar(10)
440.      , departure_arrival_time_convenient varchar(10)
441.      , ease_of_online_booking varchar(10)
442.      , gate_location varchar(10)
443.      , food_and_drink varchar(10)
444.      , online_boarding varchar(10)
445.      , seat_comfort varchar(10)
446.      , inflight_entertainment varchar(10)
447.      , onboard_service varchar(10)
448.      , leg_room_service varchar(10)
449.      , baggage_handling varchar(10)
450.      , checkin_service varchar(10)
451.      , inflight_service varchar(10)
452.      , cleanliness varchar(10)
453.      , departure_delay_in_minutes varchar(10)
454.      , arrival_delay_in_minutes varchar(10)
455.      , satisfaction VARCHAR(50)
456.  ) engine=InnoDB;
457.
458.  LOAD DATA LOCAL INFILE 'airline_passenger_satisfaction.csv'
459.      INTO TABLE tbldata
460.      FIELDS TERMINATED BY ','
461.      ENCLOSED BY ''
462.      LINES TERMINATED BY '\n'
463.      IGNORE 1 LINES
464.      (id,    Gender,    customer_type,    age,    type_of_travel,
customer_class,
465.      flight_distance,                          inflight_wifi_service,
departure_arrival_time_convenient,
466.      ease_of_online_booking,     gate_location,     food_and_drink,
online_boarding,
467.      seat_comfort,     inflight_entertainment,     onboard_service,
leg_room_service,
468.      baggage_handling,      checkin_service,      inflight_service,
cleanliness,
469.      departure_delay_in_minutes,          arrival_delay_in_minutes,
satisfaction)
470.      SET
```

```
471.          Gender = IF(Gender = '', null, Gender)
472.          ,   customer_type   =   IF(customer_type   =   '',   null,
customer_type)
473.          , age = IF(age = '', null, age)
474.          ,   type_of_travel   =   IF(type_of_travel   =   '',   null,
type_of_travel)
475.          ,   customer_class   =   IF(customer_class   =   '',   null,
customer_class)
476.          ,  flight_distance   =   IF(flight_distance   =   '',   null,
flight_distance)
477.          , inflight_wifi_service = IF(inflight_wifi_service = '',
null, inflight_wifi_service)
478.          ,            departure_arrival_time_convenient           =
IF(departure_arrival_time_convenient         =         '',         null,
departure_arrival_time_convenient)
479.          , ease_of_online_booking = IF(ease_of_online_booking = '',
null, ease_of_online_booking)
480.          ,   gate_location   =   IF(gate_location   =   '',   null,
gate_location)
481.          ,   food_and_drink   =   IF(food_and_drink   =   '',   null,
food_and_drink)
482.          ,  online_boarding   =   IF(online_boarding   =   '',   null,
online_boarding)
483.          , seat_comfort = IF(seat_comfort = '', null, seat_comfort)
484.          , inflight_entertainment = IF(inflight_entertainment = '',
null, inflight_entertainment)
485.          ,  onboard_service   =   IF(onboard_service   =   '',   null,
onboard_service)
486.          ,  leg_room_service   =   IF(leg_room_service   =   '',   null,
leg_room_service)
487.          ,  baggage_handling   =   IF(baggage_handling   =   '',   null,
baggage_handling)
488.          ,  checkin_service   =   IF(checkin_service   =   '',   null,
checkin_service)
489.          ,  inflight_service   =   IF(inflight_service   =   '',   null,
inflight_service)
490.          , cleanliness = IF(cleanliness = '', null, cleanliness)
491.          ,                departure_delay_in_minutes                =
IF(departure_delay_in_minutes = '', null, departure_delay_in_minutes)
492.          , arrival_delay_in_minutes = IF(arrival_delay_in_minutes =
'', null, arrival_delay_in_minutes)
493.          , satisfaction = IF(satisfaction = '', null, satisfaction)
494.      ;
495.
496.  INSERT INTO tbldataprocess
497.  SELECT * FROM tbldata
;
```

CODING TABLE DATA PROCESS

```
498.DROP TABLE IF EXISTS tbldataprocess;
499.CREATE TABLE tbldataprocess(
500.    id INT PRIMARY KEY
501.    , Gender VARCHAR(10)
502.    , customer_type varchar(50)
503.    , age INT
504.    , type_of_travel VARCHAR(50)
```

```
    , customer_class VARCHAR(50)
    , flight_distance INT
    , inflight_wifi_service INT
    , departure_arrival_time_convenient INT
    , ease_of_online_booking INT
    , gate_location INT
    , food_and_drink INT
    , online_boarding INT
    , seat_comfort INT
    , inflight_entertainment INT
    , onboard_service INT
    , leg_room_service INT
    , baggage_handling INT
    , checkin_service INT
    , inflight_service INT
    , cleanliness INT
    , departure_delay_in_minutes INT
    , arrival_delay_in_minutes INT
    , satisfaction VARCHAR(50)
) engine=InnoDB;

INSERT INTO tbldataprocess
SELECT * FROM tbldata
;
```

PROCEDURE PREPROCESSING

```
DROP PROCEDURE IF EXISTS preprocessing;
DELIMITER ##
    CREATE PROCEDURE preprocessing()
    BEGIN
    DECLARE i, iwhile, spinformation_int INT DEFAULT 0;
    DECLARE nama, spinformation, spinformation2 VARCHAR(255);

    DELETE FROM tbldataprocess
    WHERE
           Gender IS NULL or
           customer_type IS NULL or
           age IS NULL or
           type_of_travel IS NULL or
           customer_class IS NULL or
           flight_distance IS NULL or
           inflight_wifi_service IS NULL or
           departure_arrival_time_convenient IS NULL or
           ease_of_online_booking IS NULL or
           gate_location IS NULL or
           food_and_drink IS NULL or
           online_boarding IS NULL or
           seat_comfort IS NULL or
           inflight_entertainment IS NULL or
           onboard_service IS NULL or
           leg_room_service IS NULL or
           baggage_handling IS NULL or
           checkin_service IS NULL or
           inflight_service IS NULL or
           cleanliness IS NULL or
```

```
            departure_delay_in_minutes IS NULL or
            arrival_delay_in_minutes IS NULL or
            satisfaction IS NULL
        ;

        SET  @num := 0;
        UPDATE tbldataprocess SET id = @num := (@num+1);
        ALTER TABLE tbldataprocess AUTO_INCREMENT =1;

        -- GENDER
        SELECT count(DISTINCT gender) into i from tbldataprocess;
        SET iwhile = 0;
        WHILE iwhile<>i DO
            SELECT DISTINCT gender INTO spinformation FROM tbldataprocess order by gender ASC limit iwhile, 1;
            UPDATE tbldataprocess set gender=iwhile where gender=spinformation;
        set iwhile= iwhile +1;
        END WHILE ;

        -- Customer Type
        SELECT count(DISTINCT customer_type) into i from tbldataprocess;
        SET iwhile = 0;
        WHILE iwhile <> i DO
            SELECT DISTINCT customer_type INTO spinformation FROM tbldataprocess order by customer_type ASC limit iwhile, 1;
            UPDATE tbldataprocess set customer_type=iwhile where customer_type=spinformation;
        set iwhile= iwhile +1;
        END WHILE ;

        -- AGE
        UPDATE tbldataprocess set age=0 where age <= 27;
        UPDATE tbldataprocess set age=1 where age > 27 and age <= 51;
        UPDATE tbldataprocess set age=2 where age > 51;

        -- Type Of Travel
        SELECT count(DISTINCT type_of_travel) into i from tbldataprocess;
        SET iwhile = 0;
        WHILE iwhile <> i DO
            SELECT DISTINCT type_of_travel INTO spinformation FROM tbldataprocess order by type_of_travel ASC limit iwhile, 1;
            UPDATE tbldataprocess set type_of_travel=iwhile where type_of_travel=spinformation;
        set iwhile= iwhile +1;
        END WHILE ;

        -- Customer Class
        SELECT count(DISTINCT customer_class) into i from tbldataprocess;
        SET iwhile = 0;
        WHILE iwhile <> i DO
            SELECT DISTINCT customer_class into spinformation FROM tbldataprocess order by customer_class ASC limit iwhile, 1;
```

```
76.           UPDATE tbldataprocess set customer_class=iwhile where
   customer_class=spinformation;
77.       set iwhile= iwhile +1;
78.       END WHILE ;
79.
80.       -- FLIGHT DISTANCE
81.       UPDATE    tbldataprocess    set    flight_distance=0    where
   flight_distance <= 414;
82.       UPDATE    tbldataprocess    set    flight_distance=1    where
   flight_distance > 414 && flight_distance <= 1744;
83.       UPDATE    tbldataprocess    set    flight_distance=2    where
   flight_distance > 1744;
84.
85.       -- Inflight Wifi Service
86.       SELECT  count(DISTINCT  inflight_wifi_service)  into  i  from
   tbldataprocess;
87.       SET iwhile = 0;
88.       WHILE iwhile <> i DO
89.           SELECT      DISTINCT      inflight_wifi_service      into
   spinformation_int FROM tbldataprocess order by inflight_wifi_service
   ASC limit iwhile, 1;
90.       set iwhile= iwhile +1;
91.       END WHILE;
92.
93.       -- Departure Arrival Time Convenient
94.       SELECT count(DISTINCT departure_arrival_time_convenient) into
   i from tbldataprocess;
95.       SET iwhile = 0;
96.       WHILE iwhile <> i DO
97.           SELECT  DISTINCT  departure_arrival_time_convenient  into
   spinformation_int       FROM       tbldataprocess       order       by
   departure_arrival_time_convenient ASC limit iwhile, 1;
98.       set iwhile= iwhile +1;
99.       END WHILE;
100.
101.      -- Ease of Online Booking
102.      SELECT  count(DISTINCT  ease_of_online_booking)  into  i  from
   tbldataprocess;
103.      SET iwhile = 0;
104.      WHILE iwhile <> i DO
105.          SELECT      DISTINCT      ease_of_online_booking      into
   spinformation_int FROM tbldataprocess order by ease_of_online_booking
   ASC limit iwhile, 1;
106.      set iwhile= iwhile +1;
107.      END WHILE;
108.
109.      -- Gate Location
110.      SELECT    count(DISTINCT    gate_location)    into    i    from
   tbldataprocess;
111.      SET iwhile = 0;
112.      WHILE iwhile <> i DO
113.          SELECT DISTINCT gate_location into spinformation_int FROM
   tbldataprocess order by gate_location ASC limit iwhile, 1;
114.      set iwhile= iwhile +1;
115.      END WHILE;
116.
```

```
117.      -- Food and Drink
118.      SELECT    count(DISTINCT    food_and_drink)    into    i    from
   tbldataprocess;
119.      SET iwhile = 0;
120.      WHILE iwhile <> i DO
121.          SELECT DISTINCT food_and_drink into spinformation_int FROM
   tbldataprocess order by food_and_drink ASC limit iwhile, 1;
122.      set iwhile= iwhile +1;
123.      END WHILE;
124.
125.      -- Online Boarding
126.      SELECT    count(DISTINCT    online_boarding)    into    i    from
   tbldataprocess;
127.      SET iwhile = 0;
128.      WHILE iwhile <> i DO
129.          SELECT DISTINCT online_boarding into spinformation_int FROM
   tbldataprocess order by online_boarding ASC limit iwhile, 1;
130.      set iwhile= iwhile +1;
131.      END WHILE;
132.
133.      -- Seat Comfort
134.      SELECT count(DISTINCT seat_comfort) into i from tbldataprocess;
135.      SET iwhile = 0;
136.      WHILE iwhile <> i DO
137.          SELECT DISTINCT seat_comfort into spinformation_int FROM
   tbldataprocess order by seat_comfort ASC limit iwhile, 1;
138.      set iwhile= iwhile +1;
139.      END WHILE;
140.
141.      -- Inflight Enterteinment
142.      SELECT  count(DISTINCT  inflight_entertainment)  into  i  from
   tbldataprocess;
143.      SET iwhile = 0;
144.      WHILE iwhile <> i DO
145.          SELECT      DISTINCT      inflight_entertainment      into
   spinformation_int FROM tbldataprocess order by inflight_entertainment
   ASC limit iwhile, 1;
146.      set iwhile= iwhile +1;
147.      END WHILE;
148.
149.      -- Onboard Service
150.      SELECT    count(DISTINCT    onboard_service)    into    i    from
   tbldataprocess;
151.      SET iwhile = 0;
152.      WHILE iwhile <> i DO
153.          SELECT DISTINCT onboard_service into spinformation_int FROM
   tbldataprocess order by onboard_service ASC limit iwhile, 1;
154.      set iwhile= iwhile +1;
155.      END WHILE;
156.
157.      -- Leg Room Service
158.      SELECT    count(DISTINCT    leg_room_service)    into    i    from
   tbldataprocess;
159.      SET iwhile = 0;
160.      WHILE iwhile <> i DO
```

```
161.          SELECT DISTINCT leg_room_service into spinformation_int
   FROM tbldataprocess order by leg_room_service ASC limit iwhile, 1;
162.      set iwhile= iwhile +1;
163.      END WHILE;
164.
165.      -- Baggage Handling
166.      SELECT   count(DISTINCT   baggage_handling)   into   i   from
   tbldataprocess;
167.      SET iwhile = 0;
168.      WHILE iwhile <> i DO
169.          SELECT DISTINCT baggage_handling into spinformation_int
   FROM tbldataprocess order by baggage_handling ASC limit iwhile, 1;
170.      set iwhile= iwhile +1;
171.      END WHILE;
172.
173.      -- Checkin Service
174.      SELECT   count(DISTINCT   checkin_service)   into   i   from
   tbldataprocess;
175.      SET iwhile = 0;
176.      WHILE iwhile <> i DO
177.          SELECT DISTINCT checkin_service into spinformation_int FROM
   tbldataprocess order by checkin_service ASC limit iwhile, 1;
178.      set iwhile= iwhile +1;
179.      END WHILE;
180.
181.      -- Inflight Service
182.      SELECT   count(DISTINCT   inflight_service)   into   i   from
   tbldataprocess;
183.      SET iwhile = 0;
184.      WHILE iwhile <> i DO
185.          SELECT DISTINCT inflight_service into spinformation_int
   FROM tbldataprocess order by inflight_service ASC limit iwhile, 1;
186.      set iwhile= iwhile +1;
187.      END WHILE;
188.
189.      -- Cleanliness
190.      SELECT count(DISTINCT cleanliness) into i from tbldataprocess;
191.      SET iwhile = 0;
192.      WHILE iwhile <> i DO
193.          SELECT DISTINCT cleanliness into spinformation_int FROM
   tbldataprocess order by cleanliness ASC limit iwhile, 1;
194.      set iwhile= iwhile +1;
195.      END WHILE;
196.
197.      -- Departure Delay In Minutes
198.      UPDATE tbldataprocess set departure_delay_in_minutes=0 where
   departure_delay_in_minutes <= 12;
199.      UPDATE tbldataprocess set departure_delay_in_minutes=1 where
   departure_delay_in_minutes > 12;
200.
201.      -- Arrival Delay In Minutes
202.      UPDATE  tbldataprocess  set  arrival_delay_in_minutes=0  where
   arrival_delay_in_minutes <= 13;
203.      UPDATE  tbldataprocess  set  arrival_delay_in_minutes=1  where
   arrival_delay_in_minutes > 13;
204.
```

```
205.      -- Satisfaction
206.      SELECT count(DISTINCT satisfaction) into i from tbldataprocess;
207.      SET iwhile = 0;
208.      WHILE iwhile <> i DO
209.          SELECT DISTINCT satisfaction into spinformation FROM
   tbldataprocess order by satisfaction ASC limit iwhile, 1;
210.          UPDATE tbldataprocess set satisfaction=iwhile where
   satisfaction=spinformation;
211.      set iwhile= iwhile +1;
212.      END WHILE ;
213.
214.      END ##
215.  DELIMITER ;
216.
217.  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.
29.   DROP TABLE IF EXISTS tbldatatesting;
30.   CREATE TABLE tbldatatesting(
31.       id INT PRIMARY KEY AUTO_INCREMENT
32.       , Gender VARCHAR(10)
33.       , customer_type varchar(50)
34.       , age INT
35.       , type_of_travel VARCHAR(50)
36.       , customer_class VARCHAR(50)
37.       , flight_distance INT
38.       , inflight_wifi_service INT
```

```
39.       , departure_arrival_time_convenient INT
40.       , ease_of_online_booking INT
41.       , gate_location INT
42.       , food_and_drink INT
43.       , online_boarding INT
44.       , seat_comfort INT
45.       , inflight_entertainment INT
46.       , onboard_service INT
47.       , leg_room_service INT
48.       , baggage_handling INT
49.       , checkin_service INT
50.       , inflight_service INT
51.       , cleanliness INT
52.       , departure_delay_in_minutes INT
53.       , arrival_delay_in_minutes INT
54.       , satisfaction VARCHAR(50)
55.       , prediksi VARCHAR(255)
56.   ) engine=InnoDB;
57.
58.
59.   CREATE TABLE tblaccuracy (
60.       id INT PRIMARY KEY AUTO_INCREMENT
61.       , algoritma VARCHAR(15)
62.       , testing INT
63.       , total_data_training INT
64.       , total_data INT
65.       , total_training INT
66.       , tp INT
67.       , tn INT
68.       , fp INT
69.       , fn INT
70.       , tnull INT
71.       , fnull INT
72.       , accuracy FLOAT(4,2)
73.   );
74.
75.   CREATE TABLE tblW(
76.       id INT PRIMARY KEY AUTO_INCREMENT
77.       ,testing INT
78.       ,alpha FLOAT(30,20)
79.       ,eps FLOAT(30,20)
80.       ,w1 FLOAT(30,20)
81.       ,w2 FLOAT(30,20)
82.       ,w3 FLOAT(30,20)
83.       ,w4 FLOAT(30,20)
84.       ,w5 FLOAT(30,20)
85.       ,w6 FLOAT(30,20)
86.       ,w7 FLOAT(30,20)
87.       ,w8 FLOAT(30,20)
88.       ,w9 FLOAT(30,20)
89.       ,w10 FLOAT(30,20)
90.       ,w11 FLOAT(30,20)
91.       ,w12 FLOAT(30,20)
92.       ,w13 FLOAT(30,20)
93.       ,w14 FLOAT(30,20)
94.       ,w15 FLOAT(30,20)
```

```
95.       ,w16 FLOAT(30,20)
96.       ,w17 FLOAT(30,20)
97.       ,w18 FLOAT(30,20)
98.       ,w19 FLOAT(30,20)
99.       ,w20 FLOAT(30,20)
100.      ,w21 FLOAT(30,20)
101.      ,w22 FLOAT(30,20)
102.      ,w23 FLOAT(30,20)
103.  );
```

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.
16.            DECLARE
17.                prob_notsatisfied,                        prob_gender_ns,
   prob_customer_type_ns,      prob_age_ns,      prob_type_of_travel_ns,
   prob_customer_class_ns
18.                ,                                 prob_flight_distance_ns,
   prob_inflight_wifi_service_ns,
   prob_departure_arrival_time_convenient_ns,
   prob_ease_of_online_booking_ns
19.                ,    prob_gate_location_ns,    prob_food_and_drink_ns,
   prob_online_boarding_ns,                          prob_seat_comfort_ns,
   prob_inflight_entertainment_ns
20.                ,  prob_onboard_service_ns,  prob_leg_room_service_ns,
   prob_baggage_handling_ns,                       prob_checkin_service_ns,
   prob_inflight_service_ns
21.                ,                                    prob_cleanliness_ns,
   prob_departure_delay_in_minutes_ns, prob_arrival_delay_in_minutes_ns
22.
23.            FLOAT(30,30) DEFAULT 0;
24.
25.            DECLARE prediksi_s, prediksi_ns FLOAT(30,30);
```

```
26.           DECLARE i, testing_ke, total_training, total_testing,
   i_testing, total_data INT DEFAULT 0;
27.           DECLARE info_satisfaction VARCHAR(2);
28.
29.           SELECT COUNT(*) INTO total_data FROM tbldataprocess;
30.
31.           SET testing_ke = number_of_testing;
32.
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_training = 0.9 * total_data;
40.               SET total_testing = total_data-total_training;
41.               SET i_testing = 1;
42.               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;
43.               UPDATE                tblaccuracy                SET
   total_data_training=total_training, total_data = total_testing WHERE
   testing = testing_ke AND algoritma = "Bayesian";
44.
45.               TRUNCATE tbldatatesting;
46.               TRUNCATE tbldatatraining;
47.
48.               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)
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 id<= total_training;
51.
52.               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)
53.             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
54.             FROM tbldataprocess WHERE
55.                 id > total_training
56.                 AND id <= ( total_training + total_testing);
57.
58.             SET total_satisfied = (SELECT count(satisfaction) FROM
tbldatatraining WHERE satisfaction = 1);
59.             SET  total_notsatisfied  =  (SELECT  count(satisfaction)
FROM tbldatatraining WHERE satisfaction = 0);
60.
61.             SET prob_satisfied =  total_satisfied / total_training;
62.             SET   prob_notsatisfied   =      total_notsatisfied   /
total_training;
63.
64.             -- WHILE per baris
65.             WHILE i_testing <= total_testing DO
66.             -- GENDER
67.             SET  prob_gender_s    =  (SELECT  count(gender)  FROM
tbldatatraining WHERE gender=(SELECT gender FROM tbldatatesting where
id=i_testing) AND satisfaction =1) / total_satisfied;
68.             SET  prob_gender_ns  =  (SELECT  count(gender)  FROM
tbldatatraining WHERE gender=(SELECT gender FROM tbldatatesting where
id=i_testing) AND satisfaction =0) / total_notsatisfied;
69.
70.             -- Customer Type
71.             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;
72.             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;
73.
74.             -- AGE
75.             SET   prob_age_s      =   (SELECT   count(age)   FROM
tbldatatraining  WHERE  age=(SELECT  age  FROM  tbldatatesting  where
id=i_testing) AND satisfaction =1) / total_satisfied;
76.             SET   prob_age_ns    =    (SELECT    count(age)    FROM
tbldatatraining  WHERE  age=(SELECT  age  FROM  tbldatatesting  where
id=i_testing) AND satisfaction =0) / total_notsatisfied;
77.
78.             -- Type Of Travel
79.             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;
```

```
80.               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;
81.
82.               -- Customer Class
83.               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;
84.               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;
85.
86.               -- FLIGHT DISTANCE
87.               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;
88.               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;
89.
90.               -- Inflight Wifi Service
91.               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;
92.               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;
93.
94.               -- Departure Arrival Time Convenient
95.               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;
96.               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;
97.
98.               -- Ease of Online Booking
99.               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;
```

```
100.              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;
101.
102.              -- Gate Location
103.              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;
104.              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;
105.
106.              -- Food and Drink
107.              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;
108.              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;
109.
110.              -- Online Boarding
111.              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;
112.              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;
113.
114.              -- Seat Comfort
115.              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;
116.              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;
117.
118.              -- Inflight Enterteinment
119.              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;
120.              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;
121.
122.              -- Onboard Service
123.              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;
124.              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;
125.
126.              -- Leg Room Service
127.              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;
128.              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;
129.
130.              -- Baggage Handling
131.              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;
132.              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;
133.
134.              -- Checkin Service
135.              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;
136.              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;
137.
138.              -- Inflight Service
139.              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;
140.              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;
141.
142.              -- Cleanliness
143.              SET prob_cleanliness_s   = (SELECT count(cleanliness)
   FROM  tbldatatraining  WHERE  cleanliness=(SELECT  cleanliness  FROM
```

```
   tbldatatesting   where   id=i_testing)   AND   satisfaction   =1)   /
   total_satisfied;
144.              SET prob_cleanliness_ns = (SELECT count(cleanliness)
   FROM  tbldatatraining  WHERE  cleanliness=(SELECT  cleanliness  FROM
   tbldatatesting   where   id=i_testing)   AND   satisfaction   =0)   /
   total_notsatisfied;
145.
146.              -- Departure Delay In Minutes
147.              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;
148.              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;
149.
150.              -- Arrival Delay In Minutes
151.              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;
152.              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;
153.
154.              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;
155.              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;
156.
```

```
157.              SELECT   satisfaction   INTO   info_satisfaction   from
   tbldatatesting where id = i_testing;
158.
159.
160.              IF info_satisfaction = 0 THEN -- actual not satisfied
161.                  IF prediksi_s < prediksi_ns THEN
162.                      UPDATE    tblaccuracy    SET    tn=tn+1    WHERE
   algoritma="Bayesian" AND testing=testing_ke;
163.                  ELSEIF prediksi_s > prediksi_ns THEN
164.                      UPDATE    tblaccuracy    SET    fp=fp+1    WHERE
   algoritma="Bayesian" AND testing=testing_ke;
165.                  ELSEIF prediksi_s = 0 AND prediksi_ns = 0 THEN
166.                      UPDATE  tblaccuracy  SET  fnull=fnull+1  WHERE
   algoritma="Bayesian" AND testing=testing_ke;
167.                  END IF;
168.              ELSEIF info_satisfaction = 1 THEN -- actual satisfied
169.                  IF prediksi_s < prediksi_ns THEN
170.                      UPDATE    tblaccuracy    SET    fn=fn+1    WHERE
   algoritma="Bayesian" AND testing=testing_ke;
171.                  ELSEIF prediksi_s > prediksi_ns THEN
172.                      UPDATE    tblaccuracy    SET    tp=tp+1    WHERE
   algoritma="Bayesian" AND testing=testing_ke;
173.                  ELSEIF prediksi_s = 0 AND prediksi_ns = 0 THEN
174.                      UPDATE  tblaccuracy  SET  tnull=tnull+1  WHERE
   algoritma="Bayesian" AND testing=testing_ke;
175.                  END IF;
176.              END IF;
177.              SET i_testing = i_testing+1;
178.              END WHILE;
179.
180.              SELECT testing_ke as 'selesai uji ke';
181.
182.      END ##
183.      DELIMITER ;
184.
185.  DROP PROCEDURE IF EXISTS processb;
186.      DELIMITER ##
187.      CREATE PROCEDURE processb()
188.      BEGIN
189.          CALL bayesian(1);
190.          CALL bayesian(2);
191.          CALL bayesian(3);
192.          CALL bayesian(4);
193.          CALL bayesian(5);
194.          UPDATE                    tblaccuracy                    SET
   accuracy=((tp+tn)/(tp+tn+fp+fn+tnull+fnull))*100                where
   algoritma='Bayesian';
195.
196.      END ##
197.      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)
```

```
                        + POWER((w8t - w8),2)
                        + POWER((w9t - w9),2)
                        + POWER((w10t - w10),2)
                        + POWER((w11t - w11),2)
                        + POWER((w12t - w12),2)
                        + POWER((w13t - w13),2)
                        + POWER((w14t - w14),2)
                        + POWER((w15t - w15),2)
                        + POWER((w16t - w16),2)
                        + POWER((w17t - w17),2)
                        + POWER((w18t - w18),2)
                        + POWER((w19t - w19),2)
                        + POWER((w20t - w20),2)
                        + POWER((w21t - w21),2)
                        + POWER((w22t - w22),2))
                    )
                    ;
            RETURN(hasil);
        END; ##
        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,w16
   s,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,w16
   t,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.
14.            DECLARE tn_lvq, fp_lvq, fn_lvq, tp_lvq INT DEFAULT 0;
15.
16.            DECLARE ws, wns, wt FLOAT(30,20) DEFAULT 0;
17.            DECLARE alpha, eps, err, temp_alpha FLOAT(30,20) DEFAULT 0;
18.            DECLARE info_satisfaction VARCHAR(2);
19.
20.            DECLARE prediction INT DEFAULT 0;
21.            DECLARE  i,  testing_ke,  total_training,  total_testing,
   i_testing, i_training, temp_i_training,
22.                total_data, ptotal_data_training INT DEFAULT 0;
23.
```

```
24.           SELECT COUNT(*) INTO total_data FROM tbldataprocess;
25.           SET testing_ke = number_of_testing;
26.           SET maxepoch=5;
27.
28.               IF testing_ke = 1 THEN SET total_training = 0.9 *
   total_data;
29.                   ELSEIF testing_ke = 2 THEN SET total_training =
   0.75 * total_data;
30.                   ELSEIF testing_ke = 3 THEN SET total_training = 0.5
   * total_data;
31.                   ELSEIF testing_ke = 4 THEN SET total_training =
   0.25 * total_data;
32.                   ELSEIF testing_ke = 5 THEN SET total_training = 0.1
   * total_data;
33.                   END IF;
34.               TRUNCATE tbldatatesting;
35.               TRUNCATE tbldatatraining;
36.
37.               SET total_testing = total_data-total_training;
38.               SET i_testing = 1;
39.               SET i_training = 1;
40.               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)
41.               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
42.               FROM tbldataprocess where id<= total_training;
43.
44.               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)
45.               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
46.               FROM tbldataprocess WHERE
47.                   id > total_training
48.                   AND id <= ( total_training + total_testing);
```

```
49.
50.             INSERT    INTO    tblaccuracy(algoritma,    testing,
   total_data_training, total_data,total_training, tp, tn, fp,fn,tnull,
   fnull, accuracy) VALUES
51.             ('LVQ', testing_ke,0,0,0,0,0,0,0,0,0,0);
52.
53.             -- INITIALITATION
54.             IF pAlpha=0 THEN
55.                 SET alpha = 0.9;
56.             ELSE
57.                 SET alpha = pAlpha;
58.             END IF;
59.
60.             IF pEps=0 THEN
61.                 SET eps = 0.0000001;
62.             ELSE
63.                 SET eps = pEps;
64.             END IF;
65.
66.             -- get id from each class
67.             SELECT  id  INTO  ids  FROM  tbldatatraining  WHERE
   satisfaction = 1 ORDER BY RAND() LIMIT 1;
68.             SELECT  id  INTO  idns  FROM  tbldatatraining  WHERE
   satisfaction = 0 ORDER BY RAND() LIMIT 1;
69.
70.             SELECT  gender  INTO  w1s  FROM  tbldatatraining  WHERE
   id=ids;
71.             SELECT  gender  INTO  w1ns  FROM  tbldatatraining  WHERE
   id=idns;
72.             SELECT  customer_type  INTO  w2s  FROM  tbldatatraining
   WHERE id=ids;
73.             SELECT  customer_type  INTO  w2ns  FROM  tbldatatraining
   WHERE id=idns;
74.             SELECT age INTO w3s FROM tbldatatraining WHERE id=ids;
75.             SELECT  age  INTO  w3ns  FROM  tbldatatraining  WHERE
   id=idns;
76.             SELECT  type_of_travel  INTO  w4s  FROM  tbldatatraining
   WHERE id=ids;
77.             SELECT  type_of_travel  INTO  w4ns  FROM  tbldatatraining
   WHERE id=idns;
78.             SELECT  customer_class  INTO  w5s  FROM  tbldatatraining
   WHERE id=ids;
79.             SELECT  customer_class  INTO  w5ns  FROM  tbldatatraining
   WHERE id=idns;
80.             SELECT  flight_distance  INTO  w6s  FROM  tbldatatraining
   WHERE id=ids;
81.             SELECT flight_distance INTO w6ns FROM tbldatatraining
   WHERE id=idns;
82.             SELECT    inflight_wifi_service    INTO    w7s    FROM
   tbldatatraining WHERE id=ids;
83.             SELECT    inflight_wifi_service    INTO    w7ns    FROM
   tbldatatraining WHERE id=idns;
84.             SELECT departure_arrival_time_convenient INTO w8s FROM
   tbldatatraining WHERE id=ids;
85.             SELECT departure_arrival_time_convenient INTO w8ns FROM
   tbldatatraining WHERE id=idns;
```

```
86.               SELECT    ease_of_online_booking    INTO    w9s    FROM
   tbldatatraining WHERE id=ids;
87.               SELECT    ease_of_online_booking    INTO    w9ns    FROM
   tbldatatraining WHERE id=idns;
88.               SELECT  gate_location  INTO  w10s  FROM  tbldatatraining
   WHERE id=ids;
89.               SELECT  gate_location  INTO  w10ns  FROM  tbldatatraining
   WHERE id=idns;
90.               SELECT  food_and_drink  INTO  w11s  FROM  tbldatatraining
   WHERE id=ids;
91.               SELECT  food_and_drink  INTO  w11ns  FROM  tbldatatraining
   WHERE id=idns;
92.               SELECT  online_boarding  INTO  w12s  FROM  tbldatatraining
   WHERE id=ids;
93.               SELECT  online_boarding  INTO  w12ns  FROM  tbldatatraining
   WHERE id=idns;
94.               SELECT  seat_comfort  INTO  w13s  FROM  tbldatatraining
   WHERE id=ids;
95.               SELECT  seat_comfort  INTO  w13ns  FROM  tbldatatraining
   WHERE id=idns;
96.               SELECT    inflight_entertainment    INTO    w14s    FROM
   tbldatatraining WHERE id=ids;
97.               SELECT    inflight_entertainment    INTO    w14ns    FROM
   tbldatatraining WHERE id=idns;
98.               SELECT  onboard_service  INTO  w15s  FROM  tbldatatraining
   WHERE id=ids;
99.               SELECT  onboard_service  INTO  w15ns  FROM  tbldatatraining
   WHERE id=idns;
100.              SELECT  leg_room_service  INTO  w16s  FROM  tbldatatraining
   WHERE id=ids;
101.              SELECT  leg_room_service  INTO  w16ns  FROM  tbldatatraining
   WHERE id=idns;
102.              SELECT  baggage_handling  INTO  w17s  FROM  tbldatatraining
   WHERE id=ids;
103.              SELECT  baggage_handling  INTO  w17ns  FROM  tbldatatraining
   WHERE id=idns;
104.              SELECT  checkin_service  INTO  w18s  FROM  tbldatatraining
   WHERE id=ids;
105.              SELECT  checkin_service  INTO  w18ns  FROM  tbldatatraining
   WHERE id=idns;
106.              SELECT  inflight_service  INTO  w19s  FROM  tbldatatraining
   WHERE id=ids;
107.              SELECT  inflight_service  INTO  w19ns  FROM  tbldatatraining
   WHERE id=idns;
108.              SELECT  cleanliness  INTO  w20s  FROM  tbldatatraining  WHERE
   id=ids;
109.              SELECT  cleanliness  INTO  w20ns  FROM  tbldatatraining
   WHERE id=idns;
110.              SELECT    departure_delay_in_minutes    INTO    w21s    FROM
   tbldatatraining WHERE id=ids;
111.              SELECT    departure_delay_in_minutes    INTO    w21ns    FROM
   tbldatatraining WHERE id=idns;
112.              SELECT    arrival_delay_in_minutes    INTO    w22s    FROM
   tbldatatraining WHERE id=ids;
113.              SELECT    arrival_delay_in_minutes    INTO    w22ns    FROM
   tbldatatraining WHERE id=idns;
```

```
114.
115.              -- row used to initialitation is not use again
116.              DELETE FROM tbldatatraining WHERE id=ids;
117.              DELETE FROM tbldatatraining WHERE id=idns;
118.
119.              SET  @num := 0;
120.              UPDATE tbldatatraining SET id = @num := (@num+1);
121.              ALTER TABLE tbldatatraining AUTO_INCREMENT =1;
122.              -- END INITIALITATION
123.
124.              -- TRAINING
125.              SET temp_alpha = alpha;
126.              WHILE epoch < maxepoch DO
127.                  SET i_training = 0;
128.                  SET temp_i_training = 0;
129.                  SET alpha=temp_alpha;
130.                  algolvq: WHILE (i_training <= total_training) or
   (alpha >= eps) DO
131.                      IF (alpha >= eps) THEN
132.                          SELECT gender INTO w1t FROM tbldatatraining
   WHERE id=i_training;
133.                              SELECT  customer_type  INTO  w2t  FROM
   tbldatatraining WHERE id=i_training;
134.                              SELECT    age    INTO    w3t    FROM
   tbldatatraining WHERE id=i_training;
135.                              SELECT  type_of_travel  INTO  w4t  FROM
   tbldatatraining WHERE id=i_training;
136.                              SELECT  customer_class  INTO  w5t  FROM
   tbldatatraining WHERE id=i_training;
137.                              SELECT  flight_distance  INTO  w6t  FROM
   tbldatatraining WHERE id=i_training;
138.                              SELECT inflight_wifi_service INTO w7t
   FROM tbldatatraining WHERE id=i_training;
139.                              SELECT
   departure_arrival_time_convenient INTO w8t FROM tbldatatraining WHERE
   id=i_training;
140.                              SELECT ease_of_online_booking INTO w9t
   FROM tbldatatraining WHERE id=i_training;
141.                              SELECT  gate_location  INTO  w10t  FROM
   tbldatatraining WHERE id=i_training;
142.                              SELECT  food_and_drink  INTO  w11t  FROM
   tbldatatraining WHERE id=i_training;
143.                              SELECT online_boarding INTO w12t FROM
   tbldatatraining WHERE id=i_training;
144.                              SELECT  seat_comfort  INTO  w13t  FROM
   tbldatatraining WHERE id=i_training;
145.                              SELECT   inflight_entertainment   INTO
   w14t FROM tbldatatraining WHERE id=i_training;
146.                              SELECT onboard_service INTO w15t FROM
   tbldatatraining WHERE id=i_training;
147.                              SELECT leg_room_service INTO w16t FROM
   tbldatatraining WHERE id=i_training;
148.                              SELECT baggage_handling INTO w17t FROM
   tbldatatraining WHERE id=i_training;
149.                              SELECT checkin_service INTO w18t FROM
   tbldatatraining WHERE id=i_training;
```

```
150.                                SELECT inflight_service INTO w19t FROM
   tbldatatraining WHERE id=i_training;
151.                                SELECT  cleanliness  INTO  w20t  FROM
   tbldatatraining WHERE id=i_training;
152.                                SELECT departure_delay_in_minutes INTO
   w21t FROM tbldatatraining WHERE id=i_training;
153.                                SELECT  arrival_delay_in_minutes  INTO
   w22t FROM tbldatatraining WHERE id=i_training;
154.                                SELECT  satisfaction  INTO  w23t  FROM
   tbldatatraining WHERE id=i_training;
155.                            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);
156.                            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);
157.
158.                            IF ws < wns THEN SET cj = 1;
159.                                ELSEIF ws > wns THEN SET cj = 0;
160.                                ELSE SET cj = 1;
161.                                END IF;
162.                            SELECT    satisfaction    INTO    t    FROM
   tbldatatraining WHERE id=i_training;
163.                            -- cj 1 satisfied, 0 dissatisfied
164.                            IF cj = 1 AND t = 1 THEN
165.                                SET w1s = w1s + (alpha * (w1t - w1s));
166.                                SET w2s = w2s + (alpha * (w2t - w2s));
167.                                SET w3s = w3s + (alpha * (w3t - w3s));
168.                                SET w4s = w4s + (alpha * (w4t - w4s));
169.                                SET w5s = w5s + (alpha * (w5t - w5s));
170.                                SET w6s = w6s + (alpha * (w6t - w6s));
171.                                SET w7s = w7s + (alpha * (w7t - w7s));
172.                                SET w8s = w8s + (alpha * (w8t - w8s));
173.                                SET w9s = w9s + (alpha * (w9t - w9s));
174.                                SET  w10s  =  w10s  +  (alpha  *  (w10t  -
   w10s));
175.                                SET  w11s  =  w11s  +  (alpha  *  (w11t  -
   w11s));
176.                                SET  w12s  =  w12s  +  (alpha  *  (w12t  -
   w12s));
177.                                SET  w13s  =  w13s  +  (alpha  *  (w13t  -
   w13s));
178.                                SET  w14s  =  w14s  +  (alpha  *  (w14t  -
   w14s));
179.                                SET  w15s  =  w15s  +  (alpha  *  (w15t  -
   w15s));
180.                                SET  w16s  =  w16s  +  (alpha  *  (w16t  -
   w16s));
181.                                SET  w17s  =  w17s  +  (alpha  *  (w17t  -
   w17s));
182.                                SET  w18s  =  w18s  +  (alpha  *  (w18t  -
   w18s));
```

```
183.                        SET w19s = w19s + (alpha * (w19t -
   w19s));
184.                        SET w20s = w20s + (alpha * (w20t -
   w20s));
185.                        SET w21s = w21s + (alpha * (w21t -
   w21s));
186.                        SET w22s = w22s + (alpha * (w22t -
   w22s));
187.
188.                    ELSEIF cj = 0 AND t = 0 THEN
189.                        SET w1ns = w1ns + (alpha * (w1t -
   w1ns));
190.                        SET w2ns = w2ns + (alpha * (w2t -
   w2ns));
191.                        SET w3ns = w3ns + (alpha * (w3t -
   w3ns));
192.                        SET w4ns = w4ns + (alpha * (w4t -
   w4ns));
193.                        SET w5ns = w5ns + (alpha * (w5t -
   w5ns));
194.                        SET w6ns = w6ns + (alpha * (w6t -
   w6ns));
195.                        SET w7ns = w7ns + (alpha * (w7t -
   w7ns));
196.                        SET w8ns = w8ns + (alpha * (w8t -
   w8ns));
197.                        SET w9ns = w9ns + (alpha * (w9t -
   w9ns));
198.                        SET w10ns = w10ns + (alpha * (w10t -
   w10ns));
199.                        SET w11ns = w11ns + (alpha * (w11t -
   w11ns));
200.                        SET w12ns = w12ns + (alpha * (w12t -
   w12ns));
201.                        SET w13ns = w13ns + (alpha * (w13t -
   w13ns));
202.                        SET w14ns = w14ns + (alpha * (w14t -
   w14ns));
203.                        SET w15ns = w15ns + (alpha * (w15t -
   w15ns));
204.                        SET w16ns = w16ns + (alpha * (w16t -
   w16ns));
205.                        SET w17ns = w17ns + (alpha * (w17t -
   w17ns));
206.                        SET w18ns = w18ns + (alpha * (w18t -
   w18ns));
207.                        SET w19ns = w19ns + (alpha * (w19t -
   w19ns));
208.                        SET w20ns = w20ns + (alpha * (w20t -
   w20ns));
209.                        SET w21ns = w21ns + (alpha * (w21t -
   w21ns));
210.                        SET w22ns = w22ns + (alpha * (w22t -
   w22ns));
211.
212.                    ELSEIF cj = 0 AND t = 1 THEN
```

```
213.                            SET w1ns = w1ns - (alpha * (w1t -
   w1ns));
214.                            SET w2ns = w2ns - (alpha * (w2t -
   w2ns));
215.                            SET w3ns = w3ns - (alpha * (w3t -
   w3ns));
216.                            SET w4ns = w4ns - (alpha * (w4t -
   w4ns));
217.                            SET w5ns = w5ns - (alpha * (w5t -
   w5ns));
218.                            SET w6ns = w6ns - (alpha * (w6t -
   w6ns));
219.                            SET w7ns = w7ns - (alpha * (w7t -
   w7ns));
220.                            SET w8ns = w8ns - (alpha * (w8t -
   w8ns));
221.                            SET w9ns = w9ns - (alpha * (w9t -
   w9ns));
222.                            SET w10ns = w10ns - (alpha * (w10t -
   w10ns));
223.                            SET w11ns = w11ns - (alpha * (w11t -
   w11ns));
224.                            SET w12ns = w12ns - (alpha * (w12t -
   w12ns));
225.                            SET w13ns = w13ns - (alpha * (w13t -
   w13ns));
226.                            SET w14ns = w14ns - (alpha * (w14t -
   w14ns));
227.                            SET w15ns = w15ns - (alpha * (w15t -
   w15ns));
228.                            SET w16ns = w16ns - (alpha * (w16t -
   w16ns));
229.                            SET w17ns = w17ns - (alpha * (w17t -
   w17ns));
230.                            SET w18ns = w18ns - (alpha * (w18t -
   w18ns));
231.                            SET w19ns = w19ns - (alpha * (w19t -
   w19ns));
232.                            SET w20ns = w20ns - (alpha * (w20t -
   w20ns));
233.                            SET w21ns = w21ns - (alpha * (w21t -
   w21ns));
234.                            SET w22ns = w22ns - (alpha * (w22t -
   w22ns));
235.
236.                        ELSEIF cj = 1 AND t = 0 THEN
237.                            SET w1s = w1s - (alpha * (w1t - w1s));
238.                            SET w2s = w2s - (alpha * (w2t - w2s));
239.                            SET w3s = w3s - (alpha * (w3t - w3s));
240.                            SET w4s = w4s - (alpha * (w4t - w4s));
241.                            SET w5s = w5s - (alpha * (w5t - w5s));
242.                            SET w6s = w6s - (alpha * (w6t - w6s));
243.                            SET w7s = w7s - (alpha * (w7t - w7s));
244.                            SET w8s = w8s - (alpha * (w8t - w8s));
245.                            SET w9s = w9s - (alpha * (w9t - w9s));
```

```
246.                                SET w10s = w10s - (alpha * (w10t -
   w10s));
247.                                SET w11s = w11s - (alpha * (w11t -
   w11s));
248.                                SET w12s = w12s - (alpha * (w12t -
   w12s));
249.                                SET w13s = w13s - (alpha * (w13t -
   w13s));
250.                                SET w14s = w14s - (alpha * (w14t -
   w14s));
251.                                SET w15s = w15s - (alpha * (w15t -
   w15s));
252.                                SET w16s = w16s - (alpha * (w16t -
   w16s));
253.                                SET w17s = w17s - (alpha * (w17t -
   w17s));
254.                                SET w18s = w18s - (alpha * (w18t -
   w18s));
255.                                SET w19s = w19s - (alpha * (w19t -
   w19s));
256.                                SET w20s = w20s - (alpha * (w20t -
   w20s));
257.                                SET w21s = w21s - (alpha * (w21t -
   w21s));
258.                                SET w22s = w22s - (alpha * (w22t -
   w22s));
259.
260.                            END IF;
261.
262.                            SET alpha = alpha - (alpha * eps);
263.                            UPDATE tblaccuracy SET total_data_training
   = i_training+1 WHERE id=(SELECT COUNT(*) FROM tblaccuracy);
264.
265.                            SET i_training = i_training + 1;
266.                            SET temp_i_training = i_training;
267.                            IF (i_training = total_training) THEN
268.                                SET temp_alpha = alpha;
269.                                SELECT alpha as 'alphat', i_training as
   't';
270.                                SET alpha = eps;
271.                            ELSEIF (alpha <= eps) THEN
272.                                SET temp_alpha = alpha;
273.                                SELECT alpha as 'alphaa', i_training as
   't';
274.                                SET i_training = total_training +1;
275.                            END IF;
276.                        ELSE
277.                            LEAVE algolvq;
278.                        END IF;
279.                    END WHILE;
280.                    SET epoch = epoch + 1;
281.                    IF (temp_i_training <> 0) THEN
282.                        SET  ptotal_data_training=ptotal_data_training
   + temp_i_training;
283.                        SET pepoch=epoch;
284.                        SELECT epoch as "epoch";
```

```
285.                 END IF;
286.             END WHILE;
287.             UPDATE   tblaccuracy   SET   total_data_training   =
   ptotal_data_training WHERE id=(SELECT COUNT(*) FROM tblaccuracy);
288.             UPDATE tblaccuracy SET total_training = pepoch WHERE
   id=(SELECT COUNT(*) FROM tblaccuracy);
289.             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,w2
   0,w21,w22,w23) values
290.             (
291.                 testing_ke
292.                 ,pAlpha
293.                 , pEps
294.                 , w1s
295.                 , w2s
296.                 , w3s
297.                 , w4s
298.                 , w5s
299.                 , w6s
300.                 , w7s
301.                 , w8s
302.                 , w9s
303.                 , w10s
304.                 , w11s
305.                 , w12s
306.                 , w13s
307.                 , w14s
308.                 , w15s
309.                 , w16s
310.                 , w17s
311.                 , w18s
312.                 , w19s
313.                 , w20s
314.                 , w21s
315.                 , w22s
316.                 , w23s
317.             );
318.
319.             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,w2
   0,w21,w22,w23) values
320.             (
321.                 testing_ke
322.                 ,pAlpha
323.                 , pEps
324.                 , w1ns
325.                 , w2ns
326.                 , w3ns
327.                 , w4ns
328.                 , w5ns
329.                 , w6ns
330.                 , w7ns
331.                 , w8ns
332.                 , w9ns
333.                 , w10ns
334.                 , w11ns
```

```
                    , w12ns
                    , w13ns
                    , w14ns
                    , w15ns
                    , w16ns
                    , w17ns
                    , w18ns
                    , w19ns
                    , w20ns
                    , w21ns
                    , w22ns
                    , w23ns
                );
                -- END TRAINING

                -- SELECT i_training;
                -- SELECT alpha;
                WHILE i_testing <= total_testing DO
                    SELECT gender INTO w1t FROM tbldatatesting WHERE
   id=i_testing;
                        SELECT    customer_type    INTO    w2t    FROM
   tbldatatesting WHERE id=i_testing;
                        SELECT age INTO w3t FROM tbldatatesting WHERE
   id=i_testing;
                        SELECT    type_of_travel    INTO    w4t    FROM
   tbldatatesting WHERE id=i_testing;
                        SELECT    customer_class    INTO    w5t    FROM
   tbldatatesting WHERE id=i_testing;
                        SELECT    flight_distance    INTO    w6t    FROM
   tbldatatesting WHERE id=i_testing;
                        SELECT  inflight_wifi_service  INTO  w7t  FROM
   tbldatatesting WHERE id=i_testing;
                        SELECT  departure_arrival_time_convenient  INTO
   w8t FROM tbldatatesting WHERE id=i_testing;
                        SELECT  ease_of_online_booking  INTO  w9t  FROM
   tbldatatesting WHERE id=i_testing;
                        SELECT    gate_location    INTO    w10t    FROM
   tbldatatesting WHERE id=i_testing;
                        SELECT    food_and_drink    INTO    w11t    FROM
   tbldatatesting WHERE id=i_testing;
                        SELECT    online_boarding    INTO    w12t    FROM
   tbldatatesting WHERE id=i_testing;
                        SELECT    seat_comfort    INTO    w13t    FROM
   tbldatatesting WHERE id=i_testing;
                        SELECT  inflight_entertainment  INTO  w14t  FROM
   tbldatatesting WHERE id=i_testing;
                        SELECT    onboard_service    INTO    w15t    FROM
   tbldatatesting WHERE id=i_testing;
                        SELECT    leg_room_service    INTO    w16t    FROM
   tbldatatesting WHERE id=i_testing;
                        SELECT    baggage_handling    INTO    w17t    FROM
   tbldatatesting WHERE id=i_testing;
                        SELECT    checkin_service    INTO    w18t    FROM
   tbldatatesting WHERE id=i_testing;
                        SELECT    inflight_service    INTO    w19t    FROM
   tbldatatesting WHERE id=i_testing;
```

```
372.                     SELECT    cleanliness    INTO    w20t    FROM
   tbldatatesting WHERE id=i_testing;
373.                     SELECT  departure_delay_in_minutes  INTO  w21t
   FROM tbldatatesting WHERE id=i_testing;
374.                     SELECT arrival_delay_in_minutes INTO w22t FROM
   tbldatatesting WHERE id=i_testing;
375.                     SELECT    satisfaction    INTO    w23t    FROM
   tbldatatesting WHERE id=i_testing;
376.
377.                 SET ws = ed(
378.                         w1t, w1s
379.                         , w2t, w2s
380.                         , w3t, w3s
381.                         , w4t, w4s
382.                         , w5t, w5s
383.                         , w6t, w6s
384.                         , w7t, w7s
385.                         , w8t, w8s
386.                         , w9t, w9s
387.                         , w10t, w10s
388.                         , w11t, w11s
389.                         , w12t, w12s
390.                         , w13t, w13s
391.                         , w14t, w14s
392.                         , w15t, w15s
393.                         , w16t, w16s
394.                         , w17t, w17s
395.                         , w18t, w18s
396.                         , w19t, w19s
397.                         , w20t, w20s
398.                         , w21t, w21s
399.                         , w22t, w22s
400.                     );
401.
402.                 SET wns = ed(
403.                         w1t, w1ns
404.                         , w2t, w2ns
405.                         , w3t, w3ns
406.                         , w4t, w4ns
407.                         , w5t, w5ns
408.                         , w6t, w6ns
409.                         , w7t, w7ns
410.                         , w8t, w8ns
411.                         , w9t, w9ns
412.                         , w10t, w10ns
413.                         , w11t, w11ns
414.                         , w12t, w12ns
415.                         , w13t, w13ns
416.                         , w14t, w14ns
417.                         , w15t, w15ns
418.                         , w16t, w16ns
419.                         , w17t, w17ns
420.                         , w18t, w18ns
421.                         , w19t, w19ns
422.                         , w20t, w20ns
423.                         , w21t, w21ns
```

```
                          , w22t, w22ns
                      );


                  SELECT satisfaction INTO info_satisfaction FROM tbldatatesting WHERE id=i_testing;
                  IF ws < wns THEN SET prediction = 1;
                      ELSEIF ws > wns THEN SET prediction = 0;
                      ELSE SET prediction = 1;
                  END IF;
                  IF info_satisfaction = 0 AND prediction=0 THEN
                      SET tn_lvq=tn_lvq+1;
                  ELSEIF info_satisfaction = 0 AND prediction=1  THEN
                      SET fp_lvq=fp_lvq+1;
                  ELSEIF info_satisfaction = 1 AND prediction=0 THEN
                      SET fn_lvq=fn_lvq+1;
                  ELSEIF info_satisfaction = 1 AND prediction=1 THEN
                      SET tp_lvq=tp_lvq+1;
                  END IF;
              SET i_testing = i_testing + 1;
              END WHILE;

              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)
                  ;
              SELECT testing_ke as 'selesai uji ke', pAlpha as 'a', pEps as 'eps';
      END ##
      DELIMITER ;

  DROP PROCEDURE IF EXISTS processl;
      DELIMITER ##
      CREATE PROCEDURE processl(alpha FLOAT(30,20), eps FLOAT(30,20))
      BEGIN
          CALL lvq(1,alpha,eps);
          CALL lvq(2,alpha,eps);
          CALL lvq(3,alpha,eps);
          CALL lvq(4,alpha,eps);
          CALL lvq(5,alpha,eps);
          UPDATE       tblaccuracy       SET       accuracy       = ((tp+tn)/(tp+tn+fp+fn))*100 WHERE algoritma="LVQ";
          SELECT * FROM tblaccuracy WHERE id>((SELECT COUNT(*) FROM tblaccuracy)-5);
      END ##
      DELIMITER ;

  DROP PROCEDURE IF EXISTS processlvq;
      DELIMITER ##
      CREATE PROCEDURE processlvq()
      BEGIN
          CALL processl(0.9,0.0000001);
```

```
476.          CALL processl(0.9,0.0001);
477.          CALL processl(0.9,0.01);
478.
479.          CALL processl(0.1,0.0000001);
480.          CALL processl(0.1,0.0001);
481.          CALL processl(0.1,0.01);
482.
483.          CALL processl(0.01,0.0000001);
484.          CALL processl(0.01,0.0001);
485.
486.          CALL processl(0.05,0.0000001);
487.          CALL processl(0.05,0.0001);
488.
489.          END ##
490.      DELIMITER ;
```

Report #14310919

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