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:

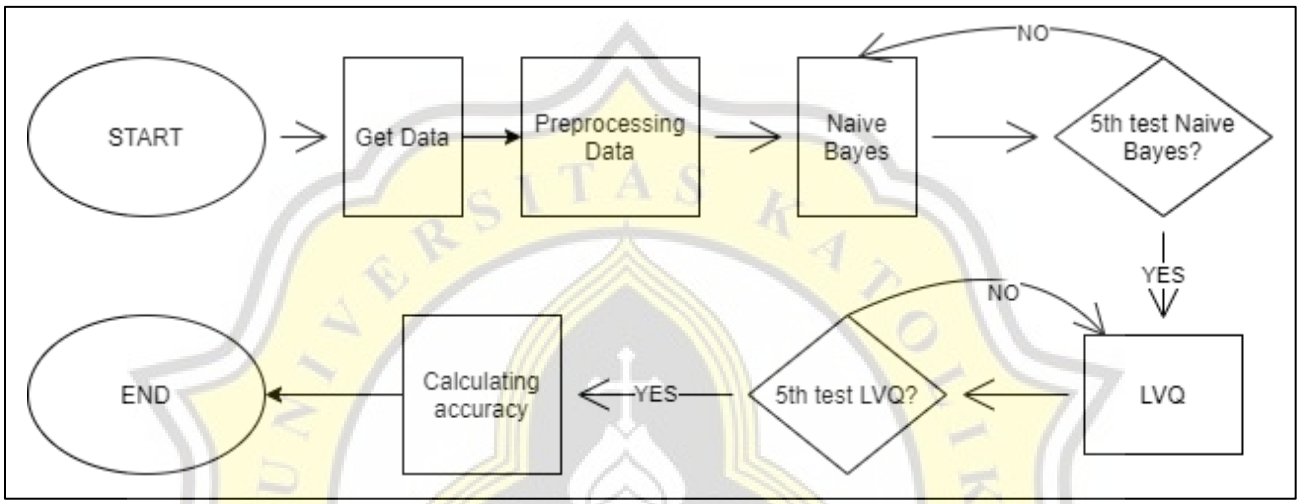

Figure 4.1 Workflow

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.

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

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.

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

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:

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

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

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.

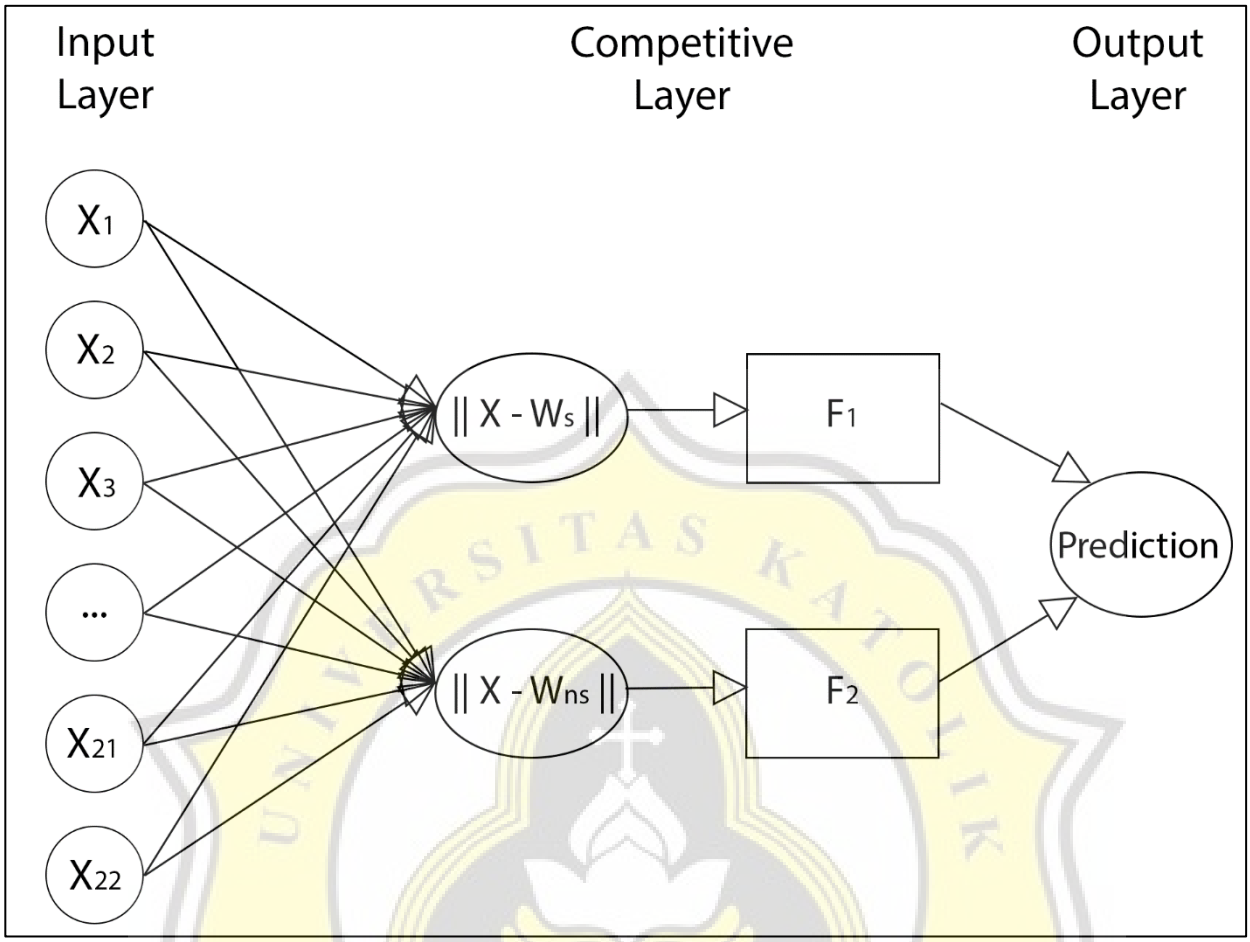

Figure 4.2 LVQ Architecture

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.

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

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

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

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

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

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

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.