

project report

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

NIKODEMUS GALIH CANDRA WICAKSONO

18.K1.0082

Faculty of Computer Science

Soegijapranata Catholic University

2020

APPROVAL AND RATIFICATION PAGE (Heading plain)

(gunakan style “Approval”)

JUDUL PROJECT ANDA

by

NIKODEMUS GALIH CANDRA WICAKSONO – 18.K1.0082

This project report has been approved and ratified

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

With approval,

|  |  |
| --- | --- |
|  | Supervisor  Dosen Pembimbing Project  NPP: ­ |
| Examiners,  1.)  [NAMA PENGUJI 1]  NPP :  2.)  [NAMA PENGUJI 2]  NPP :  3.)  [NAMA PENGUJI 3]  NPP : |  |
|  | Dean of Faculty of Computer Science  Robertus Setiawan AJi Nugroho, Ph.D  NPP: 058.1.2004.264 |

DECLARATION OF AUTHORSHIP

I, the undersigned:

Name : NIKODEMUS GALIH CANDRA WICAKSONO

ID : 18.K1.0082

declare that this work, titled "JUDUL ANDA", and the work presented in it is my own. I confirm that:

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

Semarang, MONTH, DD, YYYY

[sign here]

NIKODEMUS GALIH CANDRA WICAKSONO

18.K1.0082

ACKNOWLEDGMENT

Silahkan tuliskan anda ingin mengucapkan terima kasih atau ucapan persembahan ke siapapun yang anda rasa perlu ditulis disini

I have received a myriad of support, advice, and assistance throughout this document writing. I would like to thank my supervisors XXXXXX for formulating this topic. I would also like to thank my friend XXXXX for guiding with advice to finish this document.

I would like to thank my family and friends for giving me ceaseless love, support, and advices throughout my study in Soegijapranata Catholic University. You gave me great escape to rest my mind from my thesis.

ABSTRACT (Abstract Title)

Isi dari abstract menggunakan syle abstract content. Abstract ditulis dalam 3 paragraf. Semua ditulis dengan huruf italic dan 1 spasi Paragraf pertama berisi tentang permasalahan yang diselesaikan dalam project ini.

Paragraf kedua di sini, membahas tentang proses penyelesaian yang Anda tawarkan.

Sedangkan paragraf ketiga membahas tentang hasil akhir. Setelah itu di bagian paling bawah, sertakan keywords atau kata kunci 3-5 kata.

Keyword: kata\_kunci1, kata\_kunci2, kata\_kunci3, dst

\*Tambahkan informasi mengenai penelitian payung di sini apabila ada (konsultasikan dengan dosen pembimbing).

TABLE OF CONTENTS

COVER i

[APPROVAL AND RATIFICATION PAGE (Heading plain) ii](#_Toc85463061)

[DECLARATION OF AUTHORSHIP iii](#_Toc85463062)

[ACKNOWLEDGMENT iv](#_Toc85463063)

[TABLE OF CONTENTS vi](#_Toc85463064)

[LIST OF FIGURE viii](#_Toc85463065)

[LIST OF TABLE ix](#_Toc85463066)

[CHAPTER 1 INTRODUCTION 10](#_Toc85463067)

[1.1. Background 10](#_Toc85463068)

[1.2. Problem Formulation 11](#_Toc85463069)

[1.3. Scope 11](#_Toc85463070)

[1.4. Objective 11](#_Toc85463071)

[CHAPTER 2 LITERATURE STUDY 12](#_Toc85463072)

[CHAPTER 3 RESEARCH METHODOLOGY 15](#_Toc85463073)

[3.1. Data Collection 15](#_Toc85463074)

[3.2. Algorithm 15](#_Toc85463075)

[3.3. Coding and Design 15](#_Toc85463076)

[3.4. Analysis 16](#_Toc85463077)

[3.5. Make a Report 16](#_Toc85463078)

[CHAPTER 4 ANALYSIS AND DESIGN 17](#_Toc85463079)

[CHAPTER 5 IMPLEMENTATION AND RESULTS 19](#_Toc85463080)

[5.1. Implementation 19](#_Toc85463081)

[5.2. Results 19](#_Toc85463082)

[CHAPTER 6 CONCLUSION 20](#_Toc85463083)

[REFERENCES 21](#_Toc85463084)

[APPENDIX a](#_Toc85463085)

LIST OF FIGURE

[**Figure 4.1** Workflow 17](#_Toc85510339)

LIST OF TABLE

[**Table 3.1.** Analysis 16](#_Toc85466710)

[**Table 4.1.** Data Table 17](#_Toc85466711)

[**Table 4.2.** Modified Attribute Data Table 18](#_Toc85466712)

# CHAPTER 1 INTRODUCTION

## Background

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

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

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

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

## Problem Formulation

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

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

## Scope

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

## Objective

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

# LITERATURE STUDY

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

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

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

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

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

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

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

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

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

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

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

# RESEARCH METHODOLOGY

## Data Collection

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

## Algorithm

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

## Coding and Design

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

## Analysis

In analyzing, I will do 5 tests as follows :

Analysis

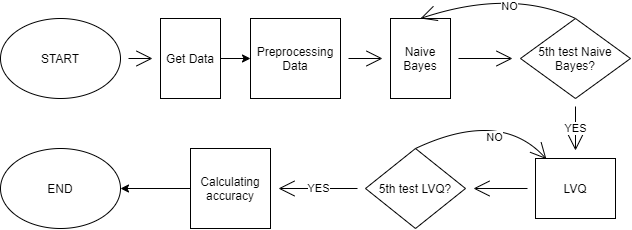
|  |  |  |
| --- | --- | --- |
|  | Training Data | Testing Data |
| I | 10% | 90% |
| II | 25% | 75% |
| III | 50% | 50% |
| IV | 75% | 25% |
| V | 90% | 10% |

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

## Make a Report

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

# ANALYSIS AND DESIGN

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

Workflow

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

Data Table

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

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

From the existing data as seen 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 there is a record of the attributes to be changed. Notes will be converted to numbers at small intervals. For example, there are too many age categories, which will then be changed to "0" where the age is <28, then "1" where the age is between 28 and 52, and finally "2" which is over 52. The value of 28 and 52 is based on quantile values ​​that can be seen on the data link is downloaded. There is also data that is not in the form of numbers will be converted to numbers. This is because the LVQ algorithm will be calculated based on the value of the attribute. So that it is converted into a number so that it can be calculated. For example, the original gender "Female" and "Male" will be changed to "0" and "1". The attributes that are changed in the preprocessing stage are as follows:

Modified Attribute Data Table

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

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

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

Distribution of Training and Testing Data

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

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

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

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

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

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

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

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

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

x = attribute class/label

y = attribute input

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

Data Sample Naïve Bayes (20 data)

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

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

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

Training Dataset Naïve Bayes

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

Testing Dataset Naïve Bayes

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

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

x = Total data class/label from training data

Total data = Total data from training data

P(x) = Probability of variable class/label

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

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

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

a = class input from testing data

x = class/label

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

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

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

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

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

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

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

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

1. The biggest results are prediction results

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

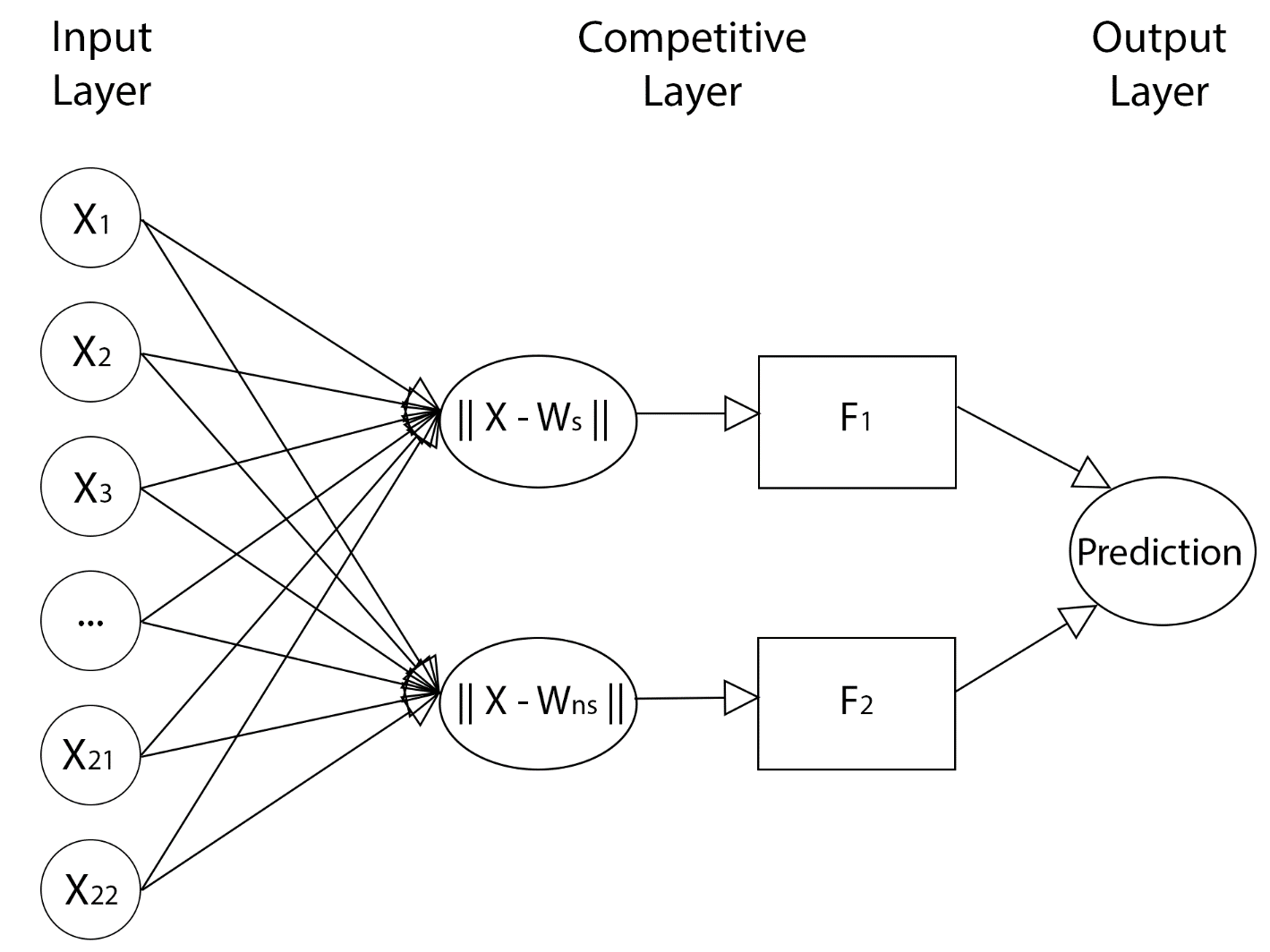
1. To find the accuracy results later then if:
   1. Label class “1” and prediction results is “1”, TP added 1
   2. Label class “0” and prediction result is “0”, TN added 1
   3. Label class “1” and prediction results is “0”, FN added 1
   4. Label class “0” and prediction results is “1”, FP added 1

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

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

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

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

LVQ Architecture

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

||X-W|| = Euclidean distance

Xn = Value from attribute n

Wcn = Weight of class/label c and attribute n

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

Data Sample LVQ (20 data)

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

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

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

Training Dataset LVQ

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

Testing Dataset LVQ

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

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

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

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

Initial Weight

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

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

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

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

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

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

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

1. If Epoch < MaxEpoch or α > eps:

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

* 1. Epoch = epoch +1.

Epoch changed to 2.

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

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

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

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

T=Target

Wj = Weight class j

α = Learning ratio

j = prediction class

X = data value

Wn = Weigh index n

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

* 1. Update the value of α.

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

α` = new learning ratio

α = learning ratio

MaxEpoch = Maximum Iteration

α` = new learning ratio

Then the value of the new alpha is 0.9 - (0.9 \* (1/16)) which is 0.84375. This new alpha value will be used as the alpha value for the next iteration.

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

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

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

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

1. Repeat step 6 for all testing datasets

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

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

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

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

# IMPLEMENTATION AND RESULTS

## Implementation

Bab implementasi adalah bab tentang narasi pemanfaatan dari data structure dan algoritma dalam bentuk aplikasi terapan.

Bab implementasi menyertakan source code, namun tidak semua source code program disertakan dalam bab ini. Ambil lah penggalan nya saja yang penting dan menjadi inti dari program Anda. Gunakan style “Code” . Jika menyertakan gambar (capture), silahkan ditambahkan caption di gambar tersebut sebagaimana penjelasan pada Chapter 5.

1. create Function sfHelloWorld (vNama varchar(30))
2. returns varchar(100)
3. begin
4. declare vHello varchar(255)
5. select concat(‘Hello ‘, vNama) into vHello;
6. return vHello;
7. End

Setiap source code diberikan nomor urut baris. Jelaskan baris perintah dan untuk apa perintah tersebut. Baris 1-2 kode program berisi perintah untuk membuat function dengan nama sfHelloWorld. Baris 3 dan 7 adalah blok baris untuk function khusus di dalamnya. Inti dari kode program ada pada baris 4 untuk deklarasi variabel vHello, baris 5 untuk menggabungkan karakter “Hello” dengan variabel vNama dan mengembalikan hasilnya pada baris ke 7.

## Results

Sub bab results berisi hasil dari uji coba algoritma dan struktur data yang diterapkan dalam bentuk aplikasi. Hasil disajikan dalam bentuk tabel, narasi atau gambar yang dapat memberikan penjelasan solusi masalah dengan bantuan program sehingga dapat ditarik kesimpulan dari penelitian anda.

# CONCLUSION

Bab ini membahas tentang kesimpulan akhir. Harus menjawab semua pertanyaan yang Anda ajukan sebagai permasalahan yang bab 1 bagian scope. Tidak sekedar menyimpulkan tapi sertakan argumentasi kuat terkait pengambilan kesimpulan tersebut.

Di bagian akhir, sertakan saran untuk penelitian lanjutan. Tidak perlu bertele-tele tapi fokuskan pada saran penelitian Anda saja, apa yang belum dilakukan disertakan di sini.

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

Jika Anda punya lampiran dari project, silahkan dilampirkan di bagian ini. Yang wajib Anda lampirkan adalah kode program (coding) lengkap dan diberikan keterangan terlebih dahulu pada bagian atas dari coding tersebut, koding ditulis dengan format font yang berbeda. Contoh:

**CODING PEHITUNGAN**

1. SELECT @a := 5;
2. SELECT @b := 5;
3. SELECT hasil:= @a \* @b;

**PROCEDURE HITUNG PERKALIAN**

1. CREATE PROCEDURE spMaksimal ()
2. BEGIN
3. DECLARE a INT;
4. DECLARE b INT;
5. DECLARE hasil INT;
6. SELECT a \* b INTO hasil;
7. SELECT hasil
8. END

Selain coding, yang dapat dijadikan lampiran adalah: hasil hitungan yang panjang dan tidak mungkin dimuat dalam laporan utama, gambar atau ilustrasi diagram yang cukup panjang namun hanya sebagai penjelasan dari diagram utama yang ada di dalam laporan, foto-foto penunjang, dan dokumen lain yang sifatnya menunjang namun dianggap penting.