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);
```

```
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;
UPDATE tbldataprocess set customer_class=iwhile where customer_class=spinformation;
set iwhile= iwhile +1;
END WHILE ;
-- FLIGHT DISTANCE
UPDATE tbldataprocess set flight_distance=0 where flight_distance <= 414;
UPDATE tbldataprocess set flight_distance=1 where flight_distance > 414 && flight_distance <= 1744;
UPDATE tbldataprocess set flight_distance=2 where flight_distance > 1744;
-- Departure Delay In Minutes
UPDATE tbldataprocess set departure_delay_in_minutes=0 where departure_delay_in_minutes <= 12;
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,0,0,0,0,0,0,0,0,0,0);

-- 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

-- TRAINING
SET temp_alpha = alpha;
WHILE epoch < maxepoch DO
SET i_training = 0;
SET temp_i_training = 0;
SET alpha=temp_alpha;
 algolvq: WHILE (i_training <= total_training) or (alpha >= eps) DO
  IF(alpha>=eps)THEN
  SELECT gender INTO w1t FROM tbldatatraining WHERE id=i_training;
  ……
  SELECT satisfaction INTO w23t FROM tbldatatraining WHERE id=i_training;

  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);
  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);

  IF ws < wns THEN
    SET cj = 1;
  ELSEIF ws > wns THEN
    SET cj = 0;
  ELSE
    SET cj = 1;
  END IF;

  SELECT satisfaction INTO t FROM tbldatatraining WHERE id=i_training;

  IF cj = 1 AND t = 1 THEN
    SET w1s = w1s + (alpha * (w1t - w1s));
    ……
    SET w22s = w22s + (alpha * (w22t - w22s));
  ELSEIF cj = 0 AND t = 0 THEN
    SET w1ns = w1ns + (alpha * (w1t - w1ns));
    ……
    SET w22ns = w22ns + (alpha * (w22t - w22ns));
  ELSEIF cj = 0 AND t = 1 THEN
    SET w1ns = w1ns - (alpha * (w1t - w1ns));
    ……
    SET w22ns = w22ns - (alpha * (w22t - w22ns));
  ELSEIF cj = 1 AND t = 0 THEN
    SET w1s = w1s - (alpha * (w1t - w1s));
    ……
    SET w22s = w22s - (alpha * (w22t - w22s));
  END IF;

  SET alpha = alpha - (alpha * eps);
  UPDATE tblaccuracy SET total_data_training = i_training+1 WHERE id=(SELECT COUNT(*) FROM tblaccuracy);
  SET i_training = i_training + 1;
  SET temp_i_training = i_training;
  IF (i_training = total_training) THEN
```

```
    SET temp_alpha = alpha;
    SET alpha = eps;
  ELSEIF (alpha <= eps) THEN
    SET temp_alpha = alpha;
    SELECT alpha as 'alphaa', i_training as 't';
    SET i_training = total_training +1;
  END IF;
  ELSE
    LEAVE algolvq;
  END IF;
  END WHILE;
    SET epoch = epoch + 1;
  IF (temp_i_training <> 0) THEN
    SET ptotal_data_training=ptotal_data_training + temp_i_training;
    SET pepoch=epoch;
  END IF;
  END WHILE;
WHILE i_testing <= total_testing DO
  SELECT gender INTO w1t FROM tbldatatesting WHERE id=i_testing;
  ……
  SELECT arrival_delay_in_minutes INTO w22t FROM tbldatatesting WHERE id=i_testing;
  SELECT satisfaction INTO w23t FROM tbldatatesting WHERE id=i_testing;

  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);

  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);

  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);
END ##
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.

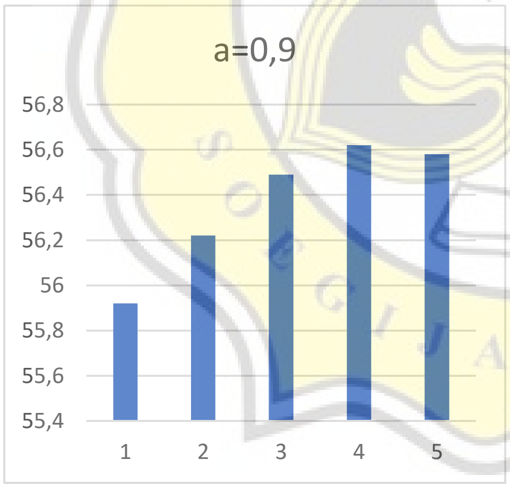

Figure 5.1 Graph of accuracy LVQ alpha=0.9

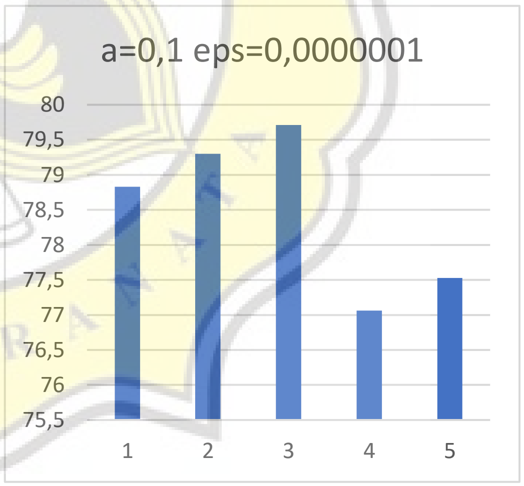

Figure 5.2 Graph of accuracy LVQ alpha=0.1 and eps=0.0000001

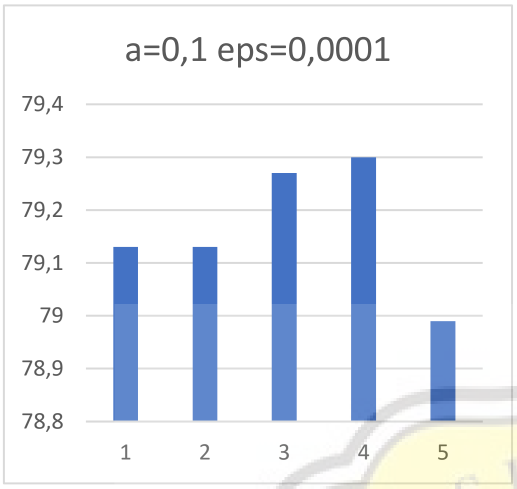

Figure 5.3 Graph of accuracy LVQ alpha=0.1 and eps=0.0001

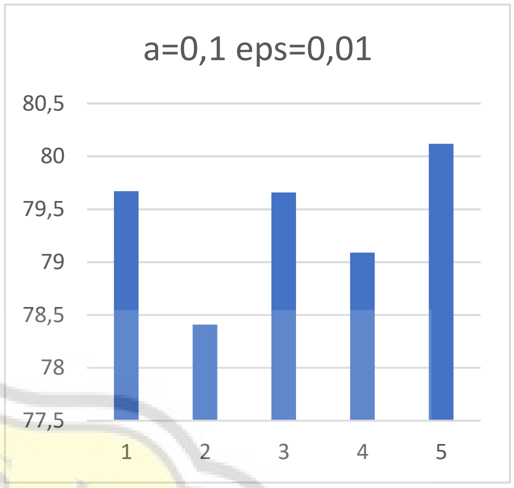

Figure 5.4 Graph of accuracy alpha=0.1 and eps=0.01

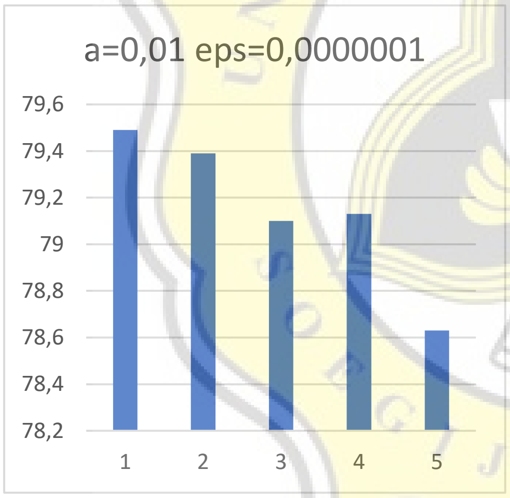

Figure 5.5 Graph of accuracy alpha=0.01 and eps=0.0000001

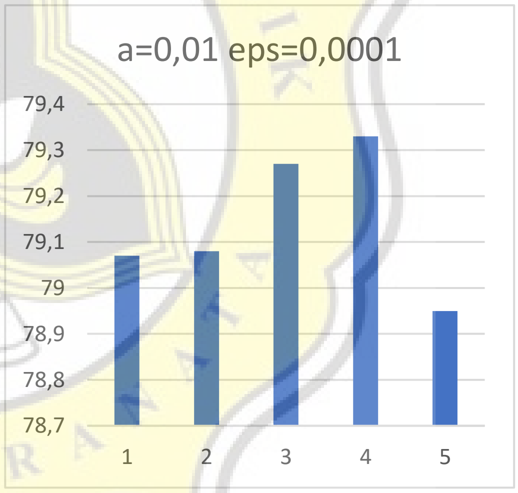

Figure 5.6 Graph of accuracy alpha=0.01 and eps=0.0001

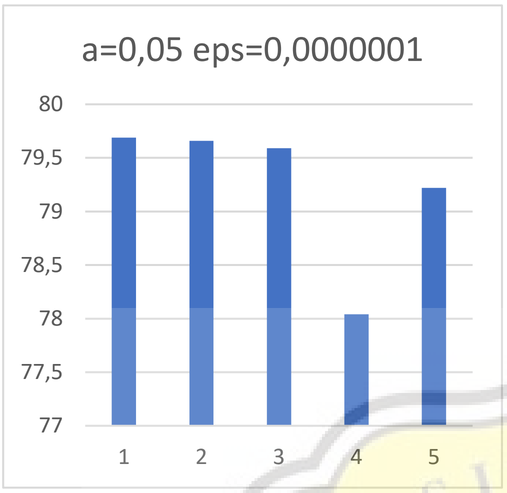

Figure 5.7 Graph of accuracy alpha=0.05 and eps=0.0000001

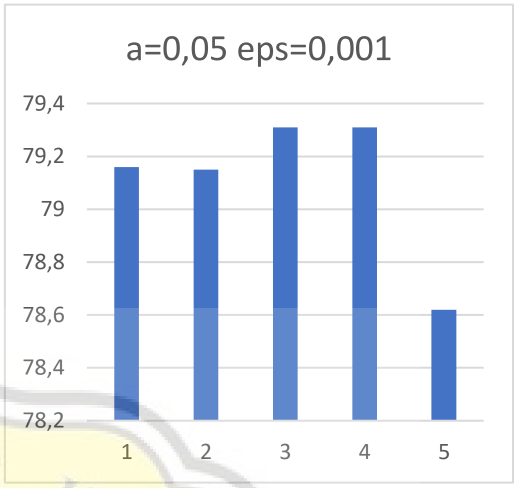

Figure 5.8 Graph of accuracy alpha=0.05 and eps=0.0001

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.