CHAPTER 3
RESEARCH METHODOLOGY

3.1 Artificial Neural Network

An artificial Neural Network is one of the artificial representations of the human brain that always tries to simulate the learning process in the human brain. An artificial neural network consists of several neurons which are often called nodes, and each neuron is connected and performs information processing as in a biological neural network system. [14].

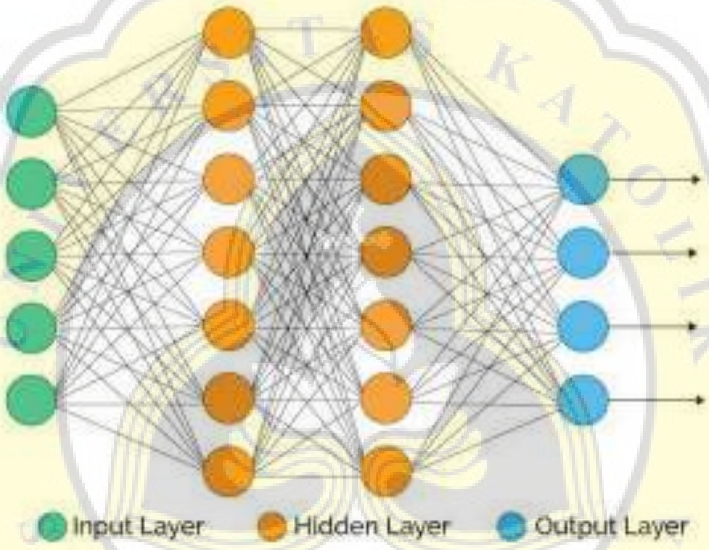

Figure 3.1 Architecture ANN [15]

The model in ANN is a function of a mathematical model that defines the function f = x -> y. In general, the ANN layer is divided into three parts, namely the input layer, the hidden layer, and the output layer [16].

3.2 Reccurental Neural Network

RNN is one of the neural network methods that is designed to be able to process data that has a sequential pattern. Sequential data has characteristics where data is processed in a sequence and data in that sequence has a close relationship with one another. This method has a hidden state to

pass the output at each stage passed. Which repeatedly the output of the previous process will return to be input in the next process [3].

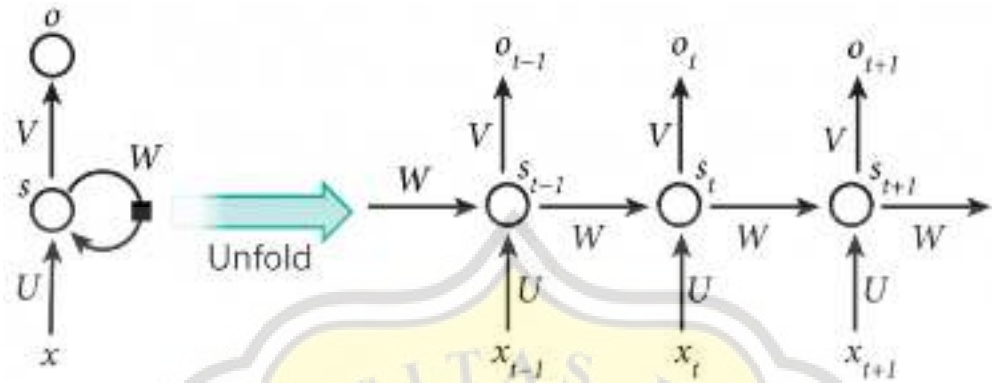

Figure 3.2 Architecture RNN [3]

For each time step t, we first calculate the state st of the input (xt) and the previous state (st-1), multiplied by the parameters U and W, and then proceed with the activation function tanh :

$$st = tanh(U . xt + W . st - 1) \#(1)$$

From st then the output t is calculated by multiplying by parameter V and passing it on to the softmax activation function :

$$\hat{y}t = softmax(V \cdot st) \#(2)$$

3.3. Activation Function

The activation function is a function used in neural networks to determine whether a neuron will be activated or not. Here are some frequently used activation functions :

- Sigmoid

The smaller the input, the output will approach zero, if the input is larger, the output will approach the value one. Because it will classify positive and negative sentiments or called binary classification so that the sigmoid output has positive and negative probabilities between values 0 and 1. The addition of the probability of positive and negative classification results will produce a value of 1. Here is the formula for Sigmoid activation [16] :

$$f(x) = sigmoid(x) = \frac{1}{1 + e^{-x}} \ \#(3)$$

- Tanh

The Tanh activation function will change its input x value into a value that has a range from -1 to 1. Just like Sigmoid, Tanh has the disadvantage that it can turn off gradients, but the advantage is that Tanh's output is zero-centered. Here is the formula for the activation of Tanh [19] :

$$tanh(x) = 2\sigma(2x) - 1 \ \#(4)$$

- Softmax

Softmax is an activation function used in the output layer. The output layer has many similarities with the fully connected layer, what distinguishes the two layers is the use of the softmax activation function in the output layer and the ReLU activation function in the fully-connected layer. Here is the formula for Softmax activation [17]:

$$\sigma(z)_j = \frac{e^{zj}}{\sum_{k=1}^{K} e^{zk}} \ for\ j = 1, \ldots, K. \#(5)$$

3.4. Adam Optimization

To produce an optimal learning process, an optimization like Adam is needed. Adam optimizes the weight change process using gradient descent, which changes the learning rate that is set dynamically so that it will avoid gradient descent being stuck at a local minimum. Comparison of performance with various types of gradient descent optimization is the best is Adam (Kingma & Ba 2015). With large models and datasets, Adam works efficiently and recommends using Adam (Ruder 2016) [16].

3.5. Binary Cross Entropy

Serves to calculate the error value of the model prediction on the training data and test data, if the error value is large then there are still many errors in the model to understand the pattern of reviews in the dataset. This study classifies polar targets, namely positive and negative sentiments or those that produce outputs between two probabilities, using Binary Cross Entropy.

BCE is calculated for every batch, if you want to know the error value per epoch then the error value of the entire batch is averaged for one epoch [16].

3.6. Batch Size dan Epoch

Epoch is when the entire dataset has gone through the training process on the Neural Network until it is returned to the beginning in one round. In a Neural Network one epoch is too big in the training process because all data is included in the training process so it will take a long time. To simplify and speed up the training process, usually, the data rate is divided per batch (Batch Size). Determining the value of the batch size usually depends on the researcher by looking at many samples [18].

3.7. Natural Language Processing (NLP)

NLP is a combination of artificial intelligence and language, used to create or manipulate computers to understand comments or words written in human language. NLP is widely used as a machine translation, text classification, sentiment analysis, spam filtering, text summarization, and so on. According to Buntoro (2017) sentiment analysis or opinion mining itself is a process of understanding, extracting, and processing textual data automatically to obtain sentiment information contained in an opinion sentence [3].

3.8. Sentiment Analysis

Opinion mining or sentiment analysis is a field of data mining that is useful for analyzing, processing, and extracting textual data on entities, such as services, products, individuals, organizations, events, or specific problems and topics. This analysis serves to obtain information from an existing data set. Sentiment analysis is new research on Natural Language Processing (NLP) and aims to find subjectivity in texts as well as extract and carry out sentiment classifications on opinions [8].

3.9. Machine Learning (ML)

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform specific tasks without using explicit instructions, relying on patterns and inference instead. It is known as part of artificial intelligence. Machine learning algorithms build mathematical models based on sample data, known as "training data", to make predictions or decisions without being explicitly programmed to do their job [19].

3.9.1 Supervised Learning

Supervised learning algorithms build a mathematical model from a data set containing the desired inputs and outputs. This data is known as training data and consists of a series of training examples. Each training instance has one or more desired inputs and outputs, also known as monitoring signals [19].

3.9.2 Unsupervised Learning

Unsupervised learning is a type of machine learning that looks for previously undetected patterns in data sets without pre-existing labels and with minimal human supervision. In contrast to supervised learning which typically uses human-labeled data, unsupervised learning, also known as self-stacking allows modeling of probability density over inputs. It forms one of the three main categories of machine learning, along with supervised and reinforced learning. Semi-supervised learning, related variant, utilizing supervised and unsupervised techniques [19].

3.10. Count-Vectorizer

Count–Vectorizer is a technique that is based on the occurrence of words in the document. In this technique, tokenization calculations can be carried out as well as many other parameters that can enhance the type of features, one of which is generating Ngrams. Count–Vectorizer also counts the number of words that occur frequently but often the number of words that appear rarely will be covered even though these words can be important in document features [20].

3.11. Data source

The data used in this study came from the social media Twitter. Twitter is an online social networking and microblogging service that allows users to send and read text-based messages from a computer or mobile device anywhere and anytime. Since its launch, Twitter has become one of the ten most visited sites on the Internet, and has been nicknamed "Messaging from the Internet". Twitter users themselves can consist of various kinds of circles whose users can interact with friends, family, and co-workers. The way to get the data is done by a process called Crawling. Crawling is a technique of collecting data on a website by entering a Uniform Resource Locator (URL). This URL is a reference to find all hyperlinks on the website. Then indexing is done to find words in the document on every link that exists. For its implementation, crawling uses an automation program and uses the Application Programming Interface (API) as a communication line in obtaining data. With the API we can collect more specific data according to existing URL links without having to know the HTML elements on a website.

In this study, the opinion data on Twitter was taken using Web Crawling with Application Programming Interface (API) keys provided by Twitter with a total of 2411 data. In the data collection process other than public opinion, researchers also took information about the Polarity Sentiment on each existing opinion. Sentiment Polarity is an element to determine the sentiment expressed by the public. Sentiment Polarity determines whether the opinion text contains positive or negative sentiments.

Table 3.1. Public Opinion Web Crawling Results

Opinion	Polarity
Things you don t have to worry about when Homeschooling	0.0
Don t elect them and remove your children to Homeschooling private schools or charters	0.4
Shoddy religious fundamentalist enforced Homeschooling	-0.15

Homeschooling my kids is going to be tough but I guess we ll make it work RiskGreaterThanReward	-0.3888888888888889
Time to start Homeschooling and limiting the social media	0.03333333333333333

3.12. Labeling

After the opinion text data and polarity are obtained through Web Crawling, the data is then analyzed by labeling each existing data. Labels are given based on the existing polarity. If the polarity shows numbers 0.0 and 0.0 and above, the data is classified as positive data. Meanwhile, polarity data showing several 0.0 and below, such as -0.1, is classified as negative data.

For the Long Short-Term Memory (LSTM) method, the label used is in the form of numbers, namely the numbers 0 and 1. The number 0 is used for texts that have negative opinions, and Number 1 is used for texts that have positive opinions. For the Support Vector Machine (SVM) method, the label used is in the form of words, namely positive and negative words.

Table 3.2. Labeling results on the LSTM method

Opinion	Polarity	Status	Label
things worry Homeschooling	0.000000	positive	1
ad Homeschooling	0.000000	positive	1
eldest child covid jab booked back home school...	0.000000	positive	1
elect remove children Homeschooling private s...	0.400000	positive	1
networked families Homeschooling need bigger...	-0.078125	negative	0

Table 3.3. Labeling results on the SVM method

Opinion	Polarity	Status	Label
things worry Homeschooling	0.000000	positive	1
ad Homeschooling	0.000000	positive	1
eldest child covid jab booked back home school...	0.000000	positive	1
elect remove children Homeschooling private s...	0.400000	positive	1
networked families Homeschooling need bigger...	-0.078125	negative	0

3.13. Preprocessing

Pre-processing is important in training data to support the process of training the algorithm so that it can correct unorganized data into organized, to simplify data processing. The collection of opinion data from Twitter social media is sometimes not the same as standard words, words that are not in the dictionary, using regional languages, or being abbreviated. To revert some text to natural text by eliminating atypical expressions to minimize noise at a later stage, pre-processing or normalization is needed to overcome this [12].

3.13.1. Cleansing

At this initial stage, the existing text data is cleaned by removing symbols, numbers, punctuation marks, redundant spaces, and characters that are not in the alphabet. And also change the abbreviated word to the original word [2].

Table 3.4. Preprocessing Data Cleansing

Sentence	Data Cleansing
They networked with 9 other families and are Homeschooling We need a bigger plan here	They networked with other families and are Homeschooling We need a bigger plan here

3.13.2. Case Folding

At this stage, all the letters that exist are equalized in form, namely to be all lowercase letters or all uppercase letters [2].

Table 3.5. Preprocessing Case Folding

Sentence	Case Folding
They networked with other families and are Homeschooling We need a bigger plan here	they networked with other families and are Homeschooling we need a bigger plan here

3.13.3. Stopwords Removal

This stage serves to delete words that are not important and have no meaning for the next process or delete words that are often repeated [2].

Table 3.6. Preprocessing Stopwords Removal

Sentence	Stopwords Removal
they networked with other families and are Homeschooling we need a bigger plan here	networked families Homeschooling need bigger plan

3.13.4. Tokenization

This stage has the role of separating the document text into a series of tokens or words [2].

Table 3.7. Preprocessing Tokenization

Sentence	Tokenization
networked families Homeschooling need bigger plan	[‘networked’, ‘families’, ‘home’, ‘schooling’, ‘need’, ‘bigger’, ‘plan’]

3.14. Training Data and Testing Data

After the opinion text data is pre-processed, the data is divided into 2, namely Training Data and Testing Data. Data Training aims to train the algorithm that we will use. While Data Testing aims to determine the performance of the previously trained algorithm. Opinion data as many as 2411 data will be divided into training data and testing data with a comparison ratio :

Table 3.8. Split Training Data and Testing Data

Training Data	Testing Data
80%	20%
50%	50%
20%	80%

3.15. Word Embedding

After the opinion text data is pre-processed and divided into Training Data and Testing Data, the opinion text data will then be converted into numbers and made a mapping or word embedding which will later become input for the Long Short-Term Memory classification method. Word embedding or word vector is a method that works by mapping words in vector form. Semantically related words will be mapped in adjacent vector values. The result is, words that have semantic similarities will be in the same area as words that have similarities with these words. The word vector process in this study uses one of the features provided by Keras called Embedding [5].

Here are some steps when doing the Word Embedding process :

3.15.1. Word Index

Data that has gone through the preprocessing process and a list has been formed will be indexed which is the process of converting unique words into numbers which will later represent the word before entering the word embedding stage [5].

3.15.2. Padding / Pad Sequences

After getting the index of each word, the next step is to do the padding for each sentence. Padding is done to equalize the length of the list of each sentence so that it can be input during the classification process. In making padding, it must be declared in advance the maximum length of all existing sentence vectors. By using padding, a list of sentences whose length is not up to the specified value will be added a value of 0 in front of it [5].

3.16. Long Short Term Memory (LSTM)

LSTM is a development of the RNN model which can also be used to manage sequential data. Initially developed by Hochreiter & Schmidhuber (1997) to overcome the problem of the RNN model, namely vanishing gradient.

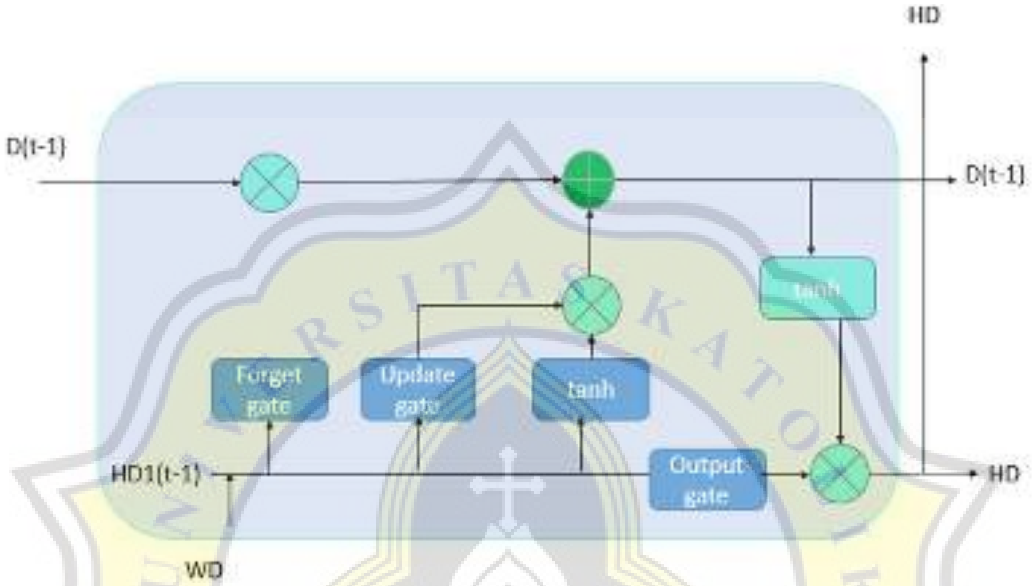

Figure 3.3 Long Short-Term Memory Network Structure [21]

The architecture of the LSTM model has a cell state that can store data and pass it on to the next LSTM. The cell state consists of the previous cell state which stores the old context and the output cell state which stores the new context. It is also supported by three gates, including input gate, forget gate, and output gate. In the forget gate, the LSTM method will learn or give a decision that a context in the previous cell state will be saved or forgotten. Using the sigmoid activation function, outputs with values close to 0 will be forgotten, and outputs with values close to 1 [3].

Here is the formula for each gate in LSTM :

- Forget Gate (ft)

  At this gate, the previous output value with the current input is combined and then passes through the sigmoid activation function. It is this gate that determines whether the previous information will be forgotten or not. Then this information is continued to the memory cell or cell state [2].

$$ft = \sigma(Wf \times [xt + ht - 1] + bf) \#(6)$$

  ft : forget gate — xt : input cell

  σ : sigmoid activation function — ht−1 : output cell previously

  Wf : weights forget gate — bf : bias forget gate

- Input Gate (it)

  At this gate the previous output value with the current input is combined, then two activation functions will be passed. One path passes through the sigmoid activation function for input values, the other path passes through the tanh activation function for candidate memory cell values [2].

$$it = \sigma(Wi \times [xt + ht - 1] + bi) \#(7)$$

$$C\tilde{}t = tanh(WC \times [xt + ht - 1] + bC) \#(8)$$

- it : input gate — C˜ t : candidate
- σ : sigmoid activation function — tanh : tanh activation function
- Wi : weights input — WC : weights candidate
- xt : input cell — bC : bias candidate
- ht−1 : output cell previously — bi : bias input gate

- Cell State (ct)

  At this stage, there is an amalgamation of the two values. The first value is the value of the forget gate which will be multiplied by the value of the previous cell state. The second value is the value of the input gate multiplied by the value of the candidate memory cell [2].

$$Ct = ft \times Ct - 1 + it \times C\tilde{}t \#(9)$$

  Ct : cell state — it : input gate

  ft : forget gate — C˜ t : candidate

  Ct−1 : cell state previously

- Output Gate (ot)

  This gate produces an output value, where this value comes from the combination of the previous value with the current value that has gone through the sigmoid activation function [2].

$$ot = \sigma(Wo \times [xt + ht - 1] + bo) \#(10)$$

  ot : output gate — xt : input cell

  σ : sigmoid activation function — ht−1 : output cell previously

  Wo : weights output gate — bo : bias output gate

- Hidden State (st)

  The hidden layer affects the value in the next process, the value of this layer comes from the output value multiplied by the value of the cell state or memory cell that has been activated with the tangent function [2].

$$ht = ot \times tanh(Ct) \#(11)$$

  ht : hidden layer — ot : output gate

  tanh : tanh activation function — Ct : cell state

3.17. Fully Connected Layer

The Fully Connected layer is a layer that is usually used in MLP applications and aims to transform the data dimensions so that the data can be classified linearly. Each neuron in the convolution layer needs to be transformed into one-dimensional data before it can be entered into a fully connected layer. Because this causes the data to lose its spatial information and is not reversible, the fully connected layer can only be implemented at the end of network [22].

3.18. Support Vector Machine (SVM)

Support Vector Machine is a machine learning technique that is quite popular for text classification and has good performance in many domains and can identify hyperplanes separately between two different classes so that the results are maximized and can also maximize the distance between the data closest to the hyperplane. Classification is done by looking for a hyperplane or decision boundary that separates a class from another class, Support Vector Machine searches for hyperplane values using support vectors and margin values [23].

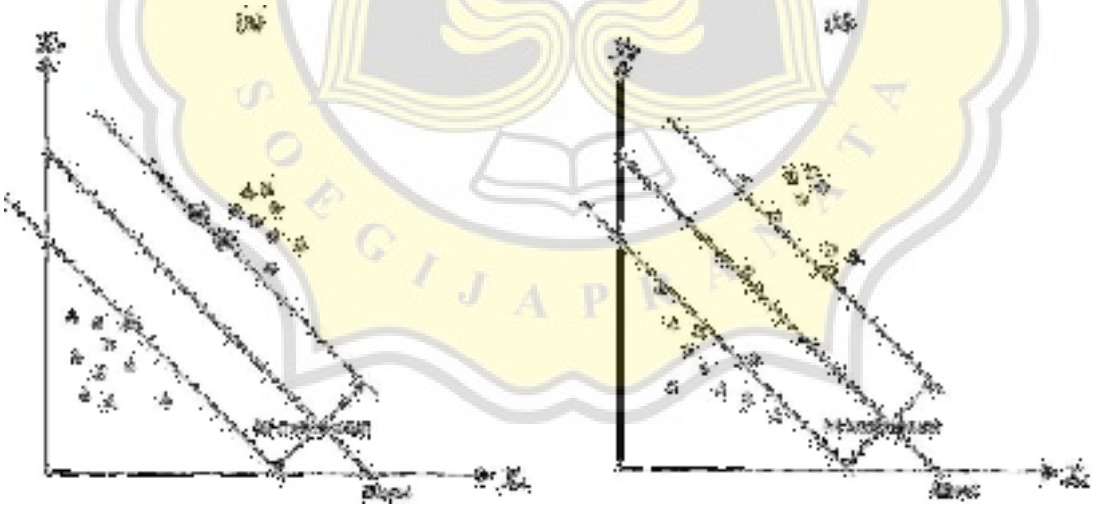

Figure 3.4 Best Hyperplane and Maximum Margin [9]

$$y = w\,T\,xi + b\,, i = 1,2, \dots, l\ \#(12)$$

$xi = [x1, x2, \dots, xk]$ is a row vector with dimension k (number of features)

$y \in \{-1, +1\}$ = target value of the data xi

l = number of data

$w = [w1, w2, \dots, wk]$ is a row vector which is a weight parameter

b = bias or error

The vector weight (w) is a vector line that is perpendicular between the center of the coordinates and the hyperplane line. Bias (b) is the line coordinates relative to the point coordinates. Equation (13) is an equation for calculating the value of b, while equation (14) is an equation to find the value of w [13].

$$b = -\frac{1}{2}(w.x^{+} + w.x^{-}) \#(13)$$

$$w = \sum_{i}^{n} = 1\ \alpha_i \square_i x_i \#(14)$$

b = bias value

$w.x+$ = weight value for positive data class

$w.x-$ = weight value for negative data class

w = vector weight

αi = nilai bobot data ke-i

yi = data class

xi = data

To determine the optimal hyperplane of the two classes using the equation:

$$Minimize\ J1\ [w] = \frac{1}{2}\ ||w||^{2} \#(15)$$

In general, there are four types of kernel functions that can be used [9]:

1. Linear Kernel

$$K(x_i, x_j) = x_i^t x_j \#(16)$$

2. Polynomial Kernel

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0 \#(17)$$

3. Gaussian Kernel (Radial Basis Function)

$$K(x_i, x_j) = \exp\{-\gamma \| x_i - x_j | \ |^2 \}, \gamma > 0 \#(18)$$

4. Sigmoid Kernel

$$K(x, x_k) = \tanh[\gamma x_i^T x_j + r] \#(19)$$

3.19. Parameter Regularization

The lambda parameter is used in the part of the cost function, namely regularization, so it is also called the regularization parameter. Lambda is a real number, so it doesn't affect the training speed [24].

3.20. Confusion Matrix

Confusion Matrix is a table or matrix that contains four values which are performance measurements of the classification problems that have been carried out. There are four values or points in the confusion matrix, namely True Positive, True Negative, False Positive, and False Negative [2].

Table 3.9. Confusion Matrix [25]

Actual Label	**Predicted Label**	
	Negative	***Positive***
Negative	TN	FP
Positive	FN	TP

Description :

- True positive (TP) : Prediction that is positive and correct according to the target.
- True negative (TN) : Prediction that is negative and correct according to the target.
- False positive (FP) : Predictions that are positive and false are not on target.
- False negative (FN) : Predictions that are negative and false are not on target.

After knowing the confusion matrix, it can also be seen the values of accuracy, precision, recall, and f1-score. Here is an explanation and formula to find out :

- Accuracy

  Accuracy is a calculation of how precisely the classification that has been built is following the existing target.

$$\frac{TP + TN}{TP + TN + FP + FN} \#(13)$$

- Precision

  Precision is the calculation of the accuracy between the target data and the prediction results from the model.

$$\frac{TP}{TP + FP} \#(14)$$

- Recall

  Recall is a calculation that describes the success of the model in finding back information.

$$\frac{TP}{TP + FN} \#(15)$$

- F1-Score

  F1-score is a calculation that describes the comparison between precision and recall. If the FN and FP values are not close, the f1-score should be used instead of the accuracy value.

$$2 . \frac{Precision . Recall}{Precision + Recall} \#(16)$$