CHAPTER 4
ANALYSIS AND DESIGN

4.1. Overview

In the previous chapter, we explained all about the methodology from this research. Now, we are going to explain about the performance of the model, especially in machine learning algorithms. If we do machine learning, of course it would explain the performance. As we already mentioned in the previous chapter, the evaluation method of this performance includes precision, recall, F1-Score, and accuracy.

In this chapter, we outline the analysis and design of this research. Firstly, we explain the analysis method which was used in this research. Second, we describe how the design and model perform in this research. It explained with the picture, hopefully the reader can understand more with the pictures.

4.2. Analysis

The goal from this research is to get research's performance by precision, recall, F1-Score, and accuracy. The first analysis is supervised learning with Adaboost(Adaptive Boosting) algorithm and the second is semi-supervised learning with Adaboost (Adaptive Boosting) algorithm. In multi classification, we differentiate them into two parts. The first is macro average and the second is micro average. Macro average is used when we have more attention about performance in each class. While micro average used when we did not care about performance in each class.

For the first is precision, precision is the comparison between true positives and the positive predictions [24]. Precision shown at the **Figure 4.1.** below.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Figure 4.1 *Precision General Formula*

The **figure 4.1** count precision in general or used when in binary classification, we used true positive divided by true positive add with false positive. True positives means how many numbers while the models predict true and the actual class is true. False positives means that how many numbers while the models predict are true but the actual class is false. Higher precision indicates fewer false positives.

$$P = \frac{\sum_c TP_c}{\sum_c TP_c + \sum_c FP_c}$$

Figure 4.2 *Precision Micro-Averaged*

Macro-precision calculated with precision from each class which is used one vs rest method with that formula. If we used one vs rest, we can imagine that each class is a binary classification. While their precisions from each class is added and then divided with the number of classes. Micro-precision counts true positive and false negative in each class and then calculated with that formula.

The second, recall measures the proportion of positives that are correctly identified as such [24]. Recall count the data which in actual fact is true. So, the division are true positives divided true positif add false negatives. Higher recall indicates fewer false negatives. Recall shown at Figure **4.2.2.** below.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Figure 4.3 *Recall General Formula*

The **Figure 4.3** to count recall in general or used when in binary classification, we used true positive divided by true positive add with false negatives. True positives means how many numbers while the models predict true and the actual class is true. False negatives means how many numbers while the models predict false but the actual class is true.

$$R = \frac{\sum_c TP_c}{\sum_c TP_c + \sum_c FN_c}$$

Figure 4.4 *Recall Micro-Averaged*

Macro-recall in **Figure 4.4** calculated with precision from each class which uses a one vs rest method with that formula. If we used one vs rest, we can imagine that each class is a binary classification. While their recall from each class is added and then divided with the number of classes. Micro- recall counts true positive and false negative in each class and then calculated with that formula.

The third is F1-Score, the F1 score is the harmonic mean of precision and recall. To calculate F1-Score using this formula below.

$$\text{ma}F_1 = \left(\sum_{j=1}^{M} \frac{2P_j R_j}{P_j + R_j} \right) / M$$

Figure 4.5 *Macro-Averaged F1 Score*

From **Figure 4.5**, the macro-averaged F1 score is obtained with precision and recall in each class and then summed together and divided by the number of classes. While the micro-averaged for F1 score formula is:

$$\text{mi}F_1 = \frac{2PR}{P + R} = \sum_{j=1}^{M} \mu_j \theta_{jj}$$

Figure 4.6 *Micro-Averaged F1 Score*

From **Figure 4.6**, micro-averaged for F1 score get two times with micro precision times with micro recall divided by micro-precision add with micro recall. So, we get a micro-averaged F1 Score in this evaluation model.

4.3. Design

After we evaluate the model's performance, we are now explained how the design of this model works. Look at this picture **Figure 4.7** below

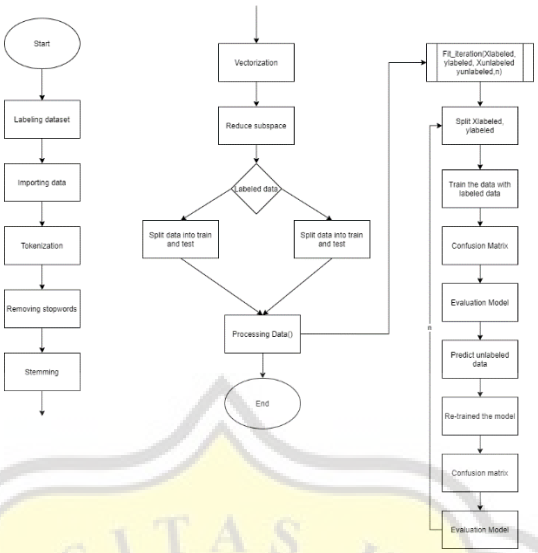
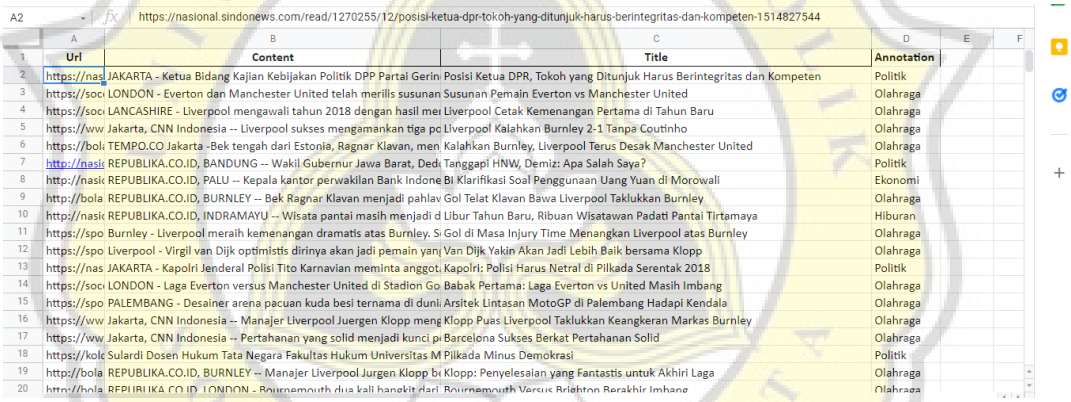


Figure 4.7 *Flowchart for News Categorization*

Below show how news labeled



A2 https://nasional.sindonews.com/read/1270255/12/posisi-ketua-dpr-tokoh-yang-ditunjuk-harus-berintegritas-dan-kompeten-1514827544

	A	B	C	D	E	F
1	**Url**	**Content**	**Title**	**Annotation**		
2	https://nas	JAKARTA - Ketua Bidang Kajian Kebijakan Politik DPP Partai Gerin	Posisi Ketua DPR, Tokoh yang Ditunjuk Harus Berintegritas dan Kompeten	Politik		
3	https://soc	LONDON - Everton dan Manchester United telah merilis susunan	Susunan Pemain Everton vs Manchester United	Olahraga		
4	https://soc	LANCASHIRE - Liverpool mengawali tahun 2018 dengan hasil me	Liverpool Cetak Kemenangan Pertama di Tahun Baru	Olahraga		
5	https://ww	Jakarta, CNN Indonesia -- Liverpool sukses mengamankan tiga po	Liverpool Kalahkan Burnley 2-1 Tanpa Coutinho	Olahraga		
6	https://bol	TEMPO.CO Jakarta -Bek tengah dari Estonia, Ragnar Klavan, men	Kalahkan Burnley, Liverpool Terus Desak Manchester United	Olahraga		
7	http://nasi	REPUBLIKA.CO.ID, BANDUNG -- Wakil Gubernur Jawa Barat, Ded	Tanggapi HNW, Demiz: Apa Salah Saya?	Politik		
8	http://nasi	REPUBLIKA.CO.ID, PALU -- Kepala kantor perwakilan Bank Indone	BI Klarifikasi Soal Penggunaan Uang Yuan di Morowali	Ekonomi		
9	http://bola	REPUBLIKA.CO.ID, BURNLEY -- Bek Ragnar Klavan menjadi pahlav	Gol Telat Klavan Bawa Liverpool Taklukkan Burnley	Olahraga		
10	http://nasi	REPUBLIKA.CO.ID, INDRAMAYU -- Wisata pantai masih menjadi d	Libur Tahun Baru, Ribuan Wisatawan Padati Pantai Tirtamaya	Hiburan		
11	https://spo	Burnley - Liverpool meraih kemenangan dramatis atas Burnley. S	Gol di Masa Injury Time Menangkan Liverpool atas Burnley	Olahraga		
12	https://spo	Liverpool - Virgil van Dijk optimistis dirinya akan jadi pemain yan	Van Dijk Yakin Akan Jadi Lebih Baik bersama Klopp	Olahraga		
13	https://nas	JAKARTA - Kapolri Jenderal Polisi Tito Karnavian meminta anggot	Kapolri: Polisi Harus Netral di Pilkada Serentak 2018	Politik		
14	https://soc	LONDON - Laga Everton versus Manchester United di Stadion Go	Babak Pertama: Laga Everton vs United Masih Imbang	Olahraga		
15	https://spo	PALEMBANG - Desainer arena pacuan kuda besi ternama di duni	Arsitek Lintasan MotoGP di Palembang Hadapi Kendala	Olahraga		
16	https://ww	Jakarta, CNN Indonesia -- Manajer Liverpool Juergen Klopp meng	Klopp Puas Liverpool Taklukkan Keangkeran Markas Burnley	Olahraga		
17	https://ww	Jakarta, CNN Indonesia -- Pertahanan yang solid menjadi kunci p	Barcelona Sukses Berkat Pertahanan Solid	Olahraga		
18	https://kol	Sulardi Dosen Hukum Tata Negara Fakultas Hukum Universitas M	Pilkada Minus Demokrasi	Politik		
19	http://bola	REPUBLIKA.CO.ID, BURNLEY -- Manajer Liverpool Jurgen Klopp b	Klopp: Penyelesaian yang Fantastis untuk Akhiri Laga	Olahraga		
20	http://bola	REPUBLIKA.CO.ID, LONDON - Bournemouth dua kali bangkit dari	Bournemouth Versus Brighton Berakhir Imbang	Olahraga		

Figure 4.8 *Labeled Dataset*

After the data is labeled by a human, now we preprocess the data until processing as we explained in the previous chapter. In processing data, we split into two parts, processing data and modelling. In the modelling we used Adaboost Algorithm with semi-supervised learning.

The first popular boosting algorithm is Adaboost (Adaptive Boosting) algorithm that works in linear combination with a weak classifier. Boosting algorithm is an ensemble learning technique for improving the training process. Adaboost is relevant with binary classification, in this project, we proposed SAMME for multi-classification. It was developed by M. Adaboost, J. Zhu, H. Zou, S. Rosset, and T. Hastie, a new algorithm that directly extends the AdaBoost algorithm to the multi-class case without reducing it to multiple two-class problems[23]. The algorithm, it explained by the picture below.

1. *Initialize the observation weights* $w_i = 1/n$, $i = 1, 2, \ldots, n$.
2. *For* $m = 1$ *to* M:
   (a) *Fit a classifier* $T^{(m)}(\boldsymbol{x})$ *to the training data using weights* w_i.
   (b) *Compute*

$$err^{(m)} = \sum_{i=1}^{n} w_i \mathbb{I}\left(c_i \neq T^{(m)}(\boldsymbol{x}_i)\right) / \sum_{i=1}^{n} w_i.$$

   (c) *Compute*

$$\alpha^{(m)} = \log \frac{1 - err^{(m)}}{err^{(m)}} + \log(K - 1). \quad (1)$$

   (d) *Set*

$$w_i \leftarrow w_i \cdot \exp\left(\alpha^{(m)} \cdot \mathbb{I}\left(c_i \neq T^{(m)}(\boldsymbol{x}_i)\right)\right),$$

   for $i = 1, \ldots, n$.
   (e) *Re-normalize* w_i.
3. *Output*

$$C(\boldsymbol{x}) = \arg\max_k \sum_{m=1}^{M} \alpha^{(m)} \cdot \mathbb{I}(T^{(m)}(\boldsymbol{x}) = k).$$

Figure 4.9 *Adaboost Multiclass Formula*

The process of the SAMME algorithm is defined into 3 steps. So, for the first is we initialized weight for each data in a row. The weight is 1/N where N is the number of data trains. Second, this step has many sub-step, we start the iteration. The goal of this iteration is to get the best value of alpha and get the total gain where the alpha is affected by the weight value. The weight value is affected the classification problem. The weight will decrease if the classification of the dataset is wrong predicted. The trueness of the classification process belongs to the data train. After getting the weight, now count the error rate of misclassified from the classification. This error rate value is used as a multiplication number to get the new weight of each data. Then, get the alpha value with the formula:

$$\alpha^{(m)} = \log \frac{1 - err^{(m)}}{err^{(m)}} + \log(K - 1).$$

Figure 4.10 *The Alpha Value SAMME Formula*

Now, we get alpha value to count the new weight by the formula:

$$w_i \leftarrow w_i \cdot \exp\left(\alpha^{(m)} \cdot \mathbb{I}\left(c_i \neq T^{(m)}(\boldsymbol{x}_i)\right)\right),$$

$$for\ i = 1, \ldots, n.$$

Figure 4.11 *The New Weight Value SAMME Formula*

This weight is used for the input variable in the next iteration. This iteration will run until some decision stumps which are declared at the beginning of this SAMME algorithm code. This iteration will get the best Alpha value and the total of gain to predict the different data. For the predict we use the formula:

Prediction

$$y = sign(\sum_t^T \alpha_t \cdot h(X))$$

Figure 4.12 *SAMME Predicted Formula*

From that, we get the predicted value from the sum of alpha and total gain of each row of the dataset. Remember, the max depth of the tree is 1. So, the value from every stump is summed together. Referring to this project, we used scratch code for this algorithm, not just used the library. So, we can explore more details about this algorithm.