CHAPTER 3
RESEARCH METHODOLOGY

3.1. Overview

In the previous chapter, we have studied literature about news in general, how Natural Language Processing (NLP) works, and how this project is different from an existing project. In this chapter, we only focus on the research methodology that is used. The methodology is divided into two parts in general. The first is about the dataset, how we preprocessed the dataset until how we processed the dataset. The second is about the algorithm that we used for news categorization. To visualize that methodology, look at this picture of this flowchart **Figure 3.1** below.

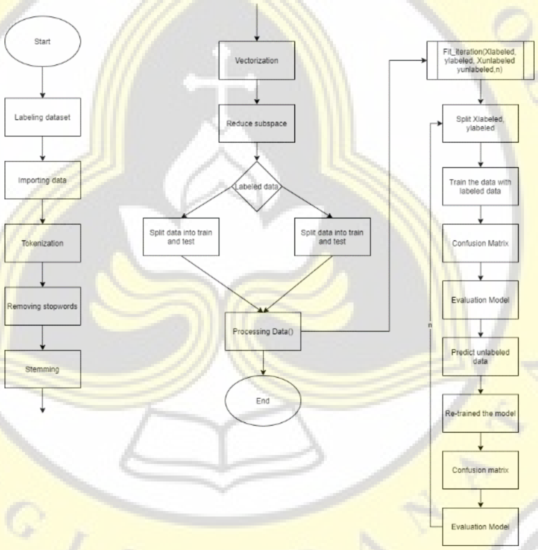


Figure 3.1 *Flowchart of This Research*

3.2. Dataset

Refer the project using the dataset from GitHub forum which the dataset got from someone who crawled data. The dataset is about a collection of news. News is a piece of information about events or a recent incident. News not only spreads through television, newspaper, and radio, but it also spreads through the Internet [17]. The news source comes from CNN, Sindonews, Tempo, Republika, Kompas, etc. There are many rows about a news article that contains in many columns such as URL, content, title, date, and timestamp. The unlabeled dataset turned into a labeled

UNIVERSITAS KATOLIK SOEGIJAPRANATA

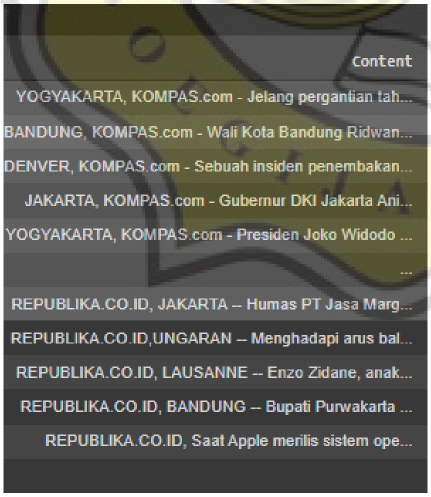
dataset which was labeled by Davin Chang from the student year 2019 Soegijapranata Catholic University.

This project divides the dataset for training and testing is 25:75, 50:50, and 75:25. Hopefully, this research can show how the division of the dataset can affect the result of the training. It created a dataset divided into 3 classes: *Politik, Ekonomi, and Olahraga*. That classification also uses Politik-304.42 words, Ekonomi-252.87 words, Olahraga-245.71 words. The total number of rows of the dataset is 365 rows.

Preprocessing is preparing data for further processing aiming to obtain a good performance of the system. It consists of case folding, tokenization, stopword removal, and stemming [17]. Preprocessing is making unstructured data into structured data [14]. In this project, unstructured data comes from the content of the news that is full of symbols like "-",",_", etc. Besides that, the unused word like where the news is published is not necessarily needed with the training later.

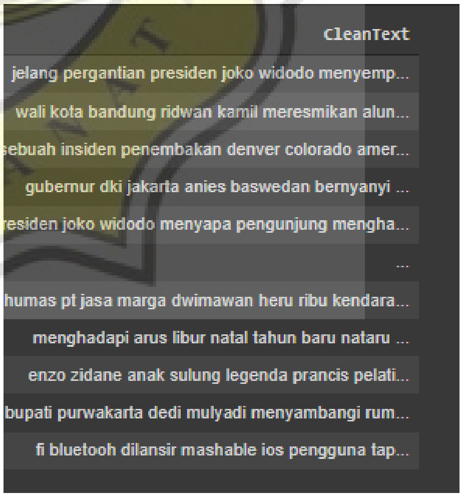
1. Stopwords Removal

   This research uses a library named Regular Expression (RegEx). This library can handle unused words in the news content. For example stopwords, punctuation, whitespace, hyperlink, and other words that are not necessarily for supervised training. It also used a list of stopwords that get from the internet which is from a .txt format file and filtered using that.



```
                                     Content
 YOGYAKARTA, KOMPAS.com - Jelang pergantian tah...
BANDUNG, KOMPAS.com - Wali Kota Bandung Ridwan...
DENVER, KOMPAS.com - Sebuah insiden penembakan...
   JAKARTA, KOMPAS.com - Gubernur DKI Jakarta Ani...
YOGYAKARTA, KOMPAS.com - Presiden Joko Widodo ...
                                                ...
 REPUBLIKA.CO.ID, JAKARTA -- Humas PT Jasa Marg...
 REPUBLIKA.CO.ID,UNGARAN -- Menghadapi arus bal...
  REPUBLIKA.CO.ID, LAUSANNE -- Enzo Zidane, anak...
   REPUBLIKA.CO.ID, BANDUNG -- Bupati Purwakarta ...
       REPUBLIKA.CO.ID, Saat Apple merilis sistem ope...
```

Figure 3.3 Dataset Before Stopwords Removal



```
                                   CleanText
jelang pergantian presiden joko widodo menyemp...
 wali kota bandung ridwan kamil meresmikan alun...
sebuah insiden penembakan denver colorado amer...
   gubernur dki jakarta anies baswedan bernyanyi ...
residen joko widodo menyapa pengunjung mengha...
                                                ...
humas pt jasa marga dwimawan heru ribu kendara...
     menghadapi arus libur natal tahun baru nataru ...
    enzo zidane anak sulung legenda prancis pelati...
bupati purwakarta dedi mulyadi menyambangi rum...
      fi bluetooh dilansir mashable ios pengguna tap...
```

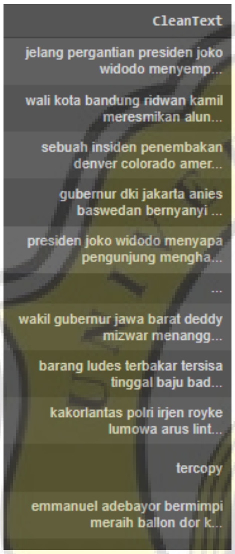
Figure 3.2 Dataset After Stopwords Removal

2. Case Folding

   After cleaning the data, this research lowered the case of the word. The purpose of this process is to make the system read the same lowercase word. So, if there are the same words just different by uppercase the system does not differentiate them.

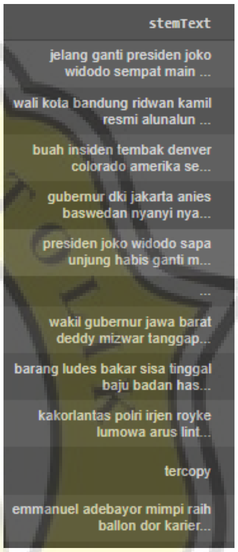
3. Stemming

   The stemming process is to make a basic form of the word. This process extracts words and removes prefixes, inserts, suffixes, or any combination of prefixes and suffix [17]. The stemming process spend 5-10 minutes for 365 rows



CleanText
jelang pergantian presiden joko widodo menyemp...
wali kota bandung ridwan kamil meresmikan alun...
sebuah insiden penembakan denver colorado amer...
gubernur dki jakarta anies baswedan bernyanyi ...
presiden joko widodo menyapa pengunjung mengha...
...
wakil gubernur jawa barat deddy mizwar menangg...
barang ludes terbakar tersisa tinggal baju bad...
kakorlantas polri irjen royke lumowa arus lint...
tercopy
emmanuel adebayor bermimpi meraih ballon dor k...

Figure 3.5 Dataset Before Stemming

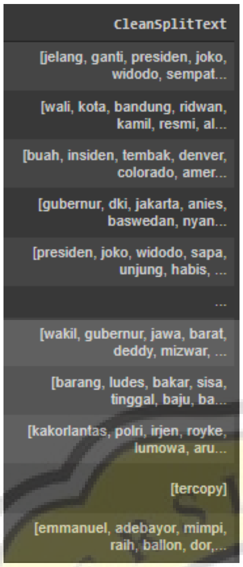


stemText
jelang ganti presiden joko widodo sempat main ...
wali kota bandung ridwan kamil resmi alunalun ...
buah insiden tembak denver colorado amerika se...
gubernur dki jakarta anies baswedan nyanyi nya...
presiden joko widodo sapa unjung habis ganti m...
...
wakil gubernur jawa barat deddy mizwar tanggap...
barang ludes bakar sisa tinggal baju badan has...
kakorlantas polri irjen royke lumowa arus lint...
tercopy
emmanuel adebayor mimpi raih ballon dor karier...

Figure 3.4 Dataset After Stemming

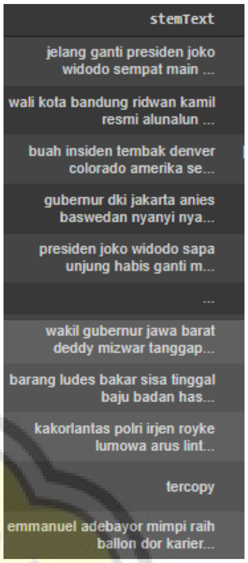
4. Tokenization

   After many several processes above, this research split sentences into a list of words where each word is called token. This process used for making the weight of each tokenize, which it used to form a matrix of weight that was used for training that only read an array only.



```
                    CleanSplitText
   [jelang, ganti, presiden, joko,
                  widodo, sempat...
     [wali, kota, bandung, ridwan,
                 kamil, resmi, al...
  [buah, insiden, tembak, denver,
                  colorado, amer...
   [gubernur, dki, jakarta, anies,
                   baswedan, nyan...
   [presiden, joko, widodo, sapa,
                    unjung, habis, ...
                                 ...
    [wakil, gubernur, jawa, barat,
                    deddy, mizwar, ...
      [barang, ludes, bakar, sisa,
                   tinggal, baju, ba...
 [kakorlantas, polri, irjen, royke,
                     lumowa, aru...
                          [tercopy]
   [emmanuel, adebayor, mimpi,
                    raih, ballon, dor,...
```

Figure 3.6 *Dataset Before Tokenization*



```
                          stemText
      jelang ganti presiden joko
           widodo sempat main ...
  wali kota bandung ridwan kamil
                resmi alunalun ...
      buah insiden tembak denver
            colorado amerika se...
      gubernur dki jakarta anies
           baswedan nyanyi nya...
     presiden joko widodo sapa
          unjung habis ganti m...
                                ...
       wakil gubernur jawa barat
          deddy mizwar tanggap...
  barang ludes bakar sisa tinggal
                 baju badan has...
       kakorlantas polri irjen royke
                  lumowa arus lint...
                           tercopy
  emmanuel adebayor mimpi raih
                  ballon dor karier...
```

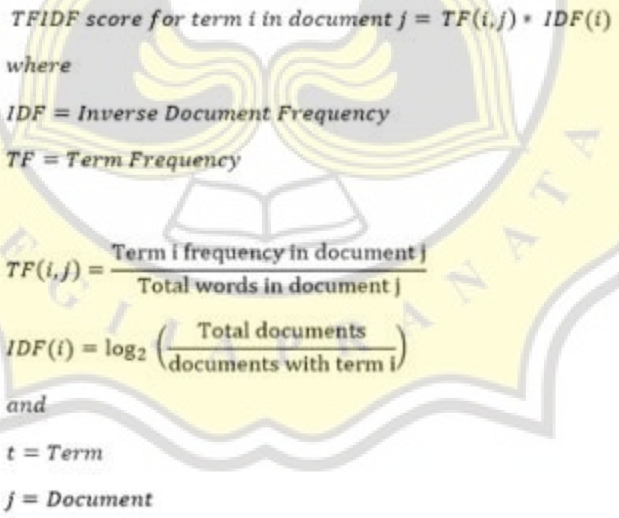
Figure 3.7 *Dataset After Tokenization*

UNIVERSITAS KATOLIK SOEGIJAPRANATA

3.2.1. Data Processing

Before making data that can be read by machine learning model or feature selection, it transforms into an array of numbers. This process is also called feature extraction. It also transforms the tokenized word into weight. This process is mentioned as count weight for every tokenized word that we get after preprocessing the text. It is done with TF-IDF. TF-IDF is a combination of TF and IDF [6]. TF stands for term frequency while IDF stands for Inverse Document Frequency. Qaiser Shahzad said TF is used to measure how many times a term is present in a document [19]. From this research, the term means the number of the unique word in a row dataset. While Inverse Document Frequency (IDF) counts how many documents with that word

Refer to this project, so that every tokenized word that we get from the dataset transforms into a list of weight. This method is also said to be feature selection [6]. Firstly, this research counts every word and makes a comparison with the count of another word. From that, this research gets the TF-IDF from every tokenized word and appends it into the list of every row of the dataset. This formula **Figure 3.2.7** can show how to calculate that :



$$TFIDF\ score\ for\ term\ i\ in\ document\ j = TF(i,j) * IDF(i)$$

$$where$$

$$IDF = Inverse\ Document\ Frequency$$

$$TF = Term\ Frequency$$

$$TF(i,j) = \frac{\text{Term i frequency in document j}}{\text{Total words in document j}}$$

$$IDF(i) = \log_2\left(\frac{\text{Total documents}}{\text{documents with term i}}\right)$$

$$and$$

$$t = Term$$

$$j = Document$$

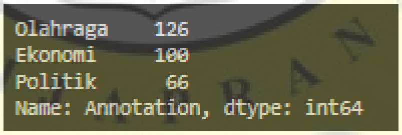
Figure 3.8 *TF-IDF Formula*

After getting the weight of every tokenized word, this project also used the SVD model. The purpose of doing this is that reduce the dimensional data. SVD is usually used after TF-IDF, a vector from the result of TF-IDF used to calculate SVD. This result from SVD will be used for the model or classifier. It should be reduced because of the difficulties of feature selection in the number of data especially in Natural Language Processing (NLP).

3.2.2. Imbalanced Class

P. Chujai, K. Chomboon, P. Teerarassamee, N. Kerdprasop, and K. Kerdprasop said that imbalanced data is information that occurs in everyday life. They have an unequal number in each class. In this project, we used 100 rows for the "Ekonomi" class, 126 rows for the "Hiburan" class, and 66 rows for the "Teknologi" class. It said the number for each class is imbalanced and it also affects the performance of learning especially in data mining. The boundary of a decision in imbalanced data chosen by most algorithms of machine learning makes bias through the majority class and hence misclassifies the minority class. The methods for solving imbalanced classification problems can be divided into three approaches as follows: (a) Data Level Approaches, (b) Algorithm Level Approaches, (c) Cost-Sensitive Approaches [20].

Referring to this project, we only focus on Data Level Approaches. This approach solves the problem in a pre-processing stage by rebalancing the class distribution using the sampling techniques. The method of both undersampling and oversampling is to reduce the most number of data and add data which is less number of data. But, this project also uses ensemble learning to solve that problem. It used ensemble learning with boosting methods to improve a model that can fix misclassification learning [20]. But, it does not rule out the possibility of misclassification from that dataset. Although imbalanced data increases, it can be misclassified by this model.



```
Olahraga    126
Ekonomi     100
Politik      66
Name: Annotation, dtype: int64
```

Figure 3.9 *Imbalanced Dataset*

3.2.3. Outliers

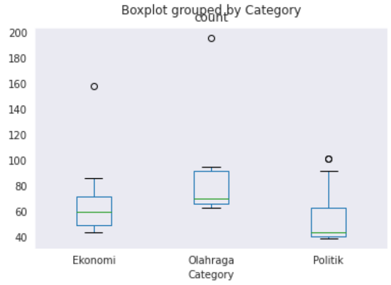
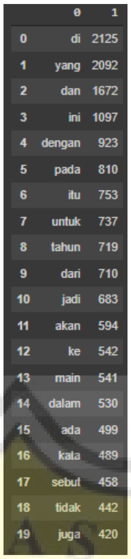


Figure 3.10 *Boxplot to Detect Outliers*

Figure 3.10 is using boxplot to detect outlier. Outlier is data that appears far from a normal distribution. It can appear in the right from normal distribution or maybe the left from the normal distribution. Irad Ben-Gal[21] introduced outliers are often considered as an error or noise, they may carry important information. Goldberg also said that an outlier is an instance that appears unreasonably far from the rest of the data [5]. There are many methods to detect outliers, in this project we only focus on a boxplot. Boxplot is a graphical display to detect outliers. It is based on the distribution quadrants Q1 and Q2. It also used standard deviation to detect where the outliers appear [21]. In this research, there are many outliers in each class. These outliers can not be very affected because there are only a small number of outliers.

3.2.4. Analysis

Preprocessing data very affected the result of the data train maybe until the classification. Natural Language Processing problems are very sensitive especially in stopwords. If a stopword did not clear anyway, it will cause serious problems in predicted data. For example in a stopwords, if we did not filter the stopwords the result of the word count would be in the picture below

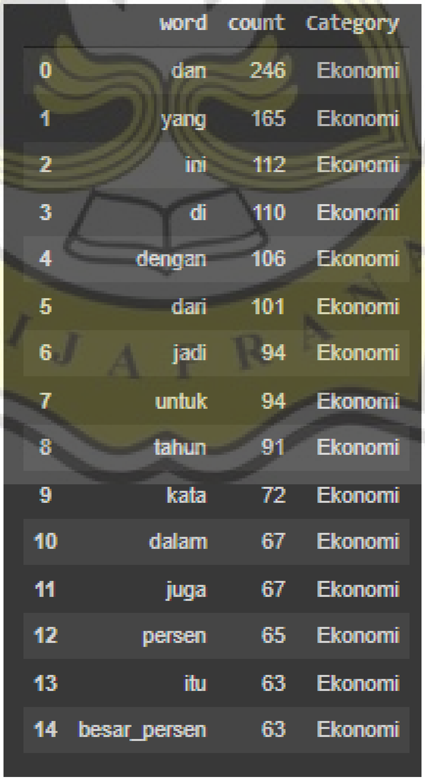


	0	1
0	di	2125
1	yang	2092
2	dan	1672
3	ini	1097
4	dengan	923
5	pada	810
6	itu	753
7	untuk	737
8	tahun	719
9	dari	710
10	jadi	683
11	akan	594
12	ke	542
13	main	541
14	dalam	530
15	ada	499
16	kata	489
17	sebut	458
18	tidak	442
19	juga	420

Figure 3.11 *Word Count without Stopwords Filtering*

In **Figure 3.11**, there are stopwords in the data that would be the dominant number. It is caused by preprocessing data that is not clean and not removing the stopwords. So, it is a simple problem that would very much affect the performance like precision and recall analysis.

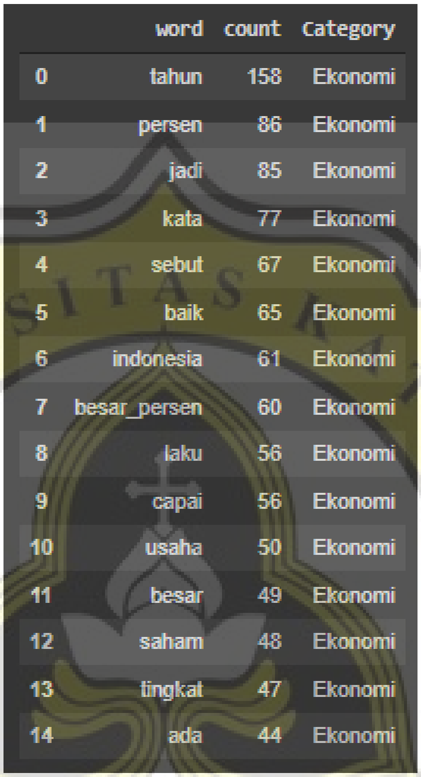
UNIVERSITAS KATOLIK SOEGIJAPRANATA



	word	count	Category
0	dan	246	Ekonomi
1	yang	165	Ekonomi
2	ini	112	Ekonomi
3	di	110	Ekonomi
4	dengan	106	Ekonomi
5	dari	101	Ekonomi
6	jadi	94	Ekonomi
7	untuk	94	Ekonomi
8	tahun	91	Ekonomi
9	kata	72	Ekonomi
10	dalam	67	Ekonomi
11	juga	67	Ekonomi
12	persen	65	Ekonomi
13	itu	63	Ekonomi
14	besar_persen	63	Ekonomi

Figure 3.12 *Ekonomi class without Stopwords Filtering*

In **Figure 3.12**, we know that there are many stopwords in that case. The most dominant word is the word “dan”. It is caused by preprocessing data without stopwords filtering. In the picture below is word count in “Ekonomi” class with stopwords filtering.



	word	count	Category
0	tahun	158	Ekonomi
1	persen	86	Ekonomi
2	jadi	85	Ekonomi
3	kata	77	Ekonomi
4	sebut	67	Ekonomi
5	baik	65	Ekonomi
6	indonesia	61	Ekonomi
7	besar_persen	60	Ekonomi
8	laku	56	Ekonomi
9	capai	56	Ekonomi
10	usaha	50	Ekonomi
11	besar	49	Ekonomi
12	saham	48	Ekonomi
13	tingkat	47	Ekonomi
14	ada	44	Ekonomi

***Figure 3.13** Ekonomi class with Stopwords Filtering*

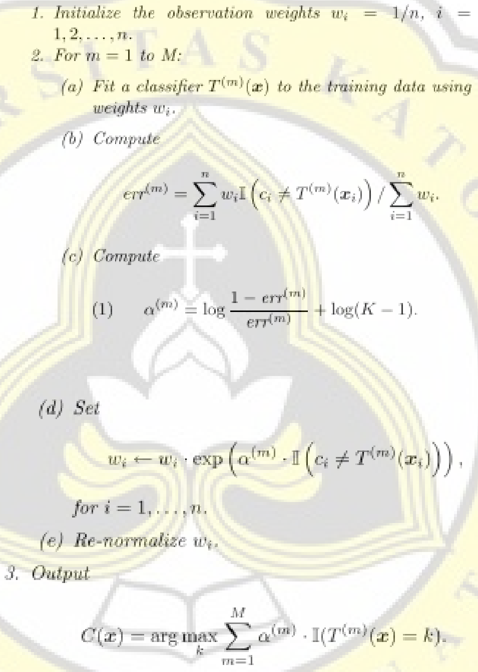
From **Figure 3.13**, we can easily guess that data above through the preprocessing data in stopwords filtering. So, the most dominant word is “tahun” without any stopwords. It can make the data clean. So, the data can be learned by machine learning well.

3.3. Algorithm

In the previous, we had explained how datasets are processed, from raw data until ready to use for training. Now, we have an algorithm that can process that dataset. Before we discuss what algorithm is used, we need to know what the algorithm is. Yanovsky said that an algorithm is a set of programs that implement functions [22]. A function comes from a mathematical formula that can solve a problem. In this subchapter, we only focus on the algorithm which is used for this classification.

3.3.1. *Adaboost Algorithm*

The first popular boosting algorithm is the 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
        (1) $\alpha^{(m)} = \log \frac{1 - err^{(m)}}{err^{(m)}} + \log(K - 1).$
    (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 3.14 Adaboost Multiclass Formula

The process of the SAMME algorithm in the **Figure 3.14** is defined into 3 steps. So, 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 by the classification problem. The weight will decrease if the classification of the dataset is wrongly predicted. The trueness of the classification process belongs to the data train. After getting the weight, now count the error rate of misclassification

UNIVERSITAS KATOLIK SOEGIJAPRANATA

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 3.15 *The Alpha Value SAMME Formula*

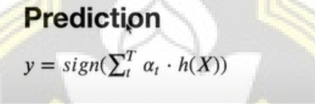
Now, we get alpha value to count the new weight by the formula:

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

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

Figure 3.16 *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 3.17 *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.

3.3.2. Supervised Learning

Referring to this project, we used supervised learning that adopted algorithms from ensemble learning. This ensemble learning used is the Adaboost algorithm (Adaptive Boosting algorithm). This research only focuses on Semi-Supervised learning, this supervised learning is only used for the comparison between them.

3.3.3. Semi-Supervised Learning

In this project, we used semi-supervised learning to classify the news based on their category. This semi-supervised learning adopted supervised learning combined with pseudo labeling. In general, pseudo labeling can be mentioned as self-training. It used both labeled data and unlabeled data. From labeled data, used for training algorithms. Then, that model was used for predicted unlabeled data. The result of predicted unlabeled data concat with the trained data before and used for the next retraining model.

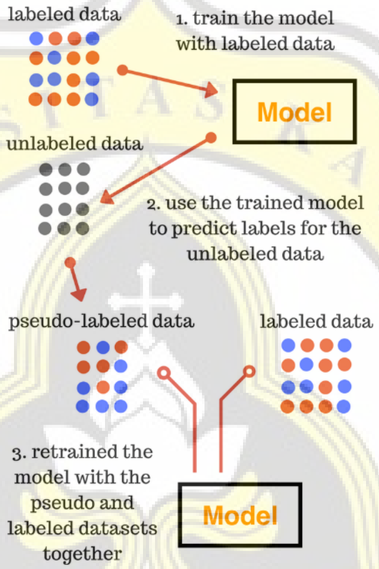

labeled data
1. train the model with labeled data
Model
unlabeled data
2. use the trained model to predict labels for the unlabeled data
pseudo-labeled data
labeled data
3. retrained the model with the pseudo and labeled datasets together
Model

Figure 3.18 *Pseudolabelling in Semi-Supervised Learning*

3.3.4. Evaluation Model

From those research methods that we have explained before, the score of the model can be calculated by the precision, recall, F1-Score, and accuracy. In each other, we differentiate them into two parts. The first is macro average and the second is micro average. Each has their analysis.