CHAPTER 3
RESEARCH METHODOLOGY

This chapter outlines the research methods from data collection to how the program is made. The general approach of each step will be given in order to make the recreation of the project more manageable.

3.1. Dataset creation

The first step to do is data gathering, as in machine learning data usually is analogized as the blood of the algorithm, and the challenge that comes with the medical dataset is that they are hard to come by. In this case, the dataset is gathered from credible journals and medical websites with an annotator to help validate and measure the accuracy of the data collected.

3.1.1. Data gathering

Websites that are credible and especially for the website, trusted and well-known. Examples of these are:

- WebMD

For WebMD, one of the main benefits that it has is a symptom checker system, so the validity of symptoms that is collected can also be validated with the symptom checker. WebMD's main source for illnesses symptoms that are used is the article "6 Oral Health: Warning Signs You Should Never Ignore", in which it gives mostly mild symptoms in the area of the mouth, afterward it also gives references towards the illnesses itself and other symptoms that can be collected.

- Alodokter

Alodokter helps to give more information towards the dataset collected, the primary reason for the inclusion is mostly to verify and to add information on symptoms and illnesses that have already been gathered.

- Google's Medical Information

The most convenient and while also reliable is Google's Medical Information given when we searched about a particular symptoms and illnesses. Google's Medical Information is sourced from the Centers for Disease Control and Prevention (CDC), National Cancer Institute (NCI), Food and Drug Administration (FDA), etc.

Not only validate and verify the illnesses' symptoms gathered through websites, medical journals and research are also used in order to add several symptoms. Such as Systemic Consequences of Poor Oral Health in Chronic Kidney Disease Patients, Eating Disorders in Adolescents and their Repercussions in Oral Health, Association between poor oral health and eating disorders: systematic review and meta-analysis, etc.

The next step is determining the diseases that will be classified and the features for classification. Throughout the research done, major illnesses that dominantly have classifiable symptoms in the mouth are Diabetes, Bulimia Oral Cancer, and Kidney Failure. Determining the features of each disease can be quite challenging as sometimes certain websites and journals do not give a thorough symptom list for illnesses, therefore the data gathered needs to be validated with the help of an annotator(s) which will be done in (3.1.2 Data Validation and Annotation). With that said there are six features (parts of the mouth) chosen to help strengthen the classification process. These are the gums (*gusi*), teeth (*gigi*), lips (*bibir*), tongue (*lidah*) throat (*tenggorokan*), and area of mouth (*area mulut* but will be stylized as *area_mulut*).

For each feature, symptoms that are gathered through research that correspond to their illnesses will be listed as common symptoms. Some parts of the mouth have more symptoms than others. The illnesses for each feature are: For gums (*gusi*): *sakit, bengkak, berdarah sulit sembuh, abses/nanah, gingivitis,* for teeth (*gigi*): *karies, sensitif setelah makan panas, maloklusi,* *sakit/nyeri, perubahan warna, lepas erosi, decayed/berlubang posisi gigi tidak pas*, for lips: (*bibir*): *gatal-gatal, sakit, membiru, kering, berdarah sulit sembuh, bengkak, bercak putih/merah*, for tongue (*lidah*): *sakit, bengkak, berdarah sulit sembuh, bercak putih/merah*, for throat (*tenggorokan*): *benjolan/pembengkakan, sakit saat menelan/dysphagia, dehidrasi*. For area of mouth (*area mulut*): *kering, bau mulut berulang, bau almond tanpa konsumsi almond, bau buah tanpa konsumsi buah, bercak merah/putih, benjolan area mulut, infeksi jamur di mulut, mati rasa, jaringan tidak berwarna, muntah-muntah*.

In order for the data to be processed, it needs to be encoded as numbers, so each particular symptom in the feature will have a distinctive number. For gums (*gusi*), teeth (*gigi*), lips (*bibir*), tongue (*lidah*), throat (*tenggorokan*), area of mouth (*area mulut*), the common symptoms and their encodings can be seen in Table 3.1-Table 3.6 respectively below

Table 3.1. Gums (*gusi*) list encoding (left: encoded as numbers, right: symptoms)

1	sakit
2	bengkak
3	berdarah sulit sembuh
4	abses/nanah
5	gingivitis

Table 3.2. Teeth (*gigi*) list encoding (left: encoded as numbers, right: symptoms)

1	karies
2	sensitif setelah makan panas
3	maloklusi
4	sakit/nyeri
5	perubahan warna
6	lepas
7	erosi
8	decayed/berlubang
9	posisi gigi tidak pas

Table 3.3. Lips (*bibir*) list encoding (left: encoded as numbers, right: symptoms)

1	gatal-gatal
2	sakit
3	membiru
4	kering
5	berdarah sulit sembuh
6	bengkak
7	bercak putih/merah

Table 3.4. Tongue (*lidah*) list encoding (left: encoded as numbers, right: symptoms)

1	sakit
2	bengkak
3	berdarah sulit sembuh
4	bercak putih/merah

Table 3.5. Throat (*tenggorokan*) list encoding (left: encoded as numbers, right: symptoms)

1	sakit
2	bengkak
3	berdarah sulit sembuh
4	bercak putih/merah

Table 3.6. Area of mouth (*area mulut*) list encoding (left: encoded as numbers, right: symptoms

1	kering
2	bau mulut berulang
3	bau almond tanpa konsumsi almond
4	bau buah tanpa konsumsi buah
5	bercak merah/putih
6	benjolan area mulut
7	infeksi jamur di mulut
8	mati rasa
9	jaringan tidak berwarna
10	muntah-muntah

And the Table 3.7 below is to give an example of how the table is created after each symptom is encoded and then given labels.

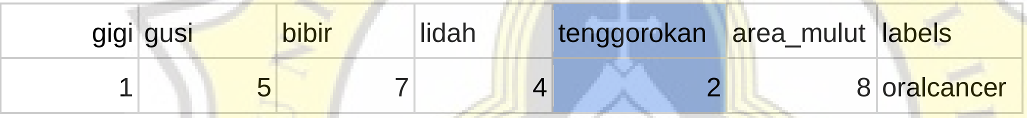
Table 3.7. Example of encoded symptoms for each feature, with labels

gigi	gusi	bibir	lidah	tenggorokan	area_mulut	labels
1	5	7	4	1	6	oralcancer
1	5	7	4	2	6	oralcancer
1	6	7	4	1	5	oralcancer
1	6	7	4	2	5	oralcancer
1	6	7	4	1	6	oralcancer
1	6	7	4	2	6	oralcancer
1	6	7	4	2	6	oralcancer

3.1.2. Data validation and annotation

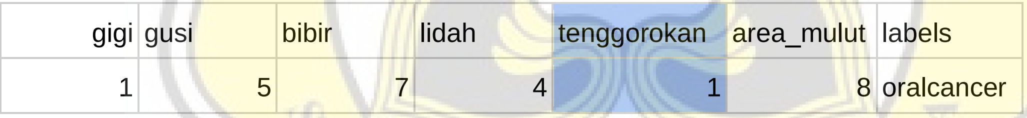
To make the dataset unbiased and more accurate, the dataset has to be annotated as the means of validation. The annotation that is used will give the dataset a comparison of each symptom in the feature and measure its accuracy. To annotate the data gathered, a willing annotator needs to create a new dataset with the same symptoms list and features (parts of the mouth) list, but are not allowed to copy the original entries. The annotator is given the original source, but also has to find their own sources so the symptoms that will be gathered are more diverse and more generalized. After that, creating a scoring based on each entry comparison is necessary to see how many entries are similar. This can be done using two new columns with the name of diff (differences) and score. Here is an example comparison of the original entry in Table 3.1 and the annotator's entry in Table 3.2.

Table 3.1. Original entry



gigi	gusi	bibir	lidah	tenggorokan	area_mulut	labels
1	5	7	4	2	8	oralcancer

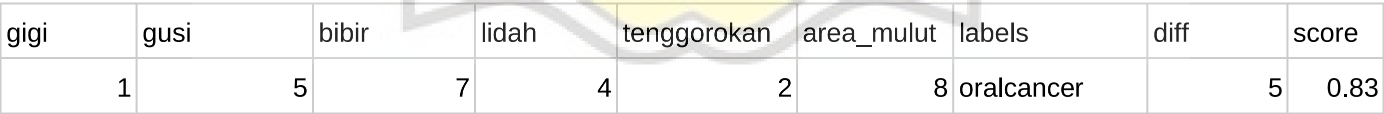
Table 3.2. Annotator's entry



gigi	gusi	bibir	lidah	tenggorokan	area_mulut	labels
1	5	7	4	1	8	oralcancer

As we can see there is a difference in the feature "tenggorokan", this needs to be taken into account. This is where the usage of differences and scoring can be done, and it can be used to accompany the original dataset as seen in Table 3.3 below.

Table 3.3. Example of the original dataset with diff and score column



gigi	gusi	bibir	lidah	tenggorokan	area_mulut	labels	diff	score
1	5	7	4	2	8	oralcancer	5	0.83

The usage of diff (differences) is used to measure how many symptoms for each feature on each entry differs from the original dataset when compared to the annotator's dataset. Because there are multiple parts of the mouth that are used for each illness in the dataset comparison, there are more possibilities of permutations for each entry. But in actuality, the majority of data has fewer differences and becomes easier to input because fewer vague and ambiguous symptoms need to be entered for each entry.

$$\text{Score} = \frac{diff}{len(v=[v_1,\dots,v_n])} \tag{1}$$

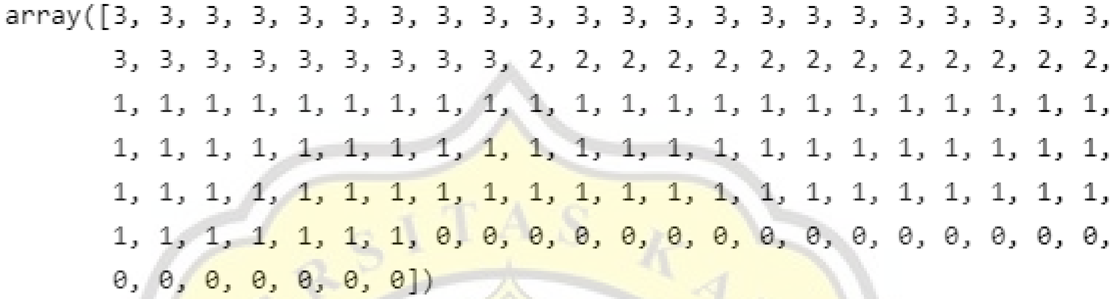
For function (1) each entry is scored to give a confidence score towards each entry ($Score$), by dividing the number of differences ($diff$) with the length (len) of non-duplicate features (parts of the mouth) ($v = [v_1, \dots, v_n]$). The score can be used to filter out entries that do not give a high amount of confidence score, therefore helping in terms of dataset accuracy and giving flexibility when experimenting with data that will be used.

3.2. Preparation and introduction for each algorithm

This subchapter will explain the general approach to algorithms that will be used for the classification and analysis. Programming will be done in the Python programming language. There are 3 algorithms used to do the prediction of illnesses, random forest, XGBoost, and TensorFlow, with XGBoost being the main algorithm used and random forest and TensorFlow are mostly used for comparison analysis, with each algorithm needing certain preprocessing.

To make the explanations more efficient, several libraries are used multiple times. Therefore, certain libraries that are explained might not be used for all the algorithms. The function of each library will be explained in a general way. Pandas is a fast, powerful, flexible, and easy to use open source data analysis and manipulation tool, built on top of the Python programming language[13]. It will be used for all three algorithms. Pandas data frame makes it easier and more flexible to separate parts of the mouth (features) and labels, check the condition of the dataset, and remove unnecessary parts for prediction and the added bonus is to make the data look more presentable. NumPy will also be used, as it gives the ability of comprehensive mathematical functions, vectorization, and indexing[14]. NumPy in this case will be used to transform the input value for prediction as an array, and also be responsible for reshaping the array. The dataset needs to be preprocessed first to fit the needs of each algorithm. Sklearn will be used for several reasons, but mainly is used for its model_selection's train_test_split in order to shorten the process of splitting the dataset into a training set and test set which will be used to evaluate the final model accuracy before deployment and also to check the accuracy metric.

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements machine learning algorithms under the Gradient Boosting framework[15]. XGBoost for this case is used for the XGBClassifier in order to classify/predict the illnesses based on the symptoms given. Because of that reason, the labels need to be encoded, the array of the encoded labels are as shown in Figure 3.1.



```
array([3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0])
```

Figure 3.1 **Array of Encoded Labels**

To encode we will need sklearn.preprocessing's "LabelEncoder". The benefit of encoding using LabelEncoder is not only to transform the labels. Transforming the labels as integers using LabelEncoder also helps the program to be more future-proof, as when there is a new illness, there is no need to manually assign illnesses new numbers.

The first comparative algorithm is the random forest (or random decision forest) algorithm, an ensemble learning method for classification, regression, and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees[16]. Random forest will be used as a comparative analysis. This algorithm is simpler but a little bit more complicated to prepare. Sklearn's ensemble for RandomForestClassifier will be used. As random forest does not need the labels to be encoded, the usage of LabelEncoder is not done for this algorithm, rather each feature and label will just be located using pandas and contained in X and y labels respectively.

The second comparative algorithm is multilayer perceptrons (MLP) using TensorFlow. MLP is a fully connected class of feedforward artificial neural network (ANN). TensorFlow, is an end-to-end open source platform for machine learning (ML). It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML, and developers easily build and deploy ML-powered applications[17]. For TensorFlow, multilayer perceptrons (MLP) will be used.. Encoding will be used for TensorFlow, therefore LabelEncoder will be of use. The requirement for features to be able to be inputted for training is that the features need to be the float32 data type.