

CHAPTER 3

RESEARCH METHODOLOGY

3.1. Overview

In the last chapter, we already gained some knowledge about previous methods and algorithms that researchers use. The differences between this research and previous research are also explained in the last chapter. After gaining some knowledge in the previous chapter, the research methodology that will be used in this research will be discussed here. The research methodologies that will be done in this research can be look in Figure 3.1 below.

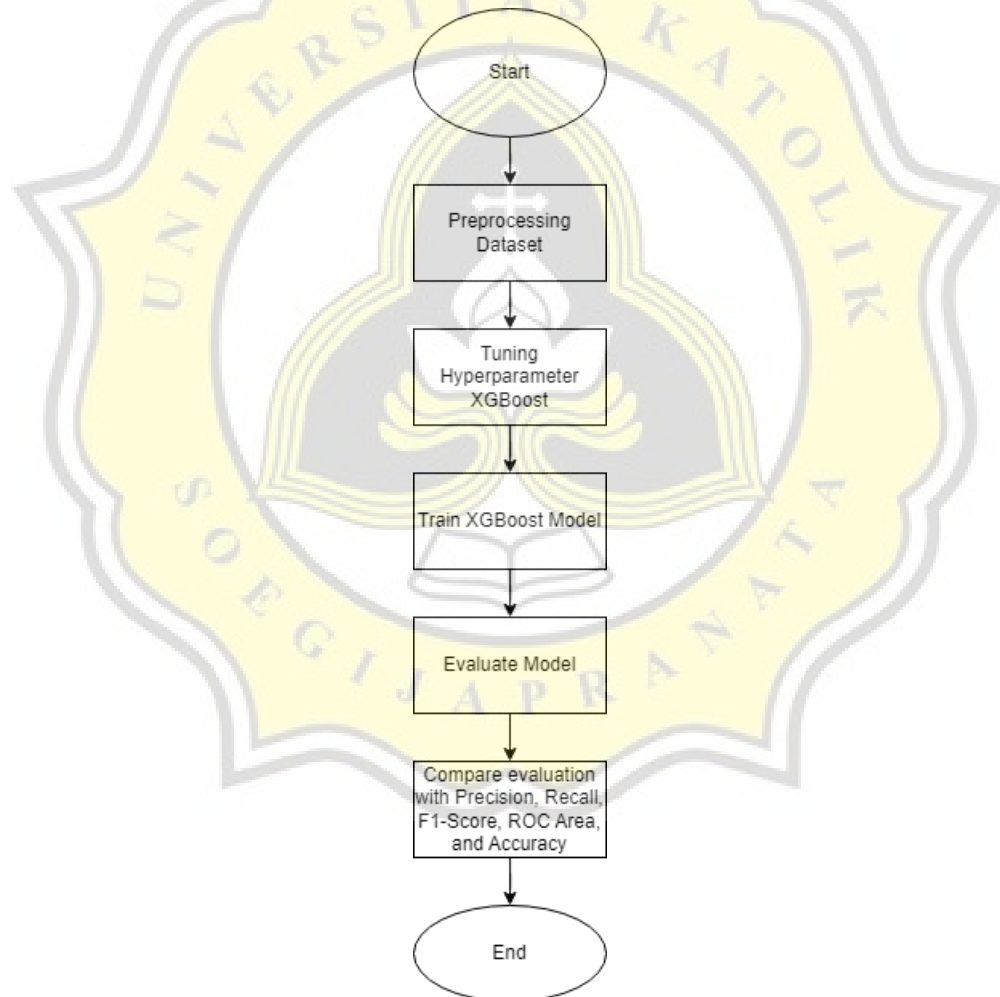


Figure 3.1 Research Methodology

3.2. Data Collection

This research will be using the heart disease dataset obtained from UCI Machine Learning Repository. The dataset can be obtained freely on the UCI Machine Learning Repository's website [14]. The original data contains 76 attributes, including the predicted or output attribute. But, all published research and experiment about heart disease using machine learning only refer to only 14 attributes [14] that are strongly related to heart disease [2]. The 14 attributes that are included can be seen in Table 3.1. The target attributes with value 1, 2, 3, and 4 that has meaning as the presence of heart disease will be all converted to value 1 which represent the presence of heart disease (True) and 0 has meaning as the absence of heart disease (False) [2]. The dataset contains 303 data rows with 164 absence of heart disease (labeled with value 0) and 139 presence of heart disease (labeled with value 1).

Table 3.1. Data attributes and description

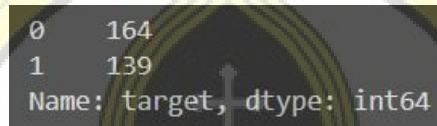
Attributes	Description
age	Age in years
sex	Male = 1 and Female = 0
cp	Chest pain type (4 values (1, 2, 3, 4))
trestbps	Resting blood pressure (in mm Hg on admission to the hospital)
chol	Serum cholestoral in mg/dl
fbs	Fasting blood sugar > 120mg / dl (1 = True and 0 = False)
restecg	Resting electrocardiographic results (3 values (0, 1, 2))
thalach	Maximum heart rate achieved
exang	Exercise induced angina (1 = Yes and 0 = No)
oldpeak	ST depression induced by exercise relative to rest
slope	The slope of the peak exercise ST segment (3 values (1, 2, 3))
ca	Number of major vessels (0-3) colored by flourosopy
thal	Thalium stress test result(3 values (3, 6 ,7))
target	Diagnosis of heart disease (5 values (0, 1, 2, 3, 4))

3.3. Data Portioning

Due to the Cleveland heart disease dataset being limited to 303 data, the 10-Fold Cross Validation will be used for data portioning. This method was chosen primarily because it had less variation than other estimators like the single hold-out approach [2].

3.4. Data Preprocessing

Before making and training the algorithm model, data need to be pre-processed. In this process, the target attribute values 1, 2, 3, and 4 will be converted to only True (1) representing the presence of heart disease. While the target attribute value 0 will remain 0 as it is False which means the absence of heart disease [2]. The output of the converted can be seen in Figure 3.2 below.



```
0    164
1    139
Name: target, dtype: int64
```

Figure 3.2 Preprocessed Target Attribute

Missing values that are contained in the Cleveland heart disease dataset will be handled by: (1) Dropping missing data rows [6], (2) Replacing them with mode, (3) Replacing them with median. Each of these missing value techniques will be analyzed to find which has a better result on the model. Outliers will also be removed as it can affect the distribution of the data using IQR (Interquartile Range) method.

3.5. Bayesian Optimization

One of the most popular automated hyperparameter tuning techniques to reach the global optimum with fewer steps is Bayesian Optimization [8]. Bayesian Optimization may analyze previous values to identify the best combination of hyperparameters, and executing the surrogate model is typically significantly less expensive than running the objective function as a whole [8]. Bayesian Optimization algorithm have higher stability, quicker calculations, and has the least time consuming than Grid Search and Random Search [10], [11]. Bayesian Optimization will be used in this research to find the optimum hyperparameter for the XGBoost algorithm.

3.6. Extreme Gradient Boosting Algorithm

The Extreme Gradient Boosting Algorithm or usually known as XGBoost Algorithm is a popular algorithm that has been frequently used in machine learning and data mining problems [13]. Real world sized issues may be resolved using XGBoost with the least amount of resources [13]. The XGBoost method is a boosting algorithm that creates trees consecutively to minimize mistakes from the previous trees.

XGBoost hyperparameters will be tuned to optimize its performance [7]. Comparatively speaking, a well tuned XGBoost can attain state-of-the-art prediction accuracy [11]. Bayesian Optimization will be used to tune the XGBoost hyperparameters. Several hyperparameters will be tuned to control overfitting¹. Learning_rate is an important hyperparameter that can help the model prevent overfitting by reducing the learning_rate to reduce the step size. N_estimators is a hyperparameter about how many times the tree will be boosted. Small n_estimators values can lead to a underfit of a model, and large n_estimators values can lead to a overfit of a model. So choosing optimal n_estimators for a model is important.

There are also max_depth, min_child_weight, and gamma that will control the model complexity. Max_depth will control how many leaves the tree will be made in a boosting round. Min_child_weight will determine whether a leaf will be pruned or not based on the minimum values that are set. The gamma hyperparameter will determine if a branch of a tree will be pruned or not. Subsample and colsample_bytree will also be tuned. Subsample will randomly choose samples based on a given ratio to make a tree. Colsample_bytree will randomly choose subsample columns on a given ratio to make a tree. Subsample and colsample_bytree will add randomness to the data to make a model that is trained to be robust to noise. The search space of learning_rate, n_estimators, max_depth, min_child_weight, gamma, subsample, and colsample_bytree can be seen in **Error! Reference source not found.** below [7], [9].

¹ “Control Overfitting” https://xgboost.readthedocs.io/en/latest/tutorials/param_tuning.html#control-overfitting (accessed on October 26 2022, 20.32 o'clock).

Table 3.2. Hyperparameter Search Space

Hyperparameter	Search Space		
	<i>Lower Bounds</i>	<i>Upper Bounds</i>	<i>Optimal Result</i>
learning_rate	0.1	1	0.1
n_estimators	100	250	220
max_depth	1	15	1
min_child_weight	0	1	0.881
gamma	0	1	0.815
subsample	0.4	1	0.842
colsample_bytree	0.4	1	0.721

3.7. Feature Selection

Feature selection is a significant step in constructing a classification model as it can increase the accuracy and reliability of the algorithm further [3], [4]. The Chi Square, Mutual Information, ANOVA, Forward Feature Selection, Backward Feature Selection, Recursive Feature Selection, and Feature Importance will be used to determine which feature is more or less important.

3.8. Model Evaluation

The model will be evaluated using precision, recall, f1-score, roc area, and accuracy. The result of the evaluation will be then compared with the AdaBoost algorithm that are used in previous research [12]. The XGBoost model result that are using feature selected dataset will also be compared with the XGBoost model without feature selected dataset to gain knowledge of whether feature selection in heart disease prediction is more effective or not.