JUSIKOM PRIMA (Jurnal Sistem Informasi dan Ilmu Komputer Prima)
Vol. 7 No. 1, August 2023 E-ISSN : 2580-2879

LIVER DISEASE CLASSIFICATION ANALYSIS USING THE XGBOOST METHOD

Yadi Sitinjak*[*1], Muhaymin[2], Marlince Nababan[3]
[1,2,3]Indonesian Prima University
Jl. Cover No. 3, Sei Putih Bar., Kec. Medan Petisah, Medan City, North Sumatra 20118
Email : Yadivanzraso.s@gmail.com*

ABSTRACT-Liver disease is a severe pathological condition that can cause liver inflammation due to viral infection, toxic agents, or bacterial invasion, interfering with normal liver function. The death rate from this disease reaches 1.2 million people annually in Southeast Asia and Africa. Liver disease can cause damage to the liver and negatively affect overall body function. To reduce disease progression, it is critical to facilitate early diagnosis, thereby enabling rapid initiation of treatment for affected individuals. Classification methods are widely used to make decisions based on new information from previous data processing through calculation algorithms. This study uses the XGBoost classification method to build a predictive model for liver disease. The results of this study confirm that the XGBoost model is a robust and efficient choice for liver disease classification based on patient data. The use of the XGBoost approach has proven its success in the category of liver disease with an accuracy of up to 95% and an accuracy balance of 95%, demonstrating the effectiveness and efficiency of this method in overcoming class imbalances in liver disease classification data.

Keywords: Xgboost, Liver, Classification, Disease

1. INTRODUCTION

Liver disease is a significant pathological condition affecting the liver, characterized by liver inflammation due to viral infection, exposure to toxic agents, or bacterial invasion, thus interfering with normal liver function [1][2]. According to data published by the World Health Organization (WHO), it has been reported that around 1.2 million people annually in the Southeast Asia and Africa region die from this particular disease [3]. Liver disease can cause damage to the liver, causing a gradual decline in the body's ability to function correctly.

Liver disease can be caused by various factors, such as congenital liver defects, viral or bacterial infections, alcohol addiction, smoking, unhealthy lifestyle choices, and other similar activities [4][5]. One of the main functions affected is the liver's ability to neutralize toxins that enter the body. Left untreated, it can harm the body [6]. To reduce disease progression, it is essential to facilitate early diagnosis, thereby enabling rapid initiation of treatment for individuals with liver disease [7].

Classification methods are widely used to make decisions based on new information from processing previous data through algorithm calculations [8]. Xgboost is the method used to build a Liver disease classification model in this study. XGBoost (Extreme Gradient Boosting) is a viral and effective machine learning algorithm for tasks such as classification and regression.

One of the main advantages of XGBoost is its superior performance, which often outperforms many other machine learning methods in competition and prediction tasks [9]. In addition, XGBoost is optimized for efficiency and scalability to work well on large data sets with many features [10]. XGBoost can also handle non-linear relationships between components and targets and has various options to address class imbalance issues in classifications.

To provide references and comparative material, the researcher conducted a literature study that included several previous studies related to the methods used to predict inflammatory liver

disease by [11][12]; in both studies, the Decision Tree method was used (C4.5) in its classification. Produces an accuracy of 70.29% and an AUC value of 0.714, Naïve Bayes produces an accuracy of 67.05% and an AUC value of 0.757, while SVM obtains an accuracy of 78%.

Based on this background presentation, developing a method for classifying liver disease is necessary. Hence, the researchers conducted a study entitled "Analysis of Liver Disease Classification Using the Xgboost Method."

2. RESEARCH METHODS

2.1 Research Methods

The liver is a vital organ in the human body; it is responsible for converting harmful compounds into essential nutrients, which are then used by the body to regulate hormone levels within its physiological framework [13]. The liver also plays a vital role in synthesizing hormones and proteins, regulating blood glucose levels, and facilitating hemostasis [14].

Classification is a fundamental machine learning procedure involving developing models capable of distinguishing between classes. This model is built by identifying certain traits or characteristics that distinguish one type from another [15]. The classification process involves making a model using pre-existing training data, which is then used to classify the newly acquired data. Classification can be defined as training or learning a model on the target function, which maps a set of properties (also known as features) to a group of available class labels [16].

XGBoost, also known as Extreme Gradient Boosting, is a machine learning technique that integrates the principles of gradient boosting and boosting. Classification models built using boosting techniques involve making predictions based on previous model errors to produce the following model [17]. The algorithm used in this context is called gradient enhancement, which uses gradient descent techniques to reduce mistakes while constructing new models effectively.

XGBoost is a machine learning algorithm that belongs to the ensemble tree family, specifically using classification and regression trees (CART) [18]. Improvement methodologies require iterative procedures aimed at improving classification performance. On the other hand, the ensemble tree approach combines multiple decision trees to produce superior performance compared to a single decision tree.

2.2 Research Workflow

In conducting research, a workflow diagram is determined so that it runs well and can be completed on time. The research flowchart that we can see in Figure 1 below

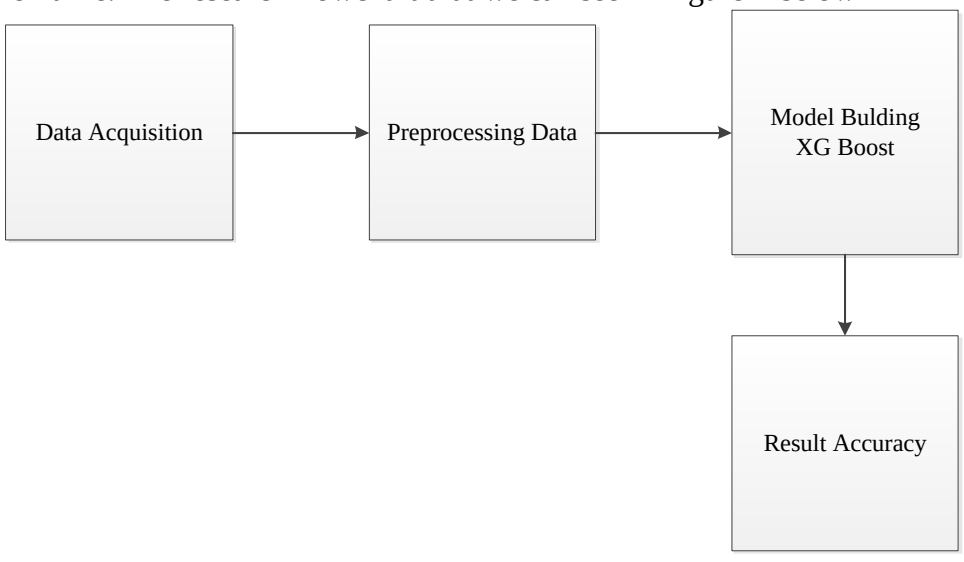

Figure 1 Research Diagram

1. Data Acquisition

Data acquisition refers to a systematic procedure of collecting data from various sources to carry out further analysis or processing activities. The process involves acquiring data from multiple sources, including databases, sensors, hardware, websites, and other relevant data sources.

2. Data Preprocessing

Data pre-processing refers to sequential procedures or methodologies that are executed before using data for further analysis or processing. The main goal of data pre-processing is to effectively clean, organize, and filter data to align it with the specific requirements of the following analysis or processing task.

3. Model Building Xgboost

XGBoost model construction involves developing predictive or classifying models with the XGBoost algorithm. XGBoost, or Extreme Gradient Boosting, is a mighty and efficient machine learning algorithm for various tasks, including classification, regression, and ranking.

4. Accuracy Result

The concept of result accuracy, also known as prediction accuracy or classification accuracy, refers to evaluating the extent to which a constructed prediction or classification model gives accurate results. Accuracy evaluation metrics are often used to assess a model's ability to predict or classify data accurately.

3. RESULTS AND DISCUSSION

3.1 Data Acquisition

Based on data obtained from the Kaggle public data provider platform[19], which is a dataset of liver disease patients, which has a total data of 483 rows and 11 columns, each column containing Age, Gender, Total_Bilirubin, Direct_Bilirubin, Alkaline_Phosphotase, Alamine_ Aminotransferase, Aspartate _Aminotransferase, Total_Protiens, Albumin, Albumin_ and_Globulin_ Ratio, Liver_Disease. The stages of data acquisition can be seen in Figure 2:

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	Liver_Disease
0	65	Female	0.7	0.1	187	16	18	6.8	3.3	0.90	1
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	1
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	1
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	1.00	1
4	72	Male	3.9	2.0	195	27	59	7.3	2.4	0.40	1
...	...	...	...	...	...	...	...	...	...	...	...
478	60	Male	0.5	0.1	500	20	34	5.9	1.6	0.37	0
479	40	Male	0.6	0.1	98	35	31	6.0	3.2	1.10	1
480	52	Male	0.8	0.2	245	48	49	6.4	3.2	1.00	1
481	31	Male	1.3	0.5	184	29	32	6.8	3.4	1.00	1
482	38	Male	1.0	0.3	216	21	24	7.3	4.4	1.50	0

483 rows × 11 columns

Figure 2 Data Acquisition

3.2 Data Processing Flow

For the research to run well, the researcher makes a data processing flow that aims to keep the study running based on the predetermined flow. The flow of processing this data is in Figure 3 below:

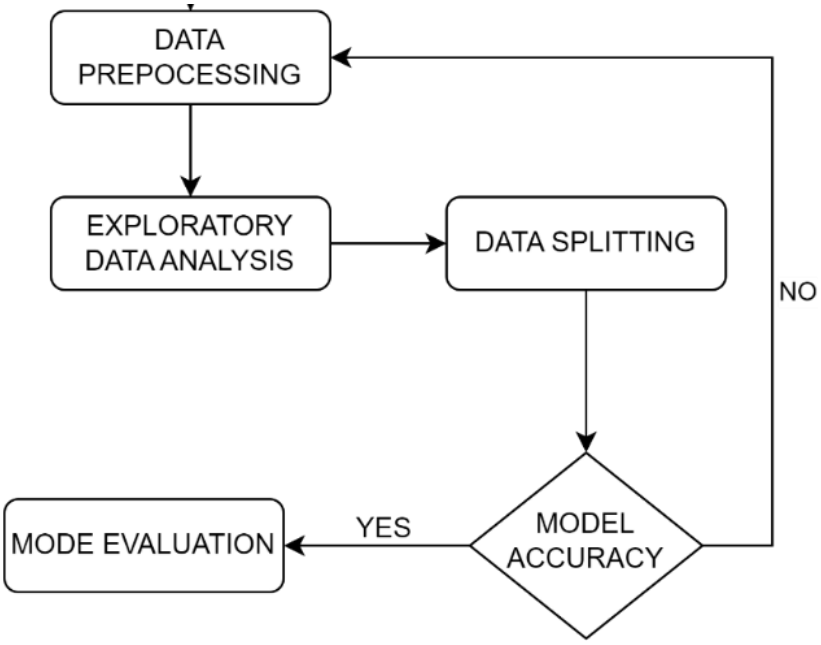

Figure 3 Data Processing Flowchart

3.3 Data preprocessing

The data obtained then enters the preprocessing stage, where the dataset related to the construction and training of machine learning models is prepared. At this stage, it also aims to improve the dataset's quality. The steps are carried out as follows:

3.3.1 Data Manipulation

At the data manipulation stage, the aim is to change the English language contained in each dataset column to Indonesian so that it can more easily understand the contents of the owned dataset. The process can be seen in Figure 4 below:

```
df.rename(columns={"Age" : "Usia",
                   "Gender" : "JenisKelamin",
                   "Total_Bilirubin" : "JumlahBilirubin(mg/dL)",
                   "Direct_Bilirubin" : "BilirubinDirek(mg/dL)",
                   "Alkaline_Phosphotase" : "AlkaliFosfatase(IU/L)",
                   "Alamine_Aminotransferase" : "AlanineAminotransferase(IU/L)",
                   "Aspartate_Aminotransferase" : "AspartateAminotransferase(U/L)",
                   "Total_Protiens " : "JumlahProtein", "Albumin " : "Albumin",
                   "Albumin_and_Globulin_Ratio " : "RasioAlbumin&Globulin",
                   "Liver_Disease" : "PenyakitLiver"}, inplace=True)
df.head()
```

Usia	JenisKelamin	JumlahBilirubin(mg/dL)	BilirubinDirek(mg/dL)	AlkaliFosfatase(IU/L)	AlanineAminotransferase(IU/L)	AspartateAminotransferase(U/L)	JumlahProtein	Albumin	RasioAlbumin&Globulin
65	Female	0.7	0.1	187	16	18	6.8	3.3	0.90
62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74
62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89
58	Male	1.0	0.4	182	14	20	6.8	3.4	1.00
72	Male	3.9	2.0	195	27	59	7.3	2.4	0.40

Figure 4 Data Manipulation

3.3.2 Missing Values Handling

This stage checks for missing data in the dataset, and if the data is missing, then the missing data will be patched with the mean, median, and max values. In this study, the missing data in the dataset was restored using the mean value. The stages of the disappeared values handling process can be seen in Figure 5 below:

```
# Melihat data yang hilang
df.isnull().sum()
```

```
Usia                                0
JenisKelamin                        0
JumlahBilirubin(mg/dL)              0
BilirubinDirek(mg/dL)               0
AlkaliFosfatase(IU/L)               0
AlanineAminotransferase(IU/L)       0
AspartateAminotransferase(U/L)      0
Total_Protiens                      0
Albumin                             0
Albumin_and_Globulin_Ratio          3
PenyakitLiver                       0
dtype: int64
```

```
# MENGISI DATA KOSONG DENGAN NILAI RATA RATA (MEAN)

df.fillna(df.mean(), inplace=True)
```

Figure 5 Missing Value Handling

3.3.3 Duplicate Removal

At this stage, an elimination process is carried out on data with duplicate values, aiming to remove the exact data. The actual removal stages can be seen in Figure 6 below:

```
df.duplicated()
```

```
0      False
1      False
2      False
3      False
4      False
       ...
478    False
479    False
480    False
481    False
482    False
Length: 483, dtype: bool
```

Figure 6 Duplicate Removal

3.3.4 Feature Engineering

At the feature engineering stage or feature engineering, the features in the dataset are engineered into two different classes, where the classes are 0 and 1, taken from the liver disease column, meaning 0 is a class that is not and 1 is a class yes. The engineering feature stages can be seen in Figure 7 below:

```
# MELIHAT JUMLAH KELAS 0 DAN 1
print(df['PenyakitLiver'].value_counts())
cls_0=df[df['PenyakitLiver']==0]
cls_1=df[df['PenyakitLiver']==1]
```

```
1    339
0    133
Name: PenyakitLiver, dtype: int64
```

Figure 7 Future Engineering

From the results obtained, see class 0 and class 1 that there is an imbalance between classes which is often called Imbalance data, where data with class 1 has more value of 399 and for class 0, and there are as many as 133. In both types, the imbalance data stage that is carried out is to resample the data to 500 for each class. Data resampling can be seen in Figure 8 below:

```
# MENAMBAH JUMLAH DATASET UNTUK MEMPERKUAT MODEL
cls_0=cls_0.sample(500,replace=True)
cls_1=cls_1.sample(500,replace=True)
df=pd.concat([cls_0,cls_1],axis=0)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 396 to 216
Data columns (total 11 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Usia                            1000 non-null   int64
 1   JenisKelamin                    1000 non-null   object
 2   JumlahBilirubin(mg/dL)          1000 non-null   float64
 3   BilirubinDirek(mg/dL)           1000 non-null   float64
 4   AlkaliFosfatase(IU/L)           1000 non-null   int64
 5   AlanineAminotransferase(IU/L)   1000 non-null   int64
 6   AspartateAminotransferase(U/L)  1000 non-null   int64
 7   JumlahProtein                   1000 non-null   float64
 8   Albumin                         1000 non-null   float64
 9   RasioAlbumin&Globulin           987 non-null    float64
 10  PenyakitLiver                   1000 non-null   int64
dtypes: float64(5), int64(5), object(1)
memory usage: 93.8+ KB
```

Figure 8. Data Resampling

3.3.5 Unused Column Removal

The final step in the data preparation process is the deletion of columns that are not used in the classification model to be implemented. The column that needs to be removed from the dataset includes Gender. The following section describes the sequential steps involved in deleting unused columns.

```
# TERDAPAT KOLOM YANG TIDAK MEMPENGARUHI KLASIFIKASI SEPERTI KOLOM JenisKelamin
df.drop(['JenisKelamin'], axis=1, inplace=True)
df.head(10000)
```

	Usia	JumlahBilirubin(mg/dL)	BilirubinDirek(mg/dL)	AlkaliFosfatase(IU/L)	AlanineAminotransferase(IU/L)	AspartateAminotransferase(U/L)	JumlahProtein	Albumin	RasioAlbumin&Globulin	PenyakitLiver
239	58	0.8	0.2	180	32	25	8.2	4.4	1.100000	0
440	42	0.8	0.2	114	21	23	7.0	3.0	0.700000	0
388	42	0.7	0.2	197	64	33	5.8	2.4	0.700000	0
161	22	2.7	1.0	160	82	127	5.5	3.1	1.200000	0
251	27	1.3	0.6	106	25	54	8.5	4.8	0.981317	0
...	...	...	...	...	...	...	...	...	...	...
219	58	0.8	0.2	123	56	48	6.0	3.0	1.000000	1
194	38	1.7	0.7	859	89	48	6.0	3.0	1.000000	1
200	33	1.5	7.0	505	205	140	7.5	3.9	1.000000	1
316	46	0.8	0.2	160	31	40	7.3	3.8	1.100000	1
146	26	0.6	0.2	142	12	32	5.7	2.4	0.750000	1

1000 rows × 10 columns

Figure 9 Remove Unused Columns

3.4 Exploratory Data Analysis

This study involved a series of exploratory data analysis (EDA) stages, which involved examining the distribution of liver disease cases based on age, Gender, Total Bilirubin (mg/dL), Direct Bilirubin (mg/dL), Alkali Phosphatase (IU/L), Alanine Aminotransferase (IU/L), Aspartate Aminotransferase (U/L), Total Proteins, Albumin, Albumin and Globulin Ratio, and Liver Disease. Figure 10 provides an overview of these stages.

```
sns.heatmap(df.corr(),annot=True, cmap='Greens', linewidths=0.1)
fig=plt.gcf()
fig.set_size_inches(10,10)
plt.show()
```

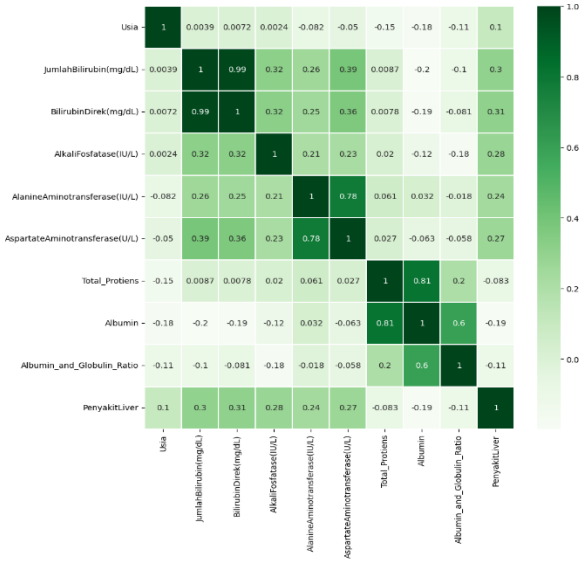

Figure 10 Exploratory Data Analysis

3.5 Data Splitting

Data splitting refers to dividing the dataset into two subsets, namely training data and testing data. The goal of data splitting is to facilitate the training of machine learning models by utilizing predefined training data sets and evaluating the performance of trained models using discrete test data sets.

3.5.1 Making the Determinant Variable (X)

This stage aims to initialize variable X which functions as an independent variable in forming a machine-learning model. This X variable includes the dataset's relevant columns, totaling 9 columns. These stages can be seen in Figure 11 below:

```
# MEMBUAT VARIABEL PENENTU (X)

X = df.drop('PenyakitLiver', axis=1)
X.head()
```

Usia	JumlahBilirubin(mg/dL)	BilirubinDirek(mg/dL)	AlkaliFosfatase(IU/L)	AlanineAminotransferase(IU/L)	AspartateAminotransferase(U/L)	Total_Protiens	Albumin
36	0.8	0.2	158	20	39	6.0	2.2
48	0.9	0.2	175	24	54	5.5	2.7
55	0.8	0.2	225	14	23	6.1	3.3
47	0.8	0.2	236	10	13	6.7	2.9
65	0.7	0.2	406	24	45	7.2	3.5

Figure 3.11 Making Variable X

3.5.2 Creating Variables (Y)

This stage also functions to create a target variable with a variable name (Y), where this variable serves as a target model in determining accuracy results. This determinant variable is used for training and testing with the "Liver Disease" column as the target variable. These stages can be seen in Figure 12 below:

```
# MEMBUAT VARIABEL TARGET (Y)

Y = df['PenyakitLiver']
Y.head()
```

```
94     0
44     0
90     0
347    0
439    0
Name: PenyakitLiver, dtype: int64
```

Figure 12 Creating Variable Y

3.5.3 Distribution of Train and Test Data

The purpose of this process is to divide the dataset into two parts. Where in this study, the data was split into 80% for training and 20% for testing, using random state or randomizing the data 101 times. These stages can be seen in the following figure:

```
# MEMBUAT FUNGSI PARTITIONING/SPLITTING

X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size=0.2,
                                                    random_state=101, shuffle=True)
```

Figure 13 Distribution of Train and Test

3.6 Model Accuracy

The Xgboost Classifier algorithm is applied as a method for creating a classification model to identify the causative factors of liver disease to achieve optimal accuracy.

3.6.1 Xgboost Accuracy

This study uses Xgboost with default settings with the library used by the Xgboost Classifier.

```
# IMPORT LIBRARY ALGORITMA
from xgboost import XGBClassifier
```

```
xgb_clf1 = xgb.XGBClassifier(objective='multi:softmax',
                             num_class=3,
                             missing=1,
                             early_stopping_rounds=10,
                             eval_metric=['merror','mlogloss'],
                             seed=42)

xgb_clf1.fit(X_train, Y_train,
             verbose=0, # set to 1 to see xgb training round intermediate results
             eval_set=[(X_train, Y_train), (X_test, Y_test)])
```

Figure 14 Import Library Xgboost

Based on the modeling that has been made, the accuracy of the Xgboost algorithm is 95%. The results of the accuracy of the Xgboost algorithm can be seen in Figure 15 below:

```
Accuracy: 0.95
Balanced Accuracy: 0.95

Micro Precision: 0.95
Micro Recall: 0.95
Micro F1-score: 0.95

Macro Precision: 0.95
Macro Recall: 0.95
Macro F1-score: 0.95

Weighted Precision: 0.95
Weighted Recall: 0.95
Weighted F1-score: 0.95
```

Figure 15 Xgboost Accuracy

3.7 Model Evaluation

The results of the confusion matrix stage of the Xgboost algorithm can be seen as true positives of 97, false positives of 2 with a total of 99 tests, true negatives of 93, and false negatives of 8 with a total of 101 tests. These stages can be seen in Figure 16 below:

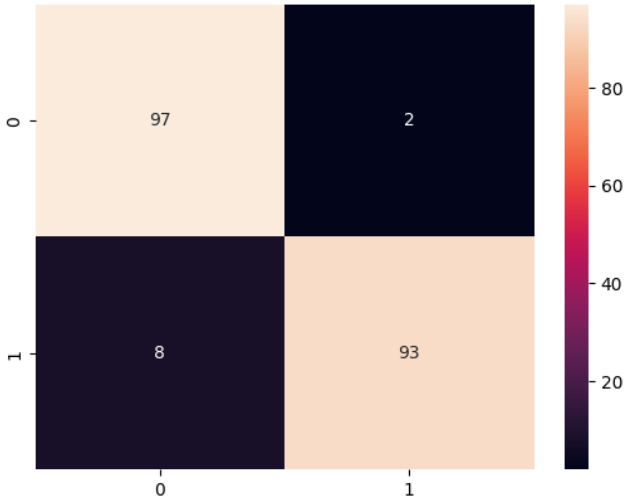

Figure 16 Confusion Matrix Xgboost

CONCLUSION

The XGBoost approach has demonstrated successful applicability in the classification of liver disease, resulting in the development of predictive models with excellent performance. The XGBoost model exhibits a high level of accuracy, as evidenced by findings of 95% accuracy and 95% accuracy balance. In addition, the model demonstrates a laudable ability to address class imbalances in data sets effectively. The evidence above suggests that using available patient data, the XGBoost technique is a robust and efficient option for liver disease classification.

BIBLIOGRAPHY

[1] APSampurna, IG Santi Astawa, NA Sanjaya Er, AAI Ngurah Eka Karyawati, IW Santiyasa, And IM Widiartha, "Attribute Selection in the Diagnosis of Liver Disease Using a Decision Tree with Genetic Algorithms," Jeliku (Electron Journal. Computer Science. Udayana), Vol. 11, no. 2, P. 329, 2022, Doi: 10.24843/Jlk.2022.V11.I02.P12.

[2] MRFRizki, "Comparison of Classification Algorithms for Liver Disease Prediction," Reputation J. Software Engineering, Vol. 1, No. 2, pp. 82–88, 2020, Doi: 10.31294/Reputasi.V1i2.109.

[3] APAyudhitama And U. Pujianto, "Analysis of 4 Algorithms in Liver Classification Using Rapidminer," J. Inform. Polynema, Vol. 6, No. 2, pp. 1–9, 2020, Doi: 10.33795/Jip.V6i2.274.

[4] AILubis, U. Erdiansyah, And R. Siregar, "Comparison of Accuracy in Naive Bayes and Random Forest in Liver Disease Classification," J. Comput. Eng. Syst. Sci., Vol. 7, No. 1, pp. 81–89, 2022.

[5] FLDCahyanti, F. Sarasati, W. Astuti, And E. Firasari, "Classification of Data Mining Using Machine Larning Algorithms for Liver Disease Prediction," Technol. J.Ilm., Vol. 14, No. 2, P. 134, 2023, Doi: 10.31602/Tji.V14i2.10093.

[6] HEMedha And DP Hapsari, “Implementing Fuzzy Decision Tree For Predicting Liver Disease Of Ilpd (Indian Liver Patient Dataset),” Vol. 5, No. 2, 2019.

[7] E.Nurlelah And MS Mardiyanto, "Selection of Attributes in the C4.5 Algorithm Using Particle Swarm Optimization to Increase the Prediction Accuracy of Liver Disease Diagnosis," J. Pilar Nusa Mandiri, Vol. 15, No. 2, pp. 195–202, 2019, Doi: 10.33480/Pilar.V15i2.706.

[8] S.Zulaikhah Hariyanti Rukmana, A. Aziz, And W. Harianto, "Optimization of the K-Nearest Neighbor (Knn) Algorithm With Normalization and Feature Selection for Liver Disease Classification," Jati (Jurnal Mhs. Tek. Inform., Vol. 6, No. 2, Pp. 439–445, 2022, Doi:

10.36040/Jati.V6i2.4722.

[9] R.Ubaidillah, M. Muliadi, DT Nugrahadi, MR Faisal, And R. Herteno, "Xgboost Implementation on Liver Patient Dataset Balance with Smote and Hyperparameter Tuning Bayesian Search," J. Media Inform. Budidarma, Vol. 6, No. 3, P. 1723, 2022, Doi: 10.30865/Mib.V6i3.4146.

[10] S.Delima, "Classification Of Electrocardiogram (Ecg) Waves Of Heart Disease Using The Xgboost Method," Vol. 10, No. 2, pp. 891–904, 2022.

[11] NT Rahman, "AnalysisDecision Tree Algorithm and Naïve Bayes in Liver Disease Patients," J. Fasilkom, Vol. 10, No. 2, pp. 144–151, 2020, Doi: 10.37859/Jf.V10i2.2087.

[12] N.Musyaffa And B. Rifai, "Support Vector Machine Model Based on Particle Swarm Optimization for Liver Disease Prediction," Jitk (Journal of Science and Computer Technology), Vol. 3, No. 2, pp. 189–194, 2018, Doi: https://Doi.Org/10.33480/Jitk.V3i2.

[13] R.Annisa, "Comparative Analysis of Data Mining Classification Algorithms for Predicting Heart Disease Sufferers," J. Tek. inform. Kaputama, Vol. 3, No. 1, pp. 22–28, 2019, [Online]. Available: Https://Jurnal. Kaputama.Ac.Id /Index.Php/Jtik/Article/View/141/156