IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

The impact of Data Mining Classification techniques on the diagnosis of Liver Disease

SHYNI A L
Research Scholar, Department of Computer Science, Noorul Islam Centre for Higher Education, Kumaracoil, Tamil Nadu

Dr.J.R JEBA
Associative Professor and Head, Department of Computer Applications, Noorul Islam Centre for Higher Education, Kumaracoil, Tamil Nadu

Abstract
Liver disease is a major public health problem, affecting millions of people worldwide. Early diagnosis and treatment are essential for improving outcomes for patients with liver disease. However, liver disease can be difficult to diagnose, as it often does not cause any symptoms until it is in the advanced stages. Data mining classification techniques have the potential to revolutionize the diagnosis of liver disease. By analyzing large datasets of patient data, these techniques can identify patterns that can be used to predict the presence of liver disease, even in patients who do not have any symptoms. This can lead to earlier diagnosis and treatment, which can improve outcomes for patients. In this paper, we review the literature on the use of data mining classification techniques for the diagnosis of liver disease. We discuss the different techniques that have been used, the results of these studies, and the limitations of the current research. We also discuss the potential impact of data mining classification techniques on the diagnosis of liver disease and the future directions of this research.

Keywords: Data Mining, Classification Techniques, SVM, Naïve Bayes, Decision Tree, KNN, Random Forest, Evaluation metrics,

Introduction

Data mining is a powerful tool that can be used to extract hidden patterns and relationships from large datasets. This makes it a valuable tool for early diagnosis of liver disease. In recent years, there has been a growing interest in using data mining to predict liver disease.
Data mining and classification techniques can be used to extract knowledge from large datasets. They can be used to solve a wide variety of problems, and they are becoming increasingly important in the modern world. Data mining is a process of extracting knowledge from large datasets. It is a subset of machine learning that uses statistical methods to find patterns in data. Data mining can be used to find hidden relationships, predict future outcomes, and identify outliers. Classification is a type of data mining task that involves assigning new data points to one of a set of predefined categories. Classification algorithms are trained on a dataset of labelled data points. The training data is used to learn the relationship between the features of the data points and their categories. Once the algorithm is trained, it can be used to classify new data points. In classification, accurate detection of disease by using training and testing data set [12]. There are many different classification algorithms available. Some of the most popular algorithms include:

- Support vector machines (SVMs)
- Naive Bayes classifiers
- Decision trees
- Random forests
- K-nearest neighbors (KNN)

The choice of classification algorithm depends on the specific data mining problem. Some factors to consider when choosing an algorithm include:

- The size of the dataset
- The number of features
- The distribution of the data
- The desired accuracy

Data mining and classification techniques are widely used in a variety of applications, including:

- Fraud detection
- Customer segmentation
- Risk assessment
- Medical diagnosis
- Natural language processing

This study findings suggest that data mining classification techniques can be used to predict liver disease with a high degree of accuracy. This information can be used to develop early detection and treatment strategies for liver disease, which can improve patient outcomes and prevent complications.

Support Vector Machines (SVMs)

Support Vector Machines (SVMs) are a type of supervised learning algorithm used for classification and regression analysis. The basis of SVMs is to find a hyperplane that best separates the classes in the input data. In a two-dimensional space, a hyperplane is simply a line that separates the data into two classes. In higher dimensions, a hyperplane is a linear subspace that separates the data into different classes. To find the best hyperplane, SVMs maximize the margin between the hyperplane and the closest data points of each class, which are called support vectors. This ensures that the hyperplane has the largest possible distance from the data points of both classes, and therefore the best generalization performance on unseen data. SVMs can also handle non-linearly separable data by using a kernel function to transform the input data into a higher-dimensional feature space where a linear separation is possible. The most commonly used kernel functions are the polynomial kernel, radial basis function (RBF) kernel, and sigmoid kernel. SVMs have been successfully applied in various fields, including computer vision, text classification, and bioinformatics, due to their ability to handle high-dimensional data and their strong theoretical foundation. However, they can be computationally expensive for large datasets and require careful tuning of their parameters.

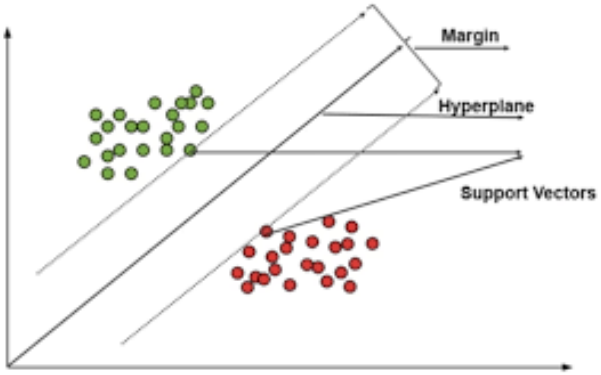

Fig: SVMs Data Classification

Naive Bayes Classifiers

The naive Bayes algorithm is a supervised learning algorithm that is used for classification tasks. They are based on Bayes' theorem, which provides a way to calculate the probability of a hypothesis given the observed evidence.
In a Naive Bayes classifier, the hypothesis corresponds to a class label, and the evidence corresponds to a set of features or attributes that describe the input data. The classifier assumes that the features are conditionally independent given the class label, which is called the "naive" assumption. This means that the probability of a certain combination of features occurring together can be calculated as the product of the probabilities of each feature occurring independently, given the class label.
To train a Naive Bayes classifier, the algorithm estimates the probabilities of each feature given each class label, and the prior probability of each class label. This is done using a training set of labelled examples.
To classify a new input data point, the Naive Bayes classifier calculates the posterior probability of each class label given the observed features, using Bayes' theorem. The class label with the highest posterior probability is then assigned to the input data point.
Naive Bayes classifiers are simple, fast, and efficient, and can work well even with a small amount of training data. They are often used in text classification tasks, such as spam filtering or sentiment analysis, where the input data consists of a bag-of-words representation of text documents

Decision Tree

Decision tree classification is a type of machine learning algorithm used for both classification and regression tasks. Classification issues are mostly handled with the use of a decision tree[12]. It is based on a tree-like model of decisions and their possible consequences. The decision tree consists of internal nodes, which represent tests on the input features, and leaf nodes, which represent the class labels or regression values.
To build a decision tree, the algorithm recursively splits the input data into subsets based on the values of the input features that best separate the classes. The best feature to split on is determined by a criterion such as information gain or Gini impurity, which measures the amount of information provided by the feature for classifying the input data. Once the tree is built, to classify a new input data point, the algorithm traverses the tree from the root to a leaf node, following the path that corresponds to the values of the input features. The class label or regression value at the leaf node is then assigned to the input data point. Decision tree classification has several advantages, such as being easy to interpret and visualize, handling both categorical and numerical input features, and handling missing values.
Fig: illustrate the diagnosis process, using decision trees, of patient that suffer from certain respiratory problem. The decision tree employees the following attributes: CT findings (CTF), X-ray findings (XRF), Chest Pain Type (CPT), and Blood Test Findings (BTF) in [19].

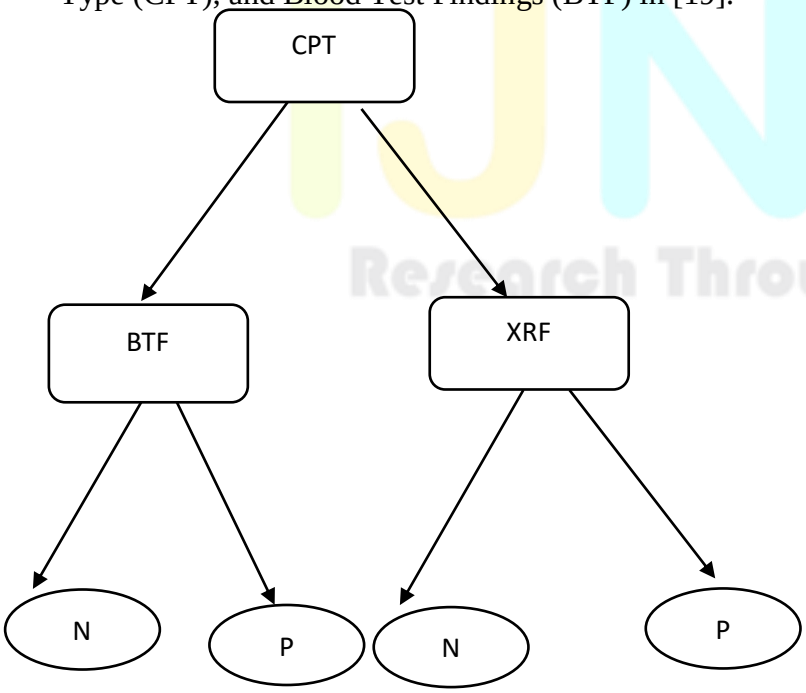

Fig: Decision Tree Classifier

Random Forest Classification

Random Forest is a powerful machine learning algorithm used for classification, regression and other tasks. It belongs to the family of ensemble learning methods, which combines multiple models to improve the accuracy and stability of predictions.

In Random Forest classification, multiple decision trees are trained on different subsets of the training data, each tree is trained on a random sample of features. The trees are then used to classify new instances by taking a majority vote of their predictions.

The algorithm has several advantages over other machine learning models, such as:

It can handle large datasets with high dimensionality.

It is less prone to overfitting than other models, thanks to the random selection of features and the aggregation of predictions from multiple trees.

It is easy to interpret and visualize, as each decision tree can be analyzed separately.

To train a Random Forest classifier, we first randomly sample a subset of the training data, with replacement. This is called bootstrapping, and it creates multiple training sets of equal size to the original dataset. Then, for each sample, we train a decision tree using a random subset of features, typically the square root of the total number of features. Finally, the predictions of all trees are combined using majority voting.

The Random Forest algorithm can be further optimized by tuning several hyperparameters, such as the number of trees, the maximum depth of the trees, and the minimum number of samples required to split a node. Cross-validation techniques can be used to find the optimal values of these hyperparameters.

The random forest algorithm can be summarized in the following steps:

1. Choose the number of trees in the forest. The number of trees in the forest is a hyperparameter that can be tuned to improve the performance of the model.
2. Create a bootstrap sample of the training data for each tree. A bootstrap sample is a random sample of the training data with replacement. This means that each data point in the training data has a chance of being selected for the bootstrap sample more than once.
3. Build a decision tree on the bootstrap sample. A decision tree is a machine learning algorithm that can be used to make predictions. Decision trees work by splitting the data into smaller and smaller groups until each group contains only data points of the same class.
4. Predict the label of a new data point by taking the majority vote of the predictions of the trees in the forest. The random forest algorithm makes a prediction for a new data point by taking the majority vote of the predictions of the trees in the forest. For example, if 5 of the 10 trees in the forest predict that the new data point is a cat and 5 of the 10 trees predict that the new data point is a dog, then the random forest algorithm will predict that the new data point is a cat.

The random forest algorithm is a powerful and versatile machine learning algorithm that can be used for a variety of tasks. It is a good choice for tasks where overfitting is a concern or where the data is not clean.

K-NN classification

K-Nearest Neighbors (KNN) is a classification algorithm that uses a simple yet effective approach to classify new observations based on their similarity to existing data points in a training set. K-Nearest Neighbor is a classification algorithm used to determine groups based on the majority of the k closest neighbours [7].

The number of neighbors that are considered is called the K value.

KNN is a non-parametric algorithm, which means that it does not make any assumptions about the underlying distribution of the data. KNN can handle both binary and multi-class classification problems, and it can also be used for regression tasks by predicting the average value of the K nearest neighbors.

Steps involved in the KNN algorithm:

1. Choose the K value. The K value is the number of nearest neighbors that will be considered when making a prediction. The K value can be chosen based on experience or by using cross-validation.
2. Calculate the distance between the new data point and each data point in the training set. The distance between two data points can be calculated using any distance metric, such as the Euclidean distance or the Manhattan distance.

3. Sort the training data points by their distance from the new data point. The training data points that are closest to the new data point will be considered its nearest neighbors.
4. Choose the labels of the K nearest neighbors. The labels of the K nearest neighbors will be used to predict the label of the new data point.
5. Predict the label of the new data point. The label of the new data point is determined by the majority vote of its K nearest neighbors. For example, if 3 of the 5 nearest neighbors are labelled "A" and 2 of the 5 nearest neighbors are labelled "B", then the new data point will be predicted to be labelled "A".

The KNN algorithm is a simple and versatile machine learning algorithm that can be used for a variety of tasks. It is a good choice for tasks where the data is not well-structured or where the underlying distribution of the data is unknown. KNN is also a good choice for tasks where speed is important.

Evaluation Metrics

The evaluation metrics for classification are accuracy, precision, recall and F1 score [13].

Accuracy

Accuracy is a common evaluation metric for classification problems. It is defined as the fraction of instances that are correctly classified. Accuracy can be calculated as:

Accuracy = (TP + TN) / (TP + TN + FP + FN)

where TP=True Positive, FP= False Positive, TN=True Negative, FN=False Negative

Precision

Precision is a measure of how accurate a classifier is when it predicts positive instances. It is defined as the fraction of predicted positives that are actually positive. Precision can be calculated as:

Precision = TP / (TP + FP)

Recall

Recall is a measure of how complete a classifier is when it predicts positive instances. It is defined as the fraction of actual positives that are predicted positive. Recall can be calculated as:

Recall = TP / (TP + FN)

F1 Score

The F1 score is a measure of a model's performance on a classification task. It is calculated as the harmonic mean of the precision and recall of the model. The F1 score is a useful metric because it takes into account both the precision and recall of the model, and it is not as sensitive to outliers as precision or recall. The F1 score is calculated as:

F1 = 2 * (precision * recall) / (precision + recall)

where:

precision = TP / (TP + FP)

recall = TP / (TP + FN)

TP = true positives

FP = false positives

Related Works

In [1], the authors compared the performance of four different ML algorithms for predicting liver disease: random forest, k-nearest neighbors (KNN), autoneural, and logistic regression. The algorithms were evaluated on the Andhra Pradesh liver disease dataset, which contains 583 liver disorder patient records with 11 attributes. The results of this study suggest that KNN is a promising algorithm for predicting liver disease. It is accurate, fast, and well-suited for datasets with complex or non-linear relationships

A work [2] published in 2019, HWOA-SA was used to classify Chronic Liver Disease(CLD) using a dataset of CT images. The results showed that HWOA-SA was able to achieve an accuracy of 98%, which is significantly higher than the accuracy of other machine learning algorithms that have been used for CLD classification. The results of this study suggest that HWOA-SA is a promising algorithm for predicting CLD. It is accurate, fast, and well-suited for datasets with complex or non-linear relationships.

In [3] the authors have analysed with real patient dataset for constructing classification models to forecast liver diagnosis. They applied five classification algorithms to given dataset and parameters such as precision, recall, and accuracy were analysed to determine performance of these classifiers. The highest accuracy is given by the Random Forest Algorithm when all algorithms are implemented without any changes. However, after adaptive boosting, the implementation of the C5.0 algorithm gives a more accurate result. The maximum precision has been obtained by the implementation of the K-Means Algorithm, whereas the maximum recall has been obtained by the implementation of the KNN Algorithm.

In [4] C.Geetha et al. proposes and evaluates two machine learning techniques for diagnosing liver disease in patients: support vector machines (SVMs) and logistic regression. The authors used a dataset of 100 patients with liver disease and 100 patients without liver disease. They trained the models on the dataset and then tested them on a held-out set of patients. The results showed that both models were able to predict liver disease with a high degree of accuracy. SVMs had an accuracy of 96%, while logistic regression had an accuracy of 95%. The authors concluded that both models are effective for diagnosing liver disease.

In [5] authors used three different machine learning algorithms to detect liver disease from blood test results: support vector machines (SMO), naive Bayes, and J48. SMO had the highest accuracy of 97.39%, while naive Bayes had the lowest accuracy of 70.72%. SMO also took the longest time to run, at 2.36 seconds, while J48 took the least amount of time, at 0 seconds. The study's findings suggest that SMO is the most accurate algorithm for detecting liver disease from blood test results. However, it is important to note that the study was conducted on a small dataset of only 100 patients. More research is needed to confirm the findings of the study on a larger dataset.

The results showed that the random forest model was able to predict Fatty Liver Disease (FLD) with an accuracy of 95% in [6]. This is a significant improvement over the accuracy of traditional diagnostic methods, such as liver biopsy, which has an accuracy of only 80%. The authors suggest that the random forest model could be used to screen patients for FLD in clinical practice. This would allow for earlier diagnosis and treatment of FLD, which could improve patient outcomes. The authors also suggest that the random forest model could be used to develop new diagnostic and treatment strategies for FLD. The findings of the study are promising, as they suggest that machine learning techniques can be used to improve the diagnosis and treatment of FLD. However, more research is needed to validate the findings of the study on a larger dataset.

The comparison of the performance of two machine learning algorithms, naive Bayes and KNN, for predicting liver disease in patients has done in [7]. The authors used the Indian Liver Patient Dataset (ILPD) from the UCI Machine Learning Repository. The ILPD dataset contains 586 patient records, each with 19 features. The authors trained and tested the models on the ILPD dataset. The results showed that the naive Bayes algorithm had a higher accuracy than the KNN algorithm. The naive Bayes algorithm had an accuracy of 83.7%, while the KNN algorithm had an accuracy of 81.6%. The authors also found that the naive Bayes algorithm was more robust to overfitting than the KNN algorithm. This is because the naive Bayes algorithm makes the assumption that the features are independent of each other. It is important to note that the study was conducted on a small dataset of only 586 patients. More research is needed to confirm the findings of the study on a larger dataset.

In [8] the authors used three different classification algorithms to predict liver disorder diseases: support vector machines (SVMs), naive Bayes, and C4.5 decision trees. The authors suggest that using a hybrid approach to liver disorder diseases prediction is a promising area of future work. By combining different data mining techniques, we can potentially improve the accuracy of our predictions.

In [9] the authors used liver function test reports to develop a model for predicting liver disorder. The authors used three different classification algorithms: support vector machines (SVMs), logistic regression, and decision trees. The logistic regression algorithm had the highest accuracy of 95.8%. The

authors also used a confusion matrix to evaluate the performance of the models. The confusion matrix showed that the logistic regression algorithm had a high true positive rate and a low false positive rate. The AUC was also almost equal to 1, which is a good indication of the model's performance. They also suggest that a larger dataset could be used to train the model and to determine the best algorithm for predicting liver disorder.

In [10] reviews on different data mining algorithms for predicting various diseases, including heart, liver, kidney, and diabetes has done. The authors found that support vector machines (SVMs) and naive Bayes are the most commonly and widely used algorithms for disease prediction. They also found that KNN, SMO, and random forest algorithms are also used, but due to their complexity they are not widely accepted and preferred for disease prediction. The authors also implemented different data mining algorithms in Weka using different data sets obtained from UCI respiratory. They used the following parameters for performance analysis of algorithms: correctly classified instances, precision, recall, and F-Measure. After simulation results and discussion, they concluded that random forest algorithm shows best accuracy for heart, liver, and kidney disease prediction. SMO performs best for diabetes prediction with an accuracy of 77.34%.

In [11] the authors used classification algorithms, namely naive Bayes and support vector machine (SVM), for liver disease prediction. The authors compared the performance of the two algorithms based on the performance factors classification accuracy and execution time. They found that SVM had a higher classification accuracy than naive Bayes, but naive Bayes had a lower execution time. The authors conclude that SVM is a better choice for liver disease prediction because it has a higher classification accuracy. However, they also note that naive Bayes is a good choice for liver disease prediction if execution time is a concern. The results of the paper are based on a limited dataset. More research is needed to confirm the findings of the paper on a larger dataset.

Process of liver disease prediction

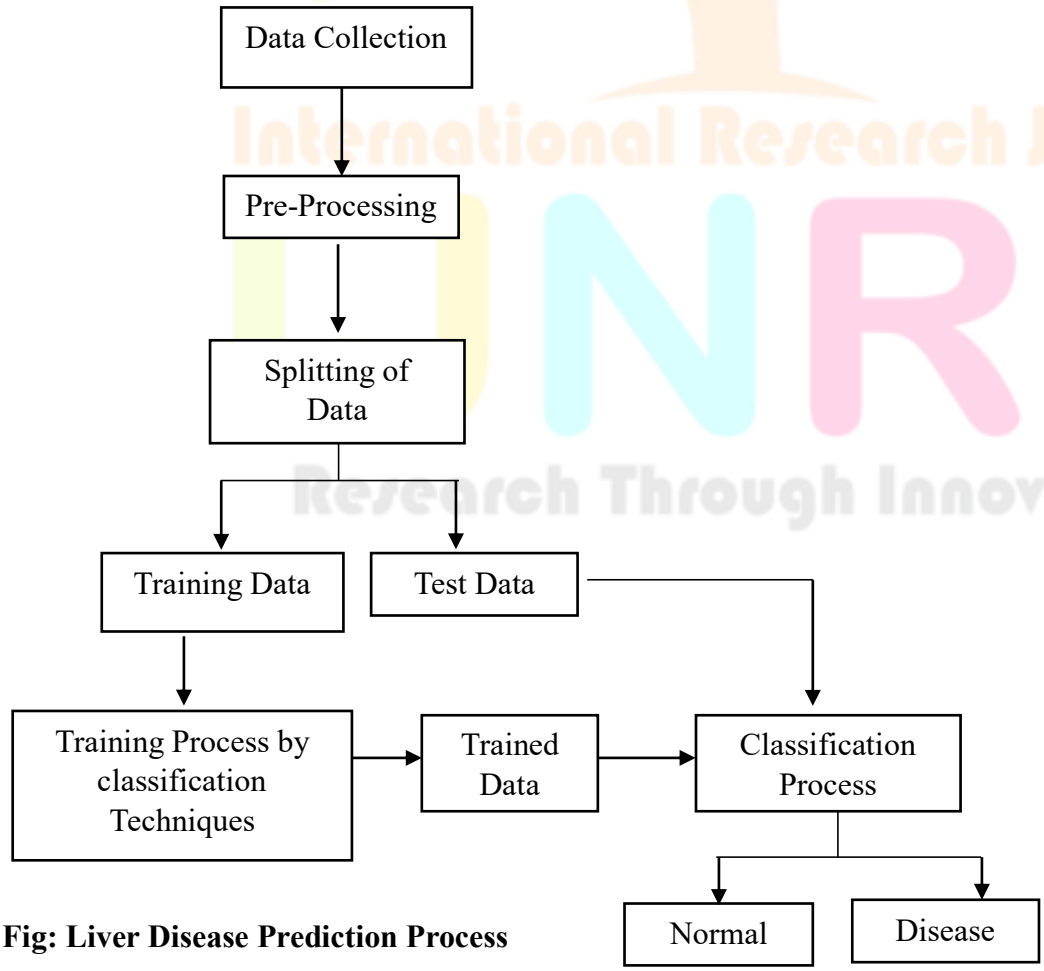

Fig: Liver Disease Prediction Process

Process of liver disease prediction using data mining classification techniques:

Data collection

The first step is to collect data on patients with liver disease. This data can be collected from a variety of sources, such as electronic health records, clinical trials, and surveys, IoT devices. The data needed for liver disease prediction varies depending on the type of liver disease being predicted. Some common data points that are used for liver disease prediction include:

- Demographic data, such as age, sex, and race
- Clinical data, such as blood pressure, cholesterol levels, and liver function tests
- Lifestyle data, such as smoking status, alcohol consumption, and diet
- Genetic data, such as the presence of certain genetic mutations that are associated with liver disease

In addition to these data points, other data points that may be used for liver disease prediction include:

- Imaging data, such as ultrasounds or CT scans
- Biomarkers, such as proteins or enzymes that are found in the blood or liver tissue
- Self-reported data, such as symptoms and medications

The data that is needed for liver disease prediction will vary depending on the type of liver disease being predicted and the specific prediction model that is being used.

Data pre-processing

Once the data has been collected, it needs to be processed for analysis. Data pre-processing is the process of cleaning, formatting, and transforming raw data into a form that is suitable for analysis. The steps involved in data pre-processing vary depending on the specific data set and the analysis that will be performed.

Data cleaning – Data cleaning involves identifying and correcting any errors or inconsistencies in the data.

Data formatting – Data formatting involves converting the data into a format that is compatible with the analysis tools that will be used.

Data transformation – Data transformation involves changing the data in a way that makes it more useful for analysis. This can involve creating new variables, aggregating data, or normalizing the data.

Feature selection

The next step is to select the features that will be used to predict liver disease. This is an important step, as the features that are selected will have a significant impact on the accuracy of the prediction model. There are a number of different methods that can be used to select features, such as the chi-squared test and the information gain ratio.

Model building

Once the features have been selected, the next step is to build a model that can be used to predict liver disease. Model building is the process of creating a model that can be used to predict the outcome of a particular event. In data mining, classification techniques are used to create models that can be used to predict the category of a data point.

There are a number of different classification techniques that can be used, such Support vector machines (SVMs) , Decision trees, Random forests. The choice of classification technique will depend on the specific problem that is being solved. Some factors that may be considered include the size of the data set, the number of features, and the desired accuracy of the model. Once a classification technique has been chosen, the next step is to build the model. This involves fitting the model to the data set. The model is fit by finding the parameters of the model that minimize the error between the model's predictions and the actual values.

Model evaluation

Once the model has been built, it needs to be evaluated to see how well it performs. This is done by splitting the data into a training set and a testing set. The training set is used to build the model, and the testing set is used to evaluate the model's performance. The performance of the model is usually measured using accuracy, precision, and recall.

Model deployment

Once the model has been evaluated and found to be satisfactory, it can be deployed in a real-world setting. This could involve using the model to predict liver disease in patients who are being seen in a clinic or hospital.

The process of liver disease prediction using data mining classification techniques is a complex one, but it can be a valuable tool for early diagnosis and treatment of liver disease.

Conclusion

This study highlights the potential impact of data mining classification techniques on the diagnosis of liver disease. The combination of data mining classification techniques can deal with large datasets and provide better accuracy rate. The use of hybrid data mining techniques has the potential to improve the accuracy of diagnosis, which can ultimately lead to better patient outcomes. Furthermore, the findings suggest that healthcare providers and researchers should continue to explore the use of data mining techniques to improve diagnosis and treatment outcomes for liver disease and other health conditions. With continued advancements in machine learning and data analytics, the potential for data mining techniques to revolutionize the field of medicine is immense.

References:

1. Ain Najwa Arbain et al. Comparison of Data Mining Algorithms for Liver Disease Prediction on Imbalanced Data, International Journal of Data Science and Advanced Analytics, 2019
2. G. Ignisha Rajathi and G. Wiselin Jiji , Chronic Liver Disease Classification Using Hybrid Whale Optimization with Simulated Annealing and Ensemble Classifier, Symmetry 2019, 11, 33
3. Sanjay Kumar and Sarthak Katyal, Effective Analysis and Diagnosis of Liver Disorder by Data Mining, Proceedings of the International Conference on Inventive Research in Computing Applications (ICIRCA 2018)
4. C.Geetha et al. Evaluation based Approaches for Liver Disease Prediction using Machine Learning Algorithms, International Conference on Computer Communication and Informatics, Informatics 2021
5. M. Sujatha et al. A Survey of Classification Techniques in Data Mining, International Journal of Innovations in Engineering and Technology (IJIET), Vol. 2 Issue 4 August 2013
6. Chieh-Chen Wu et al. Prediction of fatty liver disease using machine learning algorithms, Computer Methods and Programs in Biomedicine 170 (2019) 23–29, ScienceDirect
7. Hartatik et al. Prediction for Diagnosing Liver Disease in Patients using KNN and Naïve Bayes Algorithms, 2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS).
8. L. Alice Auxilia, Accuracy Prediction using Machine Learning Techniques for Indian Patient Liver Disease, Proceedings of the 2nd International Conference on Trends in Electronics and Informatics (ICOEI 2018)
9. Vyshali J Gogi and Dr. Vijayalakshmi M.N, Prognosis of Liver Disease: Using Machine Learning Algorithms, International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering - (ICRIEECE), 2018
10. Muhammad Nabeel et al. Review on Effective Disease Prediction through Data Mining Techniques, International Journal on Electrical Engineering and Informatics - Volume 13, Number 3, September 2021.
11. Dr. S. Vijayarani1, Mr.S.Dhayanand, Liver Disease Prediction using SVM and Naïve Bayes Algorithms, International Journal of Science, Engineering and Technology Research (IJSETR) , Volume 4, Issue 4, April 2015
12. Han J and Kamber M (2006). Data Mining: Concepts and Techniques, 2nd edition, (The Morgan Kaufmann Series).
13. A.Sivasangari et al. Diagnosis of Liver Disease using Machine Learning Models, Proceedings of the Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), IEEE Xplore