INTELLIGENT LIVER DISEASE PREDICTION SYSTEM BY USING MACHINE LEARNING TECHNIQUES

BY

T.M. KAMRUZZAMAN
ID: 171-15-9183

AND

MD. SALMAN MAHBUB
ID: 171-15-9184

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Md. Azizul Hakim
Lecturer
Department of CSE
Daffodil International University

Co-Supervised By

Sheikh Abujar
Sr. Lecturer
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JANUARY 2021

APPROVAL

This Project titled "**Intelligent Liver Disease Prediction System By Using Machine Learning Techniques**", submitted by **T. M. Kamruzzaman** and **Md. Salman Mahbub** to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 28/01/2021.

BOARD OF EXAMINERS

Professor Dr. Touhid Bhuiyan **Chairman**
Professor and Head
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Hhena

Most. Hasna Hena **Internal Examiner**
Assistant Professor
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Nusrat Jahan

Nusrat Jahan **Internal Examiner**
Senior Lecturer
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Dr. Shamim H Ripon **External Examiner**
Professor
Department of Computer Science & Engineering
East West University

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Md. Azizul Hakim, Lecturer, Department of CSE** and co- supervision of **Sheikh Abujar, Sr. Lecturer, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

____

Md. Azizul Hakim
Lecturer
Department of CSE
Daffodil International University

Co-Supervised by:

____

Sheikh Abujar
Sr. Lecturer
Department of CSE
Daffodil International University

Submitted by:

Kamruzzaman
____

T.M. Kamruzzaman

ID: - 171-15-9183
Department of CSE
Daffodil International University

Salman

____

Md. Salman Mahbub
ID: - 171-15-9184
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year thesis successfully.

We really grateful and wish our profound our indebtedness to **Md. Azizul Hakim**, **Lecturer**, Department of CSE and **Sheikh Abujar**, **Sr. Lecturer**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "*Data Mining*" and "*Machine Learning*" to carry out this project. His endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Professor Dr. Touhid Bhuiyan**, Honorable Professor and Head**,** Department of CSE for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

We all know that the liver is one of the most important organs of our body function. Once upon a time we couldn't see that a large number of people were suffering from liver diseases. But in recent years we can see that the number of patient of liver problem is increasing day by day. So this affected people should go to a medical center for checking. But in this COVID situation it is risky for going to the medical center. So in this thesis we are working for the liver affected people by which they don't need to go outside for checking the possibility of Liver disease of them. We took some data based on some basic attributes which related to liver diseases and make a classifier model for predicting the possibility of liver diseases. Then we gave some data on that classifier model. The data carries both the liver affected people and non-affected people. This data is used for training the machine about to identify the affected people and non-affected people easily. Then we run some algorithm like KNN, Naïve Bayes, Decision Tree and SVM on that model and generate some results based on these algorithms. We did the evaluation into two different approaches. Firstly we generated the complete result with all the attributes from the selected data. After that we selected some attribute from that data and run the classifier algorithm on that. It generated some different result which gave us better accuracy from all attribute based result. In this approach people will know their liver diseases possibility easily at home which will save time and hassle as well.

TABLE OF CONTENTS

CONTENTS **PAGE**

CHAPTER

LIST OF FIGURES

FIGURES **PAGE**

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Introduction

We all know, Liver is one of the most important organs of human body. It helps our body to run it smoothly. But a recent study tells that at present approximately 2 million deaths happens per year worldwide due to the various kind of liver diseases. [1] From this, 1 million suffers due to various complications of cirrhosis [2] and 1 million suffers due to viral hepatitis. If any liver failure happens, it can be a life-threating condition. But most of the time liver failure happens gradually. There are many reasons behind liver diseases [3]. Like, Infection is one of the main reasons for various kinds of liver diseases. Like, Hepatitis A, B, C viruses can infect the liver and interrupts its functions. These viruses are able to make a greater damage of our liver and it can spread through blood, contaminated food, contaminated water and close contact with an infected people. Again, immune system abnormality is one of the great reasons of liver diseases. If our immune system performs abnormal participation of our various body function then it can cause various damages to our health. It is harmful to the important organ of our body like liver. Again an important reason of liver diseases is genetics. It can cause various liver diseases like Hemochromatosis. Again the growth of cancer cells into liver can causes liver cancer.

There is a lot of risk factor for liver diseases [3] like alcoholic, obesity, diabetes, tattoos on body, blood transfusion, taking blood from others etc. Again if anyone’s family have the history of liver diseases then he\she has a great risk of being a liver patient in future.

Beside the risk factors, there are some symptoms [3] of liver diseases, but the symptoms are not always visible. Sometimes the symptoms are visible and sometimes they are invisible in our country. Some visible symptoms of liver diseases are the skin and the eyes turns into yellowish that means Jaundice, there will be abdominal pain and swelling, the skin will itching and the urine color will be dark, there will be nausea and vomiting and chronic fatigue etc.

The vast majority of individuals dependably take high utilization of sugar, salt, cholesterol and oily food which indicates the unhealthy and unbalanced food habit can causes this diseases. Again alcohol cigarette, tobacco, physical idleness is another reason of liver diseases to increasing rate.

In these circumstances, we utilized a machine learning technique in our framework which is SVM (Support Vector Machine). Our framework utilizes likewise a settled an incentive for preparing and test set to characterize those liver diseases exists or not.

1.2 Motivation

In the case of liver diseases, the symptoms are not always visible. Sometimes they are invisible but can create a problem into our body function. So when this kind of symptoms stays into a body then the people always have to go to the hospital for clinical testing which is time consuming and as well as a lengthy process. Again in this CORONA Virus pandemic it is risky too. In this situation patients are not having proper treatment in early which is bad for this type of patient's condition. That's why many patients do not get right treatment and can't understand what treatment he\she should do. So many people are suffering more and more. Our system helps to give a good result for this situation and get early result that liver diseases is there or not.

1.3 The rationale of the story

Nowadays Liver Cirrhosis, Fatty Liver, Hepatitis etc. are very serious diseases. A large number of people around the world is suffering from these diseases. Again many people lost their lives for these diseases. For taking vaccine the death rates are decreased but the affected rate is increasing day by day. Because people do not identify their symptoms at early stages. Only for these reason the rate of affected people is increasing. If we prevent it in early stages then it can be recovered properly. So in this wake of seeking and breaking down we picked liver diseases as our exploration point. For turning into an expansive number of dead on the liver diseases, the exploration subject has been chosen.

At long last, the paper has been chipping away at this to give a superior proposal that encourages us to decrease the dead number for our modern aged people groups.

1.4 Research Questions

a) Does it show the accurate value to predict liver diseases with both all attributes and the selected attributes from the datasets?

b) Does it classify liver disease by machine learning algorithm?

Effectively, a few destructive sicknesses have been distinguished for an individual. Even each sickness has an answer for avoidance it's unrealistic for everybody because of just for obviousness. Everybody needs to have an upbeat existence in where an infection is the only obstacle. Any sort of illness avoidance is conceivable if that in stay essential stage. For that reason, we assembled an expectation framework that assists with recognizing the illness stage and gives us the outcome that the person has liver disease or not. The entirety of the illnesses, sickness is viewed as one of the main infections. Numerous people groups are kicked the bucket due to this illness. This illness is the greatest enemy of the two people in the world. In our Bangladesh there is one work but not that kind of notable framework for liver diseases. At last, we chosen it as our examination subject in Bangladesh individuals for our pleasure and furthermore attempt to make a decent framework for forecast liver disease infections.

1.5 Expected Output

In our liver disease prediction system, we generate an output based on the given datasets which was expected. We all know that in this COVID situation it is risky to collect data from different clinical center. For that reason we take our datasets from ILPD (Indian Liver patient’s data) and BUPA as all the south Asian peoples has almost the same reason for liver diseases. After that we run some classifier algorithm on that datasets. We used KNN, SVM, Decision Tree and Naïve Bayes classifier to get our expected result. But we did our job into two ways. Firstly we run that classifier on all of the features of that dataset. Then we found an output from that. Then we selected some important features from that datasets then we run the classifiers on that’s again. After that we found a better accuracy from that. But among the both works the Support Vector Machine (SVM) classifier gave the best output as we expected.

1.6 Layout of the Report

- Chapter 1 have demonstrated an introduction to the project with its motivation, research questions and expected outcome.

- Chapter 2 will have “Background Study” demonstrates introduction, related works, research summary and challenges that we have faced during this research.
- Chapter 3 will have Research Methodology.

- Chapter 4 will have Experimental Results and Discussion.

- Chapter 5 will have Summary and Conclusion.

CHAPTER 2

BACKGROUND STUDY

2.1 Introduction

In this segment, we will examine about some related works, research summary and challenges that we have looked about this exploration. In related works segment, we will talk about other examination papers and their works, their strategies and precision which are identified with our work. In exploration rundown area we will give the summary of our connected works. In challenges segment, we will examine about our data collection, feature selection process and how we increased the accuracy level.

2.2 Related Works

In this section we have evaluated some connected works. Where we attempted to peruse out or discover the hole of these work. The latest papers we have surveyed. A large portion of the creators of these connected works have introduced ILPD dataset. And furthermore they have executed some ML techniques. The related works are described below in details.

M. Banu Priya, P. Laura Juliet and P.R. Tamilselvi proposed a system[4] named Performance Analysis of Liver Disease Prediction Using Machine Learning Algorithms on which they implemented a feature model construction and comparative analysis for improving prediction accuracy of Indian liver patients. They had accomplished their work by utilizing three stages .In first stage, min max standardization calculation is applied on the first liver patient datasets gathered from UCI storehouse. In second stage, by the utilization of PSO highlight choice, subset (information) of liver patient dataset from entire standardized liver patient datasets is gotten which includes just huge characteristics. Third stage, characterization calculations are applied on the informational collection. They have use MLP (Multilayer Perceptron), Bayes net, SVM, Random

Forest, J.48 algorithm. And from the above algorithms J48 gave the less Mean Absolute Error and gave the best accuracy.

Chieh-Chen Wu a , e , Wen-Chun Yeh b , Wen-Ding Hsu c , Md. Mohaimenul Islam a , e , Phung Anh (Alex) Nguyen e , Tahmina Nasrin Poly a , e , Yao-Chin Wang a , e , d , Hsuan-Chia Yang e , Yu-Chuan (Jack) Li proposed a framework[5] named Prediction of greasy liver infection utilizing AI calculation In this theory, they did a relative investigation for improving expectation exactness of Taiwan's greasy liver patients They incorporated all patient who had an underlying greasy liver screening at the city medical clinic of their nation. Grouping calculation, for example, Random Forest, Naïve Bayes, ANN, Logistic Regression (LR) were created to foresee the greasy liver infection. The region under the recipient working attributes bend (ROC) was utilized to assess the precision of the grouping models. All out 577 patients information was in their examination and 377 patients had greasy liver illness. The zone under the beneficiary working attributes said that the Random Forest Algorithm gave the best exactness of 88.48%. This paper contains some impediment. They just gathered information from one clinical focus. Be that as it may, multicenter dataset and outside approval could have better performance and more dependable. They couldn't order patients into greasy and non-greasy liver dis-ease patients because of information deficiency. They utilized an arrangement approach for programmed ML factors combination, however profound learning approach might have been utilized to improve better expectation.

Tapas Ranjan Baitharu, Subhendu Kumar Pani proposed a model [6] named Analysis of Data Mining Techniques For Healthcare Decision Support System Using Liver Disorder Dataset. In this paper they were attempting to create astute clinical choice emotionally supportive networks by utilizing some grouping calculation. First they gathered a few number of information dependent on some characteristic which increment the danger of liver illnesses. At that point they actualized some characterization calculation on that information and gather the exactness that the calculation gave. They utilized WEKA to actualize their work. The calculation that they utilized are j-48, ZeroR, Multilayer

Perceptron, 1BK, Naïve Bayes, VFI. The Multilayer Perceptron classifier Algorithm gave the best precision. They utilized an order approach for programmed ML factors combination, however profound learning approach might have been utilized to improve better forecast. What's more, multiple dataset will give the better exactness.

Dr. S. Vijayarani1, Mr.S.Dhayanand2 proposed a system named [7] Liver Disease Prediction using SVM and Naïve Bayes Algorithm. In this proposition they were attempting to foresee liver illness by utilizing two classifier calculations. First they gathered a few number of information dependent on some properties which increment the danger of liver illnesses and the qualities resemble. Family background of liver illness • Smoking • Consumption of liquor • Intake of sullied food • Obesity • Diabetes are the main feature of their work. Mainly they utilized SVM and Naïve Bayes classifier calculation to do the expectation the liver illnesses and the SVM gave the preferable precision over the Naïve Bayes calculation. In their work they just attempted two classifier calculations. There are some other calculation left that they didn't utilize like ANN, RF, j-48 and so on. This calculation may have that capacity to give the preferred outcome over the pre-owned two calculations.

Dr. N. V. Ramana Murthy1, S. Shruti2, V. Vinay Bhargav3, S. Anil Kumar4 proposed a system [8] Liver Disease Prediction and Diagnosis Expert System using Data Mining Techniques**.** The fundamental worry of this proposal is to plan and build up a clinical conclusion master framework which helps the doctors in dynamic through gathered information of liver problems by utilizing getting standards. They gathered a few dataset from UCI machine learning archive. The ILPD contains 11 unmistakable properties of complete 583 patient records where 416 are liver patients and 167 are sound patients. This dataset holds records of 142 female and 441 guys. They utilized Weka to actualize their work and they utilized a few classifier calculation like j-48(decision trees), RF, Naïve Bayes, Multilayer Perceptron calculations. The Multilayer Perceptron calculation gave the best precision put together based with respect to their information.

Kalyan Nagaraj1* and Amulyashree Sridhar2 proposed a system named[9] A Graphical User Interface for Identification of Liver Patients.

In this paper they applied Data Mining Technique for the prediction of Liver Diseases. Those predictions gave various percentage(output) of accuracy. Five types of Algorithm was applied to the process and the best Algorithm gave the best accuracy. They build a Hybrid system to develop the accuracy and a screening system (GUI) for the better representation. The GUI represented the identification of Liver Patients whether the human is affected or not by Liver problems using the installed trained methods.

At first they collected the data from respected organization. Then they pre-process the data to normalize the missing value and the missing value were replaced by NULL value. Then they selected the Feature using some methods. They randomized the data to obtain and arbitrary permuted sample. Then they split the data set for training (70%) and for testing (30%). Then they applied 4 data mining algorithms (Naive Bayes Classifier, Bagging, Random Forest, Support Vector Machine). Then they record the prediction performance of the algorithms. Then they develop a hybrid NeuroSVM model to classify the patients using Artificial Neural Network and Support Vector Machine. And as well as at last they made a Hybrid Model as a GUI in R. Hybrid model was tested for its performance by using Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE).

They have collected 583 instances based on ten different biological parameters. These are either YES (416) cases and NO (167) cases to represent liver problems. Pre-processing techniques were applied to normalize the missing values. Feature selection was performed using both Filter and Wrapper methods. Correlation of data set were removed by using correlation analysis. In GUI, feature was selected using library “Boruta” in R. Four types of algorithm algorithms (Naive Bayes Classifier, Bagging, Random Forest, Support Vector Machine) were used in R platform to for classification. And a Hybrid model was implemented using Artificial Neural Network and Support Vector Machine. Then the GUI was implemented by using “gWidgets” ,“RGtk2” and “tcltk2” libraries in R.

Naïve Bayes: 53.09% Accuracy

Bagging: 66.73% Accuracy

Random Forest: 67.67% Accuracy

Support Vector Machine: 76.22% Accuracy

NeuroSVM: 98.83% Accuracy

Attributes after Dual selection: Age, Total Bilirubin, Direct Bilirubin, Alkphos Alkaline Phosphatase, Sgpt Alanine Aminotransferase , Sgot Aspartate Aminotransferase.

Tapas Ranjan Baitharua[1], Subhendu Kumar Panib[2] proposed a model[6] namely Analysis of Data Mining Techniques For Healthcare Decision Support
System Using Liver Disorder Dataset
Liver diseases are one of the most common causes of death. In this paper they have learned a pattern through the collected data (Data collected from the healthcare industry) of Liver Disorder to create an intelligent system to help the doctors. It gives the comparative performance, effectiveness, analysis, correction of some algorithm. In this process they have used some algorithm and recorded the accuracy of every single algorithm. And the process gave the prediction and description of accuracy and it also reduces the time complexity of the system.

They have used Data Mining technique to find the prediction and description. They have used some classification technique. They also have created pre-classified examples and train the system to develop the model which can predict and classify the data. Then they applied the test data and have seen the outcome of the system which classifier can give the best accuracy and which classifier can give the worst accuracy. Data was applied to a tool for data pre-process, feature reduction, classification, regression etc.

They have used WEKA to pre-process the data and it’s a data Mining Tool

They have used some classifier which is:-

- Decision Tree J48,
- Naïve Bayes.
- Multilayer Perception,
- ZeroR,
- Nearest Neighbor(IBK),
- VFI,

- Margin Curve.

For pre-classified classification they have used Fraud Detection and Credit Risk Application

Multiplayer Perception (71.59% Accuracy)

Decision Tree J48 (68.97% Accuracy)

IBK (62.8986% Accuracy)

VFI (60.2899 Accuracy)

ZeroR (57.971% Accuracy)

Naïve Bayes (55.3623% Accuracy)

So, Multiplayer Perception gave the best output and Naïve Bayes gave the slowest output.

Kemal Akyol [1] and Yasemin Gültepe[2] proposed a system namely[9] A Study on Liver Disease Diagnosis based on Assessing the Importance of Attributes

In this paper they have tried to find out the important attributes which are the main reasons for effective prevention of Liver Disorder patients by using Machine Learning process. They have tried to create and wanted to understand balanced data and meaningful data in order to give maximum accuracy of the classifiers. And they have proved that Balanced data gives more accuracy than Unbalanced data. Those balanced and unbalanced data have gone through a process and those data was evaluated and recorded in the frame of Accuracy and Sensitive metrics. They collected some dataset from BUPA and ILDP, then they cleaned the data from null or irrelevant values. Then they normalized the data in range from 0 to 1 value. The collected data was unbalanced and they have balanced it using Random Under-Sampling (RUS), after balancing datasets were called sub-datasets. For finding most effective attribute, they have applied Stability Selection (SS) method. And then sub-datasets were send to Random Forest (RF) algorithm to find out Liver disease or not. Finally, They evaluated the performance of both balanced and unbalanced dataset and showed it in confusion matrix structure.

For finding the effective attributes for diagnosis of Liver Disorder, they have used the combination of Stability Selection and Random Forest. For balancing the dataset, they have used Random Under-Sampling (RUS). Important attribute was detected by Stability Selection Method which were obtained with 5 Fold cross-validation Technique. For evaluate the performance of datasets they have used Random Forest and have shown in Accuracy and Sensitive Metrics.

Nazmun Nahar[1] and Ferdous Ara[2] proposed a model named[10] LIVER DISEASE PREDICTION BY USING DIFFERENT DECISION TREE TECHNIQUES.

In this paper they have tried to predict the Liver Diseases using different Decision Trees techniques. They compared the performance of various Decision Trees with respect to the seven different criteria (Accuracy, Precision, Recall, Mean absolute Error, F-Measure, Kappa Statistics, and Run time) and find the best and worst technique for the collected datasets to predict the Liver Diseases at an earlier stage. They collected the dataset from UCI Machine Learning Repository and attributes are Age, Gender, Total Bilirubin, Direct Bilirubin, Alkphos Alkaline Phospotase, Sgpt Alamine Aminotransferase, Total Proteins, Albumin, Albumin and Globulin Ration and class. Then they run those data sets through WEKA. It’s a very efficient tool to classify the performance of algorithms. And then they recorded the performance for each algorithm and compared them. They have compared the performance of 7 algorithms and these are J48, LMT, Random Forest, Random Tree, REPTree, Decision Stump and Hoeffding Tree in WEKA. Decision Stump gave the highest accuracy (70.67%) and J48 gave the worst accuracy of (65.69%). In terms of Kappa Statistics Decision Stump gave the highest value of 0.397 and LMT gave the lowest value of 0.065. And in terms of run time Random Tree and Decision Stump got the lowest run time of 0.01 and LMT got the highest run time of 0.88. They suggested to collect the recent data from different places across the world for the diagnosis of Liver Diseases. And they encouraged us to apply other different trees like Classification and Regression Tree (CART).

From the above discussion we find some common technique [13] that they used on their thesis. Here is the details information about those techniques.

Using Fuzzy Logic

Fuzzy logic and set theory can be suitable for developing knowledge based systems for diagnosis of diseases in healthcare. In research the authors have proposed a method that combines the genetic algorithms for feature selection and fuzzy expert system (Mamdani Model) for effective classification. Here experiments are taken by using fuzzy tool in MATLAB. The dataset is used from UCI machine learning repository and six (06) attributes are used in the experiment. In this system, the input is the set of all selected features, whereas the output is either a value 0 or 1 (0-absence, 1-presence) of heart disease in a patient. Here, genetic algorithms and fuzzy logic are used. It is to be noted that fuzzy logic is a mathematical tool and is a subfield of AI. The problem with using only fuzzy logic in AI is that the system can only implement the rules and cannot learn as it goes along.

Using data mining techniques

In most of the works the authors have presented a comparison of different data mining algorithms that predict the risk of heart diseases, namely C5.0, Neural Network, Support Vector Machine (SVM), and K-Nearest Neighborhood (KNN). It is observed that Decision tree has greater accuracy of 93.02% than KNN (88.37%), SVM (86.05%), and Neural Network (80.23%). The results produced using decision tree is interpretable, applicable and easily understandable by different clinical practitioner. Here, 270 records with thirteen (13) features are used from the UCI and four (04) classifiers including C5.0, SVM, KNN and Neural Network are used. Data divided into training set and test set (70% and 30%, respectively). The training set is used to build the classifier and test set used to validate it. It is observed that decision tree outperforms others with 93.02% accuracy and the accuracy result of ANN is very low.

In a work the authors have presented data classification based on various ML algorithms, namely KNN, Naïve Bayes, and Decision List. A data mining tool, known as TANAGRA, is used to classify the data where the data is evaluated using 10-fold cross validation. Here the training data set of 3000 instances with fourteen (14) different attributes is used in the experiments. Depending upon the attributes, the dataset is classified into two parts: 70% of the data is used for training and 30% is used for testing. It is observed that the Naïve Bayes algorithm (52.33% accuracy with 609ms) performs better than the other two algorithms i.e., Decision List with 52% accuracy and KNN with 45.67% accuracy and the accuracy of these algorithms are very low.

Different approaches of J48 Decision Tree

Almost every work that we have reviewed we can see that the authors have presented a prediction model of various diseases using some attributes including various disease results collected from UCI database. They have applied decision tree J48 algorithm using different approaches, such as pruned, un-pruned, and reduced error pruning approach for prediction heart disease based on these attributes. Where they observed J48 reduced error pruning approach is (almost 76%) better than J48 un-pruned (almost 73%) and J48 pruned (almost 74%) approach. Also they have observed that fasting blood sugar is most important feature which gives better classification result than other features. However, the problem is fasting blood sugar is not provided good accuracy.

Liver sickness is an illness that assaults straightforwardly to the liver. Undoubtedly, the liver is a significant piece of each person. Subsequently, in the event that we need to have a sound existence, we must be careful. In the event that we discover this illness from the get-go in the underlying stage, we can without much of a stretch conquer it. Else we should languish over this infection for our future life.

After making the decision based on the current situation, we wanted to establish a system that provides better performance due to disease and understand the situation of affected patients. At last, we contact our normal objective for God's favoring, which we have thought to execute.

2.3 Challenges

In this section we will discuss about the challenges that we have faced during our research. Like, raw data collection is one of the big challenges for getting better accuracy. Without data, the prediction is not possible and the system can’t predict. And we all know that in this COVID situation the hospitals not allowed any unwanted visitor for others purposes except treatment. For that reason we had to collect our data from ILPD and BUPA. After that, another challenge is preprocessing. After doing preprocessing our data set has no null value and helps us to get a good prediction. Next, Feature scaling helps to take all feature values into the same scale with respect to value. Therefore, different algorithm has been applied to the proposed architecture. Finally, the implementation process has been established to get accurate predicted value for both all attributes and selected some important attributes. There were several challenges rising according to the working procedure. We are tried to increase and get a better result for this model by using machine learning algorithm of support Vector Machine (SVM).

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction:

We collected two datasets from online. One is ILPD (Datasets 1) dataset, another one is BUPA (Dataset 2) dataset. Dataset 1 contains 583 data and 11 attributes. Dataset 2 contains 345 data and 7 attributes. We have had some missing value from an attribute of dataset 1. We fixed it. For the better accuracy and outcome we had to find the important features from both of the datasets so that the accuracy increases. For finding the important feature, we used WEKA (Filter and Wrapper methods). Datasets were used for testing and training purpose and here are some algorithm we used Support Vector Machine (SVM), Naïve Bayes (NB), K-Nearest Neighbor (KNN) and Decision Tree. We applied it in both datasets. Now we will see how we found the important attributes and how many percentage of the accuracy increases.

3.2 Data Collection Procedure:

For our system, we tried to collect data from hospitals but because of the pandemic situation (COVID-19) we didn't make it. That's why we had to collect data from online. Both datasets were taken from the website of the University of California. We tried for an unique research. We searched many papers about Liver diseases. No one described the accuracy after selecting important attributes. That's why we tried to improve the accuracy after selecting important attributes.

Dataset Details:

We have worked on both datasets:

ILPD [12]

BUPA[11]

TABLE 3.2.1: ILPD (Dataset 1)

Attribute	Description	Possible Value
AGE(V1)	Age of the patient	NUMERIC
GENDER(V2)	Gender of the patient	NOMINAL
TB(V3)	Total Bilirubin	NUMERIC
DB(V4)	Direct Bilirubin	NUMERIC
ALKPHOS(V5)	Alkaline phosphatase	NUMERIC
SGPT(V6)	Alamine Aminotransferase	NUMERIC
SGOT(V7)	Aspartate Aminotransferase	NUMERIC
TP(V8)	Total Proteins	NUMERIC
ALB(V9)	Albumin	NUMERIC
A/G(V10)	Albumin and Globulin Ratio	NUMERIC
SELECTOR(V11)	Patient or Not (0 or 1)	NOMINAL

In this dataset, there are 11 attributes which predicts the diseases. This dataset contains 583 instances where 416 of them are Liver patients which is (71.35%) and 167 of them are non-liver patients which is (28.64%). It contains 441 male and 142 female records. Selector and gender is nominal and the rest are numeric

TABLE 3.2.2: BUPA (Dataset 2):

Attribute	Description	Possible Value
MCV	Mean Corpuscular Volume	NUMERIC
ALKPHOS	Alkaline Phosphotase	NUMERIC
SGPT	Alamine Aminotransferase	NUMERIC
SGOT	Aspartate Aminotransferase	NUMERIC
GAMMAGT	Gamma-Glutamyl Transpeptidase	NUMERIC
DRINKS	Alcoholic Beverage Drunk Per Day	NUMERIC
SELECTOR	Patient or Not(1 or 0)	NOMINAL

In this dataset, there are 7 attributes which predicts the diseases. This dataset contains 345 instances where 145 of them are 1 which is (43.47%) and 200 of them are 0 which is 57.97%. There are no missing value. Selector is only nominal and rest of them are numeric.

3.3 Statistical Analysis:

In our work, we have got two dataset. From those two datasets, in dataset 1 we have 416 liver patients and 167 non-liver patients. Again, from dataset 2 we have 200 liver patients and 145 non-liver patients. Here, we selected 70% data to train and 30% data to test. We run the program using all features using SVM, Naïve Bayes, KNN, and Decision Tree and predicted liver diseases. Then we ran the feature selection process using WEKA**.** After selecting the important feature we again run the program using SVM, Naïve Bayes, KNN, and Decision Tree for the prediction of liver diseases. How we proposed our model is given below using flowchart.

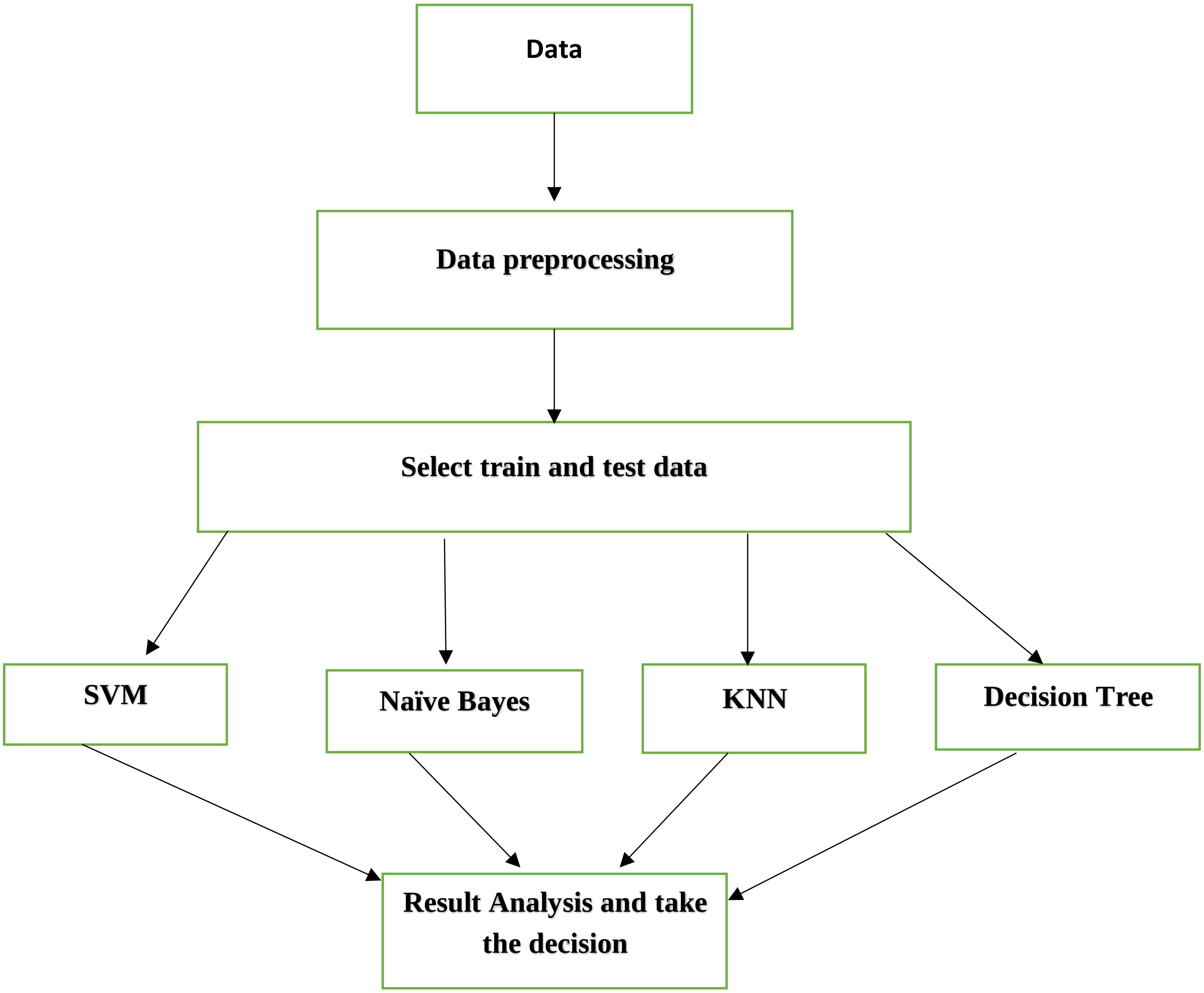

Data
Data preprocessing
Select train and test data
SVM
Naïve Bayes
KNN
Decision Tree
Result Analysis and take the decision

Figure 3.3.1: Architecture of proposed model using all attributes

How we have done our research that is shorty shown in the figure. That is our 1st step. In this step we used all of our attributes. We proposed the model for both of the datasets.

Here is another proposed model for the important attributes:

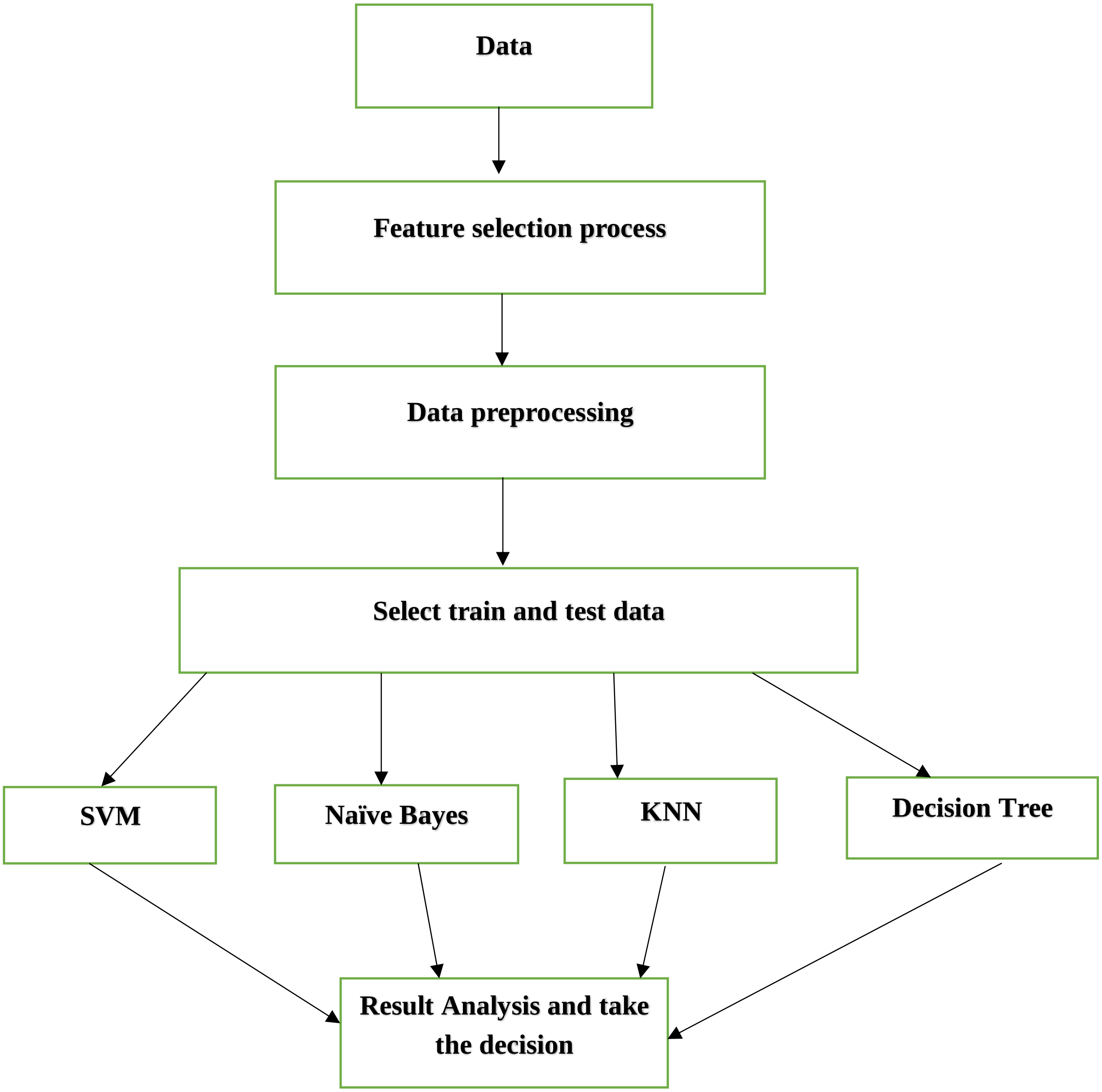

Data
Feature selection process
Data preprocessing
Select train and test data
SVM
Naïve Bayes
KNN
Decision Tree
Result Analysis and take the decision

Figure 3.3.2: Architecture of proposed model after selecting important attributes

This is our 2nd step. Here, the change in the model is feature selection process. In this process we had to find out the important features from those two datasets. Then we applied the algorithm and predicted.

3.4 Feature Selection Process:

It’s a long process [14]. We had to check every angle for finding the right and important set of features so that our accuracy can improve. We have applied both WRAPPER and FILTER method for finding the perfect set of features. For this, we have done our feature selection process using WEKA. [22]

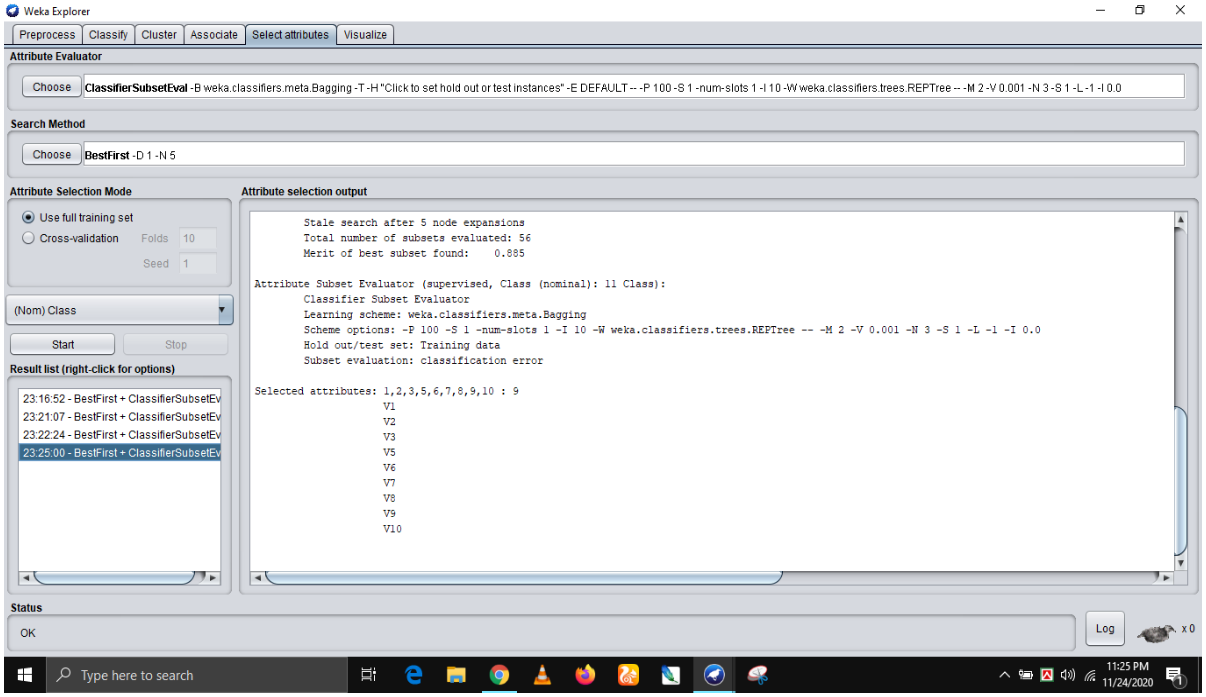
Feature selection is process where we try to reduce the number of input variables used to train a Machine Learning model so that we can improve the accuracy. Reducing the number of redundant values and less important input variables, we can improve the accuracy. As well as fewer predictors needed less time to compute. Statistical-based feature selection method involves evaluating the relationship between each input variable and target variable using statistics and selecting those important attributes that have the strongest relationship with the outcome variable. This is a fast and effective method although the statistical measures depends on the both input and target data. It’s a very challenging process for machine learning. There are three types of feature selection method, (i) Wrapper methods (ii) Filter methods and (iii) Embedded methods. In this paper we have done our work using Wrapper and filter methods.

3.4.1 Feature selection of Dataset 1:

Wrapper Method

In wrapper methods we have used a specific machine learning algorithm that we were trying to fit on a given datasets. [15] It follows a greedy search approach by evaluating all possible combinations of features against the evaluation criterion. It uses cross validation. It works so fast for a dataset with many features. There is a high chances for overfitting because it involves training of machine learning models with different combination of features. It’s computationally expensive.

After processing feature selection process for wrapper method on Dataset 1, we have got some set of attributes which is giver below:



Attribute Evaluator: ClassifierSubsetEval -B weka.classifiers.meta.Bagging -T -H "Click to set hold out or test instances" -E DEFAULT -- -P 100 -S 1 -num-slots 1 -I 10 -W weka.classifiers.trees.REPTree -- -M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0

Search Method: BestFirst -D 1 -N 5

Attribute Selection Mode: ● Use full training set ○ Cross-validation Folds 10 Seed 1

(Nom) Class

Result list (right-click for options):

- 23:16:52 - BestFirst + ClassifierSubsetEv
- 23:21:07 - BestFirst + ClassifierSubsetEv
- 23:22:24 - BestFirst + ClassifierSubsetEv
- 23:25:00 - BestFirst + ClassifierSubsetEv

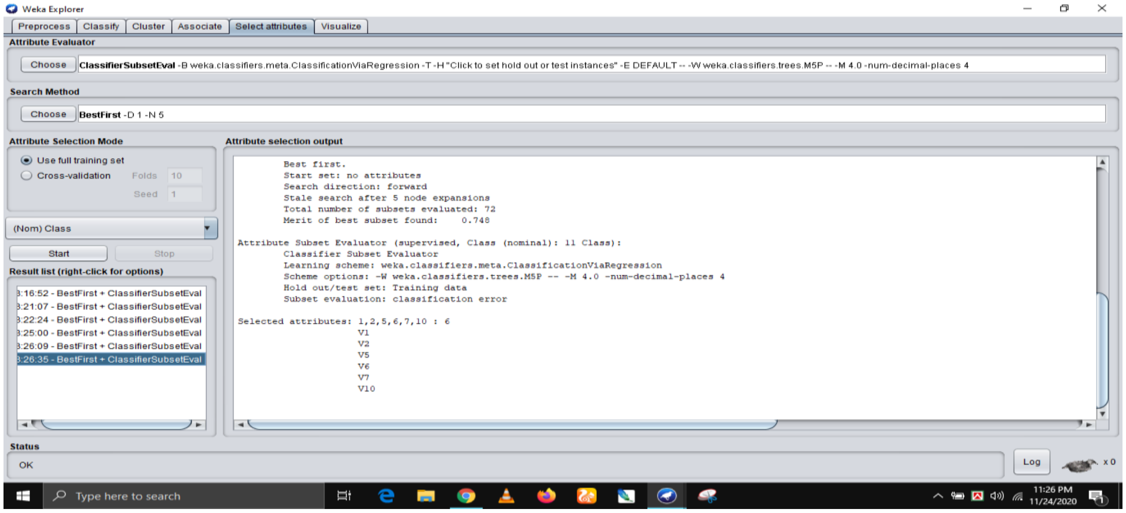
Attribute selection output:

```
        Stale search after 5 node expansions
        Total number of subsets evaluated: 56
        Merit of best subset found:    0.885

Attribute Subset Evaluator (supervised, Class (nominal): 11 Class):
        Classifier Subset Evaluator
        Learning scheme: weka.classifiers.meta.Bagging
        Scheme options: -P 100 -S 1 -num-slots 1 -I 10 -W weka.classifiers.trees.REPTree -- -M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0
        Hold out/test set: Training data
        Subset evaluation: classification error

Selected attributes: 1,2,3,5,6,7,8,9,10 : 9
                     V1
                     V2
                     V3
                     V5
                     V6
                     V7
                     V8
                     V9
                     V10
```

Status: OK

Figure 3.4.1.1: Applied wrapper method for Dataset 1 using Bagging classifier in WEKA

For this process, we have used ClassifierSubsetEval as an attribute evaluator, BestFirst search as a search method, Bagging as a classifier.



Attribute Evaluator: ClassifierSubsetEval -B weka.classifiers.meta.ClassificationViaRegression -T -H "Click to set hold out or test instances" -E DEFAULT -- -W weka.classifiers.trees.M5P -- -M 4.0 -num-decimal-places 4

Search Method: BestFirst -D 1 -N 5

Attribute Selection Mode: ● Use full training set ○ Cross-validation Folds 10 Seed 1

(Nom) Class

Result list (right-click for options):

- 3:16:52 - BestFirst + ClassifierSubsetEval
- 3:21:07 - BestFirst + ClassifierSubsetEval
- 3:22:24 - BestFirst + ClassifierSubsetEval
- 3:25:00 - BestFirst + ClassifierSubsetEval
- 3:26:09 - BestFirst + ClassifierSubsetEval
- 3:26:35 - BestFirst + ClassifierSubsetEval

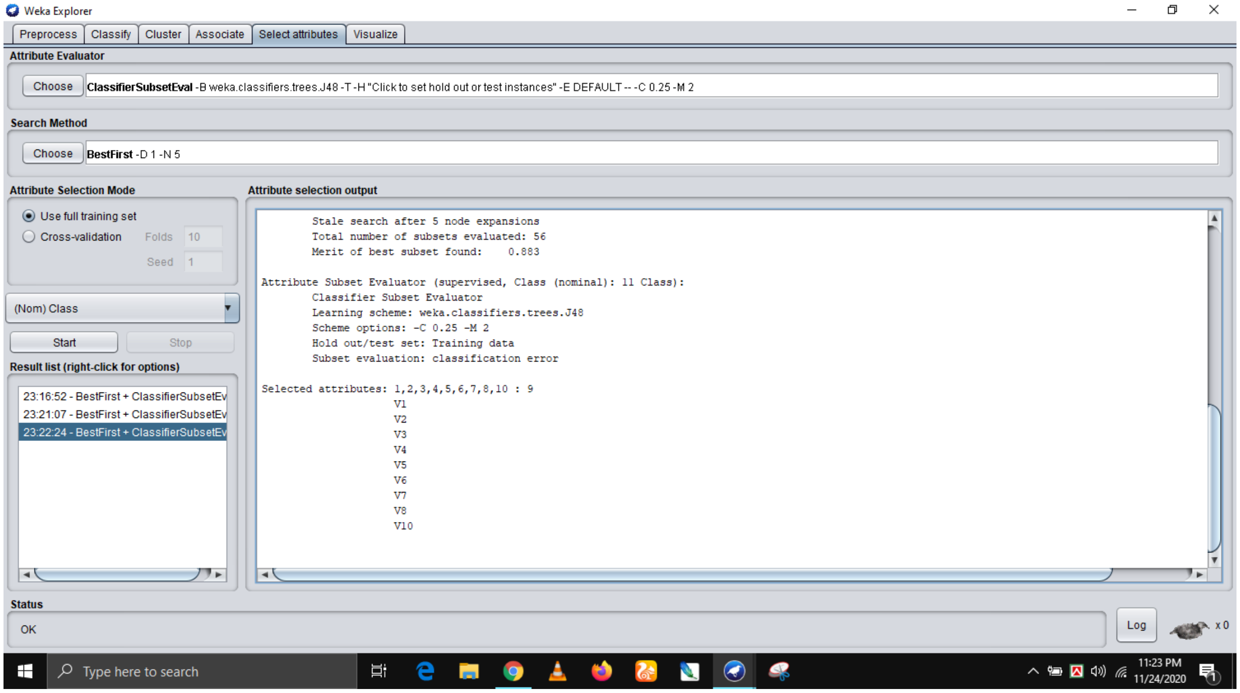
Attribute selection output:

```
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 72
        Merit of best subset found:    0.748

Attribute Subset Evaluator (supervised, Class (nominal): 11 Class):
        Classifier Subset Evaluator
        Learning scheme: weka.classifiers.meta.ClassificationViaRegression
        Scheme options: -W weka.classifiers.trees.M5P -- -M 4.0 -num-decimal-places 4
        Hold out/test set: Training data
        Subset evaluation: classification error

Selected attributes: 1,2,5,6,7,10 : 6
                     V1
                     V2
                     V5
                     V6
                     V7
                     V10
```

Status: OK

Figure 3.4.1.2: Applied wrapper method for Dataset 1 using Classification Via Regression classifier in WEKA

For this process, we have used ClassifierSubsetEval as an attribute evaluator, BestFirst search as a search method, ClassificationViaRegression as a classifier.



Attribute Evaluator

Choose **ClassifierSubsetEval** -B weka.classifiers.trees.J48 -T -H "Click to set hold out or test instances" -E DEFAULT -- -C 0.25 -M 2

Search Method

Choose **BestFirst** -D 1 -N 5

Attribute Selection Mode

● Use full training set
○ Cross-validation Folds 10
Seed 1

(Nom) Class

Result list (right-click for options)

23:16:52 - BestFirst + ClassifierSubsetEv
23:21:07 - BestFirst + ClassifierSubsetEv
23:22:24 - BestFirst + ClassifierSubsetEv

Attribute selection output

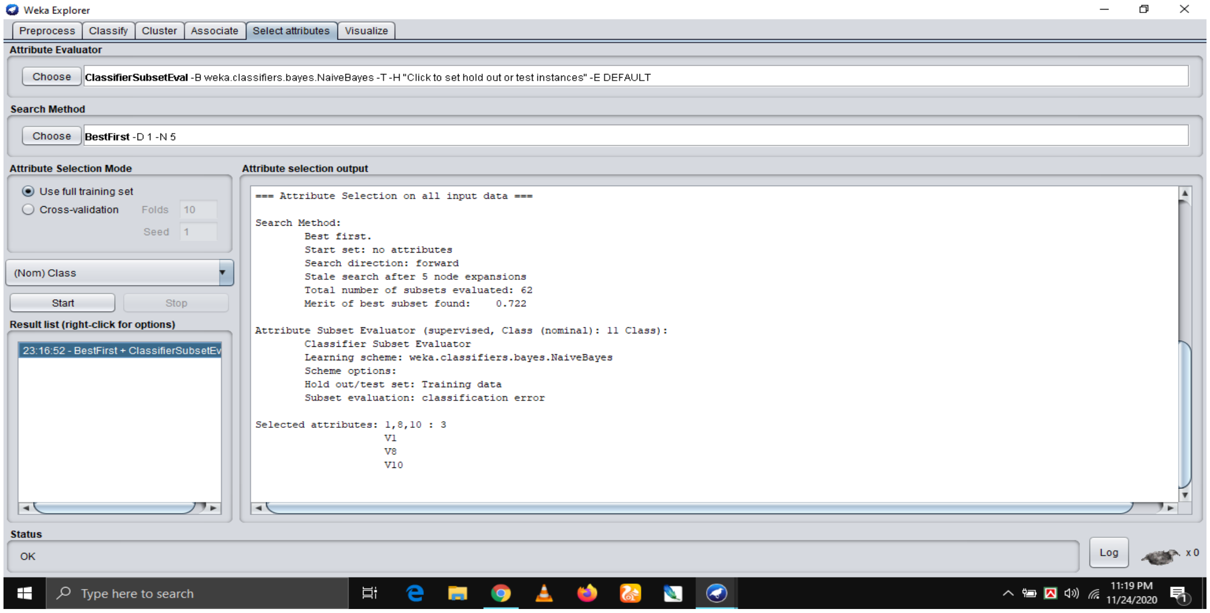
```
        Stale search after 5 node expansions
        Total number of subsets evaluated: 56
        Merit of best subset found:    0.883

Attribute Subset Evaluator (supervised, Class (nominal): 11 Class):
        Classifier Subset Evaluator
        Learning scheme: weka.classifiers.trees.J48
        Scheme options: -C 0.25 -M 2
        Hold out/test set: Training data
        Subset evaluation: classification error

Selected attributes: 1,2,3,4,5,6,7,8,10 : 9
                     V1
                     V2
                     V3
                     V4
                     V5
                     V6
                     V7
                     V8
                     V10
```

Status

OK

Figure 3.4.1.3: Applied wrapper method for Dataset 1 using J48 classifier in WEKA

For this process, we have used ClassifierSubsetEval as an attribute evaluator, BestFirst search as a search method, J48 as a classifier.



Attribute Evaluator

ClassifierSubsetEval -B weka.classifiers.bayes.NaiveBayes -T -H "Click to set hold out or test instances" -E DEFAULT

Search Method

BestFirst -D 1 -N 5

Attribute Selection Mode

● Use full training set
○ Cross-validation Folds 10
Seed 1

(Nom) Class

Result list (right-click for options)

23:16:52 - BestFirst + ClassifierSubsetEv

Attribute selection output

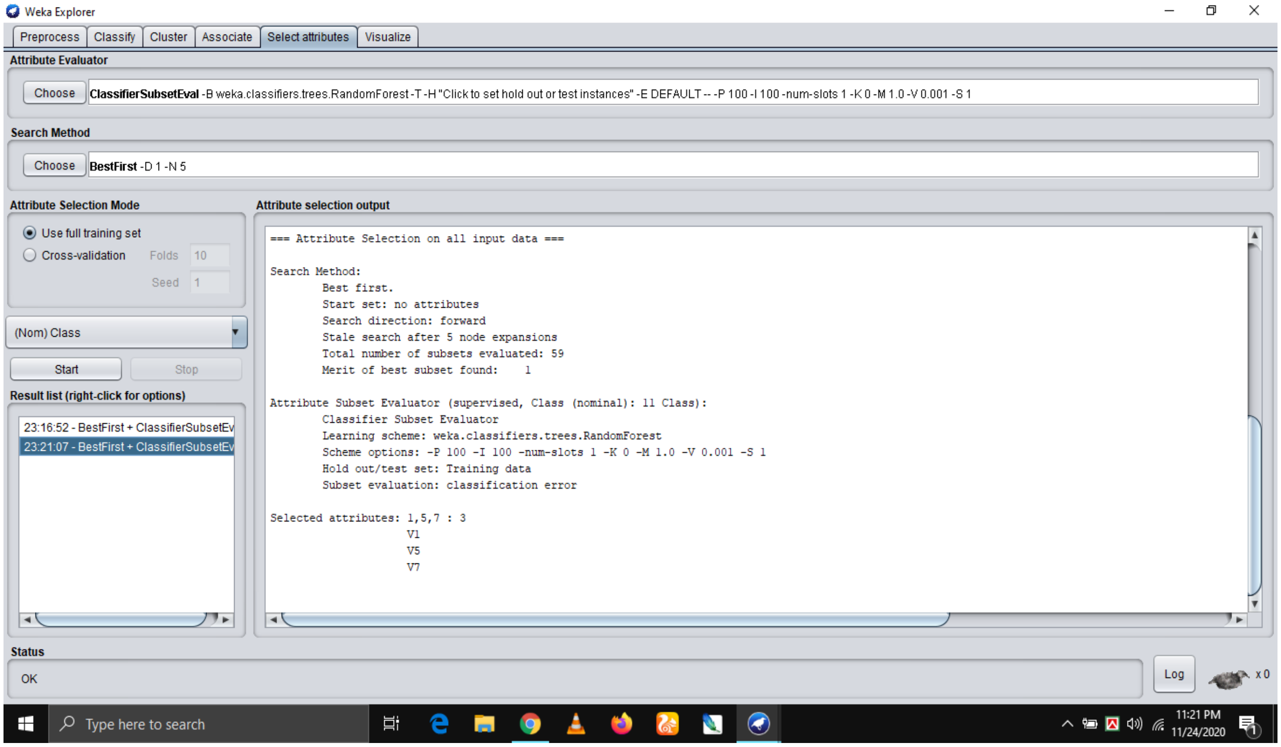
```
=== Attribute Selection on all input data ===

Search Method:
	Best first.
	Start set: no attributes
	Search direction: forward
	Stale search after 5 node expansions
	Total number of subsets evaluated: 62
	Merit of best subset found:    0.722

Attribute Subset Evaluator (supervised, Class (nominal): 11 Class):
	Classifier Subset Evaluator
	Learning scheme: weka.classifiers.bayes.NaiveBayes
	Scheme options:
	Hold out/test set: Training data
	Subset evaluation: classification error

Selected attributes: 1,8,10 : 3
                     V1
                     V8
                     V10
```

Status

OK

Figure 3.4.1.4: Applied wrapper method for Dataset 1 using Naïve Bayes classifier in WEKA

For this process, we have used ClassifierSubsetEval as an attribute evaluator, BestFirst search as a search method, NaïveBayes as a classifier.



Attribute Evaluator

ClassifierSubsetEval -B weka.classifiers.trees.RandomForest -T -H "Click to set hold out or test instances" -E DEFAULT -- -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Search Method

BestFirst -D 1 -N 5

Attribute Selection Mode

● Use full training set
○ Cross-validation Folds 10 Seed 1

(Nom) Class

Result list (right-click for options)

23:16:52 - BestFirst + ClassifierSubsetEv
23:21:07 - BestFirst + ClassifierSubsetEv

Attribute selection output

```
=== Attribute Selection on all input data ===

Search Method:
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 59
        Merit of best subset found:    1

Attribute Subset Evaluator (supervised, Class (nominal): 11 Class):
        Classifier Subset Evaluator
        Learning scheme: weka.classifiers.trees.RandomForest
        Scheme options: -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1
        Hold out/test set: Training data
        Subset evaluation: classification error

Selected attributes: 1,5,7 : 3
                     V1
                     V5
                     V7
```

Status

OK

Figure 3.4.1.5: Applied wrapper method for Dataset 1 using Random Forest classifier in WEKA

For this process, we have used ClassifierSubsetEval as an attribute evaluator, Best First search as a search method, Random Forest as a classifier.

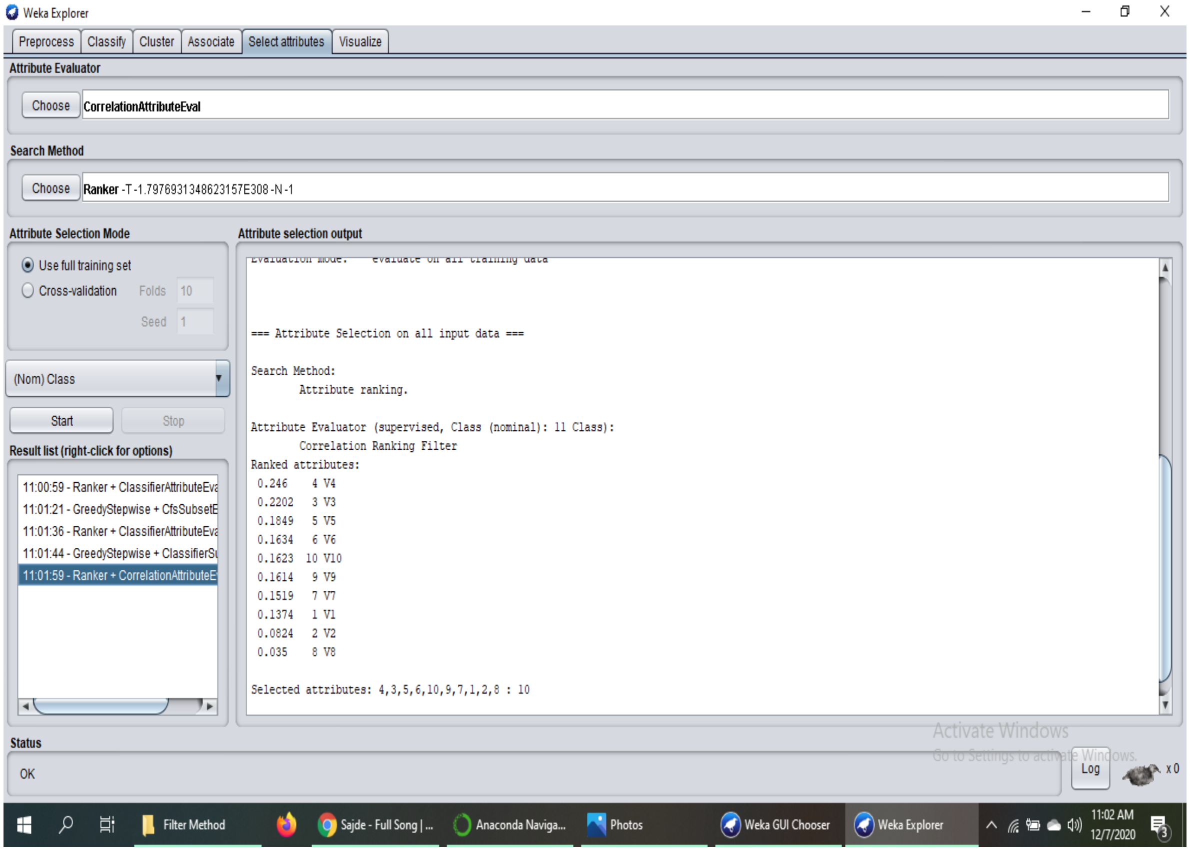
These are the output of using wrapper methods on Dataset 1. We have done this though WEKA. We have here got some set of attributes. In our paper there are five set of group we have got. Those set of attributes are given below:

1. V1,V2,V3,V5,V6,V7,V8,V9,V10 : 9
2. V1,V2,V5,V6,V7,V10 : 6
3. V1,V2,V3,V4,V5,V6,V7,V8,V10 : 9
4. V1 ,V8,V10 : 3
5. V1,V5,V7 : 3

Filter Method:

In filter method we actually ranks the attributes. Generic set of methods which do not incorporate a specific machine learning algorithm. This is a model agnostic, rely entirely on features in the datasets. Filter methods are much faster than wrapper method in terms of time complexity. There is a less chance for overfitting. It uses statistical methods for evaluation of a subset of features.

After processing feature selection process for filter method on Dataset 1, we have got some set of attributes which is giver below:



Attribute Evaluator: CorrelationAttributeEval

Search Method: Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode: ● Use full training set ○ Cross-validation Folds 10 Seed 1

(Nom) Class

Result list (right-click for options)

```
11:00:59 - Ranker + ClassifierAttributeEva
11:01:21 - GreedyStepwise + CfsSubsetE
11:01:36 - Ranker + ClassifierAttributeEva
11:01:44 - GreedyStepwise + ClassifierSu
11:01:59 - Ranker + CorrelationAttributeE
```

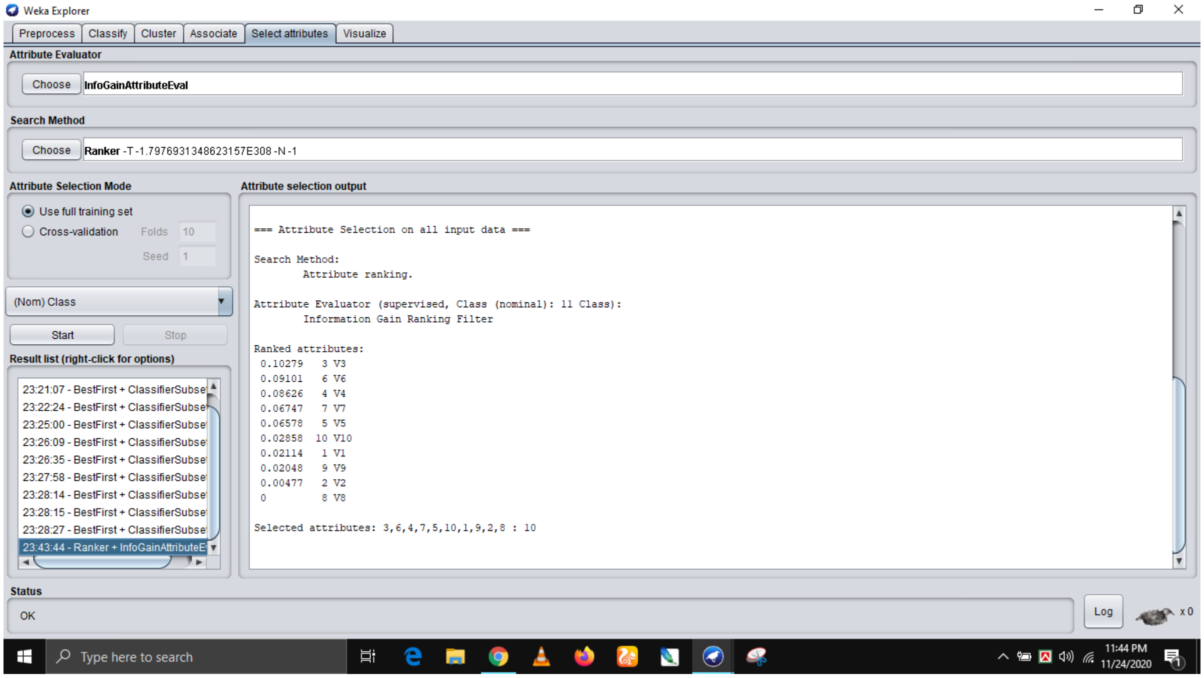
Attribute selection output

```
Evaluation mode:    evaluate on all training data



=== Attribute Selection on all input data ===

Search Method:
	Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 11 Class):
	Correlation Ranking Filter
Ranked attributes:
 0.246    4 V4
 0.2202   3 V3
 0.1849   5 V5
 0.1634   6 V6
 0.1623  10 V10
 0.1614   9 V9
 0.1519   7 V7
 0.1374   1 V1
 0.0824   2 V2
 0.035    8 V8

Selected attributes: 4,3,5,6,10,9,7,1,2,8 : 10
```

Status: OK

Figure 3.4.1.6: Applied filter method for Dataset 1 using Correlation Attribute Eval evaluator in WEKA

For this process, we have used CorrelationAttributeEval as an attribute evaluator, Ranker as a search method.



Attribute Evaluator: InfoGainAttributeEval

Search Method: Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode: ● Use full training set ○ Cross-validation Folds 10 Seed 1

(Nom) Class

Result list (right-click for options)

```
23:21:07 - BestFirst + ClassifierSubse
23:22:24 - BestFirst + ClassifierSubse
23:25:00 - BestFirst + ClassifierSubse
23:26:09 - BestFirst + ClassifierSubse
23:26:35 - BestFirst + ClassifierSubse
23:27:58 - BestFirst + ClassifierSubse
23:28:14 - BestFirst + ClassifierSubse
23:28:15 - BestFirst + ClassifierSubse
23:28:27 - BestFirst + ClassifierSubse
23:43:44 - Ranker + InfoGainAttributeE
```

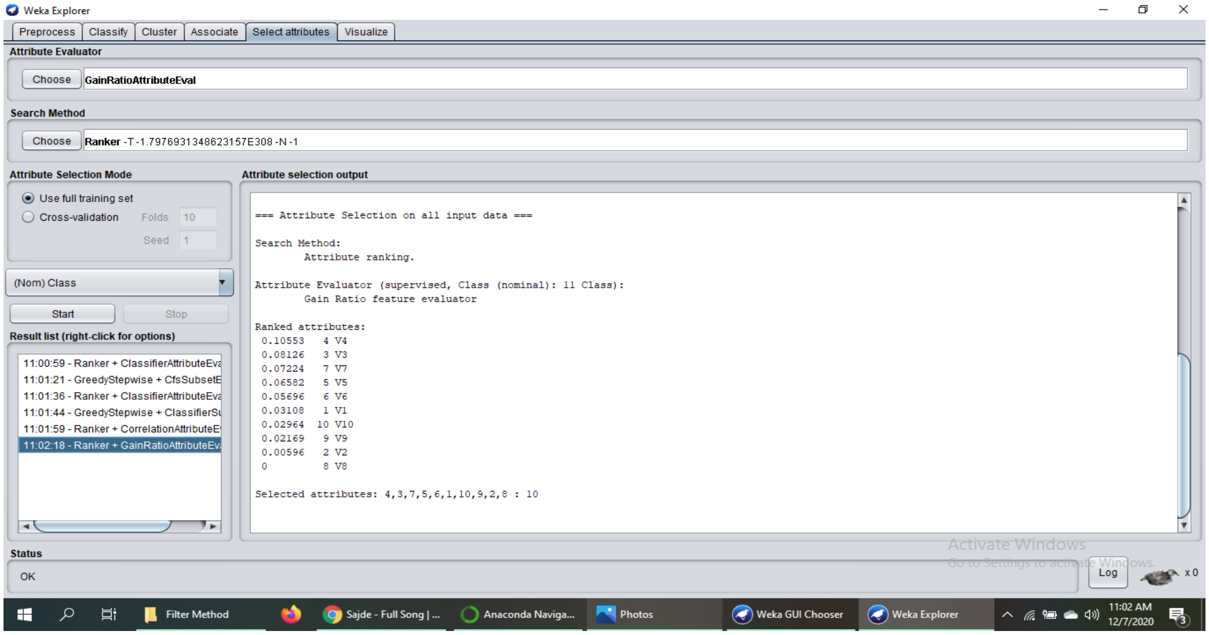
Attribute selection output

```
=== Attribute Selection on all input data ===

Search Method:
	Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 11 Class):
	Information Gain Ranking Filter

Ranked attributes:
 0.10279   3 V3
 0.09101   6 V6
 0.08626   4 V4
 0.06747   7 V7
 0.06578   5 V5
 0.02858  10 V10
 0.02114   1 V1
 0.02048   9 V9
 0.00477   2 V2
 0         8 V8

Selected attributes: 3,6,4,7,5,10,1,9,2,8 : 10
```

Status: OK

Figure 3.4.1.7: Applied filter method for Dataset 1 using Info Gain Attribute Eval evaluator in WEKA

For this process, we have used InfoGainAttributeEval as an attribute evaluator, Ranker as a search method.



Attribute Evaluator
GainRatioAttributeEval

Search Method
Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode
● Use full training set
○ Cross-validation Folds 10 Seed 1

(Nom) Class

Result list (right-click for options)
11:00:59 - Ranker + ClassifierAttributeEva
11:01:21 - GreedyStepwise + CfsSubsetE
11:01:36 - Ranker + ClassifierAttributeEva
11:01:44 - GreedyStepwise + ClassifierSu
11:01:59 - Ranker + CorrelationAttributeE
11:02:18 - Ranker + GainRatioAttributeEv

Attribute selection output

```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 11 Class):
        Gain Ratio feature evaluator

Ranked attributes:
 0.10553   4 V4
 0.08126   3 V3
 0.07224   7 V7
 0.06582   5 V5
 0.05696   6 V6
 0.03108   1 V1
 0.02964  10 V10
 0.02169   9 V9
 0.00596   2 V2
 0         8 V8

Selected attributes: 4,3,7,5,6,1,10,9,2,8 : 10
```

Status
OK

Figure 3.4.1.8: Applied filter method for Dataset 1 using Gain Ratio Attribute Eval evaluator in WEKA

For this process, we have used GainRatioAttributeEval as an attribute evaluator, Ranker as a search method.

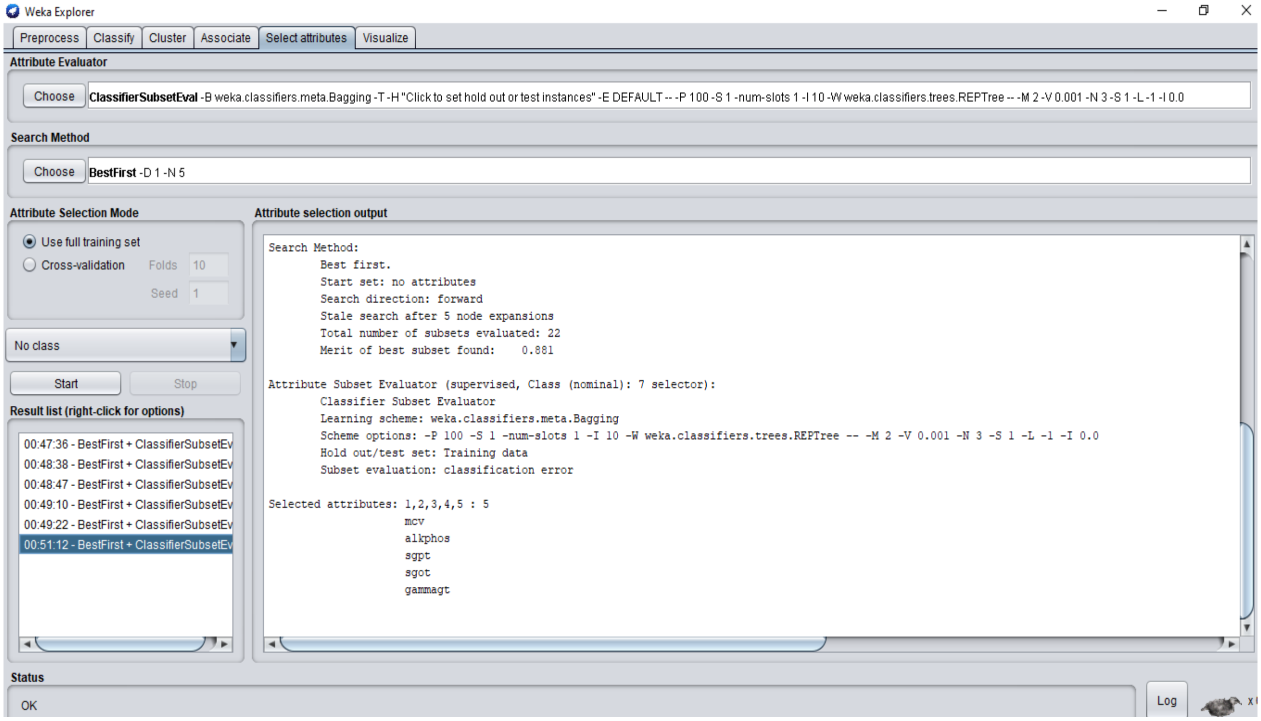
These are the output of using filter methods on Dataset 1. We have done this though WEKA. We have here selected some set of attributes using the ranker. In our paper there are three set of attributes we have got. Those set of attributes are given below:

1. V3,V4,V5,V6,V7,V9,V10 : 7 (Selected attributes according to the rank)
2. V3,V4,V5,V6,V7 : 5 (Selected attributes according to the rank)
3. V3,V4,V5,V7 : 4 (Selected attributes according to the rank)

3.4.2 Feature selection of Dataset 2:

In wrapper methods we use a specific machine learning algorithm that we are trying to fit on a given datasets. [15] It follows a greedy search approach by evaluating all possible combinations of features against the evaluation criterion. It uses cross validation. It works so fast for a dataset with many features. There is a high chances for overfitting because it involves training of machine learning models with different combination of features. It’s computationally expensive.

After processing feature selection process for wrapper method[15] on Dataset 2, we have got some set of attributes which is giver below:



Attribute Evaluator

ClassifierSubsetEval -B weka.classifiers.meta.Bagging -T -H "Click to set hold out or test instances" -E DEFAULT -- -P 100 -S 1 -num-slots 1 -I 10 -W weka.classifiers.trees.REPTree -- -M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0

Search Method

BestFirst -D 1 -N 5

Attribute Selection Mode

● Use full training set
○ Cross-validation Folds 10
Seed 1

No class

Result list (right-click for options)

```
00:47:36 - BestFirst + ClassifierSubsetEv
00:48:38 - BestFirst + ClassifierSubsetEv
00:48:47 - BestFirst + ClassifierSubsetEv
00:49:10 - BestFirst + ClassifierSubsetEv
00:49:22 - BestFirst + ClassifierSubsetEv
00:51:12 - BestFirst + ClassifierSubsetEv
```

Attribute selection output

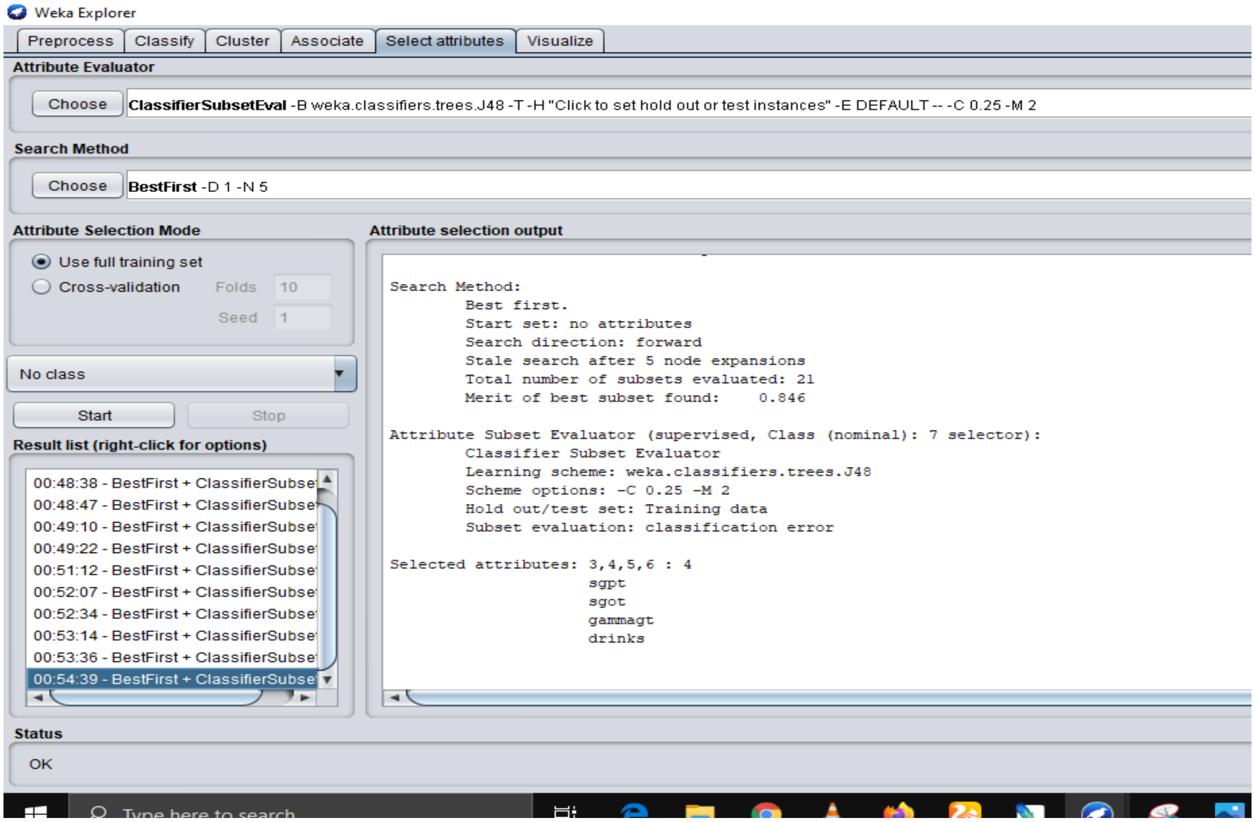
```
Search Method:
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 22
        Merit of best subset found:    0.881

Attribute Subset Evaluator (supervised, Class (nominal): 7 selector):
        Classifier Subset Evaluator
        Learning scheme: weka.classifiers.meta.Bagging
        Scheme options: -P 100 -S 1 -num-slots 1 -I 10 -W weka.classifiers.trees.REPTree -- -M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0
        Hold out/test set: Training data
        Subset evaluation: classification error

Selected attributes: 1,2,3,4,5 : 5
                     mcv
                     alkphos
                     sgpt
                     sgot
                     gammagt
```

Status

OK

Figure 3.4.2.1: Applied wrapper method for Dataset 2 using Bagging classifier in WEKA

For this process, we have used ClassifierSubsetEval as an attribute evaluator, BestFirst search as a search method, Bagging as a classifier.



ClassifierSubsetEval -B weka.classifiers.trees.J48 -T -H "Click to set hold out or test instances" -E DEFAULT -- -C 0.25 -M 2

BestFirst -D 1 -N 5

● Use full training set
○ Cross-validation Folds 10 Seed 1

No class

Result list:

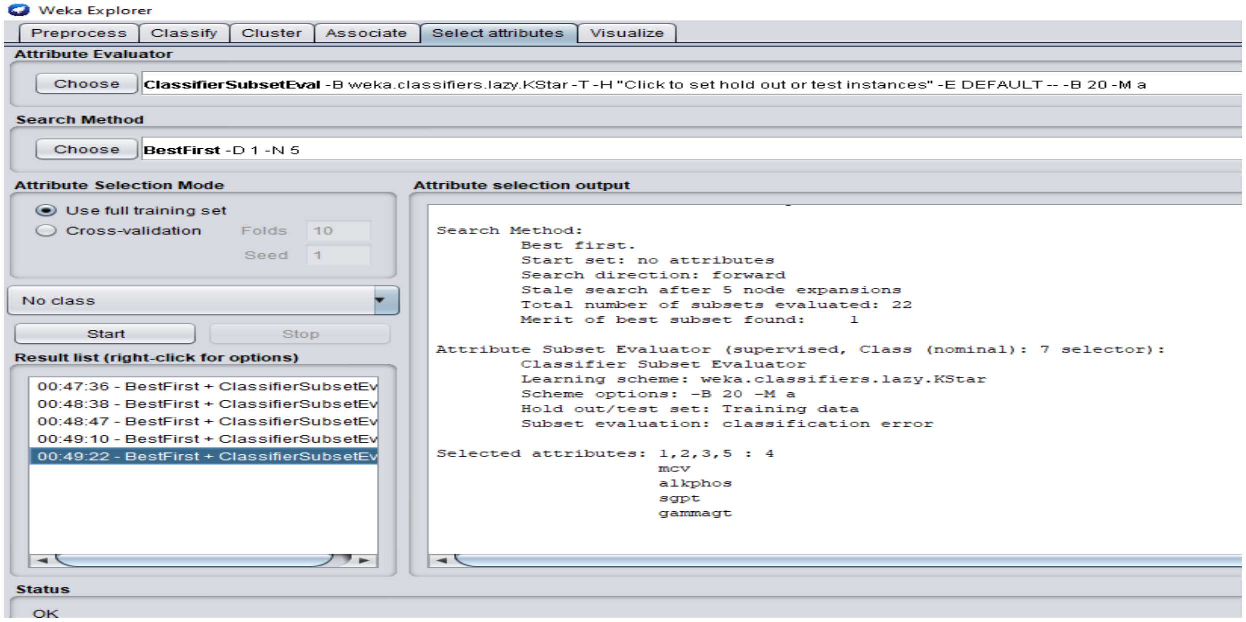
```
00:48:38 - BestFirst + ClassifierSubse
00:48:47 - BestFirst + ClassifierSubse
00:49:10 - BestFirst + ClassifierSubse
00:49:22 - BestFirst + ClassifierSubse
00:51:12 - BestFirst + ClassifierSubse
00:52:07 - BestFirst + ClassifierSubse
00:52:34 - BestFirst + ClassifierSubse
00:53:14 - BestFirst + ClassifierSubse
00:53:36 - BestFirst + ClassifierSubse
00:54:39 - BestFirst + ClassifierSubse
```

```
Search Method:
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 21
        Merit of best subset found:    0.846

Attribute Subset Evaluator (supervised, Class (nominal): 7 selector):
        Classifier Subset Evaluator
        Learning scheme: weka.classifiers.trees.J48
        Scheme options: -C 0.25 -M 2
        Hold out/test set: Training data
        Subset evaluation: classification error

Selected attributes: 3,4,5,6 : 4
                     sgpt
                     sgot
                     gammagt
                     drinks
```

Status: OK

Figure 3.4.2.2: Applied wrapper method for Dataset 2 using J48 classifier in WEKA

For this process, we have used ClassifierSubsetEval as an attribute evaluator, BestFirst search as a search method, J48 as a classifier.



Attribute Evaluator: **ClassifierSubsetEval** -B weka.classifiers.lazy.KStar -T -H "Click to set hold out or test instances" -E DEFAULT -- -B 20 -M a

Search Method: **BestFirst** -D 1 -N 5

Attribute Selection Mode:
● Use full training set
○ Cross-validation Folds 10 Seed 1

No class

Result list (right-click for options)

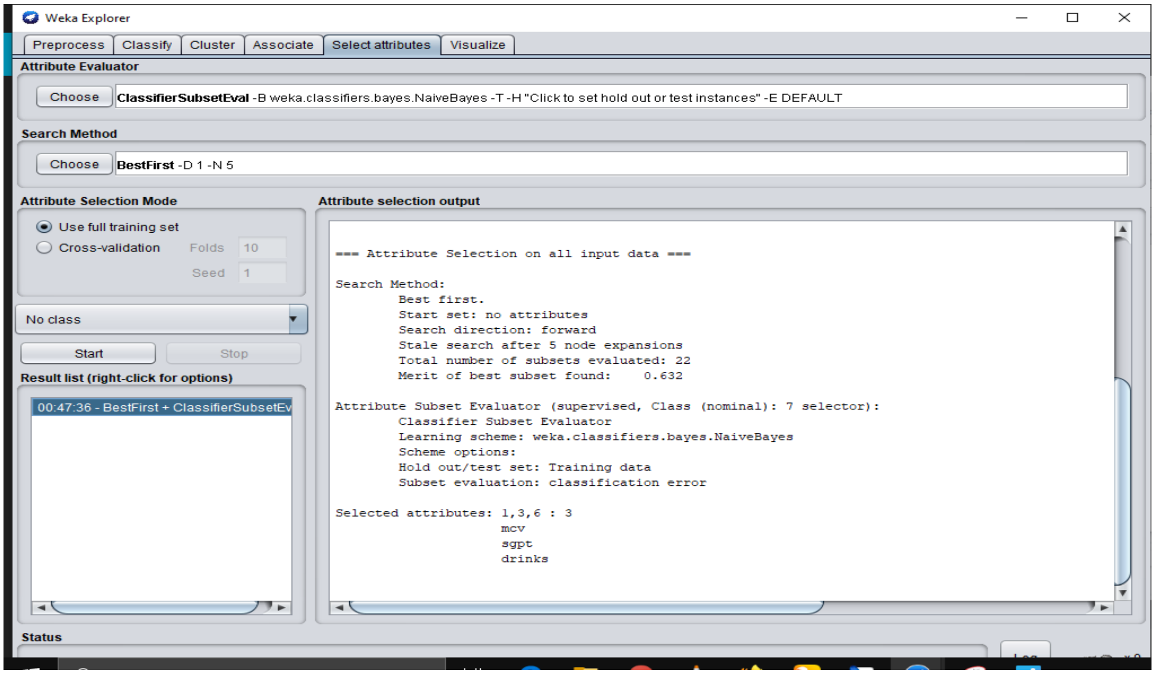
```
00:47:36 - BestFirst + ClassifierSubsetEv
00:48:38 - BestFirst + ClassifierSubsetEv
00:48:47 - BestFirst + ClassifierSubsetEv
00:49:10 - BestFirst + ClassifierSubsetEv
00:49:22 - BestFirst + ClassifierSubsetEv
```

Attribute selection output

```
Search Method:
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 22
        Merit of best subset found:    1

Attribute Subset Evaluator (supervised, Class (nominal): 7 selector):
        Classifier Subset Evaluator
        Learning scheme: weka.classifiers.lazy.KStar
        Scheme options: -B 20 -M a
        Hold out/test set: Training data
        Subset evaluation: classification error

Selected attributes: 1,2,3,5 : 4
                     mcv
                     alkphos
                     sgpt
                     gammagt
```

Figure 3.4.2.3: Applied wrapper method for Dataset 2 using KStar classifier in WEKA

For this process, we have used ClassifierSubsetEval as an attribute evaluator, BestFirst search as a search method, KStar as a classifier.



Attribute Evaluator: **ClassifierSubsetEval** -B weka.classifiers.bayes.NaiveBayes -T -H "Click to set hold out or test instances" -E DEFAULT

Search Method: **BestFirst** -D 1 -N 5

Attribute Selection Mode:
● Use full training set
○ Cross-validation Folds 10 Seed 1

No class

Result list (right-click for options)

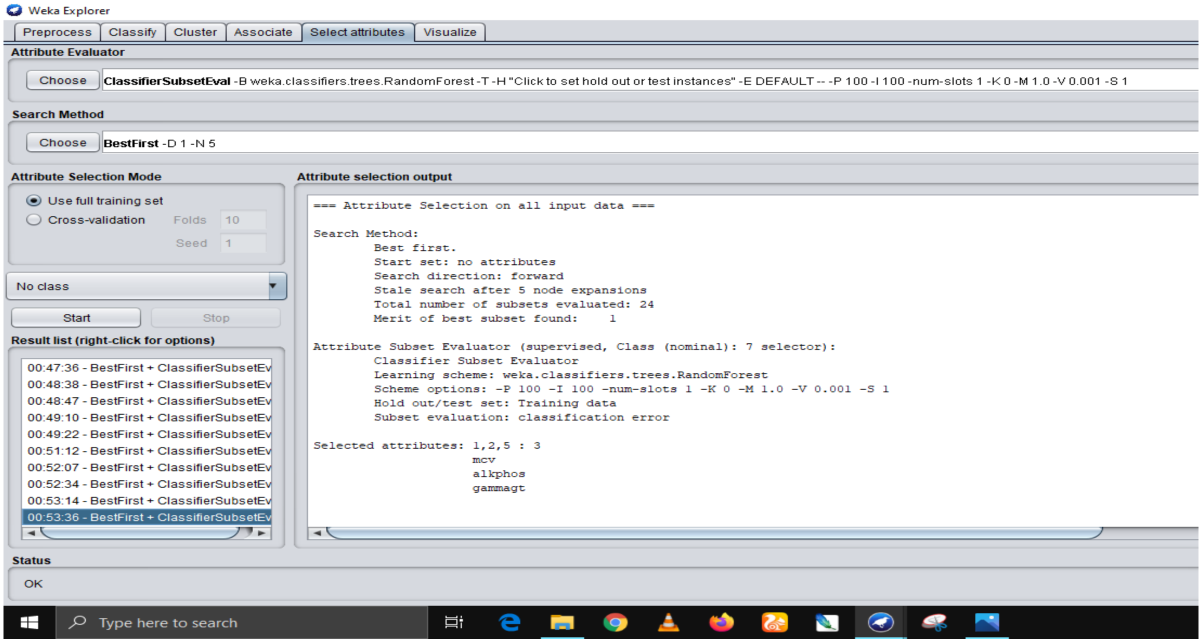
```
00:47:36 - BestFirst + ClassifierSubsetEv
```

Attribute selection output

```
=== Attribute Selection on all input data ===

Search Method:
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 22
        Merit of best subset found:    0.632

Attribute Subset Evaluator (supervised, Class (nominal): 7 selector):
        Classifier Subset Evaluator
        Learning scheme: weka.classifiers.bayes.NaiveBayes
        Scheme options:
        Hold out/test set: Training data
        Subset evaluation: classification error

Selected attributes: 1,3,6 : 3
                     mcv
                     sgpt
                     drinks
```

Figure 3.4.2.4: Applied wrapper method for Dataset 2 using Naïve Bayes classifier in WEKA

For this process, we have used ClassifierSubsetEval as an attribute evaluator, BestFirst search as a search method, NaïveBayes as a classifier.



Attribute Evaluator

ClassifierSubsetEval -B weka.classifiers.trees.RandomForest -T -H "Click to set hold out or test instances" -E DEFAULT -- -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Search Method

BestFirst -D 1 -N 5

Attribute Selection Mode

● Use full training set
○ Cross-validation Folds 10 Seed 1

No class

Result list (right-click for options)

00:47:36 - BestFirst + ClassifierSubsetEv
00:48:38 - BestFirst + ClassifierSubsetEv
00:48:47 - BestFirst + ClassifierSubsetEv
00:49:10 - BestFirst + ClassifierSubsetEv
00:49:22 - BestFirst + ClassifierSubsetEv
00:51:12 - BestFirst + ClassifierSubsetEv
00:52:07 - BestFirst + ClassifierSubsetEv
00:52:34 - BestFirst + ClassifierSubsetEv
00:53:14 - BestFirst + ClassifierSubsetEv
00:53:36 - BestFirst + ClassifierSubsetEv

Attribute selection output

```
=== Attribute Selection on all input data ===

Search Method:
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 24
        Merit of best subset found:    1

Attribute Subset Evaluator (supervised, Class (nominal): 7 selector):
        Classifier Subset Evaluator
        Learning scheme: weka.classifiers.trees.RandomForest
        Scheme options: -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1
        Hold out/test set: Training data
        Subset evaluation: classification error

Selected attributes: 1,2,5 : 3
                     mcv
                     alkphos
                     gammagt
```

Status

OK

Figure 3.4.2.5: Applied wrapper method for Dataset 2 using Random Forest classifier in WEKA

For this process, we have used ClassifierSubsetEval as an attribute evaluator, BestFirst search as a search method, RandomForest as a classifier.

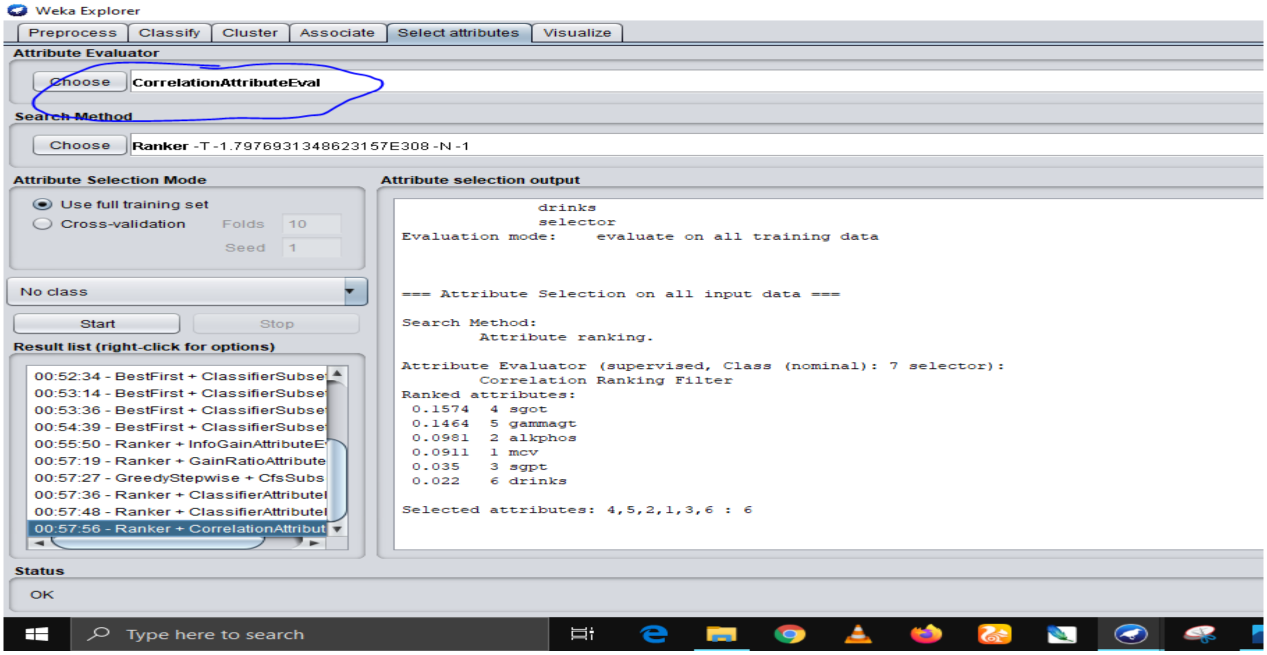
These are the output of using wrapper methods on Dataset 2. We have done this though WEKA. We have here got some set of attributes. In our paper there are five set of group we have got for wrapper methods. Those set of attributes are given below:

1. MCV, ALKPHOS, SGPT, SGOT, GAMMAGT : 5
2. SGPT, SGOT, GAMMAGT, DRINKS: 4
3. MCV, ALKPHOS, SGPT, GAMMAGT : 4
4. MCV, SGPT, DRINKS : 3
5. MCV, ALKPHOS, GAMMAGT : 3

Filter Method:

In filter method we actually rank the attributes. Generic set of methods which do not incorporate a specific machine learning algorithm. This is a model agnostic, rely entirely on features in the datasets. Filter methods are much faster than wrapper method in terms of time complexity. There is a less chance for overfitting. It uses statistical methods for evaluation of a subset of features.

After processing feature selection process for filter method[16] on Dataset 2, we have got some set of attributes which is giver below:



Attribute Evaluator

Choose CorrelationAttributeEval

Search Method

Choose Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode

● Use full training set
○ Cross-validation Folds 10 Seed 1

No class

Result list (right-click for options)

```
00:52:34 - BestFirst + ClassifierSubse
00:53:14 - BestFirst + ClassifierSubse
00:53:36 - BestFirst + ClassifierSubse
00:54:39 - BestFirst + ClassifierSubse
00:55:50 - Ranker + InfoGainAttributeE
00:57:19 - Ranker + GainRatioAttribute
00:57:27 - GreedyStepwise + CfsSubs
00:57:36 - Ranker + ClassifierAttributel
00:57:48 - Ranker + ClassifierAttributel
00:57:56 - Ranker + CorrelationAttribut
```

Attribute selection output

```
                     drinks
                     selector
Evaluation mode:     evaluate on all training data



=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 7 selector):
        Correlation Ranking Filter
Ranked attributes:
 0.1574   4 sgot
 0.1464   5 gammagt
 0.0981   2 alkphos
 0.0911   1 mcv
 0.035    3 sgpt
 0.022    6 drinks

Selected attributes: 4,5,2,1,3,6 : 6
```

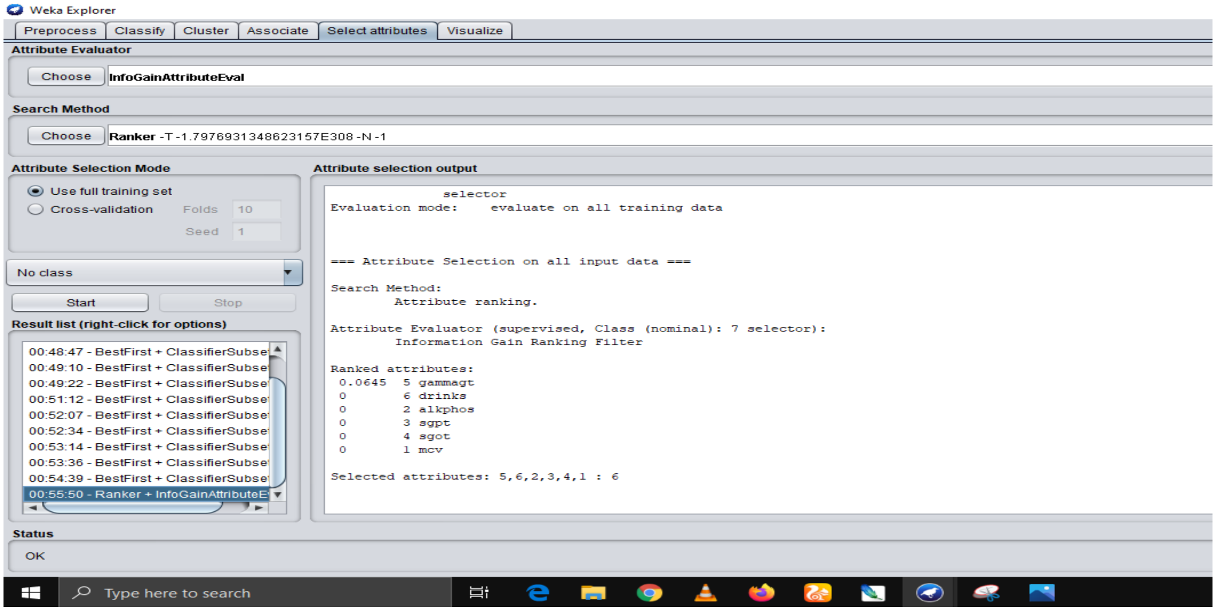
Status

OK

Type here to search

Figure 3.4.2.6: Applied filter method for Dataset 2 using Correlation Attribute evaluator in WEKA

For this process, we have used CorrelationAttributeEval as an attribute evaluator, Ranker as a search method.



Attribute Evaluator

Choose InfoGainAttributeEval

Search Method

Choose Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode

● Use full training set
○ Cross-validation Folds 10 Seed 1

No class

Result list (right-click for options)

```
00:48:47 - BestFirst + ClassifierSubse
00:49:10 - BestFirst + ClassifierSubse
00:49:22 - BestFirst + ClassifierSubse
00:51:12 - BestFirst + ClassifierSubse
00:52:07 - BestFirst + ClassifierSubse
00:52:34 - BestFirst + ClassifierSubse
00:53:14 - BestFirst + ClassifierSubse
00:53:36 - BestFirst + ClassifierSubse
00:54:39 - BestFirst + ClassifierSubse
00:55:50 - Ranker + InfoGainAttributeE
```

Attribute selection output

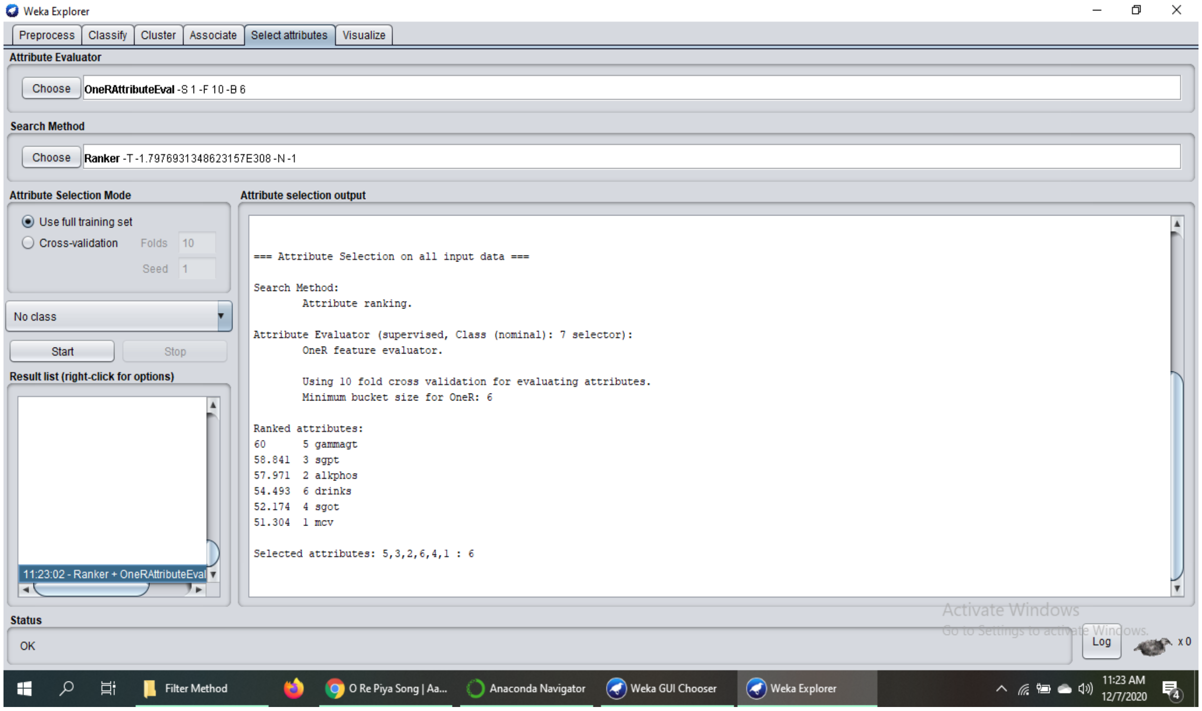
```
               selector
Evaluation mode:    evaluate on all training data



=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 7 selector):
        Information Gain Ranking Filter

Ranked attributes:
 0.0645  5 gammagt
 0       6 drinks
 0       2 alkphos
 0       3 sgpt
 0       4 sgot
 0       1 mcv

Selected attributes: 5,6,2,3,4,1 : 6
```

Status

OK

Figure 3.4.2.7: Applied filter method for Dataset 2 using Info Gain Attribute Eval evaluator in WEKA

For this process, we have used InfoGainAttributeEval as an attribute evaluator, Ranker as a search method.



Attribute Evaluator

Choose OneRAttributeEval -S 1 -F 10 -B 6

Search Method

Choose Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode

● Use full training set
○ Cross-validation Folds 10 Seed 1

No class

Result list (right-click for options)

```
11:23:02 - Ranker + OneRAttributeEval
```

Attribute selection output

```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 7 selector):
        OneR feature evaluator.

        Using 10 fold cross validation for evaluating attributes.
        Minimum bucket size for OneR: 6

Ranked attributes:
60      5 gammagt
58.841  3 sgpt
57.971  2 alkphos
54.493  6 drinks
52.174  4 sgot
51.304  1 mcv

Selected attributes: 5,3,2,6,4,1 : 6
```

Status

OK

Figure 3.4.2.8: Applied filter method for Dataset 2 using One R Attribute evaluator in WEKA

For this process, we have used OneRAttributeEval as an attribute evaluator, Ranker as a search method.

These are the output of using filter methods on Dataset 2. We have done this though WEKA. We have here selected some set of attributes using the ranker. In our paper there are three set of attributes we have got. Those set of attributes are given below:

1. MCV, ALKPHOS, SGOT, GAMMAGT : 4 (Selected attributes according to the rank)
2. GAMMAGT : 1 (Selected attributes according to the rank)
3. ALKPHOS, SGPT, GAMMAGT : 3 (Selected attributes according to the rank)
4. MCV, SGPT, SGOT, GAMMAGT, DRINKS : 5 (Selected attributes according to the rank)

3.5 Algorithms:

The frame of machine learning calculations is rising exponentially. Everything became easier for us because of algorithms in machine learning. A machine can discover the interior information and make a decision using algorithms without the experts. In this portion we will discuss about algorithms we have used to finding the accuracy. We have used four algorithm. SVM, Naïve Bayes, KNN, Decision Tree.

The algorithms details are giver below:

Support Vector Machine (SVM):

A Support Vector Machine (SVM)[18] is a formally characterized discriminative classifier by hyper plane isolation. The marked preparation of information at the end of the day, the equation yields an ideal hyper plane that arranges new ones (managed take-in), about precedents. This hyper plane in two-dimensional space is a line separating a plane into two. Parts where the numeric info factors (x) are on either side of each class in your A n-dimensional space is formed by knowledge (sections). For example, on the off chance, this would frame a two-dimensional space if you had two information variables. A hyper plane is a line that divides the variable space of the data. A hyper plane in SVM is chosen to better distinguish the focuses by their class, either class 0, in the info variable space Class or 1. You can imagine this as a line in two measurements and we can expect it to the majority of the focus of our knowledge can be absolutely separated by this section.

For example:

B0 + (B1*X1) + (B2*X2) = 0……………. (1)

Here are the coefficients (B1 and B2) from equation 1 that specify the incline of the line and the incline of the line. The learning calculation finds the catch (B0), and the two details are X1 and X2 info factors.

Naïve Bayes:

A naive Bayesian classifier[19] is essentially a theorem on Bayes with assumptions of independence between predictors. It is very simple to construct a Naïve Bayesian model and can be implemented for very large datasets. The Naïve Bayesian classifier also performs better than more advanced classification techniques. Posterior probability, From P(c), P(x), and P(x|c), P(c|x) is determined. The effect on a given class (c) of the value of a predictor (x) is independent of other predictors' values. This statement is called the conditional independence of class.

K-Nearest Neighbor (KNN):

In the controlled adapting category of calculations, KNN [20] drop. This means, casually, that we are provided a marked dataset consisting of perceptions (x, y) being prepared and may want to capture the relation between x and y. Much more formally, we are likely to take on a skill h: X-->Y with the objective that x, h(x) will unquestionably be granted a hidden perception. Foresee the corresponding yield y. Similarly, the KNN classifier is a non-parametric classifier and Calculation of opportunity-based learning. Non-parametric means that it does not produce an unambiguous presumptions concerning the utilitarian form of h, keeping away from the risks of miss modeling the fundamental dispersion of data. For example, assume our data is the learning model we use to accept a Gaussian is unusually non-Gaussian. All things considered, our estimate would make the prediction low to a great degree. Case-based learning means that a model is not specifically taken up in our estimation. Rather it preserves the events of training that are thus used as' learning' for Phase of forecast. Solidly, this means only when a query is made to our database (that is if we request that it foresee a name provided with information), the calculation will be used for the planning of opportunities to release a response. In the arrangement setting, the measurement of the K-closest neighbor basically comes down to the shaping a greater portion of the vote between the most comparable K events to a given one "concealed" perception. Similitude is defined by a metric of separation between two focusing on details. The Euclidean separation given by Euclid's is a prominent decision.

$d(x,x')=\sqrt{}(x_1-x'_1)^2+(x_2-x'_2)^2+\ldots+(x_n-x'_n)^2\ldots\ldots\ldots\ldots\ldots\ldots(2)$

From equation 2, a positive integer K is given here an inconspicuous perception x. The following two steps are carried out by the KNN classifier and a resemblance metric d:

This goes through the whole dataset that registers d between x and each perception of preparation. In the preparation data that is closest to x the package, we'll call the K focuses. Note, K is odd to counteract tie conditions. The isolation of another data point from all other plans is ultimately calculated. Focusing on details, the distinction may be of some type, e.g. Manhattan or Euclidean and so on. It chooses the K-closest focus of data at that point, where K can be any number. Now it is possible to think about how to select the K variable and what its consequences are. Classifier on this, all things considered, like most calculations for machine learning, the K in KNN is a hyper parameter that you have to select as a developer with the ultimate goal of having the most suitable match for the knowledge index. Of course, it can be considered that K controls the state of the limit of selection we discussed before. At the point where K is small, we limit the area of a given forecast and drive our forecast "more visually impaired". A bit of an opportunity for the most adaptable fit is given by K, which will have low inclination but high fluctuation. Graphically, it will be more jagged, our option limit. A higher K, on the other hand in each forecast, the midpoint of more voters. Therefore, exceptions are higher. Bigger K's estimates would have smoother limits of option which means that change is reduced but broadened predisposition.

The group quantity of K must be solved beforehand. Its downside is that for each sprint, the result is identical; as the subsequent bunches depend on the irregular underlying assignments.

We never know the true bunch using similar data, assuming it is inputted in an alternative request. If the amount of quantity is given, it could deliver a diverse group knowledge is scarce. As far as we know, datasets for the KNN are very much arranged building show. Since KNN is a non-parametric measurement, parameters for the model will not be obtained. The function of KNN () restores a vector containing an element of the test Set Characterizations.

Decision Tree:

Decision trees [21] are an important form of calculation for prescient machine learning. The conventional equations of the decision tree have been around for quite a long time. Arbitrary timberland and present-day varieties are among the most ground-breaking varieties available procedures. The humble estimation of the decision tree known by its more contemporary CART name that illustrates Trees of Classification and Regression. Decision Tree technique is used as the most important commodity for the system to take in as it gets compelling results as soon as time allows. The Tree of Choice has different kinds of calculation: Cart, ID3, C 4.5, CHH and H48 respectively. J48 is used by them, and it is the algorithm is exceptionally mainstream. J48 utilizes the technique of pruning to create a tree. This calculation continues to be a recursive method until the point where the normal procedure is found. It provides great accuracy and adaptability. The formula is made up for the accessible equations which is given below:

$E = \Sigma\ P\square\ log\square\ P\square$ …………(3)

From equation 3,

K defines the number of classes of target attributes,
Pi defines the number of occurrences of class,
i is divided by the total number of instances.

This measure is historically referred to as "decision trees" but on a few measures like R, the more recent word CART applies to them.

3.6 Selected Algorithm:

We used different algorithm to get the highest accuracy and feature selection process to increase the accuracy. We have used **ANACONDA** to finding the performances of our algorithm. In those table, we have shown the highest accuracy and the increasing accuracy after using feature selection process:

TABLE 3.6.1: All Feature Result (Dataset 1)

	Accuracy	Precision	Recall	F1 Score
SVM	0.7058571428	1	0.70285714	0.82550335
Naive Bayes (Bernoulli)	0.7117142857	1	0.72571428	0.84105960
KNN	0.73	0.84	0.74468085	0.78947368
Decision Tree	0.69	0.72950812	0.74789915	0.73858917

In this table, we can see we have four models SVM, Naïve Bayes, KNN and Decision Tree to find the accuracy. We have compared the models with one another and select the most exact Naïve Bayes. There are three types of Naïve Bayes. Gaussian Naïve Bayes, Bernoulli Naïve Bayes and Multinomial Naïve Bayes. We got the highest accuracy from Bernoulli Naïve Bayes.

TABLE 3.6.2: All Feature Result (Dataset 2)

	Accuracy	Precision	Recall	F1 Score
SVM	0.7326923076	0.4347826087	0.7407407407	0.5479452055
Naïve Bayes (Bernoulli)	0.68	0.7826086957	0.4675324675	0.5853658537
KNN	0.7519230769	0.444444	0.6896551724	0.5505849965
Decision Tree	0.6076923076	0.49056603	0.5777777778	0.5306122403

In this table, we can see we have four models SVM, Naïve Bayes, KNN and Decision Tree to find the accuracy. We have compared the models with one another and select the most exact KNN because we have got the highest accuracy from K-Nearest Neighbors.

After doing the feature selection process we have got some set of features. After applying the discussed algorithm we have got the outcome. The best output from those set of features which is given below:

TABLE 3.6.3: After Selecting V3, V4, V5, V6, V7 (Filter Dataset 1)

	Accuracy	Precision	Recall	F1 Score
SVM	0.8268571428	1	0.77142857	0.87096774
Naïve Bayes (Bernoulli)	0.7985714285	1	0.74857142	0.85620915
KNN	0.7642857142	0.89256198	0.74482758	0.81203007
Decision Tree	0.7757142857	0.78947368	0.84	0.81395348

In this table, we can see we have four models SVM, Naïve Bayes, KNN and Decision Tree to find the accuracy. We have compared the models with one another and select the most exact SVM because we have got the highest accuracy from Support Vector Machine. Here, we can see that for the same dataset after applying the filter methods the best algorithm changes. For all features, the best algorithm was Naïve Byes and after features selection the best algorithm is SVM.

TABLE 3.6.4: After Selecting 1, 3, 4, 5, 6 (Filter Dataset 2)

	Accuracy	Precision	Recall	F1 Score
SVM	0.81153846	0.5	0.83333333	0.625
Naïve Bayes (Multinomial)	0.71346153	0.87755102	0.59722222	0.71074380
KNN	0.80115384	0.57777777	0.72222222	0.64197530
Decision Tree	0.79038461	0.66666667	0.65	0.65822661

In this table, we can see we have four models SVM, Naïve Bayes, KNN and Decision Tree to find the accuracy. We have compared the models with one another and select the most exact **KNN** because we have got the highest accuracy from K-Nearest Neighbor. Here, we can see that for the same dataset after applying the filter methods the best algorithm doesn't change. For all features, the best algo was KNN and after features selection the best algorithm is SVM as well.

3.7 Proposed Algorithm:

In our model we have tried to build a model which predicts liver diseases. We have got 3 algorithms which gives us the highest accuracy. KNN, Naïve Bayes and SVM. For all those algorithm we had a same process to build the model just we needed to import different libraries for different algorithms. ANACONDA is the environment that contains all of the deep learnings and it consists of python. We have used ANACONDA environment to make our model and find the accuracy. We have used latest python version which is 3.9.0 for our work. In our method-

Step1.Firstly we select our datasets which contain liver patient and non-liver patients dataset.

Step2.Classification of dataset into patient with liver diseases and normal.

Step3: Input the dataset.

Step4: Apply machine learning algorithm in python.

Step5: Find out highest accuracy from dataset from different machine learning algorithm.

Step6: Get highest accuracy using SVM

Step7: Measure the performance of the model.

We have to admit that the model we made is great and worthy. After all we have tried every angle to find the best output.

CHAPTER 4

EXPERIMENTAL RESULTS & DISCUSSION

4.1 Experimental Results

To evaluate the performance of our projected model we used the dataset of ILPD and BUPA. The ILPD datasets contains 11 attribute and the BUPA dataset contains 7 attribute. These datasets contains both normal and affected people’s data. We have used four classifier algorithms here namely KNN, Naïve Bayes, Support Vector Machine (SVM) and Decision Tree. And we run all that classifier both all of the attributes and also some selected particular important attributes. We select the attributes by two feature selection process namely Filter method and Wrapper method. After doing that the Support vector Machine gave the best accuracy and for all attributes of ILPD, it was 76% and for some selected important attributes of ILPD it was 83%. On the other hand for all attribute of BUPA it was 73% and after selecting some important features from BUPA the accuracy was 81%. For both cases we found that when all the attributes are used to measure the accuracy then result is lower. But when we selected some important features from both datasets the accuracy was improved. That’s actually what we needed.

We also generated confusion matrix and used it to calculate the Precision, Recall, F-Score, True Positive Rate, True Negative Rate and the Accuracy of our model.

The confusion matrix is a table that describes the performance of a classification model on a set of test data. By confusion matrix we can define four term –

True Positive (TP): We predicted result as no liver disease which are actually no-liver disease.

True Negative (TN): We predicted result as liver disease which are actually liver disease.

False Positive (FP): We predicted no liver disease, but these are not actually no liver disease.

False Negative (FN): We predicted liver disease, but these are actually no liver disease.

Precision: Precision is the piece of related instances among the retrieved instances. High precision means that an algorithm returned substantially more relevant results than irrelevant ones. That means

Precision = TP/ (TP+FP)

Recall: Recall is the piece of relevant instances that have been retrieved over the total amount of relevant instances. High recall means that an algorithm returned most of the Relevant result. So the recall means

Recall = TP/ (TP+FN)

F-measure: F-score is a measure of test's accuracy by considering both precision and recall. It is a harmonic average of precision and recall.

F-Score= (2* precision*recall) / (precision + recall)

Accuracy: Accuracy refers to the familiarity of the measured value to a known value.

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

True Positive Rate: False positive rate are refers that our proposed method predict the liver disease is no liver disease when it's actually liver disease. Calculate the false positive rate by the given equation:

True Positive Rate= TP/ (TN+FP)

The features of ILPD dataset are

- Age
- Gender
- Total Bilirubin
- Direct Bilirubin
- Total Proteins
- Albumin
- Ag Ratio
- SGPT
- SGOT
- Alkphos
- Is_Patient

The features of BUPA dataset are

- MCV
- Alkphos
- SGPT
- SGOT
- GAMMAGT
- DRINKS
- Selector

TABLE 4.1: Accuracy with all features of ILPD

	Accuracy	Precision	Recall	F1 Score
SVM	0.70585714285	1	0.70285714	0.82550335
Naive Bayes (Bernoulli)	0.71171428571	1	0.72571428	0.81105960
KNN	0.73	0.84	0.74468085	0.78947368
Decision Tree	0.69	0.72950812	0.74789915	0.73858917

TABLE 4.2: Accuracy with all features of BUPA

	Accuracy	Precision	Recall	F1 Score
SVM	0.73269230	0.434782608	0.740740740	0.5479452055
Naïve Bayes (Bernoulli)	0.68	0.782608695	0.467532467	0.5853658537
KNN	0.75192307	0.444444	0.689655172	0.5505849965
Decision Tree	0.60769230	0.49056603	0.577777777	0.5306122403

TABLE 4.3: Accuracy with selected features of ILPD

	Accuracy	Precision	Recall	F1 Score
SVM	0.8268571428	1	0.77142857	0.87096774
Naïve Bayes (Bernoulli)	0.7985714285	1	0.74857142	0.85620915
KNN	0.7642857142	0.89256198	0.74482758	0.81203007
Decision Tree	0.7757142857	0.78947368	0.84	0.81395348

TABLE 4.4: Accuracy with selected features of BUPA

	Accuracy	Precision	Recall	F1 Score
SVM	0.8115384615	0.5	0.83333333	0.725
Naïve Bayes (Multinomial)	0.7134615384	0.87755102	0.59722222	0.71074380
KNN	0.801153846	0.57777777	0.72222222	0.64197530
Decision Tree	0.7903846153	0.66666667	0.65	0.65822661

In the above table we can see the accuracy, precision, recall, F1 score for both all features and selected features for both ILPD and BUPA.

CHAPTER 5

CONCLUSION & FUTURE WORKS

5.1 Conclusion

From this research we can see that there is a lot of reasons works behind our liver diseases. In this research we wanted to make a model for liver affected people for which they will get an early alert that they are suffer from this diseases or they will have a chance to be affected. Again in this COVID pandemic it will be risky for visiting the hospital for clinical test. So we try to reduce the number of clinical test. So they will know about their liver condition by this model. For COVID situation it was really impossible to visit hospital for clinical data. For this reason we use the data from ILPD & BUPA. Because this datasets are from south Asian people and we know that the people from the south Asian region have more or less the same reason for this kind of diseases. We worked both all the features and some selected important data. We tried to figure out the best accuracy among four algorithm and among all of that the KNN gave the best accuracy before feature selection and after features selection the Support Vector Machine (SVM) has given the best accuracy for both datasets.

5.2 Future Work

From this research we can see that we have to take data from online platform to build this model. So the time after pandemic it will be possible to work for this model for Bangladesh. On that time there will be clinical data available and there is a chance to give better result from now. Again we have chances to use more algorithms to be used here or a hybrid algorithm for this model which will have a possibility to provide better accuracy. Again we have used Wrapper and Filter method here to do our job, but there is a chance to use an another method which is known as embedded method can be used here. Again for feature selection there is a tools namely WEKA can be used here instead of ANACONDA.

Such a framework has the potential for fine change from the consistent preparing gave through taking care of large-scale public or worldwide multi-institutional clients, with the upside of without any problem joining newly available data to improve prediction performance.

REFFERENCES

[1] Global Burden Of Liver Disease, available at https://www.worldgastroenterology.org/publications/e-wgn/e-wgn-expert-point-of-view-articles-collection/global-burden-of-liver-disease-a-true-burden-on-health-sciences-and-economies (accessed on July 12,2020).

[2]The global, regional, and national burden of cirrhosis, available at https://www.thelancet.com/journals/langas/article/PIIS2468-1253(19)30349-8/fulltext accessed on January 22,2020.

[3]Liver Diseases, available at https://www.medicinenet.com/liver_disease/article.htm accessed on July 22, 2020.

[4] M. Banu Priya, P. Laura Juliet and P.R. Tamilselvi "Performance Analysis of Liver Disease Prediction Using Machine Learning Algorithms" IRJET, Vol. 05 , January 2018

[5] Chieh-Chen Wu a , e , Wen-Chun Yeh b , Wen-Ding Hsu c , Md. Mohaimenul Islam a , e , Phung Anh (Alex) Nguyen e , Tahmina Nasrin Poly a , e , Yao-Chin Wang a , e , d , Hsuan-Chia Yang e , Yu-Chuan (Jack) Li ," Prediction of fatty liver disease using machine learning algorithms" ELSEVIER,PP 23-29, Year 2019

[6] Tapas Ranjan Baitharu, Subhendu Kumar Pani,'Analysis of Data Mining Techniques For Healthcare Decision Support System Using Liver Disorder Dataset." ELSEVIER,pp 862-870, year 2016.

[7] Dr. S. Vijayarani1, Mr.S.Dhayanand2,"Liver Disease Prediction using SVM and Naïve Bayes Algorithm", IJSETR,Vol.4, April 4th,2015

[8] Dr. N. V. Ramana Murthy1, S. Shruti2, V. Vinay Bhargav3, S. Anil Kumar4."Liver Disease Prediction and Diagnosis Expert System using Data Mining Techniques.",IJRAT,Vol.7, March 3,2019

[9] Kemal Akyol [1] and Yasemin Gültepe[2,]"A Study on Liver Disease Diagnosis based on Assessing the Importance of Attributes"mecs-press, DOI: 10.5815/ijisa.2017.11.01 , November 8,2017

[10] Nazmun Nahar[1] and Ferdous Ara[2] ,"LIVER DISEASE PREDICTION BY USING DIFFERENT DECISION TREE TECHNIQUES.",IJDKP, Vol.8 , March 2018

[11] Liver Disorder dataset available at https://archive.ics.uci.edu/ml/datasets/Liver+Disorders , accessed on May 2018.

[12] ILPD dataset available at https://www.kaggle.com/uciml/indian-liver-patient-records ,acccessed on November 2018

[13] Ian H. Witten , Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd edition,p.208 -264, March 2009

[14] Feature selection available at https://docs.microsoft.com/en-us/analysis-services/data-mining/feature-selection-data-mining?view=asallproducts-allversions#:~:text=Feature%20selection%20refers%20to%20the,or%20features%20from%20existing%20data , accessed on May 2020

[15] Wrapper method available at https://link.springer.com/chapter/10.1007/978-3-642-37453-1_45#:~:text=The%20wrapper%20feature%20selection%20approach,merit%20of%20the%20selected%20features , accessed on September 2020

[16] Data Filter Method , Available at https://cordis.europa.eu/docs/projects/cnect/5/215455/080/deliverables/ROADIDEA-D3-1-Data-filtering-methods-V1-1.pdf accessed on September 2020

[17] Feature selection by Filter method https://link.springer.com/chapter/10.1007/978-3-540-35488-8_4 Accessed on October 2020

[18] Support Vector Machine , available at https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/#:~:text=A%20support%20vector%20machine%20(SVM,on%20a%20text%20classification%20problem accessed on February 2017

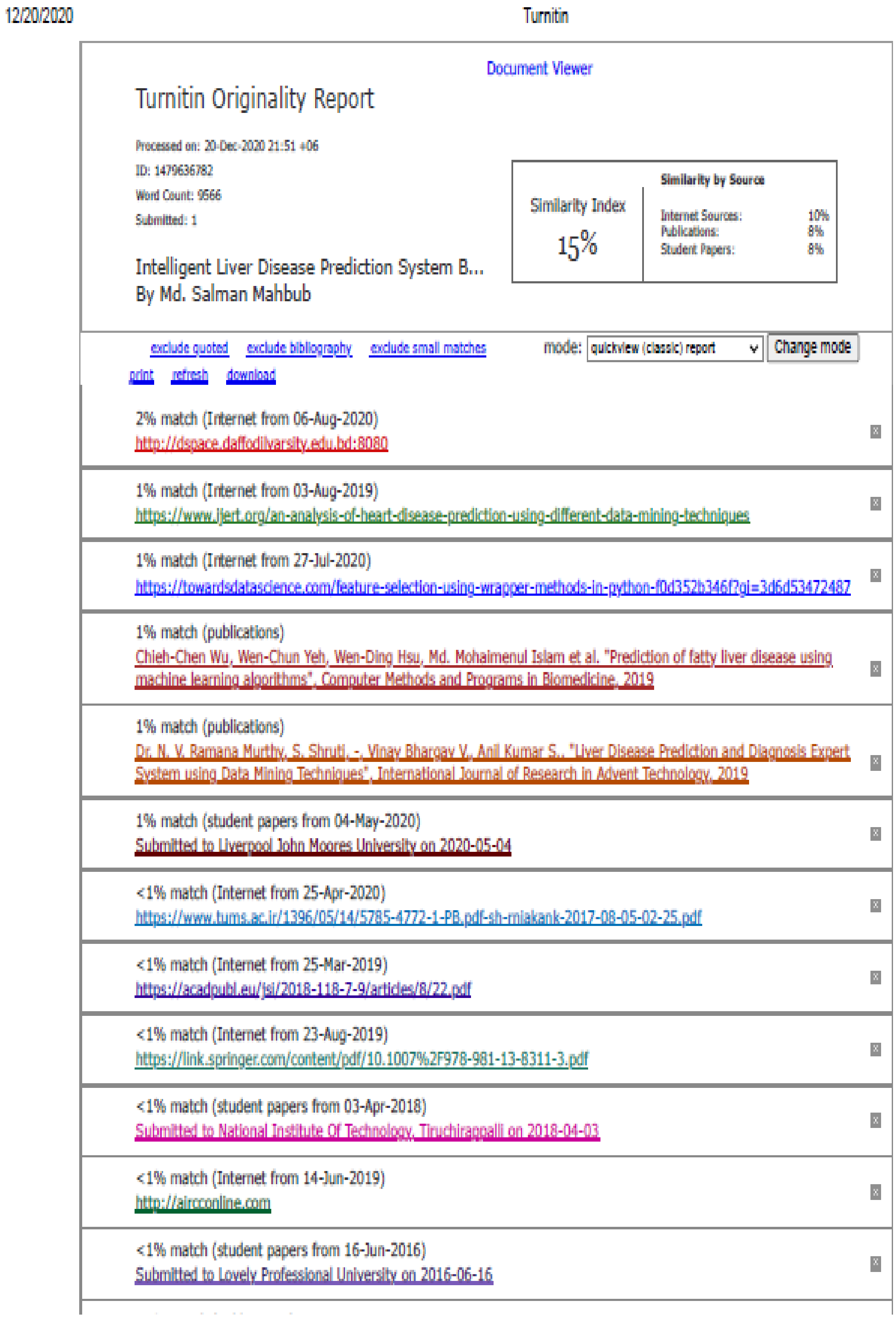
[19] Naïve Bayes classifier , available at https://en.wikipedia.org/wiki/Naive_Bayes_classifier accessed on August 2020

[20] K-Nearest Neighbors, available at https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761 accessed on june 2019

[21] Decision Tree, available at https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052 accessed on May 18,2017.

[22] WEKA, available at https://www.cs.waikato.ac.nz/ml/weka/ accessed on April 2020.

Plagiarisms Report:



12/20/2020 Turnitin

Turnitin Originality Report

Processed on: 20-Dec-2020 21:51 +06

ID: 1479636782

Word Count: 9566

Submitted: 1

Intelligent Liver Disease Prediction System B...
By Md. Salman Mahbub

Similarity Index

15%

Similarity by Source

Internet Sources:	10%
Publications:	8%
Student Papers:	8%

2% match (Internet from 06-Aug-2020)
http://dspace.daffodilvarsity.edu.bd:8080

1% match (Internet from 03-Aug-2019)
https://www.ijert.org/an-analysis-of-heart-disease-prediction-using-different-data-mining-techniques

1% match (Internet from 27-Jul-2020)
https://towardsdatascience.com/feature-selection-using-wrapper-methods-in-python-f0d352b346f?gi=3d6d53472487

1% match (publications)
Chieh-Chen Wu, Wen-Chun Yeh, Wen-Ding Hsu, Md. Mohaimenul Islam et al. "Prediction of fatty liver disease using machine learning algorithms", Computer Methods and Programs in Biomedicine, 2019

1% match (publications)
Dr. N. V. Ramana Murthy, S. Shruti, -, Vinay Bhargav V., Anil Kumar S.. "Liver Disease Prediction and Diagnosis Expert System using Data Mining Techniques", International Journal of Research in Advent Technology, 2019

1% match (student papers from 04-May-2020)
Submitted to Liverpool John Moores University on 2020-05-04

<1% match (Internet from 25-Apr-2020)
https://www.tums.ac.ir/1396/05/14/5785-4772-1-PB.pdf-sh-mlakank-2017-08-05-02-25.pdf

<1% match (Internet from 25-Mar-2019)
https://acadpubl.eu/jsi/2018-118-7-9/articles/8/22.pdf

<1% match (Internet from 23-Aug-2019)
https://link.springer.com/content/pdf/10.1007%2F978-981-13-8311-3.pdf

<1% match (student papers from 03-Apr-2018)
Submitted to National Institute Of Technology, Tiruchirappalli on 2018-04-03

<1% match (Internet from 14-Jun-2019)
http://aircconline.com

<1% match (student papers from 16-Jun-2016)
Submitted to Lovely Professional University on 2016-06-16