Comparison of Machine Learning algorithm for COVID-19 Death Risk Prediction

Praveeen Anandhanathan (prave.anand124@gmail.com)
Vellore Institute of Technology: VIT University
Priyanka Gopalan
Vellore Institute of Technology: VIT University

Research Papers

Keywords: Support vector Machine, k-Means Clustering, Decision Tree algorithm, Random Forest Method, k-Nearest Neighbor, Naïve Bayes.

Posted Date: February 4th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-196077/v1

Comparison of Machine Learning algorithm for COVID-19 Death Risk Prediction

ABSTRACT - Coronavirus disease (COVID-19) is spreading across the world. Since at first it has appeared in Wuhan, China in December 2019, it has become a serious issue across the globe. There are no accurate resources to predict and find the disease. So, by knowing the past patients' records, it could guide the clinicians to fight against the pandemic. Therefore, for the prediction of healthiness from symptoms Machine learning techniques can be implemented. From this we are going to analyse only the symptoms which occurs in every patient. These predictions can help clinicians in the easier manner to cure the patients. Already for prediction of many of the diseases, techniques like SVM (Support vector Machine), Fuzzy k-Means Clustering, Decision Tree algorithm, Random Forest Method, ANN (Artificial Neural Network), KNN (k-Nearest Neighbour), Naïve Bayes, Linear Regression model are used. As we haven't faced this disease before, we can't say which technique will give the maximum accuracy. So, we are going to provide an efficient result by comparing all the such algorithms in RStudio.

INDEX TERMS - Support vector Machine, k-Means Clustering, Decision Tree algorithm, Random Forest Method, k-Nearest Neighbor, Naïve Bayes.

1. INTRODUCTION

COVID-19 is a virus kind belongs to the virus family Coronavirus. It is first appeared in Wuhan, China. Sooner the virus starts spread across the globe in the faster manner. Globally the total infected are about 13378853 cases and 580045 deaths had happened according to WHO (Situation Report - 178) reported in July 16, 2020. The disease spreads in the rapid manner as close contact is enough to the virus to get transmitted. The rate of spread is keeps on increasing but the prevention has not yet found. We should be pre-cautious to stop the spread of the virus. Thus, a prediction is required in order to pay attention to the people who are mostly affected. Mostly the peoples who are affected are experienced symptoms in their body. Those symptoms are to be utilized in order to build a predictive model. This paper aims mostly on the symptoms and using the machine learning algorithms the symptoms are given as input and a predictive model is built. Machine Learning usually receive and analyse the input data and predict the output values as the classification. Mostly used machine learning algorithm in the field of disease prediction are Random Forest, Decision Tree, Naïve Bayes. Not only this but also more evolved algorithms are also used in the prediction of diseases. We cover most of the commonly used machine learning algorithm that are used in the field of disease prediction. All these predictive models will give a model that is related or unrelated to the prediction. So, we need to find the model that give more accuracy and also best fitted for COVID-19 prediction. To check the accurateness of the model metrics like accuracy, precision, recall, F1 score are calculated. For the verification of the best fit model the metrics such as RMSE, MSE, MAE are used. These prediction models will give the accuracy that are based on the attributes we provided, but not all these attributes will contribute for making a predictive model. Thus, a technique called feature selection is applied to it and the most important attributes/features are selected and the model is created based on those attributes. However, by applying the feature selection we obtain more accurate results rather than the previous mentioned. We adapt two feature selection algorithm such as XGBoost, Boruta algorithm. By this work we get a solution for a better prediction of COVID-19 cases.

2. RELATED WORKS

Iwendi. C et.al [1] has discussed about the machine learning algorithms such as Decision Tree, Support Vendor Machine, Gaussian Naïve, Random Forest in the prediction of COVID-19. To predict the outcome in the accurate manner they did not stop depending on factors like symptoms and age criteria. They have also included travel history, demographics, etc. Since the accurate prediction is needed the model is classified into three evaluation metrics. They concluded that the model Random Forest gives the better accuracy of 93.3%.

Kolla, Bhanu [3] has used the use ML algorithms such as KNN+NCA, Multilinear Regression and XG Boost Classifier for COVID-19 prediction. They have used two datasets. One is world dataset and another is Indian dataset. They compared the model not only with the accuracy value but also, they take R-Squared values. From their observations the model XGBoost gives the best accuracy about 42.5%

Khanday et.al [16] has used some classified the clinical reports in 4 different classes such as Term frequency/inverse document frequency (TF/IDF), Bag of words (BOW) and report length. These are then supplied to machine learning algorithm such as Logistic Regression, Multinomial Naïve Bayes for COVID-19 prediction. They state that models such as Logistic Regression and Naïve Bayes gives more accuracy of about 94%.

Pourhomayoun et.al [20] has discussed about Support Vector Machine (SVM), Artificial Neural Networks, Random Forest, Decision Tree, Logistic Regression, and K-Nearest Neighbour (KNN) algorithms to find the mortality rate among the people due to COVID-19. On building a confusion matrix they used it as classifiers and sensitivity and specificity are calculated. The results they provided gives the Neural Networks provide about 93.75% which is highest in case of their processing.

Linda Wang et.al [5] have used Convolution Neural Network in image classification, which is useful in processing the X-Ray images. Furthermore, for detailed reporting the prediction is done with Chest X-Ray images. The abnormalities in the chest characterized the infection of COVID-19. They have created an own architecture based on the X-Ray images prediction called COVID-Net which provides the best accuracy of 98.9%.

Pal et.al [6] has mentioned for each country specific dataset may have small number of observations compared to the world dataset of COVID-19. So, the proposed algorithms' accuracy may differ. Thus, in country-wise prediction they have proposed Long short -term memory (LTSM) based on neural network. They are also compared the LSTM with other neural networks such as RNN and GRU. They proposed that optimised LSTM performs well than the Neural Networks.

Vijayashree. J et.al [8] has discussed about the J48, REPTREE, SIMPLE CART algorithms on the prediction of heart diseases. On heart disease prediction they gave an accuracy of 99%. These algorithms are not used in COVID-19 cases till date.

Panda et.al [9] has discussed about the Lasso and ridge Regression in Heart disease prediction. Accuracy have been noted with Lasso and Ridge separately. Naïve Bayes on applying Lasso and Ridge regression provides an accuracy of 94.92% which is best for their process. These algorithms were never used in cases of COVID-19 till date.

Table 1. Existing Works in Various Papers

Reference	Compared Techniques	Accuracy	Precision	Recall	F1	No. of Attributes	Best Model
Iwendi C et.al [1]	Logistic Regression	84.4%	66.6%	72.7%	69.5%	14	Random Forest
	Decision Tree	86.6%	85.7%	54.5%	66.6%		
	SVM	84.4%	70%	63.6%	66.6%		
	Naïve Bayes	77.7%	53.3%	72.7%	61.5%		
	Random Forest	93.3%	100%	75%	85.7%		
Kolla, Bhanu.[3]	SVM	11.7%	-	-	-	8 (Lesser Observations – Indian Dataset)	XGBoost
	KNN+NCA	37.5%	-	-	-		
	Decision Tree	15.8%	-	-	-		
	Naïve Bayes	11.8%	-	-	-		
	Multilinear Regression	11.8%	-	-	-		
	Logistic Regression	11.8%	-	-	-		
	Random Forest	41.8%	-	-	-		
	XGBoost	42.5%	-	-	-		
Khanday et.al [16]	Logistic Regression	94%	96%	95%	96.2%	12	Logistic Regression & Naïve Bayes
	Naïve Bayes	94%	96%	95%	96.2%		
	SVM	82%	91%	86%	90.6%		
	Decision Tree	82%	92%	92%	92.5%		
Mohammad Pourhomayoun et.al [20]	Neural Network	93.75%	-	-	-	15	Neural Network
	Random Forest	91.88%	-	-	-		
	SVM	90.63%	-	-	-		
	Decision Tree	90.63%	-	-	-		
	Logistic Regression	90.00%	-	-	-		
	KNN	83.12%	-	-	-		

3. METHODOLOGY

3.1 FEATURE SELECTION

The technique Feature Selection is applied when the number of attributes in the dataset is higher. In this process we can manually or automatically select the features which contribute most for our prediction model. When we have more and more unimportant features it may lead our prediction model invalid by giving lesser accuracy. This is because when we train the model with the unimportant attribute, the model will learn based on those unimportant features. The major reasons to perform feature section is to reduce overfitting by making decision based on noise in lesser chance. Also, the foremost reason is to improve accuracy by reducing the misleading data. Also, when the number of attributes is lesser the time taken to train models with the algorithm is comparatively lesser. There are some techniques that have a specific process in creating the feature importance. They are Filter method, Wrapper method, Embedded method.

Filter Method: It acts as a pre-processing step that occurs before training a model. Here we calculate the variance of every feature and a subset from the overall feature is selected and marked as the important feature. Thus, the feature marked as important will usually have higher variance and contains more information.

Wrapper Method: In simple words the wrapper method is a sequential process of selecting the feature. It uses Greedy Search Algorithm where it tries to find the optimal feature on each and every iteration that occurs. When we start, we start with the empty subset and for every iteration one feature is selected and the results are noted to select the best features.

Embedded Method: It solves all the problem that are faced by the filter and wrapper method. First the dataset in trained using machine learning model. Then the important feature is selected based on the importance of making the prediction. Then the unimportant attributes are removed using the important feature that we obtain. It includes Regularization which is adding a penalty term to the model to overcome overfitting.

Here we use XGBoost which is an Embedded Method of feature selection and Boruta which is a Wrapper Method of feature selection. XGBoost trains the model with the helps of Decision Tree whereas Boruta trains the model using Random Forest Classifier.

3.2 FEATURE ALGORITHMS

a. XGBOOST ALGORITHM

STEP 1: Train dataset using tree classifier
STEP 2: Calculate error in the classifier
STEP 3: Error are reduced by adjusting selection parameter (High Gain)
STEP 4: Construct a boosted tree with the adjusted selection parameter
STEP 5: Important score for each attribute in noted
STEP 6: For every iteration {

i. Rank the attributes according to the importance value
ii. **If** high rank {
Attribute marked as important}
iii. **Else** {
Attribute marked as unimportant}

}

First the dataset with all attributes are selected and trained using the Simple Tree Classifier. Then the amount of error that occurs in the tree classifier is noted down and the parameters and modified to reduced the rate of error. With the newer parameters a tree called Boosted tree is built. Now the attribute with higher importance values are ranked top and the attribute with lesser importance value are ranked lower. This process of ranking the attributes based on the importance value occurs for each iteration. Then the final importance value that we obtain from the final iteration is taken into consideration and the important features are selected.

b. BORUTA ALGORITHM

STEP 1: Add randomness to the dataset by creating shadow features (shuffled copies)
STEP 2: Train dataset by random forest classifier
STEP 3: Apply feature importance measure and note the importance of each feature
STEP 4: For every iteration {

i. Compare Real feature importance with Shadow Feature
ii. **If** Real > Shadow Feature {
Update higher important feature}
iii. **Else** {
Remove unimportant feature}

}

STEP 5: Finally, the best feature will be chosen.

First the attributes in the dataset are ranked the important features randomly. This method of ranking the attributes randomly is known as Shadow features. Then the whole dataset is trained using Random Forest Classifier. From the Random Forest Classifier, the feature importance measure is done and the most important features are noted down. These ranking done by Random Forest is known as Real features. Not only one iteration of Random Forest occurs. Many numbers of iteration occur. For every iteration the real Feature get updated. At last the shadow feature and real features are compared. Here the feature with higher rank either shadow or real is selected and marked as important attributes.

3.3 CLASSIFICATION MODELS

1) RANDOM FOREST

Select the random samples from a given dataset. Then with the random forest algorithm we will construct a decision tree for every sample. Then it will get the prediction result from every decision tree. For every decision tree voting will be performed for every predicted result. At last, select the most voted prediction result as the final prediction result.

2) SUPPORT VECTOR MACHINE

On processing Support Vector Machine, it finds lines or boundaries that correctly classify the training dataset. Then, from those lines or boundaries, it picks the one that has the maximum distance from the closest data points.

3) NAÏVE BAYES CLASSIFIER

For the given dataset, a frequency table is created. Then likelihood table is created by finding the probabilities. Then, with the help of Naïve Bayesian equation we can calculate the posterior probability for each class. The highest probability is the prediction outcome

$$p(C_k \mid \mathbf{x}) = \frac{p(C_k)\, p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$$

4) DECISION TREE CLASSIFIER

The leaf is labelled with a similar class if the instances belong to similar class. For each attribute, the entropy will be calculated**.**

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Depending upon the current selection parameter (High Gain), the attributes are selected. Then the best attribute is taken as the root node and the following decision levels are made. Then the tree is being split till is attaining a proper subset.

5) LINEAR REGRESSION

From the given dataset, to make predictions with the help of relationship in the data Linear Regression is used.

$$\mathbf{y = B0 + B1 * x}$$

The line denotes y as the output variable which we want to predict. X is the input variable and B0 and B1 are the coefficient that is estimated to move the line.

6) LOGISTIC REGRESSION

In logistic regression, similar to linear regression it makes use of real-valued inputs and make prediction as to the probability of the input belonging to same class. It has three coefficient values rather the two coefficient values in linear regression.

$$\mathbf{y = B0 + B1*x1 + B2*x2}$$

7) k-NEAREST NEIGHBOUR

A random number of ***k*** neighbors that you want (*default = 5*) is selected. Take the ***k*** nearest neighbors of the new data point, according to the Euclidean Distance.

$$\mathbf{SQRT((x2\text{-}x1)^2 + (y2\text{-}y1)^2)}$$

Among the ***k*** neighbors, count the number of data points in each of the category. Assign the new data point to the category where counted the most of the neighbors.

8) NEURAL NETWORK

Neural Network contains layers such as Input, hidden and output. In each layer number of nodes varies and provide and classification at the output layers.

9) J48

J48 uses C4.5 algorithm for implementation. It created and developed by an open Java implementation WEKA Tool. When we have a dataset that consists of independent and another dependent variable, we can apply J48 algorithm on the dataset is such cases which allows us to predict new set of values. We can also prune the tree after creation of the tree [22]

10) LASSO, RIDGE, ELASTIC NET REGRESSION

If the coefficient is too large, it ends up in over-fitting on the training dataset. So, to overcome these problems, regularization can be done which penalize the larger coefficient. In Ridge Regression modification is done by adding a penalty that is equivalent to the square of coefficient values also known as L2-norm. Thus, here the penalty term is added as the square of the coefficient to the sum of the squared residual to the alpha value randomly initiated. The loss function in the Lasso Regression is modified by adding penalty that is equivalent to the absolute value of coefficient also known as L1-norm. Elastic Net combines both Ridge and Lasso Regression. The penalty is added here by using

both L1-norm and L2-norm. It can be used as Ridge and Lasso by setting the parameter to 1 or 0 [9].

3.4 PERFORMANCE MEASURE

The efficiency of the model cannot be determined only by the accuracy. It also needs some other metrics to find out the best model in the performance. So, the models are classified by some standard performance metrics like Accuracy, Precision, Recall, F1 Score. Accuracy is calculated by the simple ratio of Predicted observation to the Total observation. It is the most intuitive Performance Measure. Precision is the ratio of Predicted Positive to the Total Predicted Positive observations. Recall is the ratio of correctly predicted positive observations to the all observation in the actual class. F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both FP and FN values in account. We can't directly say when the model gives higher accuracy it gives the perfect prediction. Even though the accuracy is higher when the model failed to make the best fit, the model fails. We can say whether the model is fit or not by using the Root Mean Square Error, Mean Square Error, Mean Absolute Error. Mean Absolute Error (MAE) is the sum of errors between the actual and predicted values in the absolute form of all values. Mean Square Error (MSE) is the positive errors is the squared value. Root Mean Square Error (RMSE) is more similar to MSE where the final error value is processed with the square root. The error values and fitness of the model is inversely proportional. When these values are lower the fit of the model is higher. When the values are higher the model is not more fit.

3.5 RESULTS AND DISCUSSIONS

In this paper, we have used R Studio to predict COVID-19 of Kaggle Dataset. In RStudio we can easily import the package of every Machine Learning Techniques and they can be easier for building a model and comparing their performance. Performance of various models can be compared easily by visual representation graphs. The Prediction process using Machine Learning starts with pre-processing of data. Table 2 consists of the detailed information about the attributes. It is followed by selecting the features and target values. Then later the dataset is split into two parts namely Train data and Test data in the ratio of 80:20. Then the technique is applied in the training data and the appropriate model for that technique is created. Later this model is evaluated using testing data. Then, the filters such as XGBoost and Boruta are applied.

a. DATASET DESCRIPTION

Daily level information on affected/admitted people can provide a basis for prediction on machine learning. This data is collected by Sudalai Rajkumar. It is available online on Kaggle from 22, JAN 2020. He collected the data for Johns Hopkins University, which has made an excellent dashboard using the affected cases data. Now data is available as csv files in the Johns Hopkins GitHub repository. This dataset is licensed and available for general usage. The Usability of the Dataset is 9.7. 1354 Kernel has used this dataset for working on their projects. This dataset has daily level information on the number of affected cases, deaths and recovery from 2019 novel coronavirus. This is a time series data and so the number of cases on any given day is the cumulative number [21].

b. DATA PRE-PROCESSING

From the attributes mentioned in the dataset sym_on, hosp_vis, vis_wuhan, from_wuhan, death, recovered, symptom1, 2, 3, 4, 5, 6 are clinical data. But only symptoms and personal information such as age and gender are important for predicting. Datasets also contain some missing values. Thus, NA/N values are converted to zero or they may be neglected. The dataset now contains peoples of three categories: Death, Recovered, Critical. The value of death, recovered are only useful in predicting the model. It should be either 1 for death or recovered. If both the attributes contain value 0, then the person did not dead or recovered, he is in critical condition. Thus, these values are also neglected. Originally the dataset contains of 1085 observations. After pre-processing the number of observations is 222 observations.

Column	Description	Type
S no	Patient Id	Numeric
Location	The location where the Patient belongs to	String, Categorical
Country	Patient's Native Country	String, Categorical
Gender	Patient's Gender	String, Categorical
Age	Patient's Age	Numeric
Sym_on	The date when patient started noticing symptom	Date
Hosp_vis	The Date when patient visited hospital	Date
Vis_wuhan	Whether the Patient visited Wuhan	Numeric, Categorical
From_wuhan	Whether the Patient belongs to Wuhan	Numeric, Categorical
Death	Whether the patient is dead	Numeric, Categorical
Recovered	Whether the patient is recovered	Numeric, Categorical
Symptom1 Symptom 2 Symptom 3 Symptom 4 Symptom 5 Symptom 6	Symptoms Noticed by the patients	String, Categorical

Table 2. Dataset Attributes

c. FEATURE SELECTION PROCESS

The pre-processed dataset now contains 10 attributes in which Recovered is the exact opposite of Death. We consider the Death as the target value and it is made as feature. We can also apply XGBoost and Boruta filters to these attributes and we can select the best features in those attributes.

On applying XGBoost to those attributes we get Age, Gender, Symptom1, Symptom2, Symptom3 as the Important attributes. The importance of those attributes is mentioned in the Figure 1.

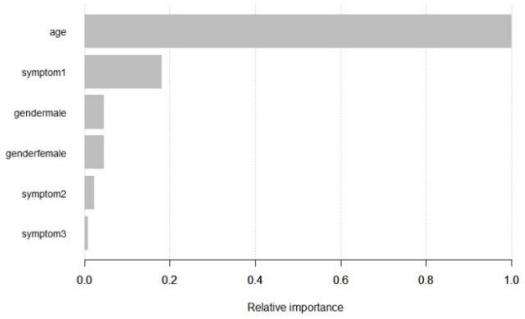

Figure 1. Importance attribute using XGBoost

Also, on applying Boruta filter we get that attributes such as Age, Gender, Symptom1 as the best features. Figure 2 shows the importance attributes obtained using Boruta filter.

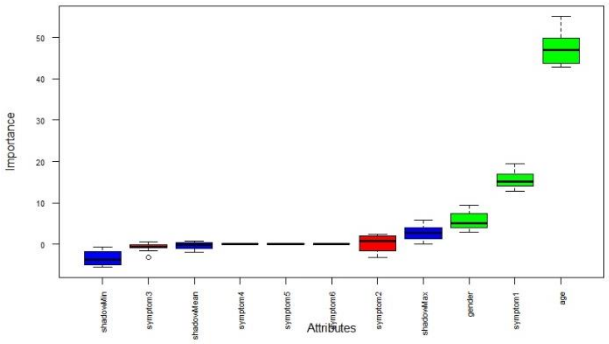

Figure 2. Importance attribute using Boruta

The accuracy values using these feature selections are obtained, and the best of those are selected. Then the models such as Random Forest Classifier, Support Vector Machine, Naïve Bayes Classifier, Decision Tree Classifier, Linear Regression, Logistic Regression, k-Nearest Neighbour, XG Boost are used for predicting. Then the prediction is repeated with Decision Tree with J48 algorithm and various regression types such as Ridge, Lasso, Elastic Net Regression.

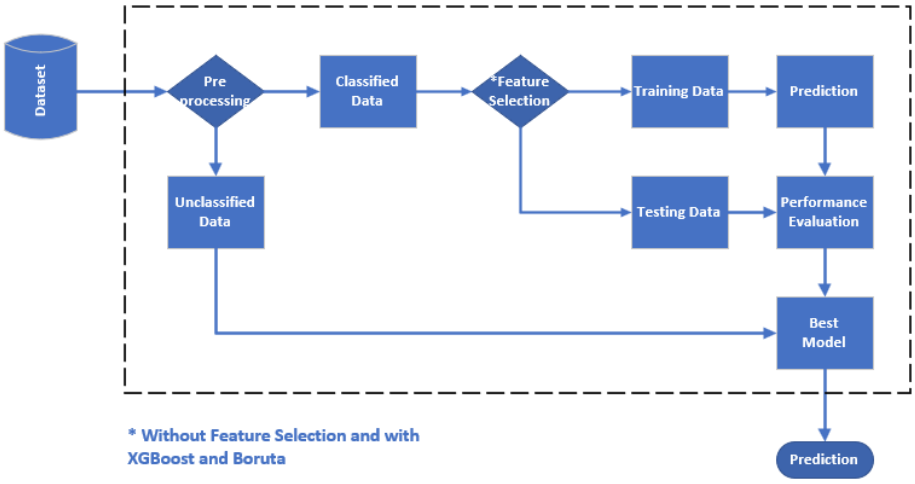

Figure 3. Experiment Workflow with Kaggle Dataset

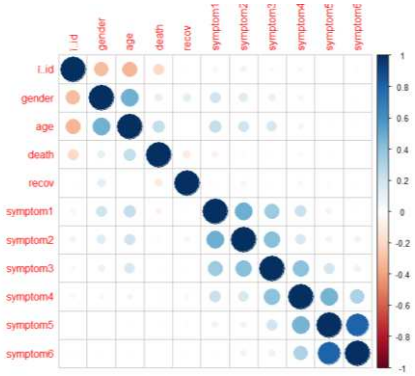

Figure 4. Co-relation plot of various attributes

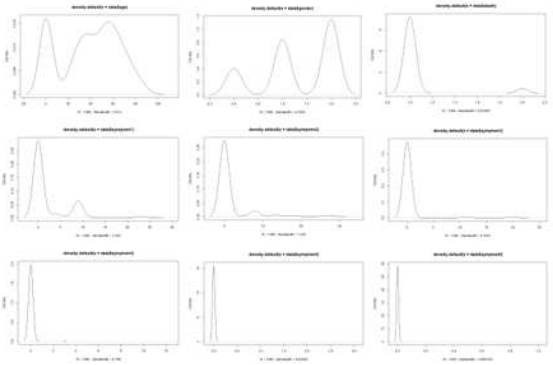

Figure 5. Density plot of various attributes

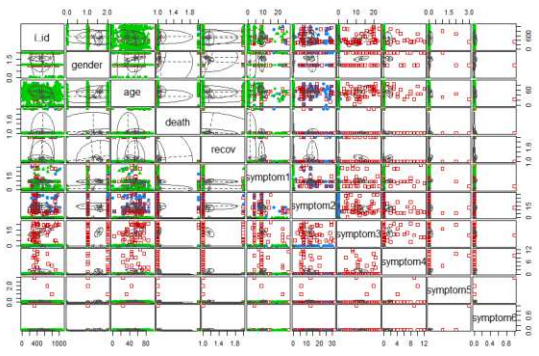

Figure 6. Clustering Classification

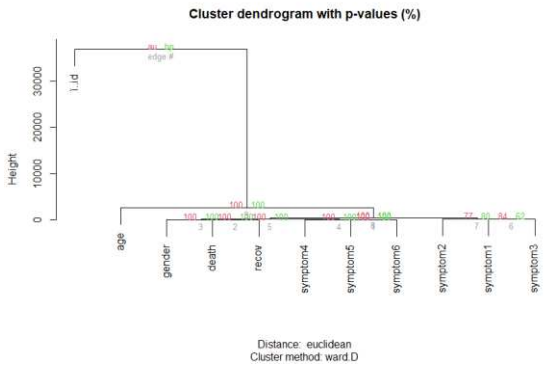

Figure 7. Clustering Dendrogram

The models are predicted for accuracy without any filters applied and also applied filters such as XGBoost and Boruta. The best accuracy of all classification methods is mentioned in Table 3. From the table we can see that the accuracy with Boruta method gives higher values. Thus, performance metrics are considered by applying Boruta Filter. Accuracy, Precision, Recall, F1 score, RMSE, MSE, MAE are considered for Boruta filter mentioned in Table 4. From Table 4 we can understand that Regression model gives higher accuracy than others. But when seeing the values such as RMSE, MSE, MAE we can say that the models Linear and Logistic Regression has more values. Thus, the model is less fit. Therefore, only the model such as Lasso, Ridge, Elastic Net Regression are the model with more accuracy and also the best fit.

Models	**Accuracy (holdout)**		
	Without Feature Selection	**With XGBoost**	**With Boruta**
Random Forest	82.1	79.5	**82.1**
Support Vector Machine	82.1	89.7	**94.9**
Naïve Bayes	92.3	92.3	**92.3**
Decision Tree	79.5	76.9	**76.9**
Linear Regression	92.3	92.3	**97.4**
Logistic Regression	92.3	92.3	**97.4**
k-Nearest Neighbor	87.2	87.2	**89.7**
Feed Forward NN	92.3	92.3	**92.3**
Back Propagation NN	82.1	92.3	**92.3**
J48	84.6	84.6	**84.6**
Lasso Regression	92.3	84.6	**97.4**
Ridge Regression	94.9	94.9	**97.4**
Elastic Net Regression	92.3	94.9	**97.4**

Table 3. Accuracy values with and without filters

Models	**Accuracy**	**RMSE**	**MSE**	**MAE**	**Precision**	**Recall Score**	**F1 Score**
Random Forest	82.1	0.424	0.179	0.179	89.3	86.2	87.7
Support Vector Machine	94.9	0.226	0.051	0.051	100	93.3	96.6
Naïve Bayes	92.3	0.277	0.077	0.077	96.4	93.1	94.7
Decision Tree	76.9	0.48	0.231	0.231	85.7	82.8	84.2
Linear Regression	**97.4**	**1.038**	**1.077**	**1.026**	**100**	**96.6**	**98.2**
Logistic Regression	**97.4**	**1.038**	**1.077**	**1.026**	**100**	**96.6**	**98.2**
k-Nearest Neighbor	89.7	0.32	0.103	0.103	89.3	96.2	92.6
Feed Forward NN	92.3	1	1.001	0.965	96.3	92.9	94.5
Back Propagation NN	92.3	1	1.001	0.965	96.3	92.9	94.5
J48	84.6	0.392	0.154	0.154	85.7	92.3	88.9
Lasso Regression	**97.4**	**0.16**	**0.026**	**0.026**	**100**	**96.6**	**98.2**
Ridge Regression	**97.4**	**0.16**	**0.026**	**0.026**	**100**	**96.6**	**98.2**
Elastic Net Regression	**97.4**	**0.16**	**0.026**	**0.026**	**100**	**96.6**	**98.2**

Table 4. Results of various methods with Boruta filter

d. BENCHMARKING OF PROPOSED MODEL

Benchmarking is needed to compare the proposed system's performance with the existing ones. Thus, Benchmarking is used to show the best model and does it improve the accuracy or not. Figure 8, Figure 9, Figure 10 are the accuracy plots for the Attributes without filters and with XGBoost and Boruta. We must know which of these models gives more accuracy. Thus, the values of all the models are plotted. The plotting is made for all metrics separately. These values under plot are obtained from applying Boruta filter. Figure 10, Figure 11, Figure 12, Figure 13, Figure 14, Figure 15, Figure 16 shows the plots of various models with respect to Accuracy, Precision, Recall, F1 Score, RMSE, MSE, MAE. From the above-mentioned plots, it is clear that all Regression Techniques on Boruta filter applied gives more accuracy rate (97.4%). But the regression techniques such as Lasso, Ridge, Elastic Net Regression gives both accuracy and also the best fit value with the error of 0.026. While seeing other accuracy values with other filers and without filters Ridge Regression model give the best performance in predicting COVID-19.

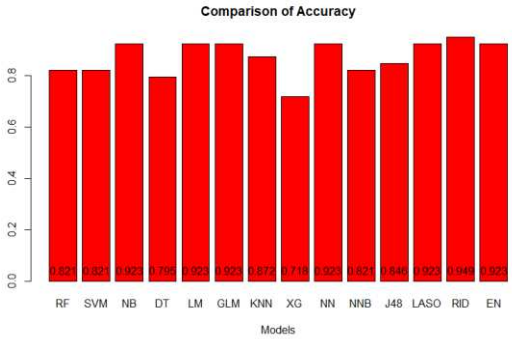

Figure 8. Accuracy comparison without filters

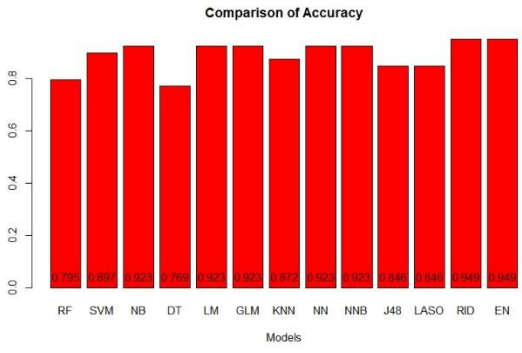

Figure 9. Accuracy comparison with XGBoost

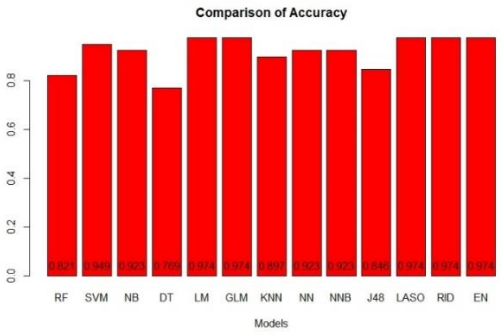

Figure 10. Accuracy comparison with Boruta

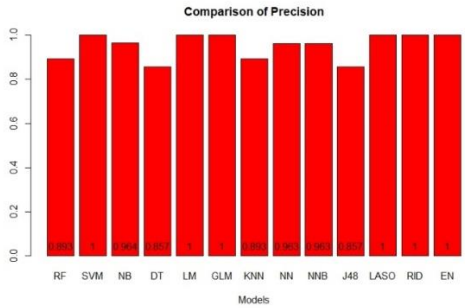

Figure 11. Precision comparison with Boruta

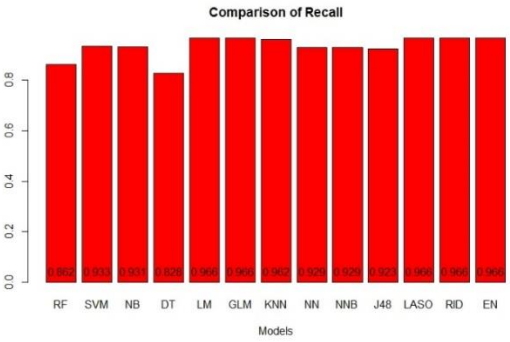

Figure 12. Recall Score comparison with Boruta

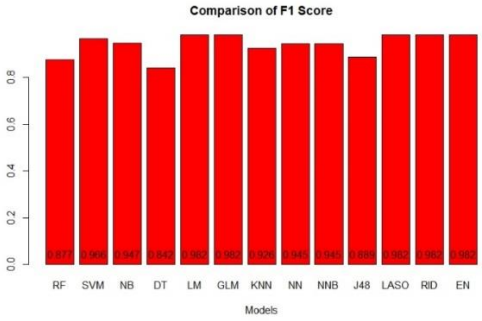

Figure 13. F1 Score comparison with Boruta

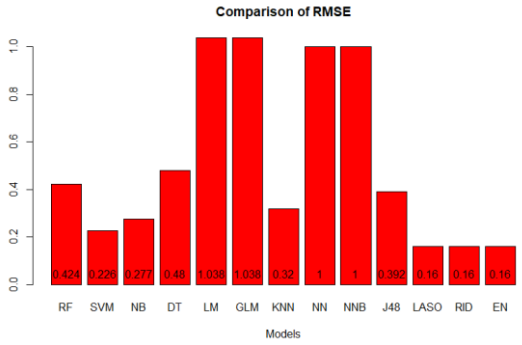

Figure 14. RMSE comparison with Boruta

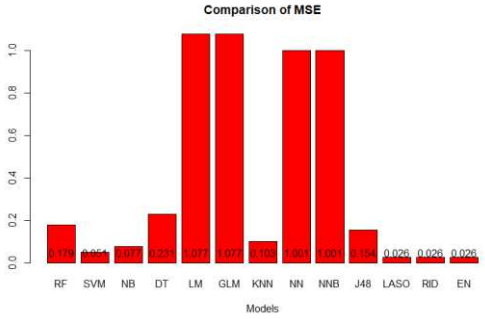

Figure 15. MSE comparison with Boruta

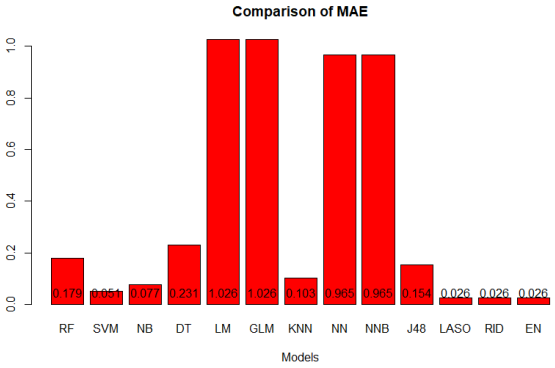

Figure 16. MAE comparison with Boruta

4. CONCLUSION

Predicting the COVID-19 using their symptoms is essential for today's world. COVID-19 is the present pandemic situation and it is a challenging issue in the medical field. By this predicting the presence of disease it helps the people in early detection and they can apply medication on the early stages that helps them to cure the disease. Machine Learning Algorithm such as RF, SVM, NB, DT, LR, XG Boost, J48, Lasso Regression, Ridge Regression, Elastic Net Regression provides predicting for various kind of disease. But not all these prediction models give the accurate results. Thus, we propose that the Ridge Regression provides a better prediction. In future the work can be added along with other application of machine learning like predicting the presence of disease using X-Ray images and by predicting the outburst of the pandemic situation using forecasting model. By combining all these applications, we can concentrate more on the areas that has higher outcomes and in that locality all people are screened by both symptoms and X-Ray prediction.

REFERENCES:

[1] Iwendi, C., Bashir, A. K., Peshkar, A., Sujatha, R., Chatterjee, J. M., Pasupuleti, S., … Jo, O. (2020). COVID-19 Patient Health Prediction Using Boosted Random Forest Algorithm. Frontiers in Public Health, 8. doi:10.3389/fpubh.2020.00357

[2] Sujath, R., Chatterjee, J. M., & Hassanien, A. E. (2020). A machine learning forecasting model for COVID-19 pandemic in India. Stochastic Environmental Research and Risk Assessment. doi:10.1007/s00477-020-01827-8

[3] Kolla, Bhanu. (2020). Analysis, Prediction and Evaluation of COVID-19 Datasets using Machine Learning Algorithms. International Journal of Emerging Trends in Engineering Research. 8. 10.30534/ijeter/2020/117852020.

[4] Li Yan, Hai-Tao Zhang, Yang Xiao, Maolin Wang, Chuan Sun, Jing Liang, Shusheng Li (2020). Prediction of criticality in patients with severe Covid-19 infection using three clinical features: a machine learning-based prognostic model with clinical data in Wuhan. doi: https://doi.org/10.1101/2020.02.27.20028027

[5] Linda Wang, Alexander Wong. COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images. arXiv:2003.09871

[6] Pal, Ratnabali & Sk, Arif Ahmed & Kar, Samarjit & Prasad, Dilip. (2020). Neural network based country wise risk prediction of COVID-19. 10.20944/preprints202004.0421.v1.

[7] Cristian Bayes, Victor Sal y Rosas, Luis Valdivieso. Modelling death rates due to COVID-19: A Bayesian approach. arXiv:2004.02386

[8] Vijayashree, J. & Iyenger, N Ch Sriman Narayana. (2016). Heart Disease Prediction System Using Data Mining and Hybrid Intelligent Techniques: A

Review. International Journal of Bio-Science and Bio-Technology. 8. 139-148. 10.14257/ijbsbt.2016.8.4.16.

[9] Panda, Debjani & Ray, Ratula & Abdullah, Azian & Dash, Satya. (2019). Predictive Systems: Role of Feature Selection in Prediction of Heart Disease. Journal of Physics: Conference Series. 1372. 012074. 10.1088/1742-6596/1372/1/012074.

[10] Dianbo Liu, Leonardo Clemente, Canelle Poirier, Xiyu Ding, Matteo Chinazzi, Jessica T Davis, Alessandro Vespignani, Mauricio Santillana. A machine learning methodology for real-time forecasting of the 2019-2020 COVID-19 outbreak using Internet searches, news alerts, and estimates from mechanistic models. arXiv:2004.04019

[11] Bo Ram Beck, Bonggun Shin, Yoonjung Choi, Sungsoo Park, Keunsoo Kang. Predicting commercially available antiviral drugs that may act on the novel coronavirus (2019-nCoV), Wuhan, China through a drug-target interaction deep learning model. doi: https://doi.org/10.1101/2020.01.31.929547

[12] Shuai Wang, Bo Kang, Jinlu Ma, Xianjun Zeng, Mingming Xiao, Jia Guo, Mengjiao Cai, Jingyi Yang, Yaodong Li, Xiangfei Meng, Bo Xu. A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19). doi: https://doi.org/10.1101/2020.02.14.20023028

[13] Qian Guo, Mo Li, Chunhui Wang, Peihong Wang, Zhencheng Fang, Jie tan, Shufang Wu, Yonghong Xiao, Huaiqiu Zhu. Host and infectivity prediction of Wuhan 2019 novel coronavirus using deep learning algorithm. doi: https://doi.org/10.1101/2020.01.21.914044

[14] Randhawa GS, Soltysiak MPM, El Roz H, de Souza CPE, Hill KA, et al. (2020) Machine learning using intrinsic genomic signatures for rapid classification of novel pathogens: COVID-19 case study. PLOS ONE 15(4): e0232391

[15] Santosh, K. C. (2020). AI-Driven Tools for Coronavirus Outbreak: Need of Active Learning and Cross-Population Train/Test Models on Multitudinal/Multimodal Data. Journal of Medical Systems, 44(5). doi:10.1007/s10916-020-01562-1

[16] Khanday, A. M. U. D., Rabani, S. T., Khan, Q. R., Rouf, N., & Mohi Ud Din, M. (2020). Machine learning based approaches for detecting COVID-19 using clinical text data. International Journal of Information Technology. doi:10.1007/s41870-020-00495-9

[17] Pinter, G., Felde, I., Mosavi, A., Ghamisi, P., & Gloaguen, R. (2020). COVID-19 Pandemic Prediction for Hungary; A Hybrid Machine Learning Approach. Mathematics, 8(6), 890. doi:10.3390/math8060890

[18] H. Kassani, Sara & Kassasni, Peyman & Wesolowski, Mike & Schneider, Kevin & Deters, Ralph. (2020). Automatic Detection of Coronavirus Disease (COVID-19) in X-ray and CT Images: A Machine Learning-Based Approach.

[19] Punn, Narinder & Sonbhadra, Sanjay & Agarwal, Sonali. (2020). COVID-19 Epidemic Analysis using Machine Learning and Deep Learning Algorithms. 10.1101/2020.04.08.20057679.

[20] Pourhomayoun, Mohammad & Shakibi, Mahdi. (2020). Predicting Mortality Risk in Patients with COVID-19 Using Artificial Intelligence to Help Medical Decision-Making. 10.1101/2020.03.30.20047308.

[21] kaggle datasets download -d sudalairajkumar/novel-corona-virus-2019-dataset https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset/

[22] Maliha, S. K., Ema, R. R., Ghosh, S. K., Ahmed, H., Mollick, M. R. J., & Islam, T. (2019). Cancer Disease Prediction Using Naive Bayes,K-Nearest Neighbor and J48 algorithm. 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT). doi:10.1109/icccnt45670.2019.8944686

[23] Kaur, Gaganjot & Chhabra, Amit. (2014). Improved J48 Classification Algorithm for the Prediction of Diabetes. International Journal of Computer Applications. 98. 13-17. 10.5120/17314-7433.

Figures

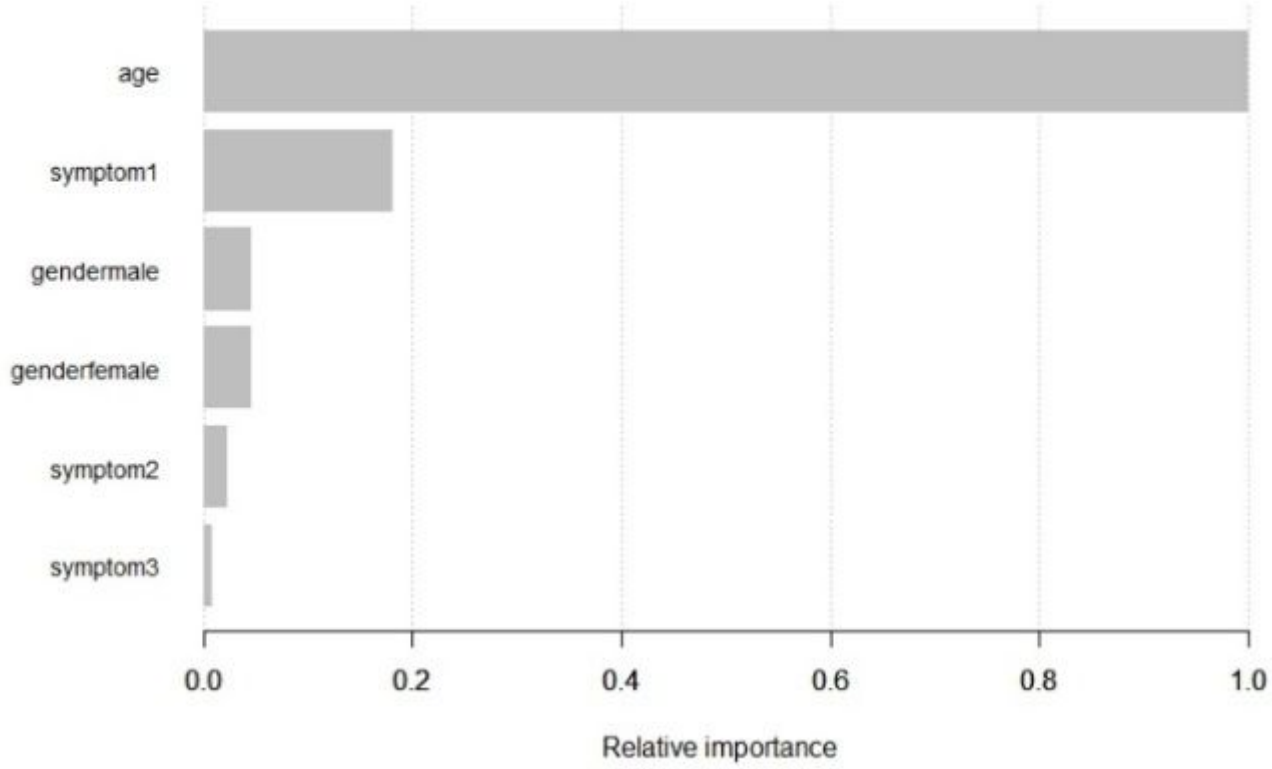

Figure 1

Importance attribute using XGBoost

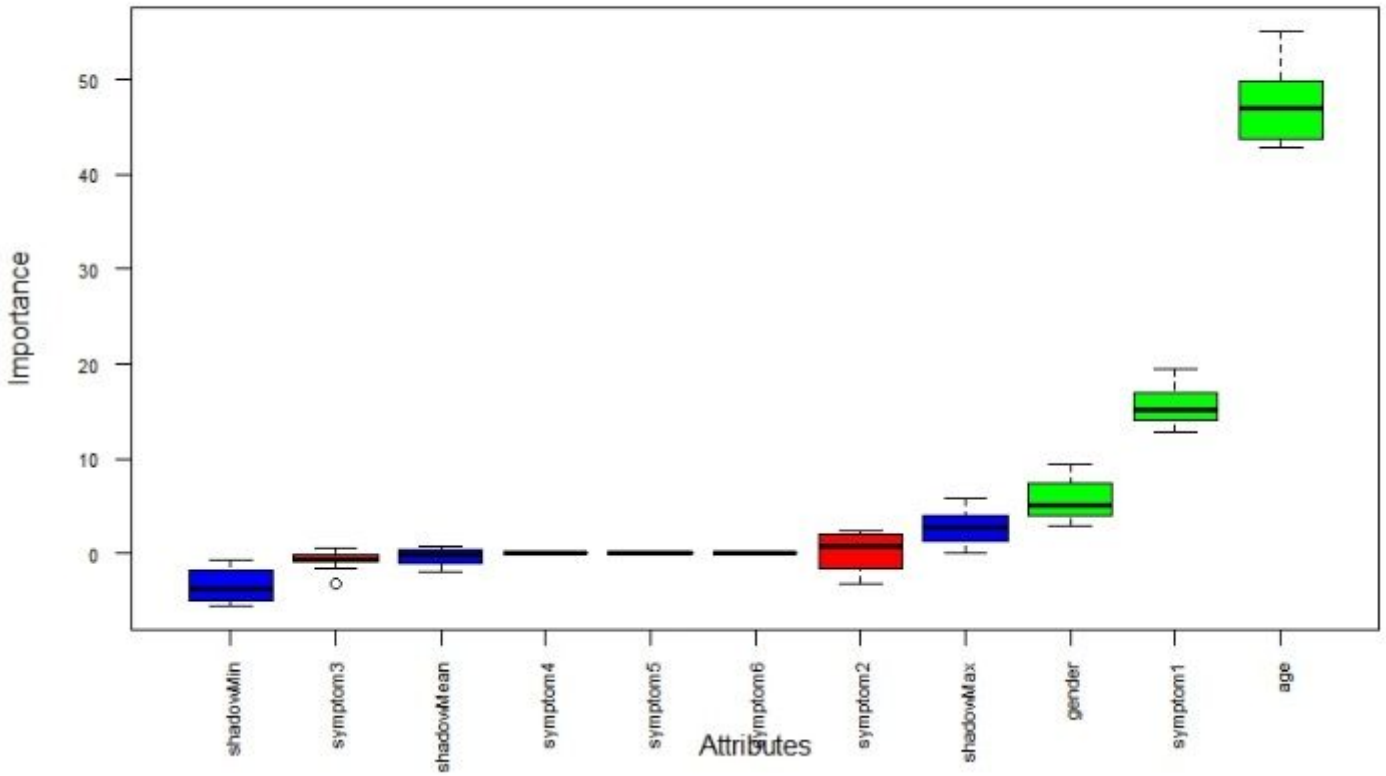

Figure 2

Importance attribute using Boruta

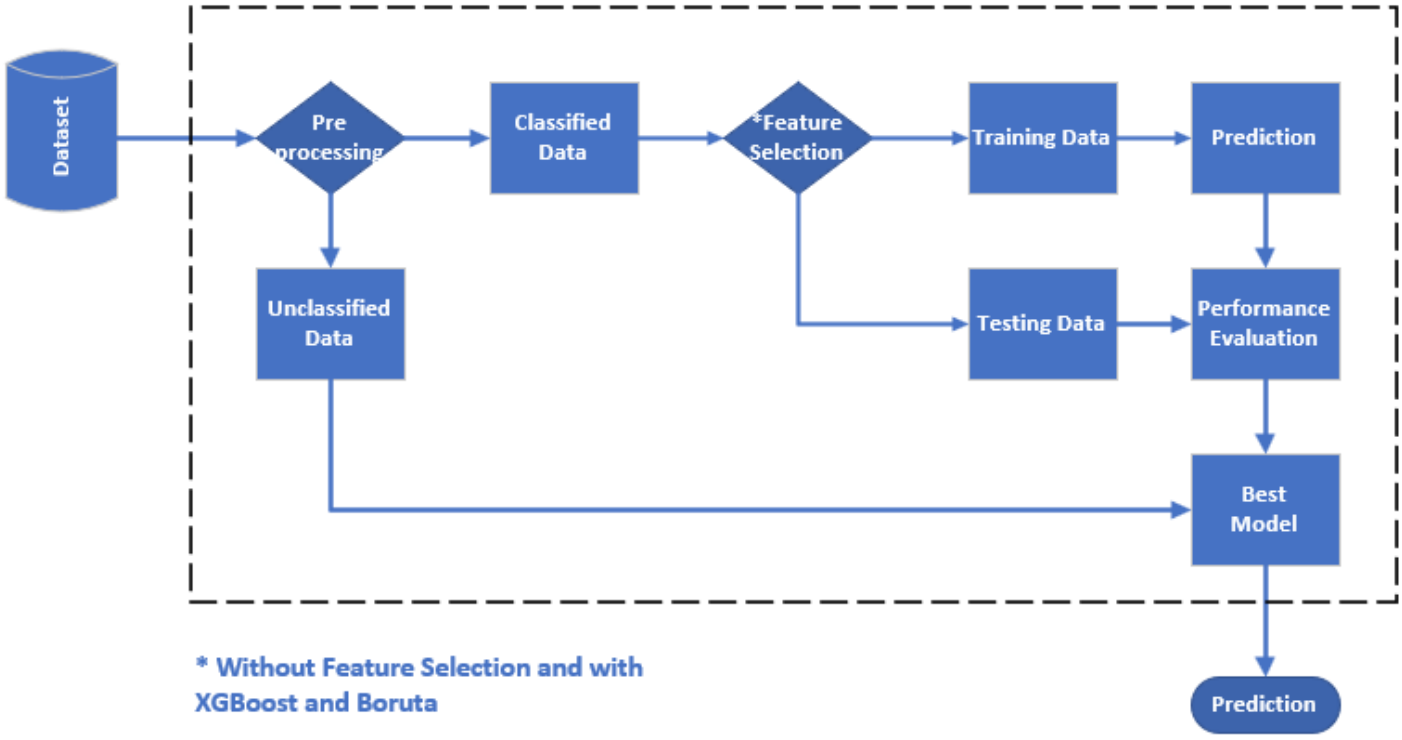

Figure 3

Experiment Workflow with Kaggle Dataset

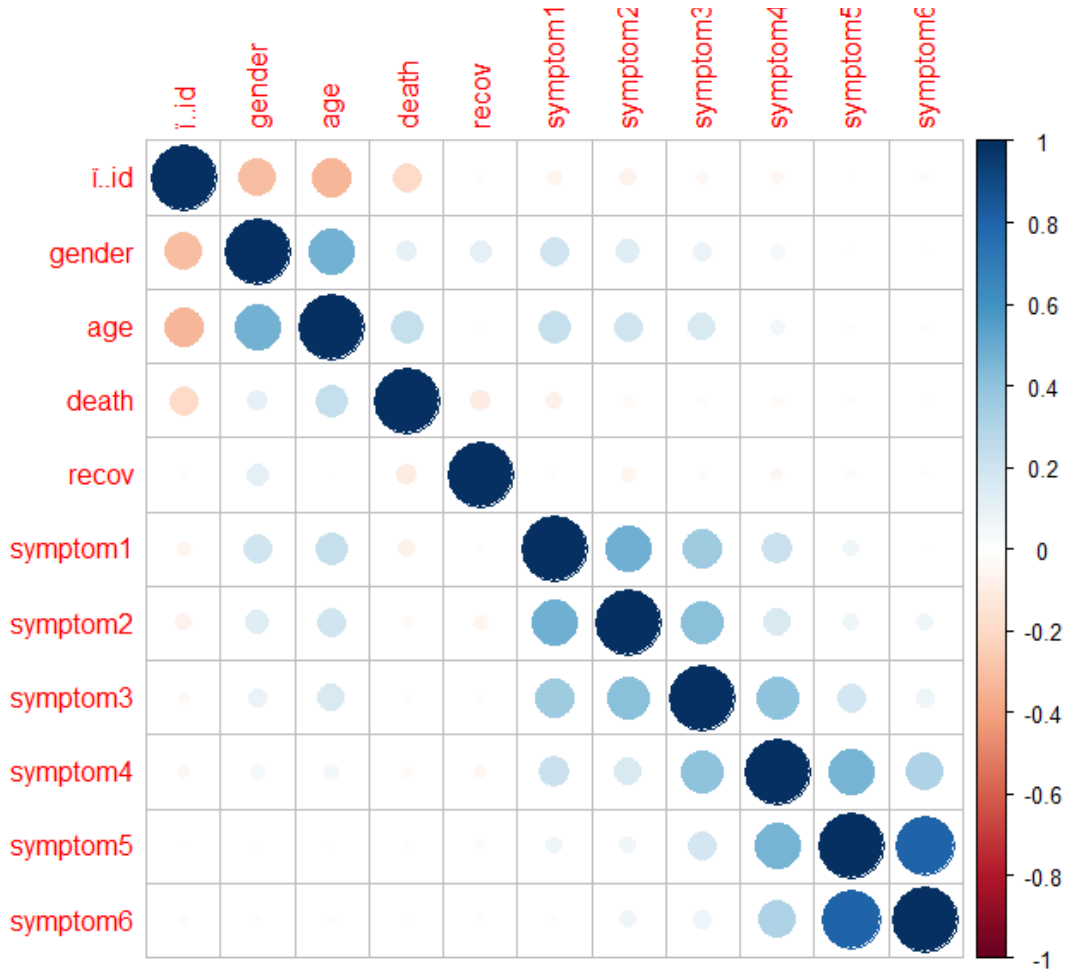

Figure 4

Co-relation plot of various attributes

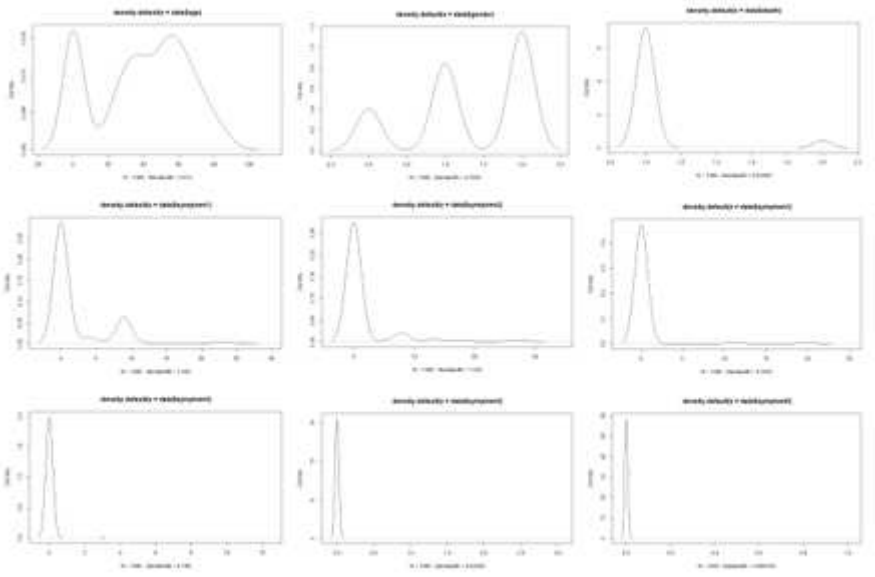

Figure 5

Density plot of various attributes

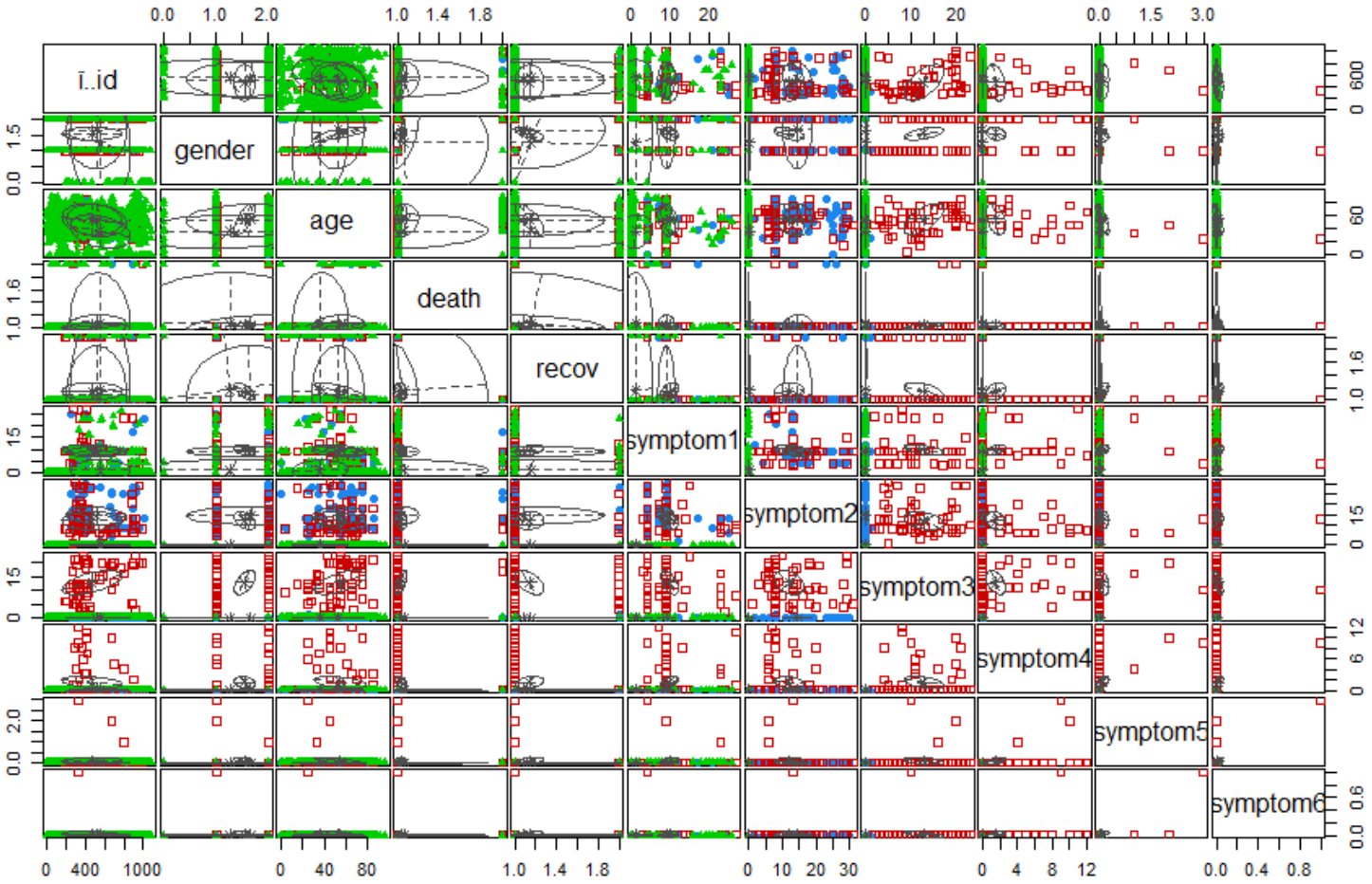

Figure 6

Clustering Classification

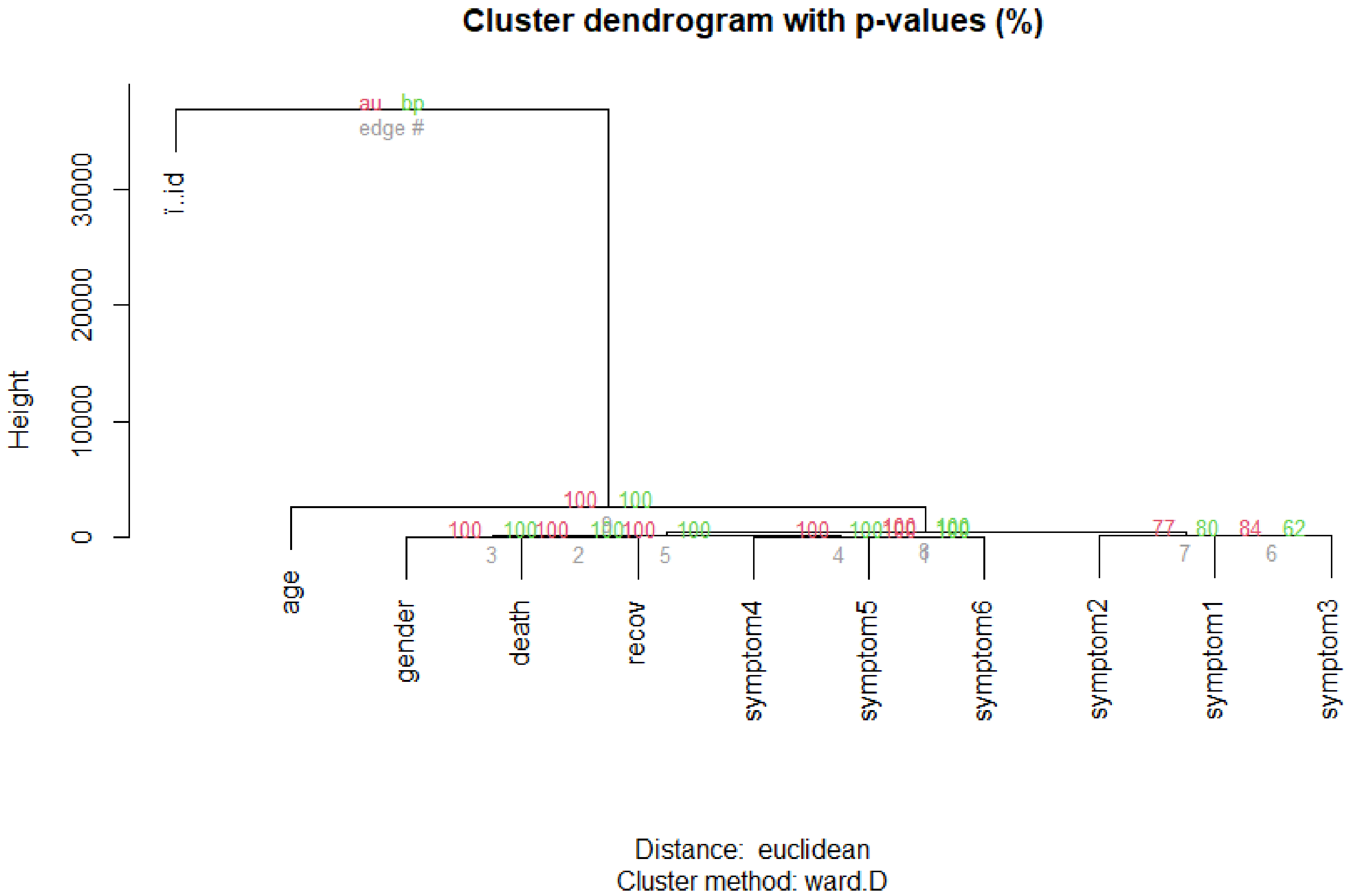

Figure 7

Clustering Dendrogram

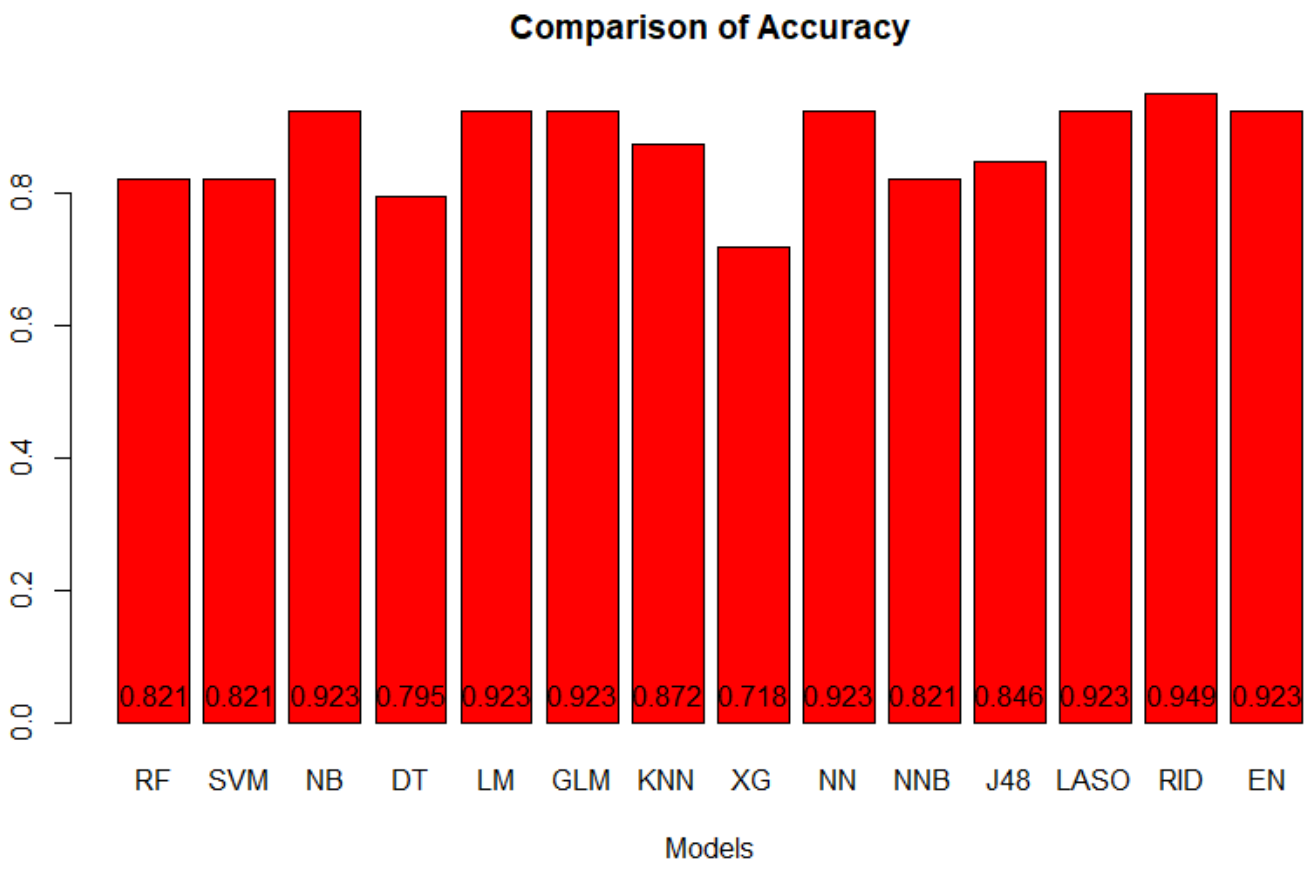

Figure 8

Accuracy comparison without filters

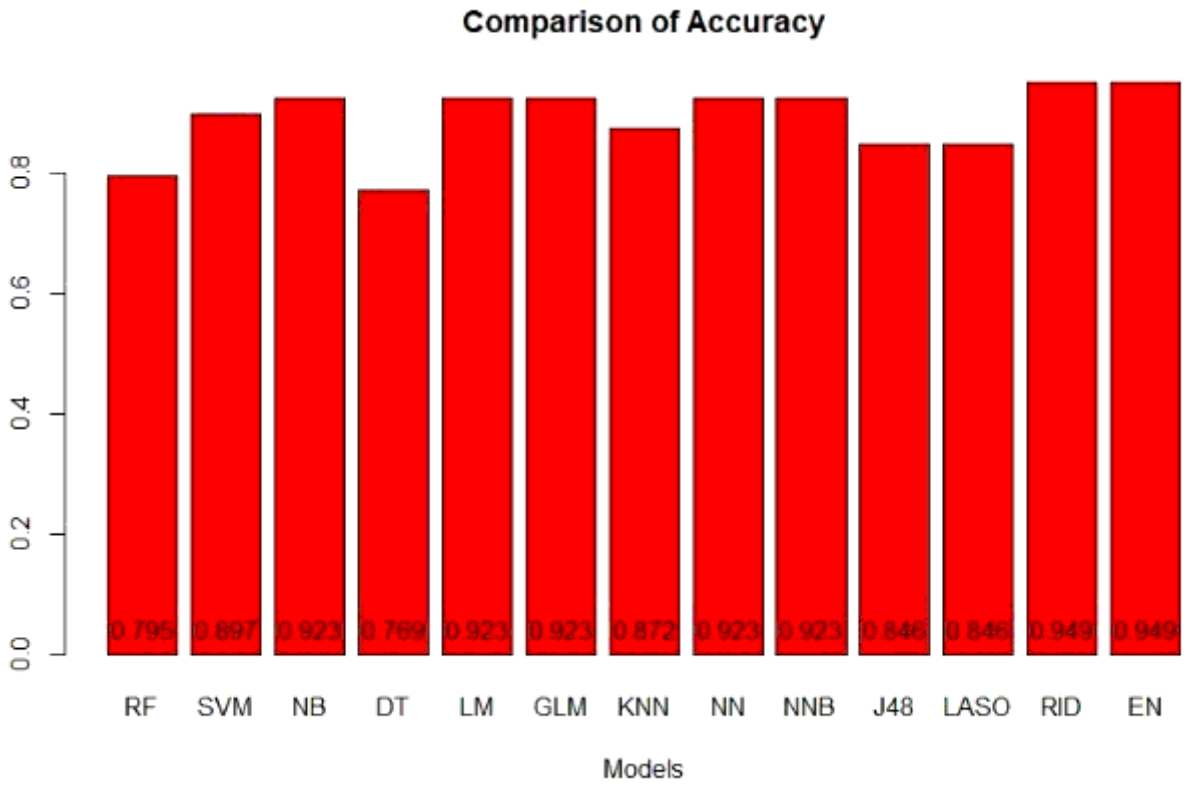

Figure 9

Accuracy comparison with XGBoost

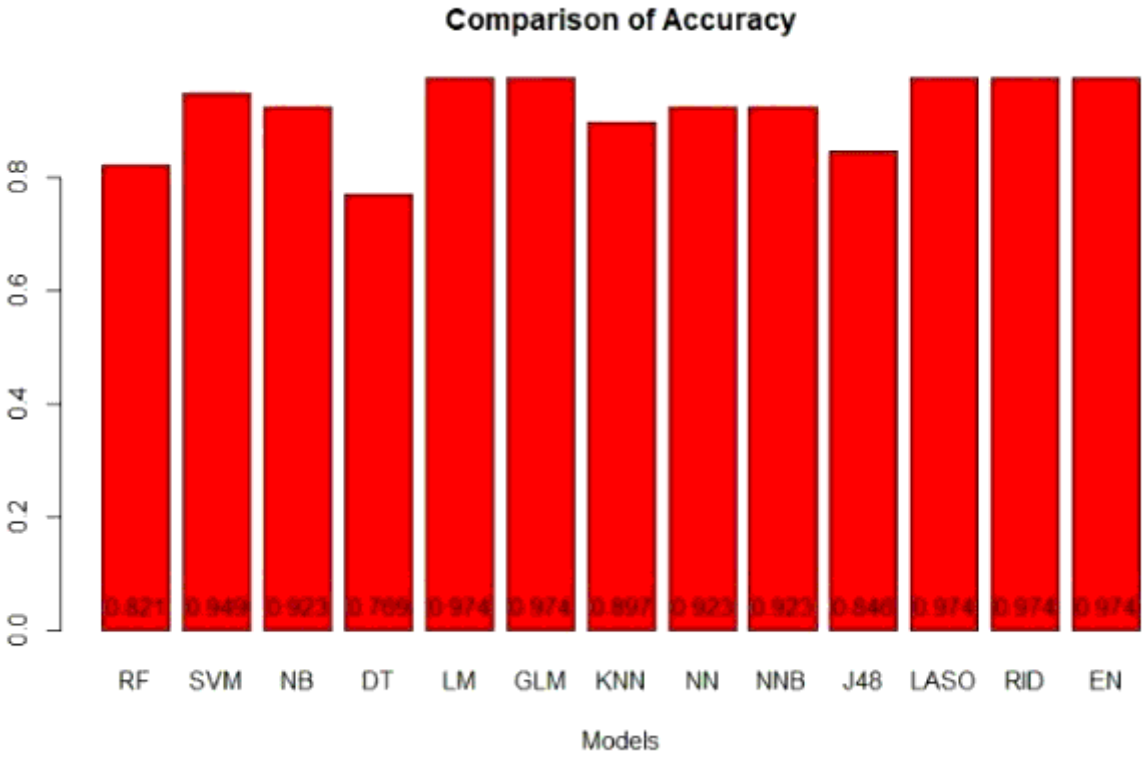

Figure 10

Accuracy comparison with Boruta

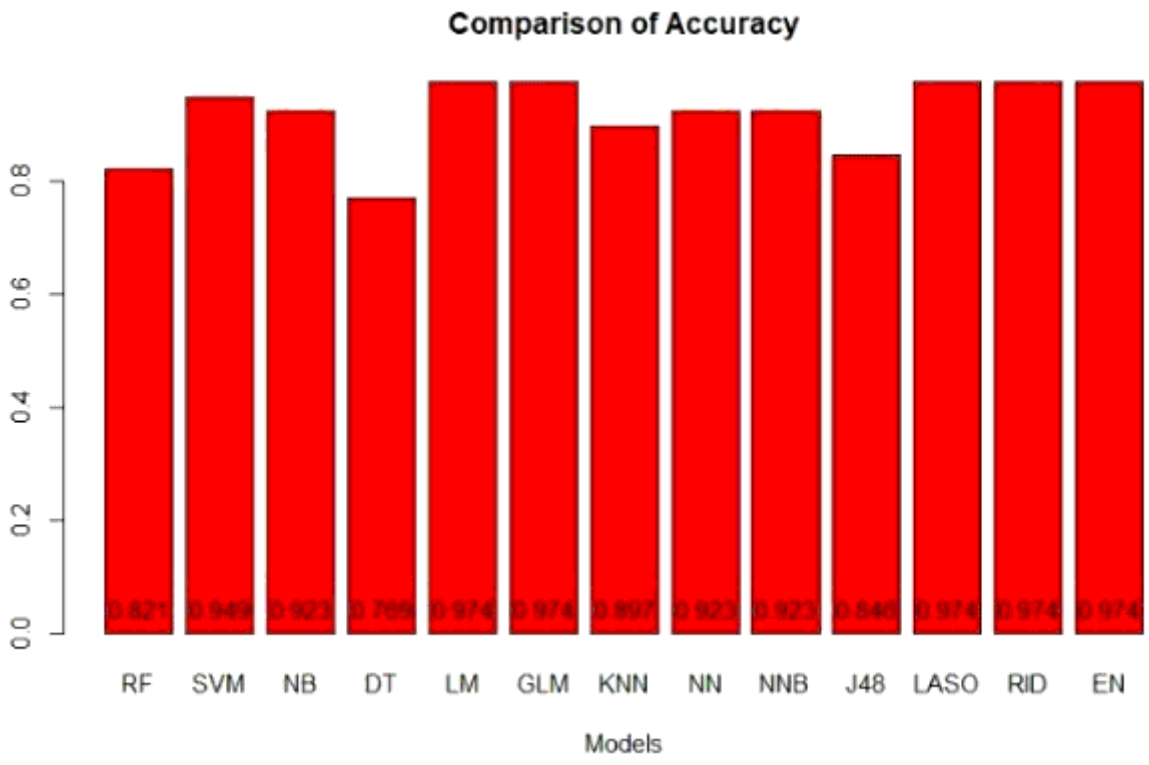

Figure 11

Precision comparison with Boruta

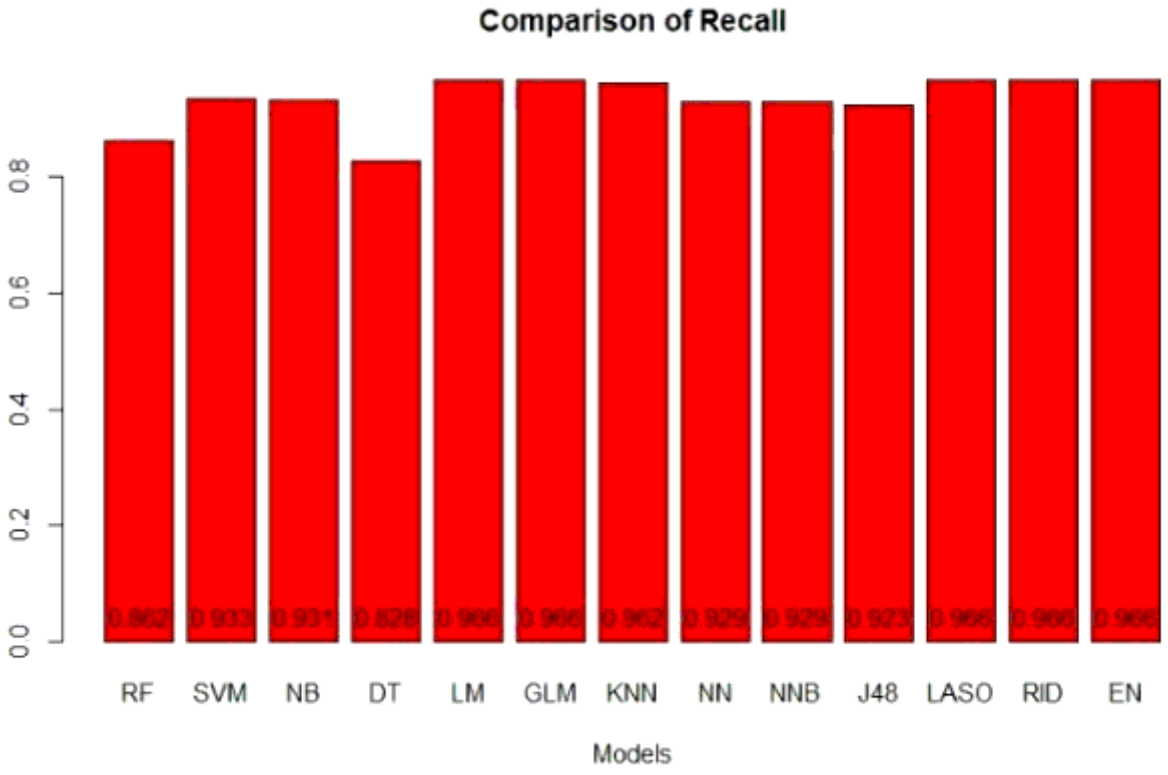

Figure 12

Recall Score comparison with Boruta

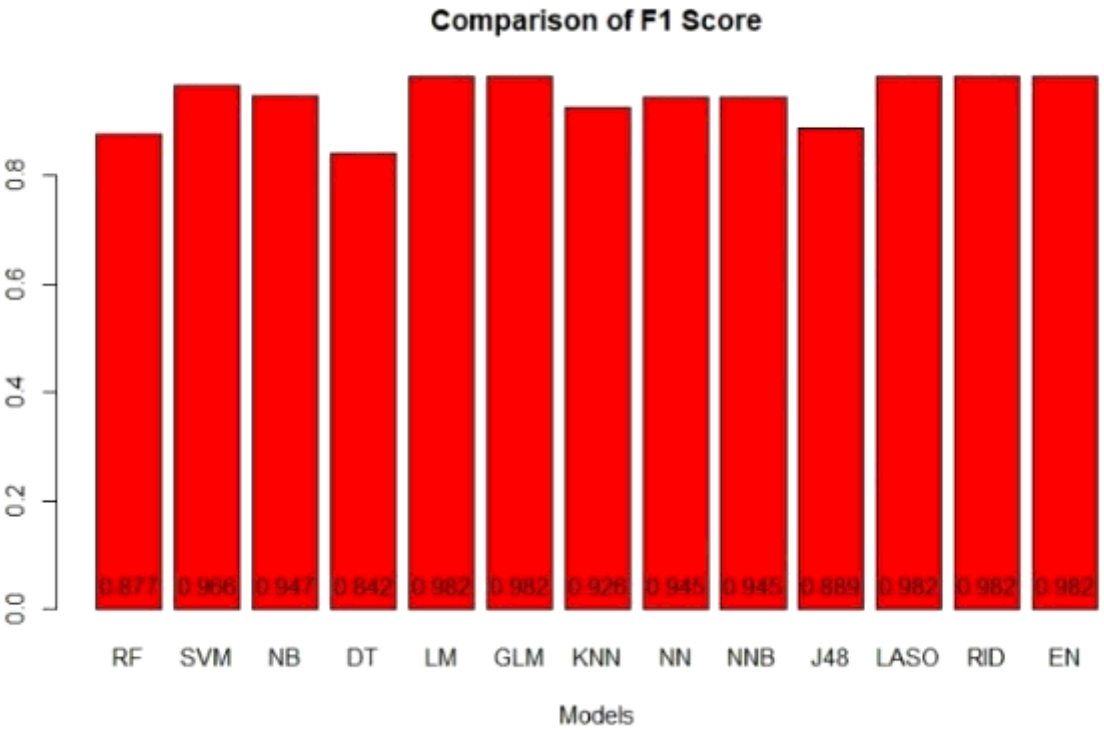

Figure 13

F1 Score comparison with Boruta

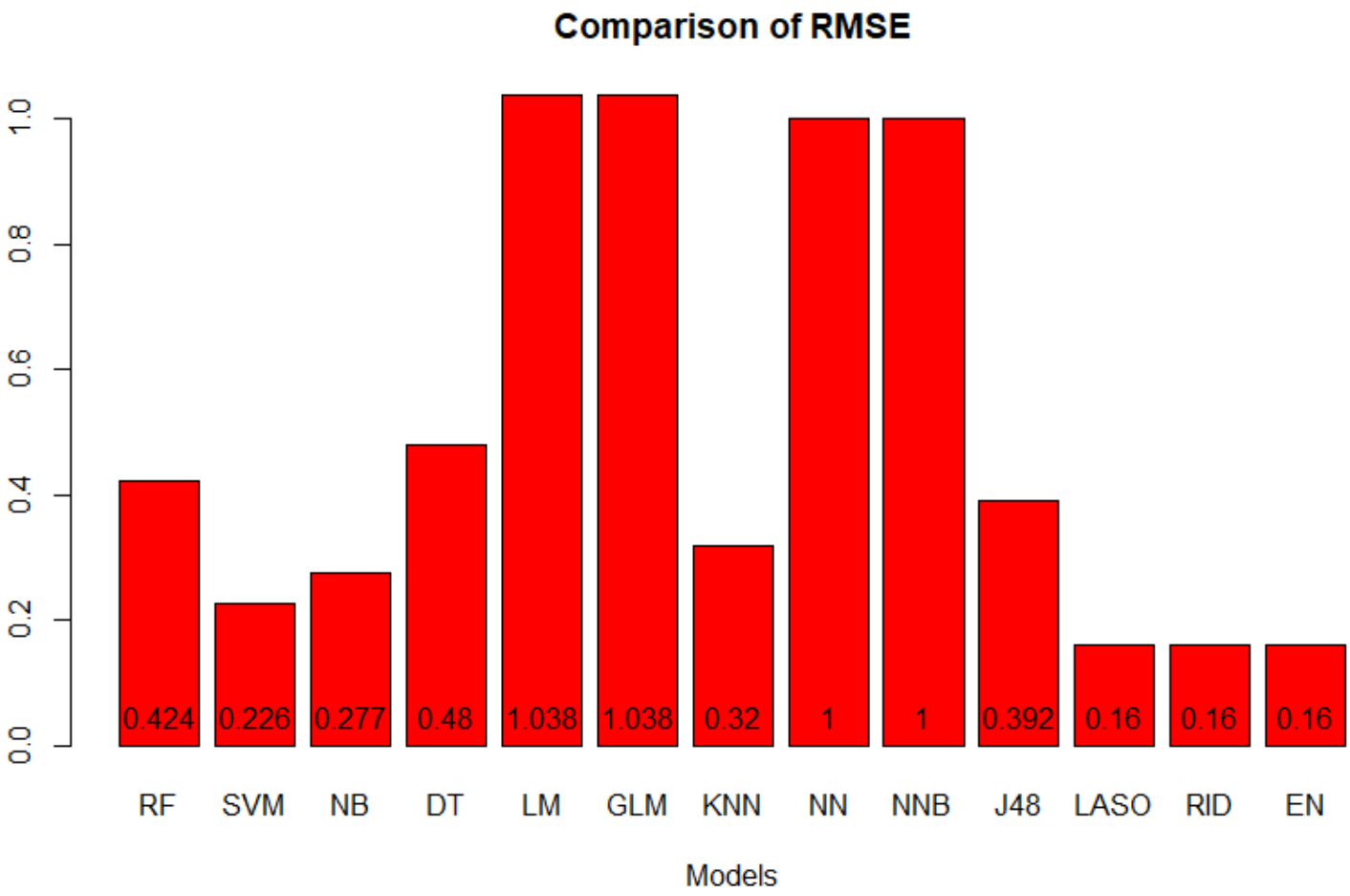

Figure 14

RMSE comparison with Boruta

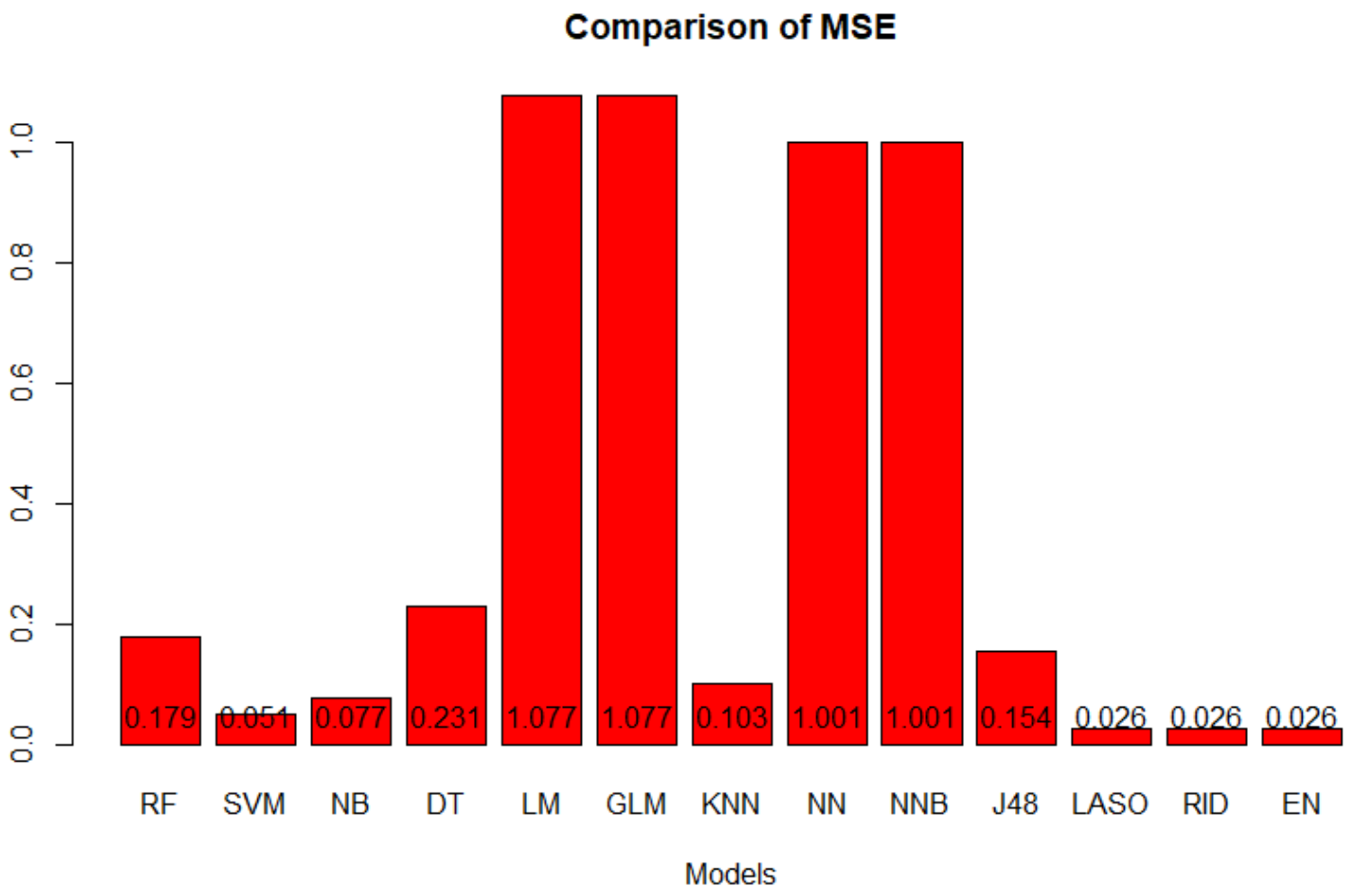

Figure 15

MSE comparison with Boruta

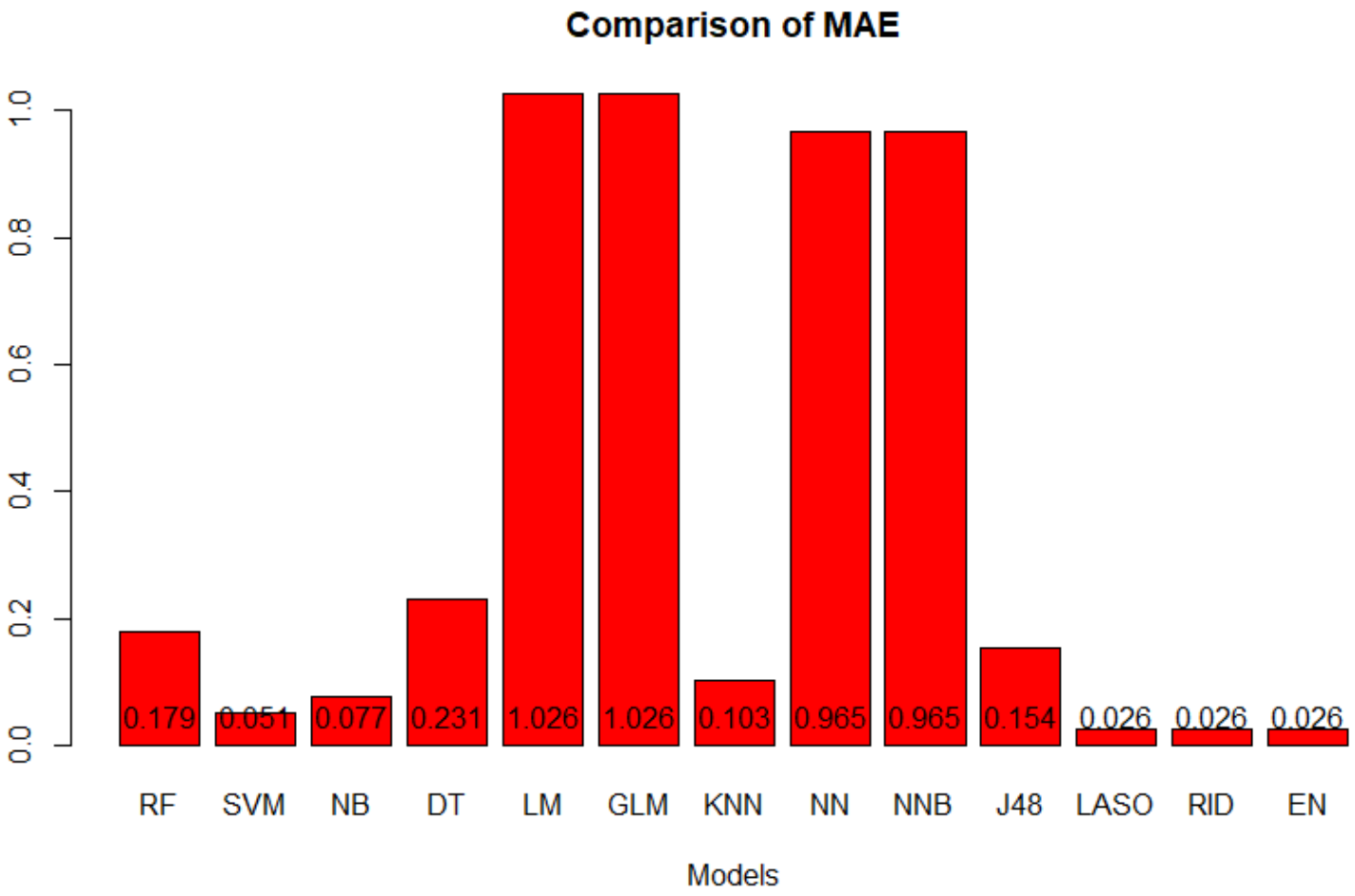

Figure 16

MAE comparison with Boruta