Date of current version December 14, 2022.

Disease Prediction by Juxtaposition of Multiple Machine Learning Models

Archit Jain[1], Faizan Ahmed Mohammed[1], and Kapil Singh Baghel[1]
[1]Computer Engineering undergraduate at Netaji Subhas University of Technology, Dwarka

ABSTRACT The field of healthcare has a vast application of machine learning in this era. Traditional ways of diagnosing a disease can only consider a limited number of factors, but with the upcoming technologies, we can consider thousands of factors in the matter of seconds. Machine learning helps in making data-driven decisions relating to key trend and driving research efficiency. We find the highest accuracy of prediction in case of multilayer perceptron with ReLU activation.

1. INTRODUCTION

With advancements in the medical field, world class treatment is reaching the patients and is saving lives. Machine learning is playing a central role in the same with its own significant developments easing its application. Machine learning algorithms can make predictions after considering many variables unlike traditional methods of diagnosing which has a limit on the factors that can be considered [11].
The dataset used in this report has symptoms represented by binary 0 or 1 signifying whether that symptom is observed or not and based on those values we form a prognosis.

We follow various steps to reach our conclusion which are depicted by the flowchart shown in Fig 1 below:

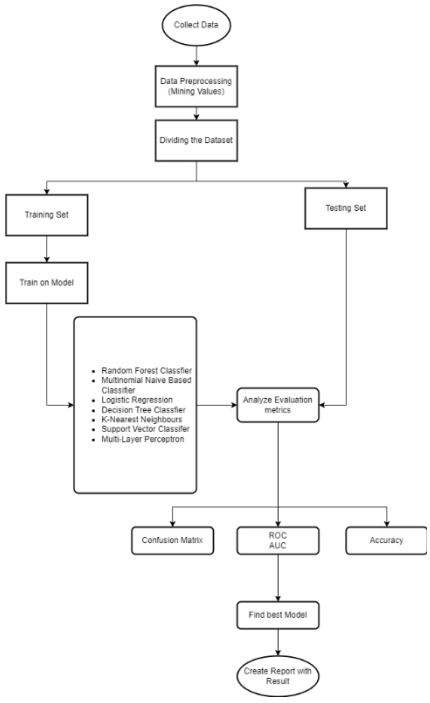


Fig 1- Flow of steps to compare ML models

a) We read the data and visualize it using bar plot. We also check for missing values and handle, if any.
b) Original dataset is then split into two sets- training and testing.
c) Our model is trained using training set. The accuracy of each model is scored on the testing set by manipulating the hyperparameters of each.
d) We plot confusion matrix using heatmap and use ROC curve (receiver operator characteristic curve) to find AUC (area under curve).
e) We compare accuracy and AUC values of different models to find the best one [6].
f) We display the predictions made by the best model on the test set.

The major challenge in this field is extracting appropriate information from our dataset and using them in an efficient way so that we can handle cases of underfitting and overfitting. This is also a sensitive field; medical conditions should not be completely reliant on machine learning at its current level but it serves as an efficient first or early indicator of a potential disease. This facility would also significantly lower disease detection costs and make it more widely available to the public. In this report, we compare the performance of various models on our dataset and check which model suits best for our purposes.

2. RELATED WORK

This section will discuss some of the existing works on different machine learning techniques related to disease prediction.

There are many factors which are related to cause and treatment of diabetes disease such as age, health history, obesity, immune system strength, weakness, and many more [1]. The objective of this study is to implement different model to also find the best fit with significant accuracy to diagnose diseases such as diabetes in patients [7]. Various machine learning algorithms are also utilized for early-stage

diabetes detection which include Random Forest Classifier [1], Support Vector Machine [4], Decision Trees [7], Logistic Regression,[5] K-Nearest Neighbors [3], Naïve Bayes Classifier [10], Multi-Layer Perceptron (MLP) [8].

Currently the approach to predict cardiovascular risk to not so appropriate, Machine-learning gives us the chance to improve accuracy by taking advantage of the complicated relationship between risk factors [2]. This paper [2] studies 24,970 incident cardiovascular events (6.6%) occurred and compared the different model with their accuracy like random forest +1.7%, logistic regression +3.2%, and neural networks +3.6%. The best performing algorithm was neural networks which predicted a total of 4,998/7,404 cases (sensitivity 67.5%) and 53,458/75,585 non-cases (specificity 70.7%), correctly predicting 355 (+7.6%) [2].

Focus is on implementation and study of performance by models such as Naive Bayes, K-Nearest Neighbor (KNN) and Random Forest classifier [1] based on the accuracy and preciseness for chronic kidney disease or CKD prediction [3][6]. The result of conducting the research is that the performance of Random Forest classifier is relatively better than both Naive Bayes and KNN [3].

The purpose of implementation of support vector machine (SVM) is to develop decision support system to diagnosis kidney disease patient [4]. Also, this paper focuses on methodology which consist of classification modelling and system development. Steps involved in a classification model consists of data collection, its preparation and grouping, and then finally classification [6]. The study resulted in a trained model which can detect a chronic condition of kidney disease based on several factors on SVM with an outstanding accuracy of 98.34% [4].

Individual patient survival often depends on a complicated relationship between multiple variables like symptoms of kidney failure, causes of kidney disease [4], medications, and their interventions in case of Kidney Disease Prediction [10]. Three data mining techniques (Artificial Neural Networks (ANN), Decision tree and Logical Regression [5]) are used to evaluate the interaction between these variables and the rate of patient's survival. The performance comparison of three of them are studied for extracting knowledge in the form of classification rules from the data. [5][6].

Clinical decision support systems have also been installed that combine various data mining techniques for prediction of diseases such as diabetes and study its progression and performance to various techniques on dataset [6].

The above discussed studies give us a good insight on the implementation of data mining into healthcare and the aim of this study is to give a valuable cumulative insight as well.

3. THEORY

3.1 Machine Learning Models

1. Random Forest Classifier

Random Forest [3] is a frequently used ML algorithm belonging to the class of supervised learning techniques. This model can be used for both types of problems-Classification and Regression.

As show in Fig 2, we observe multiple decision trees on multiple subparts of a given dataset. Each decision tree below predicts an output (accuracy is predicted in our report). Then we take their average to improve the accuracy of the overall prediction from the dataset.

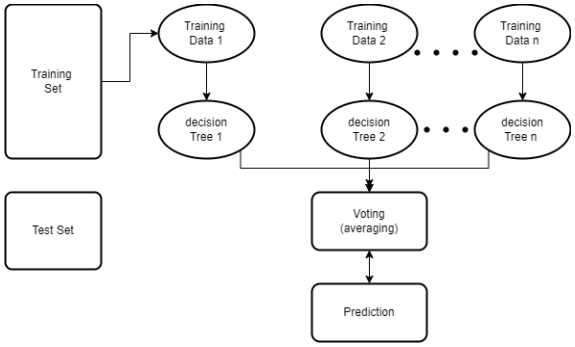


Fig 2- Working of the Random Forest ML model

2. Multinomial Naïve Bayes Classifier

The Multinomial Naïve Bayes classifier [3] is used in the case of multinomial distributed data. This classifier assumes no dependency between attributes ie. all attributes are considered independent. This algorithm uses the concept of conditional independence with the assumption that the attribute value of a given class is independent from the values in other attributes. It is primarily used for document classification. The prediction is based on the conditional probability of an object as discussed below [10].

$$P(h|D) = \frac{(P(\mathrm{D}|\mathrm{h}) \,.\, P(h))}{P(D)} \qquad (Eqn\ 1)$$

P(h|D) is known as the posterior probability: Conditional probability of the hypothesis h on observed data D.

P(D|h) is known as the likelihood probability: It is the probability of data D given that probability of a hypothesis h being true.

P(h) is Prior Probability: Probability of hypothesis h before the data is observed.

P(D) is Marginal Probability: Probability of given evidence or data.

3. Logistic Regression

Logistic regression is a widely used algorithm whose main purpose is to predict the categorical dependent variable using given independent variables [5]. Calculated probabilistic values lie in the range 0 and 1. In the figure below the y value 0.8 indicates that the probability of that event occurring is 80%. It uses a sigmoid function to plot on graph as shown in Fig 3.

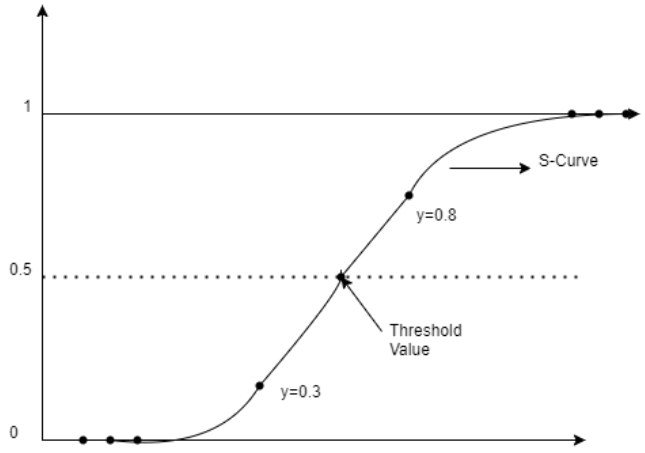


Fig 3- Graphical representation of Logistic Regression

$$\log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + \cdots + b_nx_n \quad (Eqn\ 2)$$

4. Decision Tree Classifier

It is an easy to implement, simple and widely used classifier. High dimensional data can be efficiently handles and it doesn't require any previous domain knowledge or parameter setting [7]. The results produced by our classifier are easily interpretable and readable because of its flowchart-like nature.

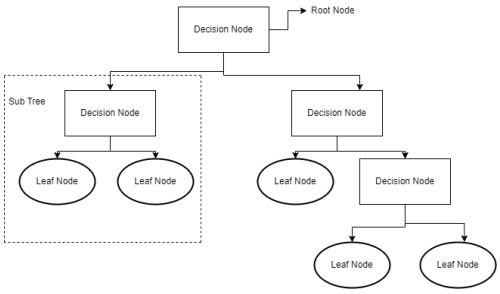


Fig 4- Model of a Decision Tree

It basically lists out a set of all possible outcomes which could later lead to more outcomes. In the above figure, the node which cannot be further broken into more noes is known as the leaf node whereas a decision node could be broken into many other nodes.

5. K-Nearest Neighbors

KNN [3] is one of the simplest algorithms that comes under the category of supervised learning. The working principle of this algorithm is how well new data is resembled to available data. The new data blend to that cluster to which it gets most resemblance. Fig 5 below shows the KNN algorithm in a graph.

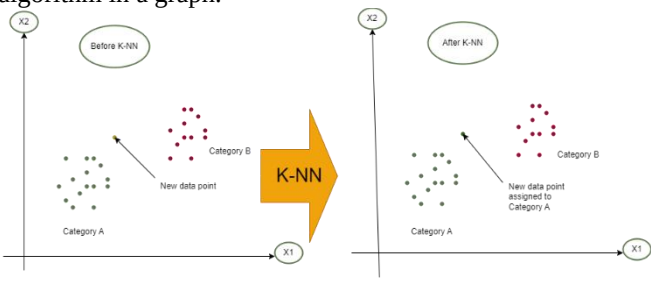


Fig 5- Graphical representation of KNN Algorithm

When a new data point is entered, the algorithm selects K neighbors and calculates their Euclidean distance between them, also, counting the number of datapoints in each of the given categories. Then it assigns a the category to the new datapoint for whom the number of neighbors is in majority.

6. Support Vector Classifier

Support Vector Machine algorithm [4] is one of the most popularly used algorithms in the category of supervised machine learning. It could be used for both applications- Regression and Classification.

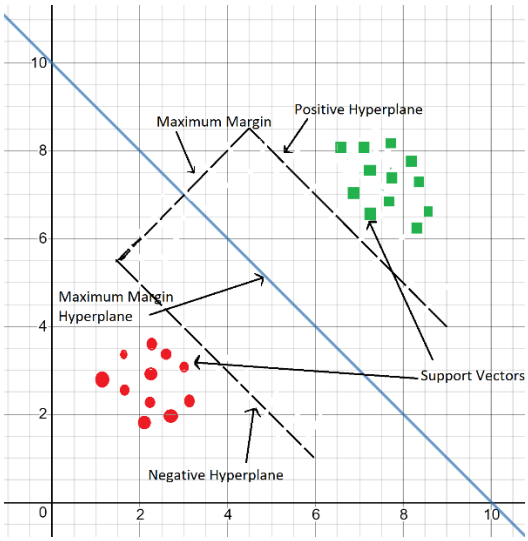


Fig 6- Understanding of Support Vector Algorithm

The main goal of Support Vector Algorithm is creating a best fit decision boundary line to segregate n-dimensional space into separate classes which will help us easily classify the new unseen data point in its correct category. This best decision boundary is called hyperplane which helps to classify data points whereas support vectors are datapoints closest to hyperplane.

7. Multi-Layer Perceptron (Neural Network)

Multi-Layer Perceptron (MLP) is one of the most complicated approaches in supervised learning. It deals with many layers of input nodes to give an output data. MLP has

many layers which are interconnected to each other and with this multilayered structure, we get our desired output. [8].

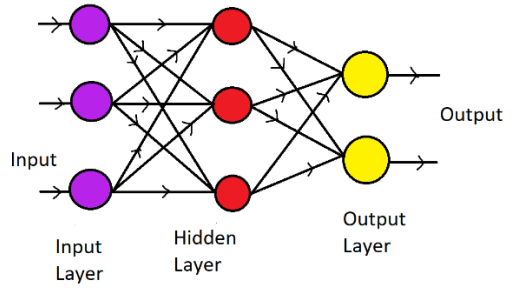


Fig 7- Multi-Layer perceptron (Neural Network)

As we can see, many nodes are connected to each other forming a very complex structure of data. Such a network of nodes can only be achieved from supervised learning approach.

A MLP has one input layer for each input. Between output layer and input layer, there can as many as hidden layers and nodes as needed to get the desired output.

a) **Logistic Activation Function**
In this algorithm, we measure the dependent and independent features of a dataset.

$$f(x) = \frac{L}{1+e^{-kx}} \qquad \textit{(Eqn- 3)}$$

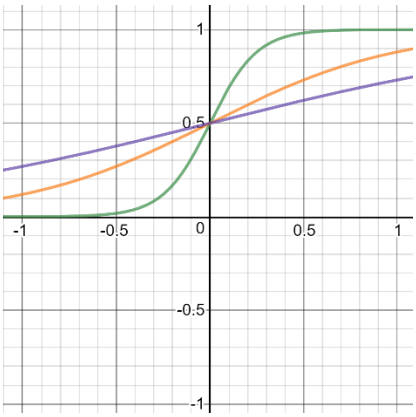


Fig 8- Logistic Activation Function graph

Input is any real value and the output ranges between 0 and 1. Output is dependent on the input i.e., the greater the input the closer is the output to 1.

b) **Rectified Linear Unit**
The rectified linear unit or ReLU for short is an activation function which is a linear function that will give the input itself as the output if positive and zero otherwise. This model is used as the default activation function for many types of neural networks because of its better performance facilitated by its easy to train perk.

The equation is given by,

$$f(x) = \max(0,) \qquad \textit{(Eqn– 4)}$$

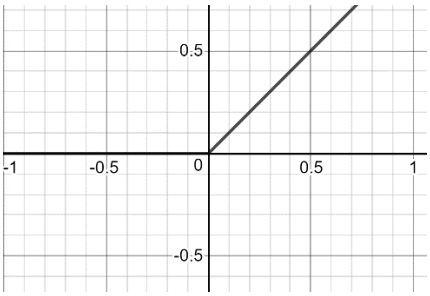


Fig 9- ReLU Activation Function graph

3.2 Evaluation Metrics

When creating a model or a group of models it is also important to make sure that they are working sufficiently well. For this we use evaluation metrics. The efficiency of various models is evaluated using various metrics. The evaluation metrics that we would be using are confusion matrix and ROC curve and its AUC [6].

1. Confusion Matrix
A confusion matrix is a metric to determine the performance of our classifier but it can only be determined for situations where the actual values for test data are known as well. It is also known as an error matrix.

		ACTUAL VALUES	
		Positive	**Negative**
PREDICTED VALUES	**Positive**	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Table 1- Confusion Matrix Representation

The calculations that can performed using a confusion matrix are:

Accuracy – Defines how correctly our model is working in predicting the right class.

$$Accuracy = \frac{TP+TN}{FP+FN+TP+TN} \qquad \textit{(Eqn- 5)}$$

Error Rate – Defines how frequently our model gives inaccurate predictions.

$$Error\ Rate = \frac{FP+FN}{FP+FN+TP+TN} \qquad \textit{(Eqn- 6)}$$

Precision - Defined as ratio of the correctly identified in the positive class to the total positive identified in the positive class.

$$Precision = \frac{TP}{TP+FP} \qquad (Eqn\text{-}7)$$

Recall – Tells us how many were predicted correctly by our model out of the total positive classes.

.

$$Recall = \frac{TP}{TP+FN} \qquad (Eqn\text{-}8)$$

2. ROC Curve

A receiver operating characteristic curve, or ROC curve, is a graph that gives us the performance of our model at all possible threshold values. The curve is plotted as TPR vs FPR at different thresholds of classification.
Thus, the curve consists of two parameters to plot:

a. True Positive Rate (TPR)

b. False Positive Rate (FPR)

True Positive Rate (TPR) is a same as recall and can be expressed as shown below:

$$TPR = \frac{TP}{TP+FN} \qquad (Eqn\text{-}9)$$

False Positive Rate (**FPR**) can be expressed as shown below:

$$FPR = \frac{FP}{FP+TN} \qquad (Eqn\text{-}10)$$

AUC is short for "Area under the ROC Curve." AUC is a measure of the entire area under the ROC curve from (0,0) to (1,1).

4. METHODOLOGY

4.1 Dataset preparation

The dataset used here is available on Kaggle. It includes 4920 rows and 133 columns. All the columns are symptoms with a binary value of 0 or 1 representing whether that symptom is observed in that data value or not. Then we have a prognosis of the disease along with it. As in our case, classification modelling will consist of data collection, preparation, grouping, classification, and extraction rules [4].

4.2 Data pre-processing and cleaning

First, we need to pre-process our dataset, that is, turn it into usable format [12]. We check for missing values, and find none. Had we found missing values, we can handle them in two ways:

1. **Deleting missing values**
   If missing value is of type MNAR (missing not at random), then do not delete it. If of type MCAR (missing completely at random), then delete it. If there are multiple missing values, we can even consider deleting the entire row.

2. **Imputing missing values**
   We can make a calculated guess about the missing value and replace the missing value in our column with the same.

   a. Replacing with mean- This is most used for numeric columns.

   b. Replacing with mode- Most occurring value is imputed.

   c. Replacing with median- The middlemost value is used for imputing.

   d. Most frequent value imputed for categorical data

   e. Impute the value "missing", which will be considered separately.

4.3 Model Training

We do implementation of various models to evaluate our dataset [2]. While doing so we use various performance metrics [6] for evaluation such as confusion matrix, AUC of ROC curve along with the accuracy of model performance is noted for comparison.

The basic steps involved in implementing any model are:

1. Choosing the correct machine learning model.
2. Training the model on our training dataset.
3. Evaluating the trained model using various evaluation metrics such as accuracy, or confusion matrix, or area under ROC curve.

Often the parameters of the model need to be tuned to handle the cases of overfitting and underfitting to give us the most ideal set of tuned parameters which help us fit our data well onto our model.

The unseen data or new data (symptoms of a new patient, in our case) are fed to the trained model to make a prediction.

For the model showing the highest accuracy (Multilayer Perceptron, ReLU activation), we can see the evaluation metrics as illustrated below:

ACCURACY

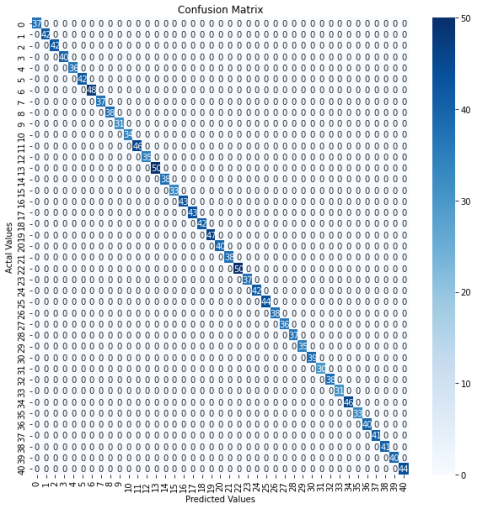
```
mlp2=MLP(hidden_layer_sizes=(50,),activation='relu',max_iter=12)
mlp2.fit(x_train, y_train)
accuracy = mlp2.score(x_test, y_test)
global_accuracy['Multilayer Perceptron ReLU Activation']=accuracy
print('Accuracy is: ', end='')
print(accuracy)
```

```
Accuracy is: 1.0
```

(Code sample #1)

CONFUSION MATRIX

```
y_pred=mlp2.predict(x_test)
cm = confusion_matrix(y_test, y_pred)
cm_df = pd.DataFrame(cm)
plt.figure(figsize=(10,10))
sns.heatmap(cm_df, annot=True, cmap='Blues')
plt.title('Confusion Matrix')
plt.ylabel('Actal Values')
plt.xlabel('Predicted Values')
plt.show()
```



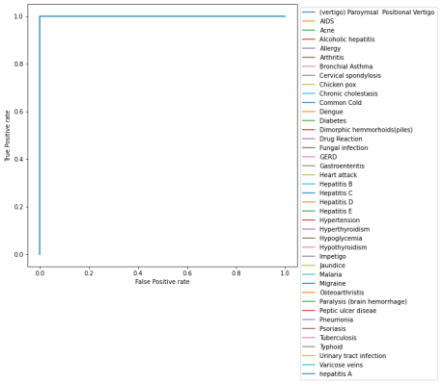
(Code sample #2)

ROC CURVE

```
plt.figure(figsize=(8,8))
y_bin=label_binarize(y_test,classes=np.unique(y_test))
false_pos_r={}
true_pos_r={}
th={}
auc_val={}
pred_prob=mlp2.predict_proba(x_test)
unique_cl=np.unique(y_test)
for i in range(len(unique_cl)):
    false_pos_r[i],true_pos_r[i],th[i]=roc_curve(y_bin[:,i],pred_prob[:,i])
    auc_val[i]=auc(false_pos_r[i],true_pos_r[i])
    plt.plot(false_pos_r[i],true_pos_r[i])
plt.xlabel("False Positive rate")
plt.ylabel("True Positive rate")

print('Average area under curve is: ', end='')
avg_auc=statistics.mean(list(auc_val.values()))
global_auc_val['Multilayer Perceptron ReLU Activation']=avg_auc
print(avg_auc)

plt.legend([i for i in unique_cl], bbox_to_anchor=(1, 1))
plt.show()
```



(Code sample #3)

4.4 Results and Evaluation

MODEL	ACCURACY	AUC
Logistic Regression	0.95443	1.00000
Support Vector Machine	0.96613	1.00000
Multinomial Naïve Bayes	0.95135	0.99976
K-Nearest Neighbors	0.96182	1.00000
Random Forest Classifier	0.98830	0.99983
Decision Tree	0.96490	0.99949
Multilayer Perceptron (Logistic Activation)	0.96490	0.99865
Multilayer Perceptron (ReLU Activation)	1.00000	1.00000

Table 2- Accuracy and AUC values of various models

The predictions as observed by ReLU activation is seen below:

```
y_pred = mlp2.predict(x_test)
pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
```

	Actual	Predicted
373	Acne	Acne
4916	Acne	Acne
1550	Hyperthyroidism	Hyperthyroidism
3081	AIDS	AIDS
3857	Chronic cholestasis	Chronic cholestasis
...	...	...
1257	GERD	GERD
3346	Tuberculosis	Tuberculosis
3384	Hepatitis D	Hepatitis D
3290	Hypertension	Hypertension
1178	Arthritis	Arthritis

1624 rows × 2 columns

(Code sample #4)

5. CONCLUSION

In our report, we compared different algorithms such as Logistic Regression, Random Forest, Naïve Bayes, KNN, Support Vector Machine (SVM), Decision Tree, and Multilayer Perceptron based on AUC of its ROC curve and accuracy evaluation metrics.

We observe that Multilayer Perceptron Model using ReLU activation shows highest accuracy of 1.0 ie. 100% correct predictions on our test dataset along with high area under curve. One reason of that could be high number of columns (which have symptoms) that need to be looked at and neural networks perform well for those situations unlike decision trees which need to make a decision at every column which increases the depth of our tree (if depth is less, accuracy suffers and if depth is more, data overfits in our tree).

It was also observed that our dataset was uniform with same number of tuples for each disease as was evident from the bar plot. Manipulating the hyperparameters in different models was showing that the data was fitting very well, so, we can say that the differences between the observed values and the model's predicted values are small and unbiased.

In some cases, we also observe that the AUC value is large, whereas accuracy is relatively lower. One reason why this might happen is if our classifier is achieving its good performance on the positive class (or high AUC) at the expense of high false negative rate (or high FNR). That is, the ROC analysis tells us something about how well the positive class sample is separated from other classes, whereas the prediction accuracy gives us a hint on the actual performance of the classifier.

The Colab Notebook for entire implementation along with dataset is available at the link below:

https://drive.google.com/drive/folders/1vlzjAbCLErwC3C-rDfQW6clUXVxyDxQC?usp=sharing

5.1 Future Scope

There are different techniques which can be further applied to improve the performance even more [9] to facilitate effective and early detection of diseases.

1. A larger, more diverse dataset can improve the factors being taken into consideration unlike traditional approach which is limited in those terms.

2. Combining models can improve accuracy and efficiency covering for each other's limitations and giving a more reliable outcome.

6. REFERENCES

[1] Palimkar, Prajyot, Rabindra Nath Shaw, and Ankush Ghosh. "Machine learning technique to prognosis diabetes disease: random forest classifier approach." In Advanced Computing and Intelligent Technologies, pp. 219-244. Springer, Singapore, 2022.

[2] Weng, Stephen F., Jenna Reps, Joe Kai, Jonathan M. Garibaldi, and Nadeem Qureshi. "Can machine-learning improve cardiovascular risk prediction using routine clinical data?." PloS one 12, no. 4 (2017): e0174944.

[3] Devika, R., Sai Vaishnavi Avilala, and V. Subramaniyaswamy. "Comparative study of classifier for chronic kidney disease prediction using naive bayes, KNN and random forest." In 2019 3rd International conference on computing methodologies and communication (ICCMC), pp. 679-684. IEEE, 2019.

[4] Ahmad, Mubarik, Vitri Tundjungsari, Dini Widianti, Peny Amalia, and Ummi Azizah Rachmawati. "Diagnostic decision support system of chronic kidney disease using support vector machine." In 2017 second international conference on informatics and computing (ICIC), pp. 1-4. IEEE, 2017.

[5] Lakshmi, K. R., Y. Nagesh, and M. Veera Krishna. "Performance comparison of three data mining techniques for predicting kidney dialysis survivability." International Journal of Advances in Engineering & Technology 7, no. 1 (2014): 242.

[6] Perveen, Sajida, Muhammad Shahbaz, Aziz Guergachi, and Karim Keshavjee. "Performance analysis of data mining classification techniques to predict diabetes." Procedia Computer Science 82 (2016): 115-121.

[7] Orabi, Karim M., Yasser M. Kamal, and Thanaa M. Rabah. "Early predictive system for diabetes mellitus disease." In Industrial Conference on Data Mining, pp. 420-427. Springer, Cham, 2016.

[8] Zhang, Hanyu, Che-Lun Hung, William Cheng-Chung Chu, Ping-Fang Chiu, and Chuan Yi Tang. "Chronic kidney disease survival prediction with artificial neural networks." In 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 1351-1356. IEEE, 2018.

[9] Singh, Smriti Mukesh, and Dinesh B. Hanchate. "Improving disease prediction by machine learning." Int. J. Res. Eng. Technol 5 (2018): 1542-1548.

[10] Dulhare, Uma N., and Mohammad Ayesha. "Extraction of action rules for chronic kidney disease using Naïve bayes classifier." In 2016 IEEE International Conference on

Computational Intelligence and Computing Research (ICCIC), pp. 1-5. IEEE, 2016.

[11] K. Shailaja, B. Seetharamulu and M. A. Jabbar, "Machine Learning in Healthcare: A Review," 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2018, pp. 910-914, doi: 10.1109/ICECA.2018.8474918.

[12] S. Mohan, C. Thirumalai and G. Srivastava, "Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques," in *IEEE Access*, vol. 7, pp. 81542-81554, 2019, doi: 10.1109/ACCESS.2019.2923707