Bangabandhu Sheikh Mujibur Rahman Science and Technology University

Abnormal baby delivery prediction using machine learning techniques.

By

Umma Sabikun Nahar :17CSE006
Md Sakib Hossain : 17CSE014

April ,2023

Department of Computer Science & Engineering

Bangabandhu Sheikh Mujibur Rahman Science and Technology University

Abnormal baby delivery prediction using machine learning techniques.

This project is submitted in the fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering (B.Sc. Engg. in CSE)

Umma Sabikun Nahar :17CSE006
Md Sakib Hossain : 17CSE014

Supervised By
Mrinal Kanti Baowaly
Associate Professor
Department of Computer Science & Engineering
Bangabandhu Sheikh Mujibur Rahman Science & Technology University

ABSTRACT

The Birth of an abnormal child is so painful for their parents. The birth of an abnormal child may have happened because of some daily behavior of their parents.we are trying to identify which behavior has much impact on the birth of an abnormal child.For this purpose we use some machine learning techniques to analyze the data.By applying this technique any people can get predictions is they able to born a normal child or not.They can identify which behavior is much affect to born an abnormal child.If they can change their behavior it’s maybe possible to born a normal child.

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty Allah for His divine blessing makes it possible to complete the final year project/thesis successfully.
We are really grateful and wish our profound indebtedness to **Mrinal Kanti Baowaly** Associate professor .Department of CSE, BSMRSTU, Gopalganj. The deep knowledge & keen interest of my supervisor in this field is helpful to carry out the project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete.

Umma Sabikun Nahar :17CSE006
Md Sakib Hossain : 17CSE014

CHAPTER 1

INTRODUCTION

1.1 Background

Firstly, we need to know about the abnormality of a child.For this question we can say that A birth defect is something visibly abnormal, internally abnormal, or chemically abnormal about your newborn baby's body. Birth defects are problems which are present at birth. In medical terms it’s called congenital anomalies. There are many different types of birth defects, and they can range from mild to severe. Defects can be structural or functional/ developmental.The defect might be caused by genetics, infection, radiation, or drug exposure, or there might be no known reason.To predict the percentage to birth an abnormal child we take some medical data which are collected from many people who have a baby.Then analysis the factor which are responsible to birth abnormal baby and we design a machine learning technique which can predict the result.

Congenital anomalies are also known as birth defects, congenital disorders or congenital malformations. Congenital anomalies can be defined as structural or functional anomalies (for example, metabolic disorders) that occur during intrauterine life and can be identified prenatally, at birth, or sometimes may only be detected later in infancy, such as hearing defects.

In simple terms, congeniality refers to the existence at or before birth.An estimated 295 000 newborns die within 28 days of birth every year, worldwide, due to congenital anomalies. Congenital anomalies can contribute to long-term disability, which may have significant impacts on individuals, families, health-care systems, and societies. The most common, severe congenital anomalies are heart defects, neural tube defects and Down syndrome. Although congenital anomalies may be the result of one or more genetic, infectious, nutritional or environmental factors, it is often difficult to identify the exact causes. Some congenital anomalies can be prevented. Vaccination, adequate intake of folic acid or iodine through fortification of staple foods or supplementation, and adequate antenatal care are just 3 examples of prevention methods.

1.2 Motivation

nowadays most married couples are concerned about giving birth to a normal child.In modern technology there are some implemented supervised machine learning techniques that detect only some specific diseases of a child after birth.These techniques do not define all the factors to give birth to an abnormal child.Which behavior of parents are responsible to born an abnormal child is not defined properly.For that reason we wish to implemented a supervised machine learning which can predict is any parents is able to born a normal child or not.And which behavior is responsible it can be detect.By using this people can easily identify which behavior should change to avoid the birth of an abnormal child.

1.3 Objectives

□ To classify which person is able to give birth to a normal child or not.

□ In this study we can identify which behavior is affected to give birth to a normal baby by using machine learning techniques.

□ The database consists of different behaviors of a person.

□ The aim is to classify people who are able to give birth to a normal baby by only asking some easy questions.

1.4 Organization of Report

The current chapter gives a brief description of the leaf diseases detection technique. Figure 1.1 shows thesis report structure.

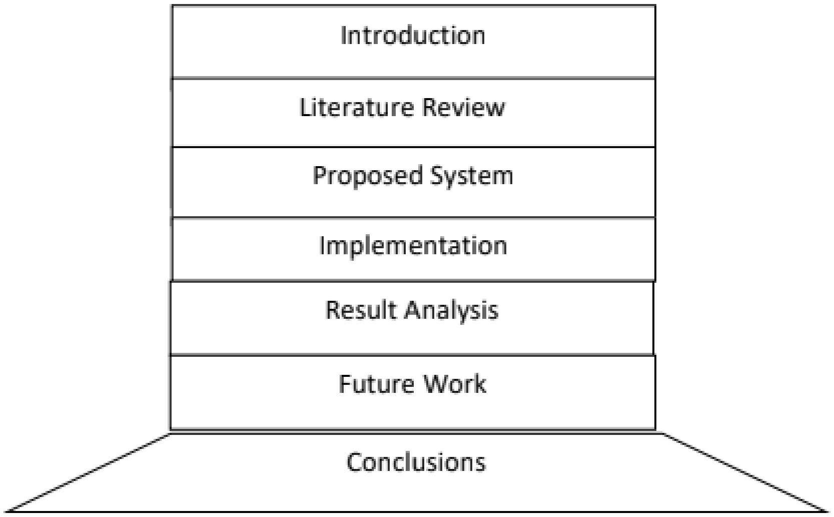


Figure 1.1: Thesis report structure.

1.5 Contributions

This system is designed to classify people who are able to birth a normal baby or not using some machine learning techniques such as Decision tree,KNN,Random forest classifier,XGBoost,GBM,LGBM. This model is able to detect which behavior of any person is responsible for the birth of an abnormal child.A dataset is created by the survey among the people who have one or more babies.This dataset is prepared in matrix format for further use of anybody.

CHAPTER 2

RELATED WORKS

In this chapter, we have discussed some related works that have been done earlier by the researchers.The aim is to identify those people who are not able to birth an abnormal child and cause it to happen.So that they can change their behavior to avoid child abnormality.

- ❏ Use of Machine Learning to Identify Children with Autism and Their Motor Abnormalities.
  Author: Alessandro Crippa
  Limitations: This work does not define all fields of abnormality.

- ❏ Analysis and Detection of Autism Spectrum Disorder Using Machine Learning Techniques.
  Author: Mary Stella. J , Dr. Shashi Kumar
  Limitations: This work does not define which behavior is responsible for the birth of an abnormal child.

- ❏ Prediction and Comparison using AdaBoost and ML Algorithms with Autistic Children.

  Author: SumanRaj,SarfarazMasood

CHAPTER 3

METHODOLOGY

3.1 Data Description

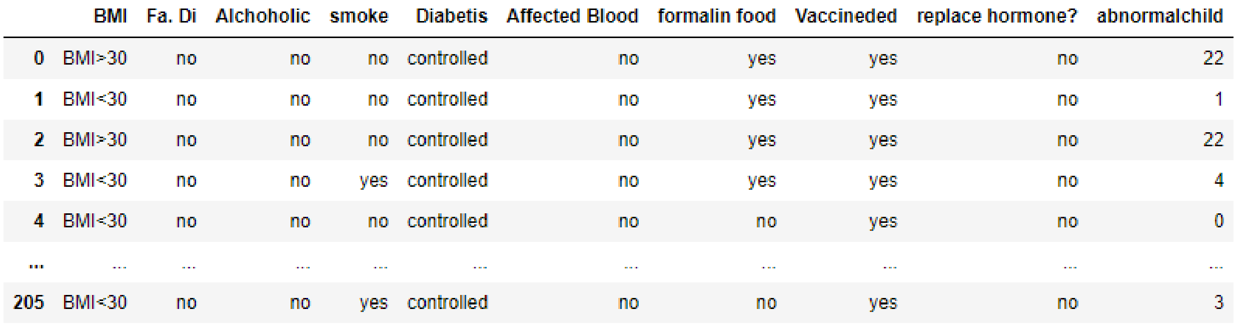
In this section, we explain about the cause or which behavior of parents are responsible for the birth of an abnormal child and also declare that which person is able to birth a normal child using machine learning techniques. First we collect a dataset and after that we train this dataset using machine learning techniques. Dataset is collected from real life.We perform a survey to collect data of some people.A medical officer helps us to gather data from medical institutes. The dataset contains about 337 people’s data.The overall system of proposed method are given below:

3.1.1 Data Collection

For one person’s data we ask 10 questions.which are:

1. How much is your BMI?
2. Is any person in your family abnormal?
3. Do you drink alcohol?
4. Are you a smoker?
5. What is your Diabetics level 's condition?
6. Are you taking blood from any person who is affected by various types of diseases?
7. Are you taking formalin food?
8. Do you have any genetic problems?
9. Have you taken all the vaccines on time?
10. Are you replacing your hormones?

Then we stored data in csv format for further use.



	BMI	Fa. Di	Alchoholic	smoke	Diabetis	Affected Blood	formalin food	Vaccineded	replace hormone?	abnormalchild
0	BMI>30	no	no	no	controlled	no	yes	yes	no	22
1	BMI<30	no	no	no	controlled	no	yes	yes	no	1
2	BMI>30	no	no	no	controlled	no	yes	yes	no	22
3	BMI<30	no	no	yes	controlled	no	yes	yes	no	4
4	BMI<30	no	no	no	controlled	no	no	yes	no	0
...	...	...	...	...	...	...	...	...	...	...
205	BMI<30	no	no	yes	controlled	no	no	yes	no	3

Figure-3.1:Raw Dataset

3.1.2 Data preprocessing

In this step we ensure that there are no null attribute value in our dataset.If any null value is present we remove the hole row corresponding the null data set.For doing this we take help a python function which is **df.isnull().sum()**
This function provides us the number of null values.

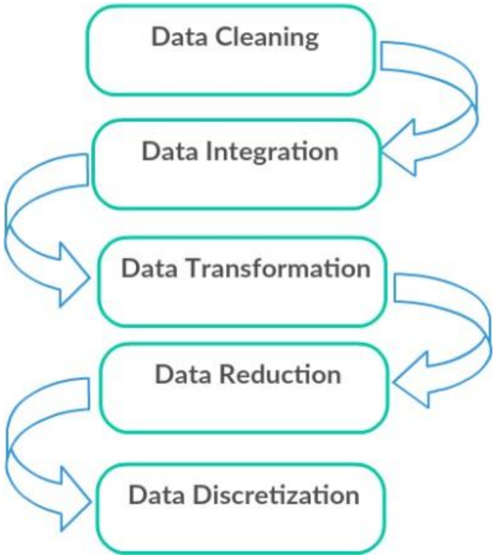
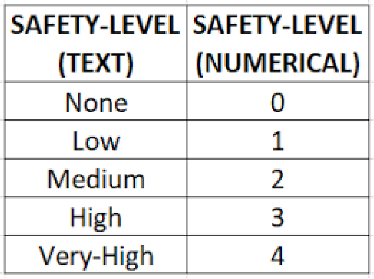


Figure-3.2: Preprocess the data

Label encoding

Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important preprocessing step for the structured dataset in supervised learning.

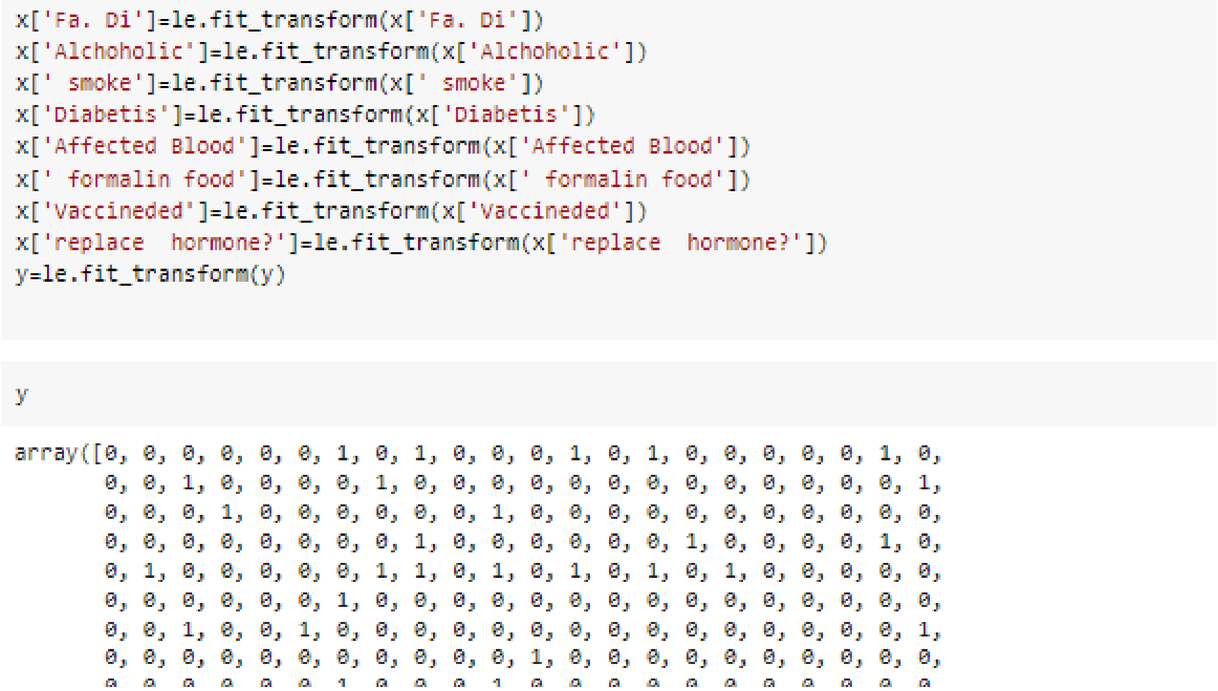
To convert categorical dataset into numeric format we need this step.In this step we convert all independent attributes to numeric format.Such as if BMI>30 then the numeric value is equal to 1 otherwise the value of BMI is 0.If anyone have Family diseases history then it denoted with 0,otherwise 1.In this similar way if drinking alcohol value is yes the numeric value is 1 otherwise 0.If smoking value is yes then it denoted with 1 otherwise 0.If diabetic level is in controlled then denote this with the numeric value is 0,otherwise 1.If anyone is suffering with affected blood it's replace with the value is 1 otherwise 0.If anybody take formalin food everyday then it will convert the numeric value is 1,otherwise 0.If anyone vaccinated properly from childhood then the yes is convert to 1 otherwise 0.Lastly if anybody replace hormone then it convert to numeric value is 1,otherwise 0.Performing this task we take help **LabelEncoder** function.The target variable able is converted to 0 and not-able is converted to 1.



SAFETY-LEVEL (TEXT)	SAFETY-LEVEL (NUMERICAL)
None	0
Low	1
Medium	2
High	3
Very-High	4

Figure-3.3:Example of label encoding

In our dataset:



```
x['Fa. Di']=le.fit_transform(x['Fa. Di'])
x['Alchoholic']=le.fit_transform(x['Alchoholic'])
x[' smoke']=le.fit_transform(x[' smoke'])
x['Diabetis']=le.fit_transform(x['Diabetis'])
x['Affected Blood']=le.fit_transform(x['Affected Blood'])
x[' formalin food']=le.fit_transform(x[' formalin food'])
x['Vaccineded']=le.fit_transform(x['Vaccineded'])
x['replace  hormone?']=le.fit_transform(x['replace  hormone?'])
y=le.fit_transform(y)
```

```
y
```

```
array([0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0,
       0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

Figure-3.4:Our dataset’s encoding

Binning

Data binning is a type of data preprocessing, a mechanism which includes also dealing with missing values, formatting, normalization and standardization. Binning can be applied to convert numeric values to categorical or to sample (quantise) numeric values.

This process is apply for target variable.Where the categorical values are converted as flows:

In this column there are two level

1. Able
2. Not able

At implementation part able is denoted by 0 and not able is denoted by 1.

3.1.3 Data's attribute description

Attribute Name	Cause for
1.BMI(body mass index)	If BMI>30 then it is the cause for abnormal children.
2.Family history	If anybody in a family is abnormal then it can happen the child would be abnormal .
3.Drinking Alcohol	It is the main issue to have an abnormal child.
4.Smoking	It’s also a main issue.
5.Diabetics Level	If the level is uncontrolled then .
6.Taking Blood	If parents get blood from any person who is affected by various types of diseases.
7.Formalin food	If parents take formalin food continuously.
8.Vaccinated	If not then the cause will happen.
9.Genetical	Is any genetic disorder of parents.

3.2 Proposed Method

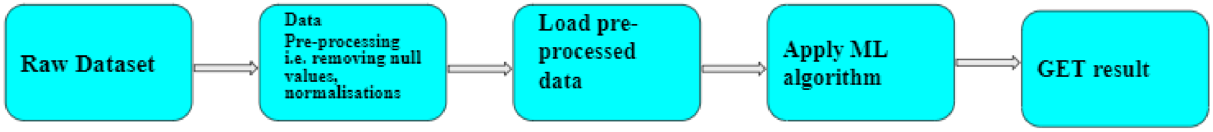


Figure-3.5:Our dataset’s encoding

CHAPTER 4

EXPERIMENT AND RESULTS

To implement a system for this work, a qualitative dataset is required. In classification based work, there are some people’s behavior information.Target factor is the person is able to birth a normal child . In this section, we describe the implementation of the methodology.

4.1 Hardware Specification

- □ Core i3-2.4GHz and Above
- □ 4GB of Random Access Memory and Above
- □ 1 TB Hard Disk

4.2 Software Specification

- ❖ Operating System : WINDOWS 10
- ❖ Language : PYTHON

4.3 Applying Model

4.3.1 Decision Tree

Decision Trees are a type of Supervised Machine Learning (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split. Decision Trees modified An example of a decision tree can be explained using the above binary tree. Let’s say you want to predict whether a person is fit given their information like age, eating habit, and physical activity, etc. The decision nodes here are questions like ‘What’s the age?’, ‘Does he exercise?’, ‘Does he eat a lot of pizzas’? And the leaves, which are outcomes like either ‘fit’, or ‘unfit’. In this case this was a binary classification problem (a yes no type problem). There are two main types of Decision Trees:

Classification trees (Yes/No types) What we’ve seen above is an example of classification tree, where the outcome was a variable like ‘fit’ or ‘unfit’. Here the decision variable is Categorical.

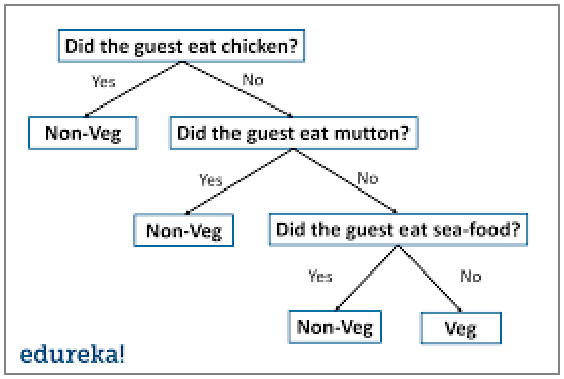

Did the guest eat chicken?
Yes
No
Non-Veg
Did the guest eat mutton?
Yes
No
Non-Veg
Did the guest eat sea-food?
Yes
No
Non-Veg
Veg
edureka!

Figure-4.1 Decision tree example

In Our dataset we apply the Decision Tree Classifier to identify if a person is able to birth a normal child or not.

4.3.2 K-Nearest Neighbors(KNN)

K-Nearest Neighbors (KNN) is one of the simplest algorithms used in Machine Learning for regression and classification problems. KNN algorithms use data and classify new data points based on similarity measures (e.g. distance function). The data is assigned to the class which has the nearest neighbors.

- □ K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- □ K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- □ K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- □ K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
- □ It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

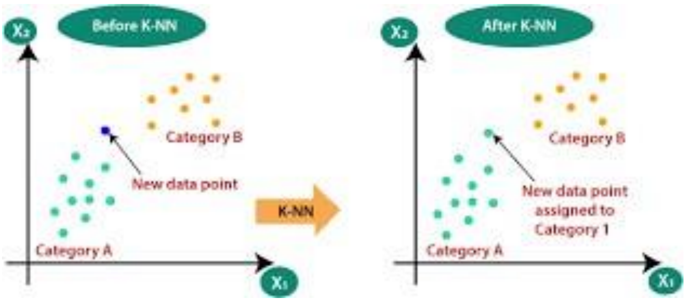


Figure-4.2:KNN Algorithm

How does K-NN work?

The K-NN working can be explained on the basis of the below algorithm:

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

Suppose we have a new data point and we need to put it in the required category. Consider the below image:

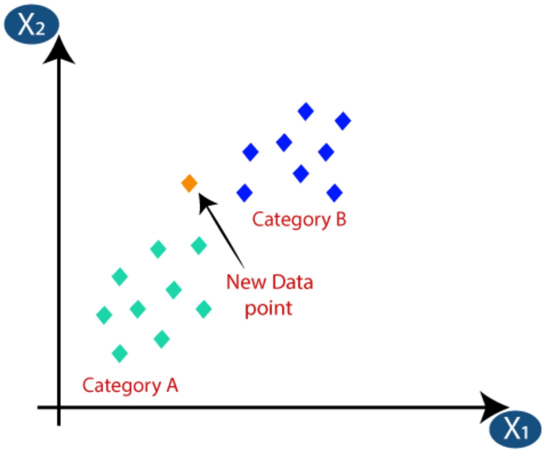


Figure-4.3:Classify new dataset

- Firstly, we will choose the number of neighbors, so we will choose the k=5.
- Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:

Euclidean distance=$d\ =\ \surd[(x2 - x1)2\ +\ (y2 - y1)2]$.

Which distance is minimum from a class then the new data point’s will belong to that class.Then we declare that the new dataset is in class A or class B.

4.3.3 Random Forest Classifier(RFC)

Random forest is a supervised learning algorithm. The "forest" it builds is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.

How Random Forest Works

Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result. Put simply: random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems. Let's look at random forest in classification, since classification is sometimes considered the building block of machine learning. Below you can see how a random forest would look like with two trees:

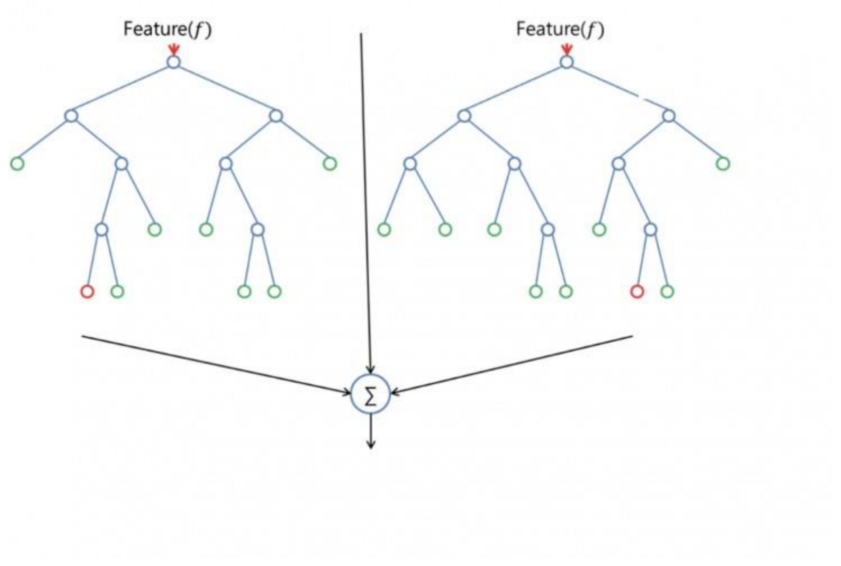


Figure-4.4:Random Forest Classifier

Random forest has nearly the same hyperparameters as a decision tree or a bagging classifier. Fortunately, there's no need to combine a decision tree with a bagging classifier because you can easily use the classifier-class of random forest. With random forest, you can also deal with regression tasks by using the algorithm's regressor. Random forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model. Therefore, in a random forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. You can even make trees more random by additionally using random thresholds for each feature rather than searching for the best possible thresholds (like a normal decision tree does).

Difference between Decision Trees and Random Forests

While a random forest is a collection of decision trees, there are some differences. If you input a training dataset with features and labels into a decision tree, it will formulate some set of rules, which will be used to make the predictions. For example, to predict whether a person will click on an online advertisement, you might collect the ads the person clicked on in the past and some features that describe his/her decision. If you put the features and labels into a decision tree, it will generate some rules that help predict whether the advertisement will be clicked or not. In comparison, the random forest algorithm randomly selects observations and features to build several decision trees and then averages the results. Another difference is "deep" decision trees might suffer from overfitting. Most of the time, random forest prevents this by creating random subsets of the features and building smaller trees using those subsets. Afterwards, it combines the subtrees. It's important to note this doesn’t work every time and it also makes the computation slower, depending on how many trees the random forest builds.

4.3.4 XGBoost

XGBoost is an algorithm that has recently been dominating applied machine learning and Kaggle competitions for structured or tabular data.

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.

In this post you will discover XGBoost and get a gentle introduction to what it is, where it came from and how you can learn more.

After reading this post you will know:

- What XGBoost is and the goals of the project.

- Why XGBoost must be a part of your machine learning toolkit.
- Where you can learn more to start using XGBoost on your next machine learning project.

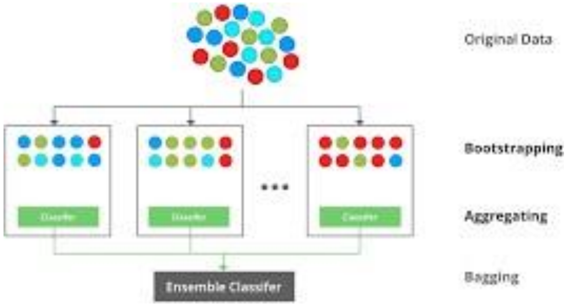


Figure-4.5:XGBoost

How does XGBoost work:

XGBoost is a popular and efficient open-source implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm, which attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models. When using gradient boosting for regression, the weak learners are regression trees, and each regression tree maps an input data point to one of its leafs that contains a continuous score. XGBoost minimizes a regularized (L1 and L2) objective function that combines a convex loss function (based on the difference between the predicted and target outputs) and a penalty term for model complexity (in other words, the regression tree functions). The training proceeds iteratively, adding new trees that predict the residuals or errors of prior trees that are then combined with previous trees to make the final prediction. It's called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models. Below is a brief illustration on how gradient tree boosting works.

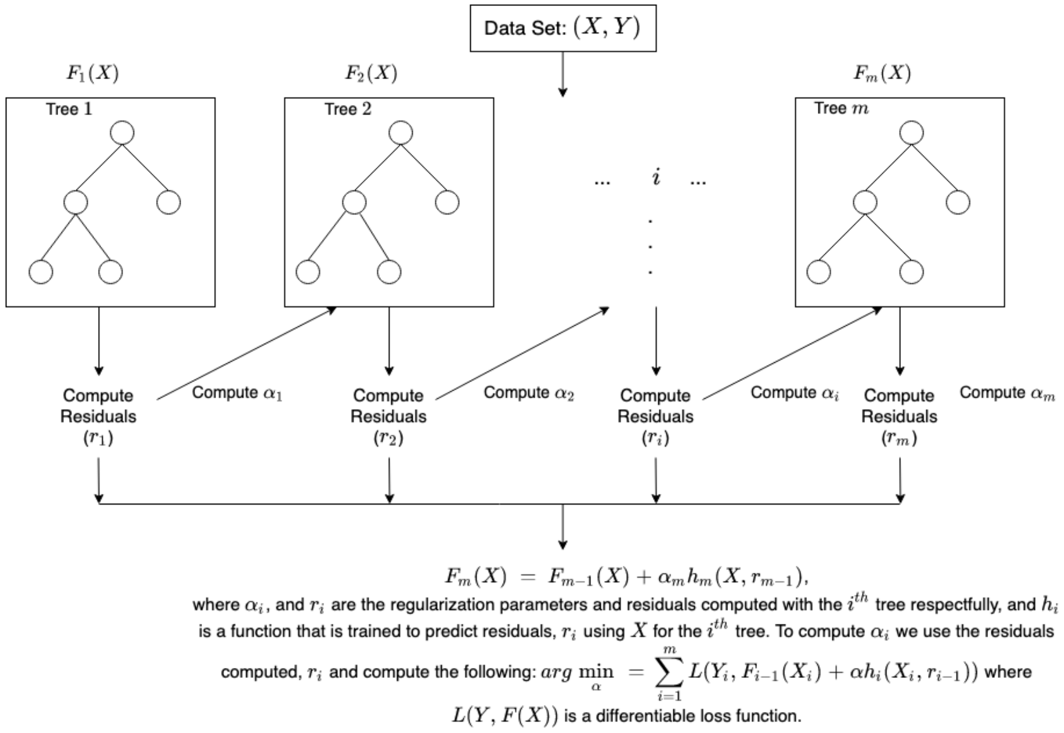

Data Set: (X, Y)
$F_1(X)$
$F_2(X)$
$F_m(X)$
Tree 1
Tree 2
... i ...
Tree m
Compute Residuals (r_1)
Compute α_1
Compute Residuals (r_2)
Compute α_2
Compute Residuals (r_i)
Compute α_i
Compute Residuals (r_m)
Compute α_m
$F_m(X) = F_{m-1}(X) + \alpha_m h_m(X, r_{m-1})$,
where α_i, and r_i are the regularization parameters and residuals computed with the i^{th} tree respectfully, and h_i is a function that is trained to predict residuals, r_i using X for the i^{th} tree. To compute α_i we use the residuals computed, r_i and compute the following: $arg \min_{\alpha} = \sum_{i=1}^{m} L(Y_i, F_{i-1}(X_i) + \alpha h_i(X_i, r_{i-1}))$ where $L(Y, F(X))$ is a differentiable loss function.

Figure-4.6:XGBoost Working Process

4.3.5 Gradient Boosting Machine(GBM)

Gradient boosting is a type of machine learning boosting. It relies on the intuition that the best possible next model, when combined with previous models, minimizes the overall prediction error. The key idea is to set the target outcomes for this next model in order to minimize the error. How are the targets calculated? The target outcome for each case in the data depends on how much changing that case's prediction impacts the overall prediction error: If a small change in the prediction for a case causes a large drop in error, then the next target outcome of the case is a high value. Predictions from the new model that are close to its targets will reduce the error. If a small change in the prediction for a case causes no change in error, then the next target outcome of the case is zero. Changing this prediction does not decrease the error. The name gradient boosting arises because target outcomes for each case are set based on the gradient of the error with respect to the prediction. Each new model takes a step in the direction that minimizes prediction error, in the space of possible predictions for each training case.

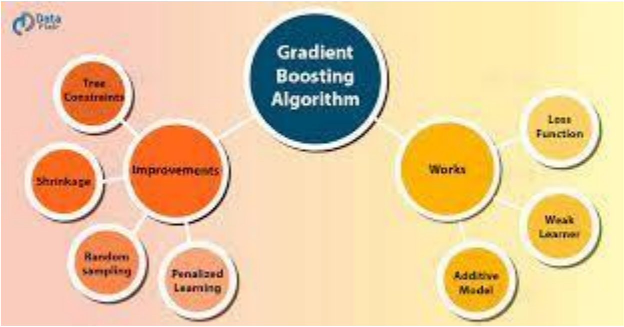


Figure-4.7:Gradient Boosting Machine

How GBM Works:

Here is the trick – the nodes in every decision tree take a different subset of features for selecting the best split. This means that the individual trees aren't all the same and hence they are able to capture different signals from the data. Additionally, each new tree takes into account the errors or mistakes made by the previous trees. So, every successive decision tree is built on the errors of the previous trees. This is how the trees in a gradient boosting machine algorithm are built sequentially.

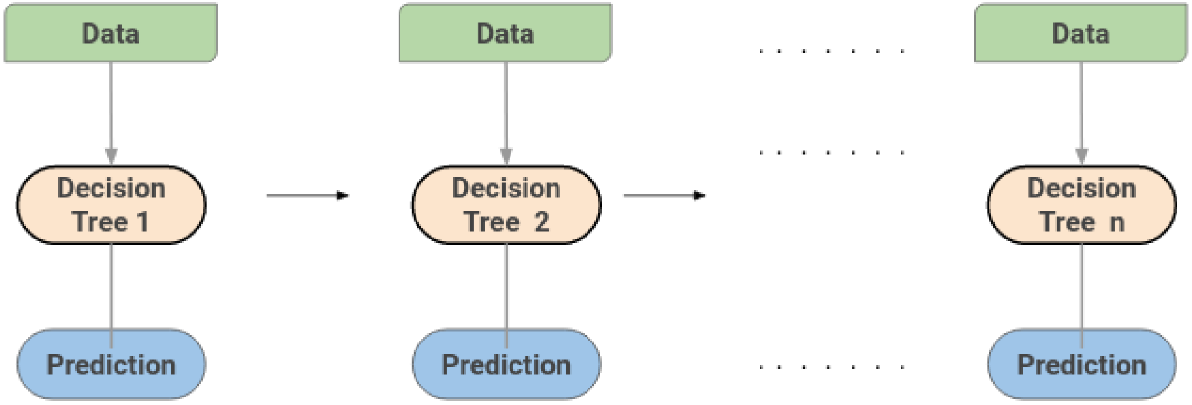


Figure-4.8:Working Process of GBM

4.3.6 Light Gradient Boosting Machine(LGBM)

The LightGBM boosting algorithm is becoming more popular by the day due to its speed and efficiency. LightGBM is able to handle huge amounts of data with ease. But keep in mind that this algorithm does not perform well with a small number of data points. Let’s take a moment to understand why that’s the case. The trees in LightGBM have a leaf-wise growth, rather than a level-wise growth. After the first split, the next split is done only on the leaf node that has a higher delta loss.

Consider the example I’ve illustrated in the below image:

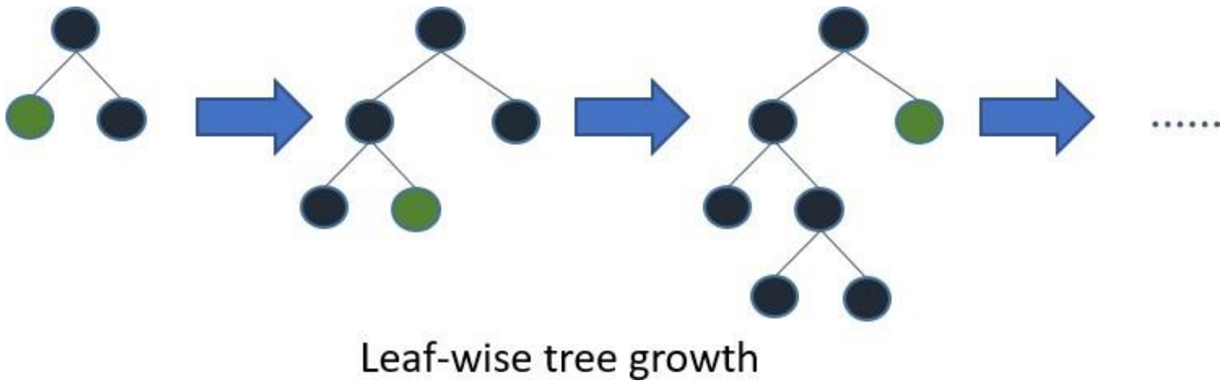


Figure-4.9:LGBM

After the first split, the left node has a higher loss and is selected for the next split. Now, we have three leaf nodes, and the middle leaf node has the highest loss. The leaf-wise split of the LightGBM algorithm enables it to work with large datasets. In order to speed up the training process, LightGBM uses a histogram-based method for selecting the best split. For any continuous variable, instead of using the individual values, these are divided into bins or buckets. This makes the training process faster and lowers memory usage.

4.4 Result

For our experiments, we used some machine learning techniques to evaluate the proposed classification models.Now we are going to show all resulting matrices.

4.4.1 Result Measures

In our experiments, we use a confusion matrix and ROC curve to show our model’s performance based on target class and output class.

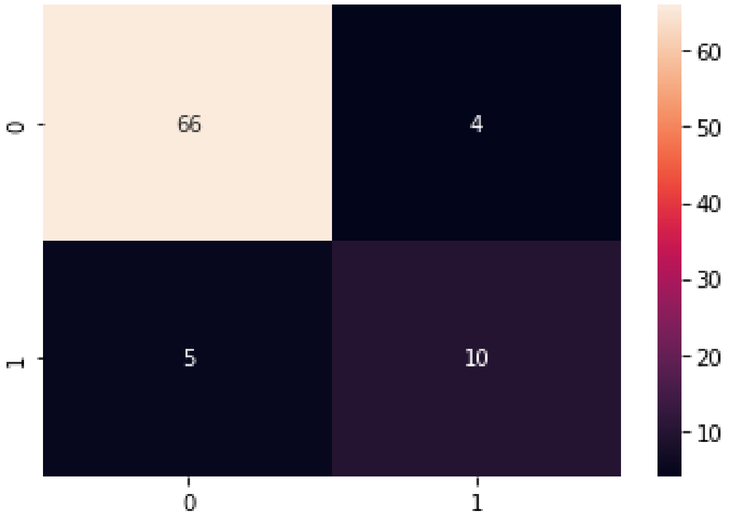


Figure-4.10: Confusion Matrix for Decision Tree

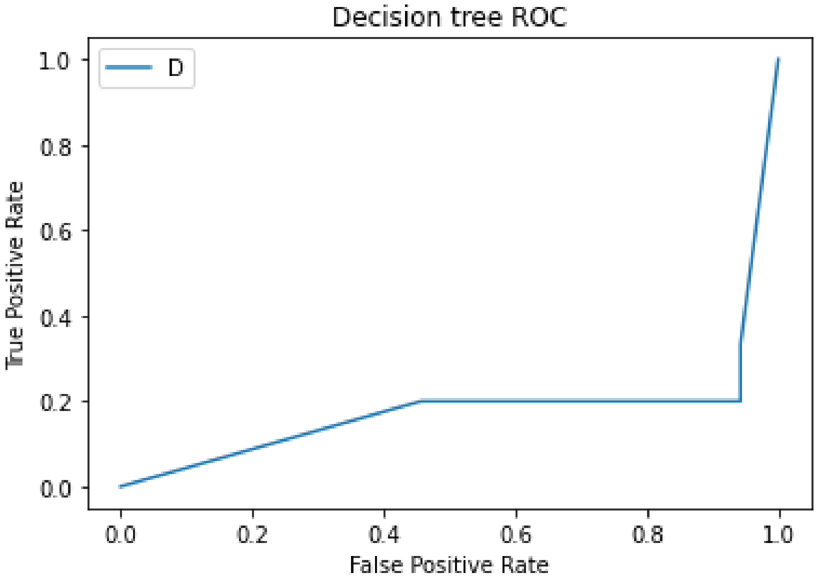


Figure-4.11:ROC for Decision Tree

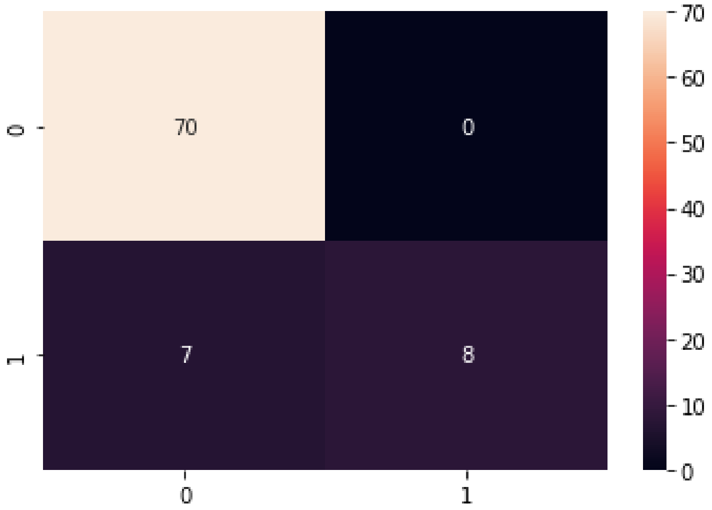


Figure-4.12:Confusion Matrix for KNN

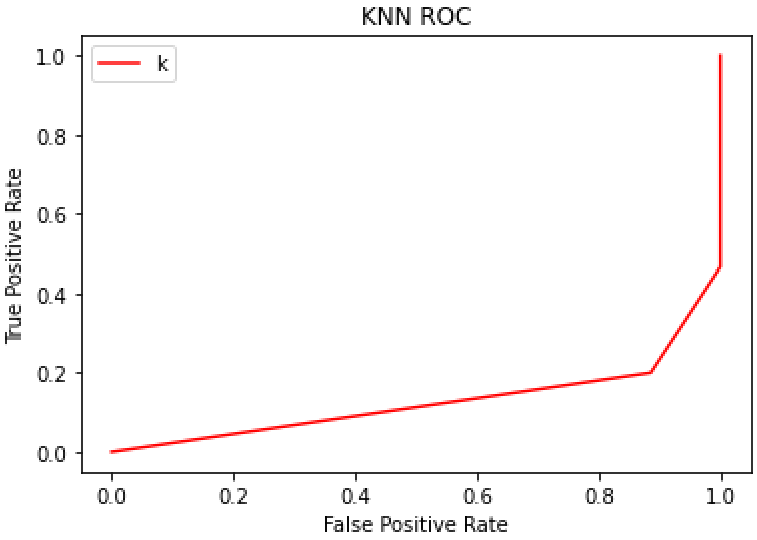


Figure-4.13: ROC for KNN

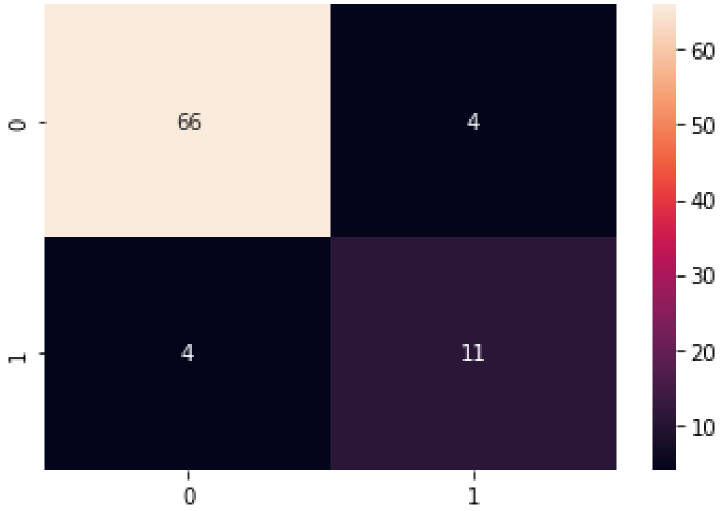


Figure-4.14:Confusion Matrix for RFC

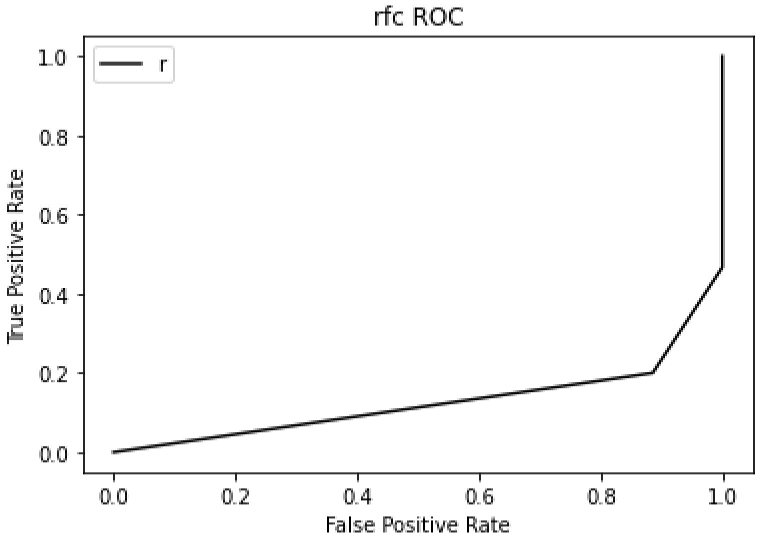


Figure-4.15: ROC for RFC

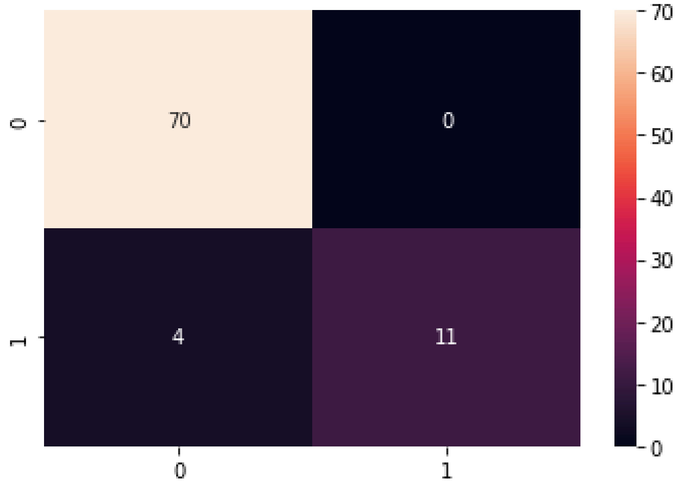


Figure-4.16:Confusion Matrix for XGBoost

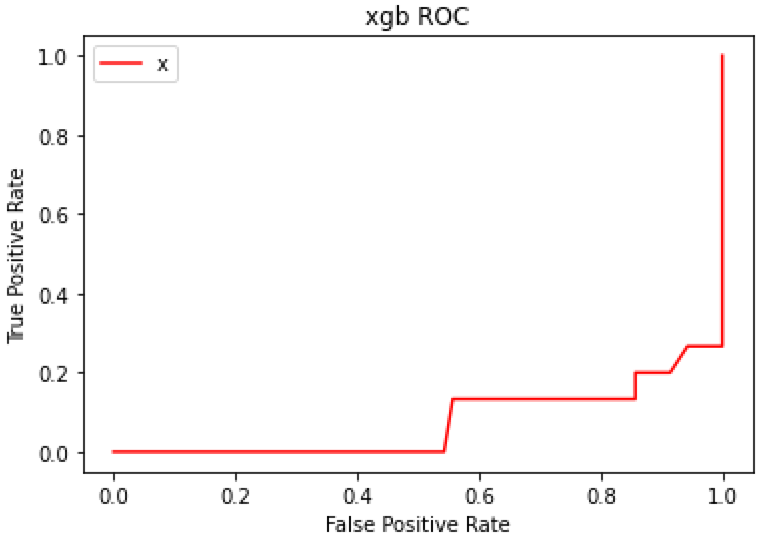


Figure-4.17: ROC for XGBoost

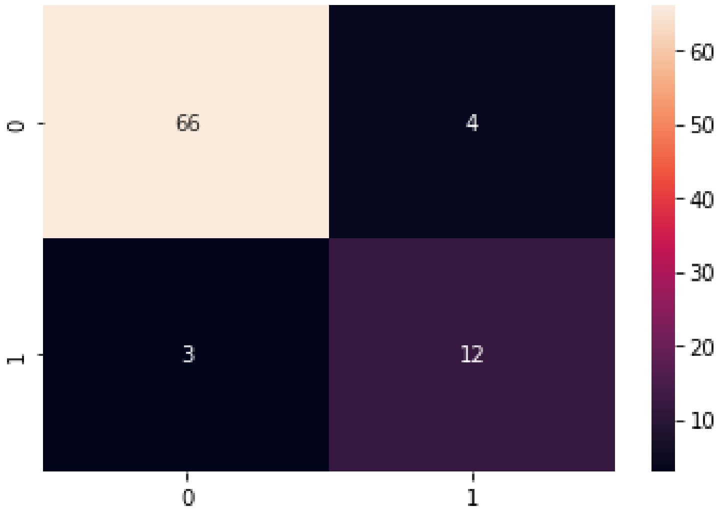


Figure-4.18:Confusion Matrix for GBM

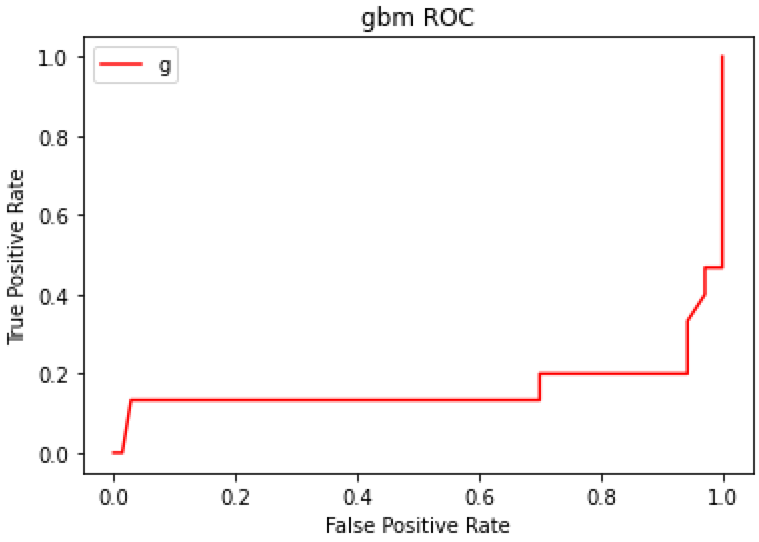


Figure-4.19: ROC for GBM

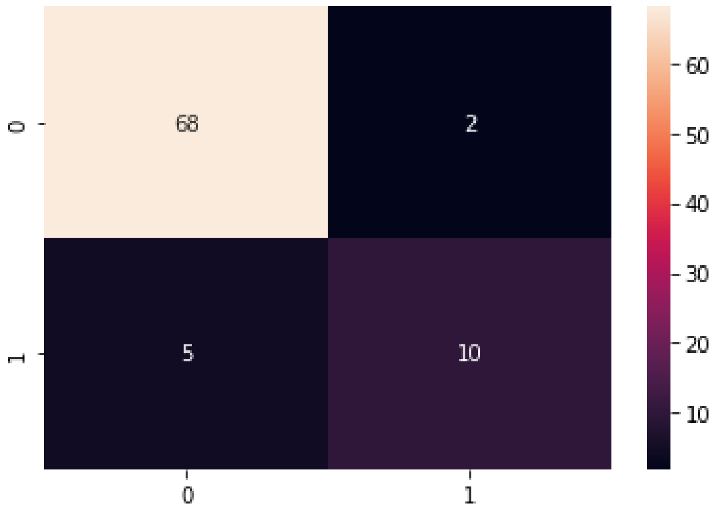


Figure-4.20:Confusion Matrix for LGBM

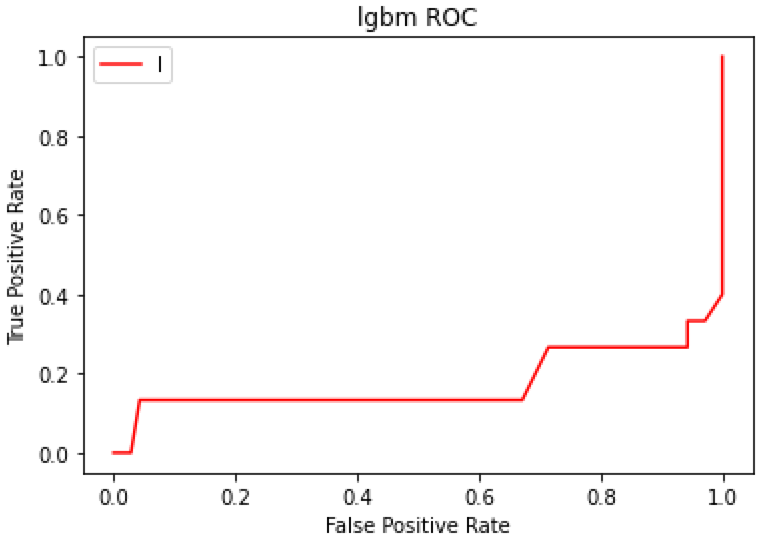


Figure-4.19: ROC for LGBM

CHAPTER 5

DISCUSSION

In any type of research, experimental results are very essential. All the researchers want to reach the highest accuracy level according to their work. This accuracy level may be different by using various algorithms and methodology. The researchers select the algorithm and methodology which give the best accuracy level for the corresponding research.

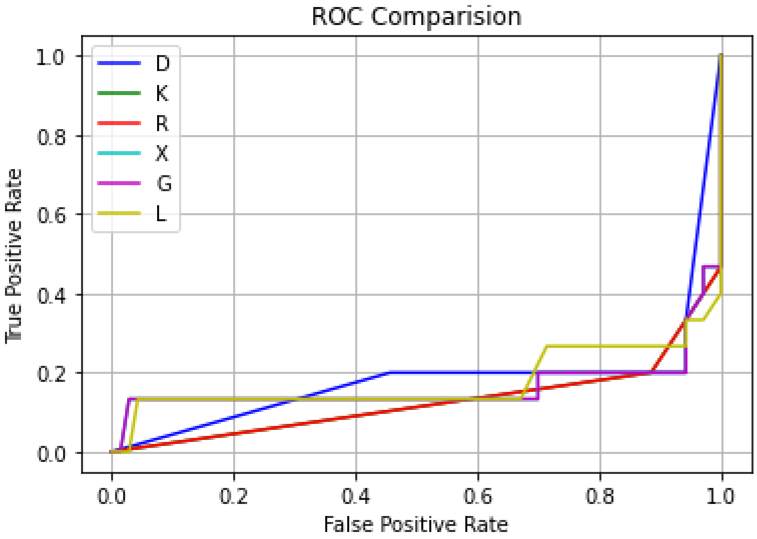


Figure-5.1: ROC Comparison

Here,
D=Decision Tree
K=KNN
R=RFC
X=XGboost
G=GBM
L=LGBM

5.1 Final Result:

Model Name	Accuracy	Precision	Recall	F1-Score	Support
Decision Tree	89%	0->0.93 1->0.71	0->0.94 1->0.67	0->0.94 1->0.69	70 15
KNN	92%	0->0.91 1->1.00	0->1.00 1->0.53	0->0.95 1->0.70	70 15
RFC	88%	0->0.94 1->0.65	0->0.91 1->0.73	0->0.93 1->0.69	70 15
XGBoost	95%	0->0.95 1->1.00	0->1.00 1->0.73	0->0.97 1->0.85	70 15
GBM	92%	0->0.96 1->0.75	0->0.94 1->0.80	0->0.95 1->0.77	70 15
LGBM	92%	0->0.93 1->0.83	0->0.97 1->0.67	0->0.95 1->0.74	70 15

CHAPTER 6

Limitation and Future work

All the researches have some lacking. No research cannot be out of error. There are some limitations. Artificial systems help us to go to the closest level of human thinking but machines cannot be human. So there may be some errors. All the factors are not enough. There are some factors which affect the result, but for the complexity of the research we avoid these factors. Overall, our aim is to reach the best accuracy level and in future determine the lacking of this research. Our future goal is to contribute to our ever developing society by providing an advanced model of skin recognition classification system.

In the future we want to classify the result in three categories. Low chance,Medium chance,High chance to birth an abnormal child.The proposed model cannot classify properly which person is able to birth an abnormal child or not.Accuracy is not satisfied.

CHAPTER 7

CONCLUSION

Detecting the responsibilities for the birth of an abnormal child,we are using various machine learning and deep learning techniques. Various performance evaluation metrics will be used to analyze the performance of the models implemented for the detection of abnormal Child. We will try our best to get the best result in this field insAllah.

Our motor measure might have potential clinical application in such cases, thus providing useful information for clinicians to support a diagnostic decision. A point of relevance of our work, in fact, is that we decided to study the predictive value of a simple reach,grasp, and drop task, because the motor system can be more easily evaluated .