**PROJECT REPORT**

ON

**THE HIGHLY RELIABLE AND ACCURATE**

**ORGAN DONAR PREDICTION APPLICATION : AI & ML BASED**

BY

BRIJ KISHOR (1901660100022)

SAMAN KHAN (1901660100052)

DINISHA JAISWAL (1901660100024)

Submitted to the department of Computer Science

In the partial fulfilment of the requirements

for the degree of

Bachelor of technology



DR. AMBEDKAR INSTITUTE OF TECHNOLOGY FOR HANDICAPPED , KANPUR

DR . A.P.J . ABDUL KALAM TECHNICAL UNIVERSITY , LUCKNOW

May , 2023

**DATE:……**

**CERTIFICATE**

*This is to certify that the Thesis entitled “****THE HIGHLY RELIABLE AND ACCURATE ORGAN DONOR PREDICTION APPLICATION: AI AND ML BASED****” is being submitted by* **BRIJ KISHOR (1901660100022), SAMAN KHAN ( 1901660100052), DINISHA JASWAL (1901660100024) ,** *an undergraduate student in the Department of Computer Science and Engineering, Dr Ambedkar institute of technology for handicapped , kanpur, India, for the award of Bachelor of Technology in Computer Science and Engineering is an original research work carried by them under my supervision and guidance. The Thesis has fulfilled all the requirements as per the regulations of Dr Ambedkar institute of technology for handicapped , Kanpur and in my opinion, has reached the standards needed for submission. The work, techniques and the results presented have not been submitted to any other University or Institute for the award of any other degree.*

*PROJECT SUPERVISOR*

*DEPARTMENT OF COMPUTER SCIENCE*

*Dr. AITH,KANPUR*

***HEAD OF DEPARTMENT***

***DEPARTMENT OF COMPUTER SCIENCE***

***Dr. AITH,KANPUR***

Declaration

*We hereby declare that the work being presented in this research project entitled,* ***ORGAN DONAR PREDICTION****submitted to Dr A.I.T.H , Kanpur in fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Engineering during the period from JAN 2023 to MAY 2023 under the supervision of* ***Dr. SHRI NATH***  *Department of Computer Science and Engineering, Dr Ambedkar institute of technology for handicapped , Kanpur*

**BRIJ KISHOR (1901660100022)**

**SAMAN KHAN ( 1901660100052)**

**DINISHA JASWAL (1901660100024)**

***Department: Computer Science & Engineering***

*Institute : Dr Ambedkar institute of technology for handicapped ,Kanpur.*

*This is to certify that the above statement made by the candidate is correct to the best of my knowledge.*

*Signed:*

*Date:*

*Acknowledgment*

We would like to express our sincere thanks and gratitude to our Super-visor **Srinath** who gave us the golden opportunity to do this wonderful project and guided us immensely through the course of the project.

We came to know about so many new things, all thanks to him. Last but not the least, We would like to thank all those who had helped directly or indirectly in this project*.*

I would like to thank my dear group members who have made their best efforts to make this project successful.

At last, I would like to extend my heartfelt thanks to my parents because without their help this project would not have been successful.

I’m extremely grateful to all the people who have helped me in completing this project. In the end, I would thank God for being able to complete this project with success.

**ABSTRACT**

The complexity of transplant medicine pushes the boundaries of innate, human reasoning. From networks of immune modulators to dynamic pharmacokinetics to variable postoperative graft survival to equitable allocation of scarce organs, machine learning promises to inform clinical decision making by deciphering prodigious amounts of available data .

A general introduction to different machine learning techniques, describing their strengths, limitations, and barriers to clinical implementation. We summarize emerging evidence that recent advances that allow machine learning algorithms to predict acute post-surgical and long-term outcomes, classify biopsy and radiographic data, augment pharmacologic decision making, and accurately represent the complexity of host immune response.

Organ donation is not meeting demand, and yet 30–60% of potential donors are potentially not identified. Current systems rely on manual identification and referral to an Organ Donation Organization (ODO). We hypothesized that developing an automated screening system based on machine learning could reduce the proportion of missed potentially eligible organ donors. Using routine clinical data and laboratory time-series, we retrospectively developed and tested a neural network model to automatically identify potential organ donors. We first trained a convolutive autoencoder that learned from the longitudinal changes of over 100 types of laboratory results.

***BACKGROUND***

Optimizing organ yield (number of organs transplanted per donor) is a potentially modifiable way to increase the number of organs available for transplant. Models to predict the expected deceased [donor](https://www.sciencedirect.com/topics/medicine-and-dentistry/donor) organ yield have been developed based on ordinary [least squares](https://www.sciencedirect.com/topics/medicine-and-dentistry/least-square-analysis) regression and [logistic regression](https://www.sciencedirect.com/topics/medicine-and-dentistry/logistic-regression-analysis). However, alternative modeling methodologies incorporating machine learning may have superior performance compared with conventional approaches.

We evaluated the predictive accuracy of 14 machine learning models for predicting overall organ yield in a cross-validation procedure. The models were parameterized using data from the Organ Procurement and Transplantation Network database from 2000 to 2018. The inclusion criteria for the study were adult deceased donors between 18 and 84 years of age that had at least 1 organ procured for transplantation.

KEY ASPECTS TO FACILITATING

MACHINE LEARNING SOLUTIONS

**Combination of multi-disciplinary skills**

The production of official statistics has always and continues to combine the knowledge and expertise from many disciplines. This is still the case, and even more so, with the proliferation of data sources (big, medium or small), users needing or wanting to exploit them and technologies enabling their use. While many of these skills are present in data science (a relatively new discipline), the breadth and depth of the skills needed in each of the disciplines cannot be found in a single or small number of individuals. Bringing together the required skills is one of the main challenges facing statistical organisations. This can be broken down into four sub-challenges: identification, acquisition, development and organisation. Some of them were addressed by the project (see: Integration). In doing so, many concrete actions by NSOs to facilitate and expand the use of ML were discovered.

**Computing infrastructure**

From the beginning, the ML project decided to focus on demonstrating added value, quality and integration. The project discussed this issue at times, but just barely scratched the surface. Going forward, this aspect should be considered among the ML 2021 projects, but this should be preceded by a scan to find and connect with any working groups or other developments already addressing this aspect.

**Research and development**

The first key aspect to having ML solutions accepted mentioned above is to alignment them with business needs. Discussions within the project subgroup on integration did not come to a full consensus on this aspect. Some emphasized the importance of starting with a business need, moving to R&D, producing a prototype and then bringing in the other areas like lT. Others emphasized the importance of building ML experience first, through R&D, which in turn allows one to identify suitable business problems which might be solved by machine learning.

TABLE OF CONTENTS

*CERTIFICATE*

*DECLARATION*

*ACKNOWLEDGEMENTS*

*ABSTRACT*

*BACKGROUND*

*KEY ASPECTS TO FACILITATING ML SOLUTIONS*

*1. INTRODUCTION*

*2 . PROBLEMS AND ISSUES IN SUPERVISED LEARNING*

*3. Overview & Problem Statement-*

*4. Implementation –*

*5. USE*

*6. APPLICATION OF MACHINE LEARNING*

*7. USE OF MACHINE LEARNING ALGORITHM*

*8 . TRAINING OF ALGORITHM*

*9 . RESULT*

*10. CODE*

*11 . CONCLUSION AND FUTURE SCOPE –*

*12 . REFERENCES*

***1 .INTRODUCTION***

Machine learning is a branch of artificial intelligence in which a computer algorithm learns from examples to generate reproducible predictions and classifications on previously unseen data . Machine learning can be supervised or unsupervised: the former referring to manually mapping an observation’s characteristics to a known outcome; the latter referring to discovery of innate patterns using unlabeled data .

An example of supervised learning would include using known clinical risk factors to predict survival. In contrast, an unsupervised model could be fed thousands of histopathology slides and learn to group them according to similarities in pixel patterns.

A further subset of machine learning includes neural networks. These networks rely on layers of calculation-performing nodes that differentially weight inputs before passing them along to other nodes, eventually producing a narrow range of outputs.

Deep neural networks consist of dozens to hundreds of layers, often with specialized functions interspersed within the layers, enabling the network to better represent complex patterns in unstructured data.

Organ transplantation is critically dependent on potential organ donor identification and conversion to actual donors. The former is a major challenge that relies heavily on the training of medical teams, an inefficient approach given the rarity of deceased organ donation, especially in small centers[3](https://www.nature.com/articles/s41598-023-35270-w#ref-CR3). Multiple retrospective cohort studies suggested that between 30 and 60% of potential organ donors are either not identified or not referred to an Organ Donation Organization (ODO)[4](https://www.nature.com/articles/s41598-023-35270-w#ref-CR4),[5](https://www.nature.com/articles/s41598-023-35270-w#ref-CR5),[6](https://www.nature.com/articles/s41598-023-35270-w#ref-CR6),[7](https://www.nature.com/articles/s41598-023-35270-w#ref-CR7). More efficient identification of potential organ donors could lead to an increase in the total number of referrals to an ODO and, therefore, to a potential increase in the number of organ donors.

Several excellent reviews have been published on machine learning in transplant medicine ([3](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8481939/#B3)–[6](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8481939/#B6)). This review builds on prior knowledge by incorporating additional applications of machine learning in predicting acute post-surgical and long-term outcomes, caring for critically ill patients, classifying biopsy and radiographic data, augmenting pharmacologic decision making, and elucidating the complexity of host immune response

The primary objective of this study was to develop a predictive model for the identification of potential organ donors among patients admitted to an intensive care unit (ICU) using routinely collected clinical data. Our secondary objectives were: (1) to compare the discriminative property of a NN model compared to a logistic regression model used as a baseline, (2) to evaluate our models in prespecified subgroups of organ donor, and (3) to evaluate the model in a prospective simulation performed over a 48-h time.

**PROBLEMS AND ISSUES WITH**

**SUPERVISED LEARNING**

Before we get started, we must know about how to pick a good machine learning

algorithm for the given dataset. To intelligently pick an algorithm to use for a supervised learning

task, we must consider the following factors [4]:

Before we get started, we must know about how to pick a good machine learning

algorithm for the given dataset. To intelligently pick an algorithm to use for a supervised learning

task, we must consider the following factors [4]:

Before we get started, we must know about how to pick a good machine learning

algorithm for the given dataset. To intelligently pick an algorithm to use for a supervised learning

task, we must consider the following factors [4]:

Before we get started, we must know about how to pick a good machine learning

algorithm for the given dataset. To intelligently pick an algorithm to use for a supervised learning

task, we must consider the following factors [4]:

Before we get started, we must know about how to pick a good machine learning

algorithm for the given dataset. To intelligently pick an algorithm to use for a supervised learning

task, we must consider the following factors [4]:

**1. Heterogeneity of Data:**

Many algorithms like neural networks and support vector machines like their

feature vectors to be homogeneous numeric and normalized. The algorithms that

employ distance metrics are very sensitive to this, and hence if the data is

heterogeneous, these methods should be the afterthought. Decision Trees can handle

heterogeneous data very easily.

**2. Redundancy of Data:**

If the data contains redundant information, i.e. contain highly correlated values,

then it’s useless to use distance based methods because of numerical instability. In

this case, some sort of Regularization can be employed to the data to prevent this

situation.

**3. Dependent Features:**

If there is some dependence between the feature vectors, then algorithms that

monitor complex interactions like Neural Networks and Decision Trees fare better

than other algorithms.

**4. Bias-Variance Tradeoff:**

A learning algorithm is biased for a particular input x if, when trained on each of

these data sets, it is systematically incorrect when predicting the correct output for x,

whereas a learning algorithm has high variance for a particular input x if it predicts

different output values when trained on different training sets. The prediction error of

a learned classifier can be related to the sum of bias and variance of the learning

algorithm, and neither can be high as they will make the prediction error to be high. A

key feature of machine learning algorithms is that they are able to tune the balance

between bias and variance automatically, or by manual tuning using bias parameters,

and using such algorithms will resolve this situation.

**5. Curse of Dimensionality:**

If the problem has an input space that has a large number of dimensions, and the

problem only depends on a subspace of the input space with small dimensions, the

machine learning algorithm can be confused by the huge number of dimensions and

hence the variance of the algorithm can be high. In practice, if the data scientist can

manually remove irrelevant features from the input data, this is likely to improve the

accuracy of the learned function. In addition, there are many algorithms for feature

selection that seek to identify the relevant features and discard the irrelevant ones, for

instance Principle Component Analysis for unsupervised learning. This reduces the

dimensionality.

**6. Overfitting:**

The programmer should know that there is a possibility that the output values may

constitute of an inherent noise which is the result of human or sensor errors. In this case, the algorithm must not attempt to infer the function that exactly matches all the

data. Being too careful in fitting the data can cause overfitting, after which the model

will answer perfectly for all training examples but will have a very high error for

unseen samples. A practical way of preventing this is stopping the learning process

prematurely, as well as applying filters to the data in the pre-learning phase to remove noises.

**2 . Overview & Problem Statement-**

* The basic idea of this topic is to predict the organ donors based on the data that we have in the dataset of the kaggle.
* The motivation for this project is clearly related to health situation in the present world. Since we can see that in the current situation of the world, its kind of very difficult to find the correct match for the organ donation and in case of emergency it really become a hectic to get a correct organ donor and initiate the process of organ replacement.
* So here keeping in mind the above problem, we basically developed a machine learning model to predict the donors from the datas present in the our datasets***.***

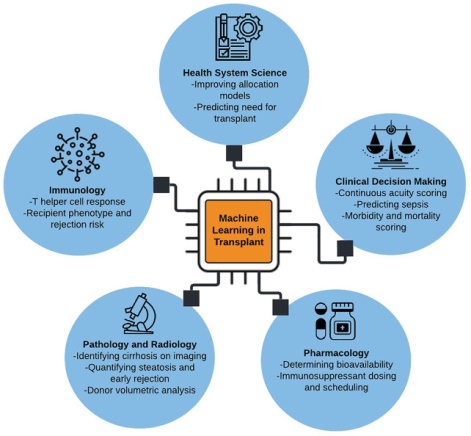
**3 . Implementation**

* In the implementation part, first we got the data set from Kaggle site. After that, we made our machine learning model in Jupyter Notebook.
* We preferred Jupyter Notebook, because its helps in running and compiling python codes in cluster, like we can run codes in clusters in Jupyter Notebook cells.
* First we splitted our implementation part in two directory, first one is Dataset and second one is Model part. In Dataset part we stored the dataset in .csv format and in the model part we proceeded with the implementation part of our code.

**4 .USE**

|  |  |
| --- | --- |
| Machine Learning | A sub-field of artificial intelligence in which a computer system performs a task without explicit instructions |
| Deep Neural Networks | A sub-field of machine learning in which computer systems learn and represent data by adjusting weighted associations among input features across a layered hierarchy of neurons or neural network |
| Supervised Learning | Algorithms learn from training sets of labeled data and then classifies new, previously unseen data |
| Unsupervised Learning | Algorithms learn from unlabeled data and generate their own classification schemes, which can discover hidden patterns |

**5 . APPLICATION OF MACHINE LEARNING**



**6 .USE OF MACHINE LEARNING ALGORITHM**

LOGISTIC REGRESSION –

Logistic regression is **a statistical analysis method to predict a binary outcome, such as yes or no, based on prior observations of a data set**. A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables.

LINEAR REGRESSION –

Linear Regression is **a machine learning algorithm based on supervised learning**. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting



RIDGE RFEGRESSION –

Ridge regression is **a model tuning method that is used to analyse any data that suffers from multicollinearity**. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values being far away from the actual values.

K-NEAREST –NEIGHBOURS –

The k-nearest neighbors algorithm, also known as KNN or k-NN, is **a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point**.

LASSO REGRESSION –

Lasso regression is like linear regression, but it uses a technique **"shrinkage"**where the coefficients of determination are shrunk towards **zero**.

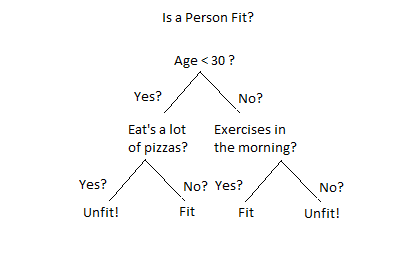
Linear regression gives you regression coefficients as observed in the dataset. The lasso regression allows you to shrink or regularize these coefficients to avoid over fitting and make them work better on different datasets.

RANDOM FOREST CLASSIFIER –

***Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."*** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

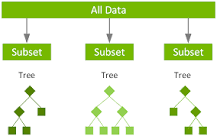
DECISION TREE CLASSIFIER –

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 .



XG BOOST CLASSIFIER –

XG Boost, which stands for Extreme Gradient Boosting, is **a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library**. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems.

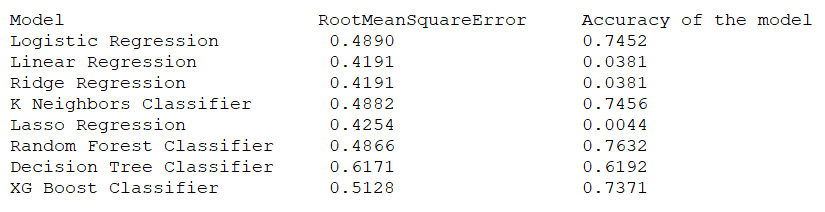


***6 . 1 - TRAINING OF ALGORITHM***

* + - Step 1: First using the sklearn library we import that model into out code.
    - Step 2: Then we fit the training part of the dataset into the imported model using the fit function present.
    - Step 3: After fitting we predict the output for the 20% testing data.
    - Step 4: At the final step, we get the accuracy of that model using the score function present in that library. That accuracy basically shows how well the model trained and tested the data according to its predefined algorithm.

**7 .RESULT**

* After training the dataset and testing, we got the following results, representing the accuracy score and root mean square value for each of the algorithm.



Therefore using the above data that has been showcased in the table shown above it could be concluded that the rand forest classifier is one of the best model to be used for the prediction of the correct and the

10-CODE

### Steps been followed are as:

Step 1:Data Exploration

Step 2: Data Preparation

Step 3: Data Visualization

Step 4:Data training

Step 5: Model Creation

Step 6: Performance Evaluation

## \*\*Data Exploration\*\*

## import all the necessary libraries

import warnings

#Ignoring unnecessory warnings

warnings.filterwarnings("ignore")

import numpy as np #for large and multi-dimensional arrays

import pandas as pd #for data manipulation and analysis

import nltk

##reading dataset

df = pd.read\_csv('../Dataset/Raw\_Data\_for\_train\_test.csv')

print(df.shape)

df.head()

df.shape

df.columns

## \*\*Data Preparation\*\*

# By printing the TRAGET\_D column we can clearly see that it has many Nan value

df["TARGET\_D"]

# Counting the Nan value present in the TARGET\_D column

df["TARGET\_D"].isna().sum()

# Now since column TARGET\_D has this much of Nan value, we can drop it from the dataset

df=df.drop(['TARGET\_D'],axis=1)

df

df['DONOR\_AGE']=df['DONOR\_AGE'].fillna(df['DONOR\_AGE'].mean())

df['DONOR\_AGE']=df['DONOR\_AGE'].astype('int64')

df['INCOME\_GROUP']=df['INCOME\_GROUP'].fillna(df['INCOME\_GROUP'].mode()[0])

df['INCOME\_GROUP']=df['INCOME\_GROUP'].astype('int64')

df['WEALTH\_RATING']=df['WEALTH\_RATING'].fillna(df['WEALTH\_RATING'].mode()[0])

df['WEALTH\_RATING']=df['WEALTH\_RATING'].astype('int64')

# Removed missing values from the dataset

df=df.dropna()

df

df.shape

# Now we need to replace some special charater like ? by some character in some column

df['URBANICITY']=df['URBANICITY'].str.replace('?','S')

df

df['SES']=df['SES'].str.replace('?','2')

df['SES']=df['SES'].astype('int64')

df['CLUSTER\_CODE']=df['CLUSTER\_CODE'].str.replace('.','40')

df['CLUSTER\_CODE']=df['CLUSTER\_CODE'].astype('int64')

df

from sklearn.preprocessing import LabelEncoder

le=LabelEncoder()

for i in list(df.columns):

if df[i].dtype=='object':

df[i]=le.fit\_transform(df[i])

## \*\*Data Visualization\*\*

# import matplotlib.pyplot as plt

# import seaborn as sns

# sns.heatmap(df.corr(),annot=True,cmap = 'YlGnBu')

import seaborn as sns

import matplotlib.pyplot as plt

sns.barplot(x=df['TARGET\_B'],y=df['IN\_HOUSE'],data=df)

sns.countplot(x=df['TARGET\_B'],

hue=df['SES'],data=df)

## \*\*Data Training\*\*

y=df['TARGET\_B']

X=df.drop(['TARGET\_B'],axis=1)

from sklearn.model\_selection import train\_test\_split

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,random\_state=0,test\_size=0.2)

# \*\*Model Creation\*\*

### \*\*Logistic Regression\*\*

# Logistic Regression

from sklearn.linear\_model import LogisticRegression

model1=LogisticRegression() # created a

model1.fit(X\_train, y\_train)

# Making predictions

pred1 = model1.predict(X\_test)

pred1

print("Accuracy of the LogisticRegression model comes to be: \n ")

print(model1.score(X\_train,y\_train))

### \*\*Linear Regression\*\*

# Linear Regression

from sklearn.linear\_model import LinearRegression

model2=LinearRegression() # created a

model2.fit(X\_train, y\_train)

# Making predictions

pred2 = model2.predict(X\_test)

pred2

print("Accuracy of the LinearRegression model comes to be: \n ")

print(model2.score(X\_train,y\_train))

### \*\*Ridge Regression\*\*

# Importing model

from sklearn.linear\_model import Ridge

model3 = Ridge()

#Fitting data into the model.

model3.fit(X\_train, y\_train)

# Making predictions on Test data

pred3 = model3.predict(X\_test)

pred3

print("Accuracy of the RidgeRegression model comes to be: \n ")

print(model3.score(X\_train,y\_train))

### \*\*K Neighbors Classifier\*\*

from sklearn.neighbors import KNeighborsClassifier

model4=KNeighborsClassifier(n\_neighbors=30)

#Fitting data into the model.

model4.fit(X\_train, y\_train)

# Making predictions on Test data

pred4 = model4.predict(X\_test)

pred4

print("Accuracy of the K Neighbors Classifier model comes to be: \n ")

print(model4.score(X\_train,y\_train))

### \*\*Lasso Regression\*\*

# Importing model

from sklearn.linear\_model import Lasso

model5 = Lasso()

#Fitting data into the model.

model5.fit(X\_train, y\_train)

# Making predictions on Test data

pred5 = model5.predict(X\_test)

pred5

print("Accuracy of the Lasso Regression model comes to be: \n ")

print(model5.score(X\_train,y\_train))

### \*\*Random Forest Classifier\*\*

from sklearn.metrics import accuracy\_score

from sklearn.ensemble import RandomForestClassifier

model6=RandomForestClassifier()

model6.fit(X\_train,y\_train)

pred6=model6.predict(X\_test)

pred6

print("Accuracy of the RandomForest model comes to be: \n ")

print(accuracy\_score(y\_test,pred6))

### \*\*Decision Tree Regression\*\*

# Importing decision tree regressor

from sklearn.tree import DecisionTreeRegressor

model7 = DecisionTreeRegressor()

#Fitting data into the model.

model7.fit(X\_train, y\_train)

# Making predictions on Test data

pred7 = model7.predict(X\_test)

pred7

print("Accuracy of the DecisionTree Classifier model comes to be: \n ")

print(accuracy\_score(y\_test,pred7))

### \*\*XG Boost Classifier\*\*

from xgboost import XGBRegressor,XGBClassifier

model8 = XGBClassifier()

#Fitting data into the model.

model8.fit(X\_train, y\_train)

# Making predictions on Test data

pred8 = model8.predict(X\_test)

pred8

print("Accuracy of the XG Boost model comes to be: \n ")

print(accuracy\_score(y\_test,pred8))

# \*\*Performance Evaluation\*\*

import numpy as np

from sklearn.metrics import mean\_squared\_error

print("Model\t\t\t RootMeanSquareError \t\t Accuracy of the model")

print("""Logistic Regression \t\t {:.4f} \t \t\t {:.4f}""".format( np.sqrt(mean\_squared\_error(y\_test, pred1)), model1.score(X\_train,y\_train)))

print("""Linear Regression \t\t {:.4f} \t \t\t {:.4f}""".format( np.sqrt(mean\_squared\_error(y\_test, pred2)), model2.score(X\_train,y\_train)))

print("""Ridge Regression \t\t {:.4f} \t \t\t {:.4f}""".format( np.sqrt(mean\_squared\_error(y\_test, pred3)), model3.score(X\_train,y\_train)))

print("""K Neighbors Classifier \t\t {:.4f} \t \t\t {:.4f}""".format( np.sqrt(mean\_squared\_error(y\_test, pred4)), model4.score(X\_train,y\_train)))

print("""Lasso Regression \t\t {:.4f} \t \t\t {:.4f}""".format( np.sqrt(mean\_squared\_error(y\_test, pred5)), model5.score(X\_train,y\_train)))

print("""Random Forest Classifier \t\t {:.4f} \t \t\t {:.4f}""".format( np.sqrt(mean\_squared\_error(y\_test, pred6)), accuracy\_score(y\_test,pred6)))

print("""Decision Tree Classifier \t\t {:.4f} \t \t\t {:.4f}""".format( np.sqrt(mean\_squared\_error(y\_test, pred7)), accuracy\_score(y\_test,pred7)))

print("""XG Boost Classifier\t\t {:.4f} \t \t\t {:.4f}""".format( np.sqrt(mean\_squared\_error(y\_test, pred8)), accuracy\_score(y\_test,pred8)))

Conclusion: Accuracy of Regression models- Linear regression, rigde regression, Lasso Regression is very low.

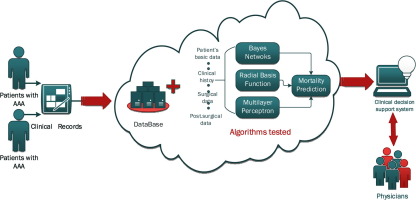
Whereas accuracy of Logistic Regression, K Neighbors Classifier, Random Forest Classifier, XGBoost Classifier, and Decision Tree Classifier is almost same.

And the Accuracy of the Random Forest is more, so it is efficient among all.most accurate organ donation.

**11. CONCLUSION AND FUTURE SCOPE**

* With the use of the Model, A medical organization can automate the health related queries without any hassle and process a large number of queries at once related to getting the correct match for the organ donors for the particular patients based on its data.
* Assistants of Doctors can use the model to serve the patients of different organ needs.
* Even a naïve person can use it for their own purpose after being consulted by a certified medical practitioner.

Conclusively, implementing this model in real life will be of great help



**12. REFERENCES**

[1] Gotlieb, Neta, et al. "The promise of machine learning applications in solid organ transplantation." NPJ digital medicine 5.1 (2022): 1-13.

[2] Briceño, Javier, María Dolores Ayllón, and Rubén Ciria. "Machine-learning algorithms for predicting results in liver transplantation: the problem of donor–recipient matching." Current Opinion in Organ Transplantation 25.4 (2020): 406-411.

[3] Data set – Kaggle (<https://www.kaggle.com/datasets/momohmustapha/donorsprediction>)

[4] Marrero, W.J., Lavieri, M.S., Guikema, S.D., Hutton, D.W. and Parikh, N.D., 2021. A machine learning approach for the prediction of overall deceased donor organ yield. Surgery, 170(5), pp.1561-1567.