Internship Report

On

Machine Learning models for prediction of diabetic

Submitted in partial fulfilment of the requirements for the award of the degree

of

Bachelor of Technology

in

Computer Science & Engineering

by

Sudhanshu Kumar	**(Roll No. 180101045)**
Verchaswa Sharma	**(Roll No.180101047)**
Dhanu Kumar	**(Roll No. 180101014)**
Dhruv Srivastava	**(Roll No. 180101015)**

Under the esteemed Supervision

of

Dr. Dilip Kumar Choubey
Assistant professor

Department of Computer Science and Engineering

Indian Institute of Information Technology Bhagalpur

October, 2021

Abstract

In this project, we were asked to experiment with a Diabetes dataset, and to explore how machine learning algorithms can be used to find the patterns in data. We were expected to gain experience using a common machine learning library, were expected to submit a report about the dataset and the algorithms used. With the suggestions algorithms and different experiences and comparing, numerical results obtained, the accuracy and efficiency of the method has been investigated and acceptable results compared to the logistic regression, MLP Classifier, random forest classifier methods were obtained. The accuracy achieved by functional classifiers Logistic regression, MLP classifier and Random forest classifier lies within the range of 75-85%. Among the three of them, Random Forest Classifier provides the best results for diabetes onset with an accuracy rate of 80.9% on the PIMA dataset. Hence, this proposed system provides an effective prognostic tool for healthcare officials. The results obtained can be used to develop a novel automatic prognosis tool that can be helpful in early detection of the disease.

Keywords: Diabetes, logistic regression, MLP classifier, random forest.

Contents

Objectives

- Internships are generally thought of to be reserved for college students looking to gain experience in a particular field. However, a wide array of people can benefit from Training Internships in order to receive real world experience and develop their skills.
- An objective for this position should emphasize the skills you already possess in the area and your interest in learning more.
- Internships are utilized in a number of different career fields, including architecture, engineering, healthcare, economics, advertising and many more.
- Some internships are used to allow individuals to perform scientific research while others are specifically designed to allow people to gain first-hand experience working.
- Utilizing internships is a great way to build your resume and develop skills that can be emphasized in your resume for future jobs. When you are applying for a Training Internship, make sure to highlight any special skills or talents that can make you stand apart from the rest of the applicants so that you have an improved chance of landing the position.

Chapter 1: Introduction

1.1 Overview

Diabetes is a major, and increasing global problem. The number of people affected with diabetes in 2000 was estimated to be 171 million worldwide. This figure is predicted to rise to 366 million by 2030, which represent around 4.4% of the estimated worldwide population. However, it has been shown that, through good management of blood glucose level (BGL), the associated and costly implications can be reduced significantly [1]. The human body requires the maintenance of blood glucose (BG) levels in a very narrow range (70-110 mg/dl) [3]. Many different factors affect these levels. The pancreas releases insulin and glycogen hormones to regulate the BG levels. Type 1 diabetes mellitus (T1D) is the consequence of an autoimmune attack on the pancreas that significantly impairs insulin prediction. Complications due to T1D include neuropathy, nephropathy, and retinopathy, with diabetes being a leading cause of renal failure, new blindness, and nontrauma amputations. These complications, in turn, impose a significant economic burden to society, wit indirect ad direct costs estimated at $245 billion for U.S. alone in 2012 [4]. Even more alarming, diabetes is a rapidly growing global epidemic for which the Centres for Disease Control estimate, will affect 1 in 3 adults in the U.S. by 2050, if current trends continue. For Type 1 (insulin-dependent) diabetes, the most common method for management is through monitoring the BGL using finger-prick blood tests taken several times a day, and adjusting the insulin doses based in those readings. For a dynamic, non linear, and complex condition as diabetes, this can be far from satisfactory. Factors such as insulin type and dose, diet, stress, exercise, illness, and pregnancy all have significant influences on the BGL. Management may be compromised through lack of data and, for some patients, an inability to interpret data adequately.

1.2 Objective

The main objective of this report is to detect innovative trends in prediction of diabetes in diabetic patients and then analyze these patterns in order to provide users with relevant and useful knowledge.

1.3 Motivation

Motivation to take up and maintain a healthy lifestyle is key to diabetes prevention and management. Motivations are driven by factors on the psychological, biological and environmental levels, which have each been studied extensively in various lines of research over the past 25 years. Here, we analyses various factor of cause of diabetics and apply different machine learning algorithms to predict the factor of diabetics, with a focus on people with diabetes.

1.4 Description of machine learning algorithms

(i) Logistic Regression:- Logistic Regression was used in the biological sciences in early twentieth century[10]. It was then used in many social science applications. Logistic Regression is used when the dependent variable(target) is categorical.

Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1". In the logistic model, the log-odds (the logarithm of the odds) for the value labeled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value).
It is also used in marketing applications such as prediction of a customer's propensity to purchase a product or halt a subscription, etc. In economics it can be used to predict the likelihood of a person ending up in the labor force, and a business application would be to predict the likelihood of a homeowner defaulting on a mortgage. Conditional random fields, an extension of logistic regression to sequential data, are used in natural language processing.

(ii) MLP Classifier:- MLP Classifier stands for Multi-layer Perceptron classifier which in the name itself connects to a Neural Network. Unlike other classification algorithms such as Support Vectors or Naive Bayes Classifier, MLPClassifier relies on an underlying Neural Network to perform the task of classification[11].
One similarity though, with Scikit-Learn [9] other classification algorithms is that implementing MLPClassifier takes no more effort than implementing Support Vectors or Naive Bayes or any other classifiers from Scikit-Learn.

(iii) Random Forest(RF) Classifier:- A random forest (RF) classifier is an ensemble classifier that produces multiple decision trees, using a randomly selected subset of training samples and variables. For classification tasks, the output of the random forest is the class selected by most trees.

1.5 Diabetes

Diabetes is one of the major deadliest diseases in the world. It is not only a disease but also have association to other kinds of diseases like heart attack, blindness, kidney diseases, etc. Normally identifying process is that patients need to visit a diagnostic center, consult their doctor, and sit tight for a day or more to get their reports. Moreover, every time they want to get their diagnosis report, they have to waste their money in vain.

Chapter 2: Literature Review

2.1 Review on classification algorithms

K.VijiyaKumar et al. [5] proposed random Forest algorithm for the Prediction of diabetes develop a system which can perform early prediction of diabetes for a patient with a higher accuracy by using Random Forest algorithm in machine learning technique. The proposed model gives the best results for diabetic prediction and the result showed that the prediction system is capable of predicting the diabetes disease effectively, efficiently and most importantly.

NonsoNnamoko et al. [6] offered envisaging diabetes onset: a collaborative supervised learning method. The outcomes are obtainable and related with analogous methods that used the same dataset in the works.

N. Joshi et al. [7] presented Diabetes Prediction Using Machine Learning Techniques aims to predict diabetes via three different supervised machine learning methods including: SVM, Logistic regression, ANN. This project pro- poses an effective technique for earlier detection of the diabetes disease

Song et al. [8] elucidated and defined using various factors such as Age, Glucose, BP, BMI, Skin Thickness etc. Diabetes Pedigree function, insulin, and pregnancy parameters not included

2.2 Summary

This project aims to help doctors and clinicians use the ML technique to predict diabetes early on. In this new Data technology, predictive analytics has earned a lot of prestige. There is a vast amount of medical evidence available today on the disease, its symptoms, the causes for the disease, and its health consequences. But this data is not adequately analyzed to predict a disease or to research it.

Chapter 3: Proposed approach

3.1 Overview

We have proposed three different approach to find best algorithm to predict diabetes in patients here we will discuss about logistic regression , MLP classifier and random forest classifier.

3.2 Dataset Description

The dataset is taken from the national institute of diabetes and digestive and kidney diseases (NIDDK). The data is stored in Kaggle and UCI data repository [1]. This dataset is mainly used to predict whether a Pima Indians female has diabetes or not. All the patients taken in the dataset are females of Pima Indians heritage of minimum age of 21 years. In order to decide a female in this dataset as diabetic, the following attributes are considered.

attributes	description
age	It shows the age in years. The range is 21 to 81 and the average age is 33.
Pregnancies	It shows that the number of times a female gets pregnant. The range is 0 to 17 and the average is 4.
Glucose	It shows the plasma glucose concentration level (2 hours). It is from 0 to 199 and the average is 121.
Blood pressure	It shows the diastolic blood pressure in mm Hg. It is from 0 to 122 and the average is 69.

Skin thickness	It shows the triceps skin thickness in mm. The range is 0 to 99 and the average is 21.
Insulin	It ranges from 0 to 846. The average is 80.
BMI	It shows body mass index in Kg/m2. The range is 0 to 67.1 and the average is 32.
Diabetes pedigree function	This function scores the likelihood of diabetes. It is from 0.078 to2.42 and the average is 0.47.
Outcome	It is either 0 or 1. Here, 0 means that a female has non-diabetic and 1 means that female is diabetes

In this dataset, the outcome used as the target class to predict that the Pima Indians female is diabetic or not. It has 768 instances (i.e., number of rows) and 9 columns (i.e., number of attributes).

We split our data in training and testing set in 70:30 ratio.

3.3 Algorithms proposed

This section discusses various supervised learning algorithms for classifying the diabetic and non-diabetics Pima Indians females. Note that these algorithms create the training dataset and testing dataset from the original dataset to classify or predict diabetes.

3.1.1 Logistic Regression

It is appropriate to use logistic regression when the dependent variable is binary [10], as we have to classify an individual in either type 1 or type 2 diabetes. Besides, it is used for predictive analysis and explains the relationship between a dependent variable and one or many independent variables, as shown in equation (1). Therefore, we used the sigmoid cost function as a hypothesis function ($h_\Theta(x)$). The aim is to minimize cost function $J(\Theta)$. It always results in classifying an example either in class 1 or class 2.

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}},$$
$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^i \log\ h_\theta\left(x^i\right) + \left(1 - y^i\right) \log\left(1 - h_\theta\left(x^i\right)\right)\right]. \quad (1)$$

3.1.2 MLP Classifier

For diabetes classification, we have fine-tuned multilayer perceptron in our experimental setup[11]. It is a network where multiple layers are joined together to make a classification method, as shown in Figure 1 The building block of this model is perceptron, which is a linear combination of input and weights. We used a sigmoid unit as an activation function. The proposed algorithm consists of three main steps. First, weights are initialized and

output is computed at the output layer (δ_k) using the sigmoid activation function. Second, the error is computed at hidden layers (δ_h) for all hidden units. Finally, in a backward manner, all network weights ($W_{i,j}$) are updated to reduce the network error.

Figure 1 shows the multilayer perceptron classification model architecture where eight neurons are used in the input layer because we have eight different variables. The middle layer is the hidden layer where weights and input will be computed using a sigmoid unit. In the end, results will be computed at the output layer. Backpropagation is used for updating weights so that errors can be minimized for predicting class labels. For simplicity, only one hidden layer is shown in the architecture, which in reality is much denser.
Input data from the input layer are computed on the hidden layers with the input values and weights initialized. Every unit in the middle layer called the hidden layer takes the net input, applies activation function "sigmoid" on it, and transforms the massive data into a smaller range between 0 and 1. The calculation is functional for every middle layer. The same procedure is applied on the output layer, which leads to the results towards the prediction for diabetes.

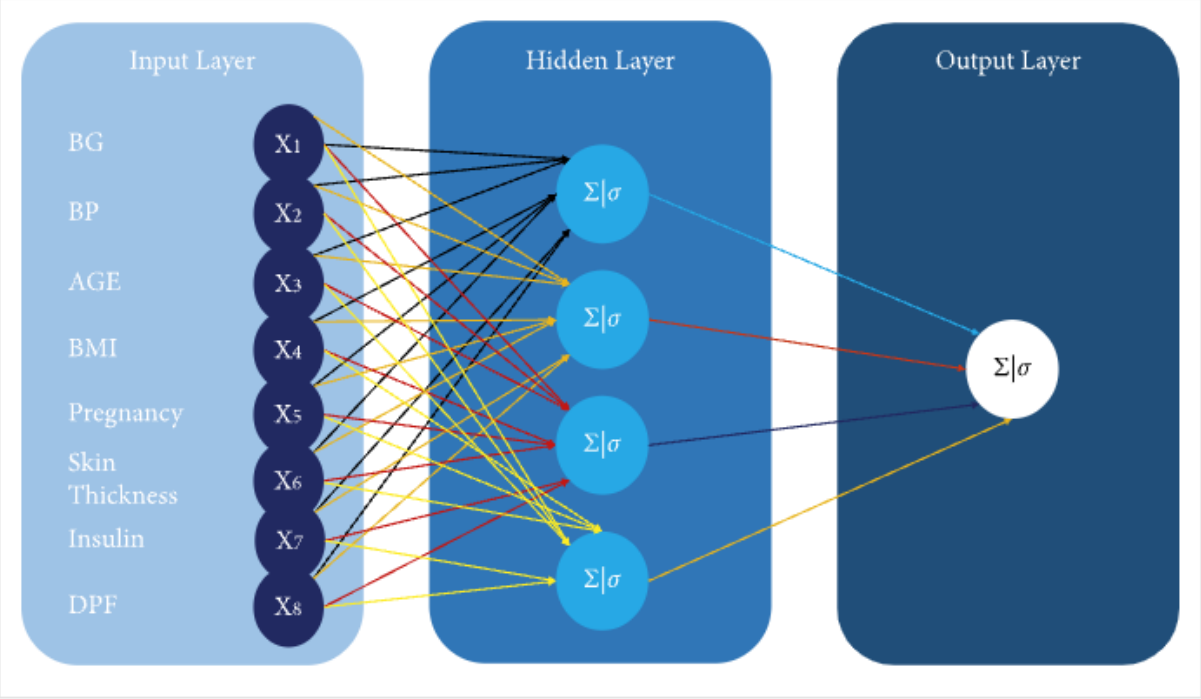

Figure 1 Proposed MLP architecture with eight variables as input for diabetes classification.

3.1.3 Random Forest(RF)

As its name implies, it is a collection of models that operate as an ensemble. The critical idea behind RF is the wisdom of the crowd, each model predicts a result, and in the end, the majority wins. It has been used in the literature for diabetic prediction and was found to be effective. Given a set of training examples $X = x_1, x_2, \ldots, x_m$ and their respective targets $Y = y_1, y_2, \ldots, y_m$, RF classifier iterates B times by choosing samples with replacement by fitting a tree to the training examples. The training algorithm consists of the following steps depicted in equation (2).
(i)For $b = 1...B$, sample with replacement n training examples from X and Y.

(ii)Train a classification tree f_b on X_b and Y_b.

$$f = \frac{1}{B}\sum_{b=1}^{B} f_b\left(x'\right). \qquad (2)$$

Chapter 4: Experimental Results

For diabetic prediction, we implemented three state-of-the-art algorithms, i.e., logistic regression, MLP Classifier, and Random Forest. Notably, we fine-tuned Random forest and compared its performance with other algorithms. It is evident that the Random Forest outperformed as compared to other algorithms implemented in this study.

n_estimator (the no of trees in the forest)	Accuracy
50	80%
100	78.35%
150	79.2%
200	80.9%

Here in table we saw that tuning the value of n_estimator in random forest we achieve the better accuracy of 80.9% from compare to other used algorithms like in Logistic Regression we get 78.35% and in MLP Classifier get 70.9% We will use the confusion matrix in MLP Classifier to determine the accuracy which is measured as the total number of correct predictions divided by the total number of predictions.

.

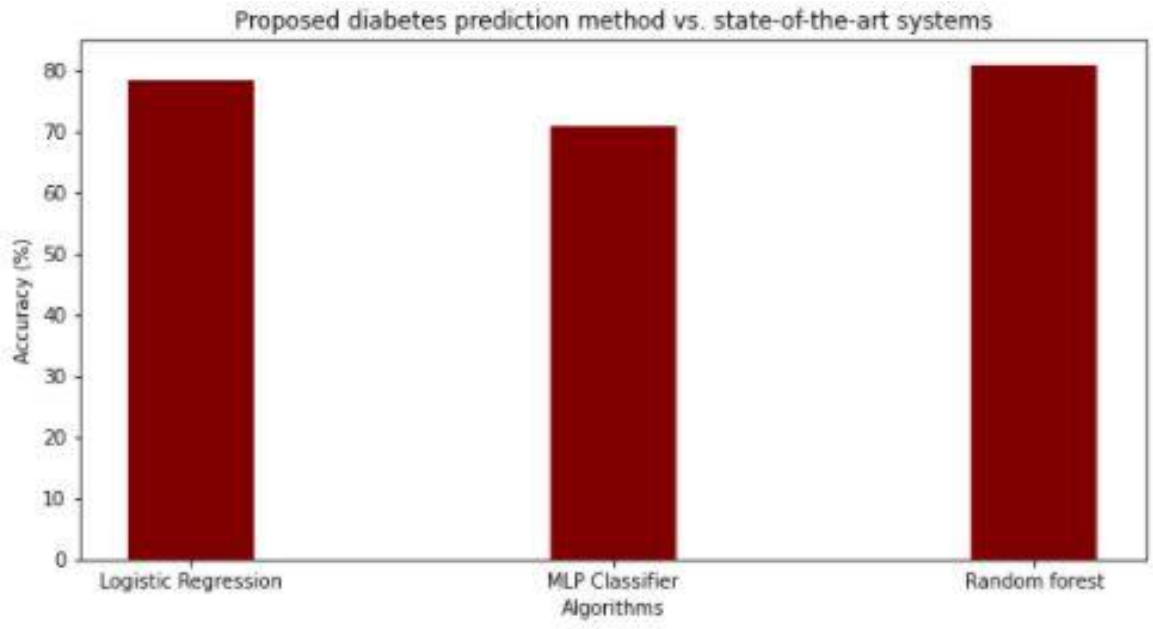

Figure 2

Chapter 5: Conclusion

In our work, we have addressed the problem of automatic glucose level prediction leveraging multi-patient data. Our aim is to learn a generalised glucose level prediction model from a multi-patient training set, and use it to predict the near future glucose levels of a new patient. Our proposed model (Logistic regression, random forest, MLP Classifier). We have combined these models to make custom cascades and stacks, in order to obtain better results. According to our experiments Random Forest obtained the best results. The Logistic regression also obtained good results.

References

[1] https://www.kaggle.com/uciml/pima-indians-diabetes-database

[2] https://turcomat.org/index.php/turkbilmat/article/view/4958/4155

[3] Contreras, I., Oviedo, S., Vettoretti, M., Visentin, R., & Veh´ı, J. (2017). Personalized blood glucose prediction: A hybrid approach using grammatical evolution and physiological models. PloS one, 12(11), e0187754.

[4] Cobelli C, Renard E, Kovatchev B. Artificial pancreas: Past, present, future. Diabetes. 2011;60: 2672–2682. pmid:22025773

[5] K.VijiyaKumar, B.Lavanya, I.Nirmala, S.Sofia Caroline, "Random Forest Algorithm for the Prediction of Diabetes ".Proceeding of International Conference on Systems Computation Automation and Networking, 2019.

[6] NonsoNnamoko, Abir Hussain, David England, "Predicting Diabetes Onset: An Ensemble Supervised Embedded and Com- munication Systems (ICIIECS), 2017.

[7] Tejas N. Joshi, Prof. Pramila M. Chawan, "Diabetes Prediction Using Machine Learning Techniques".Int. Journal of Engineer- ing Research and Application, Vol. 8, Issue 1, (Part -II) Janu- ary 2018, pp.-09-13

[8] . Song, Y., Liang, J., Lu, J., & Zhao, X. (2017). An efficient instance selection algorithm for k nearest neighbor regression. Neurocomputing, 251, 26-34

[9] https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.htm

[10] https://en.wikipedia.org/wiki/Logistic_regression#cite_note-safety-13

[11] https://analyticsindiamag.com/a-beginners-guide-to-scikit-learns-mlpclassifier/

Appendix

Code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.neural_network import MLPClassifier
from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error,r2_score
```

```
df= pd.read_csv('/content/diabetes (1).csv')
```

```
df.shape
df.describe().transpose()
```

	count	mean	std	min	25%	50%	75%	max
Pregnancies	768.0	3.845052	3.369578	0.000	1.00000	3.0000	6.00000	17.00
Glucose	768.0	120.894531	31.972618	0.000	99.00000	117.0000	140.25000	199.00
BloodPressure	768.0	69.105469	19.355807	0.000	62.00000	72.0000	80.00000	122.00
SkinThickness	768.0	20.536458	15.952218	0.000	0.00000	23.0000	32.00000	99.00
Insulin	768.0	79.799479	115.244002	0.000	0.00000	30.5000	127.25000	846.00
BMI	768.0	31.992578	7.884160	0.000	27.30000	32.0000	36.60000	67.10
DiabetesPedigreeFunction	768.0	0.471876	0.331329	0.078	0.24375	0.3725	0.62625	2.42
Age	768.0	33.240885	11.760232	21.000	24.00000	29.0000	41.00000	81.00
Outcome	768.0	0.348958	0.476951	0.000	0.00000	0.0000	1.00000	1.00

```
df['dibetes']=df['Outcome']
```

```
df.drop('Outcome',axis=1,inplace=True)
```

```
df.describe().transpose()
```

	count	mean	std	min	25%	50%	75%	max
Pregnancies	768.0	3.845052	3.369578	0.000	1.00000	3.0000	6.00000	17.00
Glucose	768.0	120.894531	31.972618	0.000	99.00000	117.0000	140.25000	199.00
BloodPressure	768.0	69.105469	19.355807	0.000	62.00000	72.0000	80.00000	122.00
SkinThickness	768.0	20.536458	15.952218	0.000	0.00000	23.0000	32.00000	99.00
Insulin	768.0	79.799479	115.244002	0.000	0.00000	30.5000	127.25000	846.00
BMI	768.0	31.992578	7.884160	0.000	27.30000	32.0000	36.60000	67.10
DiabetesPedigreeFunction	768.0	0.471876	0.331329	0.078	0.24375	0.3725	0.62625	2.42
Age	768.0	33.240885	11.760232	21.000	24.00000	29.0000	41.00000	81.00
dibetes	768.0	0.348958	0.476951	0.000	0.00000	0.0000	1.00000	1.00

```
test = df['dibetes']
train=df.drop('dibetes',axis=1,)
```

```
train.shape
```

```
(768, 8)
```

```
test.shape
```

```
(768,)
```

```
X_train,X_test,y_train,y_test=train_test_split(train,test,test_size=0.3,random_state=1)
```

```
from sklearn.linear_model import LinearRegression

linear_regression = LinearRegression()
linear_regression.fit(train, test)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
from sklearn.linear_model import LogisticRegression
```

```
logisticRegr = LogisticRegression()
```

```
logisticRegr.fit(X_train, y_train)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

```
predictions = logisticRegr.predict(X_test)
```

```
score = logisticRegr.score(X_test, y_test)
print(score)
```

```
0.7835497835497836
```

```
mlp=MLPClassifier(hidden_layer_sizes=(8,8,8),activation='relu',solver='adam',max_iter=100)
mlp.fit(X_train,y_train)
predict_train=mlp.predict(X_train)
predict_test=mlp.predict(X_test)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:571: ConvergenceWarning: Stochastic Optimizer: Maximum itera
  % self.max_iter, ConvergenceWarning)
```

```
def accuracy(confusion_matrix):
   diagonal_sum = confusion_matrix.trace()
   sum_of_all_elements = confusion_matrix.sum()
   return diagonal_sum / sum_of_all_elements
```

```
from sklearn.metrics import confusion_matrix,classification_report
cm = confusion_matrix(predict_train,y_train)
```

```
print("Accuracy of MLPClassifier : "'', accuracy(cm))
```

```
Accuracy of MLPClassifier :  0.7094972067039106
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
# creating a RF classifier
clf = RandomForestClassifier(n_estimators = 200)

# Training the model on the training dataset
# fit function is used to train the model using the training sets as parameters
clf.fit(X_train, y_train)

# performing predictions on the test dataset
y_pred = clf.predict(X_test)

# metrics are used to find accuracy or error
from sklearn import metrics
print()

# using metrics module for accuracy calculation
print("ACCURACY OF THE MODEL: ", metrics.accuracy_score(y_test, y_pred))
```

```
ACCURACY OF THE MODEL:  0.8095238095238095
```