TABLE OF CONTENTS

ABSTRACT

Diabetes mellitus, a chronic metabolic disorder, poses a significant global health challenge due to its increasing prevalence and associated complications. Early detection and prediction of diabetes risk are crucial for effective prevention and management strategies. In this project, we propose a predictive modeling approach leveraging machine learning techniques to assess the risk of developing diabetes. The dataset comprises demographic, clinical, and lifestyle-related features collected from a diverse population. Preprocessing techniques are applied to handle missing values, feature scaling, and encoding categorical variables. Various machine learning algorithms, including logistic regression, decision trees, random forests, support vector machines, and neural networks, are employed to build predictive models. Performance evaluation metrics such as accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC) are utilized to assess model performance and identify the most effective algorithm. Additionally, feature importance analysis is conducted to identify the significant predictors contributing to diabetes risk. The proposed predictive modeling framework holds the potential to aid healthcare practitioners in identifying individuals at high risk of developing diabetes, enabling early intervention and personalized preventive strategies to mitigate the burden of this debilitating disease. Diabetes is considered as one of the deadliest and chronic diseases which causes an increase in blood sugar.

Many complications occur if diabetes remains untreated and unidentified. The tedious identifying process results in visiting a patient to a diagnostic center and consulting doctor. But the rise in machine learning approaches solves this critical problem. The motive of this study is to design a model which can prognosticate the likelihood of diabetes in patients with maximum accuracy. Therefore, three machine learning classification algorithms namely Decision Tree, SVM and Naive Bayes are used in this experiment to detect diabetes at an early stage. Experiments are performed on Pima Indians Diabetes Database (PIDD) which is sourced from UCI machine learning repository. The performances of all the three algorithms are evaluated on various measures like Precision, Accuracy, F-Measure, and Recall. Accuracy is measured over correctly and incorrectly classified instances. Results obtained show Naive Bayes outperforms with the highest accuracy of 76.30% comparatively other algorithms. These results are verified using Receiver Operating Characteristic (ROC) curves in a proper and systematic manner.

INTRODUCTION

Diabetes, a complex metabolic disorder characterized by high blood sugar levels, has emerged as a global health crisis in recent decades. According to the World Health Organization (WHO), diabetes affects millions of people worldwide, with its prevalence steadily rising across all age groups and demographics. The burden of diabetes extends beyond its immediate health implications, encompassing socioeconomic and public health challenges.

The rising prevalence of diabetes presents a pressing need for innovative approaches to its prevention and management. Traditional methods of diabetes diagnosis often rely on clinical symptoms or laboratory tests, which may only detect the disease at advanced stages. Early detection of diabetes risk factors is crucial for effective prevention and timely intervention to mitigate its adverse effects on health.

In this context, predictive modeling techniques have garnered significant attention in healthcare research. Predictive models utilize data analytics, machine learning algorithms, and statistical analysis to forecast the likelihood of disease occurrence or progression based on individual characteristics and risk factors. By leveraging large datasets encompassing demographic, lifestyle, clinical, and genetic information, predictive models enable healthcare practitioners to identify individuals at high risk of developing diabetes long before clinical symptoms manifest.

The importance of diabetes prediction in healthcare cannot be overstated. Early identification of individuals at risk allows for targeted interventions such as lifestyle modifications, dietary changes, and pharmacological interventions to prevent or delay the onset of diabetes and its complications.

Furthermore, predictive models enable healthcare systems to allocate resources more efficiently, prioritize high-risk populations, and implement preventive strategies on a population-wide scale. In light of the growing diabetes epidemic and its profound impact on public health, the development and implementation of accurate and reliable predictive research. By harnessing the power of data-driven insights and predictive analytics, we can empower individuals, healthcare providers, and policymakers to combat diabetes effectively and improve health outcomes for millions worldwide.

OBJECTIVES

Diabetes prediction applications using machine learning techniques to predict the probability of an individual developing diabetes.

Here, some of the objectives related to diabetes prediction are below:

1. Developing a Robust Predictive Model:

The primary objective of the project is to develop a robust predictive model for diabetes risk assessment. This entails leveraging advanced machine learning algorithms and data analytics techniques to analyze diverse datasets encompassing demographic, lifestyle, clinical, and genetic information.

2. Real-Time Prediction Capability:

Another key objective is to develop a real-time prediction capability, enabling instantaneous risk assessment and proactive intervention. By implementing scalable and efficient algorithms, the project seeks to enable timely identification of individuals at high risk of developing diabetes, facilitating early intervention and preventive measures.

3. Personalized Risk Assessment:

The project aims to facilitate personalized risk assessment by tailoring predictive models to individual characteristics and risk factors. By considering factors such as age, gender, family history, lifestyle habits, and clinical biomarkers, the goal is to provide customized risk scores and recommendations tailored to each individual's unique profile.

4. Integration with Healthcare Systems:

A crucial objective is to integrate the predictive model seamlessly with existing healthcare systems and electronic health records (EHRs). This involves developing interoperable interfaces and data exchange protocols to facilitate seamless integration with healthcare providers' workflows and decision-making processes.

.

LITERATURE SURVEY

Diabetes prediction applications leverage machine learning techniques to analyze medical data andidentify individuals at risk of developing the disease.

Here, some of the surveys or reviews regarding diabetes prediction application are below:

1. "Survey on Clinical Prediction Models for Diabetes Prediction":

This paper provides a detailed overview of predictive models, from basic to state-of-the-art, focusing on various types of predictive models and their applications in healthcare, particularly in diabetes prediction. It covers topics such as diabetes diagnosis, self-management, and prevention using predictive analytics.

2. "A Survey on Diabetes Mellitus Prediction Using Machine Learning Techniques":

This survey discusses the major findings related to diabetes prediction using machine learning methods. It explores the state-of-the-art approaches for predicting diabetes based on available data.

3. "A Comprehensive Survey on Diabetes Type-2 (T2D) Forecast":

This study analyzes current literature on the effectiveness of machine learning techniques for early detection of type-2 diabetes. It focuses on predicting diabetes at an early stage.

4. "A Survey on Diabetes Prediction Models Using Data Mining Techniques":

This review paper provides an overview of recent studies on diabetes prediction models developed using data mining techniques. It covers issues, challenges, and advancements in this field.

5. "Framework for Early-Stage Diabetes Mellitus Risk Prediction":

Contemporary literature includes research studies involving different machine learning techniques for early-stage diabetes prediction. These studies aim to detect diabetes early, allowing for timely intervention.

6. "Review of Diabetic Prediction Models":

This paper analyzes various diabetic prediction models to identify the best quality research. It synthesizes different studies to provide a comprehensive understanding of the methods used for diabetes prediction.

EXISTING SYSTEM

The existing system was taken in order to meet the demands of this system and solve the problemsof the existing system by implementing the classifier.

There have been several machine learning-based systems developed for diabetes prediction and classification. These systems aim to identify individuals at risk of diabetes or predict the onset of the disease. Let's explore a few of them:

Classification Algorithms Used:

Random Forest (RF), Multilayer Perceptron (MLP), and Logistic Regression (LR) were employed for diabetes classification.

MLP outperformed other classifiers with an accuracy of 86.08%.

Predictive Analysis:

For predictive analysis, they used Long Short-Term Memory (LSTM), Moving Averages (MA), and Linear Regression (LR).

Disadvantages:

Limited Predictive Accuracy: Existing systems may not achieve high accuracy in predicting diabetes. While machine learning techniques have been employed, achieving consistently accurate results remains challenging.

Data Insufficiency: Some systems suffer from a lack of sufficient data. Accurate predictions require robust datasets, and inadequate information can lead to suboptimal results.

Lack of Real-Time Data: Many existing systems do not operate in real time. Timely predictions are crucial for effective management and intervention, especially in healthcare scenarios

Security Vulnerabilities: Privacy and security concerns are significant. Ensuring the confidentiality and integrity of patient data is essential, and vulnerabilities in the system can compromise patient information.

PROPOSED SYSTEM

The proposed diabetes prediction system has two main stages that work together to achieve the desired results. The first stage of the proposed system is the data preparation, and the second one is the classification. However, the input into the system is the dataset and the output will be one class which represents the healthy or the diabetic.

Here are three proposed systems for diabetes prediction:

1.K-Nearest Neighbor (K-NN) and Decision Tree (DT):

This system combines K-NN for data filtering and a DT-based classification approach.

Undesired data is eliminated using K-NN.

Features are extracted from preprocessed data.

DT assigns each data sample to its appropriate class.

2.Two-Phase System (Disease Prediction and Severity Estimation):

In the disease prediction phase:

Preprocessing is performed on the Pima dataset.

Features are extracted from the preprocessed data.

Classification is done using OWDANN.

The severity level estimation phase follows.

3.Deep Neural Network (DNN):

A highly accurate system based on DNN.

DNNs are powerful for complex data representation and predictions.

Advantages:

Early Detection and Prevention:

The proposed system leverages machine learning techniques to identify patterns associated with diabetes at an early stage.Early detection allows patients to take proactive measures for prevention, leading to better healthoutcomes.

Precision and Accuracy:

Machine learning models used in the system, such as random forest (RF), multilayer perceptron (MLP), and logistic regression (LR), achieve high accuracy rates.

For instance, MLP outperforms other classifiers with an impressive 86.08% accuracy in diabetes classification.

Cost and Time Savings:

Predictive capabilities enable patients to take preventive actions and manage diabetes early.

Early prediction not only helps prevent complications but also saves time and money.

Improved Healthcare Management: The application can provide valuable information to medical professionals, allowing them to tailor treatment plans for individual patients based on their predicted risk. This can lead to more effective management of the disease and potentially reduce complications.

Remote Monitoring: Patients in remote or underserved areas can benefit from these systems through telemedicine, ensuring they receive timely care and monitoring without needing frequent hospital visits.

Research and Development: The data collected and analyzed by these systems can be invaluable for research, leading to new discoveries and innovations in diabetes treatment and prevention.

Support for Clinical Trials: These systems can identify suitable candidates for clinical trials, accelerating the development of new treatments and therapies for diabetes.

ARCHITECTURE AND DESIGN

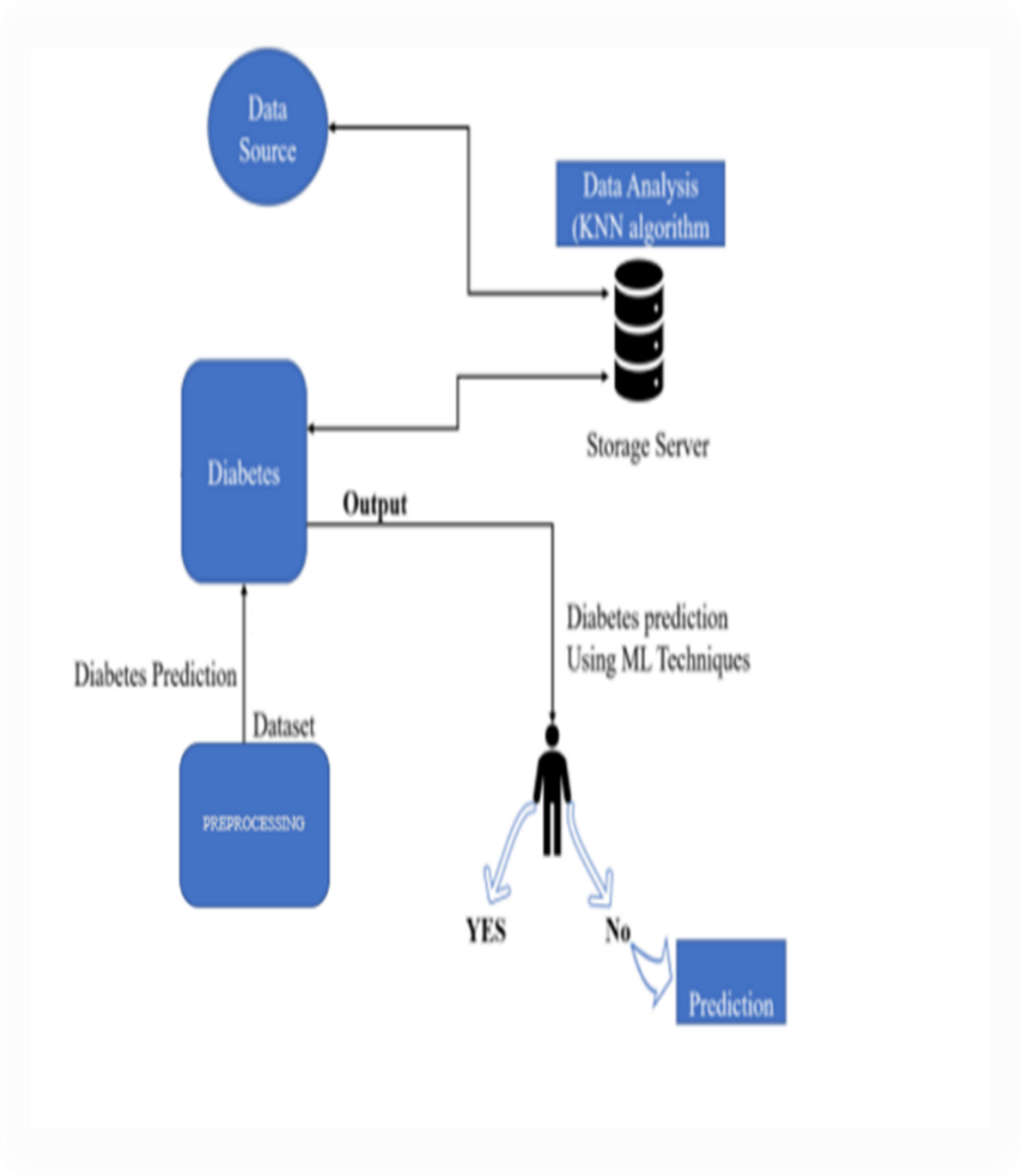


Fig-1: DIABETES PREDICTION APPLICATION

Diabetes prediction is a crucial area where machine learning techniques play a significant role. Let's explore some architectures and approaches for predicting diabetes:

Machine Learning-Based Approach:

In a study titled "Machine Learning-Based Diabetes Classification and Prediction for Healthcare Applications," researchers proposed an approach for diabetes classification, early-stage identification, and prediction1.

Architecture:

Classification:

Three classifiers were employed: Random Forest (RF), Multilayer Perceptron (MLP), and Logistic Regression (LR).

MLP outperformed other classifiers with 86.08% accuracy.

Prediction:

For predictive analysis, they used Long Short-Term Memory (LSTM), Moving Averages (MA), and Linear Regression (LR).

LSTM improved significantly with 87.26% accuracy for diabetes.

They also proposed an IoT-based hypothetical diabetes monitoring system for monitoring blood glucose levels.

The approach demonstrated adaptability in various public healthcare applications.

K-Means Clustering and SVM Approach:

A novel architecture combines K-means clustering and Support Vector Machine (SVM) for predicting diabetes patients. Features extracted from K-means are classified using an SVM classifier.

SYSTEM AND SOFTWARE REQUIREMENTS

H/W System Configuration:

Processor	:	I3/Intel Processor
RAM	:	4GB (min)
Hard Disk	:	160GB
Key Board	:	Standard Windows Keyboard
Mouse	:	Two or Three Button Mouse
Monitor	:	SVGA

S/W System Configuration:

Operating System	:	Windows 10
Front End	:	HTML, CSS, BOOTSRAP
Scripts	:	JavaScript, J query.
Server side Script	:	Python
Framework	:	Django, Flask

UML DIAGRAMS

UML represents Unified Modeling Language. UML is an institutionalized universally useful showing dialect in the subject of article situated programming designing. The fashionable is overseen, and become made by way of, the Object Management Group.

The goal is for UML to become a regular dialect for making fashions of item arranged PC programming. In its gift frame UML contains two noteworthy components: a Meta-show and documentation. Later on, a few type of methods or system can also likewise be brought to; or related with, UML.The Unified Modeling Language is a popular dialect for indicating, Visualization, Constructing and archiving the curios of programming framework, and for business demonstratingand different non-programming frameworks.

The UML speaks to an accumulation of first-rate building practices which have verified fruitful in the showing of full-size and complicated frameworks.The UML is a essential piece of creating gadgets located programming and the product development method. The UML makes use of commonly graphical documentations to specific theplan of programming ventures.

GOALS:

The Primary goals inside the plan of the UML are as in step with the subsequent:

1. Provide clients with prepared-to-utilize, expressive visual showing Language on the way to create and change massive models.

2. Provide extendibility and specialization units to make bigger the middle ideas.

3. Be free of specific programming dialects and advancement manner.

4. Provide a proper cause for understanding the displaying dialect.

5. Encourage the improvement of OO gadgets exhibit.

6. Support large amount advancement thoughts, for example, joint efforts, systems, examples and components.

7. Integrate widespread procedures.

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

Fig-2: Use Case Diagram of Diabetes Prediction Application

CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

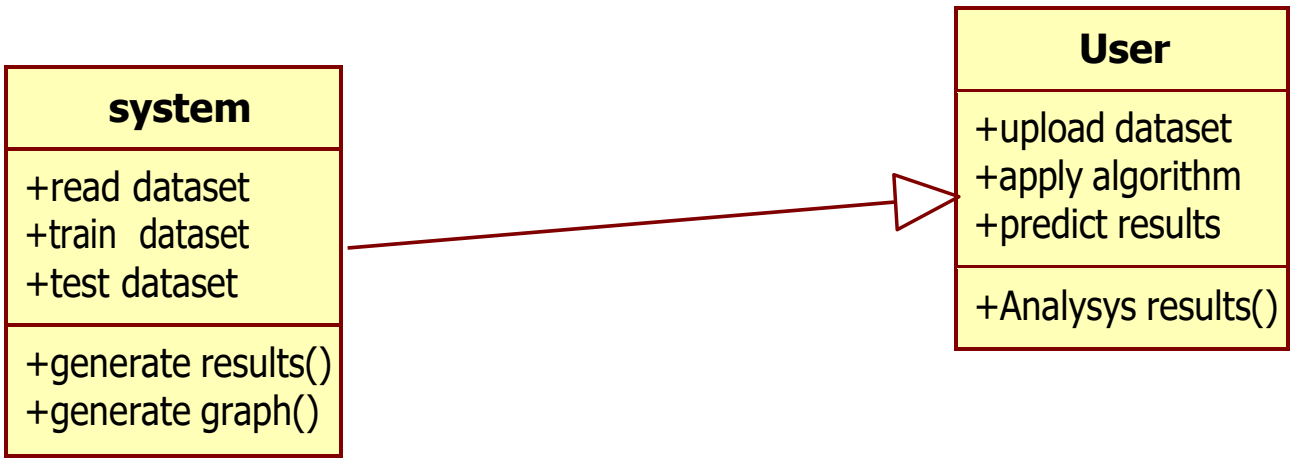


Fig-3: Class Diagram of Diabetes Prediction Application

.

SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

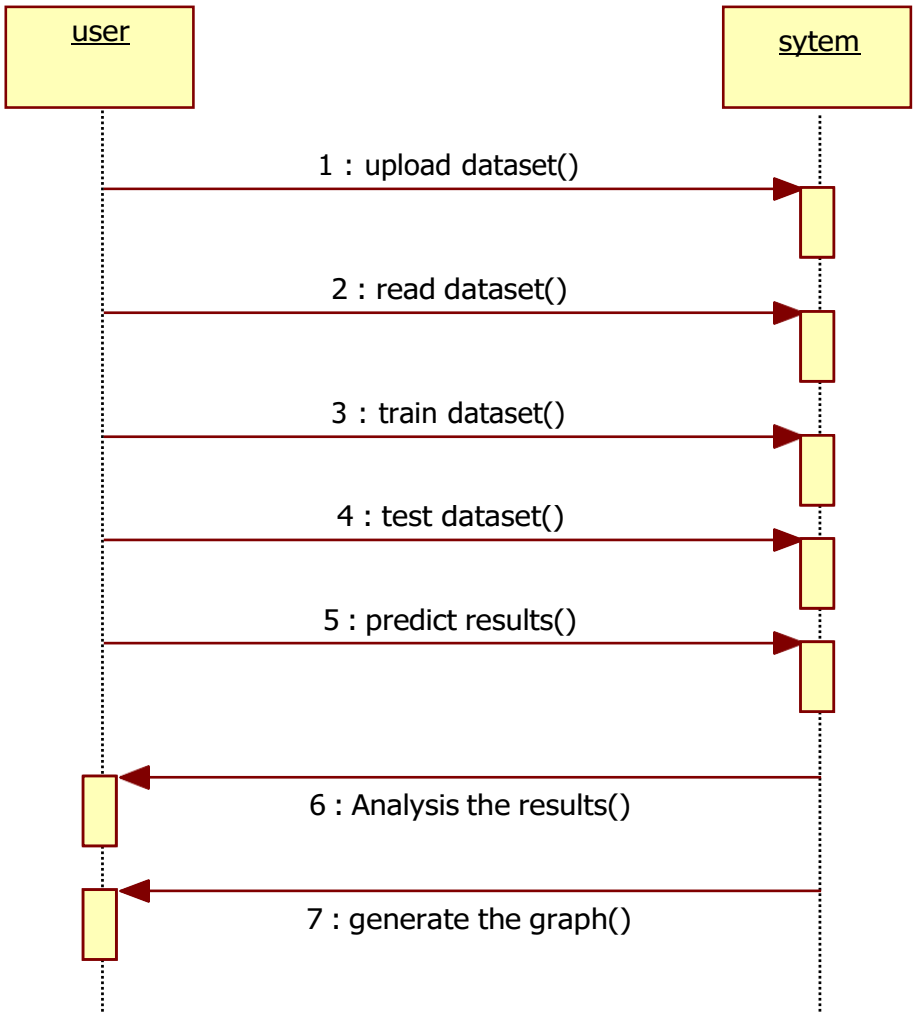


Fig-4: Sequence Diagram of Diabetes Prediction Application

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

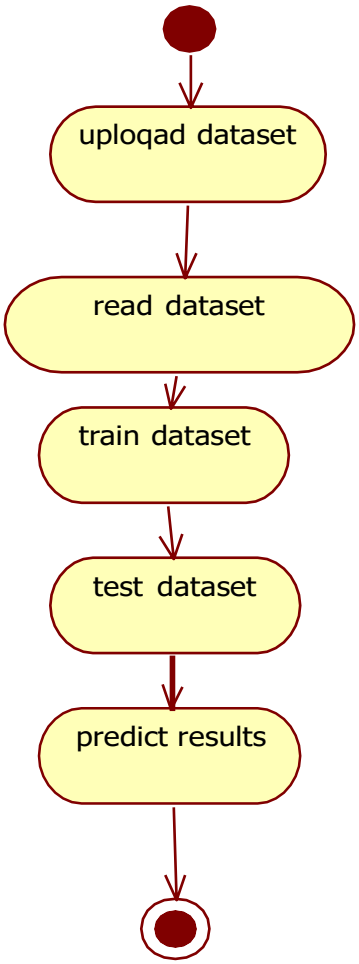


Fig-5: Activity Diagram of Diabetes Prediction Application

ALGORITHMS

SVM (Support Vector Machine):

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyper plane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyper plane which categorizes new examples.

An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification, implicitly mapping their inputs into high-dimensional feature spaces.

Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

Naive Bayes:

Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems. The technique is easiest to understand when described using binary or categorical input values.

It is called naive Bayes or idiot Bayes because the calculations of the probabilities for each hypothesis are simplified to make their calculation tractable. Rather than attempting to calculate the values of each attribute value P(d1, d2, d3|h), they are assumed to be conditionally independent given the target value and calculated as P(d1|h) * P(d2|H) and so on.

This is a very strong assumption that is most unlikely in real data, i.e. that the attributes do not interact. Nevertheless, the approach performs surprisingly well on data where this assumption does not hold.

KNN (K NEAREST NEIGHBORS):

K-Nearest Neighbors is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection.

It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data).

We are given some prior data (also called training data), which classifies coordinates into groups identified by an attribute.

K can be kept as an odd number so that we can calculate a clear majority in the case where only two groups are possible. With increasing K, we get smoother, more defined boundaries across different classifications. Also, the accuracy of the above classifier increases as we increase the number of data points in the training set.

Logistics Regression:

Logistic regression is another technique borrowed by machine learning from the field of statistics. It is the go-to method for binary classification problems (problems with two class values). In this post you will discover the logistic regression algorithm for machine.

Random Forest:

Many relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models. The low correlation between models is the key. Just like how investments with low correlations (like stocks and bonds) come together to form a portfolio that is greater than the sum of its parts, uncorrelated models can produce ensemble predictions that are more accurate than any of the individual predictions.

DECISION TREE (C4.5):

Decision trees are the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute. This process is then repeated for the sub tree rooted at the new node. Representation of tree is shown below:

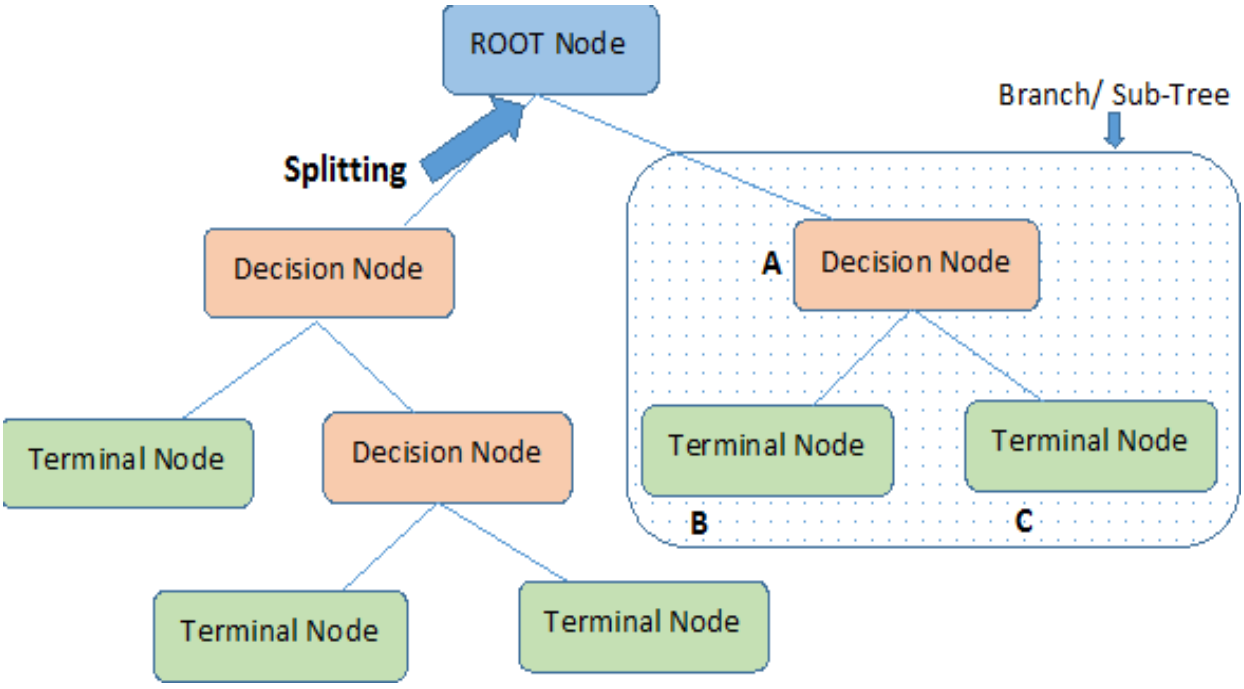


Fig-6 Decision Tree

IMPLEMENTATION

CODE:

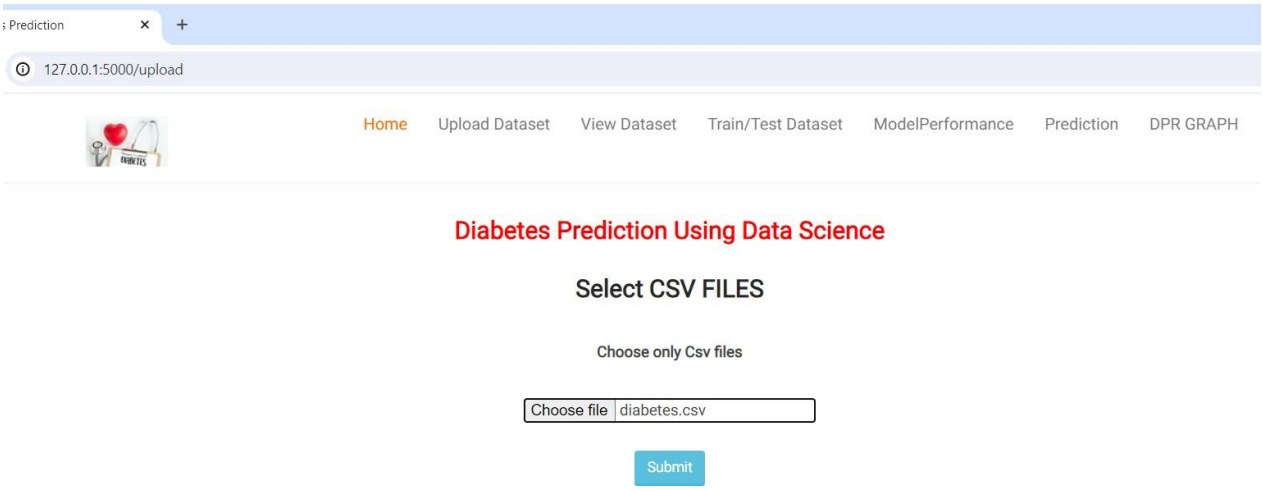
```
from flask import Flask, render_template, request, session, url_for, Response
import pandas as pd
import numpy as np
from werkzeug.utils import redirect
from sklearn.model_selection import train_test_split
from sklearn import tree
from sklearn.metrics import accuracy_score
from sklearn import linear_model
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC

from random import randint
import time
import json

accuracy =[]

app = Flask(__name__)
global dt1, LR1, RF1, NB1, KNN1, SVM1


def f(x_train, x_test, y_train, y_test):
    global X_trains, X_tests, y_trains, y_tests
    X_trains = pd.DataFrame(x_train)
    X_tests = pd.DataFrame(x_test)
    y_trains = pd.DataFrame(y_train)
    y_tests = pd.DataFrame(y_test)
    print("HELLO ++++++++++++++++++++++++++++++++++++++++++++++WORLD")
    print(X_trains)
```

```
@app.route( rule: '/uploaddataset', methods=["POST", "GET"])
def uploaddataset_csv_submitted():
    if request.method == "POST":
        csvfile = request.files['csvfile']
        result = csvfile.filename
        file = "C:/RTR/DIABETIES DATASCIENCE/" + result
        print(file)

        session['filepath'] = file

        return render_template( template_name_or_list: 'uploaddataset.html', msg='sucess')
    return render_template('uploaddataset.html')


@app.route( rule: '/viewdata', methods=["POST", "GET"])
def viewdata():
    session_var_value = session.get('filepath')
    print("Hello world")
    print("session variable is=====" + session_var_value)
    df = pd.read_csv(session_var_value)
    # print(df)
    x = pd.DataFrame(df)

    return render_template( template_name_or_list: "view.html", data=x.to_html())

    # return render_template('view.html', name=session_var_value, data=df.to_html())


@app.route( rule: '/traintestdataset', methods=["POST", "GET"])
def traintestdataset_submitted():
    if request.method == "POST":
        value = request.form['traintestvalue']
```

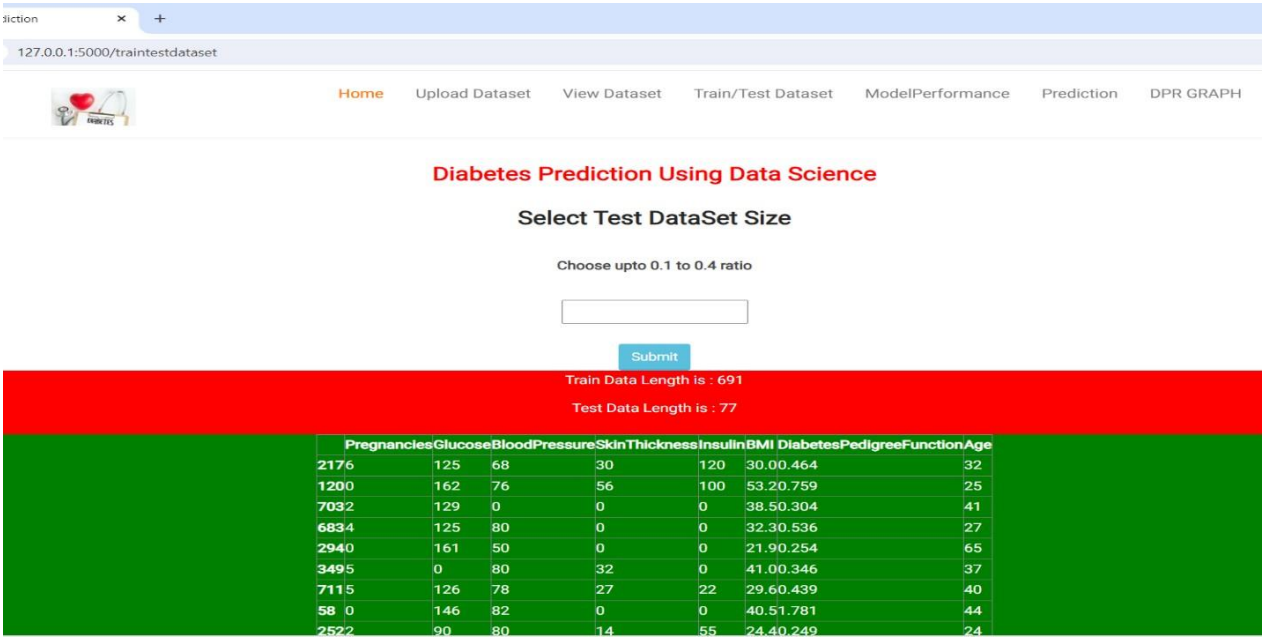
```
def traintestdataset_submitted():

        return render_template( template_name_or_list: 'traintestdataset.html', msg='sucess', data=X_train1.to_html(),
                               X_trainlenvalue=X_trainlen, y_testlenval=y_testlen)
    return render_template('traintestdataset.html')


@app.route( rule: '/modelperformance', methods=["POST", "GET"])
def selected_model_submitted():
    if request.method == "POST":
        selectedalg = int(request.form['algorithm'])


        print(X_trains)
        print(X_tests)
        print(
            "$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$")
        print(y_trains)
        print(
            "++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
        print(y_tests)

        if (selectedalg == 1):
            model = tree.DecisionTreeClassifier()

            model.fit(X_trains, y_trains)
            model.fit(X_trains, y_trains)
            y_pred = model.predict(X_tests)
            accuracyscore = accuracy_score(y_tests, y_pred)
            # accuracy_score = model.score(X_trains,y_trains)
            abc4 = accuracyscore
            accuracy.append(abc4)
```

```
def selected_model_submitted():
            print(accuracyscore)
            abc9 = accuracyscore
            accuracy.append(abc9)
            return render_template( template_name_or_list: 'modelperformance.html', msg="accuracy_score", score=accuracyscore,
                                   model="SupportVectorMachine")

    return render_template('modelperformance.html')


@app.route( rule: '/prediction', methods=["POST", "GET"])
def prediction():
    if request.method == "POST":
        list1 = []
        pre = request.form['pre']
        glu = request.form['glu']
        bp = request.form['bp']
        ski = request.form['ski']
        ins = request.form['ins']
        bmi = request.form['bmi']
        dpf = request.form['dpf']
        age = request.form['age']
        list1.extend([pre, glu, bp, ski, ins, bmi, dpf, age])
        print(list1)

        model = SVC()
        model.fit(X_trains, y_trains)
        predi = model.predict([list1])
        print(predi)
        pre = predi
        print(pre)
```

RESULTS

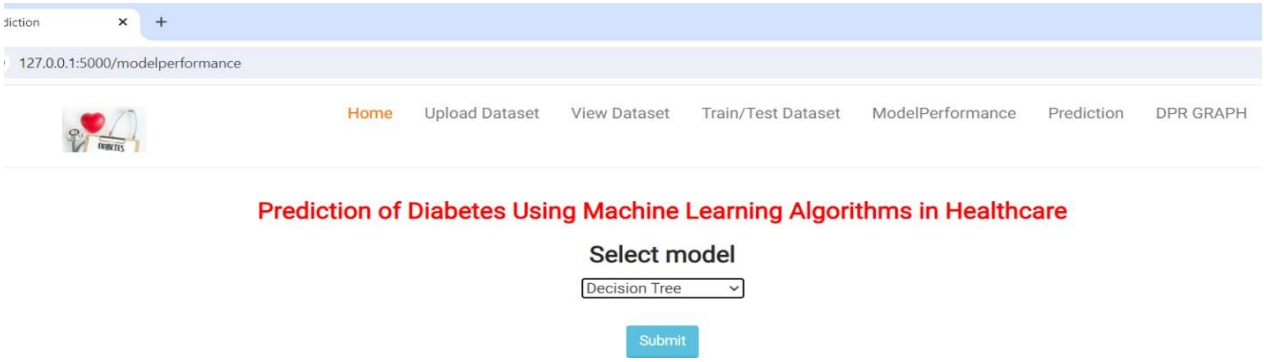


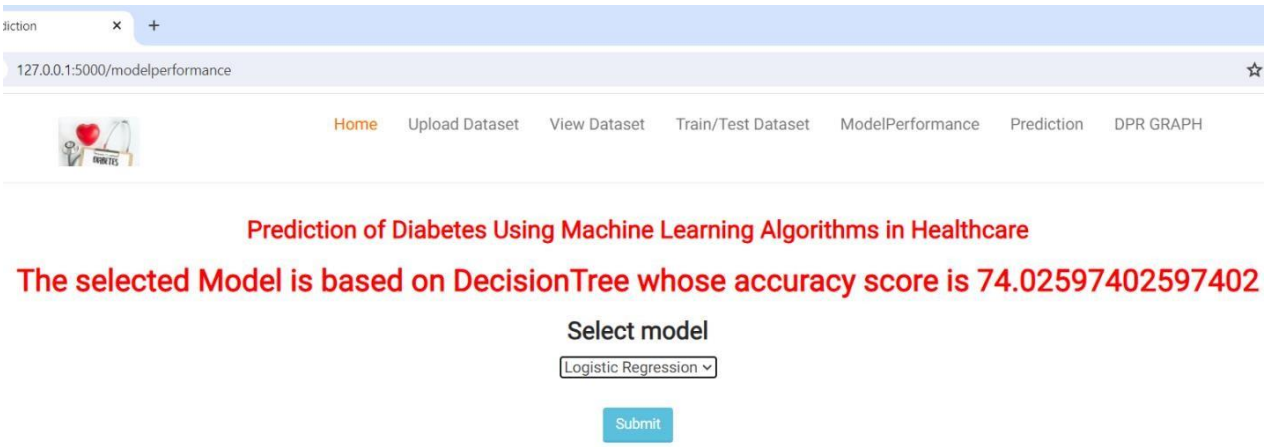
Here the datasets are uploaded. The data set files should be in the form of .csv and click the upload button.



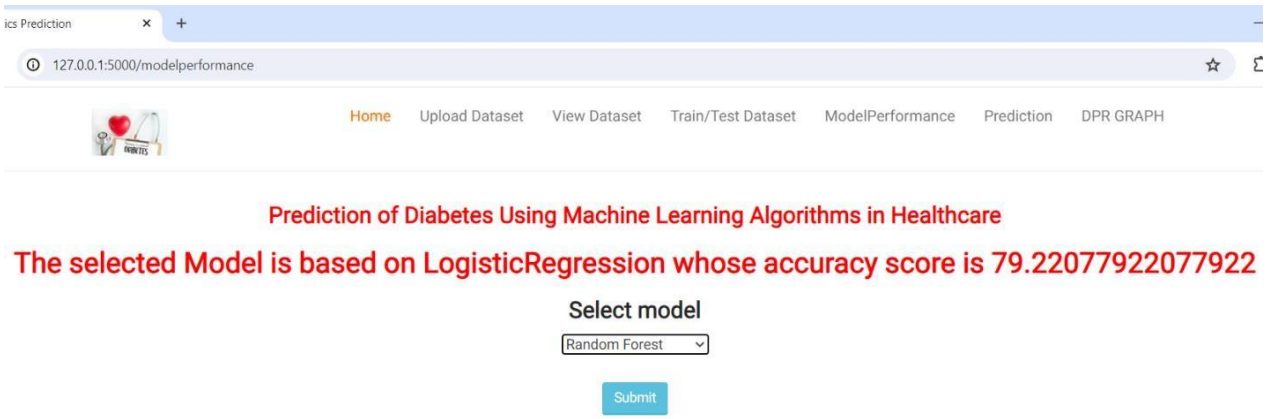
After uploading the dataset. Now select the view data set and select size and after selecting the data set size, click submit, you can see the data set you have uploaded, and you will get the values of train data length and test data length.

Now click on train/test dataset, here you select the algorithms that you have used in the program one by one and see the results of each algorithm such as Logistic Regression, Decision Tree, Random Forest, Navie Bayes, K Nearest Neighbors, Support Vector Machine.

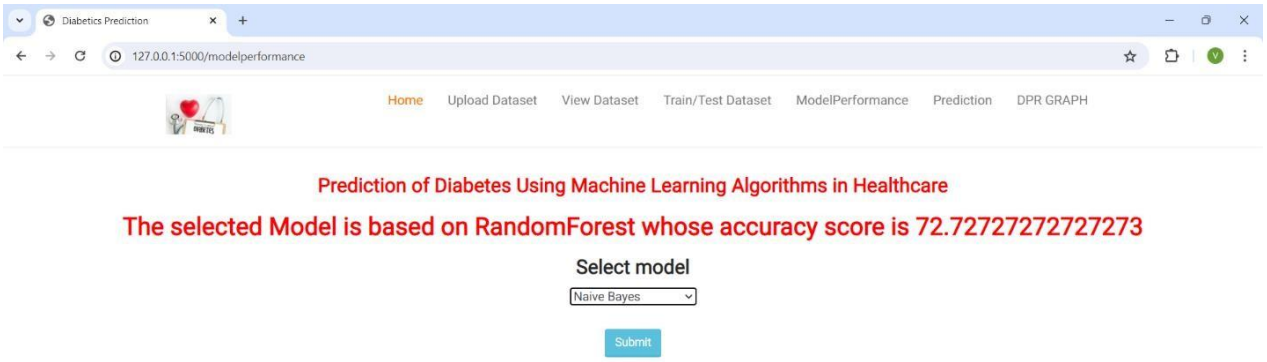




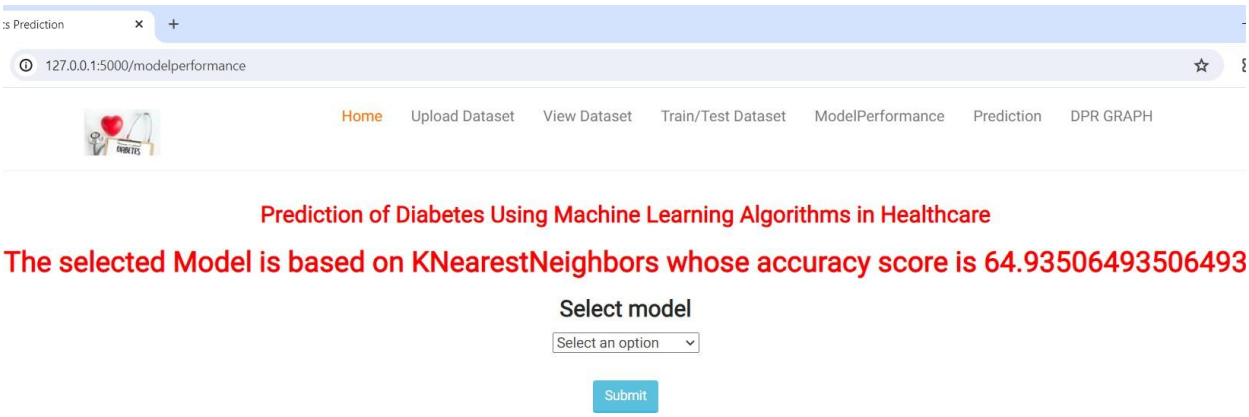
Decision Tree accuracy score is “74.02”.



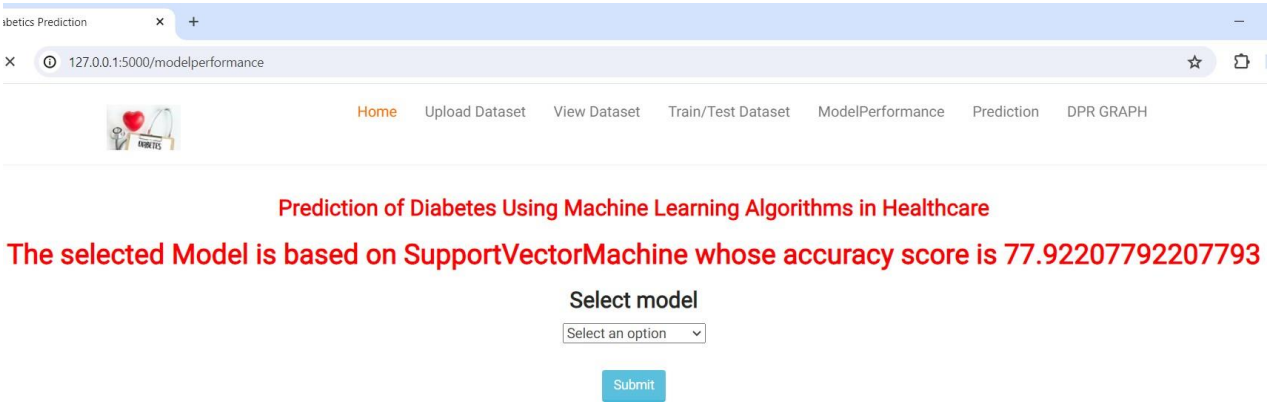
Logistic Regression accuracy score is “79.22”.



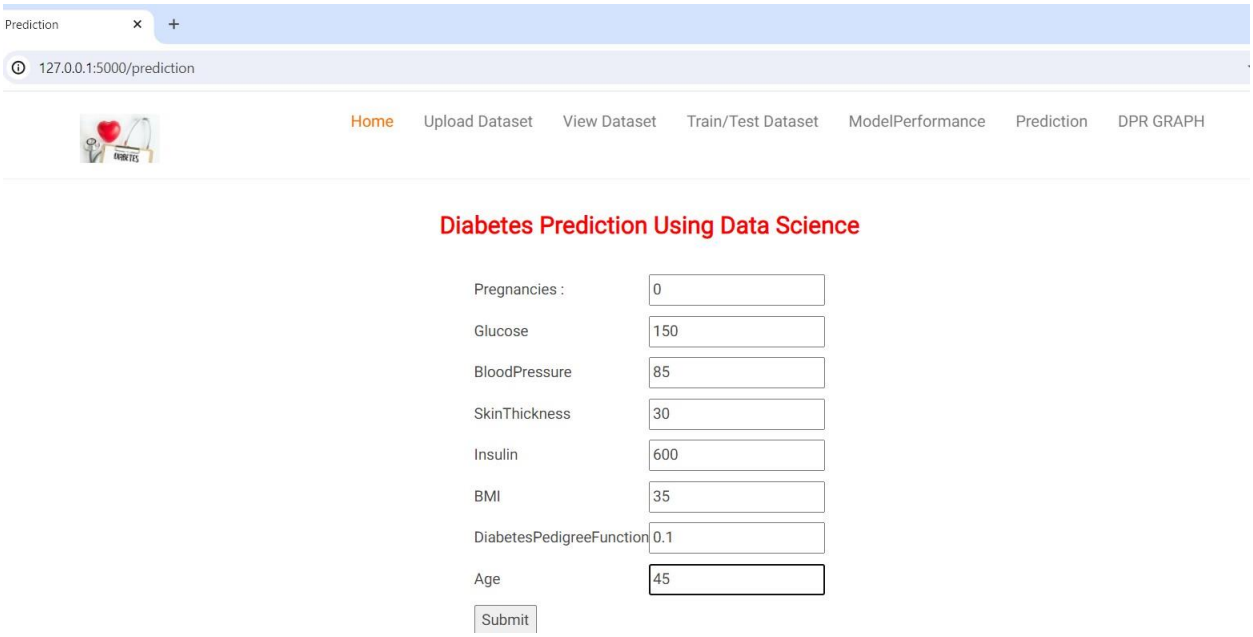
Random Forest accuracy score is “72.72”.



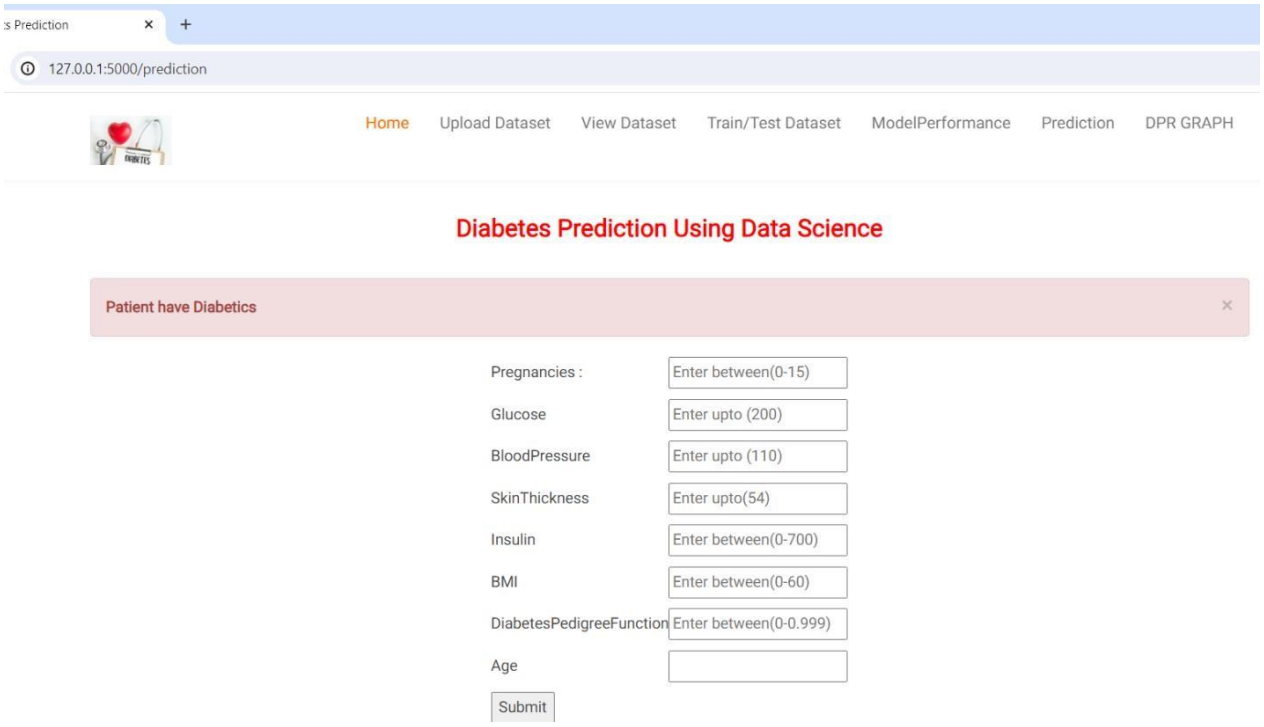
K Nearest Neighbors accuracy score is “64.93”.



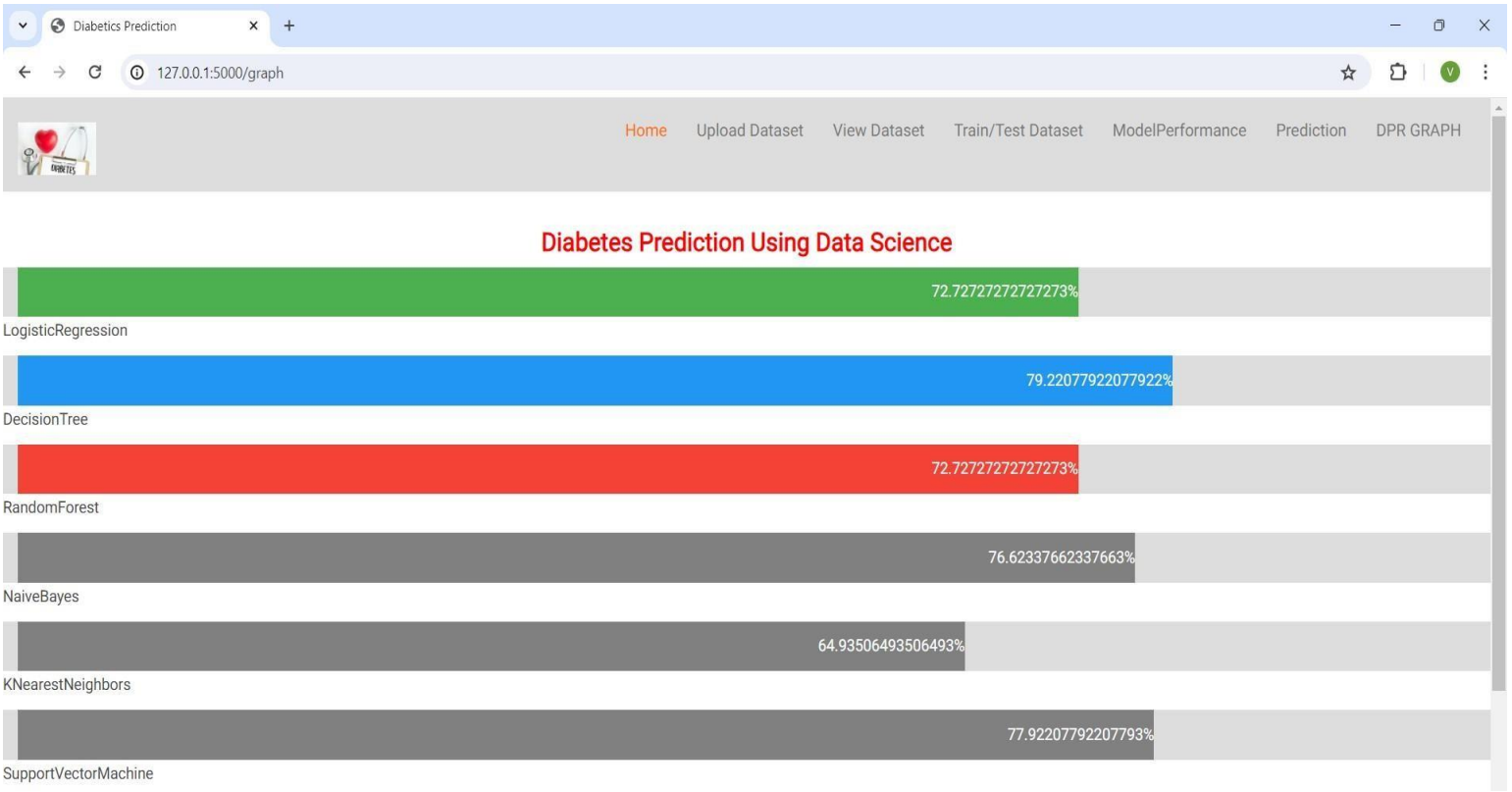
Support Vector Machine accuracy score is “77.92”



Here we have to give the information about the Pregnancies (0-15), Glucose levels (0-200), Blood Pressure levels (0-110), Skin Thickness (up to 54), Insulin Levels (0-700), BMI (up to 60), Diabetes Prediction Function (0-0.999), Age. Click on submit.

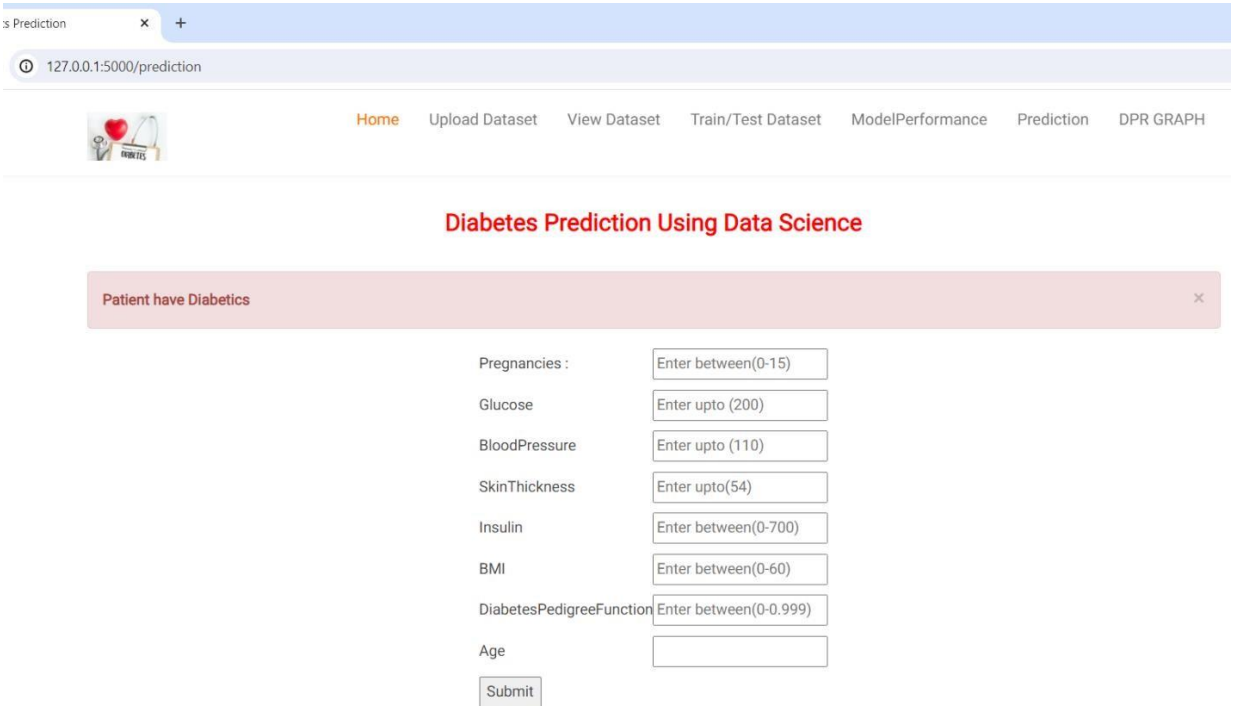


You get the result as the “patient have Diabetes”.

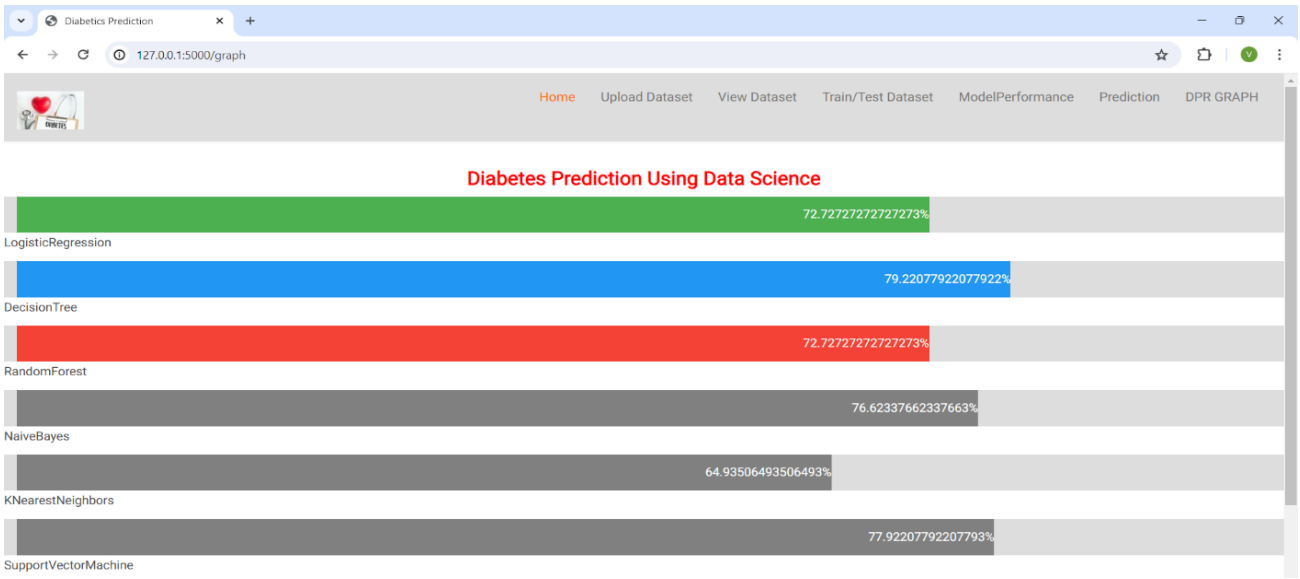


Here is the graph for the data set taken. The gives the information about which algorithm is predicting the highest accuracy for the taken data set. Form the above graph we can say Accuracy for the taken data set is “79”. Algorithm predicting the highest accuracy is “Logistic Regression”.

RESULTS WITH DISCUSSION



Here, we must give the information like pregnancies, glucose levels, blood pressure, skin thickness, insulin levels, bmi, diabetes prediction function, age and it will predict the disease. Here, the person has diabetes.



The above graph describes which machine learning algorithm gives the highest accuracy.

Accuracy for the taken data set is “79”.

The algorithm predicting the highest accuracy is “Logistic Regression”.

CONCLUSION

In our research, firstly, we have adopted several machine learning algorithms and evaluated their performances to predict the diabetes of individuals. Secondly, we have conducted several experiments and evaluated the performance of the proposal. We found that SVM outperforms the other algorithms. Finally, based on our observed results, a smart web application is developed for predicting diabetes accordingly. Any individual can submit clinical data to this web application, which can then forecast the existence or absence of diabetes. Individuals who are unsure or simply want a routine checkup may consider this application. Our model is compared with two recent studies, and the findings reveal that, depending on the dataset and the ML method used, the suggested model can offer greater accuracy ranging from 2.71% to 13.13%. While we have conducted several experiments utilizing two distinct datasets, this study still has room for additional research and development using a variety of deep learning methods.

A diabetes prediction application is a valuable tool that can help individuals assess their risk of developing diabetes. By analyzing relevant health data, such as blood glucose levels, family history, and lifestyle factors, the application can provide personalized risk scores. Users can then take preventive measures, such as adopting healthier habits or seeking medical advice, to managetheir risk effectively.

REFERENCE

[1] Belle, R. Thiagarajan, S. M. R. Soroushmehr, F. Navidi, D. A. Beard, and K. Najarian, "Big Data Analytics in Healthcare," Hindawi Publ. Corp., vol. 2015, pp. 1–16, 2015.

[2] J. Andreu-Perez, C. C. Y. Poon, R. D. Merrifield, S. T. C. Wong, and G.-Z. Yang, "Big Data for Health," IEEE J. Biomed. Heal. Informatics, vol. 19, no. 4, pp. 1193–1208, 2015

[3] E. Ahmed et al., "The role of big data analytics in Internet of Things," Computer Networks, vol. 129, no. December, pp. 459–471, 2017M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang, "Disease Prediction by Machine Learning over Big Data from Healthcare Communities," IEEE Access, vol. 5, no. c, pp. 8869–8879, 2017.

[4] L. Zhou, S. Pan, J. Wang, and A. V. Vasilakos, "Machine learning on big data: Opportunities and challenges," Neurocomputing, vol. 237, pp. 350–361, May 2017.

[5] J. B. Heaton, N. G. Polson, and J. H. Witte, "Deep learning for finance: deep portfolios," Appl. Stoch. Model. Bus. Ind., vol. 33, no. 1, pp. 3–12, Jan. 2017.

[6] Iyer A., Jeyalatha S. Sumbaly R.,"Diagnosis of diabetes using classification mining techniques".

[7] V. Krishnapraseeda, M. S. Geetha Devasena, V. Venkatesh and A. Kousalya, "Predictive Analytics on Diabetes Data using Machine Learning Techniques,".

[8] Vrushali B., and Rakhi W., "Review on Prediction of Diabetes using Data Mining Technique", International Journal of Research and Scientific Innovation (IJRSI), Volume IV, Issue IA, pp. 43-46, January 2017.

[9] H. Syed and T. Khan, "Machine Learning-Based Application for Predicting Risk of Type 2 Diabetes Mellitus (T2DM) in Saudi Arabia: A Retrospective Cross-Sectional Study,", doi: 10.1109/ACCESS.2020.3035026 in IEEE Access, vol. 8, pp. 199539-199561, 2020.