Diabetes Prediction using Data Mining

Sharvani M S[1], Siddharth Warad[2], Fayaz M[3], Darshan N S[4]

[1-4]*Student, Dept. of ECE Engineering, Dr. Ambedkar Institute of Technology, Karnataka, India*

---***---

Abstract - *The goal of the data mining methodology is to think data from a data set and change it into a reasonable structure for further use. Our examination concentrates on this part of Medical conclusion learning design through the gathered data of diabetes and to create smart therapeutic choice emotionally supportive network to help the physicians. The primary target of this examination is to assemble Intelligent Diabetes Disease Prediction System that gives analysis of diabetes malady utilizing diabetes patient's database. In this system, we propose the use of algorithms like Bayesian and K-NN (K-Nearest Neighbour) to apply on diabetes patient's database and analyze them by taking various attributes of diabetes for prediction of diabetes disease.*

Key Words: **Diabetes, Data Mining, Python, Anaconda Navigator, SVM**

1. INTRODUCTION

1.1 Data Mining

Data mining is the investigations of expansive datasets to separate covered up and beforehand obscure examples, connections and information that are hard to recognize with conventional measurable techniques.

The territories where data mining is connected, as of late; incorporate designing, showcasing, human services and monetary anticipating. Data mining in social insurance is also a rising field of high significance for giving what we can say is high anticipation and a more profound comprehension of restorative data.

The amount of accessibility of tremendous measure of patient's data which can be used to extricate valuable information, scientists have been utilizing data mining methods to help medicinal services experts in analysis of ailments.

In the usually higher part of the papers, the diabetes forecast system chips away at a little dataset, however our point is to deal with expansive dataset. The quantity of medicinal test required may influence to execution of system in this way we additionally concentrate on diminishing the therapeutic test. It relies on upon which parameter or quality is taken in the system for foreseeing diabetes.

Our expectation system will take a shot at a bigger dataset and number of therapeutic testing test required will overcome. Our system utilizes two calculations which we will apply on the same dataset for anticipating diabetes.

1.2 Diabetes

The term "diabetes" is a disease that occurs when the blood glucose in the body, also called blood sugar, is too high. Blood glucose is the main source of energy and comes from the food we eat. According to doctors, diabetes occurs when a gland known as pancreas does not release a hormone called insulin in sufficient quantity. Insulin is a hormone that carries sugar from the bloodstream to various cells to be used as energy. Lack of insulin disrupts the body's natural ability to produce and use insulin accurately. As a result of this, high levels of glucose are released in urine. In the long-term, diabetes when not properly managed can lead to organ failure, cardiovascular diseases and disrupts other functions of the body.

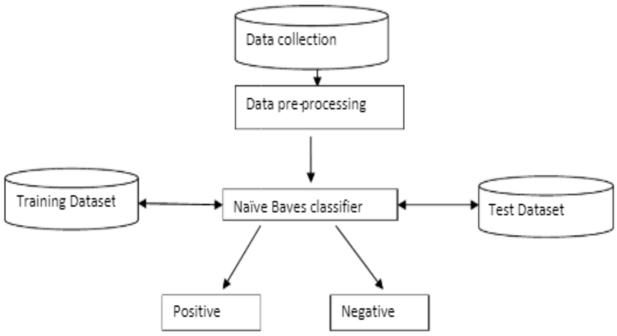


Fig-1: Classification of Diabetics and Non-diabetics

The following parameters were used for detecting and classifying the diabetes into positive and negative class, the parameters are: age, insulin, smoke cigarette, age first smoked, etc.

Functional Requirement- A functional requirement describes system should do. The functional requirement also specifies the operations and activities that a system must be able to perform. Functional Requirements should include: Descriptions of data to be entered into the system, Descriptions of work-flows performed by the system, Descriptions of system reports or other outputs. Some of the functional requirement of the proposed system includes:

i. The proposed system will provide a platform to analyze dataset for new patients.

ii. The proposed system will measure dataset for accuracy

2. LITERATURE SURVEY

Sadegh et al. have proposed, this system that comes under the category of data mining. The system performs data mining on patterns and correlation to predict the economic events. This system utilizes K-Nearest Neighbour for estimating values that will maintain a strategic distance from financial distress and bankruptcy.

In the current review, k-Nearest Neighbour characterization technique has been examined for economic estimating. Lately, after the situation of worldwide financial emergency, the quantity of bankrupt organizations has risen. Since organizations' financial distress is the principal phase of bankruptcy, utilizing financial proportions for anticipating financial distress have pulled in a lot of consideration of the scholastics and also economic and financial institutions.

Mohamed EL Kourdi et al. have proposed this system in which Naive Bayes (NB) which is a factual machine learning algorithm is utilized to order Arabic web documents. This system utilizes K-Nearest Neighbour for estimating values that will maintain a strategic distance from financial distress and bankruptcy. In the current review of k-Nearest Neighbour characterization technique, have been examined for economic estimating. Lately, after the situation of worldwide financial emergency, the quantity of bankrupt organizations has risen. Since organizations' financial distress is the principal phase of bankruptcy, utilizing financial proportions for anticipating financial distress have pulled in a lot of consideration of the scholastics and economic and financial institutions.

Kevin Beyer et al. have proposed this system that tries to explain what happens when dimensionality increases. While dimensionality builds the separation between the nearest and the most distant point gets to be distinctly irrelevant and in this manner the execution is influenced. This may prompt to wrong forecast. Additionally, increment in measurement ought to be dismissed however much as could be expected.

In example acknowledgment, the k-Nearest Neighbours algorithm (or k-NN for short) is a non-parametric strategy utilized for order and relapse. The K-NN algorithm is among the easiest of all machine learning algorithms. Both for order and relapse, it can be valuable to relegate weight to the commitments of the neighbours, so that the closer neighbours contribute more to the normal than the more far off ones.

Marius et al. have proposed this system that implements rather fast generating nearest neighbour and appropriate algorithm configuration. In this system, this system they have built up a framework that choses a fitting algorithm in view of the data bolstered which rather creates the fastest nearest neighbour. This algorithm is selected based on dimension of the data.

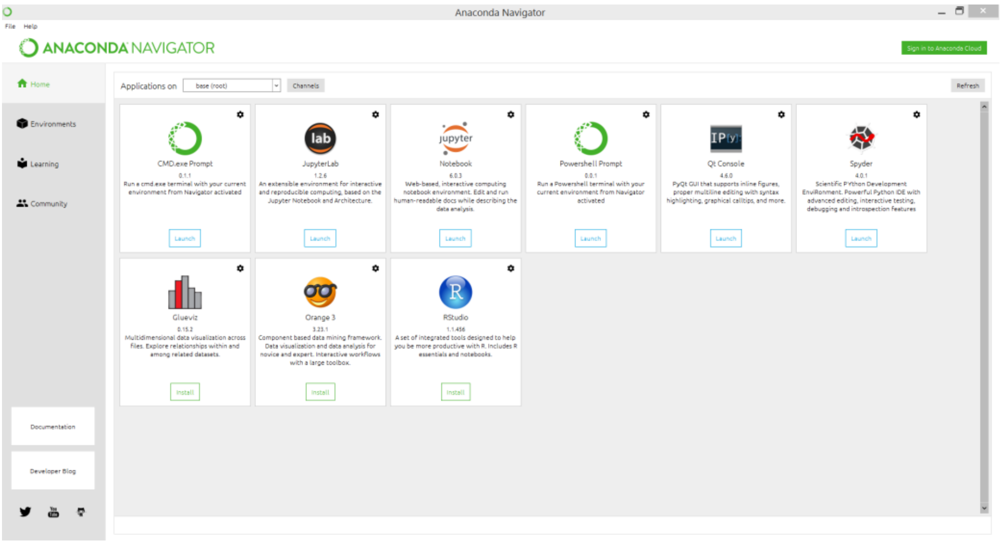
3. PYTHON AND JUPYTER NOTEBOOK

Python is an interpreted language, which means you just type in plain text to an interpreter, and things happen. There is no compilation step, as in languages such as C or FORTRAN. To start up the Python interpreter, just type python from the command line on climate. You'll get a prompt, and can start typing in python commands. Try typing in 2.5*3+5. and see what happens. To exit the Python interpreter, type ctrl-d.

Eventually, you'll probably want to put your Python programs, or at least your function definitions, in a file you create and edit with a text editor, and then load it into Python later. This saves you having to re-type everything every time you run. The standard Unix implementation of Python provides an integrated development environment called idle, which bundles a Python interpreter window with a Pythonaware text editor.

To start up idle, log in to the server from an xterm and type IDLE. You will get a Python shell window, which is an ordinary Python interpreter except that it allows some limited editing capabilities. The real power of idle comes from the use of the integrated editor. To get an editor window for a new file, just choose New Window from the File menu on the Python Shell window. If you want to work with an existing file instead, just choose Open from the File menu, and pick the file you want from the resulting dialog box. You can type text into the editor window, and cut and paste in a fashion that will probably be familiar to most computer users. You can have as many editor windows open as you want, and cut and paste between them. When you are done with your changes, select Save or Save as from the File menu of the editor window, and respond to the resulting dialog box as necessary. Once you have saved a file, you can run it by selecting Run module from the Run menu. You can actually use the integrated editor to edit just about any text file, but it has features that make it especially useful for Python files. For example, it colorizes Python key words, automatically indents in a sensible way, and provides popup advice windows that help you remember how various Python functions are used. As an exercise at this point, you should try creating and saving a short note (e.g. a letter of gratitude to your TA), and then try opening it up again in a new editor window. To exit from idle just choose Exit from the File menu of any window. An especially useful feature of the idle editor is that it allows you to execute the Python script you are working on without leaving the window. To do this, just choose Run Script from the Edit menu of the editor window. Then the script will run in the Python shell window. When the script is done running, you can type additional Python commands into the shell window, to check the values of various quantities and so forth. IDLE has various other powerful features, including debugging support. You can manage without these, but you should feel free to learn about and experiment with them as you go along. Once you have written a working Python script and saved it,say, as MyScript.py, you can run it from the command line by typing python MyScript.py. There is no need to start up IDLE just to run a script.

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than C or Pascal.



CMD.exe Prompt
0.1.1
Run a cmd.exe terminal with your current environment from Navigator activated

JupyterLab
1.2.6
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook
6.0.3
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Powershell Prompt
0.0.1
Run a Powershell terminal with your current environment from Navigator activated

Qt Console
4.6.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Spyder
4.0.1
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Glueviz
0.15.2
Multidimensional data visualization across files. Explore relationships within and among related datasets.

Orange 3
3.23.1
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Fig-2: Home page of Anaconda Navigator

4. COMPONENTS

The Jupyter Notebook combines three components:

- **The notebook web application**: An interactive web application for writing and running code interactively and authoring notebook documents.
- **Kernels**: Separate processes started by the notebook web application that runs users' code in a given language and returns output back to the notebook web application. The kernel also handles things like computations for interactive widgets, tab completion and introspection.
- **Notebook documents**: Self-contained documents that contain a representation of all content visible in the notebook web application, including inputs and outputs of the computations, narrative text, equations, images, and rich media representations of objects. Each notebook document has its own kernel.

5. ARCHITECTURAL DESIGN

System architecture is a conceptual model that defines the structure and behaviour of the system. It comprises of the system components and the relationship describing how they work together to implement the overall system.

System design is the process of the defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements .Systems design could be seen as the application of systems theory to product development. Object- oriented analysis and methods are becoming the most widely used methods for computer systems design. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user. The UML has become the standard language in object oriented analysis and design.

5.1 Dataflow Diagram

A dataflow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing. A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored.

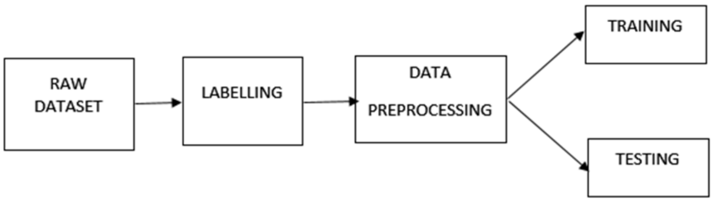


Fig-3: DFD Level Zero

6. RESULTS AND TESTING

Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate one or more properties of interest.

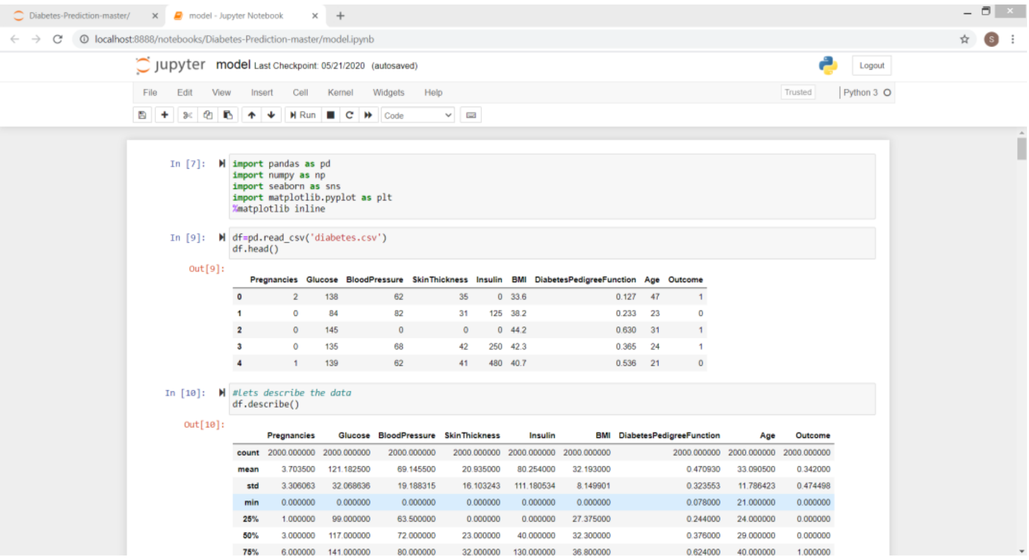
Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools. Some prefer saying Software testing as a White Box and Black Box Testing.

In our system we have done manual testing as well as stress testing to check the breakpoint of the network. The manual testing was done using selenium software while stress testing was done manually with the help of hundreds of nodes that were rented from an online server.

The first Testing was done in the first module i.e. Data Pre-processing which is to ensure that the data set does not contain any missing value or unknown value. The original CSV file is taken as input and data cleansing is performed successfully.

The second and third testing is done in second module i.e. Feature Extraction to reduce the dimensionality of dataset .The pre-processed CSV file is taken and PCA and random forest are successfully applied separately to get the reduced feature dataset.

This chapter gives the outline of all testing methods that are carried out to get a bug free system. Quality can be achieved by testing the product using different techniques at different phases of the project development. The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components sub-assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.


Diabetes-Prediction-master/
model - Jupyter Notebook
localhost:8888/notebooks/Diabetes-Prediction-master/model.ipynb
jupyter model Last Checkpoint: 05/21/2020 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [7]:  import pandas as pd
         import numpy as np
         import seaborn as sns
         import matplotlib.pyplot as plt
         %matplotlib inline
```

```
In [9]:  df=pd.read_csv('diabetes.csv')
         df.head()
```

Out[9]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	2	138	62	35	0	33.6	0.127	47	1
1	0	84	82	31	125	38.2	0.233	23	0
2	0	145	0	0	0	44.2	0.630	31	1
3	0	135	68	42	250	42.3	0.365	24	1
4	1	139	62	41	480	40.7	0.536	21	0

```
In [10]: #lets describe the data
         df.describe()
```

Out[10]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000
mean	3.703500	121.182500	69.145500	20.935000	80.254000	32.193000	0.470930	33.090500	0.342000
std	3.306063	32.068636	19.188315	16.103243	111.180534	8.149901	0.323553	11.786423	0.474498
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	63.500000	0.000000	0.000000	27.375000	0.244000	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	40.000000	32.300000	0.376000	29.000000	0.000000
75%	6.000000	141.000000	80.000000	32.000000	130.000000	36.800000	0.624000	40.000000	1.000000

Fig-4: Pre-processing of given data

6.1 Testing Process

After designing phase there is the coding phase. In this phase, every module identified and specified in the design document is independently Coded and Unit tested. Unit testing (or module testing) is the testing of different units or modules of a system. In this phase, the physical design of the system is converted into the logical programming language.

6.2 Testing Objectives

The coding is done in java before starting of the coding, we have tried to follow some coding standards and Guidelines.

The coding standards are: -

- Naming standards for the Classes and variables etc.
- Screen design standards.
- Validation and checks that need to be implemented.

The Guidelines are: -

- Code should be well documented.
- Coding style should be simple.
- Length of function should be short.

6.3 Levels of Testing

Unit Testing- In this, the programs that made up the system were tested. This is also called as program testing. This level of testing focuses on the modules, independently of one another. The purpose of unit testing is to determine the correct working of the individual modules. For unit testing, we first adopted the code testing strategy, which examined the logic of program. During the development process itself all the syntax errors etc. got rooted out. For this we developed test case that results in executing every instruction in the program or module i.e. every path through program was tested. (Test cases are data chosen at random to check every possible branch after all the loops.).Unit testing involves a precise definition of test cases, testing criteria, and management of test cases.

User Input- In User interface the data entry is done through GUI and tested. Each element is tested for valid range and invalid range of data.

Error Handling- In this system we have tried to handle all the errors that are occurred while running the GUI forms. The common errors we saw are reading the empty record and displaying a compiler message, etc.

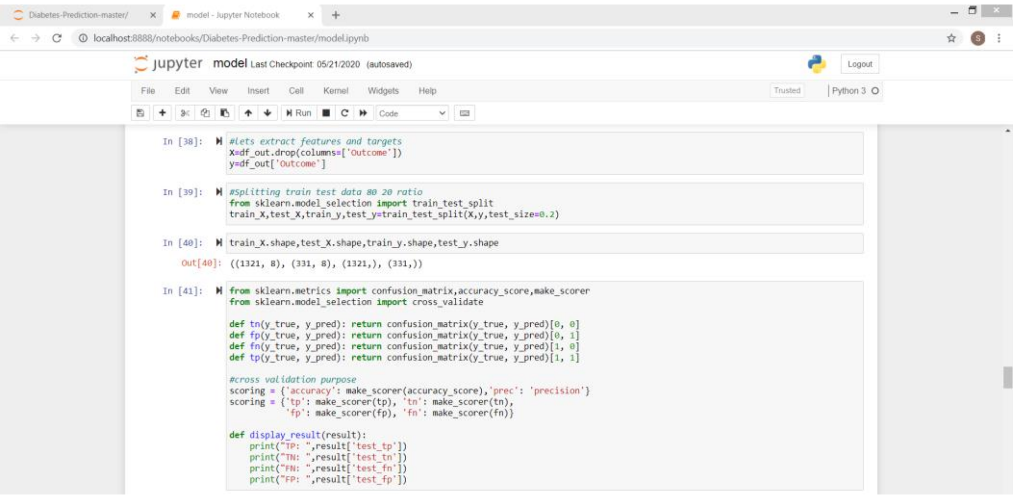
System Testing- Once we are satisfied that all the modules work well in themselves and there are no problems, we do in to how the system will work or perform once all the modules are put together. The main objective is to find discrepancies between the system and its original objective, current specifications, and system documentation. Analysts try to find molds that have been designed with different specifications, which could cause incompatibility. At this stage the system is used experimentally to ensure that all the requirements of the user are fulfilled. At this point of the testing takes place at different levels so as to ensure that the system is free from failure. Testing is vital to success of the system. System testing makes a logical assumption that whether all parts of the system are correct. Initially the system was given to the user for entry validation was provided at each and every stage, So that the user is not allowed to enter unrelated data. The training is given to user about how to make an entry. While implementing the system it was observed that the user was initially resisting the change, however the system being the need of the hour and user friendly, the fear was overcome. Entering live data of the past months records was little tedious, prior to the actual day to day transaction. The best test made on the system was whether it produces the correct outputs. All the outputs were checked out and were found to be correct. Feedback sessions were conducted and the suggested changes given by the user were made before the acceptance test. Finally the system is being accepted and made to run with live data. System tests are designed to validate a fully developed system with a view to assuring that it meets its requirements. There are three main kinds of system testing:

- Alpha Testing.
- Beta Testing.
- Acceptance Testing.

Alpha Testing: This refers to the system testing that is carried out by the test team with the organization.

Beta Testing: This refers to the system testing that is performed by a select group of friendly customers.

Acceptance Testing: This refers to the system testing that is performed by the customer to determine whether or not to accept the delivery of the system.



Diabetes-Prediction-master/
model - Jupyter Notebook
localhost:8888/notebooks/Diabetes-Prediction-master/model.ipynb
jupyter
model Last Checkpoint: 05/21/2020 (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

```
In [38]:  #lets extract features and targets
          X=df_out.drop(columns=['Outcome'])
          y=df_out['Outcome']

In [39]:  #Splitting train test data 80 20 ratio
          from sklearn.model_selection import train_test_split
          train_X,test_X,train_y,test_y=train_test_split(X,y,test_size=0.2)

In [40]:  train_X.shape,test_X.shape,train_y.shape,test_y.shape

Out[40]:  ((1321, 8), (331, 8), (1321,), (331,))

In [41]:  from sklearn.metrics import confusion_matrix,accuracy_score,make_scorer
          from sklearn.model_selection import cross_validate

          def tn(y_true, y_pred): return confusion_matrix(y_true, y_pred)[0, 0]
          def fp(y_true, y_pred): return confusion_matrix(y_true, y_pred)[0, 1]
          def fn(y_true, y_pred): return confusion_matrix(y_true, y_pred)[1, 0]
          def tp(y_true, y_pred): return confusion_matrix(y_true, y_pred)[1, 1]

          #cross validation purpose
          scoring = {'accuracy': make_scorer(accuracy_score),'prec': 'precision'}
          scoring = {'tp': make_scorer(tp), 'tn': make_scorer(tn),
                     'fp': make_scorer(fp), 'fn': make_scorer(fn)}

          def display_result(result):
              print("TP: ",result['test_tp'])
              print("TN: ",result['test_tn'])
              print("FN: ",result['test_fn'])
              print("FP: ",result['test_fp'])
```

Fig-5: Testing of the given datasets

7. FUTURE SCOPE AND CONCLUSIONS

7.1 Future Scope

The proposed system can be developed in many different directions which have vast scope for improvements in the system.

These include:

1. Increase the accuracy of the algorithms.

2. Improvising the algorithms to add more efficiency of the system and enhance its working.

3. Working on some more attributes so to tackle diabetes even more.

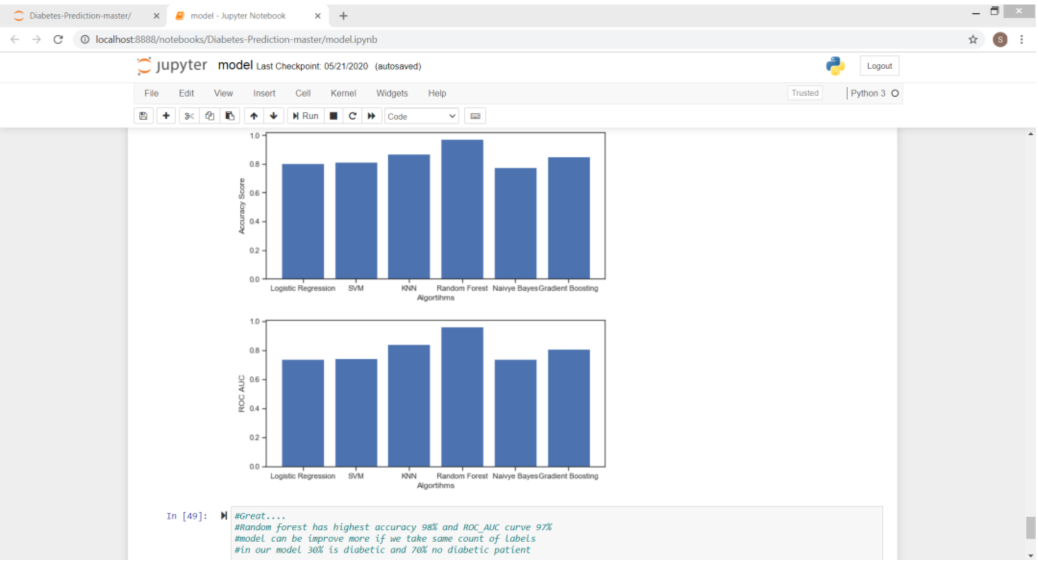
4. To make it as a complete healthcare diagnosis system to be used in hospitals.

Future work should be done on improving the accuracy of the prediction by increasing the level of training data. Its performance can be further improved by identifying and incorporating various other parameters and increasing size of training.

7.2 Conclusions

By our in-depth analysis of literature survey, we acknowledged that the prediction done earlier did not use a large dataset. A large dataset ensures better prediction. Also what it lacks is recommendation system. When we predict we will give some recommendation to the patient on how to control or prevent diabetes in case of minor signs of diabetes. The recommendations would be such, that when followed it will help the patient. Thus we will build up a system which will anticipate diabetic patient with the assistance of the Knowledge base which we have of dataset of around 2000 diabetes patients and furthermore to give suggestions on the premise of the nearness of levels of diabetes patients. Prediction will be done with the help of two algorithms Naïve Bayes and K-Nearest Neighbour and also we will compare which algorithm gives better accuracy on the basis of their performance factors. This system which will be developed can be used in HealthCare Industry for Medical Check of diabetes patients.

This application would be a tremendous asset for doctors who can have structured specific and invaluable information about their patients / others so that they can ensure that their diagnosis or inferences are correct and professional. Finally, the huge appreciations received from the doctors on having such software prove that in a place like, where diseases are on the rise, such applications should be developed to cover the entire state. The common person stands to benefit from doctors having such a tool so that he/she can be better knowledgeable as far as personal health and wellbeing is concerned



```
In [49]:  #Great....
          #Random forest has highest accuracy 98% and ROC_AUC curve 97%
          #model can be improve more if we take same count of labels
          #in our model 30% is diabetic and 70% no diabetic patient
```

Fig-6: Conclusion formed by the given datasets expressed by bar graphs in terms of accuracy and ROC & AUC

8. REFERENCES

[1] Y. Cai, D. Ji,D. Cai, "A K-NN Research Paper Classification Method Based on Shared Nearest Neighbor", Proceedings of NTCIR-8 Workshop Meeting, 2010.

[2] I. Rish, "An empirical study of the naive Bayes classifier", T.J. Watson Research Center, 2001.

[3] M. Elkourdi, A. Bensaid, T. Rachidi, "Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm", Alakhawayn University, 2001.

[4] L. Wang, L. Khan and B. Thuraisingham,"An Effective Evidence Theory based on nearest Neighbor (KNN) classification", IEEE International Conference, 2008.

[5] M.Muja, David G.Lowe, "Fast Approximate Nearest Neighbors With Automatic Algorithm Configuration", University of British Columbia.

[6] SadeghB.Imandoust, M.Bolandraftar, "Application of KNearestNeighbor (KNN) Approach for Predicting Economic Events: Theoretical Background",International Journal of Engineering Research and Applications, Vol. 3, 2013.

[7] Tina R.Patil, S. S. Sherekar, "Performance Analysis of Naïve Bayes and J48 Classification Algorithm for Data Classification", International Journal Of Computer Science And Applications, 2013.

[8] Z.Song, N.Roussopoulos, "K-Nearest Neighbor Search for Moving Query Point",T.J. Watson Research Center.

[9] Harry Zhang, "The Optimality of Naive Bayes", Faculty of Computer Science at University of New Brunswick.

[10] K Beyer, J Goldstein,R Ramakrishnan and U Shaft, "When is 'Nearest neighbor' Meaningful?" 2014. [11] Y Cai, D Ji, Dong-feng Cai, "A KNN Research Paper Classification Method Based on Shared Nearest Neighbor", at Shenyang Institute of Aeronautical Engineering.

[12] K Saxena1, Dr. Z Khan2, S Singh3," Diagnosis of Diabetes Mellitus using K Nearest Neighbor Algorithm", at Invertis University.

[13] M Panda and M Patra, "Network Intrusion Detection Using Naïve Bayes" at IJCSNS International Journal of Computer Science and Network Security, VOL7, 2007.

[14] Davis D. Lewis, "Naïve Bayes at Forty – The Independence Assumption in Information retrieval." AT&T Labs.