An Analysis of Predicting Diabetes using Machine Learning

¹Mr. Ujjwal Anand, ²Dr. Amit Sehgal, ³Mr. Shashank Tripathi, ⁴Mr. Gagandeep Singh, ⁵Mr. Rinku Sharma, ⁶Ms. Manisha ^{1,4,6} ^{1,4,6} ^{1,4,6} ⁵Project Engineer, ⁵Research Associate,

1,4,0 Innovatores, ⁵Project Engineer, ⁵Research Associate, Department of Electronics & Communication Engineering, G L Bajaj Innovation Center, Greater Noida, UP, India.

²Professor, Department of Electronics & Communication Engineering, GLBITM, Greater Noida, UP, India.

Email: ¹*ujjwalanand1997@gmail.com,* ²*amit.sehgal@glbitm.org,* ³*shashanktripathi030@gmail.com* **DOI:**

Abstract

Diabetes comes under chronic disease, in which cells are not able to use blood sugar (glucose) efficient enough for energy. This condition arrives when the cells become nonresponsive to insulin and the blood sugar increases gradually. The known types of diabetes are, Type1, Type 2 and Type 3. Type 1 and type 2 diabetes are in hyperglycemia category (caused by increase in blood sugar), while Type 3 diabetes (Alzheimer's disease) which is caused by resistance to insulin in the brain. Prediction of preliminary stage diabetes is very important as it becomes worse in next stages. This Prediction can be done using machine learning classification models, which are more widely being used for other medical purposes. To predict, if a person is diabetic we need data about insulin, blood pressure, skin thickness and glucose. This data will be fitted in classification models of machine learning with a target vector of conclusion. This will prepare a model that can predict if a certain patient is a diabetic or not. We have implemented many classification models on the data. We have also used neural networks to serve the same purpose.

Keywords: Diabetes diagnosis, Medical ontology.

INTRODUCTION

Machine learning is one of the technique to withstand computational capacity of a computer, where a computer is programmed with the ability to learn and improve its performance on specific tasks. In Short, machine learning is all about analyzing data, extracting information which is used to make predictions, check whether the prediction was correct, and if in correct, learning from that to make a more correct prediction in the future.

Machine learning is being used in many parts of our life. Nowadays, it is being much implemented in the Medical diagnostics. Medical diagnostics are a category of medical exams completed to become aware of infections, situations and sicknesses. These days, system mastering is gambling a key issue in reaching automation in clinical diagnostics.

Diagnosis via machine learning works while the condition may be reduced to a classification venture on physiological facts, in areas where we presently rely upon the clinician in an effort to visually perceive patterns that indicate the presence or form of the condition.

Classification is a technique to solve a problem where the output that we want is a category, such as "red" or "blue" or



"disease" "no and disease"[1]. А classification model makes computation to come on conclusions using observed values or labelled data. Given one or more inputs to a classification model, it can predict the value of one or more outcomes. In case of Medical Diagnostics, the output we want will be either а "disease" or "nodisease", or inmore deep class ification, which type of a certain disease is it, for example to predict which type of diabetes is it, the output we desire should be, "type 1" or "type2"[2].

The data is the most important part in machine learning, and no prediction model can be implemented without data[3].

The data can be a table of multiple records of patients, or the images of different patient's diseased parts. For example, if we want to detect tuberculosis, then we need to train our model with chest x-rays of tuberculosis patients, but if we want to detect if there is a presence of diabetes in a patient, then we will need a feature matrix containing the features of patients, and another list of conclusion if the patient is having the Diabetes or not.

Blood Pressure	Insulin	Glucose	Skin Thickness	
72	85	148	35	
66	94	89	23	
84	230	118	47	
88	235	126	41	

Table: 1. Feature	e Matrix for patients ca	ame for Diabetes diagnosis
-------------------	--------------------------	----------------------------

Then there will be another list containing the conclusion of diabetes in patients.

The machine learning model will be trained according to the feature matrix and conclusion list.

Overfitting in Machine learning models

In this condition, the model learns most of the detail and noise available in the training data to that extent, that it negatively influences the presentation of the model on new data. Actually, the noise or random variations in the exercise data is picked up and learned as concepts by the model[4]. This comes on a problem that we need to tackle in every machine learning model is that when the new data will come for prediction, it will show wrong result.

This problem of overfitting, will become worse in medical diagnostics. As in other predictionsitwouldmaypredictwrongandinc uralossofmoneyorelse, butincaseof medical diagnostics it would be a matter of life and death of a patient, i.e. if a patient is being diagnosed for diabetes and model made a prediction "negative", then the patient's health may lose in further days and this might make his/her health worse. So, we need to be very careful in avoid overfitting.

Data Pre-processing

Data pre-processing means altering raw information into a comprehensible format. Real data is mostly imperfect, in consistent and complicated. During the capturing of data there are certain features that are not required. During the pre-processing step all such features are removed. Preprocessing steps:

Data Reducing: Data reduction means selecting only desired features. We have

used Back Prorogation algorithm to determine the features which affect the data most. We found out that Glucose level, Insulin level, Skin thickness and Blood Pressure affect the data most.

Data Cleaning: Data cleaning has lot of steps such as filling missing values, normalize the data and resolving in consistency.

Cases in Data Cleaning:

CASE 1 (Using mean to handle missing values):

Missing data is one of the common problems in data which can actually have a significant effect on the conclusions that can be drawn from the **data**. So, it is important to fill all missing values. Here we are using mean to fill all missing values and mean is calculated for each column individually. The results after replacing the missing data were astonishing, with only slight changes in data we were able to get a high accuracy. CASE 2 (by replacing missing values with custom values):

In our case, we have taken avg. Values of all features of a healthy person and a diabetic

person.Thevaluewegetisthenusedtofillallth emissingvaluesinthedataset. In our case Skin thickness was missing for many rows. We replaced it with the avg. values in accordance with whether he is healthy or diabetic.

	\mathbf{j}	0	
	Skin	BP	
	Thickness		
Diabetic	30	130	
Non-	22	120	
diabetic			

Table: 2. custom values for missing data

Classification Models implemented

Decision Tree: Decision tree algorithm is a supervised earning algorithm that can be used for both regression and classification problems[5].Decision tree is a rule based algorithm that creates its own rules at the time of training. Decision tree tries to solve a problem by tree representation. When we used decision tree with case 1(i.e. replacing missing values by mean) the accuracy was not very great. It was low due to fact that it was replacing every missing value with mean which is not a good idea.

In second case where missing values were replaced by threshold of health and diabetic, the accuracy was pretty high than first time. It was because of handing missing values in a better way.

Random Forest Classifier: Like decision tree it is also supervised learning algorithm. Random forest builds multiple decision tree and uses them to give more accurate results. Higher number of trees in random forest can lead to more accurate results.

Random forest produces lightly better result than decision tree for case1.However, it was also not in acceptable state due to low accuracy. There was a slight improvement in accuracy when the random forest was used in case



There was slight (2%) increase in accuracy. It is because random forest uses number of decision trees to predict are sult.

K-Nearest Neighbor (KNN): KNN algorithm uses feature similarity to find its neighbors. During the training period it tries to distribute the data into different classes. When a new data arrives, the algorithm finds features in the data and then finds its closest neighbor or class. In this model we had only 2 classes- diabetic or non-diabetic. The closest neighbor at the end is calculated using distance vector.

KNN didn't do a good job in case 1 when the data was replaced with mean values. Its accuracy was lower than random and decision.

KNN did better in case 2. It was able to find features very accurately. However, its accuracy was little less than random forest classifier.

Support Vector Machine: SVM is mostly used for classification but it can used for both regression and classification problems. In this classification model, we plot each data point in n-dimensional space consisting of each feature at a particular data point. We then find a plane which can separate classes.

The main important aspect in svm is kernel, which decides the type of classification. There are 3 main types of kernel in svm, namely, Linear, poly and rbf.

Linear kernel is for linear classification of data, which is not much helpful in case of sparse data.

Polykernel is for polynomial classification, whereas, rbf (radial basis function) is the most popular svm kernel.

Weused"rbf"kernelandSVMdidn'tdogoodi

nbothcases.Thekernelfunctionisa measure of similarity between two sets of features. Hence, SVM is not an appropriate choice.

Naïve Bayes: Naïve Bayes uses conditional probability to predict the class of unknown data. Conditional probability[**8**] is the probability that something will happen, taken as something has occurred already[6]. $\mathbb{Z}(|) = \mathbb{Z}(|) * \mathbb{Z}()/\mathbb{Z}()$ Where

P (A) is the probability of hypothesis H being true. This is known as the prior probability.

P (B) is the probability of the evidence (regardless of the hypothesis). P (B|A) is the probability of the evidence given that hypothesis is true.

P (A|B) is the probability of the hypothesis given that the evidence is there.

In both cases Naïve Bayes didn't do a good job.

Artificial Neural Networks (ANN): [9] this works as an information processing system that works in the same way as the like biological nervous systems work, such as the brain works in processing facts[5]. It comprises of a large number of highly intersected processing elements (neurons) working in unison to solve a particular problem. An ANN is arranged in single for a precise application, such as pattern acknowledgement or information sorting, using knowledge procedure[7].Knowledge in biological systems involves adjustments to the synaptic connections that exist between the neurons[8].

However, we experimented our data on many models, but the best result that we got is on a 5-layer neural networks, Having 3 hidden layers consisting of 15, 8 and 15 neurons. In both cases we get similar results [9].



But, we do not recommend to use neural networks for medical diagnosis classification, as there is a wide range of fluctuation in results.

RESULTS

The Accuracy for each model can be

obtained from confusion matrix of each model that we get from 768 people data, where we took 192 patients data as the test set, on which the model is tested. The results from the model and the original data are matched to see if the output is correct or not.

	0	1	Decision	
0	111	19	Tree	
1	6	56		
1	0	1	Gaussian	
0	117	13		
1	30	32		
	0		KNN	
0	117	13		
1	11	51		
	0	1	Random	
0	115	15	Forest	
1	12	50		
	0	1		
0	130	0	SVM	
1	62	0	5 V IVI	
	0	1	Artificial	
0	117	13	Neural	
1	30	32	Network	

 Table: 3. Confusion Matrix for models

This Confusion Matrix Actually shows, how many test sets out of 192 patients where correct. Through this we can calculate the accuracy of each model. In the confusion matrix, the values at [0,0] and [1,1] is correct, while values at [0,1] and [1,0] are wrong

Model Name	Accuracy
DecisionTree	86.9%
Gaussian	77.6%
KNN	87.5%
Random Forest	88.7%
SVM	67.6%
ANN	73.6%

Table: 4. Accuracy of Model



However, the accuracy is quite speaking about the model to be selected as best on the data that we used to build the model.

Simulation Setup

The model that is implemented is coded

on software **anaconda** (**python 3.6.5**), which runs on a system having a configuration of 8GB RAM, 1TB Hard Drive and core i3 generation 5processor. In Anaconda, An IDE named **Spyder** is used to run the code.

- 6 ×

	X / *	C 'Wervillgwal Downlees/Didectes president michter	• • •
blion - Crithery/Djwal/Downlood/Diabetes prediction master fored py	# x	Venezie explorer	
🗅 represent 🔲 debetes.cov - Coveloade) Dabetes predictor-menter 🖸 annuay 🗍 KM ay 🗍 D	red py 🛄 🕴 🗛	± 8 % κ	0
1 # thest#linetim combain # # Deport maps as mp % Deport maps as mp % Deport sorphotils applet as plt % Deport pards; as pd	-	Name Type Sic Viue dataset Datafrané (768, 9) Sala hickness, Jose	aua.ra,
 Bit reporting the defense. Bitaset = plinead cos(fidialieres.cos)) 		vertehle explorer Miclophysic Hop	
10 x = dataset.flor[1, 0:8].values [100010.0]		Python minute	4,
N - caracet.flor[1, 8].willes		🗂 Console 1/A 📴	= 0
11 t verture Selection		In [1]:	1
If from sklearn.feature_selection import Selectillest 15 from sklearn.feature_selection import ch12		In [1]:	
16.8 SelectKBest(ch12, k 4).fit_transform(K, y)			
17 18 # Maraging Miertag Values		20 [1]:	
10 tran skieern.preprocessing import Inputer		20 [1]:	
<pre>10 inputer - Inputer(sizing values-0 , strategy-"mean", suit-0) 11 X[:,0:5] - inputer.fit transform(X[:,0:5])</pre>		in [1]:	
22			
35 # Splatting the orthogen into the Training and reat and Teat art		10 [1]:	
14 from sklearn.cross validation import train test split 15 X train, X test, y train, y test = train test split(X, y, test size = 0.25, rend)	e state = 0)	.0 [1];	
26 27 # ficting elevatries to the Treating at		.0 [1])	
18 from sklearn.neigtbors import WeigtborsClassifier			
<pre>35 classifier = KWeighborsClassifier(n meighbors = 3,metric = 'minkewski',p=2) 58 classifier(fit(X train,y train)</pre>		[n [1]]	
11 12 # Praiting DV, had jot nonetta 13 gyped = classifier.prodit(X.test) 15 # Twing via criminan Serie 18 Free sklearm.etrics import confusion_matrix 37 on = confusion_metrics_leady jore)		<pre>in [1]: Separt execptes op : Separt execute static system as pit : Separt pendes as pi : in Description the defense : Categoryread_read_read('discretes.cov') in [2]:</pre>	
58	*	Python comple Hotavyku	
	3	and a second	1 Memory 53.94

Fig: 1. Importing Libraries and reading the input data which is in CSV format.

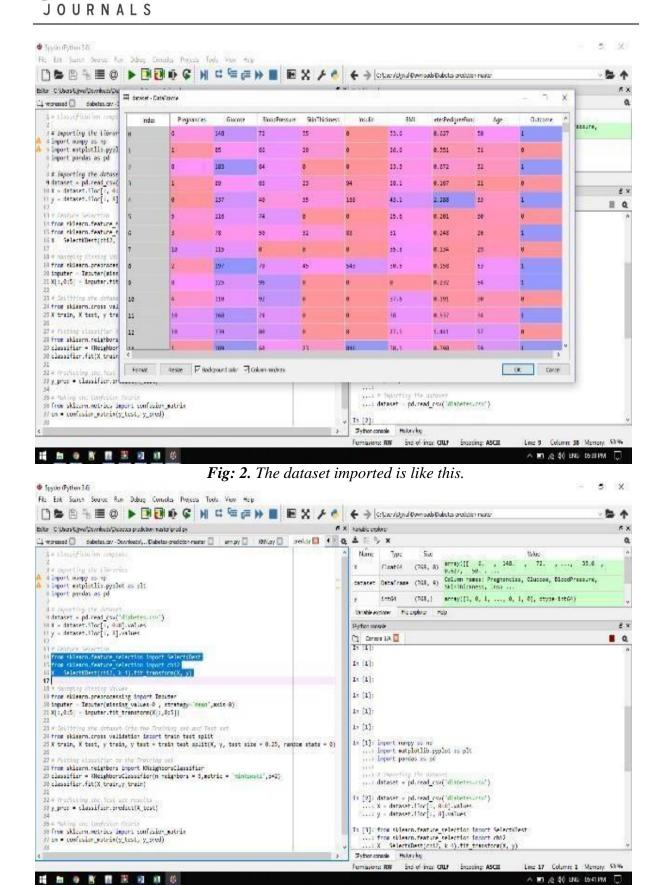


Fig: 3. Feature Selection After feature selection we got just 4 features, i.e. Glucose level, Insulin level, Skin thickness and Blood Pressure.

MAT

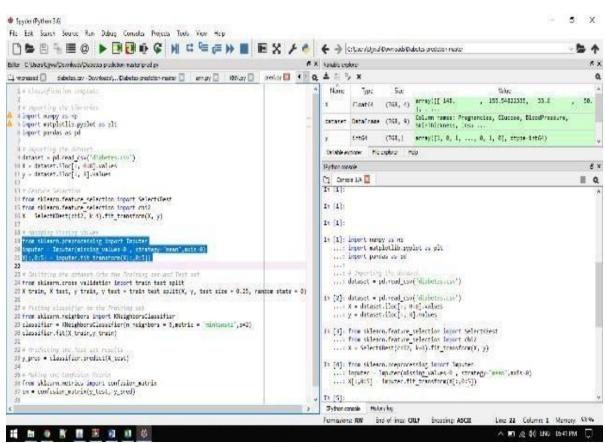
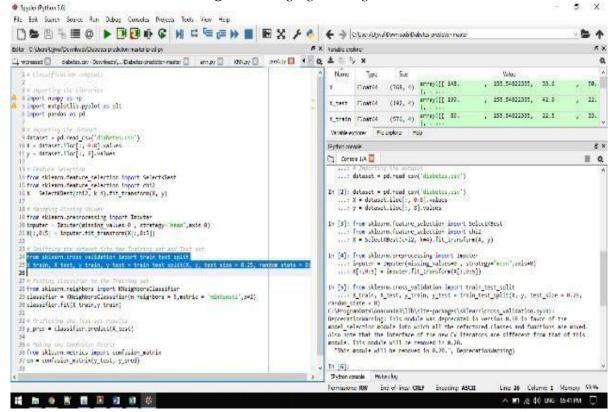


Fig: 4. Managing Missing Values





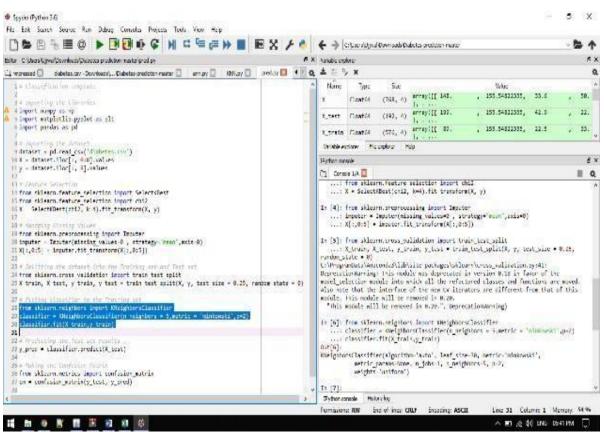


Fig: 5. Splitting the dataset in training and test set

Fig: 6. Fitting classifier to the dataset

MAT JOURNALS

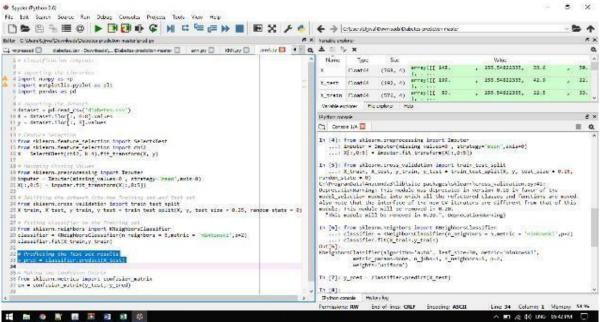


Fig: 7. Predicting the test set

is Ent Sanch Servic Run Dabus Convola Projecta Torb. View Hop 🗅 🏷 🖹 🖶 🚳 🕨 💌 💽 🐨 🚱 🔰 🛤 😂 🗡 🤇	← → C:Car /Din/Downsab Didets analytic nate	84	
ter C'BestUne/Durines/Durines problem material problem	X Apublic and ear	-	
vegressed 🔄 diabetes.cov -Downloads), Diabetes-prediction-memor 🔂 em.py 🖸 1016.ay 🔯 pred-by 🖾 🤳			9
3 = 13soff but hav interfate	* Mane Type Size Whee		
14	Y Coat64 (368, 6) array([[148, , 155,54522335, 33.6]	10.3	50.
<pre># superting dv Clearies #Deport many ss mp</pre>			
a mort satplatlis pyslot as all	X_TART (10AT(4 (19), 4) array([[197. , 155.54832335, 42.0	- 33.1	22.
a laport pardas as pd	w and a market there as arrest [[00. , 155.54012335, 22.5	68.3	23.
	X_prein Cloatid (574, 4) array(1) arr (100.0401001) array		
<pre># supporting the densest # dataset = pd.read (su('diabetes.cov'))</pre>	Winklewstower Ric crybror Holp		
18.4 = dataset.floc[:, 4:8].values	Particip modele		8
II y - dataset. flori, sj.values		-	-
	Chi Contos 1/A	1	0
12			
O Treative Startin	(1 constant and a con		- 2
If the skiest selection input selection input selections	: X[:,0:3] * imputer.fit trensform(X[:,0:5])		0
1) transmiss Selection In form :kiesen_feature_relection import_Selectifiest 15 form :kiesen_feature_relection import_chi2	<pre>:X :,0:5 * import fit transform(X):,0:5) In (5): from sklearn.tross validation import train test split</pre>		8
11 transferrer Stretten 16 foor slearn fasture selection inport Selectifiest 15 foor slearn fasture selection inport mi2 16 : Selectifiest(mi2, s.).fitz productors(X, y)	: X[:,0:3] * insuter.fit transform(X[:,0:5]) [5] (5): free allearn cross validation insort train test solit : X train, X test, y train, y test + orini test solit(X, y, test size *	0.25,	3
11 rights Starting Selection input Selectilest 15 foor skiesen feature selection input oli 15 foor skiesen feature selection input oli 15 foor skiesen feature selection (x, y) 15	<pre>: X[:,0:3] * isouter.fit transform(X[:,0:5]) In (5): from sklearn.cross validation import train test solit: X train, X test, y brain, y test * train test split(X, y, test size * random state * 0)</pre>	0.25,	0
<pre>if transfer effection input Selections if for all sens feature_relection input mi2 is for all sens feature_relection input mi2 is "selections(rel2, k 4).ftp_transform(X, y) is "selecting storing that is "selecting storing storing that is "selecting storing st</pre>	: X[:,0:5] + Messler.flt treasform(X[:,0:5]) In [5]: free sklaarn.cross valdation import train test split X train, X test, y train, y test + train test split(X, y, test size + random state = 0) C: Wheyarentbit/Amerotals/Ilb/size postage/Schartworks_valiation.sy/41:	AS-85	0
In from skiern, feature selection inport Selectifiest is from skiern, feature selection inport only is a Selectifiest(shi), k (), fit_transform(X, y) is a selecting strainer strain is from skiern, presonasting inport Insuter	<pre>: X[:,0:3] * import.fit transform(X[:,0:5]) In [5]: free sklearn cross validation import train text solit: X train, X text, y Frain, y text * train text solit(X, y, text size * random state * 0) C:\Mrsgnambti\MexecuteS\liMitic packages'svictorm\cross_validation.sy/41: DepressionHarming: This module may depression in a varian %List facer of in facer of</pre>		0
<pre>if transfer effective effection impact SelectSheet is from ekleren festure_selection impact oni2 (a SelectDheet(ehl2, k 4).ftp_transform(X, y)))))) (a subjective formation for the selection is from this encourage impact impacts is from this encourage impact is contage, mean', mole 0)</pre>	: X[:,0:5] * Mescler.flt treasform(X[:,0:5]) In [5]: free ablearm cross validation import train test split : X train, X test, y train, y test * train test split(X, y, test size * random state * 8) C:Mregarentationecondes(Ilbylic sockages(oblear/oross, validation, sy)42: Depretsionization exolution into depretation in version 8.15 in factor of the word().eltic in word() into mich all the reflected classes of formities are	e e moved.	
In from skiern, feature selection inport Selectifiest is from skiern, feature selection inport only is a Selectifiest(shi), k (), fit_transform(X, y) is a selecting strainer strain is from skiern, presonasting inport Insuter	<pre>: X[:,0:3] * import.fit transform(X[:,0:5]) In [5]: free sklearn cross validation import train text solit: X train, X text, y Frain, y text * train text solit(X, y, text size * random state * 0) C:\Mrsgnambti\MexecuteS\liMitic packages'svictorm\cross_validation.sy/41: DepressionHarming: This module may depression in a factor of in factor of in </pre>	e e moved.	
<pre>if transfer effective operation import Gelectifiest if for all serve feature_celection import onio (a _ Gelectifiestre_celection import onio) (b _ Gelectifiestre_celection (), (), (), (), (), (), (), (), (), (),</pre>	<pre>: X[:,0:3] * import.fit transform(X[:,0:5]) In [5]: free aklearm cross validation import truit text solit: X truit, X text, y train, y text * truit text solit(X, y, text size * random state * 0) C: three restantiation of the solitic pockages between version 8.12 in [source of the eversion state is not be an a depression in version 8.12 in [source of the eversion states are offer that in the right of the solitions are afformed that are and formities an else note that the interface of the new CY intractes are afformed from form for the solitions an else note that the interface of the new CY intractes are afformed from the form for the solition of the s</pre>	e e moved.	
<pre>11 transference.selection import Gelectilect 15 from ciliarm.feature_celection import only 15 from ciliarm.feature_celection import only 16 transference.selection import only 16 transference.selection import Import 28 imports - Import.feature(), estimate 28 imports - Import.feature(), estimate 20 imports - Import.feature(), estimate 20 imports - Import.feature(), estimate 20 import - Import</pre>	: X[:,0:3] * isociar fit transform(X[:,0:5]) In [5]: free shiarm cross valuation insert train test solit X train, X test, y train, y test * train test solit(X, y, test size * random state * 8) C. 'Wegeneratetteneousles'libraic packages withow trans allocation, y 42: Deprets instantary: this mobile main dependent in version 8.15 in forse of the words, electric module into the wind will the reflected classes and functions are also note that the interface of the nar of iterators are different free that a social entity of the interface of the nar of iterators are different free that a module. This module mill be removed in 8.28." " "this module mill be removed in 8.28."	e e moved.	
<pre>11 * Journal & Starting 11 for a client fature_selection input Selectifiest 15 for client fature_selection input cn12 16 * Selectifiest(pd12, * 0, 1 for transform(X, y) 19 19 * Second Selection Selection (X, pd1) 19 * Second Selection Selection 19 * Second Selection Selection 20 * Selection Selection Selection 21 * Selection Selection Selection (Selection Selection) 22 * California Selection Selection Selection Selection 23 * California Selection Selection Selection Selection 23 * California Selection Selection Selection Selection 24 * California Selection Selectio</pre>	: X[:,0:2] • insider fit treaters(X[:,0:5]) In [5]: free allern cross validation insert trait test solit : X[:ruln, X] test, y train, y test = trait test solit(X, y, test size = random state = 0) C:thregmentationscendes/likeling portage: which is the form of the merch solition means into main all the reflected size will define a first be reader that the interface of the name of kirotars are eliferent from that a which made mail the rescale to X. "This means will be rescale in 0.2.", Derecationscening) in (a) free solar meights insert the approximation first (in (b) reader will be rescale in 0.2.", Derecationscening)	e e moved. of this	
<pre>if transfer effection import Generalized if for all serve feature_celection import Generalized is for all serve feature_celection import on 52 if a Generalized for the transference of the transference of the if for a feature from the transference of the transference if for a feature feature for the transference of the transference is import - Instantiation (sector) (sector) if is instantiation (sector) (sector) if is instantiation (sector) (sector) if it is instantiation (sector) (sector) if for all sector) (sector) (sector) if for all sector) (sector) (sector) (sector) if for all sector) (sector) (sector) (sector) (sector) if for all sector) (sector) (sector) (sector) (sector) (sector) if for all sector) (sector) (se</pre>	: X[:,0:3] * isociar.fit transform(X[:,0:5]) In [3]: free shkern.cross validation insort train test solit : X train, X test, y train, y test * train test solit(X, y, test size * random state * 0 C 'Mrogenetationscendes'librair package beliar "trans a solition, sy 42; Deprets instantary: "tis mobile man dependent in version 8.22 is force of the methylaction sould into which all the reflatored classes and functions and above that the interface of the new CY laraters are different free that a sould: "this module mill be removed in 8.28." "this module mill be removed in 8.28." (the sould entit be removed in 8.28.", DemocationMeering) In [6]: free shleron sciptors insert ObeliartScips = specify = "minematic."	e e moved. of this	
<pre>if transfer effective if for existent.feture_relection inport Selectifiest if for existent.feture_relection inport mi2 if is "Selecting-relection inport mi2 if a selecting strains (in the transfer (X, y)) if the selecting strains if prior = nutre(stearing value 0, strategy rean", exit 0) if ((X, S) = inputer it transfer((X), (X, S))) if a selecting strains (inport frame selecting read for set if strain X test, y test, y test is the set is the set is solid if x test, X test, y test, y test is test test is solid if x test is test, y test, y test is test is test is solid if x test is test, y test is test is test is solid if x test is test, y test is test is test is test is solid if y test is test is test is test is test is test is solid if y test is test</pre>	<pre>:X[:,0:2] = insider.fit treasform(N[:,0:5]) In [5]: free allowing roots validation insort train test solit:X[:ruln,X] test, y train, y test = train test solit:X[:ruln,X] test, y train, y test = train test solit:X[:ruln,X] test = train test = train test = train test solit C:throgeneticidestical information the restance of the solid restance of the solid</pre>	e e moved. of this	
<pre>if transmission feature coloring input SelectCleat if free cilesen feature coloring input SelectCleat if free cilesen feature coloring input onS2 if a SelectDirector visual if ree science.proceeding input of transform(X, y) if selector coloring visual 0, criterary.cean',acte 0) if (x):(0): - inputer.tit transform(X):(0):() if ree science.cross validation feature train test split if ree science.cross validation feature test science.cross split if ree science.cross validation feature test split if ree science.cross validation feature test split if ree sci</pre>	<pre>:X[:,0:3] * isociar.flt transform(N[:,0:5]) In [3]: free shkarm.cross validation import trait test solit:X[:vain, X test, y train, y test * irain test split(X, y, test size * rendm_state * 0; ('Mrogenerationseries'):illusing packages wither viewes sufficient may split Depretationseries':illusing packages wither viewes sufficient may split Depretationseries':illusing and depretation in version 8.22 is four of the seed_selection sould into work all the refatered charace and functions and also note that the interface of the name of iterators are different free that a result, "is mobile mill be removed in 8.28.", Depretationsering) In [4]: four first algorithm into the diplorational first: classifier display table("interface") </pre>	e e moved. of this	
<pre>if transfer effective if for exploration import Selectionst if for exploration feature_relection import mi2 if for explorations into a selection import mi2 if a feature floring volume if for exploration is a selection into a selection is a import for for exploration (selection feature) if (i,d) - import for the selection is a selection into a selection if (i,d) - import for the selection is a selection into a selection if the selection relection is into a selection is a selection in the into a selection is a selection in the selection is a selection if for exploration is a selection in the selection is a interval in the selection is a selection is a selection is a if the selection release is the forward and selection if for a below in the forward and if for a below is a selection is a selection is a if for a below is a selection is a selection is a if for a below is a selection is a selection is a if for a below is a selection is a selection is a if for a below is a selection is a selection is a if for a below is a selection is a selection is a selection is a if for a below is a selection is a selection is a selection is a if for a below is a selection is a selection is a selection is a selection is a if for a below is a selection is a selec</pre>	<pre>:X[:,0:2] * Messler.flt treasform(N[:,0:5]) In [5]: free allern.cross validation import truth test split(X, y, test size = random_state = 0) Comparison of the state of the split(X, y, test size = random_state = 0) Comparison of the state of the split(X) is the state of the mercl_solution would not into mich all the reflected classes and functions and sho made that the interface of ten may Compare the state of the mercl_solution would be removed in 0.2%. Descentioned and into made that the interface of ten may Compare the state of the mercl_solution would be removed in 0.2%. Descentioned for the state "files mould will be removed in 0.2%." Descentioned and the interface of ten may Compare the state of the mercl_solution would be removed in 0.2%." Observationed and the state of the state of ten may compare the state of the state "files mould will be removed in 0.2%." Descentioned and the state of ten may compare the state of ten may compare the state of ten the state of ten may compare the state of ten may compare the state of ten may be an explored and the state of ten may compare the state of ten may compare the state of ten may be an explored and the reflection would be the reduced in 0.2%." This mould will be removed in 0.2%." Descentioned and the state of ten may compare the state of ten may be an explored and the state of ten may be an explored and the state of ten may be an explored and the state of ten may be an explored and the state of ten may be an explored and the state of ten may be an explored and the state of ten may be an explored and the state of ten may be an explored and the state of ten may be an explored and the state of ten may be an explored and the state of ten may be an explored and the state of ten may be an explored and the state of ten may be an explored and the state of ten may be an explored and the state of ten may be an explored and the state of ten may be an explored and the state of ten may be an explored and the state of ten may be an explored and the state of ten may be</pre>	e e moved. of this	100.00
<pre>if transmission feature coloring input SelectGest if for a closen feature coloring input SelectGest if for a SelectDescription input for if a SelectDescription input for if a SelectDescription input for if for a closen, performance of the select if for a closen cross validation for the train test split if for a closen cross validation for the select select if for a closen cross validation for the train test split if for a closen cross validation for the train test split if for a closen cross validation for the select select select is a select set if for a closen for the forting sel if closelifer = OkiphoraClosenform tegetors = Spectric = "miniment", s=2; i closelifer = folk, then, y, test;</pre>	<pre>:X[:,0:3] * isociar.flt transform(N[:,0:5]) In [3]: free shkern.cross validation import trait test solit:X[:vain,X] test, y train, y test * train test split(X, y, test size * randm_state * 0; ('Mrogenetationscendes'):Hvain package validar/verses solitation.sy 42; Derect intriariaring: this mobile man dependent in version 8.12 is foury of th metch_selection sould into winh all the reflected classes and functions and show take that the interface of the new CAleraters are different free that a sould this model mill be resource in 8.28.", Demonstrateming) In [6]: free shleren seightest insert Objects/Localize</pre>	e e moved. of this	1000
<pre>if transferrer.Seture.Seture inport Selectifiest if for selecting relevant inport mi2 if for selecting transferrer inport mi2 if a Selecting transferrer inport Tester if for selecting transferrer (x, y) if if the seture inport of the seture if for selecting transferrer(x, y) if the seture inport for the seture if for selecting transferrer(x, y) if the seture inport for the seture if for selecting transferrer(x, y) if the seture inport for the seture inport for the set if for seture inport for the seture inport for the set if for seture inport for the seture inport for the set if for seture inport for the for the seture inport for the set if for seture inport for the for the seture inport for the set if for seture inport for the for the set set if the set inport for the for the for the set if for seture inport for the for the set set if the set inport for the for the for the set inport for th</pre>	<pre>:X[:,0:2] * Messler.flt treasform(N[:,0:5]) In [5]: free allern.cross validation import truth test split(X, y, test size = random_state = 0) Comparison of the state of the split(X, y, test size = random_state = 0) Comparison of the state of the split(X) is the state of the mercl_solution would not into mich all the reflected classes and functions and sho made that the interface of ten may Compare the state of the mercl_solution would be removed in 0.2%. Descentioned and the interface of ten may Compare the state of ten mercl_solution.com also made that the interface of ten may Compare the state of ten also made that the interface of ten may Compare the state of ten also made that the interface of ten may Compare the state "files models all be removed in 0.2%." Descentioned that "this models all be removed in 0.2%." Observationshold the state of ten may Compare the state of ten "the split of ten state of ten "the split of ten state of ten "this models all be removed in 0.2%." "this models all be tendels all be removed in 0.2%." "the state of tendels all be removed in 0.2%." "this models all be removed in 0.2%." "this models all be removed in 0.2%." "this models all be removed in</pre>	e e moved. of this	
<pre>if transfer Stortenton if for a chieren.feture_selection input Selectifiest if for a chieren.feture_selection input cmi2 if for schieren.feture_selection input cmi2 if for a chieren.feture_selection input for the if for a chieren.profession (input for the for the if for a chieren.profession (input for the for the if (input for the chierent for the for the for the set if for a chieren.cross validation input for the set if for a chieren set (input for the for the for the for the set if for a chieren is post for the for the for the for the set if for a chieren is the for the for the for the set if for a chieren is the for the for the for the set if for a chieren is the for the for the set if for a chieren is the for the set is the set if for a chieren is the for the for the for the set if for a chieren is the for the for the for the set if for a chieren is the for the for the set if for a chieren is the for the for the for the set if for a chieren is the for the for the for the set if for a chieren is the for the for the for the for the set if for a chieren is the for the for the for the for the set if for a chieren is the for the for the for the for the set if for a chieren is the for the for the for the for the for the for the isolation is the for the fo</pre>	<pre>:X[:,0:2] = instant, it transform(N[:,0:5]) In [5]: free allern.cross validation insert train test solit:X[:::::::::::::::::::::::::::::::::</pre>	e e moved. of this	
<pre>if transferrer.Seture.Seture.Seture inport SetureSetureSet if for a chieven.Seture_setUreSet inport mi2 if for a chieven.Seture_setUreSetUreSetUreSet if a setUreSe</pre>	<pre>:X[:,0:3] * isociar.flt transform(N[:,0:5]) In [3]: free shkern.cross validation import train test split(N, y, test stars:X[:vain,X] test, y train, y tax * train test split(N, y, test stars:X[:vain,X] test, y train, y test * train test split(N, y, test stars:X[:vain,X] test, y train, y test * train test split(N, y, test stars:X[:vain,X] test, y train of the split(Y) terms sufficiently and:X[:vain,X] test, y train of the split(Y) terms sufficiently and:X[:vain,X] test split(Y) test stars:X[:vain,X] test split(Y) test split(Y) test stars:X[:vain,X] test split(Y) test split(Y) test split(Y) test stars:X[:vain,X] test split(Y) test split(Y) test split(Y) test split(Y) test:X[:vain,Y] test split(Y) test split(Y) test split(Y) test split(Y) test:X[:vain,Y] test split(Y) test split(Y) test split(Y) test split(Y) test:X[:vain,Y] test split(Y) test split(Y) test split(Y) test split(Y) test:X[:vain,Y] test split(Y) test split(Y) test split(Y) test split(Y) test:X[:vain,Y] test split(Y) test split(Y) test split(Y) test split(Y) test:X[:vain,Y] test split(Y) test split(Y) test split(Y) test split(Y) test:X[:vain,Y] test split(Y) test split(Y) test split(Y) test split(Y) test:X[:vain,Y] test split(Y) test split(Y) test split(Y) test split(Y) test:X[:vain,Y] test split(Y) test split(Y) test split(Y) test split(Y) test:X[:vain,Y] test split(Y) test split(Y) test split(Y) test split(Y) test:X[:vain,Y] test split(Y) tes</pre>	e e moved. of this	
<pre>if transfer effection import Selections: if for a charm.festure_relection import only if for a charm.festure_relection import only if is interesting transfer (0,).fir transferrance(X, y) if if controls a characteristic import festure if for a characteristic (characteristic) if (0, 0, 0,) impure the characteristic (characteristic) if (0, 0, 0,) impure the characteristic (characteristic) if controls the characteristic) if controls the characteristic (characteristic) if controls the characteristic) classifier - Scherbers import Misightorsclassifier classifier - Scherbers import from the characteristic is involved one bid a control characteristic is involved one bid a control characteristic is involved one bid a controls (characteristic) is involved one bid a control characteristic) is involved one bid a control characteristic is involved one bid a control characteristic) is involved one bid a control characteristic (characteristic) is involved one bid a control characteristic) is involved one bid a control characteristic) is involved one bid a control characteristic) is involved one bid a control characteristic (characteristic) is involved one bid a control characteristic) is involved one bid a control characteristic (characteristic) is involved one bid a control characteristic) is involved one bid a control characteristic (characteristic) is involved one bid a control characteristic) is involved one bid a control characteristic) is involved one bid a control characteristic (characteristic) is involved one bid a control characteristic) is invo</pre>	<pre>:X[:,0:2] + insider.fit treaterm(N[:,0:5]) To [5]: free allern.cross validation insert trait test solit:X[:::::::::::::::::::::::::::::::::</pre>	e e moved. of this	
<pre>if transfer effection import Selections: if fee eligent.feture_relection import felections: if fee eligent.feture_relection import fmile if fee eligent.feture_relection import fmile if fee eligent.petures if more tileren.personenting import Teacter if more tileren.personenting import Teacter if fee eligent.petures if fee eligent.petures</pre>	<pre>:X[:,0:3] * Messler.flt treasform(N[:,0:5]) In [3]: free abherm creas validation import train text split(X, y, text, size = random state = 0) C:fVeganetation exception in the state of the state of the split(X, y, text, size = random state = 0) C:fVeganetation exception in the rank of the state of the split(X, y, text, size = random state = 0) C:fVeganetation exception in the rank of the split(X, y, text, size = random state = 0) C:fVeganetation exception is a dependent in the rank of the split(X, y, text, size = random state = 0) C:fVeganetation exception is the rank of the rank of the split(X, y, text, size = rank of the interface of the new CY interfaces are different free that a secule.tria module all be reasons in 0.20.", DestructionMentig() In (0): free addemon engineses incert ObelphoreAlessation = split(X, showing, text), c:classifier = (desplit)(text)(sect) = (sect)(text)(text)(text)(text)(text), c:classifier (text)(text)(text)(text)) enginter than stifter(text)(tex</pre>	e e moved. of this	
<pre>if transfer effective if for a chieren.festure_selection inport Selectifiest if for a chieren.festure_selection inport mi2 if s Selective_relevant inport mi2 if a chieren.proceeding inport Tester if for a chieren.proceeding inport Tester if prove Tester(stering value 0, statage read_set 0) if (1, 0, 5) - Inputer.itt transform((1, 0, 5)) if (1, 0, 5) - Inputer.itt transform (1, 0, 5)) if (1, 0, 5) - Inputer.itt transform (1, 0, 5)) if (1, 0, 5) - Inputer.itt transform (1, 0, 5)) if (1, 0, 5) - Inputer.ittt transform((1, 0, 5)) if (1, 0, 5) - Inputer.ittt transform((1, 0, 5)) if (1, 0, 5) - Inputer.ittt(1, 0, 5)) if (1, 0, 5)</pre>	<pre>:X[:,0:2] + insider.fit treaterm(N[:,0:5]) To [5]: free allern.cross validation insert trait test solit:X[:::::::::::::::::::::::::::::::::</pre>	e e moved. of this	
<pre>if transfer effection import Selections: if fee eligent.feture_relection import felections: if fee eligent.feture_relection import fmile if fee eligent.feture_relection import fmile if fee eligent.petures if more tileren.personenting import Teacter if more tileren.personenting import Teacter if fee eligent.petures if fee eligent.petures</pre>	<pre>:X[:,0:3] * Messler.flt treasform(N[:,0:5]) In [3]: free abherm creas validation import train text split(X, y, text, size = random state = 0) C:fVeganetation exception in the state of the state of the split(X, y, text, size = random state = 0) C:fVeganetation exception in the rank of the state of the split(X, y, text, size = random state = 0) C:fVeganetation exception in the rank of the split(X, y, text, size = random state = 0) C:fVeganetation exception is a dependent in the rank of the split(X, y, text, size = random state = 0) C:fVeganetation exception is the rank of the rank of the split(X, y, text, size = rank of the interface of the new CY interfaces are different free that a secule.tria module all be reasons in 0.20.", DestructionMentig() In (0): free addemon engineses incert ObelphoreAlessation = split(X, showing, text), c:classifier = (desplit)(text)(sect) = (sect)(text)(text)(text)(text)(text), c:classifier (text)(text)(text)(text)) enginter than stifter(text)(tex</pre>	e e moved. of this	

Fig: 8. Making the confusion matrix

CONCLUSION

The conventional method for detecting diabetes is time consuming, a clinical doctor must take many factors into consideration. The system we have developed uses a ML model to correctly determine health status of a person. So far Random Forest model proved to be the best amongst all in terms of accuracy in our type 2 diabetes data. Random forest model uses rule based system to tell whether a person is diabetic or not. The model can not only serve as a guide for doctors specializing in diabetes but also help family practitioners and interns in prescribing medications. It should be noted that diabetes cannot be predicted in a single event. Readings are taken for one



week before any conclusion can be made from it. Our model serves the preliminary step of detecting diabetes. After that a check-up is recommended from doctor.

Future Scope

Predicting disease using AI is soon going to reduce the workload of clinical doctors. It is still under research that how accurate the results will be and continuous efforts are made to make intelligent systems. So far neural networks have outperformed very model in various domain in terms of accuracy but when it comes to predicting disease, neural networks do not perform better. Research is still going on how to make neural networks more accurate for disease. Neural predicting Networks provide a wide range of possibilities which in future may be efficient enough to predict diseases too.

REFERENCES

- 1. HLA Nomenclature in WMDA file format-details of current HLA alleles and where known their unambiguous, possible or assumed serologically equivalent antigens. http://hla.alleles.org/wmda/rel_dna_ser .txt. Accessed April 13, 20164
- 2. International Diabetes Federation. Diabetes Atlas. 5th ed. Brussels, Belgium: IDF Publications. (2011) The Global Burden of Diabetes; pp. 7–13. Available from http://www.idf.org/diabetesatlas/news/ fifth-edition-release. Accessed 25 May2015
- Georgiev, D., Houdová, L., Fetter, M., Jindra, P.: A Scalable Method for Efficient Stem Cell Donor HLA Genotype Match Determination. In: Energy, Environment, Biology and Biomedicine, Proceedings of the 2014 International Conference on Biology and Biomedicine II, pp. 28-32. INASE, Prague (2014)3
- 4. Hayuhardhika W, Putra N, Sugiyanto,

Sarno R, Sidiq M (2013) Weighted Ontology and Weighted Tree Similarity Algorithm for Diagnosing Diabetes Mellitus. IEEE International Conference on Computer, Control, Informatics and Its Applications pp.267-272.

- Chen R, Huang Y, Bau C, Chen S (2012) A recommendation system based on domain ontology and SWRL for anti-diabetic drugs selection. Expert Systems with Applications 39: 3995–4006
- 6. NMDP allele codes. https://bioinformatics.bethematchclinic al.org/hla-resources/ allele- codes/. Accessed April 13,2016
- A. S. Abdelmoneim, D. T. Eurich, J.-M. Gamble, and S. H. Simpson, "Use patterns of antidiabetic regimens by patients with type 2 diabetes," Canadian Journal of Diabetes, vol. 37, no. 6, pp. 394–400, 2013.
- 8. World Marrow Donor Association International Standards for Unrelated Hematopoietic Stem Cell Donor registries (2014), https://www.wmda.info/images/ pdf/20140101-STDC-WMDA_Standards_New_Housestyle.p df. Accessed April 7,2016
- Y. Handelsman, Z. T. Bloomgarden, G. Grunberger et al., "American Association of Clinical Endocrinologists and American College of Endocrinology - clinical practice guidelines for developing a diabetes mellitus comprehensive care plan - 2015," Endocrine Practice, vol. 21, Supplement 1, pp. 1–87,2015

practice guidelines for developing a diabetes mellitus comprehensive care plan - 2015," Endocrine Practice, vol. 21, Supplement 1, pp. 1–87,2015

