**PROFESSIONAL TRAINING REPORT**

**at**

**Sathyabama Institute of Science and Technology (Deemed to be University)**

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Engineering Degree in Computer Science and Engineering

By

**B. RUPESH SURYA**

**REG. NO. 40111084**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SCHOOL OF COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,**

**CHENNAI – 600119, TAMILNADU**

**NOVEMBER 2022**

1

**SATHYABAMA** 

**INSTITUTE OF SCIENCE AND TECHNOLOGY**

(DEEMED TO BE UNIVERSITY)

**Accredited with Grade “A” by NAAC**

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI

CHENNAI– 600119

**www.sathyabama.ac.in**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the Bonafide work of **B. RUPESH SURYA (Reg. No: 40111084)** who carried out the project entitled “**Diabetes prediction**” under my supervision from June 2021 to November2021.

**Internal Guide**

**Ms.D. Menaka M.E.,(Ph.D)**

**Head of the Department**

**Dr. L. Lakshmanan, M.E., Ph.D.,**

**Dr. S. Vigneshwari, M.E., Ph.D.**

**Submitted for Viva voce Examination held on**

**Internal Examiner External Examiner**

2

**DECLARATION**

I, **B. RUPESH SURYA** hereby declare that the project report entitled “**Diabetes prediction system”** done by me under the guidance of **Ms.D. Menaka M.E.,(Ph.D)** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

**DATE:**

**PLACE: SIGNATURE OF THECANDIDATE**

3

**ACKNOWLEDGEMENT**

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D**, **Dean**, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of **Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. D Menaka M.E.,(Ph.D)** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project

4

**TRAINING CERTIFICATE**

****5

**TABLE OF CONTENTS**

**CHAPTER NO TITLE PAGE NO**

1 **INTRODUCTION** 1 1.1 ESTIMATING DIABETES 1

1.1.1 TYPES OF DIABETES 2

1.2 ESTIMATION DIABETES DATASET INFORMATION 3

1.3 COMMON MACHINE LEARNING ALGORITHMS AND

GOALS 3

2 **AIM**

2.1 AIM 9

2.2 ATTRIBUTE INFORMATION 9

2.3 MISSING ATTRIBUTES 10

2.4 DATA PREPARATION 10

3**EXPERIMENTAL OR MATERIALS AND METHODS,**

**ALGORITHMS USED**

3.1 TYPES OF CLASSIFICATION ALGORITHMS USED 12

3.2 LOGISTICS REGRESSION ALGORITHM 12

3.3 IMPORTED LIBRARIES 12

3.4 DECISION TREE ALGORITHM 16

4 **RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS**

4.1 TRANING AND ACCURACY (MODEL ANALYSIS) 17 4.2 TESTING THE MODEL 19

5 **REFERENCE** 19 6 **APPENDIX** 20 7 **SCREENSHOTS AND OUTPUTS** 21 8 **SOURCE CODE** 30 9 **SCREENSHOTS OF GUI** 34

6

**LIST OF FIGURES**

**FIGURE NO FIGURE NAME PAGE NO**

1 excel sheet of the given Estimating Obesity dataset 3 1.1 Types of machine learning along with the field of use 4 1.2 Linear regression scatter plot 4 1.3 Linear regression scatter plot with regression line 5 1.4 Logistic regression with probability output in s shape 6 1.5 Example for decision tree 7 1.6 Random forest outlier 7 1.7 scatter plot on K- means clustering 8

2 Described dataset 10 2.1 code for reading the dataset using pandas 11 2.2 using isnull() function checking for null values 11 3.1 various python libraries for Machine Learning 13 3.2 pandas library is used to read data sheet 13 3.3 NumPy library for changing to NumPy array 14 3.4 Separation of data 14 3.5 Graph 15 3.6 Imported libraries in jupyter notebook 15 4.1 outlier of confusion matrix 17 4.2 Training and Precession data 18 4.3 Testing the model 19 6.1 Anaconda Navigator 20 6.2 Coding Environment 20 7.1 Screenshot of KNeighbors Model 21 7.2 Screenshot of KNeighbors Model 21 7.3 Screenshot of Model 22 7.4 Screenshot of Model 22 7.5 Screenshot 23 7.6 Screenshot of Scalling the data 23 7.7 Screenshot of Spliting the Dataset 24 7.8 Screenshot of Spliting the Dataset 24 7.9 Screenshot of Correlation matrix 25 7.9.1 Screenshot of Correlation matrix 25 7.9.2 Screenshot of Data 26

7.9.3 Screenshot of Data 26 7.9.4 Screenshot of Dataset 27 7.9.5 Screenshot of Dataset 27 7.9.6 Screenshot of Data 28 7.9.7 Screenshot of Dataset 28 7.9.8 Screenshot of Importing the required libraries 29 7.9.9 Screenshot of Testing the Model 29

9.1 Screenshot of GUI 34 9.2 Screenshot of GUI 34 9.3 Screenshot of GUI 35 9.4 Screenshot of Result 35

7

**CHAPTER 1 INTRODUCTION**

Diabetes is a chronic disease with the potential to cause a worldwide health care crisis. According to International Diabetes Federation 382 million people are living with diabetes across the whole world. By 2035, this will be doubled as 592 million. Diabetes mellitus or imply diabetes is a disease caused due to the increase level of blood glucose.

Various traditional methods, based on physical and chemical tests, are available for diagnosing diabetes. However, early prediction of diabetes is quite challenging task for medical practitioners due to complex interdependence on various factors as diabetes affects human organs such as kidney, eye, heart, nerves, foot etc.

All around there are numerous ceaseless infections that are boundless in evolved and developing nations. One of such sickness is diabetes. Diabetes is a metabolic issue that causes blood sugar by creating a significant measure of insulin in the human body or by producing a little measure of insulin. Diabetes is perhaps the deadliest sickness on the planet. It is not just a malady yet, also a maker of different sorts of sicknesses like a coronary failure, visual deficiency, kidney ailments and nerve harm, and so on.

Subsequently, the identification of such chronic metabolic ailment at a beginning period could help specialists around the globe in forestalling loss of human life. Presently, with the ascent of machine learning, AI, and neural systems, and their application in various domains [1, 2] we may have the option to find an answer for this issue. ML strategies and neural systems help scientists to find new realities from existing well-being-related informational indexes, which may help in ailment supervision and detection. The current work is completed utilizing the Pima Indians Diabetes Database. The point of this framework is to make an ML model, which can anticipate with precision the likelihood or the odds of a patient being diabetic. The ordinary distinguishing process for the location of diabetes is that the patient needs to visit a symptomatic focus. One of the key issues of bio-informatics.

examination is to achieve precise outcomes from the information. Human mistakes or various laboratory tests can entangle the procedure of identification of the disease. This model can foresee whether the patient has diabetes or not, aiding specialists to ensure that the patient in need of clinical

1

consideration can get it on schedule and also help anticipate the loss of human lives.

DNA makes neural networks the apparent choice. Neural networks use neurons to transmit data across various layers, with each node working on a different weighted parameter to help predict diabetes.

Presently, with the ascent of machine learning, AI, and neural systems, and their application in various domains [1, 2] we may have the option to find an answer for this issue. ML strategies and neural systems help scientists to find new realities from existing well-being related informational indexes, which may help in ailment supervision and detection. The current work is completed utilizing the Pima Indians Diabetes Database.

Genetic factors are the main cause of diabetes. It is caused by at least two mutant genes in the chromosome 6, the chromosome that affects the response of the body to various antigens. Viral infection may also influence the occurrence of type 1 and type 2 diabetes. Studies have shown that infection with viruses such as rubella, Coxsackievirus, mumps, hepatitis B virus, and

cytomegalovirus increase the risk of developing diabetes.

**1.1.1 Types of Diabetes**

**Type 1**

Type 1 diabetes means that the immune system is compromised and the cells fail to produce insulin in sufficient amounts. There are no eloquent studies that prove the causes of type 1 diabetes and there are currently no known methods of prevention.

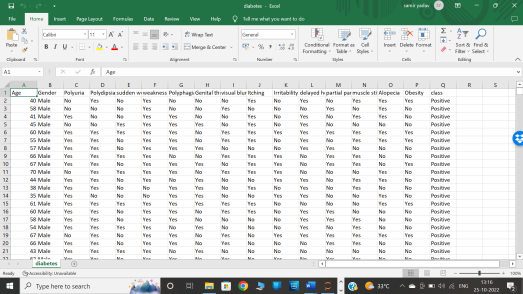
**Type 2**

Type 2 diabetes means that the cells produce a low quantity of insulin or the body can't use the insulin correctly. This is the most common type of diabetes, thus affecting 90% of persons diagnosed with diabetes. It is caused by both genetic factors and the manner of living.

2

This project aims to predict diabetes via three different supervised machine learning methods including: SVM, Logistic regression, KNN. This project also aims to propose an effective technique for earlier detection of the diabetes disease using Machine learning algorithms and end to end deployment using flask.

**1.2 ESTIMATION DIABETES DATASET INFORMATION**

**Fig-1: excel sheet of the given Estimating Obesity dataset**

**1.3 COMMON MACHINE LEARNING ALGORITHMS AND GOALS**

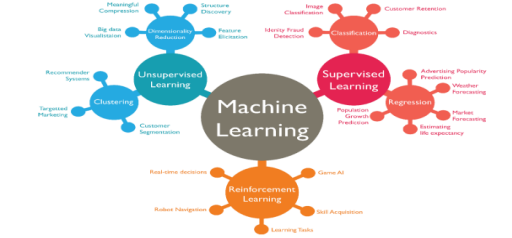
Then the variety of machine learning algorithms are classified into three categories as follows –

**Supervised learning** algorithms model the relationship between features (independent variables) and a label (target) from given set of observations. Then the model is used to predict the label of new observations using the features. Depending on the characteristics of the target variable i.e., it can be either be classification (discrete variable) or regression (continuous variable) the task is further engaged.

**Unsupervised learning** finds the structures in unlabeled data.

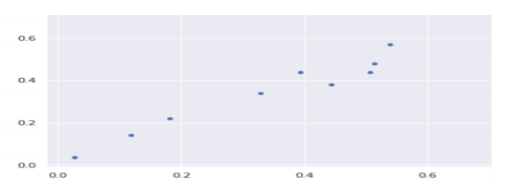
3

**Reinforcement learning** works on action-reward principle. An agent learned to reach the goal by continuously calculating the rewards that it gained from the actions

 **Fig-1.1: Types of machine learning along with the field of use**

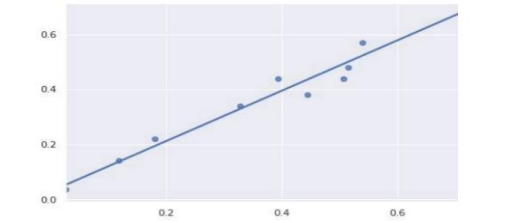
**ALGORITHMS**

**1.** Linear Regression is a supervised learning algorithm and tries to be a bridge between a continuous target variable and one or more independent variables by fitting a linear equation to the data. For choosing this algorithm, there needs be a linear relation between independent and target variable. As scatter plot shows the positive correlation between an independent variable(x-axis) and dependent variable (y-axis).

**Fig-1.2: Linear regression scatter plot**

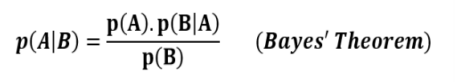
4

This would try to put regression line to represent relations. Common technique is ordinary least squares (OLS). As a result, we could get a regression line as an outcome by minimizing sum square of distance between data points and regression line.

**Fig-1.3: Linear regression scatter plot with regression line**

2. Naïve Bayes is a supervised learning algorithm used for classification problems, also called as Naïve Bayes Classifier. It assumes that features are independent of each other and there is no correlation between features. As assumption of features being uncorrelated is the reason for the name “naïve”.

**Equation:**

**** p(A|B): Probability of event A given event B has already occurred

p(B|A): Probability of event B given event A has already occurred

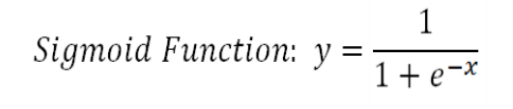
p(A): Probability of event A

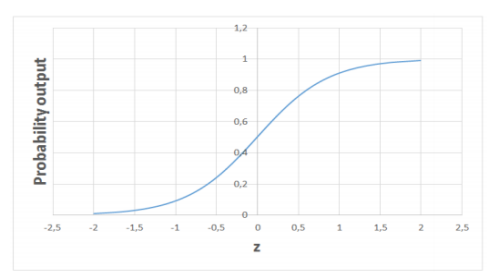
p(B): Probability of event B

5

3. Logistic Regression is a supervised learning algorithm which is mostly used for binary classification problems. Even when regression contradicts with classification, here the spot is for logistic that refers to logistic function which does the classification task. It is simple but effective classification algorithms most commonly used for binary classification problems. Logistic function also known as sigmoid function.

**Equation:**

**** Logistic regression takes linear equation as input and uses sigmoid function and logs odds to perform a binary problem. As result s shape graph will be the output

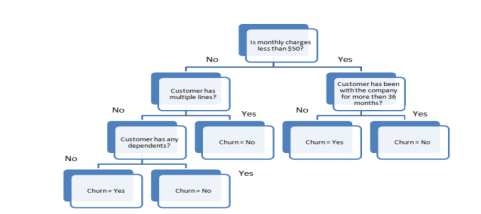
**Fig-1.4: Logistic regression with probability output in s shape**

4. Decision Trees build upon continuously to partition the data. The aim of decision tree is to increase the predictiveness as much as possible at each stage so that the model keeps

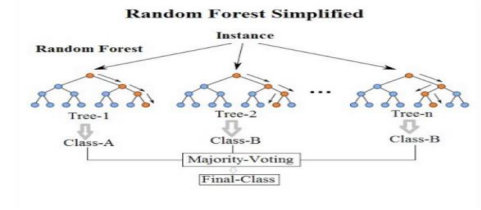
6

gaining information about the dataset. Randomly splitting will not give us valuable insight into dataset. The purity of node is inversely proportional to distribution of different classes in that node. Overfitting model would be too specific model and not be generalize well. Though it achieves high accuracy with training set but poorly on new. The depth of the tree is controlled by max\_depth parameter for decision tree algorithm in scikit-learn. Is also suitable to work on a mixture of feature data types.

**Example**

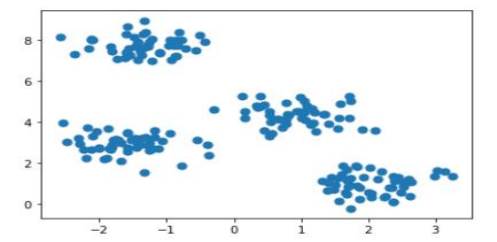
**Fig-1.5: Example for decision tree**

5. Random Forest is an ensemble of many decision trees. They are built using a method called bagging where decision trees are used as parallel estimators. When used in classification problem, the result will be based on majority of vote received from each decision tree.

**Fig-1.6: - Random forest outlier**

7

6. **K-means Clustering** is a way to group of set of data points in a way that similar data points are together. Thus, they look for dissimilarities or similarities among data points. It is an unsupervised learning so there is no label associated with data points. They try to find the underlying structures of the data. Clustering is not Classification.

**Fig-1.7: – scatter plot on K- means clustering**

8

**CHAPTER 2 AIM AND DETAILED ANALYSIS OF DATA 2.1 AIM**:

To predict the Diabetes in the given diabetes Data set using features from the given Data set.

**2.2 ATTRIBUTES INFORMATION:**

1. Data set characteristics are Multivariate

2. Attribute characteristics are categorical, integer, real

3. Associated task – Classification

4. no. of. instances – 550

5. no. of attributes – 17

**ATTRIBUTES ARE:**

1. Age: numeric

2. Gender: 0= female, 1 = male

3. Polyuria: 0 = NO, 1 = YES

4. Polydipsia: 0 = NO, 1 = YES

5. Sudden weight loss: 0 = NO, 1 = YES

6. weakness: 0 = NO, 1 = YES

7. Polyphagia: 0 = NO, 1 = YES

8. Genital thrush: 0 = NO, 1 = YES

9. visual blurring: 0 = NO, 1 = YES

10. Itching: 0 = NO, 1 = YES

11. Irritability: 0 = NO, 1 = YES

12. delayed healing: 0 = NO, 1 = YES

13. partial paresis: 0 = NO, 1 = YES

9

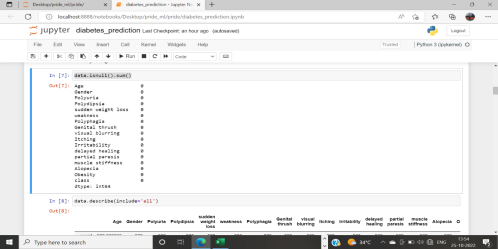
14. muscle stiffness: 0 = NO, 1 = YES

15. Alopecia: 0 = NO, 1 = YES

16. Obesity: 0 = NO, 1 = YES

17. Class (Target variable): 0 = POSITIVE, 1 = NEGATIVE

**2.3 MISSING ATTRIBUTES**: (denoted by "?")

**Fig-2: Described dataset**

**2.4 DATA PREPARATION**

In this project as a backbone tool python is used to carry out machine learning concepts. With the help of a software called Anaconda Navigator, a jupyter notebook is launched where it is already installed along with the navigator. jupyter notebook is an open-source web application that allows to create and share documents and has live code and also visualization.

After importing the required libraries the dataset will be read in the note book with help of data frame (two-dimensional labeled data structure with columns of potentially different types) and read\_csv(desired file type).

10

**Fig-2.1: code for reading the dataset using pandas**

Check the values for null using isnull() function.



**Fig-2.2: using isnull() function checking for null values**

The dataset is now ready for creating a machine learning model.

11

**CHAPTER 3 EXPERIMENTAL OR MATERIAL AND METHODS, ALGORITHMS USED**

The given data set is in the form of Classification algorithm .so, we used classification types to Predict the activity.

**3.1 Types of classification algorithms used:**

1. Logistic Regression Algorithm

2. Decision Tree Algorithm

**3.2 Logistic Regression Algorithm:**

o Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

o Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.

o Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.

o In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

**3.3 IMPORTED LIBRARIES**

Libraries are collections of prewritten code that users can use to optimize tasks. In project as python is used for implementation tool, it has the most libraries as compared to other programming languages. More than of 60% machine learning developers use and goes for python as it is easy to learn. As python has comparatively large collection of libraries let’s look at the libraries that came in handy for mammographic dataset.

12

 **Fig-3.1: various python libraries for Machine Learning**

**LIBRARIES USED:**

**1. Pandas** is a widely-used data analysis and manipulation library for python. It provides a lot of functions and methods that expedite the data analysis and preprocessing steps. IT also provides fast, flexible and expressive structures working with relational or labeled or both easy and intuitive. Considered as fundamental high-level building block in performing practical, real-world data analysis in python. Has powerful tools like Data Frame and Series for analyzing.

**Fig-3.2: pandas library is used to read data sheet**

13

**2. NumPy** stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of countless of routines for processing those arrays. Using this mathematical and logical operations on arrays can be performed. The difference in using NumPy from pandas is, it works on numerical data whereas pandas on tabular data.

**Fig-3.3: NumPy library for changing to NumPy array**

**3. sklearn** stands for Scikit-learn, a machine learning library. It is imported for various classification, regression and clustering algorithms including k-means, random forest, support vector machines, gradient boosting and DBSCAN. It is designed using libraries NumPy and SciPy. From the sklearn library and from the tree inside the library Decision Tree Classifier. It is a class capable of performing multi-class classifier on a dataset. When compared with other classifiers, Decision Tree Classifier takes input as two arrays: an array X, a parse or dense, of shape (n\_samples, n\_features) holding training samples and an array Y of integer values, shape (n\_samples), holding class labels for training sample. From sklearn another one called model\_selection for training and testing model imports train\_test\_split. It is a method setting a blueprint to analyze data and the using it to measure new data. Selecting a proper model allows to generate accurate results while making prediction. For proceeding, we need to train the model by using a specific dataset and test the model against another dataset.

By default, sklearn train\_test\_split will make random partitions for two subsets. We can also specify a random state for the operation. First, we need to split the dataset and then allocate the size for train and test. For this mammographic dataset we need train size as 80% (0.80) and test size as 20% (0.20) with the random state of 100.

**Fig-3.4: Separation of data**

14

**4. Seaborn** is a library built on top of matplotlib. It used for data visualization and exploratory data analysis. They work easily with data frames and pandas library. The graphs created can also be customized easily. It provides default styles and color palettes to make statistical plots more attractive. Also, closely integrated to the data structures from pandas.

**5. Matplotlib.pyplot** is a state-based interface to matplotlib. It provides a MATLAB-like way of plotting. It makes changes to figures.

**Fig-3.5: Graph**

****

**Fig-3.6: Imported libraries in jupyter notebook**

15

**3.4 LOGISTIC REGRESSION ALGORITHM**

This type of statistical model (also known as *logit model*) is often used for classification and predictive analytics. Logistic regression estimates the probability an event occurring, such as voted or didn’t vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and in logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure. This is also commonly known as the log odds, or the natural logarithm of odds, and this logistic function is represented by the following

**Formulas:**

Logit(pi) = 1/(1+ exp(-pi))

ln(pi/(1-pi)) = Beta\_0 + Beta\_1\*X\_1 + … + B\_k\*K\_k

16

**CHAPTER 4 RESULTS AND PERFORMANCE ANALYSIS 4.1 TRAINING AND ACCURACY (MODEL ANALYSIS)**

Confusion matrix is to evaluate the accuracy of a classification. This visual metric plots the number of predictions made for each class for each possible class in a table, with each row corresponding to the actual labels and each column corresponding to a prediction. It is beneficial for detecting which actual classes are being detected the most, and what predicted classes are being misclassified as (Bhardwaj and Tiwari, 2015; Liu et al., 2009). To further highlight the misclassifications and compare predictions with other classifiers, the confusion matrices are normalized to show a percentage rather than a count. The resulted confusion matrix has 45 incorrect prediction i.e., (25 + 20 = 45). From the matrix of y\_test and y\_pred with the help of metrics library the incorrect prediction is the outcome.



**Fig 4.1- outlier of confusion matrix**

**Accuracy** is an evaluation metric would be misleading as it would not be representative of how well the classifier fitted the data. Additionally, in breast cancer detection, detecting FPs and FNs is primordial to avoid interpreting malignant as benign and vice versa, an interpretation which could harm the patient and eventually lead to their death.

Accuracy = T P + T N/ P + N

For instance, if a dumb classifier that always classifies an image as “normal” is created, it would achieve 64.28% accuracy on the mini-MIAS dataset despite never picking up abnormal cases. Therefore, a mixture of additional metrics should be used to assess how well the model learns the mammograms data and generalize to unseen cases.

**Precision**

corresponds to the number of correct positive predictions showing the model’s ability to

17

avoid labelling negative instances as positive.

Precision = T P /T P + F P

**Recall**

Is the number of positive instances that are correctly predicted showing how well the model can find all positive instances.

Recall = T P/ T P + F

**F1 Score**

The F1 score is the 2\*((precision\*recall)/ (precision + recall)). It is also called the F Score or the F Measure. Put another way, the F1 score conveys the balance between the precision and the recall.

**Fig 4.2- Training and Precession data**

18

**4.2 TESTING THE MODEL**

**Fig 4.3- Testing the model**

**CHAPTER 5 REFERENCES:**

https://docs.python.org/3/

scikit-learn: machine learning in Python — scikit-learn 1.1.2 documentation campusX (HINDI) -- What is Machine Learning? | 100 Days of Machine Learning - YouTube

19

**CHAPTER 6 APPENDIX**

**WORKING ENVIRONMENT**

**ANACONDA NAVIGATOR** is desktop GUI used to launch applications and also manage packages in one place. Outlook

**Fig 6.1- Anaconda Navigator**

**CODING ENVIRONMENT**

Jupyter notebook from the anaconda navigator is launched along with all the preinstalled packages for python.



**Fig 6.2- Coding Environment**

20

**CHAPTER 7 SCREENSHOTS AND OUTPUTS Fig 7.1- Screenshot of KNeighbors Model**

**Fig 7.2- Screenshot of KNeighbors Model**

21

**Fig 7.3- Screenshot of Model**

**Fig 7.4- Screenshot Model**

22

**Fig 7.5- Screenshot**

**Fig 7.6- Screenshot of Scalling the data**

23

**Fig 7.7- Screenshot of Spliting the Dataset**

**Fig 7.8- Screenshot of Spliting the Dataset**

24

**Fig 7.9- Screenshot of Correlation matrix**

**Fig 7.9.1- Screenshot of Correlation matrix**

25

**Fig 7.9.2- Screenshot of Data**

**Fig 7.9.3- Screenshot of Data**

26

**Fig 7.9.4- Screenshot of Dataset**

**Fig 7.9.5- Screenshot of Data**

27

**Fig 7.9.6- Screenshot of Data**

**Fig 7.9.7- Screenshot of Dataset**

28

**Fig 7.9.8- Screenshot of Importing the required libraries**

**Fig 7.9.9- Screenshot of Testing the Model**

29

**CHAPTER 8 SOURCE CODE**

import numpy as np

import pandas as pd

import seaborn as sns

import pickle

import matplotlib.pyplot as plt

from sklearn.metrics import accuracy\_score,precision\_score, recall\_score,confusion\_matrix,f1\_score,plot\_confusion\_matrix from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.neighbors import KNeighborsClassifier from sklearn.tree import DecisionTreeClassifier

data=pd.read\_csv('diabetes.csv')

data.sample(10)

data["class"].value\_counts()

data.info()

data.isnull().sum()

data.describe(include='all')

# data preprocessing

data.nunique()

df=data.copy()

df

#converting categorical data into numerical data

df['Gender'] = df ['Gender'].replace({'Female':0,'Male':1 }) df['Gender']

for column in df.columns.drop(['Age','Gender','class']): df[column]= df[column].replace({'No':0 , 'Yes': 1})

df['class'] = df ['class'].replace({'Positive':0,'Negative':1 }) df

#Correlation

30

df.corr()

#HEATMAP

sns.heatmap(df.corr(),linewidths=.5)

sns.countplot(df['class'])

#separating the column

y=df["class"]

X=df.drop("class", axis=1)

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

X\_train, X\_test,y\_train,y\_test =

train\_test\_split(X,y,train\_size=0.8,shuffle=True,random\_state=123)

#scalling of data

scaler=StandardScaler()

X\_train=pd.DataFrame(scaler.fit\_transform(X\_train),index=X\_train.index , columns=X\_train.columns)

X\_train

y\_train

#Logistic regression

model\_1=LogisticRegression()

model\_1.fit(X\_train,y\_train)

print("Train\_score :",model.score(X\_train,y\_train)\*100) print("Test\_score :",model.score(X\_test,y\_test)\*100)

y\_pred\_1=model\_1.predict(X\_test)

y\_pred\_1

print("confusion matrix :\n",confusion\_matrix(y\_test,y\_pred\_1)) #PRECISION

print("Precesion : ",precision\_score(y\_test,y\_pred\_1))

#RECALL

print("precision :",recall\_score(y\_test,y\_pred\_1))

# F1 SCORE

print("F1 Score : ",f1\_score(y\_test,y\_pred\_1))

#decisionTree algorithm

model\_2=DecisionTreeClassifier()

model\_2.fit(X\_train,y\_train)

print("Train\_score :",model\_2.score(X\_train,y\_train)\*100) print("Test\_score :",model\_2.score(X\_test,y\_test)\*100) y\_pred2=model\_2.predict(X\_test)

31

print("confusion matrix :\n",confusion\_matrix(y\_test,y\_pred2))

#PRECISION

print("Precesion : ",precision\_score(y\_test,y\_pred2))

#RECALL

print("precision :",recall\_score(y\_test,y\_pred2))

# F1 SCORE

print("F1 Score : ",f1\_score(y\_test,y\_pred2))

from sklearn import metrics

print(metrics.classification\_report(y\_test,y\_pred2))

#KNeighbors Model

model\_3=KNeighborsClassifier()

model\_3.fit(X\_train,y\_train)

print("Train\_score :",model\_3.score(X\_train,y\_train)\*100) print("Test\_score :",model\_3.score(X\_test,y\_test)\*100) y\_pred3=model\_3.predict(X\_test)

print("confusion matrix :\n",confusion\_matrix(y\_test,y\_pred3))

#PRECISION

print("Precesion : ",precision\_score(y\_test,y\_pred3))

#RECALL

print("precision :",recall\_score(y\_test,y\_pred3))

# F1 SCORE

print("F1 Score : ",f1\_score(y\_test,y\_pred3))

y\_pred3

#TESTING THE MODEL

input\_data=(39,0,0,0,1,0,1,0,1,0,1,0,1,0,0,1)

#Changing the input data to numpy array

input\_data=np.asarray(input\_data)

# reshape the array as predicting for one instance input\_data=input\_data.reshape(1,-1)

32

#Scalling of data

std\_data=scaler.transform(input\_data) print(std\_data)

predict=model\_2.predict(std\_data)

if(predict==0):

print("\nReport ---> Positive\n\n")

else:

print("\nReport ---> Negative\n\n")

#Pickle File

pickle.dump(model\_2, open("diabetes.pkl", 'wb'))

33

**CHAPTER 9 SCREENSHOTS OF GUI**

**Fig 9.1- Screenshot of GUI**

**Fig 9.2- Screenshot of GUI**

34

**Fig 9.3- Screenshot of GUI**

**Fig 9.48- Screenshot of Result**

35