**Topic: Diabetes Prediction Using Machine Learning**

Group Members:

**2127032 Trupti Kharat**

**2127062 Shreyash Somvanshi**

**2127063 Sujay Shinde**

**2127073 Vedanti Mane**

**2127076 Prajwal Rudrapwar**

**Explanation:**

Diabetes is a chronic disease with the potential to cause a worldwide healthcare crisis. According to international Diabetes Federation 382 million people are living with diabetes across the whole world.

Motive of this project is to use the Artificial Intelligence technologies like Machine Learning in healthcare sector to increase efficiency and accuracy of results.This Diabetes Prediction using Machine Learning uses dataset containing records of patient’s bmi,insulin, glucose level, age,etc..Currently we have achieved 80% accuracy but it can be made more precise with help of appropriate data processing. This Diabetes Prediction Model makes use of various classification Algorithms to categorize patients into diabetic and non-diabetic.

**DATASET INFO:**

* Dataset link: <https://github.com/Yantra-Byte/dataset/raw/main/Diabetes.csv>
* Dataset shape:768 rows X 12 columns

**Code:**

# -\*- coding: utf-8 -\*-

"""PBL\_DiabetesFinal.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1IpgIYr-Vhc95CbLqkOEK78A8P8njEYoE

"""

import numpy as np

import pandas as pd

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

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score

dataset = pd.read\_csv(r'https://github.com/Yantra-Byte/Dataset/raw/main/Diabetes.csv')

dataset.head()

dataset.tail()

dataset.shape

dataset.info()

dataset.isnull().sum()

dataset.describe()

dataset['diabetes'].value\_counts()

X = dataset.drop(columns='diabetes',axis=1)

y = dataset['diabetes']

print(X)

print(y)

"""#Declaring Training and Testing Data"""

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, train\_size=0.6, random\_state = 250)

X\_train.shape, X\_test.shape, y\_train.shape, y\_test.shape

"""#Logistic Regression"""

model = LogisticRegression()

"""Training the LogisticRegression model with training data"""

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

"""#Model Evaluation

Accuracy score

Accuracy on training data

"""

X\_train\_pred = model.predict(X\_train)

trainingData\_accuracy = accuracy\_score(X\_train\_pred, y\_train)

print("Accuracy on Training Data : ", trainingData\_accuracy)

"""Accuracy on test data"""

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

testData\_accuracy = accuracy\_score(X\_test\_pred, y\_test)

print("Accuracy on Test Data : ", testData\_accuracy)

"""#Building a Predictive System"""

inputData = (1,138,42,36,170,45.1,2.050,35)

"""Change input data to numpy array"""

inputDataNumpyArray = np.asarray(inputData)

"""Reshape the numpy array as we are predicting for only one instance"""

inputDataReshaped = inputDataNumpyArray.reshape(1,-1)

prediction = model.predict(inputDataReshaped)

print(prediction)

if (prediction[0]==0):

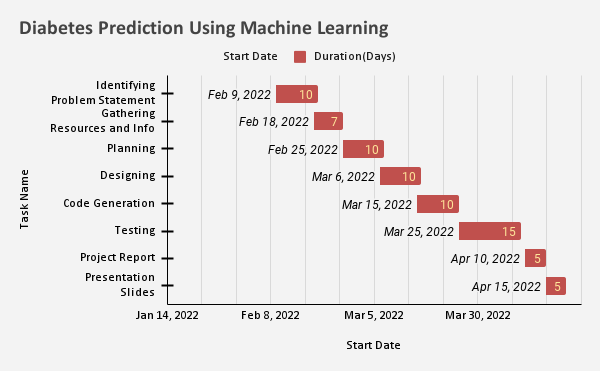
print("The patient doesn't have Diabetes. ")

else:

print("The patient is Diabetic. ")

**Project Planning:**

**(Gantt Chart)**

****

**Class Diagram:**

**Project Progress:**