**NAME OF THE COURSE : ARTIFICIAL INTELLIGENCE**

**NAME OF THE PROJECT : AI BASED DIABETES PREDICTION SYSTEM**

**PHASE - 1**

**PROBLEM STATEMENT:**

 The problem is to predict diabetes using machine learning techniques. The objective is to develop a model that accurately predicts the risk of developing diabetes in a person based on a set of features such as age, gender, weight, height, blood pressure, blood sugar levels, and other relevant factors. This project involves data preprocessing, feature engineering, model selection, training, and evaluation.

**WHAT I UNDERSTAND**

**1. Data collection:** The first step is to collect data about people's health and medical history. This data can be collected from a variety of sources, such as electronic health records (EHRs), clinical trials, and public health databases. It is important to collect data from a diverse population to ensure that the model is generalizable to a wide range of people.

**2. Data preprocessing:** Once the data has been collected, it needs to be preprocessed before it can be used to train a machine learning model. This involves cleaning the data, removing outliers, and handling missing values.

**3. Feature engineering:** Once the data has been preprocessed, you need to select the features that will be used to train the machine learning model. Not all features will be equally informative for the model, so it is important to select the features that are most relevant to predicting diabetes.

**4. Model selection:** There are many different machine learning algorithms that can be used for diabetes prediction. Some common choices include logistic regression, support vector machines, and random forests.

**5. Model training:** Once you have selected a model, you need to train it on your data. This involves feeding the model the data and allowing it to learn the relationships between the features and the target variable (i.e., whether or not a person has diabetes).

**6. Model evaluation:** Once the model is trained, you need to evaluate its performance on a held-out test set. This will give you an idea of how well the model will generalize to new data.

**7. Model deployment:** Once you are satisfied with the performance of the model, you can deploy it to production. This could involve integrating the model into an electronic health record (EHR) system or developing a stand-alone diabetes prediction app.

**8. Model monitoring:** It is important to monitor the performance of the model over time to ensure that it remains accurate. This is because new data may become available, and the prevalence of diabetes may change over time.

**9. Model retraining:** It may be necessary to retrain the model periodically with new data to ensure that it remains accurate. This is especially important if the prevalence of diabetes changes significantly over time.

**BASIC SOURCE CODE :**

import pandas as pd

import numpy as np

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

from sklearn.linear\_model import LogisticRegression # Example algorithm

# Step 1: Data Collection

# Load the dataset containing information about people's health and medical history

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

# Step 2: Data Preprocessing

# Handle missing values, duplicates, and encode categorical variables

# Address outliers if necessary

# Step 3: Feature Engineering

# Create new features or transform existing ones

# For example, you could calculate the body mass index (BMI) or the number of years a person has been overweight or obese.

# Step 4: Data Splitting

# Split the data into training and testing sets

X = data[['Age', 'Gender', 'Weight', 'Height', 'BloodPressure', 'BloodSugarLevels', 'FamilyHistoryDiabetes']] # Features

y = data['HasDiabetes'] # Target variable

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Step 5: Model Selection

# Choose a machine learning algorithm that is well-suited for classification tasks

model = LogisticRegression()

# Step 6: Model Training

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

# Step 7: Model Evaluation

# Evaluate the model's performance on the testing data

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

# Calculate metrics such as accuracy, precision, recall, and F1 score

# Step 8: Hyperparameter Tuning (if needed)

# Fine-tune the model's hyperparameters using techniques like cross-validation

# Step 9: Deployment (Optional)

# Deploy the model for making real predictions in a production environment

# Step 10: Monitoring and Maintenance (Continuous)

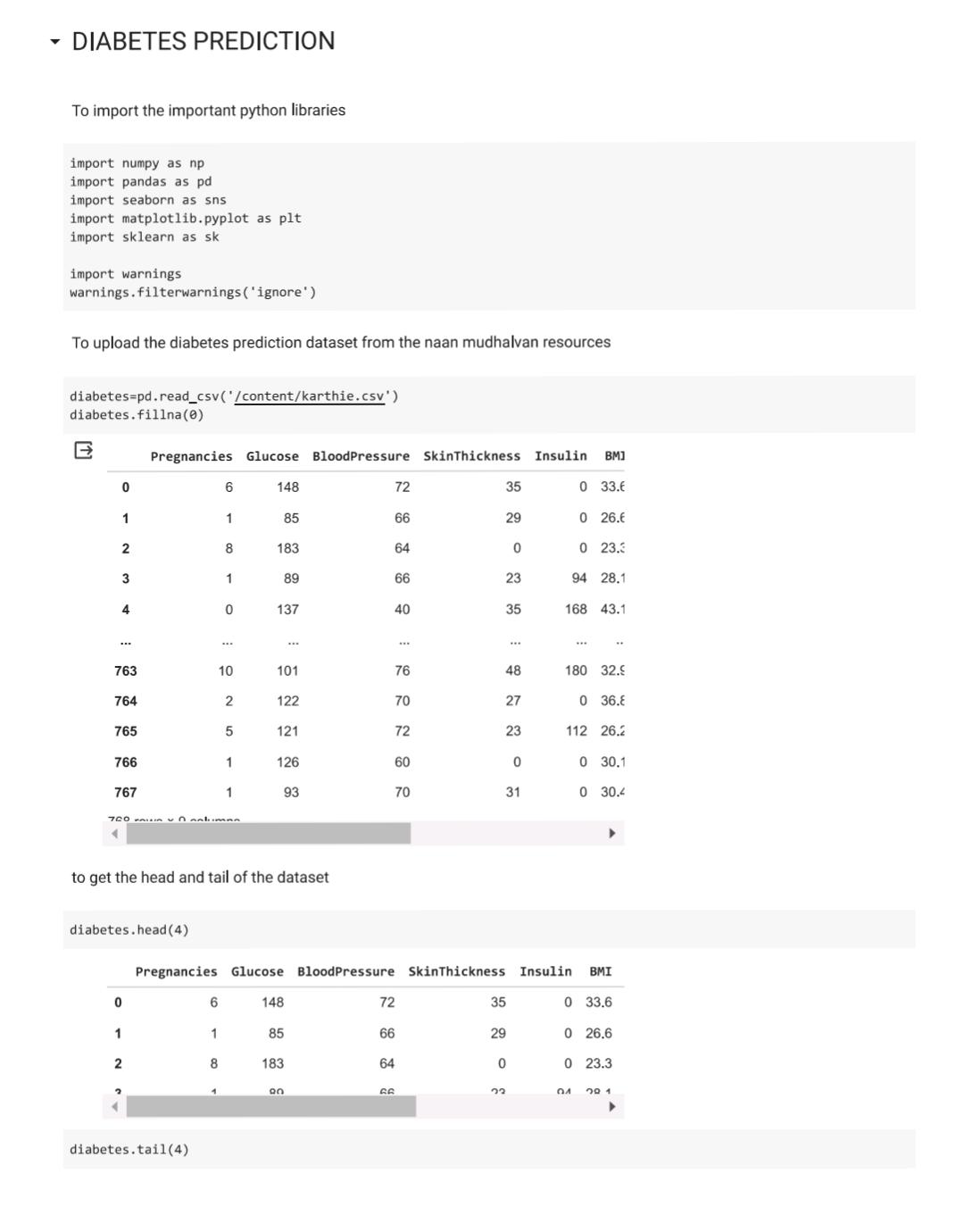
# Continuously monitor the model's performance and retrain it with new data

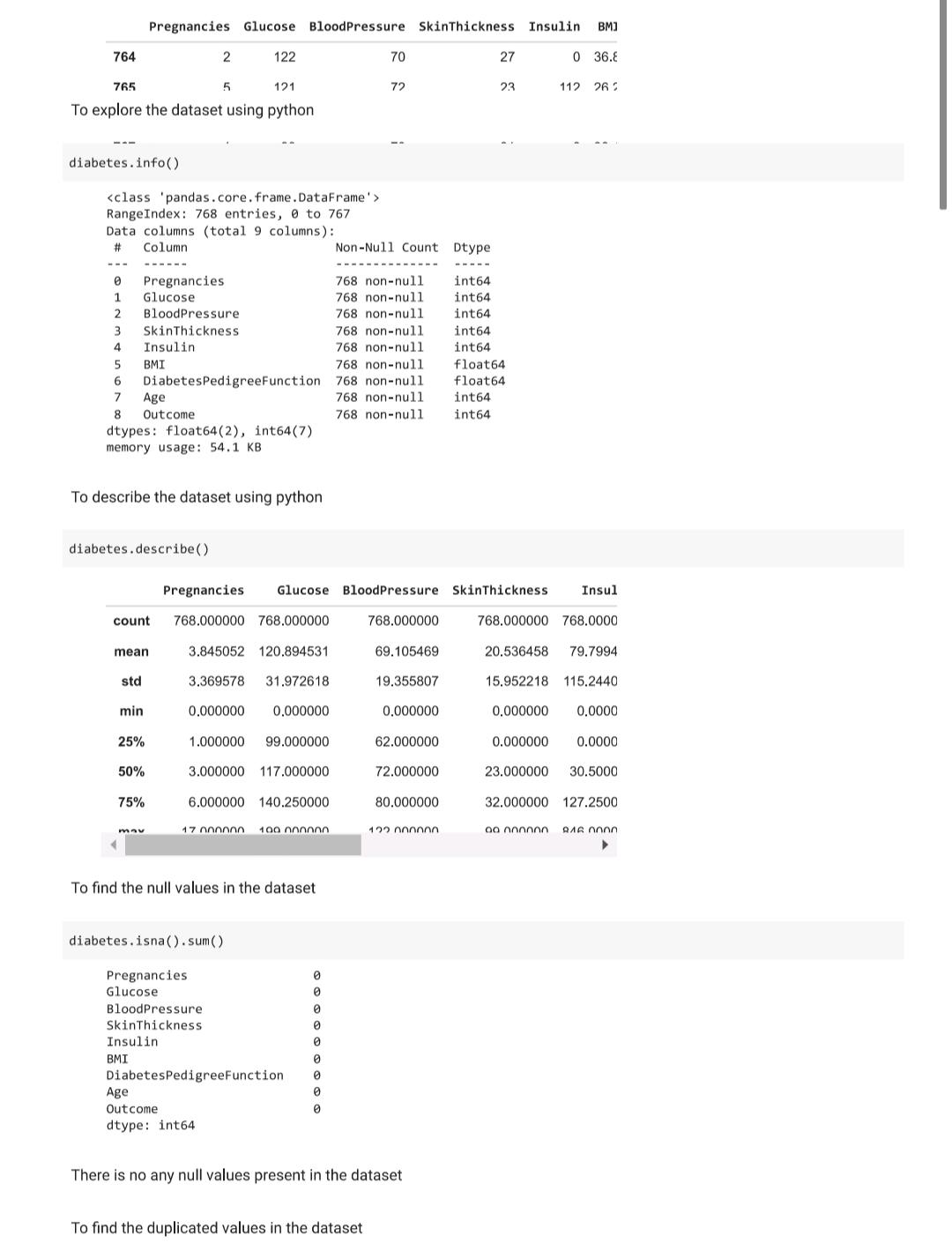
# Predicting diabetes risk for new input data

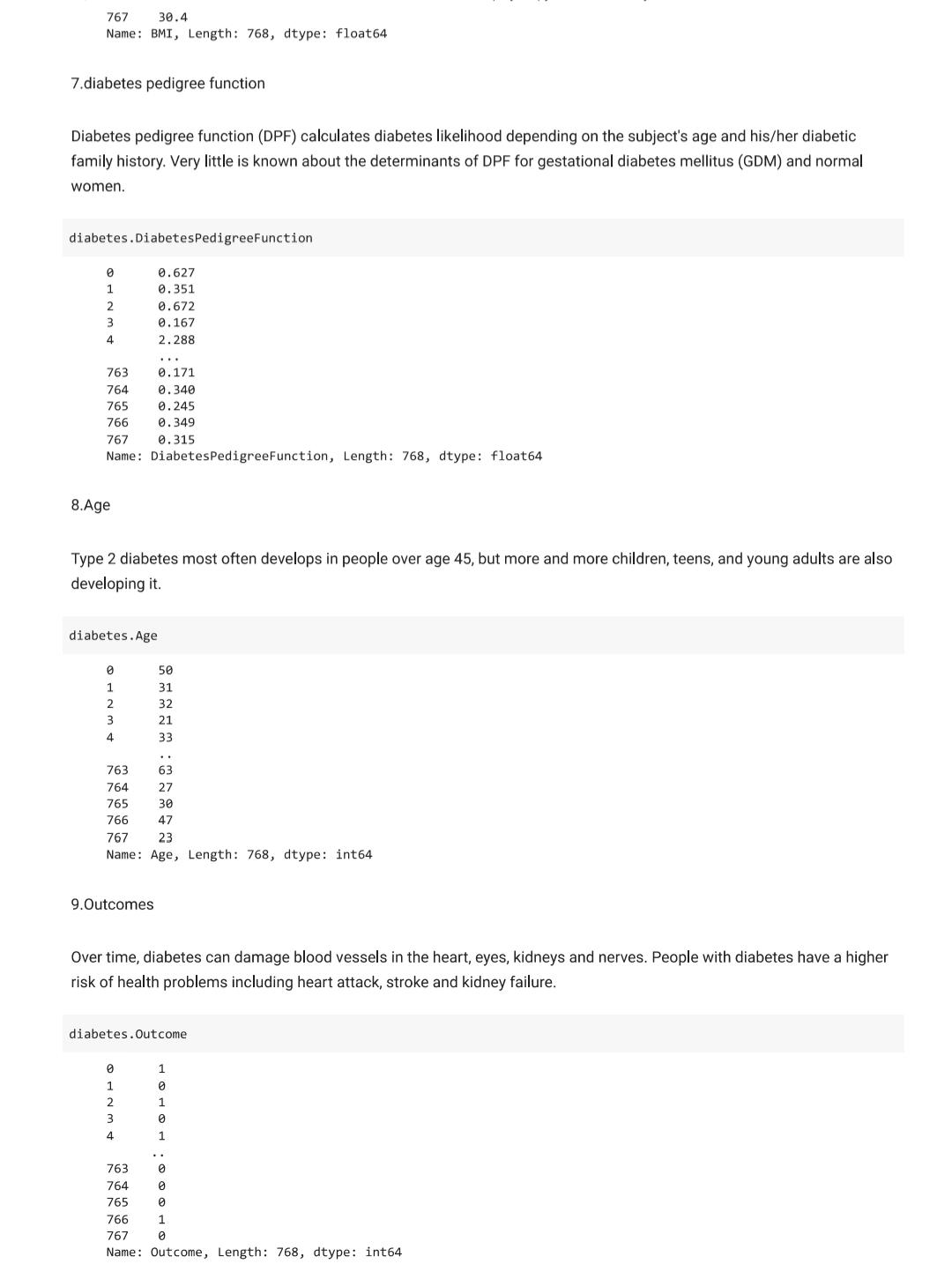
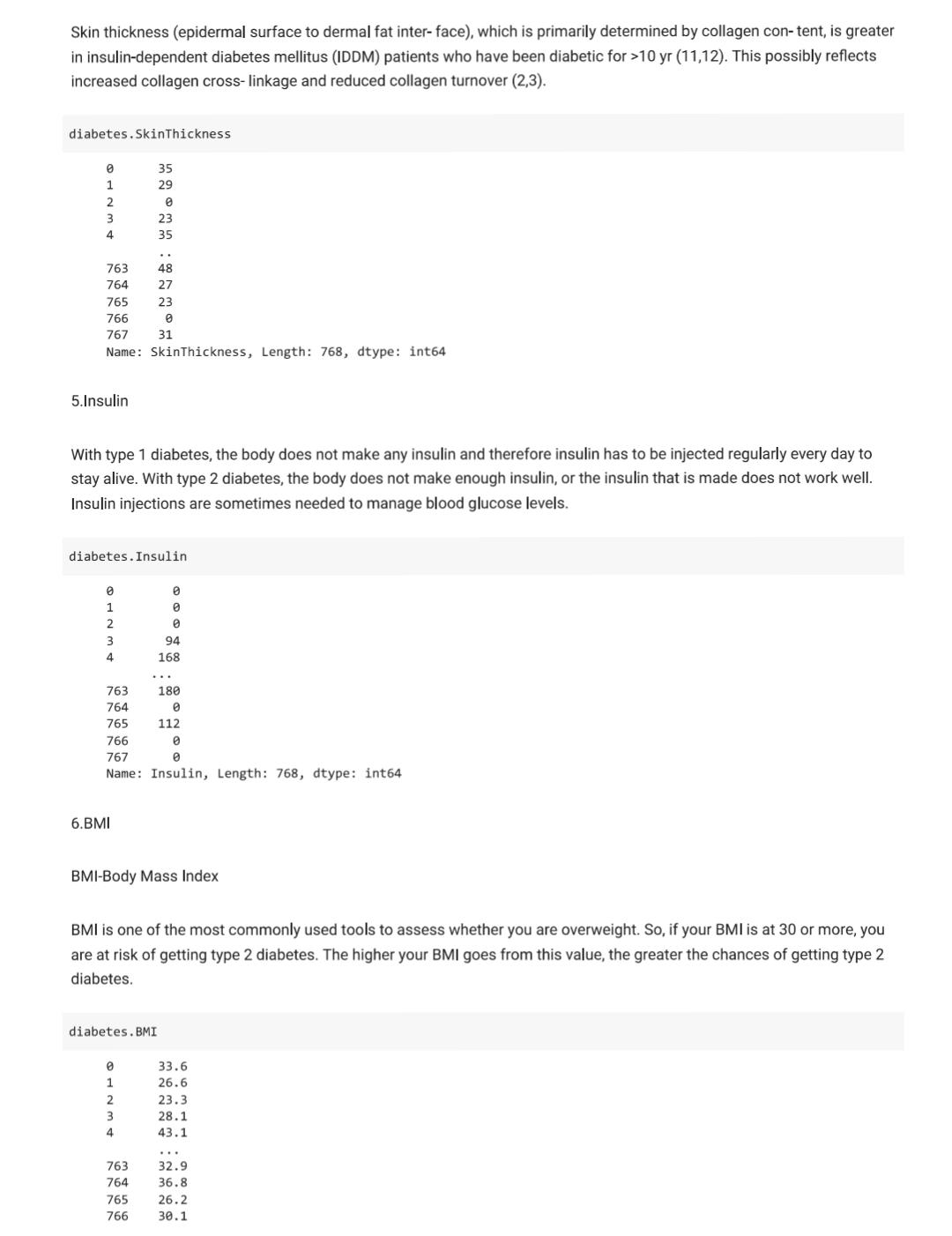
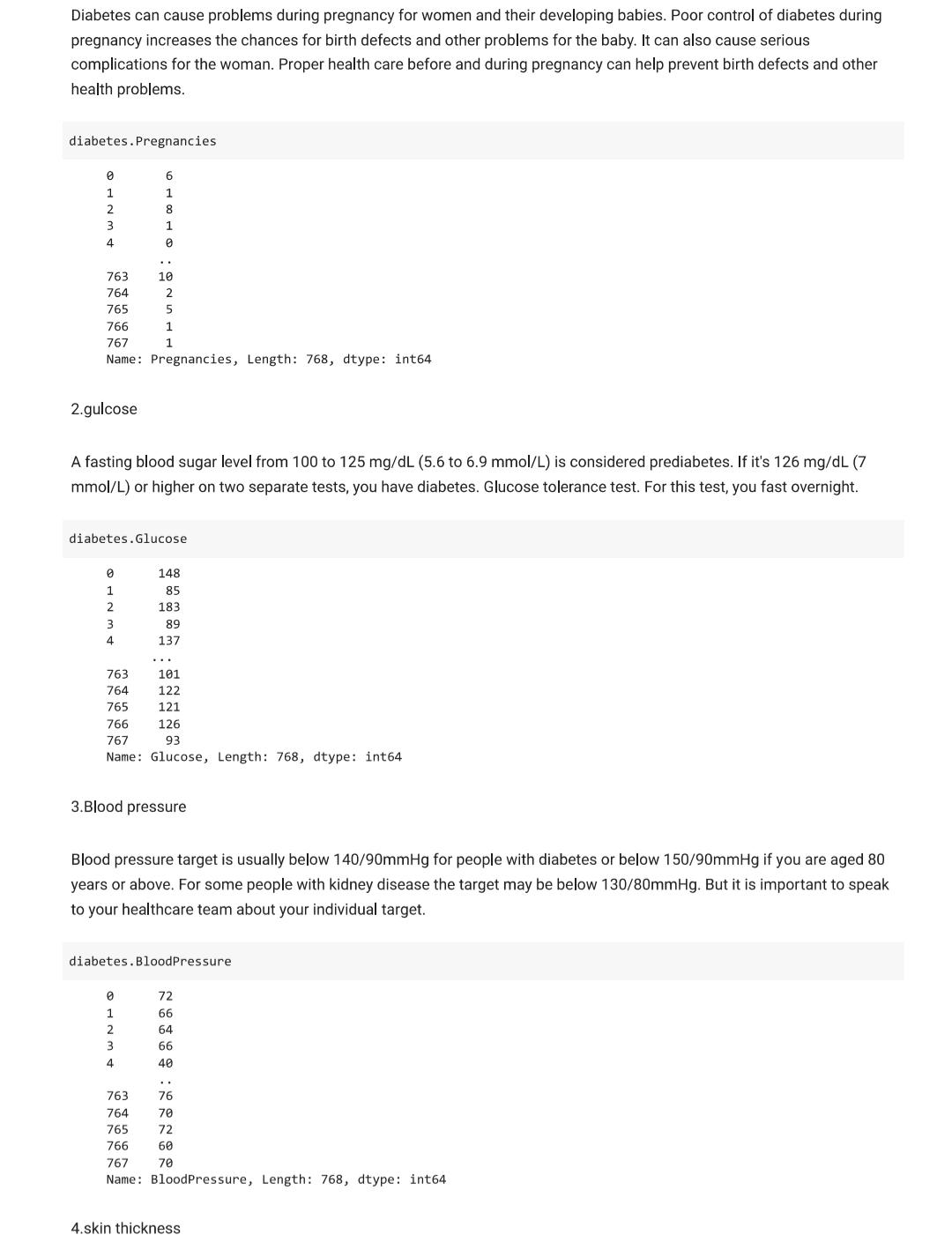
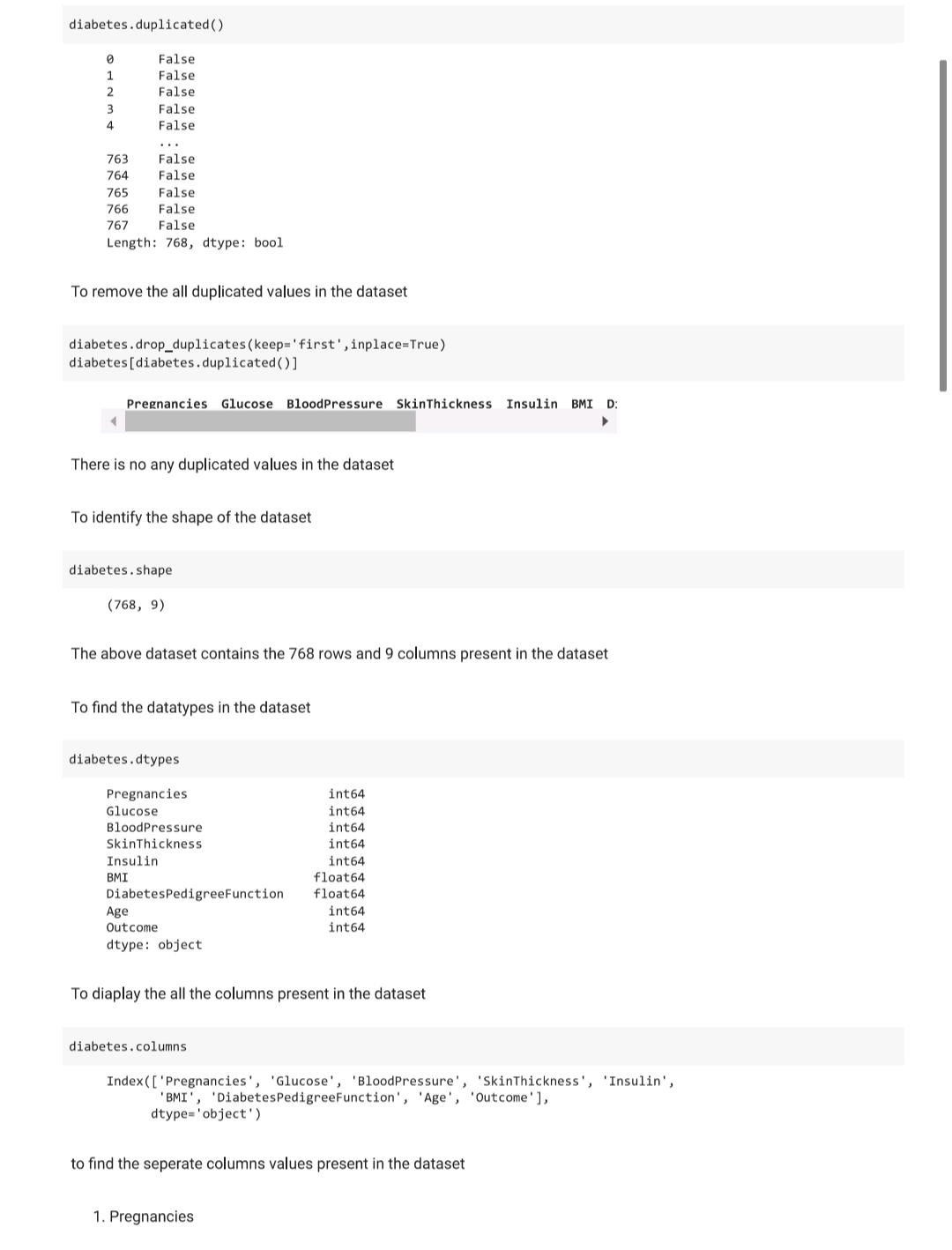
new\_data = pd.DataFrame({'Age': [35], 'Gender': ['Male'], 'Weight': [75], 'Height': [5.10], 'BloodPressure': [120], 'BloodSugarLevels': [100], 'FamilyHistoryDiabetes': [Yes]})

predicted\_risk = model.predict\_proba(new\_data)[0][1]

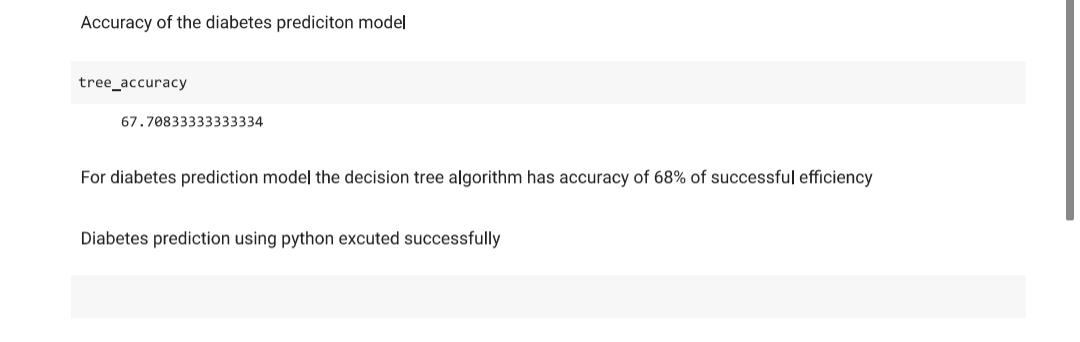
print("Predicted Diabetes Risk:", predicted\_risk)







ACCURACY:



NAME: S HEMACHANDIRAN

DEPARTMENT: BE ECE/ 3 RD YEAR

COLLAGE: PANIMALAR INSTITUTE OF TECHNOLOGY