AI based Diabetes Prediction System

PHASE 3: DEVELOPMENT PART – I

(Loading and preprocessing the dataset)

Creating python file:

Since python has more features and libraries for machine learning and Artificial Intelligence, I created a python file for this project.

Importing necessary libraries:

The development of a diabetes prediction system in Python requires the use of a variety of libraries and tools to handle data, build machine learning models, and create a user-friendly interface. The tools and libraries are:

NumPy:

NumPy is the fundamental library for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these arrays. For diabetes prediction, NumPy is essential for data manipulation and mathematical operations.

Pandas:

Pandas is foundational library that provides data structures and data analysis tools. It is used to handle and manipulate structured data, such as datasets containing patient information, medical records, and other relevant information. I can use Pandas to clean and preprocess the data for my prediction model.

Scikit-Learn:

Scikit-Learn is one of the most popular machine learning libraries in Python. It offers a wide range of tools for tasks like classification, regression, clustering, model selection, and dimensionality reduction. Scikit-Learn is particularly valuable for building and evaluating machine learning models to predict diabetes based on patient data.

Matplotlib and Seaborn:

These libraries are used for data visualization. Visualizing data can help us understand patterns, relationships, and potential features that may be important in predicting diabetes. Matplotlib and Seaborn allow us to create various types of plots, from simple scatter plots to more complex heatmaps.

TensorFlow and Keras:

If possible there is a chance to use deep learning models like TensorFlow and Keras. These libraries enable us to build and train deep learning models, which can be especially effective for complex diabetes prediction tasks that involve a large amount of data and intricate patterns.

Code:

Just some imports for usecase:

import math, time, datetime

import random as rd

Data Manipulation

import numpy as np

import pandas as pd

Visualization

import matplotlib.pyplot as plt

import missingno as msno

import seaborn as sns

plt.style.use('seaborn-whitegrid')

Preprocessing

from sklearn.preprocessing import OneHotEncoder, LabelEncoder, label\_binarize

Machine learning:

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

from sklearn.svm import LinearSVC

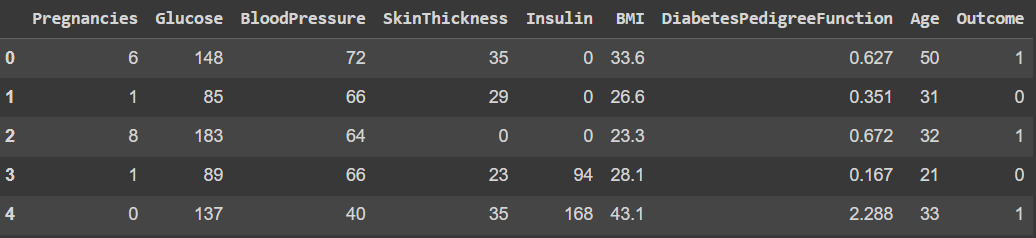
LOADING THE DATASET:

Loading the dataset to our python file using the pandas Dataframe.

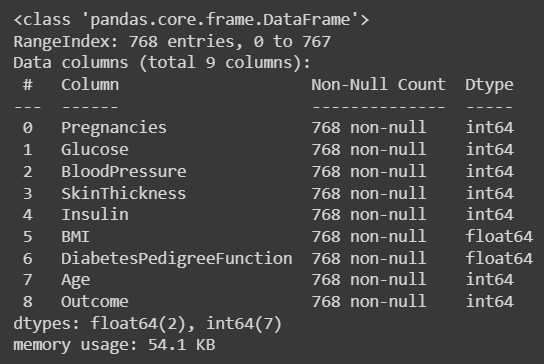
df = pd.read\_csv("/C:/Desktop/diabetes-data-set/diabetes.csv")

DATASET INSPECTION:

ds.head()

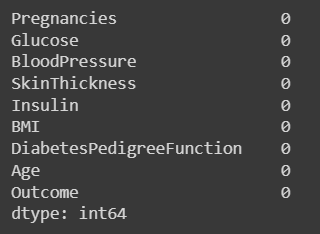


ds.info()



Checking missing values:

missing\_values = df.isnull().sum()



If so, handling missing values:

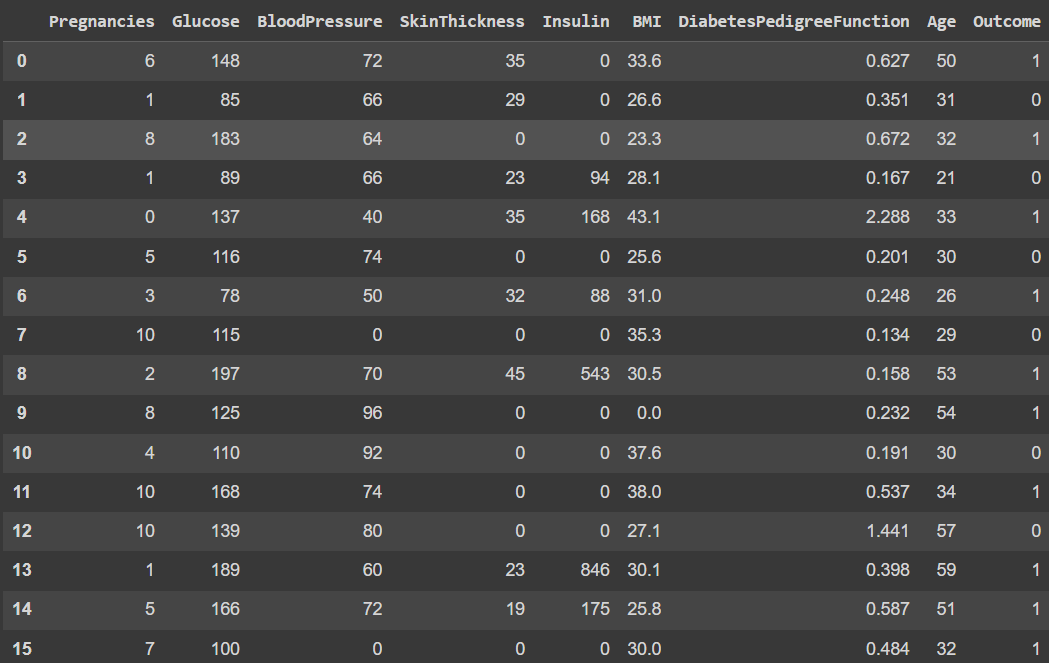
mean\_fill = df.fillna(df.mean())

df.fillna(mean\_fill, inplace=True)

Checking duplicate values:

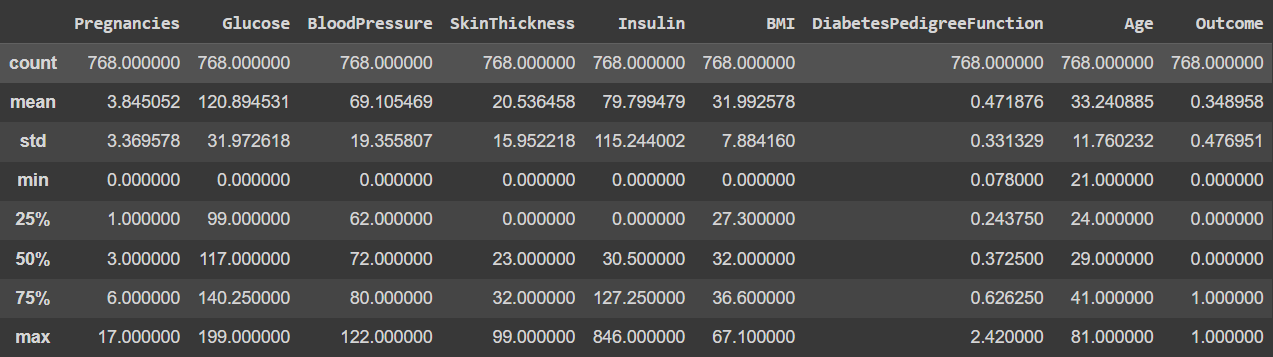
duplicate\_values = df.duplicated().sum()

df.drop\_duplicates(inplace=True)

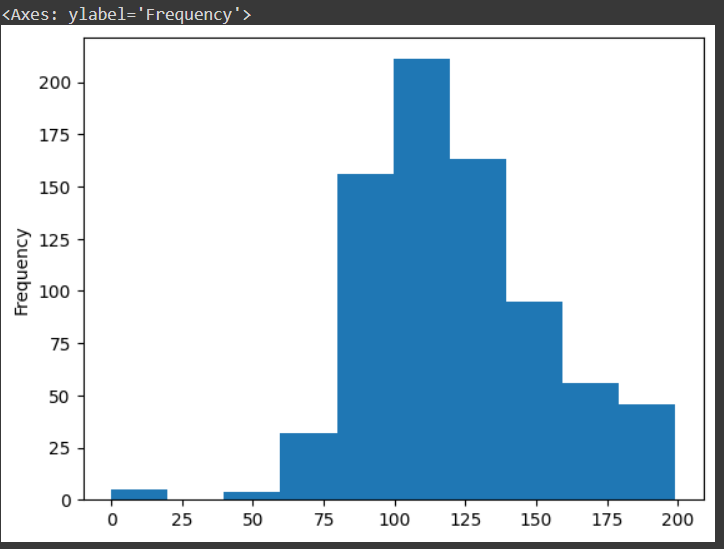


Data inspection:

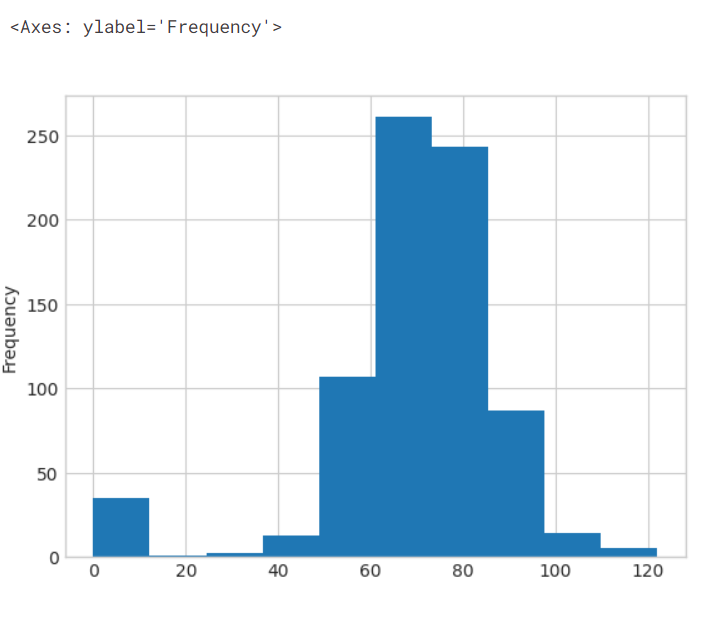
df.describe()



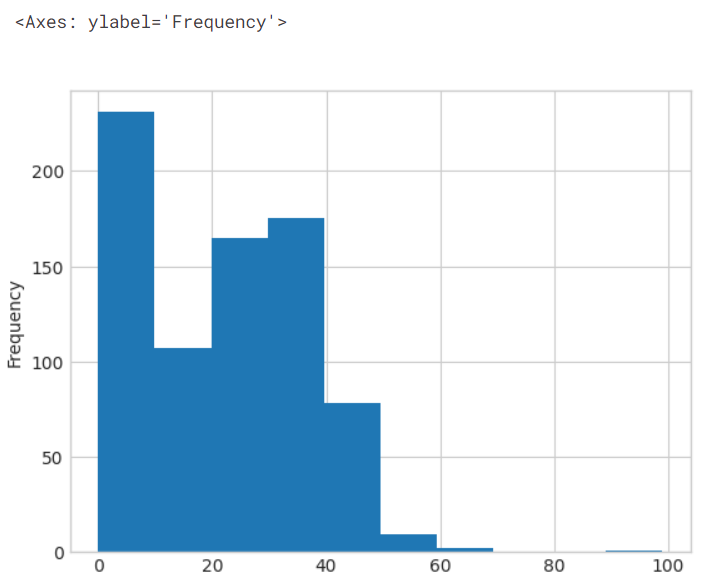
Data Analysis and Visualization:

df["Glucose"].plot.hist()

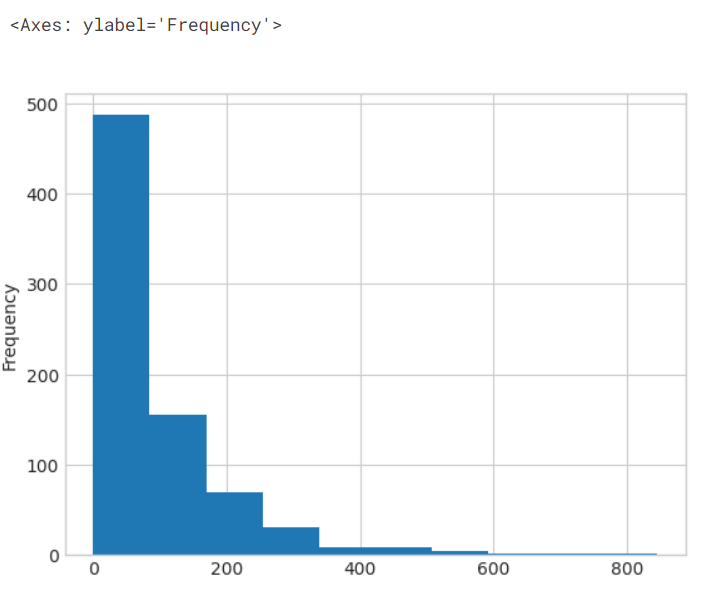
df["BloodPressure"].plot.hist()



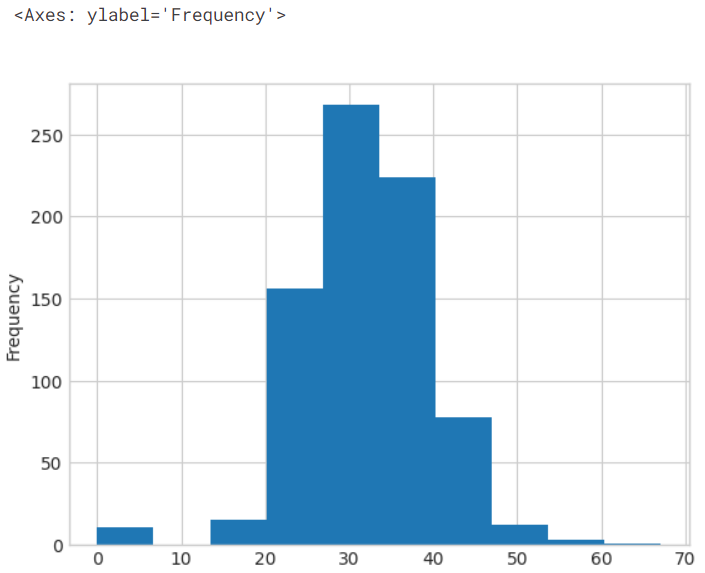
df["SkinThickness"].plot.hist()



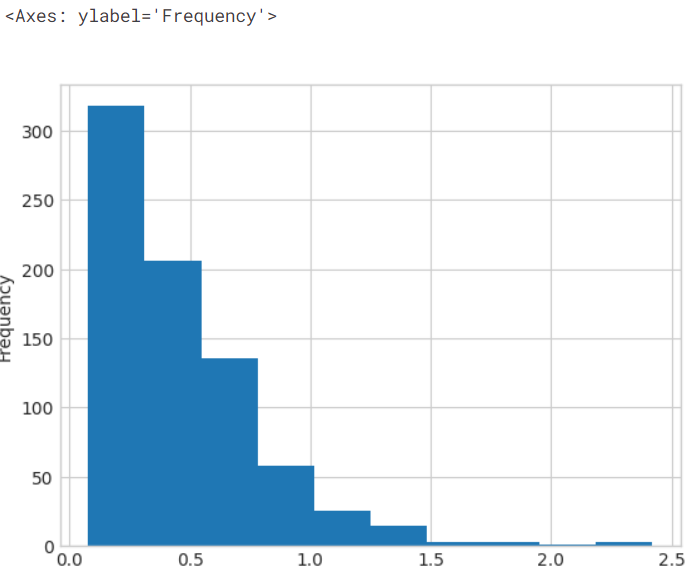
df["Insulin"].plot.hist()



df["BMI"].plot.hist()



df["DiabetesPedigreeFunction"].plot.hist()



Conclusion:

In this phase, I’ve loaded the diabetes dataset into my python program and pre-processed the dataset by cleaning the unwanted data, filled null values, etc., This cleaned dataset will be used for further machine learning models.