**PHASE 3 DOCUMENT SUBMISSION**

**PROJECT TITLE:**  AI BASED DIABETS PREDICTION SYSTEM

**PHASE 3:** DEVLOPMENT PART 1

**TOPIIC:** Build Your Project by loading and pre-processing the dataset



**INTRODUCTION**:

**1:Problem Statement**: It should begin by highlighting the growing concern of diabetes as a global health issue. Mention the increasing prevalence of diabetes and the impact it has on individuals' lives.

**2.Purpose**: Explain that the AI-based system is designed to address this issue by providing an advanced tool for predicting diabetes in individuals.

**3.Technology Overview**: Briefly describe the use of artificial intelligence and machine learning in healthcare and how it has transformed disease prediction and prevention.

**4.Scope**: Explain the scope of the system, such as the types of data it uses, the target audience (patients, healthcare professionals, etc.), and the outcomes it aims to achieve.

**5.Key Features:** Mention some of the key features of the system, such as its ability to analyze medical records, lifestyle data, and genetic information to provide personalized predictions.

**6.Benefits**: Discuss the potential benefits of using such a system, including early detection, personalized healthcare recommendations, and improved patient outcomes.

**PRE-PROCESSING DATASET :**

**Data Collection:**

Gather a dataset containing relevant features (e.g., age, BMI, family history) and target labels (diabetes status, such as diabetic or non-diabetic).

**Data Cleaning:**

Handle missing values by imputation or removal.

Remove duplicates.

**Data Exploration:**

Analyze the dataset's statistics and distributions.

Visualize the data to gain insights.

**Feature Selection:**

Choose the most relevant features for prediction.

**Feature Engineering:**

Create new features if necessary.

Normalize or scale features.

**Data Splitting:**

Divide the dataset into training, validation, and test sets.

**Handling Class Imbalance:**

Address any imbalance in the target classes through techniques like oversampling or undersampling.

**Model Selection:**

Choose an appropriate machine learning or deep learning model for diabetes prediction.

**Training**:

Train the model on the training data.

**Hyperparameter Tuning:**

Optimize model hyperparameters for better performance.

**Validation**:

Evaluate the model on the validation set to fine-tune it.

**Testing**:

Assess the model's performance on the test set to ensure it generalizes well.

**Model Evaluation:**

Measure performance using metrics like accuracy, precision, recall, F1-score, and ROC AUC.

**Deployment**:

Deploy the AI model in a suitable environment, like a web application or mobile app.

**DATASET:**

**In 1:**

# This Python 3 environment comes with many helpful analytics libraries installed

# It is defined by the kaggle/python Docker image: <https://github.com/kaggle/docker-python>

# For example, here’s several helpful packages to load

Import numpy as np # linear algebra

Import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

Import seaborn as sns

Import matplotlib.pyplot as plt

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

From sklearn import linear\_model

From sklearn.metrics import r2\_score,confusion\_matrix

# import warnings

Import warnings

# filter warnings

Warnings.filterwarnings(‘ignore’)

# Input data files are available in the read-only “../input/” directory

# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

Import os

For dirname, \_, filenames in os.walk(‘/kaggle/input’):

For filename in filenames:

Print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using “Save & Run All”

# You can also write temporary files to /kaggle/temp/, but they won’t be saved outside of the current session

**Load and Check Data:**

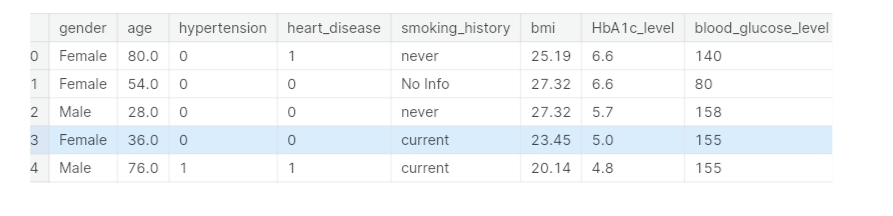
**In 2:**

**data = pd.read\_csv("/kaggle/input/diabetes-prediction-dataset/diabetes\_prediction\_dataset.csv")**

**In 3:**

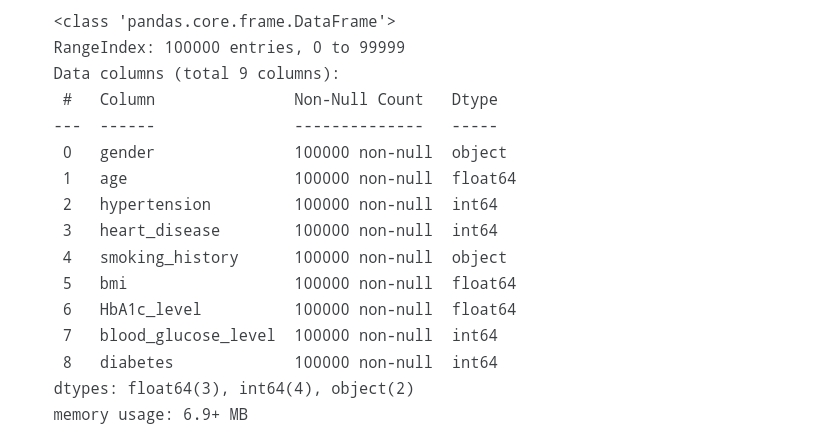
Data.head()

**Output:**

****

**In 4:**

**Data.info()**

****

**In 5:**

Def bar\_plot(variable):

‘’’

Input: variable ex: “Sex”

Output: bar plot & value count

‘’’

# get feature

Var = data[variable]

# count number of categorical variable(value/sample)

Var\_value = var.value\_counts()

# visualize

Plt.figure(figsize=(6,3))

Plt.bar(var\_value.index, var\_value,width= 1/(var.unique().size))

plt.xticks(var\_value.index, var\_value.index.values)

plt.ylabel("Frequency")

plt.title(variable)

plt.show()

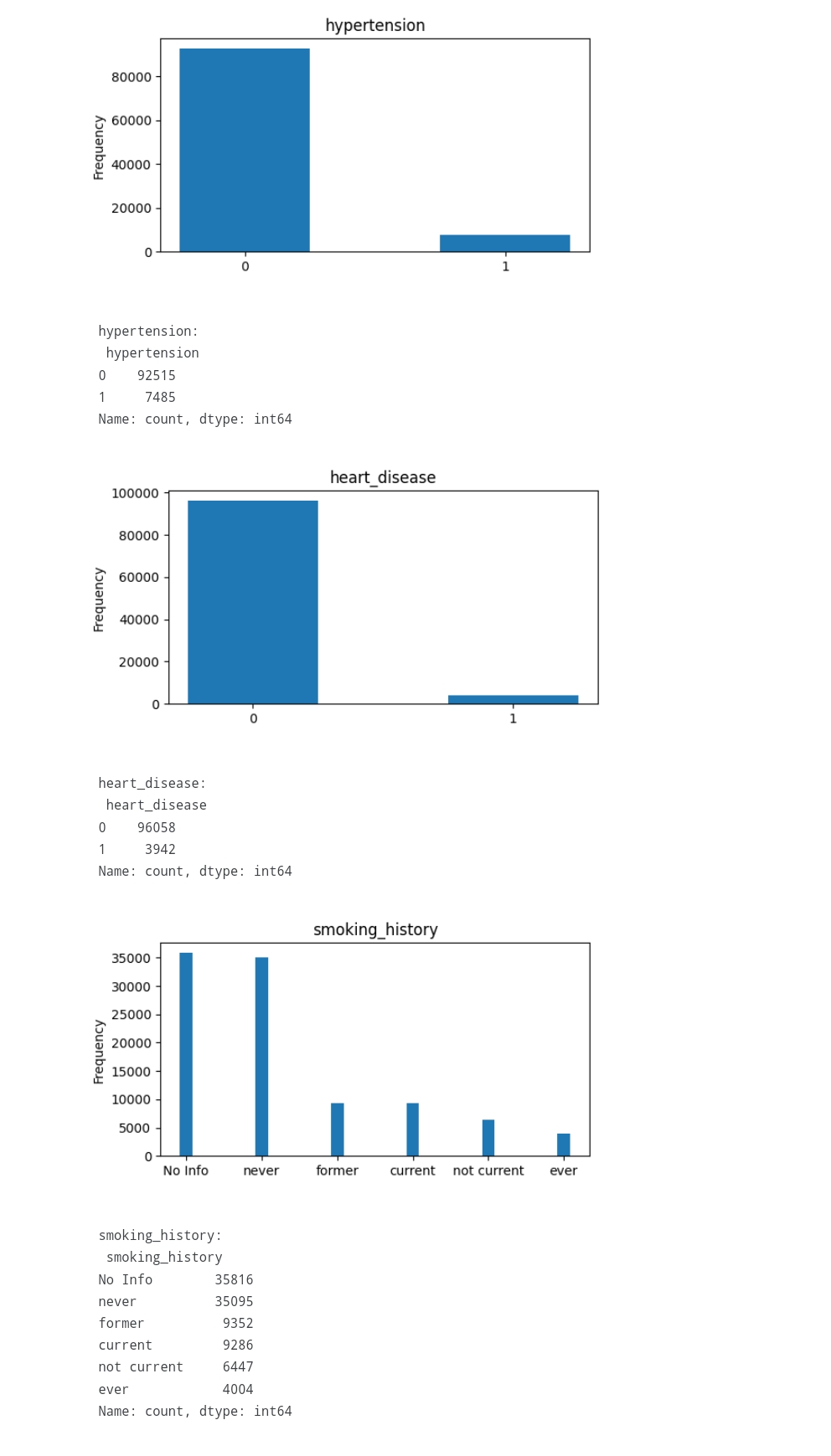
print("{}: \n {}".format(variable,var\_value))

**In 6:**

Category1 = [“hypertension”,”heart\_disease”,”smoking\_history”]

For c in category1:

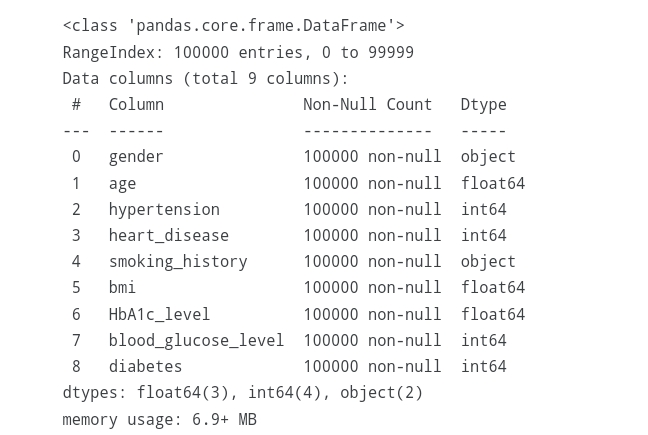
Bar\_plot(c)



**Change the Data Type:**

**Making data types integer**

**In 7:**

Data.info()

**In 8:**

Pd.unique(data.smoking\_history)

**Output:**

Array([‘never’, ‘No Info’, ‘current’, ‘former’, ‘ever’, ‘not current’],

Dtype=object)

**In 9:**

Pd.unique(data.gender)

**Output:**

Array([‘Female’, ‘Male’, ‘Other’], dtype=object)

**In 10:**

Def change\_string\_to\_int(column):

Variables=pd.unique(data[column])

For item in range(variables.size):

Data[column]=[item if each==variables[item] else each for each in data[column]]

Return data[column]

**In 11:**

Data[“gender”]=change\_string\_to\_int(“gender”)

**In 12:**

Data[“smoking\_history”]=change\_string\_to\_int(“smoking\_history”)

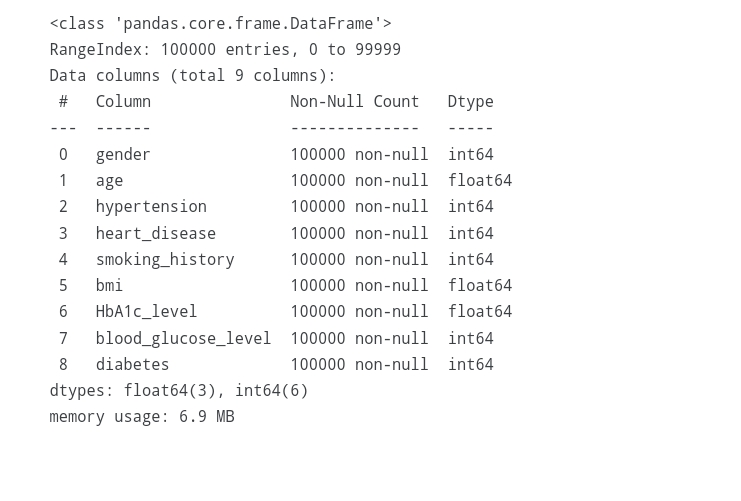
**In 13:**

Data.head()

**Output:**

**In 14:**

Data.head()

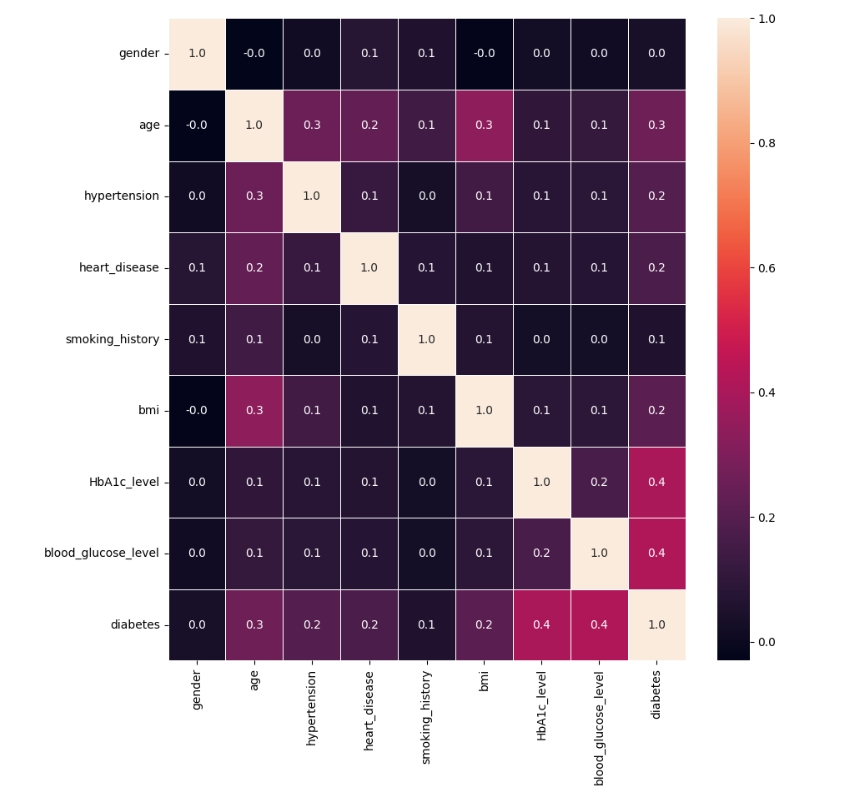
**Output 15:**

**Removing from the Data:**

**In 15:**

F,ax = plt.subplots(figsize=(10, 10))

Sns.heatmap(data.corr(numeric\_only=True), annot=True, linewidths=.5, fmt= ‘.1f’,ax=ax,)

Plt.show()

**Conclusion :**

Contribution of the Explainable AI in Diabetes Prediction system makes it easy for the end-user to understand the AI systems’ complex working. It provides a human-centered interface to the user.