DATA SCIENCE

Name: K Vinoth Kumar

Roll No: CH.EN.U4ARE22011

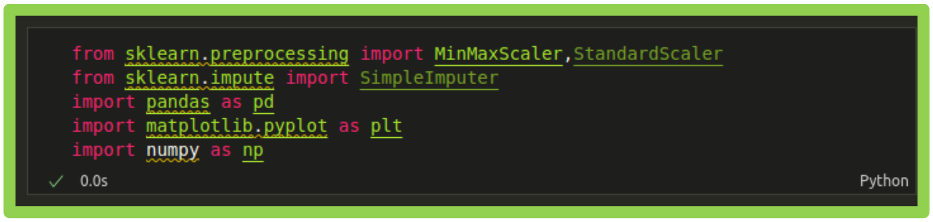
AIM :

The aim of this code is to perform comprehensive data preprocessing and exploratory data analysis on a diabetes dataset.

The Code used in this assignment can be found here

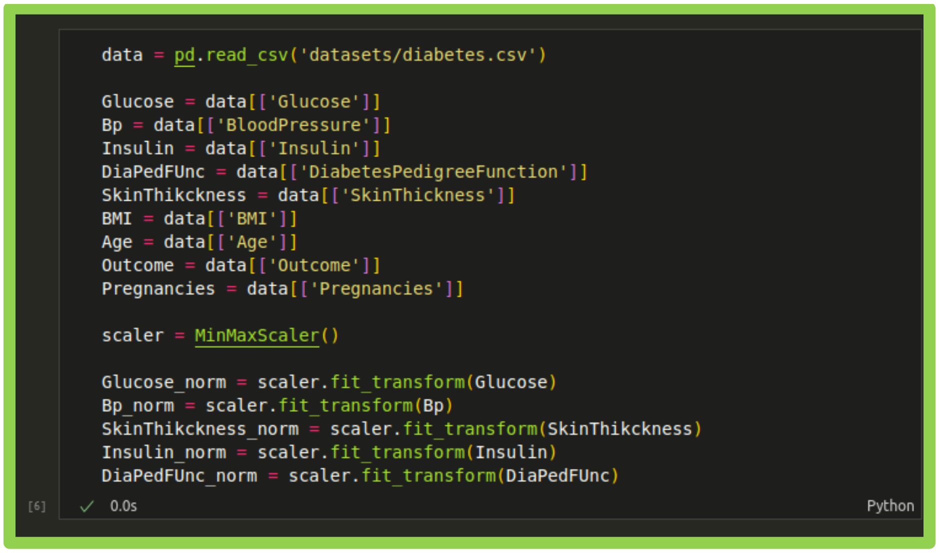
CODE :

Import Libraries for performing numerical operations, data manipulation & analysis, for creating plots.



```
from sklearn.preprocessing import MinMaxScaler,StandardScaler
from sklearn.impute import SimpleImputer
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

Data – Dataset loaded from ‘diabetes.csv’



```
data = pd.read_csv('datasets/diabetes.csv')

Glucose = data[['Glucose']]
Bp = data[['BloodPressure']]
Insulin = data[['Insulin']]
DiaPedFUnc = data[['DiabetesPedigreeFunction']]
SkinThikckness = data[['SkinThickness']]
BMI = data[['BMI']]
Age = data[['Age']]
Outcome = data[['Outcome']]
Pregnancies = data[['Pregnancies']]

scaler = MinMaxScaler()

Glucose_norm = scaler.fit_transform(Glucose)
Bp_norm = scaler.fit_transform(Bp)
SkinThikckness_norm = scaler.fit_transform(SkinThikckness)
Insulin_norm = scaler.fit_transform(Insulin)
DiaPedFUnc_norm = scaler.fit_transform(DiaPedFUnc)
```

Converted the Outcome data to a flattened NumPy array and categorized the outcomes into 'Yes' (positive) and 'No' (negative).

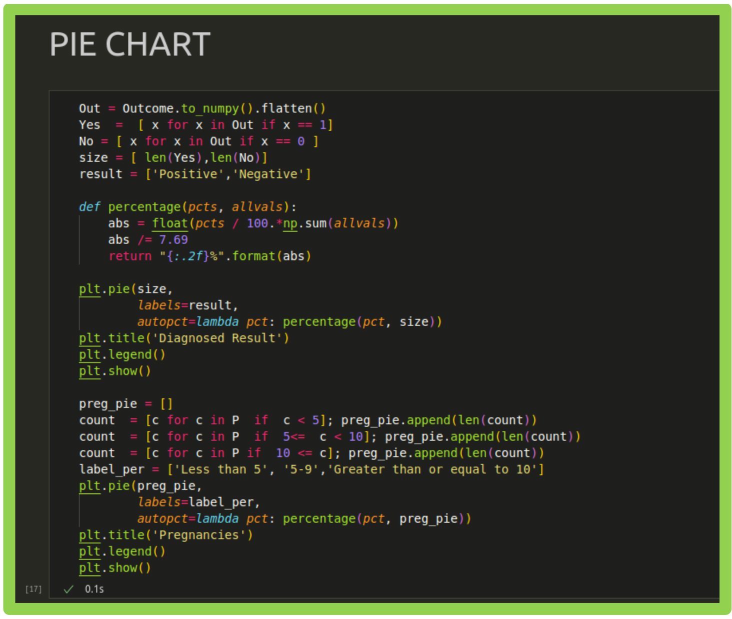
Created and displayed a pie chart for the diagnosed results.

Categorize the number of pregnancies into three categories: less than 5, 5-9, and 10 or more

Created and displayed a pie chart for the number of pregnancies:

A function is defined to calculate the percentage for the pie chart labels:

Created a pie chart with the sizes and labels specified, and used the percentage function for the label percentages.

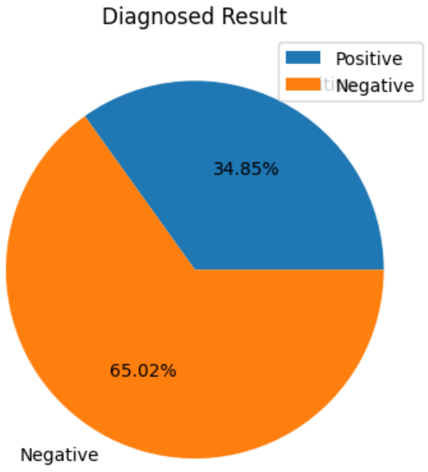


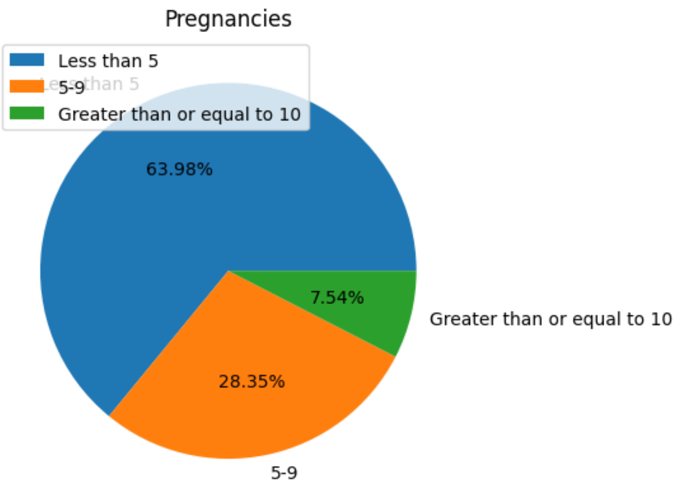
PIE CHART

```
Out = Outcome.to_numpy().flatten()
Yes  =  [ x for x in Out if x == 1]
No = [ x for x in Out if x == 0 ]
size = [ len(Yes),len(No)]
result = ['Positive','Negative']

def percentage(pcts, allvals):
    abs = float(pcts / 100.*np.sum(allvals))
    abs /= 7.69
    return "{:.2f}%".format(abs)

plt.pie(size,
        labels=result,
        autopct=lambda pct: percentage(pct, size))
plt.title('Diagnosed Result')
plt.legend()
plt.show()

preg_pie = []
count  = [c for c in P  if  c < 5]; preg_pie.append(len(count))
count  = [c for c in P  if  5<=  c < 10]; preg_pie.append(len(count))
count  = [c for c in P if  10 <= c]; preg_pie.append(len(count))
label_per = ['Less than 5', '5-9','Greater than or equal to 10']
plt.pie(preg_pie,
        labels=label_per,
        autopct=lambda pct: percentage(pct, preg_pie))
plt.title('Pregnancies')
plt.legend()
plt.show()
```

[17] ✓ 0.1s

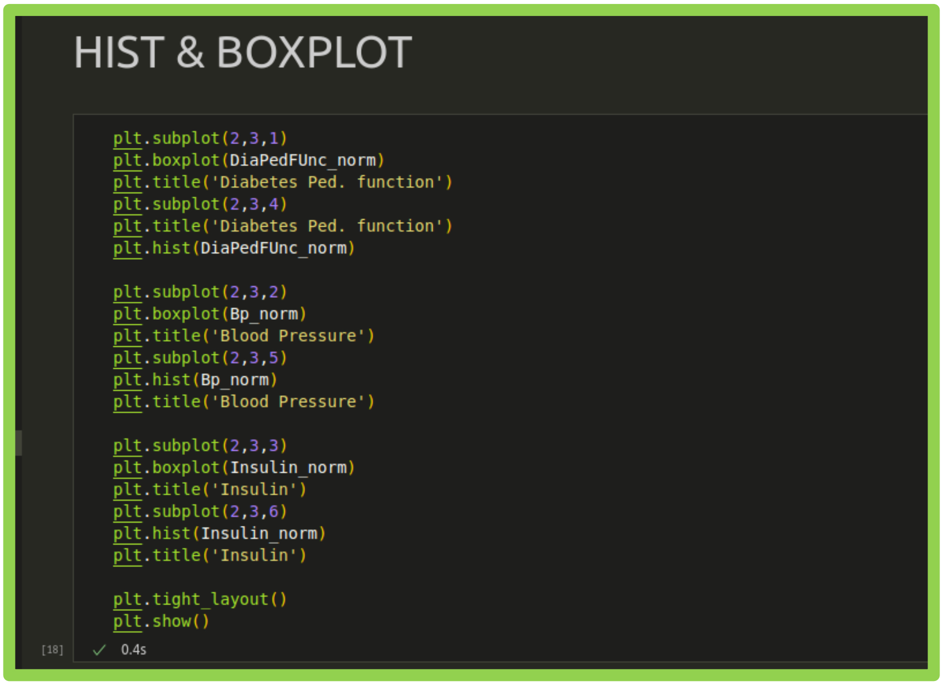
4





Plotted Boxplot for Diabetes Pedigree Function, Blood Pressure and Insulin.

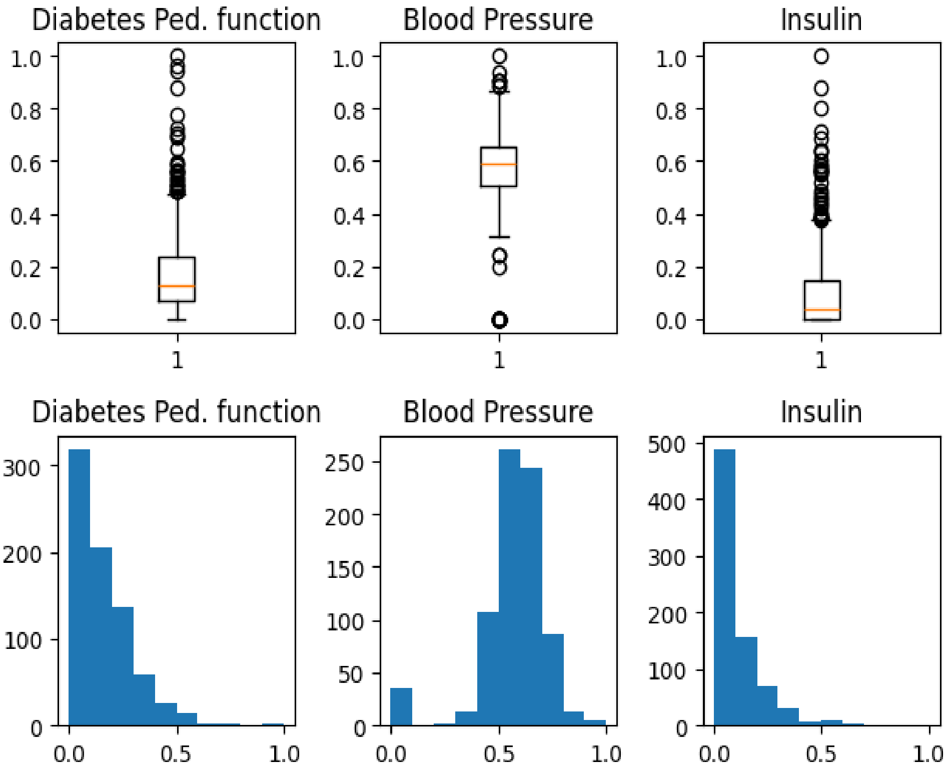
Plotted Insulin for Diabetes Pedigree Function, Blood Pressure and Insulin.



HIST & BOXPLOT

```
plt.subplot(2,3,1)
plt.boxplot(DiaPedFUnc_norm)
plt.title('Diabetes Ped. function')
plt.subplot(2,3,4)
plt.title('Diabetes Ped. function')
plt.hist(DiaPedFUnc_norm)

plt.subplot(2,3,2)
plt.boxplot(Bp_norm)
plt.title('Blood Pressure')
plt.subplot(2,3,5)
plt.hist(Bp_norm)
plt.title('Blood Pressure')

plt.subplot(2,3,3)
plt.boxplot(Insulin_norm)
plt.title('Insulin')
plt.subplot(2,3,6)
plt.hist(Insulin_norm)
plt.title('Insulin')

plt.tight_layout()
plt.show()
```

[18] ✓ 0.4s



RESULT :

The results showcase successful execution of various data preprocessing and visualization techniques, providing a comprehensive understanding of the dataset's structure, relationships, and distributions.

This thorough analysis is foundational for building effective data models and deriving meaningful insights from the data.

LEARNING OUTCOMES :

Gained proficiency in loading datasets into pandas DataFrames and extracting specific columns for focused analysis.

Acquired the ability to create various types of plots using matplotlib, including pie charts, histograms and box plots.

Enhanced the ability to perform a comprehensive data analysis by combining multiple preprocessing and visualization techniques to gain deeper insights into the dataset.

Understood how to visualize relationships between different features, which is essential for exploratory data analysis and identifying patterns or trends in the data.