Harsh Seksaria

2048011

2 - MDS

Machine Learning Lab 12 & 13

```
Hiearchical Clustering & DBSCAN

30 April, 2021
```

CHRIST (Deemed to be University)

In [10]:
```python
#Import libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [27]:
```python
#Reading dataset
fl = pd.read_csv('../diabetes.csv')
fl.head()
```

Out[27]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Pre-processing

In [28]:
```
#Null values
fl.isnull().sum()
```

Out[28]:
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Therefore, we can say that there is no null value in the dataset.

In [29]:
```
#Duplicate data
fl.duplicated().any()
```

Out[29]:
```
False
```

So there are no duplicated values.

Exploratory Data Analysis

In [30]:
```
#Dataset shape
print("No. of rows: ", fl.shape[0])
print("No. of columns: ", fl.shape[1])
```

```
No. of rows:  768
No. of columns:  9
```

In [31]:
```
#Dataset info
fl.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

In [32]:
```
#Dataset describe
fl.describe()
```

Out[32]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

In [33]:
```
#Correlation matrix
fl.corr()
```

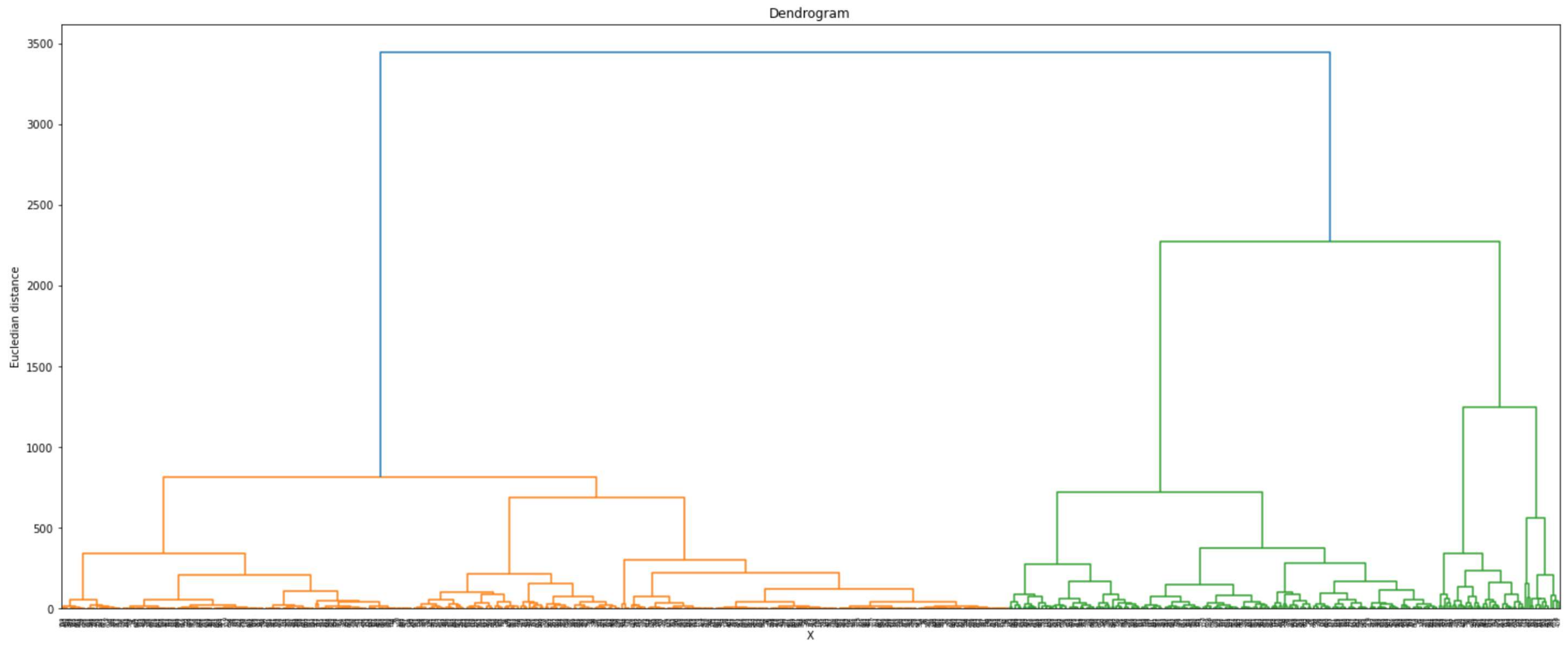
Out[33]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000000	0.129459	0.141282	-0.081672	-0.073535	0.017683	-0.033523	0.544341	0.221898
Glucose	0.129459	1.000000	0.152590	0.057328	0.331357	0.221071	0.137337	0.263514	0.466581
BloodPressure	0.141282	0.152590	1.000000	0.207371	0.088933	0.281805	0.041265	0.239528	0.065068
SkinThickness	-0.081672	0.057328	0.207371	1.000000	0.436783	0.392573	0.183928	-0.113970	0.074752
Insulin	-0.073535	0.331357	0.088933	0.436783	1.000000	0.197859	0.185071	-0.042163	0.130548
BMI	0.017683	0.221071	0.281805	0.392573	0.197859	1.000000	0.140647	0.036242	0.292695
DiabetesPedigreeFunction	-0.033523	0.137337	0.041265	0.183928	0.185071	0.140647	1.000000	0.033561	0.173844
Age	0.544341	0.263514	0.239528	-0.113970	-0.042163	0.036242	0.033561	1.000000	0.238356
Outcome	0.221898	0.466581	0.065068	0.074752	0.130548	0.292695	0.173844	0.238356	1.000000

Heirarchical Clustering

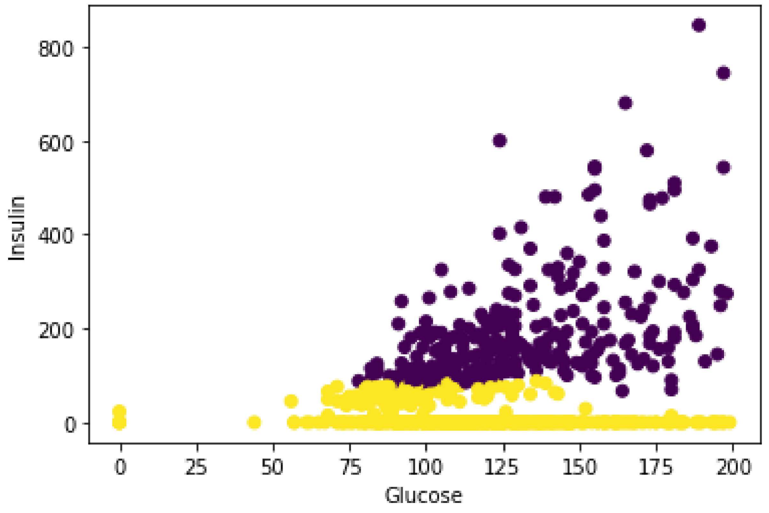
In [34]:
```
x = fl.loc[:,['Glucose', 'Insulin']]
```

In [35]:
```python
# Visualising the dendrogram
import scipy.cluster.hierarchy as sch

fig = plt.figure(figsize=(25, 10))
dendrogram=sch.dendrogram(sch.linkage(x,method='ward'))
plt.title("Dendrogram")
plt.xlabel("X")
plt.ylabel("Eucledian distance")
plt.show()
```

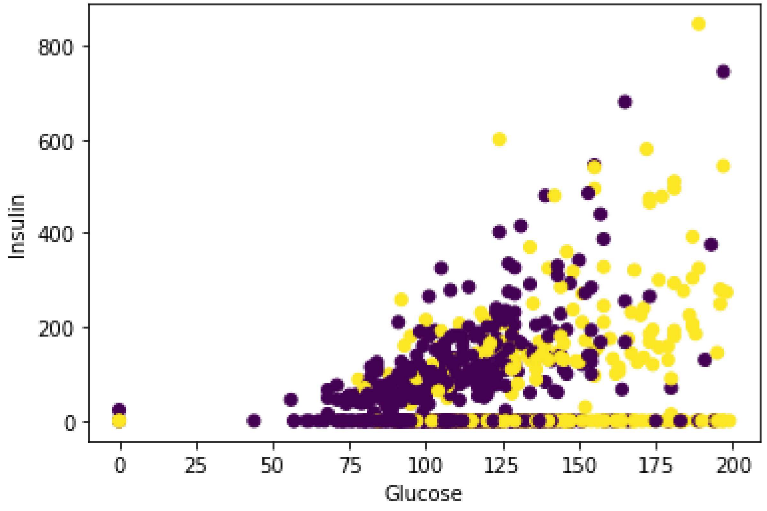


In [52]:
```
from sklearn.cluster import AgglomerativeClustering

hiearchical_clustering = AgglomerativeClustering(n_clusters = 2, affinity = 'euclidean', linkage = 'ward')
predicted_hc = hiearchical_clustering.fit_predict(x)
predicted_hc
x['Label'] = predicted_hc
```

In [45]:
```
plt.scatter(x['Glucose'], x['Insulin'], c = predicted_hc)
plt.xlabel('Glucose')
plt.ylabel('Insulin')
plt.show()
```



In [47]:
```
plt.scatter(fl['Glucose'], fl['Insulin'], c = fl['Outcome'])
plt.xlabel('Glucose')
plt.ylabel('Insulin')
plt.show()
```



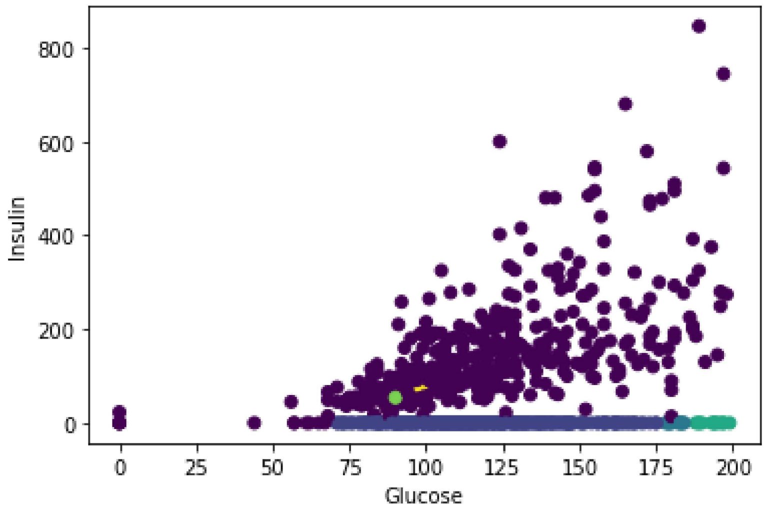
DBSCAN

In [50]:
```
from sklearn.cluster import DBSCAN

db_clustering = DBSCAN(eps=3,min_samples=5,metric='euclidean')
predicted_db = db_clustering.fit_predict(x)
predicted_db
```

Out[50]:
```
array([ 0,  0,  1, -1, -1,  0, -1,  0, -1,  0,  0,  0,  0, -1, -1,  0, -1,
        0, -1, -1, -1,  0,  2,  0, -1, -1,  0, -1, -1,  0,  0, -1,  3,  0,
        0, -1,  0,  0,  0, -1, -1,  0,  0, -1,  0,  1,  0,  0,  0,  0, -1,
       -1, -1, -1, -1,  0, -1, -1,  0, -1,  0,  0, -1, -1,  0,  0,  0,  0,
       -1, -1, -1, -1,  0, -1,  0, -1, -1,  0,  0,  0,  0,  0, -1,  0,  0,
       -1,  0, -1, -1,  0,  0, -1, -1,  0, -1, -1,  0, -1, -1, -1,  0,  0,
        0, -1,  0, -1,  0, -1, -1, -1, -1, -1, -1,  0, -1,  0,  0,  0,  0,
       -1, -1,  0, -1,  0,  0, -1, -1, -1, -1,  0, -1,  0, -1,  0, -1, -1,
       -1, -1,  0, -1,  0,  0, -1,  0, -1,  0, -1, -1,  0,  0, -1,  0, -1,
       -1,  2,  0, -1, -1,  3, -1,  0, -1, -1,  0,  0, -1,  0,  0,  0, -1,
        0, -1,  0, -1, -1, -1,  0, -1,  0,  0,  0, -1, -1,  0,  0,  2, -1,
       -1, -1, -1,  0, -1,  0,  0,  0, -1,  0, -1, -1, -1,  0,  0,  0, -1,
       -1,  0, -1,  0, -1,  1,  0,  0,  1, -1, -1, -1, -1, -1,  0,  0, -1,
        0,  0, -1, -1, -1,  0,  0, -1, -1,  0, -1, -1,  0, -1,  0, -1,  1,
        0,  0,  0, -1,  0, -1, -1,  1,  0, -1, -1,  0,  0,  0,  3,  0, -1,
        0,  0,  0, -1, -1, -1,  0,  0,  0,  0, -1,  0,  0,  0,  0,  0, -1,
        0, -1,  0, -1,  0, -1,  0, -1,  0, -1, -1,  0,  0, -1, -1, -1, -1,
       -1, -1, -1, -1, -1,  0, -1, -1, -1, -1,  0,  0, -1, -1,  0,  0, -1,
       -1, -1, -1, -1,  0, -1, -1, -1,  0, -1, -1,  1, -1,  2, -1,  0,  0,
       -1,  0, -1, -1,  1, -1, -1,  0,  3,  1,  0, -1, -1,  0,  0, -1,  1,
       -1, -1, -1,  0,  0, -1, -1,  0, -1, -1,  0,  0, -1, -1,  0,  0, -1,
        0,  3, -1, -1,  0,  0,  0, -1,  4,  0,  0, -1, -1, -1, -1, -1, -1,
       -1, -1,  4, -1,  0, -1, -1,  0, -1, -1, -1, -1,  0,  0, -1,  4, -1,
        0, -1, -1,  0, -1, -1,  0,  0,  2,  0,  0, -1,  0,  0, -1,  0,  0,
        2, -1,  0, -1, -1, -1, -1, -1,  0,  0,  0, -1, -1, -1, -1,  0, -1,
       -1,  0, -1, -1, -1,  0, -1, -1,  0,  0,  0,  0,  0,  0,  0,  2, -1,
       -1,  0,  0, -1, -1, -1, -1, -1, -1,  0, -1,  0, -1,  0,  0, -1, -1,
       -1, -1,  0, -1,  0,  0, -1, -1, -1,  0, -1,  0,  0,  0,  0,  0,  0,
       -1, -1, -1,  0, -1,  0, -1, -1,  0, -1, -1, -1,  0,  2, -1,  0,  0,
       -1,  0,  0,  0, -1, -1, -1, -1,  0, -1, -1,  0,  0, -1, -1, -1,  0,
        0, -1,  0,  0,  4, -1, -1,  0,  0, -1, -1, -1,  0,  0,  0,  0,  4,
       -1, -1,  0, -1,  0, -1,  0, -1,  0,  0, -1, -1, -1, -1, -1,  0, -1,
       -1, -1, -1, -1, -1,  2,  0, -1,  0, -1, -1, -1,  0,  0,  0,  0,  0,
       -1, -1, -1,  0, -1, -1, -1, -1, -1,  0,  0, -1, -1, -1, -1, -1,  0,
        0,  2,  0,  0,  0,  0, -1,  0,  0,  0, -1,  0,  0, -1,  0, -1, -1,
       -1, -1, -1,  0, -1,  0,  0,  0, -1,  1,  0, -1, -1, -1, -1, -1, -1,
       -1,  0, -1,  0,  0, -1,  0,  0, -1,  0,  1, -1,  0,  3,  0,  0,  0,
        0,  0, -1,  0, -1,  0,  0,  0, -1, -1, -1, -1,  0,  0,  0, -1, -1,
       -1, -1, -1,  0, -1, -1, -1,  0, -1, -1, -1, -1,  0, -1,  0,  2, -1,
       -1,  0, -1,  0,  0, -1, -1, -1,  0, -1, -1,  0,  2,  0,  0,  0, -1,
       -1,  0, -1,  0,  0, -1,  0,  0, -1, -1,  0,  0, -1, -1,  0, -1, -1,
        0, -1,  0, -1,  0,  0,  0, -1,  0,  0, -1,  0, -1, -1, -1,  0, -1,
        0, -1, -1,  0, -1,  0,  0, -1, -1, -1,  0,  0, -1,  0,  0,  0, -1,
        0, -1, -1,  0,  0, -1, -1, -1,  0, -1, -1, -1,  0, -1, -1,  0, -1,
       -1,  0,  0, -1,  0, -1,  0, -1,  0,  0,  0,  2, -1,  0,  0, -1,  0,
       -1,  0,  0], dtype=int64)
```

In [51]:

```python
plt.scatter(x['Glucose'], x['Insulin'], c = predicted_db)
plt.xlabel('Glucose')
plt.ylabel('Insulin')
plt.show()
```



In [54]:

```
correct = []
for i in range(0,767):
    if x['Label'][i] == fl['Outcome'][i]:
        correct.append(1)
    else:
        correct.append(0)
correct[0:10] # -> if we find correctly 1, if not 0



print("Hierarchical Clustering Accuracy : ", (correct.count(1)/x['Label'].size)*100)
```

```
Hierarchical Clustering Accuracy :  40.36458333333333
```