```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

data = pd.read_csv('diabetes_new.csv')

df=pd.DataFrame(data)

df
```

```
      Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin
BMI  \
0               2      138             62             35        0
33.6
1               0       84             82             31      125
38.2
2               0      145              0              0        0
44.2
3               0      135             68             42      250
42.3
4               1      139             62             41      480
40.7
...           ...      ...            ...            ...      ...   ..
.
1995            2       75             64             24       55
29.7
1996            8      179             72             42      130
32.7
1997            6       85             78              0        0
31.2
1998            0      129            110             46      130
67.1
1999            2       81             72             15       76
30.1

      DiabetesPedigreeFunction  Age  Outcome
0                        0.127   47        1
1                        0.233   23        0
2                        0.630   31        1
3                        0.365   24        1
4                        0.536   21        0
...                        ...  ...      ...
1995                     0.370   33        0
1996                     0.719   36        1
1997                     0.382   42        0
1998                     0.319   26        1
1999                     0.547   25        0

[2000 rows x 9 columns]
```

```
df.drop(['Pregnancies'],axis=1,inplace=True)
```

```
df
```

```
      Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0         138             62             35        0  33.6
1          84             82             31      125  38.2
2         145              0              0        0  44.2
3         135             68             42      250  42.3
4         139             62             41      480  40.7
...       ...            ...            ...      ...   ...
1995       75             64             24       55  29.7
1996      179             72             42      130  32.7
1997       85             78              0        0  31.2
1998      129            110             46      130  67.1
1999       81             72             15       76  30.1

      DiabetesPedigreeFunction  Age  Outcome
0                        0.127   47        1
1                        0.233   23        0
2                        0.630   31        1
3                        0.365   24        1
4                        0.536   21        0
...                        ...  ...      ...
1995                     0.370   33        0
1996                     0.719   36        1
1997                     0.382   42        0
1998                     0.319   26        1
1999                     0.547   25        0

[2000 rows x 8 columns]
```

```
data.columns
```

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',
'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

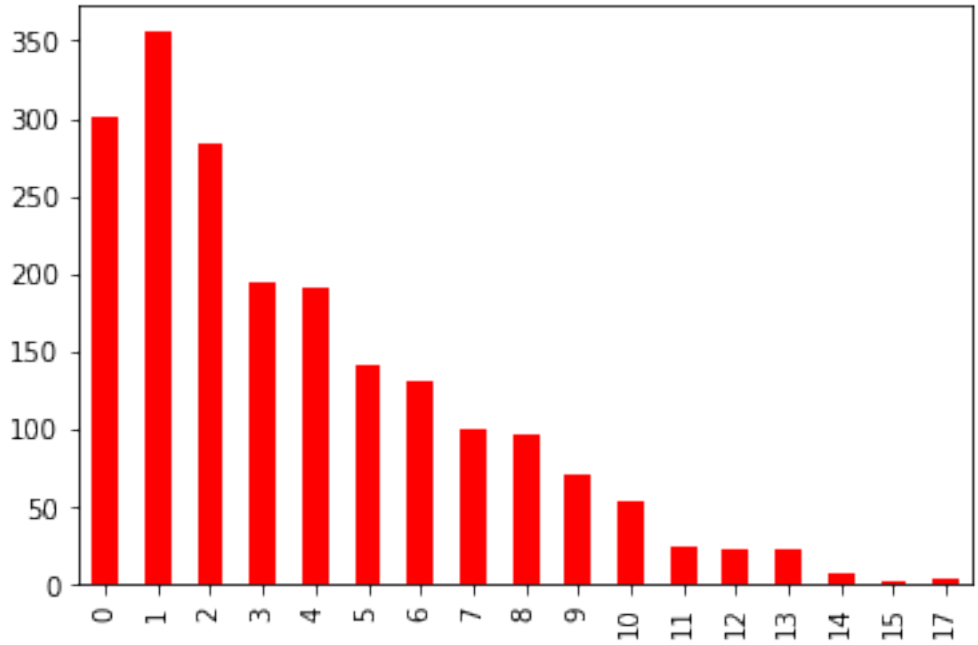
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               2000 non-null   int64
 1   Glucose                   2000 non-null   int64
 2   BloodPressure             2000 non-null   int64
 3   SkinThickness             2000 non-null   int64
 4   Insulin                   2000 non-null   int64
```

```
 5   BMI                       2000 non-null   float64
 6   DiabetesPedigreeFunction  2000 non-null   float64
 7   Age                       2000 non-null   int64
 8   Outcome                   2000 non-null   int64
dtypes: float64(2), int64(7)
memory usage: 140.8 KB
```

```
pd.value_counts(data["Pregnancies"]).sort_index().plot.bar(color="red"
)
```

```
<AxesSubplot:>
```




```
df.columns
```

```
Index(['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI',
       'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

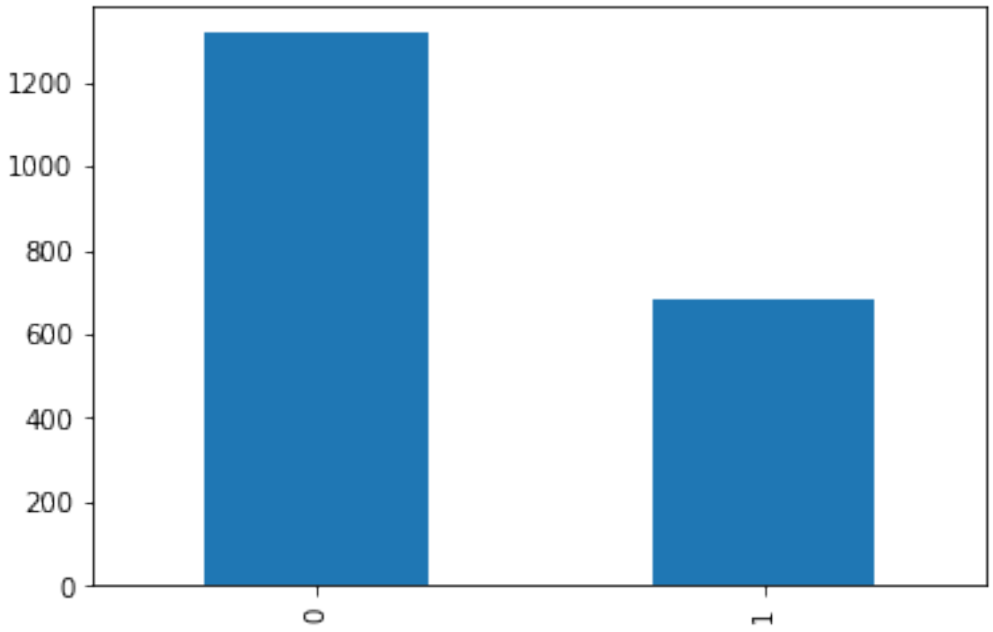
```
df["Glucose"].unique()
```

```
array([138,  84, 145, 135, 139, 173,  99, 194,  83,  89, 125,  80,
166,
       110,  81, 195, 154, 117,   0,  94,  96,  75, 180, 130, 120,
91,
       163, 122, 103, 102,  90, 111, 133, 106, 171, 159, 146,  71,
105,
       101,  88, 176, 150,  73, 187, 100,  44, 141, 114, 109,  95,
126,
       129,  79,  62, 131, 112, 113,  74, 137, 136, 107, 123, 134,
142,
       144,  92,  93, 151,  85, 155,  76, 160, 124,  78,  97, 162,
```

```
132,
       118, 170, 128, 108,  57, 147, 156, 153, 188, 152, 104, 148,
87,
       179, 143, 119, 181, 158, 196, 184, 140, 177, 197, 164, 165,
86,
       193, 191, 161, 167,  77, 115, 182, 157, 178, 116,  61, 189,
98,
       127,  82,  72, 168, 172, 175,  68, 186, 198, 121,  67, 183,
174,
       199,  56, 169, 149,  65, 190], dtype=int64)
```

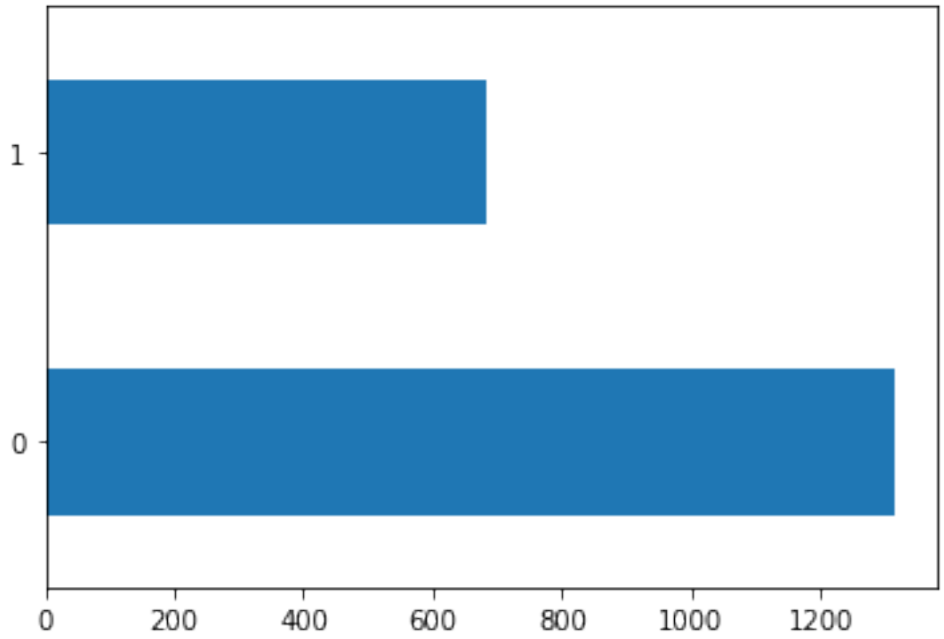
```
df["Outcome"].value_counts().plot.bar()
```

```
<AxesSubplot:>
```



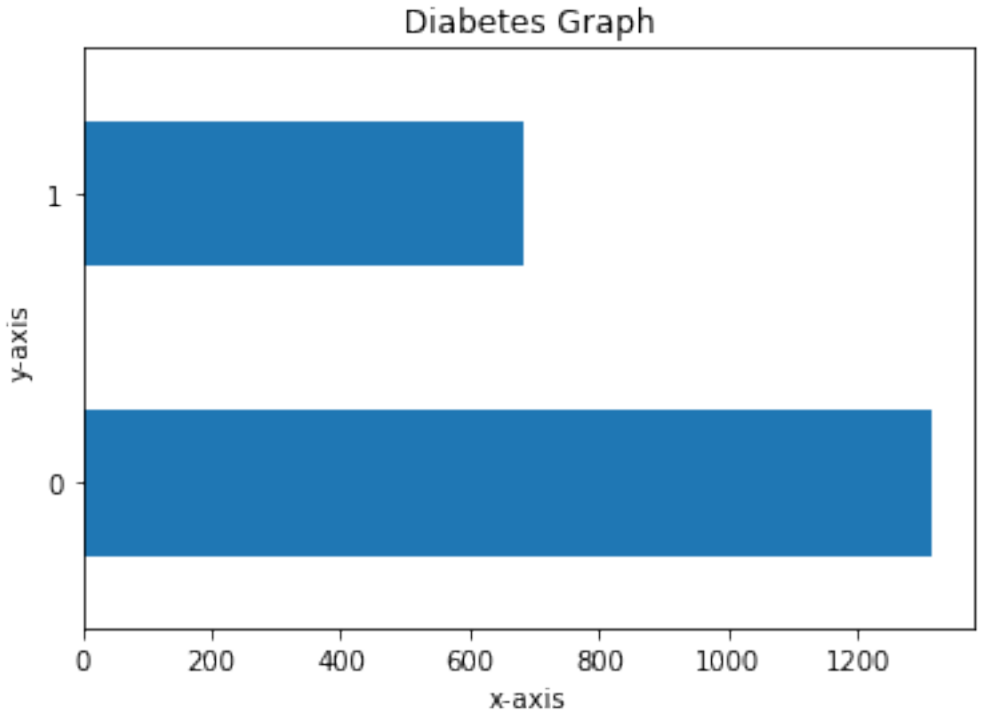

```
df["Outcome"].value_counts().plot.barh()
```

```
<AxesSubplot:>
```



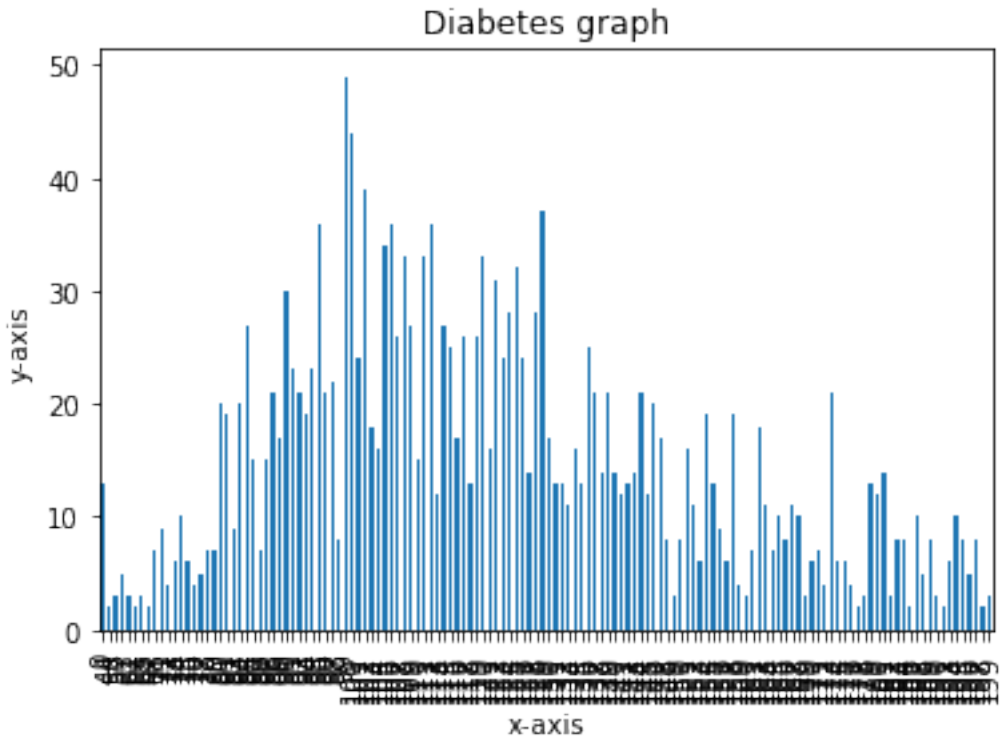

```
df["Outcome"].value_counts().plot.barh()
plt.title("Diabetes Graph")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
```

```
Text(0, 0.5, 'y-axis')
```



```
pd.value_counts(data["Glucose"]).sort_index().plot.bar("color=green")
plt.title("Diabetes graph")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
```

```
Text(0, 0.5, 'y-axis')
```




```
X=df[['Glucose','BloodPressure','SkinThickness','Insulin','BMI','Diabe
tesPedigreeFunction','Age']]
```

```
X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 7 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Glucose                   2000 non-null   int64
 1   BloodPressure             2000 non-null   int64
 2   SkinThickness             2000 non-null   int64
 3   Insulin                   2000 non-null   int64
 4   BMI                       2000 non-null   float64
 5   DiabetesPedigreeFunction  2000 non-null   float64
 6   Age                       2000 non-null   int64
dtypes: float64(2), int64(5)
memory usage: 109.5 KB
```

```
Y=data[['Outcome']]
```

```
Y.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 1 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Outcome  2000 non-null   int64
dtypes: int64(1)
memory usage: 15.8 KB
```

```
from sklearn.model_selection import train_test_split

X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.25,rand
om_state=20)

X_train.shape
```

```
(1500, 7)
```

```
X_train.head()
```

	Glucose	BloodPressure	SkinThickness	Insulin	BMI	\
1635	109	60	8	182	25.4	
1531	119	66	27	0	38.8	
698	127	88	11	155	34.5	
1848	88	58	26	16	28.4	
607	92	62	25	41	19.5	

	DiabetesPedigreeFunction	Age
1635	0.947	21
1531	0.259	22
698	0.598	28
1848	0.766	22
607	0.482	25

```
Y_train.head()
```

	Outcome
1635	0
1531	0
698	0
1848	0
607	0

```
X_test.head()
```

	Glucose	BloodPressure	SkinThickness	Insulin	BMI	\
1556	81	72	15	76	30.1	
1560	126	84	29	215	30.7	
1559	136	74	49	220	20.1	
1594	95	74	21	73	25.9	

```
906       140             85             33        0  37.4

      DiabetesPedigreeFunction  Age
1556                     0.547   25
1560                     0.520   24
1559                     0.820   44
1594                     0.673   36
906                      0.244   41
```

```
Y_test.head()
```

```
      Outcome
1556        0
1560        0
1559        1
1594        0
906         0
```

```
from sklearn.linear_model import LogisticRegression
```

```
LR=LogisticRegression()
```

```
LR
```

```
LogisticRegression()
```

```
LR.fit(X_train,Y_train)
```

```
c:\users\lenovo\appdata\local\programs\python\python37\lib\site-
packages\sklearn\utils\validation.py:993: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change
the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

```
LogisticRegression()
```

```
from sklearn.metrics import accuracy_score
```

```
from sklearn.linear_model import LogisticRegression
LR=LogisticRegression()
LR.fit(X_train, Y_train)
p1=LR.score(X_test,Y_test)*100
print(p1)
```

```
75.6
```

```
c:\users\lenovo\appdata\local\programs\python\python37\lib\site-
packages\sklearn\utils\validation.py:993: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change
the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

```
from sklearn.ensemble import AdaBoostClassifier
ADA=AdaBoostClassifier()
ADA.fit(X_train, Y_train)
p2=ADA.score(X_test,Y_test)*100
print(p2)
```

```
c:\users\lenovo\appdata\local\programs\python\python37\lib\site-
packages\sklearn\utils\validation.py:993: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change
the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

```
79.60000000000001
```

```
from sklearn.ensemble import RandomForestClassifier
RF=RandomForestClassifier(max_features='auto', n_estimators=200)
RF.fit(X_train, Y_train)
p3=RF.score(X_test,Y_test)*100
print(p3)
```

```
c:\users\lenovo\appdata\local\programs\python\python37\lib\site-
packages\ipykernel_launcher.py:3: DataConversionWarning: A column-
vector y was passed when a 1d array was expected. Please change the
shape of y to (n_samples,), for example using ravel().
  This is separate from the ipykernel package so we can avoid doing
imports until
```

```
96.2
```

```
from sklearn.tree import DecisionTreeClassifier
DC=DecisionTreeClassifier()
DC.fit(X_train, Y_train)
p4=DC.score(X_test,Y_test)*100
print(p4)
```

```
98.0
```

```
from sklearn.naive_bayes import GaussianNB
GB=GaussianNB()
GB.fit(X_train,Y_train)
p5=GB.score(X_test,Y_test)*100
print(p5)
```

```
74.4
```

```
c:\users\lenovo\appdata\local\programs\python\python37\lib\site-
packages\sklearn\utils\validation.py:993: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change
the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

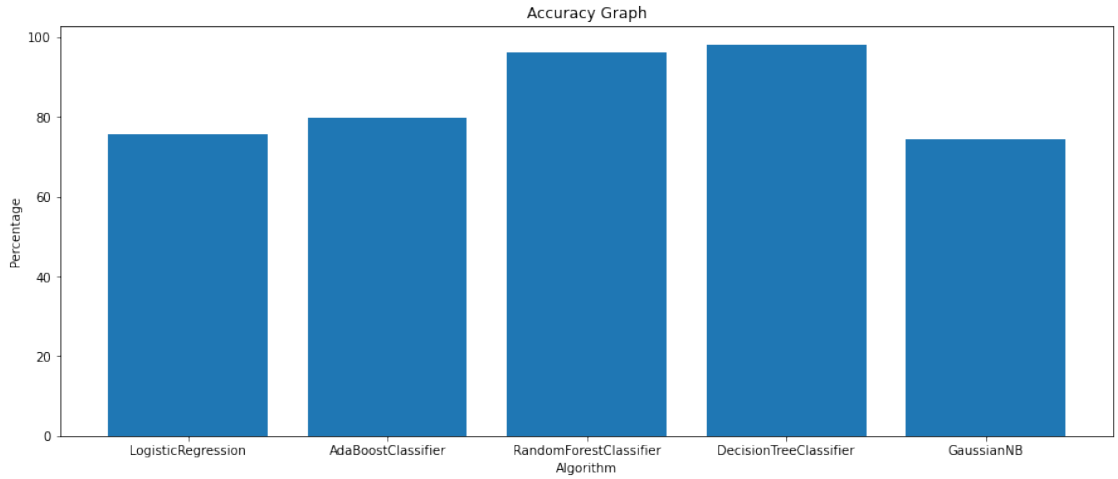
```
from sklearn.tree import DecisionTreeClassifier
```

```
DC=DecisionTreeClassifier()

DC.fit(X_train,Y_train)

DecisionTreeClassifier()

a=["LogisticRegression","AdaBoostClassifier","RandomForestClassifier",
"DecisionTreeClassifier","GaussianNB"]
b=[p1,p2,p3,p4,p5]
plt.figure(figsize=(15,6))
plt.bar(a,b)
plt.title("Accuracy Graph")
plt.xlabel("Algorithm")
plt.ylabel("Percentage")
plt.show()
```




```
result=DC.predict([[138,62,35,0,33.6,0.127,47]])

c:\users\lenovo\appdata\local\programs\python\python37\lib\site-
packages\sklearn\base.py:451: UserWarning: X does not have valid
feature names, but DecisionTreeClassifier was fitted with feature
names
  "X does not have valid feature names, but"

print([result])

[array([1], dtype=int64)]

print([result])

[array([1], dtype=int64)]

result=DC.predict([[80,0,0,0,0,0.174,22]])
result_percentage=DC.predict_proba([[80,0,0,0,0,0.174,22]])
print("It is ",round(max(result_percentage[0])*100,2),"% -----
>",result[0])

It is  100.0 % -----> 0
```

```
c:\users\lenovo\appdata\local\programs\python\python37\lib\site-
packages\sklearn\base.py:451: UserWarning: X does not have valid
feature names, but DecisionTreeClassifier was fitted with feature
names
  "X does not have valid feature names, but"
c:\users\lenovo\appdata\local\programs\python\python37\lib\site-
packages\sklearn\base.py:451: UserWarning: X does not have valid
feature names, but DecisionTreeClassifier was fitted with feature
names
  "X does not have valid feature names, but"

print([result])

[array([0], dtype=int64)]

Glucose=float(input("Enter the Glucose :"))
BloodPressure=float(input("Enter the BloodPressure:"))
SkinThickness=float(input("Enter the SkinThickness:"))
Insulin=float(input("Enter the Insulin:"))
BMI=float(input("Enter the BMI:"))
DiabetesPedigreeFunction=float(input("Enter the
DiabetesPedigreeFunction:"))
Age=float(input("Enter the Age:"))

result=RF.predict([[Glucose, BloodPressure, SkinThickness, Insulin,
BMI,DiabetesPedigreeFunction, Age]])
result_percentage=RF.predict_proba([[Glucose, BloodPressure,
SkinThickness, Insulin, BMI,DiabetesPedigreeFunction, Age]])
if(result==1):
    result="You may have Diabetes"
else:
    result="You Dont have Diabetes"
print(round(max(result_percentage[0])*100,2),"% ", result)

Enter the Glucose :

---------------------------------------------------------------------
-----
ValueError                                Traceback (most recent call
last)
~\AppData\Local\Temp\ipykernel_8732\150772915.py in <module>
----> 1 Glucose=float(input("Enter the Glucose :"))
      2 BloodPressure=float(input("Enter the BloodPressure:"))
      3 SkinThickness=float(input("Enter the SkinThickness:"))
      4 Insulin=float(input("Enter the Insulin:"))
      5 BMI=float(input("Enter the BMI:"))

ValueError: could not convert string to float:

import pickle
```

```
f=open('diabetesModel.pkl','wb')
pickle.dump(DC,f)
f.close()
```

```
---------------------------------------------------------------------------
-----
NameError                                 Traceback (most recent call
last)
~\AppData\Local\Temp\ipykernel_9172\651933054.py in <module>
      1 f=open('diabetesModel.pkl','wb')
----> 2 pickle.dump(DC,f)
      3 f.close()

NameError: name 'DC' is not defined
```