K - Neighbors Classifirer Algorithm

Name : Satbhai Anket Satbhai

Roll No. : 4286, Batch : B7

In [2]:
```
import pandas as pd
df=pd.read_csv('C:\\Users\\DELL\\Desktop\\diabetes.csv')
df.head()
```

Out[2]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Pedigree	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

In [3]:
```
df.shape
```

Out[3]:
```
(768, 9)
```

In [4]:
```
x=df.drop(['Outcome'],axis=1)
y=df['Outcome']
```

In [5]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=21
```

In [6]:
```
x_train.shape
```

Out[6]:
```
(614, 8)
```

In [7]:
```
x_test.shape
```

Out[7]:
```
(154, 8)
```

K - Neighbors Classifirer Algorithm

In [8]:
```
from sklearn.neighbors import KNeighborsClassifier
knc=KNeighborsClassifier()
knc.fit(x_train,y_train)
```

Out[8]:
```
KNeighborsClassifier()
```

In [9]:
```
y_pred=knc.predict(x_test)
```

In [10]:
```
from sklearn import metrics as mt
```

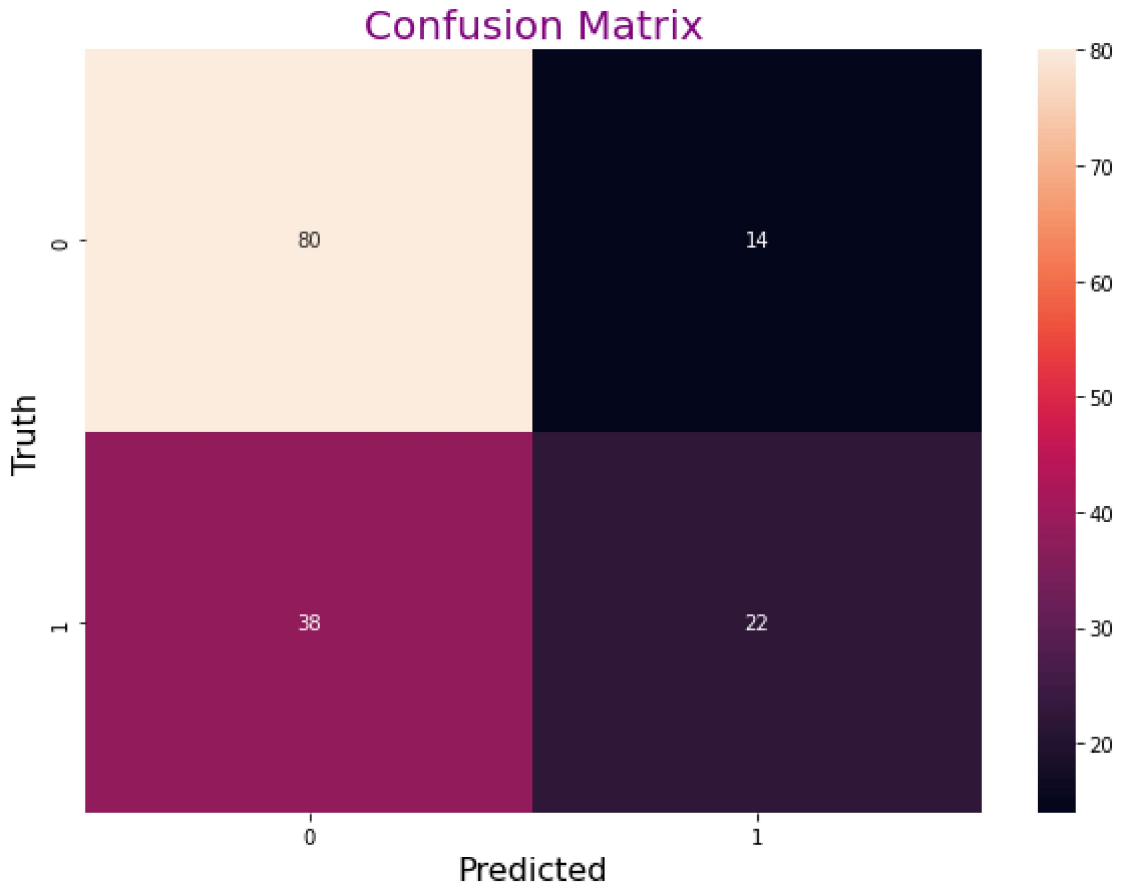
In [11]:
```
#model accuracy
mt.accuracy_score(y_test,y_pred)
```

Out[11]:
```
0.6623376623376623
```

In [12]:
```
#confusion matrix
cm=mt.confusion_matrix(y_test,y_pred)
```

In [13]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sn
plt.figure(figsize=(10,7))
sn.heatmap(cm,annot=True)
plt.xlabel('Predicted',size=16)
plt.ylabel('Truth',size=16)
plt.title('Confusion Matrix',size=20,color='purple')
```

Out[13]:
```
Text(0.5, 1.0, 'Confusion Matrix')
```



In [15]:
```
print(mt.confusion_matrix(y_test,y_pred))
```

```
[[80 14]
 [38 22]]
```

In [16]:
```
tn=80 # true negative
fp=14 # false positive
fn=38 # false negative
tp=22 # true positive
```

In [17]:
```
# accuracy
accuracy=((tn+tp)*100)/(tn+tp+fn+fp)
print('accuracy : ',accuracy,'%')
```

```
accuracy :  66.23376623376623 %
```

In [18]:
```
# precision
precision = tp/(tp+fp)
print('recision : ',precision,'%')
```

```
recision :  0.6111111111111112 %
```

In [19]:
```
# recall
recall = tp/(tp+fn)
print('recall : ',recall,'%')
```

```
recall :  0.36666666666666664 %
```

In [20]:
```
# error rate
err = (fp+fn)/(tp+tn+fp+fn)
print('error rate : ',err)
```

```
error rate :  0.33766233766233766
```