

```

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn import metrics

```

```
df = pd.read_csv("/content/sample_data/diabetes.csv")
```

```
df.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   Pedigree              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

```

```
df.describe()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	79.799479
std	3.369578	31.972618	19.355807	15.952218	115.244002	115.244002
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	30.500000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	127.250000
max	17.000000	199.000000	122.000000	99.000000	846.000000	846.000000

```
df.columns
```

✓ 0s completed at 11:49 AM



```
Pregnancies    0
Glucose         0
BloodPressure  0
SkinThickness  0
Insulin         0
BMI             0
Pedigree        0
Age            0
Outcome        0
dtype: int64
```

```
x = df.iloc[:, :-1]
y = df['Outcome']
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.20)
```

```
from sklearn.neighbors import KNeighborsClassifier
KNN = KNeighborsClassifier(n_neighbors=7)
KNN.fit(x_train, y_train)
y_pred = KNN.predict(x_test)
```

```
print("Confusion matrix : ")
cm = metrics.confusion_matrix(y_pred, y_test)
print(cm)
```

```
Confusion matrix :
[[85 22]
 [13 34]]
```

```
print("Accuracy : ", metrics.accuracy_score(y_pred, y_test))
```

```
Accuracy :  0.7727272727272727
```

```
print("Error rate : ", 1 - metrics.accuracy_score(y_pred, y_test))
```

```
Error rate :  0.2272727272727273
```

```
print("Precision Score : ", metrics.precision_score(y_pred, y_test))
```

```
Precision Score :  0.6071428571428571
```

```
print("Recall Score : ", metrics.recall_score(y_pred, y_test))
```

[Colab paid products](#) - [Cancel contracts here](#)