

```
In [195... import pandas as pd
import numpy as np
```

```
In [196... master_diabetic = pd.read_csv(r"D:\PG-DAI\MachineLearning\Dec 16 Classification KNN\diabetes.csv")
```

```
In [197... master_diabetic.value_counts()
```

```
Out[197... Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin  BMI  DiabetesPedigreeFunction  Age  Outcome
17          163         72           41          114    40.9    0.817                47    1          1
15          136         70           32          110    37.1    0.153                43    1          1
2           87         58           16           52    32.7    0.166                25    0          1
           85         65           23           0     28.9    0.773                25    0          1
           85         65           0            0     39.6    0.930                27    0          1
..
5           106         82           30            0    39.5    0.286                38    0          1
           105         72           29          325    36.9    0.159                28    0          1
           104         74            0            0    28.8    0.153                48    0          1
           103        108           37            0    39.2    0.305                65    0          1
0           57         60            0            0    21.7    0.735                67    0          1
Length: 768, dtype: int64
```

```
In [198... #Dropping the coloumns where BP is 0
master_diabetic.drop(master_diabetic[master_diabetic['BloodPressure'] ==0].index, inplace = True)
```

```
In [199... # master_diabetic[master_diabetic['SkinThickness']==0] = master_diabetic['SkinThickness'].mode()
```

```
In [200... master_diabetic['SkinThickness'].replace(to_replace=0, value=24.0, inplace=True, limit=None, regex=False, method='pad')
```

```
In [201... master_diabetic['SkinThickness']
```

```
Out[201... 0      35
1      29
2      24
3      23
4      35
..
763    48
764    27
765    23
766    24
767    31
Name: SkinThickness, Length: 733, dtype: int64
```

In []:

In []:

In [202...] `master_diabetic.isna().value_counts()`

Out[202...] Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction Age Outcome
 False False False False False False False False False 733
 dtype: int64

In [203...] `X = master_diabetic.iloc[:, :-1].values`
`y = master_diabetic['Outcome']`

In [204...] `from sklearn.model_selection import train_test_split`
`#X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30, random_state=42`
`X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25)`

In [205...] `from sklearn.neighbors import KNeighborsClassifier`
`classifier = KNeighborsClassifier()`
`classifier.fit(X_train, y_train)`

Out[205...] `KNeighborsClassifier()`In [206...] `y_pred = classifier.predict(X_test)`

In [207...] `from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, classification_report`
`result = confusion_matrix(y_test, y_pred)`
`print("Confusion Matrix:")`
`print(result)`
`result1 = accuracy_score(y_test, y_pred)`
`print("Accuracy:", result1)`

Confusion Matrix:
 [[102 19]
 [25 38]]
 Accuracy: 0.7608695652173914

In [208...] `result2 = classification_report(y_test, y_pred, digits=4)`
`print("Classification Report:")`
`print(result2)`

Classification Report:

	precision	recall	f1-score	support
0	0.8031	0.8430	0.8226	121
1	0.6667	0.6032	0.6333	63
accuracy			0.7609	184
macro avg	0.7349	0.7231	0.7280	184
weighted avg	0.7564	0.7609	0.7578	184

In [209...]

```

cnt =0
count=[]
train_score =[]
test_score = []
# Will take some time
for i in range(1,15):

    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train,y_train)
    train_score_ = knn.score(X_train,y_train)
    test_score_ = knn.score(X_test,y_test)
    cnt+=1
    count.append(cnt)
    train_score.append(train_score_)
    test_score.append(test_score_)

print("for k = ", cnt)
print("train_score is : ", train_score_, "and test score is : ", test_score_)
print("*****")
print("*****")
print("Average train score is : ",np.mean(train_score))
print("Average test score is : ", np.mean(test_score))

for k = 14
train_score is : 0.761384335154827 and test score is : 0.7934782608695652
*****
*****
Average train score is : 0.8013270882123339
Average test score is : 0.7740683229813664

```

In []:

In []:

