

```
import pandas as pd
data=pd.read_csv("Downloads/diabetes.csv")
```

```
data
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI
0	6	148	72	35	0	33.6
1	1	85	66	29	0	26.6
2	8	183	64	0	0	23.3
3	1	89	66	23	94	28.1
4	0	137	40	35	168	43.1
..
763	10	101	76	48	180	32.9
764	2	122	70	27	0	36.8
765	5	121	72	23	112	26.2
766	1	126	60	0	0	30.1
767	1	93	70	31	0	30.4

	DiabetesPedigreeFunction	Age	Outcome
0	0.627	50	1
1	0.351	31	0
2	0.672	32	1
3	0.167	21	0
4	2.288	33	1
..
763	0.171	63	0
764	0.340	27	0
765	0.245	30	0
766	0.349	47	1
767	0.315	23	0

```
[768 rows x 9 columns]
```

```
data.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI
0	6	148	72	35	0	33.6

1	1	85	66	29	0	26.6
2	8	183	64	0	0	23.3
3	1	89	66	23	94	28.1
4	0	137	40	35	168	43.1

	DiabetesPedigreeFunction	Age	Outcome
0	0.627	50	1
1	0.351	31	0
2	0.672	32	1
3	0.167	21	0
4	2.288	33	1

```
from sklearn.model_selection import train_test_split
```

```
X=data.drop('Outcome',axis=1)
```

```
y=data['Outcome']
```

X

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI
\						
0	6	148	72	35	0	33.6
1	1	85	66	29	0	26.6
2	8	183	64	0	0	23.3
3	1	89	66	23	94	28.1
4	0	137	40	35	168	43.1
..
763	10	101	76	48	180	32.9
764	2	122	70	27	0	36.8
765	5	121	72	23	112	26.2
766	1	126	60	0	0	30.1
767	1	93	70	31	0	30.4

	DiabetesPedigreeFunction	Age
0	0.627	50
1	0.351	31
2	0.672	32

3	0.167	21
4	2.288	33
..
763	0.171	63
764	0.340	27
765	0.245	30
766	0.349	47
767	0.315	23

[768 rows x 8 columns]

y

0	1
1	0
2	1
3	0
4	1
..	..
763	0
764	0
765	0
766	1
767	0

Name: Outcome, Length: 768, dtype: int64

```
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
```

x_train

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI
762	9	89	62	0	0	22.5
127	1	118	58	36	94	33.3
564	0	91	80	0	0	32.4
375	12	140	82	43	325	39.2
663	9	145	80	46	130	37.9
..
763	10	101	76	48	180	32.9
192	7	159	66	0	0	30.4
629	4	94	65	22	0	24.7

559	11	85	74	0	0	30.1
684	5	136	82	0	0	0.0

	DiabetesPedigreeFunction	Age
762	0.142	33
127	0.261	23
564	0.601	27
375	0.528	58
663	0.637	40
..
763	0.171	63
192	0.383	36
629	0.148	21
559	0.300	35
684	0.640	69

[576 rows x 8 columns]

y_train

762	0
127	0
564	0
375	1
663	1
..	..
763	0
192	1
629	0
559	0
684	0

Name: Outcome, Length: 576, dtype: int64

x_test

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI
\						
661	1	199	76	43	0	42.9
122	2	107	74	30	100	33.6
113	4	76	62	0	0	34.0
14	5	166	72	19	175	25.8
529	0	111	65	0	0	24.6
..

366	6	124	72	0	0	27.6
301	2	144	58	33	135	31.6
382	1	109	60	8	182	25.4
140	3	128	78	0	0	21.1
463	5	88	78	30	0	27.6

	DiabetesPedigreeFunction	Age
661	1.394	22
122	0.404	23
113	0.391	25
14	0.587	51
529	0.660	31
..
366	0.368	29
301	0.422	25
382	0.947	21
140	0.268	55
463	0.258	37

[192 rows x 8 columns]

y

0	1
1	0
2	1
3	0
4	1
..	..
763	0
764	0
765	0
766	1
767	0

Name: Outcome, Length: 768, dtype: int64

y_test

661	1
122	0
113	0
14	1
529	0
..	..
366	1
301	1

```
382    0
140    0
463    0
```

```
Name: Outcome, Length: 192, dtype: int64
```

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import
confusion_matrix, accuracy_score, precision_score, recall_score
```

```
knn=KNeighborsClassifier(n_neighbors=10)
```

```
knn.fit(x_train,y_train)
```

```
KNeighborsClassifier(n_neighbors=10)
```

```
y_pred=knn.predict(x_test)
```

```
y_pred
```

```
array([1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0,
      0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0,
1,
      1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1,
1,
      1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
      1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1,
      0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0,
0,
      0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
0,
      1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
      0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

```
from sklearn import metrics
acc=metrics.accuracy_score(y_pred,y_test)
```

```
acc
```

```
0.78125
```

```
err=(1-acc)
```

```
err
```

```
0.21875
```

```
def knn(x_train,y_train,x_test,y_test,n):
    n_range=range(1,n)
    results=[]
```

```

for n in n_range:
    knn=KNeighborsClassifier(n_neighbors=n)
    knn.fit(x_train,y_train)
    y_pred=knn.predict(x_test)
    acc=metrics.accuracy_score(y_pred,y_test)
    results.append(acc)
return results

import matplotlib.pyplot as plt
n=400
output=knn(x_train,y_train,x_test,y_test,n)
n_range=range(1,n)
plt.plot(n_range,output)

[<matplotlib.lines.Line2D at 0x70ea98541d80>]

```

