



```
In [20]: #Import scikit-learn dataset library
from sklearn import datasets

#Load dataset
cancer = datasets.load_breast_cancer()
```

```
In [21]: import pandas as pd
df = pd.DataFrame(cancer.data)
```

```
In [22]: df.head()
```

```
Out[22]:
```

	0	1	2	3	4	5	6	7	8	9
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883

5 rows × 30 columns

```
In [23]: df.tail()
```

```
Out[23]:
```

	0	1	2	3	4	5	6	7	8	
564	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	0.1726	0.05
565	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	0.1752	0.05
566	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	0.1590	0.05
567	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	0.2397	0.07
568	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	0.1587	0.05

5 rows × 30 columns

```
In [24]: df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 30 columns):
#   Column  Non-Null Count  Dtype
---  -
0   0        569 non-null     float64
1   1        569 non-null     float64
2   2        569 non-null     float64
3   3        569 non-null     float64
4   4        569 non-null     float64
5   5        569 non-null     float64
6   6        569 non-null     float64
7   7        569 non-null     float64
8   8        569 non-null     float64
9   9        569 non-null     float64
10  10       569 non-null     float64
11  11       569 non-null     float64
12  12       569 non-null     float64
13  13       569 non-null     float64
14  14       569 non-null     float64
15  15       569 non-null     float64
16  16       569 non-null     float64
17  17       569 non-null     float64
18  18       569 non-null     float64
19  19       569 non-null     float64
20  20       569 non-null     float64
21  21       569 non-null     float64
22  22       569 non-null     float64
23  23       569 non-null     float64
24  24       569 non-null     float64
25  25       569 non-null     float64
26  26       569 non-null     float64
27  27       569 non-null     float64
28  28       569 non-null     float64
29  29       569 non-null     float64
dtypes: float64(30)
memory usage: 133.5 KB

```

```
In [25]: df.describe()
```

Out[25]:

	0	1	2	3	4	5
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000
mean	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341
std	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813
min	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380
25%	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920
50%	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630
75%	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400
max	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400

8 rows × 30 columns

```
In [ ]: print("Features: ", cancer.feature_names)
        sprint("Labels: ", cancer.target_names)
```

```
Features: ['mean radius' 'mean texture' 'mean perimeter' 'mean area'
'mean smoothness' 'mean compactness' 'mean concavity'
'mean concave points' 'mean symmetry' 'mean fractal dimension'
'radius error' 'texture error' 'perimeter error' 'area error'
'smoothness error' 'compactness error' 'concavity error'
'concave points error' 'symmetry error' 'fractal dimension error'
'worst radius' 'worst texture' 'worst perimeter' 'worst area'
'worst smoothness' 'worst compactness' 'worst concavity'
'worst concave points' 'worst symmetry' 'worst fractal dimension']
Labels: ['malignant' 'benign']
```

```
In [27]: cancer.data.shape
```

Out[27]: (569, 30)

```
In [28]: # print the cancer labels (0:malignant, 1:benign)
        print(cancer.target)
```

[illegible]

```
In [29]: from sklearn.model_selection import train_test_split

# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(cancer.data, cancer.target
```

```
In [30]: from sklearn import svm

          clf = svm.SVC(kernel='linear')

          clf.fit(X_train, y_train)

          y_pred = clf.predict(X_test)
```

```
In [31]: from sklearn import metrics

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

Accuracy: 0.9649122807017544

```
In [32]: print("Precision:", metrics.precision_score(y_test, y_pred))

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

```
Precision: 0.9811320754716981
```

Recall: 0.9629629629629629

In []: