Support Vector Regression

Cancer dataset

Data Loaded

In [ ]:
```
from sklearn import datasets
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
cancer = datasets.load_breast_cancer()
```

In [3]:
```
print("Features: ", cancer.feature_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']
```

In [21]:
```
df=pd.DataFrame(cancer.data,columns=cancer.feature_names)
```

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

Out[22]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst radius	worst texture	worst perimeter	wor ar
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	...	25.38	17.33	184.60	2019
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	...	24.99	23.41	158.80	1956
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	...	23.57	25.53	152.50	1709
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	...	14.91	26.50	98.87	567
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	...	22.54	16.67	152.20	1575

5 rows × 30 columns

Preprocessing

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 30 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   mean radius             569 non-null    float64
 1   mean texture            569 non-null    float64
 2   mean perimeter          569 non-null    float64
 3   mean area               569 non-null    float64
 4   mean smoothness         569 non-null    float64
 5   mean compactness        569 non-null    float64
 6   mean concavity          569 non-null    float64
 7   mean concave points     569 non-null    float64
 8   mean symmetry           569 non-null    float64
 9   mean fractal dimension  569 non-null    float64
 10  radius error            569 non-null    float64
 11  texture error           569 non-null    float64
 12  perimeter error         569 non-null    float64
 13  area error              569 non-null    float64
```

```
 14  smoothness error         569 non-null    float64
 15  compactness error        569 non-null    float64
 16  concavity error          569 non-null    float64
 17  concave points error     569 non-null    float64
 18  symmetry error           569 non-null    float64
 19  fractal dimension error  569 non-null    float64
 20  worst radius             569 non-null    float64
 21  worst texture            569 non-null    float64
 22  worst perimeter          569 non-null    float64
 23  worst area               569 non-null    float64
 24  worst smoothness         569 non-null    float64
 25  worst compactness        569 non-null    float64
 26  worst concavity          569 non-null    float64
 27  worst concave points     569 non-null    float64
 28  worst symmetry           569 non-null    float64
 29  worst fractal dimension  569 non-null    float64
dtypes: float64(30)
memory usage: 133.5 KB
```

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

Out[24]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	w ra
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	...	569.00
mean	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	0.181162	0.062798	...	16.26
std	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	0.027414	0.007060	...	4.83
min	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	0.106000	0.049960	...	7.93
25%	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	0.161900	0.057700	...	13.01
50%	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	0.179200	0.061540	...	14.97
75%	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	0.195700	0.066120	...	18.79
max	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.201200	0.304000	0.097440	...	36.04

8 rows × 30 columns

In [25]:
```
df.isnull().sum()
```

Out[25]:
```
mean radius                0
mean texture               0
mean perimeter             0
mean area                  0
mean smoothness            0
mean compactness           0
mean concavity             0
mean concave points        0
mean symmetry              0
mean fractal dimension     0
radius error               0
texture error              0
perimeter error            0
area error                 0
smoothness error           0
compactness error          0
concavity error            0
concave points error       0
symmetry error             0
fractal dimension error    0
worst radius               0
worst texture              0
worst perimeter            0
worst area                 0
worst smoothness           0
worst compactness          0
worst concavity            0
worst concave points       0
worst symmetry             0
worst fractal dimension    0
dtype: int64
```

In [27]:
```
df.corr()
```

Out[27]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst radius
mean radius	1.000000	0.323782	0.997855	0.987357	0.170581	0.506124	0.676764	0.822529	0.147741	-0.311631	...	0.969539
mean texture	0.323782	1.000000	0.329533	0.321086	-0.023389	0.236702	0.302418	0.293464	0.071401	-0.076437	...	0.352573
mean perimeter	0.997855	0.329533	1.000000	0.986507	0.207278	0.556936	0.716136	0.850977	0.183027	-0.261477	...	0.969476
mean area	0.987357	0.321086	0.986507	1.000000	0.177028	0.498502	0.685983	0.823269	0.151293	-0.283110	...	0.962746
mean smoothness	0.170581	-0.023389	0.207278	0.177028	1.000000	0.659123	0.521984	0.553695	0.557775	0.584792	...	0.213120
mean compactness	0.506124	0.236702	0.556936	0.498502	0.659123	1.000000	0.883121	0.831135	0.602641	0.565369	...	0.535315
mean concavity	0.676764	0.302418	0.716136	0.685983	0.521984	0.883121	1.000000	0.921391	0.500667	0.336783	...	0.688236
mean concave points	0.822529	0.293464	0.850977	0.823269	0.553695	0.831135	0.921391	1.000000	0.462497	0.166917	...	0.830318
mean symmetry	0.147741	0.071401	0.183027	0.151293	0.557775	0.602641	0.500667	0.462497	1.000000	0.479921	...	0.185728
mean fractal dimension	-0.311631	-0.076437	-0.261477	-0.283110	0.584792	0.565369	0.336783	0.166917	0.479921	1.000000	...	-0.253691
radius error	0.679090	0.275869	0.691765	0.732562	0.301467	0.497473	0.631925	0.698050	0.303379	0.000111	...	0.715065
texture error	-0.097317	0.386358	-0.086761	-0.066280	0.068406	0.046205	0.076218	0.021480	0.128053	0.164174	...	-0.111690
perimeter error	0.674172	0.281673	0.693135	0.726628	0.296092	0.548905	0.660391	0.710650	0.313893	0.039830	...	0.697201
area error	0.735864	0.259845	0.744983	0.800086	0.246552	0.455653	0.617427	0.690299	0.223970	-0.090170	...	0.757373
smoothness error	-0.222600	0.006614	-0.202694	-0.166777	0.332375	0.135299	0.098564	0.027653	0.187321	0.401964	...	-0.230691
compactness error	0.206000	0.191975	0.250744	0.212583	0.318943	0.738722	0.670279	0.490424	0.421659	0.559837	...	0.204607
concavity error	0.194204	0.143293	0.228082	0.207660	0.248396	0.570517	0.691270	0.439167	0.342627	0.446630	...	0.186904
concave points error	0.376169	0.163851	0.407217	0.372320	0.380676	0.642262	0.683260	0.615634	0.393298	0.341198	...	0.358127
symmetry error	-0.104321	0.009127	-0.081629	-0.072497	0.200774	0.229977	0.178009	0.095351	0.449137	0.345007	...	-0.128121
fractal dimension error	-0.042641	0.054458	-0.005523	-0.019887	0.283607	0.507318	0.449301	0.257584	0.331786	0.688132	...	-0.037488
worst radius	0.969539	0.352573	0.969476	0.962746	0.213120	0.535315	0.688236	0.830318	0.185728	-0.253691	...	1.000000
worst texture	0.297008	0.912045	0.303038	0.287489	0.036072	0.248133	0.299879	0.292752	0.090651	-0.051269	...	0.359921
worst perimeter	0.965137	0.358040	0.970387	0.959120	0.238853	0.590210	0.729565	0.855923	0.219169	-0.205151	...	0.993708
worst area	0.941082	0.343546	0.941550	0.959213	0.206718	0.509604	0.675987	0.809630	0.177193	-0.231854	...	0.984015
worst smoothness	0.119616	0.077503	0.150549	0.123523	0.805324	0.565541	0.448822	0.452753	0.426675	0.504942	...	0.216574
worst compactness	0.413463	0.277830	0.455774	0.390410	0.472468	0.865809	0.754968	0.667454	0.473200	0.458798	...	0.475820
worst concavity	0.526911	0.301025	0.563879	0.512606	0.434926	0.816275	0.884103	0.752399	0.433721	0.346234	...	0.573975
worst concave points	0.744214	0.295316	0.771241	0.722017	0.503053	0.815573	0.861323	0.910155	0.430297	0.175325	...	0.787424
worst symmetry	0.163953	0.105008	0.189115	0.143570	0.394309	0.510223	0.409464	0.375744	0.699826	0.334019	...	0.243529
worst fractal dimension	0.007066	0.119205	0.051019	0.003738	0.499316	0.687382	0.514930	0.368661	0.438413	0.767297	...	0.093492

30 rows × 30 columns

In [28]:

```
sns.heatmap(df.corr(),annot=True)
```

Out[28]: <AxesSubplot:>

In [29]:

```
df.cov()
```

Out[29]:

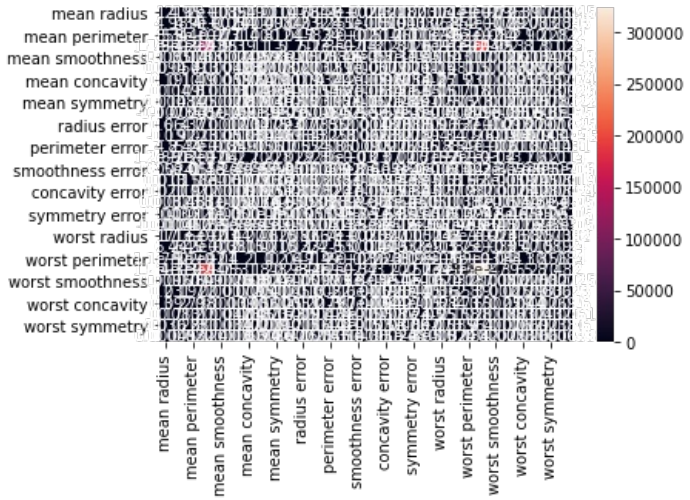
	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension
mean radius	12.418920	4.907582	85.447142	1224.483409	0.008454	0.094197	0.190128	0.112475	0.014273	-7.753706e-03
mean texture	4.907582	18.498909	34.439759	485.993787	-0.001415	0.053767	0.103692	0.048977	0.008419	-2.321158e-03
mean perimeter	85.447142	34.439759	590.440480	8435.772345	0.070836	0.714714	1.387234	0.802360	0.121922	-4.485888e-02
mean area	1224.483409	485.993787	8435.772345	123843.554318	0.876178	9.264931	19.244924	11.241958	1.459596	-7.034264e-01
mean smoothness	0.008454	-0.001415	0.070836	0.876178	0.000198	0.000490	0.000585	0.000302	0.000215	5.806859e-05
mean compactness	0.094197	0.053767	0.714714	9.264931	0.000490	0.002789	0.003718	0.001703	0.000873	2.108131e-04
mean concavity	0.190128	0.103692	1.387234	19.244924	0.000585	0.003718	0.006355	0.002850	0.001094	1.895588e-04
mean concave points	0.112475	0.048977	0.802360	11.241958	0.000302	0.001703	0.002850	0.001506	0.000492	4.572905e-05
mean symmetry	0.014273	0.008419	0.121922	1.459596	0.000215	0.000873	0.001094	0.000492	0.000752	9.289106e-05
mean fractal dimension	-0.007754	-0.002321	-0.044859	-0.703426	0.000058	0.000211	0.000190	0.000046	0.000093	4.984872e-05
radius error	0.663650	0.329037	4.661401	71.490945	0.001176	0.007286	0.013970	0.007511	0.002306	2.173204e-07
texture error	-0.189189	0.916695	-1.162988	-12.867168	0.000531	0.001346	0.003352	0.000460	0.001937	6.394310e-04
perimeter error	4.803550	2.449449	34.053028	517.009995	0.008420	0.058612	0.106443	0.055753	0.017398	5.685733e-04
area error	117.968162	50.840865	823.492755	12808.517580	0.157742	1.094708	2.239119	1.218501	0.279314	-2.896115e-02
smoothness error	-0.002355	0.000085	-0.014788	-0.176221	0.000014	0.000021	0.000024	0.000003	0.000015	8.521190e-06
compactness error	0.013001	0.014787	0.109111	1.339725	0.000080	0.000699	0.000957	0.000341	0.000207	7.078477e-05
concavity error	0.020659	0.018604	0.167296	2.205952	0.000105	0.000910	0.001663	0.000514	0.000284	9.518788e-05
concave points error	0.008180	0.004348	0.061055	0.808460	0.000033	0.000209	0.000336	0.000147	0.000067	1.486411e-05
symmetry error	-0.003039	0.000325	-0.016396	-0.210896	0.000023	0.000100	0.000117	0.000031	0.000102	2.013587e-05
fractal dimension error	-0.000398	0.000620	-0.000355	-0.018519	0.000011	0.000071	0.000095	0.000026	0.000024	1.285583e-05
worst radius	16.513749	7.329267	113.858063	1637.521341	0.014487	0.136643	0.265181	0.155721	0.024609	-8.657080e-03
worst texture	6.433100	24.110148	45.258113	621.824934	0.003118	0.080544	0.146934	0.069819	0.015274	-2.224817e-03
worst perimeter	114.288570	51.745933	792.328208	11341.789807	0.112879	1.047413	1.954350	1.116016	0.201896	-4.867133e-02
worst area	1888.227223	841.283832	13026.148359	192192.557633	1.655299	15.323436	30.682405	17.886881	2.765725	-9.320240e-01

worst smoothness	0.009625	0.007611	0.083526	0.992514	0.000259	0.000682	0.000817	0.000401	0.000267	8.139931e-05
worst compactness	0.229249	0.188010	1.742478	21.616602	0.001045	0.007194	0.009469	0.004075	0.002041	5.096572e-04
worst concavity	0.387386	0.270110	2.858506	37.634415	0.001276	0.008994	0.014704	0.006091	0.002481	5.099897e-04
worst concave points	0.172393	0.083491	1.231848	16.701789	0.000465	0.002831	0.004513	0.002321	0.000775	8.136752e-05
worst symmetry	0.035746	0.027942	0.284300	3.125809	0.000343	0.001667	0.002020	0.000902	0.001187	1.459016e-04
worst fractal dimension	0.000450	0.009260	0.022391	0.023756	0.000127	0.000656	0.000741	0.000258	0.000217	9.784499e-05

30 rows × 30 columns

In [30]:
```
sns.heatmap(df.cov(),annot=True)
```

Out[30]: <AxesSubplot:>



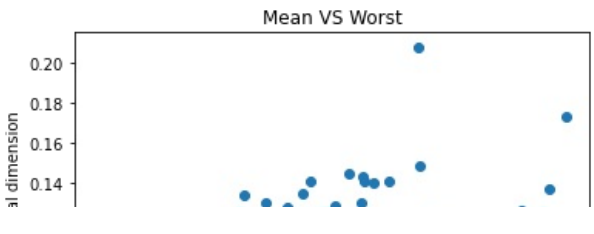
In [31]:
```
df.columns
```

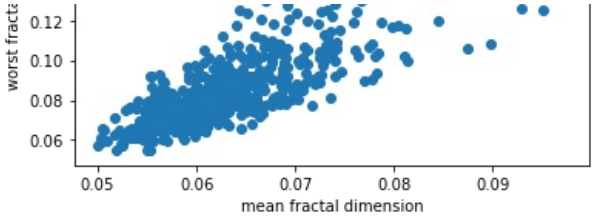
Out[31]:
```
Index(['mean radius', 'mean texture', 'mean perimeter', 'mean area',
       'mean smoothness', 'mean compactness', 'mean concavity',
       'mean concave points', 'mean symmetry', 'mean fractal dimension',
       'radius error', 'texture error', 'perimeter error', 'area error',
       'smoothness 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'],
      dtype='object')
```

Exploratory Data Anlysis

In [34]:
```
plt.scatter(df['mean fractal dimension'],df['worst fractal dimension'])
plt.xlabel('mean fractal dimension')
plt.ylabel('worst fractal dimension')
plt.title('Mean VS Worst')
```

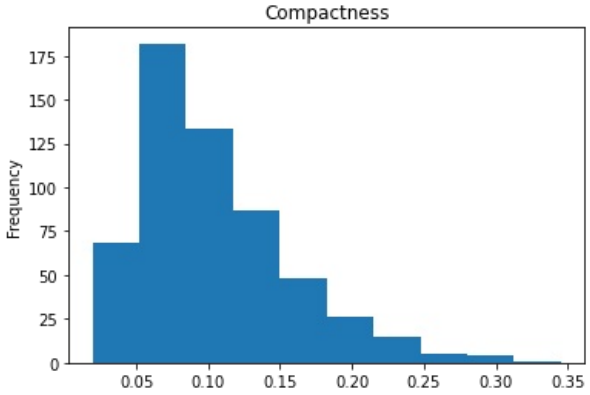
Out[34]: Text(0.5, 1.0, 'Mean VS Worst')





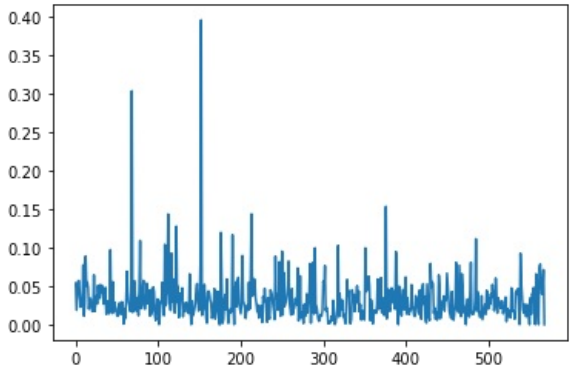
In [36]:
```
df['mean compactness'].plot.hist()
plt.title('Compactness')
```

Out[36]:
```
Text(0.5, 1.0, 'Compactness')
```



In [37]:
```
plt.plot(df['concavity error'])
```

Out[37]:
```
[<matplotlib.lines.Line2D at 0x27d4aeb6850>]
```



In [4]:
```
print("Labels: ", cancer.target_names)
```

```
Labels:  ['malignant' 'benign']
```

In [ ]:
```
sns.pairplot(df)
```

Out[ ]:
```
<seaborn.axisgrid.PairGrid at 0x27d4afcb1c0>
```

In [5]:
```
cancer.data.shape
```

Out[5]:
```
(569, 30)
```

In [6]:
```
print(cancer.data[0:5])
```

```
[[1.799e+01 1.038e+01 1.228e+02 1.001e+03 1.184e-01 2.776e-01 3.001e-01
```

```
 1.471e-01 2.419e-01 7.871e-02 1.095e+00 9.053e-01 8.589e+00 1.534e+02
 6.399e-03 4.904e-02 5.373e-02 1.587e-02 3.003e-02 6.193e-03 2.538e+01
 1.733e+01 1.846e+02 2.019e+03 1.622e-01 6.656e-01 7.119e-01 2.654e-01
 4.601e-01 1.189e-01]
[2.057e+01 1.777e+01 1.329e+02 1.326e+03 8.474e-02 7.864e-02 8.690e-02
 7.017e-02 1.812e-01 5.667e-02 5.435e-01 7.339e-01 3.398e+00 7.408e+01
 5.225e-03 1.308e-02 1.860e-02 1.340e-02 1.389e-02 3.532e-03 2.499e+01
 2.341e+01 1.588e+02 1.956e+03 1.238e-01 1.866e-01 2.416e-01 1.860e-01
 2.750e-01 8.902e-02]
[1.969e+01 2.125e+01 1.300e+02 1.203e+03 1.096e-01 1.599e-01 1.974e-01
 1.279e-01 2.069e-01 5.999e-02 7.456e-01 7.869e-01 4.585e+00 9.403e+01
 6.150e-03 4.006e-02 3.832e-02 2.058e-02 2.250e-02 4.571e-03 2.357e+01
 2.553e+01 1.525e+02 1.709e+03 1.444e-01 4.245e-01 4.504e-01 2.430e-01
 3.613e-01 8.758e-02]
[1.142e+01 2.038e+01 7.758e+01 3.861e+02 1.425e-01 2.839e-01 2.414e-01
 1.052e-01 2.597e-01 9.744e-02 4.956e-01 1.156e+00 3.445e+00 2.723e+01
 9.110e-03 7.458e-02 5.661e-02 1.867e-02 5.963e-02 9.208e-03 1.491e+01
 2.650e+01 9.887e+01 5.677e+02 2.098e-01 8.663e-01 6.869e-01 2.575e-01
 6.638e-01 1.730e-01]
[2.029e+01 1.434e+01 1.351e+02 1.297e+03 1.003e-01 1.328e-01 1.980e-01
 1.043e-01 1.809e-01 5.883e-02 7.572e-01 7.813e-01 5.438e+00 9.444e+01
 1.149e-02 2.461e-02 5.688e-02 1.885e-02 1.756e-02 5.115e-03 2.254e+01
 1.667e+01 1.522e+02 1.575e+03 1.374e-01 2.050e-01 4.000e-01 1.625e-01
 2.364e-01 7.678e-02]]
```

In [7]:

```
print(cancer.target)
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 0 0 1 0 0 1 1 1 1 0 1 0 0 1 1 1 1 1 0 1 0 0
 1 0 1 0 0 1 1 1 0 0 1 0 0 0 1 1 1 0 1 1 0 0 1 1 1 1 0 0 1 1 1 1 0 1 1 0 1 1
 1 1 1 1 1 1 0 0 0 1 0 0 1 1 1 0 0 1 0 1 0 0 1 0 0 1 1 0 1 1 0 1 1 1 1 1 0 1
 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 1 1 0 0 0 1 0
 1 0 1 1 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 1 0 1 0 1 1 0 1 0 0 0 0 1 1 0 0 1 1
 1 0 1 1 1 1 1 1 0 0 1 1 0 1 1 0 0 1 0 1 1 1 1 0 1 1 1 1 1 0 1 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 1 1 1 1 1 0 1 0 1 1 0 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1
 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 1 0 0 0 1 1
 1 1 0 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 0
 0 1 0 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 1 1
 1 0 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 0 1 1 1 1 1 0 1 1
 0 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1
 1 1 1 1 1 1 0 1 0 1 1 0 1 1 1 1 1 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0 1 0 1 0 0
 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 1]
```

Training and Testing data

In [8]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(cancer.data, cancer.target, test_size=0.3,random_state=109)
```

In [9]:

```
from sklearn import svm
clf = svm.SVC(kernel='linear')
clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)
```

In [10]:

```
clf = svm.SVC(kernel='poly')
clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)
```

In [11]:

```
clf = svm.SVC(kernel='rbf')
clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)
```

Evaluation

In [12]:

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

```
Accuracy: 0.9239766081871345
```

In [13]:
```
print("Precision:",metrics.precision_score(y_test, y_pred))
```

```
Precision: 0.8925619834710744
```

In [14]:
```
print("Recall:",metrics.recall_score(y_test, y_pred))
```

```
Recall: 1.0
```

In [ ]:
```

```