# Breast Cancer Prediction with SVM

April 1, 2021

**Breast Cancer Prediction using Support Vector Machines**

**Sohini Mukherjee**

**1.04.2021**

Importing Libraries

[4]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Getting the data

[5]:
```
from sklearn.datasets import load_breast_cancer
```

[6]:
```
cancer = load_breast_cancer()
```

[8]:
```
cancer.keys()
```

[8]:
```
dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names',
'filename'])
```

[10]:
```
df = pd.DataFrame(cancer['data'], columns=cancer['feature_names'])
```

[11]:
```
df.head()
```

[11]:

| | mean radius | mean texture | mean perimeter | mean area | mean smoothness | \ |
|---|---|---|---|---|---|---|
| 0 | 17.99 | 10.38 | 122.80 | 1001.0 | 0.11840 | |
| 1 | 20.57 | 17.77 | 132.90 | 1326.0 | 0.08474 | |
| 2 | 19.69 | 21.25 | 130.00 | 1203.0 | 0.10960 | |
| 3 | 11.42 | 20.38 | 77.58 | 386.1 | 0.14250 | |
| 4 | 20.29 | 14.34 | 135.10 | 1297.0 | 0.10030 | |

| | mean compactness | mean concavity | mean concave points | mean symmetry | \ |
|---|---|---|---|---|---|
| 0 | 0.27760 | 0.3001 | 0.14710 | 0.2419 | |
| 1 | 0.07864 | 0.0869 | 0.07017 | 0.1812 | |
| 2 | 0.15990 | 0.1974 | 0.12790 | 0.2069 | |

```
3           0.28390          0.2414              0.10520         0.2597
4           0.13280          0.1980              0.10430         0.1809

   mean fractal dimension  …  worst radius  worst texture  worst perimeter  \
0                 0.07871  …         25.38          17.33           184.60
1                 0.05667  …         24.99          23.41           158.80
2                 0.05999  …         23.57          25.53           152.50
3                 0.09744  …         14.91          26.50            98.87
4                 0.05883  …         22.54          16.67           152.20

   worst area  worst smoothness  worst compactness  worst concavity  \
0      2019.0            0.1622             0.6656           0.7119
1      1956.0            0.1238             0.1866           0.2416
2      1709.0            0.1444             0.4245           0.4504
3       567.7            0.2098             0.8663           0.6869
4      1575.0            0.1374             0.2050           0.4000

   worst concave points  worst symmetry  worst fractal dimension
0                0.2654          0.4601                  0.11890
1                0.1860          0.2750                  0.08902
2                0.2430          0.3613                  0.08758
3                0.2575          0.6638                  0.17300
4                0.1625          0.2364                  0.07678

[5 rows x 30 columns]
```

[12]:
```
cancer['target_names']
```

[12]:
```
array(['malignant', 'benign'], dtype='<U9')
```

Exploratory Data Analysis

Due to the lack of domain knowledge, I am skipping this step as I am incapable of interpreting the visual comparison between features,

Train Test Split

[13]:
```
from sklearn.model_selection import train_test_split
```

[14]:
```
X = df
y =  cancer['target']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30,␣
 ↪random_state=101)
```

Training Support Vector Classifier Model

[15]:
```
from sklearn.svm import SVC
```

```
<frozen importlib._bootstrap>:219: RuntimeWarning: numpy.ufunc size changed, may
```

```
indicate binary incompatibility. Expected 192 from C header, got 216 from
PyObject
```

[16]:
```
model = SVC()
```

[17]:
```
model.fit(X_train, y_train)
```

[17]:
```
SVC()
```

[18]:
```
predict = model.predict(X_test)
```

[19]:
```
from sklearn.metrics import classification_report, confusion_matrix
```

[20]:
```
confusion_matrix(y_test, predict)
```

[20]:
```
array([[ 56,  10],
       [  3, 102]], dtype=int64)
```

[21]:
```
print(classification_report(y_test, predict))
```

```
              precision    recall  f1-score   support

           0       0.95      0.85      0.90        66
           1       0.91      0.97      0.94       105

    accuracy                           0.92       171
   macro avg       0.93      0.91      0.92       171
weighted avg       0.93      0.92      0.92       171
```

Gridsearch

[23]:
```
from sklearn.model_selection import GridSearchCV
```

[25]:
```
pgrid = {'C':[0.1,1,10,100,1000], 'gamma':[1,0.1,0.01,0.001,0.0001]}
```

[26]:
```
grid = GridSearchCV(SVC(), pgrid, verbose=5)
```

[27]:
```
grid.fit(X_train, y_train)
```

```
Fitting 5 folds for each of 25 candidates, totalling 125 fits
[CV] C=0.1, gamma=1 …
[CV] … C=0.1, gamma=1, score=0.637, total=   0.0s
[CV] C=0.1, gamma=1 …
[CV] … C=0.1, gamma=1, score=0.637, total=   0.0s
[CV] C=0.1, gamma=1 …
[CV] … C=0.1, gamma=1, score=0.625, total=   0.0s
[CV] C=0.1, gamma=1 …
[CV] … C=0.1, gamma=1, score=0.633, total=   0.0s
[CV] C=0.1, gamma=1 …
```

```
[CV] … C=0.1, gamma=1, score=0.633, total=   0.0s
[CV] C=0.1, gamma=0.1 …
[CV] … C=0.1, gamma=0.1, score=0.637, total=   0.0s
[CV] C=0.1, gamma=0.1 …

[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    0.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   2 out of   2 | elapsed:    0.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   3 out of   3 | elapsed:    0.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   4 out of   4 | elapsed:    0.0s remaining:    0.0s

[CV] … C=0.1, gamma=0.1, score=0.637, total=   0.0s
[CV] C=0.1, gamma=0.1 …
[CV] … C=0.1, gamma=0.1, score=0.625, total=   0.0s
[CV] C=0.1, gamma=0.1 …
[CV] … C=0.1, gamma=0.1, score=0.633, total=   0.0s
[CV] C=0.1, gamma=0.1 …
[CV] … C=0.1, gamma=0.1, score=0.633, total=   0.0s
[CV] C=0.1, gamma=0.01 …
[CV] … C=0.1, gamma=0.01, score=0.637, total=   0.0s
[CV] C=0.1, gamma=0.01 …
[CV] … C=0.1, gamma=0.01, score=0.637, total=   0.0s
[CV] C=0.1, gamma=0.01 …
[CV] … C=0.1, gamma=0.01, score=0.625, total=   0.0s
[CV] C=0.1, gamma=0.01 …
[CV] … C=0.1, gamma=0.01, score=0.633, total=   0.0s
[CV] C=0.1, gamma=0.01 …
[CV] … C=0.1, gamma=0.01, score=0.633, total=   0.0s
[CV] C=0.1, gamma=0.001 …
[CV] … C=0.1, gamma=0.001, score=0.637, total=   0.0s
[CV] C=0.1, gamma=0.001 …
[CV] … C=0.1, gamma=0.001, score=0.637, total=   0.0s
[CV] C=0.1, gamma=0.001 …
[CV] … C=0.1, gamma=0.001, score=0.625, total=   0.0s
[CV] C=0.1, gamma=0.001 …
[CV] … C=0.1, gamma=0.001, score=0.633, total=   0.0s
[CV] C=0.1, gamma=0.001 …
[CV] … C=0.1, gamma=0.001, score=0.633, total=   0.0s
[CV] C=0.1, gamma=0.0001 …
[CV] … C=0.1, gamma=0.0001, score=0.887, total=   0.0s
[CV] C=0.1, gamma=0.0001 …
[CV] … C=0.1, gamma=0.0001, score=0.938, total=   0.0s
[CV] C=0.1, gamma=0.0001 …
[CV] … C=0.1, gamma=0.0001, score=0.963, total=   0.0s
[CV] C=0.1, gamma=0.0001 …
[CV] … C=0.1, gamma=0.0001, score=0.962, total=   0.0s
[CV] C=0.1, gamma=0.0001 …
[CV] … C=0.1, gamma=0.0001, score=0.886, total=   0.0s
[CV] C=1, gamma=1 …
```

```
[CV] … C=1, gamma=1, score=0.637, total=   0.0s
[CV] C=1, gamma=1 …
[CV] … C=1, gamma=1, score=0.637, total=   0.0s
[CV] C=1, gamma=1 …
[CV] … C=1, gamma=1, score=0.625, total=   0.0s
[CV] C=1, gamma=1 …
[CV] … C=1, gamma=1, score=0.633, total=   0.0s
[CV] C=1, gamma=1 …
[CV] … C=1, gamma=1, score=0.633, total=   0.0s
[CV] C=1, gamma=0.1 …
[CV] … C=1, gamma=0.1, score=0.637, total=   0.0s
[CV] C=1, gamma=0.1 …
[CV] … C=1, gamma=0.1, score=0.637, total=   0.0s
[CV] C=1, gamma=0.1 …
[CV] … C=1, gamma=0.1, score=0.625, total=   0.0s
[CV] C=1, gamma=0.1 …
[CV] … C=1, gamma=0.1, score=0.633, total=   0.0s
[CV] C=1, gamma=0.1 …
[CV] … C=1, gamma=0.1, score=0.633, total=   0.0s
[CV] C=1, gamma=0.01 …
[CV] … C=1, gamma=0.01, score=0.637, total=   0.0s
[CV] C=1, gamma=0.01 …
[CV] … C=1, gamma=0.01, score=0.637, total=   0.0s
[CV] C=1, gamma=0.01 …
[CV] … C=1, gamma=0.01, score=0.625, total=   0.0s
[CV] C=1, gamma=0.01 …
[CV] … C=1, gamma=0.01, score=0.633, total=   0.0s
[CV] C=1, gamma=0.01 …
[CV] … C=1, gamma=0.01, score=0.633, total=   0.0s
[CV] C=1, gamma=0.001 …
[CV] … C=1, gamma=0.001, score=0.900, total=   0.0s
[CV] C=1, gamma=0.001 …
[CV] … C=1, gamma=0.001, score=0.912, total=   0.0s
[CV] C=1, gamma=0.001 …
[CV] … C=1, gamma=0.001, score=0.925, total=   0.0s
[CV] C=1, gamma=0.001 …
[CV] … C=1, gamma=0.001, score=0.962, total=   0.0s
[CV] C=1, gamma=0.001 …
[CV] … C=1, gamma=0.001, score=0.937, total=   0.0s
[CV] C=1, gamma=0.0001 …
[CV] … C=1, gamma=0.0001, score=0.912, total=   0.0s
[CV] C=1, gamma=0.0001 …
[CV] … C=1, gamma=0.0001, score=0.950, total=   0.0s
[CV] C=1, gamma=0.0001 …
[CV] … C=1, gamma=0.0001, score=0.975, total=   0.0s
[CV] C=1, gamma=0.0001 …
[CV] … C=1, gamma=0.0001, score=0.962, total=   0.0s
[CV] C=1, gamma=0.0001 …
```

```
[CV] … C=1, gamma=0.0001, score=0.937, total=   0.0s
[CV] C=10, gamma=1 …
[CV] … C=10, gamma=1, score=0.637, total=   0.0s
[CV] C=10, gamma=1 …
[CV] … C=10, gamma=1, score=0.637, total=   0.0s
[CV] C=10, gamma=1 …
[CV] … C=10, gamma=1, score=0.625, total=   0.0s
[CV] C=10, gamma=1 …
[CV] … C=10, gamma=1, score=0.633, total=   0.0s
[CV] C=10, gamma=1 …
[CV] … C=10, gamma=1, score=0.633, total=   0.0s
[CV] C=10, gamma=0.1 …
[CV] … C=10, gamma=0.1, score=0.637, total=   0.0s
[CV] C=10, gamma=0.1 …
[CV] … C=10, gamma=0.1, score=0.637, total=   0.0s
[CV] C=10, gamma=0.1 …
[CV] … C=10, gamma=0.1, score=0.625, total=   0.0s
[CV] C=10, gamma=0.1 …
[CV] … C=10, gamma=0.1, score=0.633, total=   0.0s
[CV] C=10, gamma=0.1 …
[CV] … C=10, gamma=0.1, score=0.633, total=   0.0s
[CV] C=10, gamma=0.01 …
[CV] … C=10, gamma=0.01, score=0.637, total=   0.0s
[CV] C=10, gamma=0.01 …
[CV] … C=10, gamma=0.01, score=0.637, total=   0.0s
[CV] C=10, gamma=0.01 …
[CV] … C=10, gamma=0.01, score=0.613, total=   0.0s
[CV] C=10, gamma=0.01 …
[CV] … C=10, gamma=0.01, score=0.633, total=   0.0s
[CV] C=10, gamma=0.01 …
[CV] … C=10, gamma=0.01, score=0.633, total=   0.0s
[CV] C=10, gamma=0.001 …
[CV] … C=10, gamma=0.001, score=0.887, total=   0.0s
[CV] C=10, gamma=0.001 …
[CV] … C=10, gamma=0.001, score=0.912, total=   0.0s
[CV] C=10, gamma=0.001 …
[CV] … C=10, gamma=0.001, score=0.900, total=   0.0s
[CV] C=10, gamma=0.001 …
[CV] … C=10, gamma=0.001, score=0.937, total=   0.0s
[CV] C=10, gamma=0.001 …
[CV] … C=10, gamma=0.001, score=0.924, total=   0.0s
[CV] C=10, gamma=0.0001 …
[CV] … C=10, gamma=0.0001, score=0.950, total=   0.0s
[CV] C=10, gamma=0.0001 …
[CV] … C=10, gamma=0.0001, score=0.912, total=   0.0s
[CV] C=10, gamma=0.0001 …
[CV] … C=10, gamma=0.0001, score=0.975, total=   0.0s
[CV] C=10, gamma=0.0001 …
```

```
[CV] … C=10, gamma=0.0001, score=0.949, total=   0.0s
[CV] C=10, gamma=0.0001 …
[CV] … C=10, gamma=0.0001, score=0.949, total=   0.0s
[CV] C=100, gamma=1 …
[CV] … C=100, gamma=1, score=0.637, total=   0.0s
[CV] C=100, gamma=1 …
[CV] … C=100, gamma=1, score=0.637, total=   0.0s
[CV] C=100, gamma=1 …
[CV] … C=100, gamma=1, score=0.625, total=   0.0s
[CV] C=100, gamma=1 …
[CV] … C=100, gamma=1, score=0.633, total=   0.0s
[CV] C=100, gamma=1 …
[CV] … C=100, gamma=1, score=0.633, total=   0.0s
[CV] C=100, gamma=0.1 …
[CV] … C=100, gamma=0.1, score=0.637, total=   0.0s
[CV] C=100, gamma=0.1 …
[CV] … C=100, gamma=0.1, score=0.637, total=   0.0s
[CV] C=100, gamma=0.1 …
[CV] … C=100, gamma=0.1, score=0.625, total=   0.0s
[CV] C=100, gamma=0.1 …
[CV] … C=100, gamma=0.1, score=0.633, total=   0.0s
[CV] C=100, gamma=0.1 …
[CV] … C=100, gamma=0.1, score=0.633, total=   0.0s
[CV] C=100, gamma=0.01 …
[CV] … C=100, gamma=0.01, score=0.637, total=   0.0s
[CV] C=100, gamma=0.01 …
[CV] … C=100, gamma=0.01, score=0.637, total=   0.0s
[CV] C=100, gamma=0.01 …
[CV] … C=100, gamma=0.01, score=0.613, total=   0.0s
[CV] C=100, gamma=0.01 …
[CV] … C=100, gamma=0.01, score=0.633, total=   0.0s
[CV] C=100, gamma=0.01 …
[CV] … C=100, gamma=0.01, score=0.633, total=   0.0s
[CV] C=100, gamma=0.001 …
[CV] … C=100, gamma=0.001, score=0.887, total=   0.0s
[CV] C=100, gamma=0.001 …
[CV] … C=100, gamma=0.001, score=0.912, total=   0.0s
[CV] C=100, gamma=0.001 …
[CV] … C=100, gamma=0.001, score=0.900, total=   0.0s
[CV] C=100, gamma=0.001 …
[CV] … C=100, gamma=0.001, score=0.937, total=   0.0s
[CV] C=100, gamma=0.001 …
[CV] … C=100, gamma=0.001, score=0.924, total=   0.0s
[CV] C=100, gamma=0.0001 …
[CV] … C=100, gamma=0.0001, score=0.925, total=   0.0s
[CV] C=100, gamma=0.0001 …
[CV] … C=100, gamma=0.0001, score=0.912, total=   0.0s
[CV] C=100, gamma=0.0001 …
```

```
[CV] … C=100, gamma=0.0001, score=0.975, total=   0.0s
[CV] C=100, gamma=0.0001 …
[CV] … C=100, gamma=0.0001, score=0.937, total=   0.0s
[CV] C=100, gamma=0.0001 …
[CV] … C=100, gamma=0.0001, score=0.949, total=   0.0s
[CV] C=1000, gamma=1 …
[CV] … C=1000, gamma=1, score=0.637, total=   0.0s
[CV] C=1000, gamma=1 …
[CV] … C=1000, gamma=1, score=0.637, total=   0.0s
[CV] C=1000, gamma=1 …
[CV] … C=1000, gamma=1, score=0.625, total=   0.0s
[CV] C=1000, gamma=1 …
[CV] … C=1000, gamma=1, score=0.633, total=   0.0s
[CV] C=1000, gamma=1 …
[CV] … C=1000, gamma=1, score=0.633, total=   0.0s
[CV] C=1000, gamma=0.1 …
[CV] … C=1000, gamma=0.1, score=0.637, total=   0.0s
[CV] C=1000, gamma=0.1 …
[CV] … C=1000, gamma=0.1, score=0.637, total=   0.0s
[CV] C=1000, gamma=0.1 …
[CV] … C=1000, gamma=0.1, score=0.625, total=   0.0s
[CV] C=1000, gamma=0.1 …
[CV] … C=1000, gamma=0.1, score=0.633, total=   0.0s
[CV] C=1000, gamma=0.1 …
[CV] … C=1000, gamma=0.1, score=0.633, total=   0.0s
[CV] C=1000, gamma=0.01 …
[CV] … C=1000, gamma=0.01, score=0.637, total=   0.0s
[CV] C=1000, gamma=0.01 …
[CV] … C=1000, gamma=0.01, score=0.637, total=   0.0s
[CV] C=1000, gamma=0.01 …
[CV] … C=1000, gamma=0.01, score=0.613, total=   0.0s
[CV] C=1000, gamma=0.01 …
[CV] … C=1000, gamma=0.01, score=0.633, total=   0.0s
[CV] C=1000, gamma=0.01 …
[CV] … C=1000, gamma=0.01, score=0.633, total=   0.0s
[CV] C=1000, gamma=0.001 …
[CV] … C=1000, gamma=0.001, score=0.887, total=   0.0s
[CV] C=1000, gamma=0.001 …
[CV] … C=1000, gamma=0.001, score=0.912, total=   0.0s
[CV] C=1000, gamma=0.001 …
[CV] … C=1000, gamma=0.001, score=0.900, total=   0.0s
[CV] C=1000, gamma=0.001 …
[CV] … C=1000, gamma=0.001, score=0.937, total=   0.0s
[CV] C=1000, gamma=0.001 …
[CV] … C=1000, gamma=0.001, score=0.924, total=   0.0s
[CV] C=1000, gamma=0.0001 …
[CV] … C=1000, gamma=0.0001, score=0.938, total=   0.0s
[CV] C=1000, gamma=0.0001 …
```

```
[CV] … C=1000, gamma=0.0001, score=0.912, total=   0.0s
[CV] C=1000, gamma=0.0001 …
[CV] … C=1000, gamma=0.0001, score=0.963, total=   0.0s
[CV] C=1000, gamma=0.0001 …
[CV] … C=1000, gamma=0.0001, score=0.924, total=   0.0s
[CV] C=1000, gamma=0.0001 …
[CV] … C=1000, gamma=0.0001, score=0.962, total=   0.0s
[Parallel(n_jobs=1)]: Done 125 out of 125 | elapsed:    3.1s finished
```

[27]:
```
GridSearchCV(estimator=SVC(),
             param_grid={'C': [0.1, 1, 10, 100, 1000],
                         'gamma': [1, 0.1, 0.01, 0.001, 0.0001]},
             verbose=5)
```

[28]:
```
#best combination
grid.best_params_
```

[28]:
```
{'C': 1, 'gamma': 0.0001}
```

[29]:
```
grid.best_estimator_
```

[29]:
```
SVC(C=1, gamma=0.0001)
```

[30]:
```
gpred = grid.predict(X_test)
```

[31]:
```
confusion_matrix(y_test, gpred)
```

[31]:
```
array([[ 59,   7],
       [  4, 101]], dtype=int64)
```

[32]:
```
print(classification_report(y_test, gpred))
```

```
              precision    recall  f1-score   support

           0       0.94      0.89      0.91        66
           1       0.94      0.96      0.95       105

    accuracy                           0.94       171
   macro avg       0.94      0.93      0.93       171
weighted avg       0.94      0.94      0.94       171
```

[ ]: