In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:
```
from sklearn.datasets import load_breast_cancer
```

In [3]:
```
cancer = load_breast_cancer()
```

In [5]:
```
cancer.keys()
```

Out[5]:
```
dict_keys(['data', 'target', 'target_names', 'DESCR', 'feature_names'])
```

In [6]:
```
print(cancer['DESCR'])
```

```
.. _breast_cancer_dataset:

Breast cancer wisconsin (diagnostic) dataset
--------------------------------------------

Notes
-----
Data Set Characteristics:
    :Number of Instances: 569

    :Number of Attributes: 30 numeric, predictive attributes and the class

    :Attribute Information:
        - radius (mean of distances from center to points on the perimeter)
        - texture (standard deviation of gray-scale values)
        - perimeter
        - area
        - smoothness (local variation in radius lengths)
        - compactness (perimeter^2 / area - 1.0)
        - concavity (severity of concave portions of the contour)
        - concave points (number of concave portions of the contour)
        - symmetry
        - fractal dimension ("coastline approximation" - 1)
```

In [7]:
```
df_feat = pd.DataFrame(cancer['data'],columns=cancer['feature_names'])
```

In [8]:
```
df_feat.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 30 columns):
mean radius                569 non-null float64
mean texture               569 non-null float64
mean perimeter             569 non-null float64
mean area                  569 non-null float64
mean smoothness            569 non-null float64
mean compactness           569 non-null float64
mean concavity             569 non-null float64
mean concave points        569 non-null float64
mean symmetry              569 non-null float64
mean fractal dimension     569 non-null float64
radius error               569 non-null float64
texture error              569 non-null float64
perimeter error            569 non-null float64
area error                 569 non-null float64
smoothness error           569 non-null float64
compactness error          569 non-null float64
concavity error            569 non-null float64
concave points error       569 non-null float64
symmetry error             569 non-null float64
fractal dimension error    569 non-null float64
worst radius               569 non-null float64
worst texture              569 non-null float64
worst perimeter            569 non-null float64
worst area                 569 non-null float64
worst smoothness           569 non-null float64
worst compactness          569 non-null float64
worst concavity            569 non-null float64
worst concave points       569 non-null float64
worst symmetry             569 non-null float64
worst fractal dimension    569 non-null float64
dtypes: float64(30)
memory usage: 133.4 KB
```

In [10]:
```
cancer['target_names']
```

Out[10]:
```
array(['malignant', 'benign'], dtype='<U9')
```

In [11]:
```
from sklearn.cross_validation import train_test_split
```

In [14]:
```
X =df_feat
y= cancer['target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_
```

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

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

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

Out[17]:
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

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

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

In [20]:
```
print(confusion_matrix(y_test,predictions))
print('\n')
print(classification_report(y_test,predictions))
```

```
[[  0  71]
 [  0 117]]


             precision    recall  f1-score   support

          0       0.00      0.00      0.00        71
          1       0.62      1.00      0.77       117

avg / total       0.39      0.62      0.48       188


C:\Users\q21\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:113
5: UndefinedMetricWarning: Precision and F-score are ill-defined and being set
to 0.0 in labels with no predicted samples.
  'precision', 'predicted', average, warn_for)
```

In [21]:
```
# grid allow find  right  parameters
```

In [22]:
```
from sklearn.grid_search import GridSearchCV
```

In [23]:
```
param_grid = {'C':[0.1,1,10,100],'gamma':[1,0.1,0.01,0.001]}
```

In [24]:
```
grid = GridSearchCV(SVC(),param_grid,verbose =3)
```

In [25]:
```
grid.fit(X_train,y_train)
```

```
[CV] ..................... C=100, gamma=0.01, score=0.629921 -   0.0s
[CV] C=100, gamma=0.01 ................................................
[CV] ..................... C=100, gamma=0.01, score=0.629921 -   0.0s
[CV] C=100, gamma=0.001 ...............................................
[CV] ..................... C=100, gamma=0.001, score=0.905512 -   0.0s
[CV] C=100, gamma=0.001 ...............................................
[CV] ..................... C=100, gamma=0.001, score=0.921260 -   0.0s
[CV] C=100, gamma=0.001 ...............................................
[CV] ..................... C=100, gamma=0.001, score=0.937008 -   0.0s

[Parallel(n_jobs=1)]: Done  48 out of  48 | elapsed:    0.5s finished
```

Out[25]:
```
GridSearchCV(cv=None, error_score='raise',
       estimator=SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False),
       fit_params={}, iid=True, n_jobs=1,
       param_grid={'C': [0.1, 1, 10, 100], 'gamma': [1, 0.1, 0.01, 0.001]},
       pre_dispatch='2*n_jobs', refit=True, scoring=None, verbose=3)
```

In [26]:
```
grid.best_params_
```

Out[26]:
```
{'C': 1, 'gamma': 0.001}
```

In [27]:
```
grid.best_estimator_
```

Out[27]:
```
SVC(C=1, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma=0.001, kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

In [28]:
```
grid_predictions = grid.predict(X_test)
```

In [29]:
```
print(confusion_matrix(y_test,grid_predictions))
print('\n')
print(classification_report(y_test,grid_predictions))
```

```
[[ 63   8]
 [  7 110]]


             precision    recall  f1-score   support

          0       0.90      0.89      0.89        71
          1       0.93      0.94      0.94       117

avg / total       0.92      0.92      0.92       188
```

In [ ]:
```

```