Classification of Breast Cancer based on SUPPORT VECTOR MACHINES

Hanah Chang

1. Introduction

In this project, we are going to predict if the cancer diagnosis is benign or malignant. The dataset is from (https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic) Examples of key variables are:

```
- 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)
```

Our target variable is binary, Maliganat or Benign

2. Data & Libaray

By looking at types of our data, we know that our dataset 'Cancer' is a Bunch object exposing six keys. The shape of a key'data' tells us there are 569 observations and 30 independent variables.

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

In [2]:
```
# import dataset
from sklearn.datasets import load_breast_cancer
cancer = load_breast_cancer()
```

In [3]:
```
print(type(cancer))
print(cancer.keys())
print(cancer['data'].shape)
```

```
<class 'sklearn.utils.Bunch'>
dict_keys(['data', 'target', 'target_names', 'DESCR', 'feature_names', 'filename'])
(569, 30)
```

Now we turn our Bunch object into dataframe

In [4]:
```
df_cancer = pd.DataFrame(np.c_[cancer['data'], cancer['target']], columns = np.append(cancer['feature_names'], ['tar
get']))
```

In [5]:
```
df_cancer.head()
```

Out[5]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst texture	worst perimeter	worst area	worst smoothness	comp
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	...	17.33	184.60	2019.0	0.1622	
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	...	23.41	158.80	1956.0	0.1238	
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	...	25.53	152.50	1709.0	0.1444	
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	...	26.50	98.87	567.7	0.2098	
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	...	16.67	152.20	1575.0	0.1374	

5 rows × 31 columns

In [6]:
```
df_cancer.columns
```

Out[6]:
```
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',
       'target'],
      dtype='object')
```

3. Explanatory Analysis & Data Cleaning

By looking at scatter plots we can have a sense of whether classes are pretty separable. In most cases, classes are separable, except for mean smoothness. Mean/median for both classes(0,1) look similar for mean smoothness

In [7]:
```
sns.pairplot(df_cancer, hue = 'target', vars = ['mean radius', 'mean texture', 'mean area', 'mean perimeter', 'mean
 smoothness'] )
```

Out[7]: <seaborn.axisgrid.PairGrid at 0x160d2da5408>

Out of total 569 patients, 357 (62.7)% diagnosed as malagnant

In [8]:
```
df_cancer.target.value_counts()
```

Out[8]:
```
1.0    357
0.0    212
Name: target, dtype: int64
```

Prior to training our model, we are going to normalize our data. By looking at range, we know that the data needs to be scaled in a way that it ranges from 0 to 1 without distorting differences in the ranges of values. Otherwise, attributed 'mean area' for instance, will intrinsically influence the result more due to its larger value.

In [9]:
```
df_range = pd.DataFrame(df_cancer.max() - df_cancer.min())
df_range
```

Out[9]:

	0
mean radius	21.129000
mean texture	29.570000
mean perimeter	144.710000
mean area	2357.500000
mean smoothness	0.110770
mean compactness	0.326020
mean concavity	0.426800
mean concave points	0.201200
mean symmetry	0.198000
mean fractal dimension	0.047480
radius error	2.761500
texture error	4.524800
perimeter error	21.223000
area error	535.398000
smoothness error	0.029417
compactness error	0.133148
concavity error	0.396000
concave points error	0.052790
symmetry error	0.071068
fractal dimension error	0.028945
worst radius	28.110000
worst texture	37.520000
worst perimeter	200.790000
worst area	4068.800000
worst smoothness	0.151430
worst compactness	1.030710
worst concavity	1.252000
worst concave points	0.291000
worst symmetry	0.507300
worst fractal dimension	0.152460
target	1.000000

In [10]:
```
df_cancer_scaled = (df_cancer-df_cancer.min())/(df_cancer.max() - df_cancer.min())
df_cancer_scaled.head()
```

Out[10]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst texture	worst perimeter	worst area	w smoothn
0	0.521037	0.022658	0.545989	0.363733	0.593753	0.792037	0.703140	0.731113	0.686364	0.605518	...	0.141525	0.668310	0.450698	0.60
1	0.643144	0.272574	0.615783	0.501591	0.289880	0.181768	0.203608	0.348757	0.379798	0.141323	...	0.303571	0.539818	0.435214	0.34
2	0.601496	0.390260	0.595743	0.449417	0.514309	0.431017	0.462512	0.635686	0.509596	0.211247	...	0.360075	0.508442	0.374508	0.48
3	0.210090	0.360839	0.233501	0.102906	0.811321	0.811361	0.565604	0.522863	0.776263	1.000000	...	0.385928	0.241347	0.094008	0.91
4	0.629893	0.156578	0.630986	0.489290	0.430351	0.347893	0.463918	0.518390	0.378283	0.186816	...	0.123934	0.506948	0.341575	0.43

5 rows × 31 columns

4. Support Vector Machine - Training / Optimizing

In [11]:
```
X = df_cancer_scaled.drop(['target'],axis=1)
y = df_cancer_scaled['target']
```

In [12]:
```
# random split our data into train/test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state=123)
```

In [13]:
```
print("X_train:", X_train.shape, "y_train:", y_train.shape, "X_test:", X_test.shape, "y_test:", y_test.shape, )
```

```
X_train: (455, 30) y_train: (455,) X_test: (114, 30) y_test: (114,)
```

Fit our model using training data

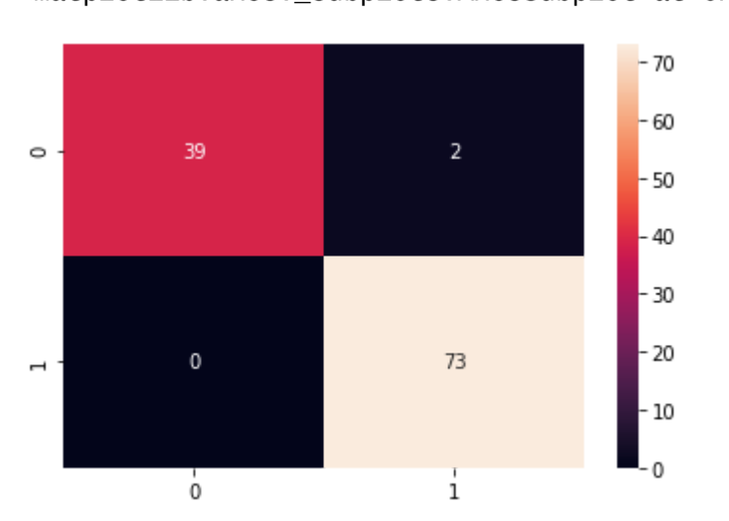
In [14]:
```
from sklearn.svm import SVC
svc_model = SVC()
svc_model.fit(X_train, y_train)
```

Out[14]:
```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

The average accuracy of the model is 98% and we accurately predicted 112 cases.

In [15]:
```
from sklearn.metrics import classification_report, confusion_matrix
y_predict = svc_model.predict(X_test)
cm = confusion_matrix(y_test, y_predict)
sns.heatmap(cm, annot=True)
```

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x160d44875c8>



In [16]:
```
print(classification_report(y_test, y_predict))
```

```
              precision    recall  f1-score   support

         0.0       1.00      0.95      0.97        41
         1.0       0.97      1.00      0.99        73

    accuracy                           0.98       114
   macro avg       0.99      0.98      0.98       114
weighted avg       0.98      0.98      0.98       114
```

In order to more generalize our model to prevent overfit, let's iterate algorithm using different values of C and Gamma parameters. We expect by decreasing C parameter we less penalize misclassifcation and by decreaseing Gamma, the decision boundary gets smoother.

In [17]:
```
# grids for all possible c, gamma premeters
grid = {'C': [0.01, 0.1, 1, 10, 100], 'gamma': [1, 0.1, 0.01, 0.001, 0.0001], 'kernel': ['rbf']}
```

In [18]:
```
from sklearn.model_selection import GridSearchCV
refit_model = GridSearchCV(SVC(),grid,refit=True,verbose=4) #verbose: num of details we wna to show
```

As the algorithm iterates, we can see that scores are changing. The best result were achievbed with C parameter 10, and Gamma parameter 0.1

In [19]:
```
refit_model.fit(X_train,y_train)
```

```
Fitting 5 folds for each of 25 candidates, totalling 125 fits
[CV] C=0.01, gamma=1, kernel=rbf ....................................
[CV] ......... C=0.01, gamma=1, kernel=rbf, score=0.615, total=   0.0s
[CV] C=0.01, gamma=1, kernel=rbf ....................................
[CV] ......... C=0.01, gamma=1, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.01, gamma=1, kernel=rbf ....................................
[CV] ......... C=0.01, gamma=1, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.01, gamma=1, kernel=rbf ....................................
[CV] ......... C=0.01, gamma=1, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.01, gamma=1, kernel=rbf ....................................
[CV] ......... C=0.01, gamma=1, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.01, gamma=0.1, kernel=rbf ..................................
[CV] ....... C=0.01, gamma=0.1, kernel=rbf, score=0.615, total=   0.0s
[CV] C=0.01, gamma=0.1, kernel=rbf ..................................
[CV] ....... C=0.01, gamma=0.1, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.01, gamma=0.1, kernel=rbf ..................................
[CV] ....... C=0.01, gamma=0.1, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.01, gamma=0.1, kernel=rbf ..................................
[CV] ....... C=0.01, gamma=0.1, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.01, gamma=0.1, kernel=rbf ..................................
[CV] ....... C=0.01, gamma=0.1, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.01, gamma=0.01, kernel=rbf .................................
[CV] ...... C=0.01, gamma=0.01, kernel=rbf, score=0.615, total=   0.1s
[CV] C=0.01, gamma=0.01, kernel=rbf .................................
[CV] ...... C=0.01, gamma=0.01, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.01, gamma=0.01, kernel=rbf .................................
[CV] ...... C=0.01, gamma=0.01, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.01, gamma=0.01, kernel=rbf .................................
[CV] ...... C=0.01, gamma=0.01, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.01, gamma=0.01, kernel=rbf .................................
```

```
[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
```

```
[CV] ...... C=0.01, gamma=0.01, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.01, gamma=0.001, kernel=rbf ................................
[CV] ..... C=0.01, gamma=0.001, kernel=rbf, score=0.615, total=   0.0s
[CV] C=0.01, gamma=0.001, kernel=rbf ................................
[CV] ..... C=0.01, gamma=0.001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.01, gamma=0.001, kernel=rbf ................................
[CV] ..... C=0.01, gamma=0.001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.01, gamma=0.001, kernel=rbf ................................
[CV] ..... C=0.01, gamma=0.001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.01, gamma=0.001, kernel=rbf ................................
[CV] ..... C=0.01, gamma=0.001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.01, gamma=0.0001, kernel=rbf ...............................
[CV] .... C=0.01, gamma=0.0001, kernel=rbf, score=0.615, total=   0.0s
[CV] C=0.01, gamma=0.0001, kernel=rbf ...............................
[CV] .... C=0.01, gamma=0.0001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.01, gamma=0.0001, kernel=rbf ...............................
[CV] .... C=0.01, gamma=0.0001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.01, gamma=0.0001, kernel=rbf ...............................
[CV] .... C=0.01, gamma=0.0001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.01, gamma=0.0001, kernel=rbf ...............................
[CV] .... C=0.01, gamma=0.0001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.1, gamma=1, kernel=rbf .....................................
[CV] .......... C=0.1, gamma=1, kernel=rbf, score=0.978, total=   0.0s
[CV] C=0.1, gamma=1, kernel=rbf .....................................
[CV] .......... C=0.1, gamma=1, kernel=rbf, score=0.934, total=   0.0s
[CV] C=0.1, gamma=1, kernel=rbf .....................................
[CV] .......... C=0.1, gamma=1, kernel=rbf, score=0.923, total=   0.0s
[CV] C=0.1, gamma=1, kernel=rbf .....................................
[CV] .......... C=0.1, gamma=1, kernel=rbf, score=0.956, total=   0.0s
[CV] C=0.1, gamma=1, kernel=rbf .....................................
[CV] .......... C=0.1, gamma=1, kernel=rbf, score=0.934, total=   0.0s
[CV] C=0.1, gamma=0.1, kernel=rbf ...................................
[CV] ........ C=0.1, gamma=0.1, kernel=rbf, score=0.945, total=   0.0s
[CV] C=0.1, gamma=0.1, kernel=rbf ...................................
[CV] ........ C=0.1, gamma=0.1, kernel=rbf, score=0.934, total=   0.0s
[CV] C=0.1, gamma=0.1, kernel=rbf ...................................
[CV] ........ C=0.1, gamma=0.1, kernel=rbf, score=0.890, total=   0.0s
[CV] C=0.1, gamma=0.1, kernel=rbf ...................................
[CV] ........ C=0.1, gamma=0.1, kernel=rbf, score=0.879, total=   0.0s
[CV] C=0.1, gamma=0.1, kernel=rbf ...................................
[CV] ........ C=0.1, gamma=0.1, kernel=rbf, score=0.890, total=   0.0s
[CV] C=0.1, gamma=0.01, kernel=rbf ..................................
[CV] ....... C=0.1, gamma=0.01, kernel=rbf, score=0.615, total=   0.0s
[CV] C=0.1, gamma=0.01, kernel=rbf ..................................
[CV] ....... C=0.1, gamma=0.01, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.1, gamma=0.01, kernel=rbf ..................................
[CV] ....... C=0.1, gamma=0.01, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.1, gamma=0.01, kernel=rbf ..................................
[CV] ....... C=0.1, gamma=0.01, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.1, gamma=0.01, kernel=rbf ..................................
[CV] ....... C=0.1, gamma=0.01, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.1, gamma=0.001, kernel=rbf .................................
[CV] ...... C=0.1, gamma=0.001, kernel=rbf, score=0.615, total=   0.0s
[CV] C=0.1, gamma=0.001, kernel=rbf .................................
[CV] ...... C=0.1, gamma=0.001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.1, gamma=0.001, kernel=rbf .................................
[CV] ...... C=0.1, gamma=0.001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.1, gamma=0.001, kernel=rbf .................................
[CV] ...... C=0.1, gamma=0.001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.1, gamma=0.001, kernel=rbf .................................
[CV] ...... C=0.1, gamma=0.001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.1, gamma=0.0001, kernel=rbf ................................
[CV] ..... C=0.1, gamma=0.0001, kernel=rbf, score=0.615, total=   0.0s
[CV] C=0.1, gamma=0.0001, kernel=rbf ................................
[CV] ..... C=0.1, gamma=0.0001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.1, gamma=0.0001, kernel=rbf ................................
[CV] ..... C=0.1, gamma=0.0001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.1, gamma=0.0001, kernel=rbf ................................
[CV] ..... C=0.1, gamma=0.0001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=0.1, gamma=0.0001, kernel=rbf ................................
[CV] ..... C=0.1, gamma=0.0001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=1, gamma=1, kernel=rbf .......................................
[CV] ............ C=1, gamma=1, kernel=rbf, score=0.989, total=   0.0s
[CV] C=1, gamma=1, kernel=rbf .......................................
[CV] ............ C=1, gamma=1, kernel=rbf, score=0.956, total=   0.0s
[CV] C=1, gamma=1, kernel=rbf .......................................
[CV] ............ C=1, gamma=1, kernel=rbf, score=0.978, total=   0.0s
[CV] C=1, gamma=1, kernel=rbf .......................................
[CV] ............ C=1, gamma=1, kernel=rbf, score=0.978, total=   0.0s
[CV] C=1, gamma=1, kernel=rbf .......................................
[CV] ............ C=1, gamma=1, kernel=rbf, score=0.967, total=   0.0s
[CV] C=1, gamma=0.1, kernel=rbf .....................................
[CV] .......... C=1, gamma=0.1, kernel=rbf, score=0.978, total=   0.0s
[CV] C=1, gamma=0.1, kernel=rbf .....................................
[CV] .......... C=1, gamma=0.1, kernel=rbf, score=0.934, total=   0.0s
[CV] C=1, gamma=0.1, kernel=rbf .....................................
[CV] .......... C=1, gamma=0.1, kernel=rbf, score=0.945, total=   0.0s
[CV] C=1, gamma=0.1, kernel=rbf .....................................
[CV] .......... C=1, gamma=0.1, kernel=rbf, score=0.945, total=   0.0s
[CV] C=1, gamma=0.1, kernel=rbf .....................................
[CV] .......... C=1, gamma=0.1, kernel=rbf, score=0.945, total=   0.0s
[CV] C=1, gamma=0.01, kernel=rbf ....................................
[CV] ......... C=1, gamma=0.01, kernel=rbf, score=0.956, total=   0.0s
[CV] C=1, gamma=0.01, kernel=rbf ....................................
[CV] ......... C=1, gamma=0.01, kernel=rbf, score=0.934, total=   0.0s
[CV] C=1, gamma=0.01, kernel=rbf ....................................
[CV] ......... C=1, gamma=0.01, kernel=rbf, score=0.890, total=   0.0s
[CV] C=1, gamma=0.01, kernel=rbf ....................................
[CV] ......... C=1, gamma=0.01, kernel=rbf, score=0.879, total=   0.0s
[CV] C=1, gamma=0.01, kernel=rbf ....................................
[CV] ......... C=1, gamma=0.01, kernel=rbf, score=0.890, total=   0.0s
[CV] C=1, gamma=0.001, kernel=rbf ...................................
[CV] ........ C=1, gamma=0.001, kernel=rbf, score=0.615, total=   0.0s
[CV] C=1, gamma=0.001, kernel=rbf ...................................
[CV] ........ C=1, gamma=0.001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=1, gamma=0.001, kernel=rbf ...................................
[CV] ........ C=1, gamma=0.001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=1, gamma=0.001, kernel=rbf ...................................
[CV] ........ C=1, gamma=0.001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=1, gamma=0.001, kernel=rbf ...................................
[CV] ........ C=1, gamma=0.001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=1, gamma=0.0001, kernel=rbf ..................................
[CV] ....... C=1, gamma=0.0001, kernel=rbf, score=0.615, total=   0.0s
[CV] C=1, gamma=0.0001, kernel=rbf ..................................
[CV] ....... C=1, gamma=0.0001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=1, gamma=0.0001, kernel=rbf ..................................
[CV] ....... C=1, gamma=0.0001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=1, gamma=0.0001, kernel=rbf ..................................
[CV] ....... C=1, gamma=0.0001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=1, gamma=0.0001, kernel=rbf ..................................
[CV] ....... C=1, gamma=0.0001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=10, gamma=1, kernel=rbf ......................................
[CV] ........... C=10, gamma=1, kernel=rbf, score=0.989, total=   0.0s
[CV] C=10, gamma=1, kernel=rbf ......................................
[CV] ........... C=10, gamma=1, kernel=rbf, score=0.956, total=   0.0s
[CV] C=10, gamma=1, kernel=rbf ......................................
[CV] ........... C=10, gamma=1, kernel=rbf, score=1.000, total=   0.0s
[CV] C=10, gamma=1, kernel=rbf ......................................
[CV] ........... C=10, gamma=1, kernel=rbf, score=0.956, total=   0.0s
[CV] C=10, gamma=1, kernel=rbf ......................................
[CV] ........... C=10, gamma=1, kernel=rbf, score=0.967, total=   0.0s
[CV] C=10, gamma=0.1, kernel=rbf ....................................
[CV] ......... C=10, gamma=0.1, kernel=rbf, score=1.000, total=   0.0s
[CV] C=10, gamma=0.1, kernel=rbf ....................................
[CV] ......... C=10, gamma=0.1, kernel=rbf, score=0.967, total=   0.0s
[CV] C=10, gamma=0.1, kernel=rbf ....................................
[CV] ......... C=10, gamma=0.1, kernel=rbf, score=0.989, total=   0.0s
[CV] C=10, gamma=0.1, kernel=rbf ....................................
[CV] ......... C=10, gamma=0.1, kernel=rbf, score=0.989, total=   0.0s
[CV] C=10, gamma=0.1, kernel=rbf ....................................
[CV] ......... C=10, gamma=0.1, kernel=rbf, score=0.967, total=   0.0s
[CV] C=10, gamma=0.01, kernel=rbf ...................................
[CV] ........ C=10, gamma=0.01, kernel=rbf, score=0.978, total=   0.0s
[CV] C=10, gamma=0.01, kernel=rbf ...................................
[CV] ........ C=10, gamma=0.01, kernel=rbf, score=0.945, total=   0.0s
[CV] C=10, gamma=0.01, kernel=rbf ...................................
[CV] ........ C=10, gamma=0.01, kernel=rbf, score=0.956, total=   0.0s
[CV] C=10, gamma=0.01, kernel=rbf ...................................
[CV] ........ C=10, gamma=0.01, kernel=rbf, score=0.945, total=   0.0s
[CV] C=10, gamma=0.01, kernel=rbf ...................................
[CV] ........ C=10, gamma=0.01, kernel=rbf, score=0.945, total=   0.0s
[CV] C=10, gamma=0.001, kernel=rbf ..................................
[CV] ....... C=10, gamma=0.001, kernel=rbf, score=0.956, total=   0.0s
[CV] C=10, gamma=0.001, kernel=rbf ..................................
[CV] ....... C=10, gamma=0.001, kernel=rbf, score=0.934, total=   0.0s
[CV] C=10, gamma=0.001, kernel=rbf ..................................
[CV] ....... C=10, gamma=0.001, kernel=rbf, score=0.890, total=   0.0s
[CV] C=10, gamma=0.001, kernel=rbf ..................................
[CV] ....... C=10, gamma=0.001, kernel=rbf, score=0.879, total=   0.0s
[CV] C=10, gamma=0.001, kernel=rbf ..................................
[CV] ....... C=10, gamma=0.001, kernel=rbf, score=0.890, total=   0.0s
[CV] C=10, gamma=0.0001, kernel=rbf .................................
[CV] ...... C=10, gamma=0.0001, kernel=rbf, score=0.615, total=   0.0s
[CV] C=10, gamma=0.0001, kernel=rbf .................................
[CV] ...... C=10, gamma=0.0001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=10, gamma=0.0001, kernel=rbf .................................
[CV] ...... C=10, gamma=0.0001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=10, gamma=0.0001, kernel=rbf .................................
[CV] ...... C=10, gamma=0.0001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=10, gamma=0.0001, kernel=rbf .................................
[CV] ...... C=10, gamma=0.0001, kernel=rbf, score=0.626, total=   0.0s
[CV] C=100, gamma=1, kernel=rbf .....................................
[CV] .......... C=100, gamma=1, kernel=rbf, score=0.978, total=   0.0s
[CV] C=100, gamma=1, kernel=rbf .....................................
[CV] .......... C=100, gamma=1, kernel=rbf, score=0.923, total=   0.0s
[CV] C=100, gamma=1, kernel=rbf .....................................
[CV] .......... C=100, gamma=1, kernel=rbf, score=0.967, total=   0.0s
[CV] C=100, gamma=1, kernel=rbf .....................................
[CV] .......... C=100, gamma=1, kernel=rbf, score=0.934, total=   0.0s
[CV] C=100, gamma=1, kernel=rbf .....................................
[CV] .......... C=100, gamma=1, kernel=rbf, score=0.967, total=   0.0s
[CV] C=100, gamma=0.1, kernel=rbf ...................................
[CV] ........ C=100, gamma=0.1, kernel=rbf, score=0.978, total=   0.0s
[CV] C=100, gamma=0.1, kernel=rbf ...................................
[CV] ........ C=100, gamma=0.1, kernel=rbf, score=0.967, total=   0.0s
[CV] C=100, gamma=0.1, kernel=rbf ...................................
[CV] ........ C=100, gamma=0.1, kernel=rbf, score=1.000, total=   0.0s
[CV] C=100, gamma=0.1, kernel=rbf ...................................
[CV] ........ C=100, gamma=0.1, kernel=rbf, score=0.967, total=   0.0s
[CV] C=100, gamma=0.1, kernel=rbf ...................................
[CV] ........ C=100, gamma=0.1, kernel=rbf, score=0.934, total=   0.0s
[CV] C=100, gamma=0.01, kernel=rbf ..................................
[CV] ....... C=100, gamma=0.01, kernel=rbf, score=1.000, total=   0.0s
[CV] C=100, gamma=0.01, kernel=rbf ..................................
[CV] ....... C=100, gamma=0.01, kernel=rbf, score=0.967, total=   0.0s
[CV] C=100, gamma=0.01, kernel=rbf ..................................
[CV] ....... C=100, gamma=0.01, kernel=rbf, score=0.989, total=   0.0s
[CV] C=100, gamma=0.01, kernel=rbf ..................................
[CV] ....... C=100, gamma=0.01, kernel=rbf, score=0.989, total=   0.0s
[CV] C=100, gamma=0.01, kernel=rbf ..................................
[CV] ....... C=100, gamma=0.01, kernel=rbf, score=0.967, total=   0.0s
[CV] C=100, gamma=0.001, kernel=rbf .................................
[CV] ...... C=100, gamma=0.001, kernel=rbf, score=0.978, total=   0.0s
[CV] C=100, gamma=0.001, kernel=rbf .................................
[CV] ...... C=100, gamma=0.001, kernel=rbf, score=0.945, total=   0.0s
[CV] C=100, gamma=0.001, kernel=rbf .................................
[CV] ...... C=100, gamma=0.001, kernel=rbf, score=0.956, total=   0.0s
[CV] C=100, gamma=0.001, kernel=rbf .................................
[CV] ...... C=100, gamma=0.001, kernel=rbf, score=0.945, total=   0.0s
[CV] C=100, gamma=0.001, kernel=rbf .................................
[CV] ...... C=100, gamma=0.001, kernel=rbf, score=0.945, total=   0.0s
[CV] C=100, gamma=0.0001, kernel=rbf ................................
[CV] ..... C=100, gamma=0.0001, kernel=rbf, score=0.956, total=   0.0s
[CV] C=100, gamma=0.0001, kernel=rbf ................................
[CV] ..... C=100, gamma=0.0001, kernel=rbf, score=0.934, total=   0.0s
[CV] C=100, gamma=0.0001, kernel=rbf ................................
[CV] ..... C=100, gamma=0.0001, kernel=rbf, score=0.890, total=   0.0s
[CV] C=100, gamma=0.0001, kernel=rbf ................................
[CV] ..... C=100, gamma=0.0001, kernel=rbf, score=0.879, total=   0.0s
[CV] C=100, gamma=0.0001, kernel=rbf ................................
[CV] ..... C=100, gamma=0.0001, kernel=rbf, score=0.890, total=   0.0s
```

```
[Parallel(n_jobs=1)]: Done 125 out of 125 | elapsed:    0.9s finished
```

Out[19]:
```
GridSearchCV(cv=None, error_score=nan,
             estimator=SVC(C=1.0, break_ties=False, cache_size=200,
                           class_weight=None, coef0=0.0,
                           decision_function_shape='ovr', degree=3,
                           gamma='scale', kernel='rbf', max_iter=-1,
                           probability=False, random_state=None, shrinking=True,
                           tol=0.001, verbose=False),
             iid='deprecated', n_jobs=None,
             param_grid={'C': [0.01, 0.1, 1, 10, 100],
                         'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
                         'kernel': ['rbf']},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring=None, verbose=4)
```

In [20]:
```
print(refit_model.best_params_)
```

```
{'C': 10, 'gamma': 0.1, 'kernel': 'rbf'}
```

5. Support Vector Machine - Testing / Result

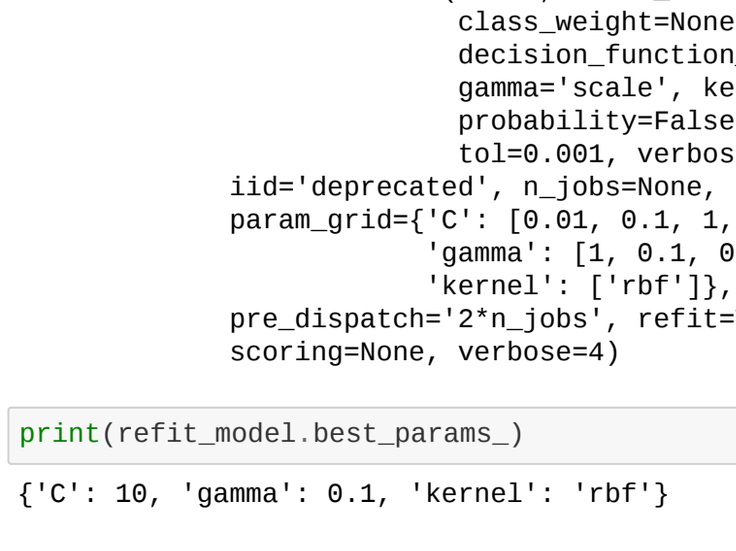
The accuracy of the model was not increased after optimization. It is likely that the accuracy before model improvement was already very high enough (98%)

In [21]:
```
y_predict2 = refit_model.predict(X_test)
```

In [22]:
```
cm = confusion_matrix(y_test, y_predict2)
```

In [23]:
```
sns.heatmap(cm, annot=True)
```

Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x160d4344708>



In [24]:
```
print(classification_report(y_test,y_predict2))
```

```
              precision    recall  f1-score   support

         0.0       1.00      0.95      0.97        41
         1.0       0.97      1.00      0.99        73

    accuracy                           0.98       114
   macro avg       0.99      0.98      0.98       114
weighted avg       0.98      0.98      0.98       114
```