Homework 3

March 12, 2018

1 Homework 3: Hyperparameter Tuning with SVMs

The final deliverable for this homework will be this Jupyter notebook, which should include all relevant code, markdown cells before each code block describing what the code does, and any write-ups/images/plots that you wish to include.

To add a block click on `Insert > Insert Cell Below`. To make a markdown cell, click the drop-down menu at the top of this page and select `Markdown`.

The starter code for this homework is purposely very minimal. You should get used to coding from scratch. Just follow all the instructions in the PDF you will be fine.

In [25]:
```
import numpy as np
import pandas as pd

from sklearn.svm import SVC
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn import preprocessing
from sklearn.metrics import accuracy_score

import matplotlib.pyplot as plt

import seaborn as sns
```

In [26]:
```
cols = ['ID', 'Clump Thickness', 'Uniformity of Cell Size',
        'Uniformity of Cell Shape', 'Marginal Adhesion',
        'Single Epithelial Cell Size', 'Bare Nuclei',
        'Bland Chromatin', 'Normal Nucleoli', 'Mitoses', 'Class']
data  = pd.read_csv("breast-cancer-wisconsin.data",names=cols)
```

In [27]:
```
#NORMALIZE DATA
data.set_index(["ID"],inplace=True)
data = (data-data.min())/(data.max()-data.min())
```

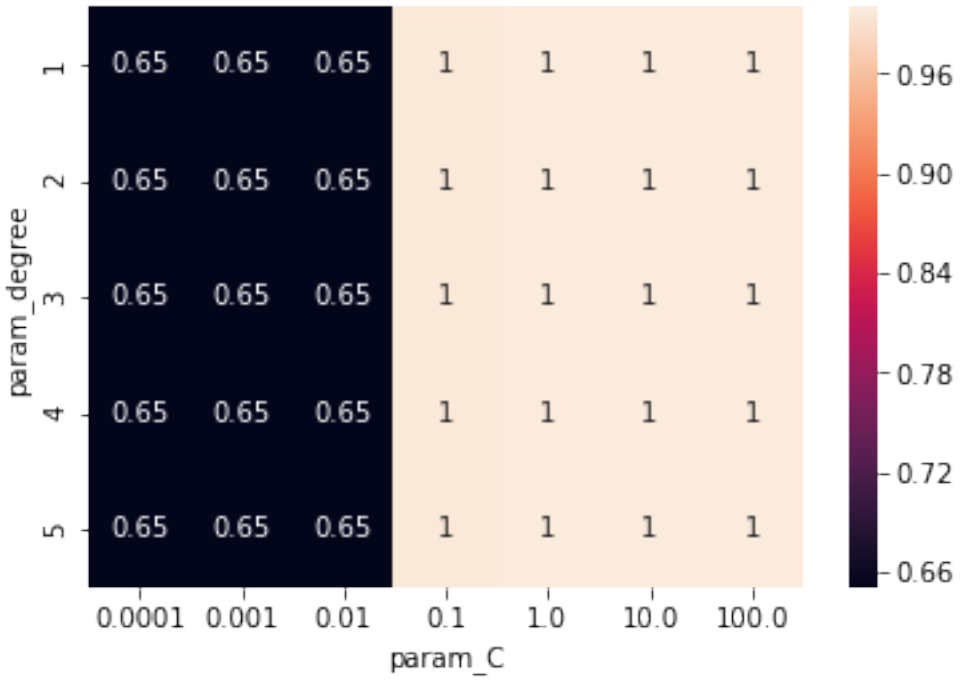
In [28]:
```
#SPLIT DATA
train_data, test_data = train_test_split(data)
```

In [29]:
```
#PREDICT
params = {
    "C": [.0001,.001,.01,.1,1,10,100],
```

```
        "degree": [1,2,3,4,5]
    }
    svc = SVC()
    clf = GridSearchCV(svc,params)
    clf.fit(train_data,train_data["Class"]);
```

In [30]:
```
#GRAPH
results = pd.DataFrame(clf.cv_results_)
pivoted = results.pivot(index="param_degree",
          columns="param_C",values="mean_test_score")
sns.heatmap(pivoted,annot=True)
print("best pair of parameters",clf.best_params_)
print("General trends: degree doesn't matter, but C values do.\nHigher C value means h
```

```
best pair of parameters {'C': 1, 'degree': 1}
General trends: degree doesn't matter, but C values do.
Higher C value means higher penalty for misclassification of data
and so higher the C value, the more the (over)fitting.
```



In [31]:
```
#PREDICT
params = {
    "C": [.0001,.001,.01,.1,1,10,100],
    "gamma":  [.001,.01,.1,1,10,100]
}
```

```
        svc = SVC()
        clf = GridSearchCV(svc,params)
        clf.fit(train_data,train_data["Class"])
        print("best pair of parameters",clf.best_params_)
```

```
best pair of parameters {'C': 1, 'gamma': 0.1}
```

In [32]:
```
        #GRAPH
        results = pd.DataFrame(clf.cv_results_)
        pivoted = results.pivot(index="param_gamma",
                  columns="param_C",values="mean_test_score")
        sns.heatmap(pivoted,annot=True);
```