

Results:

Part 1 : Digit Recognition -

Overview:

I have used SVC classifier with linear kernel for this part of assignment. Analysis is done based on varying values for C. For this dataset, the difference wasn't significant. Following are the few outputs based on variables:

SVC:

Kernel : 'linear'

C- 0.0001

Accuracy ~95%

```
Accuracy with Decision Tree..
0.861992209238
With SVM...
SVC(C=0.0001, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape=None, degree=3, gamma='auto',
    kernel='linear',
    max_iter=-1, probability=False, random_state=None,
    shrinking=True,
    tol=0.001, verbose=False)
Accuracy After Boosting..
0.9510294936
```

SVC:

Kernel : 'linear'

C- 0.00001

Accuracy ~91%

```
SVC(C=1e-05, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape=None, degree=3, gamma='auto',
    kernel='linear',
    max_iter=-1, probability=False, random_state=None,
    shrinking=True,
    tol=0.001, verbose=False)
Accuracy After SVC..
0.910962715637
```

SVC:

Kernel : 'linear'

C- 0.01

Accuracy ~96%

```
SVC(C=0.01, cache_size=200, class_weight=None, coef0=0.0,  
    decision_function_shape=None, degree=3, gamma='auto',  
    kernel='linear',  
    max_iter=-1, probability=False, random_state=None,  
    shrinking=True,  
    tol=0.001, verbose=False)
```

Accuracy After SVC..

0.961602671119

SVC:

Kernel : 'linear'

C- 1

Accuracy ~96%

```
SVC(C=1, cache_size=200, class_weight=None, coef0=0.0,  
    decision_function_shape=None, degree=3, gamma='auto',  
    kernel='linear',  
    max_iter=-1, probability=False, random_state=None,  
    shrinking=True,  
    tol=0.001, verbose=False)
```

Accuracy After SVC..

0.961602671119

SVC:

Kernel : 'linear'

C- 10

Accuracy ~96%

```
SVC(C=10, cache_size=200, class_weight=None, coef0=0.0,  
    decision_function_shape=None, degree=3, gamma='auto',  
    kernel='linear',  
    max_iter=-1, probability=False, random_state=None,  
    shrinking=True,  
    tol=0.001, verbose=False)
```

Accuracy After SVC..

0.961602671119

SVC:

Kernel : 'linear'

C- 1000

Accuracy ~96%

```
SVC(C=1000, cache_size=200, class_weight=None, coef0=0.0,  
    decision_function_shape=None, degree=3, gamma='auto',  
    kernel='linear',  
    max_iter=-1, probability=False, random_state=None,  
    shrinking=True,  
    tol=0.001, verbose=False)  
Accuracy After SVC..  
0.961602671119
```

As you can see, value of C plays an important role in accuracy of the model. For varying values of C from 0.00001 to 100000 the accuracy changes from 91% - 96%.

Part 2: Amazon Data Set:

Overview:

For this dataset as well used SVC classifier. Analysis was done based various factors. For kernel value of linear and rbf the results were observed. Also checked effect of changing value of C for linear and rbf kernel. Used value of gamma to check value for rbf kernel. The results are as follows:

SVC:

Kernel : 'linear'

C- 0.001

gamma:NA

Accuracy ~58%

Cleaning and parsing the test set movie reviews...

```
Model is ready..  
Accuracy Linear SVC..  
0.578231292517
```

SVC:

Kernel : 'linear'

C- 0.01

gamma:NA

Accuracy ~58%

```
Model is ready..  
Accuracy Linear SVC c=0.01..  
0.578231292517
```

SVC:
Kernel : 'linear'
C- 0.1
gamma:NA
Accuracy ~57%

```
Model is ready..  
Accuracy Linear SVC c=0.1..  
0.571428571429
```

SVC:
Kernel : 'linear'
C- 1
gamma:NA
Accuracy ~57%

```
In [4]: model = SVC(kernel='linear', probability=False, C=1)  
....: model=model.fit( train_data_features, train["rating"][:145035] )  
....: print("Model is ready..  
....: result = model.predict(test_data_features)  
....: #output = pd.DataFrame( data={"id":test["name"][:36208],  
"sentiment":result[:36208]})  
....: print("Accuracy Linear SVC c=1..  
....: #print(accuracy_score(output["sentiment"],test["rating"][:36208]))  
....: print(accuracy_score(result,test["rating"]))  
Model is ready..  
Accuracy Linear SVC c=1..  
0.571428571429
```

SVC:
Kernel : 'linear'
C- 10
gamma:NA
Accuracy ~57%

```
....  
Model is ready..  
Accuracy Linear SVC c=10..  
0.571428571429
```

As you can see For Linear kernel, changing value of C doesn't make much of difference.
As you can see, for values of less than 0.1 we get around 58% whereas for higher values
it is 57%. There is no significant difference in accuracy.

SVC:
Kernel : 'rbf'
C- 1
gamma:'auto'
Accuracy ~58%

```
....  
Model is ready..  
Accuracy RBF SVC c=1..  
0.578231292517
```

SVC:
Kernel : 'rbf'
C- 0.01
gamma:'auto'
Accuracy ~58%

```
....  
Model is ready..  
Accuracy RBF SVC c=0.01 gamma='auto'..  
0.578231292517
```

SVC:
Kernel : 'rbf'
C- 0.001
gamma:'auto'

Accuracy ~58%

```
....  
Model is ready..  
Accuracy RBF SVC c=0.001 gamma='auto'..  
0.578231292517
```

SVC:

Kernel : 'rbf'

C- 0.0001

gamma:'auto'

Accuracy ~58%

```
In [22]: model = SVC(kernel='rbf', probability=False, C=0.0001,  
gamma='auto')  
....: model=model.fit( train_data_features, train["rating"]  
[:145035] )  
....: print("Model is ready..")  
....: result = model.predict(test_data_features)  
....: #output = pd.DataFrame( data={"id":test["name"]  
[:36208], "sentiment":result[:36208]} )  
....: print("Accuracy RBF SVC c=0.0001 gamma='auto'..")  
....:  
#print(accuracy_score(output["sentiment"],test["rating"]  
[:36208]))  
....: print(accuracy_score(result,test["rating"]))  
Model is ready..  
Accuracy RBF SVC c=0.0001 gamma='auto'..  
0.578231292517
```

SVC:

Kernel : 'rbf'

C- 100

gamma:'auto'

Accuracy ~57%

```
....  
Model is ready..  
Accuracy RBF SVC c=100 gamma='auto'..  
0.571428571429
```

SVC:

Kernel : 'rbf'
C- 1000
gamma:'auto'
Accuracy ~57%

```
Model is ready..  
Accuracy RBF SVC c=1000 gamma='auto'..  
0.571428571429
```

SVC:
Kernel : 'rbf'
C- 1
gamma:0.1
Accuracy ~58%

```
SVC(C=1, cache_size=200, class_weight=None, coef0=0.0,  
    decision_function_shape=None, degree=3, gamma=0.1,  
    kernel='rbf',  
    max_iter=-1, probability=False, random_state=None,  
    shrinking=True,  
    tol=0.001, verbose=False)  
Model is ready..  
Accuracy RBF SVC c=1 gamma=0.1 ..  
0.578231292517
```

SVC:
Kernel : 'rbf'
C- 1
gamma:0.001
Accuracy ~58%

```

... print(accuracy_score(result, test_data))
SVC(C=1, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape=None, degree=3, gamma=0.001,
    kernel='rbf',
    max_iter=-1, probability=False, random_state=None,
    shrinking=True,
    tol=0.001, verbose=False)
Model is ready..
Accuracy RBF SVC c=1 gamma=0.1 ..
0.578231292517

```

SVC:

Kernel : 'rbf'

C- 1

gamma:0.0001

Accuracy ~58%

```

...
SVC(C=1, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape=None, degree=3, gamma=0.0001,
    kernel='rbf',
    max_iter=-1, probability=False, random_state=None,
    shrinking=True,
    tol=0.001, verbose=False)
Model is ready..
Accuracy RBF SVC c=1 gamma=0.1 ..
0.578231292517

```

SVC:

Kernel : 'rbf'

C- 0.01

gamma:0.0001

Accuracy ~58%

```

SVC(C=0.01, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape=None, degree=3, gamma=0.0001,
    kernel='rbf',
    max_iter=-1, probability=False, random_state=None,
    shrinking=True,
    tol=0.001, verbose=False)
Model is ready..
Accuracy RBF SVC c=0.01 gamma=0.0001 ..
0.578231292517

```

SVC:

Kernel : 'rbf'

C- 0.0001

gamma:0.0001

Accuracy ~58%

```
SVC(C=0.0001, cache_size=200, class_weight=None, coef0=0.0,  
    decision_function_shape=None, degree=3, gamma=0.0001,  
    kernel='rbf',  
    max_iter=-1, probability=False, random_state=None,  
    shrinking=True,  
    tol=0.001, verbose=False)
```

Model is ready..

Accuracy RBF SVC c=0.01 gamma=0.0001 ..
0.578231292517

SVC:

Kernel : 'rbf'

C- 100

gamma:0.0001

Accuracy ~56%

```
SVC(C=100, cache_size=200, class_weight=None, coef0=0.0,  
    decision_function_shape=None, degree=3, gamma=0.0001,  
    kernel='rbf',  
    max_iter=-1, probability=False, random_state=None,  
    shrinking=True,  
    tol=0.001, verbose=False)
```

Model is ready..

Accuracy RBF SVC c=0.01 gamma=0.0001 ..
0.56462585034

SVC:

Kernel : 'rbf'

C- 1000

gamma:0.0001

Accuracy ~57%

```

.....
SVC(C=1000, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape=None, degree=3, gamma=0.0001,
    kernel='rbf',
    max_iter=-1, probability=False, random_state=None,
    shrinking=True,
    tol=0.001, verbose=False)
Model is ready..
Accuracy RBF SVC c=0.01 gamma=0.0001 ..
0.571428571429

```

SVC:

Kernel : 'rbf'

C- 10000

gamma:0.0001

Accuracy ~57%

```

SVC(C=10000, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape=None, degree=3, gamma=0.0001,
    kernel='rbf',
    max_iter=-1, probability=False, random_state=None,
    shrinking=True,
    tol=0.001, verbose=False)
Model is ready..
Accuracy RBF SVC c=10000 gamma=0.0001 ..
0.571428571429

```

SVC:

Kernel : 'rbf'

C- 10000

gamma:1

Accuracy ~58%

```

.....
SVC(C=10000, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape=None, degree=3, gamma=1,
    kernel='rbf',
    max_iter=-1, probability=False, random_state=None,
    shrinking=True,
    tol=0.001, verbose=False)
Model is ready..
Accuracy RBF SVC c=10000 gamma=1 ..
0.578231292517

```

As you can see, I have tested these results against various combinations. When value of gamma is kept auto, then accuracy changes based on value on C. And once I started changing both C and gamma together, realized that it's a combination of two that matters. Based on combinations, the accuracy varied between 56-58%

