INFO

Libraries Used:

The external libraries used in Assignment 3 include scikit-learn,nltk. For each of the method used in the code, comments give a detailed description of what the method does.

The SVM classifier has 14 parameters. They are as follows:

1) **C** – It is the penalty parameter which is used for regularization. The default value is 1. It controls the effect of individual SVM's.

2) **Kernal** – Specifies the kernel to be used. Default kernel is 'RBF'

3) **Degree** – Used only with 'poly' kerna;. Default value is 3.

4) **Gamma** – Specifies kernel coefficient for 'rbf', 'poly' and 'sigmoid' kernels. Default value is 'auto'.

5) **Coef0** – It has significance only in 'poly' and 'sigmoid' kernals. It gives value for independent term in kernel. Default value is 0.

6) **Probability:** Tells the classifier to enable or disable probability estimates. Default value is 'false.

7) **Shrinking** – Specifies if the classifier should use shrinking heuristic. Default value is 'true'.

8) **Tol -** Gives value of tolerance for stopping criterion. Default value is 1e-3.

9) **Cache_size -** Specifies the size if the kernel cache in MB.

10) **Class_weight -** Sets the parameter C of class i to class_weight[i]*C for SVC. The "balanced" mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as n_samples / (n_classes * np.bincount(y)).

11) **Verbose -** Tells the classifier whether to enable verbose output or not**.** Default value is 'false'.

12) **Max_iter -** Sets the limit on iterations within solver. The default value is -1.

13) **Decision_function_shape -** Tells the classifier whether to return a one-vs- rest ('ovr') decision function of shape (n_samples, n_classes) as all other classifiers, or the original one vs-one ('ovo') decision function of libsvm which has shape (n_samples, n_classes * (n_classes - 1) / 2). The default of None will currently behave as 'ovo' for backward compatibility and raise a deprecation warning, but will change 'ovr' in 0.19.

14) **Random_state -** The seed of the pseudo random number generator to use when shuffling the data for probability estimation. The default value is none.

The analysis of different values of the above parameters is given below.

Digit Recognition Dataset

C:

For C = 1 to 10 the accuracy we got when different kernals were used did not change. In this case the value of C did not have any effect on the accuracy.

Kernal:

80-20 split:

- Linear – 97.3
- Poly – 98.69
- Sigmoid – 10.7
- RBF – 57

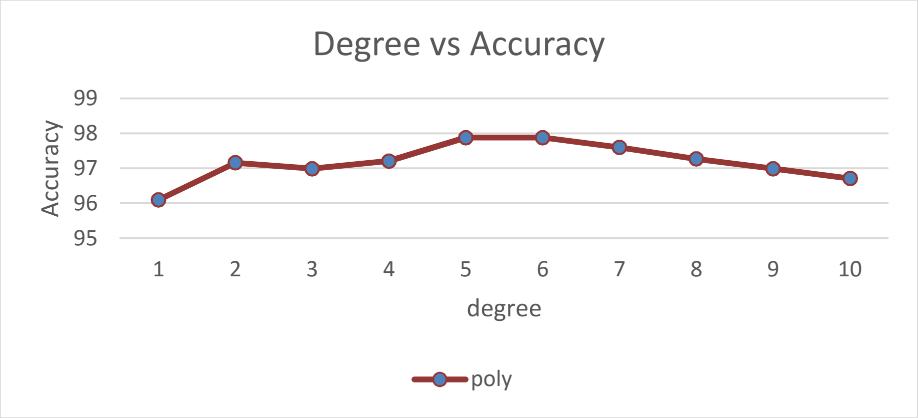
60-40 split:

- Linear – 96.87
- Poly – 98.21
- Sigmoid – 10.7
- RBF – 54

The accuracy for poly kernel is the highest. So, we finalized on setting 'poly' as the kernel to be used by the algorithm.

Degree:

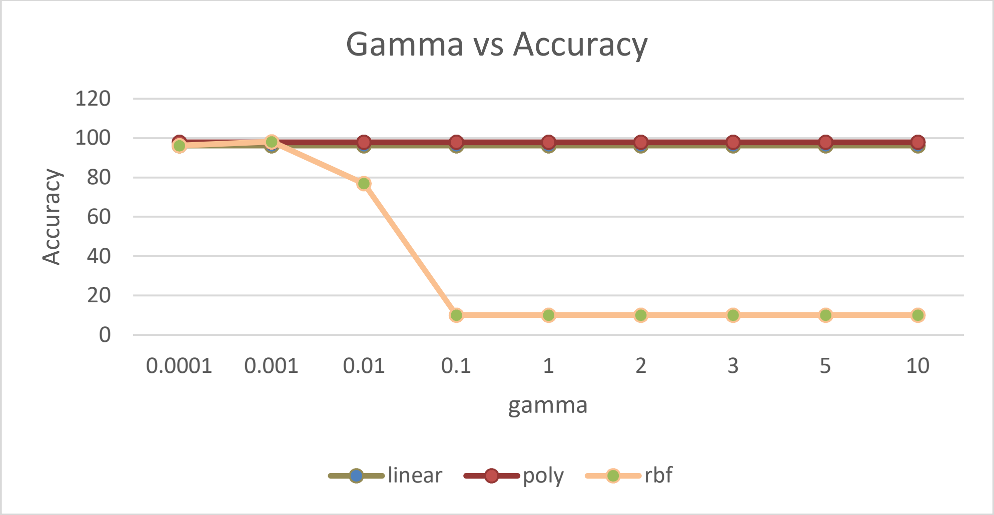
We experimented with degree values from 1 to 10 for the poly kernel.



Got maximum accuracy for degree = 5 and degree = 6.

Gamma:

The accuracy obtained for different values of gamma with different kernals is shown below.



The accuracy didn't change for linear and poly kernals. For RBF kernel the accuracy dropped drastically.

Coef0: (coef = 0 to 100)

- **Poly kernel – 98.69**
- **Sigmoid kernel – 10.7**

For Coef0 = 1 to 100 the accuracy we got when different kernals were used did not change. In this case the value of Coef0 did not have any effect on the accuracy.

Probability:

Default value is false. When the value is set to true the accuracy remained same. So, we used the default value.

Shrinking:

Default value is true. When the value is set to false the accuracy remained same. So, we used the default value.

Analysis on Amazon dataset:

C: For C = 1 to 10 the accuracy we got when different kernals were used did not change. In this case the value of C did not have any effect on the accuracy.

Kernal:

80-20 split:

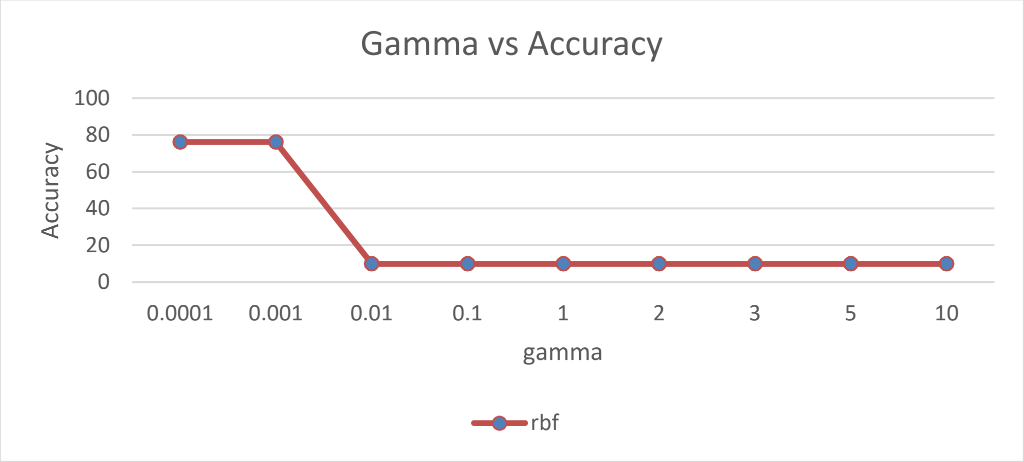
- Linear – 72
- Poly – 78
- Sigmoid – 78
- RBF – 78

60-40 split:

- Linear – 71
- Poly – 78
- Sigmoid – 76.12
- RBF – 77.67

The accuracy remained almost when poly, sigmoid, rbf kernals are used. We decided to use rbf kernel.

Gamma:



As the accuracy didn't increase beyond the default value of gamma we used the default value.

Easy Parts:

The easy part of this assignment is implementation of Digit Recognition. It took less time and analysis was quicker. So, it became easy for taking the decision of parameters in both the algorithms.

Challenging part of the Assignment:

The Amazon Baby Review dataset was challenging. Understanding what was given and what was asked from the dataset was the most difficult part. Once we got to know what to do, we tried different methods to predict the ratings from reviews. The naïve method was to searchfor words with positive and negative meanings and predict the rating. Then we used

Natural Language Processing Toolkit to analyse the data and predict the rating from the reviews.

Results:

The SVM classifier gave an accuracy of 98.69% on Digit Recognition dataset and an accuracy of 76.42% on Amazon dataset. A detailed report is given in the SVM_Results.pdf file.