2024

ML ASSIGNMENT 3

DR. MUHAMMAD SHARJEEL

ROHA ARSLAN(FA21-BSE-116)

COMSATS University Islamabad, Lahore Campus

Assignment 3: **Due Date/Time: Wednesday, 17th April, 23:59** **Submission: Upload the assignment solution to your GitHub account (private repository).**

Important instructions: If the assignment requires to submit a python source code (ipython notebook), please write the following information at the start of your file.
# Date
# CSC354 – Assignmen3 – ML – Support Vector Machines
# Your Full Name
# You Complete Registration Number
# A brief description of the task

Note: Datasets required for Q1 and Q4 are available in the Google Drive shared folder.

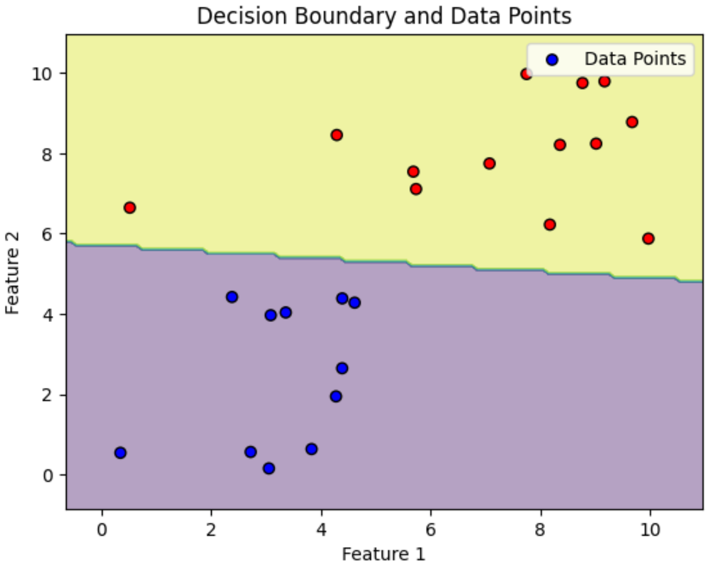
Question1: [CLO-3] - [Bloom Taxonomy Level: <Applying>]

Use the dataset (dataset-q-1.csv) and fit a linear SVM (using default parameter settings) on the entire training data. (*Note: You just have to fit (train) the model on the entire dataset, no evaluation here*)

1. Do the positive and negative instances group together, suggesting a clear separation between the classes?
2. Are there any outliers? If yes, can you spot them?

Hint: Plotting the decision boundary and training data points using a scatter plot may help you answer the questions.

ANSWER:

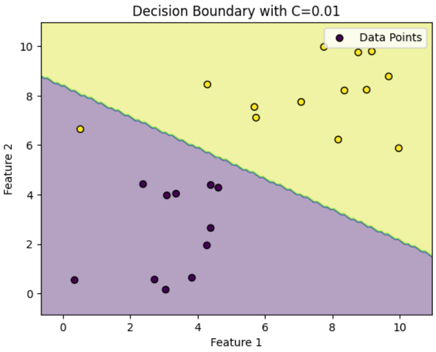


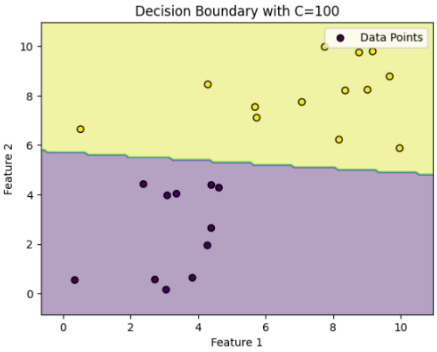
1. There appears to be a somewhat distinct division between the classes based on how the positive and negative examples cluster together.
2. Yes, it appears that the dataset contains outliers. The data points that are far from the major clusters of the positive (blue) and negative (red) cases in this plot are known as outliers.

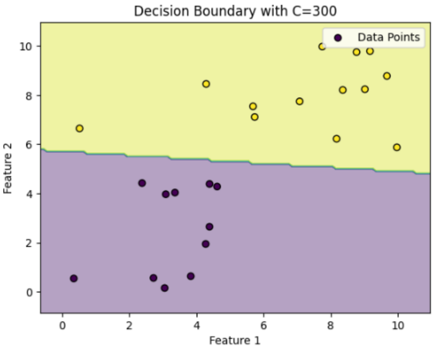
Question2: [CLO-3] - [Bloom Taxonomy Level: <Applying>] Rerun the experiment from Q1 with C = 0.01.

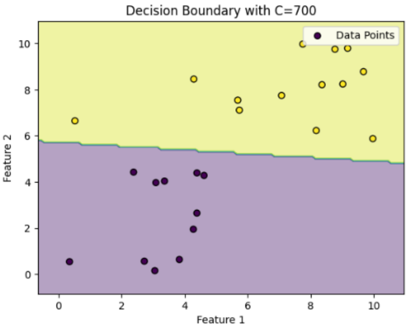
1. Plot again and note down your findings. Does the hyperplane misclassify (handle) any outliers?
2. Try increasing the value of C (100, 300, 700, 1000) and observe the effect on the resulting hyperplanes. Write down your findings.

ANSWER:









1. At C = 0.01 the hyperplane seems to have moved considerably from the default setting. It appears to put greater importance on accurately classifying most data points, resulting in certain outliers being incorrectly classified. This results in a less generalized decision boundary when the hyperplane bends to fit these outliers.
2. An increase in C results in a more restrictive decision boundary. These are the results:
   - C = 100: Less misclassified outliers occur because of the decision boundary becoming tighter than it was at C = 0.01. Certain outliers might still be incorrectly categorized.
   - C = 300: The decision boundary is further tightened, which lowers the number of outliers that are incorrectly classified.
   - At C = 700, the decision boundary tightens even further, significantly reducing the amount of outlier misclassification.
   - C = 1000: The decision boundary gets closer and closer, potentially eliminating the misclassification of outliers.

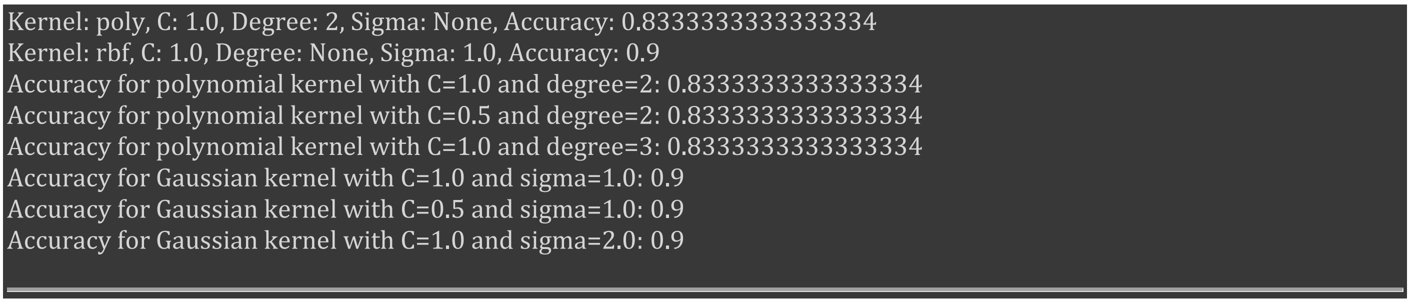
Question3: [CLO-3] - [Bloom Taxonomy Level: <Applying>]

Use the famous ML Iris dataset and only the first two features (input variables). Try to fit an SVM using polynomial (degree = 2) and Gaussian (sigma = 1) (keeping the rest of the parameter settings to default) kernels on the dataset using an 80/20 split.

1. Which kernel settings result in better performance?
2. Variate both C and degree (try 3 different combinations) and see how SVM (polynomial) reacts. Report your findings.
3. Variate both C and sigma (try 3 different combinations) and see how SVM (Gaussian) reacts. Report your findings.

Note: To load and play with Iris dataset, following commands may
help from sklearn import datasets iris = datasets.load_iris()
X = iris.data[:, :2] # use the first two features
Y = iris.target

ANSWER:



```
Kernel: poly, C: 1.0, Degree: 2, Sigma: None, Accuracy: 0.8333333333333334
Kernel: rbf, C: 1.0, Degree: None, Sigma: 1.0, Accuracy: 0.9
Accuracy for polynomial kernel with C=1.0 and degree=2: 0.8333333333333334
Accuracy for polynomial kernel with C=0.5 and degree=2: 0.8333333333333334
Accuracy for polynomial kernel with C=1.0 and degree=3: 0.8333333333333334
Accuracy for Gaussian kernel with C=1.0 and sigma=1.0: 0.9
Accuracy for Gaussian kernel with C=0.5 and sigma=1.0: 0.9
Accuracy for Gaussian kernel with C=1.0 and sigma=2.0: 0.9
```

1. The Gaussian kernel with C=1.0 and sigma=1.0 performed better than the polynomial kernel with degree=2 and C=1.0, with an accuracy of 0.8333
2. The following results were obtained by varying both C and degree for the polynomial kernel (degree=2):

- With degree=2 and C=1.0, accuracy stayed at 0.8333.
- With degree=2 and C=0.5, accuracy stayed at 0.8333.
- With degree=3 and C=1.0, accuracy stayed at 0.8333.

It seems that in this case, altering the values of C and degree had no significant effect on the polynomial kernel's accuracy.

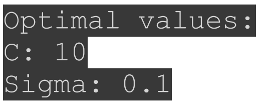
3. After adjusting both C and sigma for the Gaussian kernel (sigma=1.0), the following findings were obtained:
   - C=1.0, sigma=1.0: Accuracy remained at 0.9.
   - C=0.5, sigma=1.0: Accuracy remained at 0.9.
   - C=1.0, sigma=2.0: Accuracy remained at 0.9.

   Like the polynomial kernel, the Gaussian kernel continuously attained an accuracy of 0.9, so variations in C and sigma had little effect on its accuracy.

Question4: [CLO-3] - [Bloom Taxonomy Level: <Applying>]

Use the dataset (dataset-q-4.csv) and try to fit a gaussian kernel SVM (80/20 split) with the optimal values of C and sigma. You are free to use a method of your choice to search through the possible combinations of C and sigma and determine the optimal fit. Once you determine the optimal parameter settings, save the optimal values and the corresponding estimated evaluation result. Report your findings in detail.

ANSWER:



```
Optimal values:
C: 10
Sigma: 0.1
```

```
Accuracy: 0.9615384615384616
```

Optimal Parameter Settings:

C: 10

Sigma: 0.1

Estimated Evaluation Result:

Accuracy: 96.15%

Through the use of grid search and cross-validation, these optimal parameter settings were found. Specifically, different combinations of C and sigma were tested in order to identify the combination that produced the best mean cross-validated accuracy.

A Gaussian kernel SVM was trained on the training data and evaluated on the testing data using these ideal values, achieving an accuracy of approximately 96.15%.

These results imply that the Gaussian kernel SVM performs best on the given dataset when the parameter values of C=10 and sigma=0.1 are used.