Classifying data using Support Vector Machines(SVMs) in Python

Course Instructor : Dr. Khaled Mostafa El Sayed
TAs : Eng. Marwa Monier & Eng. Gamal Zayed

Nada Ismail, 202-001-387

15th Dec, 2023

A Problem definition and motivation

The task at hand is applying Support Vector Machines with different kernels and regularization to classify the six datasets. SVM is a powerful machine learning algorithm used for both classification and regression. In classification, it works by finding the hyperplane that best separates different classes in the feature space. The problem is to explore the performance of SVM with various configurations and assess its ability to accurately classify instances in the Aggregation dataset. Different kernels (linear, polynomial, RBF) have different ways in handling different types of data. By applying SVM with various kernels, we aim to understand how each kernel copes with the unique features of the each dataset. This information is crucial for choosing the most appropriate kernel for datasets with similar complexities.

B Data-sets

B.1 Aggregation

Description: The Aggregation dataset is designed to contain clusters with varying shapes and sizes, and the clusters are densely packed.

Use Case: Useful for testing clustering algorithms in scenarios where clusters are densely populated and have irregular shapes.

B.2 Compound

Description: The Compound dataset often contains data points arranged in several connected components, making it challenging for clustering algorithms to correctly identify the underlying structures.

Use Case:to assess the performance of clustering algorithms in scenarios where there are multiple interconnected clusters.

B.3 Flame

Description: The Flame dataset typically consists of two groups of data points resembling flame shapes. It is designed to test the performance of clustering algorithms, particularly when dealing with clusters of different sizes and shapes.

Use Case: Used in clustering algorithm evaluations, especially for algorithms dealing with clusters of varying densities and shapes.

B.4 Jian

Description: The Jain dataset contains two C shaped clusters. It's a good dataset for testing the ability of clustering algorithms to identify clusters that need to be seprated with bendy lines.

Use Case: Commonly used to evaluate the performance of clustering algorithms with irregularly shaped clusters.

B.5 Pathbased

Description: The Pathbased dataset usually contains paths of varying shapes and orientations. It is a challenging dataset for classification algorithms due to the intricate patterns of the paths.

Use Case: Useful for assessing the ability of classifiers to handle datasets with complex structures and patterns.

B.6 Spiral

Description: This dataset consists of three intertwined spirals. It is a popular example for testing the capabilities of classification algorithms that need to capture non-linear decision boundaries.

Use Case: To evaluate the performance of classifiers in handling complex, non-linear patterns.

C Approach and methodology

C.1 Data Pre-processing

C.1.1 Loading Data:

Load the dataset from the .txt file. and dropping the header

C.1.2 Visualization:

Visualize the dataset to understand its distribution and characteristics.

C.1.3 Splitting Data:

Split the dataset into training and testing sets. A common split ratio is 80% for training and 20% for testing.

C.2 Model Parameters

C.2.1 SVM with Different Kernels:

three different kernels are used: linear, polynomial, and radial basis function (RBF).

C.2.2 Regularization:

Implement SVM models both with and without regularization.Try different values for C (e.g., 0 for no regularization, 1 for regularization).

C.3 Model Training

Train SVM models for each combination of kernel and regularization using the training set.

C.4 Model Evaluation

C.4.1 Prediction:

Make predictions on the testing set using the trained models.

C.4.2 Metrics:

Evaluate the performance of each model using metrics such as accuracy, precision, recall, and F1-score.

C.4.3 Visualization:

Visualize each SVM model on the test set to understand how well the model captures the underlying patterns.

C.4.4 Comparison:

Compare the performance of models with different kernels and regularization settings.

D Implementation

D.1 Tools and Libraries

scikit-learn for SVM implementation, NumPy for numerical operations, and Matplotlib for data visualization.

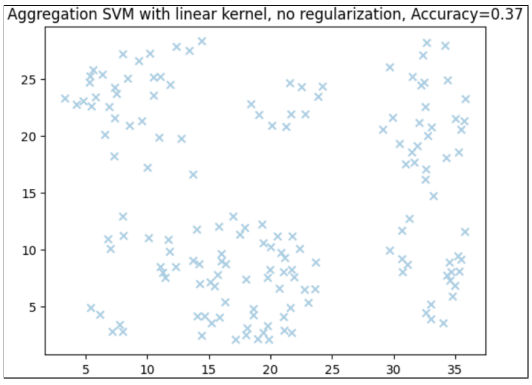
D.2 Hyperparameter Tuning

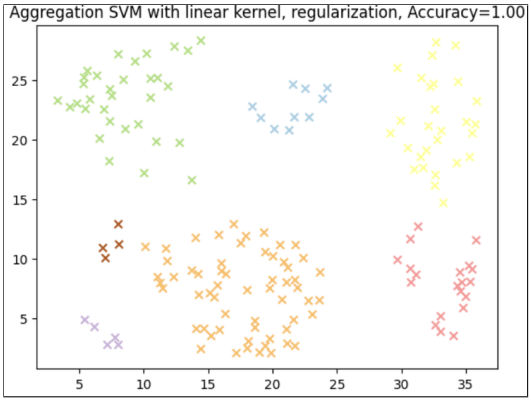
Tune hyperparameters such as the choice of kernel and regularization parameter.

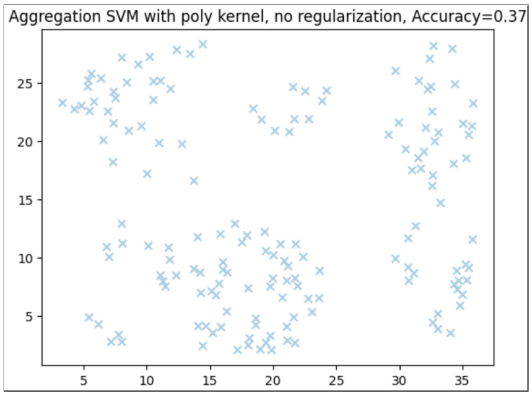
E Conclusion

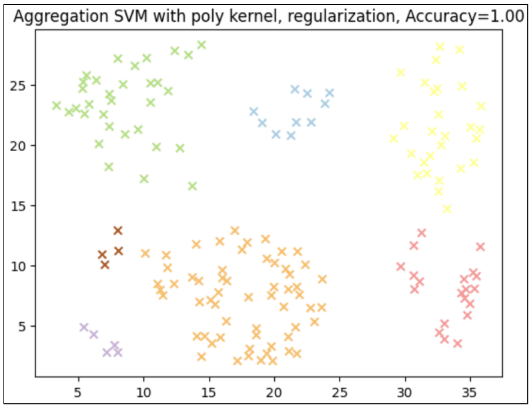
E.1 Aggregation

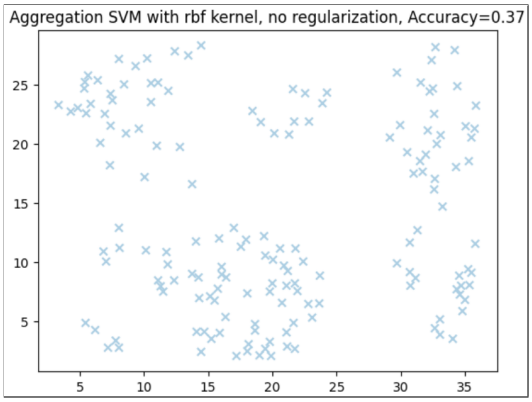
The accuracy doesn't vary according to the type of kernel. The only thing that matter is the regularization. If regularized the accuracy is 100% else it is 36%.

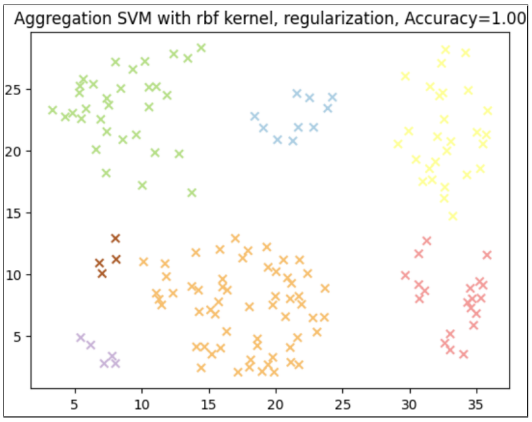






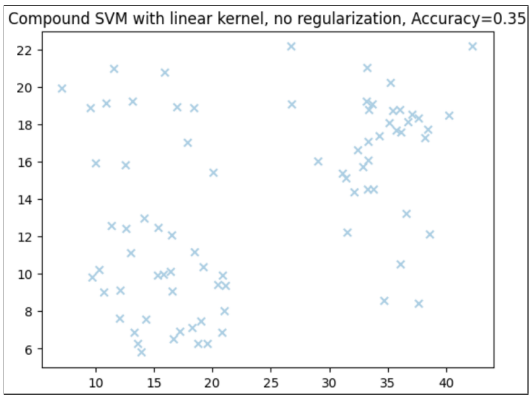


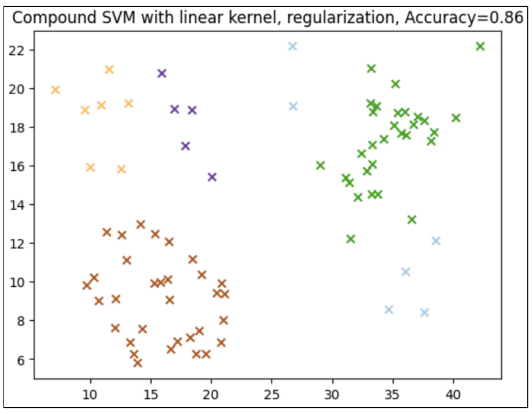


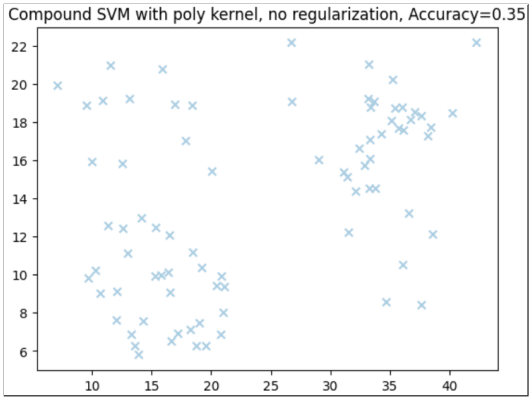


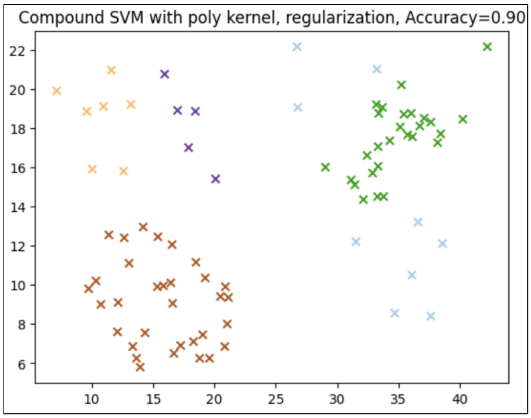
E.2 Compound

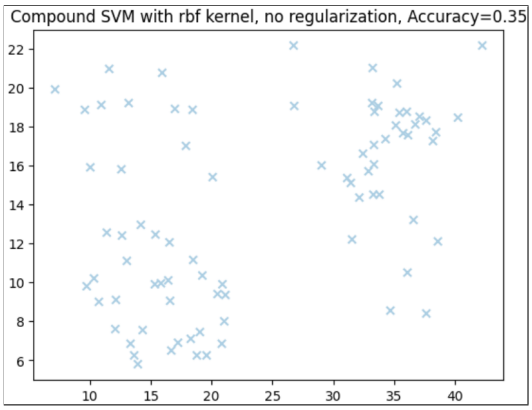
Much like the aggregation dataset the biggest difference comes from regularization. The linear kernel gives the worst results

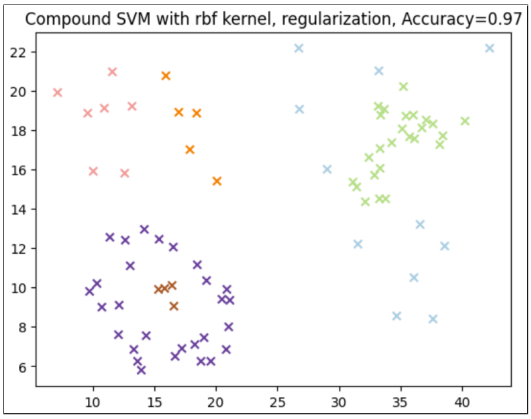






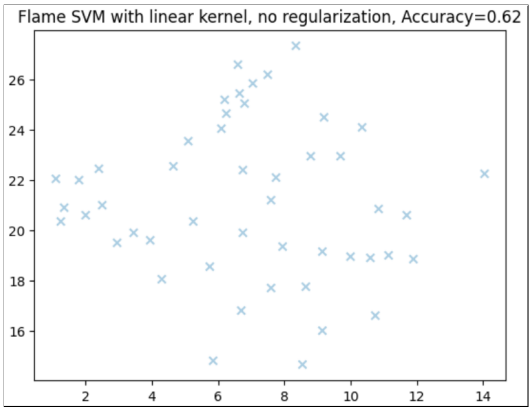


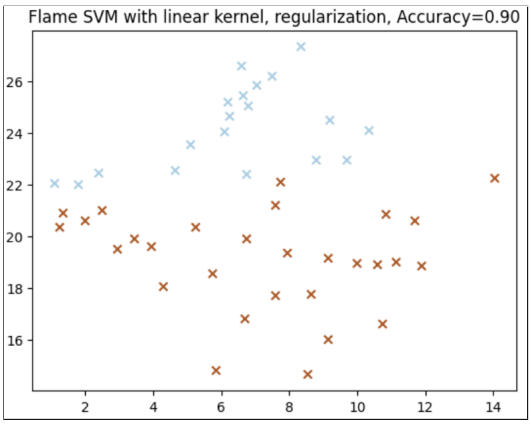


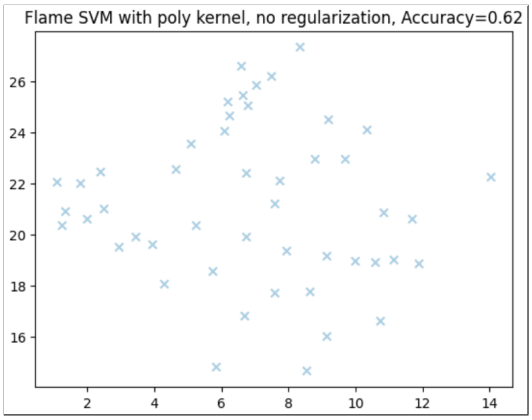


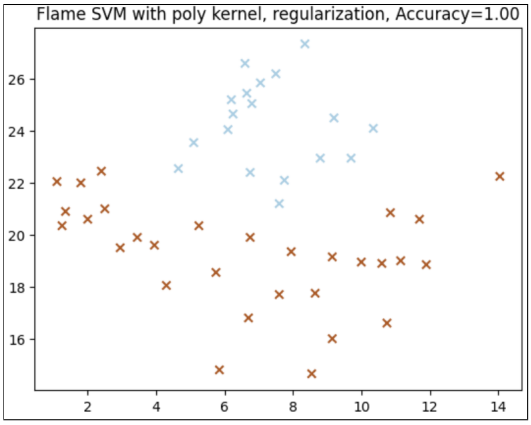
E.3 Flame

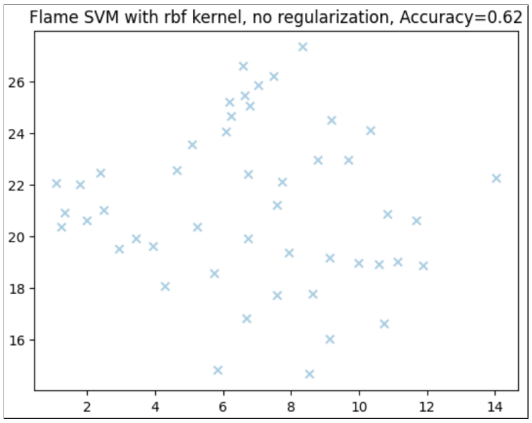
Much like the compoundnd dataset the biggest difference comes from regularization and the linear kernel gives the worst results

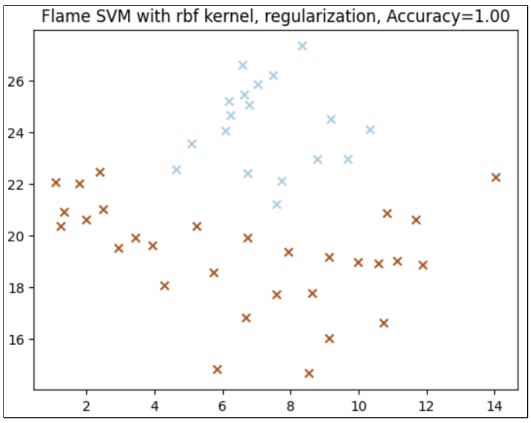






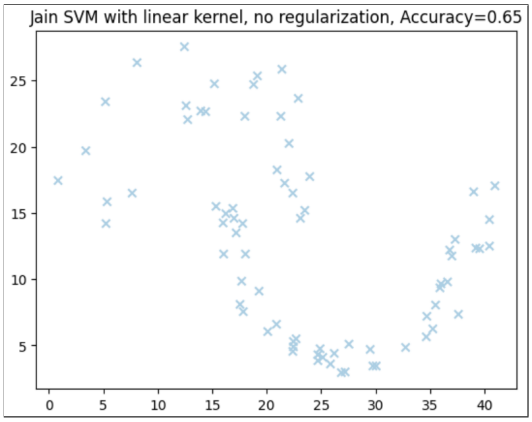


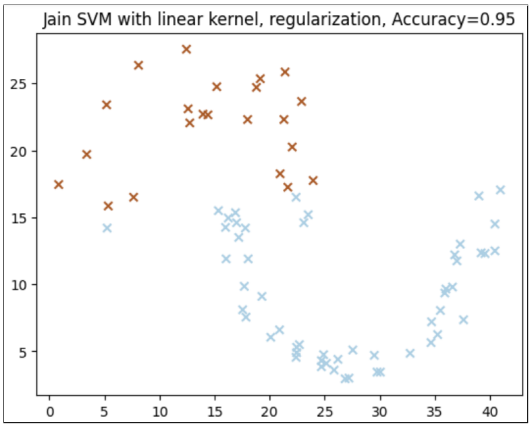


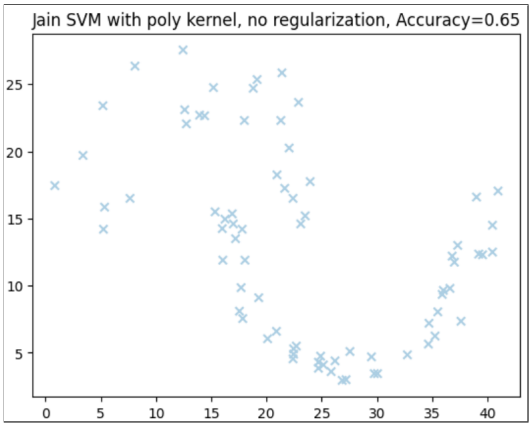


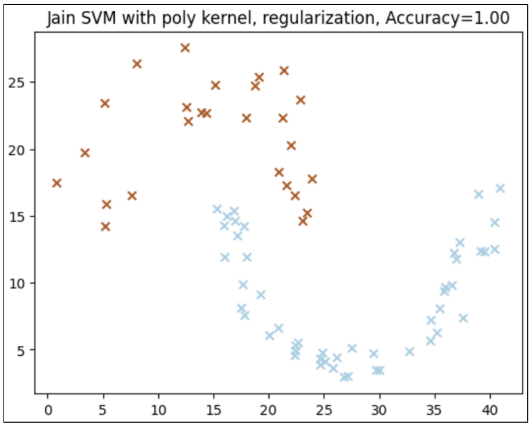
E.4 Jian

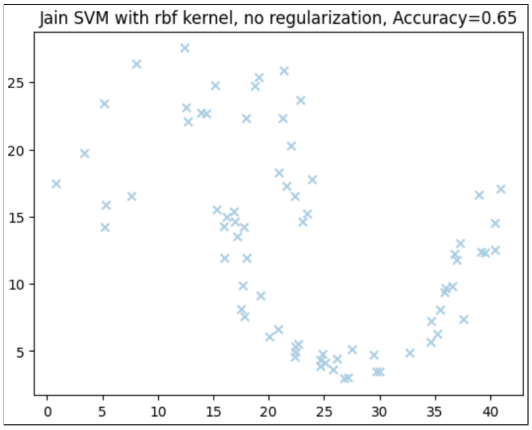
Much like the flame dataset the biggest difference comes from regularization. The linear kernel gives the worst results the reason it is not as fairing is because the two curves could be slightly separated by a line.

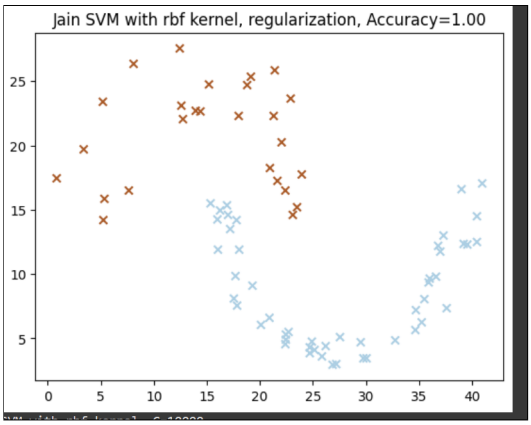






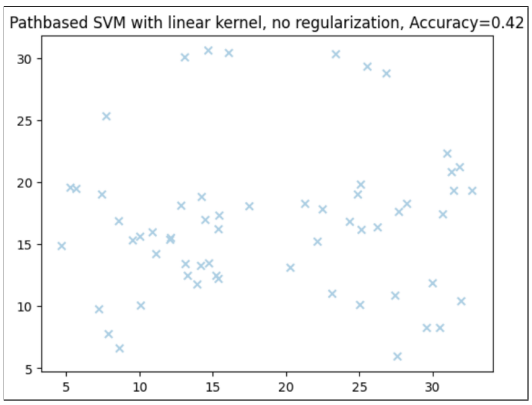


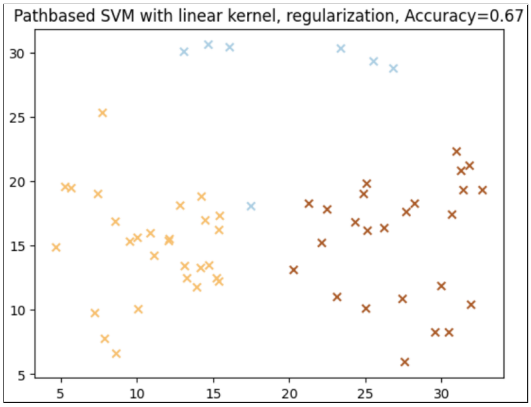


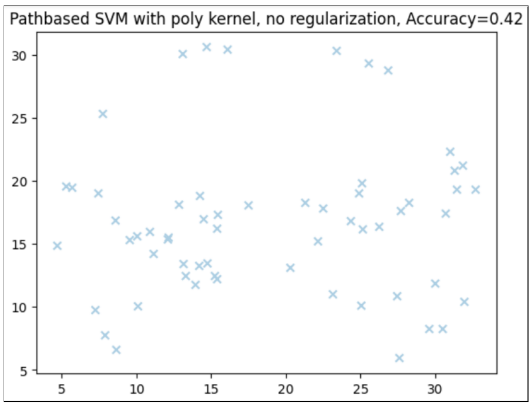


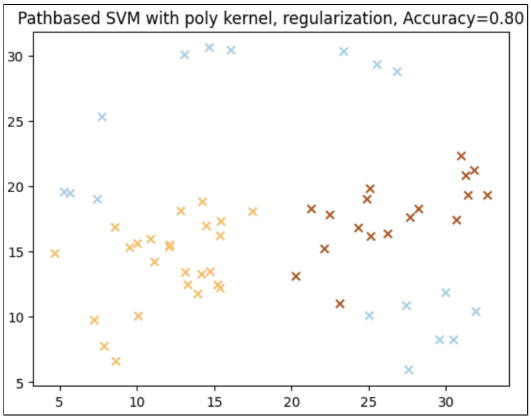
E.5 Pathbased

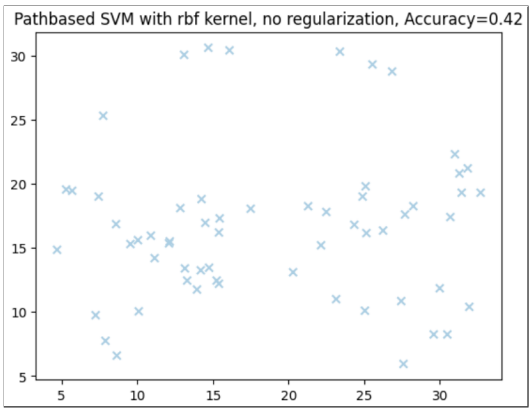
The accuracy is better with regularization. The linear kernel did very badly which is expected. Just looking at the dataset it is going to be hard to separated using lines.

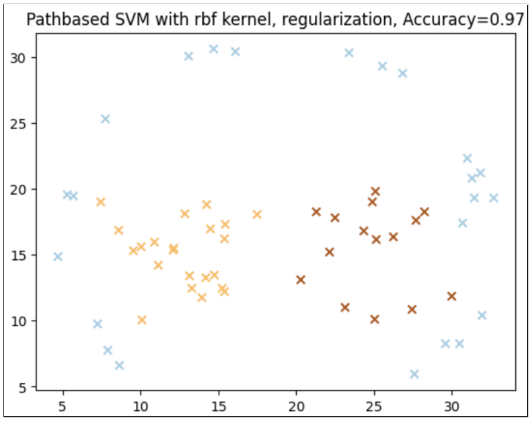






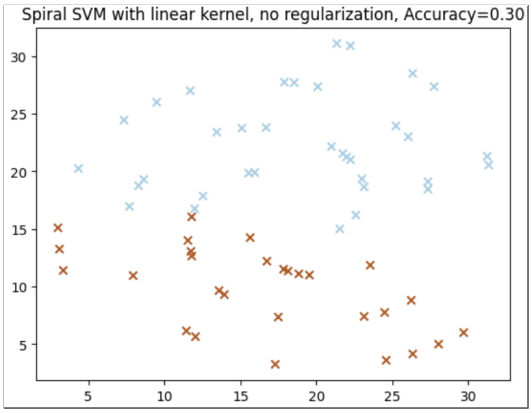


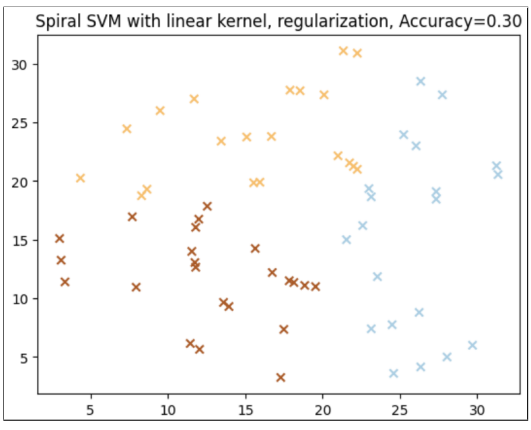


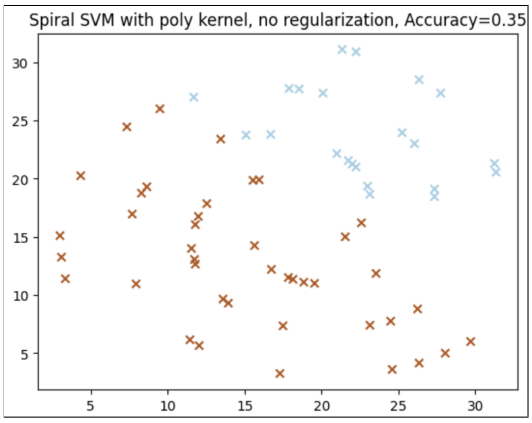


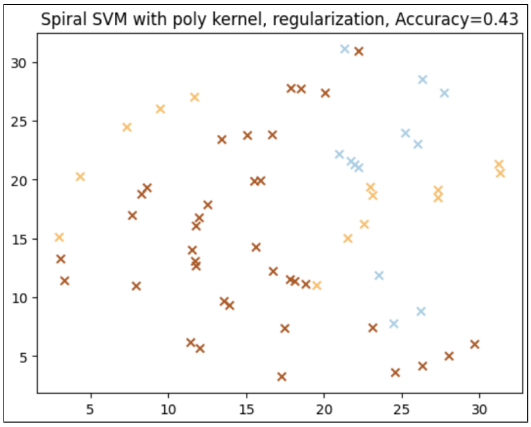
E.6 Spiral

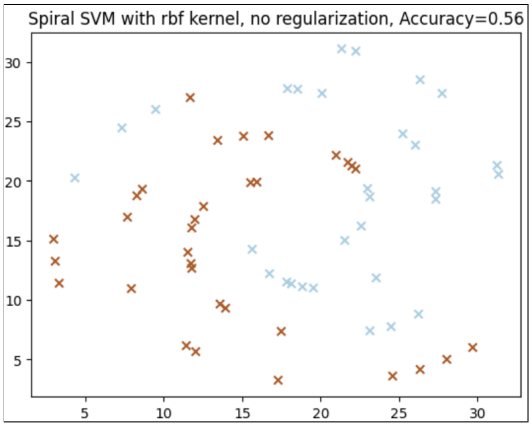
The accuracy is better or the same with regularization. The all kernels did very badly except the rbf.

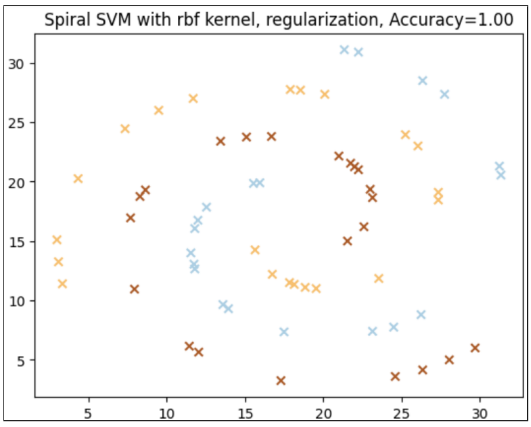





Spiral SVM with poly kernel, no regularization, Accuracy=0.35


Spiral SVM with poly kernel, regularization, Accuracy=0.43


Spiral SVM with rbf kernel, no regularization, Accuracy=0.56


Spiral SVM with rbf kernel, regularization, Accuracy=1.00