Lab Assignment 11
Prakhar Gupta
B21AI027

Question 1:

Initials-

- Downloaded dataset using **wget** command called using **os.system**
- Loaded the **.csv** file into df using pd.read_csv and set *columns=['variance','skewness','curtosis','entropy','classes']*
- Printed **df**

Part 1-

- Checked for not filled rows using **df.isnull().sum()**
- Used **df.dtypes** to find data type of **df**
- Used **df.describe()** to get insights about the dataset
- Plotted correlation matrix
- Applied **MinMaxScalar()** to every column to normalise data
- Converted **df to X,y**
- Split dataset into train,test,val in ratio 70:20:10

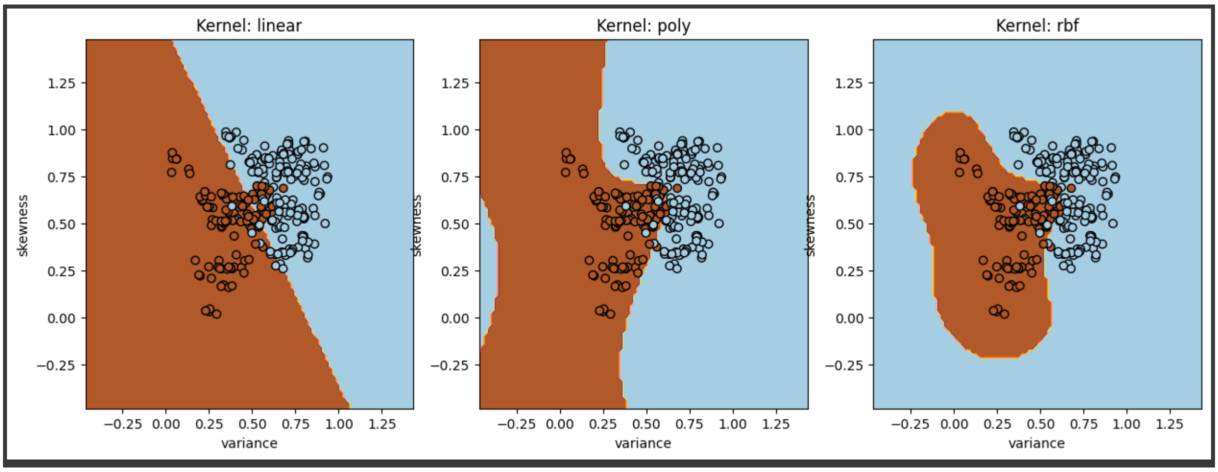
Part 2-

- Took c_vals = [0.0001, 0.01, 1, 100, 1000, 10000]
- Trained model on each C in c_vals
-
```
Classification accuracy for C = 0.0001: 0.5876
Classification accuracy for C =   0.01: 0.8978
Classification accuracy for C =      1: 1.0000
Classification accuracy for C =    100: 1.0000
Classification accuracy for C =   1000: 1.0000
Classification accuracy for C =  10000: 1.0000
```
- The accuracy increase with increase in C (the C specify how much can error is allowed low value of C means high error is allowed, this C can be used to remove overfit and underfit)
- **“Variance” and “Skewness”** feature has the highest correlation with y
- For different values of C plotted the decision boundary with data points
- Training again on these two features
```
Classification accuracy for C = 0.0001: 0.5554
Classification accuracy for C =   0.01: 0.7179
Classification accuracy for C =      1: 0.8812
Classification accuracy for C =    100: 0.8754
```

```
Classification accuracy for C =   1000: 0.8754
Classification accuracy for C =  10000: 0.8754
```

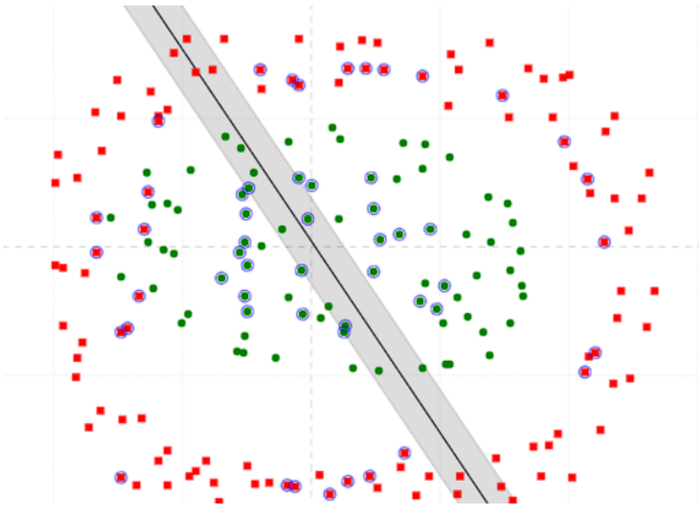
Part 3-

- Used **kernels = ['linear', 'poly', 'rbf']**
- Plotted decision boundary with dataset with different kernels considering the best feature **“Variance” and “Skewness”**
- ```
  Classification accuracy for linear kernel: 0.8759
  Classification accuracy for poly kernel: 0.9270
  Classification accuracy for rbf kernel: 0.9234
  ```
- RBF is giving the best , apart from that polynomial is also doing good

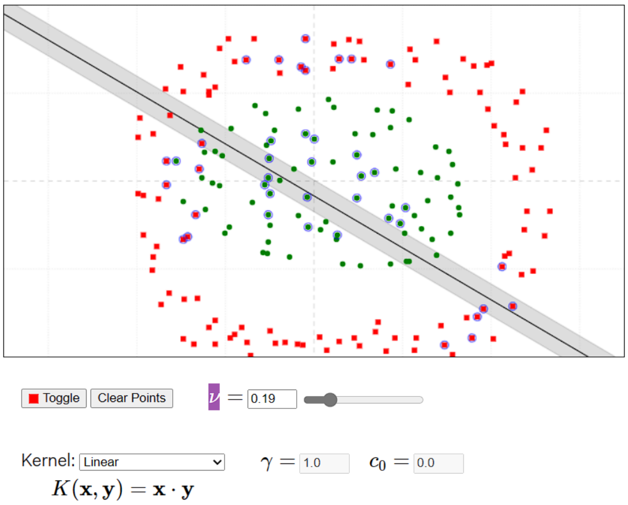


Part 4-

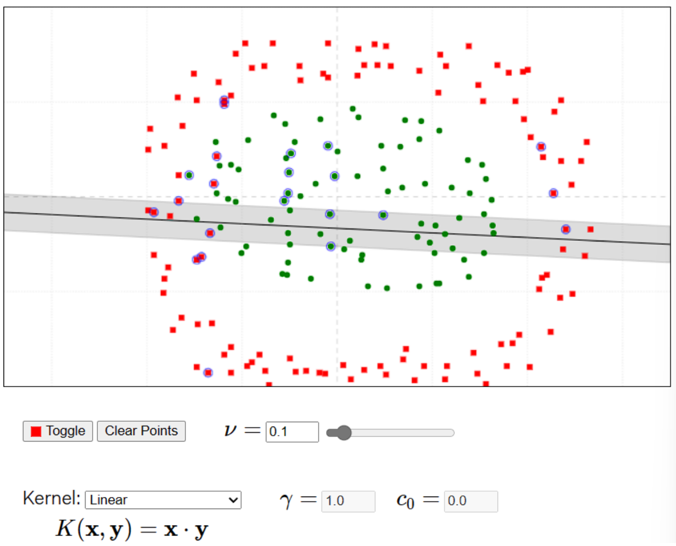
❖ **Subpart 1(First dataset)-**


  ```

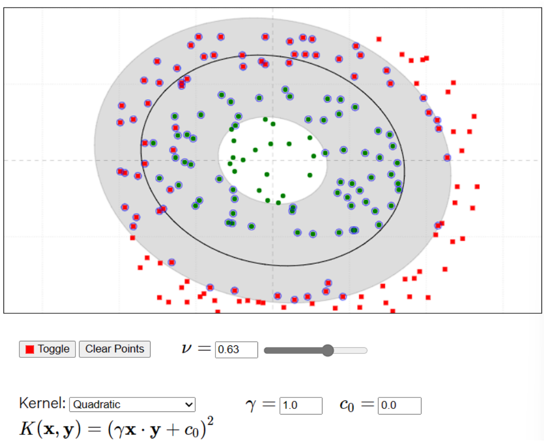
- ***v=0.15(Linear)***



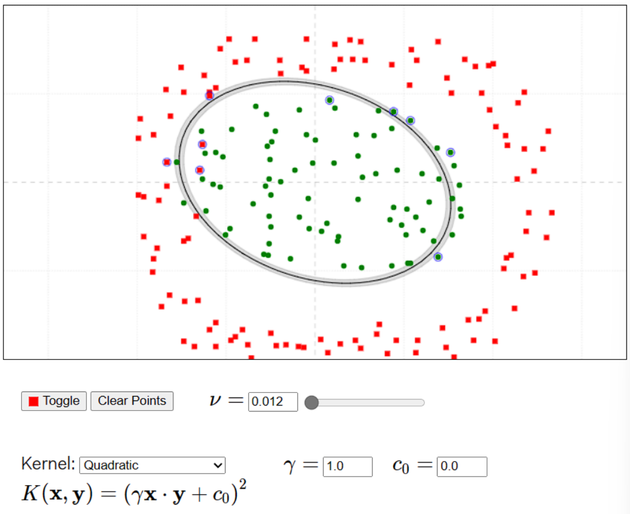
- ***v=0.1(Linear)***



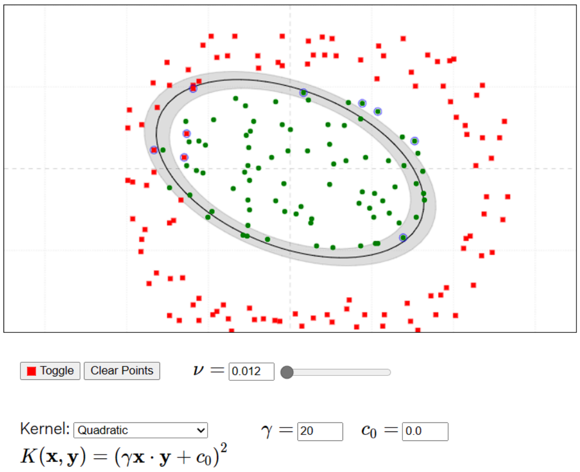
- ***v=0.63(Quadratic)***



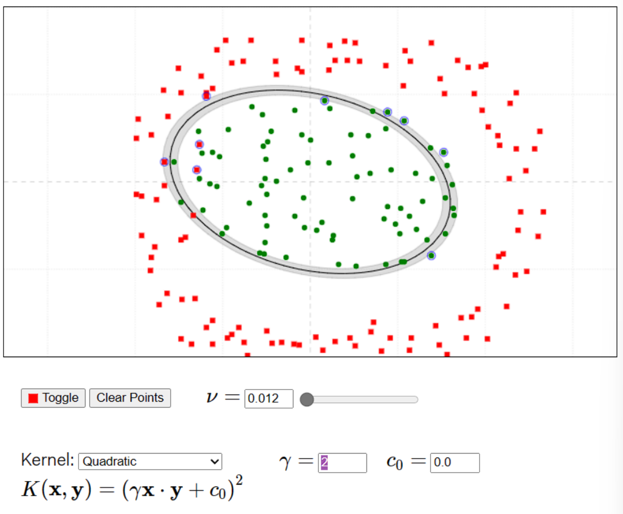
- ***v=0.012(Quadratic)***



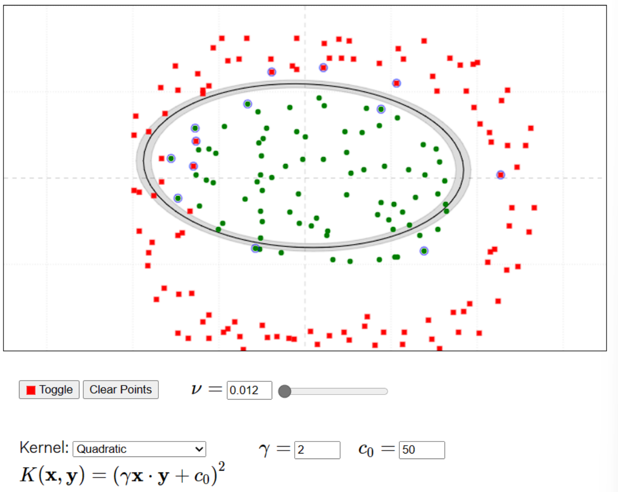
- ***γ=20(Quadratic)***



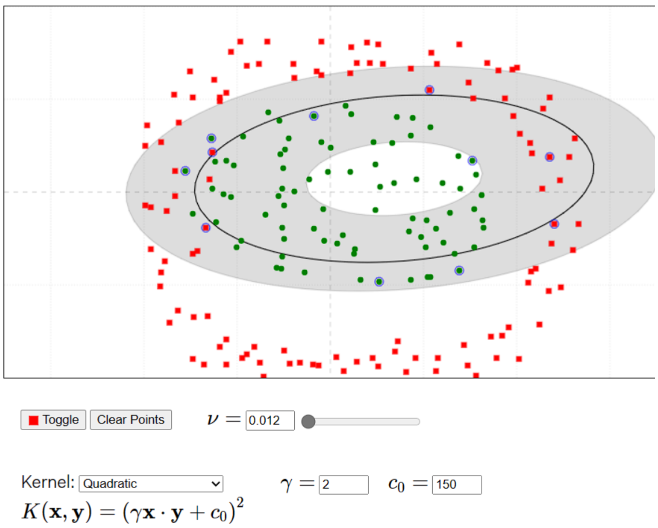
- ***γ=2(Quadratic)***



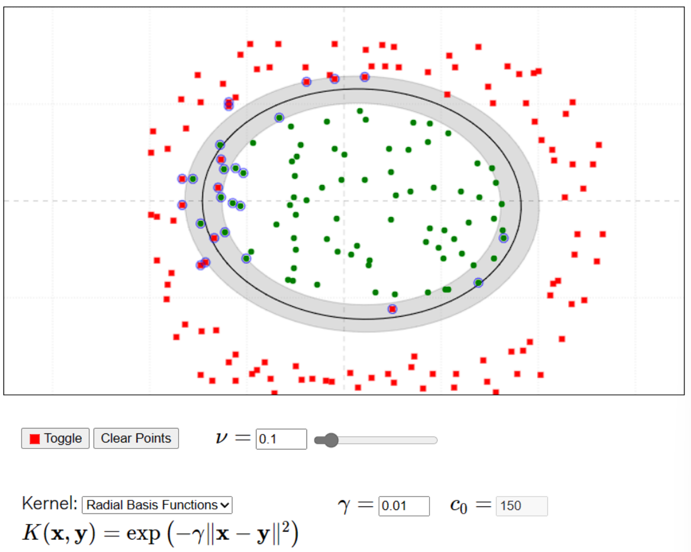
- ***c0=50(Quadratic)***



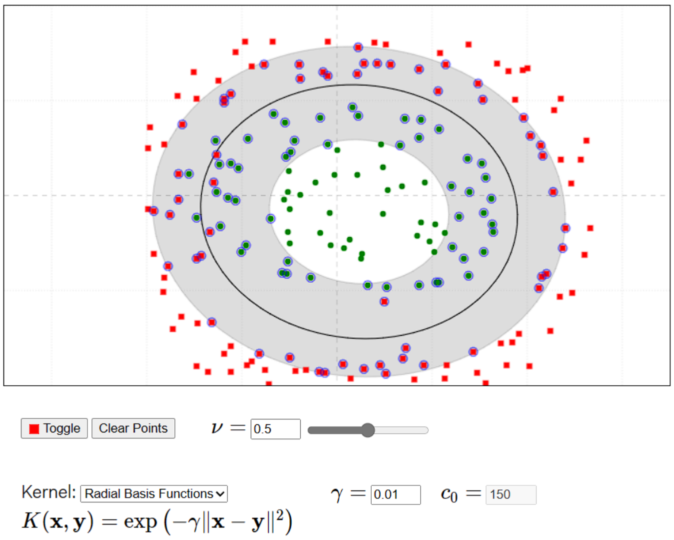
- ***c0=50(Quadratic)***



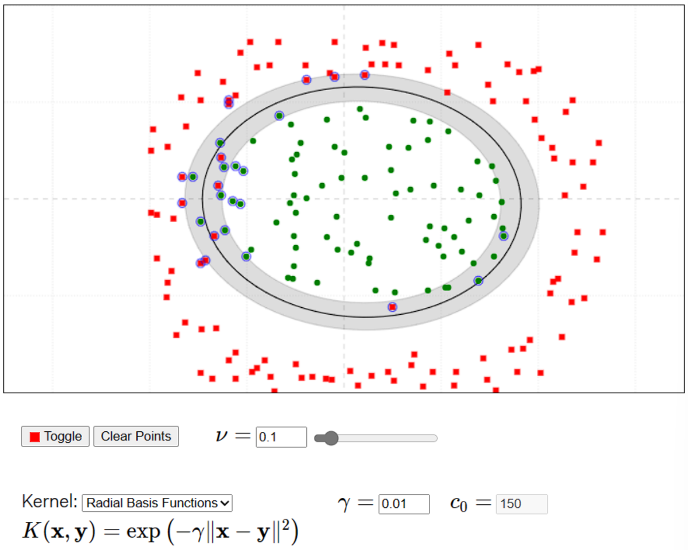
- ***v=0.1(Radial Basis Function)***



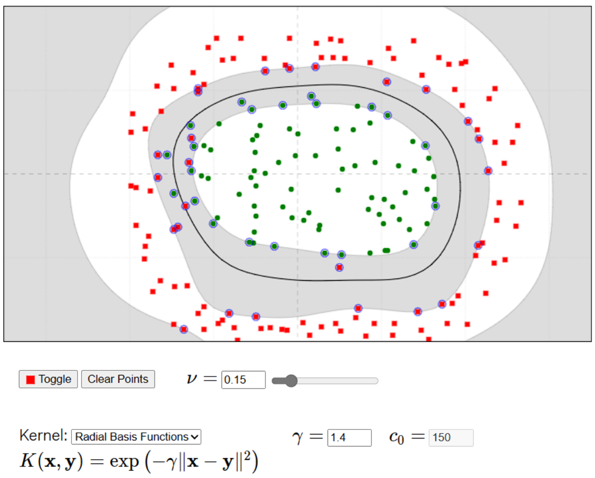
- ***v=0.5(Radial Basis Function)***



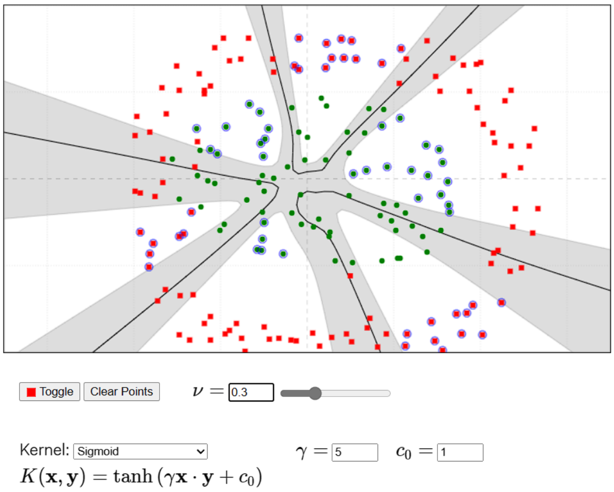
- ***γ=0.01(Radial Basis Function)***



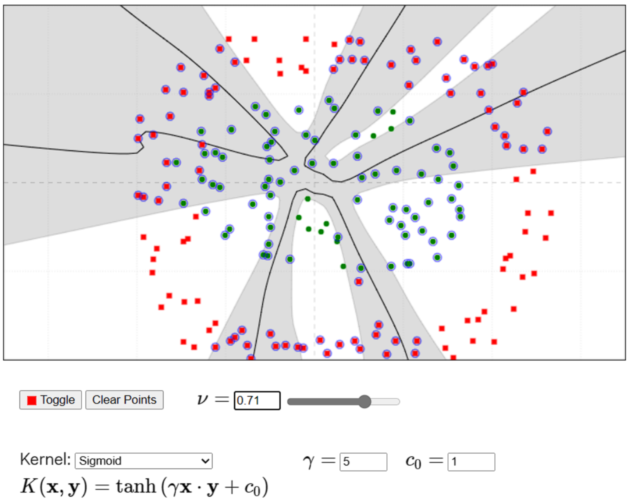
- ***γ=0.15(Radial Basis Function)***



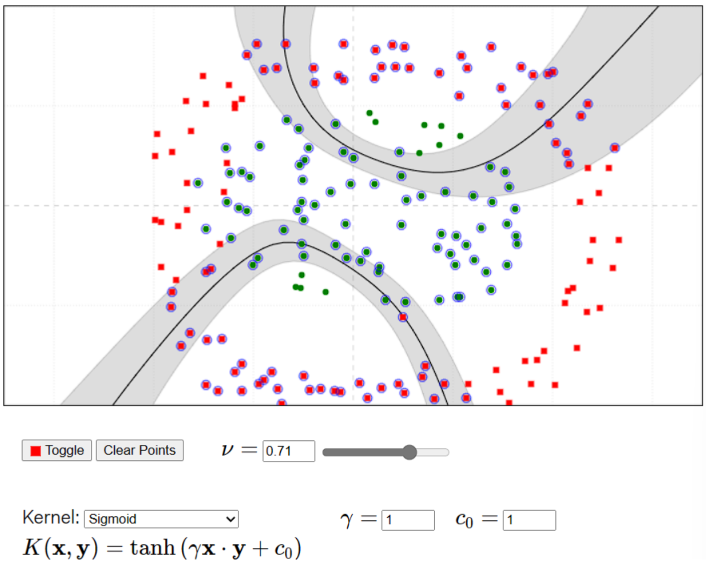
- ***v=0.3(Sigmoid)***



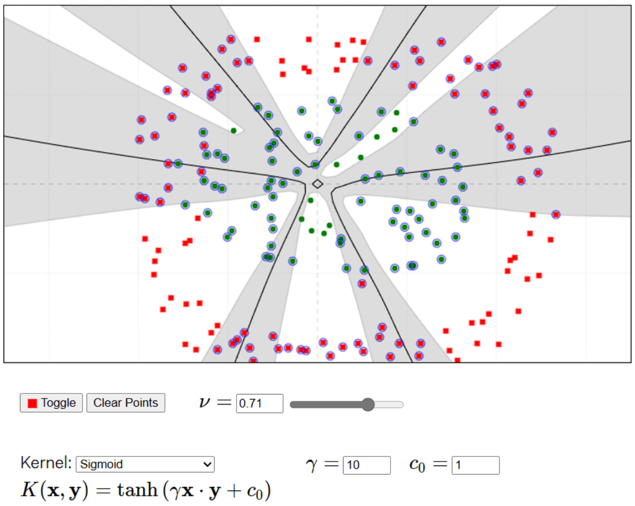
- ***v=0.71(Sigmoid)***



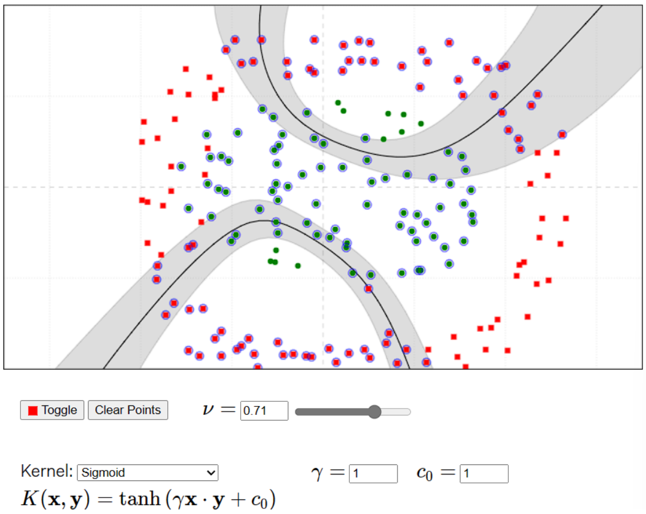
- ***γ=1(Sigmoid)***



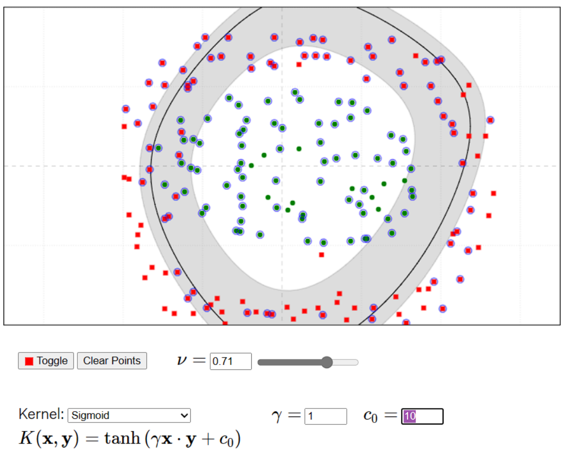
- ***γ=10(Sigmoid)***



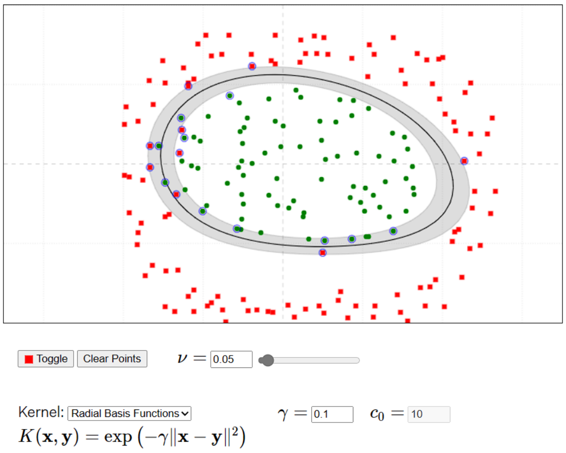
- ***C0=1(Sigmoid)***



- ***C0=10(Sigmoid)***



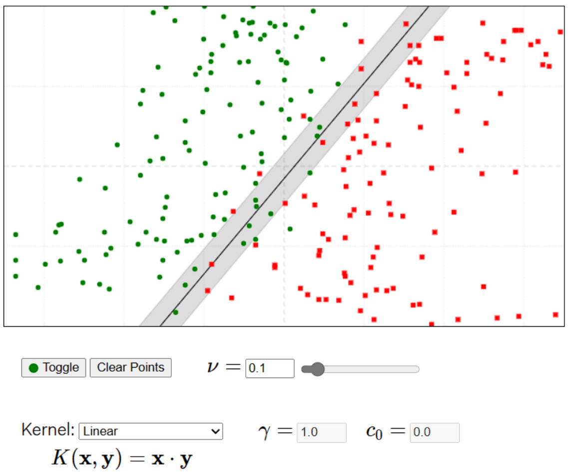
##Best Decision Boundary



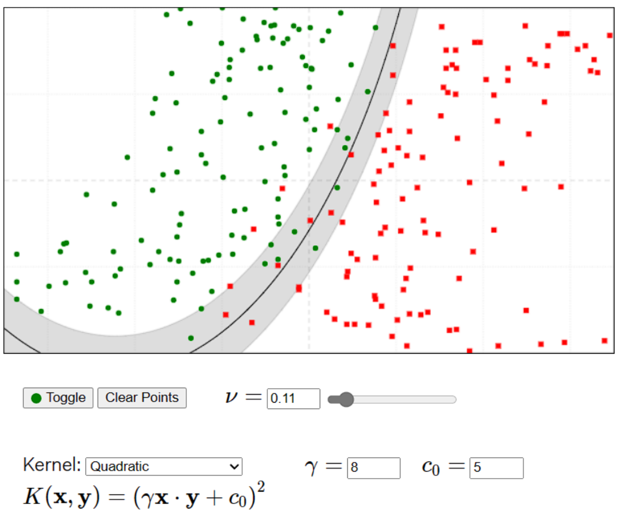
Radial Basis Function(v=0.05,γ=0.1) is performing good after hyperparameter tuning

❖ Subpart 2(Second dataset)-

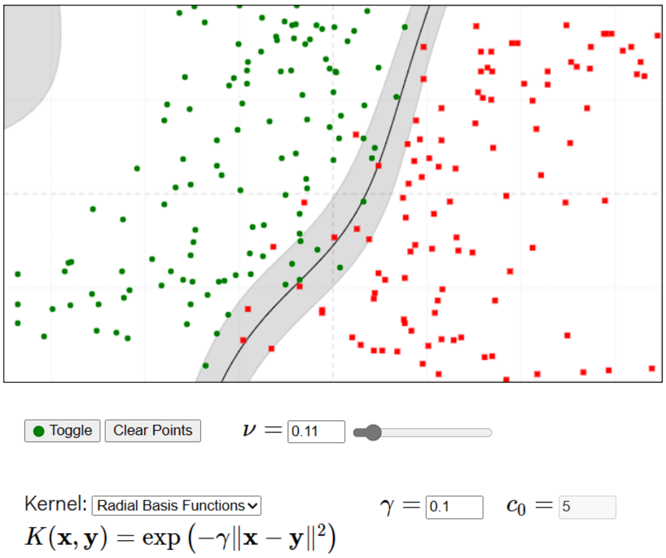
- Best Decision boundary (Linear)



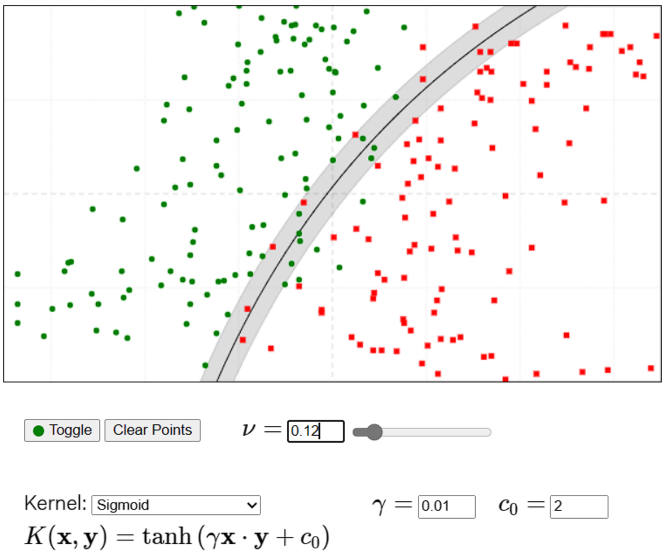
- Best Decision boundary (Quadratic)



- Best Decision boundary (Radial Basis Function)



- Best Decision boundary (Sigmoid)



Quadratic, Radial Basis Function and Sigmoid all are performing good after hyperparameter tuning

When using SVM to classify datasets, the choice of kernel and hyperparameters can significantly impact the algorithm's performance. I created two types of datasets, one being linearly separable, and the other not.

For linearly separable data, the linear kernel is a suitable choice as it is computationally efficient and can handle a large number of features. However, for non-linear data, other kernels such as RBF, sigmoid, or quadratic are more appropriate.

The RBF kernel is a popular choice for handling non-linear data. It utilises the gamma parameter, which controls the shape of the decision boundary. A smaller gamma value results in a smoother boundary, while a larger gamma value produces a more complex boundary that may lead to overfitting.

The sigmoid and quadratic kernels are other options for non-linear classification, but they may not perform as well as the RBF kernel. The sigmoid kernel utilises the mew(ν) parameter, which controls whether it is a hard margin or soft margin classification. A smaller mew produces a flatter slope, while a larger mew results in a steeper slope. On the other hand, the quadratic kernel is computationally more expensive than the linear or sigmoid kernels as it requires pairwise feature products.

The C0 hyperparameter controls the tradeoff between the margin size and the number of training errors allowed. A smaller C0 allows for more errors, while a larger C0 enforces a smaller margin and potentially overfits the data. Generally, a good starting point for C0 is 1.

In conclusion, selecting the appropriate kernel and hyperparameters is crucial for achieving optimal SVM performance. For linearly separable data, the linear kernel is a good option. For non-linear data, the RBF kernel with an appropriate gamma value is good, while the sigmoid and quadratic kernels are also good options.