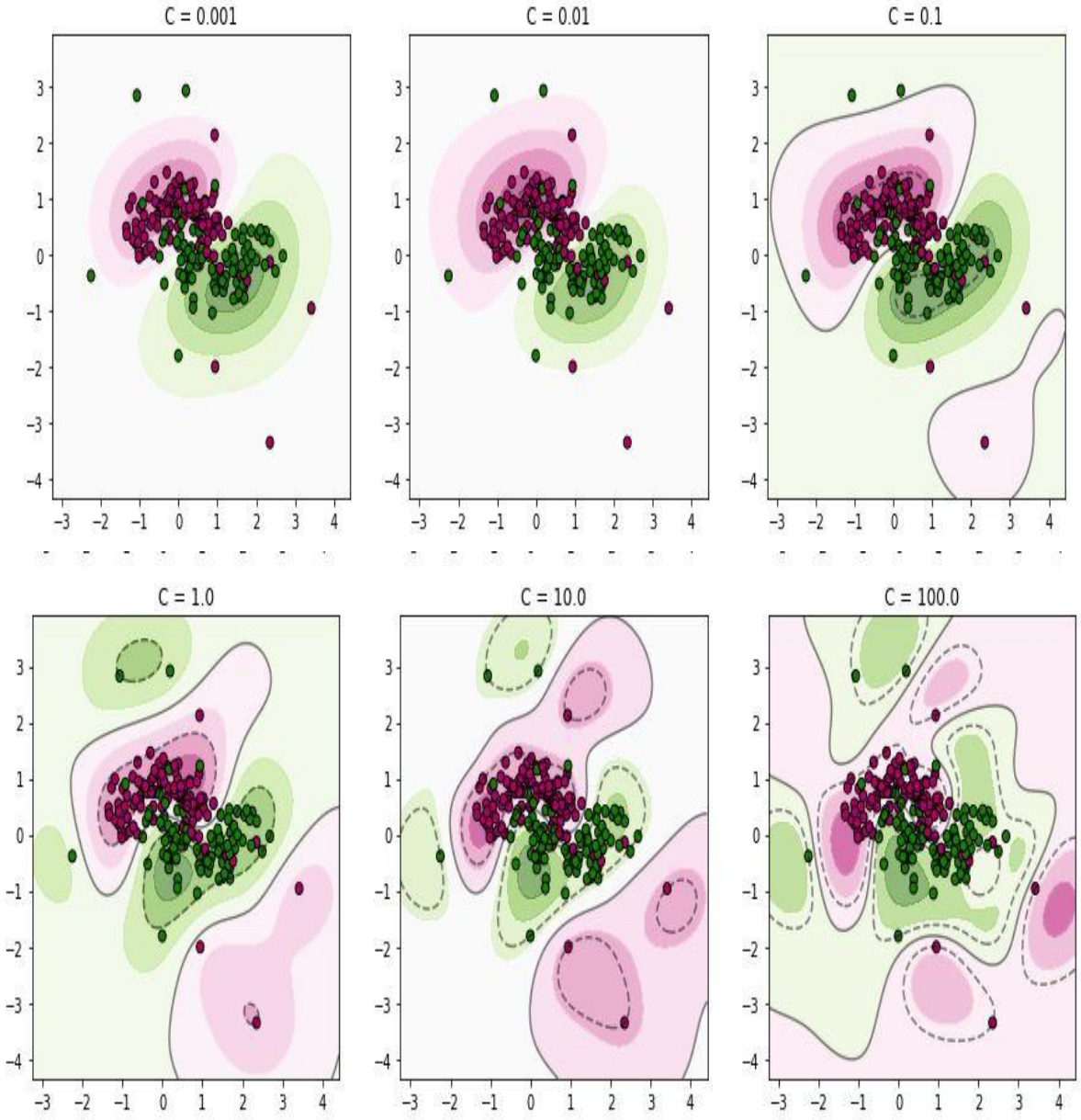
A. The C parameter modifying how much you want to avoid misclassifying for each training example. While C increasing, SVM will choose the smaller-margin hyperplane that makes it softer (allowing more slacks) to get all the training points classified, but yet overly large C could result in overfitting. While C is tiny, SVM will look for a larger margin separating hyperplane and will be less accurate and mismatch.

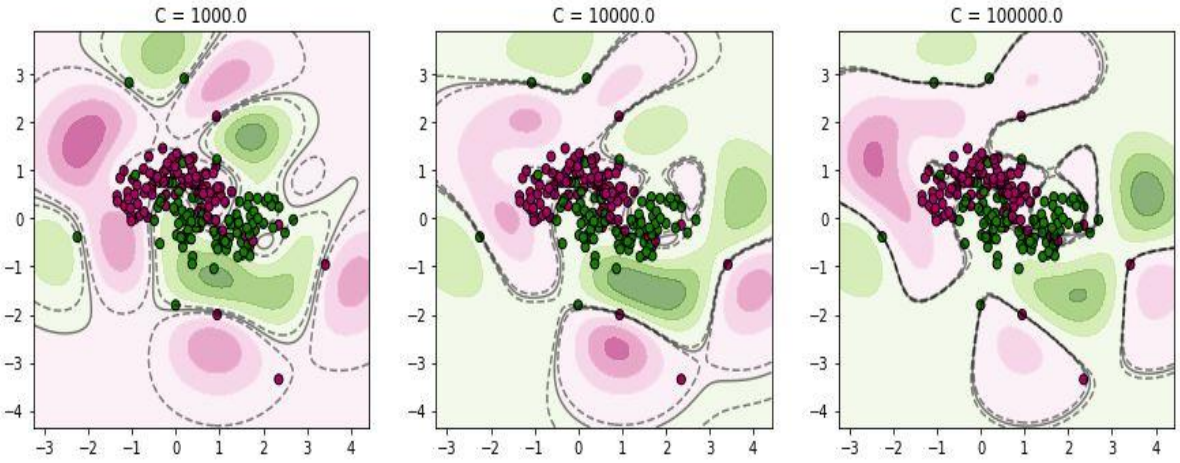
Both training and validation Error decrease while C value increase, but validation error trends to increase and bouncing while C value in after 10^2 = 100

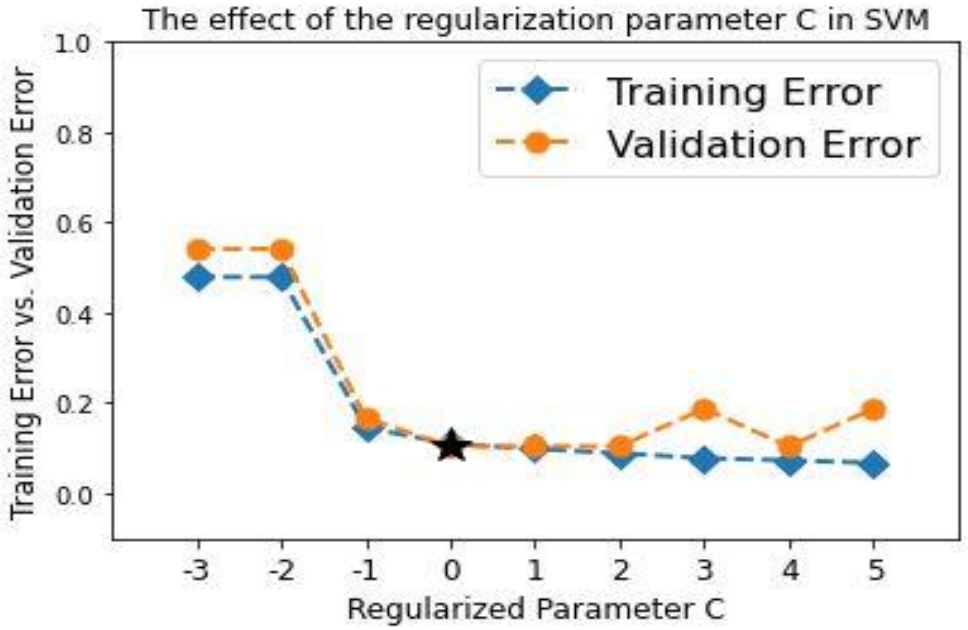
By observing the plot, the best C value is around 10^0 = 1, and the test accuracy is around 83.3%

```
Best C value is: 1.0 , Test Accuracy is: 0.833.

[52]: [<matplotlib.lines.Line2D at 0x7efe17ffc250>]
```







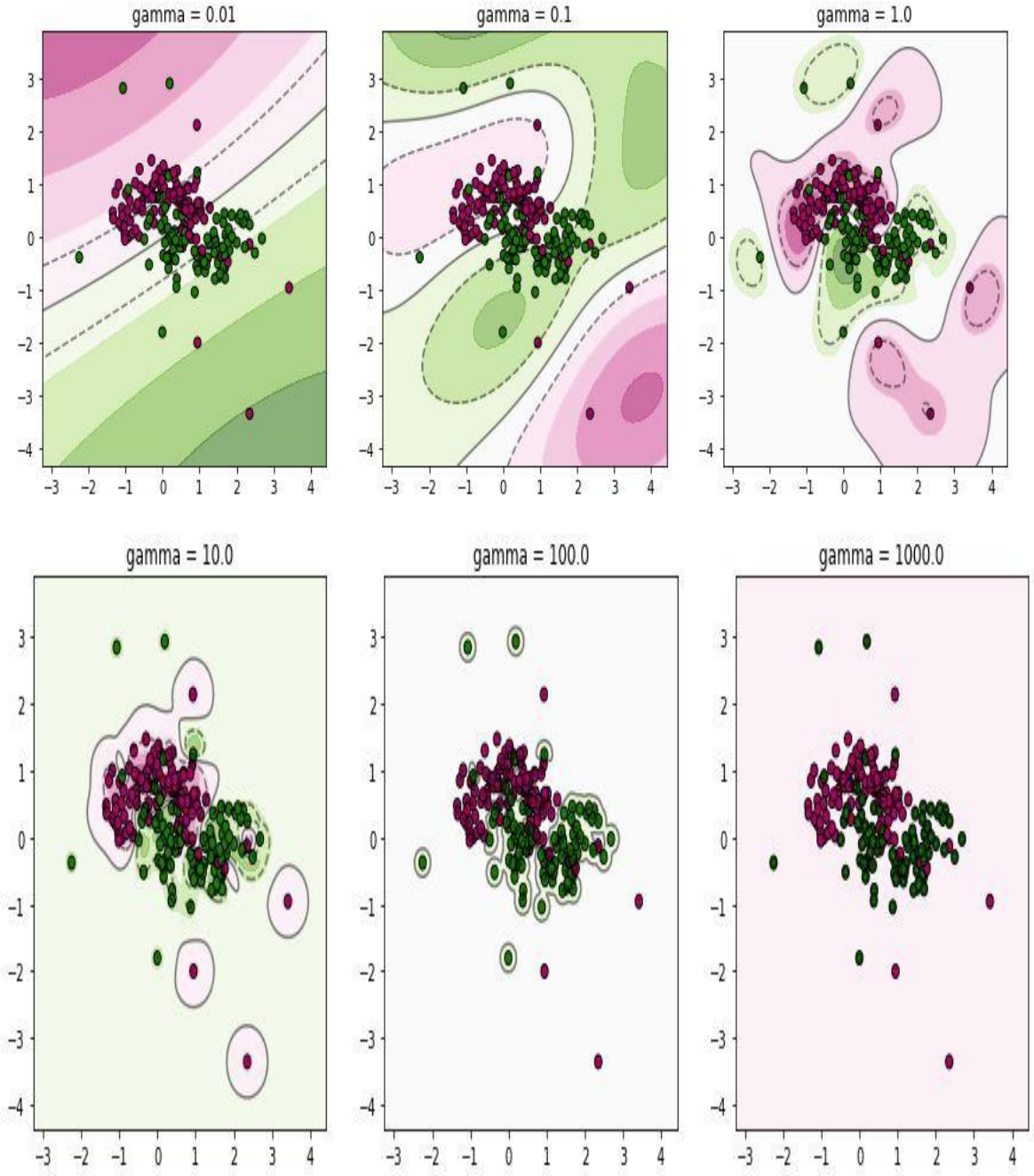
B. The gamma parameter defines how far a single training example reaches. while gamma is smaller, the single training example can reach far, and vise versa. When gamma is very small, the model is too constrained and cannot capture the shape of the data. When gamma is very large, the model makes islands of choice limits around the data center and become not so worth for classification.

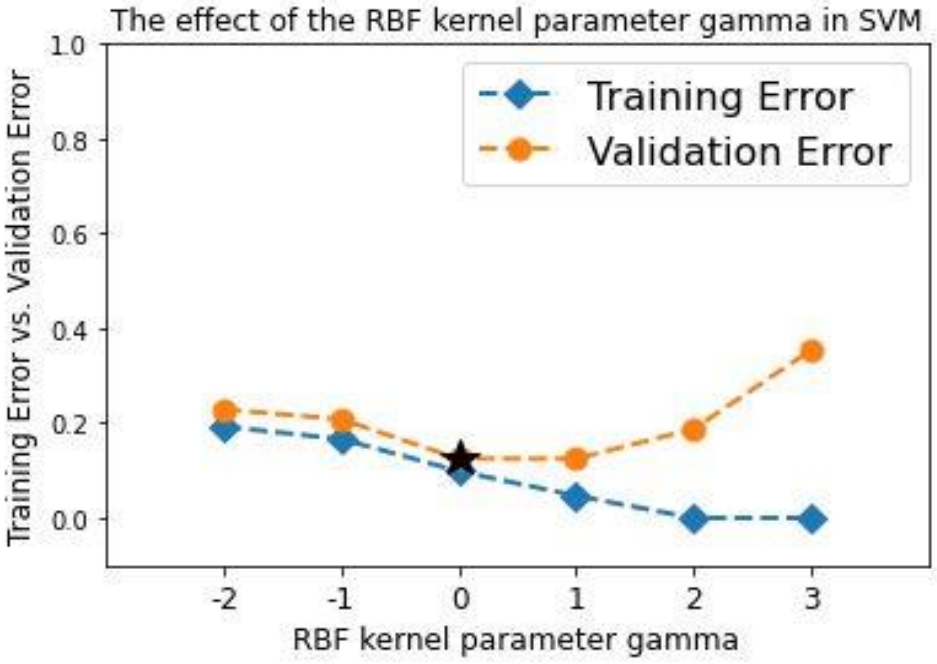
The training error decrease while gamma increase, but training error decrease at the first, when gamma reach value of 10^0 = 1, it increases.

By observing the plot, the best C value is around 10^0 = 1, and the test accuracy is around 83.3%

```
The best value of gamma is : 1.0, Test Accuracy is: 0.833.
```

[54]: `[<matplotlib.lines.Line2D at 0x7efe199983d0>]`





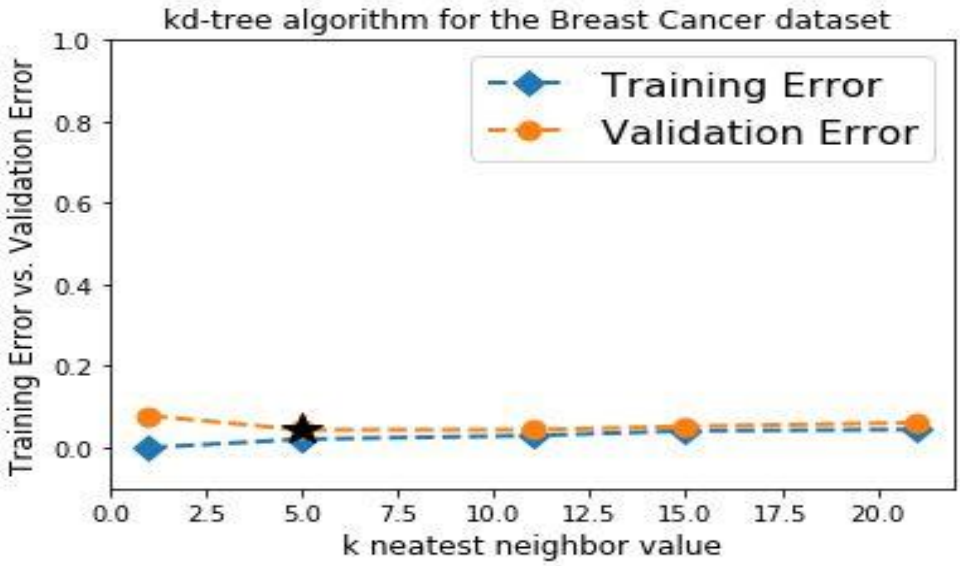
C.

```
C values :  [100.0, 1000.0, 10000.0, 10000.0]
Gamma values :  [0.01, 0.01, 0.001, 0.01]
Test error with corresponding C and gamma pairs :  [0.034782608695652195, 0.05217391304347829, 0.060869565217391286, 0.05217391
304347829]
Error Reduce :  2.608695652173909
Final test set accuracy on the model corresponding to the best C and gamma:  97.3913043478261
```

The best C and Gamma picked is while C = 100, and Gamma = 0.01 with the best accuracy of 97.39%

D. By observing the plot, the best K value is 5 with test accuracy of 95.7%

```
Best k value is: 5, Test accuracy is: 0.957.
[7]: [<matplotlib.lines.Line2D at 0x7fd1352d24d0>]
```



Conclusion:

By comparing both SVM and KNN model for the given dataset, I'd prefer the SVM model since it have the better accuracy of the data, in which 97% > 95%, but for a lager dataset and convenient KNN is also a good choice.