Binary Classification with SVM

-Anirudh Shaktawat

Abstract: The objective of this challenge is to classify the dataset with binary class labels using support vector machines (SVM). We tried SVM model with linear, polynomial and radial basis kernels on the given dataset. We then tuned the hyper-parameters of the models to select the most suitable and appropriate parameters to classify the given dataset. The tuned rbf kernel model was found to have the highest accuracy but due to the signs of overfitting, the default rbf kernel model was chosen and used to predict the class labels on the testing data.

I Dataset

The training dataset provided for this challenge consisted of two features – 'Feature 0' and 'Feature 1' and a target variable – 'Class' with binary labels (0s and 1s). A testing data was also provided with both the features and without the target variable. Both the training and testing data consisted of 200 data points.

II Model Selection and Hyper-parameter Tuning

Firstly, we trained the SVM model with the default settings without any parameter tuning with 3-fold cross validation. The mean CV score came out to be 0.785. The decision boundary is shown in Fig1.In effort to increase the accuracy, we performed Grid Search to tune the parameters of the model. There are 3 important parameters when it comes to tuning with SVM - **C**, **kernel**, and **gamma**.

C represents the penalty parameter, larger the value of C, the more we are penalizing the misclassifications. So, there is a tradeoff with respect to C, the larger we make C, the smaller will the margin be but we will be getting more of the training data correct. So, if we make C very large to get most of the training data correct, then we may compromise with the generalization property (robustness) of the model. In some cases, even when the data is truly linearly separable, we would like to tradeoff a small C for greater margins to make our model robust, especially in case of noisy data. However, when our data is not linear in the given dimensions, then to make our classifier more powerful, we do basis transformation. Kernel trick does the same thing for us, it takes the data in the given dimensions and transform it to some higher dimensions to make it linear and then applies a linear classification. This facility is provided by the kernel parameter in the SVM algorithm. Gamma is a parameter which is associated with rbf or poly kernel and deals with the measure of complexity of the model. Small gamma means less complexity and large gamma means more complexity and very large gamma may eventually lead to overfitting.

Although we could apply grid search on kernel hyper-parameter during tuning, it is better if we apply grid search for each kernel independently. It helps us in better visualization of the decision boundaries since we have only 200 datapoints which aids our decision making in choosing the right model.

III Results

SVM Model	**CV Score**	**Decision Region Boundary**
Default	0.785	Figure 1
Linear (Tuned)	0.75	Figure 2
Polynomial (Tuned)	0.72	Figure 3
Radial Basis (Tuned)	0.795	Figure 4

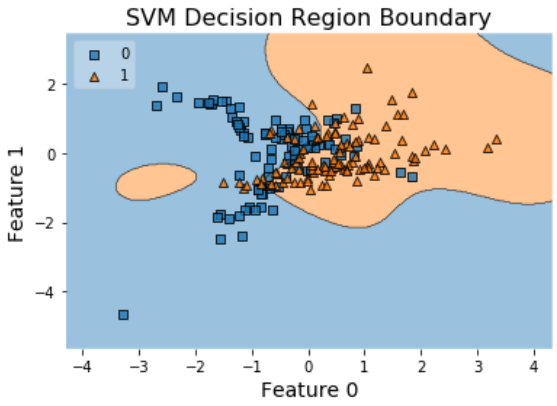

Figure 1

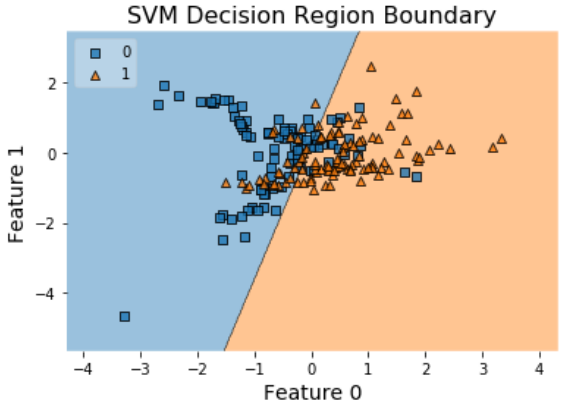

Figure 2

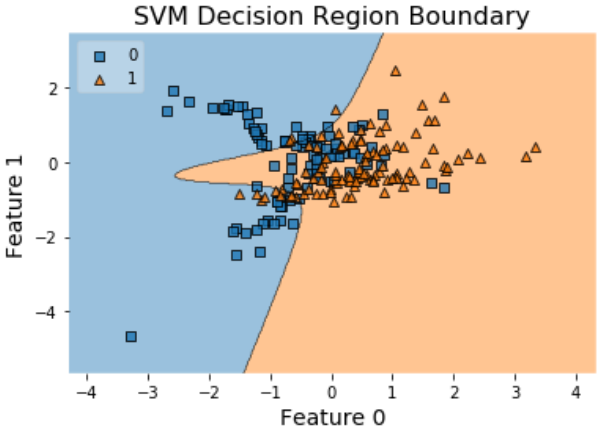

Figure 3

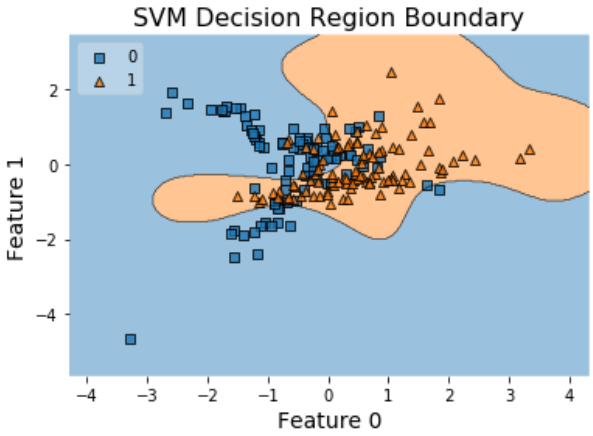

Figure 4

IV Conclusion

From all the above models, SVM model with the default settings (default kernel = rbf) and tuned SVM model with rbf kernel are giving the highest accuracy of around 78-80% as compared to models with linear and poly kernel which have accuracy of around 75% and 72% respectively.

So, it indicates that for the given data, rbf kernel is appropriate. Now, the difference between the accuracy of the default rbf kernel SVM model and the tuned rbf SVM model is not very significant. But the tuned SVM model has more complex boundaries since it has higher values of C and gamma. So, there is a danger for this model that this model might lead to overfitting and it might not generalize well on the testing data.

So, due to the danger of overfitting, we have chosen the rbf kernel SVM model with the default settings (C=1, gamma = auto).