CENG 499 - HW3

Alperen Oğuz Çakmak

January 9, 2021

1 Part 1: Decision Tree

1.1 Information Gain

Using information gain, accuracy is %95.3 and plot is named as Info_Gain.pdf.

1.2 Gain Ratio

Using gain ratio, accuracy is %95.6 and plot is named as Gain_Ratio.pdf.

1.3 Average Gini Index

Using information gain, accuracy is %95.3 and plot is named as Gini_Index.pdf.

1.4 Gain Ratio with Chi-squared Pre-pruning

With Chi-squared pre-pruning, tree's accuracy is %94.8 and plot is named as Gain_Chi.pdf.

1.5 Gain Ratio with Reduced Error Post-pruning

Test results and referring to the tree diagram.

2 Part 2: Support Vector Machine

2.1 First Part

As the C value grows, margin increases and less miss-classifications are allowed in our SVM. As it can be seen from the figures below, when C is smaller, one of the points is miss-classified, as C increases our line gets closer to the strict location between 2 classes because it starts not tolerating that missclassified points in figures 1 and 2.

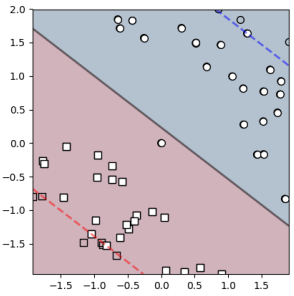

Figure 1: C = 0.01

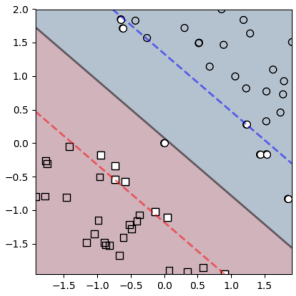

Figure 2: C = 0.1

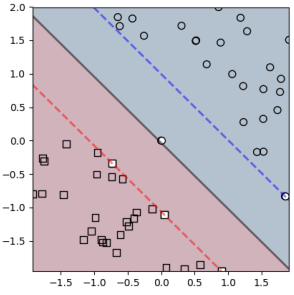

Figure 3: C = 1

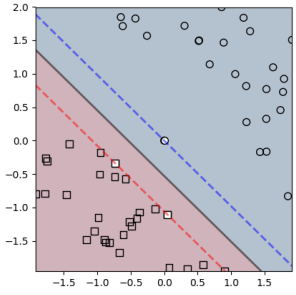

Figure 4: C = 10

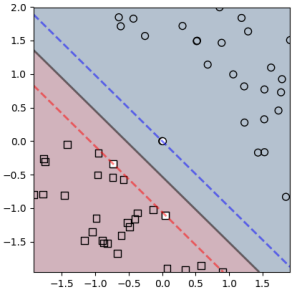

Figure 5: C = 100

2.2 Second Part

When data is not linearly seperable, we need to use more complex kernels. Kernel function changes the form of the lines. In this case, if we look at the data points, class 1 is clustered together close to the center and class 2 points are clustered as a circle which encapsulates class 1 points. So it is

not surpise that RBF did the best job classifying the points.

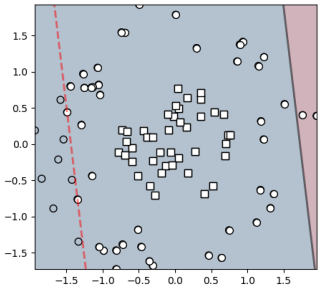

Figure 6: Linear

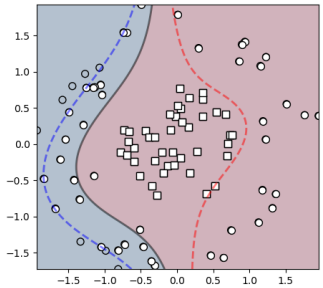

Figure 7: Polynomial

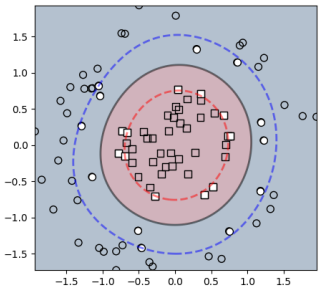

Figure 8: RBF

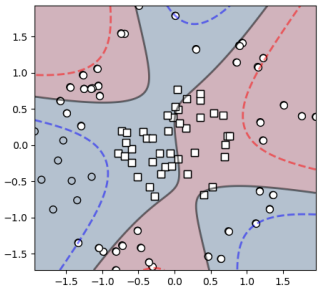

Figure 9: Sigmoid

2.3 Third Part

gamma	C				
	0.01	0.1	1	10	100
-	0.644	0.675	0.707	0.706	0.706

Table 1: Linear kernel

Best C for linear kernel SVM is 1. It has accuracy of %76.

gamma	C				
	0.01	0.1	1	10	100
0.00001	0.5376	0.5376	0.5376	0.6067	0.6441
0.0001	0.5376	0.5376	0.6113	0.6620	0.7106
0.001	0.5376	0.5898	0.6779	0.7296	0.7311
0.01	0.5376	0.5376	0.7117	0.7091	0.7091
0.1	0.5376	0.5376	0.7096	0.7096	0.7096
1	0.5376	0.5376	0.7096	0.7096	.7096

Table 2: RBF kernel

Best hyperparameters are C = 100 and gamma = 0.001. It has accuracy of %80. For this task

gamma	C				
	0.01	0.1	1	10	100
0.00001	0.5376	0.5376	0.5376	0.5376	0.5376
0.0001	0.5376	0.5376	0.5376	0.5376	0.5417
0.001	0.5376	0.5417	0.6584	0.7322	0.7240
0.01	0.7322	0.7240	0.7230	0.7230	0.7230
0.1	0.7230	0.7230	0.7230	0.7230	0.7230
1	0.7230	0.7230	0.7230	0.7230	0.7230

Table 3: Polynomial kernel

there is 2 best hyperparameter selections, C = 0.01 and gamma = 0.01, and C = 10 and gamma = 0.001. They both have %79 accuracy.

gamma	C				
	0.01	0.1	1	10	100
0.00001	0.5376	0.5376	0.5376	0.5934	0.6221
0.0001	0.5376	0.5376	0.5939	0.6226	0.6400
0.001	0.5376	0.5919	0.5534	0.5386	0.5232
0.01	0.5530	0.5253	0.5110	0.5058	0.5058
0.1	0.5504	0.5248	0.5222	0.5217	0.5212
1	0.5478	0.5161	0.5155	0.5161	0.5161

Table 4: Sigmoid kernel

For this one best hyperparameters are C = 100 and gamma = 0.0001. It has accuracy of %62.

2.4 Fourth part

2.4.1 Without handling the imbalance problem

Report test accuracy. Can accuracy be a good performance metric? Report confusion matrix and comment on it. Report additional metrics here if you want.

Test accuracy is %83 for this task which seems to be good. However, accuracy might not be the best metric in order to evaluate our classifier because for this task we have an imbalanced data. When the data is imbalanced (one class is majority and other is minority) accuracy does not give us the detailed information about our classifier.
Instead of accuracy we can use confusion matrix. With confusion matrix we can see both classes and how many predictions are correct or incorrect. Let's take a look at this task's confusion matrix:

$$\begin{bmatrix} 0 & 191 \\ 0 & 949 \end{bmatrix}$$

If we look at the first row we can see that all of our class 1 predictions are incorrect so we can say that our classifier did a terrible job with first class but on the other hand, for class 2 all of the predictions are correct so we would have %100 accuracy. As you can see there is an imbalance in predictions and we can clearly see what our classifier did well and did not well with confusion matrix.

2.4.2 Oversampling the minority class

With undersampling I got %56 test accuracy which is a very bad accuracy. Also confusion matrix is:

$$\begin{bmatrix} 129 & 62 \\ 429 & 520 \end{bmatrix}$$

Compared to normal version, oversampling improved our class 1 predictions but it has something like %66 accurracy but class 2 gives really bad results this time.

2.4.3 Undersampling the majority class

With undersampling I got %77 test accuracy which is lower than normal version. Also confusion matrix is:

$$\begin{bmatrix} 39 & 152 \\ 108 & 841 \end{bmatrix}$$

Compared to normal version, undersampling did better with class 1 but still results are not good and also it got worse for class 2

2.4.4 Setting the class_weight to balanced

I got %61 accuracy in this task which is lower than normal version. Anc confusion matrix is:

$$\begin{bmatrix} 110 & 80 \\ 355 & 594 \end{bmatrix}$$

When we look at the confusion matrix it did not do well on class 2 (always worse than normal version) and it is better at predicting class 1. But at the end it is just slightly worse than oversampling.