Ceng499-The3 Report

Seda Civelek-2237147

June 12, 2021

1 Part 1: Decision Tree

1.1 Information Gain

Test accuracy for information gain: 0.9333

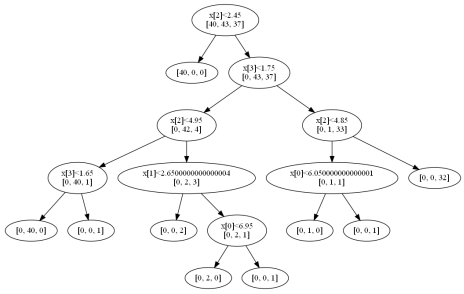


Figure 1: Decision tree using information gain

1.2 Average Gini Index

Test accuracy for average gini index: 0.9

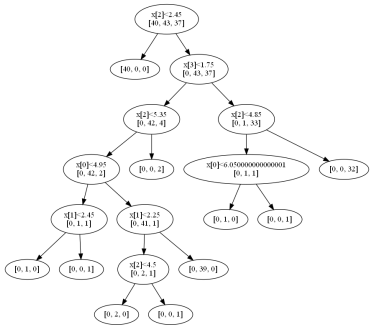


Figure 2: Decision tree using average gini index

1.3 Information Gain with Chi-squared Pre-pruning

Test accuracy for information gain with pre-pruning: 0.9666

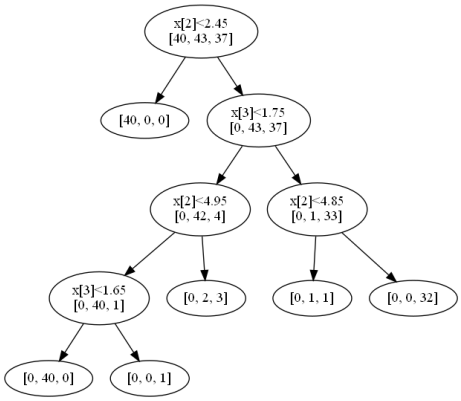


Figure 3: Decision tree using information gain with pre-pruning

1.4 Average Gini Index with Chi-squared Pre-pruning

Test accuracy for average gini index with pre-pruning: 0.9

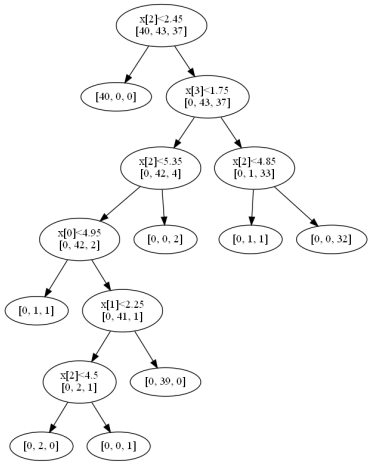
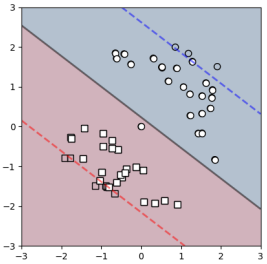


Figure 4: Decision tree using average gini index with pre-pruning

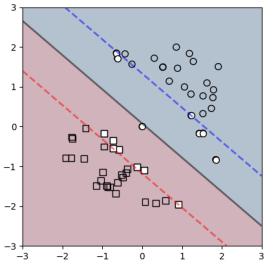
2 Part 2: Support Vector Machine

2.1 First Part

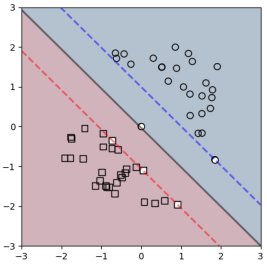
With changing C values, we can decide how much to avoid misclassifying. For large C values, we can obtain smaller margin, but more instances are correctly classified. However, for small C values, we obtain larger margin, but some of the instances will be misclassified.



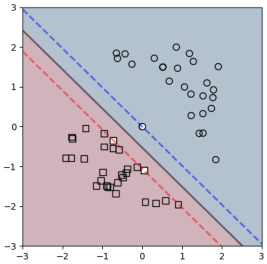
(a) C=0.01



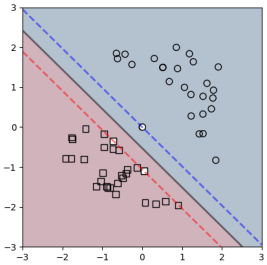
(b) C=0.1



(c) C=1



(d) C=10

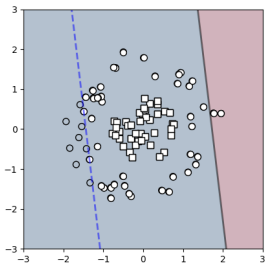


(e) C=100

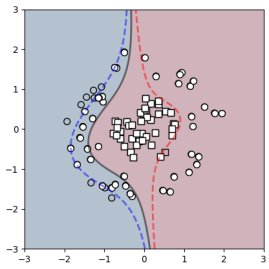
Figure 5: Plots of SVMs for each C value

2.2 Second Part

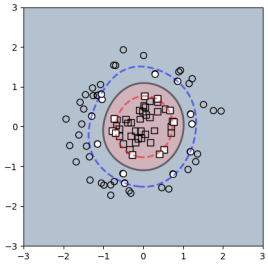
Linear kernel is useful for linearly separable data. However, the dataset is not linearly separable. We should use other kernels. With using 'rbf' kernel, we can get the best classification. However, there are some misclassified instances since we set the C value as 1.0.



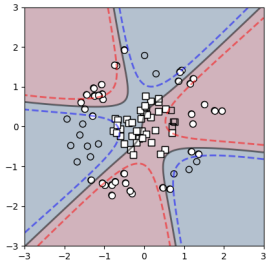
(a) kernel='linear'



(b) kernel='polynomial'



(c) kernel='rbf'



(d) kernel='sigmoid'

Figure 6: Plots of SVMs for each kernel

2.3 Third Part

Best parameters are 'C': 100, 'gamma': 0.01, 'kernel': 'rbf'.
Test accuracy with best hyperparameters: 0.8806

gamma	C				
	0.01	0.1	1	10	100
-	0.774	**0.811**	0.779	0.738	0.730

Table 1: Linear kernel

gamma	C				
	0.01	0.1	1	10	100
0.00001	0.510	0.510	0.510	0.511	0.747
0.0001	0.510	0.510	0.511	0.749	0.796
0.001	0.510	0.511	0.751	0.828	0.857
0.01	0.510	0.761	0.869	0.894	**0.896**
0.1	0.510	0.510	0.883	0.885	0.885
1	0.510	0.510	0.510	0.510	0.510

Table 2: RBF kernel

gamma	C				
	0.01	0.1	1	10	100
0.00001	0.510	0.510	0.510	0.510	0.510
0.0001	0.510	0.510	0.510	0.510	0.510
0.001	0.510	0.510	0.551	0.737	0.810
0.01	0.737	0.810	**0.875**	0.867	0.867
0.1	0.867	0.867	0.867	0.867	0.867
1	0.867	0.867	0.867	0.867	0.867

Table 3: Polynomial kernel

gamma	C				
	0.01	0.1	1	10	100
0.00001	0.510	0.510	0.510	0.510	0.737
0.0001	0.510	0.510	0.510	0.737	0.774
0.001	0.510	0.510	0.736	0.768	**0.787**
0.01	0.510	0.511	0.337	0.687	0.681
0.1	0.510	0.510	0.510	0.510	0.510
1	0.510	0.510	0.510	0.510	0.510

Table 4: Sigmoid kernel

2.4 Fourth part

2.4.1 Without handling the imbalance problem

Confusion matrix
TN: 3 FP: 76 FN: 0 TP: 1421
Test accuracy of imbalanced dataset: 0.9493
Recall of imbalanced dataset: 1.0
Precision of imbalanced dataset: 0.9492

When we look at the confusion matrix, we can see that the number of negative examples are so low. We can say that the model doesn't understand negative examples well.

2.4.2 Oversampling the minority class

Confusion matrix for oversampled dataset
TN: 31 FP: 48 FN: 15 TP: 1406
Test accuracy of oversampled dataset: 0.958
Recall of oversampled dataset: 0.9894
Precision of oversampled dataset: 0.9669

When we oversampled the minority class and making the ratio 1:1, we obtain a model that can understand negative examples more.The number of negative examples are increased. We see that looking recall and precision values is more important for imbalanced datasets.

2.4.3 Undersampling the majority class

Confusion matrix for undersampled dataset
TN: 50 FP: 29 FN: 246 TP: 1175
Test accuracy of undersampled dataset: 0.8166
Recall of undersampled dataset: 0.8268
Precision of undersampled dataset: 0.9759

When we undersampled the majority class and making the ratio 1:1, we obtain a model that can understand negative examples more. The number of negative examples are increased. However, the test accuracy is decreased. We see that looking recall and precision values is more important for imbalanced datasets.

2.4.4 Setting the class_weight to balanced

Confusion matrix for balanced dataset
TN: 38 FP: 41 FN: 21 TP: 1400
Test accuracy of balanced dataset: 0.9586
Recall of balanced dataset: 0.9852

Precision of balanced dataset: 0.9715

Setting class weight as balanced changes the weights of examples according to their class frequencies. Hence, our model understands negative examples more accurately.