DATA MINING PROJECT

RHITURAJ BHALERAO

Data preprocessing

The data has 2500 rows and 68 columns. Out of which, 67 columns are of X and the last column is of Y. Out of those 68 columns, 4 columns are categorical. The columns which are categorical are [‘x5’, ‘x13’, ‘x64’, ‘x65’]. I made dummy variables for all the categorical variables.

The ‘x5’ column has 5 categories
['A', 'B', 'C', 'D', 'E']
The ‘x13’ column has 5 categories
['Alpha', 'Beta', 'Chi']
The ‘x64’ column has 5 categories
['Ma', 'Mk', 'Mm', 'Mp']
The ‘x65’ column has 5 categories
['NPT', 'NST', 'PT', 'ST']

After creating the dummies, the columns are increased to 78 columns. So, my input variables have 78 columns.

The data was then scaled using the MinMax scalar. So, all the values in the data were scaled to between 0 and 1 by using the feature range between 0 and 1.

Splitting the Data

After scaling the data, the data was split into Train and test data samples. I split into 80% of the train data and 20% of the test data.

Up-sampling

The classes in y training sample are divided into imbalance.

Classes	Samples
-1	1516
+1	484

There is imbalance in the number of samples for 2 classes. Hence, up sampling the data to get the equal number of samples for both classes.

After up sampling, the 2 classes are balanced. Now the training sample shape is (3032, 78) and testing sample is (500,).

Classes	Samples
-1	1516
+1	1516

I also tried down-sampling but it loses sufficient amount of the data from the sample. So, I moved forward using the Up-sampling.

1. Model Building – Support Vector Machine (SVM)

First model I used was Support Vector Machine (SVM). The following parameters were used for the initial fitting of the model.
C = 1.0, gamma = 'auto-deprecated', kernel = 'rbf'
I got the following Testing and Training accuracy and the confusion matrix.

Training	Testing
74.64 %	72%

Confusion Matrix

250	125
15	110

Balanced error rate = 0.225.

After the initial model, I tuned the parameters using the Grid search CV. The following parameters were found to be the best parameters.
C = 1.0, gamma = 10.
I got the following Testing and Training accuracy and the confusion matrix.

Training	Testing
100%	75%

Confusion Matrix

375	0
125	0

Balanced error rate = 0

The model with the best parameters clearly overfits as the training accuracy is 100%. And with the 100% accuracy, I don’t get a proper confusion matrix. So, I decided to tune the C and the gamma parameters using different combinations. I found C = 100, gamma = 0.001, kernel = ‘rbf’ gives me the best results of all.

Training	Testing
81.3%	78.2%

Confusion Matrix

288	87
21	104

Balanced error rate = 0.2

Clearly, the model seems to be fitting better than the 2 models tried earlier. So, this is my **best SVM model** for fitting the training data samples.

2. Model Building – Random Forest Classifier (RFC)

First model I used was Support Vector Machine (SVM). The following parameters were used for the initial fitting of the model.
Max_depth = 10, criterion = ‘gini’, min_samples_split = ‘2’
I got the following Testing and Training accuracy and the confusion matrix.

Training	Testing
99.86%	80.6%

Confusion Matrix

339	36
61	64

Balanced error rate = 0.292.

This model with the default parameters clearly overfits as the training accuracy is 99.86%. And with such accuracy, I don't get a proper confusion matrix. So, I decided to tune the max_depth, criterion and sample split using different combinations. I found C = 10, min_samples_split = 2, criterion = 'entropy' gives me the best results of all.

After the initial model, I tuned the parameters using the Grid search CV. The following parameters were found to be the best parameters.
Max_depth = 10, criterion = 'entropy', min_samples_split = '2'.
I got the following Testing and Training accuracy and the confusion matrix.

Training	Testing
92.11	79.4%

Confusion Matrix

300	75
28	97

Balanced error rate = 0.237

This model looks better than the previous one. This model even has a better balanced error rate than the previous model. I chose this to be the best model for Random forest classifier.

Choosing the final model

I took the SVM model as my final model as the balanced error rate = 0.2 is the minimum compared to all the other models I tried. Even the accuracies look better in that model as it doesn't overfit.

Predictions using the Test data

In the Test data, there are less columns than the training data sample.
So, Preprocessing the data in the same way as I did for the training data, but there are less categories in the columns, ‘x5’, ‘x64’ and ‘x65’.

The categories missing in the test data are:

X5 = {B, D}
X64 = {Mm, Mk}
X65 = {NST}

So, I added 5 columns with values as 0 into the test data and matched the shape with the training samples.