IEE 520- STATISTICAL LEARNING FOR DATA MINING

FINAL TERM PROJECT

FALL 2017

SUBMITTED BY: MITHUN MURALIDHAR

ASU ID: 1211309824

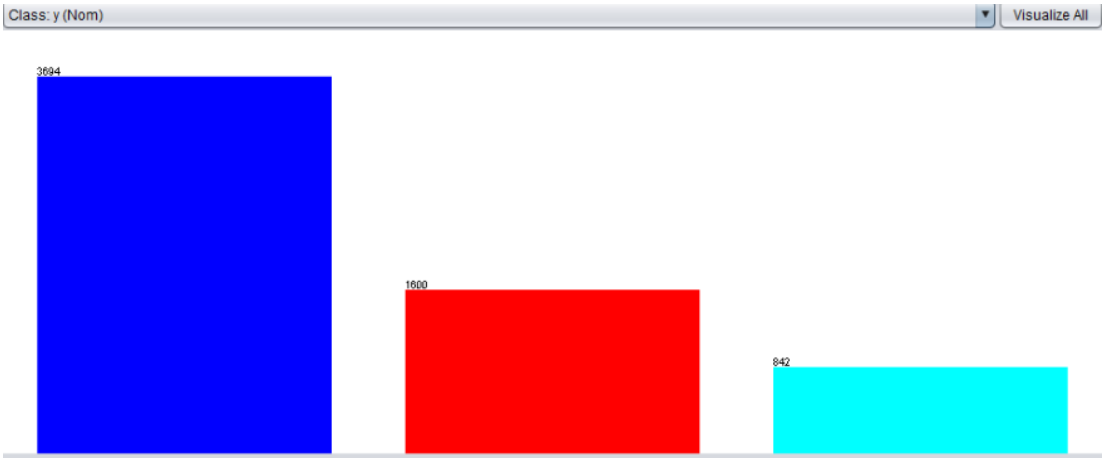
PROBLEM OBJECTIVE

Training and testing datasets are provided for analysis. Training dataset has known class attributes, which we use to draw a best possible classifier model. This classifier model is used on the testing data without known class attribute to predict the classes with good consistency.

DATASET

The dataset consist of 41 attributes of numeric and nominal types. Class label is a nominal attribute having three distinct type labels. The training dataset has 6136 instances. The given training data appears imbalanced for different class types. Ratio of differences for class 1 and 2 with class 0 (maximum counts) are 2.3 and 4.4 respectively.

Histogram of class attribute is shown below.



FILTER SELECTION APPROACH

Filter	Classifier	Accuracy %	Class 0 error rate	Class 1 error rate	Class 2 error rate	Balanced error rate
CASE 1	Bayes Net	70.9745	0.0108	0.5915	0.2691	0.2904
	J48	77.7949	0.2290	0.2763	0.1600	0.2217
	Random Forest	82.1451	0.1293	0.2513	0.1547	0.1784
	JRip	77.5497	0.1480	0.3103	0.2152	0.2245
CASE 2	Bayes Net	70.5556	0.015	0.5997	0.2686	0.2944
	J48	76.1481	0.245	0.2925	0.1780	0.2385
	Random Forest	81.5556	0.1394	0.2466	0.1672	0.1844
	JRip	75.7037	0.2022	0.3108	0.2158	0.2429
CASE 3	Bayes Net	73.9479	0.2260	0.3843	0.1710	0.2604
	J48	84.2083	0.2175	0.1685	0.0876	0.1578
	Random Forest	89.2031	0.0928	0.1445	0.0865	0.1079
	JRip	82.0938	0.1918	0.2181	0.1271	0.1790

Number of instances for class 0, 1, 2 before applying any filters were 3694, 1600 and 842 respectively.

CASE 1: Smote filter was applied to minority classes 1 and 2 to balance the instances close to the majority class 0. No spread subsample was done in this case. Number of instances for classes 0, 1 and 2 are 3694, 3200 and 3368 respectively.

CASE 2: Smote filter was applied to class 2 and class 1 by increasing the percentage to 340% and 125% respectively so that the instances in these classes come close to the majority class 0. For uniform class distribution, spread subsampling filter was applied by increasing the distribution spread to 1.0. Number of instances for classes 0, 1 and 2 are 3600.

***NOTE*:** In this case, there is some loss incurred in original majority class instances due to spread subsample filter.

CASE 3: Several iterations of smote filter was applied until the majority class instances started to increase. This was done to ensure there was no loss in original instances of majority class. Following this, spread subsample filter with distribution spread equal to 1.0 was applied to get a uniform distribution of all classes. Number of instances of all classes are 6400.

In all the cases, the order of instances were randomized.

FILTER CASE CONCLUSION AND SELECTION OF BASE CLASSIFIER

FILTER	CLASSIFIER	ACCURACY	CLASS 0 ERROR RATE	CLASS 1 ERROR RATE	CLASS 2 ERROR RATE	BALANCED ERROR RATE
CASE 3	Bayes Net	73.9479	0.2260	0.3843	0.1710	0.2604
	J48	84.2083	0.2175	0.1685	0.0876	0.1578
	Random Forest	89.2031	0.0928	0.1445	0.0865	0.1079
	JRip	82.0938	0.1918	0.2181	0.1271	0.1790

Case 3 filter provides better accuracy for all the base classifiers compared to the other two cases. This is quite evident as we obtain uniform class distribution for classes in this case without eliminating any original instance from majority class. This filter provides less balanced error rate for J48, random forest and JRip classifiers and the deviation error rate for classes is considerable. This gives unbiased predictions for the model.

Bayes network shows large deviations between class error rates and also the balanced error rate is more for this filter. Results of JRip is also not so satisfying. Random forests and J48 show considerable improvement in this filter. The balanced error rate is less, accuracy of the model is high and in-between error rate for classes is less deviated. Thus we go with Random forest and J48 as our base classifiers for future analysis.

Note: Other base classifiers such as naive bayes, kNN and SVM were observed for the training data, but their results were not satisfying.

PARAMETER SELECTION FOR BASE CLASSIFIERS

BASE CLASSIFIER	CLASSIFIER PARAMETER	ACCU-RACY	CLASS 0 ERROR RATE	CLASS 1 ERROR RATE	CLASS 2 ERROR RATE	BALANCED ERROR RATE
J48	CF 0.25, MO 2, REP=F(NL 1184, SZ 2367)	84.2083	0.2175	0.1685	0.0876	0.1578
	CF 0.1, MO 2, REP=FALSE (NL 874, SZ 1747)	84.1302	0.2129	0.1732	0.0898	0.1586
	CF 0.1, M0 2, REP= TRUE (NL 462, SZ 923)	82.3750	0.2176	0.1970	0.1140	0.1762
	CF 0.1, MO 5, REP= FALSE(NL 565, SZ 1129)	83.9427	0.2043	0.1779	0.0993	0.1605
	CF 0.1, MO 5, REP= TRUE(NL 306, SZ 611)	82.0156	0.22	0.2010	0.1184	0.1798
	C 0.1, MO 6, REP= TRUE (NL 255, SZ 509)	81.7969	0.2173	0.2062	0.1225	0.182
RANDOM FOREST	Bag size 100, max depth 0,Nof 0	89.2031	0.0928	0.1445	0.0865	0.1079
	Bag size 90, max depth 0,NOF 0	89.0677	0.0960	0.1445	0.0873	0.1092
	Bag size 90, max depth 15, NOF 0	88.7813	0.1107	0.1390	0.0867	0.1121
	Bag size 90, max depth 15, NOF 10	88.8698	0.1164	0.1321	0.0853	0.1127
	Bag size 70, max depth 30, NOF 6	88.901	0.1010	0.1457	0.0860	0.1109
	Bag size 90, depth 30 , NOF 6	89.0469	0.0965	0.144	0.0871	0.1092
	Bag size 90, max depth 15, NOF 15	89.0677	0.0960	0.1445	0.0873	0.1092
	Bag size 90 , depth 45, NOF 6	89.0677	0.0960	0.1445	0.0873	0.1092
	Bag size 100, max depth 60, NOF 10	89.0938	0.1028	0.1395	0.0848	0.1090

NL= Number of leaves, SZ= Size of tree, CF= Confidence factor, MO= Minimum no. of objects, REP= Reduced error pruning, NOF= Number of features.

J48 CLASSIFIER: As Confidence factor is decreased, the complexity reduces with negligible effect on the accuracy of the model. Thus confidence factor was reduced to 0.1 from 0.25. Minimum number of objects is another important parameter that decreases both complexity and accuracy of the model as we increase it. Reduced error pruning reduces complexity further but there is certain deviation observed in error rate and accuracy. Thus an important decision lays in deciding between complexity and balanced error rate for the model. Highlighted portion in the table suits the following inferences made, thus aiding a model with better balanced error rate.

RANDOM FOREST CLASSIFIER: Increase in depth results in following changes to the model: Increases accuracy and complexity of the model but decreases balanced error rate. Thus when we pick a parameter for depth, we should pay close attention to changes in error rate and complexity of the model. Increasing features creates correlation between trees, thus affecting binomial probability property. Number of features of the model is given by the following formula in Weka int (log_2 (#predictors) + 1) and for 40 attributes, we get number of features as 6. Based on the inference obtained, highlighted portion in the above table is considered the best parameter setting for random forest.

META-CLASSIFIER WITH SELECTED BASE LEARNERS

Adaboost Meta classifier reduces variance and bias in the model by iteratively reweighting training data focusing on previous errors.

CLASSIFIER	BASE CLASSIFIER	CLASS 0 ERROR RATE	CLASS 1 ERROR RATE	CLASS 2 ERROR RATE	BALANCED ERROR RATE	ACCURACY
Adaboost	J48	0.1643	0.1360	0.0754	0.1252	87.4688 %
Bagging	Random forest	0.0998	0.1560	0.0906	0.1154	88.4479 %
Bagging	J48	0.1648	0.1525	0.0889	0.1354	86.4583 %
Adaboost	Random forest	0.1059	0.1421	0.0870	0.1116	88.8281 %

NOTE: For Bagging with random forest and Adaboost with random forest: 5 folds were used for cross-validation, number of iterations were down to 50 to reduce computational effort and time.

TESTING: Predictions for the testing dataset is submitted in an excel file ‘test data 520’.

CONCLUSION: Adaboost with random forest is selected as the best model for this dataset as it combats over fitting in the model. Parameter changes made is Confidence factor from 0.25 to 0.1, minimum number of objects from 2 to 5, Reduced error pruning= false. This model made its predictions with better balanced error rate than rest of the models and with good accuracy of 88.8281 %.

REFERENCES

Chawla, N. V., K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. "SMOTE: Synthetic Minority Over-sampling Technique." 2011, Journal Of Artificial Intelligence Research, Volume 16, Pages 321-357, 2002.