Mining Caravan Insurance Database Technical Report

Tarek Amr (@gr33ndata)

Abstract

Caravan Insurance collected a set of 5000 records for their customers. The dataset is composed of 85 attributes, as well as extra a label stating whether a customer purchased their mobile home policies insurance policies or not. In this report we need to infer a classification model based on this data, so Caravan can tailor their email campaign to contact customers who are most likely to purchase this insurance policy.

1 Data Preparation

1.1 Data Cleansing

First, we had to make sure that there are no missing data in our labelled dataset. Then data was converted to ARFF (Attribute-Relation File Format), to be used with Weka (Hall et al., 2009) and Orange (Curk et al., 2005) later on. After that, 1-NN (K-Nearest Neighbour with k=1), using Euclidean distance measure, was applied on the dataset as proximity-based outlier detection. In the end, no missing data nor outliers were reported in the dataset.

1.2 Sampling and Balancing

The labelled dataset was randomly split into two equal subsets; training and testing datasets. One problem with our dataset is that it contains imbalanced classes. In the whole data, as well as in the training dataset, the ratio between the majority class (Caravan=0) and the minority class (Caravan=1) is about 16:1. As we will see later, such imbalance causes various problems: Classifier such as C4.5 (J48) is mislead to classify all records into the majority class, especially when applying aggressive pruning. Disabling pruning altogether did not improve the true positive and false positive rates of the minority class much. Prati et al. (2004) added that the classification performance degrades more for imbalanced data the more the classes are overlapping. When applied *ClassificationViaClustering* classifier on our dataset (Lopez et al., 2012), it gave results close to chance (true positive and false positive rates in the range of 0.5). We can deduce from this that our two classes ought to be overlapping. Similarly, *Artificial Neural Networks* fail to classify data correctly when trained on imbalanced data (DeRouin et al., 1991).

Three different approaches were applied here in order to balance the training dataset:

1. Random Over-Sampling (ROS)
2. Random Under-Sampling (RUS)
3. Synthetic Minority Over-sampling Technique (SMOTE)

As described by Chawla et al. (2011), in the first approach records are randomly duplicated from the minority class till the two classes are almost even, while in the second approach, random records are being taken from the majority class. In SMOTE, new samples are synthesised in the minority class by interpolating adjacent examples from that class. SMOTE is said to overcome the over-fitting behaviour caused by ROS, however our findings here opposed to this.

2 Feature Selection

Gao et al. (2010) concluded that *filter methods*, such as Information Gain (IG) and Chi-squared (χ^2), are more effective when applied after sampling. We also noted that out of the top 14 features selected by IG and χ^2, 13 of them were common when features were selected from un-sampled, ROS and RUS datasets. Hence it was decided to apply IG on RUS dataset, and to experiment with the top 3, 7 and 14 features.

For *wrapper methods*, sampled dataset were preferred as well, as we expect the misleading accuracy numbers obtained on imbalanced data to confuse the wrapped classifier. *Best First Forward Selection* (BFFS), *Best First Backward Selection* (BFBS) and *Genetic Search* (GS) are the main search algorithms used here for wrapper methods.

3 Classification

Two main classifiers were used here: Weka's implementation of C4.5 decision tree algorithm, known as *J48*, and *Multilayer Perceptron (MLP)* artificial neural network algorithm. Different Confidence Factor (CF) levels were tried with J48 to control the pruning level. Whereas for MLP, different numbers for hidden layers were tested, as well as the number of neurons within each layer. Learning rate for MLP were kept to its default value, '0.3', and learning rate decay was set to 'true', especially that our experiments proved such configuration to be the best.

In addition to the sampling techniques, we wrapped a MetaCost learner on top of the previously mentioned classifiers. Domingos (1999) suggested this approach as an alternative way to re-sampling the training data or tweaking the algorithms themselves to give more cost to misclassifying the minority class, i.e our class of interest. MetaCost is given a cost for misclassifying any of the 'c' classes, then it internally builds 'm' models using bootstrapped training data, then a bagging ensemble method is applied to minimize the misclassification cost. For the case of '$c = 2$', the cost matrix is given in the form of $[TP_{class_1} FP_{class_1}; TP_{class_2} FP_{class_2};]$. Different costs were tested here as we will see later.

4 Evaluation Criteria

As stated earlier, classification accuracy is misleading in our case due to classes imbalance. For example, J48 with aggressive pruning, when applied on the imbalanced training dataset, gives accuracy of about 94%, despite the fact that it blindly classifies all records into the majority class. The accuracy number is nothing in this case but the percentage of the majority class to the whole data. Prati et al. (2004) suggested using true positive rate (TP_{rate}) and false positive rate (FP_{rate}) instead. Also in our case, our main objective is to predict candidate buyers. This means that our main priority is to maximize the true

positive rate of the minority class. Also, we need to have an almost zero false positive rate with respect to that class. Therefore they are the two measurements used mainly in this report. Notice that we use TN_{rate} and FN_{rate} to measure the majority class performance. TN_{rate} is equal to $1 - FP_{rate}$ of the minority class, and vice versa.

$$TP_{rate} = \frac{TP}{TP + FN} \quad and \quad FP_{rate} = \frac{FP}{FP + TN} \tag{1}$$

ANOVA one way test (using Welch approximation) and pairwise t-test with Holm adjustment are used to test differences between sampling techniques, feature selection algorithms and classifiers. Unless otherwise mentioned, significance level of all the tests is set to 0.05.

5 Results and Discussion

5.1 Decision Tree (C4.5/J48)

Training the classifier on imbalanced dataset (no sampling) resulted in systematically classifying everything into the majority class. In fig. 1 we can see such behaviour on the un-sampled data, where TP_{rate} and FP_{rate} for the minority class were kept at zero. When pruning was disabled, the two rates were slightly improved.

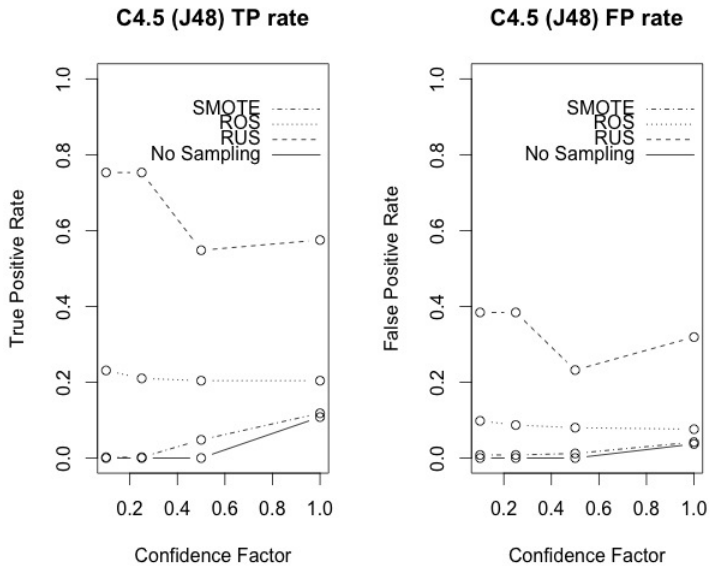
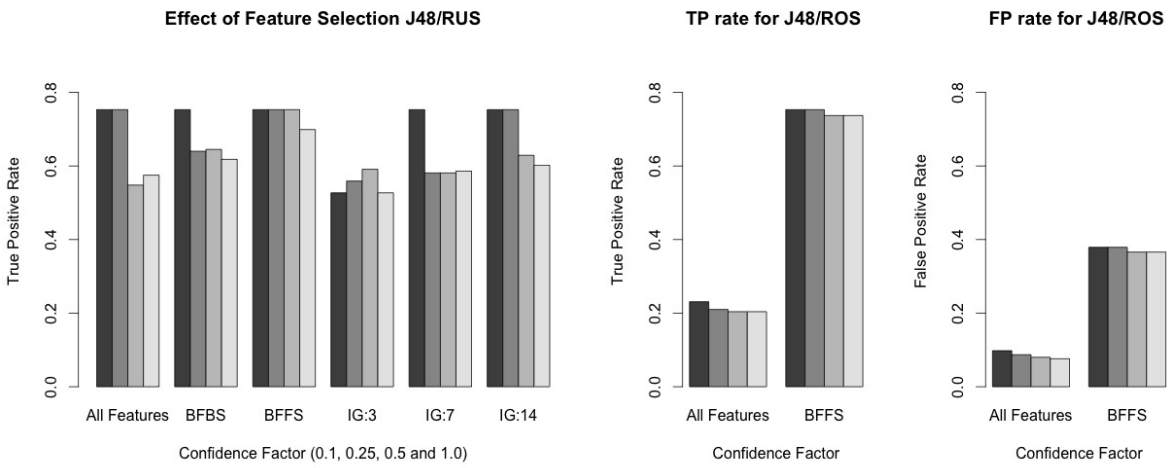


Figure 1: Effect of sampling on decision tree

Training the same classifier after balancing the data using ROS, RUS and SMOTE gave us different results. TP_{rate} on ROS and RUS were significantly higher than imbalanced dataset, whereas SMOTE was insignificantly better than imbalanced dataset. RUS gave best TP_{rate}. From RUS curve we can also see that decreasing the confidence factor (more aggressive pruning) results in higher TP_{rate}. Results for FP_{rate} were the total opposite. Training on RUS gave worst FP_{rate}, however the low FP_{rate} for SMOTE and imbalanced data are deceptive, since they are due to the classifier insistence to classify all data into the majority class.



(a) Different FS methods

(b) BFFS on ROS

Figure 2: Effect of feature selection on TP and FP rates

In fig. 2 (a) we can see that feature selection using BFFS and BFBS wrappers as well IG are not producing significant improvement on TP_{rate} when training the classifier on under-sampled dataset. However, BFFS performance is slightly better than the other methods, with more consistent results independent on pruning. Only 3 features were selected by this method (MBERARBO, PPERSAUT, APERSAUT).

Nevertheless, as seen in fig. 2 (b), BFFS significantly improves the TP_{rate} of the classifier when trained on ROS dataset. In contrast to ROS, classifiers trained SMOTE and imbalanced data did not experience any improvement with feature selection.

5.2 Multilayer Perceptron (MLP)

As in decision tree, RUS gave significantly higher TP_{rate} compared to other sampling methods for MLP with all features selected. In figure 3 we can that the best performance was achieved when having one hidden layer. Increasing or decreasing the number of hidden layers result in a model that systemically classify all records into one of the two classes. MLP layers are represented as a list of n elements, $[l_1, l_2, .., l_i, ..l_n]$, where l_i represents the number of perceptrons in the i^{th} layer.

Trying different learning rates with and without decay on RUS dataset, it was noted that decay significantly improves the TP_{rate} with small factor, while it has no significant effect on FP_{rate}.

Using wrapper method for feature selection with BFFS significantly improved the TP_{rate} of MLP when applied on RUS. Its TP_{rate} was also significantly better than Genetic Search (GS) as well as Information Gain (IG) filter methods. Nevertheless, the improvement in TP_{rate} is on the expense of FP_{rate}, which is higher compared to all other methods. It worth mentioning that we could not try BFBS with MLP due to the computational expensiveness of the two.

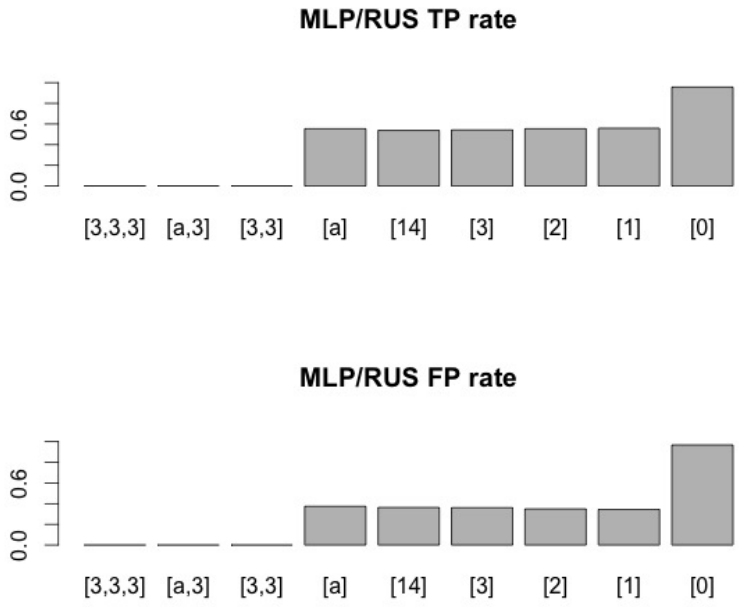


Figure 3: MLP layers are represented here as list of integers, each item represents the number of perceptrons in each layer. [0] means no hidden layers, and 'a' equals to half the summation of the number of features and number of classes.

5.3 MetaCost

We were able to produce comparable results for imbalanced dataset after wrapping J48 with MetaCost (MC). As noted by Domingos (1999), best results was achieved when misclassification cost for the minority class was set to 17.5, i.e. somewhere similar to the ratio of the majority class to the minority class in our training dataset (16.7:1). Contrary to unwrapped J48, BFFS degraded the classifier performance here.

When it comes to MLP, we failed to achieve results with MC on imbalanced data comparable to those of it the unwrapped classifier using RUS training dataset.

5.4 Best Models

Performance measures for our best four models are listed in table 1, along with model parameter, used features and sampling methods.

The TP and FP rates for the model are in the same range, however we decided to chose MLP/RUS in the end. The records in test dataset, classified as 'Caravan=1', were sorted according to the prediction performances for each model. After evaluating the top 800 records for each model, MLP/RUS had slightly more correctly predicted records than the other models. Model J48/RUS is a highly biased model, it only uses one feature/value, *'PPERSAUT=6'*, to predict class labels. The same feature is also the top node for J48/ROS and has the highest weight in MLP/RUS, yet the latter two model use more features to decide class labels.

It worth mentioning that despite the fact the four models have high TP_{rate}, we failed to lower the FP_{rate} beyond 0.3. Having highly imbalanced classes, means that for such

Model Name	J48/RUS	J48/ROS	MLP/RUS	MC/J48
Classifier	C4.5 (J48)	C4.5 (J48)	MLP	MetaCost/J48
Sampling	RUS	ROS	RUS	None
Feature Selection	Wrapper BFFS	Wrapper BFFS	Wrapper BFFS	None
Features ID's	24, 47, 68	24, 47, 68	18, 23, 34, 47, 52, 54, 59, 82	All
Parameters	CF=0.25	Un-pruned	1 Hidden layer 3 Neurons	CF=0.2 Cost=[0 1; 17.5 0]
Training TP rate	0.753	0.747	0.765	0.654
Training FP rate	0.405	0.368	0.313	0.535
Test TP rate	0.753	0.737	0.71	0.731
Test FP rate	0.379	0.366	0.415	0.374

Table 1: Best Models and their TP and FP rates

values of TP and FP rates, we get much more records from the majority class misclassified as members of the minority class than those correctly classified into the minority class.

Other than 'PPERSAUT', features with id's 24 and 68 were used by the un-pruned tree to decide the correct class label. For MLP, features with id's 18, 59, 23 came after 'PPERSAUT' with highest weights.

6 Unlabelled dataset prediction

The final predicted values for the unlabelled dataset were calculated using the model MLP/RUS, shown in table 1. Due to the high false positive rates for our models, we have more than 800 records classified as 'CARAVAN=1'. Hence, we kept the top 800 records, with highest prediction probabilities, and converted the rest to the other class, 'CARAVAN=0'.

7 Conclusion

To main issues were identified in the Caravan dataset. The first is the imbalance of the class label, while the second issue is the overlap of the two classes. In our experiments, it was proven that class imbalance can be tackled by either re-sampling the training date or using a classifier with higher cost for misclassifying the minority class. In particular, RUS and ROS were proven to be more effective compared to more complex sampling algorithms such as SMOTE. On the other hand, the class overlap issue was harder to tackle. We have seen that any trials to improve the minority class' TP_{rate} results in a more biased classifier and increase in FP_{rate}. Feature selection (FS) significantly improved the TP_{rate} for J48 with ROS data, while its effect was less significant on RUS. Wrapper feature selection methods with Best First Forward Selection were more effective compared to Information Gain as well as wrapper methods with Genetic Search. Backward selection takes long time to reach optimum set with computationally expensive algorithms such as MLP, whereas they did not improve the performance of J4.8 much compared to forward selection methods.

Caravan Car Policy Insurance
Description Report

1 Introduction

The term 'permission marketing' was coined by Godin (1999). According to that concept, customers need only to receive marketing campaigns they are interested in. Therefore, marketing departments are required today to identify customers needs and preferences before tailoring the email campaigns being sent to them.

In this report we try to identify customers interested in caravan insurance policies, based on records of your company's existing customers. In the following section we identify and explain the socio-demographic characteristics and purchase history for those customers who are most likely to purchase caravan insurance.

2 Caravan insurance customers

Based on our findings, we identified customers contribution car policies to be the most important aspect in identifying potential customers. As you can see in figure 4, a simple rule matching those with contributions between \$1,000 and \$4,999 is enough to identify about 75 % of the potential customers. However, 35% of non-potential customers might be identified with the same rule too. On the other hand, those with contributions between \$5,000 and \$9,999, as well as \$500 and \$999 are most likely not to be interested in the insurance policy.

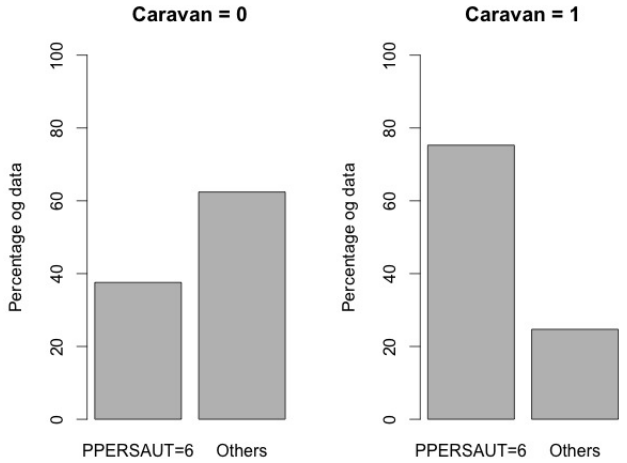


Figure 4: The relation between Caravan insurance purchase and Contribution car policies (Persaut)

In addition to contribution car policies, we have found that those living in areas with very low percentages of skilled labourers form potential customers, while inhabitants of areas with skill labourers in ranges of 24-36% and 50-62% are less likely to purchase the

policy. Similarly, inhabitants of areas with low level of unskilled workers are more likely to purchase than those living in areas with higher percentages of unskilled workers. Figure 5 displays those findings.

Figure 5: Distribution of of areas with different levels of skilled and unskilled workers among Caravan policy potential buyers

In addition to this, the education level of the customers play role in decided potential customers. Areas where those with low-level of education not exceeding 23% are good signal for potential customers, same for areas with 63-88% of low-educated personnel. Areas in between, with mixed education levels are - to a less extent - less likely to have potential buyers.

People who contribute more in fire policies, especially in ranges of $100 to $499 are potential customers, while those with low contribution, 1 to 99 are more likely not to buy. Areas with 100% car owners, contain more potential customers compared to other areas. Nevertheless, surprisingly, areas with 1 to 10% non-car-owners are less likely to purchase the policy compared to areas with 63 to 75% non-car-owners. Additionally, people with more than one car policy are more likely to buy caravan insurance.

3 Conclusion

In conclusion, caravan car policy insurgence customers are more likely those who have more than one car, with high insurance values on them, in the range of $1,000 and $4,999. They also spend on fire policies, which may reflect caravan owners tendency to go for camping or BBQ parties. They also seem to live in similar areas, which means that the company might also spend some of their marketing budget on outdoor advertisements in those areas, or on advertisements in local newspapers.

Appendix A

J48 Model

Below is J48 model built on RUS training dataset, and only 3 features (MBERARBO, PPERSAUT and APERSAUT). The confidence factor for the tree was set to 0.25.

- PPERSAUT = 0: 0 (114.0/35.0)
- PPERSAUT = 1: 0 (0.0)
- PPERSAUT = 2: 0 (0.0)
- PPERSAUT = 3: 0 (0.0)
- PPERSAUT = 4: 0 (0.0)
- PPERSAUT = 5: 0 (21.0/5.0)
- PPERSAUT = 6: 1 (186.0/64.0)
- PPERSAUT = 7: 0 (4.0)
- PPERSAUT = 8: 0 (0.0)
- PPERSAUT = 9: 0 (0.0)

The next model is built on ROS training dataset, and only 3 features (MBERARBO, PPERSAUT and APERSAUT). The confidence factor for the tree was set to 1.0.

- PPERSAUT <= 5: 0 (2265.0/600.0)
- PPERSAUT > 5
  - PPERSAUT <= 6
    - MBERARBO = 0: 1 (543.0/168.0)
    - MBERARBO = 1: 1 (686.0/176.0)
    - MBERARBO = 2: 1 (618.0/243.0)
    - MBERARBO = 3: 1 (412.0/172.0)
    - MBERARBO = 4: 1 (329.0/149.0)
    - MBERARBO = 5: 1 (169.0/64.0)
    - MBERARBO = 6
      - APERSAUT <= 1: 0 (35.0/15.0)
      - APERSAUT > 1: 1 (17.0/2.0)
    - MBERARBO = 7: 0 (8.0)
    - MBERARBO = 8: 1 (16.0/1.0)
    - MBERARBO = 9: 0 (3.0)
  - PPERSAUT > 6: 0 (32.0)

MLP Model

Below is MLP model built on RUS training dataset, and only 8 feature (MOPLLAAG, MBERARBG, MAUT0, PPERSAUT, PTRACTOR, PBROM, PBRAND and APLEZIER). The model has 1 hidden layer containing only 1 neuron. This is not the best model, yet since its a simple model and its results are not very far from the our best model, it is shown here for the sake of clarity.

Sigmoid Node 0
Inputs Weights
Threshold 1.7727134862113922
Node 2 -3.140794910819783
Sigmoid Node 1
Inputs Weights
Threshold -1.7732040177592665
Node 2 3.141541540893237
Sigmoid Node 2
Inputs Weights
Threshold -0.13614547437911242
Attrib MOPLLAAG=0 0.2577195392357572
Attrib **MOPLLAAG=1 1.3855123423874687**
Attrib **MOPLLAAG=2 0.9967060472728216**
Attrib MOPLLAAG=3 0.5800265533008822
Attrib MOPLLAAG=4 0.36869887270353036
Attrib MOPLLAAG=5 -0.4638805229516628
Attrib **MOPLLAAG=6 -1.6303603887880582**
Attrib MOPLLAAG=7 -0.7484155793079491
Attrib MOPLLAAG=8 0.19352201285123985
Attrib MOPLLAAG=9 0.21283213264648868
Attrib **MBERARBG=0 -1.06379507976582**
Attrib **MBERARBG=1 1.4889115066812142**
Attrib MBERARBG=2 0.7083385520027917
Attrib MBERARBG=3 0.07149597331443328
Attrib MBERARBG=4 -0.8359836290726058
Attrib MBERARBG=5 0.5663859153546645
Attrib **MBERARBG=6 -1.2424719515432519**
Attrib MBERARBG=7 0.7464704950338501
Attrib MBERARBG=8 -0.015433953910609133
Attrib MBERARBG=9 0.5315068256613428
Attrib MAUT0=0 0.749390362661621
Attrib MAUT0=1 -0.5737252142019146
Attrib MAUT0=2 0.1408871772380677
Attrib MAUT0=3 0.3185585768476083
Attrib MAUT0=4 -0.2636022471831909
Attrib MAUT0=5 -0.16401846966714176
Attrib MAUT0=6 0.5834728310389298
Attrib MAUT0=7 0.03515421783528215
Attrib MAUT0=8 0.12153961004871314
Attrib MAUT0=9 0.010451906671889534
Attrib PPERSAUT=0 -0.34328351120114925

Attrib PPERSAUT=1 -0.004993221975007622
Attrib PPERSAUT=2 -0.015543921693123387
Attrib PPERSAUT=3 -0.023060522140987718
Attrib PPERSAUT=4 0.024514716787820107
Attrib **PPERSAUT=5 -1.187785437846639**
Attrib **PPERSAUT=6 1.9716937259631953**
Attrib **PPERSAUT=7 -0.16723211565634444**
Attrib PPERSAUT=8 0.018769367453613864
Attrib PPERSAUT=9 0.04167027394342901
Attrib PTRACTOR=0 0.3379478638883758
Attrib PTRACTOR=1 -0.030484224501287616
Attrib PTRACTOR=2 0.007600658839037447
Attrib PTRACTOR=3 -0.24670370931064972
Attrib PTRACTOR=4 -0.05992793331095452
Attrib PTRACTOR=5 0.19427887242379277
Attrib PTRACTOR=6 -0.040269856081275174
Attrib PTRACTOR=7 0.03196935615755725
Attrib PTRACTOR=8 0.016992161115539056
Attrib PTRACTOR=9 -0.007111219697094787
Attrib PBROM=0 0.7238440215559447
Attrib PBROM=1 0.032959342292543437
Attrib PBROM=2 0.15405031630730961
Attrib PBROM=3 -0.7282247861883119
Attrib PBROM=4 0.009738731578471348
Attrib PBROM=5 -0.01256168678854254
Attrib PBROM=6 -0.04983593732702583
Attrib PBROM=7 0.018950842285448383
Attrib PBROM=8 0.021374736952674153
Attrib PBROM=9 -0.04307534550907556
Attrib PBRAND=0 0.09104292678606034
Attrib PBRAND=1 -0.6261217835807286
Attrib **PBRAND=2 -1.1867976752629994**
Attrib **PBRAND=3 0.8686530365613083**
Attrib **PBRAND=4 1.5536716787824685**
Attrib PBRAND=5 0.03629740453021645
Attrib PBRAND=6 -0.043170354944413965
Attrib PBRAND=7 0.0020896327541071943
Attrib PBRAND=8 0.04563085926419766
Attrib PBRAND=9 0.019301884420390963
Attrib APLEZIER 0.5172836649632223
Class 0 Input
Node 0
Class 1 Input
Node 1

References

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *arXiv preprint arXiv:1106.1813*, 2011.

Toma Curk, Janez Demar, Qikai Xu, Gregor Leban, Uro Petrovi, Ivan Bratko, Gad Shaulsky, and Bla Zupan. Microarray data mining with visual programming. *Bioinformatics*, 21:396–398, February 2005. ISSN 1367-4803.

E DeRouin, J Brown, H Beck, L Fausett, and M Schneider. Neural network training on unequally represented classes. *Intelligent engineering systems through artificial neural networks*, pages 135–145, 1991.

Pedro Domingos. Metacost: a general method for making classifiers cost-sensitive. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 155–164. ACM, 1999.

Kehan Gao, Taghi Khoshgoftaar, and Jason Van Hulse. An evaluation of sampling on filter-based feature selection methods. In *Proceedings of the 23rd International Florida Artificial Intelligence Research Society Conference*, pages 416–421, 2010.

Seth Godin. *Permission marketing: Turning strangers into friends and friends into customers*. Simon & Schuster, 1999.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

MI Lopez, JM Luna, C Romero, and S Ventura. Classification via clustering for predicting final marks based on student participation in forums. In *Proceedings of the 5th International Conference on Educatiional Data Mining*, pages 148–151, 2012.

Ronaldo C Prati, Gustavo EAPA Batista, and Maria Carolina Monard. Class imbalances versus class overlapping: An analysis of a learning system behavior. In *MICAI 2004: Advances in Artificial Intelligence*, pages 312–321. Springer, 2004.