Impact of different oversampling techniques in the random forest classification accuracy of the adult data set.

Juan Florez. R00184264

8.12.2019

Abstract:

The adult data set (https://archive.ics.uci.edu/ml/datasets/adult) has been around since 1994, and has been used as a classification problem, linking different data (i.e Education) to income level.
If we use this dataset to classify income level (i.e. over or under 50k), the data will be Highly unbalanced, giving around 2 out of 3 hits just by classifying everyone under 50K.

In this paper, I assess the effect that different oversampling techniques have in the default random forest classification using 300 estimators.
In order to isolate the real effect of the different balancing techniques, I did a basic data preprocessing and used the confusion matrix to monitor the impact of the balancing techniques in the model.

Random Forest method was selected randomly over SVC, KNN, and Neuroteworks, as the pourpose of this paper it to understand the impact of different class balancing techniques over one model. (trying when possible to use default values of scikit learn), and the initial run of the model gave a reasonably good accuracy in a good enough runtime.

Out of the oversampling techniques applied with default parameters, the ADASYN (Adaptive synthetic) approach, proved to improve the accuracy of the model as measured by the 'false negative' portion of the confusion matrix.

Introduction:

This paper is a hectic attempt to show my understanding of the basic process to build, calibrate and evaluate a model for machine learning over a specific dataset. (under 15 hours of work).
The first version, (hopefully delivered on time) will show the basic preprocessing of the data, the run of the models over that data and the effects that different class balancing approaches on the training data have on the overall accuracy of the models.
I will attempt to deliver a version 2 or even a 3 if I have the time after my other courses and my year's close at work, and I see that I can significantly improve my grade.
The evaluation of the models in the first version is done using the confusion matrix, particularly focussing on reducing the number of False Negatives (i.e. records that the model wrongly shows as not making more than 50k), due to time pressure, no cross-validation nor hyperparameter optimization will be performed in this first version.
The adult data set has been precleaned of null values and its data quality is good enough to allow me to isolate the effects of data balancing.

Research:

Please refear to model_v2.ipynb (Jupyter Notebook).
This code is in github at
https://github.com/juanflorezVe/PML_A2/blob/master/model_v2.ipynb

The dataset:

The data set has 15 columns, in order to increase my speed, and remove personal bias, I renamed the column from ‘a’ to ‘o’, and chose ‘o’
The one that shows either <=50k or > 50k. as the class. The dataset was downloaded from https://archive.ics.uci.edu/ml/datasets/adult
And it has 48842 instances.

Pre-processing:

Null values:

This particular dataset has been cleaned out of null values, and this is confirmed by running data.info() and data.isnull().sum()

Features encoding:

I ran a basic encoding in all the columns to ensure work with numbers. for simplicity, and as I want to isolate the effects of the balancing techniques, I did not run one hot encoding, as I do not know if it will affect my results. (see future work recommendations)

Divide data into training, validation, and test:

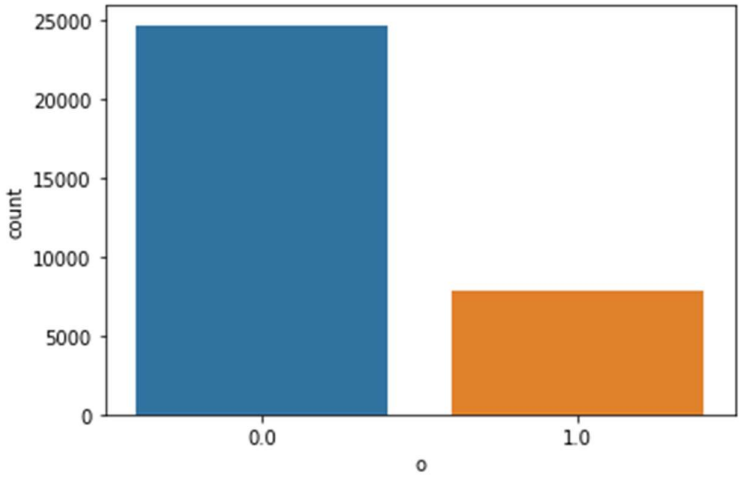
I took 20 percent of the data to learn and train the model.

Scaled data:

Using the StandardScaler, I scaled the features in the traing and test sets (notice I did not touch the class data). so there are no features obscuring other just because of big numbers.

Confirming unbalanced data:

Fig 1. shows the relative amount between classes in the TESTING set. (>50k, is class '1')
Learning the models and testing their accuracy



:

Fig 1. Clearly unbalanced training data set

In the first ran, each model was fit 1 time using train data and evaluated 1 time using test data; without any hyperparameter optimization.

- SVC
- Random Forest (300 runs)
- K Nearest Neighbours
- MLP Classifier.

Applying different balancing techniques:

First of all, it is worth noting that the balancing techniques are applied to the TRAINING DATA, meaning, the test data, and indeed the "real data set" that the model is aiming to predict should not be "contaminated" with synthetic data.

In the interest of time, I skipped random over er sampling. which is the technique to randomly add instances in order to arrive to a training set that has a balanced amount of instances of different classes.
So, I applied 3 techniques for oversampling:

- SMOTE
- ADASYN
- Borderline SMOTE
- K Means SMOTE

I created a training set out of each of the technique, and learn a Random Forest Classifier with 300 runs.
Then I plotted the confusion matrix and noted the FN value.

Bellow there is a short description of the oversampling techniques.

SMOTE:

SMOTE will create instances of the minority class in areas "that are expected". Rather than creating new instances randomly, it creates instances "in the neighbourhood or along connecting lines between similar instances.

ADASYN:
Adaptive Synthetic Sampling Method for Imbalanced Data

"The major difference between SMOTE and ADASYN is the difference in the generation of synthetic sample points for minority data points. In ADASYN, we consider a density distribution r? which thereby decides the number of synthetic samples to be generated for a particular point, whereas in SMOTE, there is a uniform weight for all minority points.".[2]

Borderline SMOTE:
“Borderline-SMOTE1 and borderline-SMOTE2, in which only the minority examples near the borderline are over-sampled.”[3].

K Means SMOTE:[4]
“K-Means SMOTE is an oversampling method for class-imbalanced data. It aids classification by generating minority class samples in safe and crucial areas of the input space. The method avoids the generation of noise and effectively overcomes imbalances between and within classes.”[5]

Conclusion

We can see a reduction of circa 90% on false negatives (from 486 to 48) Of ADASYN and Borderline SMOTE over K Means SOTE.
The reason, I believe, is that due to the nature of the dataset, the features may be heavily related (opposite to an assumption of independence expected by a basic Bayesian approach). so, by increasing the synthetic data in the areas of mayor density, the ADASYN model presents a clear advantage over the SMOTE technique. Anyway future work remains to be done in order to really confirm the results.

Future work recommendations:

Things that still can be done to evaluate the results:
run cross folding validation
run hyper parameter optimization
run feature selection

run one hot encoding in the most relevant features, ie, education, job type, family membership.
run outlawyer detection.
run under sampling
run each model a number of times and evaluate the results by the median of all the runs.

Other oversampling techniques:

See https://imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.over_sampling.BorderlineSMOTE.html#imblearn.over_sampling.BorderlineSMOTE

SMOTENC, SVMSMOTE

REFERENCES

[1] N. V. Chawla, K. W. Bowyer, L. O.Hall, W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” Journal of artificial intelligence research, 321-357, 2002.

[2] ADASYN: Adaptive synthetic sampling approach for imbalanced learning

H He, Y Bai, EA Garcia, S Li - 2008 IEEE International Joint Conference on Neural …, 2008

[3] Han H., Wang WY., Mao BH. (2005) Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In: Huang DS., Zhang XP., Huang GB. (eds) Advances in Intelligent Computing. ICIC 2005. Lecture Notes in Computer Science, vol 3644. Springer, Berlin, Heidelberg

[4] Felix Last, Georgios Douzas, Fernando Bacao, “Oversampling for Imbalanced Learning Based on K-Means and SMOTE” https://arxiv.org/abs/1711.00837

[5] https://pypi.org/project/kmeans-smote/