Advanced Machine Learning
TP

Learning from Unbalanced Data

Master Machine Learning Data Mining

Submitted by

DEVEAUX Anthony
UPENDRA Nisal

University Jean Monnet
ST Etienne, France

November 2018

0.1 Introduction to the Problem

The dataset given in this assignment is a set of credit card transactions. Since credit card transaction fraud data is highly unbalanced, here we propose two sampling methods to transform the data to achieve a better result. Unbalanced data occurs where there is a significant differences in the occurences of different the classes of data that is being predicted.[2] There are two different methods for handling an imbalanced dataset in sampling, which is the most common method used. Those are under sampling and oversampling. In undersampling, data in given to the model is changed by dropping common classes, while oversampling will duplicate rare classes.

For this study, we have split the training data, to train a model and test its accuracy and then classified the given test data using that model. We used the imblearn library for python to run the algorithms.

0.2 Oversampling using SMOTE

SMOTE creates new instances for the under represented class between the existing under represented class instances. This is an oversampling process. After these minority instances are created, it eliminates the imbalance between the over represented and under represented classes [3]. In our study, we used SMOTE to oversample the fraudulent detections such that it equalled the number of transactions labelled as legitimate.
Afterwards, we trained an KNN classifier on the oversampled data and then analyzed the accuracy. The results can be seen below (The relevant code for this results is at smote.py).

```
counts of labels before oversampling:  [0. 1.] [69855   144]
counts of labels after oversampling:  [0. 1.] [69855 69855]
Accuracy on split training data: 0.9159702281461164
Recall metric - split train data: 100.0%
Accuracy on split test data: 0.9057
Recall metric - split test data: 40.57971014492754%
```

As seen from the results, after oversampling, the recall[1] is at 100% with 91.5% accuracy on the split training data (train.csv). After using this model on the split test data from the training, the accuracy goes down to 90.5% with a recall of 40.5%. The confusion matrices for the split training data is shown below. After this we used this trained model on the test data provided (test.csv) and the predicted labels and their probabilities have been saved to

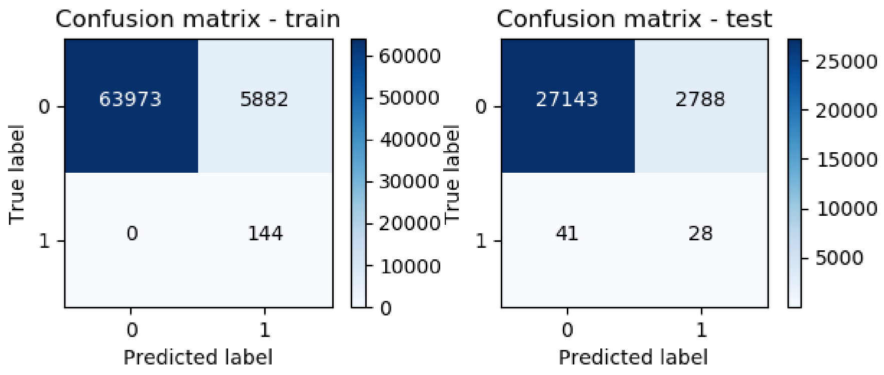

Figure 1: Oversampling using SMOTE on split training data

the smote_test_data_predict.csv and smote_test_data_predict_probabilities.csv files attached to this report.

0.3 Undersampling using TomeKLinks

Tomeklinks is an undersampling method that works by removing the instances of the over represented class. The algorithm works by pairing with an instance of the opposite class nearby and then removing the over represented class instance of each pair. This increases the space between the two classes, making the classification less biased.
In our study, we used Tomeklinks to undersample the fraudulent detections such that it equalled the number of transactions labelled as legitimate.
Afterwards, we trained a KNN classifier on the undersampled data and then analyzed the accuracy. The results can be seen below (The relevant code for this results is at tomklinks.py).

```
counts of labels before undersampling:  [0. 1.] [69855   144]
counts of labels after undersampling:  [0. 1.] [69798   144]
Accuracy on split training data: 0.9979428277546822
Recall metric - split train data: 0.0%
```

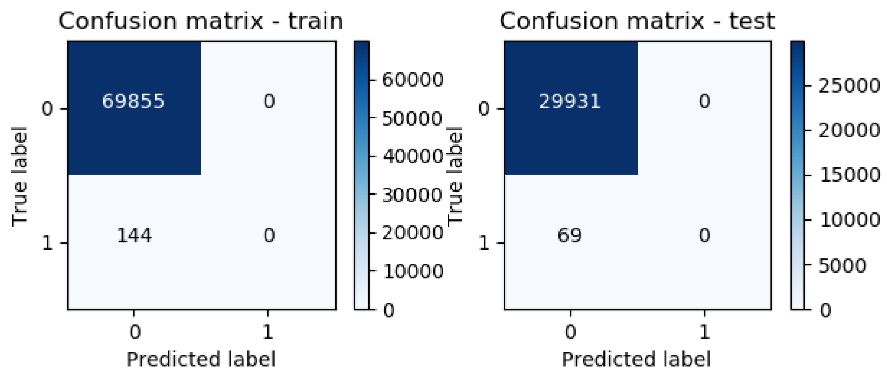

Figure 2: Undersampling using TomeKLinks on split training data

```
Accuracy on split test data: 0.9977
Recall metric - split test data: 0.0%
```

As seen from the results, after undersampling, the accuracy is at 99.7% on the split training data (train.csv). After using this model on the split test data from the training, the accuracy remains roughly the same. The confusion matrices for the split training data is shown below. After this we used this trained model on the test data provided (test.csv) and the predicted labels and their probabilities have been saved to the tomeklinks_test_data_predict.csv and tomeklinks_test_data_predict_probabilities.csv files attached to this report.

0.4 Conclusion

Allthough the Tomklinks algorithm gives better results, the confusion matrix shows that it has not correctly identified any of the fraudulent transactions. Therefor, it is not suitable for this kind of classification. This is due the fact that this method only removes a very small amount of instances of the over represented class and thus it is not enough to improve the accuracy in detecting fraud. We believe it is more suited to balance datasets that are not

extremely imbalanced.
On the other hand, even though SMOTE gives lower accuracy it is more suited to this kind of dataset, as it creates an equal number of instances for both classes. This is very important as the objective here is to identify a fraudulent transaction, and not the opposite.

References

[1] URL: https://www.kaggle.com/qianchao/smote-with-imbalance-data.

[2] *Oversampling and undersampling in data analysis.* July 2018. URL: https://en.wikipedia.org/wiki/Oversampling_and_undersampling_in_data_analysis.

[3] *SMOTE explained for noobs - Synthetic Minority Over-sampling TEchnique line by line.* URL: http://rikunert.com/SMOTE_explained.