IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Heuristic Approach of Over-Sampling and Under-Sampling in Fraud Detection

[1]Rahul Pandya, [2]Shivang S. Manhas, [3]Arieyshma Chowhan, [4]Ronit Gandhi, [5]Aditya Umalkar

[1,2] Department of Electronics and Telecommunications, Mumbai University, Mumbai, India
[3,5,6] Department of Electronics, Mumbai University, Mumbai, India
[4] Department of Information Technology, Mumbai University, Mumbai, India

Abstract: The Sorting of Imbalanced Data Sets has developed a great deal of cognizance since the evolution of Machine Learning, which is substantial in the worlds of Business, Industries, and Scientific Research. In this research paper, we present an overview related to the problem in Imbalanced Data Sets and explain certain techniques such as sampling (Near miss, SMOTE) provides some evaluation metrics, which are used on imbalanced data sets, re-iterates some captivating points drawn from the latest and known research papers related to Imbalanced classification problem. This research focuses on implementing certain techniques to derive a clear image of the classification problem and to present a brief review of existing solutions for such problems. Here, we examine the binary classification problem on Imbalanced data sets.

Index *Fraud Detection, Imbalaced data, SMOTE, Sampling, Anamoly Detection, Near Miss*

I. INTRODUCTION

In AI and Information Science, we frequently run over a term called Imbalanced Information Conveyance, Disorganized or random chunks of scattered data, which is the usual occurrence when perceptions in one of the classes are higher than different classes or the other way around. In real applications, Data sets with imbalanced class distributions are quite frequent. For example, in credit card fraud detection, legal transactions outnumber the fraud transactions or during an automated inspection of the products in a coffee-mug industry would evaluate that the defective products are significantly less compared to the non-defective ones. This distribution when compared to the classifier tends to have a biased image for the majority class. The reason that this idea might be occurring is due to the fact, while performing the engineers are more concerned to have accuracy for the result and not on the performing stage, that is, they are accuracy driven.
Considering the same case of fraud detection for credit card users, the accuracy for no fraudulent detection might achieve the 97% benchmark but despite this achievement in such cases, the rare class has much greater importance than the majority class. There are primarily two methodologies that are utilized for dealing with imbalanced class appropriation, one is SMOTE and the other is Near Miss Algorithm. There are more ways of drawing closer imbalanced datasets, for example, Cost-sensitive learning or Anomaly detection and so on.
As AI calculations will generally increment exactness by minimizing the error, they are not efficient for varied class distributions.

II. LITERATURE REVIEW

An imbalanced classification is an illustration of an issue where the dissemination of models across the realized classes is one-sided or biased. The dispersion can fluctuate from a slight predisposition to a serious unevenness where there is one model in the minority class for hundreds, thousands, or millions of models in the majority class or classes. Imbalanced classifications represent a test for prescient displaying as the greater part of the AI calculations utilized for classification algorithms were planned around the assumption of an equivalent number of models for each class. This outcomes in models that have poor performance and lower preciseness, explicitly for the minority class. This event is an issue on the grounds that normally, the minority class is more significant and hence the issue is more delicate to classification blunders for the minority class compared to majority class.

III. SYSTEM BLUEPRINT

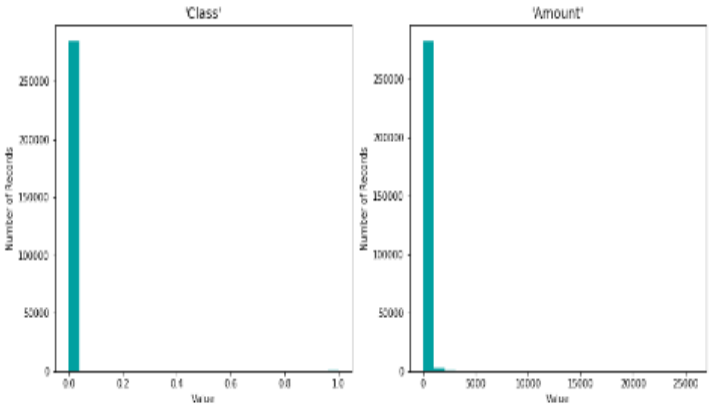
Imbalanced Data cleaning requires practically irrelevant costing. Our framework includes utilizing of a scripting language like Python or R alongside the establishment of a simple to utilize IDE. Here we have utilized Jupyter Notebook which comes worked in with Anaconda [1]. Visual Studio and Pycharm could likewise be used. System processors having i3 intel processors or more forms with a fundamental realistic card are adequate to run the libraries and ML calculations. We propose involving Jupyter Notebook as it empowers the client to download the entire bundle of libraries and programming as a group and other required ones can be effectively downloaded utilizing a straightforward and efficient command known as pip.

The following are the essential prerequisites to perform opinion analysis:

1. Pandas - it is utilized to produce the CSV for the imbalanced data.
2. Numpy bundle - it is utilized for mathematical computations in python library.
3. Seaborn - it is a library utilized for factual information representation
4. Matplotlib: it is library used for graphical representation and visualization of data

IV. DATA PREPROCESSING

The dataset utilized in this research is the credit card dataset, which is accessible openly[2]. This dataset has massive 284807 records and very distinct 31 features. The dataset contains numerical values of the PCA exchanges[3]. Because of privacy issues, the primary featured highlights have been renamed. These components are the fundamental parts gotten with PCA, the primary features that have not been changed with PCA are 'Time' and 'Amount'. Here, 'Time' contains the seconds postponed between each transaction and the fundamental transaction in the dataset, we can drop this part as a period contrast between the first and the following transaction has no association with the client being a fake[4]. The feature 'Amount' is the exchange Sum. This should be made to implement the cost-effective function for the training dataset. The element 'Class' is the response variable assigned with the values 1 in the event of the client being fraud and 0 in any case. Subsequently, we will break down these features alone. Here, the figure of envisioned information is shown.

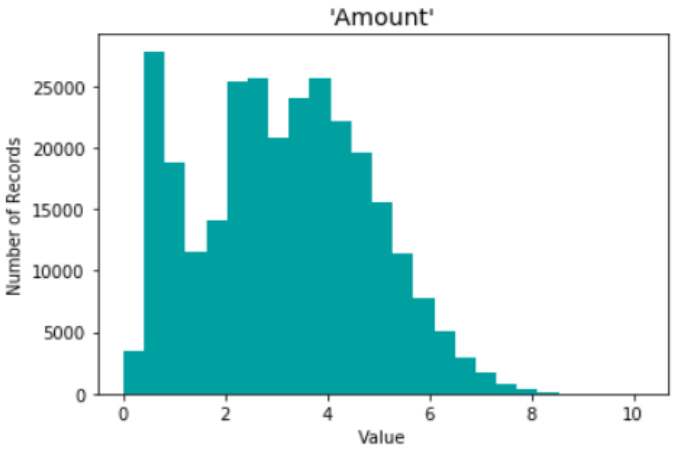


Features of Dataset

The information pre-processing part is acted in the following manner.

Changing of features that are Skewed Continuous:

For profoundly skewed distributions, considering present realities, it is a normal practice to apply a logarithmic change on the data, so the extremely enormous and tiny features don't adversely influence the exhibition of a learning calculation. Utilizing a logarithmic change fundamentally lessens the scope of values brought about by outliers. Care should be taken while applying this change nonetheless: The logarithm of 0 is indistinct, so we should interpret the values just barely over 0 to effectively apply the logarithm.



Distribution of Data in Graph

We can see that now the data distribution is improved. Now, we need to normalize within a range similar to other features to feed it into SVM classifier. After this, we are normalizing the data on the scale of 0 to 1.

	Time	V1	V2	V3	V4	V5
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321

Features Renamed

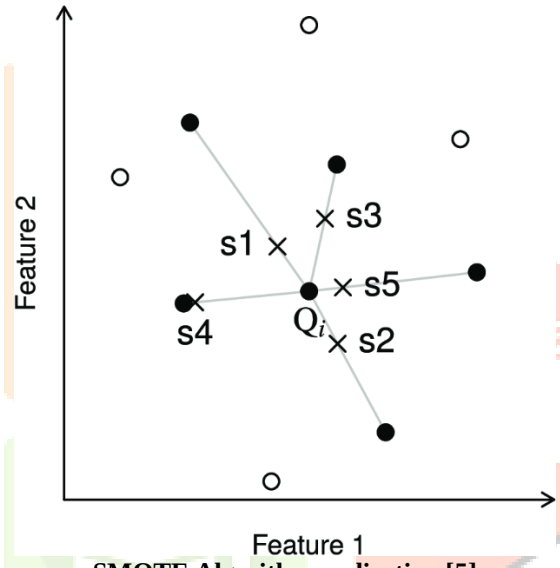
For this study secondary data has been collected. From the website of KSE the monthly stock prices for the sample firms are obtained from Jan 2010 to Dec 2014. And from the website of SBP the data for the macroeconomic variables are collected for the period of five years. The time series monthly data is collected on stock prices for sample firmsand relative macroeconomic variables for the period of 5 years. The data collection period is ranging from January 2010 to Dec 2014. Monthly prices of KSE - 100 Index is taken from yahoo finance.

V. ALGORITHMIC OVERVIEW

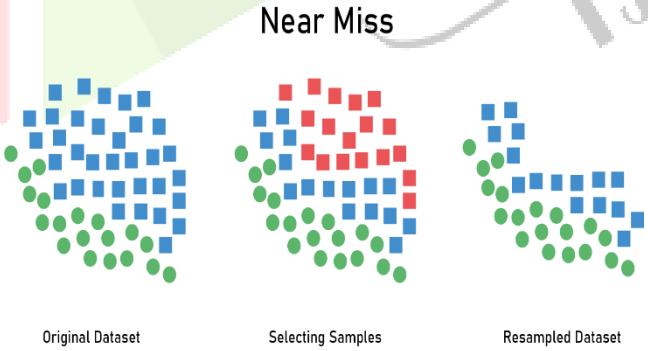
SMOTE (synthetic minority oversampling technique) is one of the most usually utilized oversampling techniques to tackle the irregularity issue.

It aims to adjust class distribution by arbitrarily expanding minority class models by imitating them. SMOTE incorporates new minority cases among existing minority examples. It produces virtual preparation records by direct interjection for the minority class. These records are produced by arbitrarily choosing at least one of the k-nearest neighbours for every model in the minority class. Post the application of the oversampling on the data, the dataset is re-set. After this step various classification models can be iterated on the newly prepared data.

Smote Algorithm synthesizes new specimens between existing minority instances. After that, it imagines new instances over the boundary of minority instances. This move creates instances to match the sample's quantity of majority classes.



SMOTE Algorithm realization [5]



Methodology of Near Miss Algorithm

Near Miss Algorithm is a widely known under-sampling technique. The end goal of this algorithm is to discard the random samples from majority classes. Unlike SMOTE, the Near miss reduces specimen from majority classes instead of increasing in minority classes. This algorithm will in general eliminate cases of majority part classes when examples of two classes that are close. To forestall the loss of data, it takes out the closest neighbour. To execute this, it first tracks down the distance between every one of the occurrences of each class. After that, it selects the n specimen of the majority having the shortest distance to minority classes. The near miss finds the n closest instances in the following way. It chooses the samples of majority class where the distance to minority class is the smallest. It also selects whose farthest distance to minority class is the smallest.

VI. MODEL

For a research purpose, we use a Credit Card Fraud Detection dataset. It composes of the transactions made via credit cards in September 2013 by European cardholders. Here, this dataset presents transaction, where we have about 492 fraudulent transactions out of 284,807 total. It's a highly imbalanced dataset, with the positive class frauds account for 0.172% of all transactions.[6] We explore the data first, which has 284807 records with 31 features. The main features are Time, Class, and Amount. We drop the Time feature as the time difference does not have any relation with fraud transactions. We focus on the Class feature after the normalization of the Amount. Splitting the data in an 80:20 ratio and fit them as train and test data sets respectively.

```
#print Classification report
print(classification_report(y_test, predictions))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	85296
1	0.88	0.62	0.73	147
accuracy			1.00	85443
macro avg	0.94	0.81	0.86	85443
weighted avg	1.00	1.00	1.00	85443

Classification Outcomes

Here, the recall of the minority class is less, and the model is biased more towards majority class. Hence, this is not a good model for use. So, we use SMOTE algorithm to oversample the minority class and make it equal to the majority class so both the classes have equal values. After prediction, we get the classification report with reduced accuracy value but improved recall value. Making the use of Near-Miss technique to under-sample the majority class values. We get similar or almost equal accuracy results after applying Near Miss technique. Here we have 0.98 accuracy value. In similar ways on an implementation of hybrid models which include use of cost functions with decision tree classifiers, Random Forest or GNB, we can tackle the problem. Preferably, used Smote and Near Miss as they require generally lesser training time with good accuracies. For better understanding, we should make use of all and then choose the best one after comparison. Decision trees work best as classifiers more often as they work by learning a hierarchy of models therefore classes are well addressed.

```
results_df = pd.DataFrame(results)
display(results_df)
#print "Columns are ", results_df.columns
```

	oversampled	undersampled	SMOTE	Decision_Tree	Random_forest	GNB
fbeta	0.926651	0.904762	0.924967	0.773481	0.788352	0.078214
precision	0.975387	0.956835	0.972827	0.823529	0.956897	0.063764
pred_time	0.020426	0.001995	0.013688	0.023349	0.533566	0.080506
recall	0.915218	0.892617	0.913729	0.761905	0.755102	0.836735
resample_time	NaN	NaN	0.779217	NaN	NaN	NaN
train_time	22.595821	16.109726	2.111507	9.562523	70.838898	0.124972

Results

Above all, we get almost better accuracy values, recall values and AUC with high precision. Higher the values, prediction accuracy is higher for detecting fraudulent transactions.

VII. OUTCOMES

Based on the data model which can be used for imbalanced datasets we can get these outcomes. K-fold cross-validation is a methodology utilized during preparing the AI calculations in which the dataset is sampled again during the training period. This approach parts the datasets into k various gatherings following which one of these is considered as testing information and the remaining are considered as training information. This approach gives the appropriate requirements to the imbalanced data. Ensembling re-test datasets incorporates re-examining the dataset so that the data which are scant or uncommon will be oversampled. Through this approach the general information can be adjusted, and the outcomes accomplished by the algorithm will be impartial. SMOTE and Near-Miss Algorithms are of this approach but can be utilized in a hybrid approach as well as discussed previously. Reducing the weight of the majority attributes and increment for the minority ones is likewise a significant methodology. Every attribute has a dedicated weight assigned to it. This enables to balance the data according to the respective relations with a single factor of consideration. The attributes with the maximum presence will have larger values of weights assigned to it and vice-versa. This gives in further developing the exactness scores and AUC successfully.

VIII. CONCLUSION

This research gives the insights about the various classification methods to be applied on imbalanced datasets. The focal thought of research reviewed making use of certain sampling techniques by making use of algorithms and performing them through Python as the programming language.
The majorly covered 5 unique strategies for managing imbalanced datasets:

1. Change the presentation metric
2. Oversampling minority class
3. Undersampling majority class
4. Change the methodology
5. Produce synthetic examples

These are only a portion of the numerous potential techniques to attempt while managing imbalanced datasets, and not a comprehensive rundown. Some other techniques to consider are gathering more information or picking different resampling proportions - you don't must have precisely a 1:1 proportion! One must continuously attempt a few methodologies and afterward conclude which is best fit. In the advanced world, data collected are exponential, and with such amount of data comes a huge number of imbalanced datasets as well which was the prime reason behind selecting the research. In this research, we have also highlighted a lot of interesting facts from many popular and latest papers and their implementation. This research work is a briefing on the analysis of Imbalanced data sets by making use of widely used techniques.

REFERENCES

[1]. **[1** Pandya, Rahul & Charak, Sujal & Moolya, Suraj & Dahivalkar, Ritish & Gadhadara, Hardik. (2021). Polarity Testing and Analysis of tweets in Twitter using Tweepy. 10.13140/RG.2.2.27330.09921.

[2]. The Dataset: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

[3]. Lebichot, B., Paldino, G.M., Siblini, W. et al. Incremental learning strategies for credit cards fraud detection. Int J Data Sci Anal 12, 165–174 (2021). https://doi.org/10.1007/s41060-021-00258-0

[4]. Dal Pozzolo, Andrea & Caelen, Olivier & Le Borgne, Yann-Aël & Waterschoot, Serge & Bontempi, Gianluca. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. Expert Systems with Applications. 41. 4915–4928. 10.1016/j.eswa.2014.02.026.

[5]. Bogner C, Seo B, Rohner D, Reineking B (2018) Classification of rare land cover types: Distinguishing annual and perennial crops in an agricultural catchment in South Korea. PLoS ONE 13(1): e0190476. https://doi.org/10.1371/journal.pone.0190476

[6]. Le Borgne, Yann-Aël & Bontempi, Gianluca. (2021). Reproducible Machine Learning for Credit Card Fraud Detection - Practical Handbook.