Classification of Machine Learning techniques effective on imbalanced Datasets

Suyash Chaudhary

Dept. of AI and Machine Learning

AI-Tech Systems

ai-techsystems.com

Aurangabad, India

suyash15122@gmail.com

***Abstract*—changing or modifying the dataset under study for any kind of research or experimentation is pretty common in the world of Machine learning. On observation, it is pretty evident that most of the datasets found online is raw i.e. they can be either unstructured or they tend to be highly biased towards a particular trend. In other words, one or more attributes can be imbalanced in their values or frequency which makes obtaining any insights or information from the dataset quite difficult and visualization equally misleading. This imbalance leads to a higher cost/error rate which is never minimized no matter how much any model based on these types of dataset is trained .This paper is focused on effect of various techniques used to balance the dataset and it will also cover the results each of these techniques provide and it will provide a classification among these techniques in order to differentiate the least effective techniques from the most effective ones. This paper will focus on some of the most common techniques which are being used to balance the datasets like normalization, metrics change, precision vs recall, F1 score, k-fold cross-validation, re-sampling of training set, etc.**

Keywords—normalization, imbalanced datasets, k-fold cross-validation, F1 score, precision, recall, under-sampling , oversampling, bootstrapping.

I. Introduction

When we dwell deeper into the world of machine learning we realize that balanced datasets are quite a rarity and even those that are balanced have been so due to some preprocessing, cleaning and manipulation steps performed on them before they were deployed. But what is an imbalanced dataset ? an imbalanced dataset is a dataset which has a sort of discrepancy among it's values or classes. there are many possibilities that leads to this discrepancy like too many values in one class compared to the other classes and a very large difference between the values of datasets like values in either of the extremes which is not an ideal situation at all.

This imbalance later affects the algorithms which use this data and models to provide prediction or classification solutions leading to inaccurate results, this can have some serious implications when the results are use in real-life applications like for instance suppose a stock market analyst is using an unbalanced dataset which is raw in nature and is based on the recent trends observed in the past and he uses the parameters based on these past observation to predict the future scenario, due to presence of incorrect predictions many investors who trusted the analyst can lose all their money that they invested and this leads to a disastrous situation. Similarly suppose a classifier is being trained to differentiate between legal and illegal drugs found in airport custom security check, if there is a lack of sufficient values or information in the class of illegal drugs compared to legal ones, the model will fail to classify accurately and it will classify even the illegal drugs as legal ones because there is quite a huge difference in both of the classes.

This is where these balancing techniques comes in, these techniques are required to be performed before the algorithms are performed on the model as part of the data preprocessing steps. each step will be performed either individually or in combination as per the accordance of the dataset under study. Next further in this paper we will discuss what these techniques are.

II. SAmpling

A. Undersampling

In a gist under-sampling is the process of balancing the dataset by reducing the size of the abundant class and making it close to the scarce class in terms of size. This method is used when the quantity of the data is sufficient and any kind of reduction or elimination of data won't have any drastic effect on the model. in short by keeping all the samples in the rare or scarce class and randomly selecting an equal number of samples in the abundant class, a balanced new dataset can be retrieved for further modeling and experimentation. One of the most well known under-sampling techniques is called Cluster Centroid based Majority Under-sampling technique or (CCMUT). In majority under-sampling, unimportant instances are removed among majority samples, in CCMUT, the determination of instances as important or unimportant is done by using the concept of clustering on feature space geometry.

Clustering is an unsupervised learning approach but CCMUT only uses the concept of finding cluster centroid as instances are already labeled into appropriate classes. This centroid is obtained by finding the average feature vectors for all the features ,over the data points belonging to the majority class in the feature space. Similar to the process of the K-means clustering Algorithm, the CCMUT deems the instances which are farthest from the cluster centroid as unimportant and it eliminates them and vice versa for the instances closest to the centroid.

B. Oversampling

In a gist, Oversampling is used when the quantity of data is insufficient. It tries to balance the dataset by increasing the size of rare or scarce samples. Rather than getting rid of abundant samples like in under-sampling. New rare samples are generated using various techniques. One might think that why increase the number of instances at all when the model is giving predictions or classification, because due to the overabundance of the instances of the majority class, the model will always predict the majority class or it will classify all the data into the majority class which is problematic. Methods that are used for balancing using oversampling are random naive over-sampling, SMOTE and Bootstrapping.

1) Random Naive

Random Naive over-sampling is the easiest way to achieve over-sampling through repetition. This method randomly selects instances or observations from the scarce or rare class and adds them to the dataset until we achieve a balance between the majority class and the rare class. One issue with this technique is that since this technique employs the use of random repetition for balancing the dataset it might not improve the error rate of the model by a larger margin.

2) Synthetic Minority Oversampling

Synthetic Minority Over-sampling Technique also known as SMOTE oversamples the rare class but it does not rely on repetition of rare class instances instead it creates new instances or observations based on the instances or observations in your data. SMOTE relies on KNN(K-Nearest Neighbor) for instance creation. First it finds the centroid of the observations of the rare class, then it finds the nearest neighbors to that centroid. then according to the extent we want to over-sample the data, the neighbors are selected for new instance creation. Now the final step involves creation of new instances by randomly choosing a point on the line connecting the instances to their nearest neighbors. SMOTE's main advantage is that it does not creates repeated instances rather it creates synthetic instances for the rare class which could have been observed in the reality thus your model is less likely to over-fit.

3) Bootstrapping

Bootstrapping is an over-sampling technique which can be used to quantify the uncertainty associated with a given model or statistical learning approach. In a gist, the bootstrap generates distinct data sets by repeatedly sampling instances from the original data set. These generated data sets can be used to determine the variability in the model instead of sampling independent datasets from the full data population. The sampling approach employed by bootstrap involves randomly selecting n instances with replacements, which means some instances can be selected multiple times while other instances are not included at all.

III. Metrics

Applying inappropriate evaluation metrics for model generated using imbalanced data can turn out to be pretty hazardous. Most of the time when an ideal dataset is not involved, using a evaluation metrics based on accuracy does not yield useful results take for instance a model into consideration which classifies almost all of its instances into only a particular class then even though the model is giving almost 99 % accuracy it still won't be of any use. So for this reason other evaluation metrics should be taken into consideration when an imbalanced dataset is involved. All the other metrics are determined using a special matrix called confusion matrix

A. .Confusion Matrix

Actual Values

Predicted Values	True Positive(TP)	False Positive(FP)
	False Negative(FN)	True Negative(TN)

A confusion matrix is a table that is often used to describe how a model has performed on the given dataset for which true values are known. It is also known as error matrix and it allows the visualization of the performance of an algorithm.It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made. The number of correct and incorrect predictions are summarized with count values and broken down by each class . To sum it up there are only two classes in a confusion matrix a positive class(1) and a negative class(0) and there are four types of predictions which are obtained from these classes

- True Positive: These are the observations which were predicted to be positive by the model and they turned out to be positive.
- True Negative: These are the observations which were predicted to be negative by the model and they turned out to be negative.
- False Positive: These are the observations which were predicted to be positive by the model but they turned out to be negative.
- False Negative: These are the observations which were predicted to be negative by the model but they turned out to be positive.

with the help of these four types of predictions we obtain the several types of evaluation metrics which are useful to the model these are

B. *Accuracy*

Accuracy is a metric for evaluating models. Informally, accuracy is the fraction of predictions our model got right. For binary data it can also be calculated in terms of positives and negatives.

Accuracy = TP + TN/TP + TN + FP + FN

C. *Precision*

Precision is the evaluation metric which is used to determine out of all the predicted positive observations how many are actually positive. It makes use of True positive and False positive

Precision = TP/TP + FP

D. *Recall*

Recall is the evaluation metric which is used to determine out of all the True positive observations how many were predicted successfully. It makes use of True Positive and False Negative

Recall = TP/TP + FN

As precision and recall both are at odds with each other, both can never be high or low simultaneously. There needs to be a balance among them as having both of these values on either of the extreme can complicate the model. When recall is high and precision is low, it means that most of the positive observations are successfully predicted but there is also a higher number of False Positives. Similarly, high precision and low recall means that the model missed a lot of positive observations but those that it predicted as positives are indeed positive observations i.e. low False Positives. That's why a need for a balance arises and this tradeoff is called the precision Vs accuracy tradeoff.

E. *F score*

The F score also known as the F1 score or F measure is an evaluation metric of the test's accuracy. It is also defined as the weighted harmonic mean of the test's precision and recall. The F score reaches the best value, meaning perfect precision and recall, at a value of 1. The worst F score, which means lowest precision and lowest recall, would be a value of 0.

The F score is used to measure a test’s accuracy, and it balances the use of precision and recall to do it. The F score can provide a more realistic measure of a test’s performance by using both precision and recall. The F score has often found its use in information retrieval for measuring search, document classification, and query classification. F score's formula is given by

2 * precision * recall / precision + recall

F. *Cohen's Kappa*

Cohen's Kappa is a very useful but underrated performance evaluation metric found in the world of machine learning. In machine learning, we often encounter the problem of multi-class classification. When such problem is encountered, other evaluation metrics like accuracy, precision and recall do not provide a complete picture of the model. A multi-class classifications problem arises when the classes observed for the given model are non-binary in nature i.e. there is presence of 3 or more classes. And the model is encountering trouble in classifying an instance into any of the observed classes. It is also useful in case of imbalanced class problems as it provides a better intuitive explanation for an instance belonging in a class compared to the F score. The Cohen's Kappa formula is given by

1 - (1 - Po / 1- Pe)

Here Po and Pe are the observed and expected agreements respectively. Agreements suggest how much close are the values numerically i.e. how accurate is the observed value compared to what the expected value stated. These Observed and Expected agreements are nothing but probabilities obtained from the expected and actual values of the confusion matrix.

Cohen's Kappa is always less than or equal to 1. Value of 0 or less indicates that the model is pretty much useless. The following is a table which was created to indicate the meaning behind the values.

Kappa	Agreement
<0	Less than chance agreement
0.01-0.20	Slight agreement
0.21-0.40	Fair agreement
0.41-0.60	Moderate agreement
0.61-0.80	Substantial agreement
0.81-0.99	Almost perfect agreement

G. *AUC-ROC Curve*

AUC stands for Area Under Curve and ROC stands for Receiver Operation Characteristic. This is also a performance evaluation metric for classifier problems at various threshold settings meaning it tells how well a model can distinguish between two different classes or categories (e.g. a person is a drug addict or not). Better models are found to be accurate when distinguishing classes whereas those who don't are not.

ROC curve is a graph plotted using a Recall Vs False Positive Rate curve. The formula for these is given by

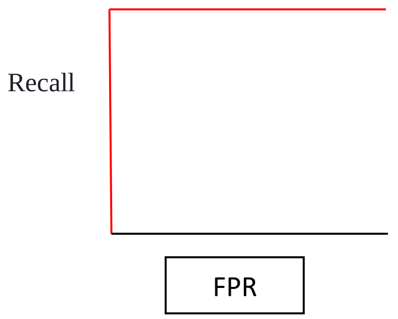
Recall=TP/TP+FN

Specificity=TN/TN+FP

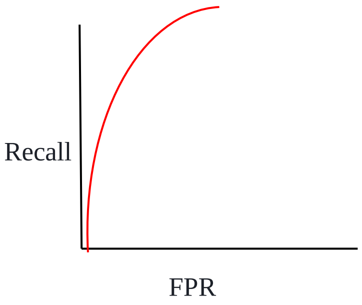
False Positive Rate = 1 - Specificity

= FP/TN+FP

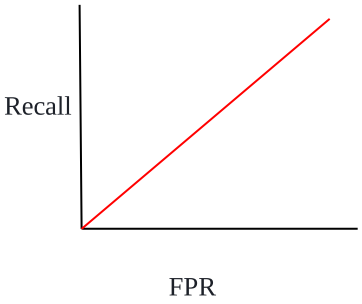
Now how do we distinguish whether a model is excellent and balanced from those that are not ? An excellent model has an AUC near to 1 which means it has a good measure of separability. And when the AUC is near 0.5, that means that the model has no class separation capacity whatsoever.



This is an ideal situation where the ROC has an AUC 1



This is a situation where the chances of a model successfully predicting a class is 70 %



This is the worst situation where the model has an AUC close to 0.5 and it has no classification capacity at all.

IV. K-Fold Cross Validation

As it is evident that in most of the scenarios that there is never enough data for feeding your model, splitting some part of it for validation set creation causes a reduction in the training data and there is a risk of loss of important instances and observations which introduces errors and inaccuracies in the model . In order to deal with this scenario, we require K-fold cross validation so that there is sufficient data for training and validation set.

This method of balancing the dataset is quite popular because of its simplicity and because it generally results in a less biased or less optimistic estimate of the model than other methods such as a simple train/test split.

The general approach for applying the K-fold Cross Validation is given by:

1. Shuffling the dataset randomly
2. Splitting the datasets into K groups
3. For each unique group

1) Take the group as a holdout or test set

2) Take the remaining groups as the training set

3) Fit the model on the training set and evaluate it on the test set

4) Retain the evaluation score and discard the model

4. Summarize the performance of the model using the sample of model evaluation metrics score.

Each observation in the data is assigned to an individual group and it stays in that group for the duration of the procedure which means that each observation is given an opportunity to be used in the hold out set one time and is used to train the model K-1 times. In this method the first fold is usually treated as the validation set and the rest K-1 folds are fit into the model. Interchanging the instances of training and test sets also adds to the effectiveness of this method. The general rule for assuming the value of k around 10 i.e. k=10. K-fold cross validation can also be executed using two approaches.

A. *Stratified K-fold Cross Validation*

in some cases when performing study or experimentation on any dataset there is a high chance of the presence of an imbalance between the instances and classes. For example, in a dataset involving housing prices, there might be quite a large number of houses who have high prices or in a classification model it is observed that there might be way more instances in the negative class as compared to the positive class. For such situations, a slight modification is made in the working mechanism of k-fold Cross Validation. Each fold now contains approximately the same percentage of instances of each target class as the complete set or in case of prediction problems, the mean response value is approximately equal in all the folds. This modified version is called the Stratified K-fold Cross Validation.

B. *Leave p-out Cross Validation*

This approach leaves p data instances out of the training data, i.e. if there are n data instances in the original sample, then n-p instances are used to train the model and p instances are used as validation set. This is repeated for all the combinations in which original instances can be separated this way and then the error is averaged for all the trials. This method can be exhaustive in the sense that it needs to train and validate the model for all the possible combinations and this can be computationally very heavy and not feasible at all.

A particular case of this method is when p=1 and this approach is known as leave one out cross validation. This method is preferred more because it is not computationally intensive and the number of possible combinations is equal to the number of the data instances in the dataset.

K-fold cross validation is one of the best balancing techniques out there because of its effectiveness in mitigating the over-fitting of the model and its accuracy in choosing the right hyper-parameter for the model i.e. the parameter with the lowest test error.

V. Conclusion

Thus we have studied various techniques which are useful when we want to balance an imbalanced dataset. Since the data that is used for study and experimentation in machine learning is so diverse, each of these techniques can have a different effect on the data. There are many scenarios which can play out while balancing a dataset, sometimes one of these techniques might be sufficient enough to balance the dataset while sometimes more than one of these techniques might be essential in balancing.

A careful study of data and the nature of the data should be done initially because understanding is the first key to obtaining meaningful solutions. Then only balancing techniques should be performed. Trying out most of these techniques and comparing their results is beneficial in the long run as this approach will provide an optimal solution when balancing is done.

VI. References

[1]Yunchun Tang, Yan Qing Zhang, Nitesh V. Chawla and Sven Krasser 'SVM Modeling for Highly Imbalanced Classification':https://www3.nd.edu/~dial/publications/tang2009svms.pdf

[2]Vaishali Ganganwar 'An overview of Classification Algorithms for Imbalanced Datasets': http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.413.3344&rep=rep1&type=pdf

[3]https://ieeexplore.ieee.org/abstract/document/4695979

[4]Foster Provost 'Machine Learning from imbalanced datasets': https://www.aaai.org/Papers/Workshops/2000/WS-00-05/WS00-05-001.pdf

[5]Yanmin Sun, Andrew K.C. Wong and Mohamed S. Kamel 'Classification of Imbalanced Data: a Review': https://www.worldscientific.com/doi/abs/10.1142/S0218001409007326