A critical analysis of Sampling Techniques for imbalanced data classification: An application to Social Media

MSc Research Project
Data Analytics

Vinitha Nagarajan
x16142233

School of Computing
National College of Ireland

Supervisor: Dr Giovani Estrada

National College of Ireland
Project Submission Sheet – 2017/2018
School of Computing

Student Name:	Vinitha Nagarajan
Student ID:	x16142233
Programme:	Data Analytics
Year:	2017
Module:	MSc Research Project
Lecturer:	Dr Giovani Estrada
Submission Due Date:	11/12/2017
Project Title:	A critical analysis of Sampling Techniques for imbalanced data classification: An application to Social Media
Word Count:	3600

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	
Date:	11th December 2017

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
3. Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A critical analysis of Sampling Techniques for imbalanced data classification: An application to Social Media

Vinitha Nagarajan
x16142233
MSc Research Project in Data Analytics

11th December 2017

Abstract

Imbalanced training datasets appear in a number of real-life problems, such as anomaly detection, network monitoring and social media. While some classes will normally have a sizeable amount of records, other classes will be underrepresented. Constructing efficient classifiers for minority classes is a challenge which has been addressed in various ways, but basically grouped into undersampling the majority class(es) and oversampling the minority class(es), or a combination of both techniques. This thesis will focus on imbalanced classification techniques for social media data from Twitter. The classification task at hand is the identification of spam tweets. Social networks are a rich source of information, but also attracts many illegitimate users who spread spam tweets. A machine learning approach is presented whereby analytical models are learnt from highly skewed datasets to predict spam messages. A range of techniques to tackle class imbalance are analysed in detail by controlling the class imbalance ratio. It is indeed possible to identify techniques of superior performance according to the imbalance ratio they can cope with. It is shown that classification performance heavily depends on the imbalance degree of the dataset and their sampling techniques.

1 Introduction

There is a massive growth of digital sources being collected and stored by organisations around the world. The gap between data production and our ability to understand it requires advanced data analytics techniques. Most data intensive fields use classification techniques to automate analysis tasks, such as bioinformatics, remote sensing, sentiment analysis or spam detection. Machine classification algorithms provide great opportunities for automation, but at same time a misclassification can lead to human injury or financial loss. Data should be well studied before building a classification model, because real world data may vary over time and thus pose challenging problems(Liu et al.; 2009). Imbalance learning is a very active research field in both academia and industry. It often happens the outliers are interesting events which can pave the way to detect device anomalies, financial fraud, security leaks or machine failures.
In other words, traditional classification algorithms focus on well-represented classes,

not on the minority class. Accuracy is often measured on the majority class, unless cost-sensitive classification is performed. Weights in the cost matrix are also difficult to estimate. Another way to deal with minority classes or rare events is sampling. In this research paper the most important sampling techniques are studied and their impact on the classifier are noted. For conducting experiments, gradually imbalanced twitter data has been selected.

2 Related Work

2.1 Sampling Techniques

Class imbalance problem are mostly dealt with undersampling and oversampling, or a combination of both techniques. The most important techniques are listed below.

Random − Oversampling: This method balances the class by eliminating some instances in the majority class.
Random − Undersampling: This method replicates the instance of the minority class to balance the distribution(Batista et al.; 2003).
Both the method has known for drawbacks, undersampling can eliminate some important instances and oversampling can lead to overfitting the dataset. To overcome these problems many studies has been done.
SMOTE: (Chawla et al.; 2002). (2002) proposed a combination of oversampling and undersampling which can achieve better results than when using only undersampling the majority class. This method can overcome the drawbacks of overfitting the dataset. In this method, the authors created a synthetic data rather than replicating a minority class.
SMOTEBorderline: (Han et al.; 2005) proposed an oversampling method based on the synthetic minority oversampling (SMOTE). Based on the SMOTE method they have developed two minority oversampling techniques Border1 and Border2, where only the border instance of minority class is oversampled.
SMOTE+TOMEK: (Batista et al.; 2003) proposed two new combination techniques called TOMEK + SMOTE and ENN + SMOTE. Tome link can be used as undersampling or cleansing of the dataset, this method used to avoid the invading of majority class into minority class and vice versa. ENN is similar to Tomeklink but it can remove more examples in depth, it removes both the classes which are misclassified by three nearest neighbours.
ENN: (Wilson; 1972) introduced a nearest neighbour rule that refine the decision boundary by cleaning up points which overlap between classes.
ADASYN: (He et al.; 2008) introduced a novel adaptive synthetic sampling approach for class imbalance focusing on the minority class. Than main idea behind this providing density distribution so that it automatically generates synthetic samples for the minority class. This is major difference between SMOTE and ADASYN, where the number of samples created for each minority class varies.

2.2 Spam detection

(Wang; 2010) published the first paper on machine learning technique for twitter spam

detection. Previous works were mostly on identifying email spam. In their paper, the authors proposed a social graph model to identify relationship among friends and followers, also proposed graph based and content based features. A Bayesian algorithm is used to classify non-spam and spam.
In (Meda et al.; 2014), authors selected five important features and applied random forest algorithm. Two phases were implemented, first phase is training of data using random forest classifier in the offline and parameters are fixed. In the second phase, real-time twitter messages were classified as spam and non-spam.
(Guo and Chen; 2014) dealt with spam detection based on geo tagged networks for geo social networks. Spam detection used this geographic information, content based and graph based features like maximum speed, mean speed, hashtags, URLs. Based on these features, a data set was created, and supervised classification algorithm were applied.
(Miller et al.; 2014) approached spam detection using clustering algorithm. Three novel contributions were introduced in their paper. Firstly, the problem is viewed as anomaly detection. Secondly, authors proposed n-gram feature from the tweet text and finally, clustering algorithms were used to identify spam (StreamKM++ and DenStream).
(Chen et al.; 2017) proposed a novel method called Lfun to overcome a drawback of statistical features which change over time. Lfun methods helps to update the classifier training method in real time.
(Liu et al.; 2017) addressed the class imbalance problem in the twitter spam detection from a different perspective. They proposed a new sampling approach called fuzzy based oversampling. A sampling method is applied to the minority class prior classification. Finally, an ensemble approach was used to increase the spam detection rate.
(Hirve and Kamble; 2016) introduced new features to the training dataset. They used supervised learning methods, but considered hashtags and URLs as their unique features. As twits were limited to 140 character other features, such as hashtags and URLs, could help in detecting spam.

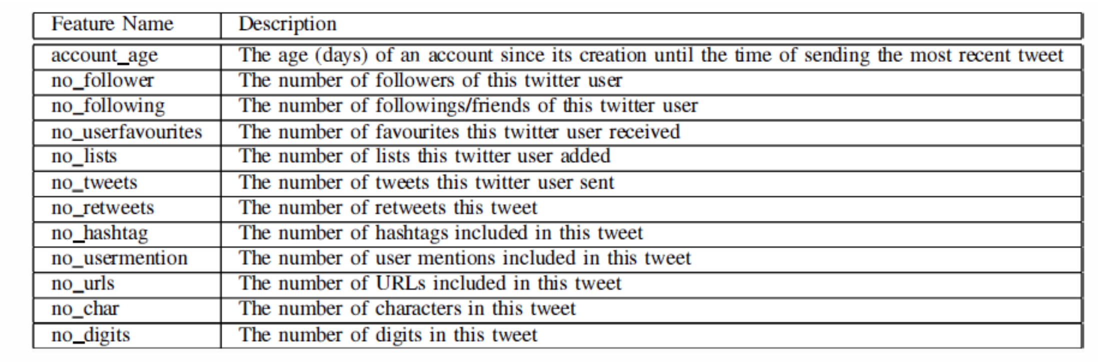
A detailed study is here presented on imbalanced data classification techniques for spam detection. In this paper class imbalance problem in social media data has been approached by different sampling techniques with various class imbalance ratio and their impacts on classification are evaluated.

3 Methodology

For this research report the Python programing language was used for all sampling and predicting analysis. Python is a free and open source programming language with many toolboxes for data manipulation and analysis, for instance Pandas, imblearn, scikit-learn, etc. Python is a language for rapid prototyping.

3.1 Background Of The Dataset

The labelled dataset used in this research was obtained from (Chen et al.; 2015). This dataset was manually labelled by the authors and made it available to the research community. They used streaming API to collect tweets with URLs from the twitter. URLs are considered as the ground truth for the dataset and through tweets also spam can be spread, but after conducting manual research on 1000 tweets only very few are spam



Feature Name	Description
account_age	The age (days) of an account since its creation until the time of sending the most recent tweet
no_follower	The number of followers of this twitter user
no_following	The number of followings/friends of this twitter user
no_userfavourites	The number of favourites this twitter user received
no_lists	The number of lists this twitter user added
no_tweets	The number of tweets this twitter user sent
no_retweets	The number of retweets this tweet
no_hashtag	The number of hashtags included in this tweet
no_usermention	The number of user mentions included in this tweet
no_urls	The number of URLs included in this tweet
no_char	The number of characters in this tweet
no_digits	The number of digits in this tweet

Figure 1: Feature Discription

without URLs so, the authors limited collection of tweets with URLs. Once labelling the dataset is done, authors extracted 12 very important light features, which has more impact on detecting the spam. The below figure 1 has the features list with their description.

3.2 Data Mining Techniques

Classification accuracy of the minitory class in highly skewed datasets is often poor. The minitory class is outnumbered by other classes and thus classifiers (e.g. rules and trees) normally overlook those classes in favour of majority ones which will yield the largest overall accuracy. A breakdown of predictions per class (e.g. confusion matrix) will usually keep the minitory class as outliers. One strategy is to construct a classifier per class, so the ensemble of classifiers will outperform any classifier alone. This research will focus on two popular methods: decision trees and random forests.

3.2.1 Decision Trees

Decision trees can used for classification and a variant is the regression trees. It is very easy to interpret and works on linearly separable data. It is robust to outliers and provide high accuracy rate with less computational effort. Decision trees do not usually perform well on highly dimensional datasets. Tree pruning is normally done to avoid overfitting the model (Crisci et al.; 2012).

3.2.2 Random Forest

Random forests is an ensemble learning method. In ensemble learning multiple machine learning algorithms are put together into large models, which leverage the accuracy of individual classification algorithms. This method is robust to outliers as it is non-parametric algorithm. At some time, it has drawback that if many trees grows then computational time become an issue for real-time processing (Crisci et al.; 2012).

Steps to build a Random Forest:
Step 1: Pick at any random X data points from the training set.
Step 2: Build the decision tree associated to those X data points.
Step 3: Choose the number of trees you want to build and repeat.
Step 4: For a new data points, make each of your N tree predict the category to which the data points belong and assign the new data points to the category that wins the

majority vote.

3.2.3 K-Nearest neighbours (K-NN)

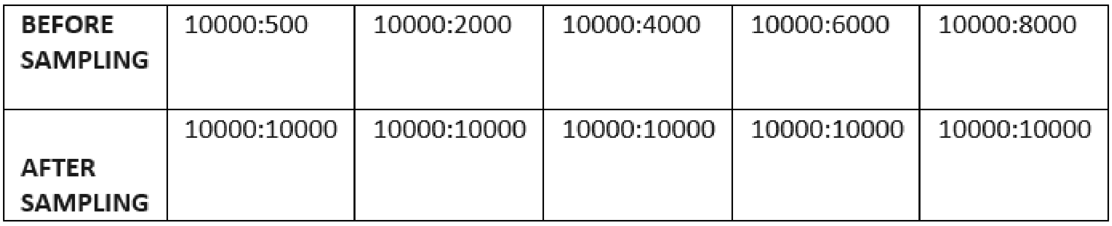
K-Nearest neighbours can be used for classification and regression purposes. This method is very easy to interpret, and fast to calculate. It is a very powerful algorithm with high accuracy. It is also called as lazy method because it does not build or generalise a model, but only memorise data points. K-NN is not robust to outliers (Crisci et al.; 2012).

Steps to build K-NN:
Step 1: Choose the number of K neighbours.
Step 2: Find the K-nearest neighbours for the new data points, according to a given distance metric (normally Euclidean distance).
Step 3: Among these K-neighbours, count the number of data points in each category.
Step 4: Assign the new data points to the category where you counted the most neighbours.

3.3 Sampling Techniques

3.3.1 Random Oversampling

This is a basic oversampling technique of the minority class, where the minority classes are replicated until the probability of both minority and majority classes are equal. It has a major drawback of overfitting the model since it replicates the minority class. Figure 2 shows the majority and minority class ratio (majority:minority) before and after random oversampling

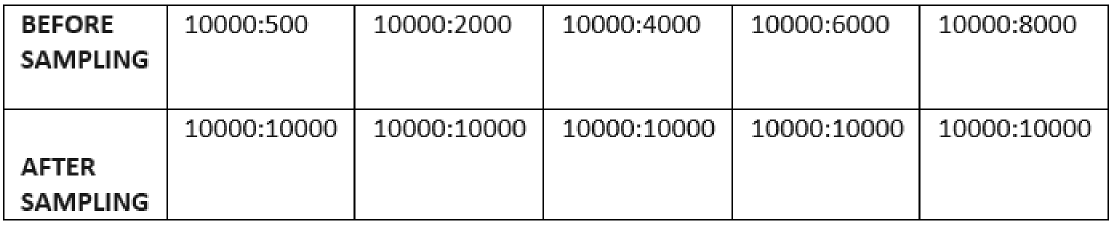


BEFORE SAMPLING	10000:500	10000:2000	10000:4000	10000:6000	10000:8000
AFTER SAMPLING	10000:10000	10000:10000	10000:10000	10000:10000	10000:10000

Figure 2: Random oversampling

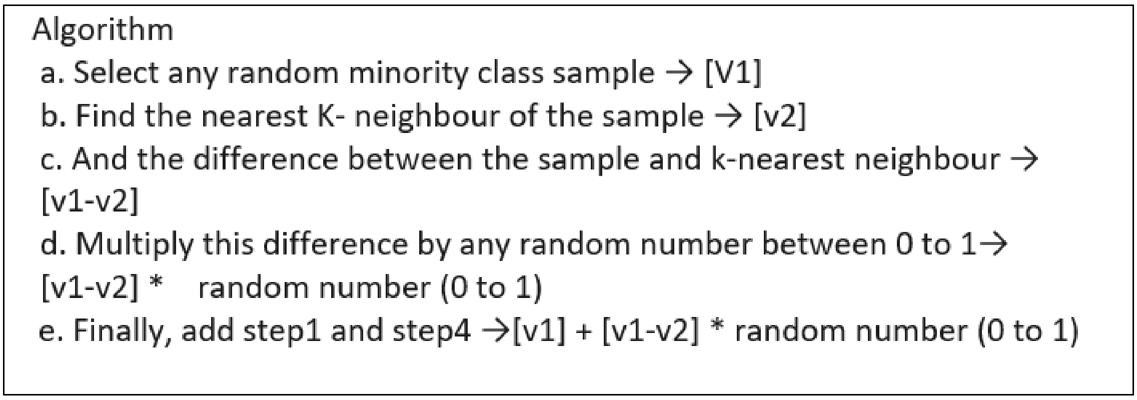
3.3.2 SMOTE

SMOTE is a different oversampling approach that creates synthetic samples rather than oversampling with replacement. Samples are created in the feature space not the data space. For each minority class a k-nearest neighbours is calculated within the decision space of the minority class. Synthetic samples are created depending upon the oversampling amount between the nearest neighbours chosen randomly. Though SMOTE has been very powerful algorithm, it has some disadvantage like generalization of the dataset(He and Garcia; 2009), this leads to overlap between the classes. Figure 3 shows a similar balancing of classes for SMOTE. The algorithm is presented in Figure 4.



BEFORE SAMPLING	10000:500	10000:2000	10000:4000	10000:6000	10000:8000
AFTER SAMPLING	10000:10000	10000:10000	10000:10000	10000:10000	10000:10000

Figure 3: SMOTE oversampling

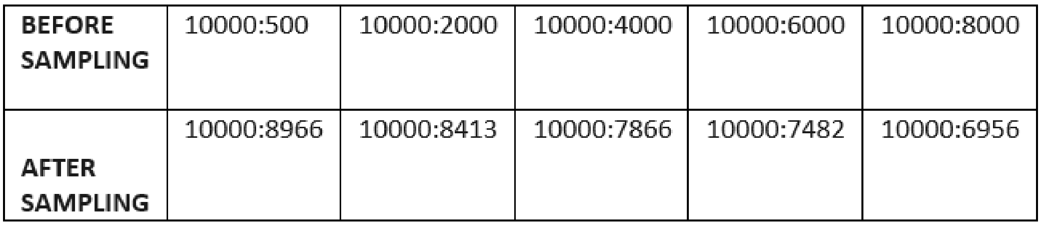


Algorithm
a. Select any random minority class sample → [V1]
b. Find the nearest K- neighbour of the sample → [v2]
c. And the difference between the sample and k-nearest neighbour → [v1-v2]
d. Multiply this difference by any random number between 0 to 1→ [v1-v2] * random number (0 to 1)
e. Finally, add step1 and step4 →[v1] + [v1-v2] * random number (0 to 1)

Figure 4: SMOTE Algorithm

3.3.3 SMOTE + ENN

'Edited Nearest Neighbour removes any samples whose majority or minority class differs from each other of at least two of its three nearest neighbours. The ENN method removes the instances of the majority class whose prediction made by K-NN method is different from the majority class. ENN method can remove both the noisy examples as borderline examples, providing a smoother decision surface. Compare Tomek link ENN tends to remove more samples, so it is expected that it will provide a more in depth data cleaning'(Walimbe; 2017).Figure 5 shows the resulting class balancing from this technique.

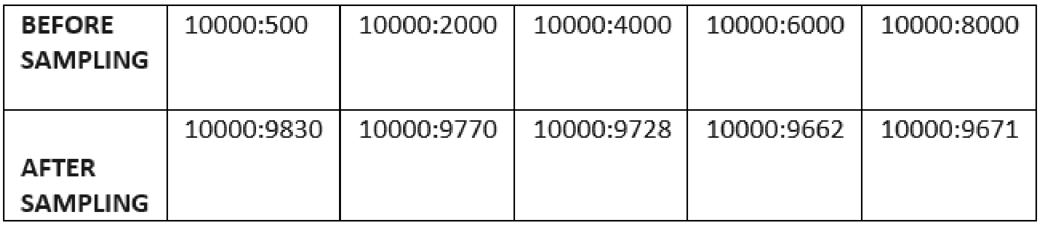


BEFORE SAMPLING	10000:500	10000:2000	10000:4000	10000:6000	10000:8000
AFTER SAMPLING	10000:8966	10000:8413	10000:7866	10000:7482	10000:6956

Figure 5: SMOTE+ENN Sampling

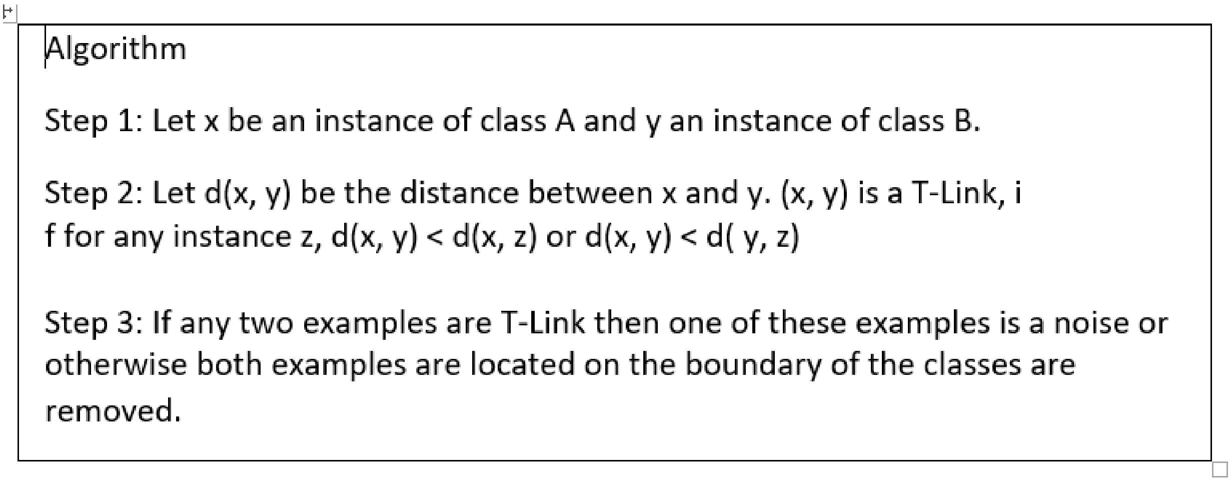
3.3.4 SMOTE + Tomek

This method combines undersampling and oversampling. TomekLink is the cleaning process of the data. Class distribution will be always problem if majority class invade into the minority class and same way after oversampling the minority class can do the same to majority class. In order to avoid overfitting and improve classification accuracy, the TomekLink cleaning method is employed. Tomek has some drawbacks that include eliminating some important features while undersampling the majority class(He and Garcia; 2009). Figure 6 shows the resulting class balancing and Figure 7 the Tomek algorithm (Elhassan et al.; 2016).



BEFORE SAMPLING	10000:500	10000:2000	10000:4000	10000:6000	10000:8000
AFTER SAMPLING	10000:9830	10000:9770	10000:9728	10000:9662	10000:9671

Figure 6: SMOTE + Tomek Sampling



Algorithm

Step 1: Let x be an instance of class A and y an instance of class B.

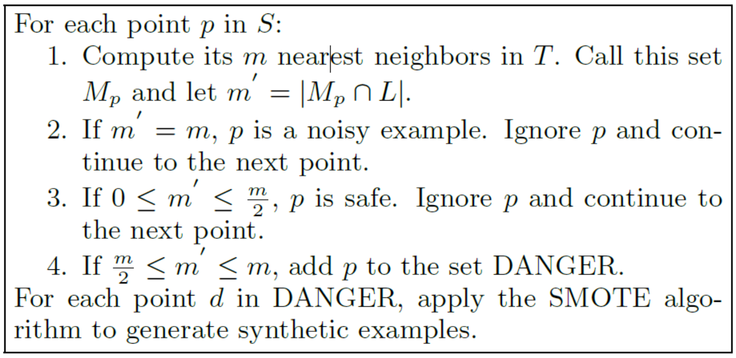
Step 2: Let d(x, y) be the distance between x and y. (x, y) is a T-Link, i
f for any instance z, d(x, y) < d(x, z) or d(x, y) < d(y, z)

Step 3: If any two examples are T-Link then one of these examples is a noise or otherwise both examples are located on the boundary of the classes are removed.

Figure 7: Tomek Algorithm

3.3.5 SMOTE Borderline

SMOTE generates new synthetic data points between the minority class using k nearest neighbours, but it does not work fine around boundary decisions. To improve more accuracy SMOTE borderline has been introduced, mostly to get higher accuracy based on border decision regions. Only borderline minority class instances are oversampled using SMOTE. However borderline has some disadvantage, as they consider border points for creating synthetic samples they tend to evade some important examples in the minority class. Following figure 8 shows the algorithm of SMOTE Borderline from author (More; 2016) and sampling results are same as shown in above figure 3 .

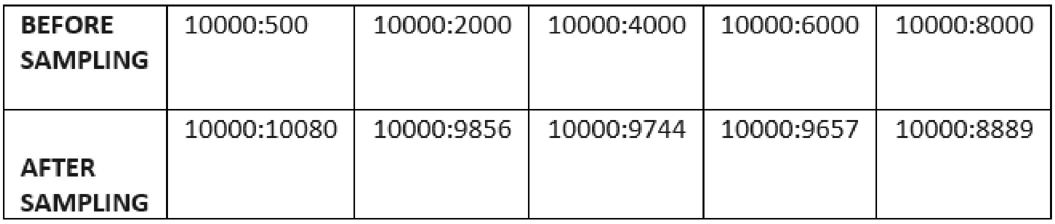


```
For each point $p$ in $S$:
    1. Compute its $m$ nearest neighbors in $T$. Call this set $M_p$ and let $m^{'} = |M_p \cap L|$.
    2. If $m^{'} = m$, $p$ is a noisy example. Ignore $p$ and continue to the next point.
    3. If $0 \leq m^{'} \leq \frac{m}{2}$, $p$ is safe. Ignore $p$ and continue to the next point.
    4. If $\frac{m}{2} \leq m^{'} \leq m$, add $p$ to the set DANGER.
For each point $d$ in DANGER, apply the SMOTE algorithm to generate synthetic examples.
```

Figure 8: Borderline Algorithm

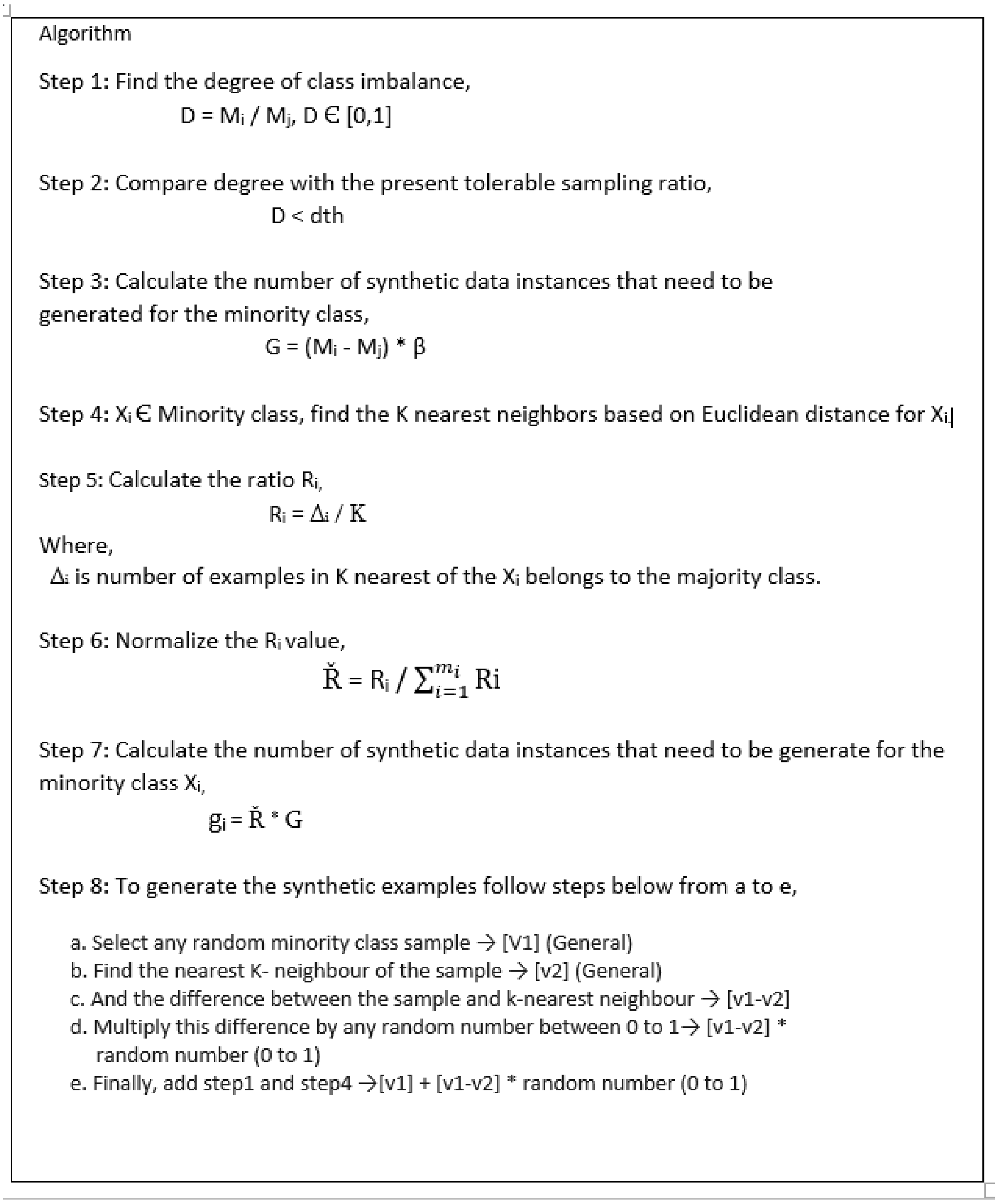
3.3.6 ADASYN

Adaptive synthetic sampling approach is based on key concept of setting density distribution to automatically generate data samples for each minority class. This method is to generate synthetic examples according to their difficult of learning the minority class examples.Figure 9 shows the resulting class balancing and Figure 10 the ADASYN algorithm.



BEFORE SAMPLING	10000:500	10000:2000	10000:4000	10000:6000	10000:8000
AFTER SAMPLING	10000:10080	10000:9856	10000:9744	10000:9657	10000:8889

Figure 9: ADASYN Sampling



```
Algorithm

Step 1: Find the degree of class imbalance,
            $D = M_i / M_j, D \in [0,1]$

Step 2: Compare degree with the present tolerable sampling ratio,
            $D < dth$

Step 3: Calculate the number of synthetic data instances that need to be
generated for the minority class,
            $G = (M_i - M_j) * \beta$

Step 4: $X_i \in$ Minority class, find the K nearest neighbors based on Euclidean distance for $X_i$.

Step 5: Calculate the ratio $R_i$,
            $R_i = \Delta_i / K$
Where,
 $\Delta_i$ is number of examples in K nearest of the $X_i$ belongs to the majority class.

Step 6: Normalize the $R_i$ value,
            $\check{R} = R_i / \sum_{i=1}^{m_i} Ri$

Step 7: Calculate the number of synthetic data instances that need to be generate for the
minority class $X_i$,
            $g_i = \check{R} * G$

Step 8: To generate the synthetic examples follow steps below from a to e,

    a. Select any random minority class sample → [V1] (General)
    b. Find the nearest K- neighbour of the sample → [v2] (General)
    c. And the difference between the sample and k-nearest neighbour → [v1-v2]
    d. Multiply this difference by any random number between 0 to 1→ [v1-v2] *
       random number (0 to 1)
    e. Finally, add step1 and step4 →[v1] + [v1-v2] * random number (0 to 1)
```

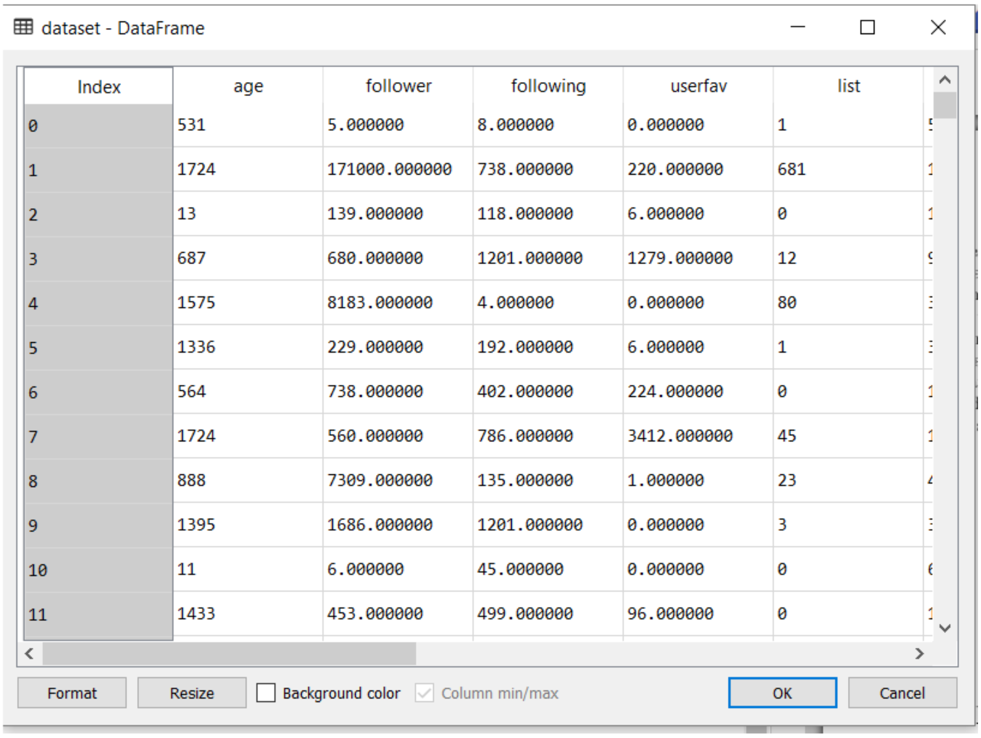
Figure 10: ADASYN Algorithm

4 Implementation

A total of 15000 instances and 13 attributes are present in the original dataset. The dataset has been manually splitted into 5 different frequency distributions of spam and non-spam classes. The splitting criteria has been chosen to cover a range of class imbalance. Five imbalance ratios were prepared:

10000:500 referred hereafter as [10]
10000:2000 as [20]
10000:4000 as [30]
10000:6000 as [40]
10000:8000 as [50]

Test and train datasets were read into Python using Pandas (read.csv) and named the dataframe as testcase. Attributes of this dataframe were separated into dependent and independent variables respectively as X and Y. Then sampling process on the dataset using various methods were conducted such as Random oversampling, SMOTE, SMOTE borderline1, SMOTE borderline2, SMOTE+Tomek, SMOTE+ENN and ADASYN using imblearn libraries. Once the sampling is done test and train data are splitted using cross validation library into ratio of 80 : 20, data mining models were built using scikit-learn (Random Forest, Decision trees, K-NN). Outputs are taken and explained under evaluation section.



Index	age	follower	following	userfav	list
0	531	5.000000	8.000000	0.000000	1
1	1724	171000.000000	738.000000	220.000000	681
2	13	139.000000	118.000000	6.000000	0
3	687	680.000000	1201.000000	1279.000000	12
4	1575	8183.000000	4.000000	0.000000	80
5	1336	229.000000	192.000000	6.000000	1
6	564	738.000000	402.000000	224.000000	0
7	1724	560.000000	786.000000	3412.000000	45
8	888	7309.000000	135.000000	1.000000	23
9	1395	1686.000000	1201.000000	0.000000	3
10	11	6.000000	45.000000	0.000000	0
11	1433	453.000000	499.000000	96.000000	0

Figure 11: View of DataFrame

5 Evaluation

In this experiment, the following five sampling techniques: Random Oversampling, SMOTE, SMOTE+Tomek, SMOTE+ENN and ADASYN, are employed to balance our skewed training dataset. Different frequency of dataset and their effect on classifiers like Random forests, Decision Trees and K-NN are evaluated and outputs are taken. Notice that SMOTE Borderline experiments are not reported, since results were very similar to SMOTE.

As discussed earlier, there is a need to evaluate the classification accuracy with some reliable evaluation metrics. As datasets were purposely created with different imbalanced

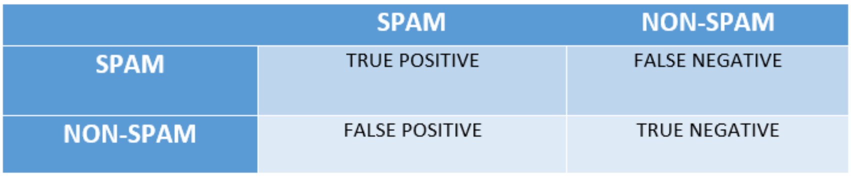
class ratios, accuracy itself cannot considered as a reliable performance measurement. Other evaluation metrics however exist, such as F-measure, recall and precision. Precision can provide you the minority class measure and false positive rate. It is a particularly important measure in social media because no legitimate customers should fall into spam (type 1 error), Recall can provide true positive and false negative measure. F-measure (also called F1 score) is the harmonic mean of recall and precision. It does provide an overall learning measure for binary classification.
The following standard formulae were computed for evaluation purposes:

$$RECALL = \frac{truepositives}{truepositives + falsenegatives}$$

$$PRECISION = \frac{truepositives}{truepositives + falsepositives}$$

$$F - MEASURE = \frac{2 * precision * recall}{precision + recall}$$

Where the acronyms are taken from the confusion matrix as follows in the Figure 12:



	SPAM	NON-SPAM
SPAM	TRUE POSITIVE	FALSE NEGATIVE
NON-SPAM	FALSE POSITIVE	TRUE NEGATIVE

Figure 12: Confusion Matrix

Following graphs shows the experiments results. In the graphs recall, precision and F-measure are shown, X-axis is constant and denotes the class imbalance ratio (1-5). The y- axis shows recall (r), precision (p) and F-measure (F).

In evaluation process first comparing the results of classification with sampling and without sampling gives an lead to the following techniques. In particular taking random forest algorithm for classification, Figure 13(a) shows that random forest algorithm is conducted on different ratio of class imbalance without sampling and their results of recall, precision, and F-measure are plotted. Observation shows highly imbalanced ratio has very less recall and F-measure, steadily increases as ratio of class imbalance change. Figure 13(b) experiments conducted with SMOTE + ENN sampling and classification is done, here results are results of recall and F-measure are very high compare without sampling, even after class imbalance ratio change still results are prominently high compared to without sampling. This gives an generalized idea about impact of sampling on imbalance learning.

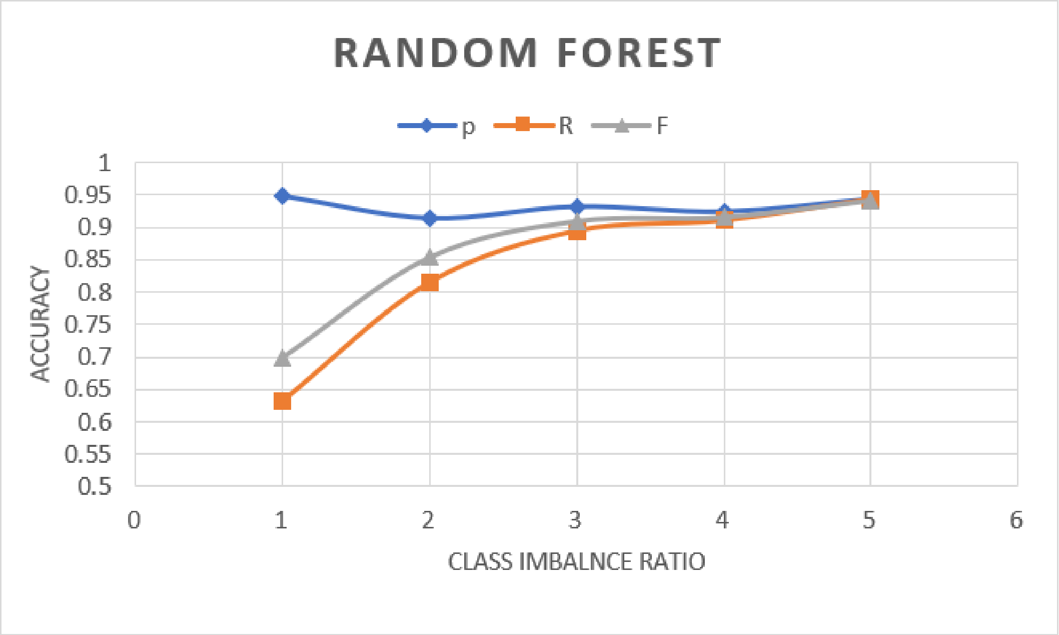


Figure 13: (a) Classification accuracy on original dataset (before sampling)

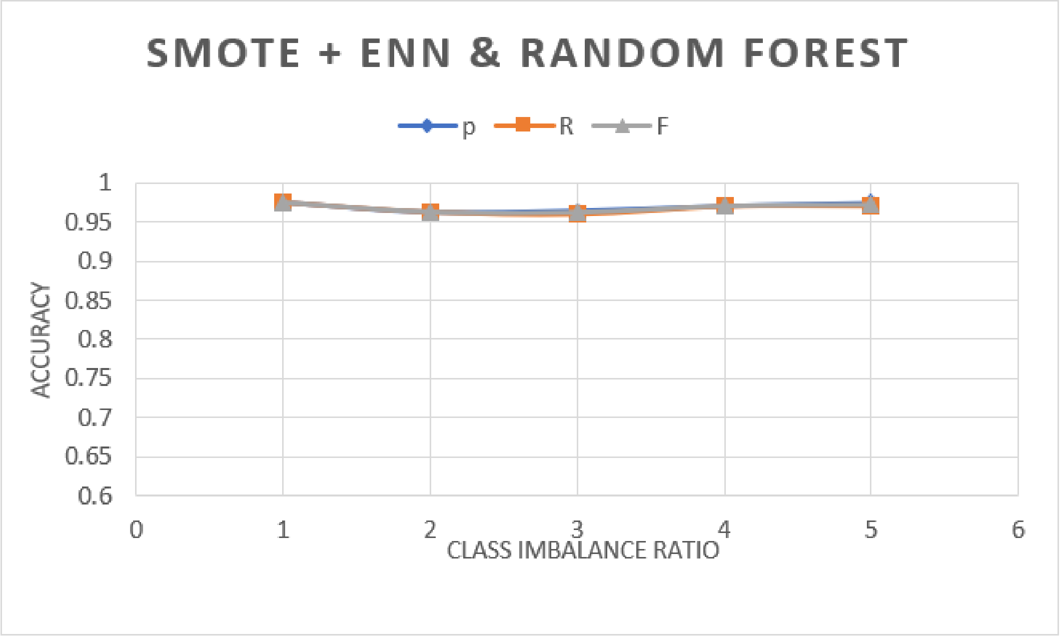


Figure 13: (b) Classification accuracy after sampling)

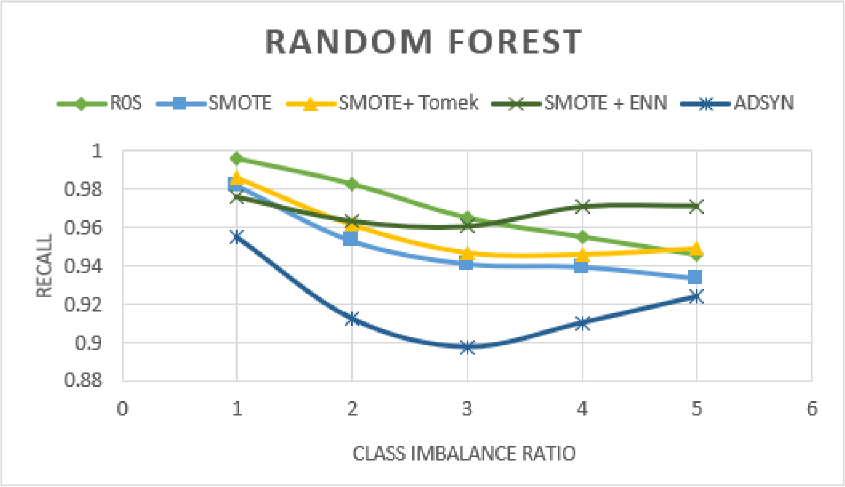


Figure 14: (a)Classification accuracy of Random Forests across the range of techniques described in this report.)

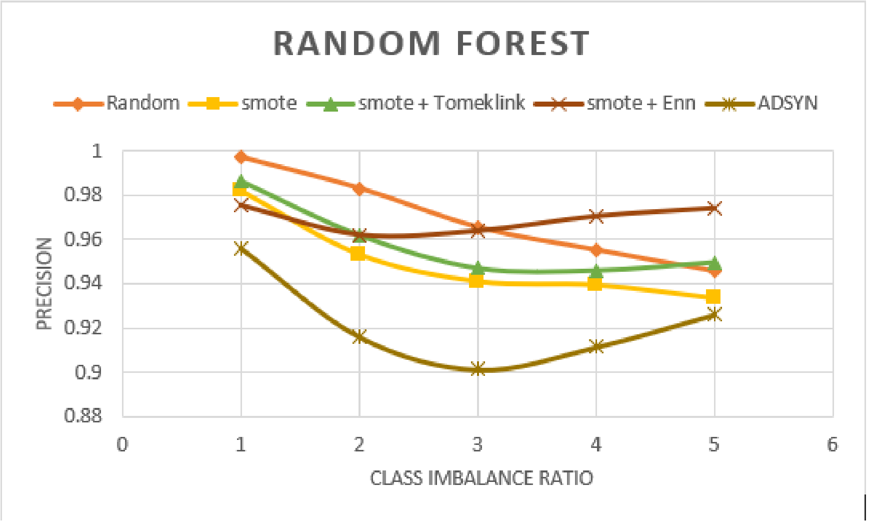


Figure 14: (b)

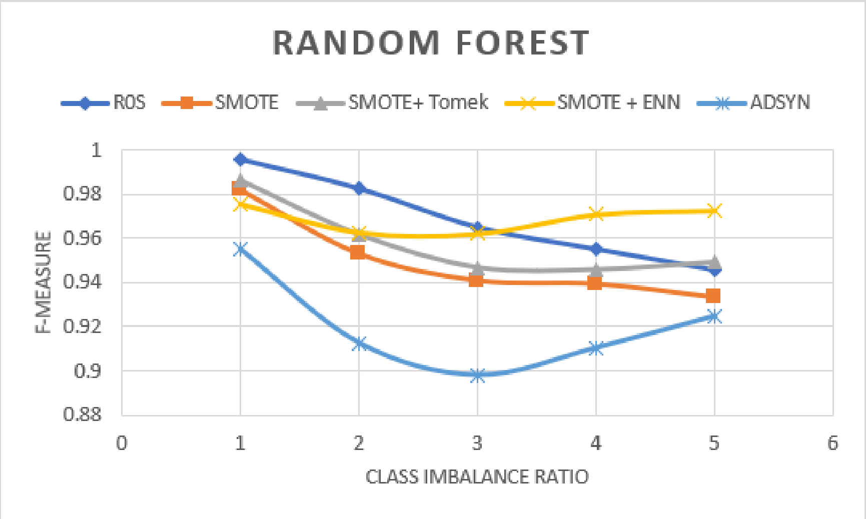


Figure 14: (c)

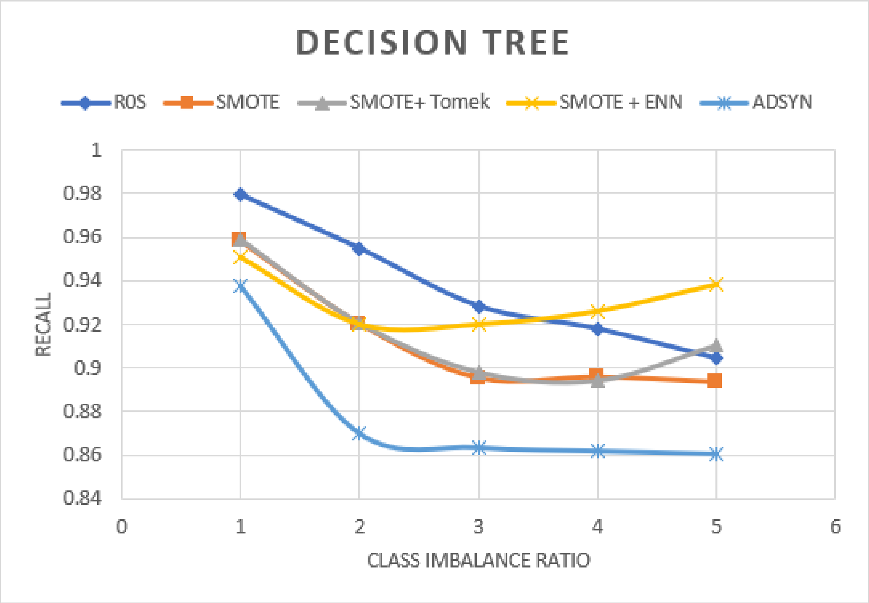


Figure 14: (d)

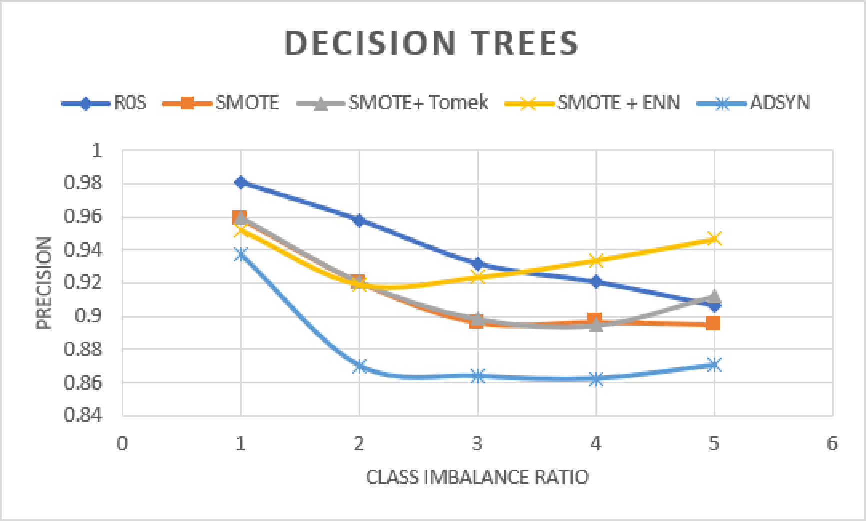


Figure 14: (e)

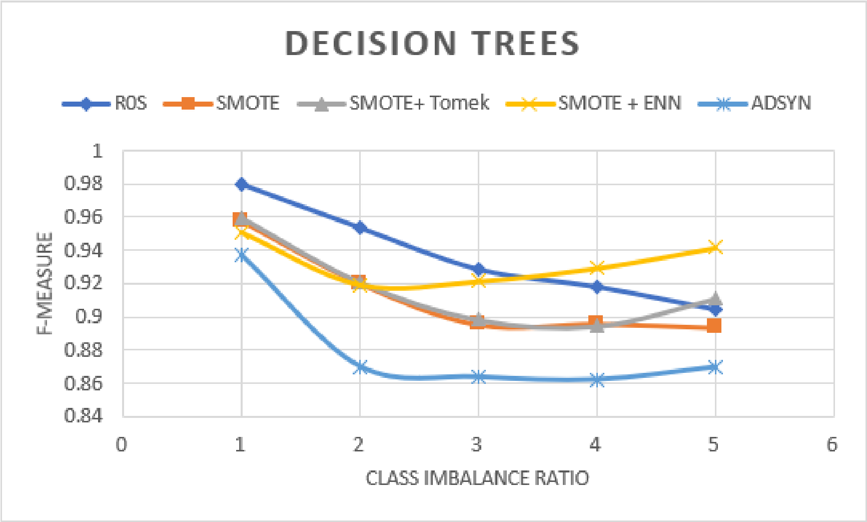


Figure 14: (f)

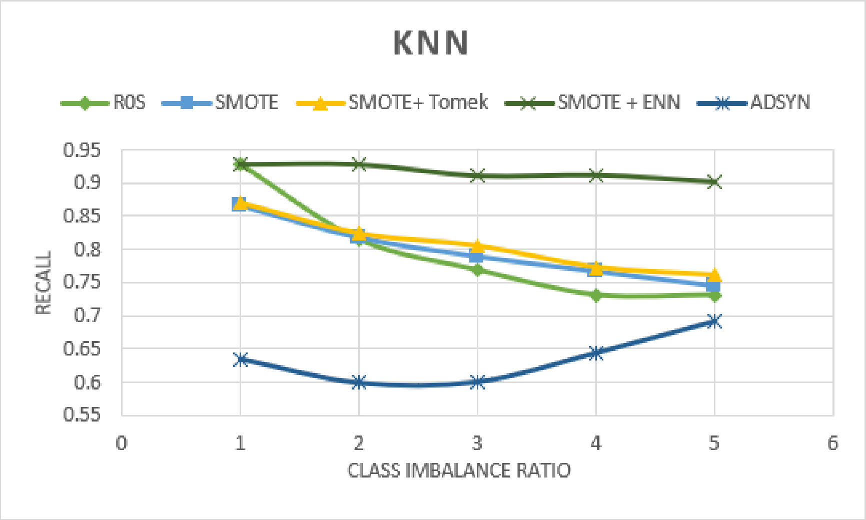


Figure 14: (g)

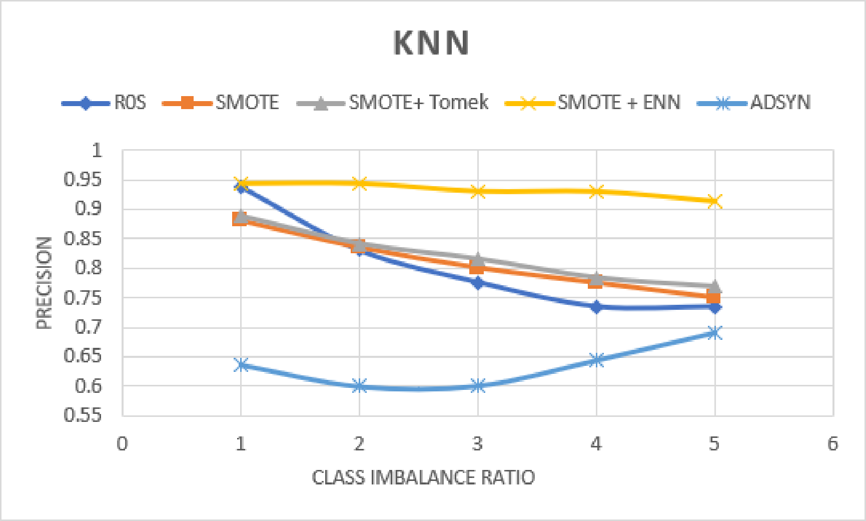


Figure 14: (h)

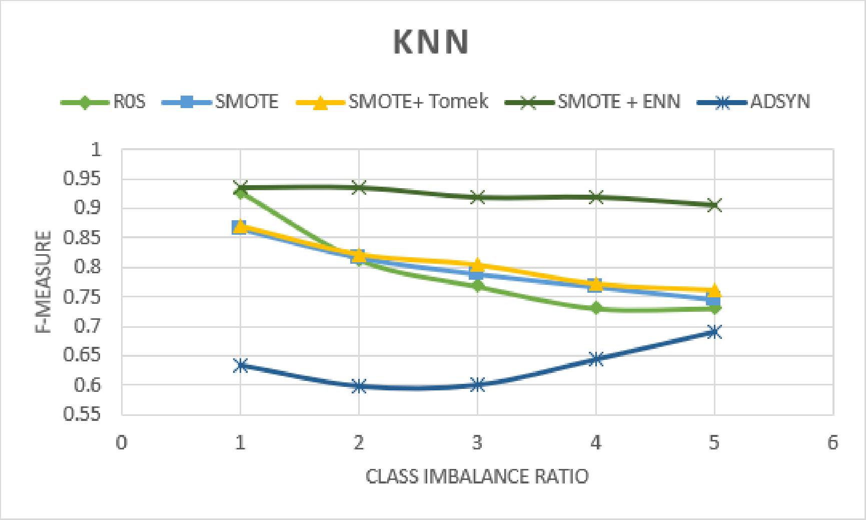


Figure 14: (i)

5.1 Discussion

Results from Figure 14(a,b,c) reveals that a critical behaviour of the evaluated sampling techniques. First of all, after analysing the result of random forest on the five sampling techniques and different ratio of class distribution. Figure 14(b) illustrates the precision graph results obtained with different class imbalance ratio, random oversampling starts with higher level but slowly reducing to lower rate as levels of spam ratios increased and it has drawback of overfitting even if it gives 0.98 precision but models tend to be erroneous. SMOTE + Tomek and SMOTE starts with higher ratio they both seems to be similar with little variations. These methods also seem to be less accurate as imbalance level increased. SMOTE + ENN starts with higher ratio of precision and continues to maintain higher precision even after levels are increased and finally ADASYN tends to start with lower rate of precision compared to the other methods and continue to reduce as ratio distribution increases. In Figure 14(a,c) results of recall and f-measure are plotted, as analysing both the graphs it seems to be quite very similar to the precision attained before, even recall and f-measure tend to be have SMOTE + ENN as best performing technique.

Figure 14(d, e, f) illustrates the results of decision trees. Random oversampling starts with higher ratio of 0.98 precision but with drawbacks. Here SMOTE, SMOTE + Tomeklink and SMOTE + ENN all seems to start with similar rate of precision. When the ratio of distribution is however increased, SMOTE and SMOTE + ENN fail to continue the higher precision trend. Notice that SMOTE + ENN has good precision rate across different levels of ratios. ADASYN starts with lower rate compared to the other methods, but after two class imbalance ratios, it slowly increases but not higher than rest of the techniques. In Figure 14(d,f) is seen that regarding recall and f-measure outputs, all sampling techniques plots appears to be very similar to precision rate except ADASYN. Recall and F-measure in ADASYN start low with low class imbalance ratio ratio. As levels of class imbalance increase the recall F-measure also slightly increase.

Regarding classification results with K-NN Figure 14(g, h, i), it is possible to see that

SMOTE + ENN balancing technique starts at high precision compared to other technique. It does maintain that high accuracy across all class imbalance ratios. Random oversampling however seems to start equally with the SMOTE + ENN at ratio 10, but soon thereafter its precision rate reduces tremendously. SMOTE and SMOTE + Tomek precision rate are similar all along the trend line. It starts at high rate and reduces as a function of class imbalance. ADASYN starts with low precision and after ratio 30 it increases slowly. Figure 14(g,i) shows the results of recall and f-measure. These two plots are very similar to precision in all sampling techniques, with no change in trend line but having small variations in the values.

Comparing the all five sampling techniques SMOTE + ENN seems to be the clear winner. It returns the most propitious result on various ratio of class imbalance and works well on all three classification algorithms. The Random Forests algorithm produces better results compared to decision trees and K-NN. A deeper analysis shows Random Forest on all balancing techniques give fair results on highly imbalanced data (10 20), but nevertheless most techniques get lower accuracy when the class imbalance ratio increases. This counterintuitive behaviour shows that balancing algorithms produce lower quality datasets when the training dataset is already balanced.

6 Conclusion and Future Work

In recent years, imbalance data is becoming ubiquitous and serious issues have to be addressed to successfully extract meaningful information. A critical analysis of various classification techniques and sampling methods on twitter datasets, with different class imbalance ratios, show that SMOTE + ENN and Random Forest is the best combination to analyse spam on twitter data. Results obtained from this study may apply to classification in other highly imbalance datasets, such as the ones obtained from intrusion detection, fraud detection and rare event detection.

The current work can be extended in a number of ways. For instance, the analysed sampling techniques clean overlapping points that belong to different classes or borderline points. Synthetic examples could be created and validated at the same time. It is therefore interesting to explore the creation and verification steps for sampling techniques.

Acknowledgement

I would take this moment to thank my supervisor Dr Giovani Estrada for his guidance and support all through my research work with his patience, motivation, and immense knowledge.

References

Batista, G. E., Bazzan, A. L. and Monard, M. C. (2003). Balancing training data for automated annotation of keywords: a case study., *WOB*, pp. 10–18.

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. (2002). Smote:

synthetic minority over-sampling technique, *Journal of artificial intelligence research* **16**: 321–357.

Chen, C., Wang, Y., Zhang, J., Xiang, Y., Zhou, W. and Min, G. (2017). Statistical features-based real-time detection of drifted twitter spam, *IEEE Transactions on Information Forensics and Security* **12**(4): 914–925.

Chen, C., Zhang, J., Chen, X., Xiang, Y. and Zhou, W. (2015). 6 million spam tweets: A large ground truth for timely twitter spam detection, *Communications (ICC), 2015 IEEE International Conference on*, IEEE, pp. 7065–7070.

Crisci, C., Ghattas, B. and Perera, G. (2012). A review of supervised machine learning algorithms and their applications to ecological data, *Ecological Modelling* **240**: 113–122.

Elhassan, T., Aljurf, M., Al-Mohanna, F. and Shoukri, M. (2016). Classification of imbalance data using tomek link (t-link) combined with random under-sampling (rus) as a data reduction method, *Journal of Informatics and Data Mining* .

Guo, D. and Chen, C. (2014). Detecting non-personal and spam users on geo-tagged twitter network, *Transactions in GIS* **18**(3): 370–384.

Han, H., Wang, W.-Y. and Mao, B.-H. (2005). Borderline-smote: a new over-sampling method in imbalanced data sets learning, *Advances in intelligent computing* pp. 878–887.

He, H., Bai, Y., Garcia, E. A. and Li, S. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning, *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, IEEE, pp. 1322–1328.

He, H. and Garcia, E. A. (2009). Learning from imbalanced data, *IEEE Transactions on knowledge and data engineering* **21**(9): 1263–1284.

Hirve, S. and Kamble, S. (2016). Twitter spam detection, *International Journal of Engineering Science* **2807**.

Liu, S., Wang, Y., Zhang, J., Chen, C. and Xiang, Y. (2017). Addressing the class imbalance problem in twitter spam detection using ensemble learning, *Computers & Security* **69**: 35–49.

Liu, X.-Y., Wu, J. and Zhou, Z.-H. (2009). Exploratory undersampling for class-imbalance learning, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **39**(2): 539–550.

Meda, C., Bisio, F., Gastaldo, P. and Zunino, R. (2014). A machine learning approach for twitter spammers detection, *Security Technology (ICCST), 2014 International Carnahan Conference on*, IEEE, pp. 1–6.

Miller, Z., Dickinson, B., Deitrick, W., Hu, W. and Wang, A. H. (2014). Twitter spammer detection using data stream clustering, *Information Sciences* **260**: 64–73.

More, A. (2016). Survey of resampling techniques for improving classification performance in unbalanced datasets, *arXiv preprint arXiv:1608.06048* .

Walimbe, R. (2017). Handling imbalanced dataset in supervised learning using family of SMOTE algorithm., `https://www.datasciencecentral.com/profiles/blogs/handling-imbalanced-data-sets-in-supervised-learning-using-family/`. [Online; accessed 10-December-2017].

Wang, A. H. (2010). Don't follow me: Spam detection in twitter, *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, IEEE, pp. 1–10.

Wilson, D. L. (1972). Asymptotic properties of nearest neighbor rules using edited data, *IEEE Transactions on Systems, Man, and Cybernetics* **2**(3): 408–421.