Handling Imbalanced Data set: A Case Study for Binary Class Problems

by Addo Danquah Richmond, Master of Marketing Research

A Thesis Submitted in Partial
Fulfillment of the Requirements
for the Degree of
Master of Science
in the field of Mathematics

Advisory Committee:

Dr. Beidi Qiang, Chair

Dr. Song Foh Chew

Dr. Andrew Neath

Graduate School
Southern Illinois University Edwardsville
May, 2020

ABSTRACT

HANDLING IMBALANCED DATA SET: A CASE STUDY FOR BINARY CLASS PROBLEMS

by

ADDO DANQUAH RICHMOND

Chairperson: Professor Beidi Qiang

This study focused on handling imbalance data with a particular focus on performance measures and synthetic oversampling techniques - SMOTE (Synthetic Minority Oversampling Technique) and ADASYN (Adaptive Synthetic). Because accuracy, precision and recall performance measures could be misleading especially when dealing with imbalance data set, this study focused on using the F1-score and the Area Under the Curve (AUC) score as the reported performance measures. While SMOTE considers a uniform weight in generating new synthetic data for all minority points, ADASYN considers a density distribution in deciding the number of synthetic samples to be generated for a particular minority data point.

The classifier learning algorithms used in this study reported a much better F1-score and AUC score after the application of SMOTE or ADASYN on the imbalance data sets compared to the performance measures (F1-score and AUC score) on the imbalance data sets.

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to Dr. Beidi Qiang for her support, guidance, insightful comments, encouragement and thoughtful questions which incited me to widen my research from various perspectives.

Again, I would like to thank my thesis committee members; Dr. Chew Song and Dr. Andrew Neath, for their immense contributions in ensuring I produce quality research work.

Also, a big thank you to my academic advisor; Dr. Chew Song for being an inspiration to me through out my course of study. A big thank you to Musenbrock Annette for all you did for me. Finally, a special thank you to my wife, Judith Addo Danquah and my son, Othniel Addo Danquah for your support, love and most importantly, your patience as I worked on my thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 Outline

In the 1990's as more data and applications of machine learning and data mining started to become prevalent, an important challenge emerged: how to achieve desired classification accuracy when dealing with data that had significantly skewed class distributions (Sun et al.,2009; He & Garcia, 2009; Lopez et al., 2013; Branco et al., 2016; Cieslak et al., 2012; Hoens et al., 2012b; Hoens Chawla, 2013; Lemaitre et al., 2017; Khan et al., 2018). A data set is said to be imbalanced, if sample from one class is in higher number than other [9] [10]. In an imbalance data set, the class having more number of instances is called the major class while the one having relatively less number of instances is called, the minor class [10]. Machine Learning algorithms tend to produce unsatisfactory classifiers when faced with imbalanced data sets. They tend to be biased towards classes which have the greater number of instances. The algorithm tend to show bias in predicting the majority class data, while ignoring the minority class. The effect of modeling or predicting using an imbalance data set can cause a greater damage than good.Take for example, in a financial institution where there is critical need to identify fraud transaction which are mostly rare. Any errors in detecting a fraudulent transaction causes a major financial blow to the company. The same applies to the medical diagnosis field where certain health conditions are rare and therefore physicians or doctors cannot afford to incorrectly diagnose a patient. The effect of such incorrect diagnosis could be extremely dangerous for the patient. It is for this and many other reasons why a good classification model should be able to achieve a higher prediction rate on both the majority and minority classes (rare occurrence).

The significance of this area of research continues to grow across all sectors in the economy and this is largely driven by the challenging problem statements from different

application areas such as face recognition, software engineering, social media, social networks, sports, politics and health industry for medical diagnosis [**21**]. Class imbalance can be addressed in two ways. One is to assign distinct costs to training examples (Pazzani, Merz, Murphy, Ali, Hume, Brunk, 1994; Domingos, 1999). The other is to re-sample the original data set, either by oversampling the minority class and/or under-sampling the majority class (Kubat Matwin,1997; Japkowicz, 2000; Lewis Catlett, 1994; Ling Li, 1998). This study will focus on re-sampling the original data set by oversampling the minority class.

1.2 Description of the research

In this study, we present a liberal overview on handling imbalance data sets with particular focus on performance measures and re-sampling methodologies. We will focus on re-sampling of the original data set using oversampling techniques, SMOTE (Synthetic Minority Oversampling Technique), and its extension ADASYN (Adaptive Synthetic) technique. This study used three different data set with different sample sizes and features to test the learning abilities of the re-sampling algorithms, SMOTE and ADASYN. We will use Logistic Regression, Support Vector Machine (SVM), Random Forest and XGBoost as our classification methods or classifiers and compare these methods before and after applying the re-sampling techniques, SMOTE and ADASYN.

Chapter 2 gives an overview of performance measures; Accuracy, Recall, Precision, F1-Score and Area Under the Curve (AUC). Chapter 3 presents the details of our re-sampling techniques, SMOTE and ADASYN. Chapter 4 presents data analysis results comparing our two oversampling techniques to each other using the classification methods. Chapter 5 concludes the study and suggests directions for future work.

CHAPTER 2

Performance Measures

The efficiency of any machine learning model is determined using performance measures such as True Positive Rate, False Positive Rate, True Negative Rate and False Negative Rate [13]. Since this study focused on binary-class problems, we will consider classification accuracy, precision, sensitivity/recall, F1-score and Area Under the Curve (AUC) score as our performance measures on the models.

In a two-class problem, we are often looking to discriminate between observations with a specific outcome, from normal observations. Therefore, each of the observation data points belongs to one of the possible four outcomes as shown in Figure 2.1:

- True positive (TP) is the proportion of positive cases that were correctly identified.
- True negative (TN) is defined as the proportion of negatives cases that were correctly classified.
- False positive (FP) is the proportion of negative cases that were incorrectly classified as positive. This is also known as Type 1 error.
- False negative (FN) is the proportion of positive cases that were incorrectly classified as negative. This is also known as Type 2 error.

2.1 Classification Accuracy

Accuracy is the most common evaluation metric for most traditional application. But accuracy is a great performance measure only when you have a symmetric or balanced data set between the classes. It is therefore recommended that we consider other measurements when evaluating model performance. From Figure 2.1, accuracy can be calculated

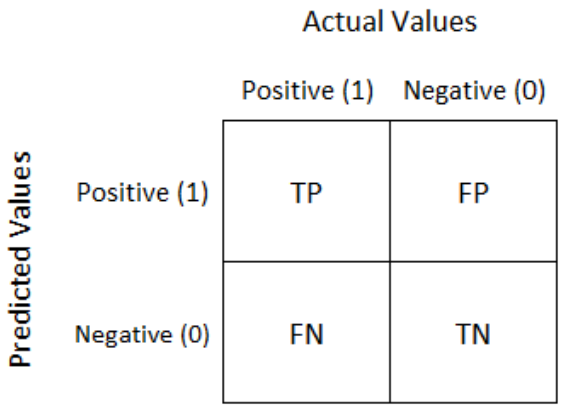

Figure 2.1: The Confusion Matrix

mathematically as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2.2 Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. In simple term, Precision is the ability of the classifier not to label an instance that is actually negative. Also called Positive Predictive Value (PPV), it can be described as a measure of a classifiers exactness. Mathematically,

$$Precision = \frac{TP}{TP + FP}$$

2.3 Recall/Sensitivity

Recall/Sensitivity is a measure of a classifiers completeness. Recall can be described as the ration of the number of positive predictions and the number of positive class values, that is, the ability of the classifier to find all positive instance. Mathematically,

$$Recall = \frac{TP}{TP + FN}$$

2.4 F1-Score

F1-score is particularly very useful compared to accuracy especially when dealing with asymmetric or imbalance data set. This is because it takes into account the weighted harmonic mean of both the classifiers exactness (Precision) and completeness (Recall/Sensitivity). Mathematically,

$$F1 - Score = \frac{2 * (Recall * Precision)}{Recall + Precision}$$

2.5 AUC (Area Under the Receiver Operator Characteristic (ROC) Curve)

It is the fraction of the total area that lies under the ROC graph. The ROC is used to compute an overall measure of quality while AUC provides a single value for performance evaluation of classifier. It can also be used as an evaluation measure for the imbalance data sets [**27**] [**28**]. AUC measure can be computed as;

$$AUC = \frac{1 + TP - FP}{2}$$

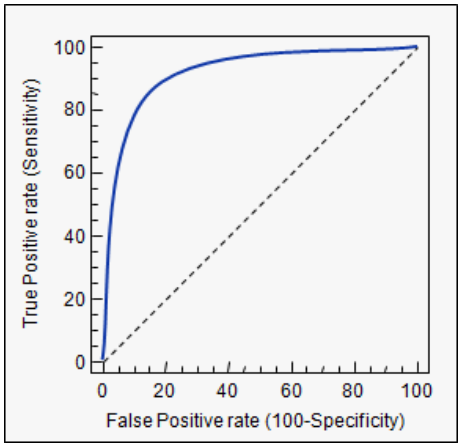

Figure 2.2: Example of a ROC plot. Two classifier curves are depicted: the dashed line represents a random classifier, whereas the solid line is a classifier which is better in overall performance quality than the random classifier.

CHAPTER 3

Oversampling Technique

For many years, there has been an extended research on the use of oversampling techniques in solving class imbalance problems. The extended research on this technique can be attributed to its ability to retain the original data set while preventing the loss of important information. Some of these researches on oversampling technique were developed by notable researchers such as (Douzas Bação, 2018), (Last, Douzas, Bacao, 2017), (Nekooeimhr Lai-Yuen, 2016), (Li, Fong , Wong, Mohagmmed, Fiaidhi , 2016), (Sun, Song , Zhu, Xu, Zhou, 2015), (Menardi Torelli, 2014) and (Bowyer, Hall, Kegelmeyer, Chawla, 2002). This technique creates a balanced data set by generating new samples to be added to the minority class. Oversampling can be done either through random oversampling where the data set is balanced through replicating the existing minority class or through synthetic oversampling where the data set is balanced through creating new synthetic minority samples by linear interpolation. The focus of this study will be on the application of synthetic oversampling in handling class imbalance because this method unlike random oversampling avoids over fitting, therefore improving the generalization ability of the classifier.

3.1 Synthetic Minority Oversampling Technique (SMOTE)

This approach, inspired by a technique that proved successful in handwritten character recognition (Ha Bunke, 1997), Synthetic Minority Oversampling Technique (SMOTE) was proposed by Chawla in 2002 [**22**]. In SMOTE algorithm, minority class is over sampled by generating synthetic examples rather than by oversampling with replacement for simple random oversampling. To avoid the issue of over fitting when increasing minority class regions, SMOTE creates new instances by working within the current feature space. New instance values are extracted from interpolation, so the original data set still has

significance. SMOTE interpolates values using a K - nearest neighboring technique for each minority class instance and generates attribute values for new data instances [**27**].

For each minority data, a new synthetic data instance is generated by taking the difference between the feature vector and its nearest neighbor belonging to the same class, multiplying it by a random number between 0 and 1 and then adding it to the minority instance. This creates a random line segment between every pair of existing features. This results in a new instance generated within the data set [13]. The cycle is replicated for the remaining minority instance [**28**].

One such downside stems from the fact that SMOTE arbitrarily tries to over-sample a minority instance with a uniform likelihood. Although this helps the approach to efficiently counter imbalances between the classes, the problems of disparity within the class and small disjoints are overlooked. Input areas that report multiple minority populations are highly likely to be further inflated, while sparsely populated minority areas are likely to remain sparse (Prati et al., 2004). This increases the complexity of the problem and lowers the learning classifier efficiency. Another downside is that SMOTE will intensify the noise present in the data further. This is likely to happen when a noisy minority sample, which is situated between instances of the majority class, and its nearest minority neighbor interpolates linearly. The approach is susceptible to noise generation, as it does not differentiate between overlapping class regions and so-called protected areas (Bunkhumpornpat et al., 2009). This impedes the classifier's ability to define the problems boundaries [**36**].

An imbalanced class distribution (LEFT) as shown in fig 3.1 contains more blue colors than orange colors. Using SMOTE, the algorithm finds the K-nearest neighbour of a data point in the minority class (orange colors) and creates some synthetic data points on the lines joining the primary point and the neighbors as shown in fig 3.1(middle). These new neighbors synthetic data points generated share similar characteristics of the

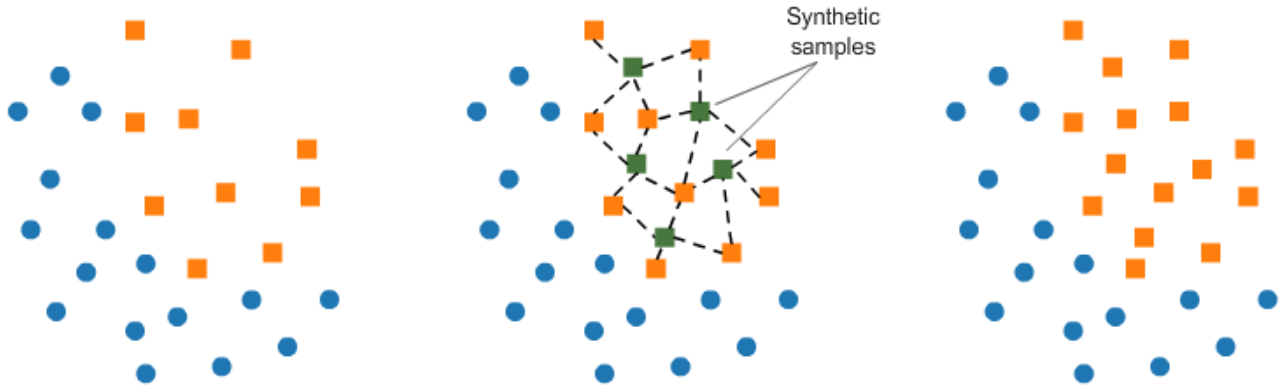

Figure 3.1: An illustration on how to create a Synthetic data in SMOTE Algorithm.

other minority data points. Theses synthetic data points now help balance the original class distribution (right), which improves the model's generalization ability. The use of SMOTE for class distribution balancing ensures no loss of data, alleviates over fitting and the process is very easy to implement. However, SMOTE should be used with extra care when dealing with higher dimensional space. Even though SMOTE performs well on low-dimensional data it is not effective in the high-dimensional setting for the classifiers.

SMOTE has been widely adopted by researchers and practitioners, likely due to its simplicity and added value with respect to random oversampling. From the original SMOTE algorithm, many other SMOTE-based algorithms, which aim to eliminate its disadvantages and improve performance on imbalanced learning has been developed. Some of these modifications include Borderline-SMOTE (Han et al., 2005), AHC(Cohen et al., 2006), ADASYN (He et al., 2008), Safe-Level-SMOTE (Bunkhumpornpat et al., 2009), DBSMOTE (Bunkhumpornpat et al., 2012), MWMOTE (Barua et al., 2014), ROSE (Menardi Torelli, 2014) and MDO (Abdi Hashemi, 2016). What distinguishes these SMOTE extensions from one another is based on how the synthetic data points are generated. While the original SMOTE method used K-nearest neighbors, these extensions uses method such as clustering, weighted distribution, border approach, mahalanobis distance and boost strapping in generating the synthetic minority points. Below is a table

of how the SMOTE algorithm works to generate new synthetic data points;

Algorithm 1 SMOTE

Input:

Let x_1, x_2,...,x_n be the minority class feature vectors in the n dimensional space of X

Let N be the number of synthetic instances to generate

Let K be the number of nearest neighbour

Output:

Synthetic set of artificial instances

1. For i in range (N) do,
2. Select randomly a minority class feature vector x_i
3. From x_i's K-nearest minority class neighbors, randomly select a neighbor $\hat{x}_i$
4. diff = $\hat{x}_i$ - x_i

δ =random number between 0 and 1

5.new Sample= x_i + diff * δ

6. Synthetic $\Longleftarrow$new Sample

endfor

To better appreciate how SMOTE works, we created 10 fictitious imbalance class data with three(3) minority class and seven (7) majority class as shown in Table 3.1. We then used this imbalance class to create two (2) minority synthetic data points by using the SMOTE algorithm steps as shown in the table above. For the purpose of illustration, we will set K to be 2.

From the above, we have manually calculated 2 synthetic data points (4.5 3) and (5 2.5) by using the steps in the SMOTE algorithm. This increased our minority class from

Algorithm 2 Example for SMOTE

Input:

Let N = 2

Let K = 2

Output:

Synthetic set of artificial instances

1. For i in range (N=1) do,

2. Select (4 3)

3. Randomly select a neighbor (5 3)

4. diff = (5 3) - (4 3) = (1 0)

$\delta = 0.5$

5.new Sample=(4 3) + [(1 0) * 0.5]

6.$Synthetic1 \Longleftarrow$(4.5 3)

1. for i in range (N=2) do,

2.Select (5 2)

3.Randomly select a neighbor (5 3)

4.diff = (5 3) - (5 2) = (0 1)

5.new Sample=(5 2) + [(0 1) * 0.5]

6.$Synthetic2 \Longleftarrow$(5 2.5)

endfor

three (3) to five (5), therefore improving the imbalance ratio significantly. We then plot the new synthetic data, (4.5 3) and (5 2.5) together with the original data set as shown in Figure 3.2.

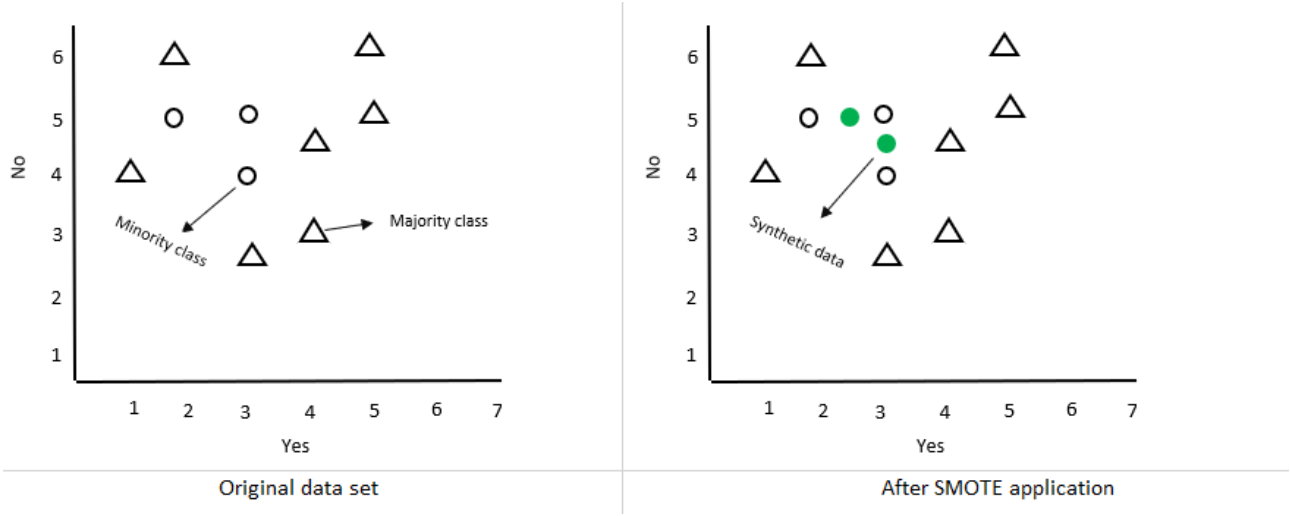

Figure 3.2: A schematic diagram of the class data before and after the application of SMOTE algorithm.

Table 3.1: Example of class imbalance data set

No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
5	4	5	2	1	3	4	4	5	5
3	3	2	6	4	2.5	3	4.5	5	6

3.2 Adaptive Synthetic (ADASYN) Sampling Approach

Adaptive Synthetic sampling approach is an extension or improvement of the SMOTE algorithm. The concept of Adaptive Synthetic (ADASYN) sampling approach for imbalanced learning was first introduced by He (2008). ADASYN tries to generate more synthetic instances on the region with less positive instances than one with more positive instances to increase the recognition of positive. This algorithm uses the number of negative neighbors in K-nearest neighbors of each positive instance to form a distribution function. The distribution function determines how many synthetic instances are generated from that positive instance [**29**]. ADASYN is based on the idea of adaptively generating minority data samples according to their distributions: more synthetic data is generated for minority class samples that are harder to learn compared to those minority samples that are easier to learn. Ultimately, ADASYN is a pseudo-probabilistic algorithm in the sense that a fixed number instances is generated for each minority instance based on a weighted distribution of its neighbors [**30**] [**31**]. Using SMOTE, each minority examples has equal chance to be selected for synthetic process but ADASYN uses the density distribution as a criterion to automatically evaluate the number of synthetic samples to be produced for each example of minority data. Therefore, ADASYN approach improves data distribution learning by reducing the bias generated by the class disparity and moving the classification decision boundary to the challenging examples (He et al, 2008). Because ADASYN is very sensitive to outliers, it is advisable to deal with outliers during data preprocessing before applying ADASYN procedure. Compared to SMOTE, ADASYN put more focus on the minority samples that are difficult to learn by generating more synthetic data points for these difficult and hard to learn minority class samples [**36**]. Compared to SMOTE where there is a uniform weight in generating new synthetic data for all minority points, ADASYN considers a density distribution in deciding the number of synthetic samples to be generated for a particular minority data point.

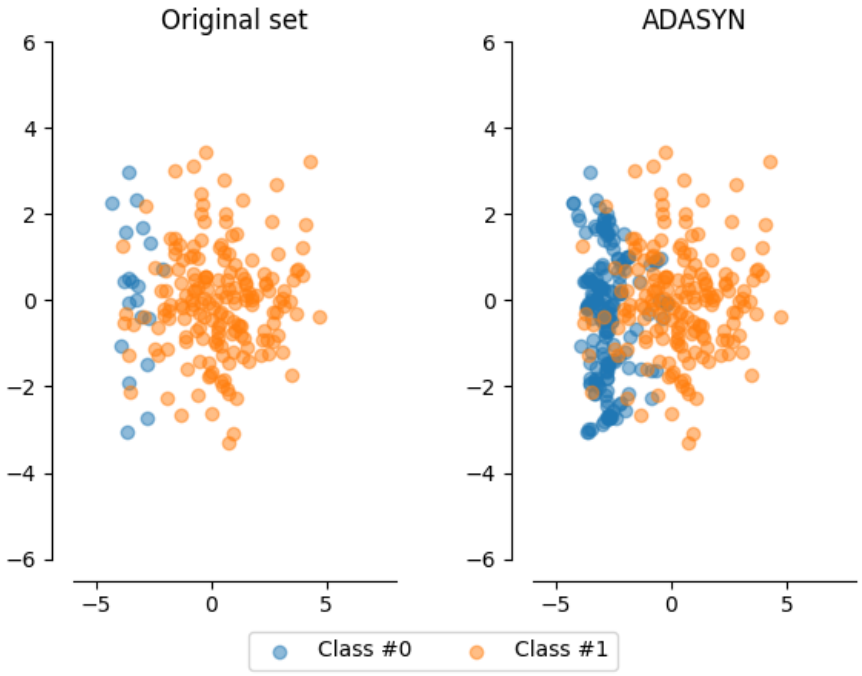

Figure 3.3: Schematic diagram of ADASYN algorithm application.

As shown in fig 3.3, the original data set has some minority class examples that are difficult for an algorithm to learn. ADASYN synthetically generates new minority samples using a density distribution based on the number of out-of-class neighbors. A minority instance surrounded by more out-of-class instances is considered hard-to-train, and is thus given a higher probability to be augmented by generating synthetic samples. The resulting data set post ADASYN, will not only provide a balanced representation of the data distribution but it will also force the learning algorithm to focus on those difficult to learn examples [**30**]. The table below shows the pseudo algorithm for ADASYN.

We again manually computed new synthetic minority data points using the ADASYN algorithm using the same data set used in SMOTE application. We then used this imbalance class to create three (3) minority synthetic data points by using the ADASYN algorithm steps as shown in the table above. For the purpose of illustration, we will set K

Algorithm 3 ADASYN

Input:

Let m be the number of minority samples

Let n be the number of majority samples

Let β be the ratio of the balance level of the synthetic samples. NB: $\beta \in (0, 1]$

Let x_i for i=1,2,3...m be the minority class feature vectors in the n dimensional space of X

Let G be the number of synthetic instances to generate

Let g_i for i=1,2,3...m be the number of synthetic data generated for each x_i

Let K be the Number of Nearest Neighbour

Let $\delta \in [0, 1]$

---

to be 2 as well.

From the table below, we have manually calculated 3 synthetic data points (4.5 3), (5 2.5) and (4.5 2.5) by using the steps in the ADASYN algorithm. This increased our minority class from three (3) to six (6), therefore improving the imbalance ratio significantly. We then plot the new synthetic data, (4.5 3), (5 2.5) and (4.5 2.5) together with the original data set as shown in Figure 3.4.

Algorithm 4 Example for ADASYN

Input:

Let m = 3, n = 7, K = 2, $\beta = 0.75$ $\delta = 0.5$

Output:

1. $G = (7 - 3) * 0.75 = 3$

2. ** There is only 1 majority class in each of the distinct neighbourhood

i. $r_i = 1/2$ for $i = 1$ to 3

ii. $\sum_{i=1}^{3} r_i$=1/2 + 1/2 + 1/2 = 3/2

3. $\hat{r}_i = 1/2$ * 2/3 = 1/3 for $i = 1$ to 3

4. $g_i = 1/3 * 3 = 1$ for $i = 1$ to 3

5. From (4 3) 2-nearest minority class neighbors, randomly select a neighbor (5 3)

i. diff = (5 3) - (4 3)

ii.$newSample_{ij}$ = (4 3) + [(1 0) * 0.5]

iii. Synthetic $\Longleftarrow$(4.5 3)

6. From (5 3) 2-nearest minority class neighbors, randomly select a neighbor (5 2)

i.diff=(5 2)−(5 3)

ii.new Sample_{ij} = (5 3) + [(0 -1) * 0.5]

iii. Synthetic $\Longleftarrow$(5 2.5)

7. From (5 2) 2-nearest minority class neighbors, randomly select a neighbor (4 3)

i.diff=(4 3)−(5 2)

ii.new Sample_{ij} = (5 2)+ [(-1 1) * 0.5]

iii. Synthetic $\Longleftarrow$(4.5 2.5)

endfor

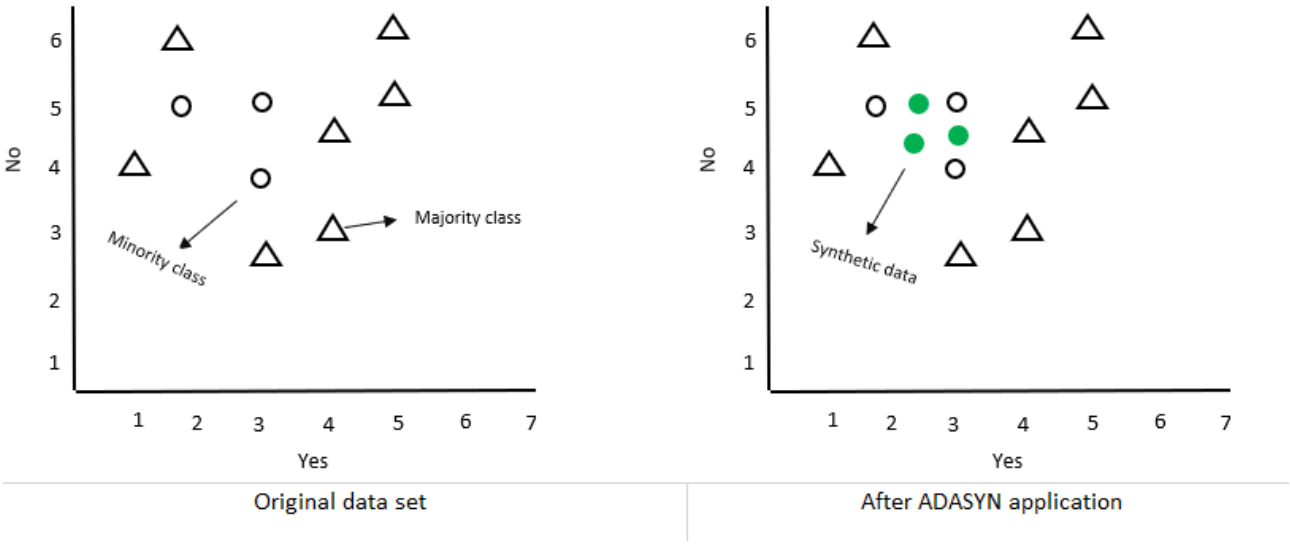

Figure 3.4: A schematic diagram of the class data before and after the application of ADASYN algorithm.

CHAPTER 4

Experiments

4.1 Data set Summary

Using a real world data set obtained from Kaggle[**1**] and UCI Machine Learning Repository [**2**], we test our SMOTE and ADASYN algorithm. The purpose of this study is to test the learning capabilities of these algorithm on two-class imbalance data sets. Table 4.1 shows a brief description of the data sets used in this study.

Table 4.1: Description of data sets

Data Set	Attributes	Sample Size	Minority	Majority
Blood Transfusion Service Center	5	748	178	570
Pima Indians Diabetes	8	768	268	500
IBM HR Analytics Employee Attrition	35	1470	237	1233

4.2 Blood Transfusion Service Center

This data set originates from the Blood Transfusion Service Center in Hsin-Chu City in Taiwan. The objective of this data set is to use certain measurements to predict whether a person will donate blood. The data set has 5 variables or features with 748 data examples. There are 178 minority class and 570 majority class. This data set is a two-class imbalance data set and hence no modification was needed.

4.3 Pima Diabetes data set

This data set is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective is to predict based on diagnostic measurements whether a

patient has diabetes [**5**]. There are 768 data examples with 8 attributes. This data set is a two class imbalance data set with a binary outcome, 0 (not diabetic) and 1 (diabetic).

4.4 IBM HR Analytics Employee Attrition

This fictitious data set was created by IBM data scientists for the purpose of uncovering the factors that lead to employee attrition. There are 1470 data sample with 237 minority class and 1233 majority class.

4.5 Data Pre-processing

All of the three different data set were checked for missing values, fortunately there were none for all of the data sets. Certain features were converted from categorical values to numerical values to boost performance measures. The outcome variables for IBM data set were re-coded from a nominal binary outcome to a numerical binary outcome. For the purpose of been able to interpret the output and to avoid information loss, we did not perform any dimensional reduction technique especially to the IBM HR Analytics Employee Attrition data set. Again, some of the classifiers used in this study inherently deal with dimension reduction while learning the data. All of our data set were standardized to provide a common ground for all the classifiers even though some classifiers can still perform well whether the data is standardized or not. The data set were split into 80% train data and 20% test data. The classifiers first learned using the train data and the test data was used to measure the classifiers learning abilities.

4.6 Classification methods

This study made use of 4 classification classifiers to test the learning abilities of the algorithms on all of the data sets. These classifiers included;

4.6.1 Logistic Regression

In contrast to Linear Regression where the dependent variable is a continuous variable, Logistic Regression is used to examine the association of (categorical or continuous) independent variable(s) with one dichotomous dependent variable. Logistic regression analyzes the relationship between multiple independent variables and a categorical dependent variable and estimates the probability of occurrence of an event by fitting data to a logistic curve [**6**]. Although Logistic Regression is a widely used technique because it is efficient, we cannot solve non-linear problems since its decision surface is linear. We have two types of logistic regression namely Binary and Multinomial Logistic Regression. For the purpose of this study, we will focus on Binary Logistic Regression.

4.6.2 Support Vector Machine

Support Vector machine (SVM) is a supervised learning model that analyze data used for classification. Support Vector machine (SVM) classifier is capable of doing linear classifications as well as non-linear classifications using kernel tricks. Support Vector Machine classifier is one of the most important learning algorithms that has been used for decades. Developing SVM model for prediction is a matter of experimenting with the choice of inputs to find the set of inputs that produce a model with the lowest prediction error [**7**]. SVM is really effective in higher dimensions but SVM algorithm is not suitable for larger data sets and performs poorly when the data set has much noise.

4.6.3 Random Forest

Random Forest is one of the best known classifiers; it is very simple and effective. Random Forest classifier is a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. In other words, Random Forest is an ensemble of randomized decision trees where each decision tree gives a vote for the prediction of target variable.

Then the Random forest chooses the prediction that gets the most vote. Random forest classifier is very efficient on both small and large data sets and works well with missing data while giving a better predictive accuracy [**35**].

4.6.4 XGBoost method

XGBoost is an efficient and scalable implementation of gradient boosting framework by (Friedman, 2001) (Friedman et al., 2000). It is one of the most popular and efficient implementations of the Gradient Boosted Trees algorithm that is based on function approximation by optimizing specific loss functions as well as applying several regularization techniques. XGBoost is actually a library that is used for developing fast and high performance. Although XGBoost is highly efficient and versatile compared to other classification methods, XGBoost can only work on numeric features and can lead to over fitting if hyper parameters are not tuned properly.

4.7 Experimental Results

Combined with SMOTE and ADASYN, 4 classifiers; Logistic Regression, Support Vector Machine (SVM), Random Forest and XGBoost are used as the learning models in our experiment. Without applying SMOTE and ADASYN algorithms, we created baseline models using the original data sets and provided the performances using each of the classifiers mentioned. The performance measures as discussed in chapter 2 are used

to illustrate the performances of these learning techniques and we compared the learning ability of the classifiers before and after performing SMOTE and ADASYN. The following tables below contain the experimental results. The best classifier for each of the three applications (baseline, SMOTE and ADASYN) as well as the best performance measures in terms of F1-score and AUC score are highlighted with bold text.

Table 4.2: Blood Transfusion Service Center Data Set

Original Data Set					
Classifier	Accuracy	Precision	Recall	F1-score	AUC
Logistic Regression	0.73	0.57	0.10	0.20	0.54
SVM	0.74	0.67	0.10	0.20	0.54
Random Forest	0.71	0.45	0.32	0.37	0.59
XGBoost	0.75	0.58	0.37	**0.45**	**0.63**
SMOTE					
Logistic Regression	0.75	0.74	0.79	0.77	0.75
SVM	0.76	0.79	0.74	0.76	0.76
Random Forest	0.80	0.82	0.78	0.80	0.80
XGBoost	0.80	0.81	0.81	**0.81**	**0.80**
ADASYN					
Logistic Regression	0.70	0.75	0.71	0.73	0.70
SVM	0.66	0.72	0.63	0.67	0.66
Random Forest	0.74	0.78	0.75	**0.77**	**0.74**
XGBoost	0.72	0.75	0.74	0.75	0.71

Table 4.3: Pima Diabetes Data Set

Original data set					
Classifier	Accuracy	Precision	Recall	F1-score	AUC
Logistic Regression	0.82	0.76	0.62	0.68	0.77
SVM	0.79	0.70	0.55	0.62	0.73
Random Forest	0.81	0.71	0.64	0.67	0.76
XGBoost	0.82	0.70	0.70	**0.70**	**0.79**
SMOTE					
Logistic Regression	0.81	0.78	0.82	0.80	0.81
SVM	0.85	0.83	0.87	**0.85**	**0.86**
Random Forest	0.85	0.83	0.87	**0.85**	**0.86**
XGBoost	0.85	0.82	0.88	0.85	0.85
ADASYN					
Logistic Regression	0.72	0.73	0.71	0.72	0.72
SVM	0.82	0.79	0.87	0.83	0.82
Random Forest	0.85	0.82	0.90	**0.86**	**0.85**
XGBoost	0.82	0.78	0.89	0.83	0.82

4.8 Discussion

This study although reported all of the five (5) performance measures, the focus will be on the last two (2) performance metrics; F1-score and AUC score because, these are better scores to consider when dealing with imbalance data set relative to accuracy, recall and precision. From the experimental results below, overall, there were significant improvement in the model performances in terms of F1-score and AUC score for each of the data set after applying SMOTE and ADASYN. As shown in Table 4.2 and Table 4.4, the baseline models reported low recall scores for all the classifiers while the accuracy and

Table 4.4: IBM HR Analytics Employee Attrition data set

Original Data Set					
Classifier	Accuracy	Precision	Recall	F1-score	AUC
Logistic Regression	0.89	0.83	0.41	**0.55**	**0.70**
SVM	0.86	0.90	0.20	0.31	0.59
Random Forest	0.84	0.71	0.10	0.18	0.55
XGBoost	0.87	0.75	0.31	0.43	0.64
SMOTE					
Logistic Regression	0.87	0.90	0.85	0.88	0.87
SVM	0.88	0.93	0.84	0.88	0.88
Random Forest	0.91	0.93	0.90	**0.91**	**0.91**
XGBoost	0.89	0.92	0.86	0.89	0.89
ADASYN					
Logistic Regression	0.87	0.89	0.84	0.86	0.87
SVM	0.89	0.91	0.86	0.88	0.88
Random Forest	0.91	0.92	0.90	**0.91**	**0.91**
XGBoost	0.89	0.91	0.86	0.88	0.88

precision scores were significantly better. The reason for low recall scores in both tables, Table 4.2 and Table 4.4, is due to low imbalance ratio. The Blood Transfusion Service Center data set (Table 4.2) had an imbalance ratio of 0.3 while the IBM HR Analytics Employee Attrition data set (Table 4.4) had an imbalance ratio of 0.19. While Table 4.3 (Pima Diabetes data set) recall scores were significantly better than those in Table 4.2 and Table 4.4, imbalance ratio was also higher, 0.54 compared to 0.19 (Table 4.4) and 0.30 (Table 4.2). This is an indication of how sensitive classifiers like Logistic, SVM and Random Forest are to class imbalance data sets. In this study, in most cases, XGBoost

classifier performed better in terms of F1-score and AUC score for the baseline models compared to Random Forest classifier due to it's high versatility. On the other hand, Random Forest performed better (Table 4.3 and Table 4.4) or similar (Table 4.2) than XGBoost classifier in terms of F1-score and AUC score after handling the class imbalance. Below are the ROC curve graphs showing the AUC score of each classifier for the baseline models and the models after applying SMOTE and ADASYN.

As seen in Figure 4.1, 4.2 and 4.3, there were improvement in the classifiers ability to distinguish between the minority and majority classes after applying SMOTE or ADASYN to the imbalance class data set. Random Forest and XGBoost were the best classifiers in distinguishing between the classes (80%) after applying SMOTE application to the Blood Transfusion Service Center data set. With the Pima Diabetes data set, SVM and Random Forest classifiers after applying SMOTE to the imbalance class data set gave the best measure of separability. That is, Random Forest and SVM were able to distinguish between the minority and majority classes 86% of the time after SMOTE application. Random Forest classifier on the IBM HR Analytics Employee Attrition data set reported an AUC score of 91% after applying both SMOTE and ADASYN.

In general, F1-score and AUC score tend to improve after dealing with the respective imbalance data set using SMOTE or ADASYN. However, choosing the best combination of sampling technique and classifier is essential to handling the problem of imbalance data set in the best possible way.

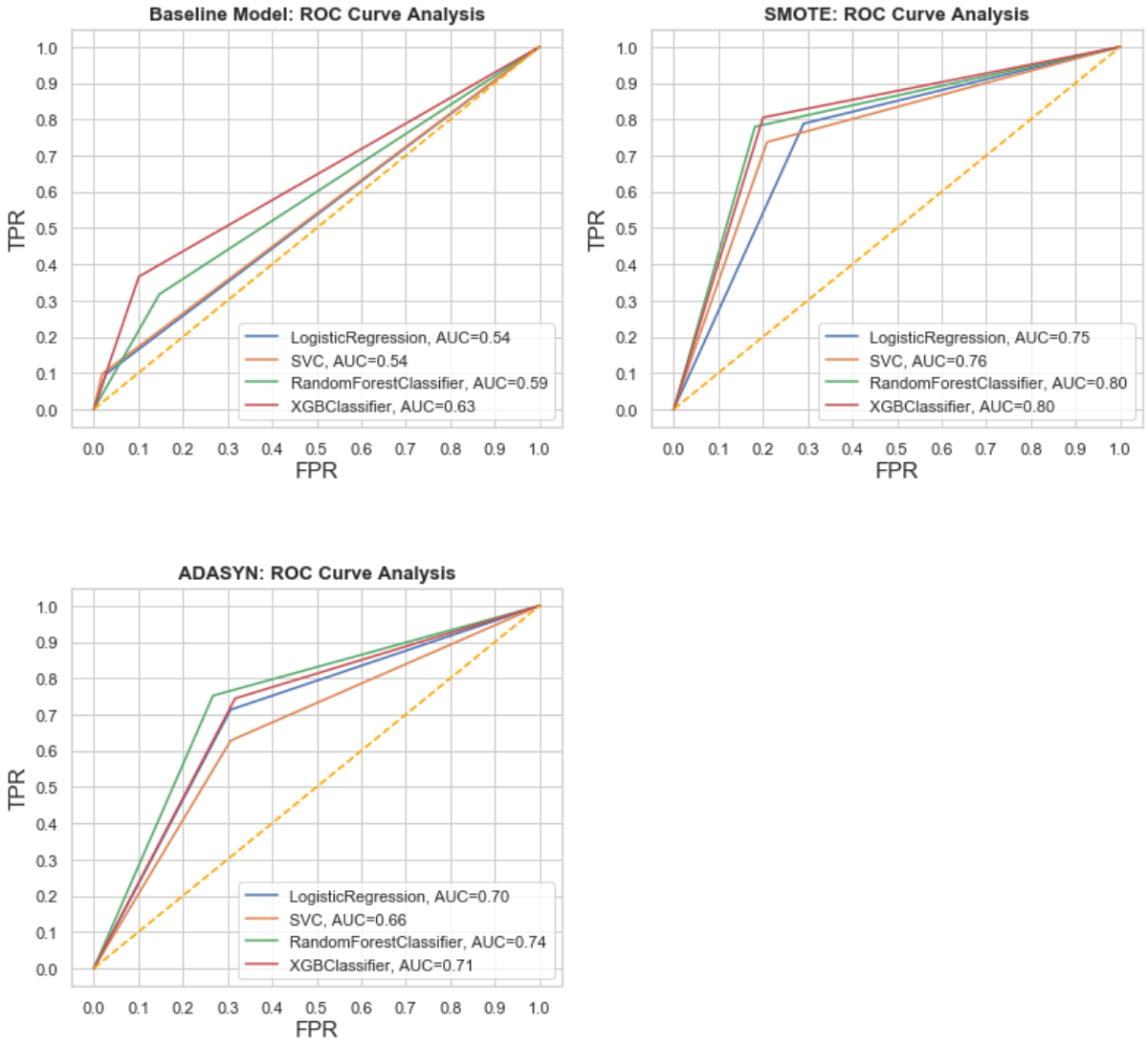

Figure 4.1: Blood Transfusion Service Center data set

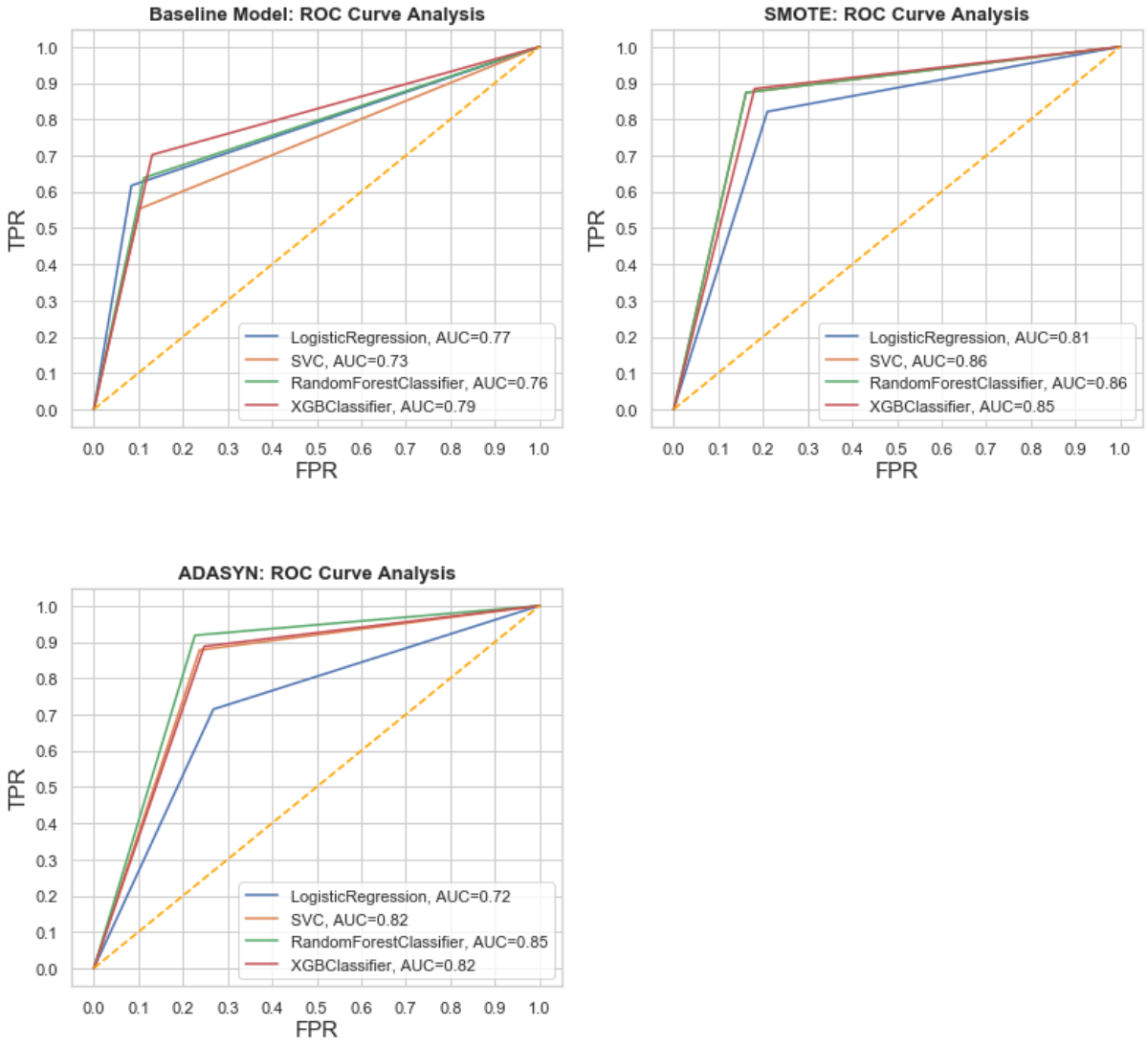

Figure 4.2: Pima Diabetes data set

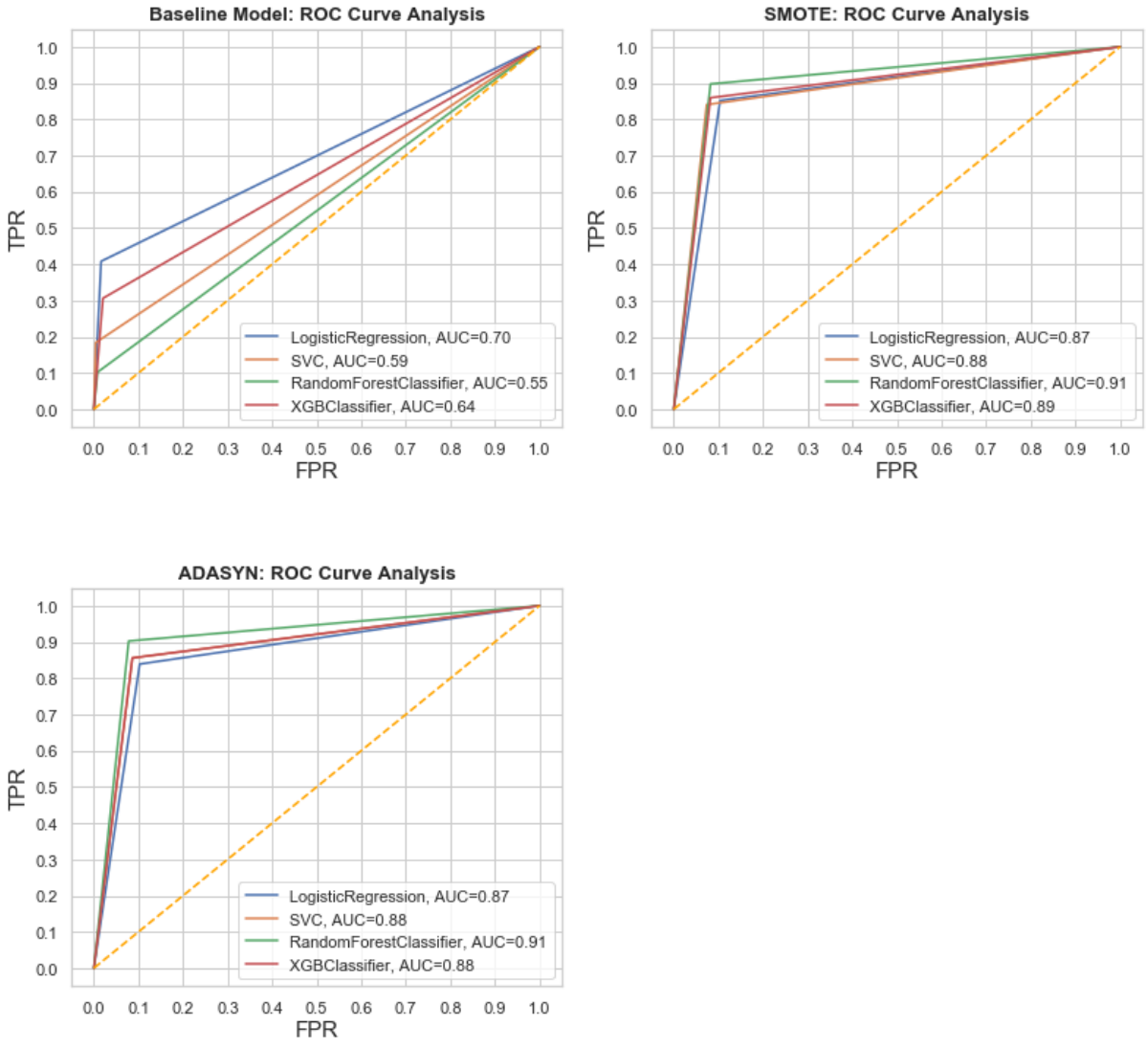

Figure 4.3: IBM HR Analytics Employee Attrition data set

CHAPTER 5

Conclusion

In this study, we presented a liberal overview of the problem of handling imbalanced data sets with particular focus on performance measures and re-sampling methodologies. Although we proposed five (5) performance measures, we focused on F1-score and the AUC score in our reporting. The reason been that, these two measures represent the best performance measures when dealing with imbalance class data set. We also focused on the SMOTE and ADASYN re-sampling methodologies. We discussed that, compared to SMOTE, ADASYN put more focus on the minority samples that are difficult to learn by generating more synthetic data points for these difficult and hard to learn minority class samples. We also discussed that while SMOTE considers a uniform weight in generating new synthetic data for all minority points, ADASYN considers a density distribution in deciding the number of synthetic samples to be generated for a particular minority data point. In this study, while both SMOTE and ADASYN when applied to the imbalance class data set, resulted in an improvement in the performance measures (F1-score and AUC score), SMOTE generally reported higher scores than ADASYN. However, there are not enough evidence to generalize based on this study that, SMOTE performs better than ADASYN in handling class imbalance. Factors not limited to the type of classifier, parameter tuning and the type of imbalance class data set can definitely affect the performances of SMOTE and ADASYN.

For future research, we will focus on how the combination of the oversampling techniques- SMOTE and ADASYN and classifiers perform on a test data that has different class ratios from the train data set.

REFERENCES

[1] https://www.kaggle.com/uciml/pima-indians-diabetes-database

[2] https://archive.ics.uci.edu/ml/index.php.

[3] ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning, Haibo He, Yang Bai, Edwardo A. Garcia and Shutao Li.

[4] https://data.world/data-society/pima-indians-diabetes-database

[5] Aida Ali, Siti Mariyam Shamsuddin, and Anca L. Ralescu,Classification with classimbalance problem:A Review

[6] Habib, Mona Soliman. Improving scalability of support vector machines for biomedicalnamed entity recognition. University of Colorado at Colorado Springs, 2008.

[7] Rokach, Lior Maimon, Oded. (2005). Decision Trees. 10.1007/0-387-25465-X9.

[8] https://towardsdatascience.com/methods-for-dealing-with-imbalanced-data-5b761be45a18.

[9] Shuo Wang, Member, and Xin Yao, "Multiclass Imbalance Problems: Analysis and Potential Solutions", IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 42, No. 4, August 2012.

[10] Nitesh V. Chawla, Nathalie Japkowicz, Aleksander Ko lcz "Special Issue on Learning from Imbalanced Data Sets "Volume 6, Issue 1 - Page 1-6.

[11] Aida Ali, Siti Mariyam Shamsuddin, and Anca L. Ralescu,Classification with class imbalance problem:A Review

[12] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. J Artif Int Res. 2002;16(1):321–357.

[13] S. Deepa and V. Bharathi, "Textural Feature Extraction and Classification of Mammogram Images using CCCM and PNN," IOSR Journal of Computer Engineering (IOSR-JCE), vol. 10, no. 6, pp. 07-13, June 2013.

[14] Guo, Xinjian Yin, Yilong Dong, Cailing Yang, Gongping Zhou, Guangtong. (2008). On the Class Imbalance Problem. Fourth International Conference on Natural Computation, ICNC '08. Vol. 4. 10.1109/ICNC.2008.871.

[15] A Deep Analysis of the Precision Formula for Imbalanced Class Distribution-Gabriel Kofi Armah, Guangchun Luo, and Ke Qin

[16] J. Soni, U. Ansari, D. Sharma and S. Soni, "Predictive data mining for medical diagnosis: An overview of heart disease prediction," International Journal of Computer Applications, vol. 17, no. 8, pp. 43-48, 2011

[17] S. Gupta, D. Kumar and A. Sharma, "Data mining classification techniques applied for breast cancer diagnosis and prognosis," Indian Journal of Computer Science and Engineering, vol. 2, no. 2, pp. 188-193, 2011.

[18] M. Kumari and S. Godara, "Comparative Study of Data Mining Classification Methods in Cardiovascular Disease Prediction," International Journal of Computer Science and Technology (IJCST), vol. 2, no. 2, pp. 304-308, June 2011.

[19] I. Y. Khan, P. Zope and S. Suralkar, "Importance of Artificial Neural Network in Medical Diagnosis disease like acute nephritis disease and heart disease," International Journal of Engineering Science and Innovative Technology (IJESIT), vol. 2, no. 2, pp. 210-217, 2013.

[20] R. Batuwita, V. Palade, AGm: a new performance measure for class imbalance learning. application to bioinformatics problems, in: Proceedings of the 8th International Conference on Machine Learning and Applications (ICMLA 2009), 2009, pp. 545–550.

[21] SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary; Alberto Fernandez Salvador,Francisco Herrera,Nitesh V. Chawla,Journal of Artificial Intelligence Research 61 (2018) 863-905

[22] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16(1):321–357, 2002.

[23] Classification of Imbalanced Data Using Synthetic Over-Sampling Techniques;Peng Jun Huang

[24] Andrew Estabrooks, Taeho Jo, and Nathalie Japkowicz. A multiple resampling method for learning from imbalanced data sets. Computational Intelligence, 20(1):18–36, 2004.

[25] Vaishali Ganganwar. An overview of classification algorithms for imbalanced datasets. International Journal of Emerging Technology and Advanced Engineering, 2(4):42–47, 2012.

[26] Thien M Ha and Horst Bunke. Off-line, handwritten numeral recognition by perturbation method. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 19(5):535–539, 1997.

[27] Barua, S., Islam, M. M., Murase, K. (2015). GOS-IL: A generalized over-sampling based online imbalanced learning framework. In Neural Information Processing - 22nd International Conference (ICONIP), pp. 680–687.

[28] S. Maheshwari, J. Agrawal, and S. Sharma, "New approach for classification of highly imbalanced datasets using evolutionary algorithms," Int. J. Sci. Eng. Res., vol. 2, no. 7, pp. 1–5, 2011.

[29] K. P. Satyasree and J. Murthy, "An exhaustive literature review on class imbalance problem," Int. J. Emerg. Trends Technol. Comput. Sci., vol. 2, pp. 109–118, May 2013

[30] Adaptive neighbor synthetic minority oversampling technique under 1NN outcast handling,Wacharasak Siriseriwan, Krung Sinapiromsaran

[31] Haibo He, Yang Bai, E. A. Garcia and Shutao Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, 2008, pp. 1322-1328.

[32] Synthetic Sampling for Multi-Class Malignancy Prediction,Matthew Yung, Eli T. Brown, Alexander Rasin, Jacob D. Furst, Daniela S. Raicu

[33] https://ieeexplore.ieee.org/document/5128907

[34] https://snlpatel0012134.wixsite.com/thinking-machine/single-post/SMOTE-Synthetic-Minority-Over-sampling-Technique

[35] Automatic Determination of Neighborhood Size in SMOTE

[36] https://www.stat.berkeley.edu/ breiman/RandomForests

[37] Oversampling for Imbalanced Learning Based on K-Means and SMOTE,Felix Last,Georgios Douzas,Fernando Bacao

[38] Nekooeimehr, I. and Lai-Yuen, S. K. (2016). Adaptive semi-unsupervised weighted oversampling (a-suwo) for imbalanced datasets. Expert Systems with Applications, 46:405–416.