Journal of Electronics and Informatics (2020)
Vol.02/ No. 02
Pages: 96-136
http://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.2.004

Navo Minority Over-sampling Technique (NMOTe): A Consistent Performance Booster on Imbalanced Datasets

Navoneel Chakrabarty
Jalpaiguri Government Engineering College,
Jalpaiguri, India
E-mail: nc2012@cse.jgec.ac.in

Sanket Biswas
Autonomous University of Barcelona,
Spain.
E-mail: sanket.biswas@e-campus.uab.cat

Abstract: Imbalanced data refers to a problem in machine learning where there exists unequal distribution of instances for each classes. Performing a classification task on such data can often turn bias in favour of the majority class. The bias gets multiplied in cases of high dimensional data. To settle this problem, there exists many real-world data mining techniques like over-sampling and under-sampling, which can reduce the Data Imbalance. Synthetic Minority Oversampling Technique (SMOTe) provided one such state-of-the-art and popular solution to tackle class imbalancing, even on high-dimensional data platform. In this work, a novel and consistent oversampling algorithm has been proposed that can further enhance the performance of classification, especially on binary imbalanced datasets. It has been named as NMOTe (Navo Minority Oversampling Technique), an upgraded and superior alternative to the existing techniques. A critical analysis and comprehensive overview on the literature has been done to get a deeper insight into the problem statements and nurturing the need to obtain the most optimal solution. The performance of NMOTe on some standard datasets has been established in this work to get a statistical understanding on why it has edged the existing state-of-the-art to become the most robust technique for solving the two-class data imbalance problem.

Keywords: imbalanced data; machine learning; classification; data mining; over-sampling; under-sampling; SMOTe; NMOTe.

I. Introduction

With the proliferating growth and expansion of computational power, the availability of more and more raw data has steered the surge of machine learning, as it emerged from a fledgling science to an applied technology, bringing a new revolution in the world of industry, scientific research and business development [1]. As more and more real world applications and open source frameworks unfolded, it has resulted in the development of novel and more efficient machine learning approaches which can beat the existing state-of-the-art methods [2]. Consequently, real world datasets tend to have their own structure and distribution. In dealing with these problems, classifiers are often faced with imbalanced data, which suggests that one of the destination classes contain much less number of instances than the other classes. The 'class imbalance problem' is one such affair that has caused a deep concern among machine learning practitioners for more than a decade now. It is indeed a serious obstacle for learning classifiers as they are biased toward the majority class and tend to mis-classify the minor instances as major ones. These real life applications in data mining include sentiment analysis [3], text mining [4], target detection [5], industrial systems monitoring [6] and so on.

In all these problems, the correct recognition of minority class plays a key role. Several data-driven approaches have been introduced in recent times which help to transform the original set of instances through re-sampling to change class distribution. Although such specialized techniques have been proposed but the identification of conditions for the efficient usage of those methods still remain an open problem to all data mining experts.The degradation of classification performance can be understood by examining the nature of more fundamental complexities of data distributions which has already been established in some related works [7] [8] [9] [10] [11].

In this section, light is thrown on some of the specialized approaches that have been taken to find a solution to the class imbalance problem. Most of these works can be categorized into four different elements:

- Sampling-based methods
- Cost-based methods
- Kernel-based methods
- Active learning-based methods

Again these methods can be further classified under 3 branches:

- Data-level approaches which modify the collection of instances in the training set to balance the distribution among classes.
- Algorithm-level approaches which directly reconstruct the existing learning algorithms and mitigate the biases towards majority groups.
- Hybrid approaches which precisely combine the advantages of both Data-level and Algorithm-level approaches.

In sampling methods, the size of the classes is modified. Two kinds of sampling techniques exist in the literature:

Under-sampling

This refers to reducing the majority class samples to reduce data imbalance. This can be done either randomly (called random under-sampling) or through an inference of some prior statistical knowledge (called informed under-sampling) [12]

Over-sampling

This method generates the minority class samples, which help to further add essential information to the original data that may enhance the performance of the classifier.

In cost-sensitive methods, unlike sampling, we do not try to modify the original data and instead try to derive the cost of mis-classification. If any minority sample gets mis-classified in majority class, then the cost matrix returns a value higher than that of the majority class sample [13].

Although the sampling based methods and cost-sensitive methods have been the most popular, studies have also been conducted on kernel based methods in order to tackle imbalanced data-sets.It is mainly used to optimize the performance of sampling approaches by creating ensembles of classifiers (like, over or under sampled SVMs) with asymmetric cost of mis-classification and help to overcome the problem of generalization.

Active-learning based methods are mainly applied to a data pool containing both labeled and unlabeled samples. To improve the performance of classification, active learning methods help to acquire labels for those unlabeled data samples. The learner then selects the unlabeled samples that lie within the vicinity of the decision boundary, and which are most uncertain. These data points with maximum confidence are then labelled manually [14].

An insight is thrown into some standard sampling approaches that have been made in recent years to find a solution to this data imbalance problem. Some of the previous state-of-the-art techniques have been discussed in brief and a critical analysis of these methods is sketched, based on their self-dependency and rigidity before introducing the novel technique to outshine them:

Chawla et al. [15] proposed a very powerful technique called Synthetic Minority Over-sampling Technique (SMOTe) for balancing imbalanced data-sets. It also investigated to conclude that a hybrid approach of both oversampling and under- sampling can actually exhibit better results, compared to methods involving only under-sampling the majority class. This algorithm was indeed the first successful stand-alone and rigid-in-formulation sampling approach to be accepted and used globally.

Journal of Electronics and Informatics (2020)
Vol.02/ No. 02
Pages: 96-136
http://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.2.004

Chawla et al. [17] then combined SMOTe and Boosting for designing a new approach called SMOTeBoost which augments the original data with artificial in- stances in minority classes by updating weights assigned to mis-classified examples and balancing the skewed distributions. They also experimented to show a marked improvement in prediction performance of learners on highly imbalanced data-sets. But deploying this model is not an independent approach as it combines boosting with stand-alone SMOTe and moreover, boosting performs poorly in case of noisy platform. So, SMOTeBoost, though effective in its performance in the end, but is a Classifter dependent Over-sampling Technique.

A few years later, Han et al. [16] then proposed two new synthetic minority over-sampling approaches called borderline-SMOTe1 and borderline-SMOTe2 in which the minority instances near the borderline were over-sampled for obtaining superior performance of model classifiers in terms of True Positive Rate (TPR) and F-value than SMOTe and random oversampling methods. Since these methods concentrate only on borderline examples and do not over-sample all the instances or any random subset of the target minority class, it proves to be far more efficient with respect to time. However there is no mention of the proper estimation of this borderline which can often lead to overlapping between two adjacent classes and affect the overall results, making them unclear and flexible in their formulation.

Wang et al. [21] proposed an ensemble sampling algorithm called SMOTeBagging and applied them for solving multi-class imbalanced datasets. Again, this approach is not stand-alone, depending on the Bagging Classifier. Moreover it deals with multi-class data-sets while our main focus is on two-class problem in this work. Chen et al. [22] proposed another adaptive novel technique called RAMOBoost (Ranked Minority Oversampling in Boosting) which ranks minority class instances at each learning iteration and adaptively shifts the decision boundary toward more such complicated majority and minority instances by using a hypothesis assessment procedure. Although this is a very decent approach, but it is not stand-alone in nature, depending on a Machine Learning Classifier (boosting).

Later, Dong et al. [18] proposed a new over-sampling technique called Random- SMOTe for learning from imbalanced data where the minority examples were sparsely distributed in sample space against an enormous amount of majority samples. They demonstrated Random-SMOTe to be more effective as compared to other random sampling approaches. Also after SMOTe, Random-SMOTe emerged as a highly-successful Stand-alone and Rigid Over-sampling Algorithm.

Seiffert et al. [23] proposed a faster alternative to the existing SMOTeBoost for enhancing learning performance from skewed data. They named it as RUSBoost (Random Under-sampling Boost). This is an under-sampling approach, which is again not Stand-alone as it involves Boosting.

Ertekin [20] devised a hybrid adaptive algorithm called VIRTUAL which combines the benefits of both oversampling and active learning methods. It has also been demonstrated that it outperforms other oversampling techniques on the grounds of generalization performance and computational complexity.

ISSN: 2582-3825 (online)

However, this approach is not rigid-in-formulation and is quite similar to the borderline approach already been proposed by Han et. al. [16].

Zheng et al. [19] highlighted some flaws of the standard SMOTe model and pro- posed a novel technique called Sigma Nearest Oversampling based on Convex Combination (SNOCC), where the number of seed samples were increased and ensured that the augmented samples were not enclosed within the line segment between two seed samples. They have also demonstrated experiments to showcase that SNOCC outperforms SMOTe and CBS (another SMOTe based algorithm). But deep-down in the formulation of SNOCC, there is no Rigidity as no mention is made, regarding the number of seed-samples to be taken.

Blagus et al. [25] applied classical SMOTe to largely imbalanced high-dimensional data. They also investigated the properties of SMOTe from theoritical and empirical point of view.

Li et al. [24] applied the oversampling method of Random-SMOTe on five different UCI imbalanced datasets and integrated it with Logit technique for achieving a significant improvement in the performance of classifiers.

Shuo Wang et al. [26] proposed two new resampling-based ensemble methods called OOB (Oversampling-based Online Bagging) and UOB (Undersampling-based Online Bagging) for the robust detection of minority class instances in online class imbalance learning.

2. Proposed Work

2.1 A new Over-sampling Method: Navo Minority Over-sampling Technique (NMOTe)

This is the new Over-sampling technique introduced in this paper. It is a technique based on nearest neighbours decided by euclidean distance between data-points in feature space. There is a percentage of Over-sampling which which indicates the number of synthetic samples to be created and this percentage parameter of Over- Sampling is always a multiple of 100. If the percentage of Over- Sampling is 100, then for each instance, a new sample will be created. Hence, the number of minority class instances will get doubled. Similarly if percentage of Over-Sampling is 200, then the number of minority class samples will get tripled. In NMOTe,

1) For each instance having minority label, k number of nearest neighbours are found such that they also belong to the same class where,

$$k = (\text{NMOTe}\%)/100$$

So, k number of euclidean distances are obtained for each minority instance.

Journal of Electronics and Informatics (2020)
Vol.02/ No. 02
Pages: 96-136
http://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.2.004

2) The k euclidean distances for each instance, are directly multiplied by a random variable between 0 and 1, excluding 0.

3) These k numeric distances, after being multiplied by the random number is added to all the features of the feature vector of every instance, so considered at each iteration.

The Pseudo Code for the Algorithm is given below as Algorithm 1.

- This is a 2-module-algorithm in which the module nearest neighbour() returns the additives for each considered instance, x belonging to the minority class.
- Number of features = n
- Original number of Minority Instances = N
- Minority Instances are stored in a (Nxn) matrix = X
- % of Over-Sampling = f (only multiple of 100)
- euclidean() is a function that calculates the Euclidean Distance between 2 points in Feature Space.
- sort() is a function that sorts a list.

```
1. nearest_neighbour(X,x,f)
2. {
3.   dis[N-1]
4.   additive[f/100]
5.   k=0
6    for j = 1 to N:
7.   {
8.     if X[j] != x:
9.     {
10.      dis[k] = euclidean(X[j],x)
11.       k = k + 1
12.    }
13. }
14. dis = sort(dis)
15. fraction = random(0,1)
16. while fraction == 0:

17. {
```

Journal of Electronics and Informatics (2020)
Vol.02/ No. 02
Pages: 96-136
http://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.2.004

```
18.   fraction = random(0,1)
19. }
20. additive = dis[: , f/100] * fraction
21. return additive
22.}

23. NMOTe(X,f)
24. {
25.   sampled instances[f*N/100][n]
26.   k = 0
27.   for i=1 to N:
28.   {
29.     additive = nearest neighbour(X, X[i], f)
30.     for j = 1 to f/100:
31.     {
32        sampled instances[k] = X[i] + additive[j]
33.       k=k+1
34.     }
35.   }
36.   return sampled instances
37. }
```

Algorithm 1. NMOTe

The NMOTe Algorithm is computationally efficient than SMOTe with less lines of code but takes longer computation time than R-SMOTe. But, on the basis of Performance Analysis, NMOTe is proved to outperform both SMOTe and R-SMOTe (in binary-class problems) in Experiments (Section 3), taking 2 binary data-sets as exemplars.

2.2 Experiment on Adult Census Data

The experiments are conducted for performance analysis of NMOTe against existing Stand-alone and Rigid Over-sampling algorithms, SMOTe and R-SMOTe. 2 binary data-sets are used for the purpose, having 2 different problem-statements and are discussed in Sections 3.1 and 3.2 respectively.

2.2.1 Dataset

The data-set was extracted from UCI-ML Repository [27]. It consists of 48,842 instances with 14 features and a binary label, income.

ISSN: 2582-3825 (online)

Journal of Electronics and Informatics (2020)
Vol.02/ No. 02
Pages: 96-136
http://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.2.004

2.2.2 Problem Statement

The binary label, Income depicts whether a person's income is greater than 50k dollars (=1) or not (=0). So, the problem statement is to Predict the Income

Level of a person from the Census Data that whether it is greater than 50k dollars or not.

2.2.3 Proposed Methodology

Tackling Missing Values

The missing values in the data are all present in categorical features.So, they are dealt using default pointer, ?.This serves as a separate category in categorical features and hence, prevents information loss.

Label Encoding of Categorical Features

Out of 14 attributes, 8 attributes are found to be categorical while 6 are found to be continuous as shown in Table 1. Label Encoding is done for all the 8 categorical features.

Feature Selection

The whole data-set is trained using Extra Trees Classifier. After training, every feature gets a feature importance score assigned by the Extra Trees Classifier. This is assigned depending upon the number of times that feature got selected as the best split for a decision tree, predicting the majority vote.

An Algorithmic Visualization of Extra Trees Classifier is given in Fig 1.

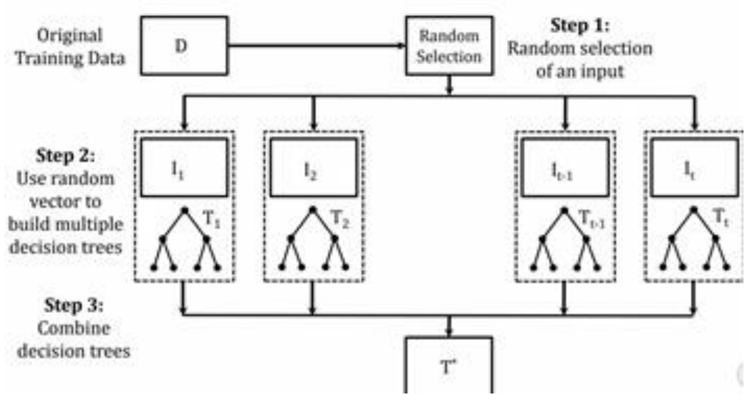
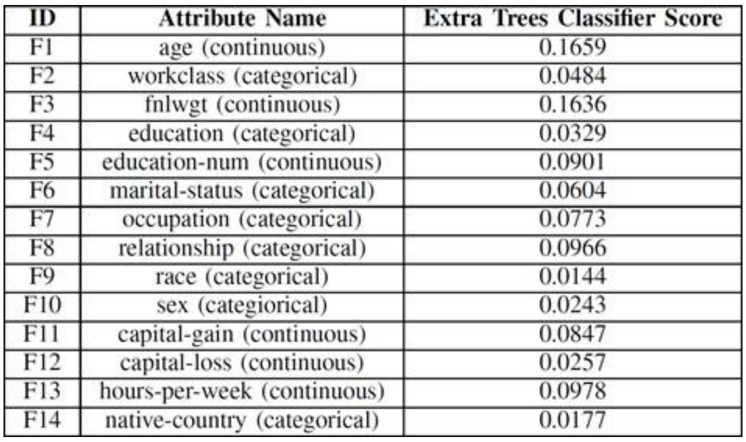

Original Training Data
D
Random Selection
Step 1: Random selection of an input
Step 2: Use random vector to build multiple decision trees
I_1
T_1
I_2
T_2
I_{t-1}
T_{t-1}
I_t
T_t
Step 3: Combine decision trees
T^*

Fig 1. Visual Representation of Extra Trees Classifier

All the features with their Extra Trees Classifier Feature Importance Scores are tabulated in Table 1.

ISSN: 2582-3825 (online)



ID	Attribute Name	Extra Trees Classifier Score
F1	age (continuous)	0.1659
F2	workclass (categorical)	0.0484
F3	fnlwgt (continuous)	0.1636
F4	education (categorical)	0.0329
F5	education-num (continuous)	0.0901
F6	marital-status (categorical)	0.0604
F7	occupation (categorical)	0.0773
F8	relationship (categorical)	0.0966
F9	race (categorical)	0.0144
F10	sex (categiorical)	0.0243
F11	capital-gain (continuous)	0.0847
F12	capital-loss (continuous)	0.0257
F13	hours-per-week (continuous)	0.0978
F14	native-country (categorical)	0.0177

Table 1. Features and Extra Trees Score

Features F9 and F14 are dropped as they are having least feature importance values.

One-Hot Encoding of Categorical Features

One-Hot Encoding is done for all the categorical features except feature, F10 as sex attribute has only 2 categories (male & female) present in the dataset.

Data Imbalance Removal or Data Balancing, Model Development, Training the Model and Results

The number of instances with label 0 have been 24720 and on the other hand, only 7841 samples of label 1 are there. Hence, there is a gross class-imbalance present in the dataset and is visually shown in Fig 2 as bar charts

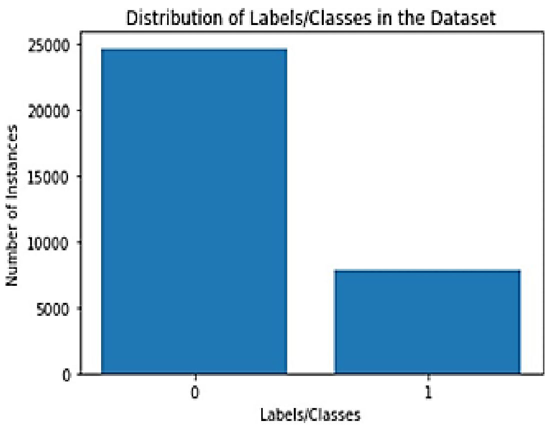


Fig 2. Bar Charts showing Class Distribution in the Income Dataset

ISSN: 2582-3825 (online)

Journal of Electronics and Informatics (2020)
Vol.02/ No. 02
Pages: 96-136
http://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.2.004

Now, Over-sampling of instances with label 1 i.e., minority instances need to be done. This Over-sampling is done in 3 strategies:

Over-sampling using SMOTe in a)
Over-sampling using R-SMOTe in b)
Over-sampling using NMOTe in c)
(a) Stage-wise Over-sampling using SMOTe: A stage-wise Over-sampling is done using SMOTe in which,

I. 100 % Over-sampling is the 1st stage: With 100% Over-sampling, the number of minority instances gets doubled, hence reducing Class Imbalance as shown in Fig 3.

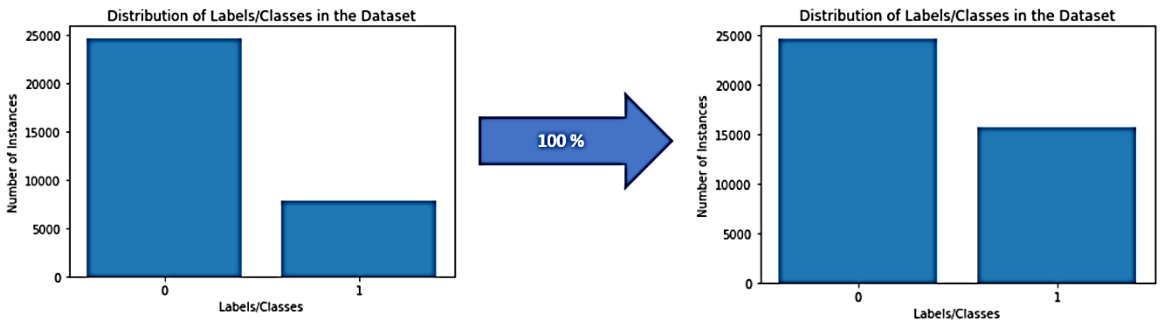


Fig 3. Class Imbalance Reduction after 100% Over-sampling

A. Now, the dataset is shuffled in a consistent way and split into Training and Validation Sets with 80% of the instances in the Training Set and 20% of the instances in the Validation Set.

B. Gradient Boosting Classifier is used as a Learning Algorithm to construct the Binary Prediction Model. It is an Ensemble and Boosting Algorithm in which series of Decision Trees are constructed and always the decision tree in the succeeding step corrects the error in the decision tree constructed in the preceding step.

The Algorithm for Gradient Boosting Classifier is given below:

```
Input: training set $Z=\{(x_1, y_1), ., (x_n, y_n)\}$
M: number of iterations
v: learning rate
1. $f_0(x)=\log(p_1/(1-p_1))$
2. for m=1...M:
3.  $g_i=dL(y_i, f_m(x_i))/df_m(x_i)$
4.  A tree, $h_m(x_i)$ is fit to target $g_i$
5.  $p_m=\text{argmax}_p Q[f_{m-1}(x)+p*h_m(x)]$
6.  $f_m(x)=f_{m-1}(x)+v*p_m*h_m(x_i)$
7. return $f_M$ (x)
```

Algorithm 4. Gradient Boosting Classifter

The GBC Model is tuned with Grid-Search. After tuning the model with Grid-Search, 350 estimators

Journal of Electronics and Informatics (2020)
Vol.02/ No. 02
Pages: 96-136
http://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.2.004

and maximum depth of 4 are obtained as the best set of hyper-parameters. The Grid-Search Summary is shown in Fig 4.

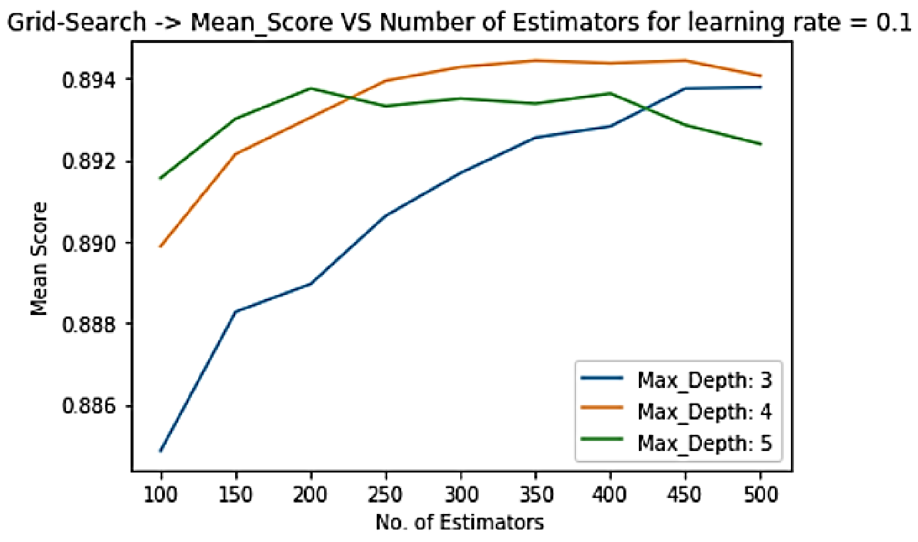


Fig 4. Grid Search Summary on Mean Score for the model after 100% SMOTe

C. Results

The model performance is evaluated on the following metrics

- The Training Accuracy describes the number of instances correctly predicted in the Training Set.
- The Validation Accuracy describes the number of instances correctly predicted in the Validation Set.
- The Sensitivity or Recall is defined as the fraction of correctly identified positives.

  Recall = TP/TP + FN

- Precision is defined as the proportion of correctly predicted positive observations of the total predicted positive observations.

  Precision = TP/TP + FP

- F1-Score is the Harmonic Mean of Recall and Precision.

ISSN: 2582-3825 (online)

- Area Under Receiver Operator Characteristic Curve (AUROC): ROC Curve is the plot of True Positive Rate vs False Positive Rate. An Area under ROC Curve of greater than 0.5, is accept- able. It is implemented using the decision function attribute of Gradient Boosting Classifier which returns values that give a measure of how far a data-point is away from the Decision Boundary from either side (negative value for opposite side). The corresponding ROC Curve is shown in Fig 5.

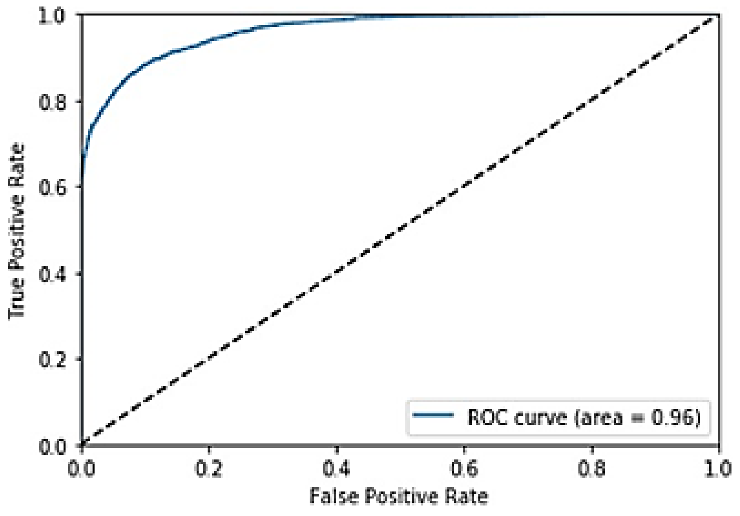


Fig 5. ROC Curve for the model after 100% SMOTe

- Confusion Matrix based on the Validation set for the model is shown in Fig 6.

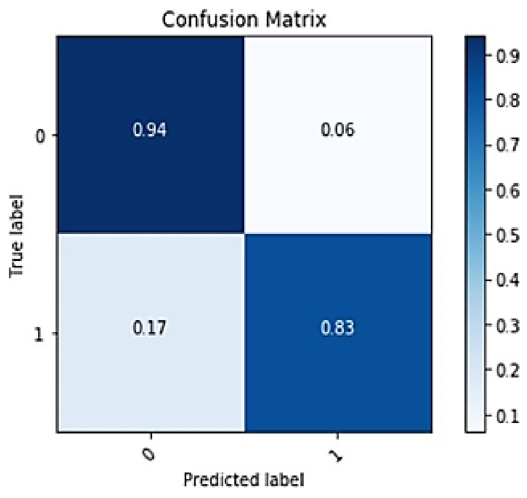


Fig 6. Normalized Confusion Matrix for the model after 100% SMOTe

All the results for the model after 100% SMOTe are shown in Table 4.

II. 200% Over-sampling is the 2nd stage
With 200% Over-Sampling, the number of minority instances gets tripled, hence reducing the Class Imbalance to a large extent as shown in Fig 7.

`

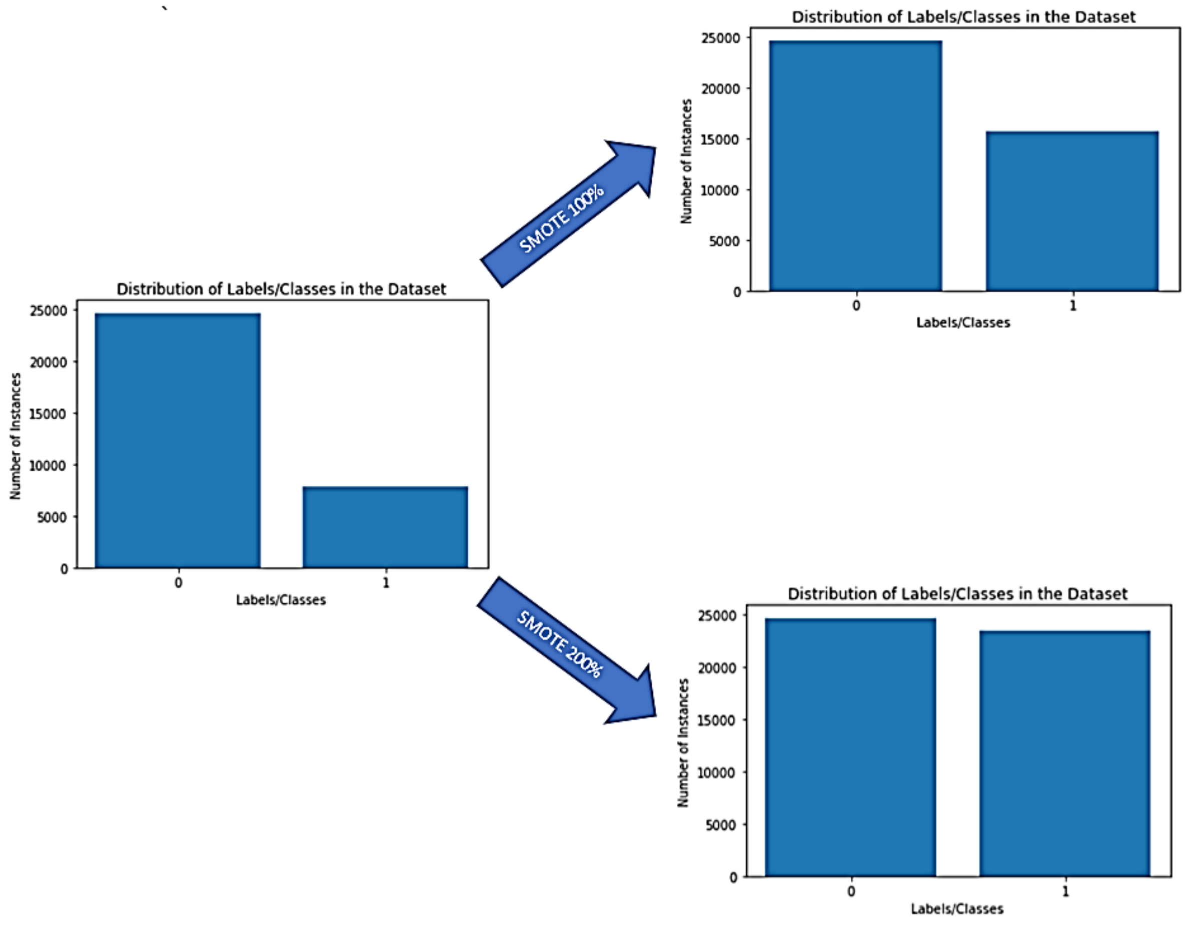


Fig 7. Reductions in Data Imbalance after 100% and 200% separately

A. The Dataset is shuffled and split as done previously in the model evaluation after 100% SMOTe.

B. Gradient Boosting Classifier is used as the Learning Algorithm.
The GBC Model is tuned with Grid-Search. After tuning the model with Grid-Search, 200 estimators and maximum depth of 5 are obtained as the best set of hyper-parameters. The summary of Grid-Search Tuning of GBC model on the basis of the Mean Score is shown in Fig 8.

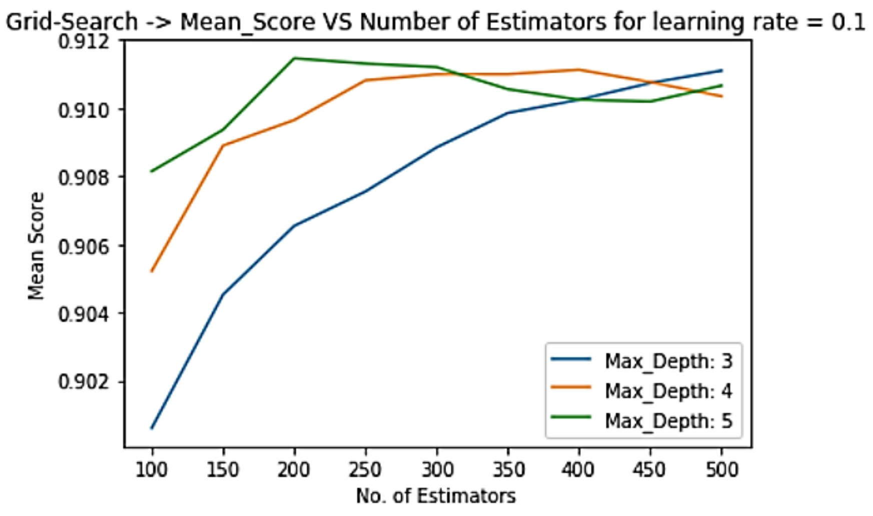


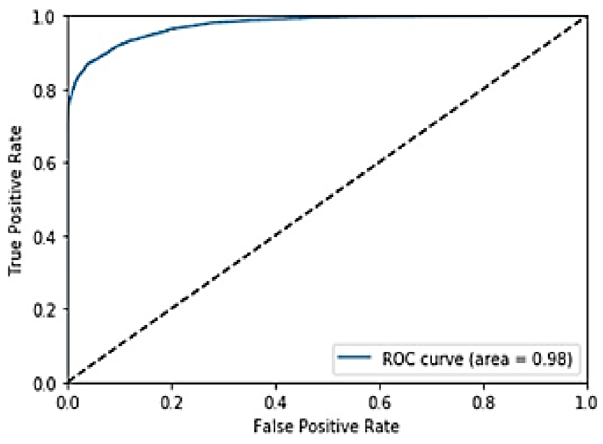
Fig 8. Grid Search Summary on Mean Score for the model after 200% SMOTe

C. Results

The model performance is evaluated on the same set of metrics as used previously in the model evaluation after 100% SMOTe:

- Training Accuracy
- Validation Accuracy
- Sensitivity or Recall
- Precision
- F1-Score

Area Under Receiver Operator Characteristic Curve (AUROC) The corresponding ROC Curve is shown in Fig 9.



ISSN: 2582-3825 (online)

Fig 9. ROC Curve for the model after 200% SMOTe

Confusion Matrix based on the Validation Set is shown in Fig 10.

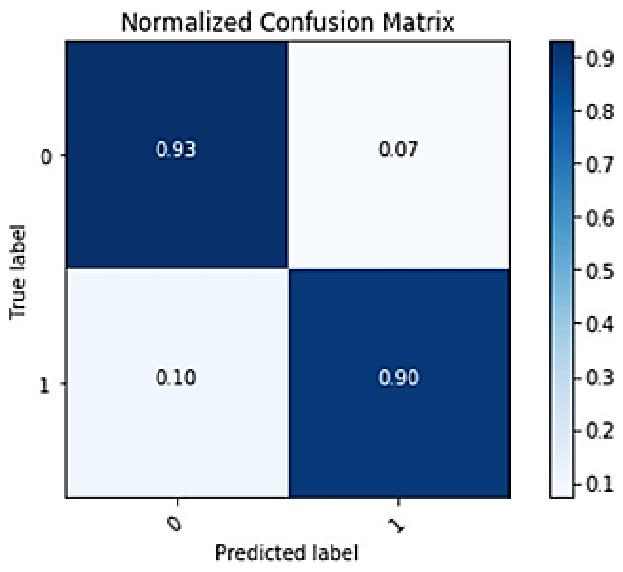


Fig 10. Normalized Confusion Matrix for the model after 200% SMOTe

All the results for the model after 200% SMOTe are shown in Table 7.

(b) Stage-wise Over-sampling using R-SMOTe: A stage-wise Over-sampling is done using R-SMOTe in which,

I. 100% Over-sampling in the 1st stage

A) Shuffling and 80-20 Splitting

B) Gradient Boosting Classifier is used as Learning Algorithm

The GBC Model is tuned using Grid-Search. After tuning, 450 estimators and maximum depth of 3 are found to be the best set of hyper-parameters. The summary of the Grid-Search Tuning of GBC on the basis of Mean Score is shown in Fig 11.

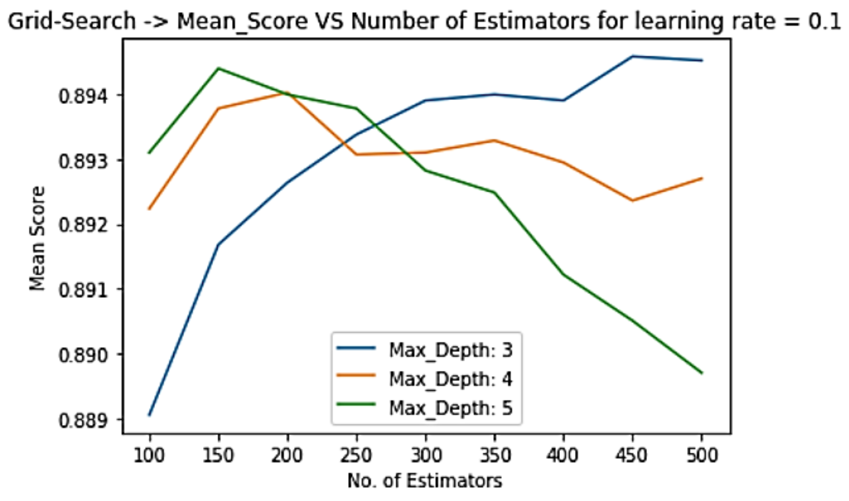


Fig 11. Grid Search Summary on Mean Score for the model after 100% R-SMOTe Results

C) The model performance is evaluated on the following metrics:

- Training Accuracy
- Validation Accuracy
- Sensitivity or Recall
- Precision
- F1-Score

- Area Under Receiver Operator Characteristic Curve (AUROC) The corresponding ROC Curve is shown in Fig 12.

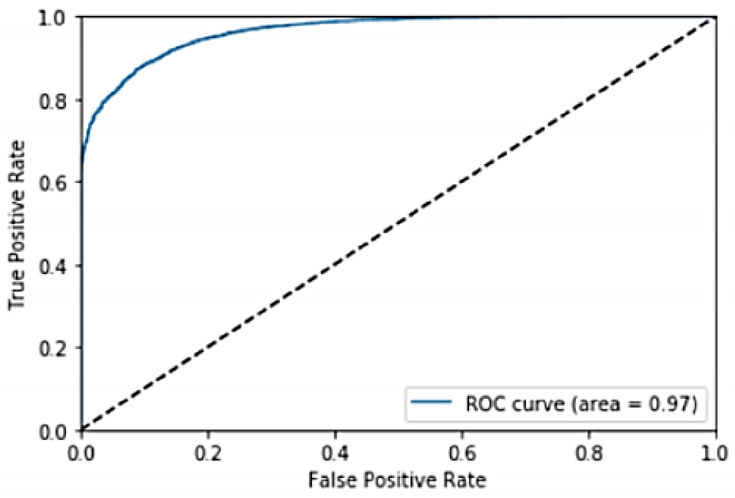


Fig 12. ROC Curve showing the Area under the Curve for the model after 100% R-SMOTe

➢ Confusion Matrix for the model is shown in Fig 13.

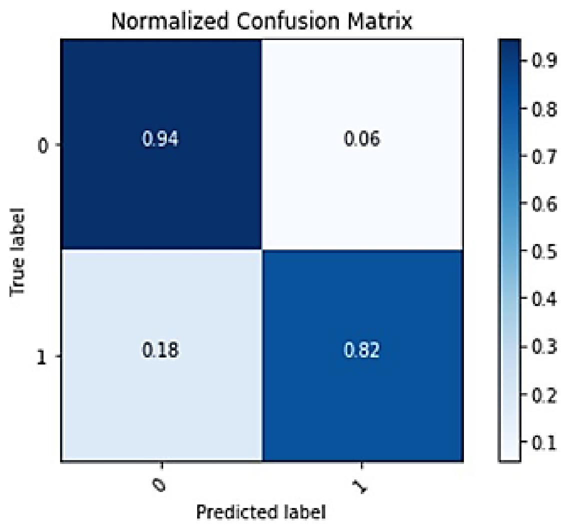


Fig 13. Normalized Confusion Matrix for the model after 100% R-SMOTe

All the results for the model after 100% R-SMOTe are shown in Table 4.

II. 200% Over-sampling is the 2nd stage

A) Shuffling and 80-20 Splitting

B) Gradient Boosting Classifier is used as Learning Algorithm

400 estimators and maximum depth of 3 are found to be the best set of hyper-parameters after applying Grid-Search on the GBC model. The Grid-Search summary is shown in Fig 14.

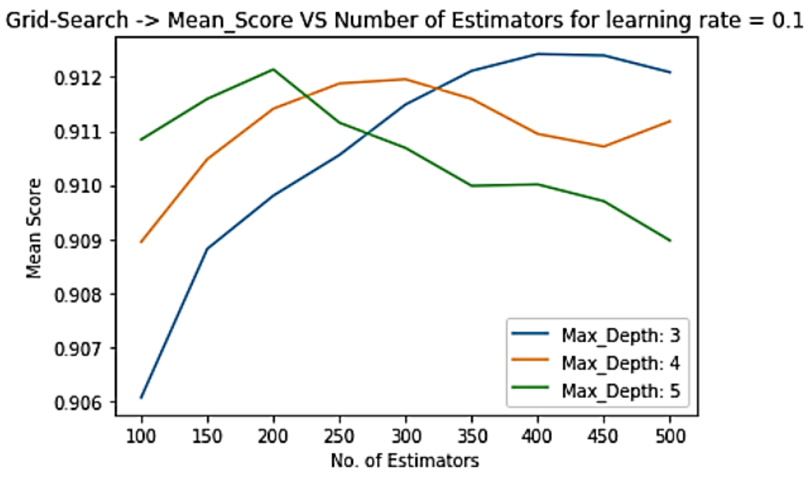


Fig 14. Grid-Search Summary on Mean Score for the model after 200% R-SMOTe

Journal of Electronics and Informatics (2020)
Vol.02/ No. 02
Pages: 96-136
http://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.2.004

C) Results

The model performance is evaluated on the same set of metrics used for the model after 100% R-SMOTe.

The corresponding ROC Curve is shown in Fig 15.

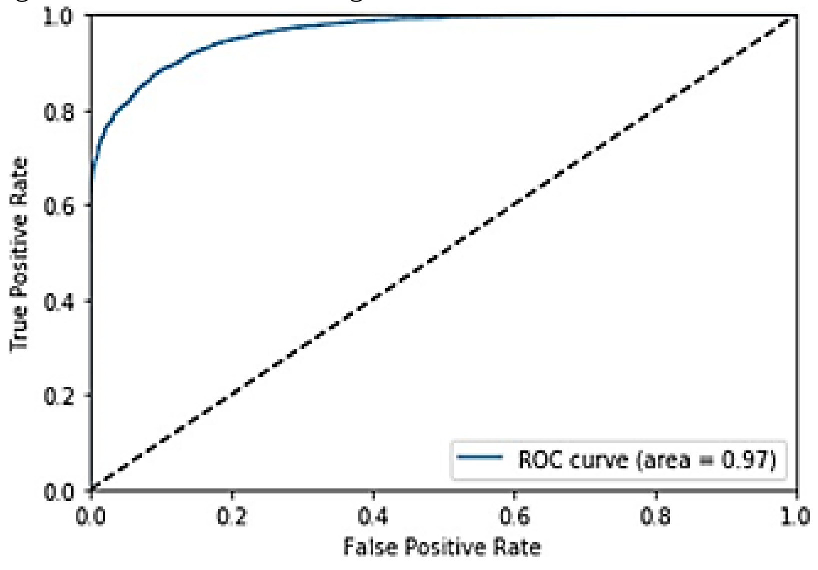


Fig 15. ROC Curve showing the Area Under the Curve for the model after 200% R-SMOTe

Confusion Matrix for the model is shown in Fig 16.

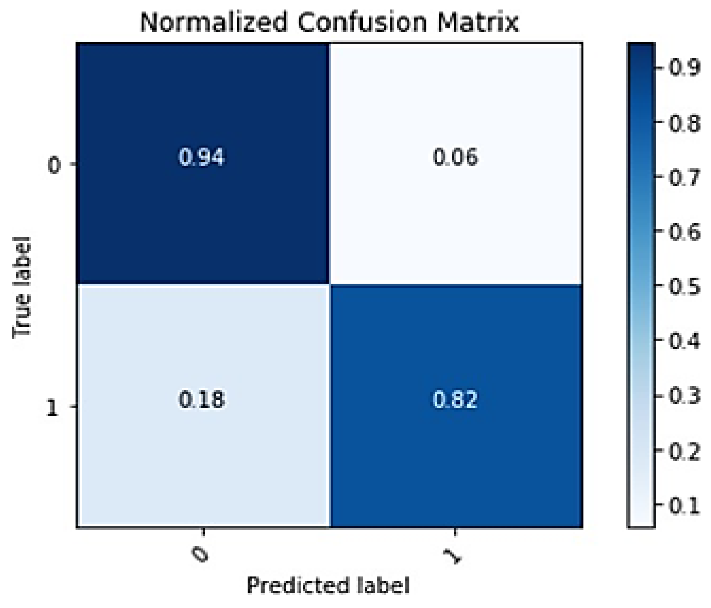


Fig 16. Normalized Confusion Matrix for the model after 200% R-SMOTe

ISSN: 2582-3825 (online)

Journal of Electronics and Informatics (2020)
Vol.02/ No. 02
Pages: 96-136
http://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.2.004

All the results for the model after 200% R-SMOTe are shown in Table 7.
Stage-wise Over-sampling using NMOTe: A stage-wise Over-sampling is done using NMOTe in which,

I. 100% Over-sampling in the 1st stage

A. Shuffling and 80-20 Splitting

B. Gradient Boosting Classifier is used as Learning Algorithm

The GBC Model is tuned using Grid-Search. After tuning, 500 estimators and maximum depth of 3 are found to be the best set of hyper-parameters. The summary of the Grid-Search Tuning of GBC on the basis of Mean Score is shown in Fig 17.

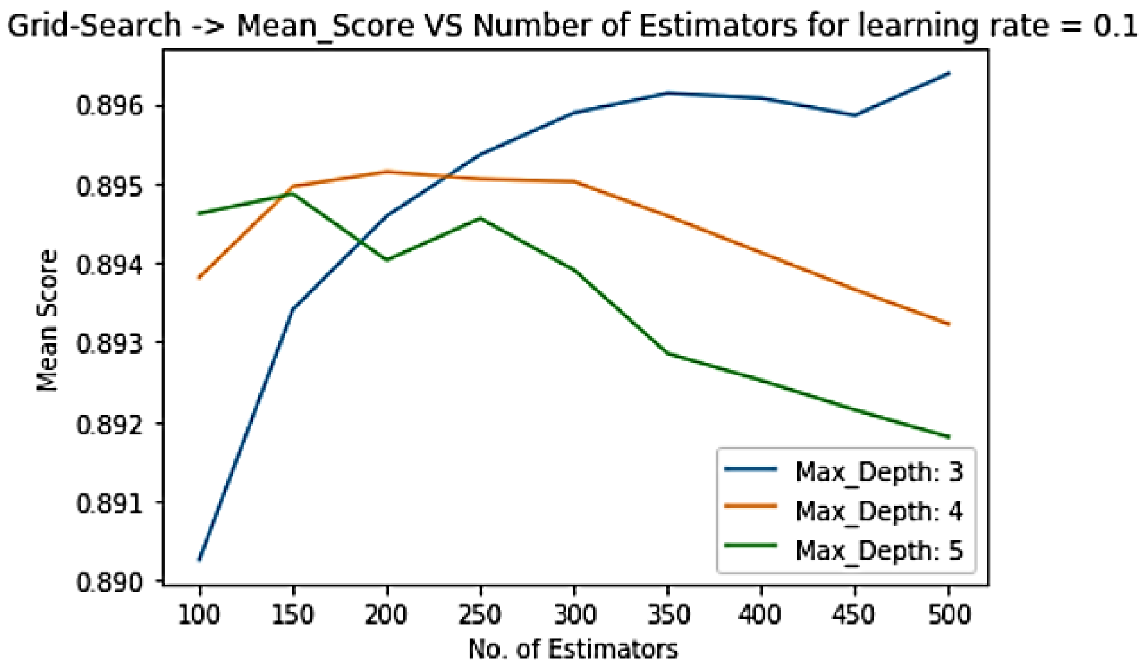


Fig 17. Grid Search Summary on Mean Score for the model after 100% NMOTe

C. Results

The model performance is evaluated on the following metrics:

- Training Accuracy
- Validation Accuracy
- Sensitivity or Recall
- Precision
- F1-Score
- Area Under Receiver Operator Characteristic Curve (AUROC) The corresponding ROC Curve is shown in Fig 18.

ISSN: 2582-3825 (online)

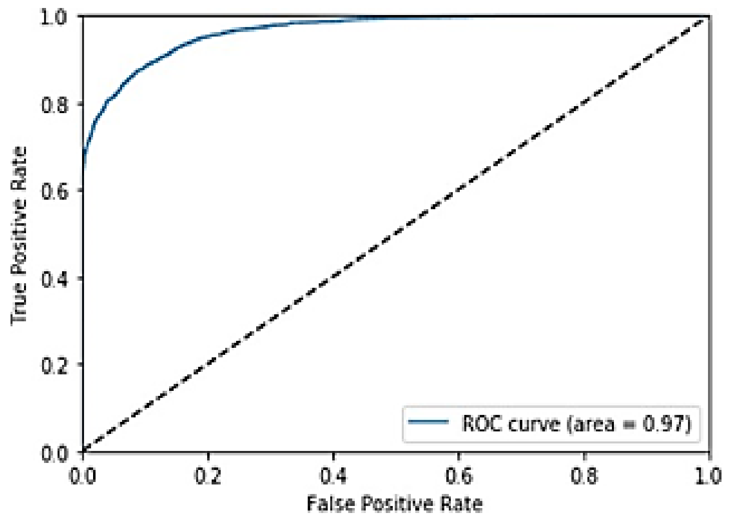


Fig 18. ROC Curve for the model after 100% NMOTe

Confusion Matrix for the model is shown in Fig 19.

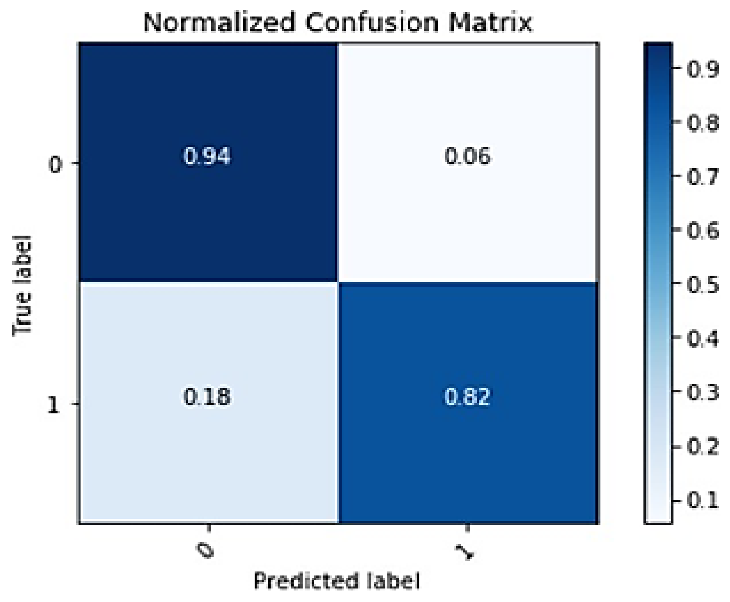


Fig 19. Normalized Confusion Matrix for the model after 100% NMOTe

All the results for the model after 100% NMOTe are shown in Table 4.
II. 200% Over-sampling is the 2nd stage
A. Shuffling and 80-20 Splitting
B. Gradient Boosting Classifier is used as Learning Algorithm

Journal of Electronics and Informatics (2020)
Vol.02/ No. 02
Pages: 96-136
http://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.2.004

500 estimators and maximum depth of 3 are found to be the best set of hyper-parameters after applying Grid-Search on the GBC model. The Grid-Search summary is shown in Fig 20.

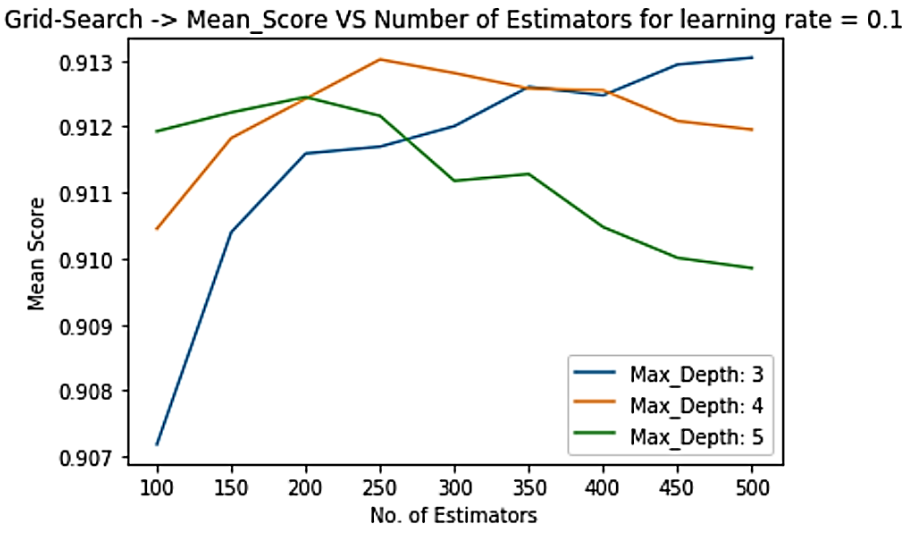


Fig 20. Grid-Search Summary on Mean Score for the model after 200% NMOTe

C. Results

The model performance is evaluated on the same set of metrics used for the model after 100% NMOTe. The corresponding ROC Curve is shown in Fig 21.

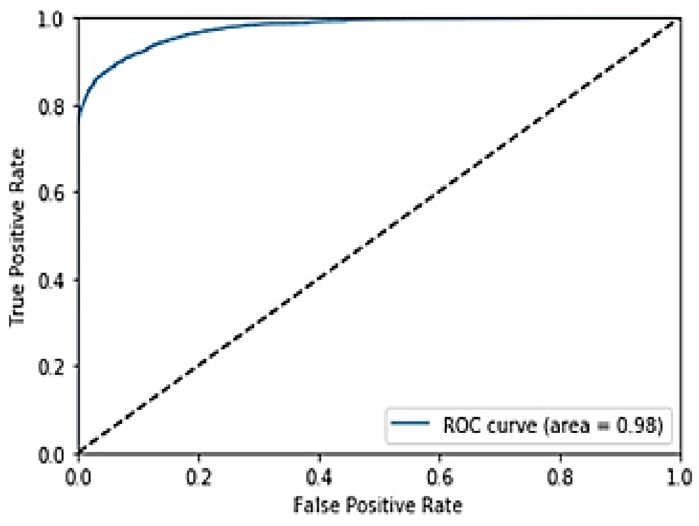


Fig 21. ROC Curve for the model after 200% NMOTe

Confusion Matrix based on the Validation Set is shown in Fig 22.

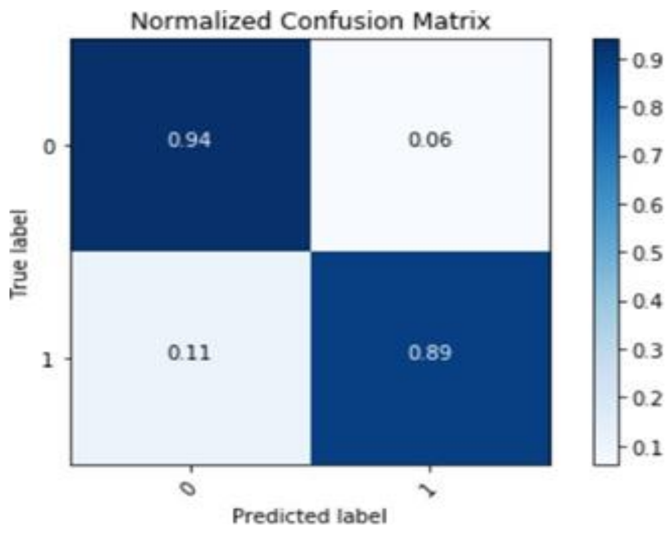


Fig 22. Normalized Confusion Matrix for the model after 200% NMOTe

All the results for the model after 200% NMOTe are shown in Table 7.

2.3 Experiment on Flight Data

2.3.1 Dataset

The Flight data-set was extracted from US Department of Transportation's Bureau of Transportation Statistics (BTS) [28] for the years 2015 & 2016 and the features associated with the dataset are tabulated in Table 2. The binary label is Arr Del 15 with values, 0 and 1 where 0 means no Arrival Delay and 1 means Arrival Delay. After that data-set is filtered to include the Flight Data of American Airlines serving the top 5 busiest airports of US. The resulting dataset contains 97,360 samples with 12 attributes and 1 label.

2.3.2 Problem Statement

Here, the Problem Statement is to Predict the Flight Arrival Delay of American Airline Flights serving top 5 busiest airports of US, Hartsfteld-Jackson Atlanta International Airport, Los Angeles International Air- port, O'Hare International Airport, Dallas/Fort Worth International Airport and John F. Kennedy International Airport.

2.3.3 Proposed Methodology

Journal of Electronics and Informatics (2020)
Vol.02/ No. 02
Pages: 96-136
http://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.2.004

Feature Selection

ID	Attribute/Feature Name	Attribute Type
F1	Year	Categorical
F2	Quarter	Categorical
F3	Month	Categorical
F4	Day_of_Month	Categorical
F5	Day_of_Week	Categorical
F6	Flight_Num	Categorical
F7	Origin_Airport_ID	Categorical
F8	Origin_World_Area_Code	Categorical
F9	Destination_Airport_ID	Categorical
F10	Destination_World_Area_Code	Categorical
F11	CRS_Departure_Time	Continuous
F12	CRS_Arrival_Time	Continuous

Table 2. Feature Study in Flight Data-set

Some features are dropped from the dataset on several grounds:

- Year (F1) is dropped as it has almost no variability throughout the dataset as flight information of years 2015 and 2016 are considered
- Quarter (F2) is also dropped as information about Quarter is already obtained from Month (F3)
- Tacking Missing Values
- All the instances having missing values are eliminated from the dataset.
- Label Encoding of Categorical Features
- Label Encoding is done for all the Categorical Features mentioned in Table 2.
- One-Hot Encoding
- Features, F7, F8, F9 and F10 are one-hot encoded.
- Data Imbalance Removal or Data Balancing, Model Development, Training the Model and Results

Here, there are 76090 instances of label 0 and only 19668 instances with label 1 or flight delay instances. Hence, there is gross class imbalance which is shown with the help of bars in Fig 23.

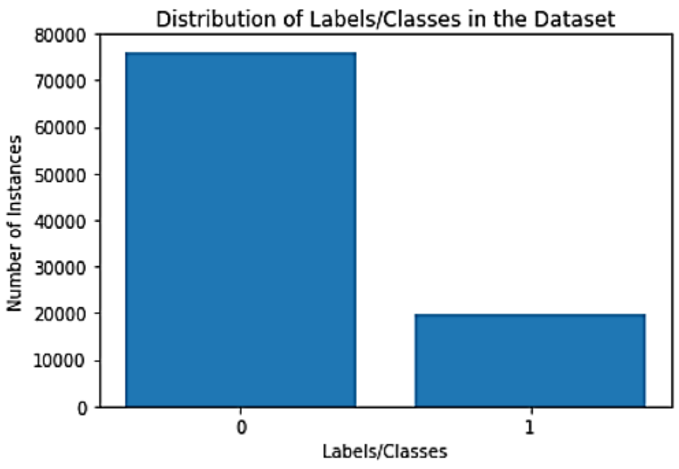


Fig 23. Bar Visualization showing the imbalance between the 2 labels in the dataset

ISSN: 2582-3825 (online)

Now, Over-sampling of instances with label 1 i.e., minority instances need to be done. This Over-sampling is done in 3 strategies:

Over-sampling using SMOTe in a)
Over-sampling using R-SMOTe in b)
Over-sampling using NMOTe in c)

(a) Stage-wise Over-sampling using SMOTe:

I. 100% Over-sampling is the 1st stage

With 100% Over-sampling, the reduction in Class Imbalance is shown in Fig 24.

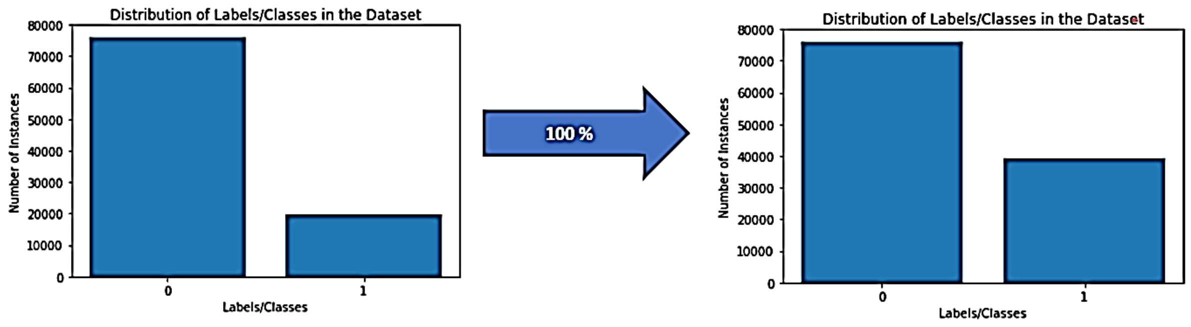


Fig 24. Reduction in Data Imbalance after 100% Over-Sampling

A) Shuffling and 80-20 Splitting

B) Gradient Boosting Classifier is used as the Learning Algorithm.

The GBC model is tuned using Grid-Search. After tuning, 250 estimators and maximum depth of 5 are found to be the best set of hyper-parameters. The Grid-Search summary on mean score is shown in Fig 25.

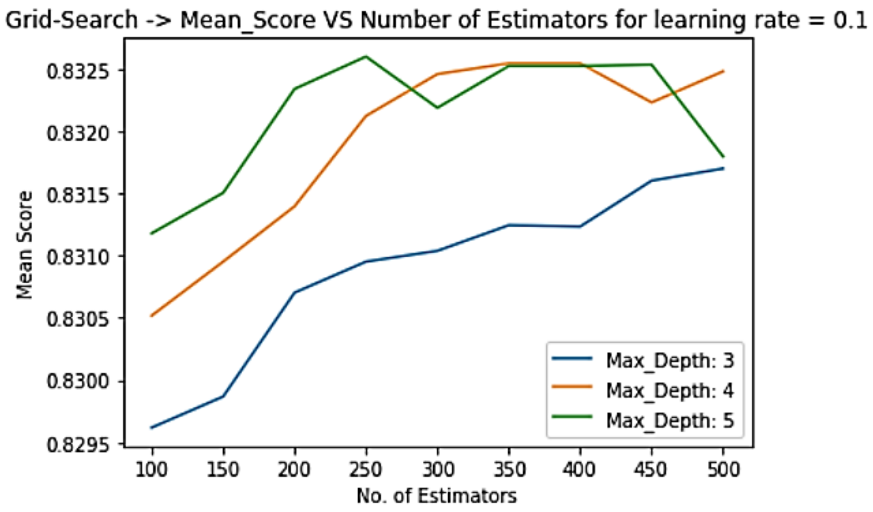


Fig 25. Grid Search Summary on Mean Score for the model after 100% SMOTe

C) Results

The model performance is evaluated on the following metrics:

- Training Accuracy
- Validation Accuracy
- Sensitivity or Recall
- Precision
- F1-Score
- Area Under Receiver Operator Characteristic Curve: The corresponding ROC Curve is shown in Fig 26.

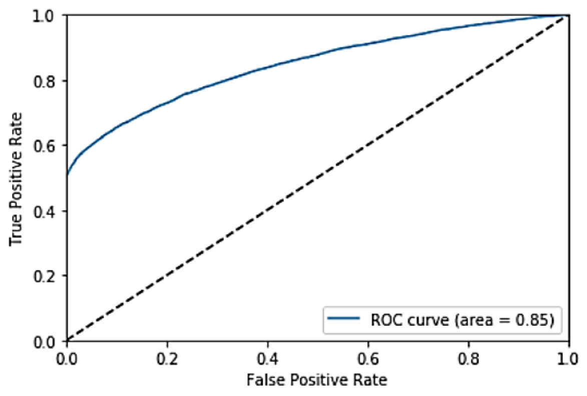


Fig 26. ROC Curve for the model after 100% SMOTe

Confusion Matrix based on the Validation Set is shown in Fig 27.

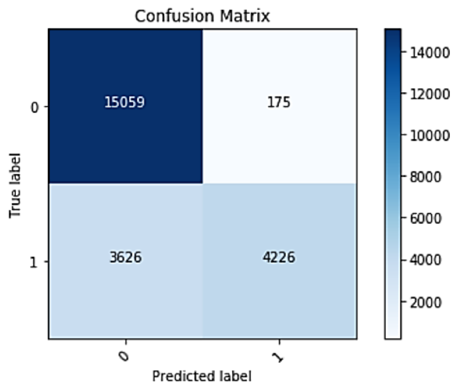


Fig 27. Confusion Matrix for the model after 100% SMOTe

All the results for the model after 100% SMOTe are shown in Table 6.

II. 200% Over-sampling is the 2nd stage

With 200% Over-Sampling, Class Imbalance got reduced to a great extent as shown in Fig 28.

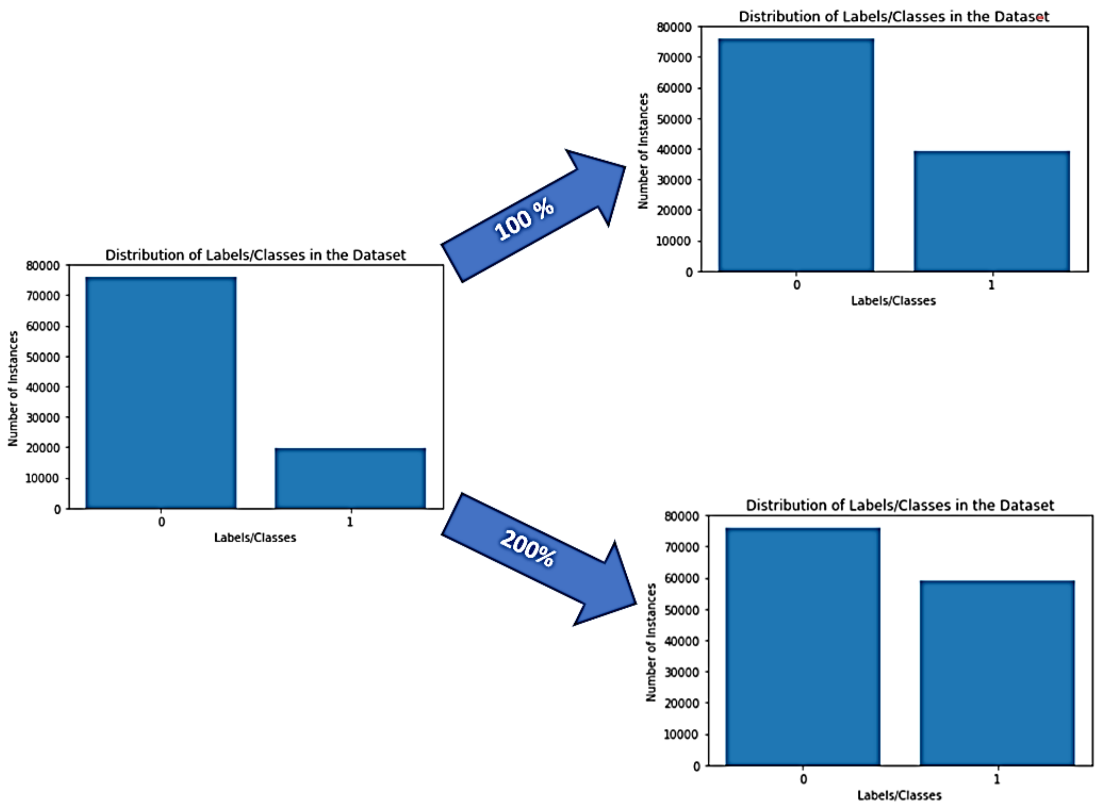


Fig 28. Reductions in Data Imbalance after 100% and 200% separately

A) Shuffling and 80-20 Splitting

B) Gradient Boosting Classifter is used as the Learning Algorithm

The Gradient Boosting Classifier Model is tuned using Grid-Search. After tuning the model with Grid-Search, 500 estimators and maximum depth of 5 are found to be the best set of hyper-parameters. The Grid-Search summary of the GBC model is shown in Fig 29.

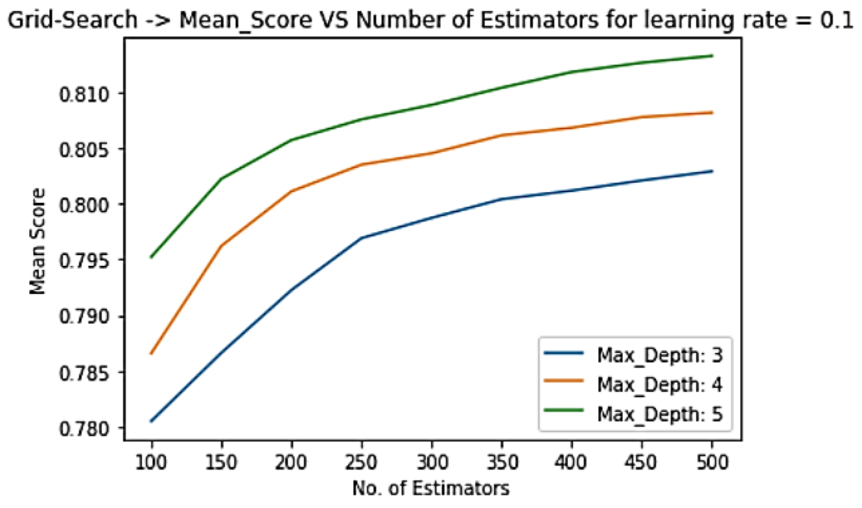


Fig 29. Grid Search Summary on Mean Score for the model after 200% SMOTe

C) Results

The model performance is evaluated on the same set of metrics used for the model after 100% SMOTe.

Area Under Receiver Characteristic Curve (AUROC) The corresponding ROC Curve is shown in Fig 30.

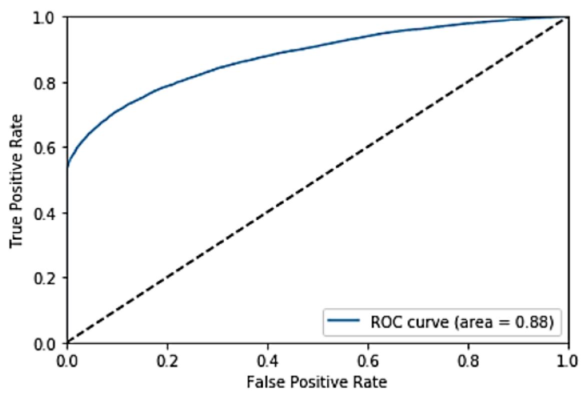


Fig 30. ROC Curve showing the Area Under the Curve for the model after 200% SMOTe

Confusion Matrix for the model is shown in Fig 31.

Journal of Electronics and Informatics (2020)
Vol.02/ No. 02
Pages: 96-136
http://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.2.004

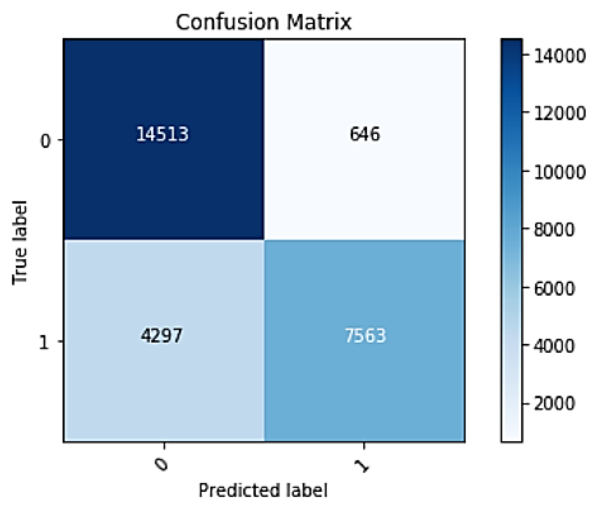


Fig 31. Confusion Matrix for the model after 200% SMOTe

All the results for the model after 200% SMOTe are shown Table 8.
(b) Stage-wise Over-sampling using R-SMOTe:
A stage-wise Over-sampling is done using R-SMOTe in which,
I. 100% Over-sampling in the 1st stage
 A. Shuffling and 80-20 Splitting
 B. Gradient Boosting Classifier is used as Learning Algorithm
Grid-Search is used for tuning the GBC Model and after tuning, 250 estimators and maximum depth of 5 are found to be the best set of hyper-parameters. The Grid-Search summary is shown in Fig 32.

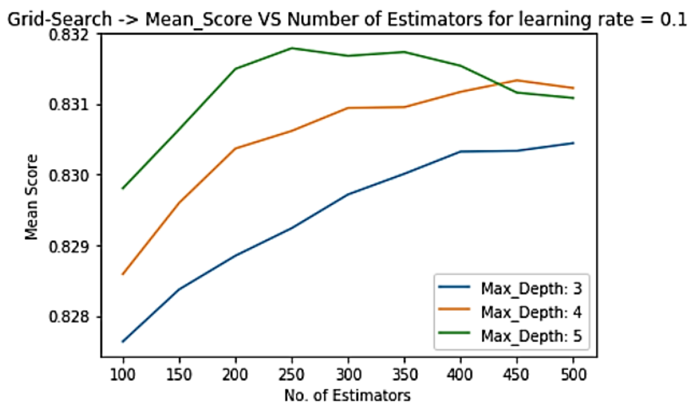


Fig 32. Grid-Search Summary on Mean Score for the model after 100% R-SMOTe

C. Results

The model performance is evaluated on the following metrics:
- Training Accuracy

ISSN: 2582-3825 (online)

- Validation Accuracy
- Sensitivity or Recall
- Precision
- F1-Score
- Area Under Receiver Operator Characteristic Curve (AUROC): The corresponding ROC Curve is shown in Fig 33.

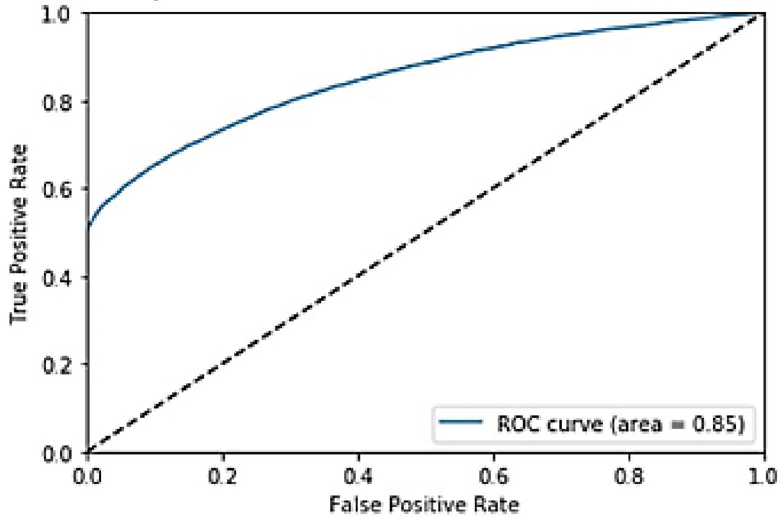


Fig 33. ROC Curve for the model after 100% R-SMOTe

Confusion Matrix based on the Validation Set is shown in Fig 34.

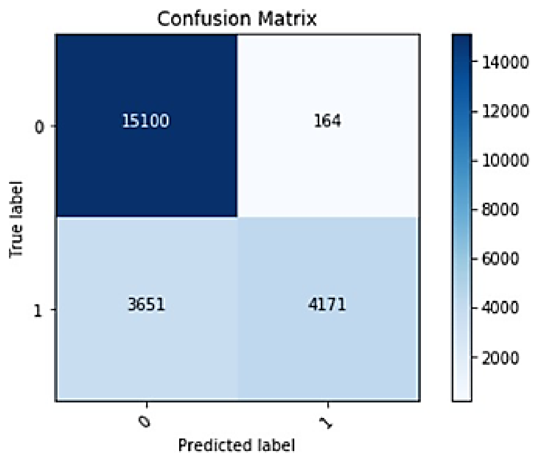


Fig 34. Confusion Matrix for the model after 100% R-SMOTe

All the results for the model after 100% R-SMOTe are shown in Table 6.

II. 200% Over-sampling is the 2nd stage

A. Shuffling and 80-20 Splitting

B. Gradient Boosting Classifier is used as Learning Algorithm

400 estimators and maximum depth of 5 are found to be the best set of hyper-parameters after applying Grid-Search on the GBC model. The Grid-Search summary is shown in Fig 35,

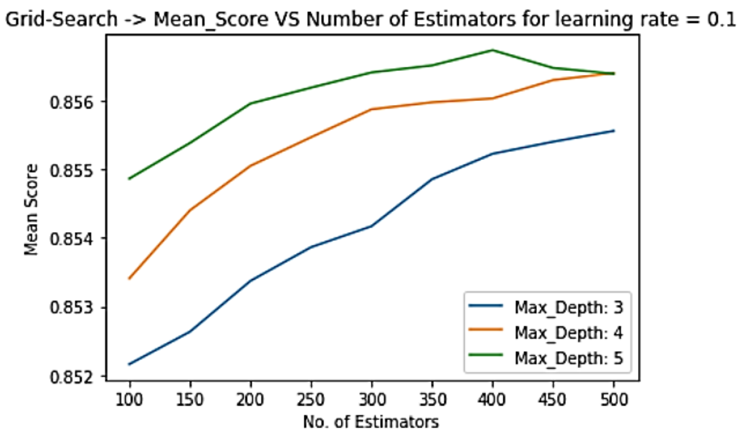


Fig 35. Grid-Search Summary on Mean Score for the model after 200% R-SMOTe

C. Results

The model performance is evaluated on the same set of metrics used for the model after 100% R-SMOTe.

The corresponding ROC Curve is shown in Fig 36.

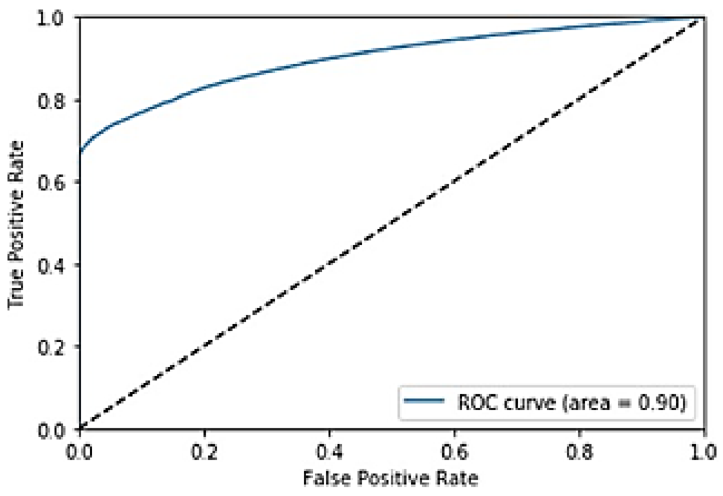


Fig 36. ROC Curve for the model after 200% R-SMOTe

Confusion Matrix based on the Validation Set is shown in Fig 37.

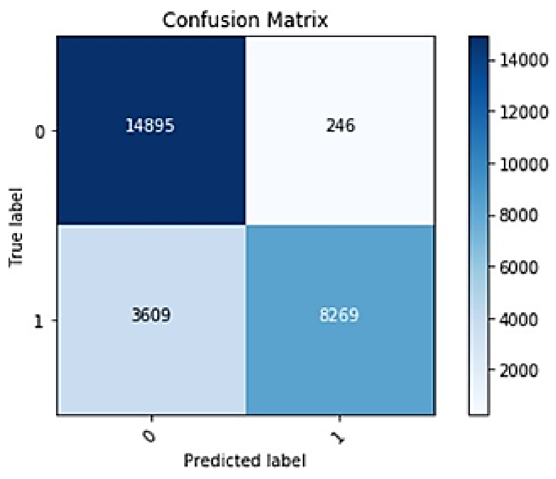


Fig 37. Confusion Matrix for the model after 200% R-SMOTe

All the results for the model after 200% R-SMOTe are shown in Table 8.

(C) Stage-wise Over-sampling using NMOTe: A stage-wise Over-sampling is done using NMOTe in which,

I. 100% Over-sampling in the 1st stage

A. Shuffling and 80-20 Splitting

B. Gradient Boosting Classifier is used as Learning Algorithm

The GBC Model is tuned using Grid-Search, which after tuning, 400 estimators and maximum depth of 5 are found to be the best set of hyper-parameters. The Grid-Search Summary is shown in Fig 38.

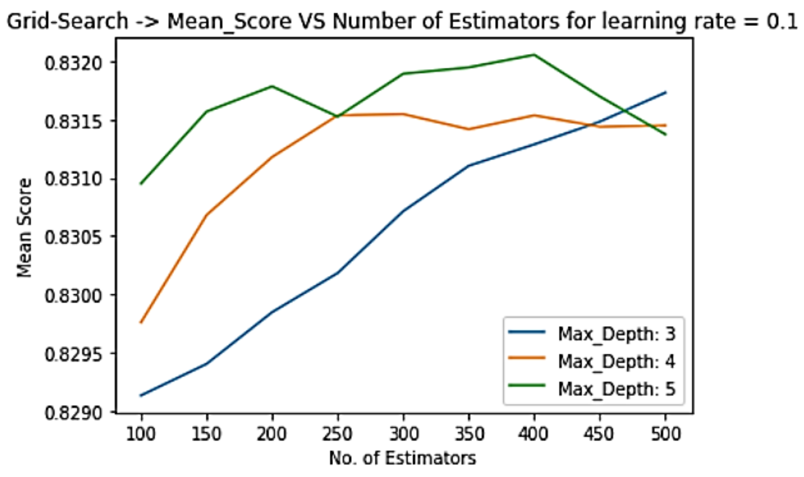


Fig 38. Grid Search Summary on Mean Score for the model after 100% NMOTe

C. Results

The model performance is evaluated on the following metrics:

- Training Accuracy
- Validation Accuracy
- Sensitivity or Recall
- Precision
- F1-Score
- Area Under Receiver Operator Characteristic Curve (AUROC): The corresponding ROC Curve is shown in Fig 39.

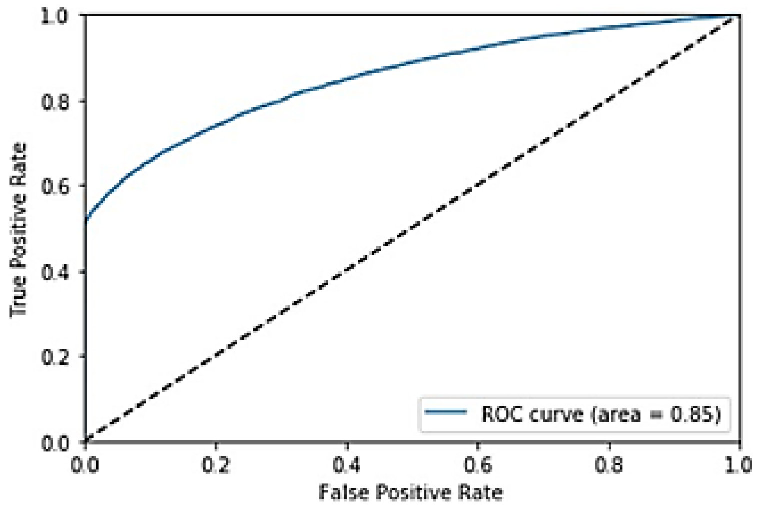


Fig 39. ROC Curve for the model after 100% NMOTe

Confusion Matrix based on the Validation Set is shown in Fig 40

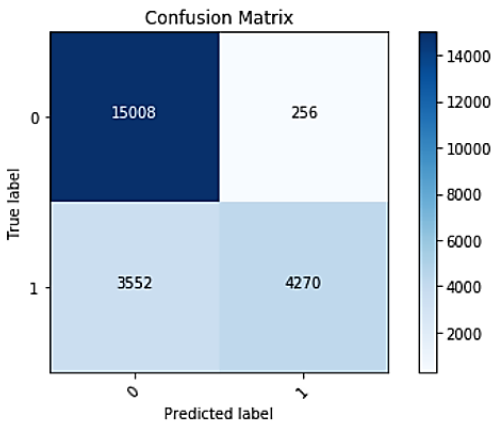


Fig 40. Confusion Matrix for the model after 100% NMOTe

All the results for the model after 100% NMOTe are shown in Table 6.
II. 200% Over-sampling is the 2nd stage
A. Shuffling and 80-20 Splitting
B. Gradient Boosting Classifter is used as Learning Algorithm

Journal of Electronics and Informatics (2020)
Vol.02/ No. 02
Pages: 96-136
http://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.2.004

300 estimators and maximum depth of 5 are found to be the best set of hyper-parameters after applying Grid-Search on the GBC model. The Grid-Search summary is shown in Fig 41.

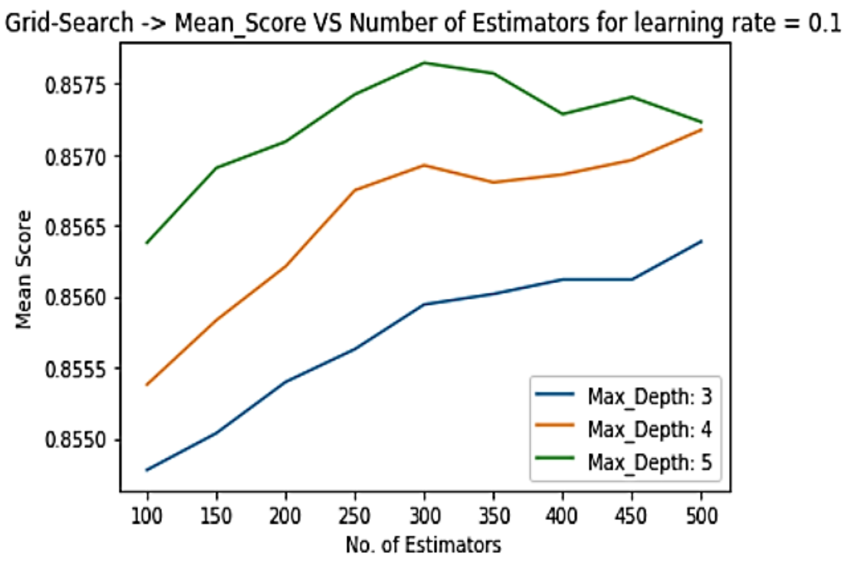


Fig 41. Grid-Search Summary on Mean Score for the model after 200% NMOTe

C. Results

The model performance is evaluated on the same set of metrics used for the model after 100% NMOTe. The corresponding ROC Curve is shown in Fig 42.

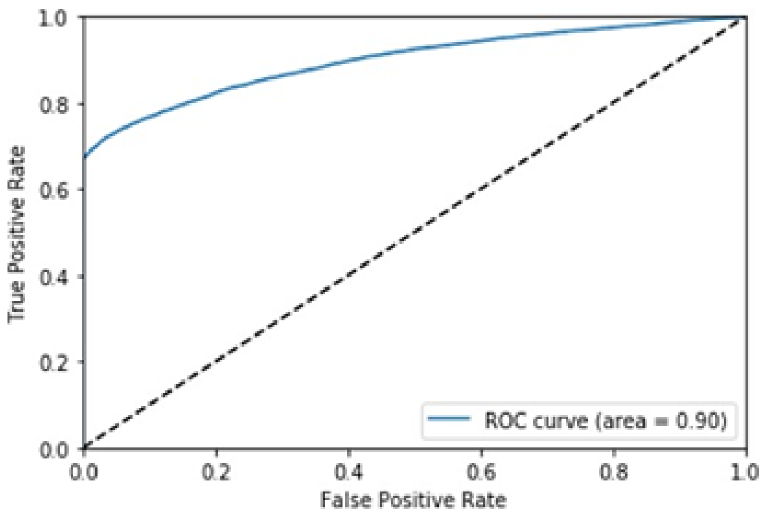


Fig 42. ROC Curve for the model after 200% NMOTe

Confusion Matrix based on the Validation Set is shown in Fig 43.

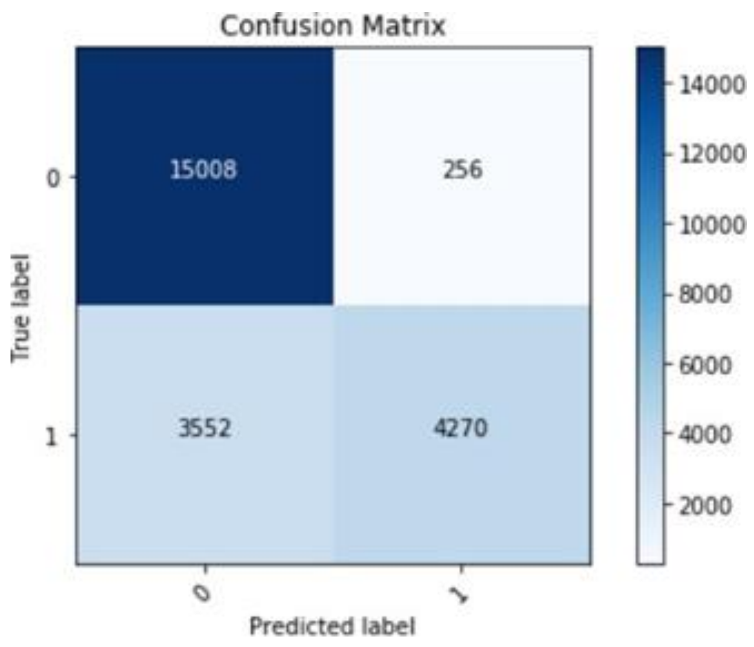


Fig 43. Confusion Matrix for the model after 200% NMOTe

All the results for the model after 200% NMOTe are shown in Table 8.

3. Implementation Details

The Data Preprocessing and Model Development are done using Python's Scikit- Learn Machine Learning Toolbox on a machine with Intel(R) Core(TM) i5-8250U processor, CPU @ 1.60 GHz 1.80 GHz and 8 GB RAM. The Visualizations are made using Python's Plotting Libraries, Matplotlib and Seaborn.

4. Results and Comparative Performance Analysis

Model Performance Evaluations are done on the following metrics:

- Fitting It refers to how close the Training and Validation Accuracies of a model are.
- Fitting = |Training Accuracy - Validation Accuracy|
- Validation Accuracy
- Precision
- Recall
- F1-Score
- AUROC
- Number of False Negatives

It refers to the number of those instances which are actually positive (having label 0) but are predicted to be negative (with label 1). This is of extra significance in a Flight Arrival Delay Model as any instance, which is predicted to be negative i.e., with label 1 or will be delayed is actually a flight on-time, i.e., positive with label 0, is heavily misleading. This results in Model Performance and Preference Degradation as such a model will be less preferred to be deployed. In Adult Census Data, this is of significance but not of extra significance over other mis-classified instances unlike the case of Flight Data. This number or count is obtained from the Confusion Matrix of the concerned model. So, less the number of False Negatives, better the Model Performance.

4.1 Over-sampling yields in Performance Boosting

Intuitively and from literature review of previous works, data balancing via over- sampling has always resulted in better performance. The same is true for our experiments as well.

4.1.1 Performance Analysis on Adult Census Data

The 1st Experiment on Adult Census Data discussed in Section 3.1 has a state-of- art model developed by Chakrabarty et al. [30] and the Results obtained by them are shown in Table 3.

Comparison Parameters		Chakrabarty et al. [30]	
Training Accuracy (in %)	Fitting	88.73	0.57
Validation Accuracy (in %)		88.16	
Validation Accuracy (in %)		88.16	
Precision		0.88	
Recall		0.88	
F1-Score		0.88	
AUROC		0.93	

Table 3. State-of-the-Art Adult Census Income Prediction Model

Now, a Comparative Analysis of the Model Performance with the inclusion of 100% Over-sampling before Model Training between the 3 Over-sampling Techniques, SMOTe, R-SMOTe and NMOTe is tabulated in Table 4.

Comparison Parameters		SMOTE		R-SMOTE		NMOTE	
Training Accuracy (in %)	Fitting	91	1.41	90.81	1.3	90.90	1.27
Validation Accuracy (in %)		89.59		89.51		89.63	
Validation Accuracy (in %)		89.59		89.51		89.63	
Precision		0.90		0.90		0.90	
Recall		0.90		0.90		0.90	
F1-Score		0.90		0.90		0.90	
AUROC		0.96		0.97		0.97	

Table 4. Comparative Performance Analysis for 100% Over-sampling between SMOTe vs R-SMOTe vs NMOTe on Adult Census Data

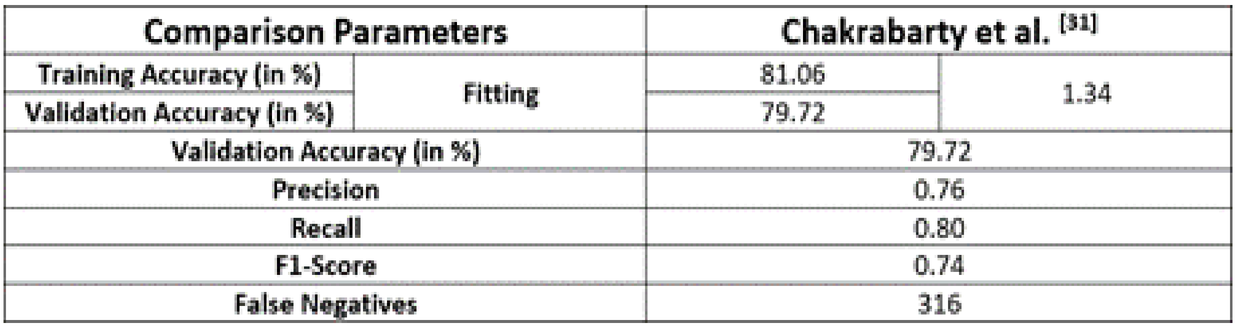
So, from Table 3 and Table 4, it is evident that Over-sampling has fetched better results in the

Journal of Electronics and Informatics (2020)
Vol.02/ No. 02
Pages: 96-136
http://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.2.004

Experiment on Adult Census Data.

4.1.2 Performance Analysis on Flight Data

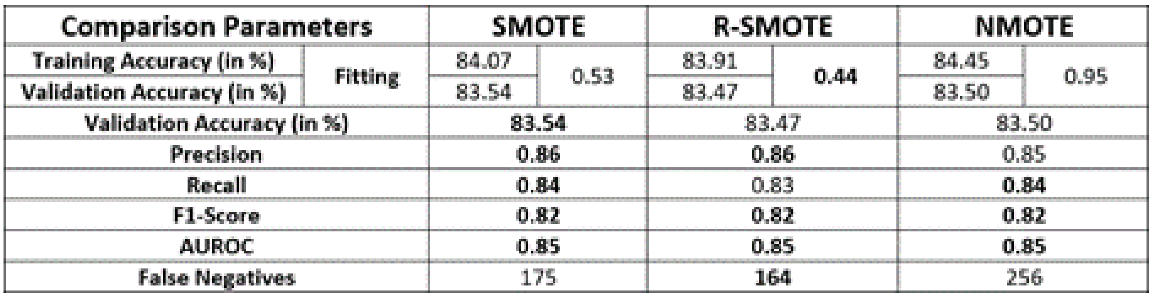
The 2nd Experiment on Flight Data discussed in Section 3.2 has the only model developed by Chakrabarty et al. [31] and the Results obtained by them are shown in Table 5.



Comparison Parameters		Chakrabarty et al. [31]	
Training Accuracy (in %)	Fitting	81.06	1.34
Validation Accuracy (in %)		79.72	
Validation Accuracy (in %)		79.72	
Precision		0.76	
Recall		0.80	
F1-Score		0.74	
False Negatives		316	

Table 5. The only Flight Arrival Delay Prediction Model for American Airlines serving only top 5 busiest airports of US

Now, the Comparative Analysis of the Model Performance with the inclusion of 100% Over-sampling before Model Training between the 3 Over-sampling Techniques, SMOTe, R-SMOTe and NMOTe is tabulated in Table 6.



Comparison Parameters		SMOTE		R-SMOTE		NMOTE	
Training Accuracy (in %)	Fitting	84.07	0.53	83.91	0.44	84.45	0.95
Validation Accuracy (in %)		83.54		83.47		83.50	
Validation Accuracy (in %)		**83.54**		83.47		83.50	
Precision		**0.86**		**0.86**		0.85	
Recall		**0.84**		0.83		**0.84**	
F1-Score		**0.82**		**0.82**		**0.82**	
AUROC		**0.85**		**0.85**		**0.85**	
False Negatives		175		**164**		256	

Table 6. Comparative Performance Analysis for 100% Over-sampling between SMOTe vs R-SMOTe vs NMOTe on Flight Data

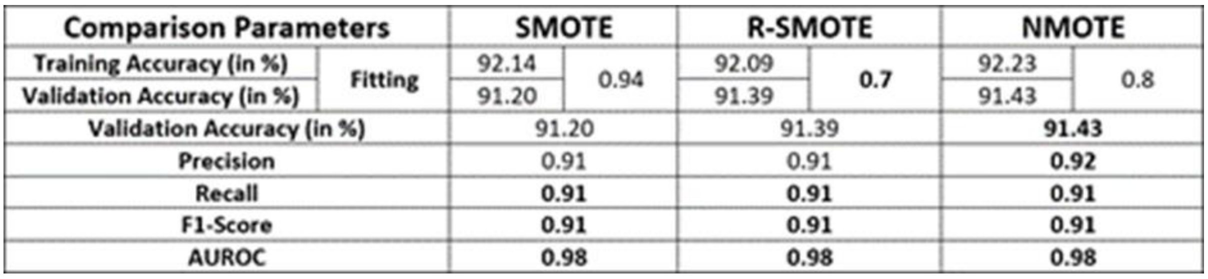
So, from Table 5 and Table 6, it is evident that Over-sampling has fetched better results in the Experiment on Flight Data.
Hence, it is therefore proved, that Over-sampling intended to Data-Balancing, improves model performance and hence, yields better results

4.2 Comparative Model Performance Analysis

4.2.1 Adult Census Data

1. 100% Over-sampling

From Table 4, it can be easily interpreted that NMOTe is a clear winner, outperforming both SMOTe and R-SMOTe. Among SMOTe and R-SMOTe, R-SMOTe has fetched better results.



Comparison Parameters		SMOTE		R-SMOTE		NMOTE	
Training Accuracy (in %)	Fitting	92.14	0.94	92.09	**0.7**	92.23	0.8
Validation Accuracy (in %)		91.20		91.39		91.43	
Validation Accuracy (in %)		91.20		91.39		**91.43**	
Precision		0.91		0.91		**0.92**	
Recall		**0.91**		**0.91**		**0.91**	
F1-Score		**0.91**		**0.91**		**0.91**	
AUROC		**0.98**		**0.98**		**0.98**	

Table 7. Comparative Performance Analysis for 200% Over-sampling between SMOTe vs R-SMOTe vs NMOTe on Adult Census Data

2. 200% Over-sampling

From Table 7, it can be concluded that NMOTe outperformed both SMOTe and R-SMOTe being only slightly behind R-SMOTe in Fitting, but leading in Validation Accuracy and Precision than R-SMOTe. Among SMOTe and R- SMOTe, R-SMOTe yielded better results with nice fitting measure and higher Validation Accuracy.

In this experiment, NMOTe proved to fetch better results than SMOTe and R- SMOTe.

4.2.2 Flight Data

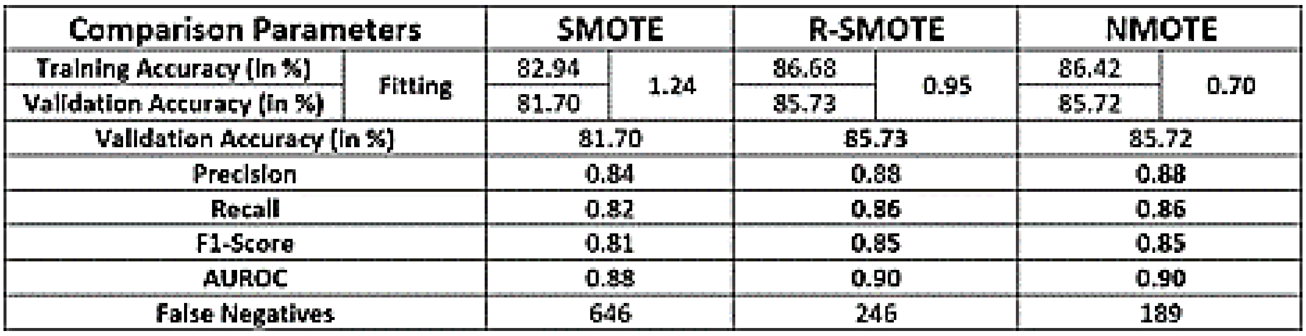
1. 100% Over-sampling

From Table 6, many important interpretations can be drawn:

- ➢ R-SMOTe leads in terms of fitting measure.
- ➢ SMOTe yielded the highest Validation Accuracy
- ➢ SMOTe and R-SMOTe fetched the highest Precision of 0.86 as compared to NMOTe yielding 0.85 SMOTe and NMOTe gave the highest Recalls of 0.84 as compared to R-SMOTe fetching 0.83.
- ➢ R-SMOTe gave the least number of False Negatives

So, all-in-all, SMOTe and R-SMOTe performed better than the newly introduced algorithm, NMOTe. But as imbalance in data still exists, 200% Over- sampling has been done and termed as 2nd stage.

Journal of Electronics and Informatics (2020)
Vol.02/ No. 02
Pages: 96-136
http://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.2.004



Comparison Parameters		SMOTE		R-SMOTE		NMOTE	
Training Accuracy (in %)	Fitting	82.94	1.24	86.68	0.95	86.42	0.70
Validation Accuracy (in %)		81.70		85.73		85.72	
Validation Accuracy (in %)		81.70		85.73		85.72	
Precision		0.84		0.88		0.88	
Recall		0.82		0.86		0.86	
F1-Score		0.81		0.85		0.85	
AUROC		0.88		0.90		0.90	
False Negatives		646		246		189	

Table 8. Comparative Performance Analysis for 200% Over-sampling between SMOTe vs R-SMOTe vs NMOTe on Flight Data

2. 200% Over-sampling

From Table 8, the following conclusions are drawn:

- ➢ Considering Table 6 and Table 8, it is observed that SMOTe results in performance degradation in 2nd stage of 200% Over-sampling, rather than boosting the performance as more the Over-sampling percentage, more the data imbalance is reduced and better should be the performance, which is not the case in 200% SMOTe on Flight Data.
- ➢ Therefore, SMOTe is proven to be highly inconsistent over-sampling technique which does not guarantee improvement in performance with reduction in data imbalance
- ➢ R-SMOTe is 0.01% ahead in Validation Accuracy than NMOTe i.e., 85.73% in case of R-SMOTe and 85.72% in case of NMOTe.
- ➢ NMOTe establishes a clear lead in terms of Fitting measure and also results in least number of False Negatives.

So, finally it can be stated that in 200% Over-sampling the Flight Data, NMOTe has outperformed R-SMOTe and SMOTe. Also, SMOTe is proven to be highly inconsistent.

So, NMOTe is established to succeed as a better Over-sampling method in this experiment also.

Hence, NMOTe, from our experiments on binary data, can be called as a Consistent Performance Booster and the most effective Stand-Alone and Rigid Over-sampling algorithm yet.

5. Conclusion

After having discussed comprehensively about the current research challenges that is often faced when learning from imbalanced data, a novel oversampling approach called Navo Minority Oversampling TEchnique (NMOTe) has been proposed in this work to find an efficient and consistent solution and applied them in contemporary real-world applications. A thorough experimentation has been conducted in this study to examine the effectiveness of this NMOTe algorithm over the existing state-of -the-art standalone and rigid techniques, SMOTe and Random-SMOTe, that has been used over the last few years. From our experiments, it has been observed that, NMOTe has boosted model performance over SMOTe and Random-SMOTe, on some metrics most of the times: Fitting, Validation Accuracy and Precision. Also, there is another point to be noted that, on special problems like Flight Arrival Delay Prediction, in which among mis-classified examples, the false negatives are considered to be the most harmful, 200% NMOTe, resulted in the least number of false-negatives.

In future, the performance of NMOTe can be further evaluated on more extensively skewed data. Also, this work is thoroughly focused on binary imbalanced data and can be further extended for multi-class problems. And lastly, we have conducted this entire research based on some extensive experimental evaluation but some mathematical modelling can be further incorporated to make this study more justified and compact.

Compliance with Ethical Standards

Funding: The study is not funded by any organization.

Conflict of Interest: The authors declare that they have no conflict of interest.

References

[1] Wing, Jeannette M. ”Computational thinking.” Communications of the ACM 49.3 (2006): 33-35

[2] Bennett, Kristin P., and Emilio Parrado-Hern´andez. ”The interplay of optimization and machine learning research.” Journal of Machine Learning Research 7.Jul (2006): 1265-1281.

[3] Xu, R., Chen, T., Xia, Y., Lu, Q., Liu, B., Wang, X.: Word embedding composition for data imbalances in sentiment and emotion classification. Cogn. Comput. 7(2), 226–240 (2015)

[4] Munkhdalai, T., Namsrai, O.-E., Ryu, K.H.: Self-training in significance space of support vectors for imbalanced biomedical event data. BMC Bioinform. 16(S–7), S6 (2015)

[5] Razakarivony, S., Jurie, F.: Vehicle detection in aerial imagery: a small target detection benchmark. J. Vis. Commun. Image Represent. 34, 187–203 (2016)

[6] Ramentol, E., Gondres, I., Lajes, S., Bello, R., Caballero, Y., Cornelis, C., Herrera, F.: Fuzzy-rough imbalanced learning for the diagnosis of high voltage circuit breaker maintenance: the SMOTE-FRST-2T algorithm. Eng. Appl. AI 48, 134–139 (2016)

[7] Japkowicz, N. (2001). Concept-learning in the presence of between-class and within-class imbalances. In Proceedings of the Canadian Conference on AI 2001 (pp. 67–77).

[8] Japkowicz, N. (2003). Class imbalance: Are we focusing on the right issue. In Proc. of 2nd Workshop on Learning from Imbalanced Data Sets (ICML) (pp. 17–23).

[9] Prati, R.C., Batista, G., Monard, M.C. (2004). Learning with class skews and small disjuncts. In Proc. of SBIA’04 (pp. 296–306)., Garcia, V., Sanchez, J., Mollineda, R. (2007)

[10] Garc´ıa, Vicente, Jose S´anchez, and Ramon Mollineda. ”An empirical study of the behavior of classifiers on imbalanced and overlapped data sets.” Iberoamerican Congress on Pattern Recognition. Springer, Berlin, Heidelberg, 2007.

[11] Napiera-la, Krystyna, Jerzy Stefanowski, and Szymon Wilk. ”Learning from imbalanced data in presence of noisy and borderline examples.” International Conference on Rough Sets and Current Trends in Computing. Springer, Berlin, Heidelberg, 2010.

[12] X.Y. Liu, J. Wu, and Z.H. Zhou, “Exploratory Under Sampling for Class Imbalance Learning,” Proc. Int’l Conf. Data Mining, pp. 965- 969, 2006.
[13] H. He and E.A. Garcia, “Learning from Imbalanced Data,” IEEE Trans. Knowledge Data Eng., vol. 21, no. 9, pp. 1263-1284, Sept. 2009.
[14] B. Settles and M. Craven, “An analysis of active learning strategies for sequence labeling tasks,” in Proc. Conf. Empirical Methods NaturalLang. Process. (EMNLP), Oct. 2008, pp. 1070–1079.
[15] Chawla, Nitesh V., et al. ”SMOTE: synthetic minority over-sampling technique.” Journal of artificial intelligence research 16 (2002): 321-357.
[16] Han, Hui, Wen-Yuan Wang, and Bing-Huan Mao. ”Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning.” International Conference on Intelligent Computing. Springer, Berlin, Heidelberg, 2005.
[17] Chawla, Nitesh V., et al. ”SMOTEBoost: Improving prediction of the minority class in boosting.” European conference on principles of data mining and knowledge discovery. Springer, Berlin, Heidelberg, 2003.
[18] Dong, Yanjie, and Xuehua Wang. ”A new over-sampling approach: random-SMOTE for learning from imbalanced data sets.” International Conference on Knowledge Science, Engi- neering and Management. Springer, Berlin, Heidelberg, 2011.
[19] Zheng, Zhuoyuan, Yunpeng Cai, and Ye Li. ”Oversampling method for imbalanced classification.” Computing and Informatics 34.5 (2016): 1017-1037.
[20] Ertekin, S¸eyda. ”Adaptive oversampling for imbalanced data classification.” Information Sciences and Systems 2013. Springer, Cham, 2013. 261-269.
[21] Chen, Sheng, Haibo He, and Edwardo A. Garcia. ”RAMOBoost: ranked minority over-sampling in boosting.” IEEE Transactions on Neural Networks 21.10 (2010): 1624-1642.
[22] Seiffert, Chris, et al. ”RUSBoost: A hybrid approach to alleviating class imbalance.” IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 40.1 (2010): 185-197.
[23] Li, Jia, Hui Li, and Jun-Ling Yu. ”Application of random-SMOTE on imbalanced data mining.” Business Intelligence and Financial Engineering (BIFE), 2011 Fourth International Conference on. IEEE, 2011.
[24] Wang, Boyu, and Joelle Pineau. ”Online bagging and boosting for imbalanced data streams.” IEEE Transactions on Knowledge Data Engineering 1 (2016): 1-1.
[25] Blagus, Rok, and Lara Lusa. ”Evaluation of smote for high-dimensional class-imbalanced microarray data.” Machine learning and applications (icmla), 2012 11th international conference on. Vol. 2. IEEE, 2012.
[26] Wang, Shuo, Leandro L. Minku, and Xin Yao. ”Resampling-based ensemble methods for online class imbalance learning.” IEEE Transactions on Knowledge and Data Engineering 27.5 (2015): 1356-1368.
[27] https://archive.ics.uci.edu/ml/datasets/Adult
[28] https://www.transtats.bts.gov/
[29] go.arcadiadata.com/rs/627-XIL-022/images/airline-id.csv
[30] Chakrabarty, Navoneel, and Sanket Biswas. ”A Statistical Approach to Adult Census Income Level Prediction.” 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN). IEEE, 2018.

[31] Chakrabarty, Navoneel, et al. "Flight Arrival Delay Prediction Using Gradient Boosting Classifier." Emerging Technologies in Data Mining and Information Security. Springer, Singapore, 2019. 651-659.

Authors Biography:

Navoneel Chakrabarty is currently pursuing Post Graduate Diploma (PGD) in the discipline of Machine Learning and Artificial Intelligence (ML & AI) from *International Institute of Information Technology (IIIT)*, Bangalore, India. His research interests include Data Mining, Data Analytics, Machine Learning, Natural Language Processing, Computer Vision, Deep Learning and Recommender Systems. His many works have been published in International Conferences including Pattern Recognition and Machine Intelligence (PReMI - 2019). He is also an Active Open Source Contributor in Towards Data Science (TDS) and Towards AI under Medium.

Sanket Biswas received his Bachelor of Technology (B.Tech), in Computer Science and Engineering from Jalpaiguri Government Engineering College, Jalpaiguri, India in 2018. He is currently pursuing his Master of Science (MS), in the discipline of Computer Vision, from the joint collaboration of *Universitat Autònoma de Barcelona(UAB)*, *Universitat Politècnica de Catalunya(UPC)*, *Universitat Pompeu Fabra(UPF)* and *Universitat Oberta de Catalunya (UOC)* in Barcelona, Spain. His current research interests lie in the area of Computer Vision and Machine Learning to study and analyse various deep learning principles in understanding documents. He is also serving as a junior researcher of the Document Analysis Group (DAG) of the Computer Vision Centre (CVC) in Barcelona, Spain.