Classification of Imbalanced data: Addressing data intrinsic characteristics

A project report submitted as a part of Major-II

Master of Technology in Computer Science and Engineering

By

Armaan Garg

2K16/SWE/05

Under the Guidance of:

Dr. Ruchika Malhotra

(Associate Professor - Computer Science Engineering)

Delhi Technological University

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

DECLARATION

I, **Armaan Garg**, Roll No: 2K16/SWE/05 student of M.Tech (**Software Engineering**), hereby declare that the project Dissertation titled " **Classification of Imbalanced data: Addressing data intrinsic characteristics** " submitted to the Department of Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work result has not been submitted to any other University or Institute for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: DTU, Delhi **Armaan Garg**

Date 2K16/SWE/05

CERTIFICATE

This is to certify that the Project Dissertation titled " **Classification of Imbalanced data: Addressing data intrinsic characteristics** " which is carried out by Armaan Garg, Roll no: 2K16/SWE/05 Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the bonafide work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted to any University or Institute in part or full for any Degree or Diploma Associateship, Fellowship or other similar title or recognition.

Place: DTU, Delhi
Date

Dr. Ruchika Malhotra
SUPERVISOR
Associate Professor
Department of Computer Science & Engineering
Delhi Technological University
Delhi-110042

ACKNOWLEDGEMENT

I would like to express my gratitude to my major project advisor **Dr. Ruchika Malhotra, Associate Professor,** Department of Computer Science & Engineering, Delhi Technological University, for the valuable support and guidance she has provided in making this project work a success. With pleasure I would like to record my sincere thanks to my respected guide for her constructive motivation and insight without which the project would not have shaped as it has.

Armaan Garg
Roll No. 2K16/SWE/05
M.Tech (Software Engineering)
E-mail: armaangarg7@gmail.com

ABSTRACT

In categorising datasets with skewed classes, classifier experiences imbalanced class dissemination. There are real applications where information in various datasets has unequal distribution. Problem arises when their is non-uniform distribution of instances between classes. For adjusting the information they have set up, different methods are used to handle them. Few of the methods are: Preprocess, cost-sensitive learning and ensemble techniques. In preprocess method, the data is modified in such a way so that the imbalanced is reduced by simply modifying the number of instances in different classes. There are few techniques under this method, they are: under sampling strategy, oversampling technique and the hybrid technique. In under sampling technique the number of majority class instances are decreased. In oversampling strategy a super set is made by imitating the instances of the minority class. In hybrid approach both subset and superset readiness strategy is utilised. In cost sensitive learning method, punishments will be upheld on to the class readiness. The cost of misclassifying the positive case is significantly higher than that of misclassifying the negative one. Ensemble classifiers, endeavour to enhance the execution of single classifiers by initiating a few classifiers and consolidating them to acquire another classifier that out plays each one of them. Later on the perspective turned into that the imbalanced information in different grouping has less impact on the execution. There are some other various issues related to data intrinsic characteristics such as sample size, class overlapping, the noisy data etc. To get better classification results, one should focus on these data intrinsic properties and should resolve the issues that arise due to the them. In the proposed work we will be looking at these data intrinsic characteristics in detail and how the issues related to these characteristics can be resolved. Algorithms have been developed corresponding to each of these issues and then integrated to get the overall performance of these algorithms. At the end a transformed dataset will be produced which will be free from these issues. WEKA tool is used to classify these datasets and measure the performance. The final result shows that the classifiers produce better results for the transformed dataset than the other datasets which do not address the issues related to data intrinsic characteristics.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1
INTRODUCTION

In real world datasets the problem of skewed distribution is very common. As a result, it has drawn huge consideration from scientists and other researchers involved in the field of machine learning, information mining and example grouping disciplines. This issue creates a situation where number of instances of a class are very less as compared to the number of instances of the other class. A class is called a majority class or negative class, when it out-numbers the other class, having less number of instances, called as minority class or positive class . Excessively few specimens in the minority class could prompt false location or the information being disregarded as commotion. These issues exist in different fields, for example, data recovery, restorative determination and content characterization. Various information level answers have been proposed earlier to handle the class imbalanced issues by under-sampling majority class (negative class) cases or oversampling minority class (positive class) occasions, or a hybrid of these two methodologies so as to manage imbalanced class distribution.

Information mining frames the centre piece of the learning disclosure process. There exist different information mining procedures viz. Classification , Clustering, and so on. Our work for the most part falls under the characterization information mining system.

1.1 Overview

Classification is one of the critical procedure of information mining. It includes utilization of the model developed by gaining from the authentic information to influence expectation about the class to name of the new information/perceptions. The classification model maps each attribute of set x to a set of predefined class marks y. It can be used to recognise the objects of various classes and also to foresee class mark of unknown instances.

Upcoming figure shows the classification task which maps attribute set x to its class label y.

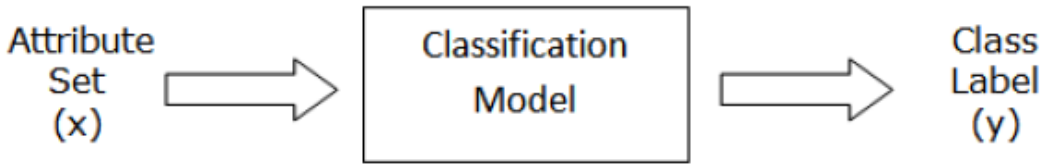

Figure 1.1.1: Classification as a task of mapping input attribute set x into its class label y

Classification problem is an inescapable issue that includes numerous different applications, ideal from static datasets to information streams. In recent times we have seen that the information streams have been exponentially increasing which are a test to conventional classifiers.

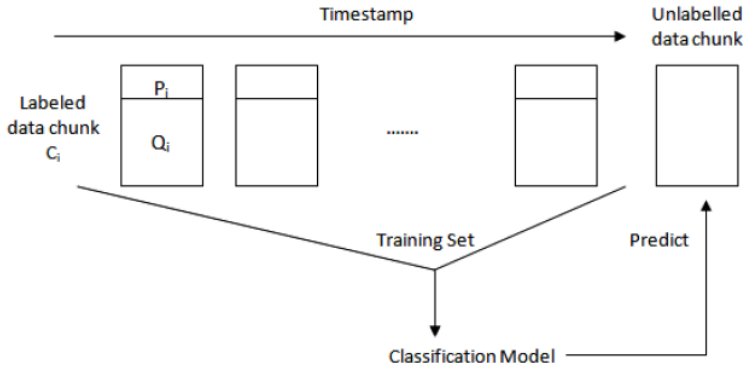

Figure 1.1.2: Classification model in data streams

In fig. 1.1.2 data chunks (sets of data) C1;C2;C3::::Ci arrive one by one. Each chunk contains positive occurrences P_i and negative examples Q_i. Assume C1;C2;C3::::Ci are named. At the time stamp m + 1, when an unlabelled piece Cm+1 arrives, the classification model assigns the unlabelled instances a tag based on the training based on labeled class instances. At the point when model give tags to the instances in Cm+1, this set can now join the training set, bringing about an ever increasing number of named information pieces. This will result in better

efficiency every time a new dataset comes in (or equal efficiency in some cases) as every time a dataset comes in, it becomes training set in the next round, so our classifier is able to learn more and more. Most examinations on stream mining accept moderately adjusted and stable information streams. In any case, numerous applications can include idea floating information streams with skewed appropriations. In information with skewed disseminations every training set has numerous less positive cases. Fig 1.1.3 demonstrates the comparable idea diagrammatically. In the meantime, misfortune capacities related with positive and negative classes are additionally lopsided.

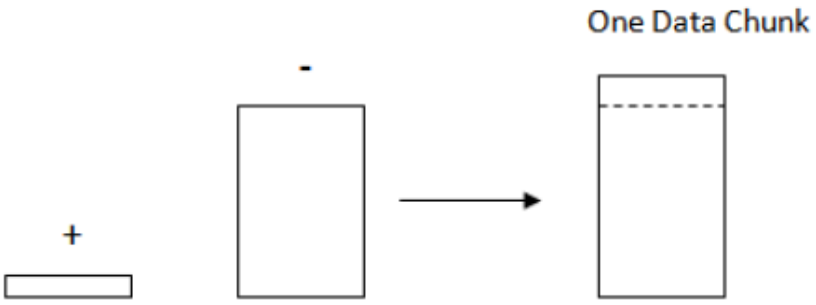

Figure 1.1.3: Skewed Distributions, each data chunk has fewer positive examples than negative examples

The rate at which science and innovation have created has brought about multiplication of information at an exponential pace. This excessive increment in information has heightened need of different applications in information mining. Class skew or class irregularity refers to a situation where in one class examples dwarf alternate class occurrences, i.e. a few classes possess most of the dataset which are known as greater part classes; while alternate classes are known as minority classes. The most crucial issue in these sorts of informational collections is that, contrasted with most of the classes, minority classes are regularly of much centrality and enthusiasm to the client.

1.1.2 SOME REAL WORLD PROBLEMS THAT EXHIBIT SUCH NATURE

Now we will be looking at some of the real world scenarios where this skewed distribution of data occurs:

- **Money related Fraud Detection**: In monetary extortion discovery, dominant part of budgetary exchanges are honest to goodness and authentic, and modest number of them might be deceitful.
- **Medicinal Fraud Detection**: In therapeutic extortion identification, the level of false claims is little, yet the aggregate misfortune is huge.
- **Ongoing Video Surveillance**: Imbalance is found in information that lands as video groupings.
- **Oil Spillage**: Oil spills identification in satellite radar pictures.
- **Space science**: Skewed informational collections exist in cosmic held likewise; just 0.001% of the items in the sky review pictures are really past the extent of current science and may prompt new revelations .
- **Spam Image Detection**: In Spam picture identification, close copy spam pictures are hard to find from the expansive number of spam pictures.
- **Content Classification**: In content arrangement, there is imbalanced information, for example, content number, class size, subclass and class crease.

1.1.3 DATA LEVEL METHODS FOR HANDLING IMBALANCE

Over-sampling

In information mining there are few preprocessing techniques, oversampling is one of them. In the oversampling technique the quantity of minority class cases in expanded by either reusing the occurrences from the past preparing/learning pieces or by making the engineered cases. Oversampling tries to strike the harmony between proportion of majority and minority classes. The most regularly utilized strategy for oversampling is SMOTE(Synthetic Minority Oversampling Technique) [8].

A system was proposed by Chen and He [9] - the SERA(Selectively Recursive Approach), in this structure they specifically consumed minority cases from past lumps into current preparing piece to adjust it.

Under-sampling

Under-sampling is another examining based strategy which takes care of the issue by diminishing the quantity of majority class occurrences. This is for the most part done by sifting through the greater part class cases or by arbitrarily choosing the fitting number of larger part class illustrations. under-sampling is generally done utilizing the grouping strategy. Using clustering the best representative from the majority class are chosen and the training chunk is balanced accordingly. A portion of the under-examining based methodologies are talked about beneath.

Zhang *et al.*[10] proposed another calculation to manage skewed information streams. They utilized clustering+sampling calculation to manage skewed information streams. Testing was completed by utilizing k-implies calculation to frame groups of negative cases in the present preparing piece and afterward they utilized the centroids of each of the bunches shaped to speak to each of those bunches. Number of bunches framed were equivalent to the quantity of positive cases in current preparing clump and therefore current preparing group was refreshed by taking every positive case alongside centroids of the groups of negative examples. Another classifier was made on these examined occurrences. Additionally size of the outfit was settled so for all classifiers show in the troupe alongside new classifier based on inspected examples. Area under ROC curve was utilized as measure to choose best classifiers to be incorporated into the group. Weights of the classifiers were allocated on the premise of AUROC ascertained and outfit along these lines shaped was utilized to order the occasions in the test piece.

Cost Sensitive Learning

Cost sensitive learning is one of the most important technique of data mining. It allocates distinctive estimations of wrongly classified punishments to each class. It has been consolidated into order calculations by considering the cost data and endeavouring to advance general cost amid the learning procedure. In cost sensitive characterization the issue is managed by modifying the learning. It works by making costs related with wrongly classified instances of minority class.

1.2 Motivation

While chipping away at distinguishing proof of the undertaking theme in the region of information mining we found that part of work has been as of now done in the diverse regions of the information mining as for the under-sampling and over-sampling systems. Then we found that the information inborn qualities which very impact the execution of the classifier has not been altogether considered. After this we focused on different real-world applications where the information inborn qualities came into the picture. Different applications like Network interruption recognition, Financial misrepresentation discovery and so forth are different zones which are characterised by stream information. We generally focus on the classifier and attempt to control them to accomplish better outcomes, however separated from the classifier properties the data inborn properties have high impact on the classifier performance. So our point here is to think about these data characteristic properties and endeavour to determine these issues keeping in mind the end goal is to show signs of improvement in the classification.

1.3 Research objective

In this project, we focus on the data intrinsic characteristics such as small disjuncts, lack of density, oversampling, the presence of noise, Borderline examples and data shifts and try to analyse each of these in detail on how they affect the classification results. After closely analysing these characteristics, our aim is to develop algorithmic solutions for each of these characteristics in order to address the issues that arise due to them. The main target is to resolve these issues to some extent, so as to improve the accuracy of the classifier.

1.4 Thesis Organization

This project report has been divided into 5 chapters. Each chapter deals with one component related to this thesis. Chapter 1 being introduction to this thesis, gives us the brief introduction about this project, thereafter chapter 2 tells about the related work. Following up is Chapter 3 which tells all about the proposed work. Chapter 4 provides us with the results followed by the final chapter, ch 5 that is the conclusion of this thesis.

CHAPTER-2
RELATED WORK

To make the imbalance datasets get properly categorised by the classifier, is one of the main issues in data mining . Imbalance is a term given to a scenario when data elements of one class are much lower than the other.This scenario occurs in many real-world cases, so its very important to address this issue. We have surveyed the attributes of the imbalanced dataset situation in characterisation, exhibiting the particular measurements for assessing execution in class imbalanced learning and identifying the proposed arrangements. Some of the papers to which we have refer to get better understanding the main issues related to using data intrinsic characteristics in the classification problem. Lopez *et al.*[1] highlighted these issues and proposed some raw solutions to how to get rid of these problems. A Technique given by Pruengkarn *et al.*[2] by forming a hybrid of the Synthetic Minority Oversampling Technique (SMOTE) and Complementary Fuzzy Support Vector Machine (CMTFSVM). This technique uses an optimised version of the membership function to get improved classification results. The comparison is done using three different classifiers. In the performed experiment a real world dataset is used and four standard datasets. The results showed that the hybrid formed using Complementary Fuzzy Support Vector Machine and Synthetic Minority Oversampling Technique gave better results as compared to that of other FSVM classifiers. So it could be concluded that the proposed work performed well when given an imbalanced dataset.

Taking in classifiers from imbalanced or skewed datasets is a critical point, emerging all the time practically speaking in characterization issues. In such issues, every one of the occurrences are marked as one class, while far less examples are named as alternate class, more often than not the more essential class. Clearly customary classifiers looking for an exact execution over a full scope of occasions are not reasonable to manage imbalanced learning assignments, since they have a tendency to group every one of the information into the greater part class, which is typically the less vital class. The investigation propose by Kotsiantis *et al.*[3] depicts different

strategies for dealing with awkwardness dataset issues. Obviously, a solitary article can't be a total audit of the considerable number of strategies and calculations, yet we trust that the references referred to will cover the major hypothetical issues, controlling the analyst in fascinating exploration bearings and recommending conceivable predisposition mixes that still can't seem to be investigated.

Information streams can be non-uniformly distributed but the classifiers accept moderately adjusted and stable information streams and can't deal with well rather skewed streams which are common place in numerous information stream applications. Class unevenness in such skewed information streams can be seen in numerous certifiable applications. In such situations gaining from skewed information streams brings about grouping one-sided towards the larger part class which brings about miss characterisation of minority class cases. The misfortunes related with miss order of minority classes can be higher in a few applications. A work was proposed on handling skewed classes by A.Godase, V.Attar[4]. In the paper they have introduced the system on the most proficient method to manage characterisation of the information streams with skewed dispersion of classes. They have utilised a group strategy of over-examining the minority class occasions utilising K closest neighbour calculation. The cases of minority class continues expanding with each preparation cycle with the limitation connected by the KNN calculation.

Examination of information streams is turning into a key zone of information mining research, as the quantity of utilizations requesting such preparing increments. Present day data innovation enables data to be gathered at a far more noteworthy rate than at any other time. Machine learning offers guarantee of an answer, yet the field for the most part concentrates on accomplishing high precision when information supply is restricted. While this has made modern arrangement calculations, many don't adapt to expanding informational collection estimate. At the point when the informational collection estimate gets to a point where it could be considered to speak to a persistent supply or information stream then incremental order calculations are required. While handling with non-stationary ideas, troupe of classifiers has a few favourable circumstances over single classifier techniques. They are anything but difficult to scale and

parallelise, they can adjust to change rapidly by pruning failing to meet expectations parts of the group and they subsequently for the most part create more precise idea depictions and proficient outcomes. Be that as it may, the viability of a calculation can't just be surveyed by precision alone. Thought should be given to the memory accessible to the calculation and the speed at which information is handled as far as both the time taken to anticipate the class of another information test and the time taken to incorporate this example in an incrementally refreshed arrangement show. There is a method proposed by Attar *et al.*[5] for a quick and light classifier for information stream order.

Work proposed by V. Chawla [11] provides with different combinations of techniques to balance the skewed data distribution. It looked at some different performance metrics as the previous known metrics, such as, accuracy seemed to be inappropriate. Apart of providing with a new technique to classify imbalanced dataset this study also focused on a particular set of performance metrics that could be more precise in evaluating the results.

One of the major problem is that we never know what is the best sample size of training get to be used to get best out of the classifiers. A study was carried out related to this issue. M. Weiss and F.Provost [12] studied how different classifiers gave different results in case of different class distributions. There was not a stand-out classifier which gave optimal results in all cases. They proposed of a budget-sensitive technique to get optimal training set which yielded a classifier that gave near optimal classification performance.

CHAPTER-3
PROPOSED WORK

In the previous section we have looked at some of the techniques applied in the field of classification. Our purpose is to focus on the data itself and not on the classifier. In this Section we will be looking into this data, the characteristics related to it. These characteristics give rise to various problems (which we will look further) that highly impact the performance of the classifier. After that we will be looking at the proposed solutions on how to tackle with the issues related to these data intrinsic characteristics.

2.1 Issues that are been addressed

2.1.1 *Small disjuncts*

The nearness of the imbalanced classes is firmly identified with the issue of little disjuncts. This circumstance happens when the ideas are spoken to inside little bunches, which emerge as an immediate consequence of underrepresented subconcepts. In spite of the fact that those little disjuncts are understood in a large portion of the issues, the presence of this sort of territories exceedingly builds the multifaceted nature of the issue on account of class unevenness, since it turns out to be difficult to know whether these illustrations speak to a genuine subconcept or are only credited to commotion.

2.1.2 *Lack of density*

One issue that can emerge in arrangement is the little specimen measure. This issue is identified with the "absence of thickness" or "absence of data", where acceptance calculations don't have enough information to make speculations about the appropriation of tests, a circumstance that turns out to be more troublesome within the sight of high dimensional and imbalanced information. it turns out to be hard for the learning calculation to get a model that can play out a decent speculation when there isn't sufficient information that speaks to the limits of the issue

and, what it is likewise most critical, when the centralization of minority cases is low to the point that they can be just regarded as commotion.

2.1.3 *Overlapping or class separability*

Overlapping happens when two dissimilar classes are very closely related to each other. Two dissimilar classes have a common range in which their data elements lie. Due to this issue it is very hard to categorise a data element laying in the overlap region, to a particular class. There are a few works which mean to consider the connection amongst covering and class irregularity. One can discover an examination where the creators propose a few trials with engineered datasets shifting the unevenness proportion and the cover existing between the two classes. Their decisions expressed that the class probabilities are not the fundamental explanation behind the frustrate in the grouping execution, yet rather the level of covering between the classes.

2.1.4 *Noisy data*

Noise can be present in any system. But in our case it has a bigger impact because of its resemblance with the minority class in terms of numbers. It becomes very difficult to separate the noise and the minority class instances. Pruning is a techniques using which we can reduce noise, but their is risk of loosing the data elements of the minority class also. So in our case we will not recommend to use pruning to remove the noise.

2.1.5 *Borderline examples*

Borderline problem can be expressed as overlapping at the edges between different classes. It becomes more difficult when the noise is also present at the borders.At the edge the elements of minority class are closer to the cluster of majority class instances as compared to the distance from their own cluster and same is for the majority class instances present at the edges, they are closer to the cluster of minority class as compared to their own cluster.

2.1.6 Dataset shift

Dataset shift problem is when some of the instances of a class are closer to the cluster of other class as compared to their own class cluster. It is another one of the most important issues that arises due to the skewed distribution of datasets. Here few instances of other classes are more closer to the cluster of other dissimilar instances than to their own type of group. So it makes it very hard for the classifier to properly detect that particular instances and not mistake them as noise.

2.2 Proposed solutions

2.2.1 Solution to Small disjuncts

We have seen various solutions provided in the related work study and we will be focusing on one of these. Their has been some discussion on getting additional data that will increase the number of minority class instances, which will help us in better classifying the minority class.Rather then increasing the data elements for both the minority and majority class, we will be increasing the number of elements for just the minority class as it is the one which we have to identify better and the majority class instances are already present in abundance so no need to increase them further.

Algorithm

- **Input:**

- Training chunk T_i ,Test chunk B_i arriving at current time-stamp S_i

- CD - A new set which contains all the positive instances from previous training data chunks.

- We calculate the number of positives instances to that of the negative instances (let the ratio be b) and we consider a balance ratio, lets say b.
- Threshold Value T

- **Output:**

- Classifier C, for the test chunk B_i begin divide the current training chunk into P and N containing positive and negative examples respectively. if b > = d then Stop (the Balance has been achieved, else

- Refine positive classes instances R_i from the CD set according to the threshold value T.

- Embed the refined positive classes instances R_i into the current training set B_i

- Classify again to check for the balance between positive class instances and the negative class instances if b > = d then Stop (the Balance has been achieved), else repeat the above procedure in a recursion until we achieve the terminating condition i.e., b > = d.

- Note that the CD set is now also containing the positive class instances from this training set and act as a input for the next test set.

2.*2.2 Solution to Lack of density*

The issue of lack of density give rise to the problem of small disjuncts. So if we remove this problem the issue of small disjuncts will be removed automatically. For resolving this issue we propose a solution where a new temporary class is generated between two relatively closer classes so as to increase the density and have a uniform distribution of elements on a whole.

Algorithm

Input: I= Dataset containing non-uniformly distributed data.

Output: O= Equivalent frequency for each class in dataset (i.e., having equivalent distributed amount of data within a range, representing different classes) by mathematically producing synthesised data.

Step 1: Partition the dataset into categories (i.e., similar data in same class and dissimilar data in different classes). Lets say class C1 and C2.

Step 2: Applying averaging between classes that are relatively closer to each other, so as to gain new set of data elements (say G_1) that lie between the classes with whose average they have been formed. These new set of data elects will form a temporary class of their own.

Step 3:Repeat the above procedure recursively to generate data (G_i) that will fill up the gaps between the classes whose data events are far too separated from each other, so as to remove the lack of density.

Step 4:Embed the new data elements (G_i) into the initial data-set and we will have an equally distributed type of data within a given range.

2.2.3 Solution to Overlapping or class separability

In various circumstances of irregularity and cover concentrating on the kNN calculation was produced. For this situation, the creators proposed two unique systems: from one perspective, they attempt to discover the connection when the unevenness proportion in the cover district is like the general lopsidedness proportion while, then again, they scan for the connection when the awkwardness proportion in the cover area is converse to the general one (the positive class is locally denser than the negative class in the cover area). We recommend applying number juggling control to the unique components in the covering area, so as to isolate them and consequently diminishing the covering.

Algorithm

Input: I= Dataset containing different classes of data with overlap between the elements of these classes.

Output: O= Each class well defined within its own space without any overlapping.

Step 1: Partition the dataset into categories (i.e., similar data in same class and dissimilar data in different classes). Lets say class C1 and C2.

Step 2: Applying addition of fixed value for each element in the set classes that are relatively overlapping to each other(choose a relatively big number and apply to one of the overlapping element in the set), so as to gain new set of data elements (say S1) that lie distant into different classes. These new set of data elects will form a temporary class of their own.

Step 3:Repeat the above procedure recursively (procedure used to remove the issue of lack of density) to generate data (D_i) that will fill up the gaps between the classes whose data events are far too separated from each other, so as to remove the overlapping.

2.*2.4 Solution to Noisy data*

As both the noisy data the positive class instances are present in small amounts, they are usually misunderstood as each other, which leads to very misleading classification results. We need to study both the noise data and the positive class instances in a very careful manner, so as to classify them properly. We suggest on recognising the potential areas which may contain noise and decrementing the value of each element in that cluster by the average value of the whole cluster. This will result in degradation of the noise, which further will result in better classification.

Algorithm

Input: I= Dataset which contains redundant data along with the useful information.

Output: O= The reduction in noise by degrading the value of the complete noise prone regions.

Step 1: Partition the dataset into categories (i.e., similar data in same class and dissimilar data in different classes). Lets say class C1 and C2. We will also see a new set of elements having

dissimilar properties to both the above classes i.e., the noise. As it will be hard to distinguish between the noise elements and the elements of the minority class (as they both are present in small amount sand it will be hard of us to study them and classify them properly), we will consider the complete set of those elements (elements that are under the impression of consisting the noise) for the removal of the noise.

Step 2: Taking the average of the all the elements of the set elements in which noisy data exists. Subtract average value from the noisy data to neutralize the noise of the data element in the set.

Step 3:Repeat the above step in a recursive manner until the average of all the elements of the set elements in which noisy data exists is under the acceptable threshold value

2.*2.5 Solution to Borderline examples*

In the borderline the instances of both the positive class and negative class are present in an overlapped manner. Blinding classifying them may result in misleading results. We suggest, first we should find out which instance in the borderline region belongs to which class, then we apply some arithmetic degradation operation to each of these instances so as to pull them back club them together with the instances of the similar class.

Algorithm

Input: I=Dataset with different classes of data that are not cleanly defined over their cluster boundaries.

Output: O= Dissimilar class elements far separated from each other (means that there will be large borderline margins between different classes).

Step 1: Partition the dataset into categories (i.e., similar data in same class and dissimilar data in different classes). Lets say class C1 and C2. Now consider the dissimilar elements that are overlapped at the borderlines.

Step 2: subtract a common value from each value of border line so that the set value can be fetched back into the set lies into the border.

Step 3:Repeat the above procedure recursively until we have attain a minimum borderline margin between different classes

Step 4:Check for the elements whose values have been subtracted in step 2, and be sure that these values are grouped together with the right set of clusters (i.e., checking for the class to which they belong).

2.2.6 *Solution to Data Shifts*

Data shift means that the instances of similar class are far apart from each other. So a situation might rise up where the positive class instance is close to the cluster of the negative class instances, In such a case the classifier will misjudge the positive class instance and they will result in misleading results. So we suggest that we analyse the outliers in each cluster by comparing the values of each element with each other within the cluster and then using the same technique to assign the outliers to their correct classes.

Algorithm

Input: I= Dataset consisting of data elements that are not well defined within their space.

Output: O = Set of similar clusters with on outliers.

Step 1: Analyse each class for the presence of outliers.

Step 2: compare each value of the set. If certain value is found to be the outlier of the set then shift that data from containing set to other set where that value will be considered to be the real value. We have to be careful as this could also be the noise.Compare the outlier will all the other sets one-by-one for resemblance.

Step 3: perform the above operation for each outlier for all the sets recursively till all the outliers are shifted to their appropriate sets.

CHAPTER-4
EXPERIMENTAL RESULTS

3.1 Performance Measures

We use FOUR evaluation metrics: Precision, Recall, F-measure and AUROC.

Confusion matrix:-

It maps the relation between what the model has predicted and what the actual result should be. If the predicted class is positive and actual class is positive as well, then we get the true positive section. If the predicted class is positive but actual class is negative then we get false positive section, on similar bases if the actual class is positive but the predicted class in negative then it is false negative and if the actual class is negative and predicted class is also negative we get true negative section.

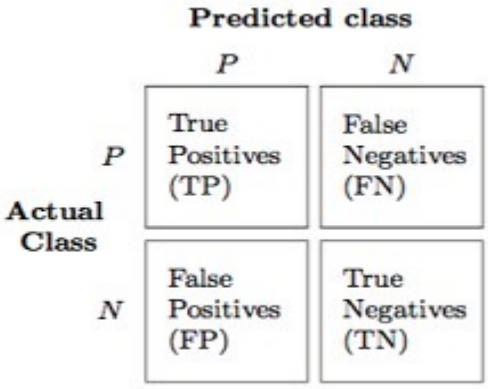

Figure 3.1.1: Confusion Matrix

We consider the possible outcomes and define: precision, recall, F-measure and Area under ROC as -

Precision: the ratio of data elements that are correctly classified (for both the minority and majority class) to total number of classified instances.

P =TP/ (TP + FP)

Recall: The ratio of the minority class instances that are correctly classified to the total number of actual minority class instances.

R =TP/ (TP + FN)

F-Measure: F-measure is calculated by taking the harmonic mean of Precision & Recall. We can say that it is essentially an average between the two percentage. It really simplifies the comparison between the classifiers.

F-measure = 2 / (1/Recall + 1/Precision)

Area Under ROC Curve: The area under ROC Curve(Receiver Operating Characteristics) gives the probability that, when one draws one majority and one minority class example at random, the decision function assigns the higher value to the minority class than the majority class sample. AUROC is not sensitive to the class distributions in the dataset. Generally it is plotted as a True Positive Rate verses False Positive Rate.

3.2 Experimental Setup

Requirements

A synthesised dataset is formed by collecting data from an online repository named MOA (Massive online analysis) [7]. I have considered the Airlines Dataset. We have considered three different training sets containing the airlines data for my experiment. The code is written in java language. To implement it we have used the NetBeans 8.1 Software and the operating system used is MAC OS Sierra version 10.13.4.

About Net Beans 8.1

NetBeans IDE is the authority IDE for Java 8. With its editors, code analyzers, and converters, you can rapidly and easily update your applications to utilize new Java 8 dialect develops, for

example, lambdas, practical activities, and strategy references.
Group analyzers and converters are given to look through various applications in the meantime, coordinating examples for change to new Java 8 dialect develops.

In this project the NetBeans software has been used to implement the algorithms on raw dataset to get well refined dataset.
Its also useful in creating charts and figures which help us understand and analyse the data even better. Using multiple read operations in our code different files were read from the directory to extract features from each file and embed into one file

Bar charts were constructed using JFrame library to get good visualisation of our data.

The supervisor supports numerous dialects from Java, C/C++, XML and HTML, to PHP, Groovy, Javadoc, JavaScript and JSP. Since the editorial manager is extensible, you can connect to help for some different dialects.

About Mac OS Sierra

macOS High Sierra (variant 10.13) is the fourteenth real arrival of macOS, Apple Inc's. work area working framework for Macintosh PCs. The successor to macOS Sierra, it was declared at the WWDC 2017 on June 5, 2017.

The name "High Sierra" alludes to the High Sierra region in California. Likewise with Snow Leopard, Mountain Lion and El Capitan, the name additionally suggests its status as a refinement of its ancestor, concentrated on execution upgrades and specialised updates instead of client highlights. Among the applications with eminent changes are Photos and Safari.

Apple File System

Apple File System (APFS) replaces HFS Plus as the default document framework in macOS out of the blue with High Sierra. It underpins 64−bit inode numbers, is intended for streak memory, and is intended to accelerate normal undertakings like copying a document and finding the span of an envelope's substance. It likewise has built−in encryption, crash−safe insurances, and disentangled information reinforcement in a hurry.

Metal 2

Metal, Apple's low-level illustrations API, has been refreshed to Metal 2. It incorporates virtual-reality and machine-learning highlights, and additionally bolster for outside GPUs. The framework's windowing framework, Quartz Compositor, bolsters Metal 2.

Media

macOS High Sierra includes bolster for High Efficiency Video Coding (HEVC), with equipment increasing speed where accessible, and also bolster for High Efficiency Image File Format (HEIF). Macintoshes with the Intel Kaby Lake processor offer equipment bolster for Main 10 profile 10-bit equipment disentangling, those with the Intel Skylake processor bolster Main profile 8-bit equipment interpreting, and those with AMD Radeon 400 arrangement illustrations likewise bolster full HEVC deciphering. In any case, at whatever point an Intel IGP is available, (for example, macbooks), the structures will just direct demands to Intel IGP. What's more, sound codecs FLAC and Opus are likewise upheld, however not in iTunes.

This was the System/Application features that were used to implement the algorithms and to form a new dataset. Now this dataset is tested using WEKA 3.8.2 and different classifiers will be used to test the dataset.

WEKA

It is a machine learning based software that has been coded in Java Language. WEKA tool provides with different set of features so that the data can be processed, visualised and analysed. It takes files of extension name .arff and the data is segregated based on the type of the data. Each feature could be individually studied. It provides with filters which could help you in modifying your data according.

There are different segments of processing available that can be performed on the data. In the classifier section which has been used by us consists of different set of classifiers like for instance: K-nearest neighbour, Support vector machine, Random Forest, Logistic regression and many more.

It provides us with an environment where we can train and test our data separately also and combined also. There are validation available for example K-cross validation where one instance is used for testing and the K-1 instances are used for training purpose.

It provides with different type of performance metrics like:-

Mean Square Error

Absolute Error

Precision

Recall

F-measure

ROC Area

The mentioned above measures Precision, recall, F-measure, ROC Area are the classification performance metrics that have been used in this project to analyse the results

There are different sections available like clustering and others which could be used to analyse the data accordingly.

Dataset Formation

We will be using different sets of such datasets, because we will be filtering out minority class instances from the previous datasets and embed them into the current dataset to get a balance between the number of instances of the majority and the minority class.

We are working on binary classification, i.e. one class is the minority class (the one with the less number of instances) and the other is the majority class (one with the dominating number of instances).

The binary classification is formed on the airlines dataset where the two classes are - the majority class (that is the flight is on time) and the minority class (that is the flight got delayed).

Data is considered from three different sources:-

- MOA (Massive Online Analysis) Repository
- US Department of Transportation (BTS)

The airlines data is considered from these real datasets and modified to form the synthetic dataset to show how all the algorithms would workout together.

- Below is shown how the dataset (unprocessed) looks like:-
- The extension of the dataset file is .arff (so that we can perforation classification in WEKA tool)

@RELATION dataset1

@ATTRIBUTE attribute_0 {9E,AA,AS,B6,CO,DL,EV,HA,OH,OO,UA,US,XE,YV}
@ATTRIBUTE attribute_1 REAL
@ATTRIBUTE attribute_2
{ACV,ALB,ANC,ATL,BHM,BIS,BNA,BOS,BQN,BTR,BTV,BWI,CAE,CAK,CHS,CLE,DAY,
DEN,DHN,DLH,DSM,ECP,EKO,EWR,FAI,FAR,GFK,GSO,HNL,HRL,ICT,IYK,JAX,LAS,L
AX,LIT,LMT,LWS,MAF,MCO,MFE,MFR,MGM,MKE,MSN,MSP,MSY,MYR,OMA,ONT,OR

D,PDX,PHL,PHX,PNS,PSE,PWM,RIC,RST,SAT,SAV,SBN,SEA,SFO,SLC,TPA,TUS,TVC,V
PS}

@ATTRIBUTE attribute_3
{ATL,CLT,DEN,DFW,DTW,HNL,IAD,IAH,ITO,JFK,KOA,LAX,LIH,MCO,MIA,MSP,OGG,
ORD,PDX,PHL,PHX,SEA,SFO,SLC,SMF}

@ATTRIBUTE attribute_4 REAL

@ATTRIBUTE attribute_5 REAL

@ATTRIBUTE attribute_6 {0,1}

@DATA

CO,269, SFO, IAH ,15.05,15.40,1

US,1558, PHX, CLT ,15.10,15.50,1

AA,2400, LAX, DFW ,20.00,20.43,1

AA,2466, SFO, DFW ,20.06,20.56,1

AS,108, ANC, SEA ,22.00,22.00,0

CO,1094, LAX, IAH ,23.11,23.43,1

AA,674, ORD, MIA ,4.00,4.29,1

B6,728, BQN, JFK ,5.18,5.20,0

CO,463, ORD, IAH ,13.10,13.49,1

OH,6423, DAY, DTW ,14.01,14.01,0

OO,6698, MFR, PDX ,23.45,23.50,0

XE,2865, MKE, IAH ,6.00,6.00,0

EV,7101, BTV, IAD ,7.00,7.10,0

…

MOA (Massive Online Analysis) Repository

MOA (Massive On-line Analysis) [7] is a system for information stream mining. It incorporates apparatuses for assessment and an accumulation of machine learning calculations. Identified with the WEKA venture, it is additionally composed in Java, while scaling to all the more requesting issues. The objective of MOA is a benchmark system for running investigations in the information stream mining setting by demonstrating

- storable settings for information streams (genuine and engineered) for repeatable examinations
- an arrangement of existing calculations and measures frame the writing for examination and
- an effectively extendable structure for new streams, calculations and assessment techniques.

MOA currently bolsters the examination of active learning (AL) classifiers. active learning is a subfield of machine learning, in which the classifier effectively chooses the cases it utilizes as a part of its preparation procedure. This is important in territories, where acquiring the name of an unlabeled case is costly or tedious, e.g. requiring human connection. AL by and large lessens the measure of preparing examples expected to achieve a specific execution contrasted with preparing without AL, in this manner diminishing expenses. An extraordinary review on active learning is given by Burr Settles.

In stream-based (or successive) active learning, one example at any given moment is introduced to the classifier. The active student needs to choose whether it asks for the occurrence's name and uses it for preparing. Ordinarily, the student is constrained by a given spending that characterizes the offer of names that can be asked.

This refresh to MOA presents another tab for active learning. It gives a few augmentations to the graphical UI and as of now contains numerous AL systems.

US Department of Transportation (BTS)

The U.S. Division of Transportation's (DOT) Bureau of Transportation Statistics tracks the on-time execution of household flights worked by huge air bearers. Rundown data on the quantity of

on-time, postponed, wiped out, and redirected flights is distributed in DOT's month to month Air

Travel Consumer Report and in this dataset of flight deferrals and cancelations.

3.3 Implementation

Using our algorithm for small disjuncts the imbalance between the majority and minority class is decreased, so that the classifier is properly able to study the minority class and is able to correctly classify such instances in the future. Different sets of data were used so that current training dataset could be more balanced using the minority class instances of the previous datasets. Net Beans 8.1 was used to implement the algorithm :-

RESULTS AND SCREENSHOTS

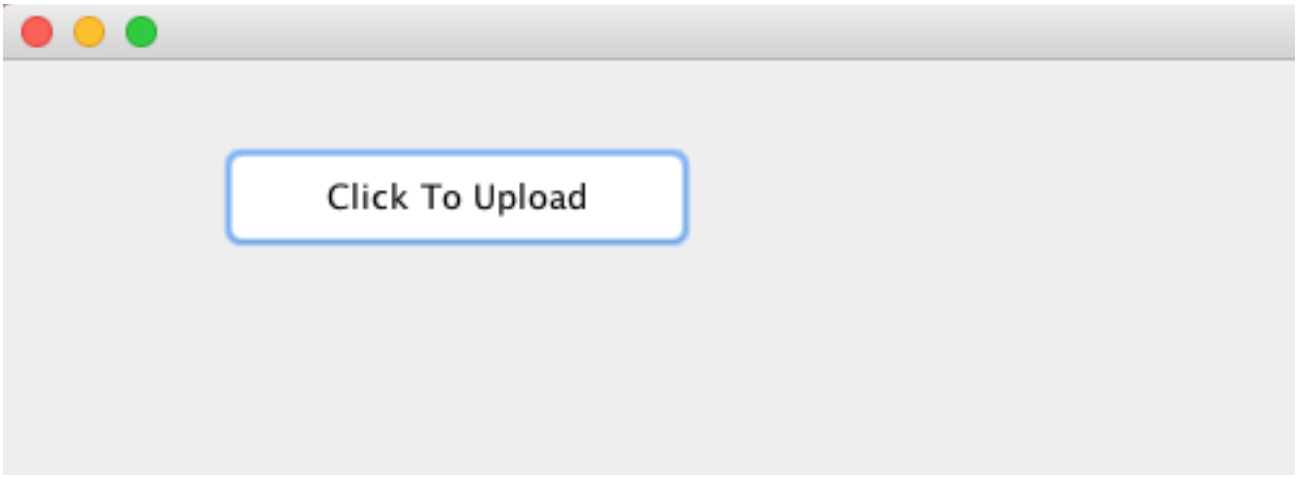

Figure 3.3.1: The Java Application first running phase

A Java application has been created, in which the dataset files are uploaded and the imbalanced is calculated accordingly. The imbalance is managed by adding minority class instances into the current training dataset. The above figure shows a window when the code is executed and it asks the user to upload the files.

Figure 3.3.2: Count of instances of various datasets

DISCUSSION

This shows the count of each class for all the various datasets. In this we have the count for the numbers of flights that were on time (majority class) and the number of flights that got delayed (minority class). In dataset 2 the few of the minority class instances from the dataset 1 (i.e. the previous dataset) are added to the current dataset (dataset2) to get some balance between the minority class and the majority class. The threshold for balance is set according to-

Number of minority class instances/Number of majority class instances >= 0.5

If this equation is satisfied we stop, otherwise we go to the next dataset i.e. the dataset 3 and we do the similar steps to achieve the balance.

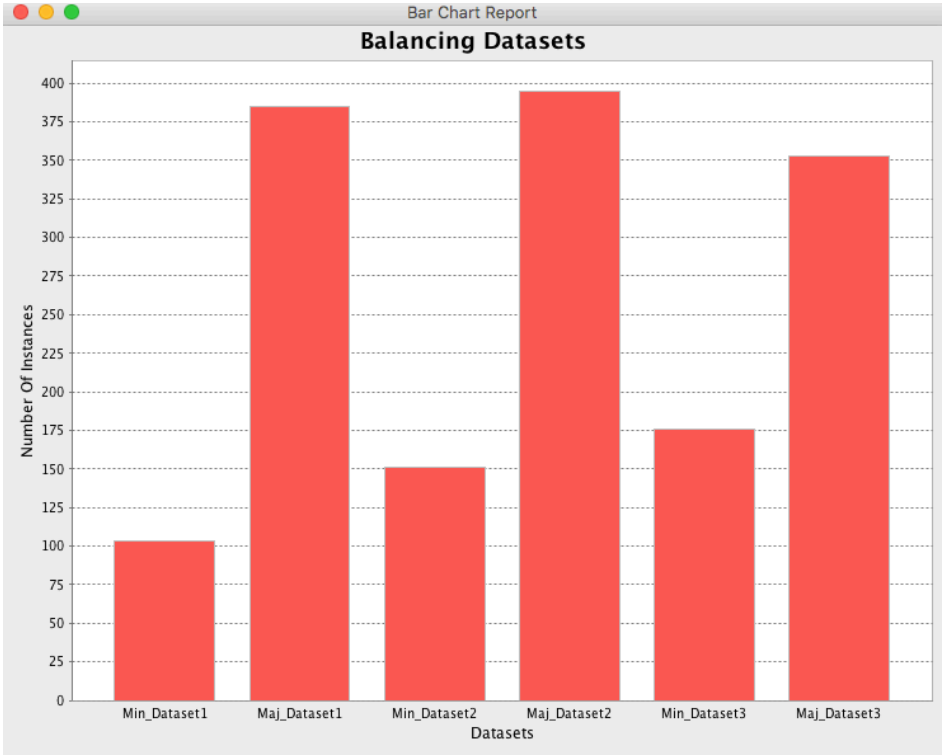

Figure 3.3.3: The bar chart of different classes based on number of instances

DISCUSSION

This figure shows the Bar Chart for the same distribution that is shown in fig-3.3.2 for better visualisation.

In the x-axis we have the various datasets each with the majority and minority class count and in the y-axis we have the number of instances that are present in each class of each dataset.

In the second phase of implementation the algorithms produced for -

- Lack of density
- Overlapping or class separability
- Noisy data
- Borderline examples
- Dataset shift

These algorithms are applied together so that we can get a refined dataset free from all theses issues that have arose due to the data intrinsic characteristics. We achieve a highly refined dataset which consists of two different well formed, well balanced classes without any intervention into each other's instances.

The transformed dataset look like:-

UA	756	SFO	DEN	16.16	16.18	2	0
9E	3718	ERI	DTW	23.1	23.5	39	1
B6	107	BUF	JFK	1	1	0	0
DL	1248	IAH	ATL	3	3.2	40	1
DL	1985	DAY	ATL	23.1	23.15	4	0
DL	2387	MKE	ATL	23.39	23.59	40	1
DL	2405	PHL	DTW	0.1	0.1	0	0
DL	734	MCI	ATL	17.1	17.13	2	0
EV	5165	AGS	ATL	19.03	19.05	1	0
MQ	3014	FAT	LAX	13.14	13.15	0	0
MQ	3332	LBB	DFW	18	18	0	0
MQ	3632	CLT	MIA	4.04	4.24	39	1
MQ	3781	LIT	ORD	2	2.09	8	0
MQ	3892	PHL	ORD	5.5	5.5	0	0
MQ	4013	LEX	ORD	14.4	14.5	10	0
OH	6629	GRR	DTW	7.08	7.08	0	0
9E	4391	MBS	DTW	7	7.1	9	0
HA	218	HNL	KOA	17.1	17.1	0	0
OO	6921	MOD	SFO	15	16	60	1
9E	3935	EWR	DTW	19.1	19.45	35	1
9E	3955	LBB	MEM	23.1	23.13	2	0
9E	4200	TYS	CVG	3	3	0	0
9E	4274	AMA	MEM	4	4	0	0
B6	735	JFK	BQN	3.59	4.05	46	1

Figure 3.3.4: The processed dataset after algorithmic transformation

- The classes have been formed as following:-

0-10 mins	no delay (majority class)
30-60 mins	delay (minority class)
11-29 mins	overlapping region
Above 60	noise

- Distribution for overlapping region:-

Case 1:

if flights are at day time

from 6:00 am to 5:00 pm

assign it class - no delay (majority class)

Case 2:

if flights are at night time

from 5:01 pm to 5:59 pm

assign it class - delay (minority class)

- The data shift problem is automatically solved as the algorithm makes the dataset again from scratch, so any misclassified instances are classified into proper classes.

- Lack of density could arise when we are working with more than 2 classes and the algorithm to solve this issue has been provided above.

- The issue of borderline examples is resolved alongside with the overlapping problem.

- So at last we have 2 well refined classes that are well balanced and are properly separated from each other alongside with the reduced noise. This all will help the classifiers to properly classify the instances of both the classes and hence, we will get highly efficient results.

- Different classifiers have been used to showcase the difference in the performance of the classifiers with the unprocessed dataset and then the transformed dataset.

Below is the list of the various classifiers that have been considered:-

- Naïve Bayesian Model
- Nearest Neighbors Model (K-NN)
- Support Vector Machine Model (Sequential Minimal Optimization SMO in WEKA)
- Random Forest Model
- Logistic Model

Brief Introduction of the classifiers:-

Naive Bayes

This classifier belongs to the probabilistic group of classifiers in the domain of machine learning. The bases of this classifier is the Bayes Theorem where the features are considered to be independent of each other. It is a very popular when it comes to classification. It is a simple model where the test (unknown) instances are assigned class tags based on the trained model.

This classifier can be used with the help of different tools. In our project we have implemented the naive Bayes classifier using the WEKA tool. It is present under the classification section of WEKA tool.

K-NN

K-nearest neighbors model can be used as classification model or regression model. For an unclassified instance as the input we consider the k classified instances in a constraint region and accordingly the unclassified instance is given a class whose instances are most in that region.
In case K=1, the unclassified instance is given the class whose neighbour is nearest to it, their is no need for count as the value of k is 1.

SVM

SMO - Sequential minimal optimization helped the support vector machine (SVM) with the problem of quadratic programming. It was developed at the Microsoft Research in 1988 by John Platt. SMO is used in the training phase of the SVM so as to get rid of the problem. It was quiet an important development as in early days if was very expensive to get rid of the quadratic programming problem of SVM using 3-party softwares.

Random Forest

Also known as - Random decision forests, It is an ensemble learning technique used for both regression and classification. It works by generating large number of decision trees in the training phase and in the test phase gives the result according to whether it is for classification or regression. It is better than decision tress as it removes its limitation of getting too precise depending on the training dataset. Its first creation was done by Tin Ham Ho in the year 1995.

Logistic Model

Logistic technique in terms of statistics is a method that is applied when the dependent feature is binary. The binary values can be assigned as ‘1’ and ‘0’ representing the cases like pass or fail, win or loss respectively or vice-versa. In our case we have a binary setup one with the class-delay and other with the class- no delay. If we are dealing with more than 2 classes for that we

have a method called multinomial logistic regression and further their are different branches of logistic model that work on different types of data.

• 10-folds cross validation is used to test and train the data.

Classifier	Unprocessed Dataset (Original)		Processed Dataset (Algorithmically Transformed	
	Correctly classified instances	Incorrectly classified instances	Correctly classified instances	Incorrectly classified instances
Naive Bayes	61	39	75	10
K-NN	60	40	71	14
SMO	62	38	70	15
Random Forest	64	36	64	21
Logistic Model	56	44	57	28

Table 3.3.1: The correctly and Incorrectly classified instances by various classifiers in different datasets

Discussion

As we can see from above table every classifier performs better with the processed dataset that has been processed using our proposed algorithms. The Incorrectly classified instances have decreased drastically, thus helping us with our problem.

• External noise was introduced using the WEKA tool to make the calculations more realistic.

In WEKA Tool-

Filters

-> Unsupervised

-> attribute

-> AddNoise

- Now we will be comparing the classifiers performances based on the performance metrics that were mentioned earlier.
- Precision
- Recall
- F-Measure
- ROC Area

Classifier	Unprocessed Dataset (Original)		Processed Dataset (Algorithmically Transformed	
	Precision	**Recall**	**Precision**	**Recall**
Naive Bayes	0.582	0.610	0.882	0.882
K-NN	0.584	0.60	0.834	0.835
SMO	0.570	0.620	0.823	0.824
Random Forest	0.56	0.64	0.750	0.753
Logistic Model	0.599	0.56	0.684	0.671

Table 3.3.2: The Precision and recall obtained by various classifiers by processing different datasets

Classifier	Unprocessed Dataset (Original)		Processed Dataset (Algorithmically Transformed	
	F-Measure	**ROC Area**	**F-Measure**	**ROC Area**
Naive Bayes	0.594	0.425	0.882	0.866
K-NN	0.591	0.505	0.834	0.841
SMO	0.588	0.490	0.820	0.80
Random Forest	0.583	0.488	0.746	0.834
Logistic Model	0.575	0.501	0.674	0.703

Table 3.3.3: The F-Measure and ROC Area obtained by various classifiers by processing different datasets

Discussion

As we can see from above tables the classifiers performs way superior on with the processed dataset as compared to when with unprocessed dataset. All the classifiers perform well, and the Naive Bayes and K-NN stands-out with the best performance results as compared to other dataset. Naive Bayes sees to perform with the highest Precision, Recall, F-Measure and ROC Area. But and the end of the day what we wanted was that our algorithmically transformed dataset performs better than the original dataset. These results shows that if we use the proposed algorithms, We will get very superior results independent of which classifier you are going to choose, it works well for everyone. After addressing all the issues that arose due to data intrinsic characteristics our algorithms have handled them pretty well giving us very high performance results.

- Presenting the Bar charts for the performance results mentioned in above table for better visualisation and analysis.

Bar Charts

It represents the categorical data in the graph in the form of rectangles with one-axis representing the type and the other axis representing the quantity. It could be plotted in both horizontal state and vertical state.

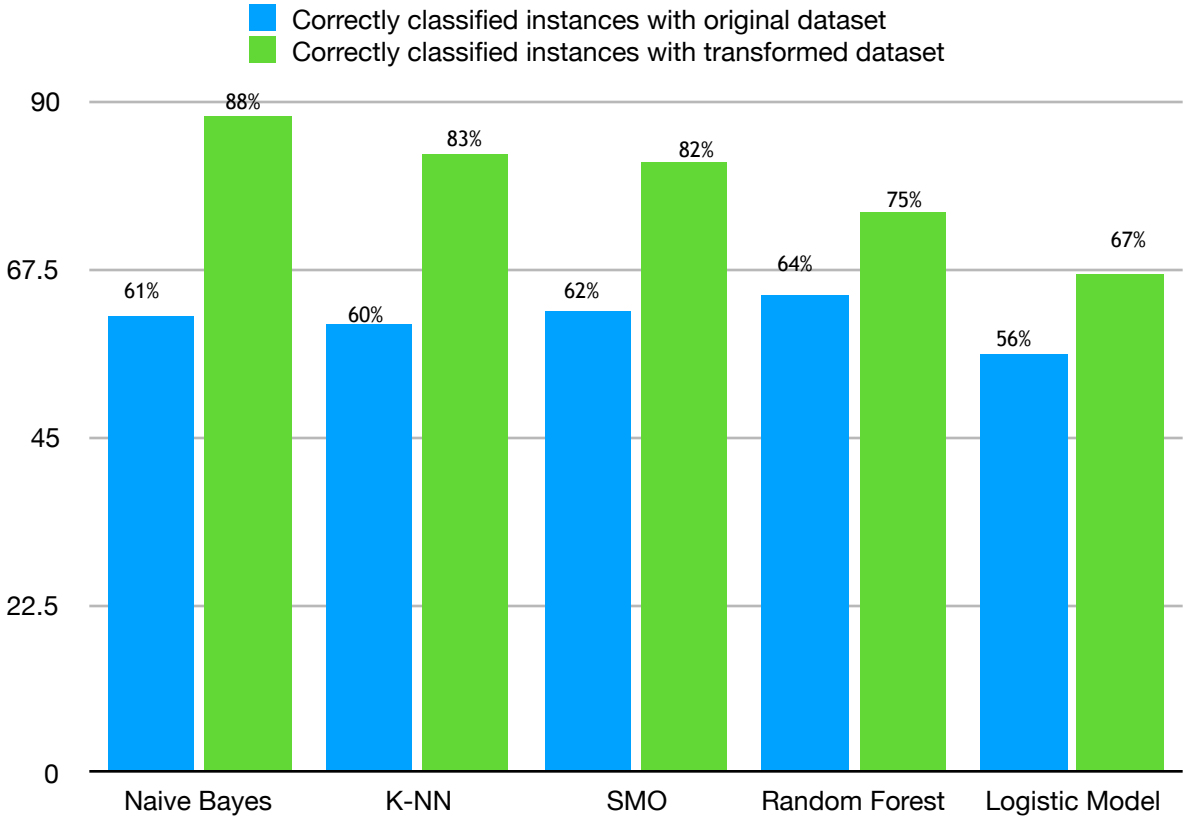

Figure 3.3.5: Bar chart for correctly classified instances by various classifiers

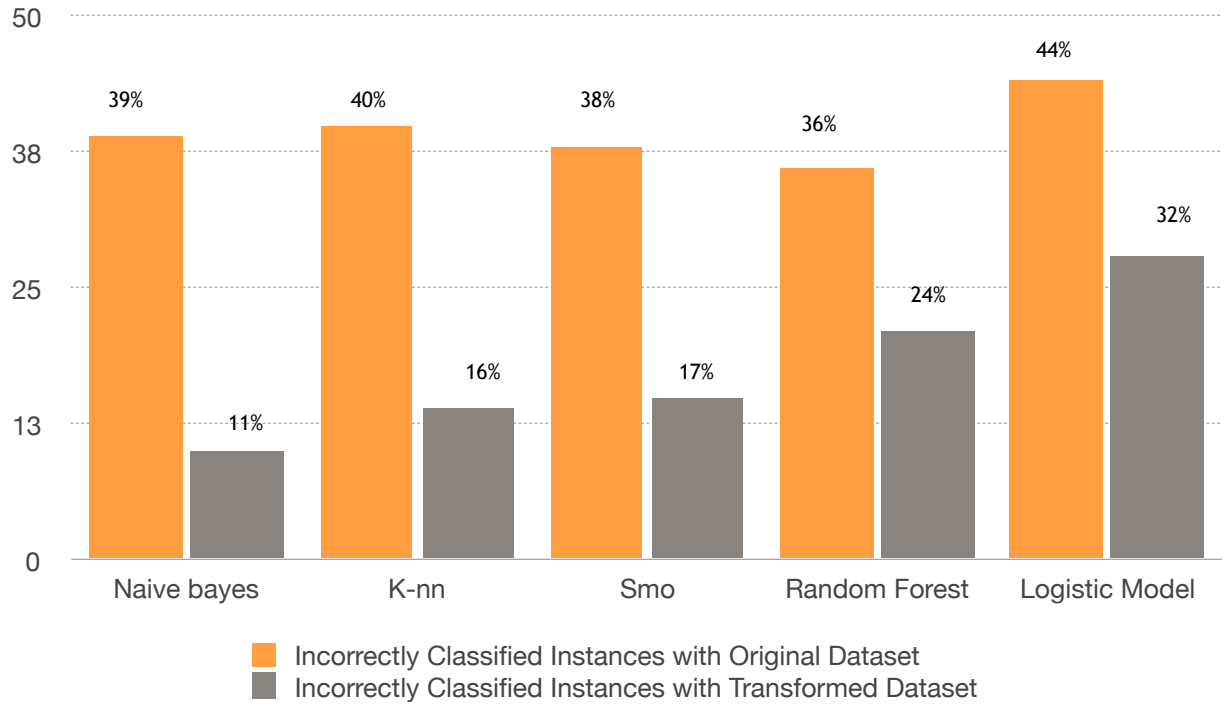

Figure 3.3.6: Bar chart for Incorrectly classified instances by various classifiers

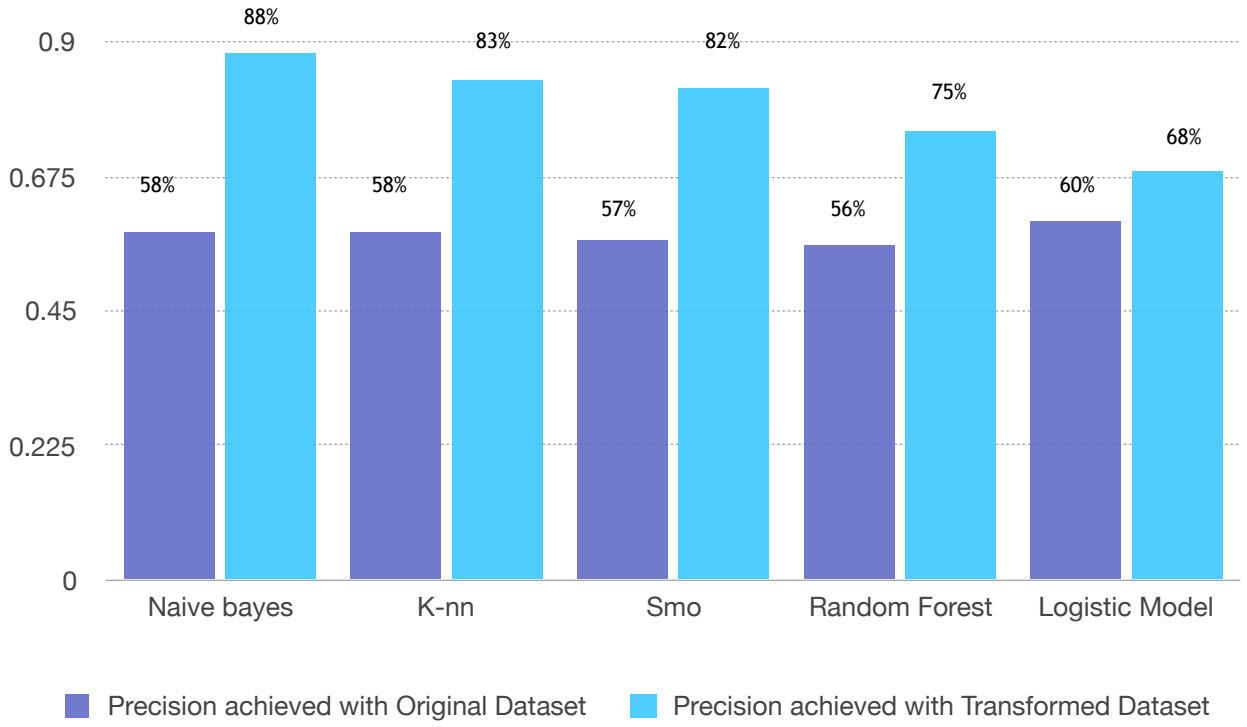

Figure 3.3.7: Bar chart for precision obtained by various classifiers on different datasets

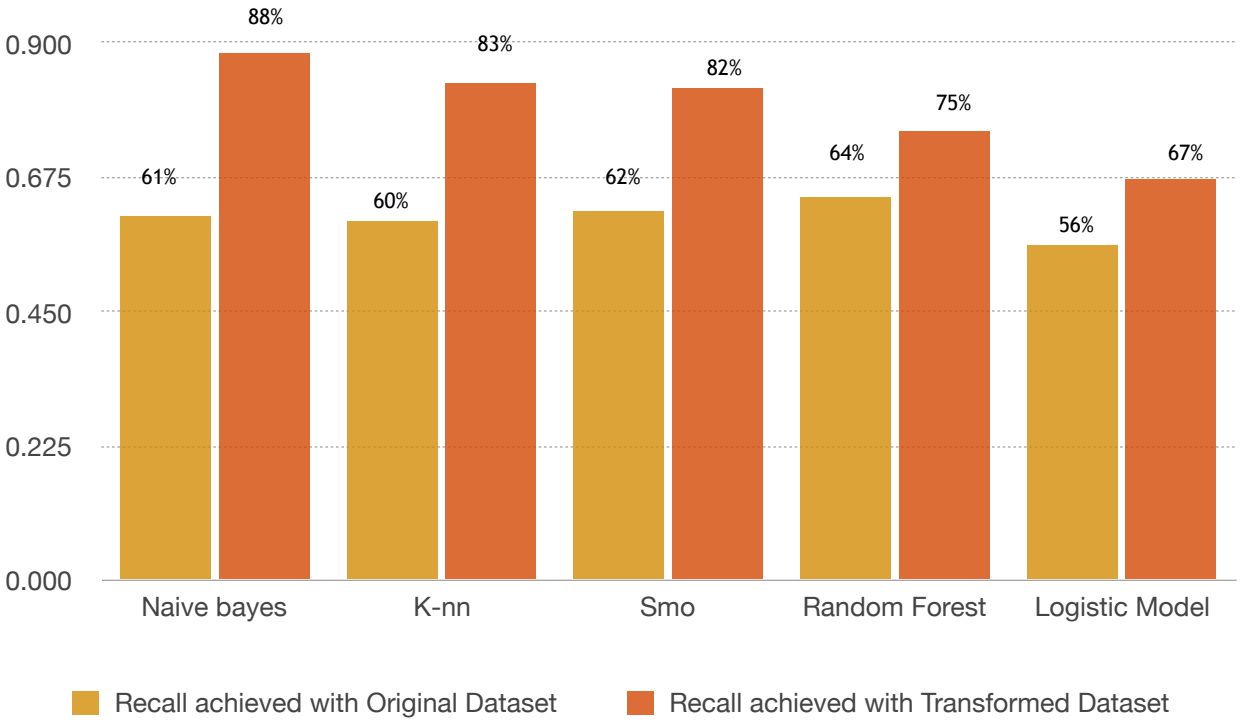

Figure 3.3.8: Bar chart for recall obtained by various classifiers on different datasets

Figure 3.3.9: Bar chart for F-Measure obtained by various classifiers on different datasets

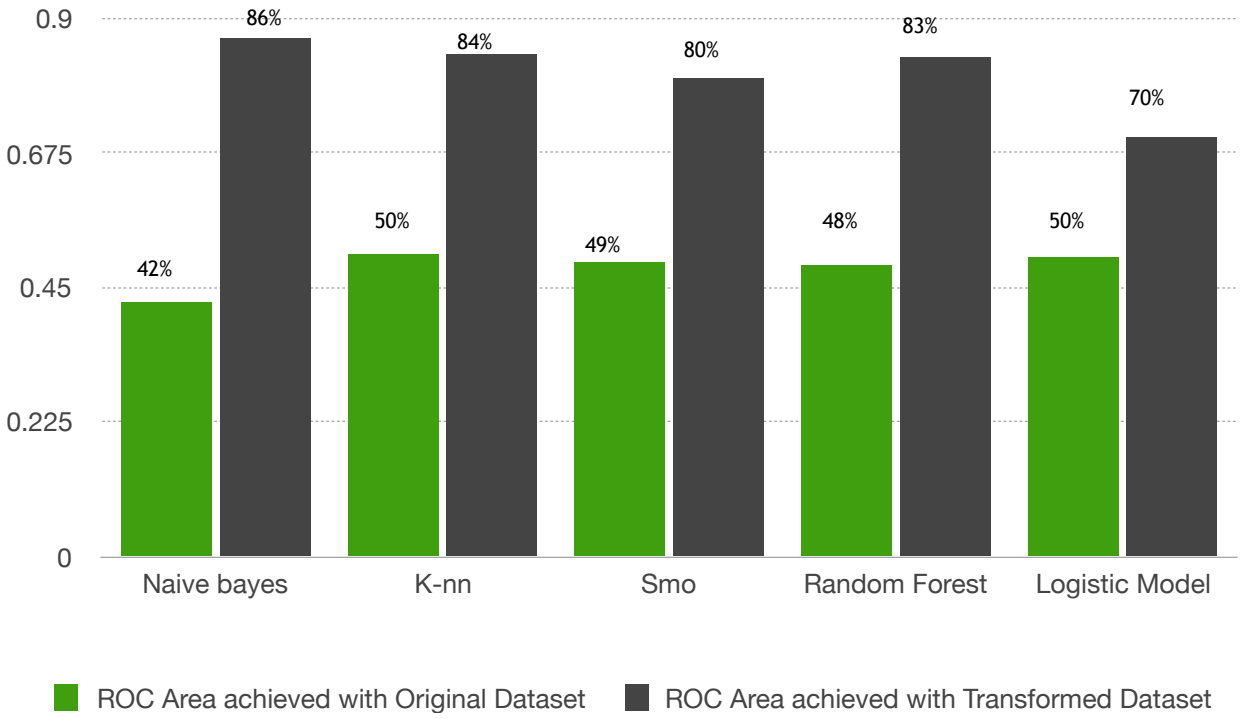

Figure 3.3.10: Bar chart for ROC Area obtained by various classifiers on different datasets

CHAPTER-5
CONCLUSION

CONTRIBUTION

This work focuses on the deep inside problems within the data that degrade the performance of the classifier. There were six major characteristics highlighted by this work, that are- problem of small disjuncts, noisy data, lack of density, overlapping between the classes, region of borderline examples and the problem of data shift. First these characteristics were properly studied one by one and individual algorithms were developed to tackle the problems that arise due to these data intrinsic characteristics. These issues limit the efficiency of the classifier hence resulting in disappointing outcomes.

The individual algorithms are combined together to give a collective result. As we can see in the result section the classifiers performed exceedingly well with the algorithmic transformed dataset. In case of transformed dataset their is an average increase of about 20% in the number of correctly classified instances, an average decrease of about 25% in the number of incorrectly classified instances. Using transformed dataset precision goes up by 20 % on an average and the recall also increases by around 20%, as a result the increase in f-measure is about 22 %. The increase in ROC area is around 30%. This goes on to prove that these data intrinsic characteristics could strongly impact the performance of the classifier. These is need to study more of these characteristics and their impact on the classification process.

Future work could lead us to new possibilities for achieving higher performance results by taking into account the hybrid classifiers and by identifying other holes in the classification process , so that the entire data could be well understood and analysed.

REFERENCES

[1] V. Lopez, A. Fernandez , S. Garcia, V. Palade and F. Herrera, ‘An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics’, Information Sciences, vol 250, pp. 113-141, 2013.

[2]R.Pruengkarn, K.W.Wong and C.Fung, ‘Imbalanced Data Classification using Complementary Fuzzy Support Vector Machine Techniques and SMOTE’, IEEE International Conference on Systems, Man and Cybernetics (SMC), vol. 1, pp. 5-8, 2017.

[3] S.Kotsiantis, D.Kanellopoulos and P.Pintel, ‘Handling imbalanced datasets: A review', GESTS International Transactions on Computer Science and Engineering, vol. 30, pp.34-40, 2006.

[4]A. Godase and V. Attar, ‘Classification of data streams with skewed distribution,’ Proceedings of the CUBE International Information Technology Conference on - CUBE 12, 2012

[5]V. Attar, P. Sinha and K. Wankhade, ‘A fast and light classifier for data streams,’ *Evolving Systems*, vol. 1, no. 3, pp. 199–207, 2010.

[6] C. Aggarwal, J. Han, J. Wang, and P. Yu, “A framework for on-demand classification of evolving data streams,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 5, pp. 577–589, 2006.

[7] A. Bifet, G. Holmes, R. Kirkby, B. Pfahringer, and M. Braun, ‘Moa: Massive online analysis’

[8] N.V. Chawla, K.W. Bowyer, L.O. Hall and W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, Journal of Artificial Intelligent Research 16 (2002) 321–357.

[9]S. Chen and H. He, "Sera: Selectively recursive approach towards nonstationary imbalanced stream data mining," in Neural Networks, 2009. IJCNN 2009. International Joint Conference on, June 2009, pp. 522-529.

[10]Y. Wang, Y. Zhang, and Y. Wang, "Mining data streams with skewed distribution by static classifier ensemble," in Opportunities and Challenges for Next-Generation Applied Intelligence, ser. Studies in Computational Intelligence, B.-C. Chien and T.-P' Hong, Eds. Springer Berlin I Heidelberg, 2009, vol. 214, pp. 65-71.

[11]Weiss, G. M., & Provost, F. (2003). Learning when training data are costly: The effect of class distribution on tree induction. *Journal of Artificial Intelligence Research*, *19*, 315-354.

[12]Chawla, N. V. (2009). Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook* (pp. 875-886). Springer US.