2019-1

Predicting Customer Retention of an App-Based Business Using Supervised Machine Learning

Jeswin Jose
Technological University Dublin

Recommended Citation

Jose, J. (2019) Predicting Customer Retention of an App-Based Business Using Supervised Machine Learning ,*Dissertation M.Sc. in Computing (Data Analytics), TU Dublin, 2019.*

Predicting Customer Retention of an App-Based Business Using Supervised Machine Learning

Jeswin Jose

D16129487

A dissertation submitted in partial fulfilment of the requirements of Dublin Institute of Technology for the degree of
M.Sc. in Computing (Data Analytics)

January 2019

DECLARATION

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Data Analytics), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the test of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Dublin Institute of Technology and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed: Jeswin Jose

Date: 04 January 2019

ABSTRACT

Identification of retainable customers is very essential for the functioning and growth of any business. An effective identification of retainable customers can help the business to identify the reasons of retention and plan their marketing strategies accordingly. This research is aimed at developing a machine learning model that can precisely predict the retainable customers from the total customer data of an e-learning business.

Building predictive models that can efficiently classify imbalanced data is a major challenge in data mining and machine learning. Most of the machine learning algorithms deliver a suboptimal performance when introduced to an imbalanced dataset. A variety of algorithm level (cost sensitive learning, one class learning, ensemble methods) and data level methods (sampling, feature selection) are widely used to address the class imbalance in the retention prediction problems.

This research employs a quantitative and inductive approach to build a supervised machine learning model that addresses the class imbalance problem and efficiently predict the customer retention. The retention Precision is used as the evaluation metrics for this research. The research evaluates the performance of different sampling methods (Random Under – Sampling, Random Over – Sampling, SMOTE) on different single and ensemble machine learning models. The results show that Random Under-Sampling used along with XGBoost classifier yields the best precision in identifying the retention class. The best model evolved in the research was also used to predict retainable customers from the recent unknown customer data, and could attain a retention precision of 57.5%.

Key words: *Retention Prediction, Churn Prediction, Machine Learning, Binary Classification, e-learning, Sampling*

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my supervisor **Prof. Sarah Jane Delany**, for her encouragement and tremendous support throughout the dissertation process.

I would like to thank **Mr. Rahul Ramesh**, CTO, Entri, for providing me the data for the research and would like to express my gratitude to **Hrishi, Sree Ram and Nidhin** (Product Team, Entri) for your outstanding support for my research project.

Finally, I would like to acknowledge the love and support that my friends and family showered during my studies. Big thanks to my dad **Adv. Jose.V.Chakiath**, my mother **Mini Jose**, my sister **Ann Maria Jose**, my local guardian **Joby John**, and all my friends. This would not have been possible without them.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

LIST OF ACRONYMS

CRISP-DM	Cross Industry Standard Process for Data Mining
CV	Cross Validation
FN	False Negative
FP	False Positive
TN	True Negative
TP	True Positive
TPR	True Positive Rate
FPR	False Positive Rate
SMOTE	Synthetic Minority Over-Sampling Technique
SVM	Support Vector Machine TN True Negative
ROS	Random Over Sampling
RUS	Random Under Sampling

1. INTRODUCTION

1.1 Background

The rapid growth of internet and its accessibility has paved way for huge advancements in the field of Education. With easy and cheap access to internet, education without physical classrooms has become possible and the business of internet-based education is expanding exponentially. As information is easily available in the internet, Education is now possible without the constraints of time, cost and space.

According to 2018 report by Forbes[1], the Online learning business worldwide will generate a revenue of $325 billion by 2025. Also, the factors like cheaper cost, compared to the university education, and customized content tailored for the end-users, makes e-learning more favourable to the aspiring students.

With huge advancements in mobile applications, mobile app-based learning, education is now easily possible for everyone without the barriers of time, age, qualifications or geography. App-based education, thus, represent a potential breakthrough in the field of learning. A report by IndiaToday[2] observed that educational apps business is edging towards a spectacular future.

The E- Learning platforms can be segmented into two types based on its characteristics like cost.

1) Free Knowledge platforms: Platforms where all the course contents are free and openly available for research or reference.

2) Paid Knowledge platforms: Learning platforms which charge the customers a fee to access the contents.

Also, based on the characteristics of the nature of education, they can be classified into two:

[1] https://www.forbes.com/sites/tjmccue/2018/07/31/e-learning-climbing-to-325-billion-by-2025-uf-canvas-absorb-schoology-moodle/#507296e43b39

[2] https://www.indiatoday.in/education-today/featurephilia/story/education-apps-1109306-2017-12-18

1) Self-taught: These are the education platforms where the customer must study themselves without an external tutor. All the course contents are available for reference for the customer and they can study and take up exams whenever they wish to.

2) Taught education: It represents a virtual classroom, with a tutor giving classes via video interface and the students can join the class online. The students can also communicate with the tutor online. These are usually costlier than the self-taught programmes.

With such advancements in the App-based learning industry, comes rapidly growing customer base for its service providers. In this competitive market, the customers are in search of better service, value for money, better content quality etc and therefore the cost of customer acquisitions have gone up. The cost of acquiring a new customer is said to be greater than retaining a current customer. So, the focus of the companies has changed from customer acquisition to customer retention.

1.2 The Company

Entri is one of the fastest growing e-learning start-ups in India with a customer base of more than two hundred thousand. It was founded in 2015 in Kerala, India with a vision of providing a platform for the people to practice and crack competitive exams both in private and public sector. Apart from the other e-learning platforms, this Android App, Entri strives to make the learning more fun and less stressful. The customer base of Entri is spanned over Asia and the Middle East. With a current install rate of 2000 installs/day, the company aims to achieve the target of 2 million customers by the end of 2019. In Last 2018, the customer base of Entri grew by 400% compared to 2017. Entri is a startup pitch winner at the prestigious Martin Trust Centre for MIT Entrepreneurship. It is also a part of Facebook's F-Start start-up program for the most promising mobile apps across the world.

1.3 Research Problem

The business in question, Entri has over hundred thousand customers and is one of the trending companies in the app-based education industry of India. This application has hundreds of mock exams which helps the students to prepare for the competitive examinations. Every new user of the app gets a free trial period of 7 days with unlimited access to all the contents (courses, quizzes and mock exams), after which they must upgrade to a premium membership to continue accessing the contents.

Currently, the customer retention proportion is 1.25%. i.e., 1.25% of the total customer base have upgraded their membership to premium. The company uses an assumption-based algorithm (called the LEAD algorithm) with no learning mechanism to identify customers with possibility of conversion and the sales team of the company approach them with promotional offers and discounts to make them convert to the premium plan.

The current LEAD algorithm has a precision of 20%. i.e., only 20% of the customers contacted by the sales team, are retained. A major investment of time and revenue is made on the sales team to contact the potential customers and convert them. The problem this research proposes to tackle is to decrease the effort of the sales team by exploring machine learning techniques to identify customers who have a better probability of conversion than the ones predicted by the current algorithm. Such data driven marketing can not only optimize the performance of the sales team but also help the business to have a better customer retention.

The aim of the research is to identify the factors that affects the retention of a customer and develop machine learning models which can predict the customer retention better than the LEAD algorithm which is currently being used by the business.

Due to the huge imbalance of data, used in the research, the class precision is used to evaluate the performance of the models developed in the research. Currently, the retention precision of the conventional LEAD algorithm used by the business is 20%. The objective of the research is to evaluate the performance of the supervised machine learning models compared to the retention precision of the conventional algorithm.

To guide the research, the research question has been formalized as:

“Can supervised machine learning models perform better than the conventional assumption-based LEAD algorithm in terms of Retention Precision for predicting the customer retention?”

1.4 Research Objectives

The key objective of the research is to experiment whether the application of supervised machine learning on the customer data can help to predict the retention of the customers precisely than the predictions generated based on the general assumption on the customer behaviour. The current approach (called LEAD algorithm), is based on certain assumptions on the customer data and it assigns a weighted score to each user based on five attributes of the user's data. It is assumed that, higher the lead score, the higher is the probability of the customer to retain. Currently, using this algorithm, the retention precision attained is only around 20%. Applying machine learning models to customer data can not only help predicting the retention but also can help the business to get insights on the reasons of customer churn and retention.

The objectives of this research are:

1) To review relevant literature on retention prediction, supervised machine learning models, class imbalance, data sampling and customer retention behaviour.

2) To collect required customer data from the business in question for the research.

3) Identify and rectify any errors, or quality issues of the data, that can affect performance of the machine learning models.

4) Prepare the data using sampling, encoding and feature extraction.

5) Build the machine learning models using supervised learning algorithms such as Random Forest, Support Vector Machine, Logistic Regression, and XGBoost.

6) Evaluate the performance of the models using retention precision as the evaluation metrics.

7) Evaluate the performance of the built model with varying class imbalance ratio by implementing sampling methods. Identify the best model that can precisely predict the customer retention.

8) Obtain recent customer data from the business. Predict the customers with most probability of retention from the recent customer data. Evaluate the precision of the model by measuring the precision of the conversions after these customers are contacted by the sales team.

9) Identify the limitations of the research and propose areas of future research.

1.5 Research Methodologies

A Quantitative research methodology is used in this research because it involves conducting experiments on the customer data to build machine learning models that can predict the customer retention. The hypothesis is accepted or rejected based on the retention precision of the machine learning model derived from the research. The result of the research is based on the experiments and a comparison is made between the retention precision of the machine learning models and the LEAD algorithm, making the reasoning of the research as Inductive.

The initial data wrangling and cleaning of the data is done using Microsoft Excel. Python programming is used for the statistical exploration of the data, data preparation, building of the machine learning models and the evaluation.

Cross Industry Standard Process for Data Mining (CRISP - DM) methodology is used in this research. It provides a structured method to execute a data mining project. It consists of six phases, which are business understanding, data understanding, data preparation, modelling, evaluation and deployment.

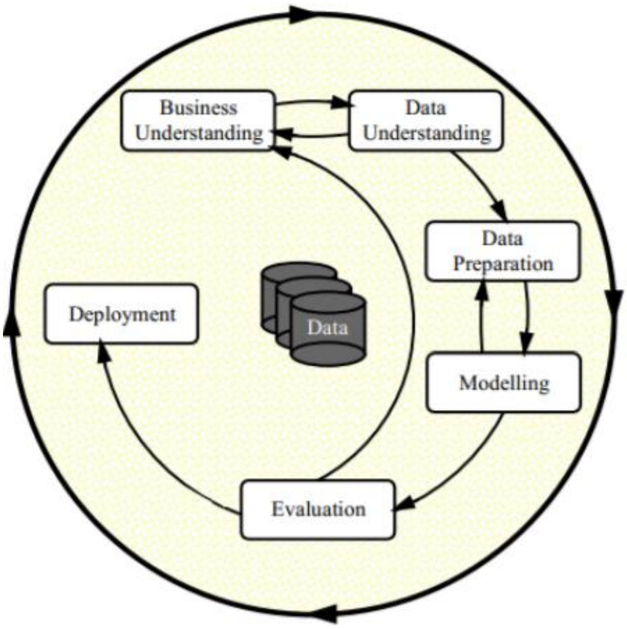


Figure 1. 1 : CRISP DM Life-Cycle

(Source: Wirth & Hipp, 2000)

1.6 Scope and Limitations

Modelling customer behaviour and application of machine learning techniques can help in predicting the customer retention and help the business to identify the reasons behind customer loss (Sharma & Panigrahi, 2011). Also, precise identification of loyal customers can help to improve the customer retention by targeting promotions and offers for them. In fact, data driven marketing using predictive analysis of customer data, can help the business thrive compared to the conventional methods of marketing.

The scope of this study is to develop a machine learning model using the customer data to predict the customer retention of an app-based e-learning platform in India. Since the current customer retention rate is only 1.25% for the business in question, the customer data is hugely imbalanced. Due to this, the classifiers often fail to correctly identify the minority data since it doesn't have enough data to learn. The major scope of the research is to tackle the class imbalance problem using sampling methods such as Random Under-sampling, Random Over sampling, and SMOTE sampling algorithm and build an efficient machine learning model that predicts the customer retention precisely.

The major limitation of the research is the un-availability of potentially important features of the customer data. The business in question, is not a data-driven organization, and doesn't record/maintain all the important customer data like customer profile details, customer geography etc . This confines the research to use only 10 features for building machine learning models. Also, huge imbalance in the customer data due to low customer retention is also another limitation to overcome.

1.7 Thesis Outline

The outline of the Thesis report document is as given below.

Chapter 2 (Literature Review) discuss the literature related to the Churn Prediction methods, Class Imbalance problem, Sampling methods, Predictive modelling especially Binary Classification algorithms.

Chapter 3 (Design and Methodology) discuss the design of the research in deep. Each phase of CRISP- DM methodology followed in the research is discussed in detail here. The process of obtaining the data, cleaning the data, transforming the data, training the machine learning models using the data, evaluation of the models, implementation of the best model are discussed in this chapter.

Chapter 4 (Implementation and Results) presents the results of the implementation of the proposed design and the results of the experiments in terms of class precision and recall, and a comparison of the results of each of the experiments.

Chapter 5 (Evaluation and Discussions) outlines the evaluation of the results of the experiments and discusses the results in light of the research question.

Chapter 6 (Conclusion) summarises the research carried out. It discusses the contribution of the research towards the research question. The chapter concludes with discussing areas of future research.

2. LITERATURE REVIEW

This chapter provides a review on the literature available on App-based education platforms, Churn/Retention prediction methods, Imbalanced data, effects of imbalanced data on training the machine learning model, sampling methods and their effect on classifiers and the evaluation metrics used for evaluating the models. The chapter concludes with the gaps in the research which forms the objective for the research.

2.1 Class Imbalance

A dataset is said to have class imbalance problem when one class of data has a significantly greater number of instances compared to the other class/classes. Class imbalance problem one of the major problems faced when dealing with real-world data. It is quite common in real world problems like Medical diagnosis of rare diseases, fraud detection in banking operations etc (Longadge, Dongre, & Malik, 2013)

Class Imbalance problem can affect the learning of a machine learning classifier, making it biased towards the major class in the dataset. Due to this, the minor classes are often ignored, and it degrades the classification performance of the classifiers. Class imbalance ratio is a metrics used to quantify the imbalance in a dataset. It is the ratio of the majority class to the minority class. The major challenge of the machine learning models dealing with the imbalanced data is to successfully identify the signal in the minority class and predict the instances of minority class with good precision.

2.2 Effect of Class Imbalance

Most of the machine learning classifiers such as decision trees or neural networks work on the assumption that the training data has equal representations of classes. But the real-world data often has huge imbalances with very low representations from the minority class. The class imbalance ratios can be as low like 99:1. This can hinder the performance of the machine learning classifiers and can cause erroneous predictions of the minority class.

The minority data is most often considered as noise by the classifiers and most classifiers have a preventing mechanism that ignores noise to prevent overfitting. Due to this, the meaningful data, i.e., the minority samples are often ignored by the classifiers during the model training. Most classifiers prefer more common classes in the presence of uncertainty (Kotsiantis, Kanellopoulos, & Pintelas, 2006). Thus, it affects the generalization ability of a model on the unknown data. (Buda, Make, & Mazurowski, 2018)

Another challenge class imbalance cause is the lack of data in the minority samples. Due to this, the classifiers fail to discover regularities or patterns within the minority data, which causes poor learning.

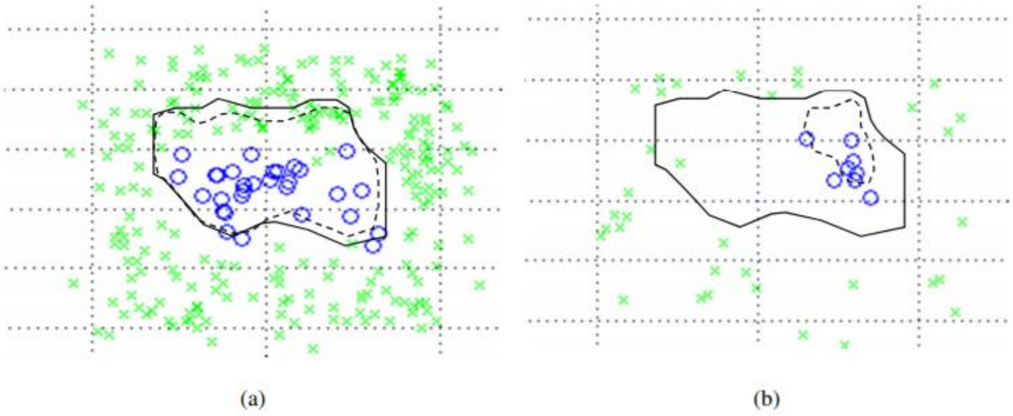


Figure 2. 1: Impact of Small Sample Size In Class Imbalance Problem

As shown in the Figure 2.1 ((a) the solid line determines the true decision boundary and (b) the dashed line defines the estimated decision boundary), when the sample has enough data (Figure 2.1 (a)), the decision boundary in the feature space is correctly identified by the classifier. Whereas, in the second diagram, when the sample doesn't contain adequate representative samples from the minority class (Figure 2.1 (b)), the decision boundary is not identified correctly by the classifier due to insufficient information.

Class overlapping, or class complexity is another problem that is caused due to class imbalance. In highly imbalanced datasets, sometimes the minority data samples get overlapped with majority data in the feature space and makes it difficult for the classifier to determine the decision boundaries for the minority class. In those cases, such data is often treated as redundant or duplicates and are avoided in the learning process (Kotsiantis et al., 2006).

2.3 Approaches in Tacking Class Imbalance Problem

Imbalance in the data hinders the classifier performance and affects the generalization of the model on the unknown data. Two main approaches in tackling class imbalance problem are:

1) Data based 2) Algorithm Based

The data level approach tries to mitigate the imbalance problem by balancing the samples of minority and majority data in the training data before building the machine learning model. Ex: Sampling, Feature Selection.

In the algorithm-based approach, dedicated machine learning algorithms are used to train the model that specifically learn the information from the minority class in the imbalanced distribution. Ex: Cost sensitive learning, Ensemble models.

Both approaches are discussed in detail in the below sections.

2.3.1 Data Level

Data level approaches usually employs a data processing task that balances the number of samples from each of its classes in the data. The main data level approach used to tackle class imbalance problem is Sampling. Sampling methods are used to generate a new representative data set from the original dataset with a more balanced distribution of the classes. The dataset obtained by sampling should consist of only the instances that are reasonably similar to the ones in the original dataset.

In the under-sampling approach, the discrepancy in the number of samples for each class is eliminated by removing samples from the majority class. Whereas, in the over sampling approach, duplicates of the minority samples are generated to match the number of majority samples in the population.

a) Random Over Sampling

Random over sampling increases the number of minority samples in the imbalanced dataset by replicating instances of minority data. Random Over Sampling is said to cause over fitting of the machine learning classifiers as the same data is replicated multiple times, hindering the generalization ability of the classifier. (Zheng, 2015)

For example, if a dataset has 100 instances of majority class and 5 instances of the minority class, to attain a balance in the data, the 5 minority instances have to be replicated 20 times each to have a 50:50 ratio in the dataset. In the real world examples like fraud analysis, detection of a rare disease etc, the imbalance ratio would be even higher, causing huge replication in the random over sampling approach. This causes multiple duplicates in the dataset, causing over fitting of the training dataset affecting the performance of the classifier to unknown data. Researchers like J. Burez and Poel (2008), have proved that under-sampling of an imbalanced dataset can lead to better prediction accuracy compared to the over-sampling methods.

b) Random Under Sampling

In random under sampling, data of the majority class are discarded randomly until the preferred balance is obtained. For example, consider a dataset containing 20 instances from the minority class and 100 instances from the majority class. In random under sampling approach, to obtain a 50:50 balance ratio, 20 instances from the 100-majority class are randomly selected and the other 80 are discarded.

The major concern with the random under sampling approach is the loss of potentially useful information. Considering the above example, the information contained in the 80 discarded instances are lost, which makes the classifier difficult to learn the decision boundaries between the minority and majority data, affecting the classifier performance (Hoens and Chawla, 2013)

c) Synthetic Minority Over-Sampling Technique (SMOTE) Technique

SMOTE is an over sampling method which generates synthetic instances of the minority class which is not an exact replica of the feature vector (minority sample). This algorithm which was proposed by (Hoens et al., 2013) is a better version of the random over-sampling method as they don’t just create duplicates of the minority samples like random over-sampling. Instead, they generate synthetic samples of the minority class by performing certain operations on the data. Here, the minority class is over-sampled by taking each minority samples and generating synthetic samples along the line joining the k minority class nearest neighbours (Hoens et al., 2013). Depending on the ratio of the sampling, the number of neighbours is chosen. For example, if the over-sampling ratio is 100%, for each sample in the feature space, one neighbour is chosen and one sample is generated along the segment joining the sample and this neighbour.

The synthetic sample is generated in the following way.

1) Take the difference between the minority sample and its nearest neighbour.

2) Multiply this difference with a random number between 0 and 1.

3) Add this to the minority sample in consideration.

4) Take the resulting feature vector as the new sample.

The new sample generated could be mathematically represented as:

$$Xnew = X + (X - X') * rand\ (0,1)$$

where, X’ = k nearest neighbor, X = sample (Hoens et al., 2013).

Due to the randomness of the multiplier used, the new sample is generated at a random point in the segment joining the minority sample and its neighbour. So, no duplicate samples are generated unlike the random over-sampling method. Thus, it avoids the problem of overfitting of the machine learning models.

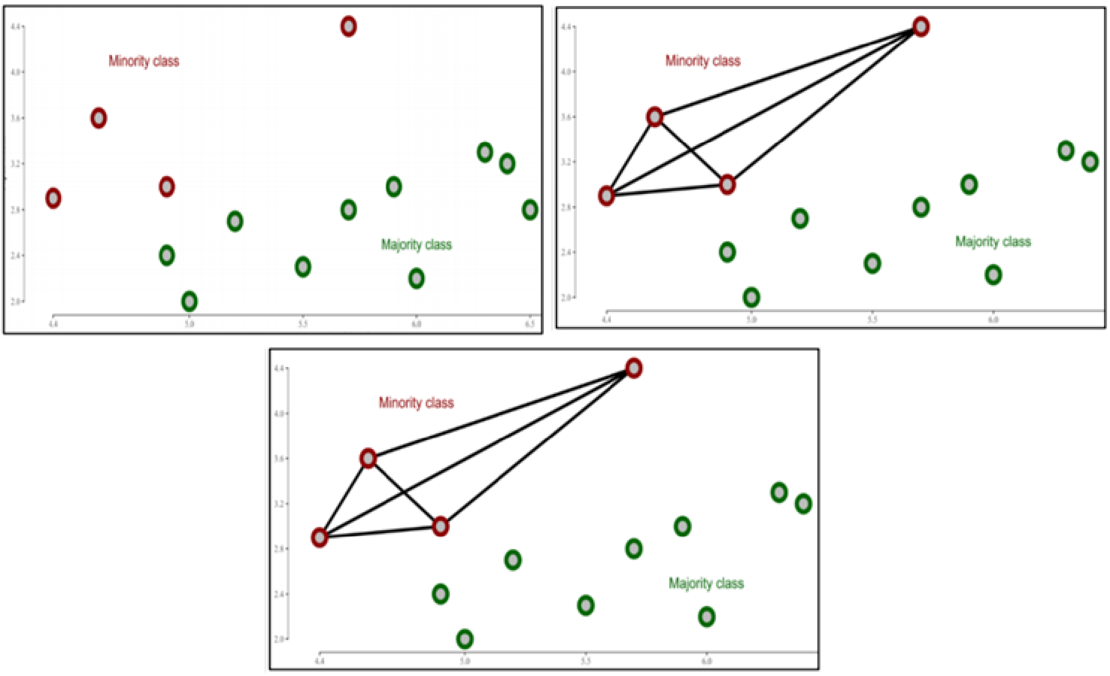


Figure 2. 2: SMOTE methodology of synthetic data generation

2.3.2 Algorithm Level

Algorithm level approaches handles the data imbalance by modifying the existing learning algorithms to fit the imbalance of the data. These dedicated algorithms are used to learn the imbalance of the data from the training data and train the model accordingly. The different examples of algorithm level approach are Ensemble methods, Cost-sensitive learning, one – Class learning etc.

a) One class learning

One class learning method, also called as Recognition based learning method, is a method proposed by Japkowicz, Myers, & Gluck (1995) where the classifier is modelled only on the minority class rather than modelling the data on both target and non-target classes. In one class learning, only the target class (minority class) is presented to the system, and the model is trained to identify only the target class, eliminating the non-target class. The experiments done by (Japkowicz et al., 1995) shows that one class learning performs better than the conventional two-class learning method. Here, the patterns of the minority target instances are learnt by the model, and the model is then used to identify targeted instances from the unseen data.

b) Cost Sensitive Learning

Misclassification errors in applications like medical diagnosis, fraud analysis etc are associated with high misclassification cost. Also, in these applications, the data will be highly imbalanced. The classical learning algorithms like Linear Regression, SVM etc assume same misclassification cost for both minority and majority classes, and the difference between the misclassification costs are ignored. Elkan (2001) proposed a method to solve this problem using cost sensitive learning. They proposed a learning algorithm that considers the misclassification cost of the classifier during the training to produce a model that has lowest cost. For a binary classification problem, a cost-sensitive learner assigns greater cost to the false negatives compared to the false positives.

The application of cost sensitive learning in real world datasets is limited because the misclassification cost should be known before constructing the model. As the cost information is dependent on many other factors which are not easily available, it is not a feasible approach for prediction regarding most of the real world applications. Also, according to Maloof et al (2003), cost – sensitive learning approach leads to over-fitting of the machine learning model.

c) Ensemble Methods

Ensemble approach is based on building multiple classifiers on the same training data and aggregating the evaluations of each model to form the final decision. The main idea behind ensemble methods is forming a strong learner by combining many weak learners.

Ensemble methods are divided in to two categories: Bagging and Boosting.

Bagging, also known as boot-strap aggregation, is a method where several subsets of the training data is created and each of these subsets are used to train it's classifier. Finally, all the decisions of each of the classifiers are aggregated to form the final decision. Random Forest algorithm is an extension of bagging method, where a collection of decision trees is used, with a random selection of the features, to give the final decision of the model.

Boosting, is an ensemble learning method with multiple classifiers trained sequentially. The goal of each classifier is to minimize the error made by the previous model in the sequence. Each misclassified instance is weighted more in the next model, so that the next model is more likely to classify it correctly. At the end of the sequence, all the weak learners are thus converted to a strong model.

2.4 Machine Learning

According to ExpertSystem[3] 'Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed'. It is used mainly for applications like classification, recognition, forecasting and prediction.

Machine learning works by 'learning' data and generating the prediction rules by recognizing the patterns in the data rather than working according to a well-defined and hard coded algorithm. The iterative nature of the machine learning enables it to adapt and evolve according to the new changes in the data.

[3] https://www.expertsystem.com/machine-learning-definition/

Machine Learning is sub divided mainly into two classes based on the mode of learning:

1) Supervised Machine Learning 2) Un-supervised machine learning

2.4.1 Supervised Machine Learning

Supervised Machine Learning is the type of machine learning where, the input variables and the respective output variables are given to the system, and it learns the mapping function from the input to the output. The input variables labelled with the respective output variable is called as the training set. In the training phase, a supervised learning algorithm is used to analyse the training examples and an inferred function is produced that maps the input variables (X) to the respective output variables (Y). This could be represented as:

$$Y = f(X)$$

Using this mapping function, the system can then predict the output for the new input variables with unknown output. This mode of learning can be compared to a teacher-assisted learning process, where the correct answers are known beforehand, and the teacher corrects the wrong answers. Due to the iterative nature of the machine learning process, the learning algorithm, corrects itself by comparing the intended output and the predicted output.

If the output of the system is discrete, it’s called a classifier, and if the output is continuous, it is called a regression function. Some examples of supervised machine learning algorithms are Logistic Regression for classification, Random Forest for regression and classification, Support Vector Machine etc.

a) Logistic Regression

Logistic regression is a regression model used to predict the output when the target variable is binary. The goal of this model is to find the mathematical model which best describes the relation between the independent variables and the target variable. The logistic function is given by,

$$\mathrm{h}\theta(\mathrm{x}) = \frac{1}{1 + e^{-z}}$$

where z is the logit function.

b) Support Vector Machine (SVM)

Support vector machine (SVM) algorithm is based on finding a hyperplane in the N – dimensional space, that divides the data into different classes. Many hyperplanes could be chosen that distinctly divides the classes of data, but the best SVM model will have the hyperplane with maximum distance from both the classes as shown below.

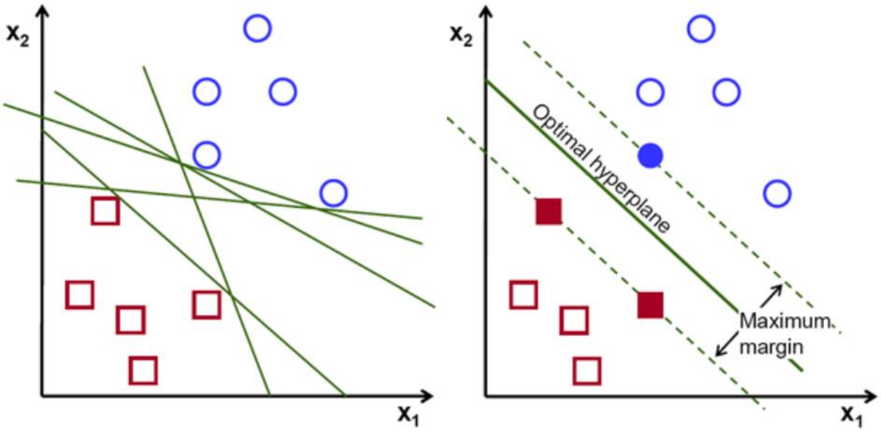


Figure 2. 3: Possible hyperplanes in SVM and optimal hyperplane

(Source :Rohith, 2018)

c) Random Forest

Random forest is a supervised machine learning algorithm used for both regression and classification problems. It is an ensemble of Decision trees trained using the ‘bagging’ method. A random forest model builds multiple weak decision trees and merges them to form a stable and efficient prediction model. A major advantage of random forest is that, it gives good prediction result with default hyper parameters. Also, compared to the other models, random forest is observed to have less over fitting problem.

d) eXtreme Gradient Boosting (XGBoost)

XGBoost or eXtreme Gradient Boosting is a machine learning algorithm developed by Chen & Guestrin (2014) which works on the principle of gradient boosting decision trees. XGBoost is popular among the machine learning applications for two reasons : 1) Execution Time 2) Model Accuracy. XGBoost uses the gradient boosting decision tree algorithm where decision trees are sequentially added in the model, and each tree corrects the errors of the previous tree in the sequence. Thus, together it produces a strong model from a collection of weak models.

2.4.2 Unsupervised Machine Learning

Unlike the supervised machine learning, the input variables don't have a corresponding output variable in the learning process of Un-supervised machine learning. In this mode of learning, the system aims at modelling the underlying patterns and hidden structures of the data. Unlike the supervised learning, there is no labelled outputs or a teacher to supervise the learning procedure.

Two main methods of un-supervised learning are :

1) Clustering: In the clustering applications of un-supervised learning, the model discovers the underlying groups or clusters of data with similar behaviour. For example, for a retail industry, it groups the customers based on their shopping behaviour.

The examples of clustering algorithms are k – means clustering, hierarchical clustering etc.

2) Association: Association is a method of unsupervised machine learning to discover interesting relationships between the data. It's a form of rule-based learning, where the system deducts hidden rules in the variables. One of the most widely used application of Association learning is the market-basket-analysis. It is used to discover the buying patterns of the customers from the sales data. For example, the rule {diapers} -> {beer} found in the sales data of Walmart indicates that if the customer buys Beer, they are more likely to buy diapers as well (Domingos, 2012)

2.5 Customer Retention Prediction

Customers are said to be an asset to the firms (Chang, 2012). A retaining customer base is often considered as a metrics to measure the growth of a company. According to Dawes (2009), retention refers to the number of customers who stays with the firm in the course of a given period. Customer retention predictions are often used by the business to implement loyalty programmes for the frequent customers and generate a long-term relationship with the customer. Acquiring a new customer is said to be six times is more expensive than retaining an existing customer. Identifying the customers with probability of retention is a key element for growth for the internet-based business when considering the various switching options available for the customers. Such loss of loyal customers can hinder the growth of the company in terms of customer base and revenue.

Customer retention prediction helps the business to identify the customers who has a probability of a long-term retainment from the total customer base. Customer Relationship Management (CRM) systems are currently used by the businesses to capture the customer activity data, personal data etc. These are often used for the prediction of the customer retention. (Magatef and Tomalieh, 2015).

Based on the Retention predictions, companies often execute retention programmes to understand the customer needs, communicate with the customer and plan the future interactions. Inexpensive retention programmes like a phone-call or an e-mail communication are common platforms for the businesses. Thus, high-valued customers with more probability of retainment can be identified and the business can target marketing or retention activities with them in focus.

2.6 Retention Prediction using Machine Learning

Retention/Churn prediction in business is of prime importance today because of the increasing customer base and increase in costs associated in acquiring new customers. Much research being done in this field for identifying the potentially retainable customers or customers with a possibility of churning. Most of the research associated with Retention Prediction are based on applying machine learning models in customer activity data or customer relationship management (CRM) data and predicting the behaviour of an unknown customer. The different approaches used in predicting retention is discussed in the next section.

2.7 Approaches in Predicting Retention

Many researches have been done on prediction of retention and churn in the recent times. Most of the research is based on applying machine learning algorithms on the customer data to predict the possible retainable customers. The similar approaches in retention prediction are discussed in this section.

A research done by Sahar (2018) analysed the performance of ten machine learning techniques such as ensemble methods, Regression methods, SVM, Decision Trees, Instance Based learning and Discriminant analysis on a telecommunication dataset to predict customer retention. He observed that the ensemble methods of machine learning outperformed the other methods with an overall accuracy of 96% and logistic regression had the least performance with an accuracy of 86.7%. Along with the classification algorithms, clustering customers according to their behavioural patterns was done in some research for better predictive accuracy. In a research done by Amjad et al, (2015), a study on the performance of hybrid models is done in comparison with the single models. The hybrid models are formed by two phases. In the first phase, the similar customers are clustered using three clustering algorithms (k-means algorithm, self-organizing maps, and hierarchical clustering). In the second phase, the data is modelled using MLP – ANN. It was observed that the hybrid models outperformed the single models in the model accuracy.

The performance of Sub Vector Machine in predicting churn, is analysed by a research done by Xia et al (2018). They compared the performance of SVM with other machine learning models like decision tree C4.5, naïve Bayesian classifiers and logistic regression on a telecommunication dataset. The found that SVM has better ability to generalize to unknown data, and good precision compared to the other models when the churn rate is big, less missing data and when the data is non-linear.

Very less studies are done in addressing the class imbalance problem of retention/churn predictions. Handling imbalance in the dataset is a major problem to tackle. A research by Burez, J and Van, P (2009) performed a study on handling the class imbalance in customer churn prediction. They compared the performance of under-sampling, boosting and cost-sensitive learners on six real-time customer churn datasets using AUC and Lift as evaluation metrics. They also studied the performance of advanced sampling techniques like CUBE and SMOTE. They concluded that under-sampling with weighted random forest as a cost-effective learner can lead to improved prediction accuracy. Chujai et al (2017) proposed a solution for resolving class imbalance by separating the data into overlapped and non-overlapped regions between the classes and clustering them based on the Euclidean distance and then generating separate classification model for both of the regions.

An ensemble-based wrapper method is proposed by Yang et al (2010) to classify imbalanced data. They created multiple balanced datasets from the original dataset with heavy class imbalance and an ensemble of base classifiers were used to model each of the subsets of the data. They proved that this method could outperform the performance of classifiers based on single inductive algorithms. Xiong et al (2010), also supports this method by comparing the performance of the sampling methods and methods based on separating the overlapping regions.

A study by Chao et al. (2008) compared the performance of machine learning algorithms such as weighted random forest, balanced random forest and sampling algorithms like SMOTE and SHRINK based on precision, recall and F-measure on oil spill data and mammography data. They discovered that random forest algorithms have superior performance compared to the other algorithms in the research. They observed that weighted random forest and balanced random forest models have better performance in terms of F1-Score and G-Mean.

Sampling methods of the data and change in performance with data sampling is studied by J. Burez and Poel (2008). Using AUC and Life as the evaluation metrics, a comparison of performance of the machine learning models with random under-sampling and advanced under – sampling are done. They concluded that under-sampling boosts the prediction accuracy in terms of AUC and Lift, but a 50:50 balanced sample is not required.

Most of the state-of-the-art approaches in churn prediction uses predictive modelling to construct models using machine learning algorithms such as Random Forest (Chao et al. (2012), Bart et al. (2015)), Logistic Regression (Burez et al. (2009), Khan et al. (2010)), Support Vector Machine (Xia et al. (2008), Jin et al. (2010)) and a few using deep learning algorithms like ANN (Amjad et al. (2015)).

2.8 Summary, Limitations and Gaps in Literature Survey

A detailed review of the state-of-the-art approaches to predicting retention has been studied for this research. Most of the research reviewed on retention prediction addresses the class imbalance problem. Sampling methods are the most used tactics in the similar research to overcome the class imbalance problem compared to the other approaches like feature selection, class separation etc.

Most of the research (Xia et al. (2008), Burez, J and Van, P (2009), Amjad et al. (2015) etc) are based on customer data of telecommunication industry and a few researchers like Seungwook et al. (2017) and Chao et al. (2008) have done churn prediction in other fields like game and app industry. No much research has been focused on the retention prediction on e-learning industry.

The research into customer related predictions are mostly biased towards churn prediction. Most researches like Xia et al. (2008), Amjad et al. (2015) are focused on predicting the churn. Only a few like Jin Su et al. (2010) have researched in the prediction of the customer retention. Also, the class imbalance problem has not been addressed by most of the researchers except for few like Chao et al. (2008), Burez and Van (2009), Xiong, Wu, & Liu (2010)

Also, most of the research are done for the purpose of research and uses archived data. It doesn’t provide much guidance on the analysis of a live and real-world application. To address the limitations and research gaps presented in this section, the research question is given as:

“*Can supervised machine learning models perform better than the conventional assumption- based LEAD algorithm in terms of Retention Precision for predicting the customer retention?*”

The next sections will discuss in detail, the research design, implementation and evaluation of experiments to address the research question.

3. DESIGN AND METHODOLOGY

This chapter presents the detailed over view of design and methodology used in the research to answer the research question. The Cross Industry Standard Process for Data Mining (CRISP - DM) methodology is followed in the research lifecycle. Python programming with Jupyter Notebooks interface is used to carry out the experiments of the research.

The aim of this research is to build a machine learning model, that can predict the customer retention with better precision than the conventional LEAD algorithm, which is currently being used in the business to predict the potentially retainable customers. Currently, the LEAD algorithm has a precision of 20% in predicting retainable customers. This research aims to build a supervised machine learning model with a retention precision greater than 20%. The overall workflow of the experiments is as shown below.

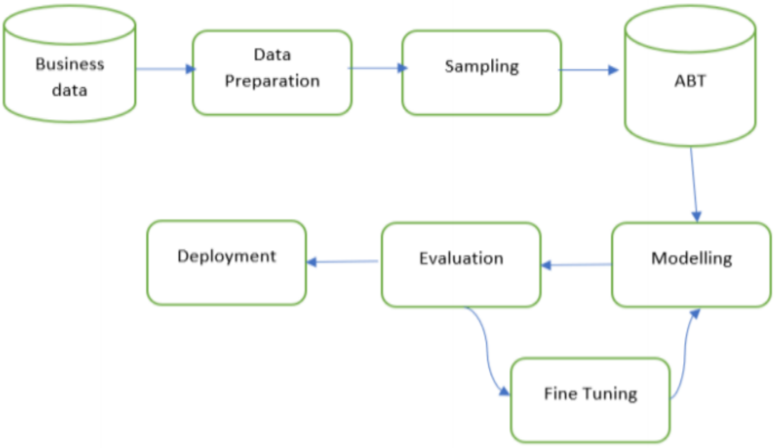


Figure 3. 1: Experiment Design

The thesis follows CRISP-DM methodology, and each of the phases of the CRISP DM methodology are described in detail below.

3.1 Business Understanding

In the business understanding phase, the problem which the research must address is studied. The business in question is an app-based e-learning platform which has over two hundred thousand customers, with a growth rate of two thousand new customers daily. Machine learning is not currently used in the business in question, for customer retention predictions. The potentially retainable customers of the business are identified using human assumptions on the customer data like ‘the customers who attempt more exams on the app are more likely to upgrade to the paid plan’ or ‘a customer who have referred someone, is more likely to retain’. The LEAD algorithm which is currently used to identify the potentially retainable customers, is based on 5 such assumptions. They are as follows:

1) Customers who attempt more exams on the app are more likely retain.
2) Customers who attempt more questions on the app are more likely retain.
3) A customer, who used a referral code to join is more likely to retain.
4) Customers who have used a promotional coupon code before, is more likely to retain.
5) Customers with more courses subscribed, are more likely to retain.

Based on these assumptions, a rank is assigned to each of the customers based on their activity data, and a dynamic leader board of customers is generated. A better rank in the leader-board is assumed to have a better possibility of being retained. The sales team of the business uses this leader-board to target their retention programmes and promotional campaigns to the customers. The customers in the top ranks of the leader-board are personally contacted by the sales team and are given promotional discounts to make them convert.

The motive of the research is to predict customer retention with more precision than the current algorithm they use, so that they can target the marketing and retention campaigns on these potentially retainable customers. Identifying customers with more probability of retention would help the business to narrow down the target to a small population of the customers, thereby reducing the cost associated with it.

Currently, the precision attained by the LEAD algorithm they currently use is only 20%. So, the aim of this research is to build a machine learning model which can predict the customer retention with a precision higher than 20%.

The hypothesis of this research is as follows:

H_0 : The supervised - machine learning models build using the customer data of Entri, cannot predict the customer retention with more than 20% retention precision.

H_A : The supervised - machine learning models build using the customer data of Entri, can predict the customer retention with more than 20% retention precision.

3.2 Data Understanding

The dataset used in the research is the customer master data of an India-based e-learning business called Entri. It consists of the data of all the customer who signed up for the app from July 1,2018 to Sep 30, 2018. The dataset has 80751 records of distinct customers with 12 features. The target variable of the dataset is 'converted', a binary feature with two values 0 and 1, which represents customer churn and retention respectively. The dataset has an imbalance ratio of 99.75:1.25 with no missing values or redundant data.

The features of the data are discussed in the following table.

Attribute	**Description**	**Type**	**Nature**
user_id	Unique ID for each user	ID	Independent
ques_attempt	Number of questions attempted by the user	Numerical	Independent
test_attempt	Number of tests attempted by the user	Numerical	Independent
no_subplatforms	Number of platforms the user has subscribed	Numerical	Independent
subexam_attempt	Number of subject exams attempted by the user	Numerical	Independent
mockexam_attempt	Number of mock exams attempted by the user	Numerical	Independent
inviteflag	Flag that shows whether the user signed up for the app using a referral code	Binary	Independent
couponflag	Number of coupons applied by the user	Numerical	Independent
score	Score obtained by the user on all exams	Numerical	Independent
perc_score	Percentage of score	Numerical	Independent
active_days	Number of days the user has been active on the app	Numerical	Independent
converted	Flag that shows if the user was upgraded or not	Binary	Target

Table 3. 1: Description of variables

More detailed understanding of the data is done using the following methods:

1) Using Descriptive Statistics, the basic quantitative analysis of the data is carried out. The measures of central tendency, range, standard deviation and skewness are measured here.

2) Missing values/ Redundant data analysis: Using the basic filter functions of Microsoft Excel, the data is checked for any missing values or duplicate records.

3) Exploratory analysis of the data: Using the matplotlib library of python, the data is visualized with histograms, scatterplots and bar-plots to identify the overall nature of the data and to understand the presence of outliers. Using the correlation matrix, the correlation of the independent features to the target variable and the correlation of the independent variables within themselves are studied. The relationship of the independent variables with the target variable and between the independent variables are studied using graphical representations.

3.3 Data Preparation

In the data preparation phase, using the insights from the Data Understanding, necessary steps are carried out to make the data fit for modelling. It includes feature extraction, encoding, Data Sampling, Outlier removal etc.

3.3.1 Feature Extraction

The retention strategies of the business are based only on the data used in the LEAD algorithm. For the research, more data features such as subexam_attempt, active_days and score were requested to the business and was added to the initial data. Two new data features were generated from the existing data as follows :

mockexam_attempt : Number of mock exams taken by the user are derived from the test_attempt and subexam_attempt data

mockexam_attempt = test_attempt – subexam_attempt

perc_score : The variable ‘Score’ is the total marks awarded for the customer for all the tests. So, customer who takes up more exams tends to have more Score. To have a better metric to capture the customer’s academic performance, the percentage of score obtained for each customer is calculated using the below formula

perc_score = (score/no_of_questions) * 100

3.3.2 Encoding

The dataset contains both numerical and categorical variables. Some machine learning models used in the research like Logistic Regression, SVM etc cannot work with categorical data directly. They assume that the variables used are numeric. For this reason, the categorical variables in the data has to be converted to numeric before feeding them to the classifiers. The dataset used in the research has two categorical variables ‘converted’ and ‘inviteflag’ with values ‘YES’ and ‘NO’. They are converted to the numerical values ‘1’ and ‘0’ using sklearn’s LabelEncoder function.

3.3.3 Data Sampling

Since highly imbalanced data can hinder the performance of the machine learning models, the data imbalance is minimized using Data Sampling in this research. Mainly 3 types of sampling are used: Random Over Sampling, Random Under Sampling and SMOTE.

Sampling ratios are varied in different experiments to obtain datasets of varying class imbalance. By applying the sampling technique, datasets with minority class proportions of 2% to 50% are achieved and used for the evaluating the performance of the machine learning experiments.

3.4 Modelling

In this phase, the pre-processed data is used to build machine learning models to predict the customer retention. Since the labelled data is available for training, supervised machine learning models are used for modelling in this research. Random Forest, Logistic Regression, SVM and XGBoost algorithms are used in the first experiment to predict the retention of the customers. In the initial experiment, the performance of these classifiers on a dataset with 75:25 class imbalance ratio is evaluated using the Stratified 10-fold cross validation method. Based on the performance of the classifiers on this experiment, the best classifier in terms of retention precision is selected for the experiments in the research. Binary classifiers are used in this research for modelling, as the target variable is dichotomous in nature. In the next set of experiments, modelling is carried out using sampled data to evaluate the performance of the models with change in imbalance ratio. The new features provided by the business was added to the best model to see if it contributes to the retention precision. A model with the best retention precision is selected from the experiments and is used for hypothesis evaluation and deployment.

3.5 Evaluation

The overall model Accuracy is not considered as a metric for the evaluation of the performance of the classifiers in the research. Since the dataset is highly imbalanced, with majority of data being customers who are not retained, the predictions of the model tend to be biased towards the ‘not retained’ class. Due to this, the model can still have a good over- all accuracy even with a poor classification performance on the minority class, i.e., retention class. So, instead of the overall model accuracy, the class precision of the retention class is used for the evaluation of the model performance in this research.

These metrics can be calculated from the confusion matrix.

Confusion Matrix:

	Actual Positive	Actual Negative
Predicted Positive	TP	FP
Predicted Negative	FN	TN

Where,

TP (True Positive): When both predicted and actual values are True

TN (True Negative): When both predicted and actual values are False

FP (False Positive): When the actual value is false, but it is predicted as true

FN (False Negative): When the actual value is true but it is predicted as false

Retention **Precision** is the measure of what proportion of the identified retained customers were correct.

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

Even though the Retention Precision is the metrics used for the evaluation of the research, other parameters like Retention Recall, Churn Precision and Churn Recall are also evaluated in this research to study how well the majority classes are identified and precise the predictions are.

Retention **Recall** is the measure of what proportion of the retained customers were identified correctly. It is given by:

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

Even though the focus of the research is on the retention class, the performance of the machine learning models on the churn class (majority class) is also analysed to understand how well the model predict the other classes as well.

The **Churn Precision** of the model are given by:

$$Churn\ Recall = \frac{TN}{TN+FP}$$

And the **Churn Recall** is given by:

$$Churn\ Precision = \frac{TN}{TN+FN}$$

3.6 Strengths and Limitations

This section summarises the strength and limitations of the design and methodology used in the research.

Since the data is highly imbalanced, Stratified k-fold cross validation method is used for training and testing the machine learning models. Hold-out method of splitting the train/test data is avoided in the research due to the class imbalance problem. Due to the iterative nature of the stratified k fold cross-validation method, all the available data are used for training and testing with the proportion of the minority and majority classes kept same in both training and testing. This has an upper hand over the conventional train and test split method where only the information contained in the training split is used to build the model. Whereas, the whole data is used for training and testing in the stratified K fold cross validation method.

The research is carried out by building different machine learning models using different features of the customers and comparing the performances of the models. Feature selection is done to eliminate the features that are irrelevant to the model. Eliminating irrelevant features from the model help to attain shorter training time and avoid over fitting. Another major strength of the research is that new features that are relevant in the prediction of customer retention were found out, apart from the conventional data features used for prediction by the business. This could help to revamp the current algorithm used by the business in predicting the retention.

The main limitation of the research is the very high imbalance of the data with only 1.25% of the data representing the retained customers. Due to the heavy imbalance in the data, the classifiers are more likely to be biased towards the majority class (churned customers). Also, due to the imbalance problem, random under-sampling of the data is used to under sample the majority data, which leads to the data loss problem. Potentially useful information is discarded when the data is under-sampled.

Also, the unavailability of customer profile data is a limitation for the thesis. Due to this, the geography, gender, age and related data, which could probably help better retention prediction, couldn’t be used to build the model to predict customer retention.

4. IMPLEMENTATION AND RESULTS

This section outlines how the research is implemented, the various stages of the CRISP – DM methodology included in the research, the experiments carried out in the research, and the results of the experiments.

The research has been implemented using the Cross Industry Standard Process for Data Mining (CRISP - DM) methodology. It is a structured approach commonly used in planning a data mining project. CRISP – DM consists of a sequence of events and can be backtracked to the previous events. The events included in the CRISP – DM methodology are: Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation and Deployment. Business Understanding has been previously outlined in the Design phase.

4.1 Data Understanding

A deep understanding of the data is required to plan and execute a machine learning research. A detailed analysis of the customer data is done using the statistical and visual analysis of the data in this research. Other factors like correlation of the features in the data, outlier detection and the data distribution of the customer data is studied here.

4.1.1 Dataset

The dataset used in the research is the customer data of an app-based e-learning platform called ‘Entri’. The data consists of 80751 records with 10 features. It contains the customer activity data from July 1, 2018 to September 30, 2018. The target variable is ‘converted’ which is a binary variable that denotes whether a customer was retained or churned. All the variables in the data are discussed in detail below.

a) ques_attempt

The ques_attempt is a numerical variable that denotes the total number of questions, a customer have attempted in the app. It is the sum of all the questions attempted in each of the tests attempted by the user. It ranges from 0 to 25788 with a standard deviation of 389.62. The mean value for the ques_attempt variable is 101.78. It could be seen that this variable is highly skewed to towards the left as shown below.

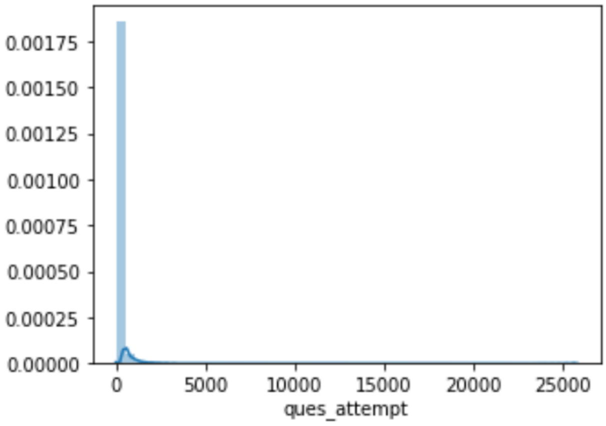


Figure 4. 1: Frequency distribution of ques_attempt

b) test_attempt

The test_attempt variable is a numerical variable that indicates the total number of tests the user has enrolled. It has a mean value of 7.8 with a standard deviation of 30.64 from the mean. The values of this variable ranges from 0 to 1873. The data is highly left skewed as shown below.

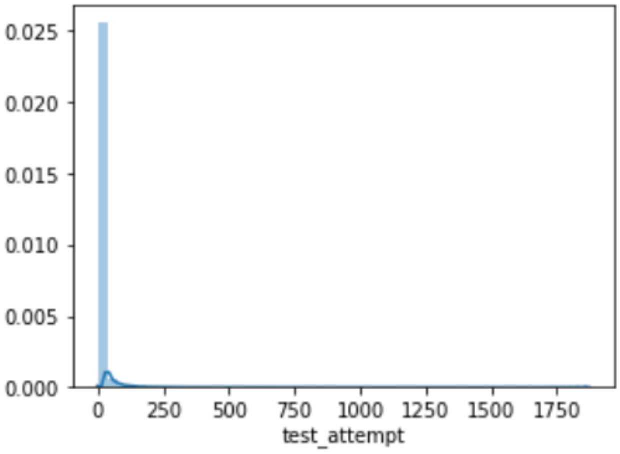


Figure 4. 2: Frequency distribution of test_attempt

c) no_subplatforms

The app has different platforms like ‘private job exams’, ‘10th grade exams’ ‘Engineering entrance exams’ etc. Each user can enrol to any of these platforms are take tests of the respective platforms. The ‘no_subplatforms’ variable denotes the number of platforms the user have subscribed to. This variable has values ranging from 0 to 30 as 30 is the total number of platforms available in the app. It has a standard deviation of 3.18 and a mean of 2.75. The data is left skewed as shown in the frequency distribution below.

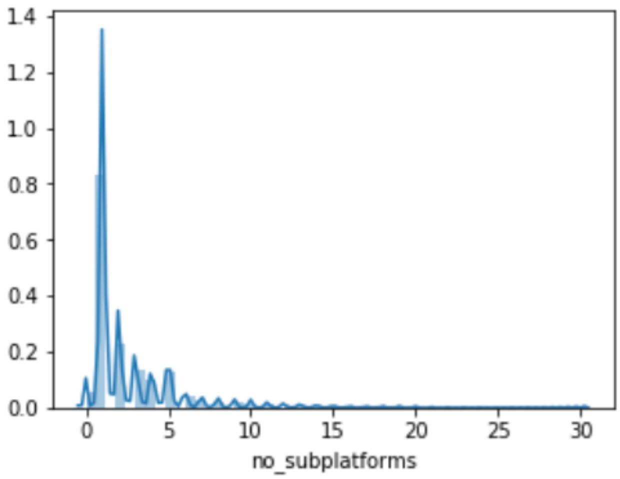


Figure 4. 3: Frequency distribution of no_subplatforms

d) subexam_attempt

The tests in the app are divided into two categories : subject exams and mock exams. Subject exams are the exams based on a particular subject, for example, Biology. Whereas, mock exams are the ones that follows the pattern of exams conducted for a particular objective, like, exam for a government job or an entrance exam for the university etc.

Subexam_attempt is a numerical variable that denotes the number of subject exams the user have enrolled in the app. It has a mean value of 5.81 with a standard deviation of 26.74. The values ranges from 0 to 1719 for this variable. The data for this variable is left skewed as shown below.

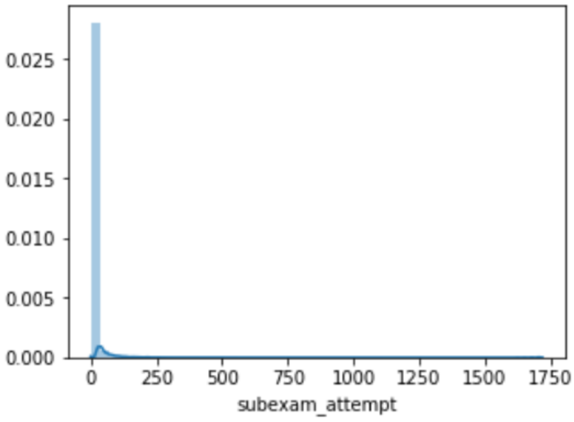


Figure 4. 4: Frequency distribution of subexam_attempt

e) mockexam_attempt

This is a numerical variable that denotes the total number of mock exams attempted by the user. It has a mean of 1.99 and a standard deviation of 8.3. The values ranges from 0 to 781 for this variable. The data is left skewed as shown in the plot below.

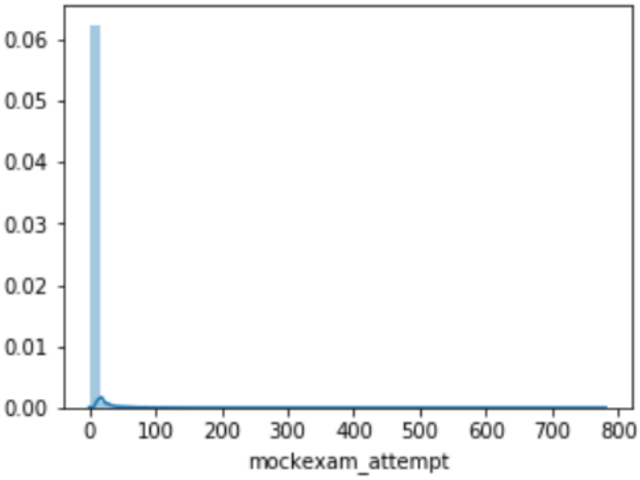


Figure 4. 5: Frequency distribution of mockexam_attempt

f) inviteflag

It is a binary variable that shows if a user joined the app via an invite from another customer or not. It has two values : 0 and 1. A '1' denotes that the user was referred to the app by another user and a '0' denotes that the user was not referred to sign up for the app. The distribution of this variable is shown in the following graph.

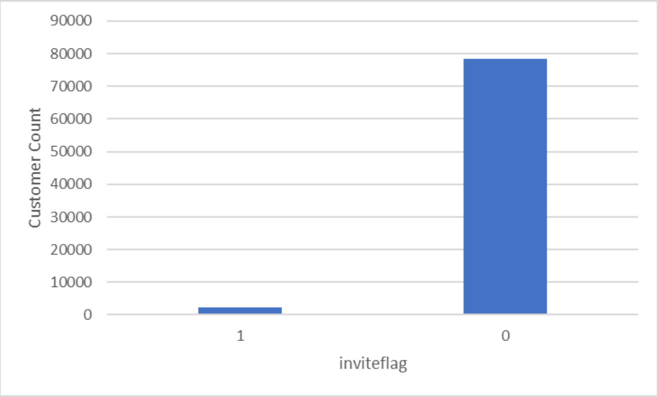


Figure 4. 6: Data distribution of inviteflag

g) couponflag

Various coupon codes have been given to the customers for promotional discounts when upgrading the membership and to increase the duration of their unlimited free access. This variable is a numerical feature that denotes the number of coupons the user has applied in the course of their app usage. It has a mean value of 2.9 and a standard deviation of 1.9. The values are limited to 29 because, it is the total number of coupons released by the business. The frequency distribution is shown below, and it is observed to be left skewed.

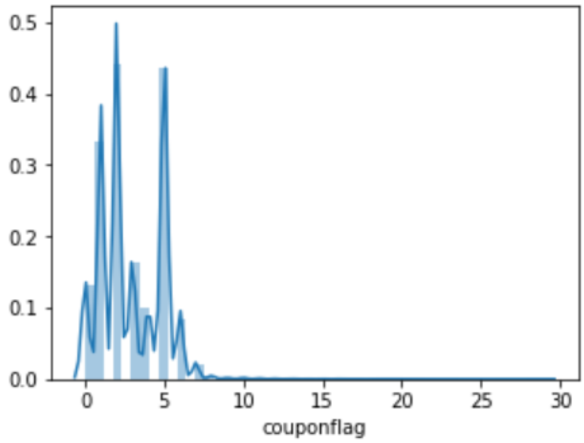


Figure 4. 7: Frequency distribution of couponflag

h) score

The score is a numerical variable that shows the number of right answers given by the user in all the exams. It is not a right metric that denotes the academic performance of a user, as more the number of questions attempted by the user, more is the chance of having a high score value. Due to this reason, a new variable has been formed from the score variable and is called 'perc score' and it will be discussed later in this section. The 'score' variable has a mean of 54.02 and a standard deviation of 235.91. The frequency distribution of score is as shown below.

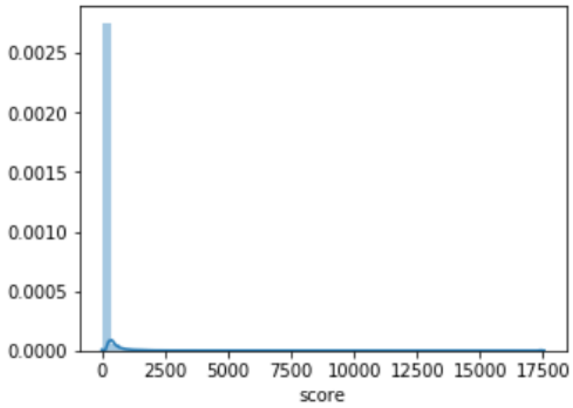


Figure 4. 8: Frequency distribution of score

i) active_days

The active_days is a numerical variable that denotes the total number of days, the user have been active on the app. i.e., have opened the app on their phone. It has a minimum of 0 and a maximum of 106 with a mean value of 1.88 and standard deviation of 4.02. The data is heavily left skewed as, most of the users who churned have used the app for less than 2 days. The frequency distribution for 'active_days' is as shown below.

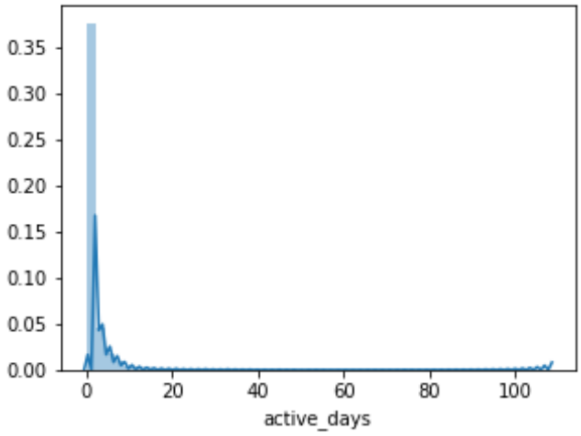


Figure 4. 9: Frequency distribution of active_days

j) perc_score

The 'perc_score' is a derived feature from the variables 'score' and 'ques'. It can be considered as a metric to measure the academic performance of a customer better the initial 'Score' feature. A score feature is the count of answers the user has correctly answered for all the appeared exams in the app, whereas, the 'perc_score' feature is an attribute which gives the proportion of correct answers given by the user in the exams. It has a mean of 41.30 and a standard deviation of 26.72 and the values ranges from 0 to 100.

The variable is derived as follows:

$$perc\ score = \left(\frac{score}{ques\ attempt}\right) * 100$$

The frequency distribution of perc_score is shown in Figure 4.10.

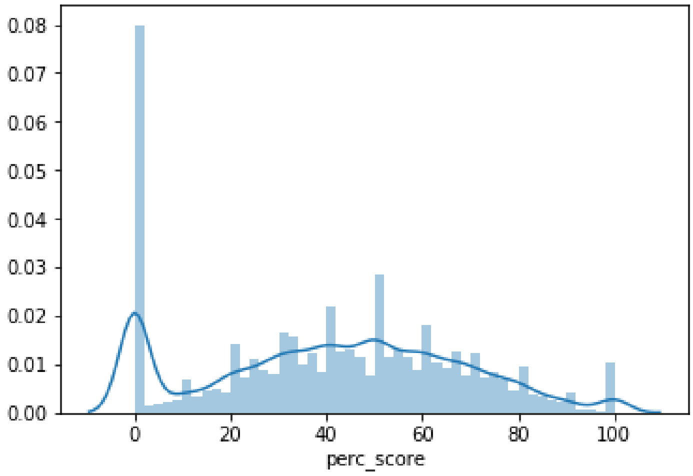


Figure 4. 10: Frequency distribution of perc_score

Discarding the customers who haven't enrolled for any exams, the data looks normal for this variable.

k) converted

This is the target variable of the research. It is a binary variable with two values 0 and 1, which represents retention and churn respectively. It denotes whether a customer upgraded to the premium version or not. The data distribution for this variable is as shown below.

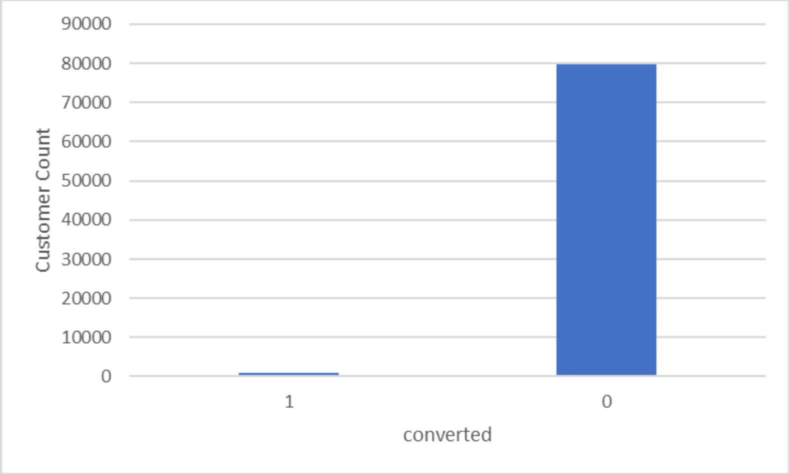
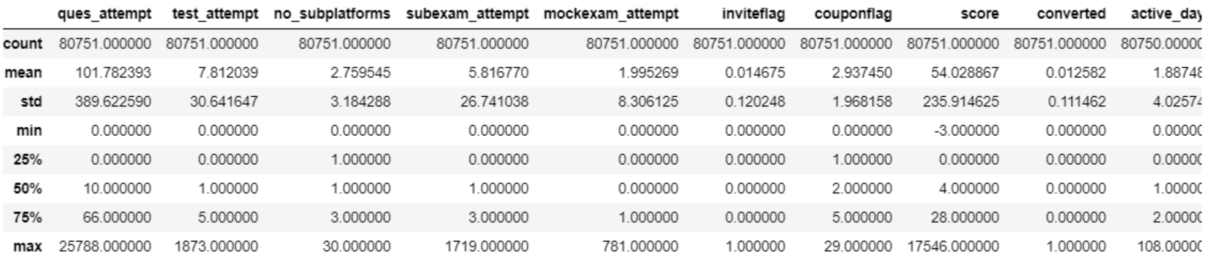


Figure 4. 11: Frequency distribution of converted

To have a better overview about data, the statistical metrics like count, mean, standard deviation, and measure of central tendency are shown in Table 4.1.



	ques_attempt	test_attempt	no_subplatforms	subexam_attempt	mockexam_attempt	inviteflag	couponflag	score	converted	active_day
count	80751.000000	80751.000000	80751.000000	80751.000000	80751.000000	80751.000000	80751.000000	80751.000000	80751.000000	80750.0000
mean	101.782393	7.812039	2.759545	5.816770	1.995269	0.014675	2.937450	54.028867	0.012582	1.8874
std	389.622590	30.641647	3.184288	26.741038	8.306125	0.120248	1.968158	235.914625	0.111462	4.0257
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	-3.000000	0.000000	0.0000
25%	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.0000
50%	10.000000	1.000000	1.000000	1.000000	0.000000	0.000000	2.000000	4.000000	0.000000	1.0000
75%	66.000000	5.000000	3.000000	3.000000	1.000000	0.000000	5.000000	28.000000	0.000000	2.0000
max	25788.000000	1873.000000	30.000000	1719.000000	781.000000	1.000000	29.000000	17546.000000	1.000000	108.0000

Table 4. 1: Descriptive Statistics of the customer data

From the above table, it could be seen that the count of all the variables are 81683, so there is no missing data. It can be seen that most of the data features are highly deviated from the mean. This calls for standardization of the data before modelling.

4.1.2 Correlation Analysis

Correlation analysis of the data is done to analyse the correlation of the independent variables and the target variables, and the correlation between the independent variables. Spearman correlation is used to analyse the correlation between the variables. The correlation heatmap is generated as shown Figure 4.12 and the correlation matrix is listed in Table 4.2.

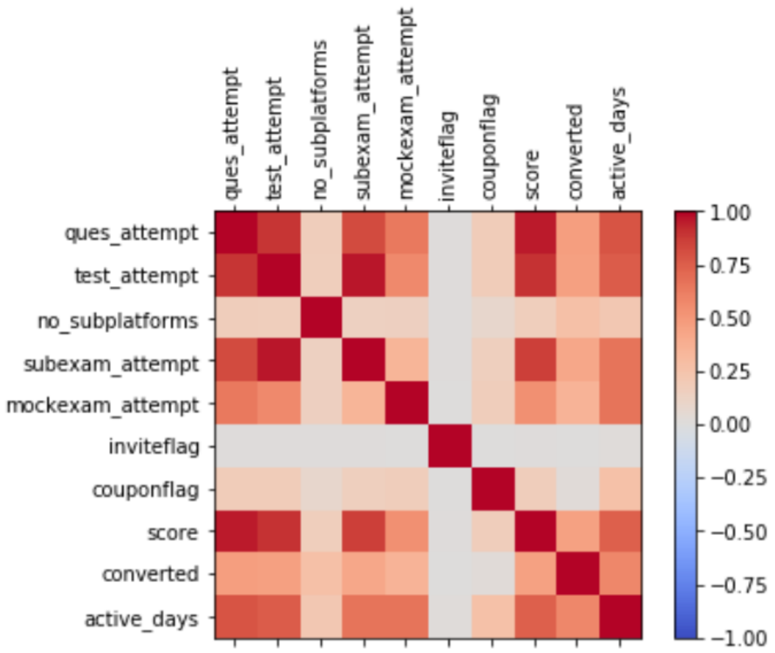
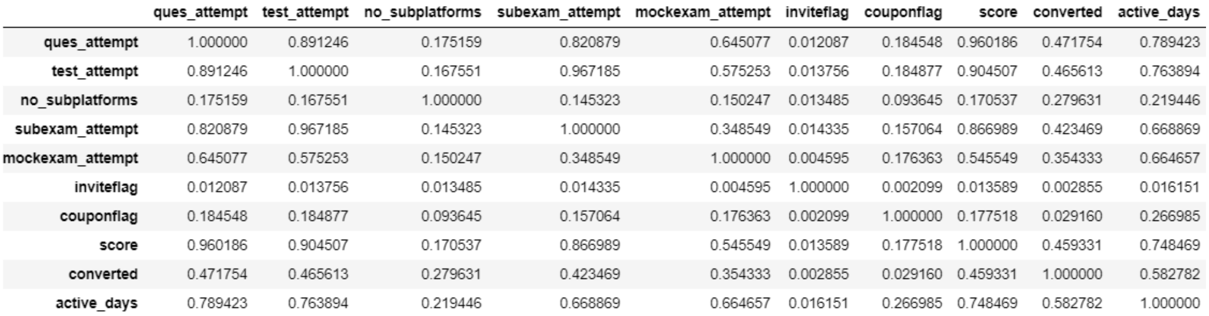


Figure 4. 12: Correlation heat-map of the variables



	ques_attempt	test_attempt	no_subplatforms	subexam_attempt	mockexam_attempt	inviteflag	couponflag	score	converted	active_days
ques_attempt	1.000000	0.891246	0.175159	0.820879	0.645077	0.012087	0.184548	0.960186	0.471754	0.789423
test_attempt	0.891246	1.000000	0.167551	0.967185	0.575253	0.013756	0.184877	0.904507	0.465613	0.763894
no_subplatforms	0.175159	0.167551	1.000000	0.145323	0.150247	0.013485	0.093645	0.170537	0.279631	0.219446
subexam_attempt	0.820879	0.967185	0.145323	1.000000	0.348549	0.014335	0.157064	0.866989	0.423469	0.668869
mockexam_attempt	0.645077	0.575253	0.150247	0.348549	1.000000	0.004595	0.176363	0.545549	0.354333	0.664657
inviteflag	0.012087	0.013756	0.013485	0.014335	0.004595	1.000000	0.002099	0.013589	0.002855	0.016151
couponflag	0.184548	0.184877	0.093645	0.157064	0.176363	0.002099	1.000000	0.177518	0.029160	0.266985
score	0.960186	0.904507	0.170537	0.866989	0.545549	0.013589	0.177518	1.000000	0.459331	0.748469
converted	0.471754	0.465613	0.279631	0.423469	0.354333	0.002855	0.029160	0.459331	1.000000	0.582782
active_days	0.789423	0.763894	0.219446	0.668869	0.664657	0.016151	0.266985	0.748469	0.582782	1.000000

Table 4. 2: Correlation matrix of the variables

It can be seen that the feature ‘active_days’ is the mostly correlated feature to the target variable ‘converted’ compared to the other features. The features like ‘ques_attempt’, ‘test_attempt’ ,’subexam_attempt’, and ‘score’ also has a positive medium correlation to the target variable. The features ‘coupon_flag’ and ‘inviteflag’ has the least correlation with the target variable.

The features ‘ques_attempt’ and ‘test_attempt’ exhibits very high positive correlation. Also, the feature ‘subexam_attempt’, ‘score’, ‘test_attempt’ and ‘ques_attempt’ are highly mutually correlated in the positive direction.

Even though these features are highly correlated to each other, they are correlated to the target variable ‘converted’ as well. Considering this fact, none of the features are removed from the dataset assuming it would worsen the predicting power of the model.

4.1.3 Outlier Analysis

The boxplots for each of the variables are plotted to study about the outliers present in the data. It can be understood from Figure 4.13 that, outliers are present for most of the variables in the data. As the data is heavily imbalanced, an analysis was done to check if the outliers fall into the minority class. Because, any outlier present in the minority class could contain useful information about the minority class, and the removal of such outliers could lead to information loss. So, each variable was analysed with respect to the target variable, to study whether the outlier removal would cause information loss.

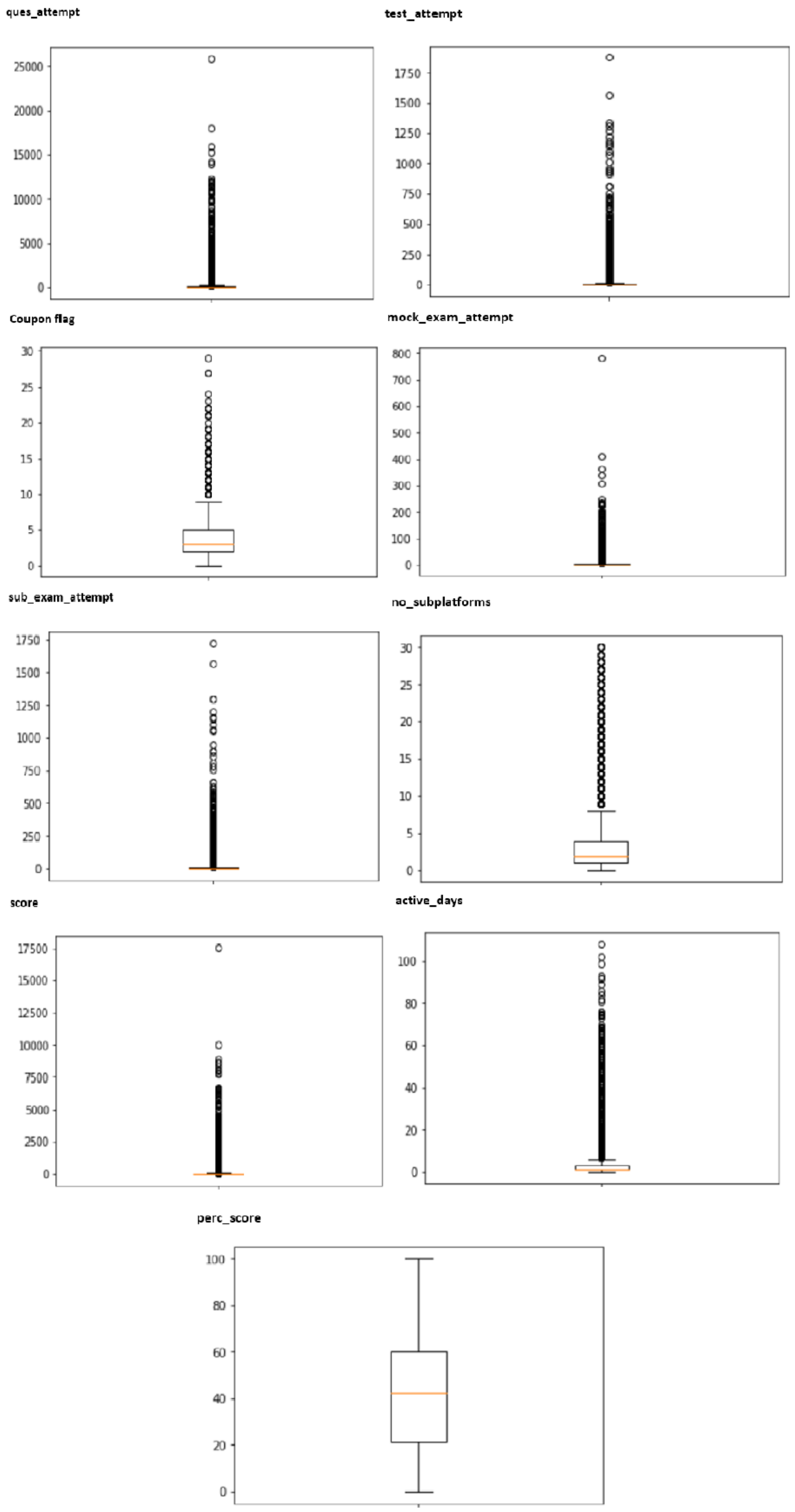


Figure 4. 13: Outlier analysis of the variables

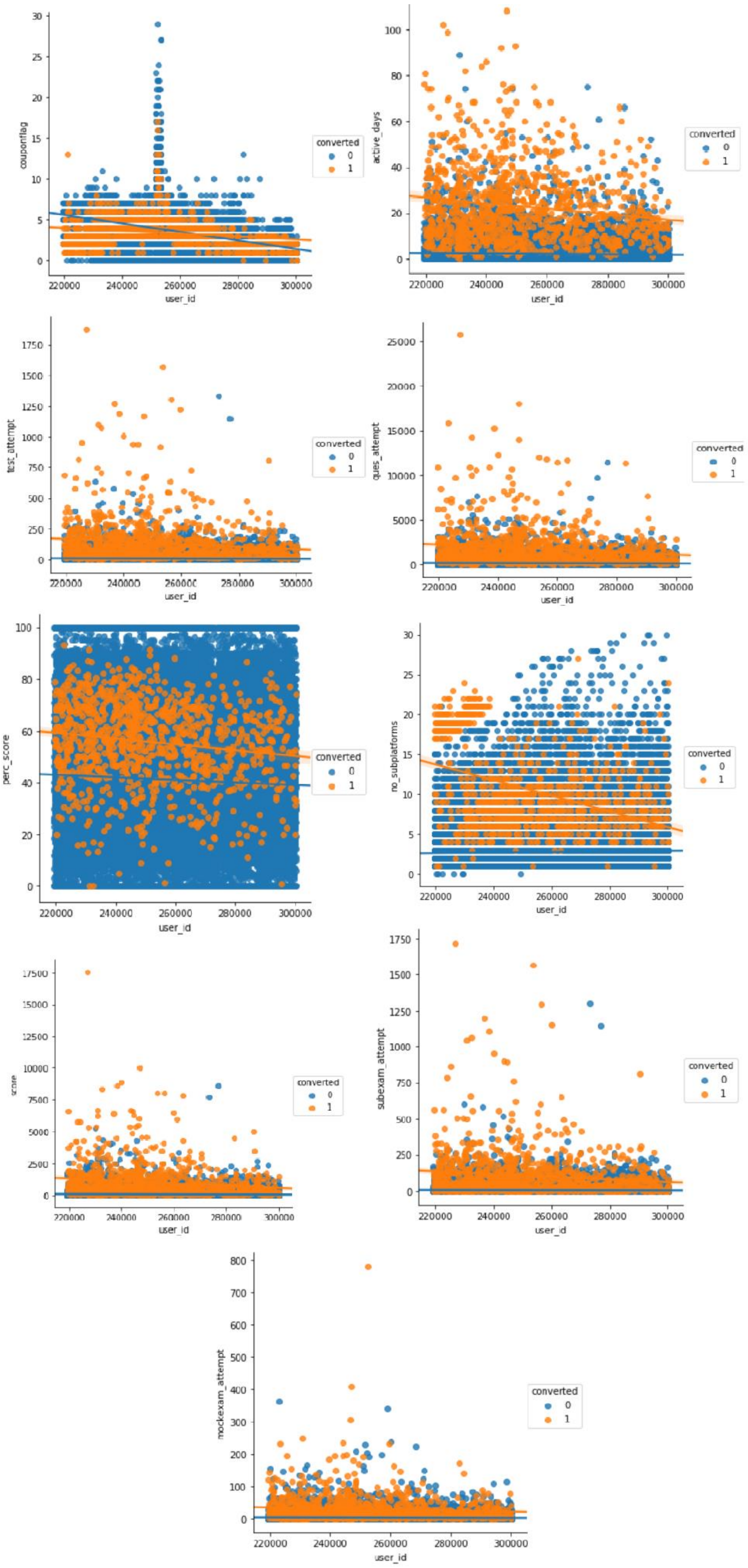


Figure 4. 14: Outlier Analysis of the variables

From the above plots, it could be observed that, except for the variables ‘couponflag’, all the outliers present in the data falls into the minority class. i.e., ‘converted’ = ‘1’ class. So, in this case, it’s assumed that, removal of these outliers would not only cause loss of information from the minority class, but also would significantly increase the class imbalance in the data. So, considering these two factors, the outliers are not removed from the data in this research.

4.2 Data Preparation

In the data pre-processing stage, based on the findings of the data understanding phase, the data is processed to make it fit for the modelling. Data Pre-Processing techniques include Encoding of the variables, Standardization, Noise elimination, Feature Extraction, Outlier Removal, and Data Splitting. The steps done in the pre-processing stage are detailed below.

4.2.1 Encoding

Encoding refers to converting one form of data to another. In this research, encoding is used to convert categorical or Boolean variables to numeric variables, to make the classifier easy to process. The data consists of two Boolean variables ‘inviteflag’ and ‘converted’. The Boolean data ‘invite_flag’ and ‘converted’ are encoded to binary values 0 and 1 (FALSE and TRUE respectively) using the LabelEncoder function of the scikit library.

4.2.2 Standardization

Standardization is a method of re-scaling the data to have a mean value of 0 and standard deviation of 1. A data with high deviation from the centre would have a negative effect on the performance of the classifiers as the standard classifiers assumes that the data has a Gaussian distribution. With the insights from the data understanding phase, it was observed that the data has a huge deviation from the mean. So, the data has to be standardized before applying the machine learning algorithms.

A standardized score called z-score, for each instance is generated using the below formula:

$$z - score = \frac{X - Xmean}{S}$$

Where, X = data Sample

Xmean = Mean of Sample

S = Standard Deviation of the Sample

4.2.2 Noise Removal

Noise refers to any unwanted signal in the data that could negatively affect the performance of the classifiers. In this section, the data is analysed in the business perspective, to identify the noise contained in it.

Analysing the ‘ques_attempt’ feature, it could be deducted that 21214 of the customers haven’t attempted any questions, (ques_attempt = 0). In the business terms, they are the customers who have installed the application, but haven’t tried out the application. These customers are considered as ‘inactive’ customers by the business and are considered that they doesn’t contain any information that can help the predictions. So, these customers are removed from the initial dataset. This not only helps to improve the signal in the data but also helps to reduce the imbalance in the data, as most of the inactive users falls to the ‘churn’ class, i.e., majority class.

Stage	No of Records	No of features	Imbalance Ratio
Initial	80751	10	98.75:1.25
After Noise Removal	59537	10	98.30:1.70

Table 4. 3: Dataset Count After Noise Removal

4.2.3 Feature Extraction

The feature ‘Score’ is the count of the correct answers of all the tests of a customer. It cannot be considered as an accurate measure of the academic performance of the customer because, more the number of questions attempted, greater could be the Score, as each correct question is awarded with 1 mark.

To tackle this, a new feature ‘perc_score’ is created, which is derived from ‘ques_attempt’ and ‘score’ by the formula :

$$perc\_score = \left(\frac{score}{ques\_attempt}\right) 100$$

This new feature represents the performance of the customer while using the app more precisely compared to the ‘score’ feature.

Stage	No of Records	No of features
Initial	59537	10
After Feature Extraction	59537	11

Table 4. 4: Dataset Count After Feature Extraction

4.2.4 Outlier Elimination

From the graphical analysis of the features, outliers were detected for the features ‘ques_attempt’, ‘test_attempt’, ‘mockexam_attempt.’, ‘active_days’, and ’no_subplatforms’. The outliers were not eliminated because, most of the outliers falls into the minority class, and eliminating outliers would cost information loss of the minority class, and increase the imbalance ratio. So, no Outlier elimination has been used in this research.

4.2.5 Data Splitting

10 fold Stratified K fold cross validation method is used in the research to evaluate the model performance. In each iteration, 9 folds of data are used for training and 1 fold for testing. A stratified approach is used in splitting the data into folds, to ensure that each folds has a representative sample of the retention and churn classes. It is implemented using StratifiedKFold function of scikit library. Each folds in the cross validation here has 5940 records with an imbalance ratio of 98.3:1.7.

4.3 Data Modelling

After the data understanding and data pre-processing phases, the pre-processed data is then used to build the machine learning models to predict the customer retention. In this phase, many machine learning models are build using different algorithms, sampling methods of varying dataset sizes and other parameters. A series of experiments are carried out in this phase to build a well performing model that can effectively predict the customer retention with best class precision.

The performance of the machine learning models is evaluated using retention precision, i.e., how precisely is the model predicting the retainable customers from the total customer population. Total Accuracy of the model is not used as a metrics to evaluate the model performance because, due to the heavy imbalance of the data, the predictions would be biased towards the majority class, i.e., churn class. In this case, the Churn Accuracy would be very high, there by resulting in a very high total accuracy. Due to this reason, the class Precision of the minority class, i.e., retention class is considered as the metrics to evaluate the performance of the model. The various experiments carried out in this phase are detailed below.

4.3.1 Experiment 1: Classifier Selection

To choose a classifier algorithm for the experiments, the performance of different classifiers such as Random Forest, Logistic Regression, SVM and XGBoost were tested on the dataset. The imbalance ratio is maintained as 75:25 (Churned:Retained) from the initial dataset to carry out this experiment. For that, from the original dataset with 59404 records (Retained: 1000 and Churned: 58404), a sample of 5000 records (Retained: 1000, Churned: 4000) was selected by taking a random sample of 4000 customers from 58404 churned customers. The model is evaluated using Stratified 10-Fold Cross – Validation method and the classifier performance is evaluated using class Precision and class Recall.

1) Logistic Regression

A logistic regression classifier is used to model the data in this experiment. It is implemented using the LogisticRegression package of sklearn library in python. The model is evaluated using the 10-fold Stratified sampling method and the default parameters were used.

Algorithm	Data Size (Count, Features)	Imbalance Ratio	Evaluation Method
Logistic Regression	5000,8	75:25	Stratified 10 Fold Cross - Validation

Table 4. 5: Experiment Design for Classifier Selection: Logistic Regression

2) Random Forest

Random forest is an ensemble of decision trees algorithm which is used for classification and regression problems. It is implemented in the research using the RandomForestClassifier package of sklearn library in python. Default parameters of the RandomForestClassifier are used to build the model.

Algorithm	Data Size (Count, Features)	Imbalance Ratio	Evaluation Method
Random Forest	5000,8	75:25	Stratified 10 Fold Cross – Validation

Table 4. 6: Experiment Design for Classifier Selection: Random Forest

3) Support Vector Machine Classifier

Linear SVM algorithm is used to build the machine learning model here. Models with different SVM kernels (‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’) are built and the performance are evaluated. The kernel with best class Precision and class Recall is selected for comparison with the other algorithms. The SVM algorithm is implemented using the SVC package of the sklearn library.

Algorithm	Data Size (Count, Features)	Imbalance Ratio	Evaluation Method
SVM (linear)	5000,8	75:25	Stratified 10 Fold Cross – Validation

Table 4. 7: Experiment Design for Classifier Selection: SVM

4) XGBoost

XGBoost, also called as, Extended Gradient Boosting algorithm, which is a collection of decision trees coupled with gradient boosting advantage. It is implemented using the XGBClassifier package of the xgboost library. XGBoost is used in this experiment to model the customer data and the performance is evaluated.

Algorithm	Data Size (Count, Features)	Imbalance Ratio	Evaluation Method
XGBoost	5000,8	75:25	Stratified 10 Fold Cross - Validation

Table 4. 8: Experiment Design for Classifier Selection: XGBoost

4.3.2 Experiment 2: Baseline Model – No Sampling

In this experiment, the original data is used for training the machine learning model to study the performance of the classifier of un-sampled data. The retention proportion of the data is maintained at 1.7%. The classifier with the most class Precision and class Recall from the previous experiment is selected for this experiment. The performance of the model is evaluated using Stratified 10 Fold cross validation.

Algorithm	Data Size (Count, Features)	Imbalance Ratio	Evaluation Method
XGBoost	59404,8	98.3:1.7	Stratified 10 Fold Cross - Validation

Table 4. 9: Experiment Design for Baseline Model – No Sampling

4.3.3 Experiment 3: Random Under-Sampling

Random under-sampling is used in this experiment to reduce the number of instances of the majority class in the data and tackle the class imbalance problem. The majority class is under - sampled at different ratios in each iteration to produce datasets of different class proportions. XGBoost classifier with default parameters is used to build the machine learning models using each of these datasets and Stratified 10-fold cross validation is used for evaluating the performance of the models. Retention Precision metric used to evaluate the performance of the model.

Iteration	Algorithm	Sampling method	Data Size (Count, Features)	Imbalance Ratio	Evaluation Method
1	XGBoost	Random Under – Sampling	2000,8	50:50	Stratified 10 Fold Cross - Validation
2	XGBoost	Random Under – Sampling	5000,8	80:20	Stratified 10 Fold Cross - Validation
3	XGBoost	Random Under – Sampling	10000,8	90:10	Stratified 10 Fold Cross - Validation
4	XGBoost	Random Under – Sampling	20000,8	95:5	Stratified 10 Fold Cross - Validation
5	XGBoost	Random Under – Sampling	50000,8	98:2	Stratified 10 Fold Cross - Validation

Table 4. 10: Random Under Sampling Experiment Design

4.3.4 Experiment 4: Random Over-Sampling

In this experiment, random over sampling is done on the minority class generate multiple instances of the minority class and tackle the class imbalance problem. In each iteration, the proportion of Over - Sampling is altered to evaluate the performance of the classifier in different imbalance proportions of the data. Here, the over-sampling is applied only to the training data and the test data is not sampled to avoid over-fitting. XGBoost algorithm is used for the classification and 10 - fold Stratified cross validation is used to evaluate the results of the experiment. Retention Precision is used as the metric for the performance evaluation of the model.

Iteration	Algorithm	Sampling method	Data Size (Count, Features)	Imbalance Ratio	Evaluation Method
1	XGBoost	Random Over - Sampling	116808,8	50:50	Stratified 10 Fold Cross - Validation
2	XGBoost	Random Over - Sampling	87170,8	66:33	Stratified 10 Fold Cross - Validation
3	XGBoost	Random Over - Sampling	69528,8	84:16	Stratified 10 Fold Cross – Validation
4	XGBoost	Random Over - Sampling	60837,8	96:4	Stratified 10 Fold Cross - Validation
5	XGBoost	Random Over - Sampling	59595,8	98:2	Stratified 10 Fold Cross - Validation

Table 4. 11: Random Over Sampling Experiment Design

4.3.5 Experiment 5: Synthetic Minority Over Sampling Technique (SMOTE)

Synthetic Minority Over-Sampling Technique is used in this experiment to over-sample the minority class. As described in section B, unlike random over-sampling, SMOTE generates synthetic samples of the minority data thereby reducing data overlapping and data complexity. Different datasets of various imbalance ratios are created by changing the SMOTE ratios in each iteration. XGBoost algorithm with default parameters is used as the classifier and

Stratified 10 fold cross validation is used for training and testing the data. Retention Precision is used as the metrics to evaluate the performance of the classifier.

Iteration	Algorithm	Sampling method	Data Size (Count, Features)	Imbalance Ratio	Evaluation Method
1	XGBoost	SMOTE	116808,8	50:50	Stratified 10 Fold Cross – Validation
2	XGBoost	SMOTE	97340,8	60:40	Stratified 10 Fold Cross – Validation
3	XGBoost	SMOTE	83434,8	70:30	Stratified 10 Fold Cross – Validation
4	XGBoost	SMOTE	73005,8	80:20	Stratified 10 Fold Cross – Validation
5	XGBoost	SMOTE	64893,8	90:10	Stratified 10 Fold Cross – Validation
6	XGBoost	SMOTE	61477,8	95:5	Stratified 10 Fold Cross – Validation
7	XGBoost	SMOTE	59901,8	97.5:2.5	Stratified 10 Fold Cross – Validation

Table 4. 12: SMOTE Experiment Design

4.3.6 Experiment 6: Dataset size selection

This experiment is carried out to determine the size of the data needed for the machine learning models to give optimal performance. For the experiments above, the data size differs depending on the set retention proportion and the sampling method used. Huge training data size can lead to higher model training time. So, in this experiment, a series of machine learning models are created in each iteration using different sizes of datasets with 50:50 imbalance ratio, by applying random under sampling for the majority class and random over-sampling for the minority class. The performance of the classifier is evaluated in terms of Retention precision. Stratified 10 - fold cross validation is used to train and test the data in this experiment.

Iteration	Algorithm	Sampling method	Data Size (Count, Features)	Imbalance Ratio	Evaluation Method
1	XGBoost	ROS + RUS	500	50:50	Stratified 10 Fold Cross - Validation
2	XGBoost	ROS + RUS	1000	50:50	Stratified 10 Fold Cross - Validation
3	XGBoost	ROS + RUS	2000	50:50	Stratified 10 Fold Cross – Validation
4	XGBoost	ROS + RUS	5000	50:50	Stratified 10 Fold Cross - Validation
5	XGBoost	ROS + RUS	7500	50:50	Stratified 10 Fold Cross - Validation
6	XGBoost	ROS + RUS	10000	50:50	Stratified 10 Fold Cross – Validation
7	XGBoost	ROS + RUS	15000	50:50	Stratified 10 Fold Cross – Validation
8	XGBoost	ROS + RUS	20000	50:50	Stratified 10 Fold Cross – Validation
9	XGBoost	ROS + RUS	25000	50:50	Stratified 10 Fold Cross – Validation
10	XGBoost	ROS + RUS	30000	50:50	Stratified 10 Fold Cross – Validation

Table 4. 13: Dataset Size Selection Experiment Design

4.3.7 Experiment 7: Addition of new data attributes

To evaluate if new data features can boost the model performance, the business was requested for additional data features, to add to the existing features of the machine learning model. A new feature ‘active_days’ were provided which indicates the number of days the customer have been active on the app. In this experiment, the new feature has been added to the existing model and the performance of the model is evaluated using Stratified 10 Fold Cross Validation method. The size of the data-set is set as 5000 with 50:50 imbalance ratio. The classifier performance is evaluated by means of Retention Precision.

Algorithm	Data Size (Count, Features)	Sampling method	Imbalance Ratio	Evaluation Method
XGBoost	5000,9	ROS + RUS	50:50	Stratified 10 Fold Cross -Validation

Table 4. 14 : Experiment Design for Addition of New Data Attributes

4.4 Results

This section outlines the results of the experiments mentioned in the previous section. Retention Precision is the metrices used to evaluate the performance of a model in this research. Total Accuracy of the model is not considered for the evaluation of the model performance, due to the class imbalance problem. Comparison of the performance of different models are also graphically represented in this section.

4.4.1 Results of Experiment 1: Classifier Selection

This experiment was done to select a classifier algorithm to be used for the research. The results of this experiment are as follows.

Algorithm	Data Size (Count, Features)	Retention Precision	Retention Recall	Churn Precision	Churn Recall
Logistic Regression	5000,8	0.813	0.566	0.992	0.997
Random Forest	5000,8	0.899	0.637	0.994	0.998
SVM	5000,8	0.783	0.666	0.994	0.996
XGBoost	5000,8	0.902	0.721	0.995	0.998

Table 4. 15: Experiment 1 - Results

From Figure 1 & 2, it can be said that XGBoost algorithm outperformed all the other machine learning models with better Class Precision and Class Recall. Churn Recall and Churn Precision remains almost constant for all the classifiers. Even though Random Forest classifier has equal Retention Precision compared to XGBoost, the Retention Recall is 8% below than XGBoost. The other classifiers like Logistic Regression and SVM have a significant difference of ~10% in the retention precision and ~15% in retention recall when compared to XGBoost classifier. Therefore, considering all these factors, XGBoost is selected as best performing classifier algorithm in the experiment.

4.4.2 Results of Experiment 2: Baseline Model – No Sampling

This experiment was done to analyse the performance of the classifier on the original data set without sampling. The results of this experiment are as shown below.

Classifier	Retention Precision	Retention Recall	Churn Precision	Churn Recall
XGBoost	0.68	0.64	0.99	0.99

Table 4. 16: Experiment 2 - Results

From the above table, it could be deducted that the performance of the classifier is lower than Experiment 1 with a loss of 22% in Retention Precision and 8% in Retention Recall. The Churn Precision and Churn Recall stays unchanged. This denotes that the class imbalance has negative effect on the performance of the XGBoost classifier and the dataset must be balanced to have better performance of the classifier.

4.4.3 Results of Experiment 3: Random Under-Sampling

Random Under – Sampling is used in this experiment to balance the dataset. Performance of the classifier for various imbalance ratios are tabulated below.

Iteration	Imbalance Ratio	Retention Precision	Retention Recall	Churn Precision	Churn Recall
1	50:50	0.954359	0.947981	0.949051	0.953883
2	80:20	0.940403	0.924427	0.962215	0.967967
3	90:10	0.920451	0.897619	0.969062	0.974351
4	95:5	0.902119	0.868271	0.973919	0.97889
5	98:2	0.886872	0.829868	0.977819	0.982482

Table 4. 17: Experiment 3 - Results

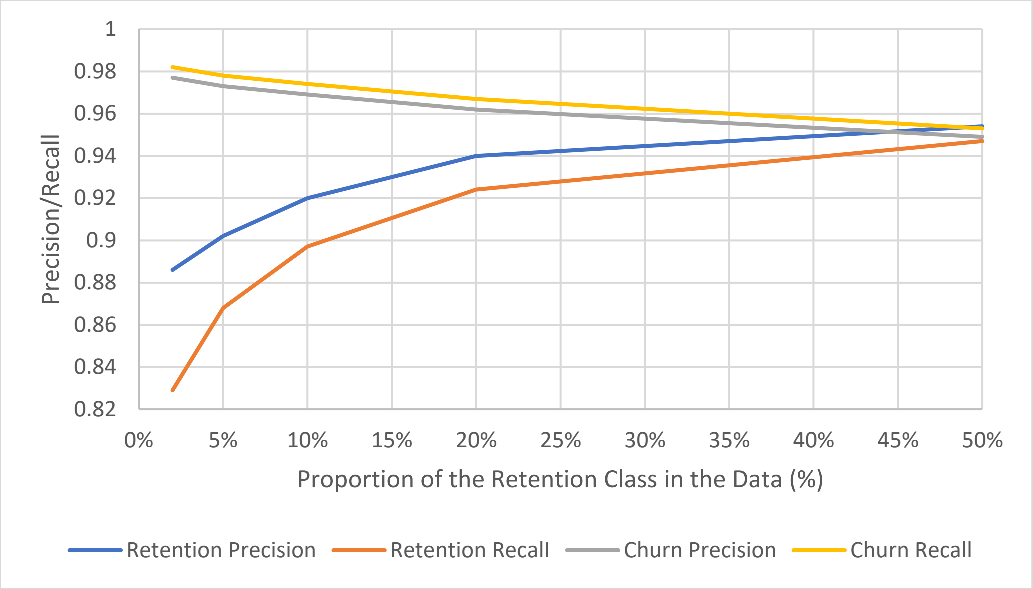


Figure 4. 15: Experiment 3 Results Plot

From Table 4.17 it is can be seen that, as the class imbalance reduces, the class Precision and Class Recall increases. The Retention Precision is 6.8% better when the class is balanced (50:50 ratio) when compared to the 98:2 Imbalance ratio. It could be also seen that there is an increase in Retention Recall by 11.2% when the class is balanced. Though the Retention Recall and Precision increases when the data is balanced, there is a slight decrease in the Churn Precision and Recall (~3% drop).

4.4.4 Results of Experiment 4: Random Over-Sampling

Random Over – Sampling is used in this experiment to balance the dataset. Performance of the classifier for various imbalance ratios are tabulated below.

Iteration	Imbalance Ratio	Retention Precision	Retention Recall	Churn Precision	Churn Recall
1	50:50	0.47	0.936	0.999	0.961
2	66:33	0.395	0.924	0.999	0.966
3	84:16	0.313	0.91	0.999	0.972
4	96:4	0.265	0.876	0.998	0.978
5	98:2	0.237	0.832	0.998	0.981

Table 4. 18: Experiment 4 - Results

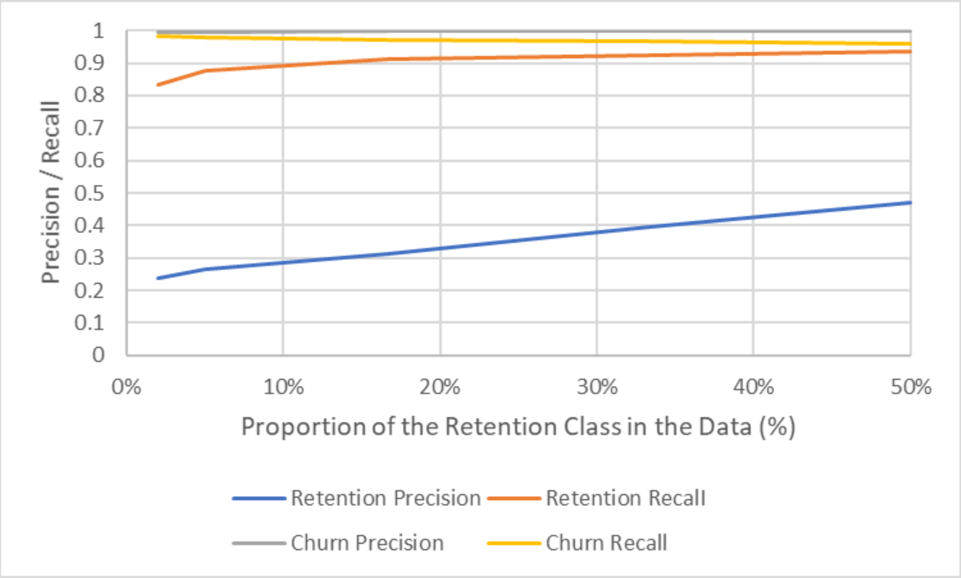


Figure 4. 16: Experiment 3 Results Plot

It can be observed from Table 4.18 that, as the class imbalance is reduced, the retention Precision and Retention Recall increases. As the imbalance is varied from 98:2 to 50:50, the retention precision increases by 23.3% and the retention recall increases by 10.4%. The churn precision and recall remain almost constant for this experiment.

Even though the retention precision increases as the imbalance reduces, compared to the Experiment 3, the retention precision of the over-sampling method is quite lower than that of the random under-sampling experiment.

4.4.5 Results of Experiment 5: Synthetic Minority Over Sampling Technique (SMOTE)

Synthetic Minority Over – Sampling Technique (SMOTE) is used in this experiment to over sample the minority class in this experiment. Performance of the classifier for various imbalance ratios are tabulated in Table 4.19.

From Figure 4.17, it is evident that as the class imbalance is reduced, the retention Precision and retention Recall increases. As the imbalance is varied from 97.5:2.5 to 50:50, the retention precision increases by 23.4% and the retention recall gains by 17%. The churn precision and recall remain almost constant for all the iterations of imbalance ratios. Even though the SMOTE has a better performance than random over-sampling method, it still cannot beat the performance of the random under-sampling method.

Iteration	Imbalance Ratio	Retention Precision	Retention Recall	Churn Precision	Churn Recall
1	50:50	0.606838	0.871287129	0.998360862	0.981651376
2	60:40	0.528169	0.851485149	0.998114156	0.984254897
3	70:30	0.493902	0.861386139	0.998240764	0.984874783
4	80:20	0.456989	0.841584158	0.997995239	0.987478304
5	90:10	0.416268	0.801980198	0.997500937	0.989709893
6	95:5	0.403756	0.742574257	0.996760125	0.991693528
7	97.5:2.5	0.372881	0.702970297	0.996273292	0.994297049

Table 4. 19: Experiment 5 - Results

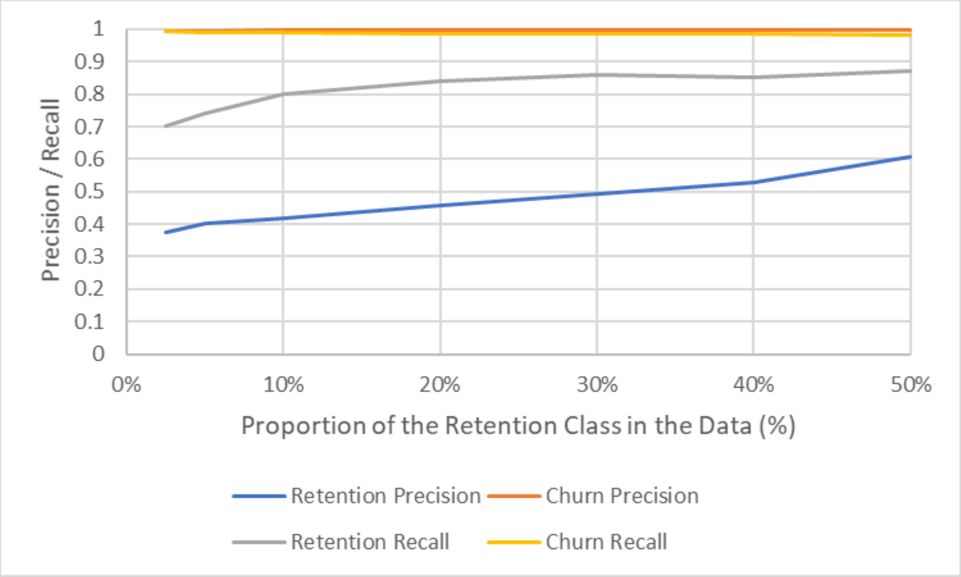


Figure 4. 17: Experiment 5 Results Plot.

A comparison of the performance of different sampling methods are tabulated in Figure 4.20.

Sampling Method	Retention Precision	Retention Recall	Churn Precision	Churn Recall
Random Under-Sampling	0.954	0.947	0.949	0.953
Random Over-Sampling	0.47	0.832	0.998	0.981
SMOTE	0.606	0.996	0.702	0.994
No Sampling	0.68	0.64	0.99	0.990

Table 4. 20: Comparison of performance of classifiers

Random – Under sampling method has the most Retention Precision when compared to Random Over – Sampling and SMOTE, with around 27.4% better than the second best performing method. Whereas, Retention Recall is the most when SMOTE algorithm is used, with a difference of 4.9% when compared to the Random Under Sampling method. Churn Precision and Churn Recall are the most for Over-Sampling methods, but the difference when compared to the Random Under-Sampling method, is negligible. Therefore, it could be deducted that Random Under-Sampling gives the best performance in terms of identifying the retention class precisely compared to the other sampling methods.

4.4.6 Results of Experiment 6: Dataset size selection

This experiment is conducted to analyse change in the performance of the classifier with change in dataset size. The imbalance ratio is maintained at 50:50 for all the iterations. The results of this experiment are tabulated in Table 4.21.

Iteration	**Data Size**	**Imbalance Ratio**	**Retention Precision**	**Retention Recall**	**Churn Precision**	**Churn Recall**
1	500	50:50	0.896	0.867	0.924	0.958
2	1000	50:50	0.909	0.875	0.924	0.959
3	2000	50:50	0.917	0.903	0.926	0.958
4	5000	50:50	0.923	0.900	0.937	0.952
5	7500	50:50	0.928	0.904	0.938	0.951
6	10000	50:50	0.925	0.903	0.939	0.947
7	15000	50:50	0.926	0.907	0.944	0.946
8	20000	50:50	0.925	0.910	0.948	0.944
9	25000	50:50	0.927	0.909	0.948	0.944
10	30000	50:50	0.928	0.908	0.948	0.941

Table 4. 21: Experiment 6 - Results

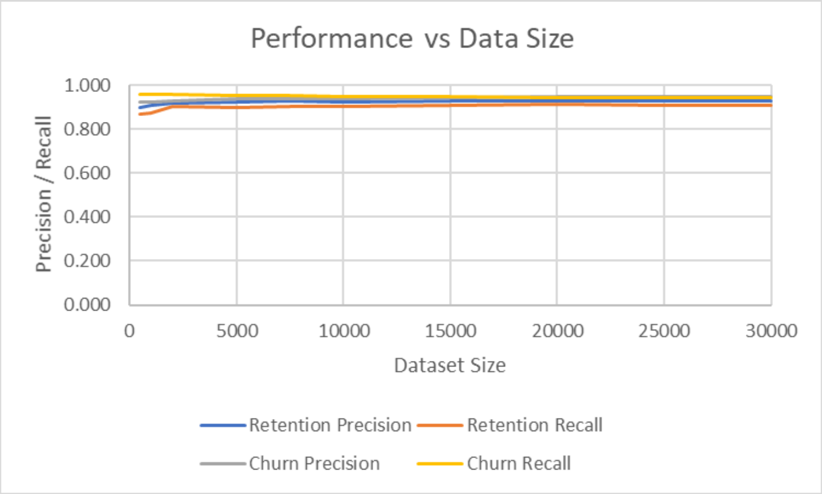


Figure 4. 18: Experiment 6 Results Plot

It could be deducted from the Figure 4.18, that the performance of the classifier is constant for all the iterations of dataset sizes after 5000. As the dataset size is increased from 500 to 5000 there is a gain of 2.7% in Retention Precision and an increase of 3.3% in Retention Recall. On analysing the performance of the classifier after 5000, it could be seen that even if the dataset size is upscaled to 30000, the retention precision increases only by 0.5% and Retention Recall by 0.8%, which is negligible when considering the factors like training time and infrastructure requirements. The churn Precision and Recall is constant for all iterations of dataset size. Therefore, optimum dataset size is determined as 5000 for the final model as shown below.

Data Size	**Imbalance Ratio**	**Retention Precision**	**Retention Recall**	**Churn Precision**	**Churn Recall**
5000,8	50:50	0.923	0.900	0.937	0.952

Table 4. 22: Performance Statistics of The Best Model

4.4.7 Results of Experiment 7: Addition of new data attributes

A new feature ‘active_days’ was added to the existing features of the model built in Experiment 6. In this experiment, the performance of the model with the new feature is analysed and is compared to the previous model.

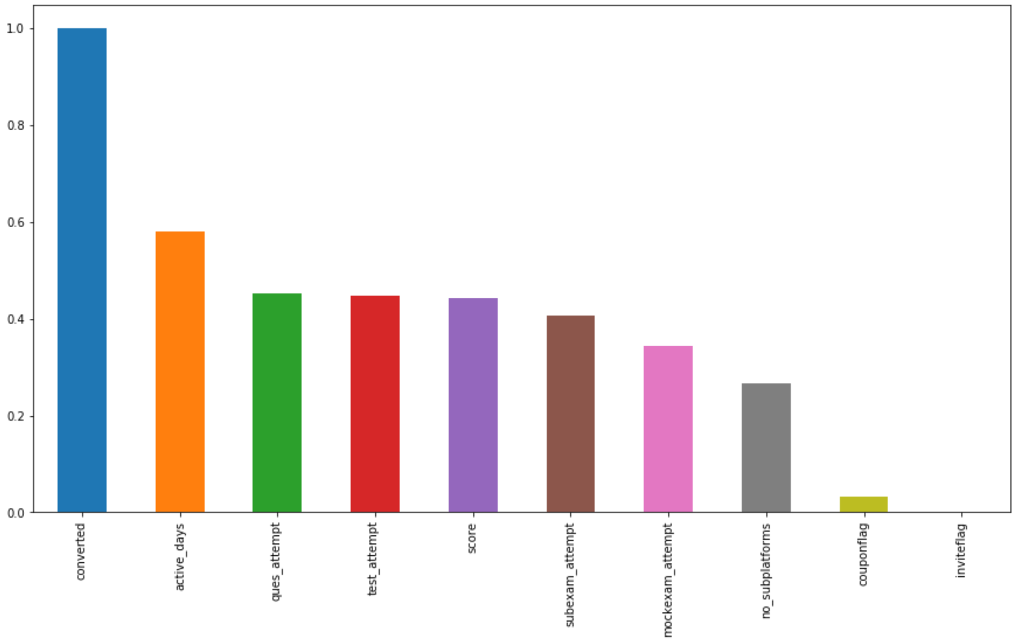


Figure 4. 19: Variable importance plot of 'active days'

Dataset Size	Imbalance Ratio	Retention Precision	Retention Recall	Churn Precision	Churn Recall
5000,9	50:50	0.912	0.822	0.917634	0.868444

Table 4. 23: Performance Statistics of The New Model with Added Attribute

It could be noted that, compared to the machine learning model built in Experiment 6, this model has a lesser Retention Precision and Recall. Comparing the results of Experiment 7 and Experiment 7, it can be observed that, even though the Retention Precision is constant, the Retention Recall drops by 7.8% when the new feature has been added to the model. It also has a slightly lesser (~4%) Churn Precision and Churn Recall when compared to the initial model. So, the new feature is said to decrease the performance of the machine learning model and is discarded from the final model.

5. EVALUATION AND DISCUSSIONS

This section analyses the results obtained from the experiments done in the research in the context of the research question and the business perspective. Also, the comparison of the sampling methods used in the research and the performance of classifiers are also discussed in this section. The section is concluded by discussing the limitations and strengths of the research.

5.1 Evaluation of the Results

A series of experiments were carried out in this research to build a high-performance machine learning model that can effectively predict the customer retention better than the currently used algorithm. The results of the experiments are analysed in this section.

The first experiment was to evaluate the performances of the different classification algorithms on a sample of 5000 records from the dataset with a 25% proportion of the retained customers and 75% of non-retained customers, to figure out the best performing classifier. From the results of this experiment, it was evident that XGBoost, a gradient boosted algorithm has the best class precision and class recall compared to the other algorithms like Logistic Regression, Random Forest Classifier and SVM as shown in Figure 5.1.

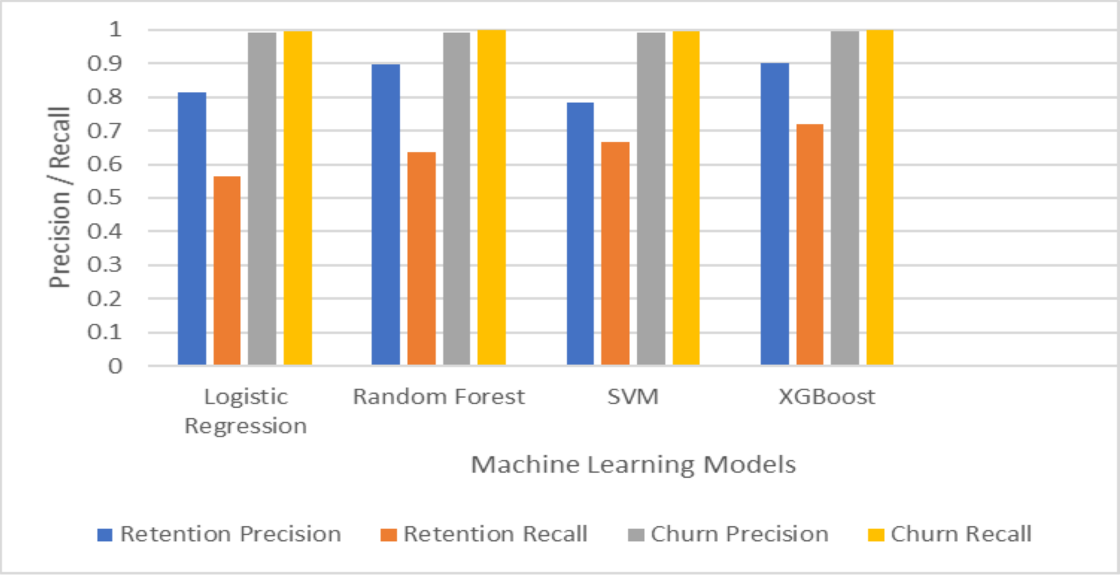


Figure 5. 1: Performance comparison of classifiers in Experiment 1

Gradient Boosted algorithms outperformed the standard algorithms like SVM, Logistic Regression and Random Forest classifiers. Also, the Random Forest classifier could deliver better performance than SVM and Logistic Regression. This agrees to the findings of Sahar (2018) that ensemble methods outperforms the standard single classifiers when applied to an unbalanced dataset. So, for the following experiments in the research, XGBoost algorithm was used as the classifier.

Experiment 2 was done to find out how well the XGBoost algorithm would work for the original data-set. For this, the original data set with the imbalance ratio of 98.3:1.7 was trained by the XGBoost algorithm. From the results, it could be seen that the performance was poor compared to the initial experiment. The retention precision was dropped by 22.1% and retention recall by 8%. This calls for implementing a sampling method that would decrease the class imbalance.

In the next set of experiments, the sampling methods like Random Under-Sampling, Random Over – Sampling and Synthetic Minority Over-Sampling Technique (SMOTE) were applied to reduce the class imbalance in the data and the performance of the XGBoost classifier on the sampled datasets are analysed. From the results, it could be seen that, random under-sampling has the most retention precision compared to the other models. But in the retention Recall, the SMOTE algorithm outperforms Random-Under-Sampling method by 5%. But since, the retention precision of Random Under Sampling is 35% better than SMOTE, it can be considered that Random-Under-Sampling is the best performing sampling method for this research.

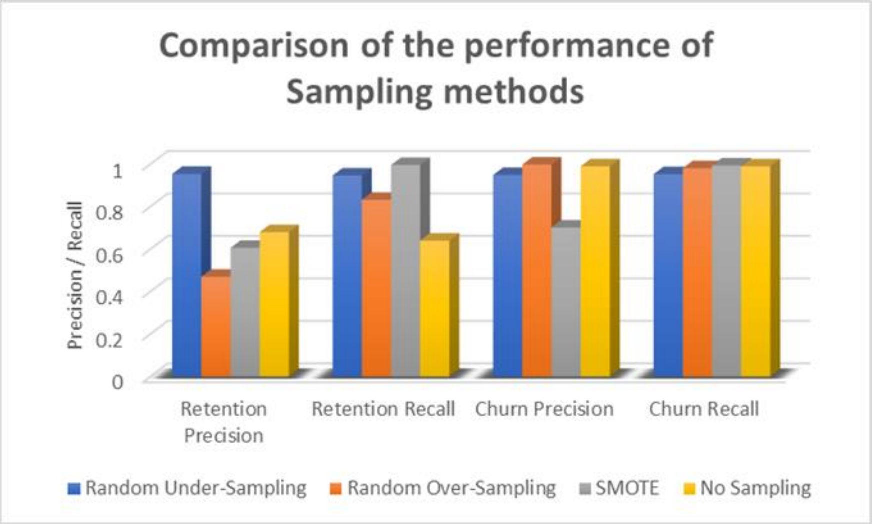


Figure 5. 2: Performance comparison of Sampling methods

It is also interesting to note that the churn Recall is almost constant for all the sampling methods. This might be due to the strong decision boundaries of the churned class in the dataset. Also, it can be seen that even without sampling the data, the churned customers are precisely identified. From this, it is evident that, due to the heavy imbalance in the data, the predictions are biased to the ‘Churn’ class. SMOTE and Random Over Sampling methods have a poor performance compared to the Under-sampling method. This might be due to the fact that, since there are only 1000 records of minority class and 54804 records of majority class in the original dataset, when over sampling is applied on the minority class, it creates multiple duplicates of the ‘Retention’ class which increases the data complexity and data over-lapping, which further causes, over fitting of the machine learning model, leading to the poor performance on the model on unknown data.

Also, the change in the performance with the change in proportion of the minority class in the training dataset is analysed in this experiment. As seen from the below graph, it can be deducted that, the classifier achieves the best retention precision and retention recall when both the training data has equal representations of the retention and churn classes, i.e., an imbalance ratio of 50:50. This contradicts with the findings of J. Burez et al (2008) that a 50:50 proportion is not required to have the best performance in retention/churn prediction. So, the imbalance ratio is set to 50:50 for the research.

Size of the dataset to be used for training was always a concern for the research considering the computational cost (training time, memory usage). The next set of experiments evaluated the performance of the XGBoost classifier on datasets of different sizes, maintaining the 50:50 class proportion. Random Under-Sampling and Random Over-Sampling are used for this experiment to yield datasets of varying size and constant imbalance ratio.

From the results, it could be seen that the classifier has a constant performance for datasets with more than 5000 records. There are fluctuating results for dataset sizes below 5000, and constant thereafter. So, this research selects 5000 as the size of the dataset for this research and for the model deployment. It is interesting to note that all the available data is not required to build a model with best performance. Reducing the training data size can help to decrease the training time and computational cost.

During the research, the business provided us with a new data attribute ‘active_days’. The business recommended this variable to be included in the model as they directly represent how many days the user was active on the app and is highly correlated with the target variable. To study the performance of the model with the new attribute included, the attribute was included in the final model and the performance is evaluated.

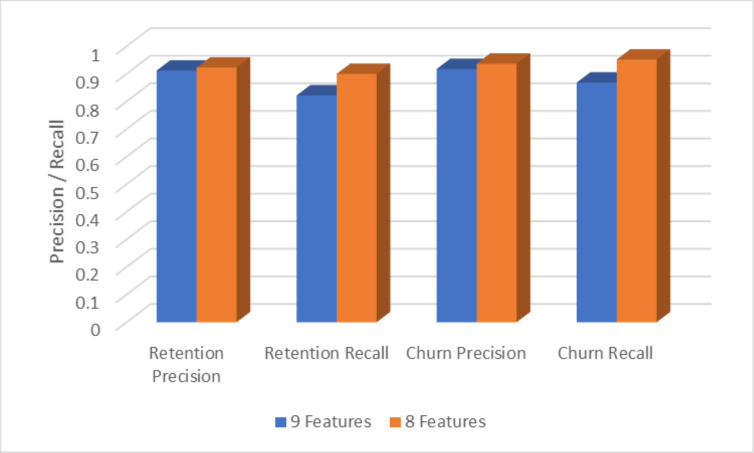


Figure 5. 3: Performance comparison of old and new models

It could be seen that when the new feature is added, it decreases the retention recall of the final model by 7.8% even though the retention precision is almost same (~1% difference). So, it can be concluded that the new attribute doesn’t significantly contributes to the performance of the final model.

5.2 Evaluation of Reasons of Retention

An analysis of the customer data was done to identify the reasons of retention/churn. From the data understanding phase, some deductions on the behaviour of the retained/churned customers are deduced from the data. The Figure 5.4 gives us a comparison of the behavioural patterns of the retained users and the churned users. Median measure is used to calculate the data for the above plots.

It can be observed that, the retained customers were actively using the app compared to the churned customers. The median active_days for the retained customer is 18 whereas, for a churned customer, it is 1. It denotes that, a customer who has less probability to retain will not be active for long in the app.

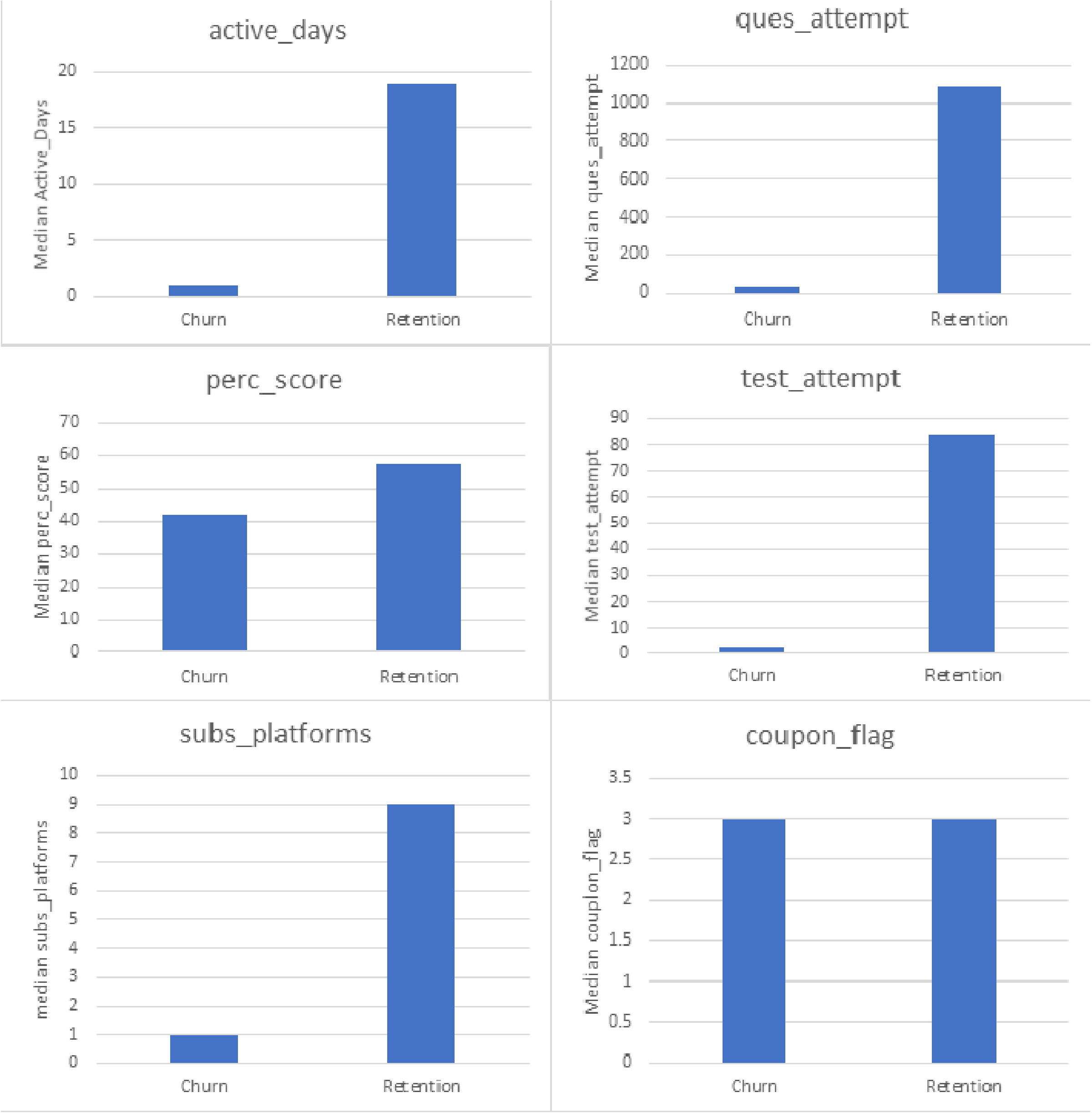


Figure 5. 4: Frequency plot – Relation between target and independent variables

The Figure 5.5 shows the number of days the customers have been active in the app before churn. It could be seen that, around 29216 of the 54804 customers were active only for 1 day before they churned and around 5486 customers haven't even signed up for the app after installing. There is a notable drop in the number of churned customers using the app after 1 day. Only 9183 customers have used the app after 4 days which accounts to only 15% of the total churned customers.

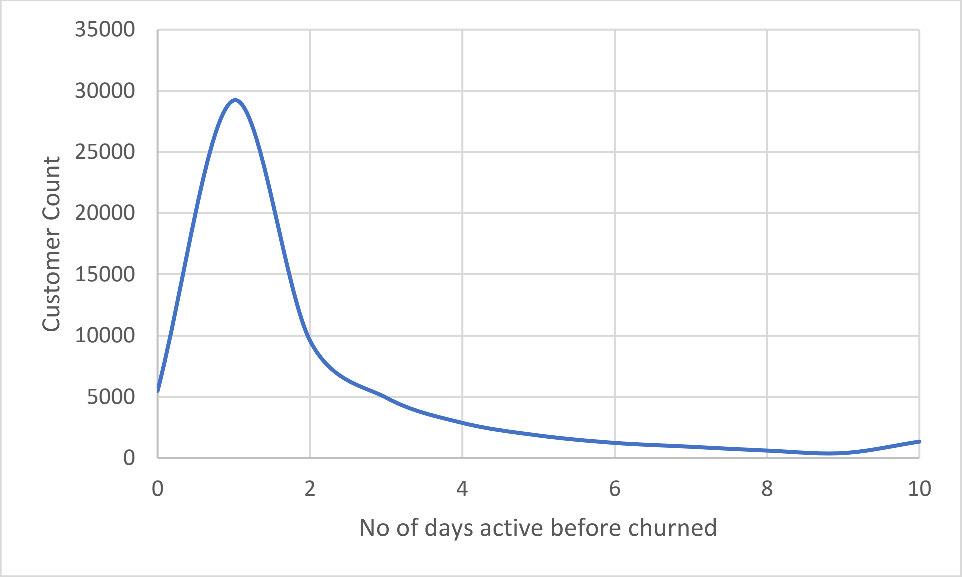


Figure 5. 5: Active days and Churn plot

Whereas, for the retained customers, 73.1% of the total retained customers have been active on the app for more than 10 days as seen from Figure 5.6.

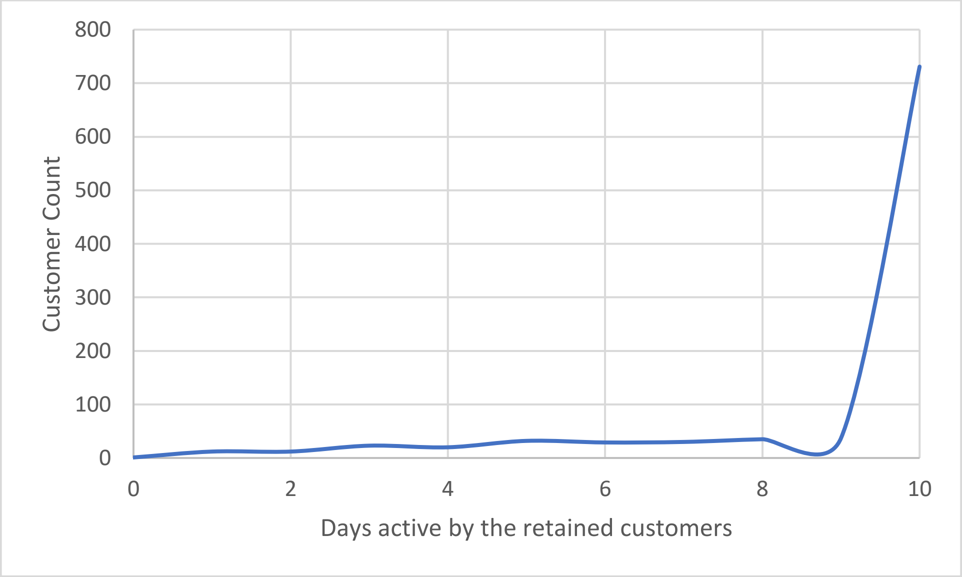


Figure 5. 6: Active days and Retention plot

Comparing the ques_attempt and the test_attempt, the retained customers have taken more exams and enrolled for more tests than the churned customers. Also, the median number of subscribed platforms is just 1 for the churned customer and is 9 for the retained customers, which implies that the customers who are subscribed to multiple platforms in the app have a better chance of retention.

The number of coupons used remains same for both retained and churned customers. When comparing the median perc_score of the customer, it could be seen that the retained customers have a median of 57.23% whereas the churned customers has 42.1%. This implies that, customers who has a better academic performance have a better chance of being retained.

Compared to the Massive Open Online Courses (MOOC), the retention rate of the business in question is very low. A study done by Onah et al (2014) observed that 11% of the users signed up for MOOC has completed a course. When compared to Entri, only 1.7% of the total 'active' customers have been retained.

5.3 Hypothesis Evaluation

The objective of the research was to build a machine learning model that can deliver a better performance in terms of retention precision, compared to the conventional algorithm which is an assumption – based algorithm with no learning attributes, used by the business to identify the potentially retainable customers. To achieve this objective, the hypothesis was coined as:

H_0 : "The supervised - machine learning models build using the customer data of Entri, cannot predict the customer retention with more than 20% retention precision."

A threshold of 20% was given because, it is the current retention precision of the business when the conventional algorithm is used.

From the research, a supervised machine learning model could be built with following precision and recall metrics:

Data Size	**Imbalance Ratio**	**Retention Precision**	**Retention Recall**	**Churn Precision**	**Churn Recall**
5000,8	50:50	0.923	0.900	0.937	0.952

Table 5. 1: Results for Hypothesis Evaluation

From table, it could be seen that the supervised machine learning model built with a dataset size of 5000 and 8 features, with 50:50 imbalance ratio has a Retention Precision of 92.3%.

So, the null hypothesis is rejected, since the Retention Precision exceeds 20% (the current retention precision of the conventional algorithm used by the business) and it can be concluded that supervised machine learning models can predict the customer retention with better retention precision than conventional assumption-based algorithm used by the business.

5.4 Model Deployment

From the research it could be deducted that the optimum performance is achieved with a dataset of 5000 records with XGBoost classifier algorithm. This model is deployed in the real customer data-set of the business to evaluate the performance of the model to the unknown real-world data. The model is run on the customer data of all customers who are have signed up for the app from October 1, 2018 – November 30, 2018. To increase the precision of the predictions, the threshold of the probability of the classifications is set as 99% using the predict.proba function of sklearn library. The list of customers who are predicted as retainable customers are given for the business sales team to plan the retention programmes. These customers were contacted by the sales team and was offered a promotional discount. After two weeks, the customer data of these customers who were predicted as retainable were generated again to evaluate the precision of the predictions.

The results are as follows.

Dataset Size	**Classifier**	**Predicted Retention**	**Predicted Churn**	**Actual Retention**
96564,8	XGBoost	40	96424	23

Table 5. 2: Deployed Model and Results

From the above results, the precision of the retention class can be calculated as.

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

$$Retention\ Precision = \frac{23}{40} = 0.575$$

So, the model was able to deliver a precision of 57.5% on the unknown data, compared to the 20% precision of the conventional LEAD algorithm. The research could increase the precision of the predictions by 37.5% compared to the conventional algorithm is the business.

5.5 Strengths of the research

The main strength of the research was its ability to precisely identify the retainable customers when compared to the algorithm which the business currently uses. The machine learning model which was evolved though this research has proved this by achieving better class precision and recall than the conventional algorithm. Also, as a part of the research, many data features of the customer data were analysed, and new features which highly represents the retained customer class were discovered. For example, the feature ‘perc_score’ was never used in the conventional business algorithm, but it was identified as a feature that has good variable importance in predicting the retention in this research.

The machine learning model which was built in this research, was deployed to the real time customer data of the business to analyse how well the model generalizes to unknown data. The model could achieve a retention precision of 57.5% retention precision, which is 37.5% better that the retention precision of the algorithm which the business currently uses.

This research proved that, not all the customer data is needed to train the model to achieve the best performance. The training data size could be scaled down from 59414 to 5000 records and still achieve moderately good performance. This could increase the speed of the model, and reduce the computational costs associated with using huge amount of data for model training.

5.6 Limitations of the Research

The primary limitation of the research was the huge difference in the number of samples of retained and churned customers in the data. Due to the heavy imbalance in the data, the over-sampling methods couldn’t perform well because, of the data over-lapping when the sampling is applied. Due to this, over-sampling methods couldn’t be efficiently utilized well in the research.

Unavailability of data features was another limitation of the research. The business in question, is not a data driven organization. They record and maintains only minimal customer data. Due to this, many of the data features that could have helped to increase the prediction performance, couldn’t be used in the research. For example, Customer age, geography, profession etc could have been better predictors of retention/churn.

6. CONCLUSION

6.1 Research Overview

The research is carried out to analyse the performance of machine learning models on the customer data of an app-based e-learning platform ‘Entri’ in predicting the customer retention. The research started with reviewing available literature, discussing traditional approaches in predicting customer retention, performance of various machine learning algorithms used for retention/churn prediction, different sampling methods used for tackling the class imbalance problem, problems caused due to class imbalance and how it hinders the generalization power of the classifiers. Also, the research question and the hypothesis were formulated that will guide the research to fulfil the research objective. A quantitative research methodology was selected, and appropriate experiments were designed to guide the research. The proposed design was implemented using supervised machine learning and the results were evaluated using retention precision metrics. The hypothesis evaluation is done using the obtained results to understand if the research objectives were achieved.

6.2 Problem Definition

The research aimed to evaluate the performance of supervised machine learning models on heavily imbalanced customer data set in predicting customer retention. To address this, the customer data set of an e-learning business ‘Entri’ was obtained and supervised machine learning algorithms combined with sampling techniques were used to predict the customer retention.

6.3 Design, Evaluation and Results

A quantitative and inductive research was designed to address the research question in this research. CRISP-DM methodology is followed throughout the research and Python programming was used to implement the research.

The business workflow is studied first by interacting with the company CTO and CMO. An overview of the current retention prediction and its performance have been studied. In the data understanding phase, the customer data set of the business is collected, and statistical analysis is done on the dataset to study the properties of the data. The data exploration is done by analysing the correlation, normality and other statistical metrics.

In the data preparation phase, the data has been cleaned to avoid the data errors, outliers etc. The categorical variables have been encoded into numeric values, and the data is standardized. The pre-processed data is then used for modelling in the Data Modelling phase. Sampling methods were used to sample the imbalanced data to a balanced dataset and their performance is evaluated. Random Under Sampling method is found to have the best class Precision and class Recall when compared to the Over sampling techniques. The performance of the classifier on different size of datasets keeping the class proportion at 50:50 is evaluated and 5000 is selected as the minimum dataset size to yield optimum performance in terms of class Precision. All the classifier performances are evaluated using Stratified 10 Fold cross validation and class Precision is used as the metrics to measure the performance of the classifiers. The retention precision of the final machine learning model evolved from the is 92.3% and retention recall is 90%. New features were added to the final model to see if there is an increase in the performance. As the new features didn't contribute to the have better performance, they are discarded from the research. The final model was deployed in real – time unknown customer data of the company to evaluate how well the model generalizes to the unknown data. The model could attain a retention precision of 57.5% which is 37.5% better than the conventional algorithm. The hypothesis evaluation is done to is confirmed that the research answers the research question.

6.4 Contributions and Impact

For academic/industrial research : Most of the retention/churn prediction literature made use of the supervised machine learning methods. But only a limited research were done into gradient boosting algorithms like XGBoost Whereas, this research makes use of XGBoost algorithm to predict the customer retention, contributing literature for the future research.

For the business : A customer retention prediction system with 37.5% better retention precision compared to the traditional algorithm could be generated through the research. This would help the business to implement more targeted marketing and retention programmes, and to identify retainable customers with improved precision. This would benefit the business by saving the misclassification cost of the traditional algorithm.

6.5 Future Work and Recommendations

This research has introduced the machine learning method to predict the customer retention from the customer data of the business in question. The machine learning model was built only using the available data which is currently been recorded by the business. In future, a better prediction model could be developed by including more data features. The factors that are potentially relatable to the retention should be identified from the business perspective and the data should be collected and maintained accordingly. For that, the business was recommended to capture Customer profile data (Gender, Age, Profession etc), app-based data (session timings, number of hits etc) from Google Play Store etc. The future research can include the above-mentioned data for building the machine learning model.

Also, the classifiers are built using default parameters in this research. Fine tuning of the classifiers and experimenting new algorithms in sampling is recommended for future work.

The primary aim of this research is to improve the retention precision of the predictions. Retention Recall is not considered as a performance metrics in this research. Future research can focus on the Recall of the retainable customers, thereby helping the business to identify more retainable customers from the customer population.

REFERENCES

Sharma, A., & Panigrahi, P. K. (2011). A Neural Network based Approach for Predicting Customer Churn in Cellular Network Services. *International Journal of Computer Applications*, 27(11), 26-31. doi:10.5120/3344-4605

Saher F. Sabbeh (2018) Machine-Learning Techniques for Customer Retention: A Comparative Study, *(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 2, 2018*, pp273-281

S. Babu, Dr. N. R. Ananthanarayanan (2014) A Review on Customer Churn Prediction in Telecommunication Using Data Mining Techniques, *International Journal of Scientific Engineering and Research (IJSER), Volume 3*, pp35 – 40

Amjad. H, Reham. D, Osama. H, Ruba. O,Hossam. F (2015) Hybrid Data Mining Models for Predicting Customer Churn, International Journal of Communications, *Network and System Sciences, Volume 8*, pp91-96 http://dx.doi.org/10.4236/ijcns.2015.85012

Xia, G.E., Jin, W.D. (2008) Model of Customer Churn Prediction on Support Vector Machine. *Systems Engineering—Theory & Practice*, 28, pp71-77.

Yabas, U., Cankaya, H. and Ince, T. (2012) Customer Churn Prediction for Telecom Services. *Computer Software and Applications Conference (COMPSAC), IEEE 36th Annual, Izmir*, pp358-359

Adwan, O., Faris, H., Jaradat, K., Harfoushi, O. and Ghatasheh, N. (2014) Predicting Customer Churn in Telecom Industry Using Multilayer Preceptron Neural Networks: Modeling and Analysis. *Life Science Journal, Volume 11*, pp75-81.

Burez, J. and Van den Poel, D. (2009) Handling Class Imbalance in Customer Churn Prediction. *Expert Systems with Applications*, 36, pp 4626 - 4636. http://dx.doi.org/10.1016/j.eswa.2008.05.027

Bart. L, Dirk. P (2005) Predicting customer retention and profitability by using random forests and regression forests techniques, *Expert Systems with Applications 29 (2005)*, pp 472-484

Rahul, J., Usharani, P., Churn Prediction in Telecommunication Using Data Mining Technology, *International Journal of Advanced Computer Science and Applications, Vol. 2, No.2*, 17-19

Afaq, K., Sanjay, J., M.M, Sapehri (2010) Applying Data Mining to Customer Churn Prediction in an Internet Service Provider, *International Journal of Computer Applications Volume 9, No 7,* pp9-14

Kim S, Choi D, Lee E, Rhee W (2017) Churn prediction of mobile and online casual games using play log data. *PLoS ONE 12(7)*: e0180735. https://doi.org/10.1371/journal.pone.0180735

Jin, Su., Kimberly, C., Tina, R., Brad, J., Customer Retention Predictive Modeling in HealthCare Insurance Industry, *Paper AD-007,* pp01-07

Zhang, Y., Liang, R., Li, Y., Zheng, Y. and Berry, M. (2011) Behavior-Based Telecommunication Churn Prediction with Neural Network Approach, *IEEE International Symposium on Computer Science and Society (ISCCS), Kota Kinabalu, 16-17 July 2011*, 307-310.

Chao, C., Andy, L., Leo, B., (2012) Using Random Forest to Learn Imbalanced Data, *PLoS ONE 12(2)*, pp01-12

Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. *In: Proceedings of the 22Nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM,* pp 785–794

Kyung, S., Taik, Lee., Hyun, K., (2005) An application of support vector machines in bankruptcy prediction model, *Expert Systems with Applications, Volume 28*, pp127–135, http://doi:10.1016/j.eswa.2004.08.009

Ali, R., Hossam, F., Jamal, A., Omar, AK. (2014), Information-An International Interdisciplinary Journal, Vol. 17(8), pp3962-3971

Jiong, M., Lijia, X., Xuliang, D., Haibo, P. (2013) Study on Customer Loyalty Prediction Based on RF Algorithm, *Journal of Computers, Vol. 8,* No. 8, pp2134-2138

Wirth, R., & Hipp, J. (2000). CRISP-DM: Towards a Standard Process Model for Data Mining, 11

Essam, S., Yehia, H., Ayman, K., Mona, N. (2012) A Proposed Churn Prediction Model, *International Journal of Engineering Research and Applications (IJERA), Vol. 2, Issue 4,* pp.693-697

Gotovac, S. (2010) Modeling Data Mining Applications for Prediction of Prepaid Churn in Telecommunication Services, *International Interdisciplinary Journal, Vol. 51, No. 3*, pp. 275-283

Cao, J.T., Zhang, H. and Zheng, Q.S. (2010) Retaining Customers by Data Mining: A Telecomunication Carrier's Case Study in China. *International Conference on E-Business and E-Government (ICEE), Guangzhou, 7-9 May 2010*, pp.3141-3144

Rushi, L., Snehlata, D., Latesh, Malik. (2013) Class Imbalance Problem in Data Mining: Review, *International Journal of Computer Science and Network (IJCSN) Volume 2, Issue 1*

Xinjian G., Yilong, Y , Cailing, D., Gongping, Y., Guangtong Z. (2009) On the Class Imbalance Problem, *Fourth International Conference on Natural Computation*, pp.192-201.

Kotsiantis, D., Kanellopoulos, S., Pintelas. P. (2006) Handling imbalanced datasets: A review, *GESTS International Transactions on Computer Science and Engineering 30 (1)* pp. 25-36

Maloof, M., (2003) Learning when data sets are imbalanced and when costs are unequal and unknown", *Proceedings of the ICML 2003 Workshop on Learning from Imbalanced Data Sets II*, pp. 73-80. Retrieved from https://www.site.uottawa.ca/~nat/Workshop2003/maloof-icml03-wids.pdf

Victoria, L., Alberto, F., Salvador, G., Vasile, P., Francisco, H. (2013) An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics, *Information Sciences 250 (2013)*, pp.113–141

Buda, M., Maki, A., Mazurowski M.A. (2018) A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259

Chen Chang (2012) A Study on the Impact of Customer Lifetime Value on Shareholder Value in Taiwan's Credit Card Market, *The Journal of Human Resource and Adult Learning Vol. 4, Num. 2, December 2008*.

Dawes, J. (2009). The Effect of Service Price Increases on Customer Retention: The Moderating Role of Customer Tenure and Relationship Breadth. *Journal of Service Research*, 11(3), 232–245. https://doi.org/10.1177/1094670508328986

Dubihlela, J., Molise - Khosa, P. (2014). Impact of e-CRM Implementation on Customer Loyalty, Customer Retention and Customer Profitability for Hoteliers along the Vaal Meander of South Africa. *Mediterranean Journal of Social Sciences*. pp 23-34

Magatef S.G, Tomalieh E.F (2015) The Impact of Customer Loyalty Programs on Customer Retention, *International Journal of Business and Social Science, Vol. 6*, No. S8(1)

Hoens T.R., Chawla N.V (2013) Imbalanced Datasets: From Sampling to Classifiers, *Imbalanced Learning: Foundations, Algorithms, and Applications, (1).*

Domingos, P. (2012) A Few Useful Things to Know about Machine Learning, *Communications of the ACM*, 55 (10), 78-87

Japkowicz, N., Myers, C., Gluck, M. (1995) A novelty detection approach to classification. *IJCAI'95 Proceedings of the 14th international joint conference on Artificial intelligence* - Volume 1, pp518-523 Retrieved from https://www.ijcai.org/Proceedings/95-1/Papers/068.pdf

Elkan, C. (2001) The Foundations of Cost-Sensitive Learning, *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI'01). Retrieved from http://web.cs.iastate.edu/~honavar/elkan.pdf*

Onah, D., Sinclair, J., Boyatt, R. (2014) Dropout rates of massive open online courses : behavioural patterns. *6th International Conference on Education and New Learning Technologies, , Barcelona, Spain, 7-9 Jul 2014,* Retrieved from https://warwick.ac.uk/fac/sci/dcs/people/research/csrmaj/daniel_onah_edulearn14.pdf

Xiong, H., Wu, J., Liu, L (2010) Classification with Class Overlapping: A Systematic Study, *The 2010 International Conference on E-Business Intelligence*

Yang, P., Liu, W., Zhou, B., Chawla, S., Zomaya (2013) A. Ensemble-based wrapper methods for feature selection and class imbalance learning. *Advances in Knowledge Discovery and Data Mining2013, Springer*. p. 544-555.

Chujai, P., Chomboon, K., Chaiyakhan, K., Kerdprasop, K., Kerdprasop, N. (2017) A Cluster Based Classification of Imbalanced Data with Overlapping Regions Between Classes, *Proceedings of the International Multi Conference of Engineers and Computer Scientists 2017 Vol I, IMECS 2017, March 15 - 17, 2017, Hong Kong.*

Gandhi R (2018) Support Vector Machine—Introduction to Machine Learning Algorithms, Retrieved from https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

Zheng, Z (2015) Oversampling Method for Imbalanced Classification, *Computing and Informatics*, *Vol. 34, 2015*