1 Introduction

With the noticeable progress in big data and machine learning technology, booming mobile internet industry and the dramatic increase in abundance of data in recent years, the machine learning applications in business have received much attention and financing from industry, academy and public. One of these practices in marketing is customer repurchase prediction. An effective and efficient prediction of churn customers rewards the store, not only financially by switching potential losses to income, but also by playing a key strategic role in competition: firms can save financing since the cost of existing customers retention is usually lower than the cost of acquiring new ones. Keeping customers loyal is important to help the company survive in highly competitive industries like online retail of standardized commodities, where customers are pretty price-sensitive and have low switching costs. Moreover, it is as well a hot topic in academic research: it covers a variety of research questions including factors influencing customers' repurchase behaviour, cost-sensitive learning, model performance evaluation and sparse matrix etc.

However, opportunity always comes with challenges: there are several decisive factors disturbing the machine learning practices in customer repurchase prediction, for instance, noise in the dataset, highly skewed class distributions, non-uniform misclassification costs and low-level features. Consequently, the empirical research in this paper is conducted to implement and evaluate different approaches to address such difficulties. From this perspective, the purpose of this paper is to analyse and evaluate how machine learning approaches applied in customer repurchase prediction can influence prediction performance. This will be tested in a given dataset representing the business scenario. Nonetheless, the generalization of the empirical research is debatable, since only one dataset was used.

The paper is organized as follows: the first section provides a literature review. A special emphasis is set on class imbalance and cost-sensitive learning, including various approaches based on resampling, model performance measures and cost sensitive classifiers. Considering the fact that no standard best technique has yet been established, different methods are applied and discussed in the main part of this paper. The conclusion summarizes the main findings and contents of this paper.

2 Literature Review

2.1 Customer Retention Study

Online customer retention is an important issue in marketing and thus very important for online businesses like e-commerce stores. Since online services have very low entry barriers, it is very easy for customers to switch to competing businesses. This makes the knowledge of factors leading to long lasting customer retention very crucial to e-commerce site owners as it can be used to secure competitive advantages.[i] Additionally, the costs of acquiring new customers greatly outweigh the costs of keeping existing ones, making customer retention more financially beneficial than customer acquisition.[ii] Studies have shown that improving customer retention by 5% can lead to profit increases ranging from 25% to as high as 75%[iii]. Thus, understanding the factors leading to customer repurchase that help to develop an effective customer-retention strategy becomes an active research field of great importance[iv]. According to the research achievements, two major factors influencing customer repurchase are customer perceived value and customer satisfaction. Between the two, customer perceived value has a stronger impact on making customers want to return to the same store. Customer perceived value describes how high the customer sees the utility of a service or product after taking into account the benefits received and the sacrifices made to obtain it. Benefits include the level of service-quality perceived as well as psychological benefits. Sacrifices can be monetary or non-monetary like time, risk and convenience sacrifices. Customer satisfaction is often defined as the discrepancy between a customer's expectations and perceptions in marketing literature.[v] The likelihood of return is higher for satisfied customers, making customer satisfaction an important factor for achieving higher retention rates, positive word of mouth and increased profits[vi]. Other factors that have a positive impact on repurchase behaviour are customer service, perceived ease of use and the image of the website. Perceived risks of using a website is an example for a detrimental factor for repurchase likelihood.[vii] Understanding how these factors influence possible customer repurchase can be helpful to construct new predictors for the data mining process.

Data mining and machine learning are used to identify potential non-returning customers and target them directly with promotion efforts. Ling and Li argue that this is more effective than mass marketing, since the promotion costs can be decreased while increasing the response rate. The goal of these efforts is to improve the return on investment of marketing campaigns or more generally to increase net profits. The nature of the customer repurchase setting typically generates certain machine learning problems: The datasets are often highly imbalanced, meaning one class is not represented nearly as much as the other class (sometimes only 1%). In this case, classifiers might simply classify every example into the majority class, thus achieving an accuracy of 99% by completely ignoring the minority class. Also different misclassification errors usually generate different amount of costs and classifiers have to be adjusted to account for this.[viii] These issues are known as the class imbalance problem and cost-sensitivity in machine learning.

Predictor variables that are commonly used in this machine learning setting tend to fall into the categories of recency, frequency and monetary (RFM).[ix] For example, monetary predictor variables can have different forms like average or highest amount paid per customer. However, it is not ideal to focus on RFM variables only. Other important predictors can be present in the dataset. Common examples are complaints and order returns. It is common practice to construct new features out of existing ones based on domain knowledge.[x]

2.2 Machine Learning with class imbalance and cost-sensitivity

In a lot of real world binary classification settings, there is a significant difference in the frequency of the two classes, known as the class imbalance problem. Furthermore, the minority class is often the more crucial class and usually has higher misclassification costs. Thus, if standard classification algorithms are used, they do not take different class frequencies and different misclassification costs into account. They will be biased towards the majority class and will be more likely to misclassify instances from the minority class, leading to greater costs. Generally speaking, there are two ways to address these problems: They can be addressed on the data level in the pre-processing step or on the algorithmic level through creating new classifiers or modifying existing ones. Additionally, it is important to adjust the evaluation metrics used, since traditional metrics like accuracy will not take class imbalance and misclassification costs into account.[xi] In this literature review, different approaches to handling class imbalance and different misclassification costs will be examined. The methods are structured into data pre-processing methods and algorithmic approaches, namely cost-sensitive learning. Lastly, common evaluation metrics for imbalanced data and cost-sensitivity will be researched.

2.2.1 Resampling

In the pre-processing step, different techniques can be used to handle class imbalance. One of these techniques is called resampling and is used to rebalance the dataset towards a more favorable class ratio. Two basic approaches to this are under-sampling and over-sampling. To evaluate which one is more effective, Drummond and Holte have compared the performances of both techniques using the decision tree learner C4.5. Under-sampling is used to decrease the frequency of the majority class and over-sampling is used to increase the frequency of the minority class in the training set. They find that under-sampling produces better results than over-sampling.[xii] Chawla et al. however present an alternative over-sampling method called Synthetic Minority Over-sampling Technique (SMOTE). Using SMOTE, the minority class is increased by creating additional synthetic examples. An additional synthetic example is created from an existing example through adjusting the features of the existing example towards one of it's same class nearest neighbours. SMOTE outperformed other methods in 44 out of 48 experiments conducted.[xiii] Wang et al however argue that the SMOTE method is flawed and they suggest using a more sophisticated version called threshold SMOTE (TSMOTE) instead. They argue that since traditional SMOTE uses the distance between two positive examples to generate synthetic examples, this distance is very large due to the sparse distribution of positive examples. TSMOTE uses the overall nearest neighbour of a positive example, which could be other positive or negative examples alike. This reduces the distance to the nearest neighbour, thus improving the quality of the synthetic examples. Their results show that TSMOTE consistently outperforms standard SMOTE and other Boosting and Bagging algorithms tested.[xiv]

2.2.2 Feature Selection

Additionally, the feature selection process can be adapted towards taking class imbalances into account. It works by improving the performance and efficiency of classifiers. If standard feature selection is used, the selected features can be biased to predict the majority class example.[xv] Tiwari proposes using a modified version of the feature selection algorithm RELIEFF. The modified RELIEFF algorithm puts higher weights on features while it is processing minority classes, which leads to higher weights on features that are stronger connected to the minority class in order to counter the class-imbalance. He finds that this approach works better for imbalanced datasets than using the traditional RELIEFF algorithm.[xvi] Yang et al. present an ensemble-based wrapper approach to perform feature selection from highly imbalanced datasets. Through sampling, various balanced datasets are created from the original imbalanced dataset. Then feature subsets are evaluated through an ensemble of base classifiers that have been trained on a balanced dataset. This approach combines ensemble feature selection and multiple sampling to create better results. They find that features selected by this approach are significantly better than features selected by traditional wrappers.[xvii] Even though there has been some research on feature selection to tackle class imbalance, it is still relatively underexplored.[xviii] Thus it is subject to change as research in this area increases.

2.2.3 Cost-Sensitive Learning

Cost-sensitive learning can be applied to take non-uniform misclassification costs into account. This can be done regardless of whether resampling or feature selection for class imbalance has already been used or not. Ling and Sheng give a structured overview of cost-sensitive learning. According to them, machine learning algorithms can be either cost-sensitive or cost-insensitive, their difference being in the way they treat misclassifications. The goal of normal learners is to achieve a high accuracy in predicting the right class, without taking into account the different costs misclassifications might have. On the contrary, cost-sensitive learning methods take the misclassification costs into account and have the goal to minimize the total costs. There are two main categories in cost-sensitive learning. The first approach is to develop classifiers that are cost-sensitive in themselves, which is called the direct method. In this case, examples should be classified in such a way that the expected cost is at it's minimum. Classifying an example *x* into class *i* has the expected cost *R(i|x)*

$$R(i \mid x) = \sum_{j} P(j \mid x) C(i, j),$$

with *P(j|x)* being the estimated probability that an example is classified into class *j*. In this approach, the classifier will only classify an example into the positive class, if the expected cost is smaller than classifying that example into the negative class. On the other hand, it is possible to design wrappers, which convert cost-insensitive classifiers into cost-sensitive ones. This approach contains two techniques, thresholding and weighting. The thresholding approach classifies the output of cost-insensitive classifiers into either class according to a threshold. In most cases the threshold is calculated with the following formula:

$$p^{*} = \frac{C(1,0)}{C(1,0) + C(0,1)} = \frac{FP}{FP + FN}.$$

C(1,0) = FP = Misclassification cost of false positive and *C(0,1) = FN = Misclassification cost of false negative*. An example is then classified as positive, if its estimated probability is higher than the threshold.[xix] Weighting on the other hand is a pre-processing step. A normalized weight is assigned to each example according to the misclassification costs. Examples of the class that has a higher misclassification cost are assigned proportionally higher weights than examples of the class that has a lower misclassification cost.[xx]

Since there are a number of ways to address class imbalance and cost-sensitivity, there have been discussions as to which methods work the best. Weiss et al. compare under-sampling and over-sampling with cost-sensitive learning. They conclude that there is no clear winner as to which method generates the best results, since this is very dependent on the dataset.[xxi] In a study by Thai-Nghe et al., a method is presented that combines sampling with cost-sensitive learning, instead of using them separately. They test various sampling techniques with cost-sensitive learning using support vector machines and find this approach to reduce misclassification costs in most cases.[xxii] Burez & Poel find that under-sampling improved accuracy in a churn prediction setting and that a weighted random forest significantly outperformed a cost-insensitive one. Boosting was found to be a very robust classifier, however never outperforming the other techniques.[xxiii] Coussements evaluates various cost-sensitive learning techniques in a customer retention setting and finds that weighting, thresholding and metacost perform the best, depending on which dataset is used.[xxiv] Standard Support Vector Machines can be modified to be applicable to class imbalance settings, through adjusting their decision boundary to remove the bias towards the majority class.[xxv]
Sampling can also be used to rebalance the data set to account for non-uniform misclassification costs. In this case it is a form of cost-sensitive learning applied in the pre-processing step. Positive and negative examples in the training data are usually adjusted to the ratio of misclassification costs.

2.2.4 Evaluation Metrics

The use of evaluation metrics to measure model performance is also influenced by cost-sensitivity and class imbalance. Since the standard accuracy measure places more weight on the majority class, it is not suited for the given setting. Hence, more applicable evaluation metrics have been used like Receiver Operating Characteristic (ROC) analysis and F-value. F-value is a combination of recall and precision and is high when both recall and precision are high. ROC analysis is perhaps the most commonly used evaluation metric in this case. ROC curves plot the true positive rate on the y-axis and the false positive rate on the x-axis.[xxvi] However, Davis and Goadrich argue that Precision-Recall (PR) curves work better to evaluate a predictor's performance on highly imbalanced data than ROC curves. PR curves can show weaknesses in algorithms that seem to be close to optimal in ROC curves. They plot the true positive rate on the x-axis and the fraction of true positive over all examples that were predicted as positives on the y-axis. In this way, the PR curves capture the class imbalance better.[xxvii] Drummond and Holte introduce cost curves, another graphical measure that is better suited for class imbalance and misclassification costs than ROC curves. These cost curves are designed to measure expected costs, instead of accuracy. They show the expected cost of a classifier on the y-axis over misclassification costs and class distribution, which are summarized in a single number on the x-axis.[xxviii]

In conclusion, one can say that literature provides no clear best method to solve these issues in a data mining application. It therefore makes sense to evaluate a number of methods for one specific dataset to identify the methods that generate the best results.

3 Empirical Research

To deal with the challenges mentioned in the introduction, a variety of strategies are implemented and evaluated. These include the appropriate experiment setting to make scientific and interpretable decisions, the comparison of performance metrics as well as resampling and cost sensitive classifiers. The default revenue maximization problem is converted to a classical cost minimization problem so, cost minimization classifiers and performance measures can be used. Lastly, the most effort is placed on data cleaning and feature engineering, especially feature construction, so as to model important factors that influence customer repurchase.

3.1 Data Preparation

3.1.1 Missing Value

As shown in Table 1 in the Appendix, there are five variables in the dataset containing missing values when na, NA, empty content and space in a cell are considered as missing data points. The distribution of missing values in the known and unknown dataset is identical. Most of the five columns are categorical variables except for the weight of the order. The overall strategy of missing value treatment is to reserve the information contained in missing data points as much as possible and imputation by using information from other columns.

3.1.1.1 Na_count

To make use of the information generated by missing values, “na_count” is created to count the number of missing values for each row. Since only 5 columns in the dataset contain missing values, the range of this new variable is consequently [0, 5]. By plotting this novel variable against order number (see Graph 1) and against order date (see Graph 2), it seems that there is no clear pattern between missing value and order number or order date in both known and unknown datasets, which implies that the missing value is evenly distributed across ID and order date. Thus, there seems to be no concern on the inconsistence of data quality.

3.1.1.2 Missing Value Imputation

In addition to summarizing information from missing values over each row, the columns containing missing values are treated in a manner according to their attributes: firstly, missing values in “form of address” are replaced by a new level “Unknown” because the missing type has a much higher weight of evidence (WOE) and information value (IV) than the other three levels (See Table 2), so that it is more reasonable to assign it an individual level.

Secondly, in cases where the account creation date is complete, 96% of account creation date records in both datasets are identical to the order date. Therefore, it is rational to replace missing observations in the account creation date with the order date. However, since there is a 0.3% difference in the repurchase rate between the subset where this variable is missing (18.3%) and that where it is not missing (18.6%), it is possible that this simple replacement may cause bias. To reserve this potential bias, the observations with missing values in account creation date are additionally accounted for in the “nosiy_observation” variable.

Thirdly, the incomplete weight of order is imputed by the average weight of each item in a specific product category conditional on the cost of shipping. More specifically, the information of product count, cost shipping and delivery are used together to group the orders into different product types. Coarse-grained classification contains three levels: whether the products are only physical goods (for instance book, imported goods or other goods), only digital goods (e-book and download audio book) or mixed products. Fine-grained product level originally contains 17 levels including book, schoolbook, audio book download etc. (See Feature Construction “basket_diversity_coarse_grained”, “basket_diversity_fine_grained). With the information of product category, the average weight of a specific category is calculated and used to impute the missing points conditional on the shipping cost situation. For example, missing values will be imputed by 0 if the products in the order are only digital goods (e-book or audio download or both). In case the order has books with shipping costs, the missing value will be imputed by the average weight of books in the order that have shipping costs. The weight of the products receives relatively more attention and careful treatment because it has high predictive power according to the embedded variable selection result (See Graph 7). Nonetheless there is still noise in the weight variable. For example, around 10% of observations have a weight below 10 grams and higher than 0 grams, which is counterintuitive in terms of product weight, so they are accounted for in the noisy observation variable.

Fourthly, missing values in postcode delivery are imputed by "Unknown", because the missing rate is very high at 96% and a lack of additional information makes a convincing imputation difficult. Considering the fact that this variable ranks among one of the variables with the least predicting power, the strategy is to simply put a label on them to avoid creating bias. Lastly, the advertising code suffers from a similar situation as postcode delivery, so it is treated in the same manner.

3.1.2 Outlier and Noise

Outliers and noise in this empirical research mislead the classifier and increase the tendency of overfitting, so it is of importance to deal with them. Starting from the four given date variables, except for order date the other three all contain a variety of noise and outliers. For instance, there are 16.8% and 16.3% of observations respectively in the known and unknown datasets that have actual delivery dates with records of "0000/00/00". In the estimated delivery date there are 73 observations with the year of 2010 and 14 observations with the year of 4746, which is inconsistent with the order date year interval of 2013-2014. To detect them completely, several columns are created: the year, month and day of the four date variables are extracted from the original columns and the difference from the estimated delivery date and the order date is calculated as "days_est_deliver". In addition, the difference between actual delivery date and the order date is marked as "days_deliver_actual", the noise and outliers are detected by the mutual comparison of above columns and original observations are modified by the following rule: "0000/00/00" in actual delivery date is because such orders contain basically only digital products, therefore the missing actual delivery date is substituted by the corresponding order date. In the case of orders expected to be delivered in 2010, the corresponding year is replaced by 2014, which is the mode value of year, and those orders which are estimated to be shipped in the year of 4746 will be replaced by their order date.

In addition to the four date variables, noise and outliers exist also in the product information. Firstly, the definition of "item_count" is not clear: the count of all items in the transactions is not equal to any combination of the sum of all reasonable ingredients in the given table. After several trials, the calculation of item count with the highest rate of similarity (95%) is modified.

The business concept of item count is still unclear. This column seems to be the original product records, which is the sum of canceled items plus the product counts remaining in the order. A piece of new information, which is the sum of all product records left in the order, could be extracted from it. "all_product_count" is consequently defined as the sum of all products left in the order after the cancelled items are dropped from the column of item count. However, the number reveals a problem that 3.1% of all observations have a product sum of zero, which means there is no product information in the records. The difference of missing product records is almost even in known (3.1%) and unknown dataset (3.0%), so it releases the concern on these abnormal points and such observations are labelled as noisy observations. Considering the noise and the outlier points, it justifies the action of recording such inconsistent records: it makes sense both in business and in data mining that it is not unusual for a customer that he is just a random client of an online store and would rarely come back. Perhaps he took the deal just because of an urgent temporary need of a specific product or an enthusiastic recommendation from a close friend or he just wanted to take advantage of a free coupon. In such a case, he or she may have an incomplete or even inconsistent personal information: an arbitrary or even forged gender, title and incompatible identity and email address. In a word, an incomplete or disaccord personal information may reveal the customer's relatively weak intention to stay in the business.

Recording such information helps the classifiers capture the real distribution of a dataset as well: a thorough investigation that captures the cumulative missing and illogical records by order would deepen the insight of the spatial distribution of the dataset. If there exists a pattern between problematic records and order ID in both training set and test set for example, this pattern would be recognized by the classifier thus improving the performance. However, such guess needs to be examined by experiments to decide the appropriate treatment of noise and outlier points and the existence of the pattern discussed above.

3.1.3 Feature Construction

3.1.3.1 Customer Perceived Value and Customer Satisfaction Modeling

To learn the effects of customer perceived value and customer satisfaction on customer repurchase and their ability in boosting the model performance, most of the time and effort in empirical research was spent on business-oriented feature construction. Thanks to the information regarding item cancelling, product remitting, time difference of shipment and the change of order frequency by day by month of the store etc. more than 43 new variables that capture the information of customer intention (e.g. complete personal information), customer dissatisfaction (e.g. in shipment or in product) as well as basket analysis (e.g. shopping only book or mixed of e-book and film) are created (See Table 2&3). The newly constructed variables have rather high Information Value (See Graph 4) and many of them remain in the final model after several rounds of feature selection (See Graph 7), so the feature construction has pretty high influence in the improvement of prediction.

3.1.3.2 Rankings of Nominal Variable

As area under the curve (AUC) is chosen as the major performance metric (See 3.4.1), it is reasonable to take the rankings of continuous variables into account because AUC, when using normalized units, is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming 'positive' ranks higher than 'negative'). [xxix] The ranking variable is not only suitable to the performance metric, but also robust to outliers because different from trimming the data, ranking variables are not vulnerable to change when different trimming criteria are employed [xxx] and it does not lose information from manipulating data points. By adding ranking features, the model will be more stable and the risk of overfitting is reduced, which is of great importance in this empirical research where the overfitting probability is pretty high (See 3.1.2). In particular, there are 41 continuous variables and therefore 41 ranking variables are created. For instance, a column with the value of 5, 10, 100 and 1 would have a corresponding rank variable with values of 2, 3, 4 and 1. When it comes to the effects, the Information Value (See Graph 4) justifies the prediction power of these ranking variables.

3.1.3.3 Interaction Terms

Except for XGBoost and other algorithms that can capture nonlinear relationships pretty well, in this empirical research some linear models like Logistic Regression (LR) and Support Vector Machines (SVM) are also implemented. Under the circumstance that an engineered feature will make it easier for the model to discover nonlinear relationships in the data, the columns to generate x*y interaction items are derived from first and second round feature selection. The first round selects variables with top 20 highest chi square or information gain and the second round selection is XGBoost feature importance (See Graph 7). 14 categorical variables selected from these two procedures are used to generate interaction terms. The reason for conducting interactions only for relatively important variables rather than for all is to avoid the noise from unimportant variables.

However, the effect of interaction terms seems not so favourable, the AUC of XGBoost decreased from 0.68 to 0.65, the average misclassification cost of the logistic regression (LR) model increases from 1.66 to 1.88 and the F1 score decreases dramatically from 0.35 to 0.04, when the interaction terms are added in. Therefore, it is not wise to throw all interactions into the model so most of them are screened out in the procedure of feature selection.

Except for the machine-dominated detection of interaction terms, the interactions are also captured through a deep Exploratory Data Analysis (EDA): for instance, the referrer, link and usage of coupon appear only in certain web styles, indicating that these three variables are inter-related. This is an important finding. So, a new variable is created to model this customer behaviour in online shopping. Another example of interaction is between newsletter and referrer: in case a customer has searched for something yet not paid, commodities may remain in the basket or be steadily recommended to him. Such information could be delivered to the client through newsletter subscription, so the interaction of newsletter and non-instant purchaser is built and examined. However, the Chi-Square test rejects the significance of their relationship.

3.2 Feature Processing

3.2.1 Discretization of Continuous Variable (Binning)

In this empirical research, equal length, equal frequency and optimal binning are all carefully conducted and compared. Considering the short time window to finish and relatively large workload, the final discretization strategy is to implement automated binning and equal frequency binning when the automated binning is not plausible.

3.2.2 Optimal Binning of Categorical Variable

Optimal binning is a method of pre-processing categorical predictors with many levels in a predictive data mining project. The existence of categorical variables with too many levels will make the analytical methods inefficient. This empirical research confronts this problem as well: there are around 10 levels in email domain and nearly 100 levels in postcode. The solution is group combining, or in another words, optimal binning of the categorical variables. By doing so, the combination could maximize the relationship of the original categorical variable to the target variable.

3.3 Feature Selection

In the feature selection step, filter methods, wrapper methods and embedded methods are evaluated regarding their performance for feature selection.

3.3.1 Filter Methods

3.3.1.1 Information Value

The information value was calculated for all variables, the results for the top 20 variables are shown in Graph 4. The selection rule is dropping all columns with very low IV.

3.3.1.2 Pearson Correlation

By calculating the Pearson correlation across the numerical variables and response variable (in this case it is treated as a nominal variable as well), the threshold for a variable to be dropped from the model is its correlation value to the target variable. It stays in the range of
[-0.03, 0.02], which means a relatively small correlation with the target variable will keep variables from the model. Further, the mutual correlations across independent variables are also taken into consideration. The threshold is 0.75, which means that in a group of variables highly correlated with other variables, only the predictor with the highest correlation value to the target variable will stay in the model so as to avoid the problem of multicollinearity. However, this rule has exceptions. For instance, when the effect of the rank version and the original version of a variable is unclear, it will be left in the feature set and further tested in wrapper methods or model tuning to check the performance in the bias variance trade-off.

3.3.1.3 Information Gain

Graph 5 shows the variables with top information gain (IG) scores among available variables. To determine the optimal percentage value for feature selection based on IG, two steps are taken: the first one is setting up a 10-fold cross validation (CV) to see if a domain knowledge based choosing of the optimal feature selection could increase the performance steadily. However, the domain knowledge based threshold does not necessarily bring out the optimal outcome. To choose a threshold of optimal percentage of top predictors more convincingly, the filter threshold determination is wrapped into different binary classification classifiers, the average performance based on a 3-fold cross validation will justify the optimal percentage.

In step one, the domain knowledge sets the optimal rate at 0.7, which means only the top 70% of predictors will be left in the model. According to the result of the cost sensitive, Logistic Regression with a threshold for positive instances (customer will repurchase) at 0.231, there is a favourable decreasing in average cost of misclassification from 1.68 to 1.64 as well as the AUC increase from 0.65 in the baseline model to 0.66, which indicates that there are more positive cases successfully captured by the model through filtering the ineffective features. However, the results from cost sensitive gradient boosting machine (GBM) and cost sensitive SVM show a decrease in performance, both models indicating the average cost of misclassification increases to 1.89 from 1.67 and 1.66 respectively. Furthermore, the AUC decreases to 0.61. The positive-negative-instance rebalanced Random Forest, AdaBoost and XGBoost are not considered in this case due to the time and computation resource limitation or the complexity in meta-parameter tuning.

By selecting the top variables based on their information gain, there is a favourable increase in model performance: The remaining top 75% of all variables in the Logistic Regression improve the average cost of misclassification from 1.69 to 1.64, however a random selection of the percentage threshold does not do so good.

3.3.1.4 Chi Square

Chi Square was used to select categorical features based on their correlation, the results being depicted in Graph 6. The results of Chi Square supplement those of IG, meaning the “unimportant” features in IG that turn out to have certain predictive power will be re-added into the model.

3.3.2 Wrapper Methods

Wrapper methods use the performance of a classifier to assess the usefulness of a feature set. There are three methods tried out: firstly, exhaustive search on cost sensitive LR. The performance is assessed by the holdout estimate of the concordance index. It indicates a remaining performance of 1.64 of average cost and 0.36 of F1 Score after removing the features through exhaustive search. Secondly, sequential forward search is implemented both by cost sensitive LR and cost sensitive Boosting, however, none of their performances improved. According to 3-fold CV the average misclassification cost increases dramatically from 1.64 to 1.71 and the F1 Score dropped to 0.327 from 0.363 in LR. By Boosting, the performance is the same as LR, the average cost being 1.71 and the F1 Score is 0.33. Lastly, the sequential backward search is conducted on a cost sensitive linear boosting classifier and it still performs worse than exhaustive searching on LR. Consequently, the wrapper feature selection will be based on the result of exhaustive searching LR.

3.3.3 Embedded Methods

XGBoost feature importance is the most important benchmark for feature selection. The final result is also based on the XGBoost feature importance result. The dataset has 87 variables left after the filter and wrapper feature selection. The 87 features are separated into four parts: the original and newly constructed categorical variables that capture the information from continuous variables in a binning manner, the ranking variables of continuous variables, the derived counting and the original continuous variables. According to the 5-fold CV the results show that the best feature combination is categorical variable + binning of continuous variables + ranking variables + the derived counting of frequency of categorical variables (0.66 AUC), other variables are deleted. However, the other models all have very similar AUC results that are stable around 0.655-0.659, therefore the feature selection is actually not helping much in the performance improvement. The final feature set has 75 variables but the prediction abilities of them are considerably different. A 5-fold CV that picks up only the top 20 variables in terms of prediction power however has a very low AUC in both train and test datasets which is an indicator of underfitting. In consequence, the feature sets are not reduced even though some of the variables are not performing so well.

Other methods, for instance principal component analysis (PCA), are not considered in the experiment because they ignore the information on target variable and focus only on the input matrix so they do not usually have a profound performance for feature selection in supervised machine learning (See Graph 7).

3.4 Class Imbalance and Cost-Sensitive Learning

Noticeably, the distribution of the target variable in the empirical research is skewed. As shown in Graph 8, the majority class (customers that do not return in the given time window) heavily outnumbers the minority class (customer repurchase). More specifically, instances labelled with 0 account for only around 18.4% in the training dataset whereas the proportion of majority is almost four times larger.

Regarding the choosing of mentioned approaches, the paper of C. Elkan 2014[xxxi] argued that changing the balance of negative and positive training examples has little effect on learned classifiers. Accordingly, the recommended way of using one of those methods is to learn a classifier from the training set as given and then to use the equations to minimize the overall cost of misclassification and calculate the optimal threshold. Based on this argument, all three strategies are implemented and compared based on the experiment outcome in order to obtain the optimal strategies or strategy combination. According to the cross validation results, the algorithmic methods, more specifically the optimal threshold of positive instance and rebalancing of positive and negative instances, are the final solutions and have the best performance overall.

3.4.1 Performance Measure

Accuracy is not a plausible performance metric for skewed datasets because it is not sufficient to calculate only overall accuracy based on the percentage of observations correctly classified without considering the misclassification costs. Accordingly, a discreet selection of performance metrics that could measure the different cost of False Negative and False Positive is suggested. The tool kits of performance measures provide a variety of potential choices including confusion matrix, AUC, precision recall curves (PRC), expected misclassification cost and G-Mean etc.

The primary measure in a marketing campaign, when the misclassification cost is known, could be the average misclassification cost: Based on the cost benefit matrix given in the assignment (Table 7), the cost information is converted to a standard cost matrix and fed to the classifiers, so normal classifiers are updated to cost-sensitive learners. Thus the average cost measures and optimizes the performance directly and so becomes the first choice as a performance measure. As argued in the literature review considering the unbalanced dataset and different misclassification costs, the classifiers will also be measured by AUC and PRC, conditionally measured by Missing Rate and Fall-out, when such information is plausible. The implementation of a variety of performance metrics is done to overcome the single performance bias and enrich the perspectives of evaluating the approaches and models.

3.4.2 Resampling Strategies

Based on 3-fold CV of a LR model fitting the original dataset (see table 8), it shows that there is a considerable decrease in Expected Cost and increase in F1 Score (AUC is unavailable in this case), when oversampling and undersampling are conducted to balance the dataset, even though the difference across theoretical optimal and empirical optimal (tuned by an extra 3-fold CV) weighting rate in both resampling strategies are pretty small.

3.4.3 Cost Sensitive Learning Strategies

3.4.3.1 Optimal Threshold

According to the function of theoretical cost-minimal classification cut-off for classifiers in Bayes Risk Minimization, the theoretical threshold for a classifier to classify an instance as positive is $t^* = 0.231$. After replacing the default threshold for positive labels (50%) by the theoretical optimal threshold (23.1%), the 3-fold CV results (Table 9) show a dramatic increase in performance by modifying the cut-off prediction. Especially the empirical optimal cut-off, which is slightly lower than the theoretical optimal cut-off, performs even better in terms of Recall Precision trade-off in that it could distinguish more positive cases at lower costs of misclassifying negative instances.

3.4.3.2 Rebalancing (Weighting)

Rebalancing is the cost sensitive learning approach in the algorithmic level. The calculation of theoretical optimal weight is similar to the cut-off, which equals to 3.33. The theoretical optimal may not be the empirical best one, so a grid search of which possible ratio performs the best in this dataset is done through a 3-fold CV. As shown in Table 10, the comparison across different weights justifies the effect of weighting in model performance even though there is no considerable difference from optimal and empirical threshold.

3.4.3.3 XGBoost

According to the official document of the machine learning algorithm XGBoost, there are two possibilities to incorporate misclassification costs into XGBoost, the first strategy is to adjust "scale_pos_weight" and use AUC as performance metric. In this case, the binary classification is in principal a ranking problem rather than calculating the probability. The second possibility is setting the parameter of "max_delta_step", which is an argument allowing for each tree's weight estimation, to a finite number (the default value is zero) so as to help convergence. The idea behind this is that a default value at 0 means that there is no constraint on the maximum delta step for each tree's weight estimation, setting it to a positive value will make the update step more conservative to overcome the problem of class-imbalance. Both strategies are implemented in the empirical work but the first one is eventually taken due to the better CV result.

It is necessary to note that the tactics to tune meta parameters of XGBoost to achieve the optimal result is not a grid search by CV algorithm because it consumes too much time and computation resources. To tune a combination of parameters could have more than 300 possibilities. Moreover, the common package "Caret" to train meta parameters is not available for XGBoost to train the critical parameter "scale_pos_weight", thus the grid search algorithm in this case is replaced by a simplified greedy strategy: 25% of the overall dataset with known labels is partitioned for validation, the rest is used to train meta parameters by the order of "max.depth", since it is relatively invariant to other parameters, then "subsample", "min_child_weight", "colsample_bytree" etc. Each parameter will start from a relatively widely applied value and will then be reduced by a small step. For example, max.depth starts from 10 and is decreased stepwise until AUC reaches its maximum. If AUC decreases however, max depth will instead be increased until AUC reaches its maximum. The other meta parameters tuning in XGBoost follow the same rule. Compared to grid search, this strategy could save time and computational resources dramatically and reach almost good parameters in just a few steps.

3.5 Model Selection

Through the whole process of data preparation, feature selection and prediction etc. different cost sensitive classifiers have been conducted and compared, however there is basically not much freedom in model selection. This is because overall the results of prediction are much lower than expected. For instance, in a 3-fold CV environment, XGBoost has the best performance in terms of AUC (average cost is not applicable to XGBoost due to the inability of writing custom loss function). By doing more than 10,000 iterations and tuning the best meta parameters, it is noticed that AUC in the training set could be higher than 0.8 but in the test set, it is not even higher than 0.68, which is clearly an indication of strong overfitting. Consequently, further tunings are done on the parameters of L1 and L2 regularization, but the improvement in performance is still minor. Compared to XGBoost, cost sensitive Logistic Regression, cost sensitive Boosting, cost sensitive SVM, cost sensitive ANN as well as cost sensitive LR with L2 regularization all have lower performance than XGBoost.

When it comes to heterogeneous ensemble learning, even though they are usually effective in boosting the performance in the very last step, they are eventually not implemented in the final model. This is because firstly the overall prediction performances across all classifiers is not high enough to do an ensemble learning, since aggregating unqualified models will not help much in the overall performance. Another reason is their performances in predicting specific kinds of instances are not clear, which is different from trees in bagging models who are performing very well in specific variables or instances though in overall they are weak, therefore there is no sufficient information to build an effective bagging of individual models.

Since the inability to do heterogeneous ensemble learning, the final prediction is only from the model with the highest CV performance in training (XGBoost). In addition to the relatively great performance XGBoost is chosen also because of its high interpretability compared to other “black boxes” like SVM or ANN where the variable importance and non-linear relationships behind are not so straightforward to be presented. By contrast XGBoost is a decision tree based method where the chart of growing procedure of a tree and variable importance rank could clearly facilitate the interpretation of decision making and help with further optimization.

In order to improve performance several approaches could be done. Firstly, more domain experience is in need and more sophisticated approaches to feature construction can be evaluated. Secondly, doing more sophisticated feature processing may help since continuous variables suffering from skewed distributions should be processed more carefully. Thirdly, it would be helpful to write custom objective functions incorporating the cost matrix into each classifier, rather than depending on the black box models, since the way cost-sensitivity is solved by the black box models is not clear to the users. Unfortunately, this is beyond our current abilities, so this step could not be implemented in the given time constraint.

4 Conclusion

In order to maximize the overall prediction revenue, several technologies and skills from marketing, statistics, data mining and machine learning that majorly deal with the problem of overfitting and imbalanced datasets are examined and compared in a scientific manner. By doing so, it is possible to quantitatively evaluate the effects of different approaches to improve the classifier's performance. Through careful interpreting and analysis of experiment results, several conclusions are conducted.

Firstly, consistent with the argument of relatively high pay-off in doing feature construction, especially features that can model the factors influencing the customer repurchase intention and behaviour, it is to be found that the newly extracted features perform considerably well in differentiating the target variables. Secondly, the existence of extreme skewed distributions of critical predictors highly increases the tendency of overfitting, which dramatically weakens the predicting performance, could to some extent be solved by discretization of nominal variables, outliers-insensitive classifiers and regularization. This indicates that the breadth and depth of knowledge and experience in solving overfitting play the key role in such practices. Last but not least, cost-sensitive learning could considerably increase the performance of prediction but the effect is highly subjected to how well the practiser understands the mechanism and principal of the algorithm as well as how good her or his programming ability is, especially the capability of writing custom loss functions. In a word, there is no "state-of-the-art" method that always brings the optimal result but the domain knowledge, profound understanding of algorithm, engineering ability and the desire for learning will help to succeed in such practices.

5 Appendix

Table 1: Missing Value Statistics and Treatment

Variable	Variable Type	Missing Rate (Known)	Missing Rate (Unknown)	Treatment
form_of_address	cardinal	13%	13%	novel level lablled as Unknown
account_creation_date	cardinal	7%	7%	impute by order_date
weight	nominal	8%	8%	impute by average weight of belonging product group
postcode_delivery	cardinal	96%	96%	novel level lablled as Unknown
advertising_code	cardinal	80%	80%	novel level lablled as Unknown

Table 2: Weight of Evidence and Information Value of form_of_address

Bin	WOE	IV
Company	-37.000	0.010
Mr	-4.900	0.001
Mrs	-8.600	0.002
Unknown	49.200	0.037

Table 3: Feature Construction based on Customer Repurchase Study

Variable	Naming	Type	Definition
Customer Inention Modeling			
conversion time	days_first_order_occurs	nominal	order date – account creation date
convert right after registeration	instant_purchase	ordinal	equals to 1 if at the same day converting otherwise 0
count of orders by day	daily_order_frequency	nominal	currence of transactions in each day
count of orders by month	monthly_order_frequency	nominal	currence of transactions in each month
Shipping Dissatisfaction Modeling			
estimated delivery time	days_est_deliver	nominal	estimated delivery date – order date
actual delivery time	days_deliver_actual	nominal	actual delivery date – order date
shipping delay	days_delivery_delay	nominal	actual delivery time – estimated delivery time
Product Dissatisfaction Modeling			
the rate of remitted items	remitted_items_rate	nominal	remitted_items/all_product_count
the level of remitted items	remitted_items_level	ordinal	binning of remitted_items_rate based on domain knowledge
Other Dissatisfaction Modeling			
fully canceled order	order_canceled	ordinal	equals to 1 if there is no product record otherwise 0
rate of products are canceled	canceled_items_rate	nominal	canceled_items/all_product_count
level of cancel rate	canceled_items_level	ordinal	binning of canceled_items_rate based on domain knowledge
cancel more than order size	cancel_more_than_records	ordinal	equals to 1 if canceled items >= order size otherwise 0
Basket Analysis			
count of all products	all_product_count	nominal	sum of 11 product count in each order
order's average price level	avg_goods_value	nominal	goods_value/item_count
price change by date	product_val_diff_by_day	nominal	delta of avg_goods_value sorted by basket category and date
count of digital goods	digital_product_count	nominal	number of ebook and audio download count
count of physical goods	physical_product_count	nominal	number of all physical products count
product type coarse grained	basket_diversity_coarse_grained	ordinal	order contains physical, digital or mixed
product type fine grained	basket_diversity_fine_grained	ordinal	order contains book, shoolbook, audio download etc.
frequency of product	basket_frequency	nominal	frequency of basket_diversity_fine_grained

Table 4: Feature Construction of Derived Information

Variable	Naming	Type	Definition
Missing Value/Outlier Dummy			
missing value count	na_count	nominal	count of missing value in each row
variable captures noise	noisy_observation	ordinal	equals to 1 if counts noise otherwise 0
Date Information			
year of order date	order_date_year	nominal	year of order date
month of order date	order_date_month	nominal	month of order date
day of order date	order_date_day	nominal	day of order date
weekday of order date	order_date_weekday	nominal	weekday of order date
year of account creation date	acc_cre_date_year	nominal	year of account creation date
month of account creation date	acc_cre_date_month	nominal	month of account creation date
day of account creation date	acc_cre_date_day	nominal	day of account creation date
weekday of account creation date	acc_cre_date_weekday	nominal	weekday of account creation date
year of estimated delivery date	de_est_date_year	nominal	year of estimated delivery date
month of estimated delivery date	de_est_date_month	nominal	month of estimated delivery date
day of estimated delivery date	de_est_date_day	nominal	day of estimated delivery date
weekday of estimated delivery date	de_est_date_weekday	nominal	weekday of estimated delivery date
year of actual delivery date	de_act_date_year	nominal	year of actual delivery date
month of actual delivery date	de_act_date_month	nominal	month of actual delivery date
day of actual delivery date	de_act_date_day	nominal	day of actual delivery date
weekday of actual delivery date	de_act_date_weekday	nominal	weekday of actual delivery date
Customer Identity			
german e-mail	email_domain_german	ordinal	if the email address is ".de"
same postcode in delivery and invoice	equal_postcode	ordinal	equals to 1 if same code otherwise 0
grouping ad code by alphabet	advertising_code_group	ordinal	ABCU, four groups

Table 5: Pearson Correlation Coefficient

	return_customer
Rank_remitted_items	0.08
Rank_real_item_in_basket	0.07
Rank_na_count	0.05
Rank_paperback_count	0.05
two_occurance	0.04
digital_product_count	0.04
three_occurance	0.04
physical_product_count	0.04
Rank_days_first_order_occurs	0.04
Rank_digital_product_count	0.03
five_occurance	0.03
four_occurance	0.03
Rank_audiobook_download_count	0.03
Rank_ebook_count	0.02
Rank_weight	0.02
six_occurance	-0.03
Rank_avg_goods_value	-0.06
one_occurance	-0.07

Table 6: Standard Cost Matrix

	Actual Negative	Actual Positive
Predict Negative	0	C(neg, POS) - B(pos, POS)
Predict Positive	C(pos, NEG) - B(neg, NEG)	0

Table 7: Cost Benefit Matrix

	Actual Negative	Actual Positive
Predict Negative	B(neg, NEG)	C(neg, POS)
Predict Positive	C(pos, NEG)	B(pos, POS)

Table 8: Prediction Performance Comparison by Over- and Undersampling

Perofrmance Metric	Default (0.5)	Theoretical Optimal Oversampling Rate (3.5)	Empirical Optimal Oversampling Rate (3.25)	Theoretical Optimal Undersampling Rate (0.286)	Empirical Optimal Undersampling Rate (0.3)
Expected Cost	1.88	1.68	1.68	1.68	1.69
F1 Score	0.02	0.34	0.34	0.34	0.33

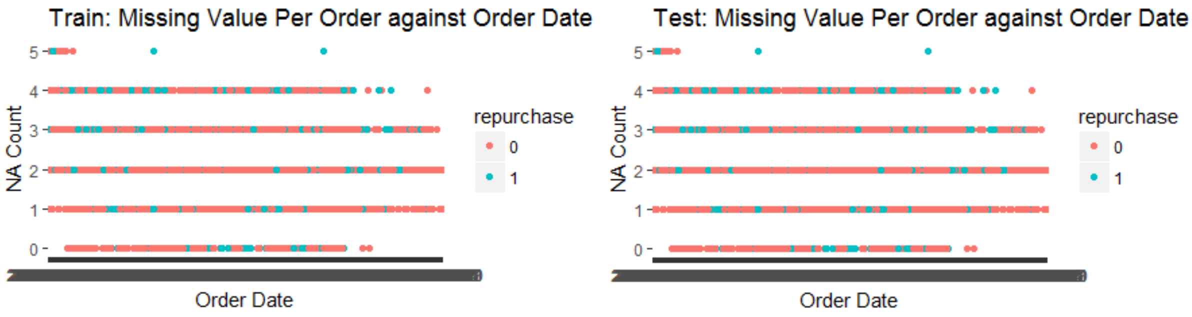
Table 9: Prediction Performance Comparison by changing classification threshold

Perofrmance Metric	Default (0.5)	Theoretical Optimal (0.231)	Empirical Optimal (0.219)
Expected Cost	1.88	1.68	1.69
F1 Score	0.02	0.33	0.34
Miss Rate	0.99	0.63	0.58
Recall	0.00	0.38	0.42
Fall-out	0.00	0.21	0.25

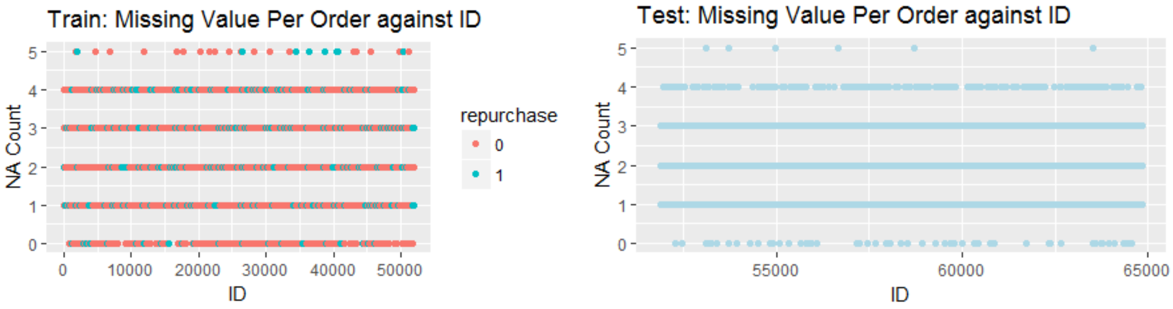
Table 10: Prediction Performance Comparison by weighting class

Perofrmance Metric	Original Weight Weight (1)	Theoretical Optimal Weight (3.33)	Empirical Optimal Weight (3.56)
Expected Cost	1.88	1.68	1.69
F1 Score	0.02	0.33	0.34

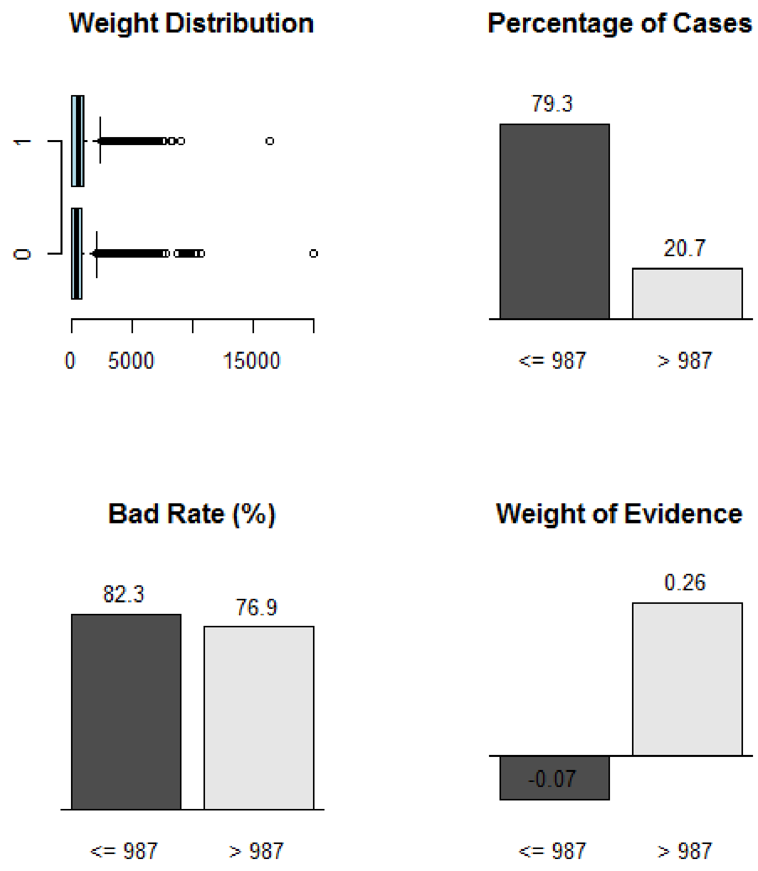
Graph 1: Row-wise missing points against order ID



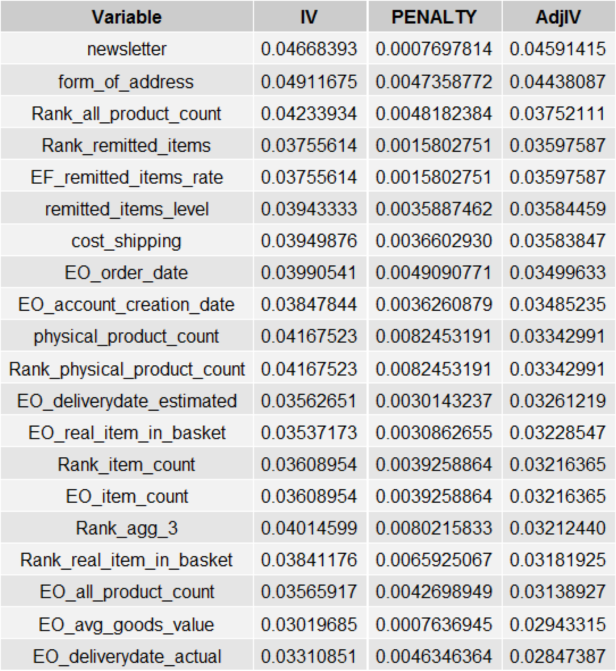
Graph 2: Row-wise missing points against order date



Graph 3: Example of Optimal Binning

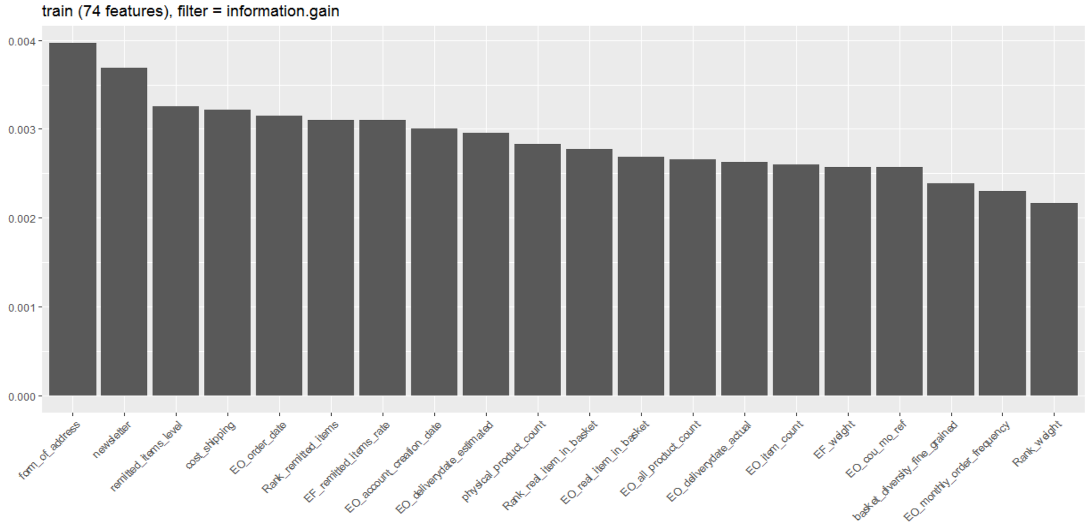


Graph 4: Top 20 columns with relatively high adjusted Information Value

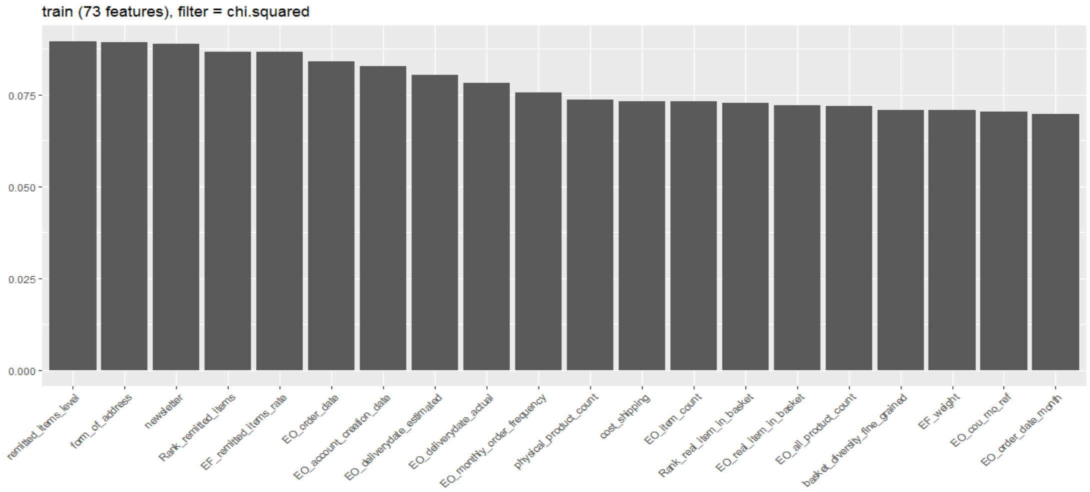


Variable	IV	PENALTY	AdjIV
newsletter	0.04668393	0.0007697814	0.04591415
form_of_address	0.04911675	0.0047358772	0.04438087
Rank_all_product_count	0.04233934	0.0048182384	0.03752111
Rank_remitted_items	0.03755614	0.0015802751	0.03597587
EF_remitted_items_rate	0.03755614	0.0015802751	0.03597587
remitted_items_level	0.03943333	0.0035887462	0.03584459
cost_shipping	0.03949876	0.0036602930	0.03583847
EO_order_date	0.03990541	0.0049090771	0.03499633
EO_account_creation_date	0.03847844	0.0036260879	0.03485235
physical_product_count	0.04167523	0.0082453191	0.03342991
Rank_physical_product_count	0.04167523	0.0082453191	0.03342991
EO_deliverydate_estimated	0.03562651	0.0030143237	0.03261219
EO_real_item_in_basket	0.03537173	0.0030862655	0.03228547
Rank_item_count	0.03608954	0.0039258864	0.03216365
EO_item_count	0.03608954	0.0039258864	0.03216365
Rank_agg_3	0.04014599	0.0080215833	0.03212440
Rank_real_item_in_basket	0.03841176	0.0065925067	0.03181925
EO_all_product_count	0.03565917	0.0042698949	0.03138927
EO_avg_goods_value	0.03019685	0.0007636945	0.02943315
EO_deliverydate_actual	0.03310851	0.0046346364	0.02847387

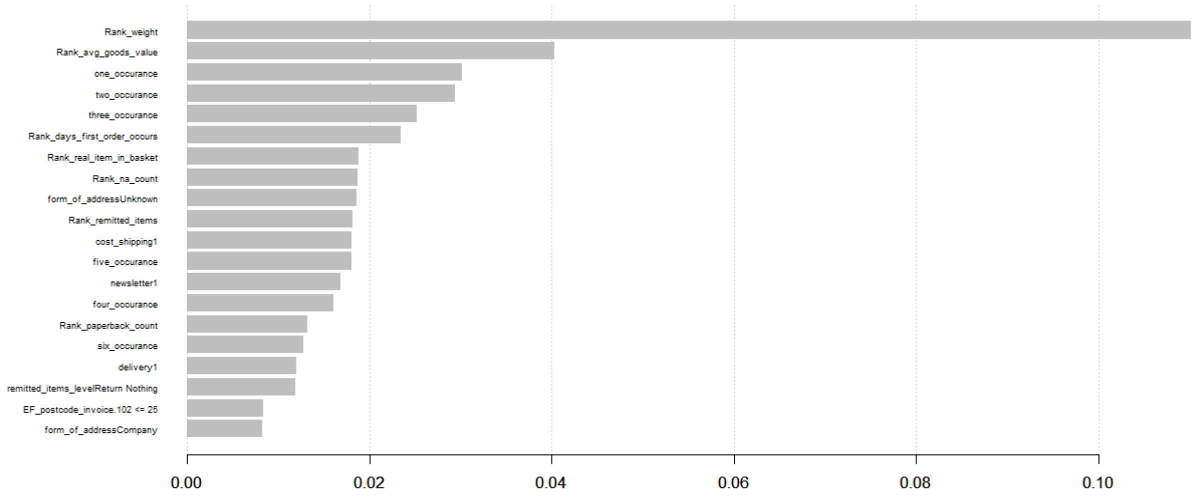
Graph 5: Importance of features based on IG



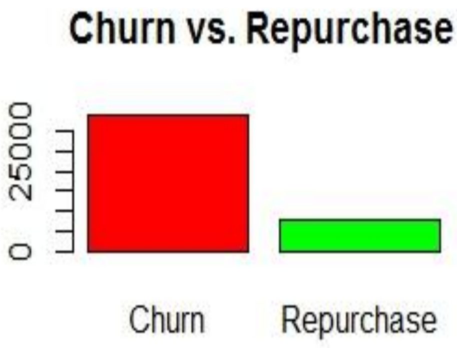
Graph 6: Importance of features based on Chi Square



Graph 7: XGBoost Feature Importance



Graph 8: Unbalanced Dataset



[i] H.Li, J. Hong, „Factors Influencing Consumers' Online Repurchasing Behavior: A Review and Research Agenda", iBusiness, 5, 161-166, 2013

[ii] C. Odindo, J.Devlin, „Customer Satisfaction, Loyalty and Retention in Financial Services", Financial Services Research Forum, n.d.

[iii] Y. Li, „Empirical Study of Influential Factors of Online Customers' Repurchase Intention", iBusiness, 8, 48-60, 2016

[iv] I-K. Chung, M-M. Lee, „A Study of Influencing Factors for Repurchase Intention in Internet Shopping Malls", n.d.

[v] H.Li, J. Hong, „Factors Influencing Consumers' Online Repurchasing Behavior: A Review and Research Agenda", iBusiness, 5, 161-166, 2013

[vi] I. Pappas, A. Pateli, M. Giannakos, V. Chrissikopoulos, „Moderating effects of onlineshopping experience on customersatisfaction and repurchaseintentions", International Journal of Retail & Distribution Management 42(3):187 – 204, 2014

[vii] I-K. Chung, M-M. Lee, „A Study of Influencing Factors for Repurchase Intention in Internet Shopping Malls", n.d.

[viii] C. Ling, C. Li, „Data Mining for Direct Marketing: Problems and Solutions", KDD-98, n.d.

[ix] S. Viaene, B. Baesens, D. Van den Poel, J. Vanthienen, G.Dedene, „Bayesian neural network learning for repeat purchase modeling in direct marketing", European Journal for Operational Research, n.d.

[x] S. Viaene, B. Baesens, T. Van Gestel, J. A. K. Suykens, D. Van den Poel, J. Vanthienen, B. De Moor, G. Dedene, „Knowledge Discovery in a DirectMarketing Case using Least

SquaresSupport Vector Machines“, International Journal of Intelligent Systems, Vol. 16, 1023-1036, 2001

[xi] V. López a, A. Fernández, S. García, V. Palade, F. Herrera, „An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics“, Information Sciences 250, 113–141, 2013

[xii] C. Drummond, R. Holte, „C4.5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling beats Over-Sampling“, n.d.

[xiii] N. Chawla, K. Bowyer, L. Hall, W. Philip Kegelmeyer, „SMOTE: Synthetic Minority Over-sampling Technique“, Journal of Artificial Intelligence Research 16, 321–357, 2002

[xiv] J. Wang, B. Yun, P. Huang, Y-A. Liu, „Applying Threshold SMOTE Algoritwith Attribute Bagging to Imbalanced Datasets, Lecture Notes in Computer Science, vol. 8171, pp. 221-228, 2013

[xv] A. Braytee, W. Liu , P. Kennedy, „A Cost-Sensitive Learning Strategy for Feature Extraction from Imbalanced Data“, Lecture Notes in Computer Science, vol. 9949, pp. 78-86, 2016

[xvi] D. Tiwari, „Handling Class Imbalance Problem Using Feature Selection“, International Journal of Advanced Research in Computer & Science Technology, Vol. 2, Issue 2, Ver. 3, 2014

[xvii] P. Yang, W. Liu, B. Zhou, S. Chawla, A. Zomaya, „Ensemble-based wrapper methods for feature selection and class imbalance learning“, n.d.

[xviii] A. Ali, S. Mariyam Shamsuddin, A. Ralescu, „Classification with class imbalance problem: A Review“, Int. J. Advance Soft Compu. Appl, Vol. 7, No. 3, 2015

[xix] C. Ling, V. Sheng, „Cost-Sensitive Learning and the Class Imbalance Problem“, Encyclopedia of Machine Learning, Springer, 2008

[xx] C. Ling, V. Sheng, „Cost-Sensitive Learning and the Class Imbalance Problem“, Encyclopedia of Machine Learning, Springer, 2008

[xxi] G. Weiss, K. McCarthy, B. Zabar, „Cost-Sensitive Learning vs. Sampling: Which is Best for Handling Unbalanced Classes with Unequal Error Costs?“, n.d.

[xxii] N. Thai-Nghe, Z. Gantner, L. Schmidt-Thieme, „Cost-Sensitive Learning Methods for Imbalanced Data“, n.d.

[xxiii] J. Burez, D. Van den Poel, „Handling class imbalance in customer churn prediction“, Expert Systems with Applications 36, pp. 4626–4636, 2009

[xxiv] K. Coussement , "Improving customer retention management through cost-sensitive learning", European Journal of Marketing, Vol. 48 Iss: 3/4, pp.477 – 495, 2014

[xxv] T. Imam, K. Ting, and J. Kamruzzaman, "z-svm: an svm for improved classification of imbalanced data," in Proceedings of the 19th Australian joint conference on Articial Intelligence: advances in Artifcial Intelligence, pp. 264-273, Springer-Verlag, 2006.

[xxvi] X. Guo, Y. Yin, C. Dong, G. Yang, G. Zhou, „On the Class Imbalance Problem“, Fourth International Conference on Natural Computation, IEE, 2008

[xxvii] J. Davis, M. Goadrich, „The Relationship Between Precision-Recall and ROC Curves“, Proceedings of the 23rd International Con- ference on Machine Learning, Pittsburgh, 2006

[xxviii] C. Drummond, R. Holte, „Cost curves: An improved method for visualizing classifier performance“, Mach Learn 65, pp.95–130, 2006

[xxix] J. Davis, M. Goadrich, „An introduction to ROC analysis“, Pattern Recognition Letters, 27, 861 – 874. 2006

[xxx] T. Chen, W. Tang, Y. Lu and X. Tu, „Rank regression: an alternative regression approach for data with outliers“, Shanghai Arch Psychiatry, 2014 Oct; 26(5): 310 – 315, 2014

[xxxi] C. Elkan, „The Fundations of Cost-Sensitive Learning“, In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI’01), 2001