Modeling and Simulation of Mobile Apps User Behavior

by

Ranasinghe Arachchige Isuru Harsha Dharmasena

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfillment of the requirements of the degree of

Master of Science

Department of Statistics

University of Manitoba

Winnipeg

Abstract

Mobile applications have become a vital part in modern businesses where products and services are offered in real-time. As many people have adopted to mobile apps, it is not uncommon that some of the applications are used for a few times and then abandoned. This "churning" effect on mobile apps has become a wide topic of interest among businesses to understand the factors affecting the user abandonment. This includes predicting and identifying the abandoning users beforehand to actively engage users to have more active and loyal app users. There is often a class imbalance problem where the retained user group is the minority class. We study and assess several over-sampling methods and under-sampling methods combined with several classification methods to improve the prediction ability and model performance of mobile app user retention using data available from a local mobile app developing company. We then discuss a non-parametric hypothesis testing strategy to compare similar ROC curves obtained by different re-sampling strategies. Finally, we propose a Bayesian network to assess which features in a particular mobile App are affecting the retention of an App user. Re-sampling techniques are then used to improve the performance of the Bayesian network and we use Structural Hamming Distances (SHD) to distinguish similar Bayesian network structures.

keywords: Classification, Churn prediction, Data imbalance, Over-sampling, Under-sampling, Bayesian network

Acknowledgment

Foremost I would like to express my sincere gratitude to my advisor Dr. Saman Muthukumarana, whose invaluable support, patience, motivation and mentorship has helped me a lot over last few years.

My special thank goes to Dr. Mike Domaratzki being in the advisory committee and for his invaluable support, enthusiasm, guidance and immense knowledge towards my MSc. thesis from the beginning of my research work. I would also like to thank Dr. Max Turgeon for being in the advisory committee and allocating his time to review and help me to complete this thesis. The suggestions and the support are appreciated to make my thesis a success.

Many thanks go to the staff, support staff, the faculty and the colleagues in the Department of Statistics, University of Manitoba. Also, I would like to thank the Department of Statistics, University of Manitoba for the financial support provided throughout my MSc. study.

Finally and most specially, I would like to thank my mother and father who supported and believed in me. I would not be able to achieve any of these without the lessons and guidance provided throughout my life. I would also thank my wife, for her support, patience and tolerance during past years to make my thesis a success.

Dedication

This work is dedicated to my mother, father, sister and my wife who have supported me with heart and soul.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this era of advanced communication technologies, mobile applications (Apps) have become primary tools in people's personal and professional lives. Mobile Apps facilitate multiple applications including but not limited to communication, social media, education, entertainment, medical, utilities and travel. Mobile Apps are not only important for the App users but also it plays a crucial role in many modern businesses. Any company providing their services through mobile Apps are interested in new user acquisition as well as retention of existing customers. The customers who continue to use the mobile App over a given period can be considered as retained users whereas this is the opposite of churned users.

Some might argue that increased number of downloads of a particular mobile App indicates a better metric on how well that App is retained among customers but it is not always might be true. One person might download an

App and abandon using after one day or may keep the mobile App without using it for a long time. So far there is no clear line indicating a mobile App user to be classified as retained or churned. Generally it is defined by the App provider considering facts such as the business model and the nature of the mobile App. Despite the nature of business, the majority of the mobile Apps face a churn rate approximately around 70% after 90 days (Perro, 2018). This retention rate of 30% indicates that only 30 out of 100 mobile App users tend to return to their mobile Apps or being "loyal" to the App. Furthermore, it has been shown that 22% users use an App only once after downloading it (Hoch, 2014).

1.1 Motivation

Software and mobile application developing companies are interested in learning the customer behavior to learn which customers are retaining given their usage patterns of the mobile App and the reasons affecting the App user to retain at the end. Mobile App user retention is itself an imbalance problem due to less instances of mobile app users retained after a certain period of time. This affects the final prediction of mobile app user retention and treating the imbalance nature is vital. Picking up the correct method to increase the performance ability of a chosen prediction algorithm is challenging with real world data. We were given a mobile App user dataset from a local App developing company to predict App user retention as well as to identify the reason for the App

users to retain after a certain amount of time. The dataset is an imbalanced problem with 3075 (80.3%) instances of users that left the mobile App (majority class) and only 755 (19.7%) instances of users who retained (minority class). This problem is a binary classification problem with 27 predictors representing in-App feature usage of App users. Generally, most of the learning algorithms and learning systems assume that the data used to learn are balanced with equal instances in each class of the response variable. However, in the real world it is not always true. The number of instances in one class might be more abundant than the others which tend to obstruct the performance of classifiers obtained through Machine Learning (ML) algorithms (Japkowicz et al., 2000).

A dataset is said to be imbalanced if the instances of each class of the response variable are not approximately equal. The imbalance can be of two types, between-class imbalance and within-class imbalance. Between-class imbalance is where some classes have more instances than others (Chawla et al., 2004). Within-class imbalance on the other hand refers to scenarios where subsets of one class have fewer instances than other subsets of same class (Weiss, 2004). Furthermore, the classes with more instances are identified as majority classes or groups while the classes with lesser instances are identified as minority classes or groups in imbalanced datasets.

It is worth to consider several existing re-sampling techniques combined together to overcome any under-performances due to imbalance nature of the dataset. Generally, using an over-sampling technique to balance the minority group would be sufficient but cleaning the noisy majority instances would

also be worthwhile on a imbalance problem. Given many over-sampling and under-sampling techniques, one might wonder what re-sampling method should be used in a given scenario. One might argue to compare few over-sampling techniques to balance the class probabilities of the response variable while another might be interested in cleaning the noisy majority instances near the class boundary. We are interested in the combinations of over-sampling and under-sampling techniques that can improve the performance of the classifiers. We set-up and observe how each re-sampling technique behave on model performance of several classifiers and choose the best combination to treat the dataset of this nature. Finally, we explore the methods to treat any complications arise when we assess and compare the simulations done with the re-sampling combinations.

Moreover, businesses are very much interested in how each feature in a particular mobile App is affected on the retention of an App user. Bayesian networks can be used to model mobile App behavior with respect to the frequency of each feature in the mobile App is used. Many applications on mobile Apps with Bayesian Networks can be found in literature such as mobile App recommendation systems (Park et al., 2007), mobile App usage modeling (Huang et al., 2012) and Android malware detection (Yerima et al., 2014). We use several Bayesian Network learning algorithms to model the mobile App user retention. Furthermore, we use re-sampling techniques to improve the performance of the Bayesian networks.

1.2 Thesis Overview

We discuss model prediction improvement with re-sampling techniques in Chapter 2. We explore existing over-sampling techniques with their method of generating new instances in the minority group in subsection 2.2.1 and we discuss under-sampling techniques to clean the majority group in subsection 2.2.2. We also focus on metrics that we can use to assess the results that we obtain in the data analysis.

In Chapter 3, we explore the theory behind causal networks and the method of learning Bayesian networks from data. We discuss several existing Bayesian structure learning algorithms in section 3.4 and we use re-sampling techniques to improve the network performance measures in the data analysis. Finally, we conclude the thesis with a discussion on our results and solutions to the problem of interest.

Chapter 2

The Imbalance Problem

Generally, most of the learning algorithms and learning systems assume that the data used to learn are balanced with equal instances in each class of the response variable. However, in the real world it is not always true. The number of instances in one class might be more abundant than the others which tend to obstruct the performance of classifiers obtained through Machine Learning (ML) algorithms.

A dataset is said to be imbalanced if the instances of each class of the response variable is not approximately equal. The imbalance can be of two types, between-class imbalance and within-class imbalance. Between-class imbalance is where some classes have more instances than others (Chawla et al., 2004). Within-class imbalance on the other hand refers to scenarios where subsets of one class have fewer instances than other subsets of same class

(Weiss, 2004). Furthermore, the classes with more instances are identified as majority classes or groups while the classes with lesser instances are identified as minority classes or groups in imbalanced datasets.

Imbalance in the proportion of 100 to 1 is frequent in fraud detection and imbalance proportion up to 100,000 to 1 has been observed in other applications (Provost and Fawcett, 2001). Imbalanced data is found in many real world classification problems such as detection of oil spills in satellite radar images (Kubat et al., 1998), telecommunication customer management (Ezawa et al., 1996), text classification (Lewis and Catlett, 1994; Lewis and Ringuette, 1994; Dumais et al., 1998) and caller user profiling (Fawcett and Provost, 1996).

Challenges in learning from imbalanced datasets

Overlapping minority and majority instances makes it difficult to separate the classes since the lack of instances in one class make the border-line weak. Previous works have shown many ways to overcome these problems. One aspect is to balance the dataset by means of over-sampling techniques. Another aspect is cleansing the imbalanced datasets by tidying up noisy majority/minority instances such that border-line of the classes becomes stronger and more distinguishable.

This imbalance issue has been addressed primarily in two ways in previous studies. One is to use unique cost values to train instances (Pazzani et al., 1994; Domingos, 1999). The other method is to re-sample the original data, either

by under-sampling the majority class and/or over-sampling the minority class (Kubat and Matwin, 1997; Japkowicz, 2000; Lewis and Catlett, 1994; Ling and Li, 1998). Under-sampling can be considered as cleaning the dataset where the classifier is mislead by noisy majority instances. Some over-sampling techniques play a role of making the classifiers stronger with synthetically generated minority instances where the classifier tends to fail in certain situations such as near the borderline.

2.1 Chapter Overview

In this chapter, we use several combinations of over-sampling and under-sampling techniques to treat the mobile app user dataset. The respective over-sample and under-sample techniques are discussed in subsection 2.2.1 and subsection 2.2.2. To assess the success-fullness of the re-sampling strategies, we use several classification models and several model metrics to compare the results. We use logistic regression classifier, Naïve Bayes classifier and Support Vector Machine (SVM) to predict the mobile app user retention. These classifiers are discussed in section 2.3. All the trained models are then evaluated using several model metrics discussed in section 2.4.

2.2 Re-Sampling Methods

2.2.1 Oversampling Techniques

Over sampling can be done in many ways and often it deals with creating new instances of the minority group such that the dataset becomes a balanced dataset. Common over sampling techniques for classification problems are as follows:

- Random Over-sampling (ROS)
- Synthetic Minority Over-sampling TEchnique (SMOTE)
- Borderline- SMOTE
- ADAptive SYNthetic over-sampling technique (ADASYN)
- Majority Weight Minority Over-sampling TEchnique (MWMOTE)

Random Oversampling (ROS)

Random naive oversampling can be considered as the most widely used oversampling techniques before other innovatory methods have discovered. Minority class samples are randomly selected and replicated to achieve the balanced dataset. In Figure 2.1 shows the concept of random over sampling on minority class (Fernández et al., 2018). One issue with random oversampling is that this method just duplicate already existing data which would not necessarily benefit

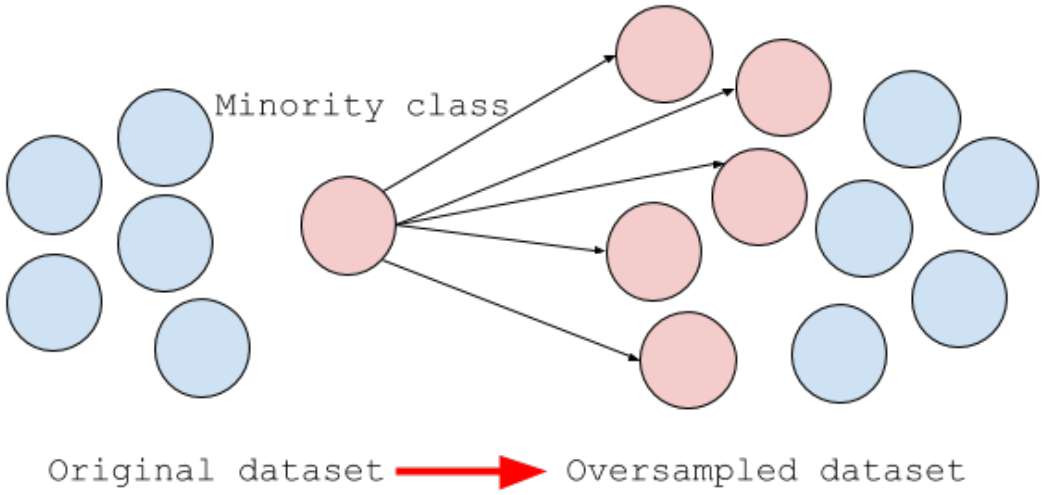


Figure 2.1: Random oversampling

the classification algorithm since duplicates would not give new information on how to classify new observations. Moreover, this often tends to increase the likelihood of overfitting and there is a chance of discarding useful data in the minority class in the process of selecting random samples as well.

Synthetic Minority Oversampling TEchnique (SMOTE)

Synthetic Minority Oversampling TEchnique (SMOTE) (Chawla et al., 2002) was introduced as an innovative method of producing "synthetic" instances of the minority class without duplicating already existing minority class instances. Consider Figure 2.2 as a hypothetical dataset consisting of imbalanced dataset with two classes. Now SMOTE finds the k-nearest neighbors of each instance

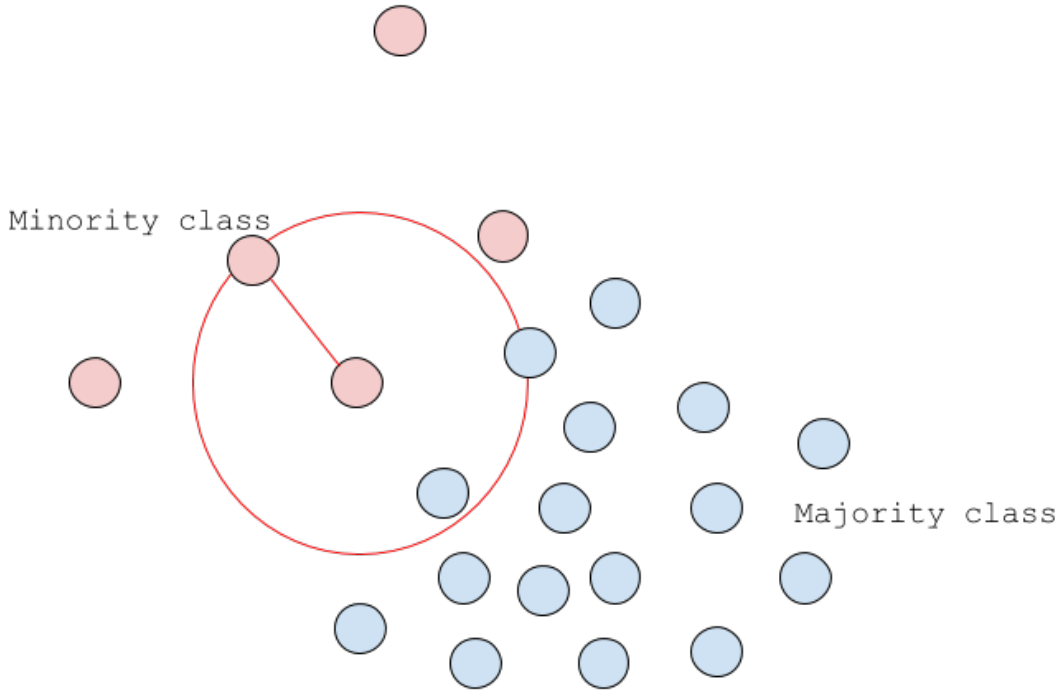


Figure 2.2: Imbalanced two class scenario

in minority class (k = 1 in Figure 2.2).

As in Figure 2.3 , the identified nearest neighbors are used to create new instances by randomly choosing a point in between the line connecting the instance with the nearest neighbor. This process can be done repeatedly for all the minority instances depending on the number of synthetic minority class instances needed as shown in Figure 2.4. As mentioned in (Chawla et al., 2002), **Algorithm** SMOTE is the pseudo-code for SMOTE algorithm.

Algorithm *SMOTE*(T, N, k)

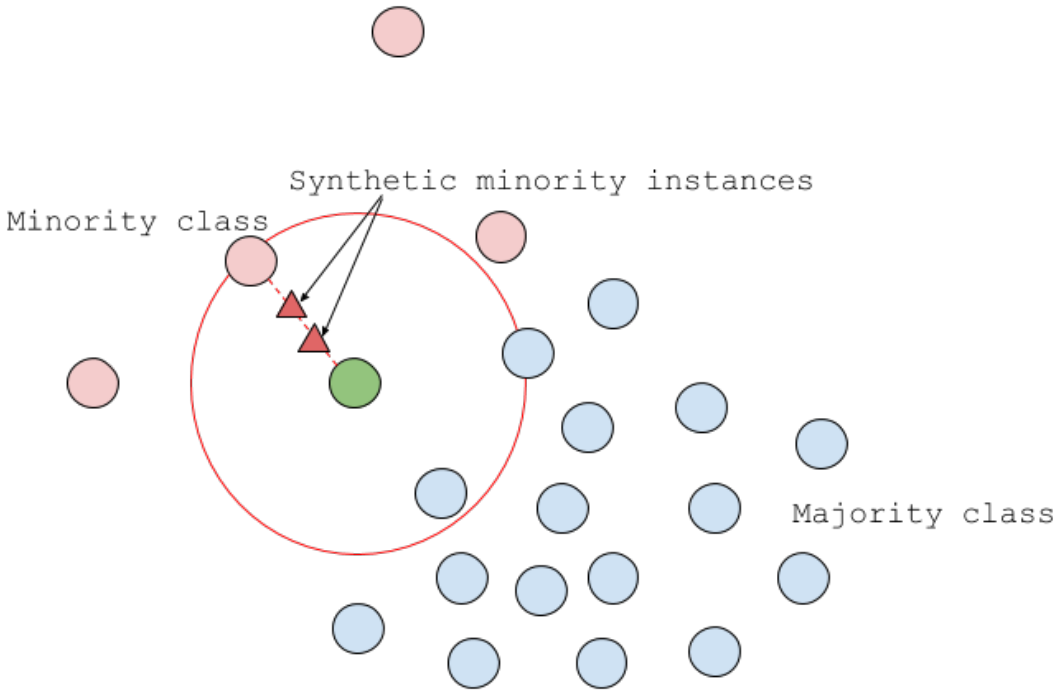


Figure 2.3: Generating new instances between the nearest neighbor and the minority instance

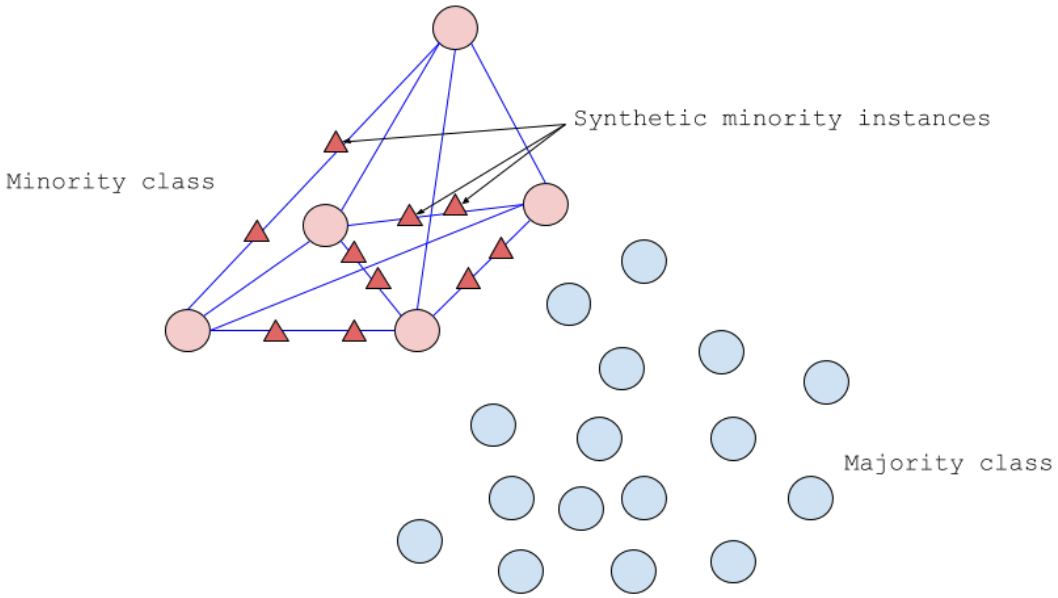


Figure 2.4: Repeatedly creating new instances for the requirement

Input: Number of minority class samples T; Amount of SMOTE $N\%$; Number of nearest neighbors k

Output: $(N/100)*T$ synthetic minority class samples

1. (* If N is less than 100%, randomize the minority class samples as only a random percent of them will be SMOTEd.*)
2. **if** $N < 100$
3. **then** Randomize the T minority class samples
4. $T = (N/100) * T$
5. $N = 100$

6. **endif**

7. $N = (int)(N/100)$ (* *The amount of SMOTE is assumed to be in integral multiples of 100.**)

8. k = Number of nearest neighbors

9. *numattrs* = Number of attributes

10. *Sample*[ ][ ] : array of original minority class samples

11. *newindex* : keeps a count of number of synthetic samples generated, initialized to 0

12. *Synthetic*[ ][ ] : array for synthetic samples

    (* *Compute k nearest neighbors for each minority class sample only.* *)

13. **for** $i \leftarrow 1$ **to** T

14. Compute k nearest neighbors for i, and save the indices in the *nnarray*

15. Populate(N, i, *nnarray*)

16. **endfor**

Populate(N, i, *nnarray*) (* *Function to generate the synthetic samples.* *)

17. **while** $N \neq 0$

18. Choose a random number between 1 and k, call it nn. This step chooses one of the k nearest neighbors of i.

19. **for** $attr \leftarrow 1$ **to** $numattrs$

20. Compute:

$dif = Sample[nnarray[nn]][attr] - sample[i][attr]$

21. Compute: gap = random number between 0 and 1

22. $Synthetic[newindex][attr] = Sample[i][attr] + gap * dif$

23. **endfor**

24. $newindex{+}{+}$

25. $N = N - 1$

26. **endwhile**

27. **return** (* *End of Populate.* *)

End of Pseudo-Code.

Taking a step further, SMOTE algorithm has been combined with standard boosting procedure to create another version of SMOTE; SMOTEBoost which is an improved version for moderately and highly imbalanced datasets (Chawla et al., 2003).

Borderline-SMOTE (BLSMOTE)

The popularity of SMOTE has led another two novel oversampling methods, borderline-SMOTE1 and borderline-SMOTE2 (Han et al., 2005) where over-sampling of minority instances conducted near the borderline. Most of the classification methods attempt to learn the borderline and instances nearby the borderline since these instances tend to be misclassified more compared to other instances away from the borderline.

This method focuses on borderline class instances and using Borderline-SMOTE1 (BLSMOTE1) and Borderline-SMOTE2 (BLSMOTE2) only the borderline instances are over-sampled given their importance in classification that instances which are away from the borderline. Unlike SMOTE, this method tries to oversample and "strengthen" the borderline minority examples by first identifying the borderline minority examples and adding synthetically generated instances to the original training dataset. The detailed procedure of Borderline-SMOTE1 is as follows.

Input If the whole training set is T, the minority class is P and the majority class is N ;

$$P = \{p_1, p_2, ..., p_{pnum}\}, N = \{n_1, n_2, ..., n_{nnum}\}$$

Procedure

1. For every $p_i (i = 1, 2, ..., pnum)$ in the minority class P, we calculate its m nearest neighbors from the whole training set T. The number of

majority examples among the m nearest neighbors is denoted by m' $(0 \leq m' \leq m)$.

2. If $m' = m$, i.e. all the m nearest neighbors of p_i are majority examples, p_i is considered to be noise and is not operated in the following steps. If $m/2 \leq m' < m$, namely the number of p_i 's majority nearest neighbors is larger than the number of its minority ones, p_i is considered to be easily misclassified and put into a set DANGER. If $0 \leq m' < m/2$, p_i is safe and needs not to participate in the following steps.

3. the examples in $DANGER$ are the borderline data of the minority class P, and we can see that $DANGER \subseteq P$. We set

$$DANGER = \{p'_1, p'_2, ..., p'_{dnum}\}, 0 \leq dnum \leq pnum$$

For each example in DANGER, we calculate its k nearest neighbors from P.

4. In this step, we generate $s * dnum$ synthetic positive examples from the data in DANGER, where s is an integer between 1 and k. For each p'_i, we randomly select s nearest neighbors from its k nearest neighbors in P. Firstly, we calculate the distances, $dist_j (j = 1, 2, ..., s)$ between p'_i and its s nearest neighbors from P, then multiply $dist_j$ by a random number $r_j (j = 1, 2, ..., s)$ between 0 and 1, finally, s new synthetic minority

examples are generated between p'_i and its nearest neighbors:

$$synthetic_j = p'_i + r_j * dist_j, j = 1, 2, ..., s$$

This procedure is repeated for each p'_i in $DANGER$ and can attain $s * dnum$ synthetic examples. This step creating synthetic instances is similar to SMOTE (Chawla et al., 2002).

In the procedure above, $p_i, n_i, p'_i, dist_j$ and $synthetic_j$ are vectors. These synthetic new data are generated along the borderline thus strengthen the borderline instances.

Borderline-SMOTE2 generate synthetic instances from each example in $DANGER$ and its positive neighbors in P as well as nearest negative neighbor in N. The distance between $DANGER$ example and the nearest negative neighbor ($dist_j$) is multiplied by a random number (r_j) between 0 and 0.5 resulting new synthetic instances closer to the minority class as the new position of the synthetic instances are calculated using,

$$synthetic_j = p'_i + r_j * dist_j, j = 1, 2, ..., s.$$

Figure 2.5 shows a simple illustration on the Borderline-SMOTE procedure and how it differentiates from SMOTE.

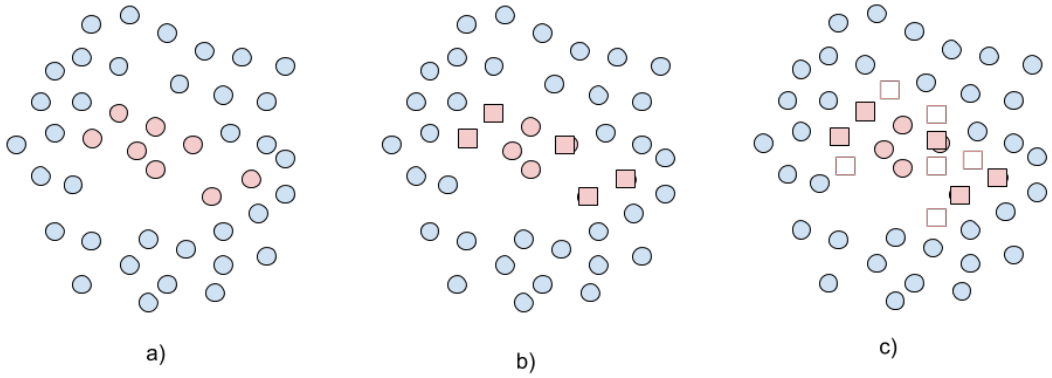


Figure 2.5: a) Sample original dataset (colors representing classes). b) Borderline minority examples chosen by algorithm (Solid squares) c) The borderline synthetic minority examples (hollow squares)

ADAptive SYNthetic generation (ADASYN)

The success in synthetic data generation for imbalanced datasets including SMOTE (Chawla et al., 2002), SMOTEBoost (Chawla, 2009) and DataBoost-IM (Guo and Viktor, 2004) has led to implementation of an adaptive method of imbalance learning called ADASYN (Haibo He et al., 2008). The most important objective of introducing this method is to reduce the bias and adaptively learning for the given data. The Pseudo-Code of **Algorithm** ADASYN (Haibo He et al., 2008) for the two - class classification problem is as follows.

Input

1. Training dataset T with t samples $\{x_i,y_i\}$, $i = 1, ..., t$, where x_i is an instance in n dimensional feature space X and $y_i \in Y = \{1, -1\}$ is the

class identity label associated with x_i. Define p_{num} and n_{num} as the number of minority class examples and the number of majority class examples respectively. Therefore, $p_{num} \leq n_{num}$ and $p_{num} + n_{num} = t$.

Procedure

1. Calculate the degree of class imbalance:

$$d = p_{num}/n_{num}$$

where $d \in (0, 1]$.

2. If $d < d_{th}$ then (d_{th} is the preset threshold for the maximum tolerated degree of class imbalance ratio):

   (a) Calculate the number of synthetic data examples that need to be generated for the minority class:

$$G = (n_{num} - p_{num}) * \beta$$

   where $\beta \in [0, 1]$ is a parameter used to specify the desired balance level after generation of the synthetic data. $\beta = 1$ means a fully balanced dataset is created after the generalization process.

   (b) For each example $x_i \in$ minority class, find K nearest neighbors based on the Euclidean distance in n dimensional space, and

calculate the ratio r_i defined as:

$$r_i = \Delta_i / K, i = 1, ..., p_{num}$$

where Δ_i is the number of examples in the K nearest neighbors of x_i that belong to the majority class, therefore $r_i \in [0, 1]$

(c) Normalize r_i according to $\hat{r}_i = r_i / \sum_{i=1}^{p_{num}} r_i$, so that $\hat{r}_i$ is a density distribution ($\sum_i \hat{r}_i = 1$)

(d) Calculate the number of synthetic data examples that need to be generated for each minority example x_i:

$$g_i = \hat{r}_i * G$$

where G is the total number of synthetic data examples that need to be generated for the minority class as defined in part a.

(e) For each minority class data example x_i, generate g_i synthetic data examples according to the following steps:

Do the **Loop** from 1 to g_i :

i. Randomly choose one minority data example, x_{zi}, from the K nearest neighbors for data x_i.

ii. Generate the synthetic data example:

$$s_i = x_i + (x_{zi} - x_i) * \lambda$$

where $(x_{zi} - x_i)$ is the difference vector in n dimensional spaces, and λ is a random number: $\lambda \in [0, 1]$

End **Loop**

Furthermore, ADASYN can also change the decision boundary automatically in order to learn from instances that are harder to learn otherwise hence improving the performance.

Majority Weight Minority Oversampling TEchnique (MWMOTE)

Majority Weight Minority Oversampling TEchnique (MWMOTE) (Barua et al., 2014) is another improvement done over existing oversampling techniques so that the minority instances that are harder to learn will be isolated and assigned weights according to their Euclidean distances to the nearest minority class instances. Because of this method all the instances generated synthetically will fall within the minority cluster. MWMOTE's objective is twofold i.e. to improve the synthetic sample generation process and to improve the sample selection process. MWMOTE completes in three major steps. First, it identifies the most important and "hard-to-learn" instances from the original minority instances S_{min} and construct a subset S_{imin}. In the second phase each instance in S_{imin} is given a weight S_w according to the magnitude of the importance. Finally in the third phase, MWMOTE generates the synthetic samples from S_{imin} using S_ws and produce the outputs S_{omin} and adding these synthetically generated samples to S_{min}.

The Pseudo- Code of **Algorithm** MWMOTE the two - class classification problem is as follows.

Algorithm MWMOTE($S_{maj}, S_{min}, N, k_1, k_2, k_3$).

Input:

1. S_{maj}: Set of majority class samples
2. S_{min}: Set of minority class samples
3. N : Number of synthetic samples to be generated
4. $k1$: number of neighbors used for predicting noisy minority class samples
5. $k2$: Number of majority neighbors used for constructing informative minority set
6. $k3$: Number of minority neighbors used for constructing informative minority set

Procedure Begin

1. For each minority example $x_i \in S_{min}$, compute the nearest neighbor set, $NN(x_i)$. $NN(x_i)$ consists of the nearest $k1$ neighbors of x_i according to euclidean distance.
2. Construct the filtered minority set, S_{minf} by removing those minority class samples which have no minority example in their neighborhood: $S_{minf} = S_{min} - \{x_i \in S_{min} : NN(x_i)$ contains no minority example$\}$

3. For each $x_i \in S_{minf}$, compute the nearest majority set, $N_{maj}(x_i)$. $N_{maj}(x_i)$ consists of the nearest $k2$ majority samples from x_i according to Euclidean distance.

4. Find the borderline majority set, S_{bmaj}, as the union of all $N_{maj}(x_i)$s, i.e.

$$S_{bmaj} = \cup_{x_i \in S_{minf}} N_{maj}(x_i)$$

5. For each majority sample $y_i \in S_{bmaj}$, compute the nearest minority set, $N_{min}(y_i)$. $N_{min}(y_i)$ consists of the nearest $k3$ minority examples from y_i according to Euclidean distance.

6. Find the informative minority set, S_{imin}, as the union of all $N_{min}(y_i)$s, i.e., $S_{imin} = \cup_{y_i \in S_{bmaj}} N_{min}(y_i)$

7. For each $y_i \in S_{bmaj}$ and for each $x_i \in S_{imin}$, compute the information weight, $I_w(y_i, x_i)$.

8. For each $x_i \in S_{imin}$, compute the selection weight $S_w(x_i)$ as $S_w(x_i) = \sum_{y_i \in S_{bmaj}} I_w(y_i, x_i)$

9. Convert each $S_w(x_i)$ into selection probability $S_p(x_i)$ according to

$$S_p(x_i) = S_w(x_i) / \sum_{z_i \in S_{imin}} S_w(z_i)$$

10. Find the clusters of S_{min}. Let M clusters be formed which are L_1, L_2, ..., L_M.

11. Initialize the set, $S_{omin} = S_{min}$.

12. Do for $j = 1, ..., N$,

    (a) Select a sample x from S_{imin} according to probability distribution $\{S_p(x_i)\}$. Let x be a member of the cluster L_k, $1 \leq k \leq M$.

    (b) Select another sample y, at random, from the members of the cluster L_k.

    (c) Generate one synthetic data, s, according to $s = x + \alpha * (y - x)$, where α is arandom number in the range $[0, 1]$.

    (d) Add s to $S_{omin} : S_{omin} = S_{omin} \cup s$.

    (e) End Loop **End**

    Output : the oversampled minority set, S_{omin}.

MWMOTE uses a clustering approach so that it ensures the generated synthetic instances are located within the minority class area avoiding any random synthetic noise generation.

2.2.2 Under-sampling Techniques

In under sampling, we downsize the actual dataset such that the dependent variable categories become a ratio of atleast 10:1. Common under-sampling

techniques for classification problems are as follows,

- Random under-sampling (RUS)
- Edited Nearest Neighborhood Rule (ENN)
- Neighborhood Cleaning Rule (NCL)
- Tomek links (TL)
- One-Sided Selection (OSS)
- Under-sampling Based on Clustering (SBC)

Random under-sampling (RUS)

Random under-sampling involves removal of random instances from the majority class with or without replacement. This is considered as the earliest under-sampling techniques used. This may increase the variance of the classifier, hence potentially may discard useful and important instances from the original dataset (Fernández et al., 2018).

Edited Nearest Neighborhood Rule (ENN)

The Edited Nearest Neighborhood (ENN) algorithm removes instances from the a class that are misclassified by their k nearest neighbors (Wilson, 1972; Tomek, 1976). This method does not require any prior knowledge of the distribution,

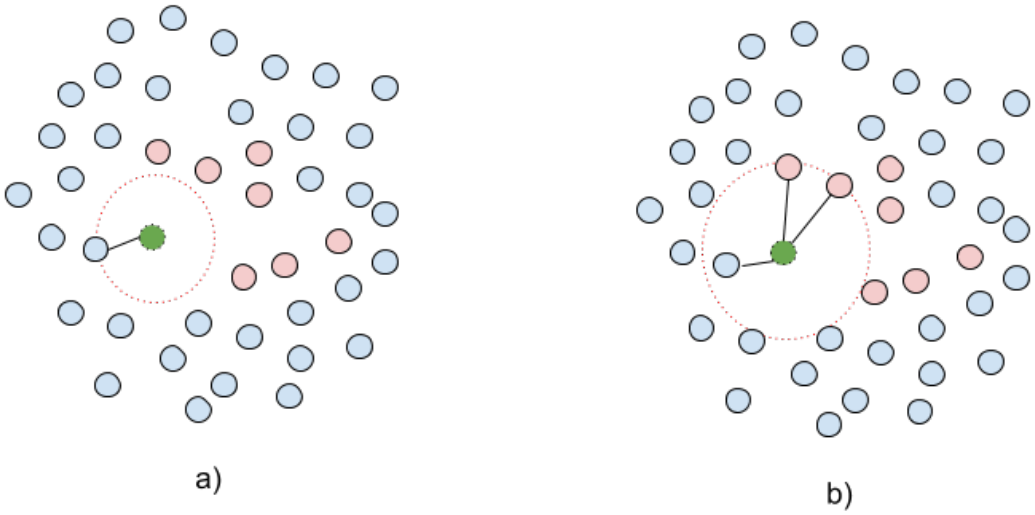


Figure 2.6: a) Nearest neighbor rule. b)k- Nearest neighbor rule

hence considered a non-parametric decision rule. These decision rules rely on the training instances with known class membership to make decisions on the class membership of unknown instances. Often Euclidean metrics are used to classify an unknown instance to the class of its nearest neighbor in the measurement space. Although this method is simple to implement, the asymptotic nearest neighbor error is said to be never two times worse than the Bayes (optimal) error (Cover and Hart, 2018). i.e.

$$p_e^{nn-rule} < P_e(2 - P_e eN(N-1))$$

Where P_e is the Bayes error and N is the size of the training set. According to 2.6, the clear modification, k-nearest neighbor decision rule is better. Despite the simplicity, the requirement of assessing the whole training dataset in order

to make a single membership makes this method space expensive. To overcome this issue many methods have been implemented to edit the training with proximity graphs such as Voronoi, Delaunay triangulation (Bhattacharya et al., 1981).

Neighborhood Cleaning Rule (NCL)

Neighborhood Cleaning Rule (NCL) (Laurikkala et al., 2001) uses Wilson (1972) 's Edited Nearest Neighbor rule (ENN) to remove majority examples i.e. under-sample. This algorithm tends to get rid of any instance whose class differ from the class of at least two of its three nearest neighbors. NCL uses ENN to clean the dataset.

Neighborhood cleaning rule can be described as follows;

1. Split data T into the class of interest C and the rest of data O.
2. Identify noisy data A_1 in O with edited nearest neighbor rule.
3. For each class C_i in O, if ($x \in C_i$ in 3 - nearest neighbors of misclassified $y \in C$) and ($|C_i| \geq 0.5 * |C_i|$) then $A_2 = \{x\} \cup A_2$.
4. Reduced dataset $S = T - (A_1 \cup A_2)$.

Tomek links (TL)

Tomek Links (TL) (Tomek, 1976) can be defined as follows; given two instances E_i and E_j in different classes, and $d(E_i, E_j)$ is the distance between E_i and E_j. A (E_i, E_j) pair is called a TL if there is no instance E_l such that $d(E_i, E_l) < d(E_i, E_j)$ or $d(E_j, E_l) < d(E_i, E_j)$. If two instances create a TL, then either one or both instances are on borderline. This method can be used to under-sample and clean the borderline majority instances.

One-Sided Selection (OSS)

One-sided Selection (OSS) (Kubat and Matwin, 1997) is another under-sampling technique which uses Tomek Links (TL) and by applying Condensed Nearest Neighborhood Rule (CNN) (Hart, 1968). Tomek Links are used as an under-sampling method to detect and remove borderline majority instances. Then CNN is used to remove instances from the majority instances that are away from the borderline.

The above concept can be put into a simple algorithm as follows;

1. Let T be the original training set.

2. Initially, C contains all positive instances from T and one randomly selected negative instance.

3. Classify T with the 1- NN rule using the instances in C, and compare the assigned labels with the original ones. Move all misclassified instances into C that is now compatible with T but at the same time being smaller.

4. Remove from C all negative instances participating in Tomek Links. This removes those negative instances that are believed borderline and/or noisy. All positive instances are retained. The resulting set that is under-sampled can be referred as D.

Under-sampling Based on Clustering (SBC)

Under-sampling Based on Clustering (SBC) uses k number of clusters to randomly select majority instances from each cluster based on the imbalance percentage within those clusters (Yen and Lee, 2009). Let $Size_{MA}$ be the number of majority instances and $Size_{MI}$ be the number of minority instances of a dataset of size N. Then using the following steps we can re-sample using identified clusters as follows:

1. Determine the ratio of $Size_{MA}$ to $Size_{MI}$ in the training dataset.

2. Cluster all the samples (instances) in the dataset into k clusters

3. Determine the number of selected majority instances in each cluster by using the expression:

$$SSize^i_{MA} = (m * Size_{MI}) * \frac{Size^i_{MA}/Size^i_{MI}}{\sum_{i=1}^{k} Size^i_{MA}/Size^i_{MI}}$$

   where, $Size_{MA} : Size_{MI} = m : 1 (m \geq 1)$. Then majority instances are randomly selected from each cluster.

4. Combine the selected majority instances and minority instances to yield the under-sampled training dataset.

Under sampling is not often considered as a better choice to overcome the imbalance problem since that method discards important information about the majority group. Hence that method becomes data inefficient compared to a method that retains the majority data while dealing with the minority problem.

2.3 Binary Classification Methods

In this section, we present the binary classification models used to evaluate the performance changes by re-sampling techniques. We use logistic regression, naïve Bayes classifier and Support Vector Regression (SVR) to predict the retention of mobile app users.

Logistic Regression

The logistic regression model is widely implemented in binary classification problems (Hastie et al., 2009) as it provides predictions of the probabilities that can be converted to the form of 0 or 1 values. It usually fits data with maximum likelihood method and models the logit of the probabilities as a linear function of predictors. Assuming this linearity in the function with only one explanatory variable (x), the logistic function can be written as follows:

$$\hat{y} = \frac{1}{1 + \exp^{-(\beta_0 + \beta_1 x)}} \tag{2.1}$$

where β_1 is the coefficient of the explanatory variable x and β_0 is the intercept.

We note that the fact that logistic regression requires far less computational resources compared to some classifiers like support vector machines (SVM) can be considered as a benefit. Furthermore, the linear function of the logistic model provides the significance of each response variable towards the outcome of the response variable (Hastie et al., 2009).

Naive Bayes model

For given number of examples n, $x = (x_1, x_2, ..., x_n)$ with instance probabilities $p(C_k|x_1, x_2, ..., x_n)$ for each of K possible outcomes of classes C_k, using Bayes

theorem, $p(C_k|x_1, x_2, ..., x_n)$ can be decomposed as

$$p(C_k|x) = \frac{p(C_k)p(x|C_k)}{p(x)}.$$

Since the denominator is effectively constant, the numerator is equivalent to the joint probability model $p(C_k, x_1, ..., x_n)$ and by using the chain rule on repeated conditional probability this joint probability can be rewritten as

$$\begin{aligned} p(C_k, x_1, x_2, ..., x_n) &= p(x_1, ..., x_n, C_k) \\ &= p(x_1|x_2, ..., x_n, C_k)p(x_2, ..., x_n, C_k) \\ &= ... \\ &= p(x1|x_2, ..., x_n, C_k)p(x_2|x_3, ...x_n, C_k)...p(x_{n-1}|x_n, C_k)p(x_n|C_k)p(C_k). \end{aligned}$$

Assuming that all the explanatory variables are mutually independent, we can write the "naive" conditional independence as

$$p(x_i|x_{i+1}, ..., x_n, C_k) = p(x_i|C_k).$$

Then, the joint model can be expressed as

$$\begin{aligned} p(C_k|x_1, ..., x_n) &\propto p(C_k, x_1, ..., x_n) \\ &= p(C_k)p(x_1|C_k)p(x_2|C_k)p(x_3|C_k)... \\ &= p(C_k)\prod_{i=1}^{n} p(x_i|C_k). \end{aligned}$$

For the Naive Bayes classifier problem, the maximum a posteriori or MAP decision rule is used and the Bayes classifier which assigns the class label $\hat{y} = C_k$ for some k as follows:

$$\hat{y} = \underset{k\in 1,...K}{\operatorname{argmax}}\, p(C_k)\prod_{i=1}^{n} p(x_i|C_k)$$

The assumption on the independence of predictor variables makes it efficient in high dimensional datasets and it also requires less computational power though the algorithm seems straightforward (Hastie et al., 2009). We use `e1071` R package (Meyer et al., 2019) to train naïve Bayes classifiers for this study.

Support Vector Machine (SVM)

Hyperplanes are used to classify datasets with a high-dimensional feature space in support vector machine (SVM) (Cortes and Vapnik, 1995). In order to find

the optimal hyperplane that would maximize the distance between margins, a SVM uses kernels such as radial basis function (RBF) to calculate distance between high dimensional data points. The Support vectors are the optimal marginal data points that anchor the hyperplane. The function to predict the class of a new sample with weights, where the weights of the hyperplane that provide the maximum margin which is trained on the train set is as follows:

$$\hat{y} = w.u + b = (\sum_{i=1}^{l} a_i y_i x_i).u + b \tag{2.2}$$

where x_i are the input features with set of weights w whose linear combination predicts y_i s for l instances with bias value b and a_i slack variables that are introduced in the maximization problem.

Support Vector Regression (SVR)

Support Vector Regression (SVR) (Drucker et al., 1997) is the regression version of SVM which is often used in high dimensional regression problems. Interestingly, SVR maintains all the properties from SVM while attempting to find a match between some vector and the position in the curve found by SVR which is not acting as a decision boundary. Support vectors participate in finding the best match between data instances and the actual function that is represented by them. When the distance between support vectors and regression curve is maximized, it becomes more closer to the actual curve. Like

Table 2.1: Confusion matrix for binary classification problem

	Predicted	
Actual	Positive	Negative
Positive	*True Positive(TP)*	*False Negative(FN)*
Negative	*False Positive(FP)*	*True Negative(TN)*

SVM, SVR can also use kernels in order to regress non-linear functions. In our problem, we use a variation of SVR, which is nu-SVR which the number of support vectors is limited. For the purpose of this study, we use `kernlab` R package (Karatzoglou et al., 2019) and the `ksvm()` function with only changing the type to `nu-svr` and kernel to `rbfbot` that will yield SVR model by using nu-SVR and radial-basis kernel.

2.4 Evaluation Metrics

Performance of classifiers have been primarily assessed using tools such as precision, recall and accuracy to reflect the effect of imbalanced data (Ling and Li, 1998; Provost and Fawcett, 2001). More information about the actual and predicted classes of a given binary classifier can be obtained using a confusion matrix in Table 2.1.

Here in Table 2.1, represents a confusion matrix of a binary classification problem having **positive** (1) and **negative** (0) class values. It is possible to extract a number of widely used performance metrics like precision, recall,

accuracy, F1 score from a confusion matrix like in Table 2.1. The methods of computing previously mentioned performance metrics are as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.4}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.5}$$

$$\text{F1} - \text{score} = 2 \times \frac{Precision.Recall}{Precision + Recall} \tag{2.6}$$

The above performance matrices that use values from both classes in a confusion matrix as Table 2.1 would be sensitive to class skewness and might mislead especially in an imbalance situation. For example, when we use accuracy or error rate (1-accuracy) it is a disadvantage in an imbalance problem since it consider both classification errors (either positive or negative) to be equally important. To address this issue, it would be better to consider metrics that consider classes independently as follows:

$$\text{False negative rate} = FN_{rate} = \frac{FN}{TP + FN} \tag{2.7}$$

$$\text{False positive rate} = FP_{rate} = \frac{FP}{FP + TN} \tag{2.8}$$

$$\text{True negative rate} = TN_{rate} = \frac{TN}{FP + TN} \tag{2.9}$$

$$\text{True postive rate} = TP_{rate} = \frac{TP}{TP + FN} \tag{2.10}$$

These performance measures are independent from class probabilities and costs. Furthermore, Receiver Operating Characteristic (ROC) curve (Provost and Fawcett, 1997) can be used to analyze the relationship between FN rate and FP rate (or TN rate and TP rate). It characterizes the performance of a binary classifier across all trade offs between the sensitivity of the classifier (TP_{rate}) and the false alarm (FP_{rate}). ROC analysis also allows the comparison of multiple classification functions simultaneously. Furthermore, area under curve (AUC) of ROC curve represents the expected model performance in a single scalar and is equivalent to the Wilcoxon rank test and other statistical measures of evaluating classification and ranking models (Hand, 1997). F1 score can also be considered as a sound measurement for classification problems since it encircles the trade-off between precision and recall and reflects how well a classifier is in a single measurement (Powers and Ailab, 2011).

2.4.1 k-Fold Cross Validation

In k-fold cross validation, a given dataset D is partitioned into k equal and mutually exclusive partitions (folds) $D_1, D_2, ..., D_k$. Then each partition is used to test the model which is trained on the the remainder partitions combined

together as the training set. Hence, k-fold cross validation make sure that the candidate model is trained on many possible combinations of data to obtain a better estimate on the model metrics preventing possible overfitting. Although k-fold cross validation is computationally intensive, reduced bias in the results and decrease in variance of the estimate with the increasing of number of folds (k) can be considered as the key advantages. Typically the value of k set to 5 or 10.

In our study, we obtain multiple ROC curves for every re-sampling strategy. In order to compare those ROC curves, there are few methods proposed in literature. One is to fit a parametric model and test the equality of the parameters (Dorfman and Alf, 1969; Metz et al., 1984). A redefined non-parametric test was introduced by DeLong et al. (1988) to compare the AUC for paired and unpaired data. Furthermore, Venkatraman and Begg (1996) have developed a complete non-parametric test to compare two ROC curves when the data are paired and continuous. This test is also capable of distinguishing two ROC curves crossing each other but have equal AUCs. We use De Long's non-parametric hypothesis test to compare similar ROC curves obtained by different re-sampling strategies.

2.4.2 Averaging ROC Curves

ROC represents the trade-off between sensitivity and specificity of a given prediction model simply showing the capability of distinguishing between models

while area under curve (AUC) provide an aggregate measure of performance of the model across all possible classification thresholds. Since AUC only measures the model prediction quality irrespective of the classification threshold chosen, obtaining ROC curves alongside is beneficial when we are interested in minimizing one type of error (either false negatives or false positives). In a study like this, obtaining ROC curves that represent 10-fold cross validation result is challenging. In literature, several methods are discussed for multi-reader multi-case (MRMC) ROC studies in medical imaging systems (Chen and Samuelson, 2014). Here, we will discuss about the methods prevailing on averaging ROC curves and then try choose a method of averaging ROC curves for our analysis.

Consider the ROC curves from 10-fold cross validation for the logistic regression model using the original training dataset as in Figure 2.7a. To obtain an average ROC curves we have few options.

- Average by calculating mean TPR and FPR values from folds
- Average sensitivity (S_e) at each specificity (S_p)
- Average specificity at each sensitivity
- Average $\frac{S_e+S_p}{2}$ at each fixed $\frac{S_e-S_p}{2}$

The first method is simply taking the mean value of each TPR and FPR value to get a mean ROC curve as shown in Figure 2.7b.We can generalize last three options by following algorithm (Chen and Samuelson, 2014):

- Rotate the axes (FPR,TPR) in ROC space counter-clockwise for an angle θ to the (u, v) space:

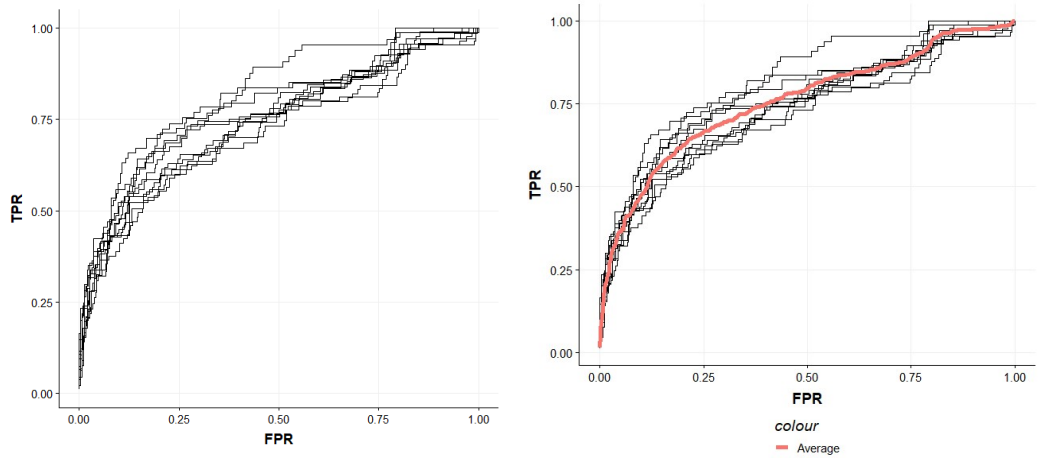
$$\begin{cases} u = FPRcos\theta + TPRsin\theta \\ v = -FPRsin\theta + TPRcos\theta \end{cases}$$

- Average ROC curves in (u, v) space by averaging v for each u
- Rotate the averaged curve in (u, v) space back to ROC space:

$$\begin{cases} FPF = ucos\theta - vsin\theta \\ TPF = -usin\theta + vcos\theta \end{cases}$$

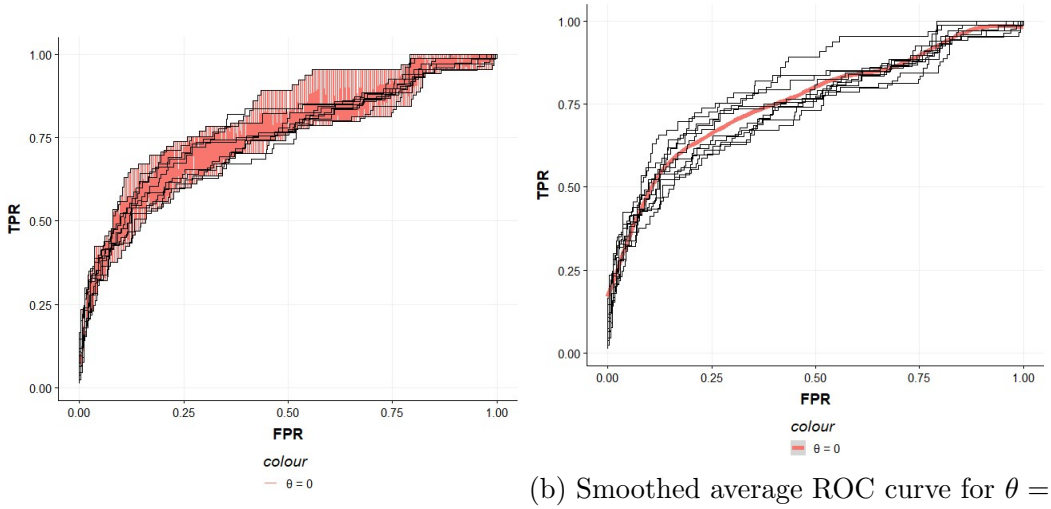
The parameter θ influences the direction along which the ROC curves are averaged. With this algorithm, the method of averaging sensitivity (S_e) at each specificity (S_p) is when $\theta = 0$ (Figure 2.8). Similarly, averaging specificity at each sensitivity corresponds to $\theta = \frac{\pi}{2}$ (Figure 2.9) while the last method is when $\theta = \frac{\pi}{4}$ (Figure 2.10).

Furthermore, we will retain all data of every ROC curve in 10-fold cross validation and try to obtain a smoothed ROC curve that represent an average ROC curve as given in Figure 2.11. When we compare the corresponding



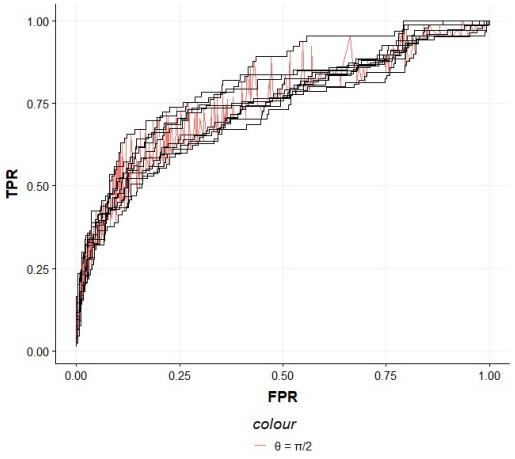
(a) ROC curves for 10-fold cross validation (b) Mean ROC curve

Figure 2.7: ROC curves for 10-fold cross validation with mean ROC curve

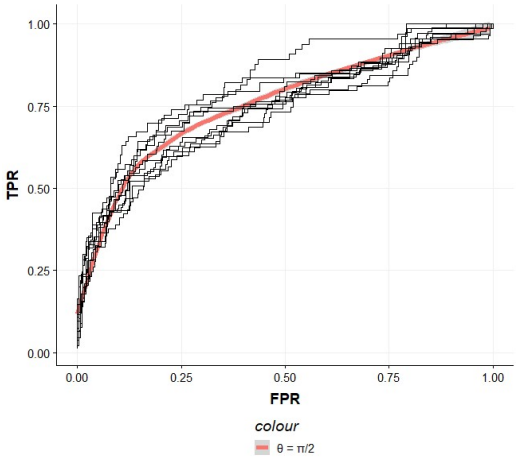


(a) Average ROC curve for $\theta = 0$ (b) Smoothed average ROC curve for $\theta = 0$

Figure 2.8: Average ROC curve and respective smoothed ROC curve by averaging sensitivity at each specificity ($\theta = 0$)

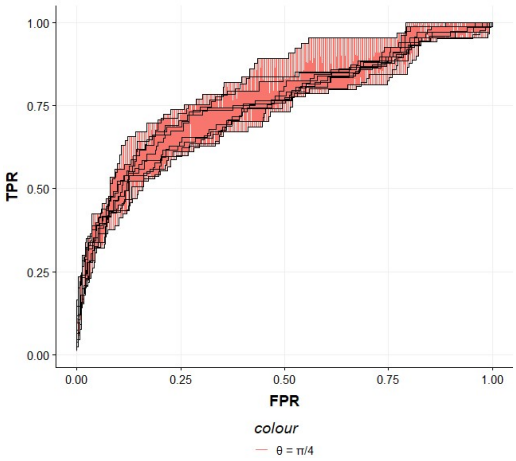


(a) Average ROC curve for $\theta = \frac{\pi}{2}$

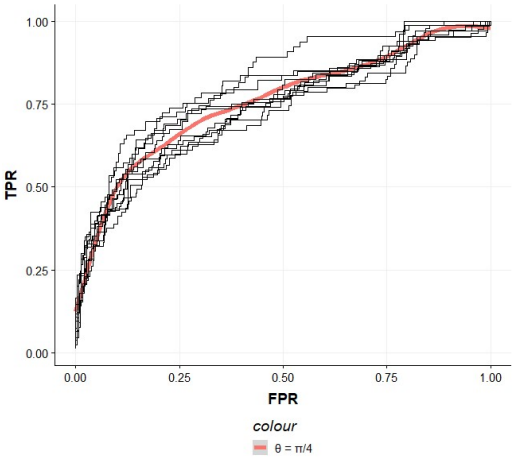


(b) Smoothed average ROC curve for $\theta = \frac{\pi}{2}$

Figure 2.9: Average ROC curve and respective smoothed ROC curve by averaging specificity at each sensitivity ($\theta = \frac{\pi}{2}$)

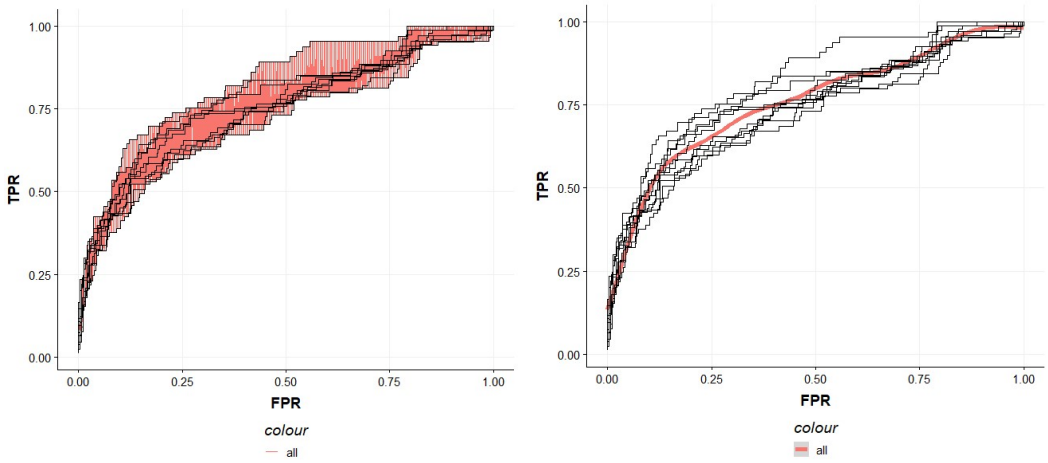


(a) Average ROC curve for $\theta = \frac{\pi}{4}$



(b) Smoothed average ROC curve for $\theta = \frac{\pi}{4}$

Figure 2.10: Average ROC curve and respective smoothed ROC curve by averaging $\frac{S_e+S_p}{2}$ at each fixed $\frac{S_e-S_p}{2}$ ($\theta = \frac{\pi}{4}$)



(a) Average ROC curve by combining all 10-fold data

(b) Smoothed average ROC curve by combining all 10-fold data

Figure 2.11: Average ROC curve and respective smoothed ROC curve by all 10-fold ROC data combined

Method	Average AUC
all	0.7612
mean	0.7563
θ=0	0.7615
$\theta = \pi/2$	0.7166
$\theta = \pi/4$	0.7570

Table 2.2: Average AUC from each averaging method

resultant average curves, each can be compared together as shown in Figure 2.12. According to this plot the average ROC curves seem to be similar but when we consider area under curves for each averaging method (Table 2.2), we can observe that ROC curve obtained by averaging specificity at each sensitivity yields the least AUC value.

For the purpose of assessing the performance of classifiers with re-sampling

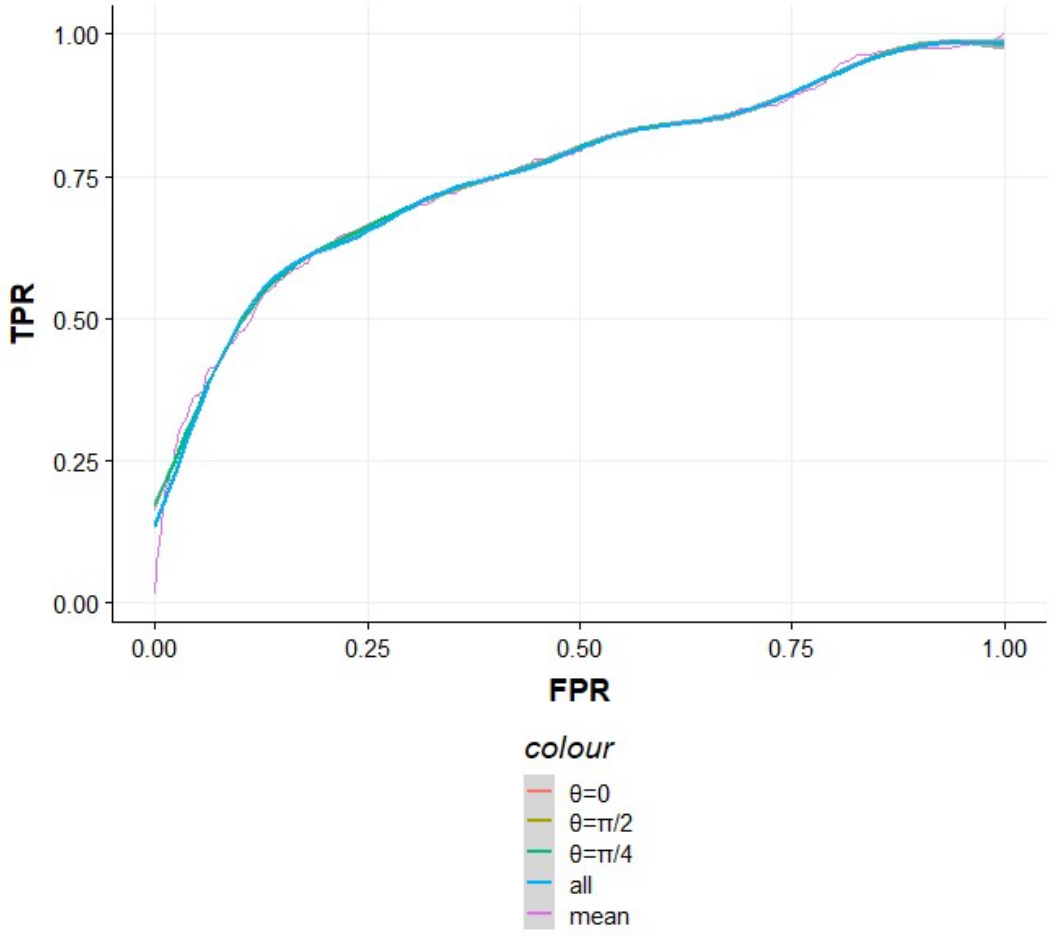


Figure 2.12: Average ROC curves from all ROC curve averaging methods

strategies, we use the method of retaining all 10-fold prediction data in order to obtain an average ROC curve for a given model.

2.5 Data Analysis - Re-sampling Methods

The mobile App user dataset consists of 27 explanatory variables and a response variable with only 19.7% of instances for App users that are retained at the end of the time period. We combine over-sampling and under-sampling methods to treat the imbalance percentage of the response variable and use logistic regression, Naïve Bayes and Support Vector Regression to classify retention of the mobile App users. We assess F1 score and area under curve (AUC) values with 10 - fold cross validation and then the best re-sampling combinations to achieve best F1 score is picked. In order to compare ROC curves, we use several ROC curve averaging methods to get the average ROC curves for the results from 10 - fold cross validation. Furthermore, we use non-parametric tests to compare any similar ROC curves.

2.5.1 Simulation Study

We use every over-sampling techniques (OS) discussed in subsection 2.2.1 to over-sample the training dataset with four levels of (35%, 40%, 45%, 50%) over-sampling percentages and the resultant datasets are then under-sampled

using the under-sampling techniques (US) discussed in subsection 2.2.2. Under-sampling is done to clean the majority group and then the final re-sampled training dataset is obtained. For an example, 50% represents balancing the response variable using the over-sampling technique while 40% means that we over-sample minority instances such that the new imbalance percentage of the response variable is 40% and then use under-sampling techniques to clean the majority group so that the final training dataset is a balanced dataset for most of re-sampling combinations.

2.5.2 Results

The F1 score changes with the over-sampling percentage for logistic regression model are given in Figure 2.13. According to the F1 score changes with over-sampling percentage, we can observe that over-sampling done using ADASYN shows constant improvement over the over-sampling percentage while oversampling with Borderline-SMOTE 1 (BLSMOTE1) reduces F1 score after 40%. According to Figure 2.14, over-sampling from Borderline-SMOTE 1 results in poor F1 scores for naïve Bayes models compared with other over-sampling techniques. Over-sampling with ADASYN does not improve F1 score significantly for the naïve Bayes models. According to Figure 2.15, the worst F1 score for SVR is where no over-sampling method is used on the training dataset while ADASYN and Borderline-SMOTE 2 over-sampling techniques improves F1 scores over the over-sampling percentages of the minority group. Figure 2.16

Table 2.3: Highest F1 score for each classifier by re-sampling methods with 10-fold cross validation

Percentage	OS	US	Model	F1 Score	sd_F1	AUC	sd_AUC
40	SMOTE	ENN	Logistic	0.5270	0.0254	0.7633	0.0261
50	MWMOTE	OSS	SVR	0.4952	0.0402	0.7312	0.0372
40	MWMOTE	ENN	NaiveBayes	0.4621	0.0298	0.7353	0.0317

Table 2.4: Least F1 score for each classifier by re-sampling methods with 10 fold cross validation

Percentage	OS	US	Model	F1 Score	sd_F1	AUC	sd_AUC
35	No Oversampling	ENN	Logistic	0.3846	0.0477	0.7535	0.0324
45	BLSMOTE_1	RUS	NaiveBayes	0.2501	0.1021	0.6021	0.0919
50	No Oversampling	TL	SVR	0.0443	0.0209	0.7197	0.0379

summarizes all F1 scores from the three models and overall logistic regression model yields better F1 scores in most of the cases. This is true for area under curves of each model as shown in Figure 2.17.

Considering all combinations of over-sampling percentages, over-sampling techniques and under-sampling techniques Table 2.3 shows the top performing combinations of each classifier according to F1 scores. Table 2.4 shows the most under performing combinations for each classifier while Table 2.5 shows classifier performance obtained from training dataset without using any re-sampling technique. As seen in Table 2.3, the top performing classifier according to F1 score is logistic regression model obtained using a train dataset obtained by over-sampling minority instances using SMOTE until the percentage of minority group is 40% and then cleaning the majority group using ENN. The second best classifier is SVR which used a train dataset that was balanced using MWMOTE and then clean the majority group using OSS.

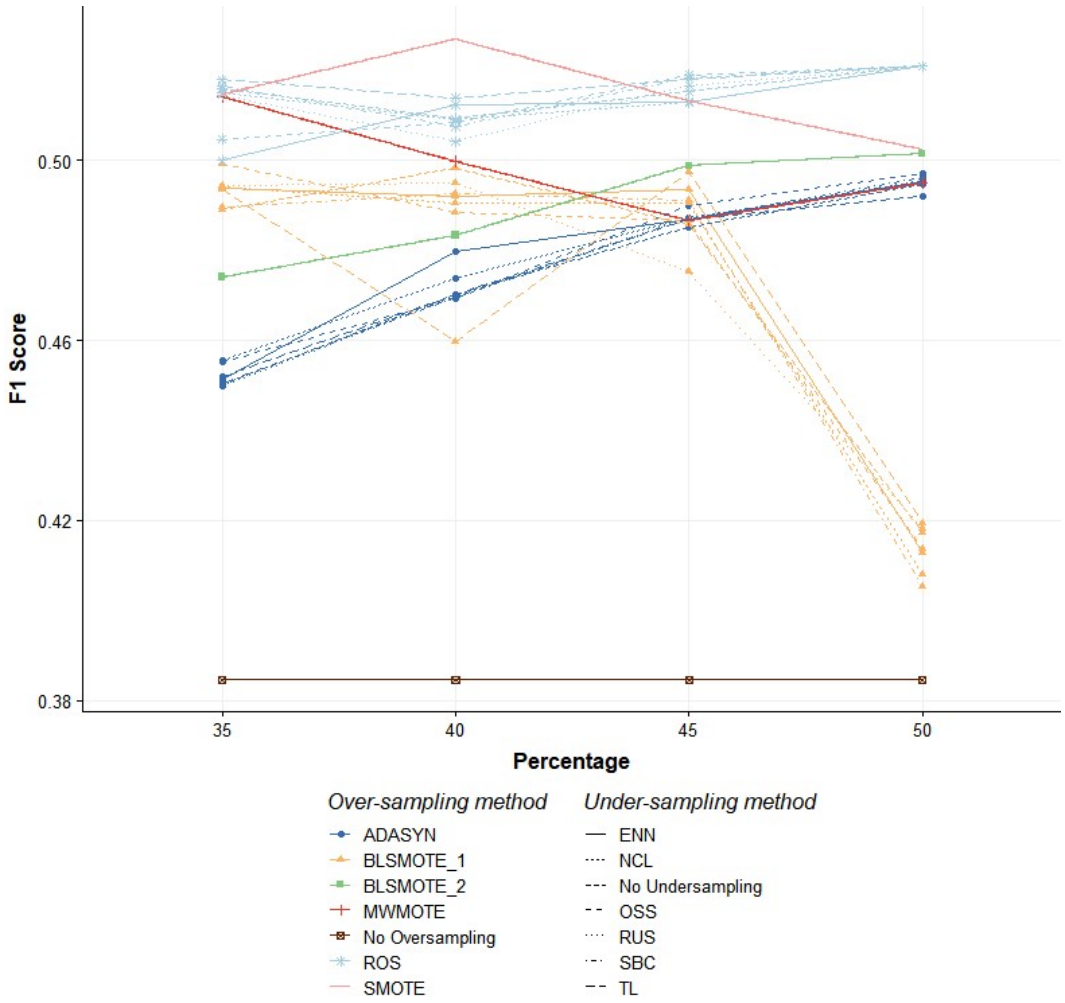


Figure 2.13: F1 score change of logistic model with different imbalance combinations

Table 2.5: Classifier performance by F1 score without using any re-sampling strategies with 10-fold cross validation

Model	F1 Score	sd_F1	AUC	sd_AUC
NaiveBayes	0.4079	0.0498	0.7374	0.0418
Logistic	0.3846	0.0477	0.7535	0.0324
SVR	0.0500	0.0218	0.7168	0.0363

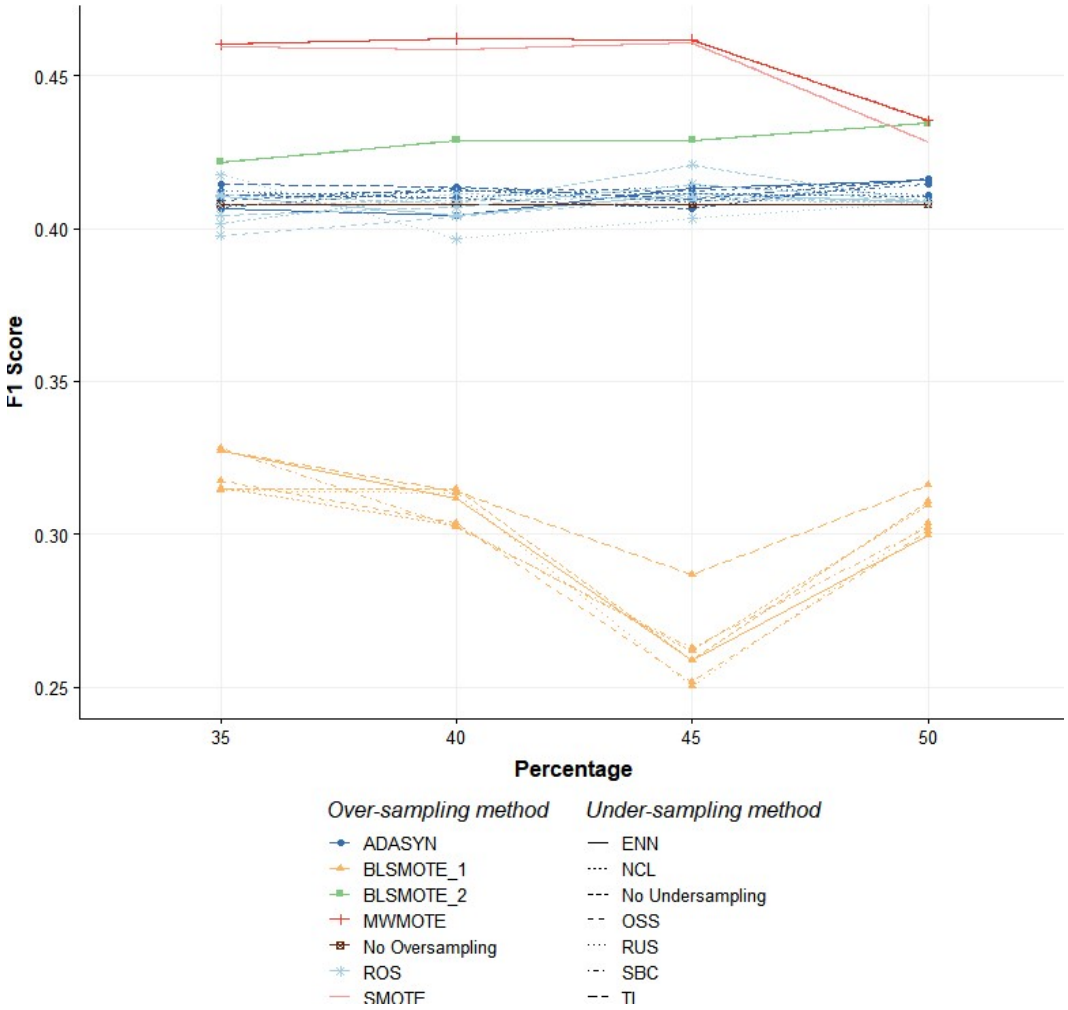


Figure 2.14: F1 score change of naïve Bayes model with different imbalance combinations

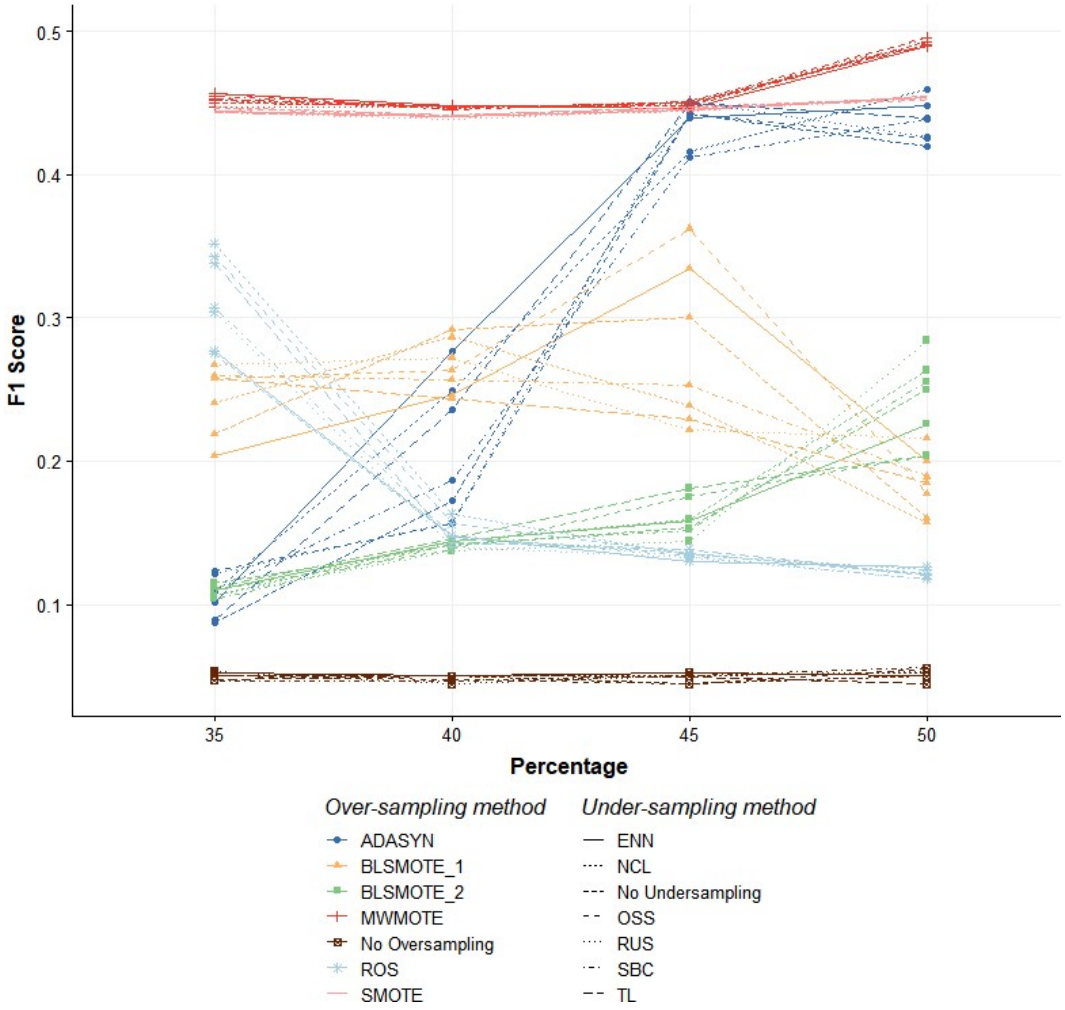


Figure 2.15: F1 score change of SVR model with different imbalance combinations

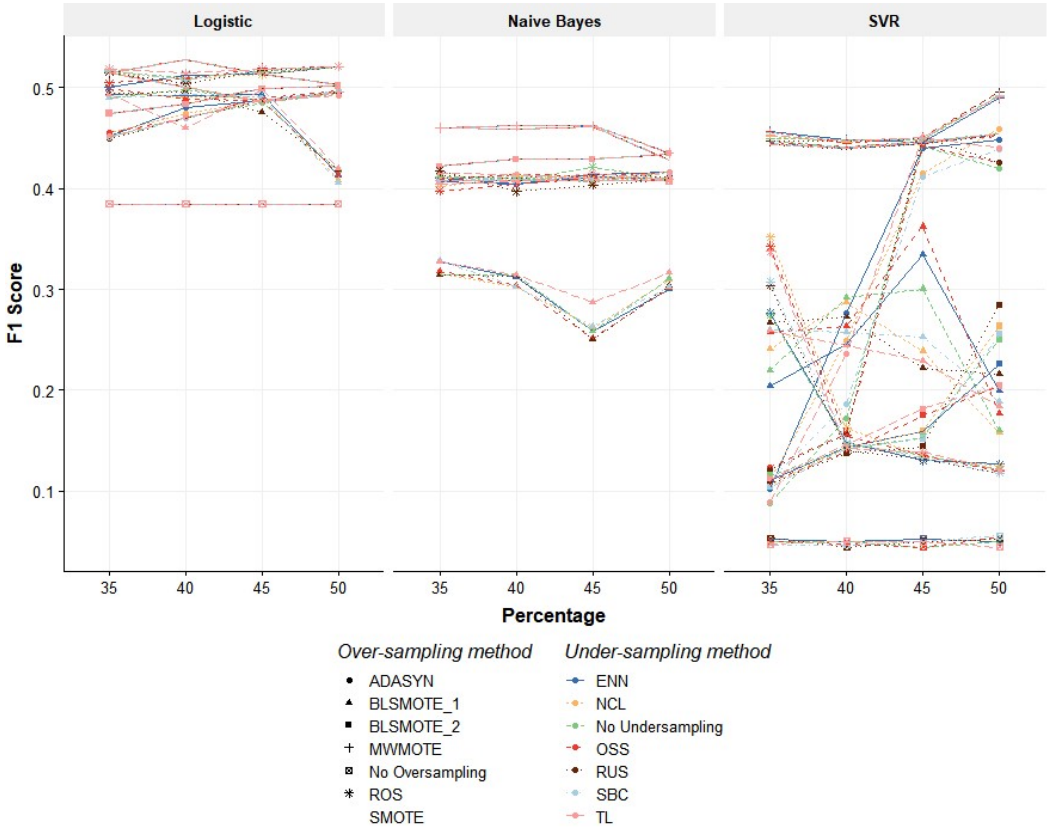


Figure 2.16: F1 score change of models with different imbalance combinations

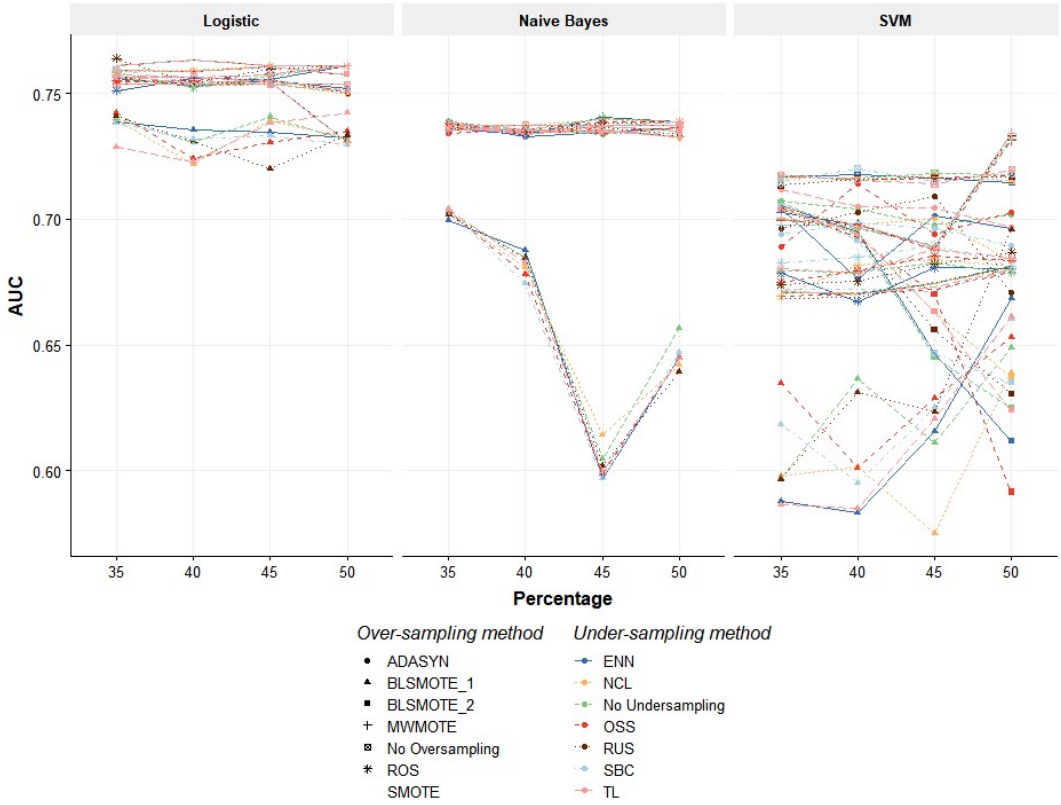


Figure 2.17: AUC change of models with different imbalance combinations

Table 2.6: P-values by De Long's non-parametric hypothesis tests to compare ROC curves

vs	Logit	NaiveBayes	SVR
Logit	.	0.00006*	0.00145*
NaiveBayes	0.00006*	.	0.31605
SVR	0.00145*	0.31605	.

To compare and differentiate the similarities between ROC curves with similar AUC values, we use De Long's non-parametric hypothesis testing for the best three models with highest F1 scores from each classifier as shown in Table 2.3. The respective p-values are given in Table 2.6.

2.6 Discussion

The results gained in our study show significant improvement in prediction of model performances. According to Table 2.3, the best re-sampling combination to get the highest F1 score is when training logistic regression model with over-sampling minority group with SMOTE till the new imbalance percentage is 40% and then under-sample the majority group using Edited Nearest Neighbors algorithm. This re-sampling combination yields a F1 score of 0.53 and area under-curve value of 0.76. This is a significant improvement over F1 score of 0.38 obtained by logistic regression models without re-sampling the dataset. On the other hand, SVR model fails on training using the dataset without any re-sampling but significantly improve with over-sampling the minority group with MWMOTE until the dataset is balanced and then cleaning the majority

group using One-Sided Selection under-sampling technique. The change in F1 scores with respect to the over-sampling percentages implies the necessity of identifying the optimum blend of re-sampling to the original data prior to decision making.

De Long's hypothesis testing can be used as a tool to distinguish similar ROC curves for situations as similar to this study. ROC curves for the best naïve Bayes and SVR models are similar and with the hypothesis testing it yields a p-value of 0.31605 failing to reject null hypothesis at 95% confidence level implying that the two ROC curves are similar, i.e. the performance of the two models over each cut-off point is approximately similar while the ROC curve for the best logistic regression model differ from the naïve Bayes model and the SVR model according to the hypothesis test p-values which are both less than 0.05.

Chapter 3

Bayesian Networks

A Bayesian Network (BN) or a Bayesian Belief Network can be considered as a probabilistic graphical model (Pearl, 1988; Parsons, 2011) representing conditional dependencies between variables with a directed acyclic graph (DAG). It uses Bayesian inference for probability computations. Using Bayesian Network, the joint probability distribution of random variables can be represented using conditional independence. Being graphical models, they contain a considerable portion that can be illustrated as a graph. There are many reasons to choose a Bayesian Network for a particular problem. First is the necessity of concrete class of models that are needed for evaluation. Second, use of probability theory as the foundation is acceptable, which is a classical and tried theory that has withstood time and has become one of the most fundamental concepts in sciences.

Most of modern AI application domains include uncertainty, where it needs to be dealt from the start in a principled way and with more explicit manner. Although we have a number of options such as decision trees, artificial neural networks and Markov networks to represent uncertainty, the ability of learning and representing directed causal relationships among variables in a dataset makes Bayesian Networks standout from the rest of models. Hence, Bayesian networks are extensively used in domains such as Biology (Needham et al., 2007), Medicine (Lucas, 2001), Chemistry (Hibbert and Armstrong, 2009), Physics (Rabiei et al., 2018) and in the sciences in general for the purpose of learning of causal networks.

To get a better understanding about the causal networks we can consider a hypothetical problem as shown in Figure 3.1. This toy domain can be considered as a causal model which describe how the style, price and location of a certain retail item influences the purchase choice of a customer. According to the example, style and location are independent while the price is depending on the style of the retail item. On the other hand purchase choice of a customer is entirely depending on the style, price and the location of the item.

Bayesian Networks in Modern Businesses

Modern businesses are interested in data driven decision making to achieve production goals as well as retaining their customer base. In literature we can find many examples where Bayesian Networks are used to find business

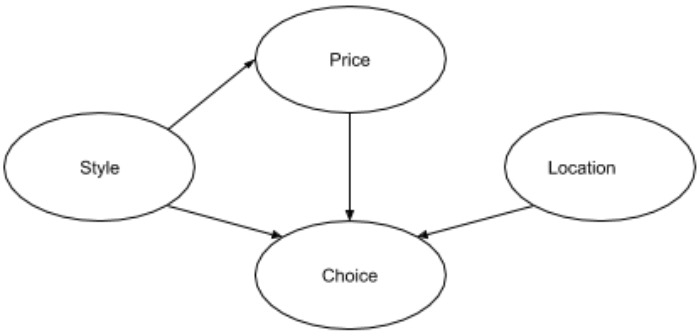


Figure 3.1: Hypothetical example of a Bayesian network modeling the choice of a customer purchasing an item from a shop

solutions. One such scenario is business process modeling where business process model and notation (BPMN) is achieved using Bayesian Network to identify dependencies between BPMN and additional background knowledge about a given business process (Bobek et al., 2013). An example scenario where Bayesian Networks can be used in business analytics is discussed by Ashcroft (2012). Human resource talent retention in firms is modeled using Bayesian Networks to take decisions on offering employment given the history of an applicant to the firm. Modeling business customer satisfaction using Bayesian Networks to identify key factors affecting the customer satisfaction is discussed by Chakraborty et al. (2016).

3.1 Causal Discovery

Bayesian Networks falls under a broad class of models which can be used to represent nested, acyclic statistical models of almost any kind of joint probability distribution. Their unique characteristic is their ability of obtaining directional relations between variables to represent cause and effect relationships compared to other graphical models such as Markov networks. Furthermore, BNs are capable of representing the independence between variables though the directed acyclic graph. The two features are closely related: direct effect of the causal relationships present are the independencies and the algorithms rely on their presence. But the reverse which is ability to present independencies does not guarantee models that wraps causal relationships. Decision trees can be taken as an example of such scenario. The ability to represent directional relationships is an important reason for our focus on BNs in this thesis.

3.1.1 Notations

The notations and symbols that are used to elaborate the rest of the chapter is summarized in Table 3.1. We might interchange the terms “node”, “variable”, “attribute”, and “feature” throughout this chapter and similarly for the terms “edge” and “arc”.

Table 3.1: Notations and symbols used in Chapter 3

D	Dataset
N	Number of instances in dataset i.e. $\|D\|$
$X, Y, Z, ...$	One dimensional variables
$x, y, z, ...$	Values of X, Y, Z
S, T	Sets
U	Universe; set of variables/nodes
n	Number of variables
E	Set of edges of a BN
T	Set of parameters of local pdfs for entire BN i.e. $p_{ijk}, i = 1, ..., n, j = 1, ..., q_i, k = 1, ..., r_i$
m	Number of edges of the BN
G	Directed acyclic graph (DAG) of a BN
B	Bayesian network, consists of DAG and parameters
$B(X)$	Markov blanket of variable
$N(X)$	Set of direct neighbors of variable X in Bayesian network
Pa_i	Set of parents of X_i
ps_{ij}	Set of values for value assignment j of each member of the set of parents Pa_i of X_i
r_i	Number of values of discrete variable X_i
q_i	Number of configurations of the set of parents of X_i
$c_1, c_2, ..., c_k$	Counts of a multinomial distribution with K bins
$p_1, p_2, ..., p_k$	Parameters (bin probabilities) of a multinomial distribution with K bins
$\alpha_i, \beta_j, \gamma_k$	Hyperparameters of multinomial distributions

3.2 Probability Distribution Representation

Apart from representation of causal relationships, BNs can be used to represent joint probability distributions (pdfs) concisely. This can be taken as the most common application today. Each variable attached in the network and their local pdfs enables this representation with the original purpose of quantifying the strength of the causal relationships picture in the BN with its structure. These local pdfs mathematically shows the behavior of a given variable under every possible value assigned by their parent(s). To describe this behavior, one might need a number of parameters exponential to the number of parents

(when the local pdfs are multinomial distributions. This is the most common choice for categorical variables) and since this value is usually smaller than the number of variables in the domain. Specifically, given the structure and the local probability distributions of a BN, the joint probability distribution of the domain of n variables $Pr(X_1, X_2, ..., X_n)$ can be calculated as

$$Pr(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} Pr(X_i|Pa_i)$$

where Pa_i are the parents of X_i in the Bayesian network whose structure is G. The conditional probabilities $Pr(X_i|Pa_i)$ defining the pdf of variable x_i given a value assignment of its parents Pa_i in the graph in this equation are exactly those local pdfs specified for each variable in the domain.

3.3 Assumptions for Learning the Causal Structure

Although BN model encodes a set of independencies that exist in the domain, their existence in actual population depends on the extent to which these assumptions hold. They are:

- **Causal Sufficiency Assumption:** No common unobserved (hidden or latent) variables exist in the domain that are parent of one or more

observed variables of the domain.

- **Markov Assumption:** Given a BN model B, any variable is independent of all its non-descendants in B, given its parents.

- **Faithfulness Assumption:** A BN graph G and a probability distribution P are faithful to one another iff every one and all independence relations valid in P are those entailed by the Markov assumption on G.

3.4 Learning Bayesian Networks

The most challenging task when using Bayesian networks is learning their structure. Research in this direction is essential because of the usefulness in many end-user applications as well as in many domains such as Biology, Medicine, Chemistry, Physics and in sciences in general where causal networks are beneficial.

An overview of the existing techniques that are used to learn Bayesian networks are presented in this section. In section subsection 3.4.1, the way of learning the parameters of BNs given the structure is described. In the subsequent sections the focus changes to the learning the structure of BNs. There are two major classes of Bayesian network structure learning algorithms. One is the "score" based structure learning where it chooses a BN based on how "well" it fits the given data and attempt to find the BN with optimal "score". This is discussed in subsection 3.4.2. In subsection 3.4.3, constraint based

structure learning algorithms are discussed. Last, hybrid structure learning algorithms are discussed in subsection 3.4.4 which uses constraint based strategy at the beginning to reduce space of candidate DAGs and a maximizing phase to use score based strategy in finding the optimal DAG in the space.

3.4.1 Learning the Parameters

Learning parameters from a fixed network structure is a well-known problem in statistics. In Bayesian approach, the problem can be stated as follows. A prior distribution is assumed over the parameters of the local pdfs before using the data. Moreover, the conjugacy of the prior is sensible. A conjugate prior is a distribution where the posterior over the parameters belong to the same family as the prior but with different hyperparameters.

In our thesis, we try to obtain network using network structure learning approaches given the data. Even though, as an example we shall present use of multinomials for the local pdfs on leaning the parameters.

For multinomial distributions, the conjugate prior comes from the Dirichlet family. Denoting the probability of each bin $p_{ijk}, k = 1, ..., r_i$ in the local pdf of variable X_i for the parent configuration pa_{ij}, the Dirichlet distribution over these parameters is expressed by:

$$Pr(p_{ij1}, p_{ij2}, ..., p_{ijr_i}|G) = Dir(\alpha_{ij1}, \alpha_{ij2}, ..., \alpha_{ijr_i}) = \Gamma(\alpha_{ij}) \prod_{k=1}^{r_i} \frac{{p_{ijk}}^{\alpha_{ijk}-1}}{\Gamma(\alpha_{ijk})}$$

where α_{ijk} are its hyper parameters and $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$. Assuming local and global parameter independence (Spiegelhalter and Lauritzen, 1990; Cooper and Herskovits, 1992; Heckerman et al., 1995), the distribution over the set of parameters p of the whole Bayesian network is,

$$Pr(p|G) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \Gamma(\alpha_{ij}) \prod_{k=1}^{r_i} \frac{{p_{ijk}}^{\alpha_{ijk}-1}}{\Gamma(\alpha_{ijk})}$$

Conditional on the dataset D, the posterior probability over the parameters is also a member of the Dirichlet family, since it is conjugate prior to the multinomial. It is,

$$Pr(p_{ij1}, p_{ij2}, ..., p_{ijr_i}|G, D) = Dir(N_{ij1} + \alpha_{ij1}, N_{ij2} + \alpha_{ij2}, ..., N_{ijr_i} + \alpha_{ijr_i})$$

and

$$Pr(p|G, D) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \Gamma(\alpha_{ij} + N_{ij}) \prod_{k=1}^{r_i} \frac{{p_k}^{\alpha_{ijk}+N_{ijk}-1}}{\Gamma(\alpha_{ijk} + N_{ijk})}$$

where N_{ijk} is the number of samples in the bin k of the pdf for X_i for parent configuration pa_{ij}. Note that N_{ijk} are the sufficient statistics of that pdf. Using this distribution to predict the value of any quantity $Q(X_1, X_2, ..., X_n)$ depending on the variables of the domain, one averages over all possible values

of the parameters, weighted by the posterior probability of each value,

$$Pr(Q(X_1, X_2, ..., X_n)|G, D) = \int Q(X_1, X_2, ..., X_n) Pr(p|G, D) dp.$$

The posterior estimate for p_{ijk} is,

$$\hat{p}_{ijk} = \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}}.$$

In scenarios where data are abundant, the hyperparameters are ignored and the fraction $\frac{N_{ijk}}{N_{ij}}$ is used instead.

3.4.2 Score Based Methods

Score based Bayesian network leaning method is one of the most popular method of constructing BNs from data, especially for the purpose of pdf estimation. In these methods, a function f is used to score a network (DAG) with respect to the given data and a search method is used to look for the network with the best score. Bayesian and non-Bayesian scoring metrics have been used in the literature (Neapolitan, 2003; Heckerman et al., 1995; de Campos, 2006). BN learning from data is an NP-hard problem (Chickering, 1996) and hence many heuristics and other metaheuristic methods such as genetic algorithms (Larranaga et al., 1996), simulated annealing (Chickering et al., 1995), tabu

search (Acid and de Campos, 2003) and ant colony optimization have been proposed to guide the search. The score is assigned to each prospect BN, typically one that measures how "well" that BN describes the dataset D. Assuming a structure G, its score is

$$Score(G, D) = Pr(G|D).$$

In other words, it is the posterior probability of G given the dataset. A score-based algorithm attempts to maximize this score. Computation of the above can be cast into a more convenient from by using Bayes' law,

$$Score(G, D) = Pr(G|D) = \frac{Pr(D|G)Pr(G)}{Pr(D)}.$$

To maximize this we need only maximize the numerator, since the denominator does not depend on G.There are several ways to assess $Pr(G)$ from prior information Heckerman (2008). To assume a uniform prior over structures, for this section we will ignore $Pr(G)$. To calculate $Pr(D|G)$, the Bayesian approach averages over all possible parameters, weighing each by their posterior probability:

$$Pr(D|G) = \int Pr(D|G, p)Pr(p|G)dp$$

Cooper and Herskovits (1992) first showed that for multinomial local pdfs this is,

$$Pr(D|G) = \prod_{i=1}^{n} \prod_{j=1}^{q_{ij}} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij+N_{ij}})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}.$$

where α_{ijk} and N_{ijk} are hyperparameter and counts for the pdf of X_i for parent configuration j. In the large sample limit the term $Pr(D|G,p)Pr(p|G)$ can be reasonably approximated as a multivariate Gaussian (Kass et al., 1988; Kass and Raftery, 1995). By approximating the mean of the Gaussian with the maximum-likelihood value $\hat{p}$ and ignoring terms that do not depend of the dataset size N, we end up with the **BIC score** approximation:

$$BICscore(G,D) = logPr(D|\hat{p},G) - \frac{d}{2}logN$$

which was first derived by Schwarz (1978). The term $\hat{p}$ is the set of maximum-likelihood estimates of the parameters $\mathbf{p}$ of the BN, while d is the number of free parameters of the multivariate Gaussian, *i.e.* its number of dimensions, which coincides with the number of free parameters of the multinomial local pdf's *i.e.* $d = \sum_{i=1}^{n} q_i(r_i - 1)$. BIC score is useful for the fact that it is not depending on the prior over the parameters, which makes popular in practice in cases where prior information is difficult to obtain or not available.

As described above, score based learning algorithms try to optimize the score, returning the structure G which maximizes it. This results considerable

problems since the space of all possible structures is at least exponential in the number of variables n: there are $n(n-1)/2$ possible undirected edges and $2^{n(n-1)/2}$ possible structures for every subset of these edges. There may be more than one orientation of the edges as well. A brute force approach of computing every possible structure is unrealistic but instead heuristic search algorithms are employed in many problems. In this thesis, we focus on Hill-climbing and tabu search which are two popular greedy search mathematical optimization algorithms used as score based BN structure leaning algorithms. The two algorithms are discussed below.

Hill-Climbing search (HC)

Hill-climbing search is a local greedy search that starts with an arbitrary solution to a problem. Afterwards it tries to improve the solution by making incremental changes to the original solution. This is done until the incremental change done to the previous solution does not improve the solution anymore. Hill-climbing search can be better than other advanced search algorithms such as simulated annealing and tabu search in terms of time to perform the search. The simplicity in the algorithm makes it a popular choice amongst optimizing algorithms. The pseudocode for the algorithm that construct BN from a dataset D from hill-climbing search is as follows:

$B = BIChillclimb(D)$

1. $E \leftarrow \emptyset$

2. $T \leftarrow ProbabilityTables(E, D)$; This estimates the parameters of local pdfs given BN structure.

3. $B \leftarrow \langle U, E, T \rangle$

4. $score \leftarrow -\infty$

5. do:

   (a) $maxscore \leftarrow score$

   (b) for each attribute pair (X, Y) do

   (c) for each $E' \in \{E \cup \{X \rightarrow Y\},$
   $E - \{X \rightarrow Y\},$
   $E - \{X \rightarrow Y\} \cup \{Y \rightarrow X\}\}$

   (d) $T' \leftarrow ProbabilityTables(E', D)$

   (e) $B' \leftarrow \langle U, E', T' \rangle$

   (f) $newscore \leftarrow BICscore(B', D)$

   (g) if $newscore > score$ then
   $B \leftarrow B'$
   $score \leftarrow newscore$

6. while $score > maxscore$

7. Return B

Tabu search

Tabu search is a metaheuristic greedy search algorithm that uses a local/ neighborhood search procedure to move from one potential solution to the other. To avoid any solution becoming stuck in a local optimum, tabu search explores the neighborhood carefully during the solution search procedure. Similar to simulated annealing, tabu search is capable of doing down-hill moves. Tabu search specifically maintains a memory structure (tabu tenure) that keeps track on solutions that it obtained in previous iterations. This prevents the search process taking a non-improving move from a local optima. A simpler version of tabu search algorithm is as follows:

problem : maximize objective f

1. Randomly select an initial solution i in the search space S, and set $i^* = i$ and $k = 0$, where i^* is the best solution so far, and k is the iteration counter;

2. Set $k = k + 1$ and generate the subset V of the *admissible* neighborhood solutions of i (non- tabu/allowed)

3. Choose the best j in V and set $i = j$

4. if $f(i) > f(i^*)$, then set $i^* = i$;

5. update the *tabu* and the aspiration conditions;

6. if a stopping condition is met → stop

   else go to step 2

3.4.3 Constraint Based Methods

The main goal of constraint based structure learning is to recover a structure that best captures the independences in a given dataset (Spirtes et al., 2000; Neapolitan et al., 2004). These constraint based BN learning algorithms use conditional independence tests to find out conditional independence constraints from data. The work by Verma and Pearl (1991) on Inductive Causation (IC) algorithm provides a framework for learning the structure of BN using conditional independence tests. however, the problem with this IC algorithm is that it cannot be applied to any real world problem due to the exponential number of conditional independence relationships that would need to be examined going forward with the algorithm. To over come this issue many other algorithms such as PC (Spirtes et al., 2000), grow-shrink(GS) (Margaritis, 2003) and Incremental Association Markov Blanket (IAMB) (Tsamardinos et al., 2003) have been developed. In our thesis, we consider grow-Shrink Algorithm and Incremental Association Markov Blanket (IAMB) used to learn BN from given data.

Grow-Shrink algorithm (GS)

Grow Shrink algorithm consist of two phases namely a growing phase and a shrinking phase. It is based on the Grow-Shrink Markov blanket algorithm (Margaritis, 2003) and it is a simple forward selection Markov blanket detection algorithm. The GS algorithm is as follows:

1. [**Compute Markov Blankets**]

   for all $X \in U$, compute the Markov blanket B(X)

2. [**Compute Graph Structure**]

   For all $X \in U$ and $Y \in B(X)$, determine Y to be a direct neighbor of X if X and Y are dependent given S for all $S \subseteq T$, where T is the smaller of $B(X) - \{Y\}$ and $B(Y) - \{X\}$

3. [**Orient Edges**]

   For all $X \in U$ and $Y \in N(X)$, orient $Y \rightarrow X$ if there exists a variable $Z \in N(X) - N(Y) - \{Y\}$ such that Y and Z are dependent given $S \cup \{X\}$ for all $S \subseteq T$, where T is the smaller of $B(Y) - \{X, Z\}$ and $B(Z) - \{X, Y\}$

4. [**Remove Cycles**]

   Do the following while there exist cycles in the graph:

   (a) Compute the set edges $C = \{X \rightarrow Y$ such that $X \rightarrow Y$ is part of a cycle $\}$

(b) Remove from the current graph the edge in C that is part of the greatest number of cycles, and put it in R

5. [**Reverse Edges**]

   Insert each edge from R in the graph in reverse order of removal in step 4, reversed

6. [**Propagate Directions**]

   For all $X \in U$ and $Y \in N(X)$ such that neither $Y \to X$ nor $X \to Y$, execute the following rule until it no longer applies: If there exist a directed path from X to Y, orient $X \to Y$

Incremental Association Markov Blanket (IAMB)

This is a two-phase selection scheme and can be considered as a variant of Grow-Shrink algorithm. Although this algorithm is efficient on time, it is considered to be performing poor on data efficiency (Schluter, 2011). This algorithm consist of two phases, a forward phase and a backward phase. The approach can be understood simply as in forward phase estimates of the Markov blankets are kept in a set and in backward phase the false positives are identified and removed from the set. Detailed IAMB algorithm and it's variants can be found in (Tsamardinos et al., 2003)

3.4.4 Hybrid Methods

Hybrid structure learning algorithms on the other hand tries to retain the best from score based structure learning algorithms and constraint based structure learning algorithms. Usually these algorithms starts with a skeleton BN from constraint based approach and then constraint on DAGs considered in the scoring phase. We use max-min Hill climbing (MMHC) (Brown et al., 2004; Tsamardinos et al., 2006) and more generalized 2-phase Restricted Maximization(RSMAX2) (Scutari, 2009) in our thesis on learning the BN structure using data.

Max-Min Hill Climbing algorithm (MMHC)

This algorithm is based on Max-Min Parents and Children (MMPC) (Tsamardinos et al., 2003) local search constraint based algorithm and Hill Climbing score based learning algorithm. MMPC is used by MMHC to rebuild the skeleton of the BN before a constraint greedy search is performed to align the edges. MMHC first identifies the parents and children set of each variable, then performs a greedy hill climbing search in the BN space. The search begins with an empty graph and then edge addition and deletion and directional changes leading to the highest increment of network score is retained and continued recursively until the highest score is obtained.

2-phase Restricted Maximization (RSMAX2)

Unlike MMHC, RSMAX2 is more generalized and we can input both constraint based methods and score based methods for learning instead of using a fixed combination of constraint based and score based learning algorithms. By default Grow Shrink algorithm and Hill Climb algorithm is used in RSMAX2.

Bayesian networks can be mainly divided in two types as discrete BNs and continuous BNs. Discrete BNs contain discrete data often with categorical variables and multinomial distribution is used to represent the conditional probabilities of nodes. In continuous BNs all the variables are continuous and the mostly considered Gaussian BNs (GBN) assumes that all the nodes are following a normal distribution. Furthermore, GBNs assume that the root nodes (parents) are described by their marginal distributions and each node has a variance that is specific to that node and does not depend on the value of the parents. Hence the joint distribution of all nodes is a multivariate normal distribution. In scenarios where both discrete and continuous nodes present, hybrid BNs can be used but the structure learning is yet to be implemented given it's flexibility towards real world problems. Continuous BNs perform better than hybrid BNs when there is only few observations to learn the structure. It also yields greater accuracy than discretization for continuous variables but discretization on the other hand yields better BNs compared to misspecified distributions of nodes and assumptions on the conditional probabilities. In this thesis, we focus on discrete BNs than other types of BNs

for structure learning.

3.5 Data Analysis - Bayesian Network Structure Learning

In this section, we try to obtain a causal network that most accurately represents mobile app user retention that is obtained from a local mobile app developing company. The dataset consist of the number of times each feature in a mobile App for a given amount of time and the final status of the customer, i.e. the app user is retained or left the mobile app. There are 27 features available in the mobile app for the user to access and for each time that user access that feature, it is recorded. This had end up with a dataset with 27 features having numerical counts but given the sparsity of the dataset and lack of retained customers, discretization of counts of each app feature is challenging. Figure 3.2 shows the distribution of in-app feature usage distribution. To discretize the features, although there are several standard methods of discretizing a variable by binning, we choose to make every in-app feature in to two discrete values 0 or 1 given the sparsity of the features. Value 0 is assigned when there are no usage in a given app feature by a customer and value 1 is given when there is alteast one record of using a given app feature.

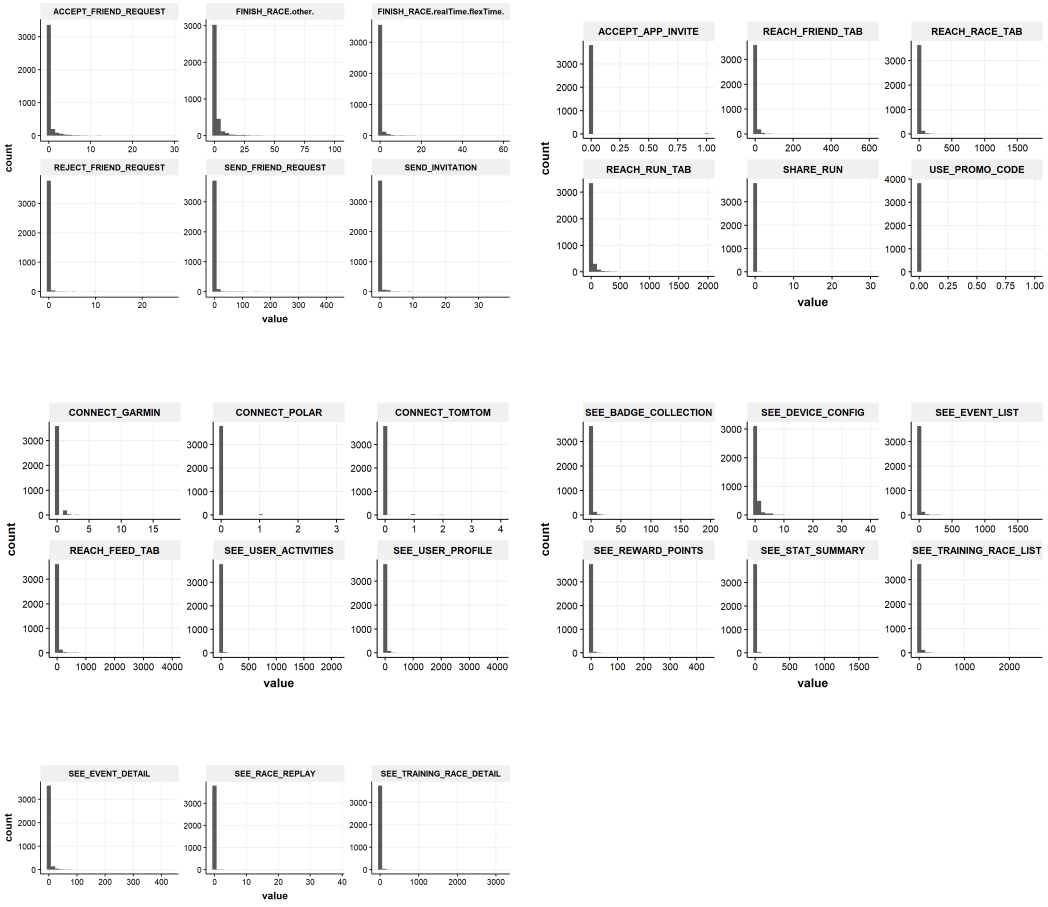


Figure 3.2: Distribution of mobile in-app feature usage by app users

3.5.1 Bayesian Structure learning on Mobile App Data

We use score based, constraint based and hybrid structure learning algorithms discussed in subsection 3.4.2, subsection 3.4.3 and subsection 3.4.4 to learn the structure of the Bayesian networks by the discretized data. The final outcome of causal structure depends on the structure learning algorithm. Using the

`bnlearn` R package (Scutari, 2009), we obtain BNs from the mobile app user dataset. The standard BN structure learning algorithms present in `bnleanr` is used and the input dataset and over-sampling/under-sampling percentages are changed accordingly.

In score based Bayesian structure learning algorithms, we used Hill-climbing algorithm and tabu search to obtain two Bayesian networks on our mobile app dataset as given in Figure 3.3 and Figure 3.4. These two Bayesian networks do not have any disconnected nodes and the respective BNs give different node structures representing the dependability of in-app features towards the customer retention (node "`Retained`"). But in both BNs, the immediate parents of the node "`Retained`" are "`ACCEPT_FRIEND_REQUEST`", "`FINISH_RACE.other.`" and "`FINISH_RACE.realRime.flexTime`". The "`Retained`" node is also a parent of "`SEE_EVENT_DETAIL`" node.

On the other hand, Bayesian networks obtained by Grow-Shrink and IAMB constraint based learning algorithms show many disconnected nodes with some subcycle graphs as shown in Figure 3.5 and Figure 3.6. According to the two constraint based learning algorithms, the node "`Retained`" has no parents but it is the parent for other nodes in the BN structure.

Bayesian structures learned from the two hybrid learning algorithms, max-min hill-climbing and RSMAX2 algorithms also show disconnected nodes from the main Bayesian network as shown in Figure 3.7 and Figure 3.8.

When comparing network scores calculated between Bayesian networks

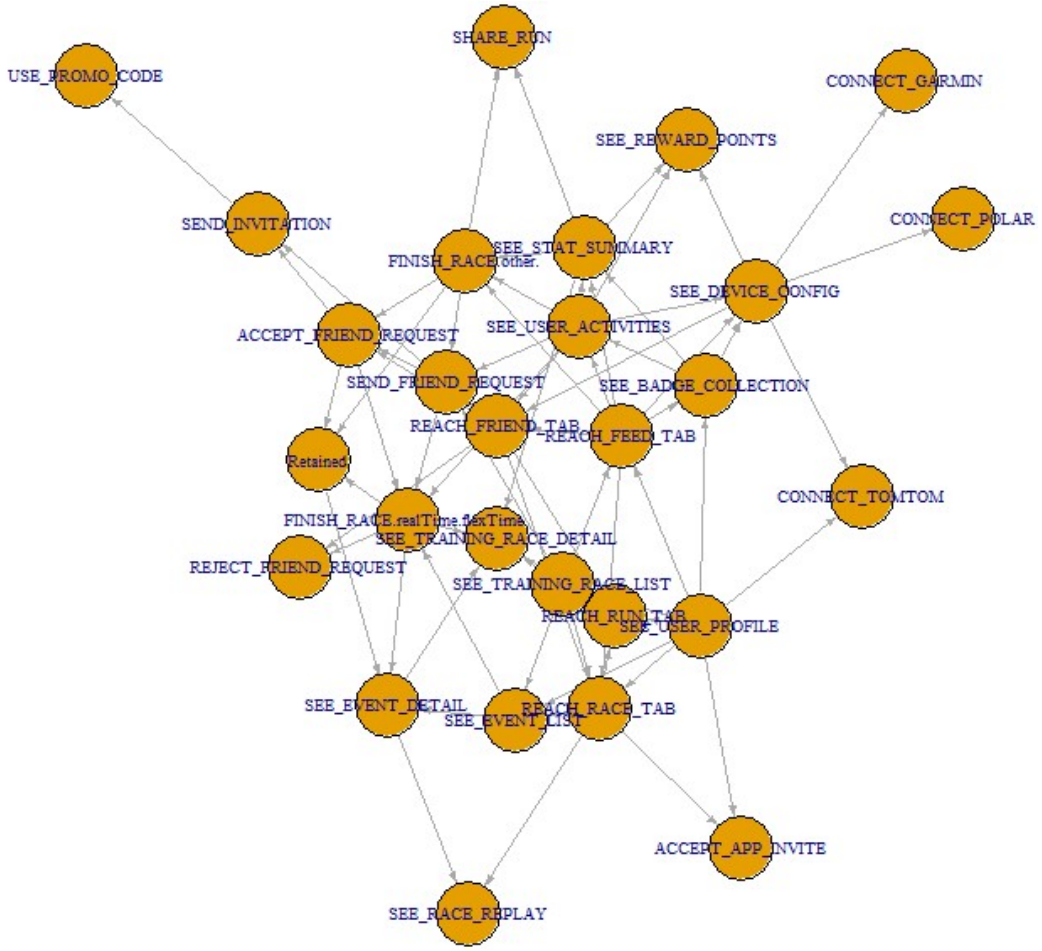


Figure 3.3: Bayesian network by Hill-climbing score based learning algorithm

Table 3.2: Network scores for the Bayesian networks

Model	BIC	AIC	logLik
GS	.	.	.
IAMB	.	.	.
HC	-27850.35	-27259.67	-27070.67
Tabu	-27837.14	-27233.96	-27040.96
MMHC	-28231.59	-27897.19	-27790.19
RSMAX2	-28936.61	-28677.20	-28594.20

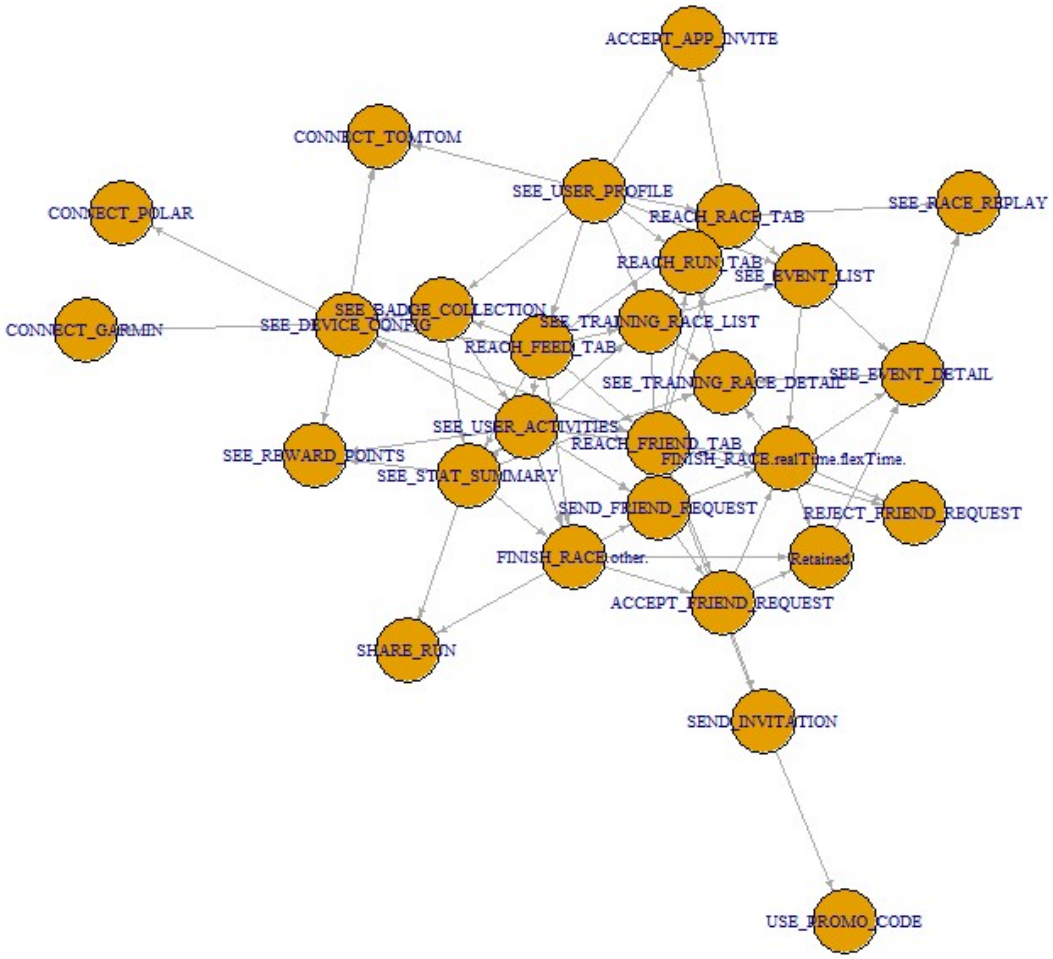


Figure 3.4: Bayesian network by tabu search score based based learning algorithm

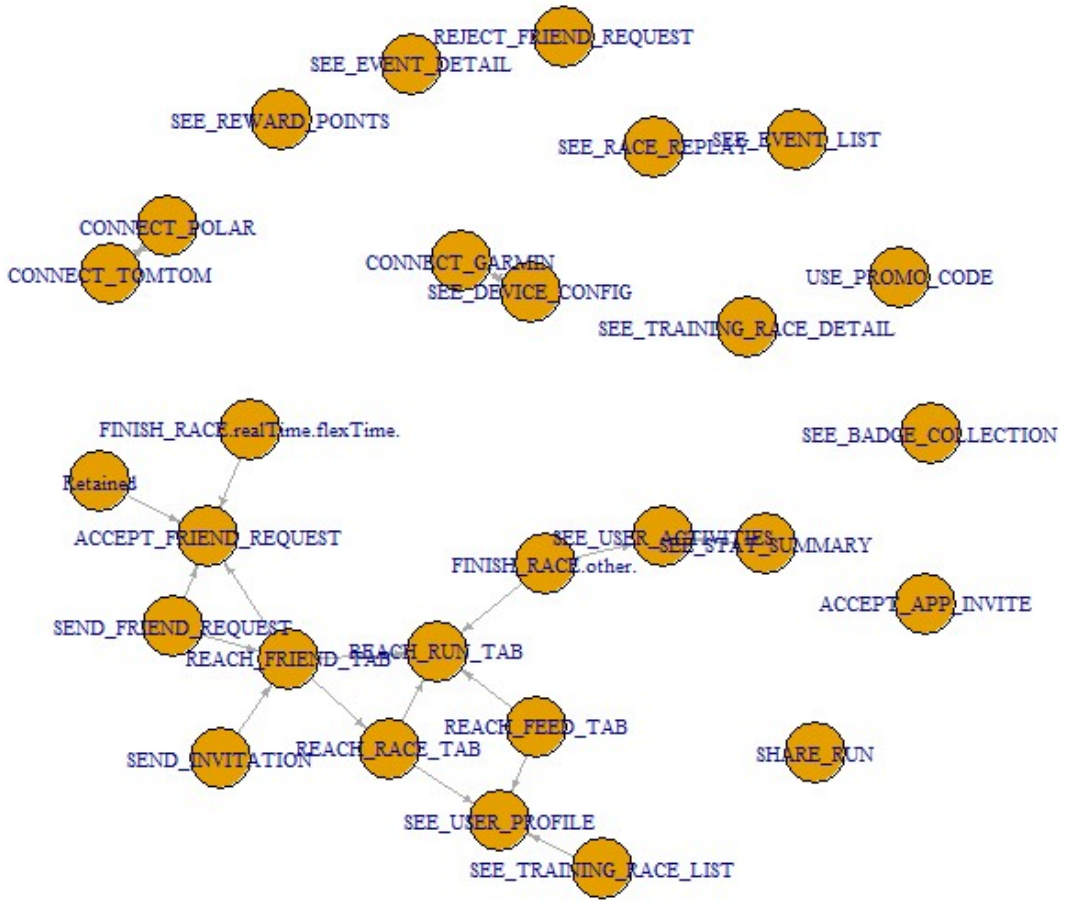


Figure 3.5: Bayesian network by Grow-Shrink constraint based learning algorithm

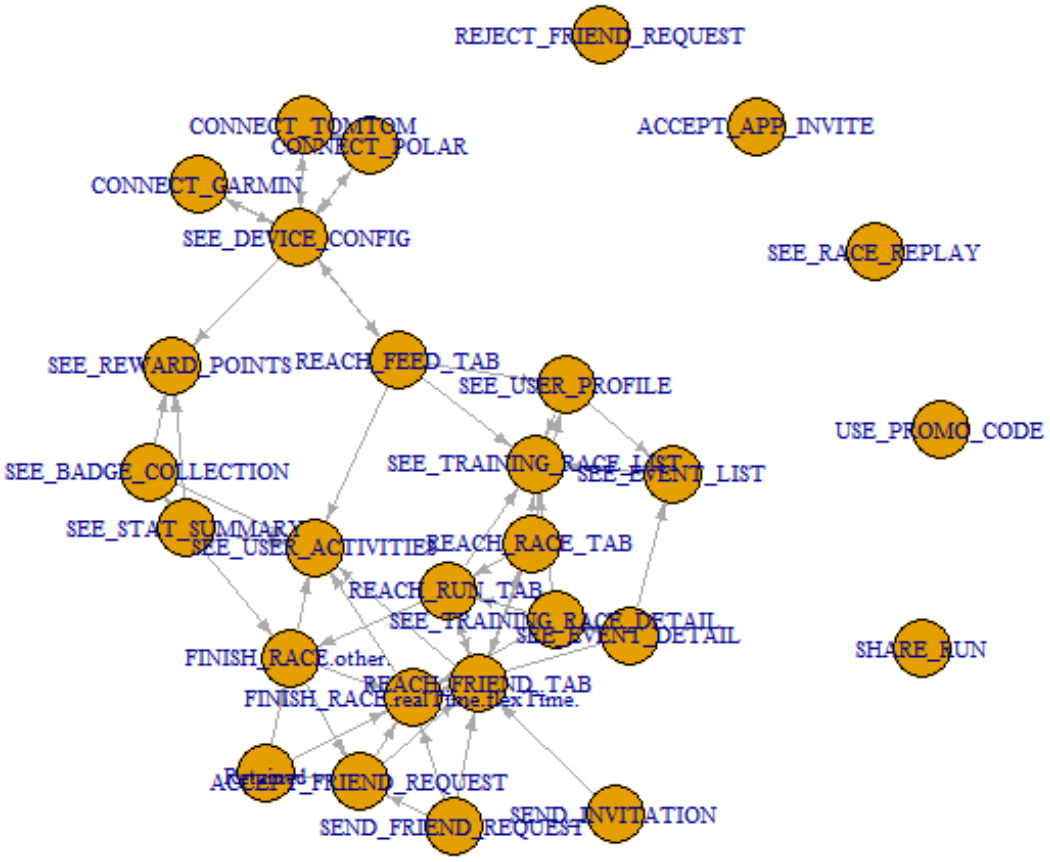


Figure 3.6: Bayesian network by IAMB constraint based learning algorithm

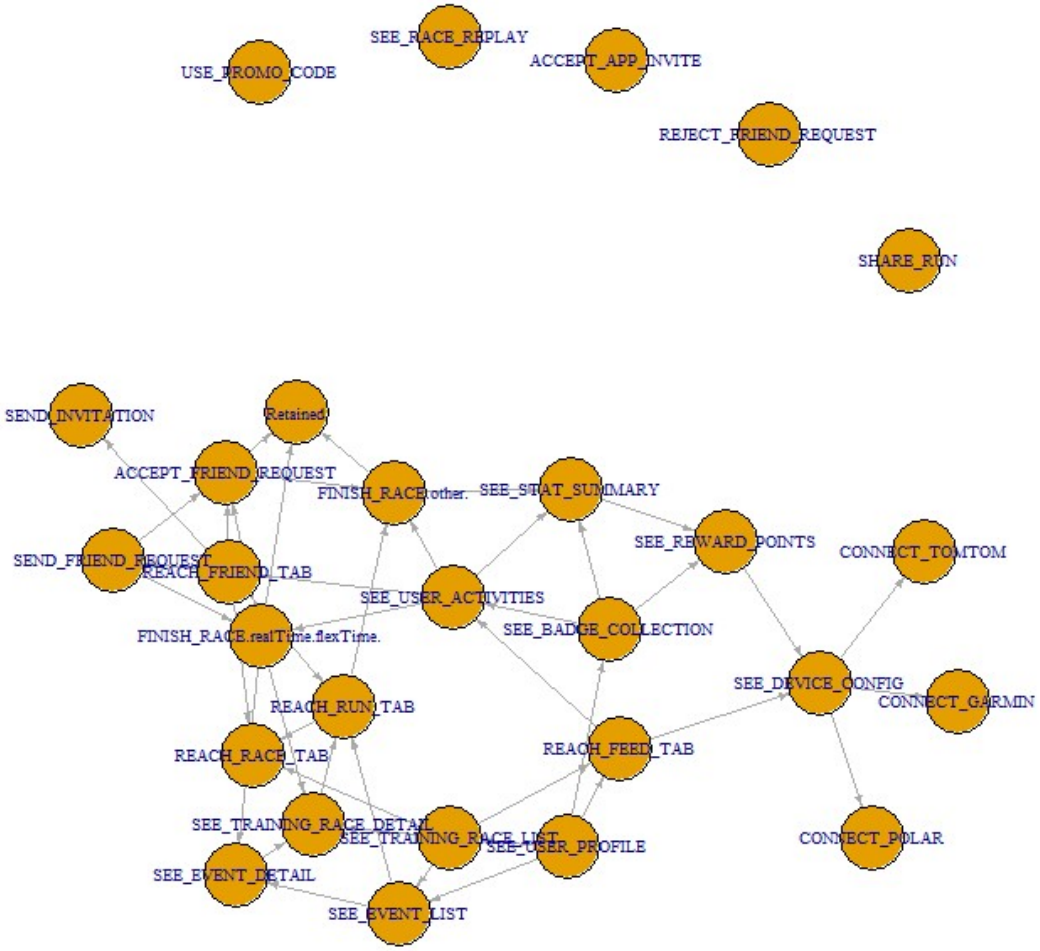


Figure 3.7: Bayesian network by max-min Hill-climbing hybrid learning algorithm

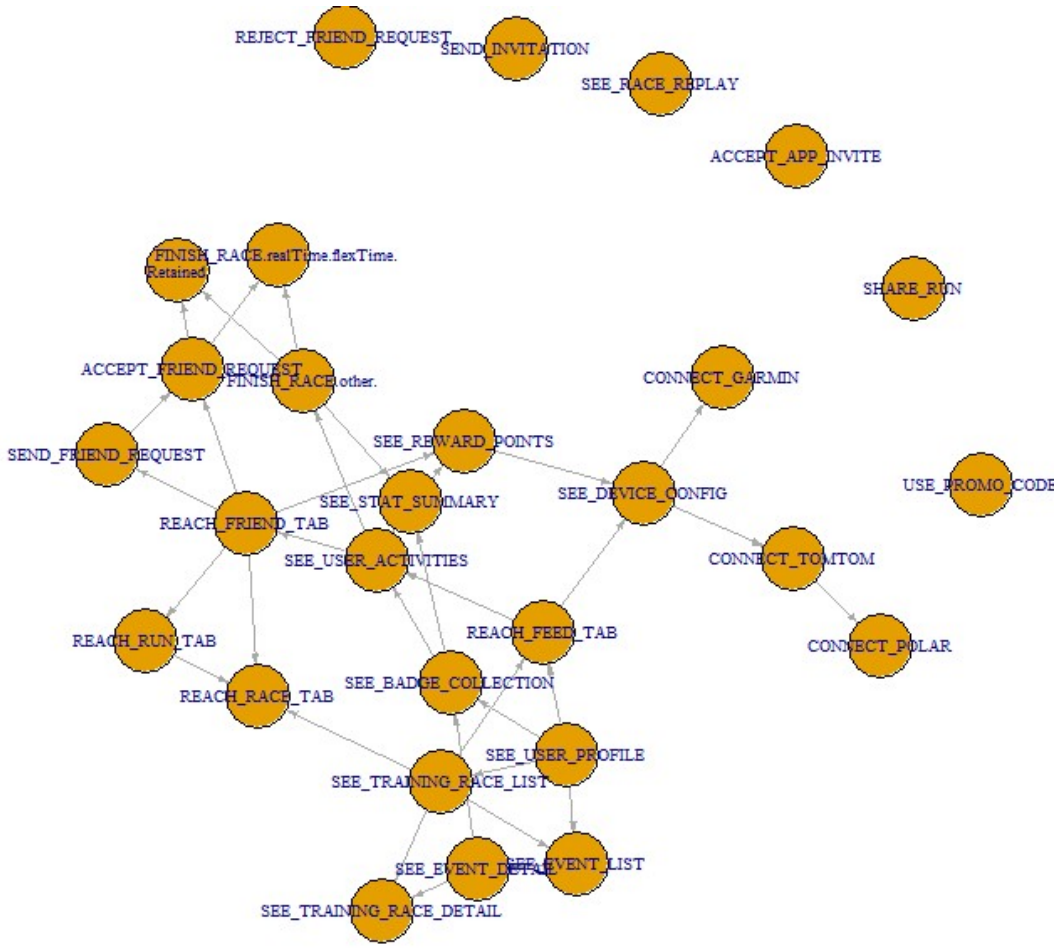


Figure 3.8: Bayesian network by RSMAX2 hybrid learning algorithm

learned as shown in Table 3.2, we can observe that BIC (Schwarz, 1978), AIC (Akaike, 1974) and log-likelihood values are lowest for the BN learned using RSMAX2. Obtaining of the network scores of the two constraint based Grow-Shrink and IAMB algorithms were failed due to their resultant BNs are partially directed.

3.5.2 Bayesian Structure Learning from Imbalance Data

In our thesis, we study about improvement of performance of selected classifiers in chapter 2. We use several over-sampling (subsection 2.2.1) and under-sampling techniques (subsection 2.2.2) to treat the imbalance nature of data and the resultant dataset is then used to model using selected classifiers. The results shows that the re-sampling techniques significantly improve the classifier performance. Our dataset of interest, the mobile app user retention dataset is a unbalanced dataset and it is worth exploring the network structure performance with respect to the same re-sampling techniques that are used to balance the target variable, i.e. mobile app user retention status ("`Retained`"). We use the same combinations of re-sampling techniques to treat the "`Retained`" variable as explored in subsection 2.5.1. In section 3.5.2, we describe the simulation study conducted in order to assess the effect of re-sampling techniques towards the BN performance.

Simulation Study

First, we will explore all combinations of re-sampling techniques to obtain the best network score. We will then compare BNs with best network scores versus the BNs that we obtained in subsection 3.5.1. According to Figure 3.9, Figure 3.10 and Figure 3.11, we can observe that when the over-sampling percentage of the minority group increases, the network scores (BIC, AIC and log-likelihood) also increases. Furthermore, we can see that in re-sampling scenarios where the minority group is not over-sampled, the network score did not improve with respect to the under-sampling of majority group.

Results

By the simulation study described in section 3.5.2, we can see BN structure model improvement with respect to the amount of over-sampling done to the minority group. In Table 3.3, we can see that the best BIC score of `-56456.52` is obtained by using the RSMAX2 with using Borderline-SMOTE 1 (BLSMOTE_1) to over-sample until the dataset is balanced and then clean the majority group using One-Sided Selection (OSS) rule. On the other hand, other BN structure learning algorithms like MMHC, HC and Tabu shows lesser network scores compared to score gained by RSMAX2.

It is worth to mention that, according to section 3.5.2, similar network scores can be seen between re-sampling techniques for each BN learning algorithm. The complete network scores for every re-sampling combination is in Table B.1. To

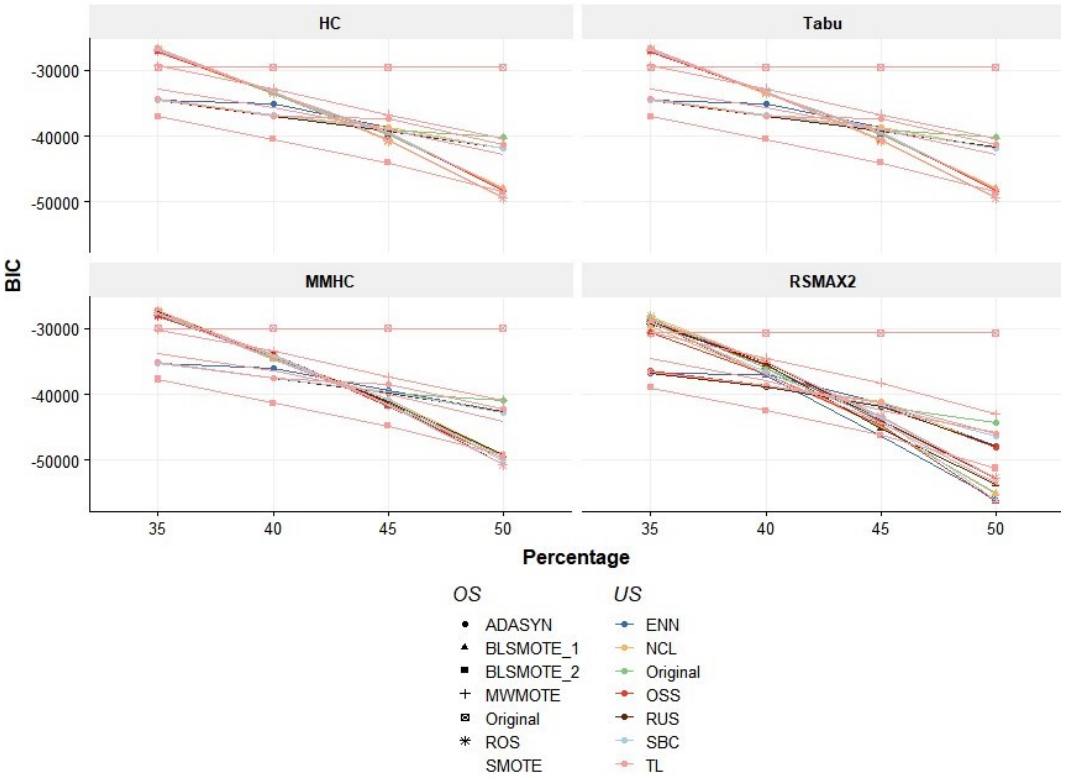


Figure 3.9: BIC scores of Bayesian networks obtained with re-sampling techniques

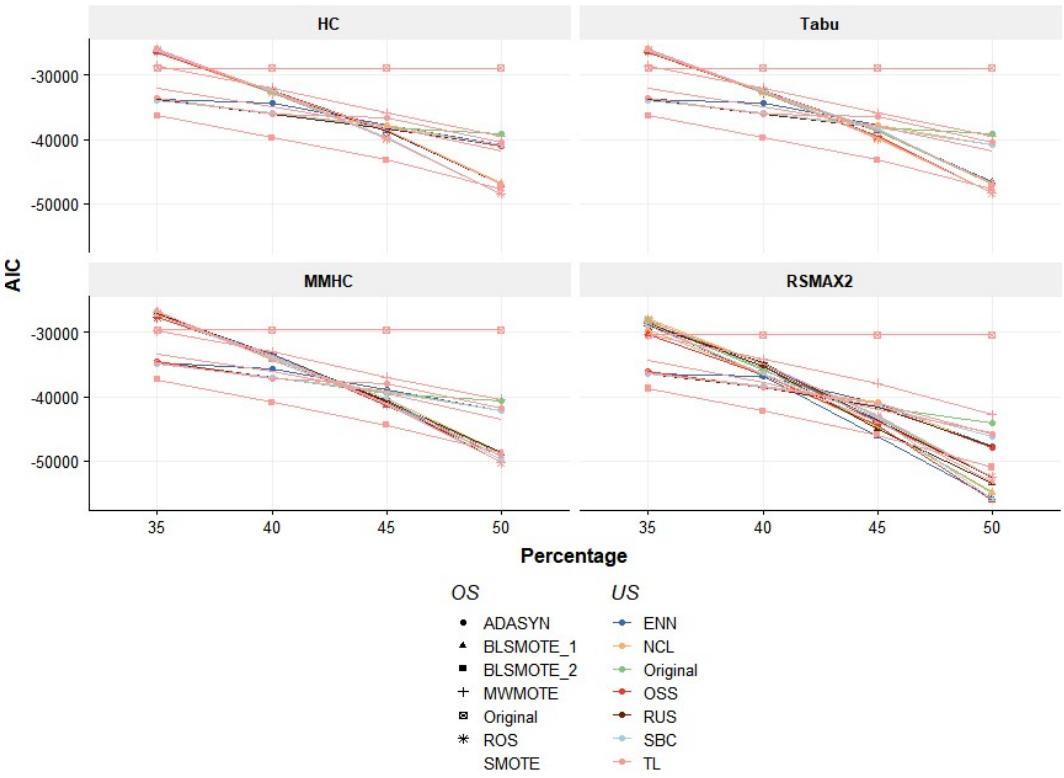


Figure 3.10: AIC scores of Bayesian networks obtained with re-sampling techniques

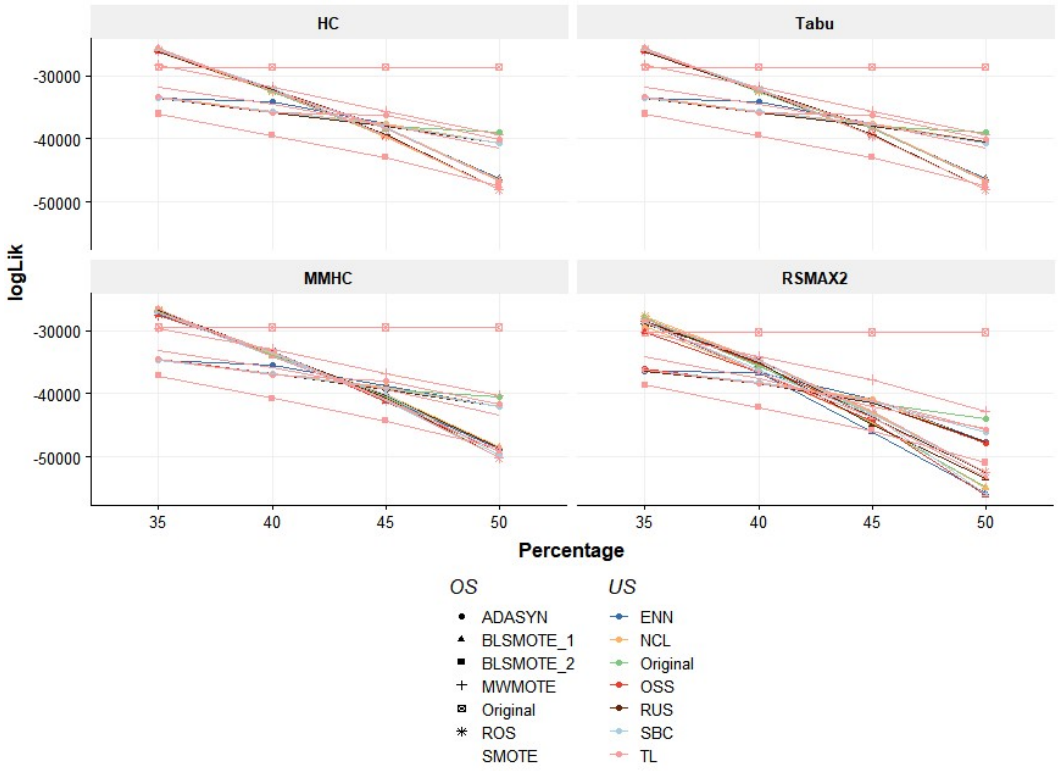


Figure 3.11: Log-Likelihood scores of Bayesian networks obtained with re-sampling techniques

Table 3.3: First two Bayesian networks with best network scores for each learning algorithm

Percentage	OS	US	Model	BIC	AIC	logLik
50	BLSMOTE_1	OSS	RSMAX2	-56456.52	-56248.21	-56186.21
50	BLSMOTE_1	SBC	RSMAX2	-56139.00	-55940.77	-55881.77
50	ROS	Original	MMHC	-50788.49	-50307.71	-50164.71
50	ROS	RUS	MMHC	-50788.49	-50307.71	-50164.71
50	ROS	Original	HC	-49440.75	-48415.31	-48110.31
50	ROS	RUS	HC	-49440.75	-48415.31	-48110.31
50	ROS	Original	Tabu	-49428.99	-48396.83	-48089.83
50	ROS	RUS	Tabu	-49428.99	-48396.83	-48089.83

understand about the magnitude of differences between the BNs obtained by re-sampling techniques, we can use Structural Hamming Distance (SHD) (de Jongh and Druzdzel, 2009) which is based on versions of hamming distance on Bayesian networks proposed by Tsamardinos et al. (2006), Acid and de Campos (2003) and Perrier et al. (2008) . In simple terms, structural hamming distance implies how many number of edge insertions, deletions or flips needed to obtain a graph from a given graph. The lower the SHD, the closer the two BNs in terms of structure. The structural hamming distance between the two RSMAX2 BNs with best network scores is 16. On the other hand, the other three pairs of network structures of using similar learning algorithms yield SHD value 0 when compared with their different re-sampling combinations as shown in Table 3.3. This implies that despite the different under-sampling technique used, the Bayesian structures does not improve by means of network score.

The respective BNs yielded from the best performing re-sampling combination of each BN structure learning algorithms are given in Figure 3.12,

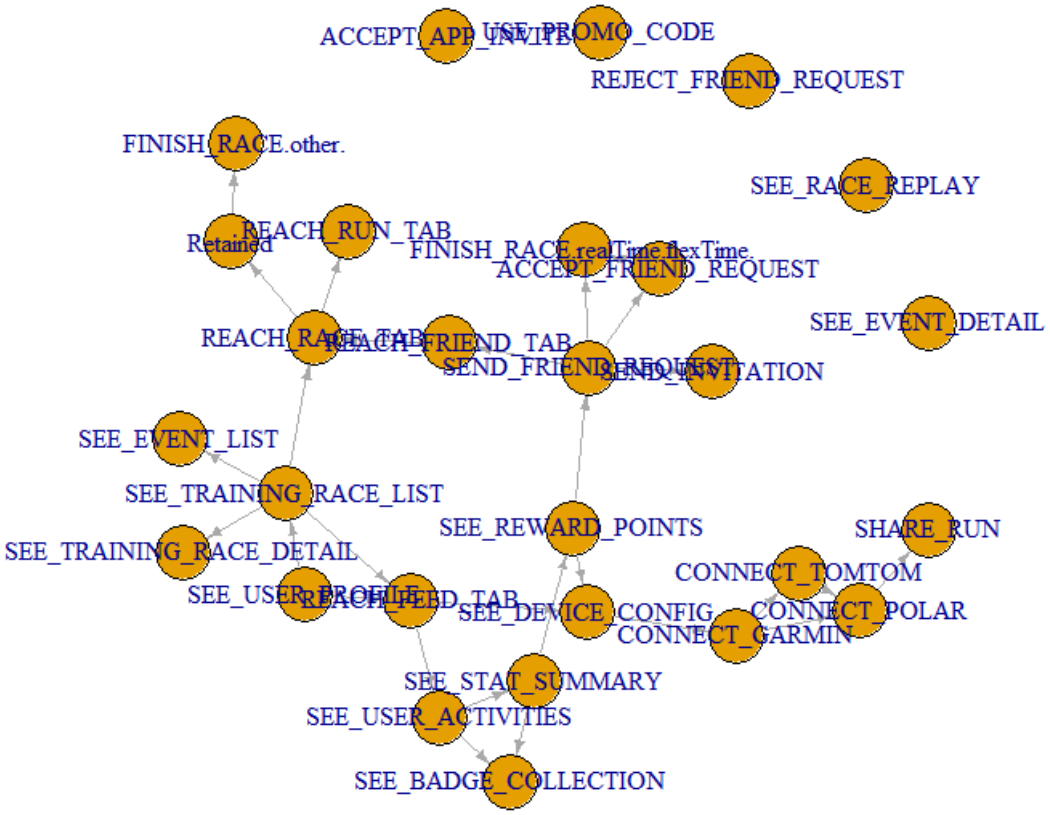


Figure 3.12: Best Bayesian network obtained by RSMAX2 learning algorithm

Figure 3.13, Figure 3.14 and Figure 3.15 respectively.

3.6 Discussion

From the simulation study and results, we can see a general improvement in BN structures in terms of network scores. Almost all structure learning algorithms tend to perform better when the minority group is over-sampled until the dependent variable ("`Retained`") is balanced. By observing the

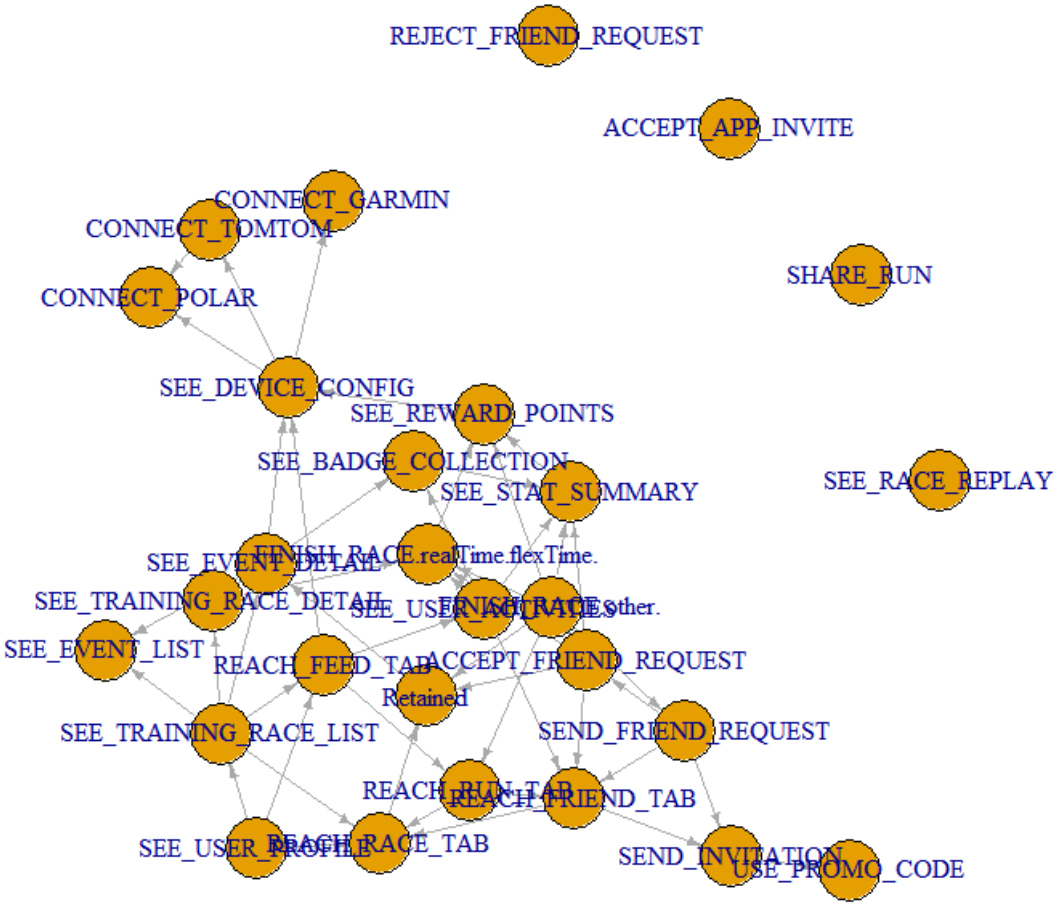


Figure 3.13: Best Bayesian network obtained by MMHC learning algorithm

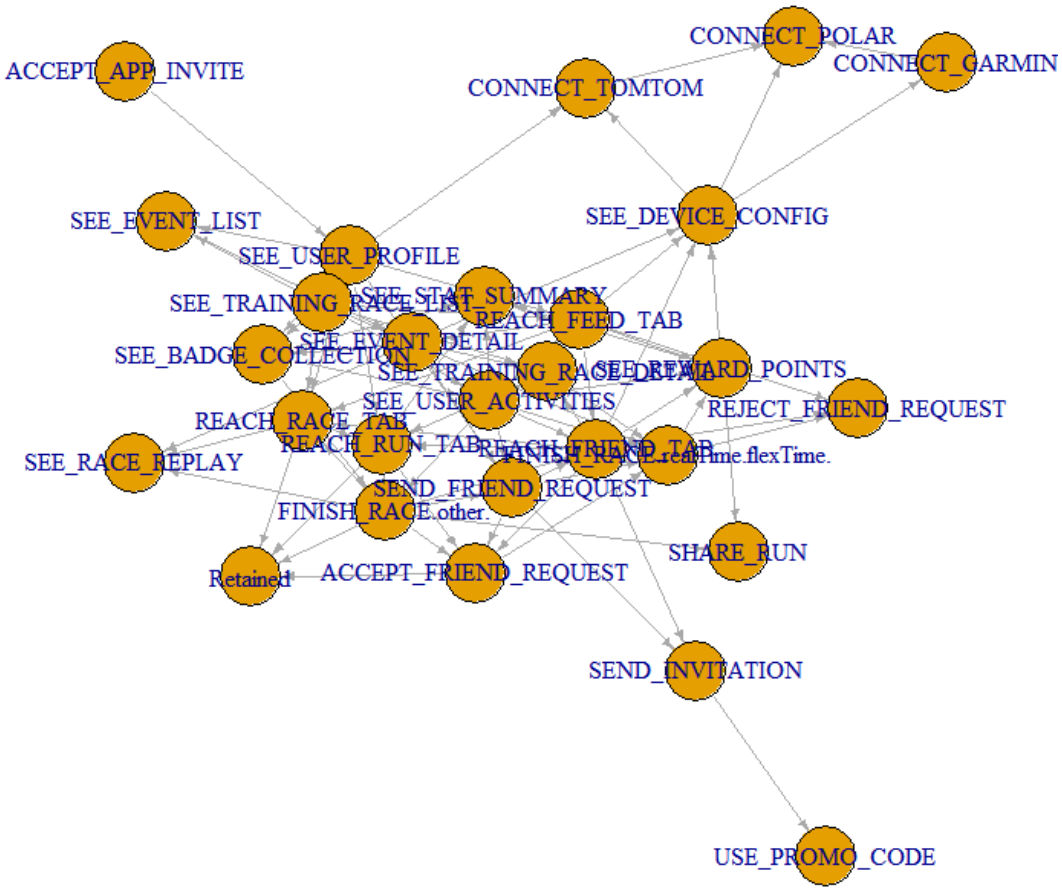


Figure 3.14: Best Bayesian network obtained by Hill-Climbing algorithm

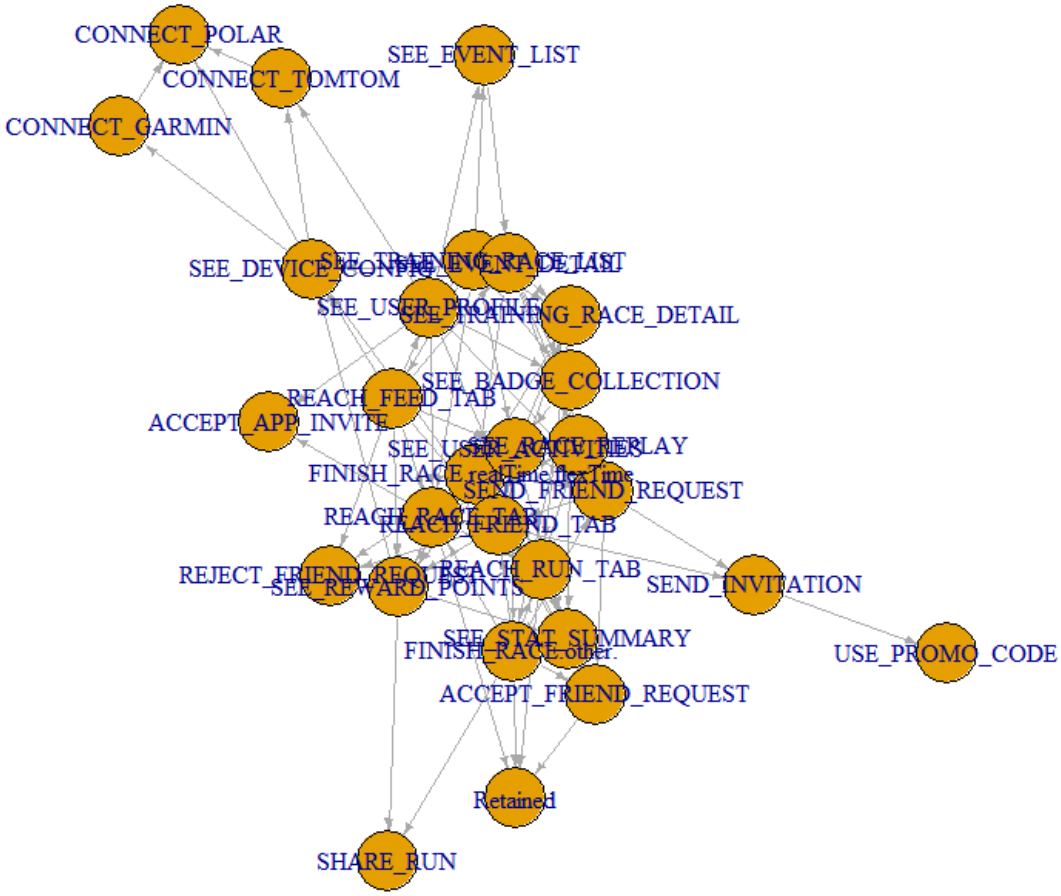


Figure 3.15: Best Bayesian network obtained by Tabu search learning algorithm

network score results we can see when the minority group is over-sampled till the dataset is balanced, the network structure learning does not improve by under-sampling methods simply because our target is to achieve a final balanced dataset by beans of re-sampling. But in the re-sampling scenario where we use Borderline-SMOTE 1 to over-sample till the dataset is balanced, the BLSMOTE1 algorithm produces instances of the minority group more that the original majority group which then get cleansed by the under-sampling methods. Because of this reason, we are left with different network models apparently performing better than rest of the re-sampling combinations. It might be worth investigating about possibilities of making the minority group a majority group by over-sampling and cleaning the noisy instances using under-sample methods afterwards.

All the BNs with highest network scores obtained by different learning algorithms contains the node "`Retained`". The node(s) that are immediate parents and/or the node(s) that are immediate child(ren) differ from each BN. But the node "`REACH_RACE_TAB`" is an immediate parent to the node "`Retained`" in every best BN by each learning algorithm.

Chapter 4

Conclusion

Re-sampling techniques on the mobile App use dataset show a significant improvement on binary classifiers. Identifying the best combination of over-sampling the minority group and under-sampling the majority group by cleaning the noisy instances play a vital role in model training. Depending on the re-sampling algorithm the final results vary so it is better to compare multiple re-sampling strategies prior to finalizing a training dataset. For cross validated results, in order to obtain an average roc, there are several standard methods as well as some näive ways. Depending on the dataset, simpler ways might yield better or similar results as more complex algorithms that are invented for more complex scenarios.

Bayesian Networks yield causal relationships between features in a dataset and from the Bayesian networks obtained by the dataset implies that some in-app features have some dependencies between them and some features

influences App retention. Re-sampling seems to improve network score overall with respect to the initial network scores obtained from Bayesian network structures learned by the dataset without over-sampling. The choice of re-sampling technique combination depends on the requirement of obtaining a higher network score when there are several Bayesian Networks to be chosen from. The validity of the Bayesian network can be assessed with respect to the network score value but the domain expertise on the relationships between the features plays a significant role on constructing a valid causal network. Several queries on hypothetical scenarios can be used to calculate the conditional probabilities and hence perform the decision making.

This study can be further expanded to a temporal domain in future, where the in-App feature usage and the status of the mobile App user is monitored continuously through time. The changes in usage of features can be modeled using temporal Beyasian networks and hence, mobile App user retention pattern can be modeled with respect to time. Furthermore, lack of demographic information about the mobile App user data limits Bayesian networks on targeting the App users more precisely and to understand about the mobile App user so that the company's product can be catered to the customer requirement.

Bibliography

Acid, S. and L. M. de Campos (2003). Searching for bayesian network structures in the space of restricted acyclic partially directed graphs. *J. Artif. Int. Res. 18*(1), 445–490. (Cited on pages 67 and 91.)

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control 19*(6), 716–723. (Cited on page 86.)

Ashcroft, M. (2012). Bayesian networks in business analytics. In *Proceedings of the Federated Conference on Computer Science and Information Systems*, pp. 955–961. (Cited on page 59.)

Barua, S., M. M. Islam, X. Yao, and K. Murase (2014). MWMOTE - Majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Transactions on Knowledge and Data Engineering 26*(2), 405–425. (Cited on page 23.)

Bhattacharya, B. K., R. S. Poulsen, and G. T. Toussaint (1981). Application of

proximity graphs to editing nearest neighbor decision rule. In *International Symposium on Information Theory, Santa Monica*. (Cited on page 29.)

Bobek, S., M. Baran, K. Kluza, and G. J. Nalepa (2013). Application of bayesian networks to recommendations in business process modeling. In *AIBP@ AI* IA*, pp. 41–50. (Cited on page 59.)

Brown, L. E., I. Tsamardinos, and C. F. Aliferis (2004). A novel algorithm for scalable and accurate bayesian network learning. In *Medinfo*, pp. 711–715. (Cited on page 75.)

Chakraborty, S., K. Mengersen, C. Fidge, L. Ma, and D. Lassen (2016). A bayesian network-based customer satisfaction model: a tool for management decisions in railway transport. *Decision Analytics 3*(1), 4. (Cited on page 59.)

Chawla, N. V. (2009). Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook*, pp. 875–886. Springer. (Cited on page 20.)

Chawla, N. V., K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research 16*, 321–357. (Cited on pages 11, 12, 19 and 20.)

Chawla, N. V., N. Japkowicz, and A. Kotcz (2004). Special issue on learning from imbalanced data sets. *ACM SIGKDD explorations newsletter 6*(1), 1–6. (Cited on pages 3 and 7.)

Chawla, N. V., A. Lazarevic, L. O. Hall, and K. W. Bowyer (2003). Smoteboost: Improving prediction of the minority class in boosting. In *European conference on principles of data mining and knowledge discovery*, pp. 107–119. Springer. (Cited on page 16.)

Chen, W. and F. W. Samuelson (2014). The average receiver operating characteristic curve in multireader multicase imaging studies. *The British journal of radiology 87*(1040), 20140016. (Cited on page 41.)

Chickering, D., D. Geiger, and D. Heckerman (1995). Learning bayesian networks: Search methods and experimental results. *Preliminary Papers of the Fifth International Workshop on Artificial Intelligence and Statistics*. (Cited on page 66.)

Chickering, D. M. (1996). *Learning Bayesian Networks is NP-Complete* (Learning from Data: Artificial Intelligence and Statistics V ed.). Springer-Verlag. (Cited on page 66.)

Cooper, G. F. and E. Herskovits (1992). A bayesian method for the induction of probabilistic networks from data. *Mach. Learn. 9*(4), 309–347. (Cited on pages 65 and 68.)

Cortes, C. and V. Vapnik (1995). Support-vector networks. *Machine learning 20*(3), 273–297. (Cited on page 35.)

Cover, T. M. and P. E. Hart (2018). Nearest Neighbor Pattern Classification. Technical report. (Cited on page 28.)

de Campos, L. M. (2006). A scoring function for learning bayesian networks based on mutual information and conditional independence tests. *J. Mach. Learn. Res. 7*, 2149–2187. (Cited on page 66.)

de Jongh, M. and M. J. Druzdzel (2009). A comparison of structural distance measures for causal bayesian network models. *Recent Advances in Intelligent Information Systems, Challenging Problems of Science, Computer Science series*, 443–456. (Cited on page 91.)

DeLong, E., D. DeLong, and D. Clarke-Pearson (1988). Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics 44*(3), 837—845. (Cited on page 40.)

Domingos, P. (1999). Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, New York, NY, USA, pp. 155–164. Association for Computing Machinery. (Cited on page 8.)

Dorfman, D. D. and E. Alf (1969). Maximum-likelihood estimation of parameters of signal-detection theory and determination of confidence intervals—rating-method data. *Journal of Mathematical Psychology 6*(3), 487 – 496. (Cited on page 40.)

Drucker, H., C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik (1997). Support vector regression machines. In *Advances in neural information processing systems*, pp. 155–161. (Cited on page 36.)

Dumais, S., J. Platt, D. Heckerman, and M. Sahami (1998). Inductive learning algorithms and representations for text categorization. In *Proceedings of the Seventh International Conference on Information and Knowledge Management*, CIKM '98, New York, NY, USA, pp. 148–155. Association for Computing Machinery. (Cited on page 8.)

Ezawa, K., M. Singh, and S. W. Norton (1996). Learning goal oriented bayesian networks for telecommunications risk management. In *In Proceedings of the 13th International Conference on Machine Learning*, pp. 139–147. Morgan Kaufmann. (Cited on page 8.)

Fawcett, T. and F. Provost (1996). Combining data mining and machine learning for effective user profiling. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pp. 8–13. AAAI Press. (Cited on page 8.)

Fernández, A., S. García, F. Herrera, and N. V. Chawla (2018). SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary. (Cited on pages 10 and 27.)

Guo, H. and H. L. Viktor (2004). Learning from imbalanced data sets with boosting and data generation. Technical Report 1. (Cited on page 20.)

Haibo He, Yang Bai, E. A. Garcia, and Shutao Li (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 1322–1328. IEEE. (Cited on page 20.)

Han, H., W.-Y. Wang, and B.-H. Mao (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. Technical report. (Cited on page 17.)

Hand, D. J. (1997). *Construction and Assessment of Classification Rules*. John Wiley and Sons. (Cited on page 39.)

Hart, P. (1968). The condensed nearest neighbor rule (corresp.). *IEEE transactions on information theory 14*(3), 515–516. (Cited on page 30.)

Hastie, T., R. Tibshirani, and J. Friedman (2009). The elements of statistical learnin. (Cited on pages 33 and 35.)

Heckerman, D. (2008). *A Tutorial on Learning with Bayesian Networks*, pp. 33–82. Berlin, Heidelberg: Springer Berlin Heidelberg. (Cited on page 67.)

Heckerman, D., D. Geiger, and D. Chickering (1995). Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning 20*, 197–243. (Cited on pages 65 and 66.)

Hibbert, D. and N. Armstrong (2009). An introduction to bayesian methods for analyzing chemistry data: Part ii: A review of applications of bayesian meth-

ods in chemistry. *Chemometrics and Intelligent Laboratory Systems 97*(2), 211–220. (Cited on page 58.)

Hoch, D. (2014). App retention improves - apps used only once declines to 20%. http://info.localytics.com/blog/app-retention-improves. (Cited on page 2.)

Huang, K., C. Zhang, X. Ma, and G. Chen (2012). Predicting mobile application usage using contextual information. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, New York, NY, USA, pp. 1059–1065. Association for Computing Machinery. (Cited on page 4.)

Japkowicz, N. (2000). The class imbalance problem: Significance and strategies. In *In Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI*, pp. 111–117. (Cited on page 9.)

Japkowicz, N. et al. (2000). Learning from imbalanced data sets: a comparison of various strategies. In *AAAI workshop on learning from imbalanced data sets*, Volume 68, pp. 10–15. Menlo Park, CA. (Cited on page 3.)

Karatzoglou, A., A. Smola, K. Hornik, and M. A. Karatzoglou (2019). Package 'kernlab'. *CRAN R Project*. (Cited on page 37.)

Kass, R., L. Tierney, and J. B. Kadane (1988). Asymptotics in bayesian computation. (Cited on page 68.)

Kass, R. E. and A. E. Raftery (1995). Bayes factors. *Journal of the American Statistical Association 90*(430), 773–795. (Cited on page 68.)

Kubat, M., R. C. Holte, and S. Matwin (1998). Machine Learning for the Detection of Oil Spills in Satellite Radar Images. *Machine Learning 30*(2), 195–215. (Cited on page 8.)

Kubat, M. and S. Matwin (1997). Addressing the curse of imbalanced training sets: One-sided selection. In *ICML*. (Cited on pages 9 and 30.)

Larranaga, P., M. Poza, Y. Yurramendi, R. H. Murga, and C. M. H. Kuijpers (1996). Structure learning of bayesian networks by genetic algorithms: a performance analysis of control parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence 18*(9), 912–926. (Cited on page 66.)

Laurikkala, J., S. Quaglini, P. Barahona, and S. Andreassen (2001). Improving Identification of Difficult Small Classes by Balancing Class Distribution Induced Pluripotent Stem Cells View project Document classification View project Improving Identification of Difficult Small Classes by Balancing Class Distribution. pp. 63–66. (Cited on page 29.)

Lewis, D. D. and J. Catlett (1994). Heterogeneous uncertainty sampling for supervised learning. In W. W. Cohen and H. Hirsh (Eds.), *Machine Learning Proceedings 1994*, pp. 148 – 156. San Francisco (CA): Morgan Kaufmann. (Cited on pages 8 and 9.)

Lewis, D. D. and M. Ringuette (1994). A comparison of two learning algorithms for text categorization. (Cited on page 8.)

Ling, C. X. and C. Li (1998). Data mining for direct marketing: Problems and solutions. In *KDD*. (Cited on pages 9 and 37.)

Lucas, P. (2001). *Bayesian networks in medicine: a model-based approach to medical decision making*. na. (Cited on page 58.)

Margaritis, D. (2003). Learning bayesian network model structure from data. Technical report, Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science. (Cited on pages 72 and 73.)

Metz, C. E., P.-L. Wang, and H. B. Kronman (1984). *A New Approach for Testing the Significance of Differences Between ROC Curves Measured from Correlated Data*, pp. 432–445. Dordrecht: Springer Netherlands. (Cited on page 40.)

Meyer, D., E. Dimitriadou, K. Hornik, A. Weingessel, F. Leisch, C.-C. Chang, C.-C. Lin, and M. D. Meyer (2019). Package 'e1071'. *The R Journal*. (Cited on page 35.)

Neapolitan, R. (2003). *Learning Bayesian Networks*. Prentice Hall. (Cited on page 66.)

Neapolitan, R. E. et al. (2004). *Learning bayesian networks*, Volume 38. Pearson Prentice Hall Upper Saddle River, NJ. (Cited on page 72.)

Needham, C. J., J. R. Bradford, A. J. Bulpitt, and D. R. Westhead (2007). A

primer on learning in bayesian networks for computational biology. *PLOS Computational Biology 3*(8), 1–8. (Cited on page 58.)

Park, M.-H., J.-H. Hong, and S.-B. Cho (2007). Location-based recommendation system using bayesian user's preference model in mobile devices. In J. Indulska, J. Ma, L. T. Yang, T. Ungerer, and J. Cao (Eds.), *Ubiquitous Intelligence and Computing*, Berlin, Heidelberg, pp. 1130–1139. Springer Berlin Heidelberg. (Cited on page 4.)

Parsons, S. (2011). Probabilistic graphical models: Principles and techniques. *The Knowledge Engineering Review 26*(2), 237–238. (Cited on page 57.)

Pazzani, M., C. Merz, P. Murphy, K. Ali, T. Hume, and C. Brunk (1994). Reducing misclassification costs. In W. W. Cohen and H. Hirsh (Eds.), *Machine Learning Proceedings 1994*, pp. 217 – 225. San Francisco (CA): Morgan Kaufmann. (Cited on page 8.)

Pearl, J. (1988). Probabilistic reasoning in intelligent systems. In *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann. (Cited on page 57.)

Perrier, E., S. Imoto, and S. Miyano (2008). Finding optimal bayesian network given a super-structure. *Journal of Machine Learning Research 9*(Oct), 2251–2286. (Cited on page 91.)

Perro, J. (2018). Mobile apps: What's a good retention rate?

http://info.localytics.com/blog/mobile-apps-whats-a-good-retention-rate. (Cited on page 2.)

Powers, D. and Ailab (2011). Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. *J. Mach. Learn. Technol 2*, 2229–3981. (Cited on page 39.)

Provost, F. and T. Fawcett (1997). Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, KDD'97, pp. 43–48. AAAI Press. (Cited on page 39.)

Provost, F. and T. Fawcett (2001). Robust Classification for Imprecise Environments. *Machine Learning 42*(3), 203–231. (Cited on pages 8 and 37.)

Rabiei, E., M. White, A. Mosleh, S. Lyer, and J. Woo (2018). Component reliability modeling through the use of bayesian networks and applied physics-based models. In *2018 Annual Reliability and Maintainability Symposium (RAMS)*, pp. 1–7. (Cited on page 58.)

Schluter, F. (2011). A survey on independence-based markov networks learning. *Artificial Intelligence Review 42*. (Cited on page 74.)

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*. (Cited on pages 68 and 86.)

Scutari, M. (2009). Learning bayesian networks with the bnlearn r package. *arXiv preprint arXiv:0908.3817*. (Cited on pages 75 and 79.)

Spiegelhalter, D. J. and S. L. Lauritzen (1990). Sequential updating of conditional probabilities on directed graphical structures. *Networks 20*(5), 579–605. (Cited on page 65.)

Spirtes, P., C. N. Glymour, R. Scheines, and D. Heckerman (2000). *Causation, prediction, and search*. MIT press. (Cited on page 72.)

Tomek, I. (1976). An Experiment with the Nearest-Neighbor Rule. *IEEE Transactions on Systems, Man, and Cybernetics SMC-6*(6), 448–452. (Cited on pages 27 and 30.)

Tsamardinos, I., C. F. Aliferis, and A. Statnikov (2003). Time and sample efficient discovery of markov blankets and direct causal relations. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 673–678. (Cited on page 75.)

Tsamardinos, I., C. F. Aliferis, A. R. Statnikov, and E. Statnikov (2003). Algorithms for large scale markov blanket discovery. In *FLAIRS conference*, Volume 2, pp. 376–380. (Cited on pages 72 and 74.)

Tsamardinos, I., L. E. Brown, and C. F. Aliferis (2006). The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning 65*(1), 31–78. (Cited on pages 75 and 91.)

Venkatraman, E. S. and C. B. Begg (1996). A distribution-free procedure for comparing receiver operating characteristic curves from a paired experiment. *Biometrika 83*(4), 835–848. (Cited on page 40.)

Verma, T. and J. Pearl (1991). *Equivalence and synthesis of causal models.* UCLA, Computer Science Department. (Cited on page 72.)

Weiss, G. M. (2004). Mining with rarity: A unifying framework. *SIGKDD Explor. Newsl. 6*(1), 7–19. (Cited on pages 3 and 8.)

Wilson, D. L. (1972). Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. *IEEE Transactions on Systems, Man and Cybernetics 2*(3), 408–421. (Cited on pages 27 and 29.)

Yen, S.-J. and Y.-S. Lee (2009). Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications 36*(3, Part 1), 5718 – 5727. (Cited on page 31.)

Yerima, S. Y., S. Sezer, and G. McWilliams (2014). Analysis of bayesian classification-based approaches for android malware detection. *IET Information Security 8*(1), 25–36. (Cited on page 4.)

Appendix A

Appendix : Imbalance Problem

Percentage	OS	US	Model	fscore	sd_f_score	AUC	sd_auc
35	ADASYN	ENN	Logit	0.45	0.04	0.76	0.03
35	ADASYN	ENN	NaiveBayes	0.41	0.05	0.74	0.04
35	ADASYN	ENN	SVR	0.10	0.06	0.70	0.03
35	ADASYN	NCL	Logit	0.46	0.04	0.75	0.03
35	ADASYN	NCL	NaiveBayes	0.41	0.05	0.74	0.04
35	ADASYN	NCL	SVR	0.11	0.05	0.71	0.03
35	ADASYN	Original	Logit	0.45	0.04	0.76	0.03
35	ADASYN	Original	NaiveBayes	0.41	0.05	0.74	0.04
35	ADASYN	Original	SVR	0.09	0.04	0.71	0.04
35	ADASYN	OSS	Logit	0.46	0.04	0.75	0.03
35	ADASYN	OSS	NaiveBayes	0.41	0.05	0.73	0.05
35	ADASYN	OSS	SVR	0.12	0.08	0.69	0.04
35	ADASYN	RUS	Logit	0.45	0.04	0.76	0.03

35	ADASYN	RUS	NaiveBayes	0.41	0.05	0.74	0.04
35	ADASYN	RUS	SVR	0.12	0.04	0.70	0.05
35	ADASYN	SBC	Logit	0.45	0.04	0.76	0.03
35	ADASYN	SBC	NaiveBayes	0.41	0.05	0.74	0.04
35	ADASYN	SBC	SVR	0.11	0.05	0.69	0.04
35	ADASYN	TL	Logit	0.45	0.04	0.76	0.03
35	ADASYN	TL	NaiveBayes	0.41	0.04	0.74	0.04
35	ADASYN	TL	SVR	0.09	0.05	0.71	0.03
35	BLSMOTE_1	ENN	Logit	0.49	0.04	0.74	0.02
35	BLSMOTE_1	ENN	NaiveBayes	0.33	0.07	0.70	0.04
35	BLSMOTE_1	ENN	SVR	0.20	0.10	0.59	0.07
35	BLSMOTE_1	NCL	Logit	0.49	0.03	0.74	0.03
35	BLSMOTE_1	NCL	NaiveBayes	0.32	0.07	0.70	0.04
35	BLSMOTE_1	NCL	SVR	0.24	0.11	0.60	0.05
35	BLSMOTE_1	Original	Logit	0.49	0.03	0.74	0.04
35	BLSMOTE_1	Original	NaiveBayes	0.31	0.08	0.70	0.04
35	BLSMOTE_1	Original	SVR	0.22	0.09	0.60	0.06
35	BLSMOTE_1	OSS	Logit	0.50	0.03	0.74	0.04
35	BLSMOTE_1	OSS	NaiveBayes	0.32	0.08	0.70	0.04
35	BLSMOTE_1	OSS	SVR	0.26	0.11	0.63	0.05
35	BLSMOTE_1	RUS	Logit	0.49	0.03	0.74	0.03
35	BLSMOTE_1	RUS	NaiveBayes	0.31	0.06	0.70	0.05
35	BLSMOTE_1	RUS	SVR	0.27	0.14	0.60	0.08
35	BLSMOTE_1	SBC	Logit	0.49	0.03	0.74	0.03
35	BLSMOTE_1	SBC	NaiveBayes	0.33	0.06	0.70	0.04

35	BLSMOTE_1	SBC	SVR	0.26	0.13	0.62	0.06
35	BLSMOTE_1	TL	Logit	0.49	0.04	0.73	0.03
35	BLSMOTE_1	TL	NaiveBayes	0.33	0.07	0.70	0.04
35	BLSMOTE_1	TL	SVR	0.26	0.15	0.59	0.07
35	BLSMOTE_2	ENN	Logit	0.47	0.03	0.76	0.03
35	BLSMOTE_2	ENN	NaiveBayes	0.42	0.05	0.74	0.04
35	BLSMOTE_2	ENN	SVR	0.11	0.04	0.71	0.03
35	BLSMOTE_2	NCL	Logit	0.47	0.03	0.76	0.03
35	BLSMOTE_2	NCL	NaiveBayes	0.42	0.05	0.74	0.04
35	BLSMOTE_2	NCL	SVR	0.11	0.06	0.70	0.03
35	BLSMOTE_2	Original	Logit	0.47	0.03	0.76	0.03
35	BLSMOTE_2	Original	NaiveBayes	0.42	0.05	0.74	0.04
35	BLSMOTE_2	Original	SVR	0.12	0.05	0.71	0.03
35	BLSMOTE_2	OSS	Logit	0.47	0.03	0.76	0.03
35	BLSMOTE_2	OSS	NaiveBayes	0.42	0.05	0.74	0.04
35	BLSMOTE_2	OSS	SVR	0.11	0.06	0.70	0.03
35	BLSMOTE_2	RUS	Logit	0.47	0.03	0.76	0.03
35	BLSMOTE_2	RUS	NaiveBayes	0.42	0.05	0.74	0.04
35	BLSMOTE_2	RUS	SVR	0.11	0.05	0.70	0.03
35	BLSMOTE_2	SBC	Logit	0.47	0.03	0.76	0.03
35	BLSMOTE_2	SBC	NaiveBayes	0.42	0.05	0.74	0.04
35	BLSMOTE_2	SBC	SVR	0.10	0.04	0.71	0.03
35	BLSMOTE_2	TL	Logit	0.47	0.03	0.76	0.03
35	BLSMOTE_2	TL	NaiveBayes	0.42	0.05	0.74	0.04
35	BLSMOTE_2	TL	SVR	0.11	0.05	0.70	0.03

35	MWMOTE	ENN	Logit	0.51	0.03	0.76	0.03
35	MWMOTE	ENN	NaiveBayes	0.46	0.04	0.74	0.03
35	MWMOTE	ENN	SVR	0.46	0.03	0.70	0.02
35	MWMOTE	NCL	Logit	0.51	0.03	0.76	0.03
35	MWMOTE	NCL	NaiveBayes	0.46	0.04	0.74	0.03
35	MWMOTE	NCL	SVR	0.45	0.03	0.70	0.03
35	MWMOTE	Original	Logit	0.51	0.03	0.76	0.03
35	MWMOTE	Original	NaiveBayes	0.46	0.04	0.74	0.03
35	MWMOTE	Original	SVR	0.45	0.04	0.70	0.03
35	MWMOTE	OSS	Logit	0.51	0.03	0.76	0.03
35	MWMOTE	OSS	NaiveBayes	0.46	0.04	0.74	0.03
35	MWMOTE	OSS	SVR	0.46	0.03	0.70	0.03
35	MWMOTE	RUS	Logit	0.51	0.03	0.76	0.03
35	MWMOTE	RUS	NaiveBayes	0.46	0.04	0.74	0.03
35	MWMOTE	RUS	SVR	0.45	0.03	0.70	0.03
35	MWMOTE	SBC	Logit	0.51	0.03	0.76	0.03
35	MWMOTE	SBC	NaiveBayes	0.46	0.04	0.74	0.03
35	MWMOTE	SBC	SVR	0.45	0.03	0.70	0.03
35	MWMOTE	TL	Logit	0.51	0.03	0.76	0.03
35	MWMOTE	TL	NaiveBayes	0.46	0.04	0.74	0.03
35	MWMOTE	TL	SVR	0.45	0.03	0.70	0.03
35	Original	ENN	Logit	0.38	0.05	0.75	0.03
35	Original	ENN	NaiveBayes	0.41	0.05	0.74	0.04
35	Original	ENN	SVR	0.05	0.02	0.72	0.03
35	Original	NCL	Logit	0.38	0.05	0.75	0.03

35	Original	NCL	NaiveBayes	0.41	0.05	0.74	0.04
35	Original	NCL	SVR	0.05	0.03	0.72	0.04
35	Original	Original	Logit	0.38	0.05	0.75	0.03
35	Original	Original	NaiveBayes	0.41	0.05	0.74	0.04
35	Original	Original	SVR	0.05	0.02	0.72	0.04
35	Original	OSS	Logit	0.38	0.05	0.75	0.03
35	Original	OSS	NaiveBayes	0.41	0.05	0.74	0.04
35	Original	OSS	SVR	0.05	0.02	0.72	0.04
35	Original	RUS	Logit	0.38	0.05	0.75	0.03
35	Original	RUS	NaiveBayes	0.41	0.05	0.74	0.04
35	Original	RUS	SVR	0.05	0.02	0.71	0.04
35	Original	SBC	Logit	0.38	0.05	0.75	0.03
35	Original	SBC	NaiveBayes	0.41	0.05	0.74	0.04
35	Original	SBC	SVR	0.05	0.02	0.71	0.04
35	Original	TL	Logit	0.38	0.05	0.75	0.03
35	Original	TL	NaiveBayes	0.41	0.05	0.74	0.04
35	Original	TL	SVR	0.05	0.02	0.72	0.04
35	ROS	ENN	Logit	0.50	0.04	0.75	0.03
35	ROS	ENN	NaiveBayes	0.41	0.06	0.74	0.04
35	ROS	ENN	SVR	0.28	0.15	0.68	0.02
35	ROS	NCL	Logit	0.52	0.03	0.76	0.02
35	ROS	NCL	NaiveBayes	0.40	0.06	0.74	0.04
35	ROS	NCL	SVR	0.35	0.15	0.67	0.02
35	ROS	Original	Logit	0.52	0.04	0.76	0.03
35	ROS	Original	NaiveBayes	0.41	0.04	0.74	0.04

35	ROS	Original	SVR	0.27	0.16	0.68	0.04
35	ROS	OSS	Logit	0.50	0.03	0.75	0.03
35	ROS	OSS	NaiveBayes	0.40	0.05	0.74	0.04
35	ROS	OSS	SVR	0.34	0.17	0.67	0.02
35	ROS	RUS	Logit	0.51	0.03	0.76	0.03
35	ROS	RUS	NaiveBayes	0.42	0.05	0.74	0.04
35	ROS	RUS	SVR	0.30	0.17	0.67	0.03
35	ROS	SBC	Logit	0.52	0.04	0.76	0.03
35	ROS	SBC	NaiveBayes	0.41	0.05	0.74	0.04
35	ROS	SBC	SVR	0.31	0.16	0.68	0.03
35	ROS	TL	Logit	0.52	0.03	0.76	0.03
35	ROS	TL	NaiveBayes	0.40	0.06	0.74	0.04
35	ROS	TL	SVR	0.34	0.14	0.68	0.03
35	SMOTE	ENN	Logit	0.51	0.03	0.76	0.03
35	SMOTE	ENN	NaiveBayes	0.46	0.04	0.74	0.04
35	SMOTE	ENN	SVR	0.44	0.03	0.67	0.04
35	SMOTE	NCL	Logit	0.51	0.03	0.76	0.03
35	SMOTE	NCL	NaiveBayes	0.46	0.04	0.74	0.04
35	SMOTE	NCL	SVR	0.44	0.03	0.67	0.04
35	SMOTE	Original	Logit	0.51	0.03	0.76	0.03
35	SMOTE	Original	NaiveBayes	0.46	0.04	0.74	0.04
35	SMOTE	Original	SVR	0.45	0.03	0.67	0.04
35	SMOTE	OSS	Logit	0.51	0.03	0.76	0.03
35	SMOTE	OSS	NaiveBayes	0.46	0.04	0.74	0.04
35	SMOTE	OSS	SVR	0.45	0.03	0.67	0.04

35	SMOTE	RUS	Logit	0.51	0.03	0.76	0.03
35	SMOTE	RUS	NaiveBayes	0.46	0.04	0.74	0.04
35	SMOTE	RUS	SVR	0.44	0.03	0.67	0.04
35	SMOTE	SBC	Logit	0.51	0.03	0.76	0.03
35	SMOTE	SBC	NaiveBayes	0.46	0.04	0.74	0.04
35	SMOTE	SBC	SVR	0.44	0.03	0.67	0.04
35	SMOTE	TL	Logit	0.51	0.03	0.76	0.03
35	SMOTE	TL	NaiveBayes	0.46	0.04	0.74	0.04
35	SMOTE	TL	SVR	0.44	0.03	0.67	0.04
40	ADASYN	ENN	Logit	0.48	0.03	0.75	0.03
40	ADASYN	ENN	NaiveBayes	0.40	0.05	0.73	0.04
40	ADASYN	ENN	SVR	0.28	0.19	0.68	0.04
40	ADASYN	NCL	Logit	0.47	0.03	0.75	0.03
40	ADASYN	NCL	NaiveBayes	0.41	0.05	0.74	0.04
40	ADASYN	NCL	SVR	0.25	0.18	0.70	0.04
40	ADASYN	Original	Logit	0.47	0.04	0.75	0.03
40	ADASYN	Original	NaiveBayes	0.41	0.04	0.74	0.04
40	ADASYN	Original	SVR	0.17	0.11	0.70	0.03
40	ADASYN	OSS	Logit	0.47	0.04	0.75	0.03
40	ADASYN	OSS	NaiveBayes	0.41	0.04	0.74	0.04
40	ADASYN	OSS	SVR	0.16	0.13	0.71	0.02
40	ADASYN	RUS	Logit	0.47	0.04	0.75	0.03
40	ADASYN	RUS	NaiveBayes	0.41	0.04	0.73	0.04
40	ADASYN	RUS	SVR	0.16	0.11	0.70	0.03
40	ADASYN	SBC	Logit	0.47	0.04	0.75	0.03

40	ADASYN	SBC	NaiveBayes	0.41	0.04	0.74	0.04
40	ADASYN	SBC	SVR	0.19	0.13	0.70	0.05
40	ADASYN	TL	Logit	0.47	0.04	0.75	0.03
40	ADASYN	TL	NaiveBayes	0.41	0.04	0.74	0.04
40	ADASYN	TL	SVR	0.24	0.17	0.70	0.03
40	BLSMOTE_1	ENN	Logit	0.49	0.02	0.74	0.03
40	BLSMOTE_1	ENN	NaiveBayes	0.31	0.06	0.69	0.05
40	BLSMOTE_1	ENN	SVR	0.25	0.11	0.58	0.07
40	BLSMOTE_1	NCL	Logit	0.49	0.03	0.72	0.04
40	BLSMOTE_1	NCL	NaiveBayes	0.30	0.07	0.68	0.05
40	BLSMOTE_1	NCL	SVR	0.29	0.14	0.60	0.06
40	BLSMOTE_1	Original	Logit	0.50	0.02	0.73	0.03
40	BLSMOTE_1	Original	NaiveBayes	0.31	0.06	0.68	0.05
40	BLSMOTE_1	Original	SVR	0.29	0.12	0.64	0.06
40	BLSMOTE_1	OSS	Logit	0.49	0.01	0.72	0.03
40	BLSMOTE_1	OSS	NaiveBayes	0.30	0.07	0.68	0.05
40	BLSMOTE_1	OSS	SVR	0.26	0.13	0.60	0.05
40	BLSMOTE_1	RUS	Logit	0.50	0.03	0.73	0.04
40	BLSMOTE_1	RUS	NaiveBayes	0.31	0.06	0.68	0.04
40	BLSMOTE_1	RUS	SVR	0.27	0.10	0.63	0.05
40	BLSMOTE_1	SBC	Logit	0.49	0.02	0.73	0.04
40	BLSMOTE_1	SBC	NaiveBayes	0.30	0.07	0.67	0.05
40	BLSMOTE_1	SBC	SVR	0.26	0.11	0.60	0.06
40	BLSMOTE_1	TL	Logit	0.46	0.06	0.72	0.03
40	BLSMOTE_1	TL	NaiveBayes	0.31	0.06	0.68	0.05

40	BLSMOTE_1	TL	SVR	0.24	0.11	0.59	0.04
40	BLSMOTE_2	ENN	Logit	0.48	0.03	0.76	0.03
40	BLSMOTE_2	ENN	NaiveBayes	0.43	0.05	0.74	0.04
40	BLSMOTE_2	ENN	SVR	0.14	0.05	0.70	0.03
40	BLSMOTE_2	NCL	Logit	0.48	0.03	0.76	0.03
40	BLSMOTE_2	NCL	NaiveBayes	0.43	0.05	0.74	0.04
40	BLSMOTE_2	NCL	SVR	0.14	0.06	0.69	0.03
40	BLSMOTE_2	Original	Logit	0.48	0.03	0.76	0.03
40	BLSMOTE_2	Original	NaiveBayes	0.43	0.05	0.74	0.04
40	BLSMOTE_2	Original	SVR	0.14	0.06	0.69	0.03
40	BLSMOTE_2	OSS	Logit	0.48	0.03	0.76	0.03
40	BLSMOTE_2	OSS	NaiveBayes	0.43	0.05	0.74	0.04
40	BLSMOTE_2	OSS	SVR	0.14	0.06	0.69	0.03
40	BLSMOTE_2	RUS	Logit	0.48	0.03	0.76	0.03
40	BLSMOTE_2	RUS	NaiveBayes	0.43	0.05	0.74	0.04
40	BLSMOTE_2	RUS	SVR	0.14	0.06	0.69	0.03
40	BLSMOTE_2	SBC	Logit	0.48	0.03	0.76	0.03
40	BLSMOTE_2	SBC	NaiveBayes	0.43	0.05	0.74	0.04
40	BLSMOTE_2	SBC	SVR	0.14	0.05	0.69	0.03
40	BLSMOTE_2	TL	Logit	0.48	0.03	0.76	0.03
40	BLSMOTE_2	TL	NaiveBayes	0.43	0.05	0.74	0.04
40	BLSMOTE_2	TL	SVR	0.15	0.06	0.69	0.03
40	MWMOTE	ENN	Logit	0.50	0.03	0.76	0.02
40	MWMOTE	ENN	NaiveBayes	0.46	0.03	0.74	0.03
40	MWMOTE	ENN	SVR	0.45	0.03	0.70	0.04

40	MWMOTE	NCL	Logit	0.50	0.03	0.76	0.02
40	MWMOTE	NCL	NaiveBayes	0.46	0.03	0.74	0.03
40	MWMOTE	NCL	SVR	0.45	0.03	0.70	0.04
40	MWMOTE	Original	Logit	0.50	0.03	0.76	0.02
40	MWMOTE	Original	NaiveBayes	0.46	0.03	0.74	0.03
40	MWMOTE	Original	SVR	0.45	0.03	0.70	0.04
40	MWMOTE	OSS	Logit	0.50	0.03	0.76	0.02
40	MWMOTE	OSS	NaiveBayes	0.46	0.03	0.74	0.03
40	MWMOTE	OSS	SVR	0.45	0.02	0.70	0.04
40	MWMOTE	RUS	Logit	0.50	0.03	0.76	0.02
40	MWMOTE	RUS	NaiveBayes	0.46	0.03	0.74	0.03
40	MWMOTE	RUS	SVR	0.45	0.03	0.70	0.04
40	MWMOTE	SBC	Logit	0.50	0.03	0.76	0.02
40	MWMOTE	SBC	NaiveBayes	0.46	0.03	0.74	0.03
40	MWMOTE	SBC	SVR	0.45	0.03	0.70	0.04
40	MWMOTE	TL	Logit	0.50	0.03	0.76	0.02
40	MWMOTE	TL	NaiveBayes	0.46	0.03	0.74	0.03
40	MWMOTE	TL	SVR	0.45	0.03	0.70	0.04
40	Original	ENN	Logit	0.38	0.05	0.75	0.03
40	Original	ENN	NaiveBayes	0.41	0.05	0.74	0.04
40	Original	ENN	SVR	0.05	0.02	0.72	0.03
40	Original	NCL	Logit	0.38	0.05	0.75	0.03
40	Original	NCL	NaiveBayes	0.41	0.05	0.74	0.04
40	Original	NCL	SVR	0.05	0.02	0.72	0.04
40	Original	Original	Logit	0.38	0.05	0.75	0.03

40	Original	Original	NaiveBayes	0.41	0.05	0.74	0.04
40	Original	Original	SVR	0.05	0.02	0.72	0.04
40	Original	OSS	Logit	0.38	0.05	0.75	0.03
40	Original	OSS	NaiveBayes	0.41	0.05	0.74	0.04
40	Original	OSS	SVR	0.05	0.02	0.72	0.03
40	Original	RUS	Logit	0.38	0.05	0.75	0.03
40	Original	RUS	NaiveBayes	0.41	0.05	0.74	0.04
40	Original	RUS	SVR	0.04	0.03	0.72	0.04
40	Original	SBC	Logit	0.38	0.05	0.75	0.03
40	Original	SBC	NaiveBayes	0.41	0.05	0.74	0.04
40	Original	SBC	SVR	0.05	0.02	0.72	0.03
40	Original	TL	Logit	0.38	0.05	0.75	0.03
40	Original	TL	NaiveBayes	0.41	0.05	0.74	0.04
40	Original	TL	SVR	0.05	0.02	0.72	0.04
40	ROS	ENN	Logit	0.51	0.04	0.76	0.03
40	ROS	ENN	NaiveBayes	0.40	0.05	0.73	0.04
40	ROS	ENN	SVR	0.15	0.04	0.67	0.03
40	ROS	NCL	Logit	0.51	0.03	0.76	0.03
40	ROS	NCL	NaiveBayes	0.41	0.05	0.74	0.04
40	ROS	NCL	SVR	0.16	0.05	0.68	0.04
40	ROS	Original	Logit	0.51	0.03	0.75	0.02
40	ROS	Original	NaiveBayes	0.41	0.06	0.74	0.04
40	ROS	Original	SVR	0.15	0.06	0.68	0.03
40	ROS	OSS	Logit	0.51	0.03	0.76	0.03
40	ROS	OSS	NaiveBayes	0.40	0.05	0.74	0.05

40	ROS	OSS	SVR	0.16	0.04	0.68	0.02
40	ROS	RUS	Logit	0.50	0.04	0.75	0.03
40	ROS	RUS	NaiveBayes	0.40	0.06	0.73	0.04
40	ROS	RUS	SVR	0.14	0.04	0.68	0.03
40	ROS	SBC	Logit	0.51	0.03	0.75	0.03
40	ROS	SBC	NaiveBayes	0.41	0.06	0.74	0.05
40	ROS	SBC	SVR	0.15	0.05	0.68	0.02
40	ROS	TL	Logit	0.51	0.03	0.76	0.02
40	ROS	TL	NaiveBayes	0.41	0.06	0.73	0.04
40	ROS	TL	SVR	0.14	0.05	0.68	0.02
40	SMOTE	ENN	Logit	0.53	0.03	0.76	0.03
40	SMOTE	ENN	NaiveBayes	0.46	0.04	0.73	0.04
40	SMOTE	ENN	SVR	0.44	0.04	0.67	0.04
40	SMOTE	NCL	Logit	0.53	0.03	0.76	0.03
40	SMOTE	NCL	NaiveBayes	0.46	0.04	0.73	0.04
40	SMOTE	NCL	SVR	0.44	0.04	0.67	0.04
40	SMOTE	Original	Logit	0.53	0.03	0.76	0.03
40	SMOTE	Original	NaiveBayes	0.46	0.04	0.73	0.04
40	SMOTE	Original	SVR	0.44	0.04	0.67	0.04
40	SMOTE	OSS	Logit	0.53	0.03	0.76	0.03
40	SMOTE	OSS	NaiveBayes	0.46	0.04	0.73	0.04
40	SMOTE	OSS	SVR	0.44	0.04	0.67	0.04
40	SMOTE	RUS	Logit	0.53	0.03	0.76	0.03
40	SMOTE	RUS	NaiveBayes	0.46	0.04	0.73	0.04
40	SMOTE	RUS	SVR	0.44	0.03	0.67	0.04

40	SMOTE	SBC	Logit	0.53	0.03	0.76	0.03
40	SMOTE	SBC	NaiveBayes	0.46	0.04	0.73	0.04
40	SMOTE	SBC	SVR	0.44	0.04	0.67	0.03
40	SMOTE	TL	Logit	0.53	0.03	0.76	0.03
40	SMOTE	TL	NaiveBayes	0.46	0.04	0.73	0.04
40	SMOTE	TL	SVR	0.44	0.04	0.67	0.04
45	ADASYN	ENN	Logit	0.49	0.02	0.75	0.03
45	ADASYN	ENN	NaiveBayes	0.41	0.04	0.73	0.04
45	ADASYN	ENN	SVR	0.44	0.11	0.70	0.02
45	ADASYN	NCL	Logit	0.49	0.02	0.75	0.03
45	ADASYN	NCL	NaiveBayes	0.41	0.04	0.74	0.04
45	ADASYN	NCL	SVR	0.42	0.13	0.70	0.03
45	ADASYN	Original	Logit	0.49	0.02	0.75	0.03
45	ADASYN	Original	NaiveBayes	0.41	0.04	0.73	0.04
45	ADASYN	Original	SVR	0.44	0.10	0.70	0.03
45	ADASYN	OSS	Logit	0.49	0.02	0.76	0.03
45	ADASYN	OSS	NaiveBayes	0.41	0.04	0.73	0.04
45	ADASYN	OSS	SVR	0.44	0.09	0.69	0.03
45	ADASYN	RUS	Logit	0.49	0.02	0.76	0.03
45	ADASYN	RUS	NaiveBayes	0.41	0.04	0.74	0.04
45	ADASYN	RUS	SVR	0.45	0.11	0.71	0.03
45	ADASYN	SBC	Logit	0.49	0.02	0.76	0.03
45	ADASYN	SBC	NaiveBayes	0.41	0.04	0.73	0.04
45	ADASYN	SBC	SVR	0.41	0.13	0.70	0.04
45	ADASYN	TL	Logit	0.49	0.02	0.76	0.03

45	ADASYN	TL	NaiveBayes	0.41	0.04	0.73	0.04
45	ADASYN	TL	SVR	0.45	0.10	0.70	0.03
45	BLSMOTE_1	ENN	Logit	0.49	0.03	0.73	0.03
45	BLSMOTE_1	ENN	NaiveBayes	0.26	0.10	0.60	0.09
45	BLSMOTE_1	ENN	SVR	0.33	0.15	0.62	0.06
45	BLSMOTE_1	NCL	Logit	0.49	0.03	0.74	0.03
45	BLSMOTE_1	NCL	NaiveBayes	0.26	0.09	0.61	0.09
45	BLSMOTE_1	NCL	SVR	0.24	0.10	0.58	0.06
45	BLSMOTE_1	Original	Logit	0.49	0.03	0.74	0.03
45	BLSMOTE_1	Original	NaiveBayes	0.26	0.08	0.60	0.08
45	BLSMOTE_1	Original	SVR	0.30	0.13	0.61	0.04
45	BLSMOTE_1	OSS	Logit	0.49	0.04	0.73	0.03
45	BLSMOTE_1	OSS	NaiveBayes	0.25	0.09	0.60	0.09
45	BLSMOTE_1	OSS	SVR	0.36	0.13	0.63	0.05
45	BLSMOTE_1	RUS	Logit	0.48	0.05	0.72	0.08
45	BLSMOTE_1	RUS	NaiveBayes	0.25	0.10	0.60	0.09
45	BLSMOTE_1	RUS	SVR	0.22	0.11	0.62	0.05
45	BLSMOTE_1	SBC	Logit	0.49	0.03	0.73	0.04
45	BLSMOTE_1	SBC	NaiveBayes	0.26	0.09	0.60	0.09
45	BLSMOTE_1	SBC	SVR	0.25	0.10	0.63	0.07
45	BLSMOTE_1	TL	Logit	0.50	0.03	0.74	0.04
45	BLSMOTE_1	TL	NaiveBayes	0.29	0.09	0.60	0.09
45	BLSMOTE_1	TL	SVR	0.23	0.08	0.62	0.04
45	BLSMOTE_2	ENN	Logit	0.50	0.03	0.76	0.03
45	BLSMOTE_2	ENN	NaiveBayes	0.43	0.04	0.74	0.04

45	BLSMOTE_2	ENN	SVR	0.16	0.04	0.65	0.05
45	BLSMOTE_2	NCL	Logit	0.50	0.03	0.76	0.03
45	BLSMOTE_2	NCL	NaiveBayes	0.43	0.04	0.74	0.04
45	BLSMOTE_2	NCL	SVR	0.16	0.04	0.66	0.05
45	BLSMOTE_2	Original	Logit	0.50	0.03	0.76	0.03
45	BLSMOTE_2	Original	NaiveBayes	0.43	0.04	0.74	0.04
45	BLSMOTE_2	Original	SVR	0.15	0.05	0.65	0.05
45	BLSMOTE_2	OSS	Logit	0.50	0.03	0.76	0.03
45	BLSMOTE_2	OSS	NaiveBayes	0.43	0.04	0.74	0.04
45	BLSMOTE_2	OSS	SVR	0.18	0.10	0.67	0.03
45	BLSMOTE_2	RUS	Logit	0.50	0.03	0.76	0.03
45	BLSMOTE_2	RUS	NaiveBayes	0.43	0.04	0.74	0.04
45	BLSMOTE_2	RUS	SVR	0.14	0.05	0.66	0.04
45	BLSMOTE_2	SBC	Logit	0.50	0.03	0.76	0.03
45	BLSMOTE_2	SBC	NaiveBayes	0.43	0.04	0.74	0.04
45	BLSMOTE_2	SBC	SVR	0.15	0.04	0.65	0.05
45	BLSMOTE_2	TL	Logit	0.50	0.03	0.76	0.03
45	BLSMOTE_2	TL	NaiveBayes	0.43	0.04	0.74	0.04
45	BLSMOTE_2	TL	SVR	0.18	0.10	0.66	0.05
45	MWMOTE	ENN	Logit	0.49	0.02	0.75	0.02
45	MWMOTE	ENN	NaiveBayes	0.46	0.04	0.74	0.04
45	MWMOTE	ENN	SVR	0.45	0.03	0.69	0.03
45	MWMOTE	NCL	Logit	0.49	0.02	0.75	0.02
45	MWMOTE	NCL	NaiveBayes	0.46	0.04	0.74	0.04
45	MWMOTE	NCL	SVR	0.45	0.03	0.69	0.03

45	MWMOTE	Original	Logit	0.49	0.02	0.75	0.02
45	MWMOTE	Original	NaiveBayes	0.46	0.04	0.74	0.04
45	MWMOTE	Original	SVR	0.45	0.03	0.69	0.03
45	MWMOTE	OSS	Logit	0.49	0.02	0.75	0.02
45	MWMOTE	OSS	NaiveBayes	0.46	0.04	0.74	0.04
45	MWMOTE	OSS	SVR	0.45	0.03	0.69	0.03
45	MWMOTE	RUS	Logit	0.49	0.02	0.75	0.02
45	MWMOTE	RUS	NaiveBayes	0.46	0.04	0.74	0.04
45	MWMOTE	RUS	SVR	0.45	0.03	0.69	0.03
45	MWMOTE	SBC	Logit	0.49	0.02	0.75	0.02
45	MWMOTE	SBC	NaiveBayes	0.46	0.04	0.74	0.04
45	MWMOTE	SBC	SVR	0.45	0.03	0.69	0.03
45	MWMOTE	TL	Logit	0.49	0.02	0.75	0.02
45	MWMOTE	TL	NaiveBayes	0.46	0.04	0.74	0.04
45	MWMOTE	TL	SVR	0.45	0.03	0.69	0.03
45	Original	ENN	Logit	0.38	0.05	0.75	0.03
45	Original	ENN	NaiveBayes	0.41	0.05	0.74	0.04
45	Original	ENN	SVR	0.05	0.02	0.72	0.04
45	Original	NCL	Logit	0.38	0.05	0.75	0.03
45	Original	NCL	NaiveBayes	0.41	0.05	0.74	0.04
45	Original	NCL	SVR	0.05	0.02	0.72	0.04
45	Original	Original	Logit	0.38	0.05	0.75	0.03
45	Original	Original	NaiveBayes	0.41	0.05	0.74	0.04
45	Original	Original	SVR	0.04	0.03	0.72	0.03
45	Original	OSS	Logit	0.38	0.05	0.75	0.03

45	Original	OSS	NaiveBayes	0.41	0.05	0.74	0.04
45	Original	OSS	SVR	0.04	0.03	0.72	0.03
45	Original	RUS	Logit	0.38	0.05	0.75	0.03
45	Original	RUS	NaiveBayes	0.41	0.05	0.74	0.04
45	Original	RUS	SVR	0.05	0.02	0.72	0.03
45	Original	SBC	Logit	0.38	0.05	0.75	0.03
45	Original	SBC	NaiveBayes	0.41	0.05	0.74	0.04
45	Original	SBC	SVR	0.05	0.02	0.71	0.04
45	Original	TL	Logit	0.38	0.05	0.75	0.03
45	Original	TL	NaiveBayes	0.41	0.05	0.74	0.04
45	Original	TL	SVR	0.05	0.02	0.71	0.04
45	ROS	ENN	Logit	0.51	0.04	0.76	0.03
45	ROS	ENN	NaiveBayes	0.41	0.05	0.74	0.04
45	ROS	ENN	SVR	0.13	0.05	0.68	0.03
45	ROS	NCL	Logit	0.51	0.04	0.76	0.03
45	ROS	NCL	NaiveBayes	0.41	0.05	0.74	0.04
45	ROS	NCL	SVR	0.13	0.06	0.68	0.03
45	ROS	Original	Logit	0.52	0.04	0.76	0.03
45	ROS	Original	NaiveBayes	0.42	0.05	0.74	0.04
45	ROS	Original	SVR	0.14	0.06	0.68	0.03
45	ROS	OSS	Logit	0.52	0.04	0.76	0.03
45	ROS	OSS	NaiveBayes	0.41	0.05	0.74	0.04
45	ROS	OSS	SVR	0.14	0.06	0.69	0.02
45	ROS	RUS	Logit	0.52	0.04	0.76	0.03
45	ROS	RUS	NaiveBayes	0.40	0.06	0.74	0.04

45	ROS	RUS	SVR	0.13	0.06	0.68	0.03
45	ROS	SBC	Logit	0.52	0.04	0.76	0.03
45	ROS	SBC	NaiveBayes	0.41	0.05	0.74	0.05
45	ROS	SBC	SVR	0.13	0.05	0.69	0.02
45	ROS	TL	Logit	0.52	0.04	0.76	0.03
45	ROS	TL	NaiveBayes	0.41	0.05	0.74	0.04
45	ROS	TL	SVR	0.14	0.06	0.69	0.03
45	SMOTE	ENN	Logit	0.51	0.03	0.76	0.03
45	SMOTE	ENN	NaiveBayes	0.46	0.04	0.74	0.04
45	SMOTE	ENN	SVR	0.44	0.03	0.67	0.04
45	SMOTE	NCL	Logit	0.51	0.03	0.76	0.03
45	SMOTE	NCL	NaiveBayes	0.46	0.04	0.74	0.04
45	SMOTE	NCL	SVR	0.44	0.03	0.67	0.04
45	SMOTE	Original	Logit	0.51	0.03	0.76	0.03
45	SMOTE	Original	NaiveBayes	0.46	0.04	0.74	0.04
45	SMOTE	Original	SVR	0.45	0.04	0.67	0.04
45	SMOTE	OSS	Logit	0.51	0.03	0.76	0.03
45	SMOTE	OSS	NaiveBayes	0.46	0.04	0.74	0.04
45	SMOTE	OSS	SVR	0.45	0.03	0.67	0.04
45	SMOTE	RUS	Logit	0.51	0.03	0.76	0.03
45	SMOTE	RUS	NaiveBayes	0.46	0.04	0.74	0.04
45	SMOTE	RUS	SVR	0.44	0.03	0.67	0.04
45	SMOTE	SBC	Logit	0.51	0.03	0.76	0.03
45	SMOTE	SBC	NaiveBayes	0.46	0.04	0.74	0.04
45	SMOTE	SBC	SVR	0.45	0.03	0.67	0.04

45	SMOTE	TL	Logit	0.51	0.03	0.76	0.03
45	SMOTE	TL	NaiveBayes	0.46	0.04	0.74	0.04
45	SMOTE	TL	SVR	0.45	0.04	0.67	0.04
50	ADASYN	ENN	Logit	0.50	0.03	0.75	0.03
50	ADASYN	ENN	NaiveBayes	0.42	0.05	0.74	0.04
50	ADASYN	ENN	SVR	0.45	0.09	0.70	0.04
50	ADASYN	NCL	Logit	0.50	0.03	0.75	0.03
50	ADASYN	NCL	NaiveBayes	0.41	0.04	0.73	0.04
50	ADASYN	NCL	SVR	0.46	0.04	0.68	0.03
50	ADASYN	Original	Logit	0.49	0.03	0.75	0.03
50	ADASYN	Original	NaiveBayes	0.42	0.04	0.74	0.04
50	ADASYN	Original	SVR	0.42	0.13	0.70	0.03
50	ADASYN	OSS	Logit	0.50	0.03	0.75	0.03
50	ADASYN	OSS	NaiveBayes	0.42	0.04	0.74	0.04
50	ADASYN	OSS	SVR	0.43	0.12	0.70	0.02
50	ADASYN	RUS	Logit	0.50	0.03	0.75	0.03
50	ADASYN	RUS	NaiveBayes	0.41	0.05	0.73	0.04
50	ADASYN	RUS	SVR	0.43	0.10	0.67	0.04
50	ADASYN	SBC	Logit	0.50	0.03	0.75	0.03
50	ADASYN	SBC	NaiveBayes	0.41	0.05	0.73	0.04
50	ADASYN	SBC	SVR	0.44	0.09	0.69	0.04
50	ADASYN	TL	Logit	0.49	0.02	0.75	0.03
50	ADASYN	TL	NaiveBayes	0.42	0.05	0.74	0.04
50	ADASYN	TL	SVR	0.44	0.09	0.70	0.04
50	BLSMOTE_1	ENN	Logit	0.41	0.09	0.73	0.03

50	BLSMOTE_1	ENN	NaiveBayes	0.30	0.10	0.65	0.10
50	BLSMOTE_1	ENN	SVR	0.20	0.17	0.67	0.07
50	BLSMOTE_1	NCL	Logit	0.41	0.10	0.73	0.04
50	BLSMOTE_1	NCL	NaiveBayes	0.31	0.11	0.64	0.11
50	BLSMOTE_1	NCL	SVR	0.16	0.12	0.64	0.08
50	BLSMOTE_1	Original	Logit	0.42	0.10	0.73	0.04
50	BLSMOTE_1	Original	NaiveBayes	0.31	0.10	0.66	0.09
50	BLSMOTE_1	Original	SVR	0.16	0.11	0.65	0.08
50	BLSMOTE_1	OSS	Logit	0.41	0.09	0.73	0.03
50	BLSMOTE_1	OSS	NaiveBayes	0.30	0.10	0.65	0.10
50	BLSMOTE_1	OSS	SVR	0.18	0.14	0.65	0.09
50	BLSMOTE_1	RUS	Logit	0.42	0.09	0.73	0.04
50	BLSMOTE_1	RUS	NaiveBayes	0.30	0.11	0.64	0.10
50	BLSMOTE_1	RUS	SVR	0.22	0.17	0.70	0.05
50	BLSMOTE_1	SBC	Logit	0.41	0.10	0.73	0.03
50	BLSMOTE_1	SBC	NaiveBayes	0.30	0.10	0.65	0.10
50	BLSMOTE_1	SBC	SVR	0.19	0.10	0.66	0.08
50	BLSMOTE_1	TL	Logit	0.42	0.09	0.74	0.03
50	BLSMOTE_1	TL	NaiveBayes	0.32	0.09	0.64	0.09
50	BLSMOTE_1	TL	SVR	0.18	0.08	0.66	0.08
50	BLSMOTE_2	ENN	Logit	0.50	0.03	0.76	0.03
50	BLSMOTE_2	ENN	NaiveBayes	0.43	0.05	0.73	0.04
50	BLSMOTE_2	ENN	SVR	0.23	0.13	0.61	0.05
50	BLSMOTE_2	NCL	Logit	0.50	0.03	0.76	0.03
50	BLSMOTE_2	NCL	NaiveBayes	0.43	0.05	0.73	0.04

50	BLSMOTE_2	NCL	SVR	0.26	0.15	0.64	0.06
50	BLSMOTE_2	Original	Logit	0.50	0.03	0.76	0.03
50	BLSMOTE_2	Original	NaiveBayes	0.43	0.05	0.73	0.04
50	BLSMOTE_2	Original	SVR	0.25	0.14	0.63	0.06
50	BLSMOTE_2	OSS	Logit	0.50	0.03	0.76	0.03
50	BLSMOTE_2	OSS	NaiveBayes	0.43	0.05	0.73	0.04
50	BLSMOTE_2	OSS	SVR	0.20	0.10	0.59	0.04
50	BLSMOTE_2	RUS	Logit	0.50	0.03	0.76	0.03
50	BLSMOTE_2	RUS	NaiveBayes	0.43	0.05	0.73	0.04
50	BLSMOTE_2	RUS	SVR	0.28	0.13	0.63	0.05
50	BLSMOTE_2	SBC	Logit	0.50	0.03	0.76	0.03
50	BLSMOTE_2	SBC	NaiveBayes	0.43	0.05	0.73	0.04
50	BLSMOTE_2	SBC	SVR	0.26	0.14	0.63	0.05
50	BLSMOTE_2	TL	Logit	0.50	0.03	0.76	0.03
50	BLSMOTE_2	TL	NaiveBayes	0.43	0.05	0.73	0.04
50	BLSMOTE_2	TL	SVR	0.20	0.09	0.62	0.05
50	MWMOTE	ENN	Logit	0.50	0.04	0.73	0.04
50	MWMOTE	ENN	NaiveBayes	0.44	0.04	0.73	0.04
50	MWMOTE	ENN	SVR	0.49	0.04	0.73	0.03
50	MWMOTE	NCL	Logit	0.50	0.04	0.73	0.04
50	MWMOTE	NCL	NaiveBayes	0.44	0.04	0.73	0.04
50	MWMOTE	NCL	SVR	0.49	0.03	0.73	0.04
50	MWMOTE	Original	Logit	0.50	0.04	0.73	0.04
50	MWMOTE	Original	NaiveBayes	0.44	0.04	0.73	0.04
50	MWMOTE	Original	SVR	0.49	0.04	0.73	0.04

50	MWMOTE	OSS	Logit	0.50	0.04	0.73	0.04
50	MWMOTE	OSS	NaiveBayes	0.44	0.04	0.73	0.04
50	MWMOTE	OSS	SVR	0.50	0.04	0.73	0.04
50	MWMOTE	RUS	Logit	0.50	0.04	0.73	0.04
50	MWMOTE	RUS	NaiveBayes	0.44	0.04	0.73	0.04
50	MWMOTE	RUS	SVR	0.49	0.04	0.73	0.04
50	MWMOTE	SBC	Logit	0.50	0.04	0.73	0.04
50	MWMOTE	SBC	NaiveBayes	0.44	0.04	0.73	0.04
50	MWMOTE	SBC	SVR	0.49	0.03	0.73	0.04
50	MWMOTE	TL	Logit	0.50	0.04	0.73	0.04
50	MWMOTE	TL	NaiveBayes	0.44	0.04	0.73	0.04
50	MWMOTE	TL	SVR	0.49	0.04	0.73	0.04
50	Original	ENN	Logit	0.38	0.05	0.75	0.03
50	Original	ENN	NaiveBayes	0.41	0.05	0.74	0.04
50	Original	ENN	SVR	0.05	0.03	0.71	0.04
50	Original	NCL	Logit	0.38	0.05	0.75	0.03
50	Original	NCL	NaiveBayes	0.41	0.05	0.74	0.04
50	Original	NCL	SVR	0.05	0.02	0.72	0.04
50	Original	Original	Logit	0.38	0.05	0.75	0.03
50	Original	Original	NaiveBayes	0.41	0.05	0.74	0.04
50	Original	Original	SVR	0.05	0.02	0.72	0.03
50	Original	OSS	Logit	0.38	0.05	0.75	0.03
50	Original	OSS	NaiveBayes	0.41	0.05	0.74	0.04
50	Original	OSS	SVR	0.06	0.02	0.72	0.03
50	Original	RUS	Logit	0.38	0.05	0.75	0.03

50	Original	RUS	NaiveBayes	0.41	0.05	0.74	0.04
50	Original	RUS	SVR	0.05	0.02	0.72	0.03
50	Original	SBC	Logit	0.38	0.05	0.75	0.03
50	Original	SBC	NaiveBayes	0.41	0.05	0.74	0.04
50	Original	SBC	SVR	0.06	0.02	0.72	0.03
50	Original	TL	Logit	0.38	0.05	0.75	0.03
50	Original	TL	NaiveBayes	0.41	0.05	0.74	0.04
50	Original	TL	SVR	0.04	0.02	0.72	0.04
50	ROS	ENN	Logit	0.52	0.04	0.76	0.03
50	ROS	ENN	NaiveBayes	0.41	0.05	0.74	0.04
50	ROS	ENN	SVR	0.13	0.04	0.68	0.03
50	ROS	NCL	Logit	0.52	0.04	0.76	0.03
50	ROS	NCL	NaiveBayes	0.41	0.05	0.74	0.04
50	ROS	NCL	SVR	0.12	0.05	0.68	0.02
50	ROS	Original	Logit	0.52	0.04	0.76	0.03
50	ROS	Original	NaiveBayes	0.41	0.05	0.74	0.04
50	ROS	Original	SVR	0.12	0.05	0.68	0.02
50	ROS	OSS	Logit	0.52	0.04	0.76	0.03
50	ROS	OSS	NaiveBayes	0.41	0.05	0.74	0.04
50	ROS	OSS	SVR	0.12	0.04	0.68	0.02
50	ROS	RUS	Logit	0.52	0.04	0.76	0.03
50	ROS	RUS	NaiveBayes	0.41	0.05	0.74	0.04
50	ROS	RUS	SVR	0.12	0.04	0.69	0.03
50	ROS	SBC	Logit	0.52	0.04	0.76	0.03
50	ROS	SBC	NaiveBayes	0.41	0.05	0.74	0.04

50	ROS	SBC	SVR	0.12	0.05	0.68	0.02
50	ROS	TL	Logit	0.52	0.04	0.76	0.03
50	ROS	TL	NaiveBayes	0.41	0.05	0.74	0.04
50	ROS	TL	SVR	0.12	0.06	0.68	0.02
50	SMOTE	ENN	Logit	0.50	0.03	0.76	0.03
50	SMOTE	ENN	NaiveBayes	0.43	0.05	0.73	0.04
50	SMOTE	ENN	SVR	0.45	0.05	0.68	0.05
50	SMOTE	NCL	Logit	0.50	0.03	0.76	0.03
50	SMOTE	NCL	NaiveBayes	0.43	0.05	0.73	0.04
50	SMOTE	NCL	SVR	0.45	0.05	0.68	0.05
50	SMOTE	Original	Logit	0.50	0.03	0.76	0.03
50	SMOTE	Original	NaiveBayes	0.43	0.05	0.73	0.04
50	SMOTE	Original	SVR	0.45	0.05	0.68	0.05
50	SMOTE	OSS	Logit	0.50	0.03	0.76	0.03
50	SMOTE	OSS	NaiveBayes	0.43	0.05	0.73	0.04
50	SMOTE	OSS	SVR	0.45	0.05	0.68	0.05
50	SMOTE	RUS	Logit	0.50	0.03	0.76	0.03
50	SMOTE	RUS	NaiveBayes	0.43	0.05	0.73	0.04
50	SMOTE	RUS	SVR	0.45	0.05	0.68	0.05
50	SMOTE	SBC	Logit	0.50	0.03	0.76	0.03
50	SMOTE	SBC	NaiveBayes	0.43	0.05	0.73	0.04
50	SMOTE	SBC	SVR	0.45	0.05	0.68	0.05
50	SMOTE	TL	Logit	0.50	0.03	0.76	0.03
50	SMOTE	TL	NaiveBayes	0.43	0.05	0.73	0.04

50	SMOTE	TL	SVR	0.45	0.05	0.68	0.05

Table A.1: F1 score and AUC values with 10-Fold cross validation

Appendix B

Appendix : Bayesian Networks

Percentage	OS	US	Model	BIC	AIC	logLik
35	Original	Original	HC	-29515.91	-28900.23	-28703.23
35	Original	RUS	HC	-29515.91	-28900.23	-28703.23
35	Original	ENN	HC	-29515.91	-28900.23	-28703.23
35	Original	NCL	HC	-29515.91	-28900.23	-28703.23
35	Original	OSS	HC	-29515.91	-28900.23	-28703.23
35	Original	SBC	HC	-29515.91	-28900.23	-28703.23
35	Original	TL	HC	-29515.91	-28900.23	-28703.23
35	ROS	Original	HC	-26905.76	-26279.97	-26074.97
35	ROS	RUS	HC	-26994.30	-26362.40	-26155.40
35	ROS	ENN	HC	-26978.00	-26297.26	-26074.26
35	ROS	NCL	HC	-26734.32	-26090.21	-25879.21
35	ROS	OSS	HC	-27048.64	-26386.22	-26169.22
35	ROS	SBC	HC	-26962.48	-26287.84	-26066.84

35	ROS	TL	HC	-27025.11	-26313.85	-26080.85
35	SMOTE	Original	HC	-32811.36	-32020.61	-31771.61
35	SMOTE	RUS	HC	-32811.36	-32020.61	-31771.61
35	SMOTE	ENN	HC	-32811.36	-32020.61	-31771.61
35	SMOTE	NCL	HC	-32811.36	-32020.61	-31771.61
35	SMOTE	OSS	HC	-32811.36	-32020.61	-31771.61
35	SMOTE	SBC	HC	-32811.36	-32020.61	-31771.61
35	SMOTE	TL	HC	-32811.36	-32020.61	-31771.61
35	MWMOTE	Original	HC	-29321.72	-28596.70	-28363.70
35	MWMOTE	RUS	HC	-29321.72	-28596.70	-28363.70
35	MWMOTE	ENN	HC	-29321.72	-28596.70	-28363.70
35	MWMOTE	NCL	HC	-29321.72	-28596.70	-28363.70
35	MWMOTE	OSS	HC	-29321.72	-28596.70	-28363.70
35	MWMOTE	SBC	HC	-29321.72	-28596.70	-28363.70
35	MWMOTE	TL	HC	-29321.72	-28596.70	-28363.70
35	ADASYN	Original	HC	-34514.07	-33776.36	-33545.36
35	ADASYN	RUS	HC	-34509.83	-33797.70	-33574.70
35	ADASYN	ENN	HC	-34532.12	-33819.88	-33596.88
35	ADASYN	NCL	HC	-34326.12	-33639.97	-33424.97
35	ADASYN	OSS	HC	-34309.42	-33572.53	-33341.53
35	ADASYN	SBC	HC	-34577.13	-33864.79	-33641.79
35	ADASYN	TL	HC	-34336.06	-33624.66	-33401.66
35	BLSMOTE_1	Original	HC	-26851.44	-26145.65	-25914.65
35	BLSMOTE_1	RUS	HC	-26656.59	-25999.68	-25784.68
35	BLSMOTE_1	ENN	HC	-26715.79	-26028.33	-25803.33

35	BLSMOTE_1	NCL	HC	-27116.12	-26410.33	-26179.33
35	BLSMOTE_1	OSS	HC	-27148.31	-26479.19	-26260.19
35	BLSMOTE_1	SBC	HC	-26799.02	-26081.01	-25846.01
35	BLSMOTE_1	TL	HC	-26626.06	-25895.83	-25656.83
35	BLSMOTE_2	Original	HC	-37054.83	-36296.38	-36061.38
35	BLSMOTE_2	RUS	HC	-37054.83	-36296.38	-36061.38
35	BLSMOTE_2	ENN	HC	-37054.83	-36296.38	-36061.38
35	BLSMOTE_2	NCL	HC	-37054.83	-36296.38	-36061.38
35	BLSMOTE_2	OSS	HC	-37054.83	-36296.38	-36061.38
35	BLSMOTE_2	SBC	HC	-37054.83	-36296.38	-36061.38
35	BLSMOTE_2	TL	HC	-37054.83	-36296.38	-36061.38
40	Original	Original	HC	-29515.91	-28900.23	-28703.23
40	Original	RUS	HC	-29515.91	-28900.23	-28703.23
40	Original	ENN	HC	-29515.91	-28900.23	-28703.23
40	Original	NCL	HC	-29515.91	-28900.23	-28703.23
40	Original	OSS	HC	-29515.91	-28900.23	-28703.23
40	Original	SBC	HC	-29515.91	-28900.23	-28703.23
40	Original	TL	HC	-29515.91	-28900.23	-28703.23
40	ROS	Original	HC	-33502.51	-32760.05	-32525.05
40	ROS	RUS	HC	-33313.56	-32571.11	-32336.11
40	ROS	ENN	HC	-33264.98	-32408.79	-32137.79
40	ROS	NCL	HC	-33363.78	-32596.05	-32353.05
40	ROS	OSS	HC	-33289.64	-32572.47	-32345.47
40	ROS	SBC	HC	-33212.24	-32343.41	-32068.41
40	ROS	TL	HC	-33367.47	-32606.06	-32365.06

40	SMOTE	Original	HC	-35714.16	-34836.13	-34563.13
40	SMOTE	RUS	HC	-35714.16	-34836.13	-34563.13
40	SMOTE	ENN	HC	-35714.16	-34836.13	-34563.13
40	SMOTE	NCL	HC	-35714.16	-34836.13	-34563.13
40	SMOTE	OSS	HC	-35714.16	-34836.13	-34563.13
40	SMOTE	SBC	HC	-35714.16	-34836.13	-34563.13
40	SMOTE	TL	HC	-35714.16	-34836.13	-34563.13
40	MWMOTE	Original	HC	-32748.37	-32056.23	-31837.23
40	MWMOTE	RUS	HC	-32748.37	-32056.23	-31837.23
40	MWMOTE	ENN	HC	-32748.37	-32056.23	-31837.23
40	MWMOTE	NCL	HC	-32748.37	-32056.23	-31837.23
40	MWMOTE	OSS	HC	-32748.37	-32056.23	-31837.23
40	MWMOTE	SBC	HC	-32748.37	-32056.23	-31837.23
40	MWMOTE	TL	HC	-32748.37	-32056.23	-31837.23
40	ADASYN	Original	HC	-36842.24	-35993.72	-35730.72
40	ADASYN	RUS	HC	-36906.13	-36057.28	-35794.28
40	ADASYN	ENN	HC	-35170.78	-34386.88	-34139.88
40	ADASYN	NCL	HC	-36875.47	-36026.68	-35763.68
40	ADASYN	OSS	HC	-36886.76	-36037.97	-35774.97
40	ADASYN	SBC	HC	-36823.00	-35948.56	-35677.56
40	ADASYN	TL	HC	-36888.37	-36039.60	-35776.60
40	BLSMOTE_1	Original	HC	-33493.53	-32699.98	-32448.98
40	BLSMOTE_1	RUS	HC	-33377.58	-32558.74	-32299.74
40	BLSMOTE_1	ENN	HC	-33433.74	-32640.19	-32389.19
40	BLSMOTE_1	NCL	HC	-33633.99	-32865.74	-32622.74

40	BLSMOTE_1	OSS	HC	-33311.84	-32480.35	-32217.35
40	BLSMOTE_1	SBC	HC	-33471.34	-32652.49	-32393.49
40	BLSMOTE_1	TL	HC	-33307.17	-32576.85	-32345.85
40	BLSMOTE_2	Original	HC	-40486.94	-39718.24	-39483.24
40	BLSMOTE_2	RUS	HC	-40486.94	-39718.24	-39483.24
40	BLSMOTE_2	ENN	HC	-40486.94	-39718.24	-39483.24
40	BLSMOTE_2	NCL	HC	-40486.94	-39718.24	-39483.24
40	BLSMOTE_2	OSS	HC	-40486.94	-39718.24	-39483.24
40	BLSMOTE_2	SBC	HC	-40486.94	-39718.24	-39483.24
40	BLSMOTE_2	TL	HC	-40486.94	-39718.24	-39483.24
45	Original	Original	HC	-29515.91	-28900.23	-28703.23
45	Original	RUS	HC	-29515.91	-28900.23	-28703.23
45	Original	ENN	HC	-29515.91	-28900.23	-28703.23
45	Original	NCL	HC	-29515.91	-28900.23	-28703.23
45	Original	OSS	HC	-29515.91	-28900.23	-28703.23
45	Original	SBC	HC	-29515.91	-28900.23	-28703.23
45	Original	TL	HC	-29515.91	-28900.23	-28703.23
45	ROS	Original	HC	-40595.80	-39718.38	-39449.38
45	ROS	RUS	HC	-40612.96	-39709.44	-39432.44
45	ROS	ENN	HC	-40599.19	-39747.87	-39486.87
45	ROS	NCL	HC	-40783.62	-39906.20	-39637.20
45	ROS	OSS	HC	-40571.77	-39707.40	-39442.40
45	ROS	SBC	HC	-40604.33	-39779.10	-39526.10
45	ROS	TL	HC	-40641.47	-39842.34	-39597.34
45	SMOTE	Original	HC	-38985.04	-38049.70	-37762.70

45	SMOTE	RUS	HC	-38985.04	-38049.70	-37762.70
45	SMOTE	ENN	HC	-38985.04	-38049.70	-37762.70
45	SMOTE	NCL	HC	-38985.04	-38049.70	-37762.70
45	SMOTE	OSS	HC	-38985.04	-38049.70	-37762.70
45	SMOTE	SBC	HC	-38985.04	-38049.70	-37762.70
45	SMOTE	TL	HC	-38985.04	-38049.70	-37762.70
45	MWMOTE	Original	HC	-36757.94	-35931.76	-35674.76
45	MWMOTE	RUS	HC	-36757.94	-35931.76	-35674.76
45	MWMOTE	ENN	HC	-36757.94	-35931.76	-35674.76
45	MWMOTE	NCL	HC	-36757.94	-35931.76	-35674.76
45	MWMOTE	OSS	HC	-36757.94	-35931.76	-35674.76
45	MWMOTE	SBC	HC	-36757.94	-35931.76	-35674.76
45	MWMOTE	TL	HC	-36757.94	-35931.76	-35674.76
45	ADASYN	Original	HC	-39101.82	-38207.23	-37932.23
45	ADASYN	RUS	HC	-39166.26	-38271.44	-37996.44
45	ADASYN	ENN	HC	-38706.26	-37814.42	-37539.42
45	ADASYN	NCL	HC	-38715.08	-37821.97	-37546.97
45	ADASYN	OSS	HC	-38980.15	-38138.53	-37879.53
45	ADASYN	SBC	HC	-39007.42	-38113.02	-37838.02
45	ADASYN	TL	HC	-37347.15	-36530.44	-36275.44
45	BLSMOTE_1	Original	HC	-39820.06	-38786.51	-38469.51
45	BLSMOTE_1	RUS	HC	-39656.51	-38681.65	-38382.65
45	BLSMOTE_1	ENN	HC	-39749.84	-38748.90	-38441.90
45	BLSMOTE_1	NCL	HC	-39630.53	-38570.91	-38245.91
45	BLSMOTE_1	OSS	HC	-39569.11	-38561.65	-38252.65

45	BLSMOTE_1	SBC	HC	-39482.55	-38455.53	-38140.53
45	BLSMOTE_1	TL	HC	-39619.18	-38605.20	-38294.20
45	BLSMOTE_2	Original	HC	-44082.88	-43211.70	-42948.70
45	BLSMOTE_2	RUS	HC	-44082.88	-43211.70	-42948.70
45	BLSMOTE_2	ENN	HC	-44082.88	-43211.70	-42948.70
45	BLSMOTE_2	NCL	HC	-44082.88	-43211.70	-42948.70
45	BLSMOTE_2	OSS	HC	-44082.88	-43211.70	-42948.70
45	BLSMOTE_2	SBC	HC	-44082.88	-43211.70	-42948.70
45	BLSMOTE_2	TL	HC	-44082.88	-43211.70	-42948.70
50	Original	Original	HC	-29515.91	-28900.23	-28703.23
50	Original	RUS	HC	-29515.91	-28900.23	-28703.23
50	Original	ENN	HC	-29515.91	-28900.23	-28703.23
50	Original	NCL	HC	-29515.91	-28900.23	-28703.23
50	Original	OSS	HC	-29515.91	-28900.23	-28703.23
50	Original	SBC	HC	-29515.91	-28900.23	-28703.23
50	Original	TL	HC	-29515.91	-28900.23	-28703.23
50	ROS	Original	HC	-49440.75	-48415.31	-48110.31
50	ROS	RUS	HC	-49440.75	-48415.31	-48110.31
50	ROS	ENN	HC	-49440.75	-48415.31	-48110.31
50	ROS	NCL	HC	-49440.75	-48415.31	-48110.31
50	ROS	OSS	HC	-49440.75	-48415.31	-48110.31
50	ROS	SBC	HC	-49440.75	-48415.31	-48110.31
50	ROS	TL	HC	-49440.75	-48415.31	-48110.31
50	SMOTE	Original	HC	-42813.99	-41765.13	-41448.13
50	SMOTE	RUS	HC	-42813.99	-41765.13	-41448.13

50	SMOTE	ENN	HC	-42813.99	-41765.13	-41448.13
50	SMOTE	NCL	HC	-42813.99	-41765.13	-41448.13
50	SMOTE	OSS	HC	-42813.99	-41765.13	-41448.13
50	SMOTE	SBC	HC	-42813.99	-41765.13	-41448.13
50	SMOTE	TL	HC	-42813.99	-41765.13	-41448.13
50	MWMOTE	Original	HC	-40394.14	-39532.48	-39267.48
50	MWMOTE	RUS	HC	-40394.14	-39532.48	-39267.48
50	MWMOTE	ENN	HC	-40394.14	-39532.48	-39267.48
50	MWMOTE	NCL	HC	-40394.14	-39532.48	-39267.48
50	MWMOTE	OSS	HC	-40394.14	-39532.48	-39267.48
50	MWMOTE	SBC	HC	-40394.14	-39532.48	-39267.48
50	MWMOTE	TL	HC	-40394.14	-39532.48	-39267.48
50	ADASYN	Original	HC	-40114.05	-39170.74	-38879.74
50	ADASYN	RUS	HC	-41797.87	-40893.51	-40618.51
50	ADASYN	ENN	HC	-41828.14	-40923.70	-40648.70
50	ADASYN	NCL	HC	-41846.07	-40948.15	-40675.15
50	ADASYN	OSS	HC	-41846.07	-40948.15	-40675.15
50	ADASYN	SBC	HC	-41816.38	-40911.97	-40636.97
50	ADASYN	TL	HC	-41371.74	-40468.97	-40193.97
50	BLSMOTE_1	Original	HC	-48071.91	-47020.28	-46707.28
50	BLSMOTE_1	RUS	HC	-48084.14	-46985.48	-46658.48
50	BLSMOTE_1	ENN	HC	-48024.73	-46731.20	-46346.20
50	BLSMOTE_1	NCL	HC	-47919.91	-46767.49	-46424.49
50	BLSMOTE_1	OSS	HC	-48208.83	-47069.85	-46730.85
50	BLSMOTE_1	SBC	HC	-48150.82	-46984.96	-46637.96

50	BLSMOTE_1	TL	HC	-48122.19	-46996.65	-46661.65
50	BLSMOTE_2	Original	HC	-48518.79	-47715.40	-47476.40
50	BLSMOTE_2	RUS	HC	-48518.79	-47715.40	-47476.40
50	BLSMOTE_2	ENN	HC	-48518.79	-47715.40	-47476.40
50	BLSMOTE_2	NCL	HC	-48518.79	-47715.40	-47476.40
50	BLSMOTE_2	OSS	HC	-48518.79	-47715.40	-47476.40
50	BLSMOTE_2	SBC	HC	-48518.79	-47715.40	-47476.40
50	BLSMOTE_2	TL	HC	-48518.79	-47715.40	-47476.40
35	Original	Original	Tabu	-29502.71	-28874.52	-28673.52
35	Original	RUS	Tabu	-29502.71	-28874.52	-28673.52
35	Original	ENN	Tabu	-29502.71	-28874.52	-28673.52
35	Original	NCL	Tabu	-29502.71	-28874.52	-28673.52
35	Original	OSS	Tabu	-29502.71	-28874.52	-28673.52
35	Original	SBC	Tabu	-29502.71	-28874.52	-28673.52
35	Original	TL	Tabu	-29502.71	-28874.52	-28673.52
35	ROS	Original	Tabu	-26899.87	-26267.97	-26060.97
35	ROS	RUS	Tabu	-26991.59	-26353.59	-26144.59
35	ROS	ENN	Tabu	-26973.85	-26287.01	-26062.01
35	ROS	NCL	Tabu	-26726.12	-26051.48	-25830.48
35	ROS	OSS	Tabu	-27048.64	-26386.22	-26169.22
35	ROS	SBC	Tabu	-26944.51	-26251.56	-26024.56
35	ROS	TL	Tabu	-27012.01	-26306.84	-26075.84
35	SMOTE	Original	Tabu	-32801.60	-32004.51	-31753.51
35	SMOTE	RUS	Tabu	-32801.60	-32004.51	-31753.51
35	SMOTE	ENN	Tabu	-32801.60	-32004.51	-31753.51

35	SMOTE	NCL	Tabu	-32801.60	-32004.51	-31753.51
35	SMOTE	OSS	Tabu	-32801.60	-32004.51	-31753.51
35	SMOTE	SBC	Tabu	-32801.60	-32004.51	-31753.51
35	SMOTE	TL	Tabu	-32801.60	-32004.51	-31753.51
35	MWMOTE	Original	Tabu	-29293.51	-28543.59	-28302.59
35	MWMOTE	RUS	Tabu	-29293.51	-28543.59	-28302.59
35	MWMOTE	ENN	Tabu	-29293.51	-28543.59	-28302.59
35	MWMOTE	NCL	Tabu	-29293.51	-28543.59	-28302.59
35	MWMOTE	OSS	Tabu	-29293.51	-28543.59	-28302.59
35	MWMOTE	SBC	Tabu	-29293.51	-28543.59	-28302.59
35	MWMOTE	TL	Tabu	-29293.51	-28543.59	-28302.59
35	ADASYN	Original	Tabu	-34514.07	-33776.36	-33545.36
35	ADASYN	RUS	Tabu	-34509.83	-33797.70	-33574.70
35	ADASYN	ENN	Tabu	-34532.12	-33819.88	-33596.88
35	ADASYN	NCL	Tabu	-34326.12	-33639.97	-33424.97
35	ADASYN	OSS	Tabu	-34309.42	-33572.53	-33341.53
35	ADASYN	SBC	Tabu	-34577.13	-33864.79	-33641.79
35	ADASYN	TL	Tabu	-34336.06	-33624.66	-33401.66
35	BLSMOTE_1	Original	Tabu	-26834.11	-26109.99	-25872.99
35	BLSMOTE_1	RUS	Tabu	-26649.47	-25974.23	-25753.23
35	BLSMOTE_1	ENN	Tabu	-26709.74	-26034.50	-25813.50
35	BLSMOTE_1	NCL	Tabu	-27061.69	-26349.79	-26116.79
35	BLSMOTE_1	OSS	Tabu	-27119.91	-26420.23	-26191.23
35	BLSMOTE_1	SBC	Tabu	-26791.88	-26067.76	-25830.76
35	BLSMOTE_1	TL	Tabu	-26603.89	-25879.77	-25642.77

35	BLSMOTE_2	Original	Tabu	-37053.86	-36298.64	-36064.64
35	BLSMOTE_2	RUS	Tabu	-37053.86	-36298.64	-36064.64
35	BLSMOTE_2	ENN	Tabu	-37053.86	-36298.64	-36064.64
35	BLSMOTE_2	NCL	Tabu	-37053.86	-36298.64	-36064.64
35	BLSMOTE_2	OSS	Tabu	-37053.86	-36298.64	-36064.64
35	BLSMOTE_2	SBC	Tabu	-37053.86	-36298.64	-36064.64
35	BLSMOTE_2	TL	Tabu	-37053.86	-36298.64	-36064.64
40	Original	Original	Tabu	-29502.71	-28874.52	-28673.52
40	Original	RUS	Tabu	-29502.71	-28874.52	-28673.52
40	Original	ENN	Tabu	-29502.71	-28874.52	-28673.52
40	Original	NCL	Tabu	-29502.71	-28874.52	-28673.52
40	Original	OSS	Tabu	-29502.71	-28874.52	-28673.52
40	Original	SBC	Tabu	-29502.71	-28874.52	-28673.52
40	Original	TL	Tabu	-29502.71	-28874.52	-28673.52
40	ROS	Original	Tabu	-33483.22	-32766.04	-32539.04
40	ROS	RUS	Tabu	-33313.56	-32571.11	-32336.11
40	ROS	ENN	Tabu	-33231.34	-32425.70	-32170.70
40	ROS	NCL	Tabu	-33363.78	-32596.05	-32353.05
40	ROS	OSS	Tabu	-33289.64	-32572.47	-32345.47
40	ROS	SBC	Tabu	-33173.42	-32355.15	-32096.15
40	ROS	TL	Tabu	-33322.94	-32542.58	-32295.58
40	SMOTE	Original	Tabu	-35706.03	-34821.56	-34546.56
40	SMOTE	RUS	Tabu	-35706.03	-34821.56	-34546.56
40	SMOTE	ENN	Tabu	-35706.03	-34821.56	-34546.56
40	SMOTE	NCL	Tabu	-35706.03	-34821.56	-34546.56

40	SMOTE	OSS	Tabu	-35706.03	-34821.56	-34546.56
40	SMOTE	SBC	Tabu	-35706.03	-34821.56	-34546.56
40	SMOTE	TL	Tabu	-35706.03	-34821.56	-34546.56
40	MWMOTE	Original	Tabu	-32739.25	-32015.50	-31786.50
40	MWMOTE	RUS	Tabu	-32739.25	-32015.50	-31786.50
40	MWMOTE	ENN	Tabu	-32739.25	-32015.50	-31786.50
40	MWMOTE	NCL	Tabu	-32739.25	-32015.50	-31786.50
40	MWMOTE	OSS	Tabu	-32739.25	-32015.50	-31786.50
40	MWMOTE	SBC	Tabu	-32739.25	-32015.50	-31786.50
40	MWMOTE	TL	Tabu	-32739.25	-32015.50	-31786.50
40	ADASYN	Original	Tabu	-36842.24	-35993.72	-35730.72
40	ADASYN	RUS	Tabu	-36906.13	-36057.28	-35794.28
40	ADASYN	ENN	Tabu	-35170.78	-34386.88	-34139.88
40	ADASYN	NCL	Tabu	-36875.47	-36026.68	-35763.68
40	ADASYN	OSS	Tabu	-36886.76	-36037.97	-35774.97
40	ADASYN	SBC	Tabu	-36823.00	-35948.56	-35677.56
40	ADASYN	TL	Tabu	-36888.37	-36039.60	-35776.60
40	BLSMOTE_1	Original	Tabu	-33484.52	-32678.32	-32423.32
40	BLSMOTE_1	RUS	Tabu	-33366.70	-32541.53	-32280.53
40	BLSMOTE_1	ENN	Tabu	-33414.96	-32615.08	-32362.08
40	BLSMOTE_1	NCL	Tabu	-33621.53	-32846.95	-32601.95
40	BLSMOTE_1	OSS	Tabu	-33300.69	-32462.88	-32197.88
40	BLSMOTE_1	SBC	Tabu	-33439.38	-32601.57	-32336.57
40	BLSMOTE_1	TL	Tabu	-33307.17	-32576.85	-32345.85
40	BLSMOTE_2	Original	Tabu	-40486.94	-39718.24	-39483.24

40	BLSMOTE_2	RUS	Tabu	-40486.94	-39718.24	-39483.24
40	BLSMOTE_2	ENN	Tabu	-40486.94	-39718.24	-39483.24
40	BLSMOTE_2	NCL	Tabu	-40486.94	-39718.24	-39483.24
40	BLSMOTE_2	OSS	Tabu	-40486.94	-39718.24	-39483.24
40	BLSMOTE_2	SBC	Tabu	-40486.94	-39718.24	-39483.24
40	BLSMOTE_2	TL	Tabu	-40486.94	-39718.24	-39483.24
45	Original	Original	Tabu	-29502.71	-28874.52	-28673.52
45	Original	RUS	Tabu	-29502.71	-28874.52	-28673.52
45	Original	ENN	Tabu	-29502.71	-28874.52	-28673.52
45	Original	NCL	Tabu	-29502.71	-28874.52	-28673.52
45	Original	OSS	Tabu	-29502.71	-28874.52	-28673.52
45	Original	SBC	Tabu	-29502.71	-28874.52	-28673.52
45	Original	TL	Tabu	-29502.71	-28874.52	-28673.52
45	ROS	Original	Tabu	-40586.10	-39702.15	-39431.15
45	ROS	RUS	Tabu	-40606.39	-39696.35	-39417.35
45	ROS	ENN	Tabu	-40588.30	-39717.40	-39450.40
45	ROS	NCL	Tabu	-40772.08	-39888.13	-39617.13
45	ROS	OSS	Tabu	-40513.69	-39590.60	-39307.60
45	ROS	SBC	Tabu	-40566.02	-39708.17	-39445.17
45	ROS	TL	Tabu	-40602.22	-39770.46	-39515.46
45	SMOTE	Original	Tabu	-38956.52	-37988.60	-37691.60
45	SMOTE	RUS	Tabu	-38956.52	-37988.60	-37691.60
45	SMOTE	ENN	Tabu	-38956.52	-37988.60	-37691.60
45	SMOTE	NCL	Tabu	-38956.52	-37988.60	-37691.60
45	SMOTE	OSS	Tabu	-38956.52	-37988.60	-37691.60

45	SMOTE	SBC	Tabu	-38956.52	-37988.60	-37691.60
45	SMOTE	TL	Tabu	-38956.52	-37988.60	-37691.60
45	MWMOTE	Original	Tabu	-36757.94	-35931.76	-35674.76
45	MWMOTE	RUS	Tabu	-36757.94	-35931.76	-35674.76
45	MWMOTE	ENN	Tabu	-36757.94	-35931.76	-35674.76
45	MWMOTE	NCL	Tabu	-36757.94	-35931.76	-35674.76
45	MWMOTE	OSS	Tabu	-36757.94	-35931.76	-35674.76
45	MWMOTE	SBC	Tabu	-36757.94	-35931.76	-35674.76
45	MWMOTE	TL	Tabu	-36757.94	-35931.76	-35674.76
45	ADASYN	Original	Tabu	-39086.84	-38185.74	-37908.74
45	ADASYN	RUS	Tabu	-39151.27	-38249.95	-37972.95
45	ADASYN	ENN	Tabu	-38692.63	-37794.30	-37517.30
45	ADASYN	NCL	Tabu	-38700.19	-37800.58	-37523.58
45	ADASYN	OSS	Tabu	-38965.71	-38117.59	-37856.59
45	ADASYN	SBC	Tabu	-38992.41	-38091.51	-37814.51
45	ADASYN	TL	Tabu	-37346.16	-36523.04	-36266.04
45	BLSMOTE_1	Original	Tabu	-39819.82	-38779.75	-38460.75
45	BLSMOTE_1	RUS	Tabu	-39616.85	-38583.31	-38266.31
45	BLSMOTE_1	ENN	Tabu	-39748.51	-38741.05	-38432.05
45	BLSMOTE_1	NCL	Tabu	-39596.20	-38477.89	-38134.89
45	BLSMOTE_1	OSS	Tabu	-39557.33	-38543.34	-38232.34
45	BLSMOTE_1	SBC	Tabu	-39475.77	-38435.70	-38116.70
45	BLSMOTE_1	TL	Tabu	-39619.15	-38592.13	-38277.13
45	BLSMOTE_2	Original	Tabu	-44082.88	-43211.70	-42948.70
45	BLSMOTE_2	RUS	Tabu	-44082.88	-43211.70	-42948.70

45	BLSMOTE_2	ENN	Tabu	-44082.88	-43211.70	-42948.70
45	BLSMOTE_2	NCL	Tabu	-44082.88	-43211.70	-42948.70
45	BLSMOTE_2	OSS	Tabu	-44082.88	-43211.70	-42948.70
45	BLSMOTE_2	SBC	Tabu	-44082.88	-43211.70	-42948.70
45	BLSMOTE_2	TL	Tabu	-44082.88	-43211.70	-42948.70
50	Original	Original	Tabu	-29502.71	-28874.52	-28673.52
50	Original	RUS	Tabu	-29502.71	-28874.52	-28673.52
50	Original	ENN	Tabu	-29502.71	-28874.52	-28673.52
50	Original	NCL	Tabu	-29502.71	-28874.52	-28673.52
50	Original	OSS	Tabu	-29502.71	-28874.52	-28673.52
50	Original	SBC	Tabu	-29502.71	-28874.52	-28673.52
50	Original	TL	Tabu	-29502.71	-28874.52	-28673.52
50	ROS	Original	Tabu	-49428.99	-48396.83	-48089.83
50	ROS	RUS	Tabu	-49428.99	-48396.83	-48089.83
50	ROS	ENN	Tabu	-49428.99	-48396.83	-48089.83
50	ROS	NCL	Tabu	-49428.99	-48396.83	-48089.83
50	ROS	OSS	Tabu	-49428.99	-48396.83	-48089.83
50	ROS	SBC	Tabu	-49428.99	-48396.83	-48089.83
50	ROS	TL	Tabu	-49428.99	-48396.83	-48089.83
50	SMOTE	Original	Tabu	-42813.99	-41765.13	-41448.13
50	SMOTE	RUS	Tabu	-42813.99	-41765.13	-41448.13
50	SMOTE	ENN	Tabu	-42813.99	-41765.13	-41448.13
50	SMOTE	NCL	Tabu	-42813.99	-41765.13	-41448.13
50	SMOTE	OSS	Tabu	-42813.99	-41765.13	-41448.13
50	SMOTE	SBC	Tabu	-42813.99	-41765.13	-41448.13

50	SMOTE	TL	Tabu	-42813.99	-41765.13	-41448.13
50	MWMOTE	Original	Tabu	-40394.14	-39532.48	-39267.48
50	MWMOTE	RUS	Tabu	-40394.14	-39532.48	-39267.48
50	MWMOTE	ENN	Tabu	-40394.14	-39532.48	-39267.48
50	MWMOTE	NCL	Tabu	-40394.14	-39532.48	-39267.48
50	MWMOTE	OSS	Tabu	-40394.14	-39532.48	-39267.48
50	MWMOTE	SBC	Tabu	-40394.14	-39532.48	-39267.48
50	MWMOTE	TL	Tabu	-40394.14	-39532.48	-39267.48
50	ADASYN	Original	Tabu	-40113.95	-39164.16	-38871.16
50	ADASYN	RUS	Tabu	-41780.48	-40869.55	-40592.55
50	ADASYN	ENN	Tabu	-41810.82	-40899.80	-40622.80
50	ADASYN	NCL	Tabu	-41820.29	-40863.18	-40572.18
50	ADASYN	OSS	Tabu	-41820.29	-40863.18	-40572.18
50	ADASYN	SBC	Tabu	-41799.71	-40888.72	-40611.72
50	ADASYN	TL	Tabu	-41345.08	-40383.22	-40090.22
50	BLSMOTE_1	Original	Tabu	-48051.37	-46979.59	-46660.59
50	BLSMOTE_1	RUS	Tabu	-48058.26	-46973.04	-46650.04
50	BLSMOTE_1	ENN	Tabu	-48008.08	-46680.95	-46285.95
50	BLSMOTE_1	NCL	Tabu	-47915.05	-46749.20	-46402.20
50	BLSMOTE_1	OSS	Tabu	-48195.80	-47043.38	-46700.38
50	BLSMOTE_1	SBC	Tabu	-48137.33	-46958.03	-46607.03
50	BLSMOTE_1	TL	Tabu	-48085.41	-46959.86	-46624.86
50	BLSMOTE_2	Original	Tabu	-48512.83	-47696.00	-47453.00
50	BLSMOTE_2	RUS	Tabu	-48512.83	-47696.00	-47453.00
50	BLSMOTE_2	ENN	Tabu	-48512.83	-47696.00	-47453.00

50	BLSMOTE_2	NCL	Tabu	-48512.83	-47696.00	-47453.00
50	BLSMOTE_2	OSS	Tabu	-48512.83	-47696.00	-47453.00
50	BLSMOTE_2	SBC	Tabu	-48512.83	-47696.00	-47453.00
50	BLSMOTE_2	TL	Tabu	-48512.83	-47696.00	-47453.00
35	Original	Original	MMHC	-29953.83	-29622.55	-29516.55
35	Original	RUS	MMHC	-29953.83	-29622.55	-29516.55
35	Original	ENN	MMHC	-29953.83	-29622.55	-29516.55
35	Original	NCL	MMHC	-29953.83	-29622.55	-29516.55
35	Original	OSS	MMHC	-29953.83	-29622.55	-29516.55
35	Original	SBC	MMHC	-29953.83	-29622.55	-29516.55
35	Original	TL	MMHC	-29953.83	-29622.55	-29516.55
35	ROS	Original	MMHC	-27531.28	-27149.70	-27024.70
35	ROS	RUS	MMHC	-27598.79	-27290.47	-27189.47
35	ROS	ENN	MMHC	-27612.68	-27231.10	-27106.10
35	ROS	NCL	MMHC	-27242.90	-26901.00	-26789.00
35	ROS	OSS	MMHC	-28004.46	-27668.67	-27558.67
35	ROS	SBC	MMHC	-27599.23	-27284.81	-27181.81
35	ROS	TL	MMHC	-27589.82	-27235.71	-27119.71
35	SMOTE	Original	MMHC	-33772.14	-33359.30	-33229.30
35	SMOTE	RUS	MMHC	-33772.14	-33359.30	-33229.30
35	SMOTE	ENN	MMHC	-33772.14	-33359.30	-33229.30
35	SMOTE	NCL	MMHC	-33772.14	-33359.30	-33229.30
35	SMOTE	OSS	MMHC	-33772.14	-33359.30	-33229.30
35	SMOTE	SBC	MMHC	-33772.14	-33359.30	-33229.30
35	SMOTE	TL	MMHC	-33772.14	-33359.30	-33229.30

35	MWMOTE	Original	MMHC	-30127.55	-29766.60	-29650.60
35	MWMOTE	RUS	MMHC	-30127.55	-29766.60	-29650.60
35	MWMOTE	ENN	MMHC	-30127.55	-29766.60	-29650.60
35	MWMOTE	NCL	MMHC	-30127.55	-29766.60	-29650.60
35	MWMOTE	OSS	MMHC	-30127.55	-29766.60	-29650.60
35	MWMOTE	SBC	MMHC	-30127.55	-29766.60	-29650.60
35	MWMOTE	TL	MMHC	-30127.55	-29766.60	-29650.60
35	ADASYN	Original	MMHC	-35245.06	-34842.67	-34716.67
35	ADASYN	RUS	MMHC	-35202.92	-34781.39	-34649.39
35	ADASYN	ENN	MMHC	-35216.08	-34791.29	-34658.29
35	ADASYN	NCL	MMHC	-35260.21	-34870.86	-34748.86
35	ADASYN	OSS	MMHC	-35009.86	-34588.78	-34456.78
35	ADASYN	SBC	MMHC	-35272.74	-34851.08	-34719.08
35	ADASYN	TL	MMHC	-35045.44	-34624.34	-34492.34
35	BLSMOTE_1	Original	MMHC	-27604.91	-27210.76	-27081.76
35	BLSMOTE_1	RUS	MMHC	-27411.68	-27014.48	-26884.48
35	BLSMOTE_1	ENN	MMHC	-27872.86	-27536.77	-27426.77
35	BLSMOTE_1	NCL	MMHC	-27603.21	-27221.29	-27096.29
35	BLSMOTE_1	OSS	MMHC	-28046.36	-27701.10	-27588.10
35	BLSMOTE_1	SBC	MMHC	-27661.66	-27291.96	-27170.96
35	BLSMOTE_1	TL	MMHC	-27121.68	-26739.76	-26614.76
35	BLSMOTE_2	Original	MMHC	-37736.37	-37320.03	-37191.03
35	BLSMOTE_2	RUS	MMHC	-37736.37	-37320.03	-37191.03
35	BLSMOTE_2	ENN	MMHC	-37736.37	-37320.03	-37191.03
35	BLSMOTE_2	NCL	MMHC	-37736.37	-37320.03	-37191.03

35	BLSMOTE_2	OSS	MMHC	-37736.37	-37320.03	-37191.03
35	BLSMOTE_2	SBC	MMHC	-37736.37	-37320.03	-37191.03
35	BLSMOTE_2	TL	MMHC	-37736.37	-37320.03	-37191.03
40	Original	Original	MMHC	-29953.83	-29622.55	-29516.55
40	Original	RUS	MMHC	-29953.83	-29622.55	-29516.55
40	Original	ENN	MMHC	-29953.83	-29622.55	-29516.55
40	Original	NCL	MMHC	-29953.83	-29622.55	-29516.55
40	Original	OSS	MMHC	-29953.83	-29622.55	-29516.55
40	Original	SBC	MMHC	-29953.83	-29622.55	-29516.55
40	Original	TL	MMHC	-29953.83	-29622.55	-29516.55
40	ROS	Original	MMHC	-34235.83	-33790.36	-33649.36
40	ROS	RUS	MMHC	-34112.54	-33698.67	-33567.67
40	ROS	ENN	MMHC	-33905.38	-33459.91	-33318.91
40	ROS	NCL	MMHC	-34110.70	-33690.50	-33557.50
40	ROS	OSS	MMHC	-33976.81	-33550.29	-33415.29
40	ROS	SBC	MMHC	-34208.46	-33744.03	-33597.03
40	ROS	TL	MMHC	-33991.84	-33552.69	-33413.69
40	SMOTE	Original	MMHC	-36474.70	-36040.50	-35905.50
40	SMOTE	RUS	MMHC	-36474.70	-36040.50	-35905.50
40	SMOTE	ENN	MMHC	-36474.70	-36040.50	-35905.50
40	SMOTE	NCL	MMHC	-36474.70	-36040.50	-35905.50
40	SMOTE	OSS	MMHC	-36474.70	-36040.50	-35905.50
40	SMOTE	SBC	MMHC	-36474.70	-36040.50	-35905.50
40	SMOTE	TL	MMHC	-36474.70	-36040.50	-35905.50
40	MWMOTE	Original	MMHC	-33479.10	-33071.40	-32942.40

40	MWMOTE	RUS	MMHC	-33479.10	-33071.40	-32942.40
40	MWMOTE	ENN	MMHC	-33479.10	-33071.40	-32942.40
40	MWMOTE	NCL	MMHC	-33479.10	-33071.40	-32942.40
40	MWMOTE	OSS	MMHC	-33479.10	-33071.40	-32942.40
40	MWMOTE	SBC	MMHC	-33479.10	-33071.40	-32942.40
40	MWMOTE	TL	MMHC	-33479.10	-33071.40	-32942.40
40	ADASYN	Original	MMHC	-37481.80	-37062.38	-36932.38
40	ADASYN	RUS	MMHC	-37508.34	-37030.66	-36882.66
40	ADASYN	ENN	MMHC	-36028.11	-35577.45	-35435.45
40	ADASYN	NCL	MMHC	-37470.48	-36992.83	-36844.83
40	ADASYN	OSS	MMHC	-37488.77	-37011.12	-36863.12
40	ADASYN	SBC	MMHC	-37458.56	-37013.27	-36875.27
40	ADASYN	TL	MMHC	-37550.36	-37111.45	-36975.45
40	BLSMOTE_1	Original	MMHC	-34471.71	-34067.03	-33939.03
40	BLSMOTE_1	RUS	MMHC	-34065.09	-33635.12	-33499.12
40	BLSMOTE_1	ENN	MMHC	-34263.14	-33845.81	-33713.81
40	BLSMOTE_1	NCL	MMHC	-34373.93	-33937.63	-33799.63
40	BLSMOTE_1	OSS	MMHC	-34119.89	-33721.53	-33595.53
40	BLSMOTE_1	SBC	MMHC	-34648.29	-34230.97	-34098.97
40	BLSMOTE_1	TL	MMHC	-34694.96	-34318.73	-34199.73
40	BLSMOTE_2	Original	MMHC	-41363.04	-40914.91	-40777.91
40	BLSMOTE_2	RUS	MMHC	-41363.04	-40914.91	-40777.91
40	BLSMOTE_2	ENN	MMHC	-41363.04	-40914.91	-40777.91
40	BLSMOTE_2	NCL	MMHC	-41363.04	-40914.91	-40777.91
40	BLSMOTE_2	OSS	MMHC	-41363.04	-40914.91	-40777.91

40	BLSMOTE_2	SBC	MMHC	-41363.04	-40914.91	-40777.91
40	BLSMOTE_2	TL	MMHC	-41363.04	-40914.91	-40777.91
45	Original	Original	MMHC	-29953.83	-29622.55	-29516.55
45	Original	RUS	MMHC	-29953.83	-29622.55	-29516.55
45	Original	ENN	MMHC	-29953.83	-29622.55	-29516.55
45	Original	NCL	MMHC	-29953.83	-29622.55	-29516.55
45	Original	OSS	MMHC	-29953.83	-29622.55	-29516.55
45	Original	SBC	MMHC	-29953.83	-29622.55	-29516.55
45	Original	TL	MMHC	-29953.83	-29622.55	-29516.55
45	ROS	Original	MMHC	-41271.07	-40739.40	-40576.40
45	ROS	RUS	MMHC	-41356.61	-40857.56	-40704.56
45	ROS	ENN	MMHC	-41354.37	-40861.84	-40710.84
45	ROS	NCL	MMHC	-41560.93	-41081.45	-40934.45
45	ROS	OSS	MMHC	-41519.81	-41059.89	-40918.89
45	ROS	SBC	MMHC	-41663.83	-41243.06	-41114.06
45	ROS	TL	MMHC	-41491.09	-40985.51	-40830.51
45	SMOTE	Original	MMHC	-39934.01	-39490.78	-39354.78
45	SMOTE	RUS	MMHC	-39934.01	-39490.78	-39354.78
45	SMOTE	ENN	MMHC	-39934.01	-39490.78	-39354.78
45	SMOTE	NCL	MMHC	-39934.01	-39490.78	-39354.78
45	SMOTE	OSS	MMHC	-39934.01	-39490.78	-39354.78
45	SMOTE	SBC	MMHC	-39934.01	-39490.78	-39354.78
45	SMOTE	TL	MMHC	-39934.01	-39490.78	-39354.78
45	MWMOTE	Original	MMHC	-37435.36	-36978.87	-36836.87
45	MWMOTE	RUS	MMHC	-37435.36	-36978.87	-36836.87

45	MWMOTE	ENN	MMHC	-37435.36	-36978.87	-36836.87
45	MWMOTE	NCL	MMHC	-37435.36	-36978.87	-36836.87
45	MWMOTE	OSS	MMHC	-37435.36	-36978.87	-36836.87
45	MWMOTE	SBC	MMHC	-37435.36	-36978.87	-36836.87
45	MWMOTE	TL	MMHC	-37435.36	-36978.87	-36836.87
45	ADASYN	Original	MMHC	-39860.52	-39411.60	-39273.60
45	ADASYN	RUS	MMHC	-39865.19	-39406.40	-39265.40
45	ADASYN	ENN	MMHC	-39452.55	-39001.76	-38862.76
45	ADASYN	NCL	MMHC	-39561.01	-39109.58	-38970.58
45	ADASYN	OSS	MMHC	-39733.39	-39288.21	-39151.21
45	ADASYN	SBC	MMHC	-39707.82	-39249.23	-39108.23
45	ADASYN	TL	MMHC	-38460.45	-38053.70	-37926.70
45	BLSMOTE_1	Original	MMHC	-41246.29	-40750.71	-40598.71
45	BLSMOTE_1	RUS	MMHC	-41087.74	-40562.82	-40401.82
45	BLSMOTE_1	ENN	MMHC	-41254.08	-40810.67	-40674.67
45	BLSMOTE_1	NCL	MMHC	-40781.76	-40295.96	-40146.96
45	BLSMOTE_1	OSS	MMHC	-41860.28	-41420.13	-41285.13
45	BLSMOTE_1	SBC	MMHC	-40799.26	-40284.12	-40126.12
45	BLSMOTE_1	TL	MMHC	-41471.51	-41063.96	-40938.96
45	BLSMOTE_2	Original	MMHC	-44871.24	-44427.37	-44293.37
45	BLSMOTE_2	RUS	MMHC	-44871.24	-44427.37	-44293.37
45	BLSMOTE_2	ENN	MMHC	-44871.24	-44427.37	-44293.37
45	BLSMOTE_2	NCL	MMHC	-44871.24	-44427.37	-44293.37
45	BLSMOTE_2	OSS	MMHC	-44871.24	-44427.37	-44293.37
45	BLSMOTE_2	SBC	MMHC	-44871.24	-44427.37	-44293.37

45	BLSMOTE_2	TL	MMHC	-44871.24	-44427.37	-44293.37
50	Original	Original	MMHC	-29953.83	-29622.55	-29516.55
50	Original	RUS	MMHC	-29953.83	-29622.55	-29516.55
50	Original	ENN	MMHC	-29953.83	-29622.55	-29516.55
50	Original	NCL	MMHC	-29953.83	-29622.55	-29516.55
50	Original	OSS	MMHC	-29953.83	-29622.55	-29516.55
50	Original	SBC	MMHC	-29953.83	-29622.55	-29516.55
50	Original	TL	MMHC	-29953.83	-29622.55	-29516.55
50	ROS	Original	MMHC	-50788.49	-50307.71	-50164.71
50	ROS	RUS	MMHC	-50788.49	-50307.71	-50164.71
50	ROS	ENN	MMHC	-50788.49	-50307.71	-50164.71
50	ROS	NCL	MMHC	-50788.49	-50307.71	-50164.71
50	ROS	OSS	MMHC	-50788.49	-50307.71	-50164.71
50	ROS	SBC	MMHC	-50788.49	-50307.71	-50164.71
50	ROS	TL	MMHC	-50788.49	-50307.71	-50164.71
50	SMOTE	Original	MMHC	-44089.21	-43579.68	-43425.68
50	SMOTE	RUS	MMHC	-44089.21	-43579.68	-43425.68
50	SMOTE	ENN	MMHC	-44089.21	-43579.68	-43425.68
50	SMOTE	NCL	MMHC	-44089.21	-43579.68	-43425.68
50	SMOTE	OSS	MMHC	-44089.21	-43579.68	-43425.68
50	SMOTE	SBC	MMHC	-44089.21	-43579.68	-43425.68
50	SMOTE	TL	MMHC	-44089.21	-43579.68	-43425.68
50	MWMOTE	Original	MMHC	-40958.32	-40483.59	-40337.59
50	MWMOTE	RUS	MMHC	-40958.32	-40483.59	-40337.59
50	MWMOTE	ENN	MMHC	-40958.32	-40483.59	-40337.59

50	MWMOTE	NCL	MMHC	-40958.32	-40483.59	-40337.59
50	MWMOTE	OSS	MMHC	-40958.32	-40483.59	-40337.59
50	MWMOTE	SBC	MMHC	-40958.32	-40483.59	-40337.59
50	MWMOTE	TL	MMHC	-40958.32	-40483.59	-40337.59
50	ADASYN	Original	MMHC	-41023.87	-40560.31	-40417.31
50	ADASYN	RUS	MMHC	-42648.54	-42161.83	-42013.83
50	ADASYN	ENN	MMHC	-42679.12	-42192.37	-42044.37
50	ADASYN	NCL	MMHC	-42699.54	-42212.76	-42064.76
50	ADASYN	OSS	MMHC	-42699.54	-42212.76	-42064.76
50	ADASYN	SBC	MMHC	-42750.27	-42270.11	-42124.11
50	ADASYN	TL	MMHC	-42216.29	-41730.44	-41582.44
50	BLSMOTE_1	Original	MMHC	-49334.55	-48827.22	-48676.22
50	BLSMOTE_1	RUS	MMHC	-49279.81	-48765.76	-48612.76
50	BLSMOTE_1	ENN	MMHC	-49429.55	-48885.26	-48723.26
50	BLSMOTE_1	NCL	MMHC	-49246.14	-48638.01	-48457.01
50	BLSMOTE_1	OSS	MMHC	-49866.93	-49339.44	-49182.44
50	BLSMOTE_1	SBC	MMHC	-50211.57	-49794.95	-49670.95
50	BLSMOTE_1	TL	MMHC	-49816.44	-49339.34	-49197.34
50	BLSMOTE_2	Original	MMHC	-49260.69	-48769.92	-48623.92
50	BLSMOTE_2	RUS	MMHC	-49260.69	-48769.92	-48623.92
50	BLSMOTE_2	ENN	MMHC	-49260.69	-48769.92	-48623.92
50	BLSMOTE_2	NCL	MMHC	-49260.69	-48769.92	-48623.92
50	BLSMOTE_2	OSS	MMHC	-49260.69	-48769.92	-48623.92
50	BLSMOTE_2	SBC	MMHC	-49260.69	-48769.92	-48623.92
50	BLSMOTE_2	TL	MMHC	-49260.69	-48769.92	-48623.92

35	Original	Original	RSMAX2	-30634.15	-30374.75	-30291.75
35	Original	RUS	RSMAX2	-30634.15	-30374.75	-30291.75
35	Original	ENN	RSMAX2	-30634.15	-30374.75	-30291.75
35	Original	NCL	RSMAX2	-30634.15	-30374.75	-30291.75
35	Original	OSS	RSMAX2	-30634.15	-30374.75	-30291.75
35	Original	SBC	RSMAX2	-30634.15	-30374.75	-30291.75
35	Original	TL	RSMAX2	-30634.15	-30374.75	-30291.75
35	ROS	Original	RSMAX2	-28542.55	-28295.29	-28214.29
35	ROS	RUS	RSMAX2	-29188.51	-28965.67	-28892.67
35	ROS	ENN	RSMAX2	-28881.92	-28656.02	-28582.02
35	ROS	NCL	RSMAX2	-28099.53	-27855.31	-27775.31
35	ROS	OSS	RSMAX2	-28845.75	-28577.12	-28489.12
35	ROS	SBC	RSMAX2	-28423.71	-28167.29	-28083.29
35	ROS	TL	RSMAX2	-28422.32	-28190.32	-28114.32
35	SMOTE	Original	RSMAX2	-34500.90	-34240.49	-34158.49
35	SMOTE	RUS	RSMAX2	-34500.90	-34240.49	-34158.49
35	SMOTE	ENN	RSMAX2	-34500.90	-34240.49	-34158.49
35	SMOTE	NCL	RSMAX2	-34500.90	-34240.49	-34158.49
35	SMOTE	OSS	RSMAX2	-34500.90	-34240.49	-34158.49
35	SMOTE	SBC	RSMAX2	-34500.90	-34240.49	-34158.49
35	SMOTE	TL	RSMAX2	-34500.90	-34240.49	-34158.49
35	MWMOTE	Original	RSMAX2	-30555.07	-30306.13	-30226.13
35	MWMOTE	RUS	RSMAX2	-30555.07	-30306.13	-30226.13
35	MWMOTE	ENN	RSMAX2	-30555.07	-30306.13	-30226.13
35	MWMOTE	NCL	RSMAX2	-30555.07	-30306.13	-30226.13

35	MWMOTE	OSS	RSMAX2	-30555.07	-30306.13	-30226.13
35	MWMOTE	SBC	RSMAX2	-30555.07	-30306.13	-30226.13
35	MWMOTE	TL	RSMAX2	-30555.07	-30306.13	-30226.13
35	ADASYN	Original	RSMAX2	-36625.56	-36366.88	-36285.88
35	ADASYN	RUS	RSMAX2	-36713.47	-36473.96	-36398.96
35	ADASYN	ENN	RSMAX2	-36740.16	-36500.62	-36425.62
35	ADASYN	NCL	RSMAX2	-36481.05	-36244.89	-36170.89
35	ADASYN	OSS	RSMAX2	-36367.87	-36119.05	-36041.05
35	ADASYN	SBC	RSMAX2	-36603.71	-36351.35	-36272.35
35	ADASYN	TL	RSMAX2	-36548.28	-36309.02	-36234.02
35	BLSMOTE_1	Original	RSMAX2	-28255.54	-27986.67	-27898.67
35	BLSMOTE_1	RUS	RSMAX2	-28913.08	-28665.60	-28584.60
35	BLSMOTE_1	ENN	RSMAX2	-28936.87	-28726.05	-28657.05
35	BLSMOTE_1	NCL	RSMAX2	-29699.32	-29479.34	-29407.34
35	BLSMOTE_1	OSS	RSMAX2	-30575.75	-30377.15	-30312.15
35	BLSMOTE_1	SBC	RSMAX2	-29110.51	-28899.69	-28830.69
35	BLSMOTE_1	TL	RSMAX2	-28847.03	-28590.38	-28506.38
35	BLSMOTE_2	Original	RSMAX2	-39026.27	-38745.49	-38658.49
35	BLSMOTE_2	RUS	RSMAX2	-39026.27	-38745.49	-38658.49
35	BLSMOTE_2	ENN	RSMAX2	-39026.27	-38745.49	-38658.49
35	BLSMOTE_2	NCL	RSMAX2	-39026.27	-38745.49	-38658.49
35	BLSMOTE_2	OSS	RSMAX2	-39026.27	-38745.49	-38658.49
35	BLSMOTE_2	SBC	RSMAX2	-39026.27	-38745.49	-38658.49
35	BLSMOTE_2	TL	RSMAX2	-39026.27	-38745.49	-38658.49
40	Original	Original	RSMAX2	-30634.15	-30374.75	-30291.75

40	Original	RUS	RSMAX2	-30634.15	-30374.75	-30291.75
40	Original	ENN	RSMAX2	-30634.15	-30374.75	-30291.75
40	Original	NCL	RSMAX2	-30634.15	-30374.75	-30291.75
40	Original	OSS	RSMAX2	-30634.15	-30374.75	-30291.75
40	Original	SBC	RSMAX2	-30634.15	-30374.75	-30291.75
40	Original	TL	RSMAX2	-30634.15	-30374.75	-30291.75
40	ROS	Original	RSMAX2	-36064.24	-35820.97	-35743.97
40	ROS	RUS	RSMAX2	-35045.05	-34776.51	-34691.51
40	ROS	ENN	RSMAX2	-35085.36	-34810.50	-34723.50
40	ROS	NCL	RSMAX2	-35091.65	-34832.58	-34750.58
40	ROS	OSS	RSMAX2	-35581.91	-35332.32	-35253.32
40	ROS	SBC	RSMAX2	-35156.00	-34900.09	-34819.09
40	ROS	TL	RSMAX2	-35158.60	-34886.89	-34800.89
40	SMOTE	Original	RSMAX2	-37970.41	-37697.03	-37612.03
40	SMOTE	RUS	RSMAX2	-37970.41	-37697.03	-37612.03
40	SMOTE	ENN	RSMAX2	-37970.41	-37697.03	-37612.03
40	SMOTE	NCL	RSMAX2	-37970.41	-37697.03	-37612.03
40	SMOTE	OSS	RSMAX2	-37970.41	-37697.03	-37612.03
40	SMOTE	SBC	RSMAX2	-37970.41	-37697.03	-37612.03
40	SMOTE	TL	RSMAX2	-37970.41	-37697.03	-37612.03
40	MWMOTE	Original	RSMAX2	-34469.88	-34194.92	-34107.92
40	MWMOTE	RUS	RSMAX2	-34469.88	-34194.92	-34107.92
40	MWMOTE	ENN	RSMAX2	-34469.88	-34194.92	-34107.92
40	MWMOTE	NCL	RSMAX2	-34469.88	-34194.92	-34107.92
40	MWMOTE	OSS	RSMAX2	-34469.88	-34194.92	-34107.92

40	MWMOTE	SBC	RSMAX2	-34469.88	-34194.92	-34107.92
40	MWMOTE	TL	RSMAX2	-34469.88	-34194.92	-34107.92
40	ADASYN	Original	RSMAX2	-38664.35	-38367.53	-38275.53
40	ADASYN	RUS	RSMAX2	-38793.50	-38528.84	-38446.84
40	ADASYN	ENN	RSMAX2	-37034.74	-36758.63	-36671.63
40	ADASYN	NCL	RSMAX2	-38586.95	-38309.40	-38223.40
40	ADASYN	OSS	RSMAX2	-38763.40	-38492.30	-38408.30
40	ADASYN	SBC	RSMAX2	-38590.65	-38306.70	-38218.70
40	ADASYN	TL	RSMAX2	-38766.12	-38495.04	-38411.04
40	BLSMOTE_1	Original	RSMAX2	-35977.47	-35677.12	-35582.12
40	BLSMOTE_1	RUS	RSMAX2	-35515.08	-35195.76	-35094.76
40	BLSMOTE_1	ENN	RSMAX2	-37092.75	-36824.02	-36739.02
40	BLSMOTE_1	NCL	RSMAX2	-36945.54	-36711.59	-36637.59
40	BLSMOTE_1	OSS	RSMAX2	-36890.23	-36653.11	-36578.11
40	BLSMOTE_1	SBC	RSMAX2	-36594.04	-36350.60	-36273.60
40	BLSMOTE_1	TL	RSMAX2	-36909.36	-36643.79	-36559.79
40	BLSMOTE_2	Original	RSMAX2	-42537.67	-42246.55	-42157.55
40	BLSMOTE_2	RUS	RSMAX2	-42537.67	-42246.55	-42157.55
40	BLSMOTE_2	ENN	RSMAX2	-42537.67	-42246.55	-42157.55
40	BLSMOTE_2	NCL	RSMAX2	-42537.67	-42246.55	-42157.55
40	BLSMOTE_2	OSS	RSMAX2	-42537.67	-42246.55	-42157.55
40	BLSMOTE_2	SBC	RSMAX2	-42537.67	-42246.55	-42157.55
40	BLSMOTE_2	TL	RSMAX2	-42537.67	-42246.55	-42157.55
45	Original	Original	RSMAX2	-30634.15	-30374.75	-30291.75
45	Original	RUS	RSMAX2	-30634.15	-30374.75	-30291.75

45	Original	ENN	RSMAX2	-30634.15	-30374.75	-30291.75
45	Original	NCL	RSMAX2	-30634.15	-30374.75	-30291.75
45	Original	OSS	RSMAX2	-30634.15	-30374.75	-30291.75
45	Original	SBC	RSMAX2	-30634.15	-30374.75	-30291.75
45	Original	TL	RSMAX2	-30634.15	-30374.75	-30291.75
45	ROS	Original	RSMAX2	-43616.35	-43339.10	-43254.10
45	ROS	RUS	RSMAX2	-44049.75	-43782.29	-43700.29
45	ROS	ENN	RSMAX2	-44167.43	-43913.01	-43835.01
45	ROS	NCL	RSMAX2	-43462.22	-43178.45	-43091.45
45	ROS	OSS	RSMAX2	-44106.44	-43838.98	-43756.98
45	ROS	SBC	RSMAX2	-43509.55	-43209.47	-43117.47
45	ROS	TL	RSMAX2	-43330.76	-43004.58	-42904.58
45	SMOTE	Original	RSMAX2	-42372.13	-42056.01	-41959.01
45	SMOTE	RUS	RSMAX2	-42372.13	-42056.01	-41959.01
45	SMOTE	ENN	RSMAX2	-42372.13	-42056.01	-41959.01
45	SMOTE	NCL	RSMAX2	-42372.13	-42056.01	-41959.01
45	SMOTE	OSS	RSMAX2	-42372.13	-42056.01	-41959.01
45	SMOTE	SBC	RSMAX2	-42372.13	-42056.01	-41959.01
45	SMOTE	TL	RSMAX2	-42372.13	-42056.01	-41959.01
45	MWMOTE	Original	RSMAX2	-38220.70	-37912.08	-37816.08
45	MWMOTE	RUS	RSMAX2	-38220.70	-37912.08	-37816.08
45	MWMOTE	ENN	RSMAX2	-38220.70	-37912.08	-37816.08
45	MWMOTE	NCL	RSMAX2	-38220.70	-37912.08	-37816.08
45	MWMOTE	OSS	RSMAX2	-38220.70	-37912.08	-37816.08
45	MWMOTE	SBC	RSMAX2	-38220.70	-37912.08	-37816.08

45	MWMOTE	TL	RSMAX2	-38220.70	-37912.08	-37816.08
45	ADASYN	Original	RSMAX2	-41809.33	-41523.06	-41435.06
45	ADASYN	RUS	RSMAX2	-41873.23	-41586.88	-41498.88
45	ADASYN	ENN	RSMAX2	-41289.82	-41020.65	-40937.65
45	ADASYN	NCL	RSMAX2	-41189.50	-40897.21	-40807.21
45	ADASYN	OSS	RSMAX2	-41775.91	-41502.95	-41418.95
45	ADASYN	SBC	RSMAX2	-41709.56	-41423.36	-41335.36
45	ADASYN	TL	RSMAX2	-41306.10	-41075.50	-41003.50
45	BLSMOTE_1	Original	RSMAX2	-45063.05	-44772.87	-44683.87
45	BLSMOTE_1	RUS	RSMAX2	-45242.99	-45001.72	-44927.72
45	BLSMOTE_1	ENN	RSMAX2	-46456.44	-46202.12	-46124.12
45	BLSMOTE_1	NCL	RSMAX2	-44878.94	-44621.36	-44542.36
45	BLSMOTE_1	OSS	RSMAX2	-44698.43	-44421.29	-44336.29
45	BLSMOTE_1	SBC	RSMAX2	-43730.11	-43407.33	-43308.33
45	BLSMOTE_1	TL	RSMAX2	-44250.65	-43944.17	-43850.17
45	BLSMOTE_2	Original	RSMAX2	-46288.38	-45960.44	-45861.44
45	BLSMOTE_2	RUS	RSMAX2	-46288.38	-45960.44	-45861.44
45	BLSMOTE_2	ENN	RSMAX2	-46288.38	-45960.44	-45861.44
45	BLSMOTE_2	NCL	RSMAX2	-46288.38	-45960.44	-45861.44
45	BLSMOTE_2	OSS	RSMAX2	-46288.38	-45960.44	-45861.44
45	BLSMOTE_2	SBC	RSMAX2	-46288.38	-45960.44	-45861.44
45	BLSMOTE_2	TL	RSMAX2	-46288.38	-45960.44	-45861.44
50	Original	Original	RSMAX2	-30634.15	-30374.75	-30291.75
50	Original	RUS	RSMAX2	-30634.15	-30374.75	-30291.75
50	Original	ENN	RSMAX2	-30634.15	-30374.75	-30291.75

50	Original	NCL	RSMAX2	-30634.15	-30374.75	-30291.75
50	Original	OSS	RSMAX2	-30634.15	-30374.75	-30291.75
50	Original	SBC	RSMAX2	-30634.15	-30374.75	-30291.75
50	Original	TL	RSMAX2	-30634.15	-30374.75	-30291.75
50	ROS	Original	RSMAX2	-52783.50	-52480.91	-52390.91
50	ROS	RUS	RSMAX2	-52783.50	-52480.91	-52390.91
50	ROS	ENN	RSMAX2	-52783.50	-52480.91	-52390.91
50	ROS	NCL	RSMAX2	-52783.50	-52480.91	-52390.91
50	ROS	OSS	RSMAX2	-52783.50	-52480.91	-52390.91
50	ROS	SBC	RSMAX2	-52783.50	-52480.91	-52390.91
50	ROS	TL	RSMAX2	-52783.50	-52480.91	-52390.91
50	SMOTE	Original	RSMAX2	-45874.17	-45596.24	-45512.24
50	SMOTE	RUS	RSMAX2	-45874.17	-45596.24	-45512.24
50	SMOTE	ENN	RSMAX2	-45874.17	-45596.24	-45512.24
50	SMOTE	NCL	RSMAX2	-45874.17	-45596.24	-45512.24
50	SMOTE	OSS	RSMAX2	-45874.17	-45596.24	-45512.24
50	SMOTE	SBC	RSMAX2	-45874.17	-45596.24	-45512.24
50	SMOTE	TL	RSMAX2	-45874.17	-45596.24	-45512.24
50	MWMOTE	Original	RSMAX2	-43098.30	-42815.42	-42728.42
50	MWMOTE	RUS	RSMAX2	-43098.30	-42815.42	-42728.42
50	MWMOTE	ENN	RSMAX2	-43098.30	-42815.42	-42728.42
50	MWMOTE	NCL	RSMAX2	-43098.30	-42815.42	-42728.42
50	MWMOTE	OSS	RSMAX2	-43098.30	-42815.42	-42728.42
50	MWMOTE	SBC	RSMAX2	-43098.30	-42815.42	-42728.42
50	MWMOTE	TL	RSMAX2	-43098.30	-42815.42	-42728.42

50	ADASYN	Original	RSMAX2	-44346.44	-44113.05	-44041.05
50	ADASYN	RUS	RSMAX2	-47972.56	-47752.23	-47685.23
50	ADASYN	ENN	RSMAX2	-47999.26	-47778.91	-47711.91
50	ADASYN	NCL	RSMAX2	-48155.13	-47941.34	-47876.34
50	ADASYN	OSS	RSMAX2	-48155.13	-47941.34	-47876.34
50	ADASYN	SBC	RSMAX2	-46408.40	-46171.61	-46099.61
50	ADASYN	TL	RSMAX2	-45975.67	-45739.31	-45667.31
50	BLSMOTE_1	Original	RSMAX2	-55122.79	-54894.33	-54826.33
50	BLSMOTE_1	RUS	RSMAX2	-53779.30	-53550.83	-53482.83
50	BLSMOTE_1	ENN	RSMAX2	-56009.48	-55767.57	-55695.57
50	BLSMOTE_1	NCL	RSMAX2	-55175.46	-54940.28	-54870.28
50	BLSMOTE_1	OSS	RSMAX2	-56456.52	-56248.21	-56186.21
50	BLSMOTE_1	SBC	RSMAX2	-56139.00	-55940.77	-55881.77
50	BLSMOTE_1	TL	RSMAX2	-53609.06	-53340.28	-53260.28
50	BLSMOTE_2	Original	RSMAX2	-51323.98	-51008.01	-50914.01
50	BLSMOTE_2	RUS	RSMAX2	-51323.98	-51008.01	-50914.01
50	BLSMOTE_2	ENN	RSMAX2	-51323.98	-51008.01	-50914.01
50	BLSMOTE_2	NCL	RSMAX2	-51323.98	-51008.01	-50914.01
50	BLSMOTE_2	OSS	RSMAX2	-51323.98	-51008.01	-50914.01
50	BLSMOTE_2	SBC	RSMAX2	-51323.98	-51008.01	-50914.01
50	BLSMOTE_2	TL	RSMAX2	-51323.98	-51008.01	-50914.01

Table B.1: Bayesian network score results by simulation of re-sampling data