VRIJE UNIVERSITEIT AMSTERDAM

MASTER PROJECT

BUSINESS ANALYTICS

Customer Churn Analysis and Prediction Modelling at

Icecat, a Content Syndicator

Yi-Ting Lin

September 2020

Supervisor:
Martijn Hoogeveen

Supervisor:
dr. Dennis Dobler

Second Reader:
dr. Jakub Tomczak

Acknowledgments

This report is a master graduation project with a six months internship at Icecat for the Master Business Analytics at the Vrije Universiteit Amsterdam. I appreciate the opportunity to work on a real business project with Icecat. During my internship, I have learned how to communicate with people from different backgrounds and gained a lot of hands-on experience. Besides the work experience, I have learned to look at things from a different angle and to approach tasks with a flexible mindset.

This project works under the supervision of Dennis Dobler on behalf of the Vrije Universiteit and Martijn Hoogeveen on behalf of Icecat. I would like to thank both of them for being my supervisors and for their guidance and support throughout the project. Moreover, I would like to thank Vazha Abramishili and Emre Tan at Icecat for their involvement during my internship, and Jakub Tomczak, for being my second reader. Finally, my special thanks go to my family for their support, encouragement, and love.

Yi-Ting Lin

September 2020

Content

List of Abbreviations

B2B	Business to Business
B2C	Business to Customer
EDA	Exploratory Data Analysis
DT	Decision Tree
LR	Logistic Regression
RF	Random Forest
KNNs	K-Nearest Neighbors
SVM	Support Vector Machine
T-Links	Tomek Links
SMOTE	Synthetic Minority Oversampling Technique
CV	Cross-Validation
SK	Stratified k-fold
RSK	Repeated Stratified k-fold
SSS	Stratified Shuffle & Split
BENELUX	Belgium, the Netherlands, and Luxembourg
SPPO	Spain, Portugal
DACH	Germany, Austria, Switzerland
NORDICS	Denmark, Finland, Iceland, Norway, and Sweden
ROEU	Rest of Europe
ROW	Rest of the World

1 Introduction

Business Understanding

Customer acquisition and retention are both necessary for growing and keeping business. Long-term business success can be achieved by reducing costs or increasing revenue. Compared to winning a new customer, the cost of keeping a current one is lower [1] and company profits can be increased to 25% - 95% by just increasing customer retention rates by 5% [2]. On the other hand, customer churn, existing customers stopping with buying products or services from a company, will decrease the revenues and profits of a company as significantly. Churn can be a big problem for any business independent of the field they operate in. Business sales can be divided into two types: Business-to-Consumer (B2C) and Business-to-Business (B2B). B2C is a company doing business with residential customers. On the other hand, if companies have a purchase relationship with other companies, it is called B2B. Compared to B2C, the churn of a single client has a significant impact on the business-to-business (B2B) model [3], as the number of customers is usually smaller, but transaction values are higher [4]. Hence, for a B2B company, having a churn analysis and building a predictive churn model helps to get critical insight into customer behavior and to improve its strategy for long term business survival.

Icecat

Icecat, a B2B company, is an online publisher and syndicator of product information and product reviews for the global e-commerce market. There are two types of clients: Manufacturers or Brand owners, and Online channel partners, including resellers, distributors, e-commerce platforms, and comparison websites. Brands/Manufacturers pay Icecat for providing their product data as free product content to their e-commerce partners. Channel partners use this free product data or pay a subscription fee to Icecat for high-quality product content. Based on this contractual subscription business model, Icecat is sensitive to the customer churn problem. Churn analysis can help Icecat to identify the cause of the churn and to make choices that might improve customer retention.

The revenue contribution ratio between Brands/Manufacturers and Channel partners is about 70% vs. 30%. Icecat is interested in the question: What if we change our business strategy by increasing the number of Brand/Manufacturer customers? Will it increase or decrease the churn rate and revenue of our future business?

Goals of the Project

Two main parts of the project: (1) A churn analysis. especially focused on the type of customer, to adjust the revenue business strategy. (2) A churn prediction model to identify a new customer or a current customer who is at a high risk of churning next year. The following questions can be answered at the end of the report.

(1) A churn analysis
    - Which type of client has a higher churn rate?
    - What are the characteristics of a churner?

(2) A churn prediction model
    - Which machine learning algorithms can generate the best performance result?
    - Which resampling technique can be best used in B2B situations?

The following parts of the paper are organized as follows: Section 2 and 3 describe the relevant literature regarding customer churn, applications in the B2B model, the particular imbalanced data problem, and the common models and algorithms used for churn prediction. Section 4 shows the analyzed data and presents data pre-processing methods. Besides, it also answers the research questions about the cause of churn and the characteristics of churn customers. After this stage, data is prepared and ready for prediction. Section 5 presents the experimental results, and the model's performance and evaluation. Finally, Section 6 gives the conclusions and some suggestions for further research.

2 Literature review

Churn Problem

Customer churn means that a company loses customers, which results in a loss in revenue. Nowadays, the churn problem has become more critical than in the past due to the booming of the eCommerce market [5]. With the growth of online shopping, it provides customers an easier way to access information, to compare products and services. For customers, this kind of change makes them switch between product or service providers easier and less costly. As a result, the churn rate might be increased. Hence, companies have to deal with the growing competition with their competitors and a higher churn rate problem. Therefore, to have better churn management is important for succeeding in business in the long term.

Churn in Subscription B2B Companies

Since Icecat's business model is based on subscription, we are interested in subscription churn [6][7]. However, most of the research about customer churn in contractual business is in B2C (Table 2.1) rather than in B2B [13]. Further, those contractual churn studies are mainly about the telecommunication industry. As described in the introduction, B2B companies are typically having fewer customers but higher transaction values per sale. Besides, according to the Pareto principle (80/20 rule), 20% of customers may even generate 80% of the total revenue [14]. It implicates that with limited company resources, a company should keep the most profitable and valuable customers. Additionally, losing a high-end customer will even have a more serious impact on a B2B company's revenue [15] than a low-end one. Therefore, a predictive churn model can have positive implications for customer retention in B2B.

The Problem with Imbalanced Classes

A good churn prediction model should not only have high accuracy but also a good performance outcome. However, if there is the problem of class imbalance, it might result in misleading accuracy. For instance, in this churn prediction case, there are two classes: a customer will churn, '*yes*', and not churn, '*no*', where the ratio is about 15%/85%. The model is to predict churn '*yes*', but usually they are in the minority as they represent only 15% of total data points. If we fit the model to such skewed data, usually the accuracy score in the training set is higher than in the test set. This difference is an overfitting problem and it can impact the model performance. Resampling is the technique to fix the overfitting by randomly resampling the training set by either to removing data points from the majority called undersampling or to duplicate data points from the minority called oversampling. However, random undersampling may lead to a poor prediction in the test set, if deleted samples are carrying useful. In the meanwhile, random oversampling just duplicates samples from the minority in the training set, and it can lead to overfitting problem as well. The mentioned problems can be improved by two advanced resampling methods: Tomek links (T-Links) [16] is used to undersample the majority and the synthetic minority oversampling technique (SMOTE) [17][18][19]

is used to oversample the minority.

Stratified sampling is another way to handle imbalanced classes. Unlike resampling, there is no need to resample classes before cross-validation [20]. Instead, it preserves the imbalanced class distribution for each fold while doing the cross-validation. Details of resampling and stratified cross-validation will be explained in Section 3.2 and 3.3.

Churn Models and Model Evaluation

In this customer churn prediction model, we would like to predict "Will this customer churn or not churn?" The target variable, 'Churn', is labeled either '*yes*' or '*no*', thus this is a binary classification problem. Due to the simplicity and interpretability, the two most commonly used methods for binary classification are logistic regression (LR) and classification trees [21] [22]. We apply Logistic Regression (LR), Decision Tree (DT), Random Forest (RF) [23], K-Nearest Neighbors (KNNs) [24], and Support Vector Machine (SVM) [25] for churn prediction. Regarding model evaluation, the confusion matrix, and the Receiver Operating Characteristic (ROC) Curve and the Area Under Curve (AUC) score are used as performance measurements for the classification problem. Beside, Accuracy, Precision, Recall, and F1 score are the four metrics that can be computed from the confusion matrix and to quantify the model performance. All of the above will be explained in detail in Section 3.

.

Paper	Industry	Prediction model
C. Bolancé. et al. (2016) [8]	Insurance	Logistic Regression, Conditional Tree, Neural Network, Support Vector Machine
Sahar F. Sabbeh. et al. (2018)[9]	telecommunication	Decision Trees, K-Nearest Neighbors, Support Vector Machines, Logistic Regression, Random Forest, Boosting trees, and Stochastic Gradient Boosting Bayesian, and Multi-layer perceptron
Keramati, A. et al. (2014)[12]	telecommunication	Decision Tree, Artificial Neural Networks, K-Nearest Neighbors, and Support Vector Machine
I. Ullah, et al. (2019)[10]	telecommunication	Random Forest
Ahmad, A.K. et al. (2019)[11]	telecommunication	Decision Tree, Random Forest, Gradient Boosted Machine Tree " GBM" and Extreme Gradient Boosting

Table 2.1: Papers about contractual churn in B2C

3 Methodology

3.1 Predictive Classification Methods

Our churn prediction model is a binary classification problem, it classifies and predicts the current or new customers as “churn” or “not churn”. The target, “Churn”, is labeled “yes” or “no”, and algorithms learn from training data to predict the target from the input data. Since the target is labeled, it is called supervised learning. We present the most common learning methods, logistic regression, and classification trees, and its models used for the churn prediction are in Section 3.11 to 3.15.

3.1.1 Logistic Regression

Logistic regression (LR) predicts the probability between 0 and 1 for the target variable (y) and uses the threshold value to classify it to either 1 or 0 in Figure 3.1. The S-curve is the Sigmoid function (1) used to fit values.

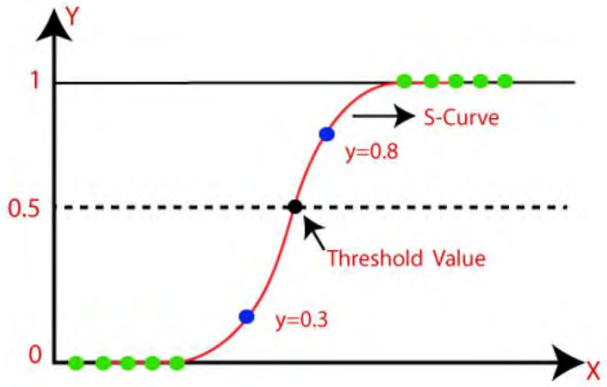

Figure 3.1: Logistic Regression with S-curve shape and the threshold value of 0.5. (Source: JavaTpoint, javatpoint.com).

$$F(x) = \frac{1}{1+e^{-x}} = \frac{e^{x}}{e^{x}+1} \quad (1)$$

3.1.2 Decision Tree

A decision tree (DT) consists of a Root node, Branch nodes, and Leaf node (class label). The Root node and Branch node present a feature attribute, and the Leaf node presents the target output. The process from the root to the leaf node is based on conditions. For example, in Figure 3.2, a new Brand\Manufacturer customer who pays more than 5, 000 € is classified as a non-churn customer.

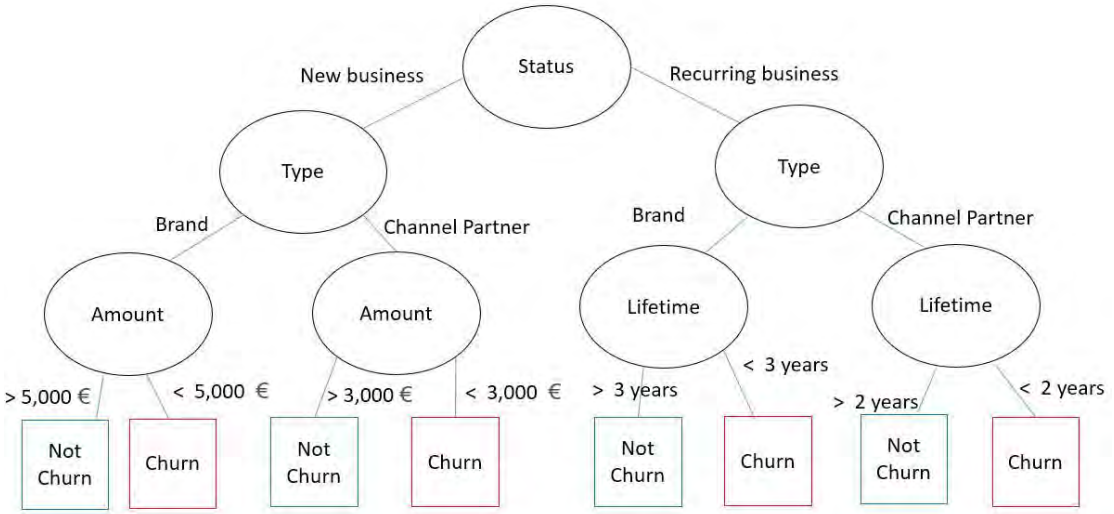

Figure 3.2: Example of a Decision Tree

3.1.3 Random Forest

Compared to the DT, a single tree building on an entire dataset, the Random Forest (RF) is a collection of a large number of DTs (Figure 3.3). Firstly, it randomly selects observations and variables from the original dataset. Secondly, each node of the tree is randomly selected from the sub-dataset selected from the first step and runs the process of building a tree until no more leaf can be made. Once a tree has been built, it repeats the whole process to generate a whole bunch of trees. The key to classification is based on "voting", results of all individual trees are collected and calculated, the final winner (class) is the one receiving the most votes.

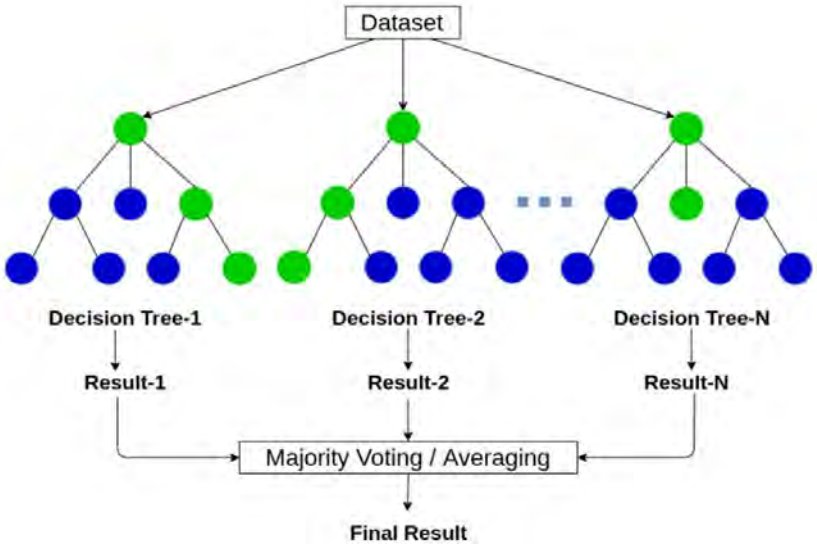

Figure 3.3: Example of a Random Forest (https://www.analyticsvidhya.com/)

3.1.4 K-Nearest Neighbors

K-Nearest Neighbors (KNNs) finds the nearest neighbors to a new observation. K is the number of nearest neighbors. In Figure 3.4 you can see how KNNs works. For instance, a new observation is needed to be classified. The first step is to calculate the distance between the new point to other points and then to find the closest neighbors. The final step is to classify the new point by "votes" from its K neighbors. In this case, for K = 3, the result of the voting of this data point is class B.

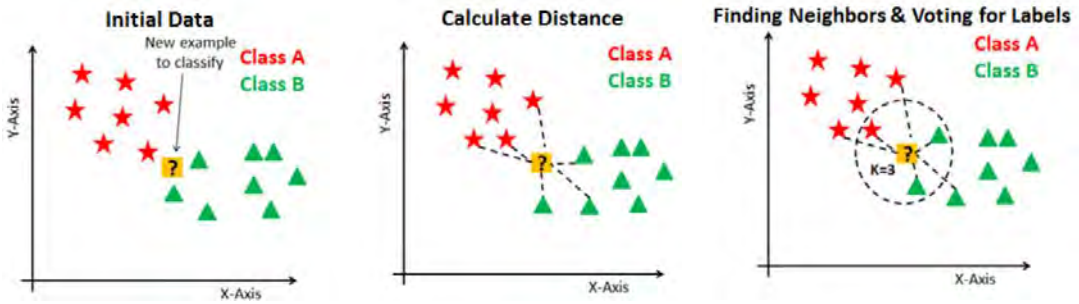

Figure 3.4: KNNs process (www.datacamp.com)

3.1.5 Support Vector Machine

Support Vector Machine (SVM) is to separate different classes by the hyperplane as best as possible. It first generates hyperplanes and finds an optimal one with the maximum margin and the minimum error.

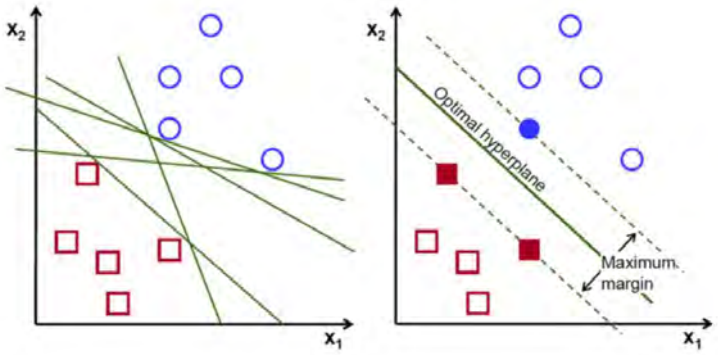

Figure 3.5: SVM process (https://towardsdatascience.com)

3.2 Resampling Technique

As mentioned before, the class imbalance is the common problem in churn prediction and usually, it will result in overfitting due to high accuracy prediction in the majority but low prediction in minority. However, for most of the cases, we want to have high accuracy prediction in minority rather than in the majority. One of the techniques to deal with such a problem is resampling [26]. There are two ways of resampling, undersampling and upsampling. Undersampling removes random samples from the majority class. On the contrary, oversampling duplicates random samples from the minority class.

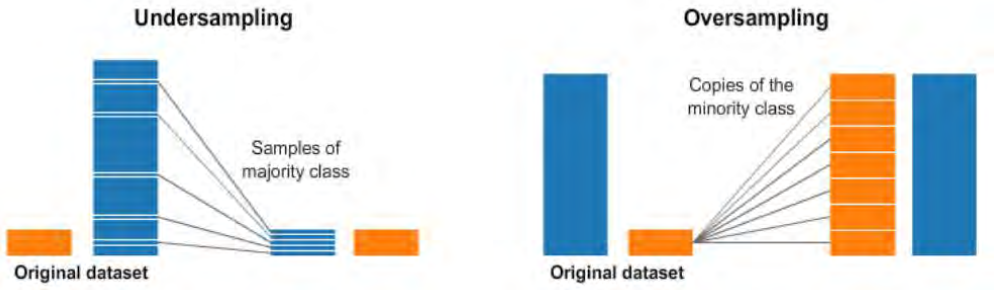

Figure 3.6: Visualization of undersampling and oversampling (towardsdatascience.com)

Although resampling can help to balance class, it does have drawbacks [27]. Random undersampling might neglect some valuable information from the majority class and the model is not train well and leads to bad model performance. On the other hand, random oversampling is just adding the duplicates to the training set and it might result in overfitting [28]. Here, we use advanced resampling methods instead of just randomly removing or adding samples.

Advanced Undersampling: Tomek links

Tomek Links (T-Links) find the two closest samples of each class as shown in a green circle, and then remove the sample from the majority class, the blue one will be removed). This can help to improve model performance by removing overlapping samples.

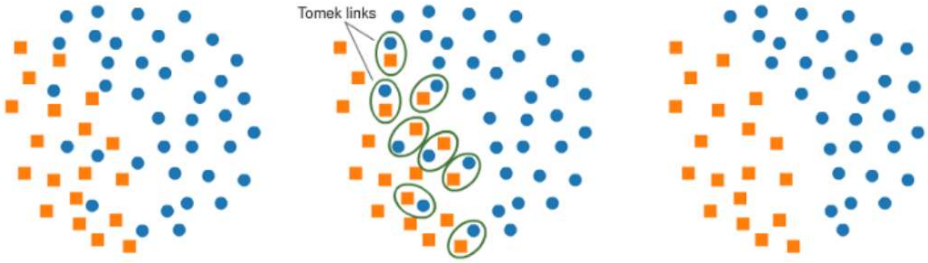

Figure 3.7: T-Links process (https://www.kaggle.com/)

Advanced Oversampling: Synthetic Minority Oversampling Technique

Synthetic Minority Oversampling Technique (SMOTE) first randomly selects a sample from the minority class and then finds its k-nearest neighbors from the same class. Next, is to add synthetic (create a similar new sample) between the selected sample and its nearest neighbors. Repeat the above process until the total number of samples in the minority is the same as in the majority (Figure 3.8). Unlike random oversampling by just randomly duplicate samples and add to the training set, SMOTE adds some variety to the data.

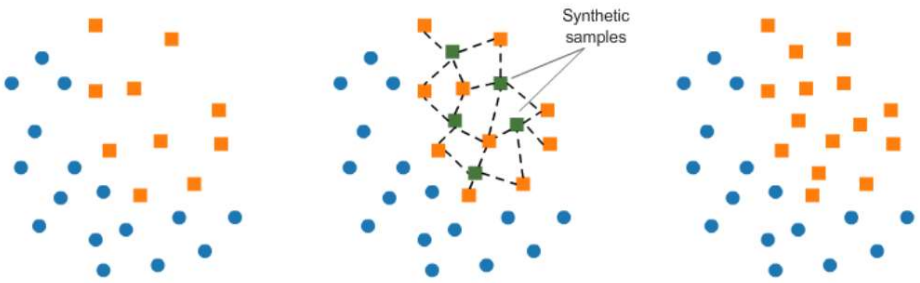

Figure 3.8: SMOTE process (https://www.kaggle.com/)

3.3 Cross-validation

Cross-validation is used to evaluate models by splitting data into training and validation set instead of using the entire dataset while training a classifier. After training, the validation set is used to test the model performance. Cross-validation iterates the above process many times.

- **K-Fold Cross-Validation**

The K-Fold cross-validation works as the following steps:

1 KF randomly divides the entire dataset into k folds equally.
   - One of the folds is used as a test set and the rest of the folds are used as a training set.
2 KF selects the first fold as the test set.
   - The test set is selected one by one in order. For instance, in the second iteration, the second fold will be selected as the test.
3 Repeat steps 1 and 2 for *k* times.

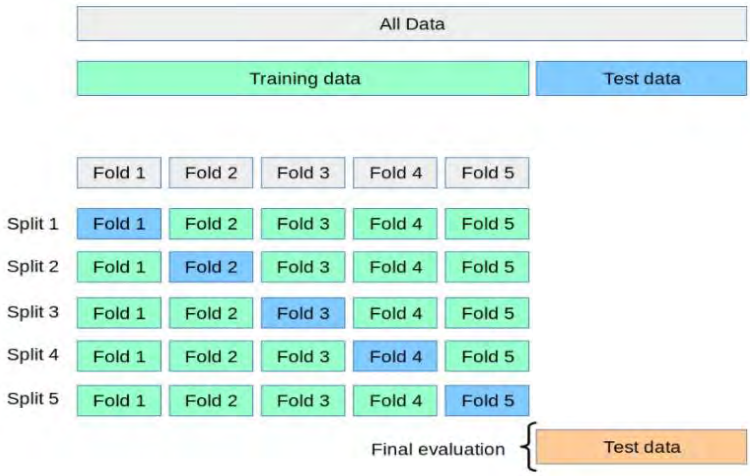

Figure 3.9: Visualization of a k-fold validation when k=5 (www.scikit-learn.org)

- **Stratified k-fold Cross-Validation**

The difference between k-fold and Stratification k-fold cross-validation (SK) is the later one can preserve the imbalanced class distribution for each fold.

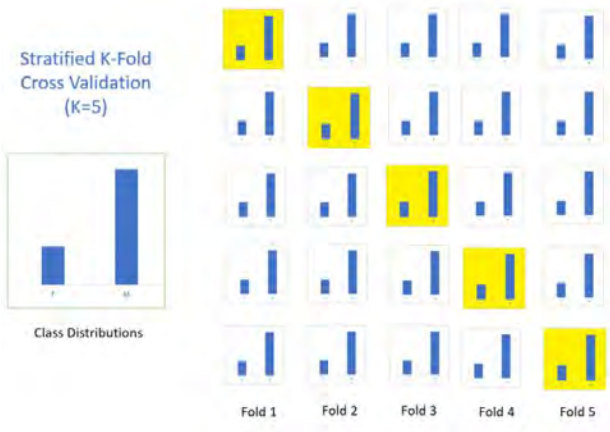

Figure 3.10: Visualization of a Stratified k-fold validation when k = 5 (https://www.analyticsvidhya.com/)

- **Repeated Stratified k-fold Cross-Validation**

Repeated Stratified k-fold cross-validation (RSK) is similar to Stratified k-fold but RSK repeats n times of the cross-validation process. For instance, for a 5 fold cross-validation, if the number of repeats is 3 times, then there are 5 *3 = 15 models to be fitted and evaluated.

- **Stratified Shuffle & Split**

Stratified Shuffle &Split (SSS) randomly selects samples from the entire dataset during each iteration and then returns stratified splits. SSS preserves the imbalanced class distribution like SK and RSK.

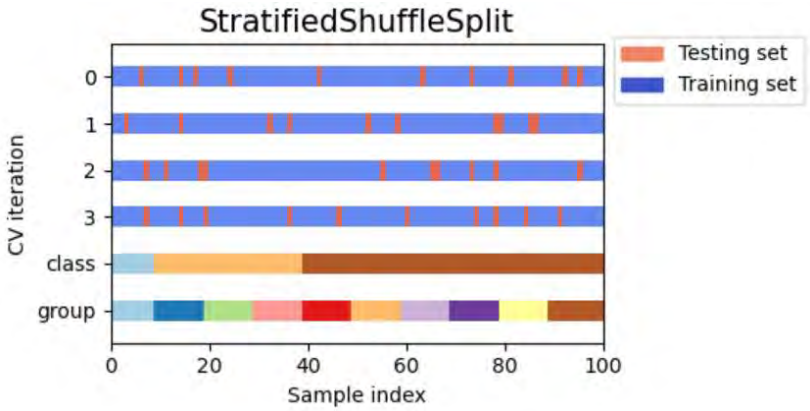

Figure 3.11 : Visualization of a StratifiedShuffleSplit validation (www.scikit-learn.org)

SK, RSK, and SSS are the stratified methods that can be used for dealing with imbalanced classes.

3.4 Model Evaluation

For classification problems, the confusion matrix, Receiver Operating Characteristic (ROC) Curves, and Area Under the Curve (AUC) are the most common methods to evaluate model performance. Accuracy, Precision, Recall, and F1 score are the four scores that can be calculated from the confusion matrix.

Confusion Matrix

The Confusion matrix shows the correct and incorrect predictions for each class in Table 3.1.

- **True Positive (TP)**: A customer is actually a churned customer and the model predicted the customer as churn.
- **True Negative (TN)**: A customer is actually not a churned customer and the model predicted the customer as a non-churn customer.
- **False Positives (FP)**: A customer is actually a non-churn customer, but the model predicted the customer as a churned customer.
- **False Negative (FN)**: A customer is actually a churned customer, but the model predicted the customer as a non-churn customer.

	Predicted Class		
Actual Class		Positive (1) Churn	Negative (0) Non-Churn
	Positive (1) Churn	True Positive (TP)	False Negative (FN)
	Negative (0) Non-Churn	False Positive (FP)	True Negative (TN)

Table 3.1: Confusion matrix

Based on the confusion matrix, the following measurements can be calculated and used for evaluating model performance.

- **Accuracy:** It gives a basic idea of the model by performance how many percentages of observations are classified correctly. For example, 80% of accuracy means that 80% of the case, the model can classify the new observation correctly.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

- **Precision:** It shows that the percentage of the model identifies churn customers correctly among predicted as churn customers.

$$Precision = \frac{TP}{(TP + FP)}$$

- **Recall:** It measures the percentage of the model identifies churn customers correctly among actually churn customers.

$$Recall = \frac{TP}{(TP + FN)}$$

- **F1 Score:** It calculates the weighted average of precision and recall. If both precision and recall are equally important, then the F1 score is a good measurement to select the model.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

In our churn prediction model, Recall is more important than Precision. Because even a non-churn customer is classified as a churned customer, it would not affect the revenue. However, if the model can not identify a true churn customer and predict he/she as a non-churn customer, then we will lose this customer.

Receiver Operating Characteristic Curve and Area Under the Curve

A Receiver Operating Characteristic Curve (ROC) that plots the False Positive rate against the True Positive rate (Figure 3.12). Area Under the Curve (AUC) is the percentage of the ROC area, which is under the ROC curve. Unlike Accuracy, Precision, Recall, and F1 score, they are relative to each other, but AUC is a single value used to summarize the model performance. The higher an AUC score, the better the model is in separate classes. If the benchmark model is just a random guess, its ROC plot will be a diagonal line as the black dash line in Figure 3.12. The AUC score of the benchmark model is 0.5. Hence, if a model's AUC is less than 0.5, this model is not good and it is unable to separate classes.

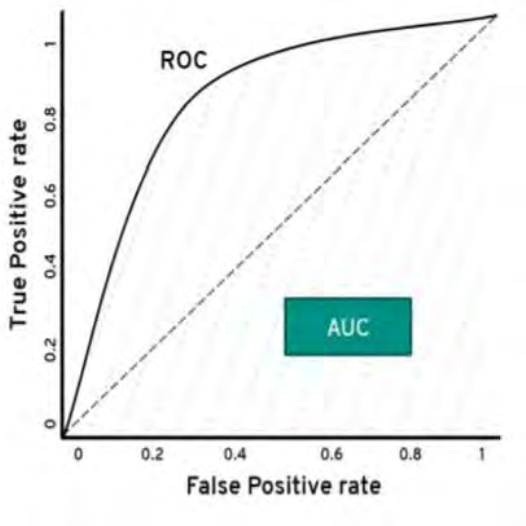

Figure 3.12: Visualization of ROC and AUC

4.3 Data Preprocessing

This section is about feature scaling, feature selection, and apply T-Links and SMOTE to the dataset. At the end of the section, the data is ready for running models.

4.3.1 Feature Scaling

To run the machine learning algorithms, we need to convert categorical feature variables to numbers and also to scale continuous ones.

Categorical encoding

Three common categorical encoding techniques:

- **Ordinal Encoding**
  It is used for categories with an ordered relationship, e.g. low churn risk and high churn risk.

- **One-Hot Encoding**
  It is used for categories without an ordered relationship, e.g. JP, NL
  This encoding assigns 1 and 0 to each category. If there have k categories, k variables will be created. This will increase the dimensionality.

- **Dummy Variable Encoding**
  It is similar to One-Hot Encoding, but k-1 variables will be created.

The categorical variables are: "Country", "Churn_risk", "Region", "Type" and "Status2018". "Churn_risk" is the only ordinal variable (high churn risk is superior to low churn risk). We use ordinal encoding for "Churn_risk" and use dummy variable encoding for the rest of the categorical variables.

Numerical Feature Scaling

We use the Log 10 transformation to transform the continuous variable "Amount" into a bell-shaped distribution.

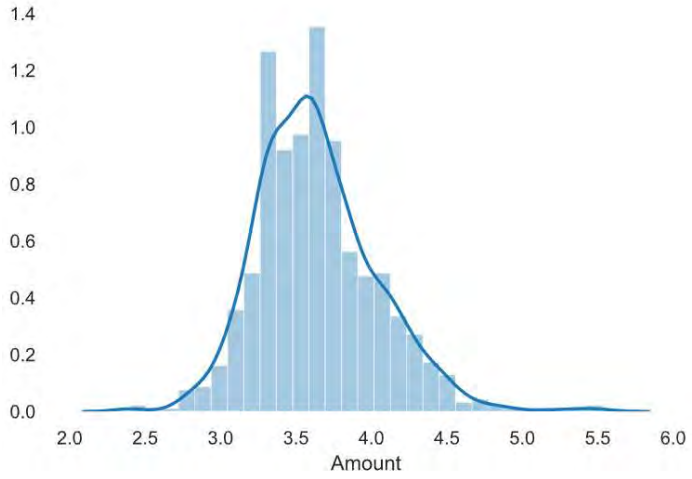

Figure 4.18: Log transformation for Amount

4.3.2 Feature Selection

After encoding categorical variables, there are now 65 features in our dataset. To use all the features may lead to overfitting and a high computation cost for training the model. Feature selection can generalize the model and reduce the computation cost. Feature selection means that we select the most relevant features and assign a score to each feature based on a threshold. If the threshold value is passed a feature is selected.

A common method used to apply on feature selection is using Random Forest. After running several experiments, the optimal number of the features is 25 with a threshold value of 0.05. The accuracy for all 65 features is about 0.8232 and the accuracy for 25 features is about 0.8279. By using feature selection, we reduce the dimensionality and enhance generalization of the model, while maintaining a similar accuracy result. Figure 4.19 displays the top 10 feature importances with a threshold of 0.05.

25 selected features

Amount	*Country_CH*	*Country_NL*	*Region_DACH*	*Region_SPPO*
Lifetime	*Country_DE*	*Country_NO*	*Region_Italy*	*Region_UK*
Country_AU	*Country_GB*	*Country_SI*	*Region_Nordics*	*Type_Channel partners*
Country_BE	*Country_IT*	*Country_US*	*Region_ROEU*	*Region_BENELUX*
Country_CA	*Country_MX*	*Churn_risk*	*Region_ROW*	*Status2018_recurring business*

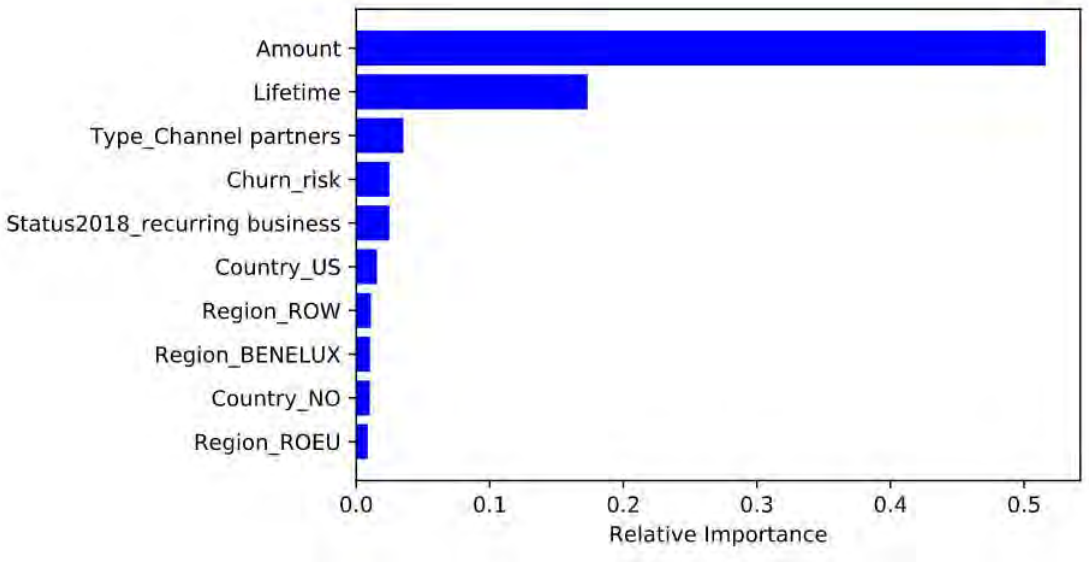

Figure 4.19: Top 10 feature importance with threshold 0.005

5 Experimental Results and Performance Evaluation

This section presents the experimental results and performance evaluation for each model. The experimental settings are :

- Resampling: Original, T-Links, SMOTE
- Cross-validation (CV):
  - K-fold (K)
  - Stratified k-fold (SK)
  - Repeated Stratified k-fold (RSK)
  - Stratified Shuffle &Split (SSS)
- Model results with and without tuning parameters

5.1 Logistic Regression

First of all, we check the accuracy of the training set and of the test set to see if there is an overfitting issue. Except for the last model, k-fold with SMOTE, there is no overfitting in other experiments. We tuned the model with the *solver* parameter and different algorithms. The best-tuned value of the *solver* is "liblinear" which has almost the same model performance as the default one. Based on the evaluation results in Table 5.2, the best model performance for LR is the one using SK with the optimal tuned parameter values of '*solver = liblinear*'.

Parameters	Description	Tuned Range	Default
solver	Algorithm to use in the optimization problem.	'newton-cg', 'lbfgs', **'liblinear'**, 'sag'	'lbfgs'

Table 5.1: Tuned parameters for LR

Resampling technique	CV	Original data points {0 : 1}	y_train {0 : 1}	LR without tuning parameters						LR with tuning parameters					
				Accuracy training set	Accuracy test set	Precision	Recall	F1	AUC	Accuracy training set	Accuracy test set	Precision	Recall	F1	AUC
Original	SK	725 : 132	580 : 106	0.8574	0.8495	0.8229	0.8495	0.7782	0.67 ± 0.04	0.8574	0.8506	0.8282	0.8506	0.7782	0.67± 0.04
	RSK	725 : 132	580 : 106	0.8575	0.8483	0.8015	0.8483	0.7752	0.67 ± 0.05	0.8572	0.8489	0.8041	0.8489	0.7753	0.67± 0.05
	SSS	725 : 132	507 : 92	0.8574	0.8426	0.7336	0.8426	0.7957	0.67 ± 0.05	0.8574	0.8419	0.7335	0.8419	0.7957	0.67± 0.05
T-Links	K	725 : 132	548 : 101	0.8516	0.846	0.7952	0.846	0.7492	0.67 ± 0.04	0.8514	0.8459	0.8106	0.8459	0.7342	0.68± 0.03
SMOTE	K	725 : 132	584 : 584	0.7686	0.6756	0.7912	0.6756	0.7268	0.65± 0.03	0.7663	0.6861	0.7898	0.6861	0.7094	0.64± 0.02

Table 5.2: Evaluation results for LR

5.2 Decision Tree

Accuracy on the training set before tuning the model is nearly 100%, but it is much lower on the test set. So DT obviously has an overfitting issue. This is due to the default setting for the *max depth* is None. It will lead to nodes that are expanded until all leaves are pure. The deeper the tree, the better the accuracy of the training set. We can solve the problem by limiting the depth of the tree. We tuned *max depth* to different ranges of 2, 3, 5, 10. Based

on the evaluation results in Table 5.4, the best model performance for DT is the one using RSK with the optimal tuned parameter values of '*max depth = 3'*.

Parameters	Description	Tuned Range	Default
max_depth	The maximum depth of the tree.	10, 5, 4, **3**, 2	None

Table 5.3: Tuned parameters for DT

Resampling technique	CV	Original data points {0 : 1}	y_train {0 : 1}	DT without tuning parameters						DT with tuning parameters					
				Accuracy training set	Accuracy test set	Precision	Recall	F1	AUC	Accuracy training set	Accuracy test set	Precision	Recall	F1	AUC
Original	SK	725 : 132	580 : 106	0.9866	0.7701	0.7673	0.7701	0.7741	0.56 ± 0.05	0.8498	0.8413	0.7309	0.8413	0.7723	0.61 ± 0.03
	RSK	725 : 132	580 : 106	0.986	0.7736	0.7672	0.7736	0.7587	0.56 ± 0.05	0.8489	0.8419	0.7231	0.8419	0.7782	0.63 ± 0.06
	SSS	725 : 132	507 : 92	0.9856	0.7395	0.7568	0.7395	0.7261	0.55 ± 0.04	0.8521	0.8333	0.7196	0.8333	0.7697	0.62 ± 0.06
T-Links	K	725 : 132	537 : 105	0.9848	0.7561	0.7691	0.7561	0.7909	0.57 ± 0.05	0.8416	0.8402	0.7152	0.8402	0.7699	0.62 ± 0.05
SMOTE	K	725 : 132	587 : 587	0.991	0.6966	0.7704	0.6966	0.7364	0.56 ± 0.04	0.6711	0.5567	0.7853	0.5567	0.6569	0.57± 0.08

Table 5.4: Evaluation results for LR

5.3 Random Forest

RF is one of the DT family members, it also has an overfitting problem due to the default setting for *max depth* is None. DT is a single tree, but RF is a collection of trees. *n_estimators* is the parameter that controls the number of trees in the forest. We then ran multiple experiments with tunning *n_estimators* and *max depth* with different tuned ranges. Based on the evaluation results in Table 5.6, the best model performance for RF is the one by using SK or RSK with the optimal tuned parameter values of '*n_estimators = 500'* and '*max depth = 4*'.

Parameters	Description	Tuned Range	Default
n_estimators	The number of trees in the forest	**500**, 1000, 100	100
max_depth	The maximum depth of the tree.	10, 9, 8, 7, 6, 5, **4**, 3	None

Table 5.5: Tuned parameters for RF

Resampling technique	CV	Original data points {0 : 1}	y_train {0 : 1}	RF without tuning parameters						RF with tuning parameters					
				Accuracy training set	Accuracy test set	Precision	Recall	F1	AUC	Accuracy training set	Accuracy test set	Precision	Recall	F1	AUC
Original	SK	725 : 132	580 : 106	0.9863	0.8039	0.7626	0.8039	0.7569	0.61 ± 0.04	0.8463	0.846	0.7157	0.846	0.7782	0.66 ± 0.05
	RSK	725 : 132	580 : 106	0.9859	0.8062	0.7598	0.8062	0.7543	0.61 ± 0.05	0.8461	0.846	0.7157	0.846	0.7782	0.66 ± 0.05
	SSS	725 : 132	507 : 92	0.9856	0.7969	0.7608	0.7969	0.7794	0.62 ± 0.03	0.8464	0.845	0.714	0.845	0.774	0.66 ± 0.05
T-Links	K	725 : 132	522 : 118	0.9839	0.7912	0.7634	0.7912	0.8677	0.59 ± 0.07	0.8361	0.846	0.7158	0.846	0.7699	0.66 ± 0.07
SMOTE	K	725 : 132	580 : 580	0.9909	0.734	0.7758	0.734	0.7254	0.62 ± 0.05	0.7144	0.5765	0.7865	0.5765	0.6426	0.65 ± 0.06

Table 5.6: Evaluation results for RF

5.4 K-Nearest Neighbors

n_neighbors is the most important parameter to tune for KNNs. The experiments are executed with a different tuned range for *n_neighbors*. The evaluation result is shown in Table 5.6 where the one for RF is the one using SK with the optimal tuned parameter values of '*n_neighbors = 25'*.

Parameters	Description	Tuned Range	Default
n_neighbors	Number of neighbors to use	5, 10, 15, 20, **25**	5

Table 5.7: Tuned parameters for KNNs

Resampling technique	CV	Original data points {0 : 1}	y_train {0 : 1}	KNNs without tuning parameters						KNNs with tuning parameters					
				Accuracy training set	Accuracy test set	Precision	Recall	F1	AUC	Accuracy training set	Accuracy test set	Precision	Recall	F1	AUC
Original	SK	725 : 132	580 : 106	0.8597	0.8156	0.7583	0.8156	0.7481	0.59 ± 0.06	0.8462	0.8483	0.7478	0.8483	0.8049	0.65 ± 0.02
	RSK	725 : 132	580 : 106	0.8603	0.8197	0.7633	0.8197	0.7634	0.59 ± 0.07	0.8467	0.8472	0.7473	0.8472	0.7753	0.64 ± 0.06
	SSS	725 : 132	507 : 92	0.8568	0.8186	0.7654	0.8186	0.7907	0.59 ± 0.02	0.8477	0.8434	0.7245	0.8434	0.774	0.63 ± 0.03
T-Links	K	725 : 132	540 : 111	0.855	0.8004	0.7478	0.8004	0.8	0.57 ± 0.05	0.8376	0.8472	0.7514	0.8472	0.8114	0.62 ± 0.03
SMOTE	K	725 : 132	581 : 581	0.8319	0.6173	0.7693	0.6173	0.6507	0.59 ± 0.03	0.6932	0.4714	0.7802	0.4714	0.5422	0.60 ± 0.07

Table 5.6: Evaluation results for KNNs

5.5 Support Vector Machine

SVM has no overfitting except for the experiment with SMOTE. We tuned the two parameters *kernel* and *gamma*. However, there is no so big improvement after tuning these parameters. Based on Table 5.8, the best model is the one using SSS with the optimal tuned parameter values of '*kernel = rbf' and 'gamma = auto'*. It gives the best performance result for SVM.

Parameters	Description	Tuned Range	Default
kernel	the kernel type to be used in the algorithm	{'linear', 'poly', '**rbf**', 'sigmoid', 'precomputed'},	'rbf'
gamma	Kernel coefficient for **'rbf'**, 'poly' and 'sigmoid'.	{'scale', **auto'**}	'scale'

Table 5.7: Tuned parameters for SVM

Resampling technique	CV	Original data points {0 : 1}	y_train {0 : 1}	SVM without tuning parameters						SVM with tuning parameters					
				Accuracy training set	Accuracy test set	Precision	Recall	F1	AUC	Accuracy training set	Accuracy test set	Precision	Recall	F1	AUC
Original	SK	725 : 132	580 : 106	0.846	0.846	0.7157	0.846	0.7782	0.55 ± 0.09	0.846	0.846	0.7157	0.846	0.7782	0.57 ± 0.01
	RSK	725 : 132	580 : 106	0.846	0.846	0.7157	0.846	0.7782	0.57 ± 0.08	0.846	0.846	0.7157	0.846	0.7782	0.58 ± 0.09
	SSS	725 : 132	507 : 92	0.8464	0.845	0.714	0.845	0.774	0.60 ± 0.03	0.8464	0.845	0.714	0.845	0.774	0.60 ± 0.03
T-Links	K	725 : 132	544 : 102	0.8363	0.8459	0.7159	0.8459	0.7453	0.61 ± 0.08	0.8366	0.846	0.7164	0.846	0.7534	0.60 ± 0.05
SMOTE	K	725 : 132	579 : 579	0.7088	0.6313	0.7974	0.6313	0.7117	0.65 ± 0.06	0.6924	0.6151	0.8096	0.6151	0.7134	0.66 ± 0.03

Table 5.8: Evaluation results for SVM

5.6 Model Comparison

We can compare the results of all implemented algorithms in Table 5.9. The model with the best performance is LR using SK with optimal tuned parameter values of '*solver = liblinear*'. Followed by the RF with the optimal tuned parameter values of '*n_estimators = 500'* and '*max depth = 4*'. The results of the experiments with T-Links and SMOTE are less good than using stratified cross-validation. This might be due to the size of the original data set being small.

Model evaluation result								
Model	CV	Parameters	Accuracy training set	Accuracy test set	Precision	Recall	F1	AUC
LR	SK	solver = liblinear	0.8574	0.8506	0.8282	0.8506	0.7782	0.67± 0.04
DT	RSK	max depth = 3	0.8489	0.8419	0.7231	0.8419	0.7782	0.63 ± 0.06
RF	SK/RSK	n_estimators = 500 max depth = 4	0.8463	0.846	0.7157	0.846	0.7782	0.66 ± 0.05
KNNs	SK	n_neighbors = 25	0.8462	0.8483	0.7478	0.8483	0.8049	0.65 ± 0.02
SVM	SSS	gamma = auto	0.6924	0.6151	0.8096	0.6151	0.7134	0.66 ± 0.03

Table 5.9: Comparison results

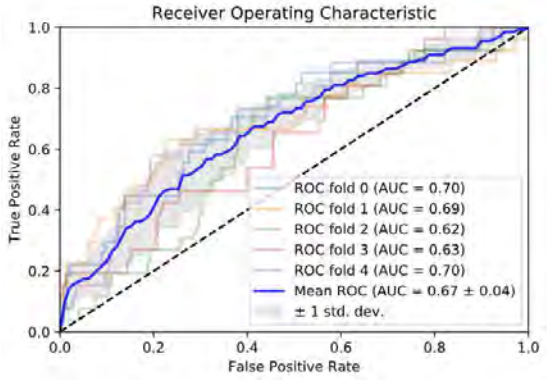

Figure 5.1: The optimal ROC/AUC for LR

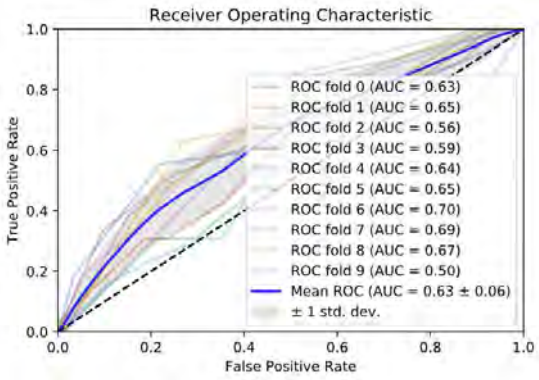

Figure 5.2: The optimal ROC/AUC for DT

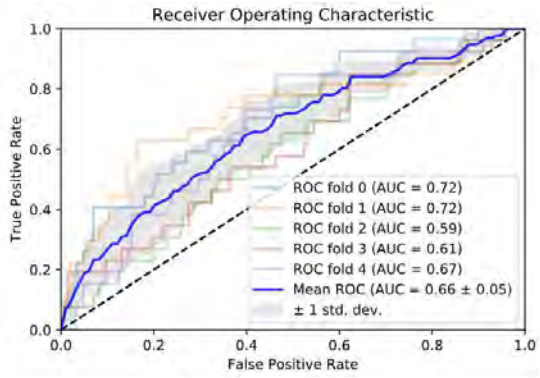

Figure 5.3: The optimal ROC/AUC for RF

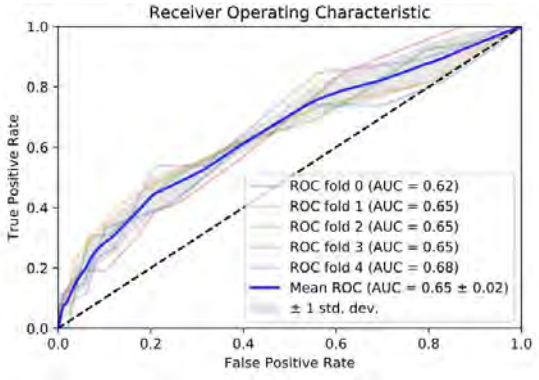

Figure 5.4: The optimal ROC/AUC for KNNs

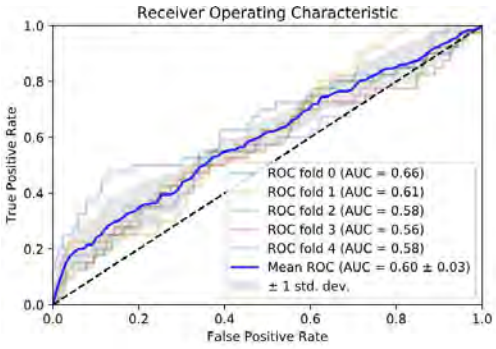

Figure 5.5: The optimal ROC/AUC for SVM

Reference

[1] Blattberg, Robert C.; Deighton, John., Harvard Business Review. Jul/Aug1996, Vol. 74 Issue 4, p136-144. 9p.

[2] Harvard Business Review, Comparative table of the C2C, B2C, and B2B business models https://hbswk.hbs.edu/archive/the-economics-of-e-loyalty

[3] Marketplaces by types of participants: C2C, B2C and B2B https://wiki.rademade.com/marketplace-c2c-b2c-b2b

[4] P. Rauyruen, K.E. Miller, Relationship quality as a predictor of B2B customer loyalty, J. Bus. Res., 60 (1) (2007), pp. 21-31

[5] Global Ecommerce Market Ranking, eshopworld, November 2018

[6] Burez, J., & Van den Poel, D. (2007). CRM at a pay-TV company: Using analytical models to reduce customer attrition by targeted marketing for subscription services. Expert Systems with Applications, 32(2), 277.

[7] Coussement, K., & Van den Poel, D. (2008a). Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. Expert Systems with Applications, 34(1), 313–327.

[8] Bolancé, Catalina & Guillen, Montserrat & Padilla Barreto, Alemar. (2016). Predicting Probability of Customer Churn in Insurance. 10.1007/978-3-319-40506-3_9.

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

[10] I. Ullah, B. Raza, A. K. Malik, M. Imran, S. U. Islam and S. W. Kim (2019), A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector, IEEE Access, vol. 7, pp. 60134-60149

[11] Ahmad, A.K., Jafar, A. & Aljoumaa, K. (2019) Customer churn prediction in telecom using machine learning in big data platform. J Big Data 6, 28

[12] Keramati, A., Jafari-Marandi, R., Aliannejadi, M., Ahmadian, I., Moza ari, M., and Abbasi, U. (2014). Improved churn prediction in telecommunication industry using data mining Applied Soft Computing Volume 24, November 2014, Pages 994-1012

[13] Ali Tamaddoni Jahromia, Stanislav Stakhovycha, Michael Ewing (2014). Managing B2B customer churn, retention and profitability. Industrial Marketing Management 43, 1258–1268

[14] Xu, M. and Walton, J. (2005). Gaining customer knowledge through analytical CRM. Industrial Management & Data Systems, 105(7):955-971.

[15] Stevens, R. (2005). B-to-B customer retention: Seven strategies for keeping your customers.

[16] Tomek, I. (1976). Two Modifications of CNN. IEEE Transactions on Systems Man and Communications SMC-6, 769–772.

[17] Silvia Cateni, Valentina Colla, Marco Vannucci, (2014) A method for resampling imbalanced datasets in binary classification tasks for real-world problems, Neurocomputing Volume 135, 5 July 2014, Pages 32-41

[18] N.V. Chawla , K.W. Bowyer , L.O. Hall , W.P. Kegelmeyer , SMOTE: syntheticminority over-sampling technique, J. Artif. Intell. Res. 16 (3) (2002) 321–357

[19] Bing Zhu, Bart Baesens, Seppe K.L.M. vanden Broucke(2017), An empirical comparison of techniques for the class imbalance problem in churn prediction, Information Sciences 408, 84–99

[20] Ron Koha((1995), A Study of CrossValidation and Bootstrap for Accuracy Estimation and Model Selection, International Joint Conference on Articial Intelligence (IJCAI)

[21] S. Neslin , S. Gupta , W. Kamakura , J. Lu , C. Mason , Detection defection: measuring and understanding the predictive accuracy of customer churn models, J. Market. Res. 43 (2) (2006) 204–211 ..

[22] A. Keramati , R. Jafari-Marandi , M. Aliannejadi , I. Ahmadian , M. Mozaffari , U. Abbasi , Improved churn prediction in telecommunication industry using data mining techniques, Appl. Soft Comput. 24 (2014) 994–1012

[23] Irfan Ullah, Basit Raza, Ahmad Kamran Malik, Muhammad Imran,Saif Ul Islam, and Sung Won Kim(2019). A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector. Digital Object Identifier 10.1109/ACCESS.2019.2914999

[24] Ron Koha, A k-nearest neighbor classification rule based on Dempster-Shafer Theory1(1995), International Joint Conference on Articial Intelligence (IJCAI)

[25] Kristof Coussement, Dirk Van den Poel. Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques Expert Systems with Applications 34 (2008) 313–327

[26] J. Burez, D. Van den Poel,(2009)Handling class imbalance in customer churn prediction, Expert Systems with Applications 36 (2009) 4626–4636

[27] Weiss, G. M. (2004). Mining with rarity: A unifying framework. SIGKDD Explorations, 6(1), 7–19.

[28] Drummond, C., & Holte, R.C. (2003). C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In: Workshop on learning from imbalanced data sets II, international conference on machine learning