E-Commerce Customer Churn Prediction

Created by:
Delabriliano Ismail

TABLE OF CONTENTS

01

Business Problem Understanding

Introduction

Many of the early-stage e-commerce business focuses on "Customer Acquisition", which is the act of gaining new customers.

Acquisition is particularly important for early-stage e-commerce looking to grow their customer base. But it's not a sustainable way to grow the company revenue in the long term.

In order to be sustainable in the long-term, an e-commerce business must also focus on customer retention. Knowing when a customer will churn, can help a company in general to retain their customer better for a long-term profit.

Fundamentally, churn occurs when a customer stops consuming from a company. A high churn rate equals to a low retention rate. Churn affects the size of a company customer base and has a big impact on the company's customer lifetime value.

Problem Statement

Maintaining a churn-rate is crucial for an e-commerce business long-term profit. However, not all company have a system that can detect which of their customer that will churn. This situation can have a bad consequences for the company, when they give a benefit/promotion to a non-churning consumer and let the other customer churned. In this case, the company will compound an expense and lost a potential revenue from the churned consumer. Or, if a company want to play safe, they can give the benefit/promotion for all of their customer base. However, it will require a quite large of expenditure without any certainty that the benefit was given to the right target.

Goals

Based on the introduction and the problem statement, the aims for this project is to develop a machine learning model that can classify on which customer of a company in general and e-commerce company specifically, that will churn on not churn.

By having a system that can classify which customer will churn, an e-commerce company can give a benefit/promotion with a clearer target, and this will reduce their expense compared to when a company give the benefit/promotion to all of their customer base. And this will also have an impact on the company's customer retention rate, and in return, the company will have a more sustainable revenue in the future.

In short, this project is aims to develop a machine learning model, that can be used by marketing/sales/any department related to benefit/promotion, that can classify which customer will churn and not churn, that can help a company to maintain their customer retention rate to generate more sustainable revenue.

02

Analytical Approach

Metrics

Class 0 = Non-Churn (Negative)
Class 1 = Churn (Positive)

- Type 1 error: False Positive

This error will increase the company expense by giving benefit and promotion to the non-churn customer and ignore the churning customer.

- Type 2 Error: False Negative

This error will make a company to overlook their churning customer without giving them any benefit/promotion to maintain their retention.

F1 Score

Based on the consequences, we will be focusing on both errors type. We want to make sure that the model can detect a churning customer as many as possible and ensure that the company give the benefit and promotion to the right customer (churned customer). We also have to notes that the data is imbalance (which will be shown later in the EDA). So the metrics that we will use is **F1 Score**.

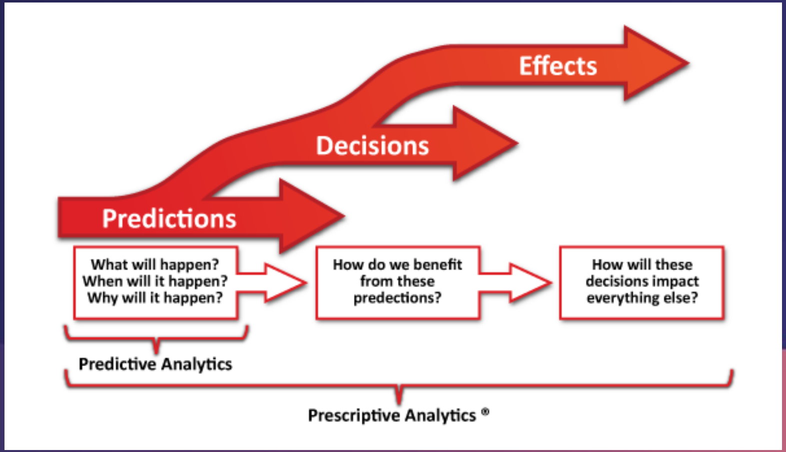
Precision and Recall are the two building blocks of the F1 score. The goal of the F1 score is to combine the precision and recall metrics into a single metric. At the same time, the F1 score has been designed to work well on imbalanced data.

$$F_1\text{--score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$$

Source: https://towardsdatascience.com/the-f1-score-bec2bbc38aa6

Analytic Approach

Based on the Problem statement and goals, we will be conducting the **Prescriptive Analytics** where we will predict which customer will churn, make a decision based on the prediction (giving benefit and promotion), and analyze how these decision impact the business.



03

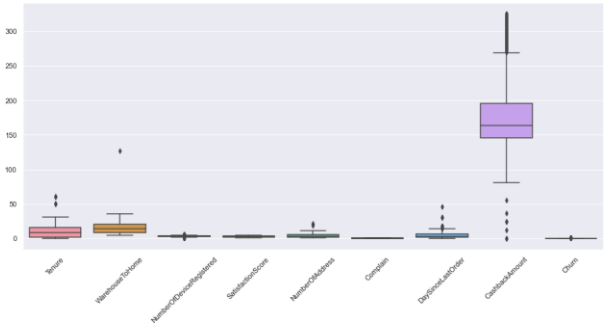
EDA and Data Preparation

Quick EDA

- Imbalance Dataset

```
0    3267
1     674
Name: Churn, dtype: int64
```

- Contains Outlier



- Contains Missing Value

	Features	DataType	Null	Null Percentage	Unique	Unique Sample
0	Tenure	float64	194	4.92	36	[0.0, 17.0]
1	WarehouseToHome	float64	169	4.29	33	[nan, 17.0]
2	NumberOfDeviceRegistered	int64	0	0.00	6	[3, 1]
3	PreferedOrderCat	object	0	0.00	6	[Mobile Phone, Grocery]
4	SatisfactionScore	int64	0	0.00	5	[4, 1]
5	MaritalStatus	object	0	0.00	3	[Married, Single]
6	NumberOfAddress	int64	0	0.00	14	[2, 21]
7	Complain	int64	0	0.00	2	[1, 0]
8	DaySinceLastOrder	float64	213	5.40	22	[46.0, 13.0]
9	CashbackAmount	float64	0	0.00	2335	[121.07, 209.74]
10	Churn	int64	0	0.00	2	[1, 0]

Handling Missing Value

- **Tenure** → Filled with PreferedOrderCategory and CashbackAmount
- **WarehouseToHome** → Filled with Median
- **DaySinceLastOrder** → Filled with Mode

Feature Selection

For our feature selection process, we will use the spearman correlation to choose which features we will be selecting. We will drop the features with low correlation with target, and drop features that have a really high correlation with the target.

From the correlation analysis, we know that 'Tenure' feature have the highest correlation (-0.42) with the target and 'WarehouseToHome' have the lowest correlation (0.075). However, we will not drop any features due to the features are already pre-selected from the source. And to avoid an underfit model, we will be using all the features that are available.

04

Model Building

Pipeline Preprocessing

Features Encoding

- **PreferedOrderCat:** We will encode the `PreferedOrderCat` features with One Hot Encoding (OHE) because the features does not have an order and only have 6 unique values.

- **MaritalStatus:** We will also encode the `MaritalStatus` features with One Hot Encoding (OHE) because the features also does not have an order and only have 3 unique values.

Model Benchmarking

For the test dataset we can see that the same 3 best performing models are XGboost, Random Forest, and Logistic Regression.

However, two out of three best algorithms is tree-based algorithm. And because the decision tree scoring is close to the logistic regression, we will choose decision tree instead.

Now we will use these 3 models with handled outliers and we will compare the scoring again before and after we handle the outliers.

	F1
XGB Testing CV (Mean)	0.618490
RF Testing CV (Mean)	0.580615
LogReg Testing CV (Mean)	0.574250
DT Testing CV (Mean)	0.564115
KNN Testing CV (Mean)	0.273810

Outlier Handling

We will not be dropping the outliers because we can see that the cashback amount contains many outlier, and the other reason we can't be sure that the outliers was due to measurement error, which in this case, the outliers is recommended to not be dropped

Source: (https://www.analyticsvidhya.com/blog/2021/05/why-you-shouldnt-just-delete-outliers/

Instead, we will be capping the outliers. The outlier that got past the upper bound will be change to the upper bound value, and the outlier outside of the lower bound will be change to the lower bound value

Source 2: https://medium.com/analytics-vidhya/detect-and-handling-outliers-53723d8ec17a)

Best Model Results

From the classification report on both models, we can see that the F1 score for class '1' has improved with our hyperparameter tuning. Thus, we will use the tuned models with XGBoost algorithms with SMOTE technique.

Classification Report Default XGBoost with SMOTE :

	precision	recall	f1-score	support
0	0.96	0.95	0.95	654
1	0.76	0.81	0.79	135
accuracy			0.92	789
macro avg	0.86	0.88	0.87	789
weighted avg	0.93	0.92	0.93	789

Classification Report Tuned XGBoost with SMOTE :

	precision	recall	f1-score	support
0	0.97	0.95	0.96	654
1	0.78	0.87	0.82	135
accuracy			0.94	789
macro avg	0.88	0.91	0.89	789
weighted avg	0.94	0.94	0.94	789

05

Conclusion

Recommended Model

From the classification report, we can conclude that our best model and chosen model is the XGBoost with SMOTE technique and tuned hyperparameters.

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way. The same code runs on major distributed environment (Hadoop, SGE, MPI) and can solve problems beyond billions of examples. ***Source: https://xgboost.readthedocs.io/en/stable/***

Whereas Synthetic Minority Oversampling Technique (SMOTE) is a statistical technique for increasing the number of cases in the dataset in a balanced way. The component works by generating new instances from existing minority classes that is supplied as input. The implementation of SMOTE does not change the number of majority classes. ***Source: https://docs.microsoft.com/en-us/azure/machine-learning/component-reference/smote***

From the classification report, we can summarize that if the model is implemented, we can detect 87% of all the churning customer (recall score) and 78% of our model customer churning prediction is precise (precision score)

Before-After

	Pred 1	Pred 0
Act 1	117	18
Act 0	33	621

Let's say we are an e-commerce company in USA.

According to [Invesp](https://www.invespcro.com/blog/global-online-retail-spending/) (Digital Consulting Firm), the average revenue per online shopper on e-commerce business in USA is $1,804.

Using our last confusion matrix on test dataset, let's say we currently have 789 customers (117 + 18 + 33 + 621).

Before-After

	Pred 1	Pred 0
Act 1	117	18
Act 0	33	621

Before

- 789 customer at the beginning of a month
- 135 customer will churn (117+18)
- Revenue from each customer = $1,804
- Revenue = 789 * 1804 = $1.423.356

- Marketing Cost = 20% of Revenue (Average marketing cost for e-commerce) Source: https://boldist.co/marketing-strategy/ecommerce-digital-marketing-budget/
- Marketing Cost = 20% * $1.423.356 = $284.671
- Average Customer Acquisition Cost for e-commerce (retail) = $10 Source: https://www.propellercrm.com/blog/customer-acquisition-cost

Let's say the cost for retaining a customer is the same with the cost for acquiring new customer:
- Customer Retention Cost = $10

Before-After

	Pred 1	Pred 0
Act 1	117	18
Act 0	33	621

Before

Based on those data, the company with no system to detect a churning customer, will give all the customer base a benefit/promotion from the marketing budget. And spend the rest on acquiring new customer.

- Spent on customer retention: 789 * $10 = $7.890
- Spend the rest on new customer acquisition = $284.671 - $7.890 = $276.781
- New customer acquired = $276.781/10 = 27.678
- Customer for next month = 27.678 + 789 = 28.467 total customer
- Potential revenue for next month = 28.467 * $1.804 = **$51.354.468**

Before-After

	Pred 1	Pred 0
Act 1	117	18
Act 0	33	621

After

- 789 customer at the beginning of a month
- 135 customer will churn (117 True Positive + 18 False Negative)
- 150 customer detected will churn (117 True Positive + 33 False Positive)
- Revenue from each customer = $1,804
- Revenue = 789 * 1804 = $1.423.356

- Marketing Cost = 20% of Revenue
- Marketing Cost = 20% * $1.423.356 = $284.671
- Average Customer Acquisiton Cost for e-commerce (retail) = $10
- Customer Retention Cost = $10

After implementing ML models, the company will no longer give all the customer a benefit/promotion. Instead, they will only give the benefit/promotion to the detected customer that will churn.

Before-After

	Pred 1	Pred 0
Act 1	117	18
Act 0	33	621

After

- Spent on customer retention: 150 * $10 = $1.500
- Spend the rest on new customer acquisition = $284.671 - $1.500 = $283.171
- Churned customer = 33 (False Positive)
- New customer acquired = $283.171/10 = 28.317
- Customer for next month = 28.317 + 789 - 33 = 29.073 total customer
- Potential revenue for next month = 29.073 * $1.804 = **$52.447.632**

Impact

We can see the potential revenue differences from implementing the ML models on our hypothetical case above. from the hypothetical case above, By implementing the ML models, the company can gain up to **1 million USD** ($ 1.093.164 to be exact) of potential revenue on the next month from the higher number of acquired customers.

Limitation

The model was built on a dataset that was already pre-selected and limited, so the author realize there will be a limitation for the models/project in which the prediction will be less accurate on a certain condition. The limitation of this project are:

- Only limited amount of features used in this model

- Due to the outlier handling, there is a possibility that this model will got the prediction wrong when the data considered as outlier. The outlier criteria on each columns is:

Features	Outlier
Tenure	Less than -19 / More than 37 year
WarehouseToHome	Less than -9 / More than 39 km
NumberOfDeviceRegistered	Less than 2 / More than 5 device
PreferedOrderCat	Category outside of 'Laptop & Accessory', 'Mobile', 'Fashion', 'Others', 'Mobile Phone', 'Grocery'
SatisfactionScore	Outside of scale from 1 to 5
MaritalStatus	Status outside of 'Single', 'Married', 'Divorced'
NumberOfAddress	Less than -4 / more than 12 address
Complain	less than -1,5 / more than 2,5 complain
DaySinceLastOrder	less than -5,5 / more than 14,5 days
CashbackAmount	Less than 71,375 / more than 269,575

Implementation and Recommendation

This project/machine learning model is recommended to be implemented by marketing/sales/any department related to benefit/promotion. The department can implement it in end of month of each month, to calculate or forecast the expected revenue on the next month. This model also can be implemented whenever the management or BODs want to measure their churn rate.

Future Recommendation
To improve this project/machine learning models, future projects can considers:

- Adding more features that is related to the target (customer Churn), such as Age, Redeemed vouchers/promo, Gender, etc.
- Use other algorithm such as SVM or LGBM and try other feature engineering such as scaling.
- Use other method to fill missing value / handle outlier

THANKS!