A picture containing icon

Description automatically generated

**CCT College Dublin**

**Assessment Cover Page**

|  |  |
| --- | --- |
| **Module Title:** | Strategic Thinking |
| **Assessment Title:** | CA 2 |
| **Lecturer Name:** | James Garza ([james@cct.ie](mailto:james@cct.ie)) |
| **Student Full Name:** | [Riccardo Possieri](https://moodle.cct.ie/user/view.php?id=13536&course=2625) |
| **Student Number:** | sba23439 |
| **Assessment Due Date:** | 15th December 2023 23:59 |
| **Date of Submission:** | 26/11/2023 |

**Declaration**

|  |
| --- |
| By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution. |

Table of contents

1. Introduction

2. Goal

3. Import Datset & libraries

4. Overview

5. EDA

6. Data Pre-processing

7. Splitting

8. Modelling

9. Oversampling

10. Conclusions

11. References

12. GitHub repo link

1. Introduction

The *'E Commerce Dataset'* and *'E Comm details'* Excel files have been helpful to understand and give us a comprehensive exploration into customer churn within a company. As we discussed in the first CA1, we are going to predict and understand the factors influencing customer attrition using the heatmap with correlation matrix, measuring the strength of the linear relationship between our target variable *Churn* and another variable, one by one. This second part of capstone project involved the use of two principal models for our case and having a prediction.

2. Goal

Our main goal in this project was to create a machine learning model that could predict and help reduce customer churn for the e-commerce company. Customer churn refers to when customers stop using the service, and understanding and predicting this behavior is crucial for businesses. To achieve this goal, we focused on a key variable called *'Churn*.' We made *'Churn'* our target variable. The idea was to use other relevant features, or characteristics, in the dataset to build a machine learning algorithm that could accurately predict whether a customer is likely to churn or not.

3. Import Dataset & Libraries

The initial step involved importing necessary datasets (*'E Commerce Dataset'* and *'E Comm details'*) and libraries essential for data analysis, manipulation, and machine learning.

4. Overview

The *'E Commerce Dataset'* consists of 20 features and 5630 observations, with a mix of numerical and categorical columns.

Target Variable – *Churn –* imbalanced variable

No churn: 4682

Churn: 948

A blue and orange rectangular chart

Description automatically generated

Visualization

Visualizations, including box plots, were employed to find some patterns and relationships between features and the target variable. Key observations include gender-specific churn rates and city-tier-specific order counts.

*Churn* by *Preferred Payment Mode*. Debit and credit cards looked like the most frequently utilized payment modes.

A graph of a bar chart

Description automatically generated with medium confidence

*Churn* by *Preferred Login Device*. The analysis of churn in relation to the preferred login device indicated us that people who use a computer as their preferred login device demonstrated a higher percentage of churn compared to those who opt for a mobile phone as their preferred login device.

A graph of a login device

Description automatically generated

5. EDA

In the exploration of the *'E Commerce Dataset*,' we did a comprehensive analysis to understand the nuances and patterns within the data. The following conclusions have been drawn from the various aspects of the dataset: We proceed to follow the variables description we did in the first step and analyse, handling missing values first, and pre-processing them one by one, if necessary, later. "Tenure" had 9% of zeros because of the longevity in the company and it had 4.7% of missing values which we proceeded to handle with a central tendency with the mean to maintain the overall distribution of this feature. After that, we replaced the missing values of *'WarehouseToHome'* with the median. Consequently, we asked ourself the following questions:

* Is there any realtionship between *Churn* and *Gender*?
* what about *HourSpendOnApp* comparing that with *Gender*?
* Which *CityTier* has the highest *OrderCount*?

An examination of the relationship between *Gender* and *Churn* revealed that since the percentage of *Churn* in male customers is slightly higher than that of female customers, we could assume that we should focus more on addressing the needs of male customers to positively affect our target variable and keep working on Female gender to increase the Total number of customers.

For the next question, the first step has been handling 4.5% of missing values with a mean and keep save the same frequency because, as the html report shows, *HourSpendOnApp* had a tendency as three hours. Then we focused on a relationship between *HourSpendOnApp* with *Gender*. We performed the independent *t-test* to determine if there was a significant difference between the mean of *HourSpendOnApp* in Male & *HourSpendOnApp* in Female:

Mean *HourSpendOnApp* (Male): 2.920991368409478

Mean *HourSpendOnApp* (Female): 2.947420571972913

T-test p-value: 0.17027252509743426

There was no strong statistical evidence to suggest that the average "*HourSpendOnApp*" was significantly different between male and female customers. The p-value of 0.17027252509743426 was greater than the conventional threshold *of 0.05*, indicating that any observed differences in means could have been due to random variation. The FacetGrid showed out our analysis and we can see they actually look almost the same. This suggested us that, on average, the time spent on the mobile application was consistent across gender categories.

A graph of two people

Description automatically generated

To handle all the missing values we still had, we proceed to apply the mean to the last four variables which had missing values too and they were:

* *OrderAmountHikeFromlastYear* had 265 (4.7%),
* *CouponUsed* had 256 (4.5%),
* *OrderCount* had 258 (4.6%),
* *DaySinceLastOrder* had 307 (5.5%)

Once we didn't have any missing values and we could have focused on *CityTier* variable and comparing it with other features and answering ourselves the last question: Which *CityTier* has the highest *OrderCount*?

With a preliminary study, we saw *CityTier* 1 had the highest value with 10836,4328 orders counted.

* 10836.432800
* 627.032018
* 5471.600335

Following that, we transformed those numbers into INT values, and we got:

* 10835
* 627
* 5471

The exploration of *'CityTier'* unveiled that *CityTier 1* exhibited the highest *'OrderCount*,' indicating a higher volume of orders compared to other city tiers.

Conclusions of EDA

The exploratory data analysis not only provided us insights into individual features, but it also put a good understanding for subsequent modeling. By addressing gender-specific churn rates, understanding the significance of certain features, and recognizing the influence of city tiers on order counts, we were better equipped to give better conclusions from the forthcoming predictive models. We've made sure that our dataset was now complete and understandable. This ensured that the data we had was good enough for training and evaluating accurate models. Subsequently, we were ready to start preprocess data and building models.

6. Data Pre-processing

We did some important things to get the data ready for using models. First, we made sure certain columns, like *'CouponUsed'* and *'OrderAmountHikeFromlastYear*,' only had whole numbers (integers). This helped us to use machine learning tools in the forward step. Next, we used a Label encoding to turn categories into numbers. This made it easier for the computer to understand and use them for predictions.

Finally, we looked at how different factors in our data relate to the main thing we're interested in, which was 'Churn'. The numbers and visualizations we got from this helped us understand those relationships better. 1 meant the two features were going on the same directions, -1 the opposite. If the number was 0 it meant there was no connection between them.

A screenshot of a computer screen

Description automatically generated

Features positively correlated with *'Churn'*: 'Complain,' *'MaritalStatus,' 'NumberOfDeviceRegistered,' 'SatisfactionScore,' 'PreferedOrderCat,' 'CityTier,' 'WarehouseToHome,' 'NumberOfAddress.*'  
Features negatively correlated with 'Churn': *'Tenure,' 'DaySinceLastOrder,' 'CashbackAmount.'*

A graph with different colored bars

Description automatically generated

The heatmap visually represented these correlations, offering a comprehensive overview of feature interactions.

We would have used PCA to identify the most important features in our dataset and then doing these processes, but PCA is an unsupervised technique, so it means that it doesn’t take the target variable into account. This was the reason why we did not apply the PCA nor the LDA, which it is used to maximize the separation of the classes and not the relation between features and *Churn* variable.

7. Data Splitting

The dataset was split into training and testing sets, with 10% reserved for testing. We decided to use 10% to not make the Assessment not to much long and keep follow the questions of the Professor. We also didn’t do the cross-validation part, so we didn’t check which was the best parameter to use in each model. We finally chose two models: Logistic Regression and Random Forest Classifier.

8. Modelling

Logistic Regression was trained on the dataset, and its accuracy on the test set was found to be 85.26%

* *Precision: 0.63*
* *Recall: 0.28*
* *F1-score: 0.39*

Random Forest Classifier demonstrated superior performance compared to Logistic Regression, achieving an accuracy of 97.87%.

* *Precision: 0.91*
* *Recall: 0.71*
* *F1-score: 0.80*

Based on our report, it looked like that the Random Forest model performed better than the Logistic Regression model, especially when considering metrics like precision, recall, and F1-score for the positive class (Churn = 1). In general, the RF model indicated better performance in identifying customers who are likely to churn.

* Logistic Regression Accuracy: 85.26%
* Random Forest Accuracy: 97.87%

9. Oversampling

It is an important step we decided to do only at the end because we wanted to explore and understand the natural distribution of our data. We Trained initial LR & RF models without balancing to gauge their performance and how well our model was handling the imbalanced classes and then, comparing it once we balanced our target variable. To further enhance model performance, oversampling of the minority class (Churn = 1) we performed using SMOTE (Synthetic Minority Over-sampling Technique). We resampled the dataset and standardized it before training Logistic Regression and Random Forest models.

* Logistic Regression Accuracy (with oversampling): 75.13%
* Random Forest Accuracy (with oversampling): 97.69%

10. Conclusions

In our capstone project, we did a study of a dataset about customer behavior in an e-commerce company to understand and predict customer churn. Our main goal was to build a model that could predict if a customer might leave the company and get the factors that influence this. We started our project by examining the information in the dataset. To make sense of the data, we created visual representations. These plots allowed us to discover that the rate at which customers stop using the service (churn rate) varies between different genders. Additionally, we observed that the location of customers, categorized by city tiers, has an impact on how many orders they place. This means that customers from different cities tend to order different amounts. These insights from our initial analysis helped us understand important factors that might influence customer churn in the e-commerce company.

Then, we did a thorough analysis of the data, handling missing values and checking relationships between variables. We made sure our dataset was complete and ready for building models.

Regarding the models we used, we wanted to see how well our models predicted customer churn, but we noticed that the data had more information about customers who didn't churn. This made the models biased, especially the one using Random Forest. So, we tried a technique called SMOTE to fix this. SMOTE helped our models see both types of customers (those who churn and those who don't) more equally and balanced them. For sure, when we balanced our target variable, we loose information, but (CohenJ2021)Balancing data gives us the same amount of information to help predict each class and therefore gives a better idea of how to respond to test data. Usually, this is something we don’t do. You can sometimes upsample test data anyway just to see if your model works well on minority classes as well. What’s most important to keep in mind is that you don’t want to upsample data and only then do a data split into train and test set. This will likely result in having elements of train data copied perfectly into test data and artificially boost your model scores. The only time you would ever upsample test data is after a data split, just like you only perform data balancing on train data.

After using SMOTE, we checked how well our models did. For the Logistic Regression model, the overall accuracy slightly went down, but it got better at finding customers who churned actually. The Random Forest model, even after using SMOTE, still performed very well. It stayed accurate and balanced in finding both types of customers. This made it a strong choice for handling customer churn. In the end, using techniques like SMOTE helped our models do a better job, especially in finding customers who might stop using our service. These findings will help us make our models even better and decide how to use them effectively to deal with the challenges of customer churn.

11. References

https://seaborn.pydata.org/generated/seaborn.violinplot.html

https://www.tutorialspoint.com/python\_data\_science/python\_p\_value.htm

Brownlee, J. (2020). Imbalanced Classification with Python. Machine Learning Mastery

Müller, A.C. and Guido, S. (2018) Introduction to machine learning with python: A guide for data scientists. Sebastopol: O’Reilly Media

McKinney, W. (2022) Python for Data Analysis: Data wrangling with pandas, NumPy, and Jupyter. Beijing: O’Reilly

iguazio.com (2014). What is Model Accuracy in Machine Learning. [online] Iguazio. Available at: https://www.iguazio.com/glossary/model-accuracy-in-ml/ [Accessed 2 December 2023].

Cohen, J. (2021). Machine Learning: Target Feature Label Imbalance Problem and Solutions. [online] Medium. Available at: https://towardsdatascience.com/machine-learning-target-feature-label-imbalance-problem-and-solutions-98c5ae89ad0.

‌

12. GitHub repo link

<https://github.com/riccardopossier/CA2_Strategic_Thinking_Possieri_Riccardo/tree/main>