A picture containing icon

Description automatically generated

**CCT College Dublin**

**Assessment Cover Page**

|  |  |
| --- | --- |
| **Module Title:** | Strategic Thinking |
| **Assessment Title:** | CA 3 Final Submission |
| **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:** | 12th May 2024 23:59 |
| **Date of Submission:** | 12/05/2024 |

**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 Dataset & libraries

4. Overview

5. EDA

6. Data Pre-processing

7. Splitting & Hyperparameter Tuning and Cross-Validation

8. Modelling

9. Oversampling

10. Conclusions

11. References

12. GitHub repo link

1. Introduction

In this project we are going to talk about churn. Churn is a measurement of the percentage of all those accounts that delete or cancel or choose not to renew their subscriptions. Basically, it is a measure of how many customers are going to stop using a service or product. Based on studies, retaining an existing customer is so much cheaper than acquiring a new one. For this reason, companies started to proactively identify customers at risk of churning and implement strategies in order to retain them.

The *'E Commerce Dataset'* and *'E Comm details'* Excel files are helpful to understand and give us a comprehensive exploration into customer churn within a company. As we discussed in the first CA1 and CA2, we predict and understand the factors influencing customer attrition using machine learning models and applying statistical methods to get a more precise result.

2. Goal

Our main goal in this project is to create a machine learning model that could predict and help reduce customer churn for the e-commerce company. To achieve this goal, we focus on a key variable called *'Churn*.' We do *'Churn'* our target variable. It is important to know that churn can be caused by different reasons such as: customer no longer values the product, motivating factors to use the product no longer exists, customer frustrated with product user experience, the product lacks a mandatory capability required by the user, and so on and so forth. The dataset we are going to use has different features and our goal is to determine which ones are the most important to predict the churn.

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. We use:

* Pandas: one of the most powerful libraries for data analysis and data manipulation.
* Numpy: the fundamental package for numerical computations.
* Matplotlib: the extension .pyplot is used for creating visualization information such as histograms.
* Seaborn: it is a built in library on top of matplotlib which provides a great level of interface for creating statistical data visualizations.
* Missingno: a library designd to visualize the missing values within a dataset.
* Plotly: enabling users to explore the data dynamically.
* scikit-learn: it is the perfection for splitting data into training and testing sets in evaluation model step. It is also implements cross-validation techniques.
* imbalanced-learn: used to balance the target variable. It is useful to oversampling, undersampling and generating synthetic samples to balance the dataset.
* Shap: Shapley additive exPlanations is the library used to attribute the prediction of an instance to its features values and to understand how each feature can contributes to the prediction of the model.
* Scipy: it is a built in library on top of numpy and it is useful for additional functionalities for optimization, integration, interpolation, etc.

4. Overview

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

|  |  |
| --- | --- |
| **Variable** | **Discerption** |
| CustomerID | Unique customer ID |
| Churn | Churn Flag |
| Tenure | Tenure of customer in organization |
| PreferredLoginDevice | Preferred login device of customer |
| CityTier | City tier |
| WarehouseToHome | Distance in between warehouse to home of customer |
| PreferredPaymentMode | Preferred payment method of customer |
| Gender | Gender of customer |
| HourSpendOnApp | Number of hours spend on mobile application or website |
| NumberOfDeviceRegistered | Total number of deceives is registered on particular customer |
| PreferedOrderCat | Preferred order category of customer in last month |
| SatisfactionScore | Satisfactory score of customer on service |
| MaritalStatus | Marital status of customer |
| NumberOfAddress | Total number of added added on particular customer |
| Complain | Any complaint has been raised in last month |
| OrderAmountHikeFromlastYear | Percentage increases in order from last year |
| CouponUsed | Total number of coupon has been used in last month |
| OrderCount | Total number of orders has been places in last month |
| DaySinceLastOrder | Day Since last order by customer |
| CashbackAmount | Average cashback in last month |

Target Variable – *Churn –* imbalanced variable

A blue and orange rectangular chart

Description automatically generated

No churn: 4682

Churn: 948

Alerts of missing values, unique values, and zeros in the dataset. Information got from the Ecommerce html file:

|  |  |
| --- | --- |
| [Tenure](file:///C:\Users\Riccardo\OneDrive\Desktop\Higher%20Diploma\STRATEGIC%20THINKING\capstone-project-riccardopossier\ecommerce.html#pp_var_8156666784084329739) has 264 (4.7%) missing values | **Missing** |
| [WarehouseToHome](file:///C:\Users\Riccardo\OneDrive\Desktop\Higher%20Diploma\STRATEGIC%20THINKING\capstone-project-riccardopossier\ecommerce.html#pp_var_-1414389591538786379) has 251 (4.5%) missing values | **Missing** |
| [HourSpendOnApp](file:///C:\Users\Riccardo\OneDrive\Desktop\Higher%20Diploma\STRATEGIC%20THINKING\capstone-project-riccardopossier\ecommerce.html#pp_var_-4154967207578089478) has 255 (4.5%) missing values | **Missing** |
| [OrderAmountHikeFromlastYear](file:///C:\Users\Riccardo\OneDrive\Desktop\Higher%20Diploma\STRATEGIC%20THINKING\capstone-project-riccardopossier\ecommerce.html#pp_var_3407155520802751395) has 265 (4.7%) missing values | **Missing** |
| [CouponUsed](file:///C:\Users\Riccardo\OneDrive\Desktop\Higher%20Diploma\STRATEGIC%20THINKING\capstone-project-riccardopossier\ecommerce.html#pp_var_2156153851446334540) has 256 (4.5%) missing values | **Missing** |
| [OrderCount](file:///C:\Users\Riccardo\OneDrive\Desktop\Higher%20Diploma\STRATEGIC%20THINKING\capstone-project-riccardopossier\ecommerce.html#pp_var_205417457139346341) has 258 (4.6%) missing values | **Missing** |
| [DaySinceLastOrder](file:///C:\Users\Riccardo\OneDrive\Desktop\Higher%20Diploma\STRATEGIC%20THINKING\capstone-project-riccardopossier\ecommerce.html#pp_var_-5854907745481663617) has 307 (5.5%) missing values | **Missing** |
| [CustomerID](file:///C:\Users\Riccardo\OneDrive\Desktop\Higher%20Diploma\STRATEGIC%20THINKING\capstone-project-riccardopossier\ecommerce.html#pp_var_5092741353282496797) has unique values | **Unique** |
| [Churn](file:///C:\Users\Riccardo\OneDrive\Desktop\Higher%20Diploma\STRATEGIC%20THINKING\capstone-project-riccardopossier\ecommerce.html#pp_var_5926730789674045169) has 4682 (83.2%) zeros | **Zeros** |
| [Tenure](file:///C:\Users\Riccardo\OneDrive\Desktop\Higher%20Diploma\STRATEGIC%20THINKING\capstone-project-riccardopossier\ecommerce.html#pp_var_8156666784084329739) has 508 (9.0%) zeros | **Zeros** |
| [Complain](file:///C:\Users\Riccardo\OneDrive\Desktop\Higher%20Diploma\STRATEGIC%20THINKING\capstone-project-riccardopossier\ecommerce.html#pp_var_-6256616895293254303) has 4026 (71.5%) zeros | **Zeros** |
| [CouponUsed](file:///C:\Users\Riccardo\OneDrive\Desktop\Higher%20Diploma\STRATEGIC%20THINKING\capstone-project-riccardopossier\ecommerce.html#pp_var_2156153851446334540) has 1030 (18.3%) zeros | **Zeros** |
| [DaySinceLastOrder](file:///C:\Users\Riccardo\OneDrive\Desktop\Higher%20Diploma\STRATEGIC%20THINKING\capstone-project-riccardopossier\ecommerce.html#pp_var_-5854907745481663617) has 496 (8.8%) zeros | **Zeros** |

A colorful circle with a white background

Description automatically generated

As we can see in the previous plot as well, almost 17% of the customers is churned, and the gender is divided into 40% females, and 60% males.

Visualization of categorical variables

A group of pie charts

Description automatically generated

As per Pie Charts, we can assume the most frequent value for each of all categorical variables in our dataset. Mobile phone is the most used to login to the platform. Debit card is the most used method of payment. Laptop and accessories are the most preferred order category and Married is the most frequent maritial status value.After that*,* 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.

A graph of a bar chart

Description automatically generated with medium confidence

| **State** | **Frequency** | **Relative Frequency** |
| --- | --- | --- |
| 0 | Debit Card | 2314 | 0.411012 |
| 1 | Credit Card | 1501 | 0.266607 |
| 2 | E wallet | 614 | 0.109059 |
| 3 | UPI | 414 | 0.073535 |
| 4 | COD | 365 | 0.064831 |
| 5 | CC | 273 | 0.048490 |
| 6 | Cash on Delivery | 149 | 0.026465 |

As we saw in piecharts, debit and credit cards look like the most frequently utilized payment modes, but it is also interesting how cash on delivery has the highest rate of churn considering the total people using that method: almost 120 people churned of 370 people using COD.

Churn by Preferred Login Device.

A graph of a login device

Description automatically generated

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

Churn by HourSpendOnApp and SatisfactionScore

A comparison of a graph

Description automatically generated with medium confidence

As per boxplot, we can assume that hours spent on app is not a very effective feature to compare with churn because eighter the customer churned or not, they still spent between 2 and 3 hours on the app. Otherwise, satisfaction score is showing up 3 as the satisfaction score of no churn customers.

5. EDA

We did a comprehensive analysis to understand the nuances and patterns within the data. "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.

* Is there any realtionship between *Churn* and *Gender*?
* what about *HourSpendOnApp* comparing that with *Gender*?
* Which *CityTier* has the highest *OrderCount*?
* Is Tenure vs Churn an important relationship to see?

Answering the first question, we already know that there are more males than females in the dateset. Nevertheless, the percentages are almost equals:

Churn 0 1 Total Churn Percentage

Gender

Female 1898 348 2246 15.494212

Male 2784 600 3384 17.730496

HourSpendOnApp has a tendency as three hours as the html report shows. First at all we decide to handle the 4.5% of missing values with a mean and keep save the same frequency. Then we focus on a relationship between HourSpendOnApp with Gender. After the analysis, we could assume we should focus more on addressing the needs of male customers to positively affect our target variable and keep working on Female gender. For the next question, the FacetGrid from seaborn showed out our analysis and we could see they look almost the same.

A graph of two people

Description automatically generated

Finally, 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. Then we focused on *CityTier* variable and comparing it with other features and answering ourselves the last question: Which *CityTier* has the highest *OrderCount*? The exploration of *'CityTier'* unveiled that *CityTier 1* exhibited the highest *'OrderCount*,' indicating a higher volume of orders compared to other city tiers.

OrderCount

CityTier

1 10835

2 627

3 5471

And then, we plot the most important plot: Tenure vs Churn which showed us up that the median tenure for churned customers is significantly lower (1) than non-churned customers (10). It suggests that customers with lower tenure are more likely to churn.

A screenshot of a graph

Description automatically generated

6. Data Pre-processing

In this process we first encode and convert all the categorical columns to numerical representations and then we did the correlation matrix. 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. The results of churn\_corr\_vector have been obtained by the formula of:

Where:

is the mean of churned customers.

is the mean of no churned customers.

is the standard deviation of churn.

is the number of observations of churned customers.

is the number of observations of no churned customers.

is the total number of observations of churn.

The point-biserial correlation is a correlation between a binary variable (our target variable) and a continuous variable. A dichotomous variable has only two categories which we already know thanks to the overview and EDA we processed before. A continuous variable can take any numeric value, which means it can take all the other variables we have in our data frame. In this case, more specifically, we study the point-biserial correlation that is essentially the same as the biserial correlation, but it is specifically used when the continuous variables are measured on an interval or ratio scale. The basic difference between biserial correlation and point biserial correlation is that the former can involve any scale, while the latter is strictly limited to interval or ratio scales. In this project, is it required to be between -1 and 1.

A screenshot of a computer screen

Description automatically generated

A graph with different colored bars

Description automatically generated

1. Complain: 𝑟𝑝𝑏=0.250188

2. MaritalStatus: 𝑟𝑝𝑏=0.140316

3. NumberOfDeviceRegistered: 𝑟𝑝𝑏=0.107939

4. SatisfactionScore: 𝑟𝑝𝑏=0.105481

5. PreferedOrderCat: 𝑟𝑝𝑏=0.104569

6. CityTier: 𝑟𝑝𝑏=0.08470

7. WarehouseToHome: 𝑟𝑝𝑏=0.069544

8. NumberOfAddress: 𝑟𝑝𝑏=0.043931

9. Gender: 𝑟𝑝𝑏=0.029264

10. HourSpendOnApp: 𝑟𝑝𝑏=0.018126

11. CouponUsed: 𝑟𝑝𝑏=−0.001430

12. OrderAmountHikeFromlastYear: 𝑟𝑝𝑏=−0.007075

13. CustomerID: 𝑟𝑝𝑏=−0.019083

14. PreferredPaymentMode: 𝑟𝑝𝑏=−0.026519

15. OrderCount: 𝑟𝑝𝑏=−0.028308

16. PreferredLoginDevice: 𝑟𝑝𝑏=−0.051099

17. CashbackAmount: 𝑟𝑝𝑏=−0.154118

18. DaySinceLastOrder: 𝑟𝑝𝑏=−0.156152

19. Tenure: 𝑟𝑝𝑏=−0.335513

We know that negative correlation means that as one variable increases, the other tends to decrease. For example: the lower is the Tenure, the higher is the probability to get a churn, as we saw in the first step of the Overview of the capstone. This is why we chose positive & negative values as well of the correlation matrix we just did. But in our case, let's also take the example if "Days Since Last Order". So, while it might seem counterintuitive that as "Days Since Last Order" increases, the likelihood of churn also increases, it's still a negative correlation because it's the trend of the data points that matters, where longer intervals between purchases are associated with higher churn rates.

Considering that, we have considered just 8 features as the most important ones to predict churn because of the correlation matrix.

7. Data Splitting & Hyperparameter Tuning and Cross-Validation

These steps involve optimizing the performance of our model and assessing its generalization ability using techniques like GridSearchCV and k-fold cross-validation. We ensure that the optimization process is based on the original distribution of the data, so we tune the hyperparameters and evaluating the model's performance before the oversampling we do at the end. We used cv=10 and with the GridsearchCV we defined grid\_model as the variable with the best parameter chosen for each model and then: best\_model = grid\_logreg.best\_estimator\_.

It is important to remark that when we evaluate the different hyperparameters in SVM model such as C setting, there is the risk of overfitting so there couldn’t be a prediction because the model is adapting too much to the training set. This could happen because these parameters can be tuned until the model performs optimally and it means that it could include information from the test set into the model and compromise the result. To mitigate the problem we do training, validation and testing set of our data, so first we trian the model on the training set. Then, the performance of the model is evaluated on the validation set and if and only if the experiment appears successful based on validation set, we run the final evaluation on the testing set.

However, if we split our dataset into three sets (training set , validation set , and testing set) we reduce the number of samples used for the training model, and the results could vary depending on the specific random selection of the (train, validation) pairs. This is the reason why we employ the technique called cross validation (CV). With this technique we don’t need to use the validation test anymore.

A graph of progress bar

Description automatically generated

*Introduction to Machine Learning with Python (n.d.)*

We applied k-fold cross-validation where ‘*k*’ is the number of subsets we decided to split our data into, which are 10. First at all, we divided our data into 70% training data, and 30% testing data. Subsequently, we subdivided our training data into 10 splits and 10 folds, using the parameter *cv=10*.

8. Modelling

In our machine learning project, we train different models and then we are going to select the one with the best performance. However, there is the possibility to improve the model. One important factor in performances is their hyperparameters. As mentioned above, we use Grid search CV. This method is a faster way to find the optimal values for a given model instead of running a loop for every possible combination of each hyperparameter of each model selected. So it's more efficient than manual looping, as it can leverage parallel processing and optimize it.

For didactical reasons, we decided to evaluate only a few hyperparameters for each ML model:

* For logistic regression we used parameters\_logreg = {'C': [0.001, 0.01, 0.1, 1, 10, 100]} and we obtained Best Parameter: {'C': 0.1}.
* For random forest we used parameters\_rf = {'n\_estimators': [50, 100, 150, 200],'max\_depth': [None, 10, 20, 30], 'min\_samples\_split': [2, 5, 10], 'min\_samples\_leaf': [1, 2, 4] and we obtained Best Parameters: {'max\_depth': 20, 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 200}.
* For support vector machine we used parameters\_svm = {'C': [0.1, 1, 10], 'gamma': [0.01, 0.1, 1], 'kernel': ['linear', 'rbf']} and we obtained Best Parameters: {'C': 10, 'gamma': 0.1, 'kernel': 'rbf'}

Based on our report, it looked like that the Random Forest model performed better than the Logistic Regression and SVM models. His confusion matrix result is:

[[1384, 21]

[76, 208]]

A blue squares with white text

Description automatically generated

Also, his confusion matrix looks something balanced even though there are 76 false negatives. In the last chapter confusion matrix will be explained more in detail.

It is important to note that gridsearchCV is a good way to evaluate a ML model because of the accuracy, f1 score, etc. This function helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, we can select the best parameters from the listed hyperparameters. (…) GridSearchCV tries all the combinations of the values passed in the dictionary and evaluates the model for each combination using the Cross-Validation method. Hence after using this function we get accuracy/loss for every combination of hyperparameters and we can choose the one with the best performance. Team, G.L. (2024).

Grid search CV gives us various metrics, but we are not obligated to use all of them. Accuracy, a straightforward measure, calculates the percentage of correctly classified instances. The confusion matrix, however, gives us information in a deeper detail. It offers a detailed breakdown of the model's performance across different categories. This breakdown includes true positives (correctly identified positives), true negatives (correctly identified negatives), false positives (incorrectly classified positives), and false negatives (incorrectly classified negatives).

We also did the ROC curve, and the Random Forest ROC AUC was the best one.

1. Logistic Regression ROC AUC: 0.8430
2. Random Forest ROC AUC: 0.9715
3. Support Vector Machine ROC AUC: 0.8519

A graph of a logistic curve

Description automatically generated

AUC-ROC Curve , or receiver operating characteristic curve, is a graph which is able to show up how well the model performs. It is an evaluation metric for binary classification problems like our project. Basically it plots a probability curve about the sensitivity, or TPR against FPR, or ().

*TPR*, or or Recall is the formulation of:

*TNR* or is the formulation of:

And finally, *FPR* is the formulation of:

The Area Under the Curve (AUC) is the measure of the ability of a binary classifier to distinguish between classes and is used as a summary of the ROC curve so the higher the AUC, the better the model’s performance at distinguishing between the positive and negative classes. When AUC = 1, the classifier can correctly distinguish between all the Positive and the Negative class points. If, however, the AUC had been 0, then the classifier would predict all Negatives as Positives and all Positives as Negatives. When 0.5<AUC<1, there is a high chance that the classifier will be able to distinguish the positive class values from the negative ones. This is so because the classifier is able to detect more numbers of True positives and True negatives than False negatives and False positives. When AUC=0.5, then the classifier is not able to distinguish between Positive and Negative class points. Meaning that the classifier either predicts a random class or a constant class for all the data points. (Bhandari, 2024).

Before talking about the results, we proceed to oversample the target variable and compare the results we got before and after the oversampling method.

9. Oversampling

We Trained initial LR, RF and SVM 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.

A blue and orange rectangular bars

Description automatically generated

Imbalanced data, where one class (the majority) significantly outnumbers another (the minority), is a frequent challenge in machine learning tasks like classification. This imbalance can lead to biased models that favor the majority class and perform poorly on the minority class. There are different methods to tackle this problem: one of them is to random oversampling the minority class, picking up a small random quantity of its data and duplicate it. We repeat the process iteratively until we reach the same amount of data of the majority class. This is how Random Oversampling works. Another method, which is the method we use in this project is SMOTE.

SMOTE (Synthetic Minority Over-sampling Technique) is a popular approach to address this issue by oversampling the minority class, creating a more balanced dataset for training. SMOTE works by creating synthetic minority class samples. Al Tobi, A. M. H. (2020).

The first step is to identify the minority class that is the [1] which is churned customers.

A diagram of red and blue dots

Description automatically generated

Riccardo homework

To explain how smote works we need to plot the minority class.

A graph with red dots

Description automatically generated

Riccardo homework

SMOTE creates synthetic samples rather than duplicating existing ones and then SMOTE selects one or more of its nearest neighbours and generates synthetic examples. To do that, we need to create a vector from a random point to its nearest neighbour. Once we got the vector we choose a random point between 0 and 1 to scale the synthetic point’s vector and it will give us as many synthetic points as we want.

A graph with red dots and green arrows

Description automatically generated

Riccardo homework

Once the examples are generated, SMOTE selects the nearest neighbours of a minority class instance using distance metrics such as Euclidean distance.

A diagram of a constellation

Description automatically generated with medium confidence

Riccardo homework

We evaluate the best model on the test set using the best estimator found during hyperparameter tuning after the oversampling:

* Logistic Regression:

Best Parameter: {'C': 0.01}

* Random Forest Classifier:

Best Parameters: {'max\_depth': 30, 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 200}

* Support Vector Machine (SVM):

Best Parameters: {'C': 10, 'gamma': 1, 'kernel': 'rbf'}

The best parameters have not changed comparing them before and after the oversampling technique. Also, the best ML model is still Random Forest Classifier, and it improved its values:

Before oversampling:

*Random Forest ROC AUC: 0.9715*

After oversampling:

*Random Forest ROC AUC: 0.9730*

The Random Forest model can be considered the best model for this classification task and after the oversample technique. Also, the confusion matrix of the RF model looks very good and performs well as the FN and TN are very small comparing to the TP:

conf\_matrix = [[1368, 37], [57 , 227]]

A screenshot of a graph

Description automatically generated

Confusion matrix has been made for the first time by the British statistician Karl Pearson. It is with no doubt one of the best way to evaluate the model in Machine Learning. Let’s suppose that we have 50 fishes, 47 are yellow and 3 are red. The model we run has got the 96 % of accuracy: , but it is a overfitted results because the model is saying us that almost the fishes are yellow but we should focus on identifying the red fish, which is the positive class. Here is where we introduced the confusion matrix. The columns correspond to what the model has predicted, and the rows correspond to the true labels. Since it is a classification problem where we have 0 and 1 as possible output, where 0 is the negative class and 1 the positive class. The top left corner are all the values that contain true negative (TN), where the no churn customers were correctly identified by the model. The true positive (TP) are in the bottom right corner and it shows us the correctly identified churned customers by the model. So now we have the bottom left corner where we have the False negative (FN). False negative are the churned customers but predicted to be no churn customers. Lastly, the top right corner are the False positive (FP). False Positive are the no churn customers but predicted to be churned customers.

In summary, confusion matrix shows us up where the model did well and where it did mistakes. It is very useful when the two classes are very imbalanced and we are more interested in positive class, in this case churned customers.

10. Conclusions

Churn is an important study to work with because it can be very beneficial for all the companies who want to try to save money and getting better revenue. This study has been made using the E commerce dataset. It was not very hard to work with and we did not implement any feature engineering. Once we got the understanding of the dataset to make sense of the data, we created visual representations.

Since the first moment we realized we were working with a strong imbalanced target variable (more than 1:5), but we still decided to work with it as imbalanced to do the Oversampling only after that and rerun the models. Subsequently, we plot all the categorical features, we compared our target variable with other different features, and we also plot some statistical techniques such as relative frequency to getting a better understanding of the percentage of each value presented in the features studied. Additionally, we observed that the location of customers, categorized by city tiers, had an impact on how many orders they placed. This meant that customers from different cities tend to order different amounts and it is a very important strategic information due to the revenue of the company.

We also applied techniques such as Label encoder to transform all the categorical features into numerical representations and we plot the correlation matrix to see the relationship between churn and all the other data. In the data preprocessing step, we discussed the point-biserial correlation, which played a crucial role in determining the features we selected before implementing the models. We then used cross validation and GridSearchCV techniques to use the best parameters for each model.

Regarding the models we used, we split the data into 30% of testing and 70% of training. We wanted to see how well our models predicted customer churn. We did not use all the combinations of all the possible parameters for each model because it has been an academical work and we wanted to show our knowledge and not spending too much time running the best model. Despite that, we tried lot of different combinations, and we got the best one.

According to the correlation matrix and the ROC curve, Random Forest Classifier model has been the model which reached the highest score to predict future churn. But even though the model reached good results, we were worried about the possibility of being an overfitted model where the model is telling us that almost the customers are not churn but we should focus on identifying the churned customers. To do that we run a technique called SMOTE to balance them and we created as much synthetic values as we needed to balance the target variable.

After balancing the data, we got better results for the model we wanted to apply. Confusion matrix showed that it almost doesn’t do many errors so we can consider it as a good model.

A graph with a bar

Description automatically generated

We finally run the feature importance plot after oversampling based on their importance score. The plot indicates that tenure, cashback amount, day since last order, satisfaction score, complain, preferred order category, number of devices registered, and marital status are the most important features for predicting churn after oversampling in this order.

A screenshot of a computer screen

Description automatically generated

Now, regarding the SHAP (Shapley Additive Explanations) summary plot, it provides insights into how each feature contributes to individual predictions. The SHAP summary plot ranks features based on the magnitude of their impact on model predictions across all instances in the test set.

SHAP (SHapley Additive exPlanations) values are a way to explain the output of any machine learning model. It uses a game theoretic approach that measures each player's contribution to the final outcome. In machine learning, each feature is assigned an importance value representing its contribution to the model's output. SHAP values are a common way of getting a consistent and objective explanation of how each feature impacts the model's prediction. SHAP values are based on game theory and assign an importance value to each feature in a model. Features with positive SHAP values positively impact the prediction, while those with negative values have a negative impact. The magnitude is a measure of how strong the effect is. Awan, A.A. (2023).

The SHAP summary plot highlighted complain and marital status as significant features, it suggests that these features contribute significantly to the model's predictions. However, we must keep in mind that SHAP values provide a per-instance explanation and so, they can vary for each data point.

Said that, we could consider we have got good results. Considering that churn prediction is one of the most common machine learning practise in everyday businesses, we applied all the knowledge we have and we got our conclusions and we have defined how we could handle the problem of churn for the e-commerce company and try to predict the techniques to implement in the future to raise their revenue.

11. References

* Seaborn.violinplot seaborn.violinplot - seaborn 0.13.2 documentation. Available at: https://seaborn.pydata.org/generated/seaborn.violinplot.html
* Awan, A.A. (2023) An introduction to shap values and machine learning interpretability, DataCamp. Available at: https://www.datacamp.com/tutorial/introduction-to-shap-values-machine-learning-interpretability
* Python - P-value (no date) Tutorialspoint. Available at: <https://www.tutorialspoint.com/python_data_science/python_p_value.htm>
* Sklearn.model\_selection.GRIDSEARCHCV (no date) scikit. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.GridSearchCV.html
* Team, G.L. (2024) Hyperparameter tuning with GRIDSEARCHCV, Great Learning Blog: Free Resources what Matters to shape your Career! Available at: https://www.mygreatlearning.com/blog/gridsearchcv/
* 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
* bhartiprasad17 (2021) Customer churn prediction, Kaggle. Available at: <https://www.kaggle.com/code/bhartiprasad17/customer-churn-prediction>
* Sklearn.metrics.confusion\_matrix (no date) scikit. Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html>
* cross-validation: Evaluating estimator performance (no date) scikit. Available at: https://scikit-learn.org/stable/modules/cross\_validation.html
* 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/>
* Bhandari, A. (2024) Guide to AUC ROC curve in machine learning : What is specificity?, Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/
* Al Tobi, A. M. H. (Creator) & Duncan, I. M. M. (Supervisor), University of St Andrews, 2020 Available at: https://research-portal.st-andrews.ac.uk/en/datasets/sta2018-full-thesis-data
* K&#305;l&#305;&ccedil;, &#304;lyurek (2023b) ROC curve and AUC: Evaluating model performance, Medium. Available at: <https://medium.com/@ilyurek/roc-curve-and-auc-evaluating-model-performance-c2178008b02>
* T., B. (2023) A complete shap tutorial: How to explain any black-box ML model in Python, Medium. Available at: <https://medium.com/towards-data-science/a-complete-shap-tutorial-how-to-explain-any-black-box-ml-model-in-python-7538d11fae94>
* Gusarova, M. (2023) Explain any machine learning model in Python, shap, Medium. Available at: <https://medium.com/@data.science.enthusiast/explain-any-machine-learning-model-in-python-shap-4caf1c343f3>
* Nguyen, H.M., Cooper, E.W. and Kamei, K. (2011) ‘Borderline over-sampling for Imbalanced Data Classification’, International Journal of Knowledge Engineering and Soft Data Paradigms, 3(1), p. 4. doi:10.1504/ijkesdp.2011.039875.
* (No date) Chatgpt. Available at: <https://chat.openai.com/>
* aig3rim (no date) Interpret\_random\_forest\_classifier\_using\_SHAP/RANDOM\_FOREST\_CLASSIFIER\_SHAP.ipynb at master · aig3rim/interpret\_random\_forest\_classifier\_using\_shap, GitHub. Available at: https://github.com/aig3rim/Interpret\_random\_forest\_classifier\_using\_SHAP/blob/master/Random\_forest\_classifier\_SHAP.ipynb
* Class 26: Biserial and Point Biserial correlation || NET preparation || dr justine’s classroom (2020) YouTube. Available at: <https://www.youtube.com/watch?v=BuJiv4-Kcu4>

12. GitHub repo link

https://github.com/CCT-Dublin/capstone-project-riccardopossier