**CCT College Dublin**

**Assessment Cover Page**

*To be provided separately as a word doc for students to include with every submission.*

|  |  |
| --- | --- |
| **Module Title:** | Machine Learning |
| **Assessment Title:** | CA1 Machine Learning 2023 |
| **Lecturer Name:** | Muhammed Iqbal |
| **Student Full Name:** | Ariel Goldman and Daniel Murphy |
| **Student Number:** | Ariel Goldman - sbs23073 Daniel Murphy – sbs23106 |
| **Assessment Due Date:** | 30/04/2023 |
| **Date of Submission:** | 30/04/2023 |

**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. |

Contents

[Business Understanding 1](#_Toc133763331)

[Data Understanding (Preliminary EDA) 1](#_Toc133763332)

[Data Preparation for EDA 2](#_Toc133763333)

[Exploratory Data Analysis 2](#_Toc133763334)

[Contact with Bank X Months Inactive 3](#_Toc133763335)

[PDF & Box Plot Analysis 4](#_Toc133763336)

[Positive Correlations 4](#_Toc133763337)

[Negative Correlations 4](#_Toc133763338)

[EDA Conclusion 5](#_Toc133763339)

[Machine Learning Models 6](#_Toc133763340)

[Data Preparation for Models: 6](#_Toc133763341)

[Accessing Models with Cross-Validation 6](#_Toc133763342)

[Hyperparameters 7](#_Toc133763343)

[Model Selection 8](#_Toc133763344)

[Support Vector Machine 8](#_Toc133763345)

[Accuracy, Precision, and Recall based on SVM on Imbalanced Data 10](#_Toc133763346)

[Interpretation & Explanation of Balanced & Imbalanced Dataset 11](#_Toc133763347)

[Grid Search to Find Optimal Hyperparameters (KERNEL='RBF') 12](#_Toc133763348)

[Grid Search to Find Optimal Hyperparameters (KERNEL='Linear') 13](#_Toc133763349)

[Linear Kernel Final Model Results 14](#_Toc133763350)

[Final Model Performance Interpretation & Evaluation of RBF & Linear Kernels. 14](#_Toc133763351)

[Random Forest 15](#_Toc133763352)

[Accuracy, score, and recall based on RF on balanced data. 15](#_Toc133763353)

[Performance of Random Forest 16](#_Toc133763354)

[Hyperparameter Tuning 16](#_Toc133763355)

[Grid Search to Find Optimal Hyperparameters 17](#_Toc133763356)

[Classification Report 17](#_Toc133763357)

[Principal Component Analysis 18](#_Toc133763358)

[Contribution of each team member in the project using a Pie Chart 19](#_Toc133763359)

[Individual Contributions of Ariel Goldman 20](#_Toc133763360)

[Individual Contributions of Daniel M. Murphy 23](#_Toc133763361)

[References 25](#_Toc133763362)

There are **2990 words** in this report excluding this sentence, headings, references, etc.

# Business Understanding

The overarching motivation of this project is to identify which factor is most significant in contributing to customer attrition. By predicting this major problem for business, stakeholders can make data driven decisions in order to lower customer attrition.

<https://www.kaggle.com/datasets/thedevastator/predicting-credit-card-customer-attrition-with-m>

Customer churn is a serious problem for businesses. Retaining customers is much more cost-effective than acquiring new customers. This is true in the credit card industry. Research by Reichheld and Sasser (1990) shows that as the defection rate drops by 5%, the average lifespan of a customer doubles, thus raising profits by 75% (appendix 1).

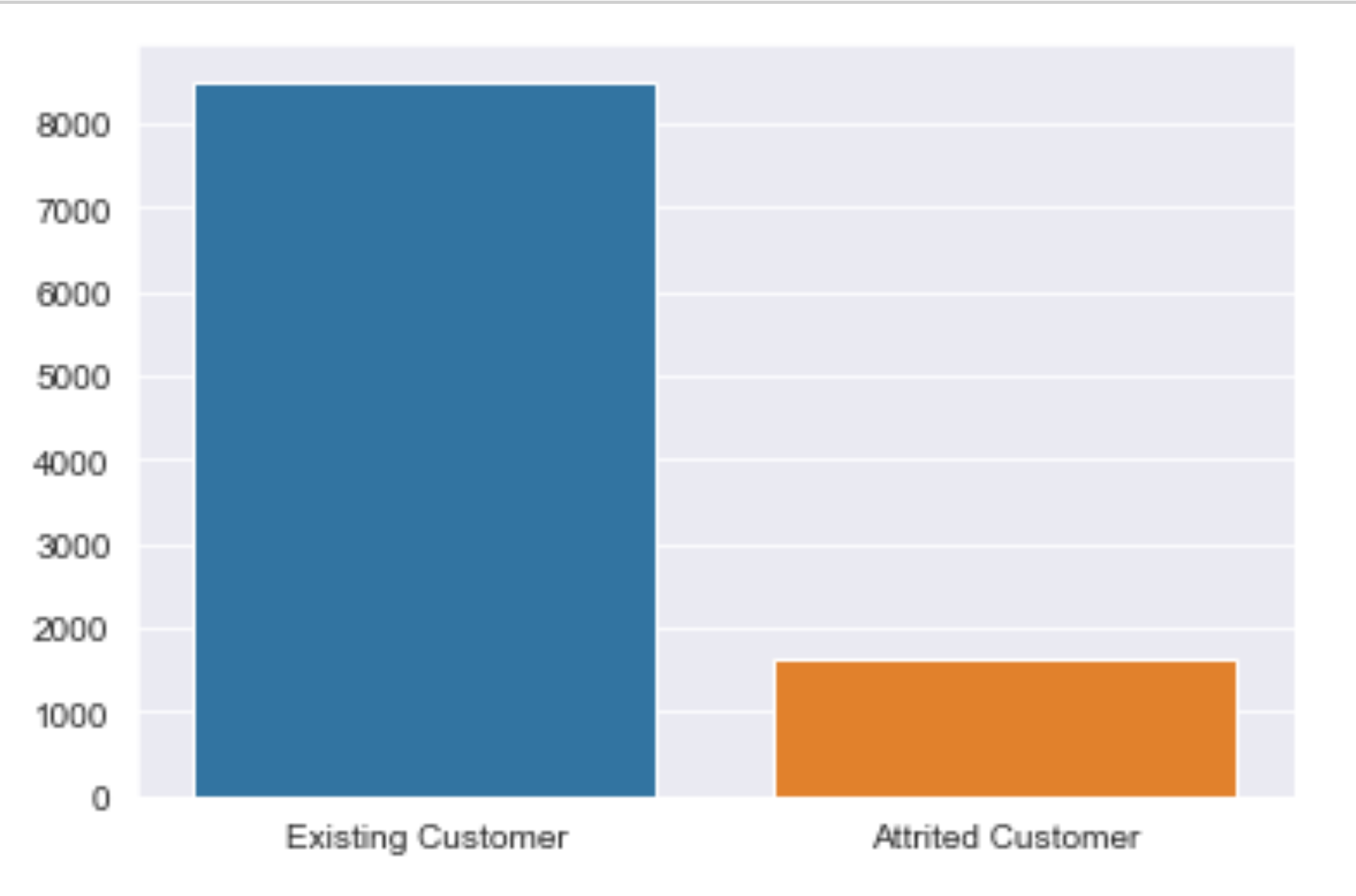
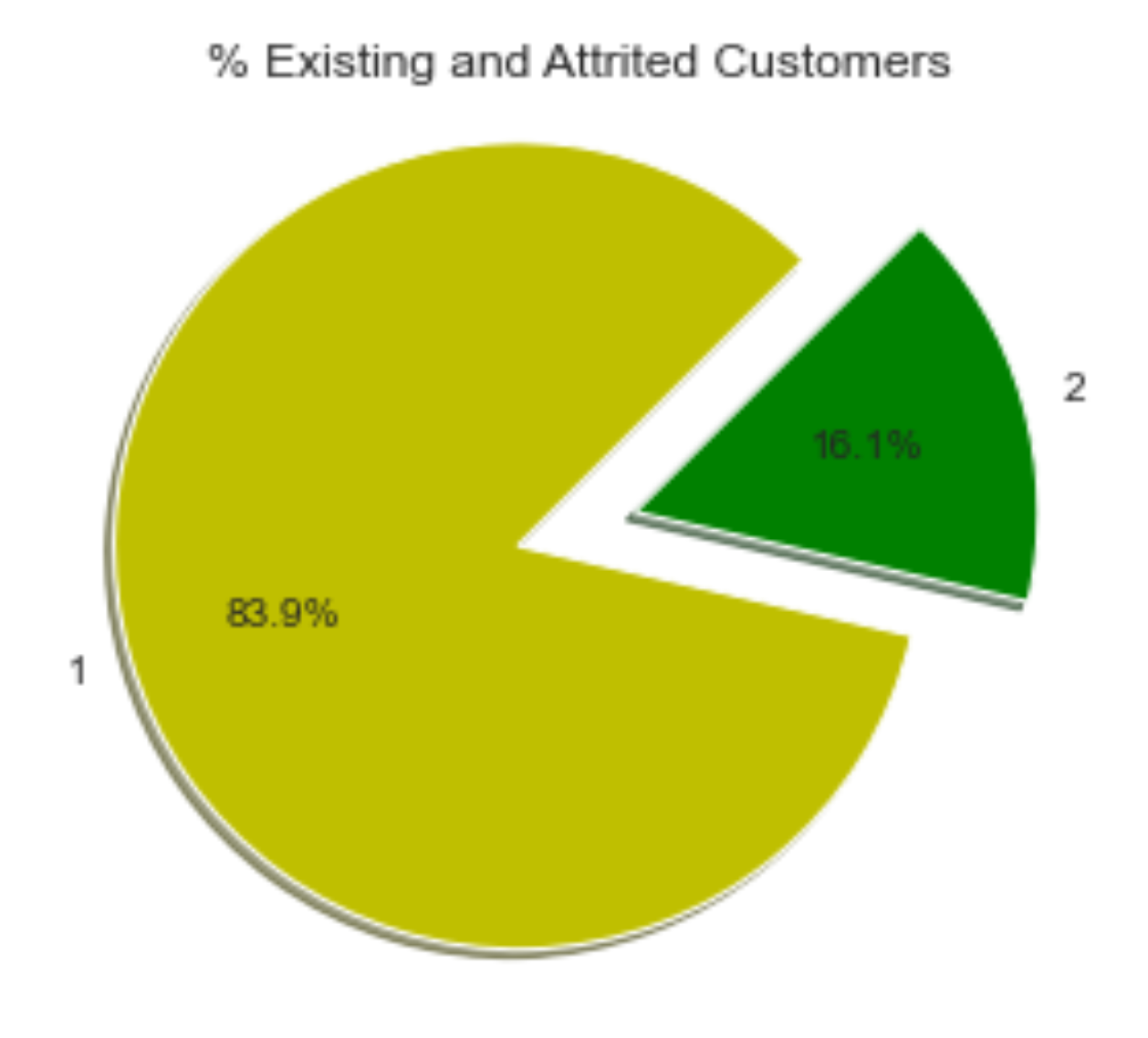
Our project goals are justified by:

* Feature Importance: By encoding categorical data, using a correlation matrix and perform EDA to identify the most significant feature for predicting attrition.
* Prediction & Classification: We implement four ML models (SVM, Random Forest, Logistic Regression, & Naive Bayes) in order to determine the probability of customer churn.
* Dimensionality Reduction: We applied PCA in order to reduce dimensionality and easily identify the strongest features for predicting churn.

# Data Understanding (Preliminary EDA)

Using basic functions & graphs we first got a glimpse of the data, before ensuring it was clean and ready for analysis.

Once clean, we looked at our target variable (‘Attrition\_Flag), and its relationship with other features.

****

As can be seen above, there is an imbalance that we will need to address.

# Data Preparation for EDA

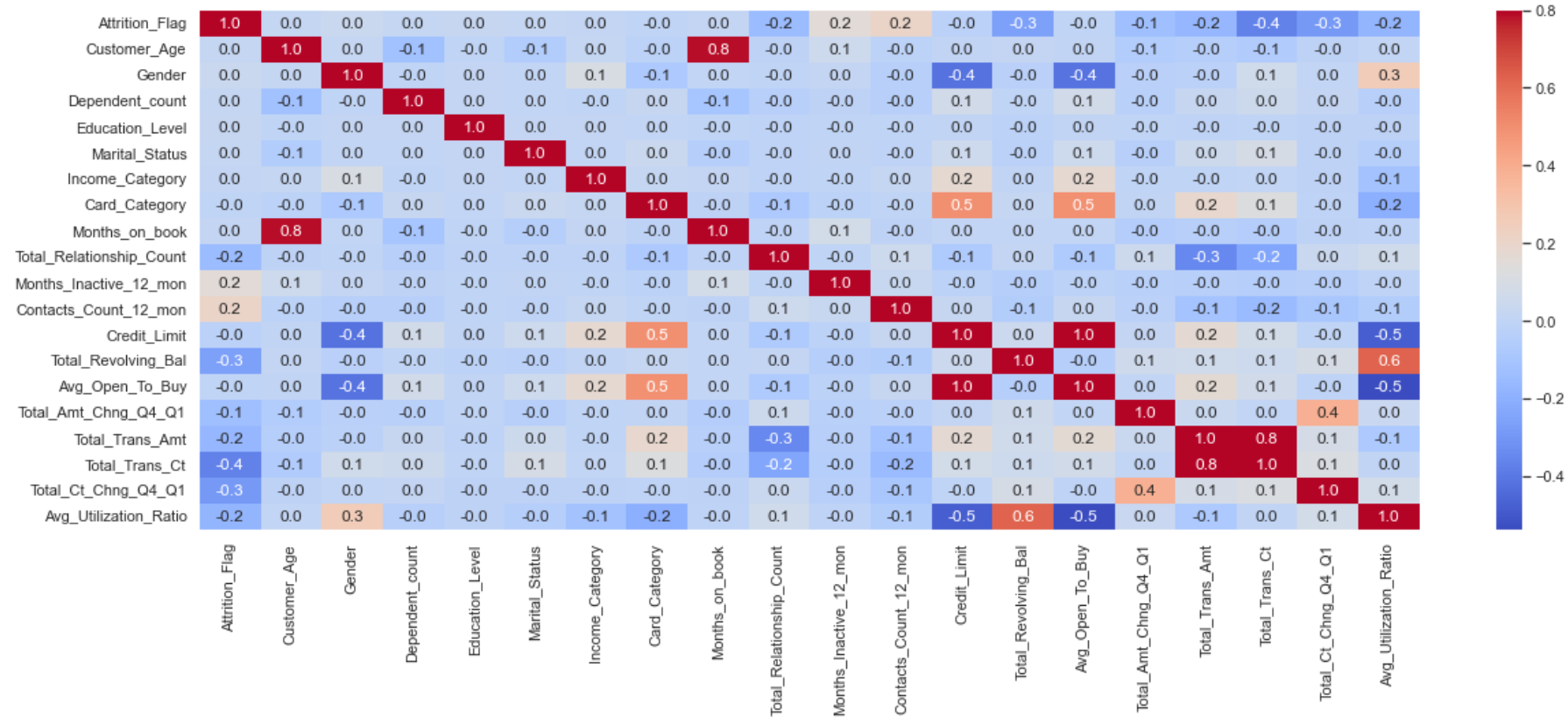
We created two new subsets (existing customers in one, churned customers in another) to create comparisons.



In addition, we encoded the data (using label encoding for its simplicity of implementation & understanding). This allowed us to analyse the object data easily.

# Exploratory Data Analysis

By encoding the data, we were able to produce the following correlation matrix.

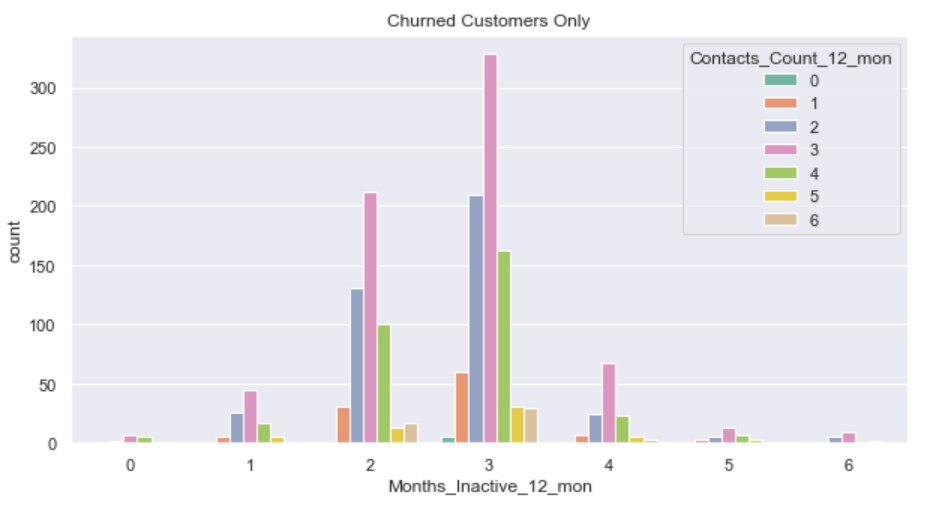


The above matrix shows that the months inactive & contact count features are the only positively correlated columns to the target variable.

The matrix also has features that have a negative correlation with attrition.

These are all features of interest that we decided to explore more.

## Contact with Bank X Months Inactive



By comparing the distribution of churned customers (CCs) above, and existing customers (EXCs) below, we can gain some insights.

Both normally distributed, CCs tend to have periods of longer inactivity - majority of cases at three, compared to EXCs being spread. Suggesting that prolonged inactivity could be a relatively good indicator of customer attrition.

CCs contact count peaks at 3 (2 for existing customers). This along with contact count being positively correlated to the attrition flag suggests contact strategy could be improved.

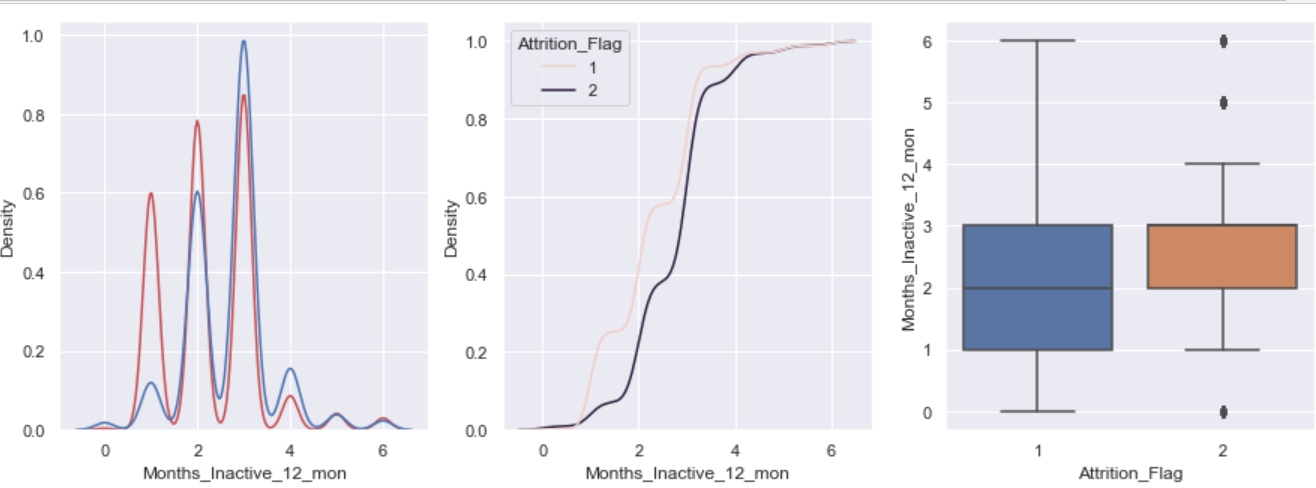


## PDF & Box Plot Analysis

### Positive Correlations

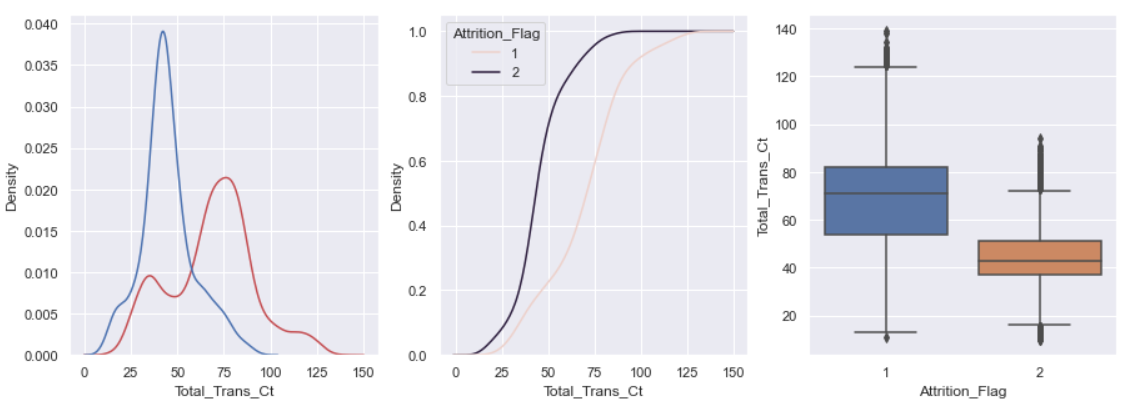
Shetty’s (2021) helped me investigate the features below.

The Month\_Inactive PDF, below (left), shows the CCs line (blue) peaks higher than the EXCs line (red) at x-axis = 3. This suggests a strong correlation between Months\_Inactive and churn. The attrition class also has a smaller box plot, suggesting the data is more clustered. Despite this, the peaks of existing & CCs at the same values indicates this feature alone isn’t a strong enough indicator of churn.



The PDF graph for the Contact\_Count\_12 have a similar problem with both lines peaking at similar values, however, as the EXCs line remains higher, it suggests the correlation isn’t as strong.

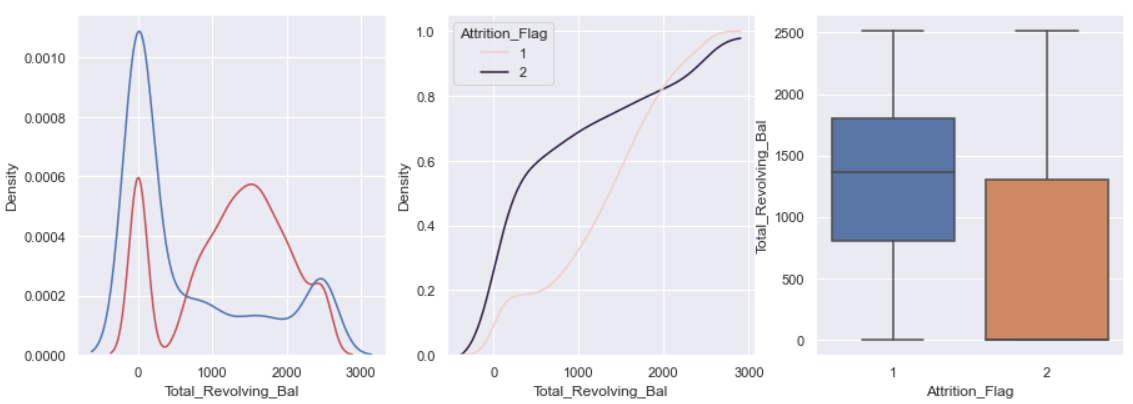
### Negative Correlations



Total transaction count PDF has different density peaks, suggesting this is a good feature for differentiating EXCs, and CCs. CCs have a lower count, suggesting that customers with lower transaction counts are more likely to leave the bank.

The attrition density suggests a more concentrated group of data for churn cases (around 35/40 value). This is supported by the small box plot box.

A quick look at the PDF & box plot of total\_ct\_chng tells us that it is not a good feature for predicting churn. The PDF peaks are in a similar value range, and the box plots are also too similar to differentiate between.



Lastly, the total revolving balance above. The EXC’s twin peaks in the PDF suggest a binomial distribution. Graph implies a lower revolving balance is more likely to result in churn. However, box plot shows big spread in both classes.

# EDA Conclusion

As a single feature, ‘Total\_Trans\_Ct,’ is the best predictor of attrition for the reasons mentioned above. Despite that, it alone might not be the best predictor of attrition.

Given more time, feature engineering expertise, & better domain knowledge, we could create new features better at predicting attrition. The strongest correlated features to the ‘Attrition\_Flag’- the most negatively correlated features - could be combined through feature interaction in order to create a new feature. This new feature should be better at predicting customer churn (Krishnamurthy, 2022).

# 

# Machine Learning Models

Step one: Split the data. Three splits. Allows us to find best for predicting churn, and avoid overfitting (Joseph and Vakayil, 2021). Xu and Goodacre (2018) point out, data splitting when done correctly can improve the **generalisation** performance of a model - meaning it will perform better on unseen data (important in how the model performs in the real-world).

# Data Preparation for Models:

SPLIT, SCALE & SMOTE

Next, we normalised the data & addressed the imbalance. We applied the MinMaxScaler as our dataset has no negative values and is non-distorting (Hale, 2019). We scaled the X\_train & X\_test separately. This is important in order to prevent data leakage (Brownlee, 2016).This is the reason for order of operation (split then scale). SMOTE used to address the imbalance. Only applied SMOTE to the training dataset in order to prevent overfitting.

# Accessing Models with Cross-Validation

Cross-Validation

Cross-validation assesses the skill of a model by breaking the dataset into smaller subsets or “folds.” Used due to being easy to implement & understand, as well as generally resulting in lower bias or less optimistic model performance (Brownlee, 2018).

K-Fold Cross-Validation

This method divides the dataset into equal sized folds. While one fold becomes the validation set, the remaining ‘K-1’ folds are used as the training set. Model is trained & then evaluating the performance on the validation set (Brownlee, 2016). We reduced the number of folds from 20 (cv=20), to 5, due to the notebook getting hung on that particular cell - this affects accuracy.

Varying Training/Test Splits

By changing data split, we can analyse the impact the training set size has on model performance. This helps us understand the trade-off between bias and variance. A smaller dataset may lead to higher bias or **underfitting** (when a machine model fails to capture the patterns in a dataset, resulting in poor performance). Large training set may reduce bias but increase variance or **overfitting** (when the model learns the dataset’s patterns too well and doesn’t generalise to new, unseen data)(Bento, 2020).

Comparing Accuracies, Precision & Recall

We can analyse the model’s performance by comparing accuracies, precision & recall at different splits. As the training size increases, so too, does the accuracy. However, as mentioned above, this can lead to overfitting and thus reduce the model's generalisation.

Confusion Matrix

Helps us visualise the performance of the model. We can compare how each model split has performed in relation to the others using the confusion matrix and comparing false positives, etc.

Cross-Validation Conclusion

Using the techniques above gives us insights into the model's performance. By comparing the results of different splits, we can then assess the most appropriate training/test split for our needs.

# Hyperparameters

Hyperparameter tuning can be performed after k-fold cross validation. By identifying and altering the hyperparameters, we can optimise the model. Comparing the model’s performance before and after hyperparameter tuning, we can analyse the model’s generalisation ability.

## Model Selection

Focused on two models: Support Vector Machine (SVM) and Random Forest. Chosen due to being strong in handling binary classification problems like predicting customer churn. we encoded the data, as well as performed MinMax scaling (which is particularly good for SVM)(Lanenok, 2015), and performed SMOTE as mentioned.

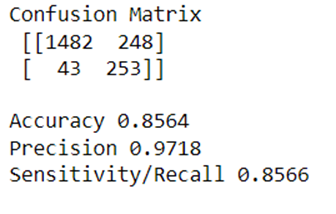
We explored Naive Bayes & Linear Regression also, but to adhere to the word count we omitted our analysis. Code still presents in Jupyter notebook.

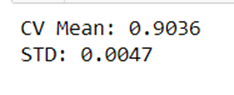
# Support Vector Machine

We will first display the results of the model in relation to the balanced and unbalanced dataset. Then we will interpret and explain the results.

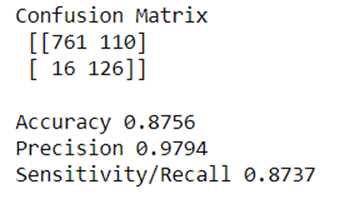
Accuracy, Precision, and Recall based on SVM on Balanced Data

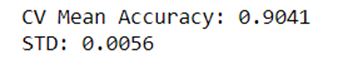
**80/20 Split - random\_state=102**



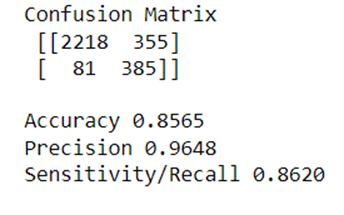


**90/10 Split - random\_state=102**





**70/30 Split - random\_state=109**





**Use of Cross Validation - Performance on the entire dataset**

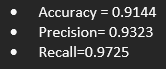
## Accuracy, Precision, and Recall based on SVM on Imbalanced Data

**Confusion Matrix on Imbalanced dataset - test set (20%).**

There are 2026 customers in our test set (20%).

Chart, treemap chart

Description automatically generated

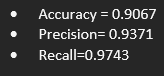


**Confusion Matrix on Imbalanced dataset - test set (10%).**

There are 1013 customers in our test set (10%).

Chart, treemap chart

Description automatically generated

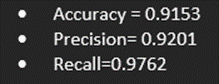


**Confusion Matrix on Imbalanced dataset - test set (30%).**

There are 3039 customers in our test set (30%).

Chart, treemap chart

Description automatically generated



## Interpretation & Explanation of Balanced & Imbalanced Dataset

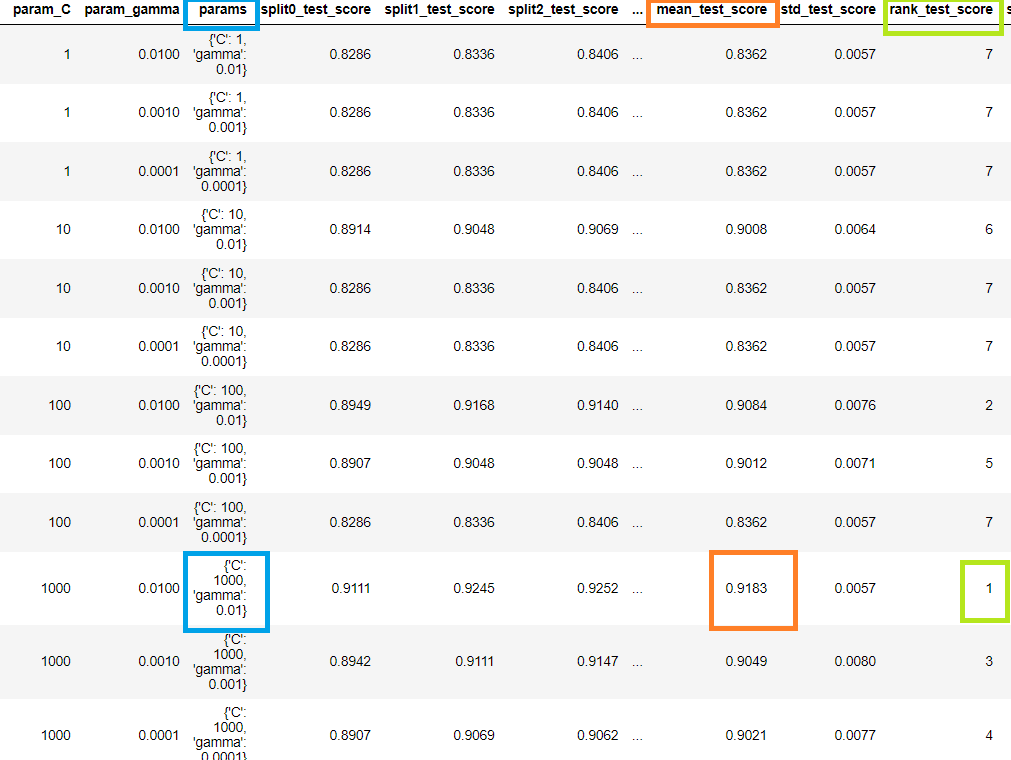
Comparing the balanced & imbalanced datasets, we see the accuracy is higher for the imbalanced. This does not mean that the model on the imbalanced dataset performs better at identifying churn. The high accuracy can refer to the model’s ability to correctly predict the majority (existing customer). This is supported by the recall being above 97% in each imbalance split.

The balanced dataset has a lower accuracy but higher precision. This suggests that there are less false positives. We can presume the model is more selective in predicting churn, and less likely to misclassify non-churn customers as churned.

The imbalanced dataset accuracy is higher, but considering the precision & recall, we begin to understand, the balanced dataset model has a higher precision and can be seen to perform better at predicting churn, as well as, returning lower false positives.

## 

# Grid Search to Find Optimal Hyperparameters (KERNEL='RBF')

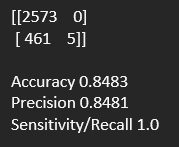


The best test score is 0.9183 corresponding to hyperparameters {'C': 1000, 'gamma': 0.01}

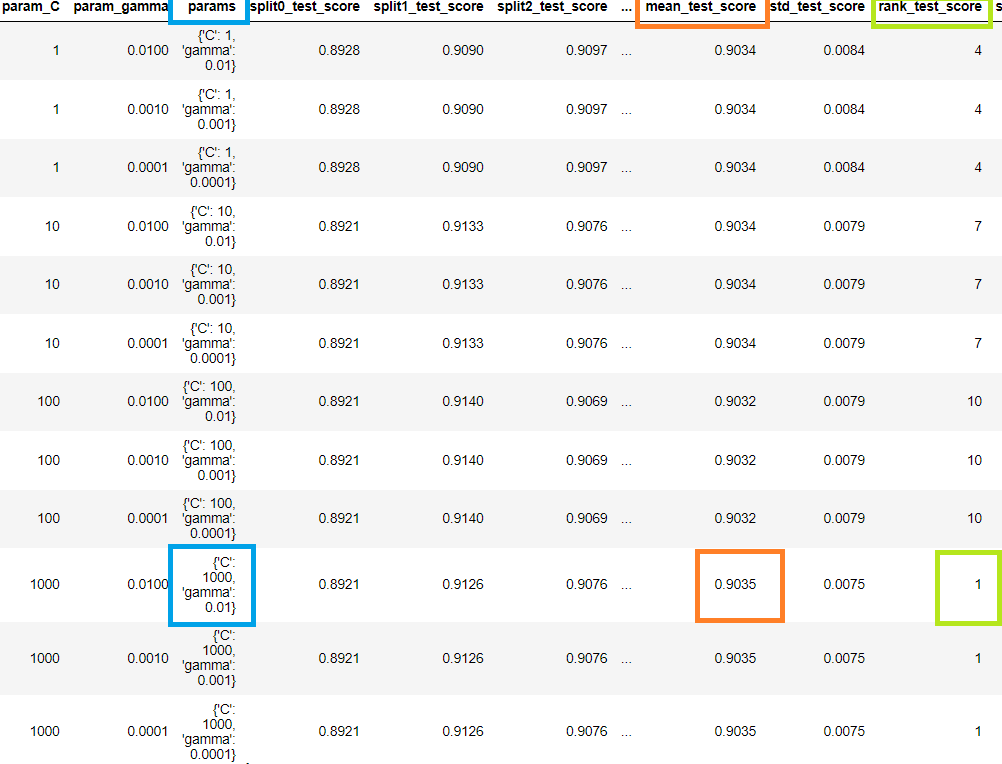
Chart, line chart

Description automatically generated

RBF Kernel Final Model Results



## Grid Search to Find Optimal Hyperparameters (KERNEL='Linear')



The best test score is 0.9035 corresponding to hyperparameters {'C': 1000, 'gamma': 0.01}

Chart

Description automatically generated

## 

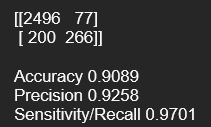
## 

## 

## 

## 

## Linear Kernel Final Model Results



## Final Model Performance Interpretation & Evaluation of RBF & Linear Kernels.

The RBF Kernel has an accuracy score of 84.83% & a perfect recall of 100%. This means that it identified all the true negatives (no churn) correctly. However, its lower precision of 84.81% suggests that it is not as good at identifying the true positive cases (churn). In fact, the confusion matrix shows that it only correctly predicted 5 churn cases and misclassified 461.

Conversely, the Linear Kernel model has a higher overall accuracy of 90.89%. The recall is slightly lower, meaning it identifies 97.01% of the true negatives (no churn) correctly. More importantly, it has a higher precision rate (92.58%). This means that it’s better at identifying churn (true positives). This is shown in the confusion matrix, where 266 out of a possible 466 cases were correctly identified.

The above demonstrates that the Linear Kernel SVM performs better for predicting customer churn. This is shown clearly through the confusion matrix and the better balance between precision and recall.

# Random Forest

## Accuracy, score, and recall based on RF on balanced data.

80/20 Split - random\_state = 102

Classification Report

Table

Description automatically generated

90/10 Split - random\_state = 102

Classification Report

Table

Description automatically generated with medium confidence

70/30 Split - random\_state = 109

Classification Report

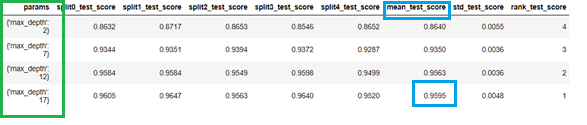
Table

Description automatically generated

## Performance of Random Forest

The random Forest model is performing quite well across all three data splits. The highest accuracy achieved was 99% in the case of 70% training and 30% testing split, and 100% accurate in class 1(Existing Customers). In general, this model performs well on the given binary classification problem.

## Hyperparameter Tuning



# Plotting accuracies with max\_depth

Chart, line chart

Description automatically generated

### Grid Search to Find Optimal Hyperparameters

* We can now find the optimal hyperparameters using GridSearchCV.

We can get accuracy of 0.9001 using {'max\_depth': 10, 'max\_features': 5, 'min\_samples\_leaf': 100, 'min\_samples\_split': 200, 'n\_estimators': 100}

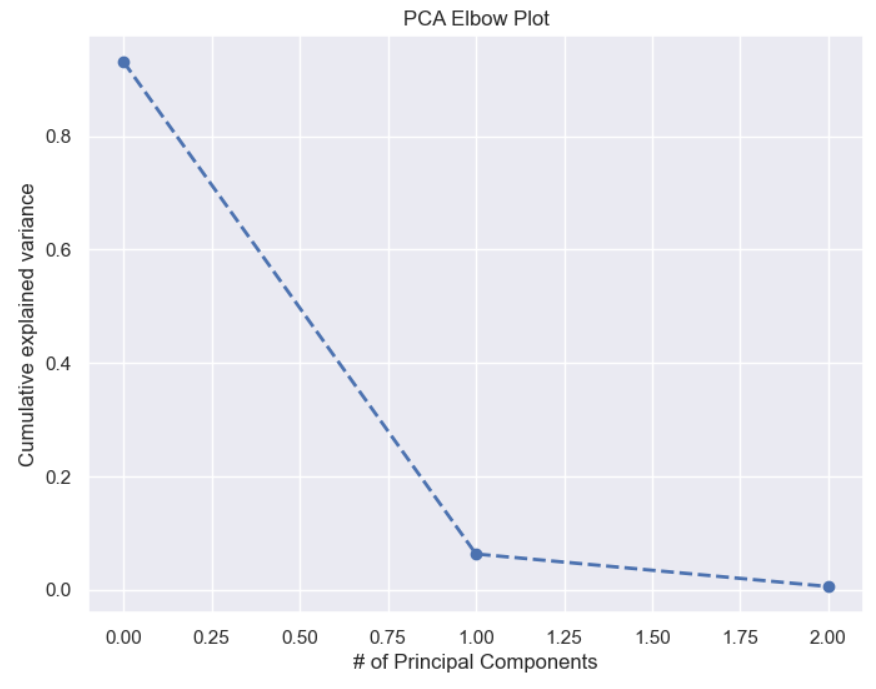
## Classification Report

By using optimal hyperparameters through the GridSearchCV, the accuracy achieved was .9001, which is relatively high. Additionally, the precision and recall for class 1 (Existing Customers) were both high, indicating that the model is able to correctly identify positive cases with high accuracy. However, the recall for class 2 (Attrited Customers) was lower than the recall for class 1, indicating that the model may have more difficulty correctly identifying negative cases.

Table

Description automatically generated

# Principal Component Analysis



Based on the above plot, it can be observed that the PCA describes a significant amount of variance in the data, while the amount of variance explained by each successive PCA decreases rapidly. In this scenario, it seems that with 3 Principal Components, a substantial portion of the variance present in the data can be accounted for, as the cumulative explained variance at around 0.95. This means that we can retain 95% of the variance in the data with 3 principal components while reducing the dimensionality of the dataset.

# Contribution of each team member in the project using a Pie Chart

Chart, pie chart

Description automatically generated

# Individual Contributions of Ariel Goldman

**My role in the team**

My responsibility, as part of this team, was to apply Machine Learning models where we have to predict whether a customer will churn the bank or not according to their banking data. The main objective of this project was to create and develop at least two Machine Learning models, however, we have decided to implement four models in order to obtain more outcomes from them and compare them to the main ones we have used. Therefore, we will be able to make predictions whether a customer would stop doing business with the bank and take actions to retain them.

**Approach**

After an exhaustive implementation of the Exploratory Data Analysis, I proceeded to train and test various Machine Learning models using Scikit-learn Libraries in Jupyter Notebook. The main models applied were Support Vector Machine and Random Forest. I experimented with different hyperparameters for Support Vector Machine using RBF Kernel and Linear Kernel to find the Optimal C and Gamma parameters and for Random Forest the Hyperparameter Tuning to find the optimal ‘n’ estimators and the optimal accuracy score.

Throughout the process of applying Machine Learning models, I came across different challenges that required a deep analysis. One of them was to encode categorical variables into numerical variables because otherwise, Machine Learning models would not work with categorical variables as the models typically require numerical input, so leaving categorical variables could cause errors or inaccuracies in the analysis. Therefore, encoding categorical variables into numerical variables helped me to develop more accurate Machine Learning models.

Another challenge was dealing with class imbalance. The target variable (dependent variable) had significant imbalance data between both classes, where Existing Customers represented 84% while the Attrited Customers 16% respectively, which could lead to biased prediction. For that reason, I analysed the outcomes with and without applying Synthetic Minority Oversampling Technique (SMOTE). With this possible approach, I could address class imbalance and along with this technique, I used an Evaluation Metrics Library that was more suitable for imbalanced data.

The images are to illustrate the percentage of customers who will exit or not the bank and the other one is to visualise the analysis applying SMOTE Vs not applying SMOTE:

Chart, pie chart

Description automatically generated

**A picture containing square

Description automatically generated**

**Reflection**

On the whole, I was pleased with the performance of the models we developed. They achieved high accuracy and performed well on the test data, indicating that they could be effective in predicting customer churn. I also learned important lessons during the CA. For instance, I realised that it is crucial to choose the right metrics to evaluate the models. Additionally, I also learned that it is important to consider weighing pros and cons between different models, such as balancing and interpretability to ensure that the model best meets the needs of the project.

**Conclusion**

This CA was a great learning experience for me. I gained practical experience applying Machine Learning models to real-words data and learned a lot about the importance of choosing the right Python Libraries. Additionally, I also learned the value of collaboration and communication in the team project, as we were able to share ideas and insights to improve the models and achieve our goals.

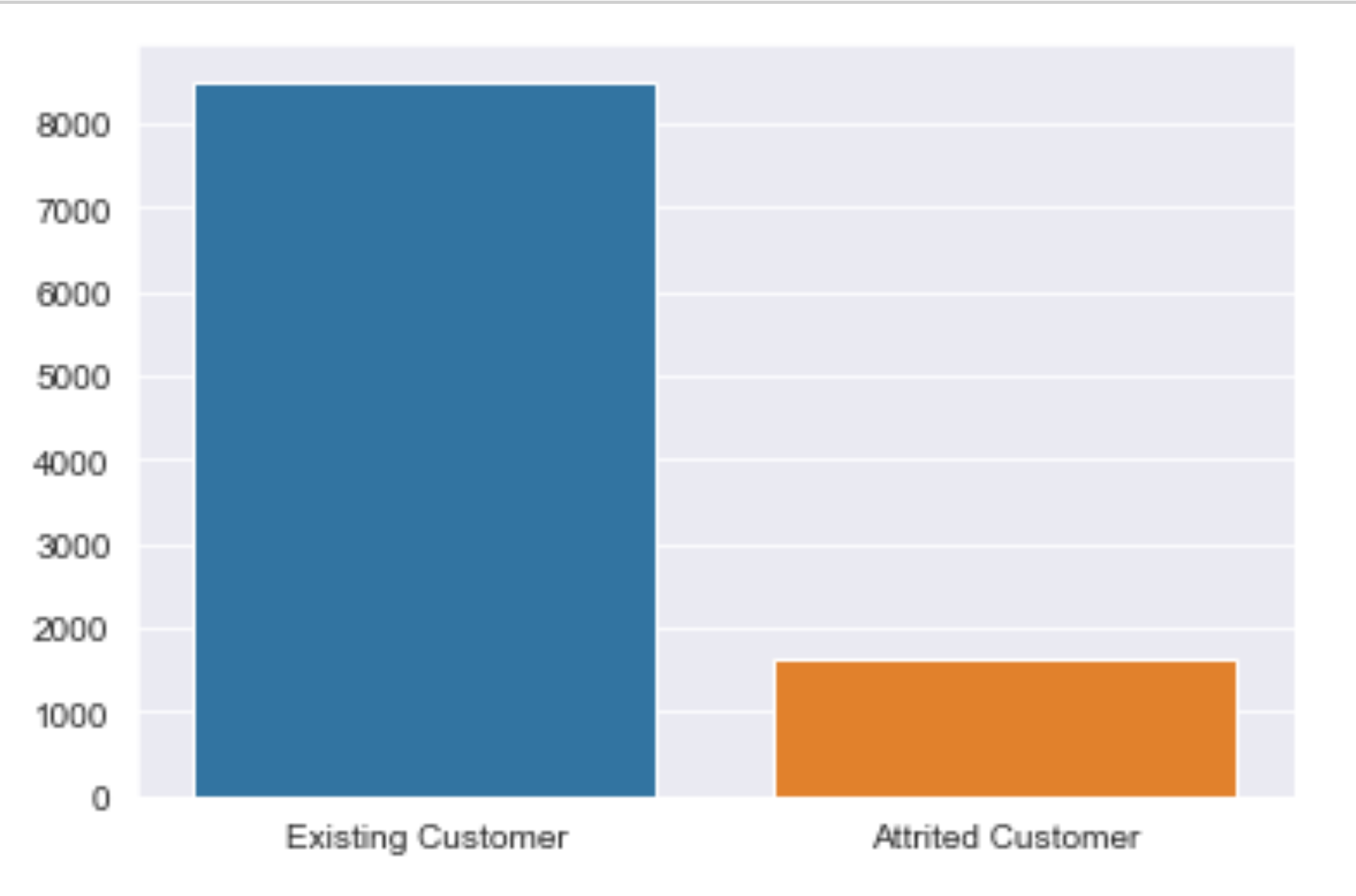
# Individual Contributions of Daniel M. Murphy

My contributions to this project can be split into distinct categories:

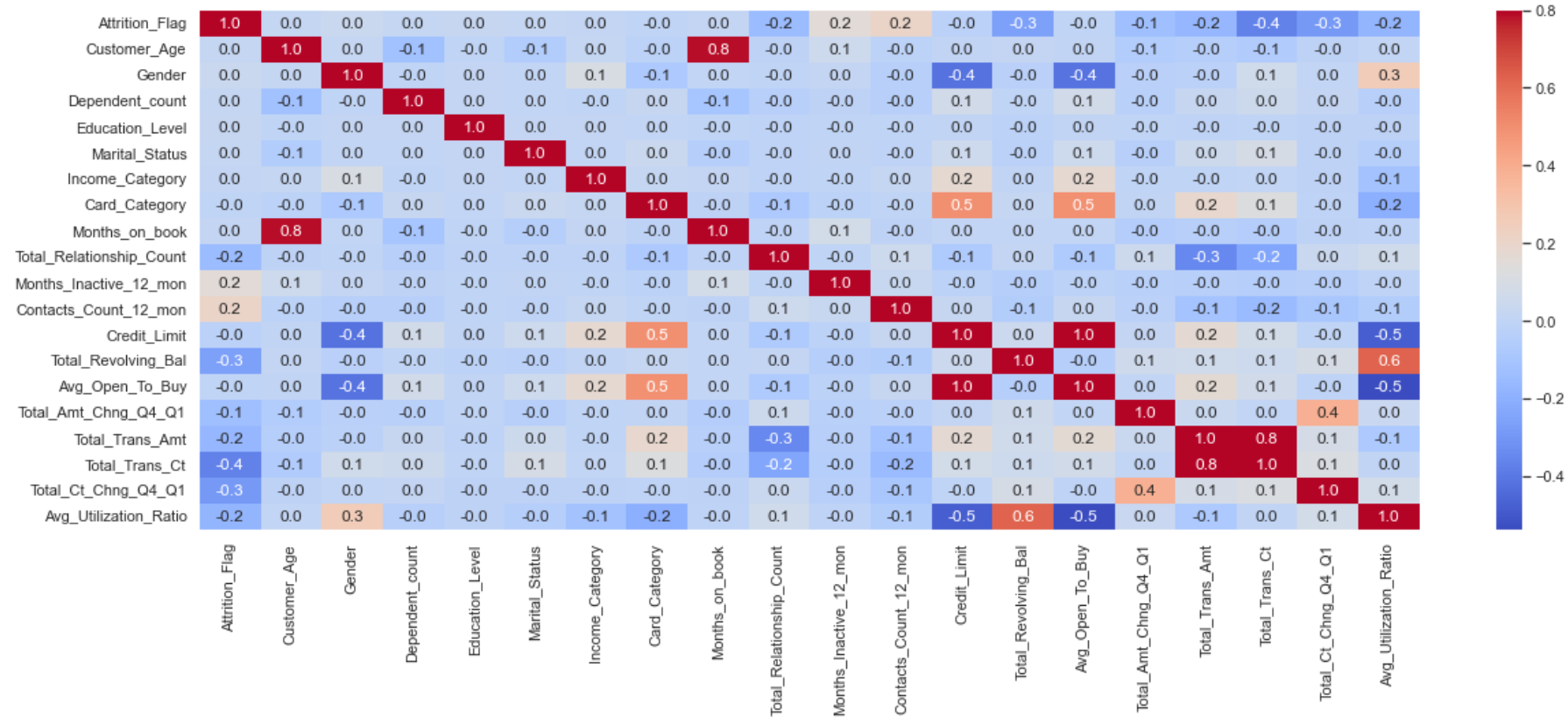
* Data Exploration
* Report Writing
* Machine Learning Model Evaluation
* Final Touches

**Data Exploration**

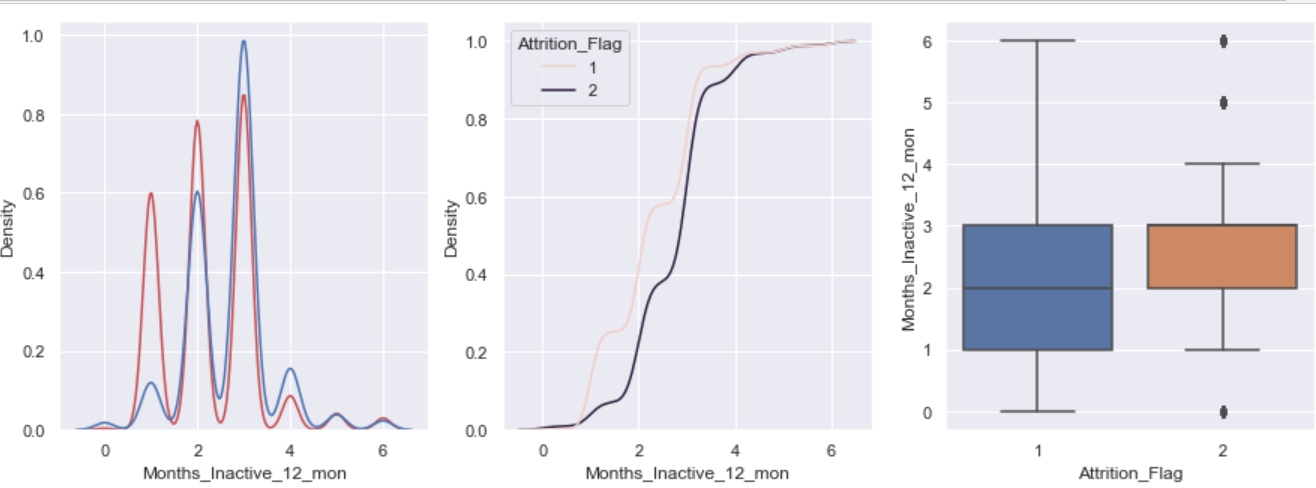
I started by characterising the dataset. I used common panda functions in order to get a glimpse of the data we were going to work with. Performing EDA, I identified the class imbalance that would need to be addressed shown below.



This process also involved finding the features most strongly correlated with the target variable through a correlation heatmap shown below.



Taking things a step further, I was able to explore interesting features in more details through comparing bar charts, and writing the code to produce PDF, CDF, & box plots (a sample of these are shown below).



# 

**Research & Report Writing**

I researched the domain area of customer churn, predicting customer churn and the customer churn in the credit card industry specifically. This enabled me to write an informed business understanding section. It also gave me a better context in which to understand the data - which helped as I wrote the data understanding and data preparation for EDA part of the report. Having written the EDA code, it made sense for me to write about the process, comparing and analysing the features of interest through visualisations. Finally, I summarized and concluded the EDA section of the Report.

For the report, I also researched which models were best to choose for our specific problem. I made sure to reference this in the report. Moreover, I wrote the sections of preparing the data for the Machine Learning Models, the techniques used, explaining the reasons for different accuracies across different splits, the difference in the balanced and imbalanced dataset as well as evaluating the RBF & Linear Kernel performance. I will talk about this more in the following section, however, as far as the report goes, it should be noted that I did a great deal of background research and ensure to back up and reference all of my findings referring to programming experts or the data itself.

I also researched articles and have formed the reference section of the report.

**Machine Learning Model Section**

Ariel wrote all the code for the Machine Learning Section of the project. He did great work. As stated briefly above, I analysed and interpreted a great deal of this code. I researched articles which were relevant to our project in order to know which model suits our project best. Moreover, I learned about and explained the techniques we used, such as, cross-validation & hyperparameters. I also interpreted and explained the differences in the balanced and imbalanced datasets, as well as the final performance of the RBF & Linear Kernel.

Perhaps my most important contribution to the Machine Learning section of the project was to identify an issue in the data preparation process Ariel implemented. Through my research, I came across an article by Brownlee (2016). In it, it’s explained that in order to prevent data leakage, the order of operation must be (Split, Scale, SMOTE). Ariel had originally scaled the data first. This allowed me to alter the code slightly (which was surprisingly difficult) in order to be able to apply the scaler separately to the X\_train and X\_test datasets.

**Final Touches**

To finish the project properly, I did the following:

* Ensuring the word count was correct.
* Code refactoring & formatting: Improving the code structure and ensuring a consistent style.
* Code linting: As mentioned above in finding order of operation issues.
* Markdown clean-up & notebook organisation.
* Editing the report to ensure tidiness, correct grammar, and spelling.
* Forming the reference list.

# 

# References

Bento, C. (2020). *Bias-Variance tradeoff in Machine Learning models: A practical example*. [online] Medium. Available at: https://towardsdatascience.com/bias-variance-tradeoff-in-machine-learning-models-a-practical-example-cf02fb95b15d [Accessed 26 Apr. 2023].

Brownlee, J. (2016). *Data Leakage in Machine Learning*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/data-leakage-machine-learning/.

Brownlee, J. (2018). *A Gentle Introduction to k-fold Cross-Validation*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/k-fold-cross-validation/.

Hale, J. (2019). *Scale, Standardize, or Normalize with Scikit-Learn*. [online] Medium. Available at: https://towardsdatascience.com/scale-standardize-or-normalize-with-scikit-learn-6ccc7d176a02#:~:text=Use%20MinMaxScaler%20if%20you%20want [Accessed 22 Apr. 2023].

Joseph, V.R. and Vakayil, A. (2021). SPlit: An Optimal Method for Data Splitting. *Technometrics*, 64(2), pp.166–176. doi:https://doi.org/10.1080/00401706.2021.1921037.

Krishnamurthy, S. (2022). *Feature interactions: An overview*. [online] Medium. Available at: https://towardsdatascience.com/feature-interactions-524815abec81 [Accessed 23 Apr. 2023].

Lanenok (2015). *machine learning - When to use Random Forest over SVM and vice versa?* [online] Data Science Stack Exchange. Available at: https://datascience.stackexchange.com/questions/6838/when-to-use-random-forest-over-svm-and-vice-versa#:~:text=For%20a%20classification%20problem%20Random [Accessed 29 Apr. 2023].

Reichheld, F. and Sasser, W.E. (1990). Zero Defections: Quality Comes to Services. *Harvard Business Review*. [online] 1 Sep. Available at: https://store.hbr.org/product/zero-defections-quality-comes-to-services/90508?fromSkuRelated=519X-PDF-ENG&ab=store\_idp\_relatedpanel\_-\_zero\_defections\_quality\_comes\_to\_services\_90508.

Shetty, R. (2021). *Predicting a Failure in Scania’s Air Pressure System (APS)*. [online] Towards Data Science. Available at: https://towardsdatascience.com/predicting-a-failure-in-scanias-air-pressure-system-aps-c260bcc4d038 [Accessed 30 Mar. 2023].

Sunil, R. (2017). *Learn How to Use Support Vector Machines (SVM) for Data Science*. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/.

Xu, Y. and Goodacre, R. (2018). On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning. *Journal of Analysis and Testing*, 2(3), pp.249–262. doi:https://doi.org/10.1007/s41664-018-0068-2.