Student Number: F220309

AI and Machine Learning COP528

Understanding and Predicting Credit Card Customer Churn

# **Introduction**

The banking sector is heavily customer dependent and with countless banks operating worldwide, the market is extremely competitive. Customer churn or customer attrition is when a customer stops giving a company their business (Xie et al, 2009). Reichheld and Teal (1996) suggested that by increasing customer retention by only 5% a company could increase profits by 25% to 95%, depending on the industry. Therefore, insights into customer behaviour are crucial for customer retention and could assist in attracting new customers too if the reason these customers are leaving is addressed (AL-Najjar, AL-Rousan & AL-Najjar, 2022).

This report is being compiled to address customer attrition in the credit card section of a bank. A machine learning pipeline was developed to pre-process and analyse the data to try to understand more about the bank's customer churn. The data was then split into training and test sets. Models (Random Forest and SVC) were then trained, and the class of existing customer or churned customer were predicted. Finally, the models were evaluated through accuracy and other classification evaluation metrics with accompanying visualisations. Results, showed that experiment b) Random Forest with final adjusted parameters was the best model. However, further testing and analysis needs to be conducted to address the algorithms misclassification error.

# **Data and Preliminary Analysis**

## The dataset contained the credit card portfolios of 10,127 customers. There were no null values in the dataset. There were 21 features for each customer. Numerical features included client number, age, dependent count, months on book, number of products held by customer (total relationship count), months inactive, number of contact in the last 12 months, credit limit, total revolving balance on credit card, open to buy credit line (average of last 12 months), change in transaction amount (Q4 over Q1), total transaction amount (last 12 months), total transaction count (last 12 months), change in transaction count (Q4 over Q1), and average card utilisation ratio.

Categorical features included attrition flag, gender, education level, marital status, income category, and card category.

The histogram matrix shows the graphical distribution of numerical features (Fig.1). There is a large spike in months on book at around 35 months meaning that many customers seemed to be leaving at this same point in time. This suggested that perhaps an offer ended at around 35 months causing them to leave (top right graph).

A few of the graphs seem to have caps, one of which is age. Age seemed to be capped at 26 which would need to be confirmed with the client as this could cause problems further down the line as machine learning (ML) algorithms may learn that 26 is the minimum age and would be unable to classify or predict any new data for anyone under the age of 26 even though most banks allow anyone over 18 to apply for a credit card (1st row, 2nd graph). Average card utilisation ratio showed that there were many customers that never used their credit card (last graph).

Chart, engineering drawing

Description automatically generated

*Fig 1.* *Histogram Matrix of Numerical Features.*

Most of the graphs were right skewed meaning that most of the numerical features were not normally distributed and had means greater than the median. The features also had different ranges of measurement. Therefore, the data would need to be transformed before algorithms could be run on the data.

The target feature, attrition flag, had 8500 existing customers and 1627 churned customers. This meant that the dataset was unbalanced with only 16% of the data set containing the class we wanted to predict (churners).

The data was labelled, and the target variable was binary. Therefore, this was a classification task, supervised machine learning algorithms were used to predict the class of attrited customer based of the rest of the features.

Chart, diagram

Description automatically generated

*Fig2. Scatter Matrix of Selected Numerical Features.*

A scatter matrix was created showing selected attributes based on their standard deviations, looking to see if there was a relationship between these attributes. There seemed to be three main graphs showing relationships. Months on book and customer age, average open to buy and credit limit and total transaction counts and total transaction amounts.

There was a moderate positive relationship between months on book and customer age. Although, as also seen in the Fig 1. there is a line cutting across the graph which suggests that people of all ages are closing their accounts at around 35 months.

Credit limit and average open to buy were perfectly positively correlated meaning that these 2 attributes are potentially measuring the same thing or something very similar and one may need to be dropped during the experiment phase as an irrelevant attribute.

Total transaction count and total transaction amount seemed to have a weak positive relationship and the data seemed to cluster into 3 points.

Fig 3. showed the mode, number of levels in each category (unique levels) and the frequency of the mode (Mode freq) for each of the categorical features.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Attrition Flag** | **Gender** | **Education Level** | **Marital Status** | **Income category** | **Card Category** |
| **Mode** | Existing customer | Female | Graduate | Married | < $40K | Blue |
| **Unique levels** | 2 | 2 | 7 | 4 | 6 | 4 |
| **Mode Freq** | 8500 | 5358 | 3128 | 4687 | 3561 | 9436 |

*Fig 3. Descriptive Statistics for Categorical Features*

# **Methods**

**Pre-processing**

Computations were carried out using Python Jupyter Notebook. Following the pipeline outlined in the introduction, the data first underwent pre-processing. The Client number column was dropped as it is not a predictive feature. Many machine learning algorithms only take in numerical values such as support vector classifier (SVC). Therefore, the categorical features needed to be transformed to numerical values. Marital status was nominal level data and was one hot encoded; one hot encoding uses a binary vector of 0s and 1s. A new column is created for each level in the marital status category e.g., married, divorced, single and unknown. If a customer was married, then a 1 was placed in the married column and 0s were placed in all the other level options (Géron, 2019). The original marital status column was dropped as it no longer provided useful information. As income, education level and card category were ordinal levels of data, all unknowns in the income and education level columns were filled with the mode (less than $40k and graduate respectively). Each of the columns was then encoded in order. Income: < $40K, $40K - 60K, $60K - $80K, $80K-$120K, > $120K (from 0 to 4). Education level: uneducated, high school, college, graduate, post-graduate, doctorate (0 to 5) and card category: blue, silver, gold, platinum (from 0 to 3). Attrition flag and gender were nominal levels of data and were arbitrarily binary encoded as there are only two levels in each feature. Attrition flag’s existing customers were encoded as 0 and attrited customers as 1. Females were encoded as 1 and males as 0.

The numerical data needed to be transformed due to large differences in measurement ranges. Standardisation is the transformation of numerical data to a common scale, meaning that the impact of each feature on the model’s output would be more balanced. In standardisation each feature is scaled independently by subtracting the mean value of a feature from each value in the feature and dividing the result by the standard deviation of the feature. This shifts the distribution to make a mean of zero and a standard deviation of one (Géron, 2019).

The data was then split into training and test sets with 80% of the data used for training and 20% for testing, this was due to there being a small sample of the target class (churners) so more training data was needed for the algorithm to train on. Gholamy et al. (2018) suggested a test and train split between 80/20 and 70/30 would be optimal as a general rule, although the size of data set and the number of features influences this. Train, test, split was chosen as it is a simple, fast and reproducible method of splitting the data for evaluation which is commonly used in machine learning.

**Models**

The models that were chosen to undertake this supervised classification task were Random Forest and SVC.

Decision Tree is a hierarchical supervised learning algorithm where the first node represents the whole data set. The tree then splits the data into smaller sets based on the value of a chosen feature, each branch represents the outcome of the test, and the leaf nodes represent the final class label (Myles et al., 2004).

Random Forest is an ensemble learning method used for various tasks such as classification and regression (Biau et al., 2016). Random forest conducts and combines a number of randomised Decision Trees and collects the average of their predictions. Unlike a Decision Tree alone, Random Forest is suggested to be able to handle data with complex feature interactions and can handle non-linear relationships between the data points and the target class, as is the case with the current data set (Biau et al., 2016). Random forest has various parameter settings, for instance the number of decision trees can be adjusted through n\_estimators, more trees typically means higher accuracy up to a certain point but often at the expense of computation time (Ellis, 2022). Max\_features sets the number of features that should be considered when looking for the best split in each Decision Tree. The higher the subset of features for Random Forest to select from the more likely there are to be meaningful features related to the target. Criterion can be adjusted and are used to measure the homogeneity of the branches and quality of the splits (Ellis, 2022). Max\_depth sets Random Forests maximum depth of levels. The higher the max depth the better predictive performance will be. However, there is the possibility of over fitting. Overfitting is when a model tries to fit to too much of the data and can capture noise rather than the underlying patterns. Ways to avoid overfitting is to carefully adjust the parameters, use regulation techniques such as L1 or L2, use cross validation or increase the size of the training data. Underfitting on the other hand is when the model is not complex enough and does not capture the patterns or relationships in the data (Brownlee, 2019).

SVC is a variant of Super Vector Machine (SVM) and is also a supervised learning algorithm that can handle both linear and non-linear data. SVC finds hyperplanes (decision boundaries) that best separate the data into different classes. Points closest to the hyperplanes are referred to as support vectors and they determine the orientation and position of the hyperplanes (Fraj, 2018). SVC has a parameter option called the kernel function which transforms the data into a higher dimensional space where it is linearly separated. Different kernels can support different types of data. Rbf and poly use non-linear hyperplanes and is used with non-linear data (Fraj, 2018). Both Random Forest and SVC are resistant to overfitting if their parameters are carefully tuned (Fraj, 2018; Biau et al., 2016).

Principal component analysis (PCA) is a feature reduction technique. The original features of the dataset are turned into a new set of uncorrelated features called principal components while capturing as much of the variability from the original dataset as possible (Jolliffe, 2005).

**Evaluation**

For classification tasks the evaluation metrics used are prediction accuracy, precision score, recall, F1 score, confusion matrix, receiver operating characteristic curve (ROC) and area under the curve (AUC) (Sunasra, 2020).

Prediction accuracy is the percentage of correctly classified data points the model was able to predict compared to the ground truth. However, prediction accuracy alone is not a good indicator of performance as it does not account for imbalanced data sets as is the case in this report. When data sets are imbalanced the model may learn to correctly classify the majority class and may be worse at predicting the minority class, which is the class we are more concerned with (churners). Therefore, other metrics are required to understand the errors that the model is making by taking into consideration true positives, data points that are correctly predicted as positive. True negatives, data points that are correctly predicted as negative. False positives, data points that are incorrectly predicted as positive and false negatives, data points that are incorrectly predicted as negatives.

Precision score measures how often the model’s positive predictions are correct.

Recall measures how well the model is able to correctly identify all positive samples or is able to avoid false negatives.

F1 score is the harmonic mean of precision and recall, ranging between 0 and 1. Higher scores indicate a better overall measure of the model’s performance (Sunasra, 2020).

ROC curve is a graphical representation that shows the classification performance of a model at all classification thresholds. The curve plots true positive rates against false positive rates. AUC measures the area underneath the entire ROC curve and indicates the ability of the model to distinguish between positive and negative classes (Sunasra, 2020).

Confusion matrix is a table which provides a summary of all true positives, true negatives, false positives, and false negatives in the model’s prediction.

# **Experiments and Results**

After pre-processing and splitting the data, the first set of experiments that were run used the Random Forest model.

a) The first Random Forest was run with no changes to the parameters.

b) the second experiment run was Random Forest with the number of trees set to 250, max depth of each decision tree set as 6, max features to consider when looking for the best split was set to 18 and criterion set to gini.

c) The 3rd experiment run was Random Forest with altered hyperparameters run after PCA was conducted with 5 components set.

The second set of experiments used SVC.

d) SVC was run with the rbf kernel selected.

e) SVC run with poly kernel and 3 degrees.

f) PCA was applied with 8 components and SVC was run with rbf kernel.

The best model was b) the Random Forest model

with manually adjusted parameters. Parameters were repeatedly adjusted with different variations and evaluated until arriving at the set of parameters in b). The final parameters selected were a max of 250 trees, with a max depth of 6, a subset of 18 features and the ‘gini’ criteria to determine quality of the split. However, the results of this model were only marginally better than the random forest with default parameters set this could be due to random forest models being less sensitive to hyperparameter adjustments compared to other models (Ellis, 2020). The worst performing model was c) Random Forest with adjusted parameters after a PCA (5). PCA works best when the most important features have the most variation which can be deduced from further analysis into each features standard deviation and may not be the case in this data set. The best preforming SVC model set up was d) SVC with rbf kernel, the rbf kernel uses a non-linear hyperplane and performed better than the poly kernel. This may be due to the rbf kernel being more flexible than the poly kernel at creating complex decision boundaries (Fraj, 2018). Fig 4. shows the prediction accuracy, precision score, recall score and F1 score for each of the models. The best performance model b) was highlighted green, and the worst c) was highlighted in red.

In Fig 4. Experiment b) shows a high overall prediction rate of 95.77%. The precision score showed that out of all the predicted churned customers 90.57% were actually churned customers. The recall score showed that only 82.26% of all actually churned customers were correctly classified as churned customers by the model. The F1 score showed a good score of 0.9 as it is close to 1 and shows that the model achieved a good balance between precision and recall.

|  | a | b | c | d | e | f |
| --- | --- | --- | --- | --- | --- | --- |
| **Prediction Accuracy (%)** | 95.67 | 95.77 | 85.04 | 92.84 | 92.20 | 88.80 |
| **Precision Score (%)** | 92.94 | 90.57 | 64.63 | 86.99 | 87.56 | 80.86 |
| **Recall Score (%)** | 77.48 | 81.26 | 16.21 | 65.44 | 60.25 | 40.06 |
| **F1 Score** | 0.8 | 0.9 | 0.3 | 0.7 | 0.7 | 0.5 |

*Fig 4. Evaluation Metrics for All Experiments*

Fig 5. Shows the ROC curve for experiment b) it also shows an AUC score of 0.98 which indicates that the model was almost perfectly able to distinguish well between churners and existing customers which seemed unlikely considering how small the sample of churned customers was and how low the recall score was. Therefore, a confusion matrix was conducted to explore this further (See Fig 6.).

Chart, line chart

Description automatically generated

*Fig 5. ROC Curve and AUC for Experiment h.*

Chart, treemap chart

Description automatically generated

*Fig 6. Confusion Matrix for Experiment b*

As churned customers were coded as 1 in the pre-process stage the confusion matrix shows that the model found 1671 true positives, 28 false positives, 58 false negatives and 269 true negatives. However, during preliminary analysis it was shown that there were 8500 existing customers and 1627 churned customers. Therefore, the model cannot have correctly predicted 1671 churned customers as there are not even that many in the whole data set and there seems to be a classification error which would require further analysis and an in depth look at the preprocessing stage.

# **Reflection**

In conclusion, this report has attempted to predict credit card customer churn. Preliminary analysis was conducted to gain an overview of the data, pre-processing was then conducted to transform the data into a format that was useable for the algorithms and the data was then split into train and test splits (80/20). Random forests and SVC’s were then conducted with various parameters altered and each of the models were evaluated, with visualisations being created for experiment b. The best performing model was b), Random Forest with the number of trees set to 250, max depth of each decision tree was set to 6, max features set to 18 and criterion set to gini. The worst performing model was model c) Random Forest with PCA (5).

However, further analysis would be required as after looking at the confusion matrix for experiment b) it appeared that the model was incorrectly classifying non-churned customers as churned customers, there could be a few for reasons for this. One being the class imbalance. A key issue in customer churn prediction is that the data tends to be imbalanced. There tends to be more existing customers than churned customers which makes it difficult to accurately predict the minority class of churners (Mann & Mann, 2023). However, this could be addressed by undersampling the majority class (removing existing customer samples to balance the data set), oversampling (creating more samples for the churned class to balance the classes). Class weights could be adjusted to give more importance to the minority class during training (Brownlee, 2019).

Other improvements that could be made to increase the model’s performance are to use cross validation rather than train test split. Cross validation is an evaluation method for machine learning models. Data is divided into k equal size folds with 1-fold being kept for testing. This process is repeated k number of times and each k fold is used once as the test set. For example, 10-fold cross validation would hold the first fold as the test set and then the model would be trained on the other 9 folds and then would be evaluated on the test fold. This process would be repeated with each fold being held as the test set in turn. The average is then calculated across all 10-folds (Daniels, 2012). This process is much more effective than train test split because it uses all of the data as the training and test set rather than 1 training set and 1 testing set and provides a more representative model evaluation. Cross validation can also help to avoid overfitting where the model performs well on that particular split but not on new unseen data (Daniels, 2012). In addition to this, other feature selection or feature reduction methods than PCA could be tried. 21 features are a large amount of features and it is likely that some of the features are redundant, such as average open to buy which was perfectly correlated with credit limit. Although, Random Forest is already fairly robust to irrelevant features, removing irrelevant features could improve performance and speed up computational time (Cheng, 2021). Finally, other algorithms could be created and evaluated once the core issue of misclassification was identified and dealt with, to see if they would perform better than the Random Forest.

##### **References**

AL-Najjar, D., Al-Rousan, N., and AL-Najjar, H. (2022). “Machine Learning to Develop Credit Card Customer Churn Prediction”. *Journal of Theoretical and Applied Electronic Commerce Research*, 17(4), pp. 1529–1542. <https://doi.org/10.3390/jtaer17040077>.

Biau, G. and Scornet, E. (2016) “A random forest guided tour,” *Test*, 25(2), pp. 197–227. Available at: <https://doi.org/10.1007/s11749-016-0481-7>.

Brownlee, J. (2019) *Overfitting and Underfitting With Machine Learning Algorithms*. Available at: <https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/>. (Accessed: March 24, 2023)

Cheng, A. (2021) *Machine Learning: Step-By-Step - Towards Data Science*. Available at: <https://towardsdatascience.com/machine-learning-step-by-step-6fbde95c455a>.

Daniels, B. (2012) “CROSSFOLD: Stata module to perform k-fold cross-validation,” *RePEc: Research Papers in Economics* [Preprint]. Available at: <https://econpapers.repec.org/RePEc:boc:bocode:s457426>. (Accessed: March 24, 2023)

Ellis, C. (2022) *Hyperparameter tuning in random forests*. Available at: <https://crunchingthedata.com/hyperparameter-tuning-in-random-forests/>. (Accessed: March 24, 2023)

Fraj, M.B. (2018) *In Depth: Parameter tuning for SVC - All things AI - Medium*. Available at: <https://medium.com/all-things-ai/in-depth-parameter-tuning-for-svc-758215394769>. (Accessed March 24, 2023)

Géron, A. (2019) *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. “O’Reilly Media, Inc.”

Gholamy, A., Kreinovich, V. and Kosheleva, O. (2018). “Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation”. *Departmental Technical Reports (CS)*, pp. 1–6.

Jolliffe, I.T. (2005) “Principal Component Analysis,” *Encyclopedia of Statistics in Behavioral Science* [Preprint]. <https://doi.org/10.1002/0470013192.bsa501>.

Mann, J. and Mann, H. (2023) “Customer Churn Prediction Model using Explainable Machine learning,” *International Journal of Computer Science Trends and Technology (IJCST)*, 11(1), pp. 33–38.

Myles, A.J., Feudale, R.N., Liu, Y., Woody, N.A., Brown, S.s*.* (2004) “An introduction to decision tree modeling,” *Journal of Chemometrics*, 18(6), pp. 275–285. <https://doi.org/10.1002/cem.873>.

Reichheld, F.F. and Teal, T.H. (1996) “The loyalty effect: The hidden force behind growth, profits, and lasting value,” *Long Range Planning*, 29(6), p. 909. <https://doi.org/10.1016/s0024-6301(97)82843-9>.

Sunasra, M. (2020) *Performance Metrics for Classification problems in Machine Learning*. Available at: <https://medium.com/@MohammedS/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b>. (Accessed: March 20, 2023)

Xie, Y., Li, X., Ngai, E.W.T., and Ying, W. (2009). “Customer churn prediction using improved balanced random forests”. *Expert Systems with Applications*, 36(3), pp. 5445-5449. <https://doi.org/10.1016/j.eswa.2008.06.121>.

**Code References**

Gabriel Atkin (2020) *Credit Card Customer Attrition Prediction - Data Every Day #113*. Available at: <https://www.youtube.com/watch?v=pyeUcmmLeiM>.(Accessed: March 24, 2023).

Géron, A. (2019) *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. “O’Reilly Media, Inc.”

Stef (2019b) *Pandas scatter\_matrix: Labels vertical (x) and horizontal (y) without being cut-off*. Available at: <https://stackoverflow.com/questions/58623528/pandas-scatter-matrix-labels-vertical-x-and-horizontal-y-without-being-cut>. (Accessed: March 20, 2023).

Ståle, H. and Egil Refsnes, J. (no date) *Python Machine Learning - Confusion Matrix*. Available at: <https://www.w3schools.com/python/python_ml_confusion_matrix.asp>. (Accessed: March 20, 2023).