Credit Card Attrition

Data Science for Business : Technical

Siddharth Singh, Jiabao Zhuang, Joe Kaewbaidhoon, Mingzhi Li
ss16915@stern.nyu.edu, jz5863@stern.nyu.edu, yk2912@stern.nyu.edu, ml8579@stern.nyu.edu
New York University
Stern School of Business

Codebase GitHub Link:
https://github.com/sidsingh1809/CreditCardAttrition-DSProject

Abstract

This report investigates the potential of machine learning to predict credit card customer attrition (churn). Publicly available data on customer demographics and credit card activity was analyzed to identify key factors influencing churn behavior. These factors, including average transaction amount and credit limit, were then leveraged to build machine learning models capable of predicting customers at risk of closing their accounts. While limitations exist in the data, such as the absence of details on card type and issuing institution, this project successfully demonstrates the potential for machine learning to revolutionize customer retention strategies within the financial services industry. Future work will focus on incorporating more specific data points and navigating ethical considerations surrounding data privacy and responsible marketing practices to ensure successful model deployment and implementation.

Business Understanding

Credit card companies spend millions of dollars to attract new customers to their cards. In 2017, Chase launched the Sapphire Reserve card with a 100,000-point bonus worth at least $1,500 if used to purchase travel on their Chase Travel Portal. This bonus is on top of other benefits such as a $300 travel credit and points for spending and travel. Even though today's bonuses are not as large as the legendary Chase offers, many credit card companies still offer sign-on bonuses worth at least $200-$800.

Credit card companies generally make their money from two sources: (1) Annual fees and (2) interest. In 2017, Chase Sapphire Reserve's annual fee was $450. Today, that fee is $550. It will be tough for Chase to recoup the acquisition cost within a year. These companies can only profit if customers do not churn their cards. In a 2017 interview with Chase CEO, Jamie Dimon, the CEO stated: "The benefit of the card gets booked over 7 years. The card was so successful it cost us $200 million, but we expect that to have a good return on it."

Card churning happens when customers sign up for new credit cards to earn the sign-on bonus, then close or underutilize the account. A quick search will reveal many sites such as this one that guide readers through the most optimal way to churn cards.

Group G9 used data from Kaggle and data science modeling to predict attrition (Target Variable = "Attrition Flag") based on various data features. The ability to predict attrition can help multiple parties at the bank:

- Marketing: Send retention offers to customers who are predicted to close the account
- Customer Service Agents: Choose to give retention offers only to those whose behavior matches the profile of customers most likely to close their account
- Product Managers: Use Predicted Attrition Rate as an early indicator. Adjust the card benefits package to retain customers. See how the adjusted package affects predicted and actual attrition rates.

Our model predicts data based on behavior on the account. Thus, we cannot predict if a customer will churn before they open an account. Otherwise, we would not give a large welcome bonus to those customers with high churn rates.

Sample Cost Matrix:

	Closed acc	Not closed acc
Predicted to close	-Retention Bonus + Benefits of customer staying	-Retention Bonus
Predicted to open	0	0

Data Understanding

The primary data source came from this Kaggle collection. The original data source came from Leaps Analytica but the link provided in Kaggle is broken, so we could not get to the original site.

The team explored other data sources. One of our team members works at Mastercard, but due to the proprietary nature of the data, the team member could not extract and share it. Generally, information about credit card consumer behavior is hard to find on the open internet. We found credit card delinquency rates from the FRED database, but aggregate-level data does not help predict attrition at the personal level.

Because this analysis was based on one data source, we did not need to conduct any data join operations.

Summary of issues with the data source:

- Unknown primary source
- Only 10,000 samples
- Only 16% of the data includes attritted customers
- Because the source is unknown, the geographic region is uncertain. We will assume it is US-based data.
- The data was posted in 2021 with no further updates. The data collection date is unspecified. The pandemic may affect customer behaviors - it's unclear if the data was collected pre or post-pandemic.

Data Preparation

This data source consists of 23 columns. The team faced no major data challenges. Analysis and transformations are described below.

Dropping Columns

Reading the data specs, we knew that the last two columns could be ignored. These columns were Naïve Bayes columns that the data uploader used. In this project, we will do our own Naïve Bayes analysis.

The first column, CUSTNUM, could also be ignored. Without the key from the bank, these CUSTNUMs have no meaning to us. We don't care about the identity of the individual. We are predicting attrition based on customer behavior, not identity.

```
[ ] # Load data
    df = pd.read_csv("BankChurners.csv")

    # Asked to ignore last two columns
    df = df[df.columns[:-2]]
    df = df.drop('CLIENTNUM', axis=1)
    df.head()
```

Fig 1 – Code block to initially drop columns

Numerical vs Categorical

Of the remaining 20 columns, 14 are numerical, and 6 are categorical.

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10127 entries, 0 to 10126
Data columns (total 20 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Attrition_Flag            10127 non-null  object
 1   Customer_Age              10127 non-null  int64
 2   Gender                    10127 non-null  object
 3   Dependent_count           10127 non-null  int64
 4   Education_Level           10127 non-null  object
 5   Marital_Status            10127 non-null  object
 6   Income_Category           10127 non-null  object
 7   Card_Category             10127 non-null  object
 8   Months_on_book            10127 non-null  int64
 9   Total_Relationship_Count  10127 non-null  int64
 10  Months_Inactive_12_mon    10127 non-null  int64
 11  Contacts_Count_12_mon     10127 non-null  int64
 12  Credit_Limit              10127 non-null  float64
 13  Total_Revolving_Bal       10127 non-null  int64
 14  Avg_Open_To_Buy           10127 non-null  float64
 15  Total_Amt_Chng_Q4_Q1      10127 non-null  float64
 16  Total_Trans_Amt           10127 non-null  int64
 17  Total_Trans_Ct            10127 non-null  int64
 18  Total_Ct_Chng_Q4_Q1       10127 non-null  float64
 19  Avg_Utilization_Ratio     10127 non-null  float64
dtypes: float64(5), int64(9), object(6)
memory usage: 1.5+ MB
```

Fig 2 – Data Types

Outliers

Outliers are plus or minus 1.5*IQR from the upper or lower quartile. Using box plots, we found some columns with clear outliers.

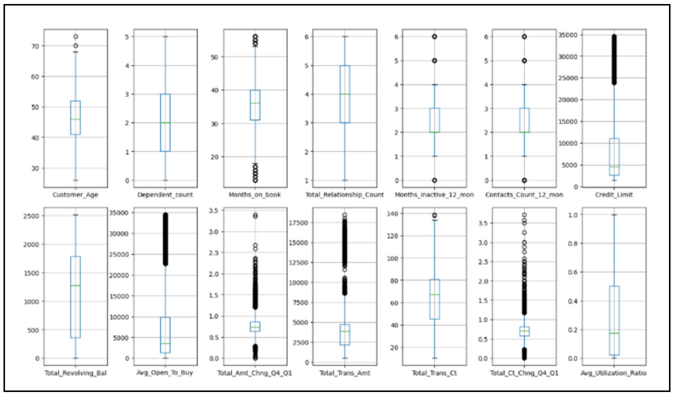

Fig 3 – Boxplot of various features

Analysis of columns with notable outliers:

Column	Rationale
Months on Book	The minimum month-on-book duration is 13 months. The maximum is 56. This range is within the expected range of a typical cardholder. Examples of atypical numbers may include negative numbers or months that exceed the average adult lifespan.
Credit Limit	Credit card companies give higher credit limits to those with higher incomes and better credits. Most customers will have average incomes and average credits, this is why the box plot median is low. The income in the US is not distributed normally. It is not surprising that the top whisker is much longer than the bottom whisker. Those with above-average incomes and credit scores are rewarded. The maximum limit of $35k is anecdotally sound.
Avg Open to Buy	This is the remaining credit limit. This distribution mimics the credit limit box plot. Those with high credit limits tend to have more credit remaining to use.
Total Transaction Amounts	Parallel to my analysis of the remaining credit, we see that the max transaction amount is only $18k. This is approximately half the maximum credit limit. Thus, it's not surprising to see that those with a lot of credit limit will also have high remaining credit. The credit limit, "average open to buy", and "total transaction amount" align and make logical sense.
Total Amount and Count Change	These two box plots almost mimic each other. The bottom number is at or above 0 which makes sense. The upper range is a ratio of the change between quarters. Because we validated that the transaction amounts and counts do not have any unexplained outliers, the ratio should be ok.

Based on the rationale above, no outliers were omitted from the analysis.

Missing Data

The data contains no missing data.

```
na_count = df.isna().sum()

na_count
```

```
Attrition_Flag              0
Customer_Age                0
Gender                      0
Dependent_count             0
Education_Level             0
Marital_Status              0
Income_Category             0
Card_Category               0
Months_on_book              0
Total_Relationship_Count    0
Months_Inactive_12_mon      0
Contacts_Count_12_mon       0
Credit_Limit                0
Total_Revolving_Bal         0
Avg_Open_To_Buy             0
Total_Amt_Chng_Q4_Q1        0
Total_Trans_Amt             0
Total_Trans_Ct              0
Total_Ct_Chng_Q4_Q1         0
Avg_Utilization_Ratio       0
dtype: int64
```

Fig 4 – No missing data

Dummy Variables

The categorical variables were assigned dummy variables.

```
categorical_features = df.select_dtypes(include=[object])
categorical_features.describe().T
```

	count	unique	top	freq
Attrition_Flag	10127	2	Existing Customer	8500
Gender	10127	2	F	5358
Education_Level	10127	7	Graduate	3128
Marital_Status	10127	4	Married	4687
Income_Category	10127	6	Less than $40K	3561
Card_Category	10127	4	Blue	9436

Fig 5 – Categorical variables that will be assigned dummy variables

For the target variable (Attrition Flag) and gender, the dummy variables were manually assigned since they were binary:

```
# Manual encoding
df['Attrition_Flag'] = df['Attrition_Flag'].replace({'Attrited Customer': 1, 'Existing Customer': 0})
df.Gender = df.Gender.replace({'F':1,'M':0})
```

Fig 6 – Code block to separate target variable

For variables with two or more levels, we used the pipeline below to implement the one-hot encoder and simple imputer for categorical and numerical columns.

```
categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='constant', fill_value='missing')),
    ('onehot', OneHotEncoder(handle_unknown='ignore', sparse_output = False))
])
```

Fig 7 – Code block for categorical transformation

Feature Engineering

```
# Interaction features
df_out['Avg_Transaction_Amt'] = df_out['Total_Trans_Amt'] / df_out['Total_Trans_Ct']
df_out['Used_Credit'] = df_out['Credit_Limit'] * df_out['Avg_Utilization_Ratio']
# Aggregated features
df_out['Inactive_Months_Ratio'] = df_out['Months_Inactive_12_mon'] / df_out['Months_on_book']
# Rate of change features
df_out['Amt_Chng_Rate'] = df_out['Total_Amt_Chng_Q4_Q1'] / df_out['Months_on_book']
df_out['Ct_Chng_Rate'] = df_out['Total_Ct_Chng_Q4_Q1'] / df_out['Months_on_book']
# Composite features
df_out['Total_Credit_Utilization'] = df_out['Total_Revolving_Bal'] + df_out['Avg_Open_To_Buy']
# Flag features
df_out['Zero_Revolving_Balance'] = (df_out['Total_Revolving_Bal'] == 0).astype(int)
```

Fig 8 – Code block for feature engineering

Variable	Rationale
Average Transaction Amount ($)	Definition: Total Trans Amount divided by total transaction count Rationale: A high average transaction amount is a better indication of usage than the total transaction amount or total transaction count. With this metric, we can tell if the user made one big purchase vs multiple purchases at average amounts.
	Hypothesis: those who use the card regularly for everyday purchases should be less likely to close the account.
Used Credit ($)	Definition: Credit Limit * Average Utilization Ratio This is the average balance on the card. Hypothesis: Those with higher balances will keep the card open. They won't be able to pay off the card balance for closure.
Inactive Months Ratio (ratio)	Inactive Months in the past 12 months divided by months on book Hypothesis: If the ratio is close to 1, they have spent most of their time at the bank inactive. They are more likely to leave.
Amount Change Rate (ratio)	Total Amount Change from Q4 to Q1 divided by months on book Hypothesis: The rate of change matters less the longer you've been a customer.
Count Change Rate (ratio)	Total Count Change from Q4 to Q1 divided by months on book Hypothesis: The rate of change matters less the longer you've been a customer.
Total Credit Utilization (ratio)	Total Revolving Balance from Q4 to Q1 divided by Credit Limit Hypothesis: Customers with higher credit utilization are less likely to close their accounts because they can not afford to pay off their balance.
Total Revolving Balance (Boolean)	Create a T/F flag. True (1) if the cardholder does not carry a revolving balance. Hypothesis: Customers who carry a balance are more likely to stay. They pay interest on the card. Unlikely to pay it off and close the account.

Finding Important Features

The team performed a Feature Importance Analysis to show the important features. We try to harness the power of gradient boosting, to shed light on the most important features that are affecting our attrition model.

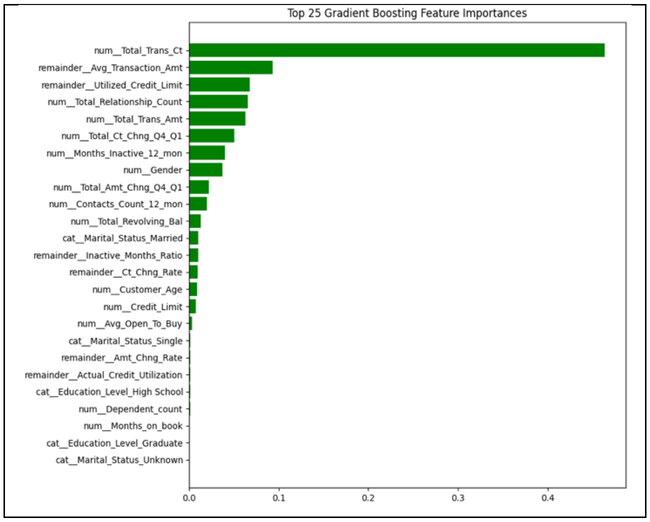

Fig 9 – Top 25 important features

The following features were most important:
```
['num__Avg_Open_To_Buy'
'num__Credit_Limit' 'num__Customer_Age'
 'remainder__Ct_Chng_Rate'
'remainder__Inactive_Months_Ratio'
 'cat__Marital_Status_Married'
'num__Total_Revolving_Bal'
 'num__Contacts_Count_12_mon'
'num__Total_Amt_Chng_Q4_Q1'
'num__Gender'
 'num__Months_Inactive_12_mon'
'num__Total_Ct_Chng_Q4_Q1'
 'num__Total_Trans_Amt'
'num__Total_Relationship_Count'
 'remainder__Utilized_Credit_Limit'
'remainder__Avg_Transaction_Amt'
 'num__Total_Trans_Ct']
```

For details on their correlation, see the Correlations section under “Exploratory Data Analysis”.

Exploratory Data Analysis

First, we performed high-level analysis using .head, .shape, .describe, and .info.

```
df.head()
```

	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Cont
0	Existing Customer	45	M	3	High School	Married	$60K - $80K	Blue	39	5	1	
1	Existing Customer	49	F	5	Graduate	Single	Less than $40K	Blue	44	6	1	

Fig 10 – df.head() output

```
df.shape

(10127, 20)
```

Fig 11 – Dataset size after initial preprocessing

```
numerical_features.describe().round(4).T
```

	count	mean	std	min	25%	50%	75%	max
Customer_Age	10127.0	46.3260	8.0168	26.0	41.000	46.000	52.000	73.000
Dependent_count	10127.0	2.3462	1.2989	0.0	1.000	2.000	3.000	5.000
Months_on_book	10127.0	35.9284	7.9864	13.0	31.000	36.000	40.000	56.000
Total_Relationship_Count	10127.0	3.8126	1.5544	1.0	3.000	4.000	5.000	6.000
Months_Inactive_12_mon	10127.0	2.3412	1.0106	0.0	2.000	2.000	3.000	6.000
Contacts_Count_12_mon	10127.0	2.4553	1.1062	0.0	2.000	2.000	3.000	6.000
Credit_Limit	10127.0	8631.9537	9088.7767	1438.3	2555.000	4549.000	11067.500	34516.000
Total_Revolving_Bal	10127.0	1162.8141	814.9873	0.0	359.000	1276.000	1784.000	2517.000
Avg_Open_To_Buy	10127.0	7469.1396	9090.6853	3.0	1324.500	3474.000	9859.000	34516.000
Total_Amt_Chng_Q4_Q1	10127.0	0.7599	0.2192	0.0	0.631	0.736	0.859	3.397
Total_Trans_Amt	10127.0	4404.0863	3397.1293	510.0	2155.500	3899.000	4741.000	18484.000
Total_Trans_Ct	10127.0	64.8587	23.4726	10.0	45.000	67.000	81.000	139.000
Total_Ct_Chng_Q4_Q1	10127.0	0.7122	0.2381	0.0	0.582	0.702	0.818	3.714
Avg_Utilization_Ratio	10127.0	0.2749	0.2757	0.0	0.023	0.176	0.503	0.999

Fig 12 – Numerical features description

```
categorical_features.describe().T
```

	count	unique	top	freq
Attrition_Flag	10127	2	Existing Customer	8500
Gender	10127	2	F	5358
Education_Level	10127	7	Graduate	3128
Marital_Status	10127	4	Married	4687
Income_Category	10127	6	Less than $40K	3561
Card_Category	10127	4	Blue	9436

Fig 13 – Categorical features description

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10127 entries, 0 to 10126
Data columns (total 20 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Attrition_Flag            10127 non-null  object 
 1   Customer_Age              10127 non-null  int64  
 2   Gender                    10127 non-null  object 
 3   Dependent_count           10127 non-null  int64  
 4   Education_Level           10127 non-null  object 
 5   Marital_Status            10127 non-null  object 
 6   Income_Category           10127 non-null  object 
 7   Card_Category             10127 non-null  object 
 8   Months_on_book            10127 non-null  int64  
 9   Total_Relationship_Count  10127 non-null  int64  
 10  Months_Inactive_12_mon    10127 non-null  int64  
 11  Contacts_Count_12_mon     10127 non-null  int64  
 12  Credit_Limit              10127 non-null  float64
 13  Total_Revolving_Bal       10127 non-null  int64  
 14  Avg_Open_To_Buy           10127 non-null  float64
 15  Total_Amt_Chng_Q4_Q1      10127 non-null  float64
 16  Total_Trans_Amt           10127 non-null  int64  
 17  Total_Trans_Ct            10127 non-null  int64  
 18  Total_Ct_Chng_Q4_Q1       10127 non-null  float64
 19  Avg_Utilization_Ratio     10127 non-null  float64
dtypes: float64(5), int64(9), object(6)
memory usage: 1.5+ MB
```

Fig 14 – Data Types

Customer Demographics

Evaluate whether the source data is significantly skewed from the general population demographics. This could have an impact on the likelihood of attrition. For example, if the sample skews significantly old, there could be more attrition due to death than a regular sample. On the other hand, younger demographics may be more likely to churn. They are more tech-savvy and more connected to media that advises on how to churn cards

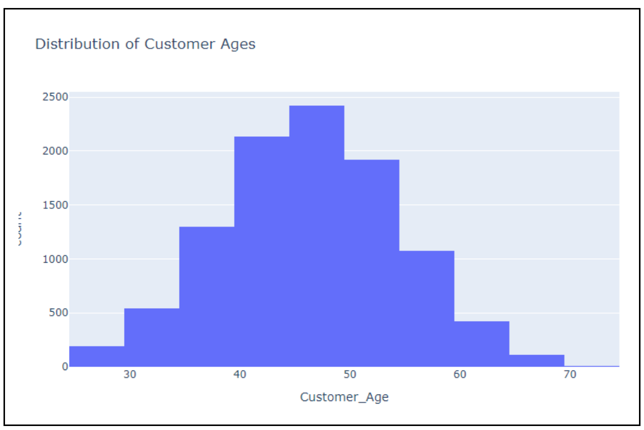

Fig 15 – Customer age distribution

The customer age distribution is almost normal. This approximates the data from Gallup. Those ages 40 - early 50 have the highest number of credit cards.

We also looked at customer gender for significant skew.

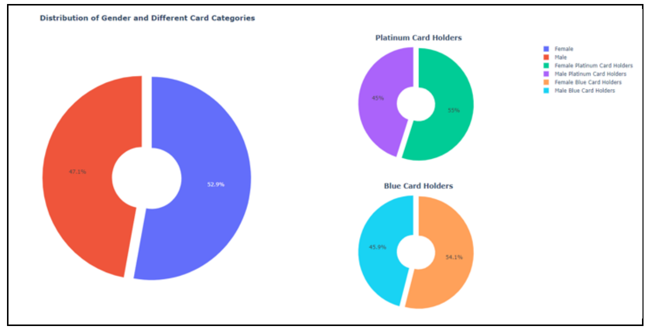

Fig 16 – Customer gender and card type split

We found no significant skew even when we account for card types. We do not need to adjust our data to account for skew.

Education Level

Education levels can determine the amount of financial literacy and health. In a recent study, up to 69% of Americans do not understand compound interest. Churning for sign-on bonuses or actively closing credit cards shows financial proactiveness. This is the education level in our data:

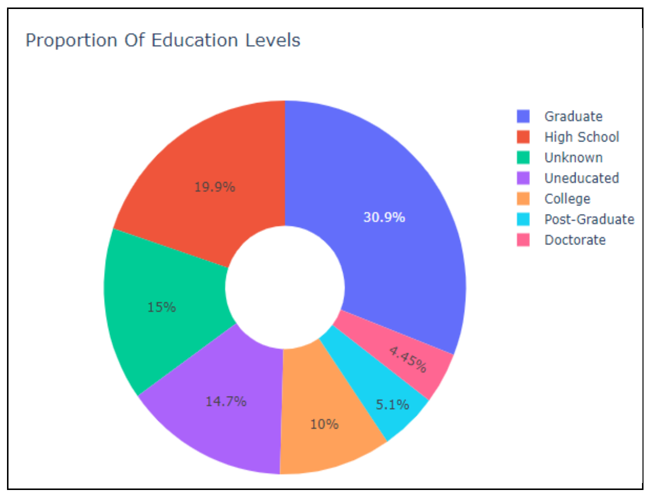

Fig 17 – Customer Education level proportions

Assuming 'Unknown' does not significantly represent formal education, the data shows that over 70% of our customers have formal education. Among these, nearly 10% have attained education levels higher than a graduate degree.

This is rather unsurprising. Those who qualify for credit cards must have a decent credit score and income. Those with formal tend to have higher incomes than those who do not. The population of credit card holders will likely have a higher education level than the general population.

Correlations

A correlation heatmap is a colorful visualization tool used to understand relationships between different variables in your data. The color intensity of each box indicates how strongly those two variables are linked. We can see see patterns and potential relationships at a glance.

Based on Fig 18, most of our variables were uncorrelated. Correlation matters because highly correlated variables can often be dropped during the modeling stage.

The most important correlations and possible explanations:

1. Customer Age and Months on Book - this metric implies that people do not frequently change financial institutions. Additionally, it may indicate that people choose their banks when they're younger and stay with the same financial institution. Thus, as you age, your months-on-book increases.
2. Total Revolving Balance and Avg Utilization Ratio - If you can't pay off your credit card monthly, you'll have higher utilization rates.
3. Avg Open to Buy (Remaining Credit) and Credit Limit - if you have a higher credit limit, you'll have more remaining credit. This assumes that those with higher credit do not spend significantly more.
4. Avg Open to Buy (Remaining Credit) and Avg Utilization Ratio - if you use your credit card a lot, you will have higher utilization % and lower remaining credit.

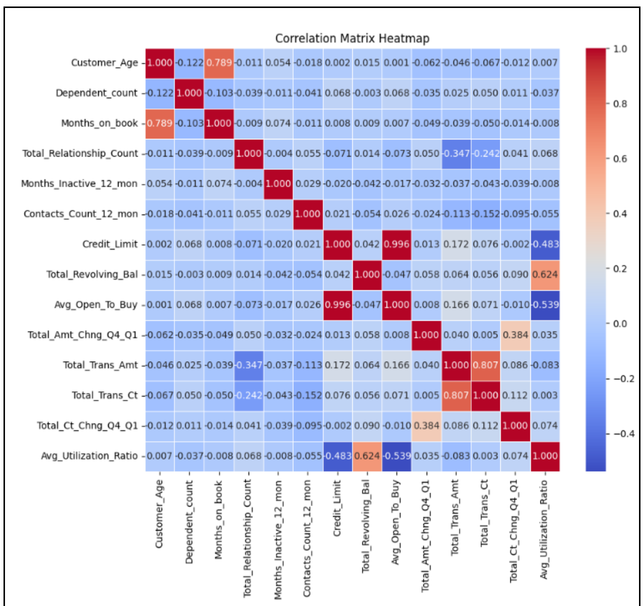

Fig 18 – Correlation matrix Heatmap

Target Variable Distribution

The team evaluated the distribution of attrition vs non-attrition to determine if our dataset was imbalanced.

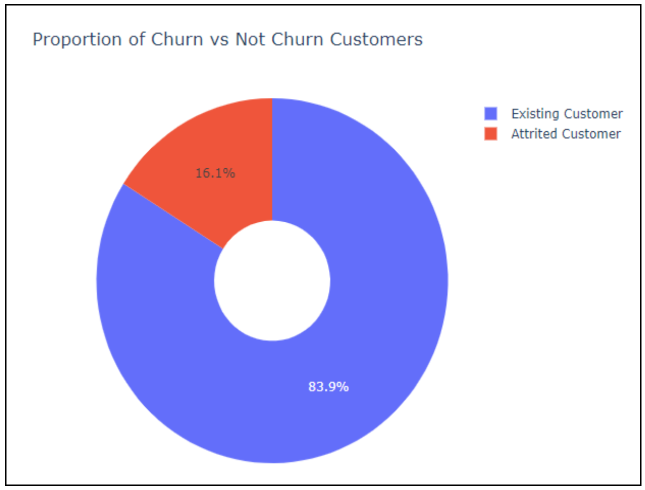

Fig 19 – Customer Attrition proportion

In our dataset, 16% of customers are attrited. The team will use SMOTE to balance the class distribution, supplemented by adjusted class weights in the modeling process. This combined method will help the model capture churn tendencies more effectively. Evaluation will focus on metrics like the F1-score and AUC-ROC for a nuanced assessment of model performance.

Modeling

Predicting attrition is a classification problem, not a regression one. Thus, we used classification models.

Models

- Naïve Bayes (Baseline model)
- Logistic Regression
- KNN
- Random Forest
- Gradient Boosting
- Neural Nets (Multi-Layer Perceptron)
- Ensemble model

Hyperparameter tuning

A dictionary named param_grids is defined containing potential hyperparameter values for each model. GridSearchCV with 3-fold cross-validation is used to find the best hyperparameters for each model. The ROC curves are generated using get_model_roc to compare model performance.

```
Fitting 3 folds for each of 1 candidates, totalling 3 fits
Best parameters for NaiveBayes: {}
Best accuracy for NaiveBayes: 0.8040729304514042

Fitting 3 folds for each of 3 candidates, totalling 9 fits
Best parameters for LogisticRegression: {'C': 0.1}
Best accuracy for LogisticRegression: 0.8826643140714601

Fitting 3 folds for each of 16 candidates, totalling 48 fits
Best parameters for KNN: {'metric': 'manhattan', 'n_neighbors': 3, 'weights': 'distance'}
Best accuracy for KNN: 0.9039111895309513

Fitting 3 folds for each of 27 candidates, totalling 81 fits
Best parameters for RandomForest: {'max_depth': None, 'min_samples_split': 2, 'n_estimators': 500}
Best accuracy for RandomForest: 0.9772092339361859

Fitting 3 folds for each of 8 candidates, totalling 24 fits
Best parameters for GradientBoosting: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 200}
Best accuracy for GradientBoosting: 0.9836053521540951
```

Fig 19 – Initial Training set fit output with best hyperparameter setting

Comparison Metrics

After applying the chosen hyperparameters, we calculate metrics like:

- Accuracy Score: Measures the proportion of correct predictions.
- Confusion Matrix: This shows how often the model predicted each class correctly or incorrectly.
- Classification Report: Provides precision, recall, F1-score, and support for each class.

By comparing these metrics across all the models, we identify the one that achieves the best balance between accuracy, precision, and recall for churn prediction in this specific dataset.

GaussianNB

Naïve Bayes is a good baseline choice for credit card attrition prediction due to its simplicity, speed, and interpretability. However, it assumes features are independent and follow Gaussian distributions, which may not hold for real-world data.

- Pros: Simple, fast to train, interpretable.
- Cons: Prone to overfitting, might not capture complex relationships in the data.

LogisticRegression

Logistic regression is another good option for credit card attrition prediction. It's interpretable, relatively simple, and works well for linear relationships between features and churn. However, it struggles with non-linear data and requires features to be on a similar scale.

- Pros: Interpretable, good for understanding the impact of features on churn.
- Cons: Assumes linear relationships between features and the target variable, might not be suitable for non-linear data.

KNeighborsClassifier

KNN is an intuitive algorithm for credit card attrition prediction. It classifies customers based on the behavior of their nearest neighbors in the data. This makes it easy to understand, that if similar customers churn, the model predicts a high churn probability for that customer.

- Pros: Easy to implement, works well with high-dimensional data.
- Cons: Sensitive to irrelevant features and outliers, and can be computationally expensive for large datasets (Curse of dimensionality).

RandomForestClassifier

Random Forest is a powerful option for churn prediction. It combines multiple decision trees for better accuracy and handles complex data patterns that simpler models might miss. While less interpretable than some models, it often yields better results, making it a strong competitor for credit card churn tasks.

- Pros: Handles non-linear data well, robust to overfitting.
- Cons: Possible "black box", less interpretable than LogisticRegression.

GradientBoostingClassifier

Gradient Boosting is another powerful option for attrition prediction. Like Random Forest, it is great for complex data patterns and achieves high accuracy. However, it's less interpretable and requires more computational resources.

- Pros: Very powerful, can learn complex patterns in data.
- Cons: Prone to overfitting if not tuned carefully, can be computationally expensive.

MLPClassifier (Multi-Layer Perceptron)

MLPClassifier is a powerful neural network option for the attrition problem. It can learn complex patterns in credit card data, potentially beating simpler models. However, it's like a black box - hard to understand why it predicts churn. It also requires a lot of computing power to train and fine-tune.

- Pros: Highly flexible, can learn complex non-linear relationships.
- Cons: Requires careful hyperparameter tuning, can be prone to overfitting, interpretability can be difficult.

Ensemble model

Ensemble methods combine predictions from multiple models, often leading to better performance and robustness. It is a powerful approach for credit card attrition prediction. Instead of relying on a single model, it combines the strengths of multiple ones.

The base models make predictions on the training data, and those predictions become new features for the meta-model. The meta-model then learns how to best combine these combined predictions from various models to deliver a potentially more accurate final churn prediction.

Stacking offers the advantage of potentially superior accuracy by leveraging the diverse strengths of different models. However, it's also more complex to train and understand when compared to a single model.

Benefits of Stacking:

- Improved Accuracy: By combining predictions from multiple models, stacking can often achieve higher accuracy than any single base model.
- Leverages Diverse Strengths: Different base models have different strengths and weaknesses. Stacking can leverage the combined power of these diverse approaches.

Drawbacks of Stacking:

- Increased Complexity: Stacking models are more complex than single models, requiring training multiple models and potentially more computational resources.
- Interpretability Challenge: Understanding how the stacking model arrives at its final prediction can be difficult due to the combined influence of multiple base models.

Alternative models

- Support Vector Machines (SVM): Powerful for high-dimensional data with clear class separation, but can be computationally expensive for large datasets.
- XGBoost: Similar to GradientBoostingClassifier but often considered more powerful and efficient.
- Decision Trees: Easier to interpret than Random Forest but can be prone to overfitting as individual trees become complex.

Evaluation

Following extensive feature engineering and meticulous model building, we employed ROC Curves to visualize the models' predictive performance. We complemented this with a comprehensive evaluation metric table, including accuracy, precision, recall, F1-score, and AUC, for straightforward comparison between the models.

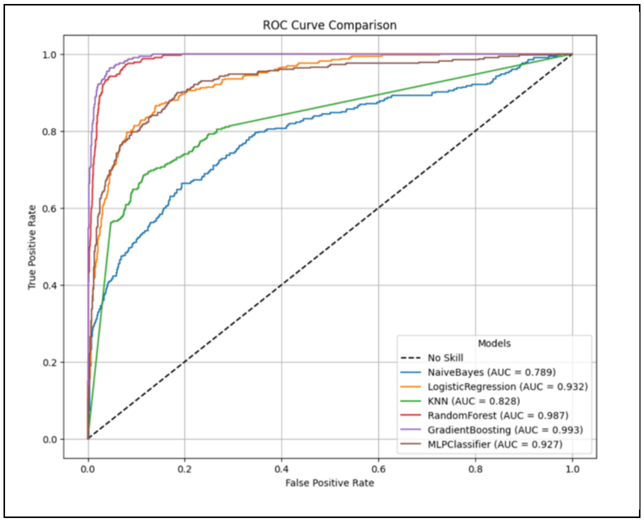

Fig 20 – ROC plot for all the different models employed

	Accuracy	**Precision (macro avg)**	**Recall (macro avg)**	**F1 (macro avg)**	**AUC**
Naive Bayes	0.76	0.3773	0.6636	0.4812	0.7886
Logistic Regression	0.86	0.5546	0.8379	0.6675	0.9317
KNN	0.83	0.4946	0.7003	0.5797	0.8284
Random Forest	0.95	0.8928	0.8410	0.8661	0.9872
Gradient Boosting	**0.96**	0.9043	**0.8960**	**0.9001**	**0.9925**
MLP Classifier	0.91	0.8250	0.6055	0.6984	0.9268
Stacking Classifier	0.94	**0.9323**	0.8443	0.8806	0.982

The evaluation table revealed Gradient Boosting as the champion across numerous crucial metrics. Although exhibiting a slightly lower precision score compared to the Ensemble Model (Stacking Classifier), Gradient Boosting shone in accuracy, recall, F1-score, and AUC. This superior class discrimination capability underscores its effectiveness in pinpointing customers at risk of churn.

While the Ensemble model, combining predictions from multiple models, achieves the highest precision, its performance in other metrics is not as robust. Similarly, the Random Forest model demonstrates strong performance with high accuracy, excellent precision, recall, F1-score, and AUC, albeit slightly below Gradient Boosting.

Gradient Boosting Model Analysis

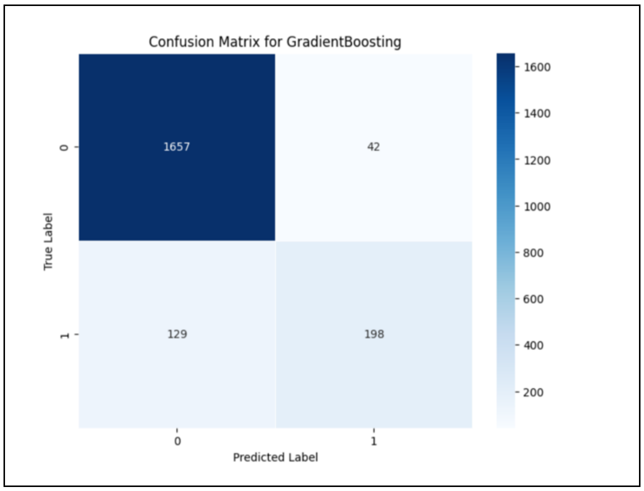

Fig 21 – Confusion Matrix for Gradient Boosting model

Delving deeper into the confusion matrix for Gradient Boosting, the top performer in our evaluation, yielded several noteworthy observations. The model excelled at accurately predicting non-attrited instances, as evidenced by the 1657 true negatives in the top-left cell. Additionally, it successfully identified 198 cases of attrition. However, there were 42 false positives and 129 false negatives. These represent instances that were incorrectly classified as attrited when they were not, and instances of attrition that the model missed. While the model demonstrates a strong ability to correctly classify non-churned customers, there is room for improvement in reducing both false positives and false negatives, which are critical for effective churn prediction. Further analysis and refinement might be necessary to enhance the model's performance in accurately recognizing churn, ultimately optimizing its predictive capabilities.

Cost-Benefit Analysis

As mentioned earlier, the team also conducted a cost-benefit analysis. This analysis assumed a 40% cost-to-benefit ratio for customer retention. In other words, retaining a customer would incur a cost of $400 (representing the retention bonus) while yielding an estimated benefit of $1000.
Here's a breakdown of the cost-benefit terminology used in the analysis:

- True Positive (TP): This scenario occurs when the model predicts a customer will churn, and they actually do. The company incurs a cost of $400 for the retention bonus but retains the customer's benefits, valued at $1000.
- False Positive (FP): This happens when the model predicts a customer will churn, but they don't. The company needlessly spends the $400 retention bonus.
- False Negative (FN): This occurs when the model predicts a customer will not churn, but they do. The company loses the potential benefits of retaining the customer without incurring the retention bonus cost.

```
Sample Cost Matrix:
                    Does not Close  Close Account
Predicted to Close             600           -400
Predicted to Open                0              0
```

Fig 22 – Sample Cost Matrix

Profit Curve Analysis

Leveraging the cost-benefit assumptions, we conducted a profit curve analysis. This analysis helps us identify the optimal threshold for targeting customers for retention efforts. The threshold represents the minimum likelihood of churn a customer must exhibit to be flagged for retention.

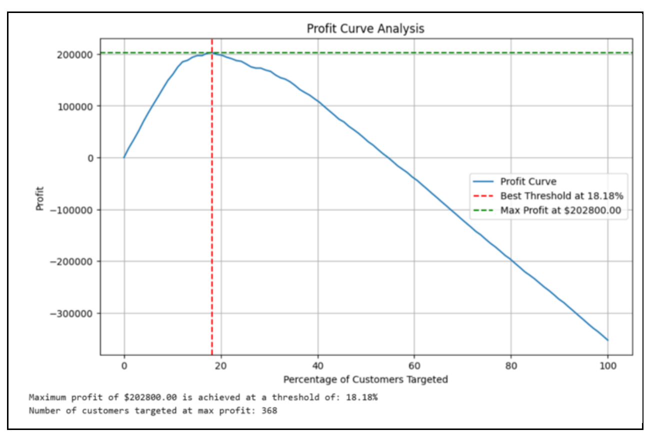

Fig 23 – Profit Curve

Our analysis revealed that a maximum profit of $202,800 can be achieved at a threshold of 18.18%, targeting 368 customers. This signifies that at this threshold, the model strikes the best balance between the cost of retention efforts (e.g., retention bonuses) and the benefits of retaining customers, resulting in the highest overall profit.

Deployment, Risks, and Future Work

Model Limitation

Generally, it is unwise to extrapolate the model to situations that do not match the training data.

Examples of factors that may impact how our model performs in the real world:

1. Type of Financial Institution: a local credit union, a national bank, or a regional bank will have very different clientele.
2. Types of Credit Cards: Credit cards are marketed differently. To apply the same model to predict attrition for Chase Reserve, a premium travel card, as Bankamericard, a no-fee, no-point starter card suited for balance transfers may not work. The source data does split customers into blue vs platinum card holders but provides no detail on what those tiers mean.
3. Geographic Location: Different countries, states, and cities have different cultures and different regulations that may affect customer attrition behavior.

Unfortunately, the dataset used to train the model does not provide much information about the three factors above.

Deployment

Given the limitations described above, the model should be deployed in stages. These are the steps I would take if I were a bank.

1. Identify Target Customer Segments that you would like to retain. These are going to be either those with high net worth or customers who are paying high interest.
2. Deploy the model to a specific geographic location within the US. For example, the tri-state NJ, NY, PA area.
3. Monitor the accuracy and F-Score of the model over time based on the model's prediction vs their actual attrition. Don't take any marketing action. Do this for 3 months.
4. Send out marketing material to remind customers who are predicted to leave of your card benefits (low cost) or send out retention offers (higher cost)
5. Expand to other geographic regions.

Results Monitoring

There are two things to monitor:

1. How well the model predicts attrition.
2. Whether attrition rates are declining.

The bank should already have information about account status. For monitoring, there are no special actions needed to collect the data.

Additional Study and Ethical Concerns

Based on our discussion in the "Model Limitation" section, we assume that different customer segments will behave differently regarding attrition. Here are ways we think we can improve our results:

1. Collect data about the financial institution, credit card types, and geographic location.
2. Perform Descriptive Functions to cluster consumers into groups. Run the model based on each cluster. Do cluster-specific models yield significantly better results? Does feature weight change significantly between clusters?
3. Simply collect more data points.

Most of the information used to conduct this attrition analysis is considered PII (Personally identifiable information) and sensitive financial data. Thus, it will be difficult to share this data between banks unless it is anonymized and aggregated. Any analysis will have to be done at the bank level, not industry wide. Large banks may have a scale and capital advantage over smaller banks when it comes to running these data science projects.

Lastly, these models will likely result in consumers getting more ads. If banks are providing value to their customers to convince them to stay, these ads may be worth the costs. However, if banks simply bombard consumers with phone calls and emails without providing new retention incentives, this is a net loss for consumers.

Conclusion

This report investigated the effectiveness of various classification algorithms for credit card attrition prediction. We explored models including Naive Bayes, Logistic Regression, KNeighborsClassifier, Random Forest, and

MLP Classifier. We also implemented a stacking ensemble model that combines predictions from these models.

Among the evaluated approaches, Gradient Boosting Classifier achieved the highest test accuracy of 96%. This result suggests its effectiveness in identifying customers at risk of churn. However, it's important to consider the trade-off between accuracy and interpretability. Gradient Boosting can be more complex to understand compared to simpler models.

Future research directions could involve exploring variations of Gradient Boosting with different hyperparameter tuning or feature engineering strategies specifically tailored for credit card attrition data. Additionally, investigating other advanced models, like XGBoost, which builds upon Gradient Boosting principles, might be beneficial.

Overall, this study highlights Gradient Boosting Classifier as a powerful tool for credit card attrition prediction, offering superior accuracy in identifying potential churners. Further exploration within this family of models and feature engineering techniques holds promise for even better results.

Appendix

Team Contributions

There was approximately equal contribution from all members of the team:

- All members of the team reviewed and wrote some pieces of the code. However, Jiabao and Siddharth are the most fluent in Python and we ended up using Jiabao' file as the master for all other edits. Towards the end of the project, we locked up the master file so only Jiabao can make edits. Siddharth also played an active role in adding more models.
- The team regularly met to determine our data analysis strategy. For example, we discussed feature engineering together. We reviewed the need for SMOTE. We reviewed our model results and tweaked the model parameters together.
- The team also worked on the report together. Joe was the primary contributor for the Business Context, Data Understanding, Data Prep, EDA, and the "Deployment, Risks, and Future Work" sections. Sid wrote much of the modeling section. Mingzhi put together the Evaluation section
- The team split the slides and worked on each section independently. Mingzhi played a key part in making sure the slides were cohesive and visually appealing. The team rehearsed the presentation together.

Link to Raw Data :
https://www.kaggle.com/datasets/sakshigoyal7/credit-card-customers

Link to GitHub Repo of our Code :
https://github.com/sidsingh1809/CreditCardAttrition-DSProject.git

References

[1] Sakshi Goyal. 2023. Credit Card Customers Dataset. Retrieved May 3, 2024, from https://www.kaggle.com/datasets/sakshigoyal7/credit-card-customers

[2] CBS News. 2024. "A jaw-dropping credit card bonus is about to expire." Retrieved May 3, 2024, from https://www.cbsnews.com/news/a-jaw-dropping-credit-card-bonus-is-about-to-expire/

[3] CNBC. n.d. Best credit card sign-up bonuses. Retrieved May 3, 2024, from https://www.cnbc.com/select/best-credit-card-sign-up-bonuses/

[4] Dimon, J. (2017, January 18). Jamie Dimon: I wish Chase Sapphire Reserve cost JPMorgan more money. CNBC. Retrieved May 3, 2024, from https://www.cnbc.com/2017/01/18/jamie-dimon-i-wish-chase-sapphire-reserve-cost-jpmorgan-more-money.html

[5] Gen Y Finance Guy. 2017. $1000 Bonus for Chase Sapphire Reserve Card. Retrieved May 3, 2024, from https://genyfinanceguy.com/2017/01/15/1000-bonus-chase-sapphire-reserve-card/

[6] St. Louis Federal Reserve. n.d. "Commercial Bank Assets to GDP for Democratic Republic of the Congo," Federal Reserve Economic Data. Retrieved May 3, 2024, from https://fred.stlouisfed.org/series/DRCCLACBS.

[7] Fuzz. 2021. "Machine Learning Classification Models." Retrieved May 3, 2024, from https://medium.com/fuzz/machine-learning-classification-models-3040f71e2529.

[8] Analytics Vidhya. (2021, August). Ensemble Stacking for Machine Learning and Deep Learning. Retrieved May 15, 2023, from https://www.analyticsvidhya.com/blog/2021/08/ensemble-stacking-for-machine-learning-and-deep-learning/

[9] GeeksforGeeks. (n.d.). Scikit-learn Classification Metrics. Retrieved May 3, 2024, from https://www.geeksforgeeks.org/sklearn-classification-metrics/?ref=ml_lbp