Bank Customer Attrition model

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

https://github.com/sfs-projects/ml

Objective

- The main objective of this analysis is to build a machine learning model to predict customer attrition for a bank.
- The primary focus of the model is on prediction rather than interpretation, with the goal of helping the bank identify customers who are at risk of leaving and take proactive measures to retain them.
- The benefits of this analysis include reducing customer churn, improving customer satisfaction, and increasing revenue for the bank.
- By accurately identifying the customers who are most likely to leave, the bank can develop targeted retention strategies that address the specific needs and concerns of those customers, which can ultimately lead to increased loyalty and profitability for the bank.

Description

- Dataset source: https://www.kaggle.com/datasets/thedevastator/predicting-credit-card-customer-attrition-with-m
- The dataset contains customer information to help analysts predict customer attrition, such as age, gender, marital status, income category, relationship with credit card provider, spending behavior.

Columns and description:

➔ CLIENTNUM - Unique identifier for each customer. (Integer)

➔ Attrition_Flag - Flag indicating whether or not the customer has churned out. (Boolean)

➔ Customer_Age - Age of customer. (Integer)

➔ Gender - Gender of customer. (String)

➔ Dependent_count - Number of dependents that customer has. (Integer)

➔ Education_Level - Education level of customer. (String)

➔ Marital_Status - Marital status of customer. (String)

➔ Income_Category - Income category of customer. (String)

➔ Card_Category - Type of card held by customer. (String)

➔ Months_on_book - How long the customer has been on the books. (Integer)

➔ Total_Relationship_Count - Total number of relationships a customer has with the credit card provider. (Integer)

➔ Months_Inactive_12_mon - Number of months customers has been inactive in the last twelve months. (Integer)

➔ Contacts_Count_12_mon - Number of contacts customers have had in the last twelve months. (Integer)

➔ Credit_Limit - Credit limit of customer. (Integer)

➔ Total_Revolving_Bal - Total revolving balance of customers. (Integer)

➔ Avg_Open_To_Buy - Average open to buy ratio of customers. (Integer)

➔ Total_Amt_Chng_Q4_Q1 - Total amount changed from quarter 4 to quarter 1. (Integer)

➔ Total_Trans_Amt - Total transaction amount. (Integer)

➔ Total_Trans_Ct - Total transaction count. (Integer)

➔ Total_Ct_Chng_Q4_Q1 - Total count changed from quarter 4 to quarter 1. (Integer)

➔ Avg_Utilization_Ratio - Average utilization ratio of customers. (Integer)

Data exploration, data cleaning and feature engineering, EDA

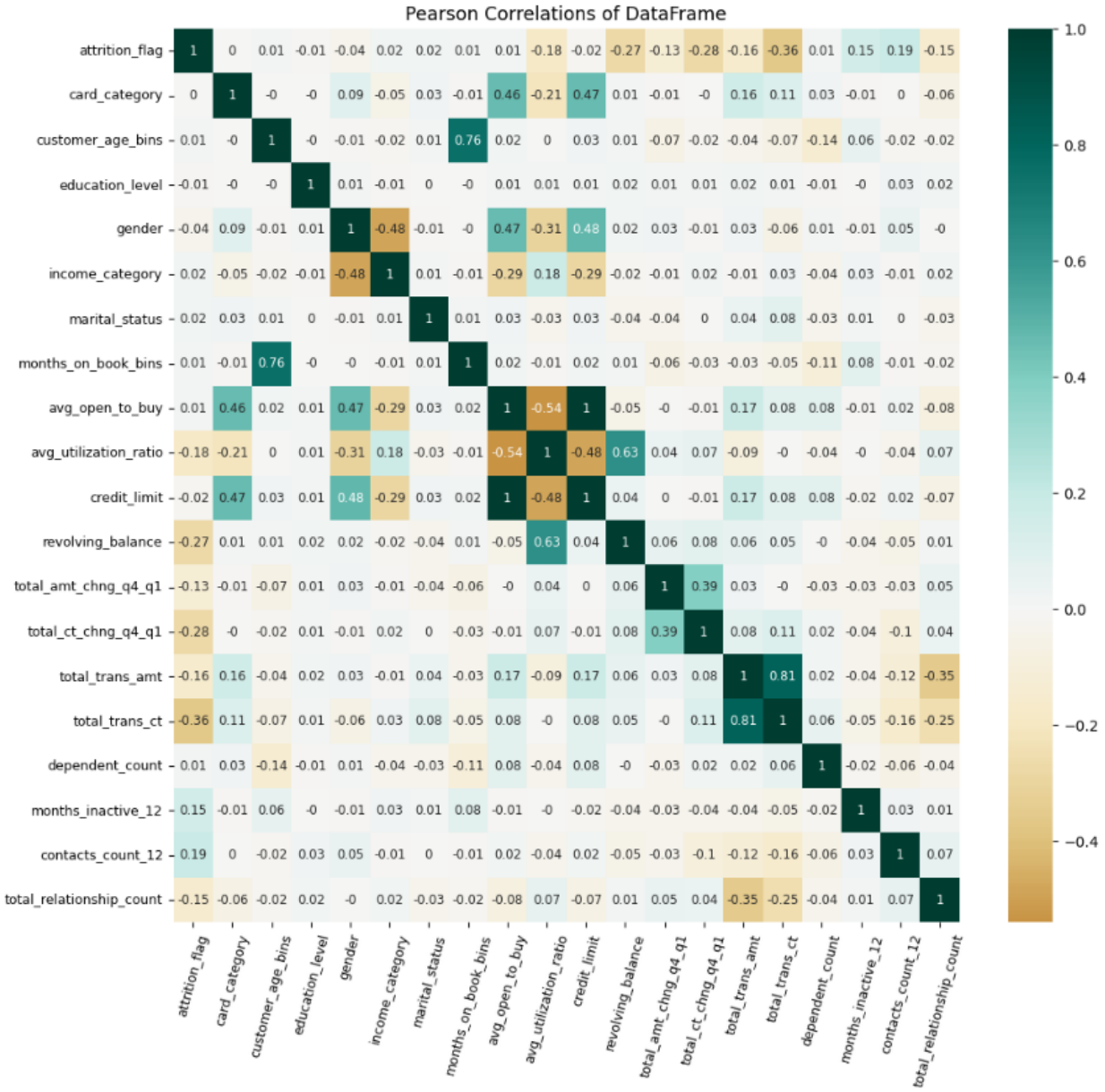
Steps included:

- Renaming columns
- Removing null values or rows containing *'Unknown'* values
- Variable encoding
- Turning some numerical columns into bins
- Displaying pie charts distribution for categorical columns
- Plotting histograms for numerical columns

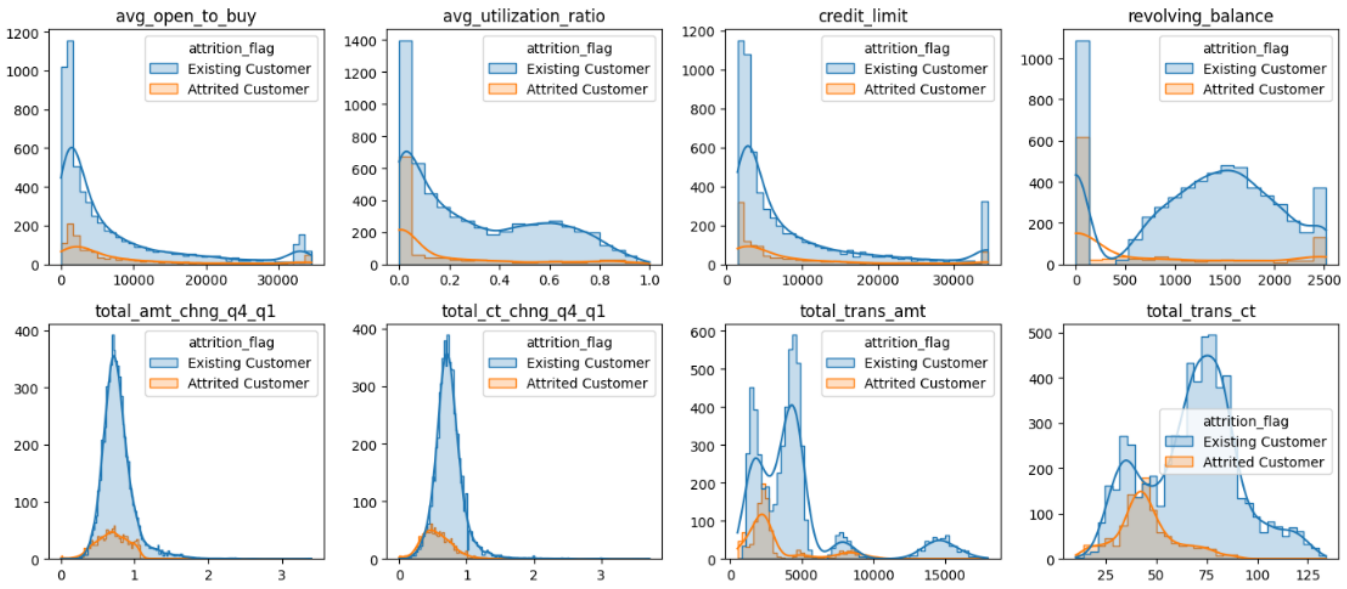
- Plotting correlations heatmap
- Plotting bars and histograms on variables in relation to the target column

	count	mean	std	min	25%	50%	75%	max
attrition_flag	7081.0	0.1572	0.3640	0.0	0.0000	0.0000	0.0000	1.0
card_category	7081.0	0.0601	0.2315	0.0	0.0000	0.0000	0.0000	1.0
customer_age_bins	7081.0	0.4254	0.1952	0.0	0.2857	0.4286	0.5714	1.0
education_level	7081.0	0.5153	0.2968	0.0	0.4000	0.4000	0.6000	1.0
gender	7081.0	0.5234	0.4995	0.0	0.0000	1.0000	1.0000	1.0
income_category	7081.0	0.6493	0.3458	0.0	0.2500	0.7500	1.0000	1.0
marital_status	7081.0	0.6680	0.3098	0.0	0.5000	0.5000	1.0000	1.0
months_on_book_bins	7081.0	0.5457	0.2138	0.0	0.4286	0.5714	0.7143	1.0
avg_open_to_buy	7081.0	0.2122	0.2646	0.0	0.0361	0.0941	0.2749	1.0
avg_utilization_ratio	7081.0	0.2826	0.2790	0.0	0.0260	0.1862	0.5155	1.0

	count	mean	std	min	25%	50%	75%	max
credit_limit	7081.0	0.2133	0.2759	0.0	0.0320	0.0861	0.2809	1.0
revolving_balance	7081.0	0.4638	0.3227	0.0	0.1839	0.5093	0.7076	1.0
total_amt_chng_q4_q1	7081.0	0.2239	0.0657	0.0	0.1852	0.2164	0.2526	1.0
total_ct_chng_q4_q1	7081.0	0.1916	0.0643	0.0	0.1570	0.1885	0.2202	1.0
total_trans_amt	7081.0	0.2222	0.1984	0.0	0.0903	0.1899	0.2419	1.0
total_trans_ct	7081.0	0.4395	0.1920	0.0	0.2742	0.4597	0.5645	1.0
dependent_count	7081.0	0.4676	0.2583	0.0	0.2000	0.4000	0.6000	1.0
months_inactive_12	7081.0	0.3904	0.1659	0.0	0.3333	0.3333	0.5000	1.0
contacts_count_12	7081.0	0.4091	0.1842	0.0	0.3333	0.3333	0.5000	1.0
total_relationship_count	7081.0	0.5639	0.3089	0.0	0.4000	0.6000	0.8000	1.0



- There is a strong positive correlation between 'total_trans_amt' and 'total_trans_ct', with a correlation coefficient of 0.81. This suggests that customers who make more transactions also tend to spend more, as the total transaction amount and total transaction move together.
- Most of the demographic variables seem to have a very low correlation with the target variable, with correlation coefficients ranging from -0.04 to 0.02. This suggests that the demographic variables may not be strong predictors of customer churn. However, it is worth noting that some demographic variables may still have an impact on customer churn that is not captured by the correlation analysis."
- The highest correlations with the target variable are with 'total_trans_ct', 'total_ct_chng_q4_q1', and 'revolving_balance', with correlation coefficients of -0.36, -0.28, and -0.27, respectively. These negative correlations suggest that customers who have more transactions, a higher change in their transaction count from the fourth quarter to the first quarter, and a higher revolving balance are less likely to churn.



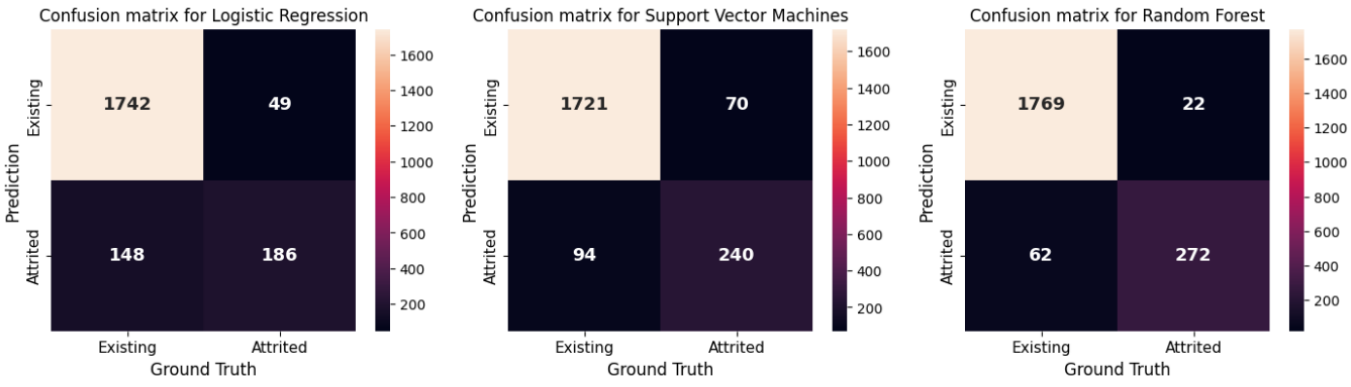
- The histograms show that a higher number of attrited customers are found on the left sides (mostly) where there's a higher density.
- We can see that for the 'Revolving_Balance' column, the proportion of attrited customers is higher for the lower values of the balance, indicating that customers with a lower balance are more likely to churn.
- Similarly, for the 'Total_Trans_Ct' column, the proportion of attrited customers is higher for the values between 10 and 75, indicating that customers with a moderate and low number of transactions are more likely to churn. The proportion of attrited customers decreases for high numbers of transactions.

Models

Hypertuned 3 classification models; plotting confusion matrix for each.

```
Best hyperparameters for Logistic Regression: {'C': 10, 'penalty': 'l2', 'solver': 'liblinear'}
Best hyperparameters for Support Vector Machines: {'C': 10, 'gamma': 'scale', 'kernel': 'poly'}
Best hyperparameters for Random Forest: {'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 150}
```

	accuracy	precision	recall	f1	auc	confusion_matrix
Logistic Regression	0.907294	0.791489	0.556886	0.653779	0.764764	[[1742, 49], [148, 186]]
Support Vector Machines	0.922824	0.774194	0.718563	0.745342	0.839739	[[1721, 70], [94, 240]]
Random Forest	0.960471	0.925170	0.814371	0.866242	0.901044	[[1769, 22], [62, 272]]



- For customer attrition prediction in banking, recall and precision are both important metrics to consider, as they can have different impacts on business outcomes.
- High recall is desirable because it means that the model is able to correctly identify as many at-risk customers (true positives) as possible, thereby reducing the number of customers who leave the bank without warning. This can help the bank take proactive measures to retain customers, such as offering them better incentives or improving their customer service experience.
- On the other hand, high precision is also important to ensure that the bank doesn't waste resources trying to retain customers who are not actually at risk of leaving (false positives).

Metrics

Next step was to implement oversampling, undersampling or class weighting to improve the recall/precision as a whole since we're dealing with an imbalanced set.

		accuracy	precision	recall	f1	auc	confusion_matrix
Original	Logistic Regression	0.904000	0.753906	0.577844	0.654237	0.771334	[[1728, 63], [141, 193]]
	Support Vector Machines	0.923294	0.802120	0.679641	0.735818	0.824187	[[1735, 56], [107, 227]]
	Random Forest	0.953412	0.918149	0.772455	0.839024	0.879807	[[1768, 23], [76, 258]]
SMOTE	Logistic Regression	0.851294	0.517308	0.805389	0.629977	0.832622	[[1540, 251], [65, 269]]
	Support Vector Machines	0.901647	0.642369	0.844311	0.729625	0.878325	[[1634, 157], [52, 282]]
	Random Forest	0.952941	0.858896	0.838323	0.848485	0.906320	[[1745, 46], [54, 280]]
ClassWeight	Logistic Regression	0.831529	0.478947	0.817365	0.603982	0.825768	[[1494, 297], [61, 273]]
	Support Vector Machines	0.872471	0.562624	0.847305	0.676225	0.862234	[[1571, 220], [51, 283]]
	Random Forest	0.952000	0.900000	0.781437	0.836538	0.882623	[[1762, 29], [73, 261]]
UnderSampling	Logistic Regression	0.658824	0.309268	0.949102	0.466519	0.776896	[[1083, 708], [17, 317]]
	Support Vector Machines	0.688471	0.331276	0.964072	0.493109	0.800573	[[1141, 650], [12, 322]]
	Random Forest	0.922824	0.685590	0.940120	0.792929	0.929859	[[1647, 144], [20, 314]]

- Random Forest looks to perform the best across all models.
- The most balanced scores for precision and recall are by using the Random Forest with SMOTE technique, which keeps both metrics close to 85%.
- With the Random Forest model using SMOTE, which has a balanced precision and recall score of around 85%, we can accurately identify the customers who are at high risk of churning. This can be incredibly valuable for the bank because they can develop targeted retention strategies for these customers. In addition, the model can help the bank save resources by allowing them to focus their retention efforts on the customers who are most likely to churn. This can be much more efficient than trying to retain all customers, which can be a resource-intensive process.

Key findings and insights

- 30% of the dataset has been cut back by removing 'unknown' rows.
- Demographic variables may not be strong predictors of customer churn.
- Most important features in predicting the target column are 'total_trans_ct', 'total_ct_chng_q4_q1', and 'revolving_balance'.
- Best scores for precision and recall are by using the Random Forest model with SMOTE technique

Suggestions

- Imputation or collection of additional data. One limitation of the current analysis is the missing values which cut back on ~30% of the dataset.
- Consider re-visiting the analysis after obtaining additional data or engineering more features, and see if the performance of the model improves.
- Transforming the data to a normal distribution can improve the performance of the model. This is especially true for some parametric models, such as logistic regression.
- Refining the modeling approach. Other models could be explored or grid search could be utilized with different hyperparameters to fine-tune the model and potentially improve its performance.