Churn Prediction Model: Analysis and Report

Contents

Title: Churn Prediction Model

Domain: Retail

Introduction

Customer Churn is a process in which the customer no longer wants to remain in the system.

Customer churn, or the rate at which customers stop doing business with a company, is a critical metric for businesses, particularly those with recurring revenue models. High churn rates can significantly impact a company's revenue and long-term growth. Understanding and predicting which customers are at risk of leaving is essential for businesses to take proactive measures to retain them.

This report presents a predictive model developed to forecast customer churn based on historical customer data. Additionally, the EDA highlights the factors contributing to customer churn and identifies areas for potential improvement. The model identifies key factors contributing to churn and provides actionable insights to help the business implement retention strategies. The primary objective is to enhance customer retention, improve lifetime value, and reduce churn-related losses by accurately predicting which customers are likely to leave

Problem Statement

The goal is to build a customer churn prediction model to identify which customers will likely stop purchasing from a business.

Approaches

Tools Used: Python

Packages: Pandas, NumPy, matplotlib, Seaborn, Scikit-Learn, Sklearn

Data Collection

The dataset utilized for this project was provided by the "**Guvi team"**. It includes various attributes related to customer demographics, transaction history, and engagement metrics. This comprehensive dataset enables a thorough analysis of customer behavior and the identification of factors influencing churn. Below is the data description of the Churn Dataset.

Dataset Description

- CLIENTNUM - Client number, Unique identifier for the customer holding the account
- Attrition_Flag - Internal event (customer activity) variable - if the account is closed then 1 else 0
- Customer_Age -Demographic variable - Customer's Age in Years
- Gender - Demographic variable - M=Male, F=Female
- Dependent_count - Demographic variable - Number of dependents
- Education_Level - Demographic variable - Educational Qualification of the account holder (example: high school)
- Marital_Status - Demographic variable - Married, Single, Divorced, Unknown
- Income_Category - Demographic variable - Annual Income Category of the account holder
- Card_Category - Product Variable - Type of Card (Blue, Silver, Gold, Platinum)
- Months_on_book - Period of relationship with bank
- Total_Relationship_Count-Total no. of products held by the customer
- Months_Inactive_12_mon - No. of months inactive in the last 12 months

- Contacts_Count_12_mon-No. of Contacts in the last 12 months
- Credit_Limit-Credit Limit on the Credit Card
- Total_Revolving_Bal-Total Revolving Balance on the Credit Card
- Avg_Open_To_Buy-Open to Buy Credit Line (Average of last 12 months)
- Total_Amt_Chng_Q4_Q1- Open to Buy Credit Line (Average of last 12 months)
- Total_Trans_Amt-Total Transaction Amount (Last 12 months)
- Total_Trans_Ct- Total Transaction Count (Last 12 months)
- Total_Ct_Chng_Q4_Q1-Change in Transaction Count (Q4 over Q1)
- Avg_Utilization_Ratio-Average Card Utilization Ratio

Data Preprocessing

The preprocessing phase involved several critical steps to prepare the dataset for analysis and modeling

1. **Data Description**: Initially, I examined the dataset to understand the various data types and their distributions. This step helped in identifying the nature of each variable and informed subsequent preprocessing actions.
2. **Statistical Summary**: A descriptive analysis of the dataset was conducted to compute key statistics, including mean, median, and other relevant metrics. This analysis provided insights into the central tendencies and variability within the data.
3. **Dimensionality Check**: I assessed the dataset for its structure by checking the number of rows and columns. This ensured clarity on the dataset's size and scope for analysis.
4. **Unique Values Assessment**: An evaluation of the unique values in categorical columns was performed to understand the diversity of categories within the dataset, which is crucial for effective feature engineering.
5. **Missing Values Identification**: Using the **isnull().sum() function**, I identified and confirmed the absence of any null values in the dataset, ensuring that data integrity was maintained for analysis.
6. **Duplicate Records Check**: Finally, I utilized **the Churn.duplicated().sum()** function to identify any duplicate entries within the dataset. This step was essential to maintain the accuracy of the analysis, and I found that there were no duplicate records present.

These preprocessing steps ensured that the dataset was clean and ready for the modelling phase, setting a solid foundation for effective churn prediction.

Feature Engineering

In the feature engineering phase, several new features were created, and existing ones were transformed to enhance the model's predictive power

1. **Income Column Creation**: A new numerical column, **Income**, was derived from the existing **Income Category**. The income ranges were mapped to specific values as follows:
    - 'Less than $40K': 20,000
    - '$40K - $60K': 50,000
    - '$60K - $80K': 70,000
    - '$80K - $120K': 100,000
    - '$120K +': 140,000

- 'Unknown': None

To handle the 'Unknown' category, I utilized the **fillna()** function to replace these values with the median income, ensuring that missing data did not adversely impact the analysis.

2. **Age Categorization**: The age column was categorized into defined bins: '18-30', '31-40', '41-50', '51-60', '61-70', and '71-80'. This transformation allows for easier analysis of customer segments based on age groups.
3. **Credit Utilization Ratio**: A new feature, **Credit Utilization Ratio**, was calculated by dividing the **Total_Revolving_Bal** by the **Credit_Limit**. This ratio serves as a significant indicator of financial health.
4. **Relationship Count**: An aggregated feature was created by subtracting the **Months_Inactive_12_mon** from **Total_Relationship_Count**. This feature aims to capture the overall engagement of a customer with the company.
5. **Attrition Flag Transformation**: The **Attrition Flag** was replaced with binary values (0 and 1) to facilitate model training.
6. **Label Encoding**: Categorical variables such as **Gender**, **Education Level**, **Marital Status**, and **Card Category** were converted into a numerical format using label encoding, enabling the model to effectively interpret these features.
7. **Dropping Irrelevant Features**: Finally, irrelevant features were removed from the dataset to streamline the analysis. The following columns were dropped:
    - 'Income_Category'
    - 'CLIENTNUM'
    - 'Age_Category'
    - 'Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_1'
    - 'Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_2'

The cleaned dataset was updated accordingly with the command
Ch = Churn.drop(columns=columns_to_drop)

These feature engineering steps enhanced the dataset's quality and relevance, contributing to more accurate predictions in the churn model.

EDA (Exploratory Data Analysis)

Exploratory Data Analysis (EDA)

The Exploratory Data Analysis (EDA) phase provided valuable insights into the dataset, allowing for a deeper understanding of customer behaviour and churn patterns.

The following analyses and visualizations were performed:

1. **Customer Attrition Distribution**: A pie chart was utilized to visualize the distribution of customer attrition, which shows
    - **16**.07% of the customers are '**Attrited Customers**"
2. **Customer Age Distribution**: A histogram was created to analyze the age distribution of customers, revealing trends and identifying age groups that may be more susceptible to churn.
3. **Correlation Analysis**:
    - A correlation matrix was generated to assess relationships among numerical variables, helping identify potential multicollinearity issues.

    - The correlation between target variables was also examined, providing insights into the factors influencing churn.

4. **Relationship Analysis**:
    - A box plot illustrated the relationship between age and churn, highlighting age groups with higher churn rates.
    - Another box plot analysed the relationship between **Total_Trans_Amt** and churn, identifying spending patterns of customers who have churned.
5. **Categorical Feature Distribution**: A bar graph depicted the distribution of categorical features such as **Gender** and **Card Category**, providing insights into customer demographics and product preferences.
6. **Attrition Analysis by Education Level and Marital Status**:
    - The attrition count was analysed within different education levels, revealing which categories were most affected.
    - Similar analysis was conducted for marital status, identifying the gender most impacted by attrition.
7. **Income Analysis**: The average income was calculated by gender, highlighting financial trends among different demographic groups.
8. **Product Analysis**:
    - An analysis was performed to determine which products customers were most likely to abandon, providing insights into customer preferences and potential areas for improvement.
    - The analysis also explored the maximum relationship period customers maintained with the bank, revealing insights into customer loyalty.
9. **Products Held by Gender**: The total number of products held by customers was compared against gender, providing insights into product distribution and preferences.
10. **Card Category Analysis**:
    - A bar plot illustrated the count of products held by customers across different card categories, helping identify popular products.
    - Additionally, the analysis explored which gender preferred which product within these categories.
11. **Credit Limit Analysis**:
    - An average credit limit was plotted against average age, revealing financial behaviours across age demographics.
    - The distribution of credit limits was further examined using a box plot to identify any outliers or trends.
12. **Inactive Customers**: The analysis also included an examination of customer inactivity, counting the number of customers who were inactive in the last month, highlighting potential retention challenges.

The insights gained during this phase are crucial for developing effective strategies to mitigate customer churn.

Model Building

The **"logistic regression model"** was built to predict customer churn.

Below are the detailed steps undertaken for model development:

Importing Libraries:

- Several essential libraries were imported from sklearn to facilitate data splitting, feature scaling, and model evaluation.

Feature Selection:

- Based on the dataset and exploratory data analysis, important features were selected to build the model. These features were believed to have a significant impact on customer churn.

Data Splitting:

- The dataset was split into training and testing sets in a **70:30** ratio. This allowed the model to learn from **70%** of the data while keeping **30%** for performance evaluation.

Feature Scaling:

- Feature scaling was applied using **Standard Scaler** to standardize the numerical features, ensuring that each feature contributed equally to the model's learning. This step was necessary because logistic regression assumes that the input features are standardized

Model Training:

- The logistic regression model was initialized and trained using the scaled training data. The fit() method was used to train the model on the selected features.

Model Prediction:

- After training the model, predictions were made on the test data using the **predict() method**. These predictions were compared against the actual values to evaluate model performance.

Model Evaluation:

- The model's performance was evaluated using various classification metrics:
  - **Accuracy**: Measured the proportion of correct predictions.
  - **Confusion Matrix**: Provided insights into true positives, true negatives, false positives, and false negatives.
  - **Classification Report**: Included key metrics such as precision, recall, and F1-score to offer a comprehensive view of the model's performance.

To enhance the churn prediction model, a "**Random Forest classifier"** was implemented. Random Forest is a robust and versatile algorithm, well-suited for handling complex datasets, and it helps improve prediction accuracy.

Below are the key steps followed in building and evaluating the Random Forest model:

Importing Libraries:

  - Several essential libraries were imported to handle data splitting, scaling, model training, and evaluation, as well as hyperparameter tuning for Random Forest.

Setting up Hyperparameters:

- A range of hyperparameters was defined to optimize the Random Forest model, including:
  - n_estimators: Number of trees in the forest.
  - max_depth: Maximum depth of the tree.
  - min_samples_split: Minimum number of samples required to split a node.
- These hyperparameters were set up for tuning via GridSearchCV, which helps in identifying the best combination of parameters.

Model Initialization and Fitting:

- The Random Forest classifier was initialized, and grid search was employed to identify the best parameters by evaluating the model with cross-validation on the training dataset.

Model Prediction:

- Using the best parameters obtained from the grid search, the final Random Forest model was trained and predictions were made on the test data.

Model Evaluation:

- The performance of the model was evaluated using various metrics:
  - **Accuracy**: The overall correctness of the predictions.
  - **Confusion Matrix**: To evaluate the true positives, true negatives, false positives, and false negatives.
  - **Classification Report**: A detailed report that included precision, recall, and F1-score for both classes (churn and non-churn).

Feature Selection and Importance:

- After training the model, the importance of each feature in predicting churn was determined using the feature importances_ attribute. This allowed for identifying the most influential features in the dataset.

Feature Importance Plot:

- To better visualize the feature's importance, a bar plot was created, showcasing the most influential features in the model. This helped in understanding which factors played a key role in predicting customer churn.

Findings and Insights

Model Evaluation: Logistic Regression Model

After building and evaluating the "**Logistic Regression Model"**, below are the key insights derived from the model's performance

Confusion Matrix Interpretation

- The model correctly predicted **2,463** customers who did not churn (True Negatives) and **273** customers who churned (True Positives).
- However, **223** customers who churned were incorrectly classified as non-churners (False Negatives), and **80** customers who did not churn were incorrectly classified as churners (False Positives).
- This highlights that while the model performs well in identifying non-churners

Precision, Recall, and F1-Score:

- For **non-churners (class 0)**:
  - The precision is **0.92**, indicating that 92% of customers predicted not to churn they did not churn.
  - The recall is **0.97**, which means the model successfully identified 97% of the actual non-churners.
  - The F1-score is **0.94**, showing a strong balance between precision and recall.
- For **churners (class 1)**:
  - The precision is **0.77**, meaning that 77% of customers predicted to churn and they churned.

    - The recall is **0.55**, indicating that the model only identified 55% of actual churners, which suggests it is missing many true churners.
    - The F1-score is **0.64**, which reflects a moderate performance in predicting churners.
- These results suggest that the model is better at predicting non-churners than churners, which could lead to missed opportunities in retaining customers who are likely to churn.

Accuracy:

- The overall accuracy of the model is **90%**, which indicates that 90% of the predictions (both churn and non-churn) are correct.

Macro Average vs. Weighted Average:

- The **macro average** precision, recall, and F1-score (0.85, 0.76, 0.79 respectively) provide an unweighted average for both classes, highlighting the performance gap between predicting churners and non-churners.
- The **weighted average** precision, recall, and F1-score (0.89, 0.90, 0.89 respectively) take into account class imbalance and suggest that the model's overall performance is strong, but improvements can be made in predicting churners more effectively

Model Evaluation: Random Forest

After building and evaluating the "**Random Forest model"**, below are the key insights derived from the model's performance

The performance of the Random Forest model was evaluated using the **confusion matrix, classification report, and accuracy score**. The results are as follows

Confusion Matrix Interpretation:

- The model correctly predicted **2,510** customers who did not churn (True Negatives) and **412** customers who churned (True Positives).
- Only **33** non-churning customers were incorrectly classified as churners (False Positives), while **84** churning customers were incorrectly classified as non-churners (False Negatives)
- This shows that the Random Forest model performs well in distinguishing between customers who churn and those who do not, with a low false positive and false negative rate.

Precision, Recall, and F1-Score:

- For **non-churners (class 0)**
    - **Precision** is **0.97**, meaning that **97%** of the customers predicted not to churn, they did not churn.
    - **Recall** is **0.99**, the model correctly identifies that **99%** of all non-churners.
    - The **F1-score** is **0.98**, indicating a strong balance between precision and recall for this class.
- For **churners (class 1)**:
    - **Precision** is **0.93**, indicating that 93% of the customers predicted to churn and they churned.
    - **Recall** is **0.83**, meaning the model identifies 83% of all actual churners, which is quite robust.
    - The **F1-score** is **0.88**, showing that the model effectively balances precision and recall when predicting churners.
    - These results suggest the **Random Forest model** is highly accurate in predicting non-churners while also significantly improving the prediction of churners compared to other models.
- **Accuracy**:

    - The overall accuracy of the model is **96.15%**, indicating that the Random Forest correctly predicts the outcome for 96% of the customers. This high accuracy makes it a very reliable model for predicting customer churn.
- **Macro Average vs. Weighted Average**:
    - The **macro average** precision, recall, and F1-score are **0.95**, **0.91**, and **0.93**, respectively, suggesting that the model performs well across both classes, with a slight imbalance in identifying churners.
    - The **weighted average** precision, recall, and F1-score are all **0.96**, indicating that the model performs well even in the presence of class imbalance, where there are more non-churners than churners.
- **Feature Importance**:
    - After evaluating the model, feature importance analysis was conducted to identify which features contributed most to the prediction of customer churn.
    - Key features such as **Total_Trans_Amt**, **Credit Limit**, and **Total_Revolving_Bal** were found to have the most significant impact on predicting churn, suggesting that customer transaction **behavior and credit utilization** are strong indicators of churn risk.

Recommendations

Based on the insights gained from the churn prediction model and the feature importance analysis, the following recommendations are suggested to reduce customer churn

- **Enhance Customer Engagement**
- **Improve Credit Utilization and Support**
- **Focus on Long-Term Customers**
- **Monitor Customer Inactivity**
- **Segment-Specific Strategies**
- **Optimize Customer Support for Higher-Risk Groups**
- **Expand Product Offerings and Upsell Opportunities**
- **Improve Service for Customers with High Credit Limit Usage**

Source Code: Github