CIIT College of Arts and Technology

Predicting Bank Customer Churn Using Data Mining Techniques: A Case Study with Altair AI Studio

Submitted by:
Babala, Patrick Miguel M.
Student ID: 17-22-3831
Denzon, Christian
Student ID: 17-22-3707

Bachelor of Science in Computer Science
Major in Software Engineering
3rd Semester, School Year 2024-2025

Submitted to:
Sir Rizaldy Rapsing

Table of Contents

1. PROBLEM DESCRIPTION

Customer churn refers to the situation wherein clients cease utilizing a bank's services. This study focuses specifically on analyzing churn behavior among credit card service users. Understanding the reasons behind churn can help financial institutions reduce attrition rates and improve customer retention. This project aims to analyze customer data and build predictive models to identify potential churners. By discovering key patterns, the bank can proactively implement strategies to keep valuable clients.

2. DATASET DESCRIPTION

The dataset used in the project was retrieved from Kaggle and it is named as BankChurners.csv, which contains demographic and transactional information of 10,128 customers. The dataset includes both numerical and categorical features relevant to customer behavior and bank usage.

- **Total Records:** 10,128 records (downsized to 10,000 due to AI Studio constraints)
- **Target Attribute:** Attrition_Flag (1 = Churned, 0 = Existing Customer)

2.2. Key Attributes

Customer_Age represents the age of the customer and provides demographic context. **Gender, Dependent_count, Education_Level, Marital_Status,** and **Income_Category** reflect the customer's socio-demographic profile, which may influence banking behavior. **Credit_Limit** and **Avg_Open_To_Buy** offer insight into the customer's financial capacity and available credit. Behavioral engagement is captured through **Total_Relationship_Count** and **Total_Revolving_Bal**, indicating the depth of their relationship with the bank and their credit utilization. Lastly, **Total_Trans_Amt** and **Total_Trans_Ct** reflect spending patterns, while T**otal_Ct_Chng_Q4_Q1** shows the change in transaction count between two quarters, signaling any sudden shifts in activity that could indicate churn risk.

3. DATA EXPLORATION

3.1. Target Attribute

The **Attrition_Flag** represents whether the customer has churned or not. It is a binary categorical attribute and serves as the label for classification.

3.2. Regular Attributes

The dataset includes transactional patterns (**Total_Trans_Amt, Total_Trans_Ct**), behavioral trends (**Total_Ct_Chng_Q4_Q1, Total_Relationship_Count**), and credit information (**Credit_Limit, Total_Revolving_Bal)**.

3.3. Descriptive Statistics

- **Average Age**: ~46 years
- The majority of customers are **married** with a **college education**.
- **Churn Rate**: Approximately **16.1%** of customers have churned in the dataset.

3.4. Problem Insights

Initial **EDA** (exploratory data analysis) shows churned customers tend to have lower transaction frequency and fewer relationships with the bank. Attributes like **Total_Trans_Ct** and **Total_Ct_Chng_Q4_Q1** show strong correlation with churn behavior.

4. DATA VISUALIZATION

4.1. Churn Distribution

The churn distribution graph shows a significant class imbalance: most customers are non-churners, while a smaller portion represents churned customers. This highlights the need for careful model evaluation to avoid bias toward the majority class.

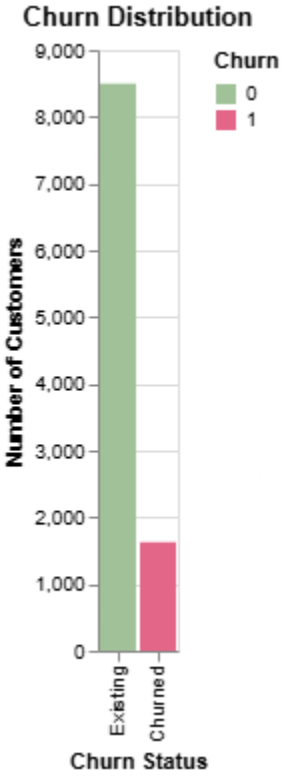

Figure I. Churn Distribution Bar Graph (Google Colab, 2025)

4.2. Average Transaction Count By Churn Status

The average transaction count by churn status graph shows that customers who did not churn have a significantly higher average number of transactions compared to those who churned. This suggests that active engagement through frequent transactions is a strong indicator of customer retention.

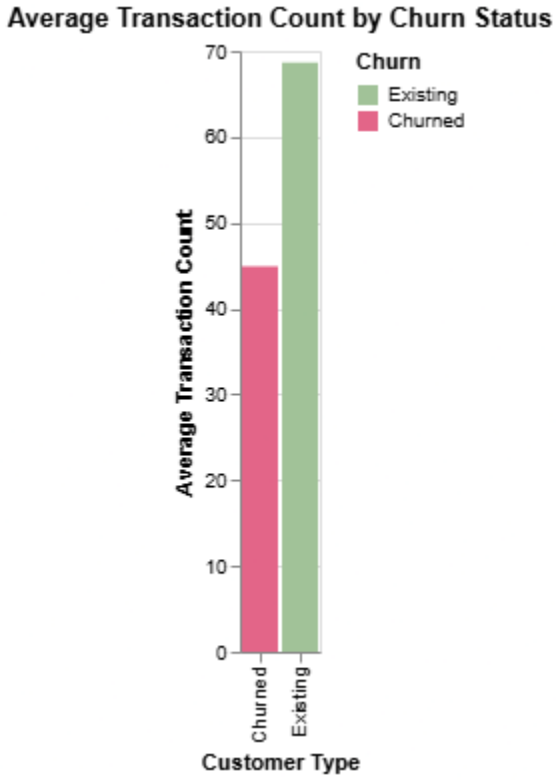

Figure II. Average Transaction Count By Churn Status Bar Graph (Google Colab, 2025)

4.3. Total Relationship Count by Churn Status

Churned customers typically have fewer relationships with the bank than loyal ones — helping highlight customer engagement as a key churn factor.

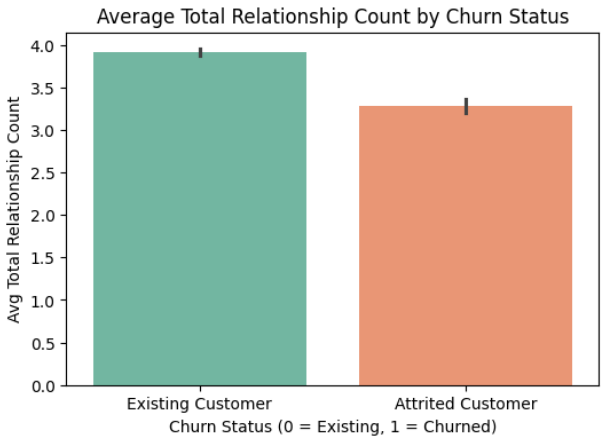

Figure III. Total Relationship Count by Churn Status Bar Graph (Google Colab, 2025)

5. DATA PREPARATION

Data preparation was a critical phase to ensure the dataset's readiness for effective model training and evaluation. The original dataset, sourced from a financial institution, contained customer information relevant to churn behavior, including transaction frequency, account activity, demographic data, and product usage. This section outlines the specific steps applied in **Altair AI Studio** to clean and structure the data, namely: attribute reduction, normalization, handling of data types, and outlier detection.

5.1. Attribute Reduction

To improve model performance and reduce complexity, attribute reduction was performed by identifying and retaining only the most relevant features. Initially, the dataset contained over 20 variables. Attributes that showed high correlation with each other or demonstrated little to no variance (e.g., columns with constant values) were removed. For instance, fields such as **CLIENTNUM** and **Naive_Bayes_Classifier_Attrition_Flag** were excluded due to redundancy or lack of predictive power. The final set of attributes included Total_Trans_Ct, Total_Trans_Amt, Customer_Age, Credit_Limit, and Contacts_Count_12_mon, which were statistically and visually validated to have meaningful influence on customer churn.

5.2. Normalization

Normalization was applied to scale continuous variables to a comparable range, particularly for models sensitive to input magnitude (e.g., Logistic Regression). Variables such as **Total_Revolving_Bal, Avg_Open_To_Buy,** and **Total_Relationship_Count** were normalized using **min-max scaling** within Altair AI Studio. This process ensured uniform feature

influence during the training process and improved algorithm convergence speed.

5.3. Data Types

Each variable was reviewed and corrected for proper data type assignment. Continuous numerical values such as **Total_Trans_Amt, Total_Trans_Ct,** and **Credit_Limit** were kept as floating-point types. Categorical variables like **Gender, Marital_Status,** and **Card_Category** were encoded using appropriate label encoders within Altair's preprocessing tools. The target variable **Attrition_Flag** was also converted into a binary format (1 for churn, 0 for non-churn) to facilitate supervised learning.

5.4. Outliers

Outlier analysis was conducted to identify and manage extreme values that could distort model learning. Boxplots and statistical thresholds (e.g., IQR method) were utilized to flag outliers in features like **Total_Trans_Amt** and **Months_on_book**. For example, a small number of customers had transaction amounts significantly above the 95th percentile. Instead of outright removal, these outliers were capped (winsorized) to preserve data integrity while reducing their influence. This approach maintained the dataset's natural distribution without skewing the model.

6. MODEL SELECTION

In order to effectively predict customer churn, several classification models were considered and evaluated using **Altair AI Studio**. Model selection focused on identifying algorithms that are not only capable of handling structured data with both numerical and categorical features but also robust in interpreting non-linear patterns present in customer behaviors. This section discusses the types of models tested, the rationale behind choosing specific classification models, and a comparative evaluation based on key performance metrics such as **accuracy, precision, recall,** and **F1 score**.

6.1. Types of Models

The primary goal of this study was to identify the most suitable classification algorithms for churn prediction. Given the binary nature of the target variable (*Attrition_Flag*), supervised learning methods were used. Three models were selected for experimentation based on their popularity in classification tasks, interpretability, and predictive performance: **Logistic Regression, Random Forest Classifier,** and **Gradient Boosted Trees**. Each of these models was trained and evaluated using the preprocessed dataset in Altair AI Studio.

6.2. Classification Models

6.2.1. Logistic Regression

Logistic Regression was selected as the baseline model due to its simplicity and interpretability. This algorithm is well-suited for binary classification problems and provides probabilistic outputs that aid in understanding the likelihood of churn. Logistic Regression assumes a linear relationship between the independent variables and the log-odds of the target class. Despite this limitation, it performed reasonably well on the dataset, offering insight into how each feature influences churn. Additionally, it is computationally efficient and serves as a solid benchmark for evaluating more complex models.

6.2.2. Random Forest Classifier

The Random Forest Classifier was chosen for its ability to handle high-dimensional data and capture complex, non-linear relationships. It operates by constructing an ensemble of decision trees and aggregating their predictions. This reduces overfitting and increases generalizability. In this project, the Random Forest model demonstrated strong performance across multiple metrics. It also provided valuable feature importance rankings, which helped in identifying the most significant predictors of customer churn, such as **Total_Trans_Ct, Total_Trans_Amt,** and **Customer_Age**. Its robustness and high accuracy made it a strong candidate for deployment.

6.2.3. Gradient Boosted Trees

Gradient Boosted Trees were included due to their proven success in many Kaggle competitions and practical machine learning applications. This model builds trees sequentially, with each new tree correcting the errors made by the previous ones. As a result, Gradient Boosted Trees can achieve higher accuracy than traditional ensemble methods but at the cost of longer training time. In this study, the model yielded the highest performance in terms of F1 score and balanced precision-recall. It was particularly effective in handling class imbalance and subtle churn indicators, which are critical in real-world churn analysis.

6.3. Model Evaluation

Each of the three models was trained and evaluated using a standard **80/20 train-test split** within Altair AI Studio. The evaluation metrics included **accuracy, precision, recall,** and **F1-score** to ensure a balanced assessment of both false positives and false negatives. Among the three, **Gradient Boosted Trees** achieved the best overall performance, closely followed by **Random Forest**, while **Logistic Regression** performed well but was slightly limited by its **linear assumptions**. The comparative results

helped in selecting the most effective model for the final prediction task and highlighted the trade-offs between model complexity and interpretability.

6.4. Model Visualization

Among the evaluated models, Gradient Boosted Trees performed the best, achieving an **accuracy** of 98.22%, **high precision** (94.73%), and **high recall** (94.02%), with an **AUC** of 0.997. While Random Forest also demonstrated strong precision, its recall was significantly lower (52.27%), indicating it missed more true positives. Logistic Regression provided a more balanced trade-off between precision and recall but underperformed compared to the tree-based models.

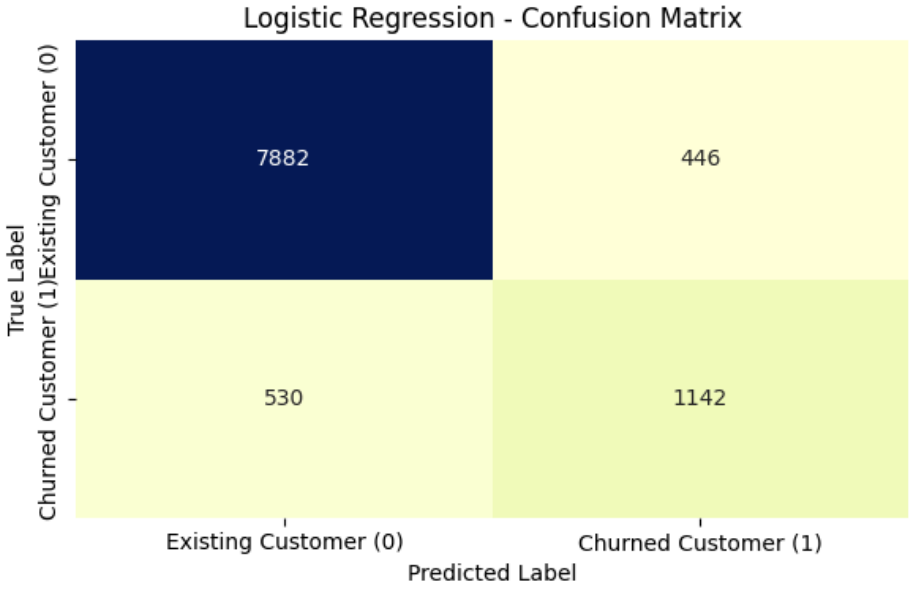

Figure IV. Logistic Regression Confusion Matrix (Google Colab, 2025)

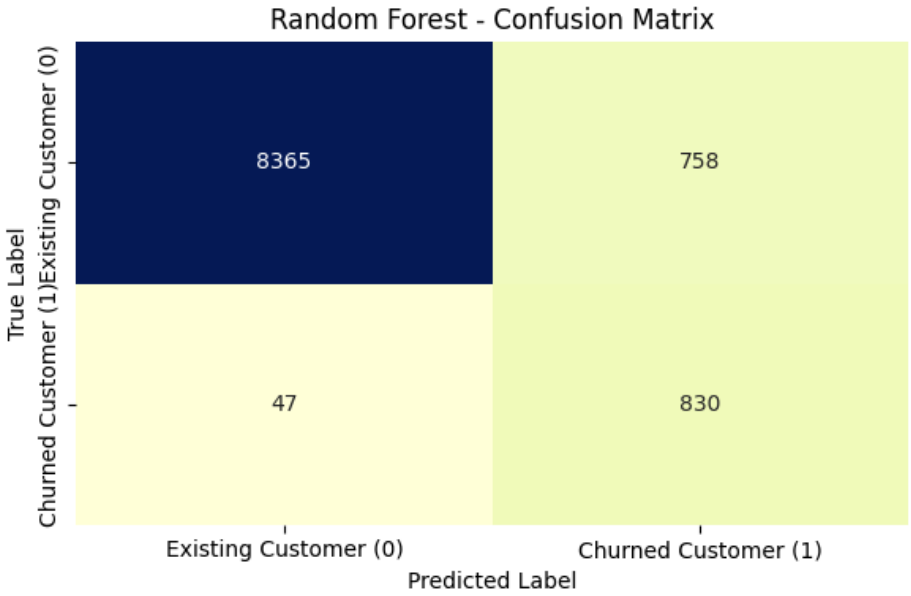

Figure V. Random Forest Confusion Matrix (Google Colab, 2025)

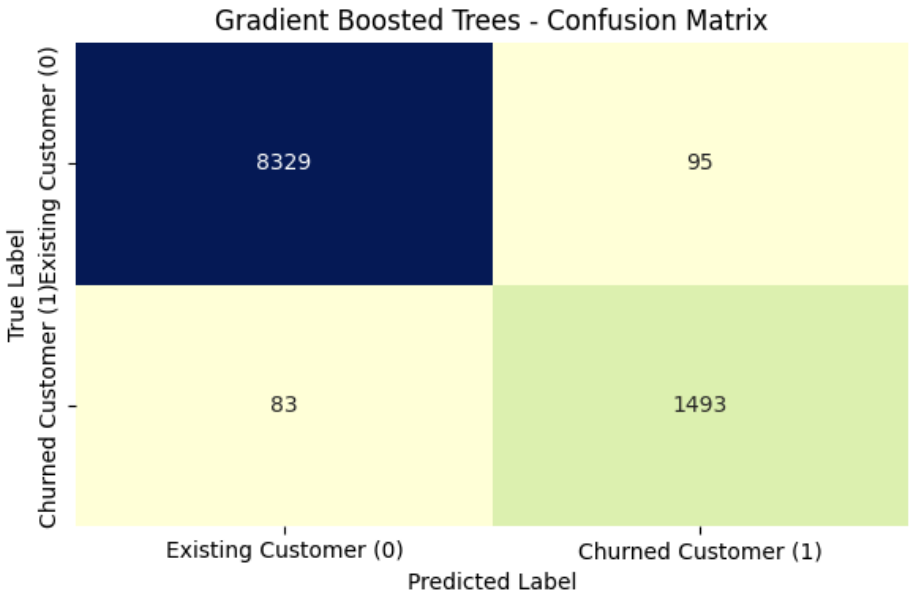

Figure VI. Gradient Boosted Trees Confusion Matrix (Google Colab, 2025)

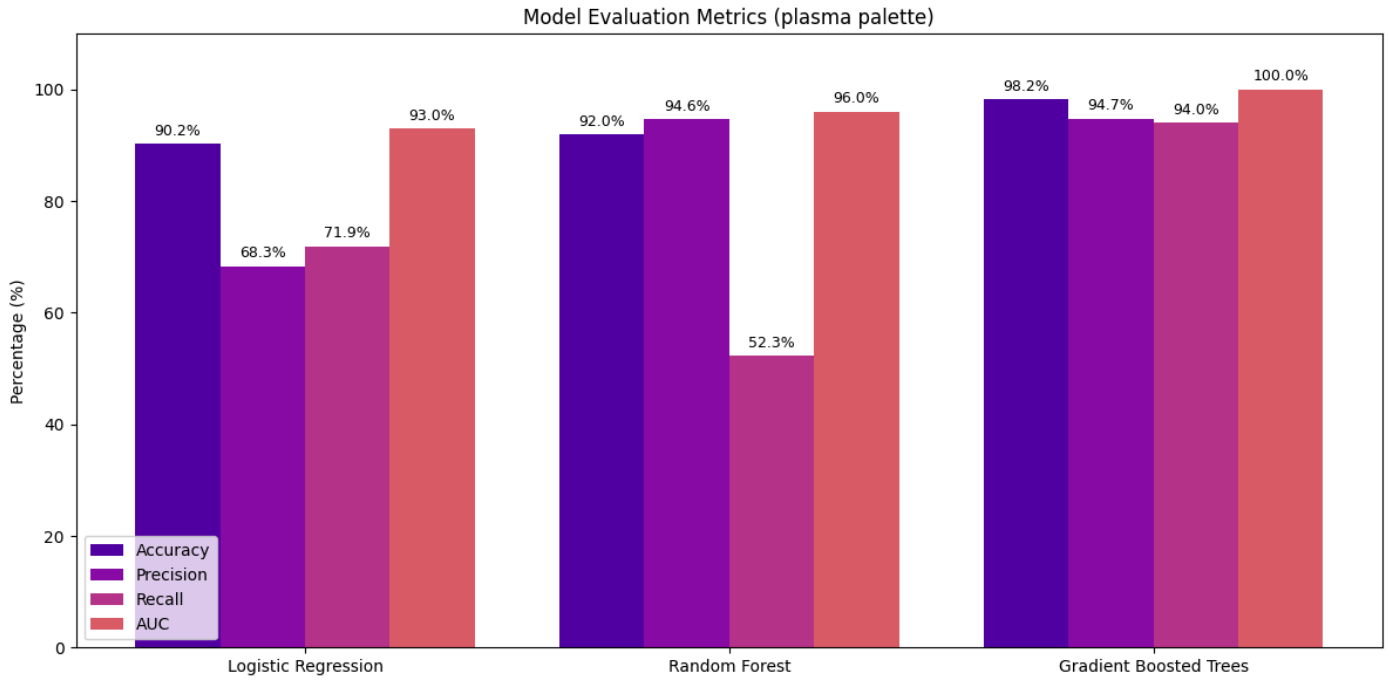

Figure VII. Model Evaluation Metrics of Logistic Regression, Random Forest, and Gradient Boosted Trees (Google Colab, 2025)

7. RAPIDMINER PROCESS FLOWS

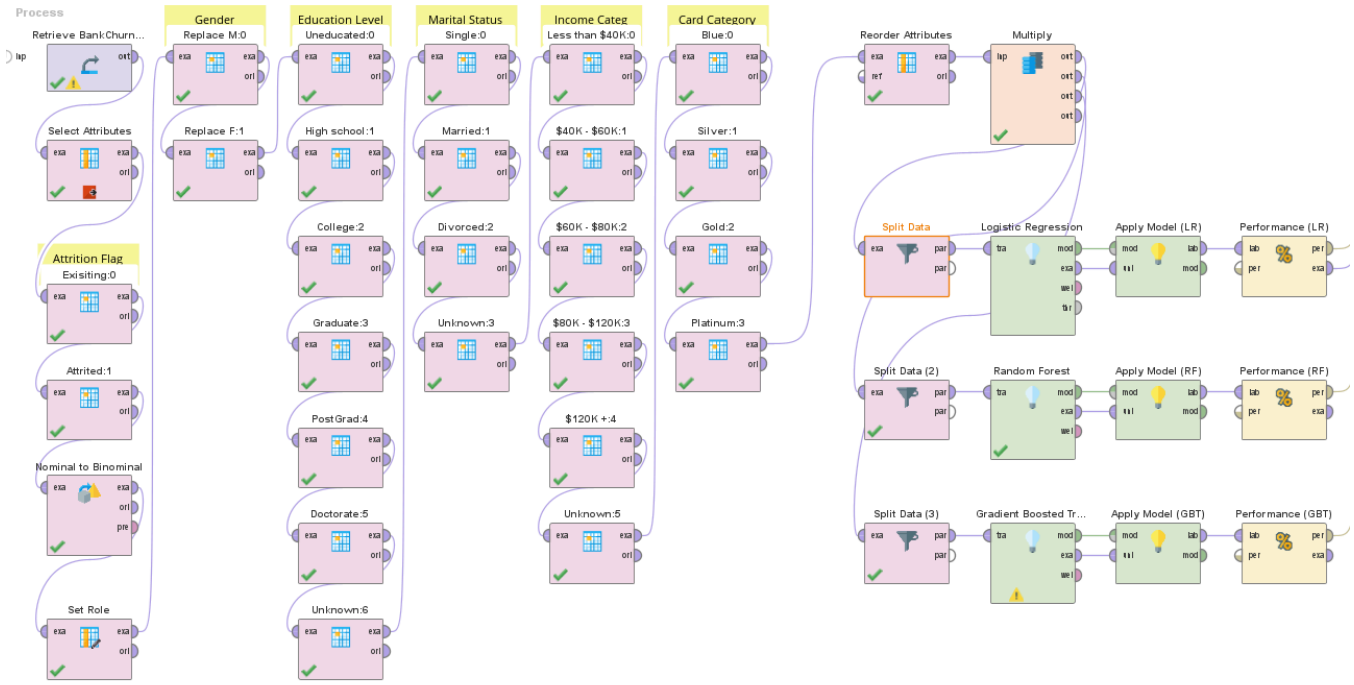

Figure VIII. (AI Studio)

7.1. Data Retrieval

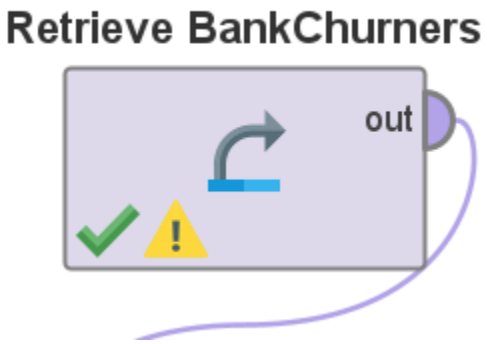

Figure VIII. (AI Studio)

The data used in this project was obtained from the BankChurners.csv file, which was imported into AI Studio. The imported file is then dragged onto the process for retrieval.

7.2. Select Attributes

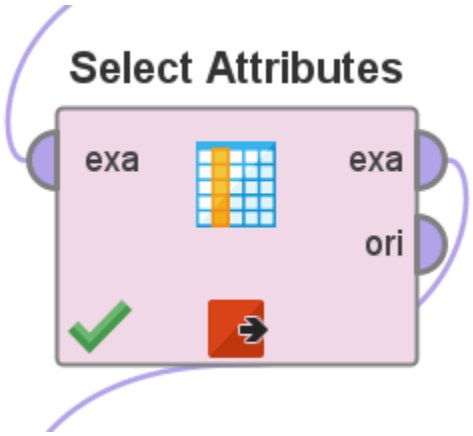

Figure IX. (AI Studio)

The **Select Attributes** operator was used to remove irrelevant attributes from the dataset, in order to make sure that only the relevant attributes are present in the model training.

7.3. Conversion of Attrition_Flag

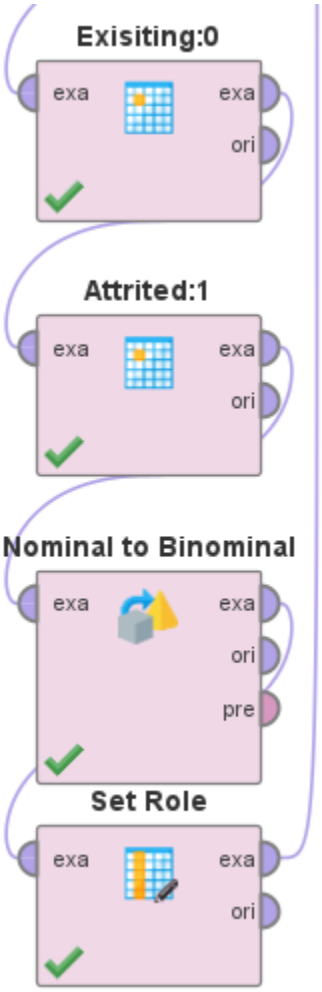

Figure X. (AI Studio)

Certain models such as the Logistic Regression model cannot handle categorical data. Before conversion, the *Attrition_Flag* attribute contained values such as *"Attrited Customer"* and *"Existing Customer"*, which are not compatible with these models. To solve this problem, the **Replace** operator is used to convert these values into binary format. The **Nominal to Binomial** operator is then used to change the classification type into binomial. Lastly, the **Set Role** operator is used to set the *Attrition_Flag* attribute to be the target attribute by setting it as the label.

7.4. Replace Value

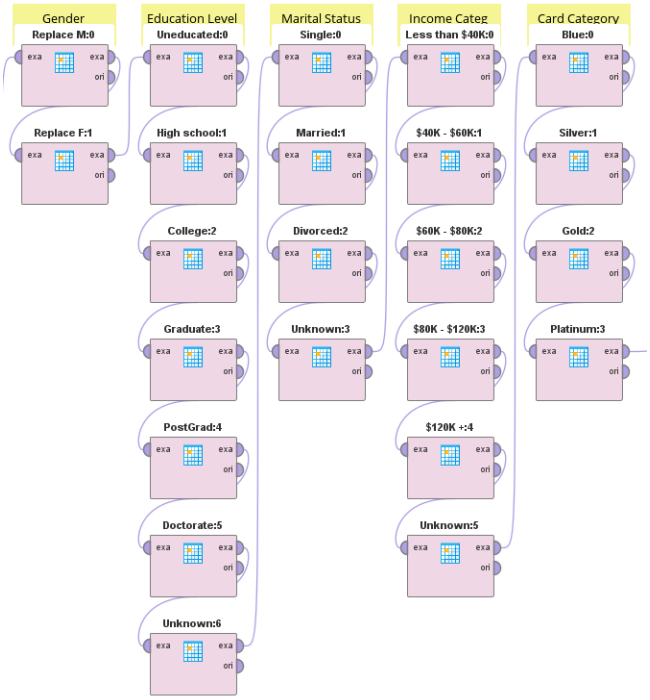

Figure XI. (AI Studio)

The dataset contains several categorical attributes such as Gender, Education Level, and Marital Status, which have text values that cannot be interpreted by machine learning models. The issue is addressed by using the **Replace** operator to convert the text values into numerical values.

7.5. Reorder Attributes and Multiply

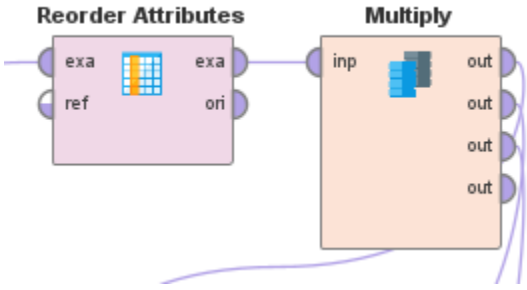

Figure XII. (AI Studio)

The **Reorder Attributes** operator is applied in the process to move the target attribute at the end of the dataset. This is done because models such as Logistic Regression and Random Forest expect the label/target attribute to be at the end of the dataset.

7.6. Data Splitting and Model Application

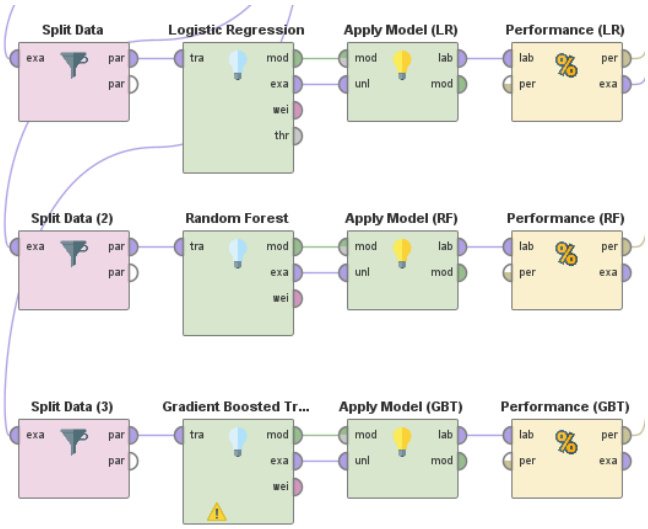

Figure XIII. (AI Studio)

The **Split Data** operator is used with a 80/20 ratio to divide the dataset into training and testing subsets. 80% of the data will be for training and the remaining 20% will be for performance evaluation. The three chosen models: **Logistic Regression**, **Random Forest**, and **Gradient Boosted Trees**, are also applied in this section. To check effectiveness, the **Performance** operator is used at the end of the process to check performance metrics.

8. EXPERIMENTS AND RESULTS

The performance of the three selected models for this project is evaluated using confusion matrices, which are generated in AI Studio by using the **Performance** operator. The matrices provide detailed information of the performance of each model when it comes to predicting customer churn by summarizing the number of correct and incorrect results.

8.1. Logistic Regression Model

accuracy: 90.24%

	true 0	true 1	class precision
pred. 0	7882	446	94.64%
pred. 1	530	1142	68.30%
class recall	93.70%	71.91%	

Figure XIV. (AI Studio)

The Logistic Regression model shows a balanced performance between detecting churned and non-churned customers. It achieved a **71.91% recall rate for churned customers** indicating that it is successful in identifying over 70% of churned customers. However, the model only achieved a precision of **68.30%** when it came to predicting customer churn. For non-churned customers, the model showed great performance with a **recall rate of 93.70%** and **precision of 94.64%**.

8.2. Random Forest Model

accuracy: 91.95%

	true 0	true 1	class precision
pred. 0	8365	758	91.69%
pred. 1	47	830	94.64%
class recall	99.44%	52.27%	

Figure XV. (AI Studio)

The Random Forest model showed great performance in predicting non-churned customers with a **91.69%** in precision and a recall rate of **99.44%**. When it came to predicting churn, it achieved a **precision of 94.64%**.

However, the recall for churned customers is only **52.27%**, meaning that the model missed almost half of actual churners.

8.3. Gradient Boosted Trees Model

accuracy: 98.22%

	true 0	true 1	class precision
pred. 0	8329	95	98.87%
pred. 1	83	1493	94.73%
class recall	99.01%	94.02%	

Figure XVI. (AI Studio)

The Gradient Boosted Trees achieved a **recall rate of 94.02%** and **precision of 94.73%** when predicting for churned customers. When predicting for non-churned customers, it returned great results, achieving a **recall rate of 99.01%** and a **precision of 98.87%**.

8.4. Conclusion

The performance of the Logistic Regression model shows that while the model is great at predicting for non-churned customers, it struggled with getting a good performance when it came to predicting for customers who will churn. While the lowest result is only **68.30%** for churn prediction precision, the model is far from ideal for churn management.

The Random Forest model produced great results in predicting non-churned customers, achieving high precision and recall rate. On the other hand, the prediction for churned customers only yielded high precision but very poor recall rate. This indicates that this model is unsuitable for churn prediction.

Among the three models that were tested, the Gradient Boosted Trees produced stellar results in both predicting for churned and non-churned customers. The high accuracy and recall rate of this model indicates that the Gradient Boosted Trees model will be the most suitable choice for churn management.