Tackling Credit Card Churn: Overcoming Imbalanced Data for Accurate Customer Retention Prediction

Harishankar Nagar (2018IMSST021)
Integrated M.Sc. Statistics

Under the supervision of
Dr. Mahesh Barale

Department of Statistics
Central University of Rajasthan

Outline

- Objective
- Data Description & Cleaning
- Exploratory Data Analysis
- Feature Selection & Engineering
- Experiment
- Results
- Conclusion

Objective

- Credit card churn prediction faces imbalanced datasets with only approximately 16% churn cases.
- Machine learning algorithms tend to prioritize overall accuracy, leading to biased predictions favoring the majority class (non-churn).
- Biased predictions result in poor identification of churn cases, causing missed revenue opportunities and challenges in retaining valuable customers.
- Various methods have been developed to handle the class imbalance problem that can be broadly divided into two main approaches:
  1. Data-driven approach to balance the class distribution
  2. Algorithm-driven approach adjusts the learning algorithm
- In this project, Our objective is to use a data-driven approach by utilizing three techniques.
  1. Undersampling for Equitable Representation
  2. Oversampling for Amplified Insights
  3. Hybrid Sampling for Synthesized Prowess

Data Description

The dataset under study contains diverse attributes reflecting customer interactions in the credit card domain and contains 10,127 entries.

1. Attrition_Flag
2. Customer_Age
3. Gender
4. Dependent_count
5. Education_Level
6. Marital_Status
7. Income_Category
8. Card_Category
9. Months_on_book
10. Total_Relationship_Count
11. Months_Inactive_12_mon
12. Contacts_Count_12_mon
13. Credit_Limit
14. Total_Revolving_Bal
15. Avg_Open_To_Buy
16. Total_Amt_Chng_Q4_Q1
17. Total_Trans_Amt
18. Total_Trans_Ct
19. Total_Ct_Chng_Q4_Q1
20. Avg_Utilization_Ratio

Data Cleaning

- **Data Quality Assurance:** Before diving into analysis, we took careful steps to ensure that our data was of high quality and trustworthy.
- **Handling Missing Values:** One important step was checking for missing information in the data. The good news is that we found no missing values at all, which means our data was complete and reliable.
- **Data Structuring:** We organized our data in a way that's ready for analysis. This step prepares the data to be easily used for exploring, creating new features, selecting models, and evaluating results.
- **No Duplicates:** We checked for any duplicate entries in our dataset, and we're happy to report that we didn't find any. Each data point is unique.

Exploratory Data Analysis (EDA)

We performed EDA in three steps

1. Analysis of Categorical Column
2. Analysis of Numerical Column
3. Correlation Structure

Analysis of Categorical Column

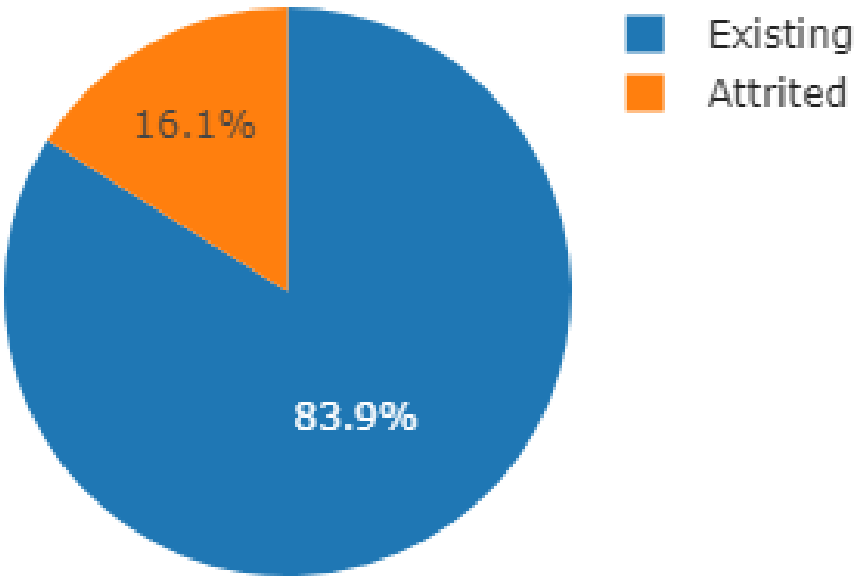


Figure: Pie Chart of Attritiob_Flag

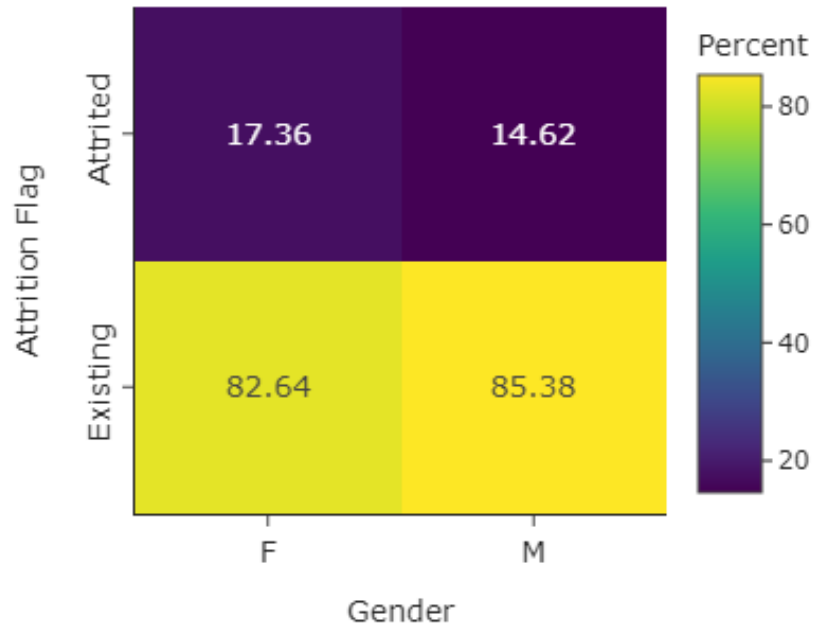


Figure: Gender and Attrition_Flag

Exploratory Data Analysis (EDA)

- We analyzed the contingency table of other variables too and observed that there are slight differences in the percentage attrition (ranges between 14%-20%) of each category of each variable.
- After analyzing the contingency table, We also performed a chi-square test of independence test to check the association between categorical variables and target variables.
- H0: There is no association between the two categorical variables.

Variable Name	**Degree of Freedom**	**Chi-Square Statistics**	**P-Value**
Gender	1	13.865	0.00019
Education Level	6	12.511	0.05148
Marital Status	3	6.056	0.10891
Income Category	5	12.83	0.025
Card Category	3	2.234	0.52523

Analysis of Numerical Variables

- We employed box plots to visualize the distribution of numerical variables. Each box plot represented a specific numerical feature.
- We compared the distribution of each numerical variable separately for the two categories of the target variable, "Attrited" and "Existing."
- Box plots allowed us to identify trends and differences in the distribution of numerical variables for both categories. Patterns in these plots revealed variations that could be significant for predictive modeling.
- The insights gained from the box plot analysis informed our feature selection process, helping us identify relevant numerical features for further analysis and model building.
- Only two variables showed an almost similar distribution of both categories i.e. Customer_Age and Months_on_book.

Exploratory Data Analysis (EDA)

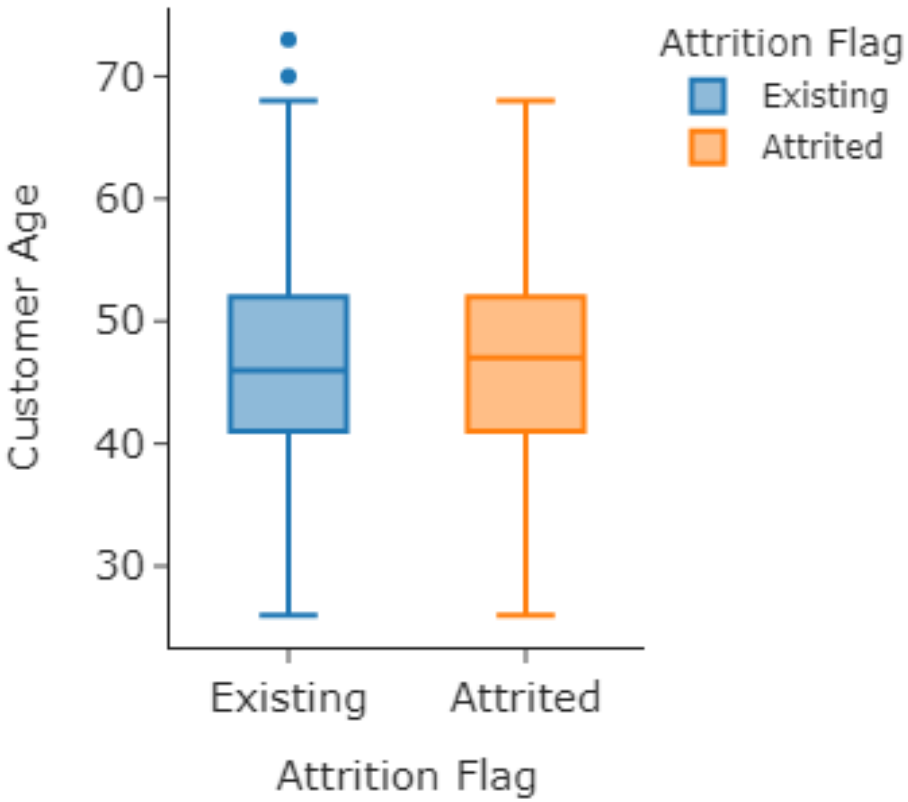


Figure: Box Plot of Customer_Age

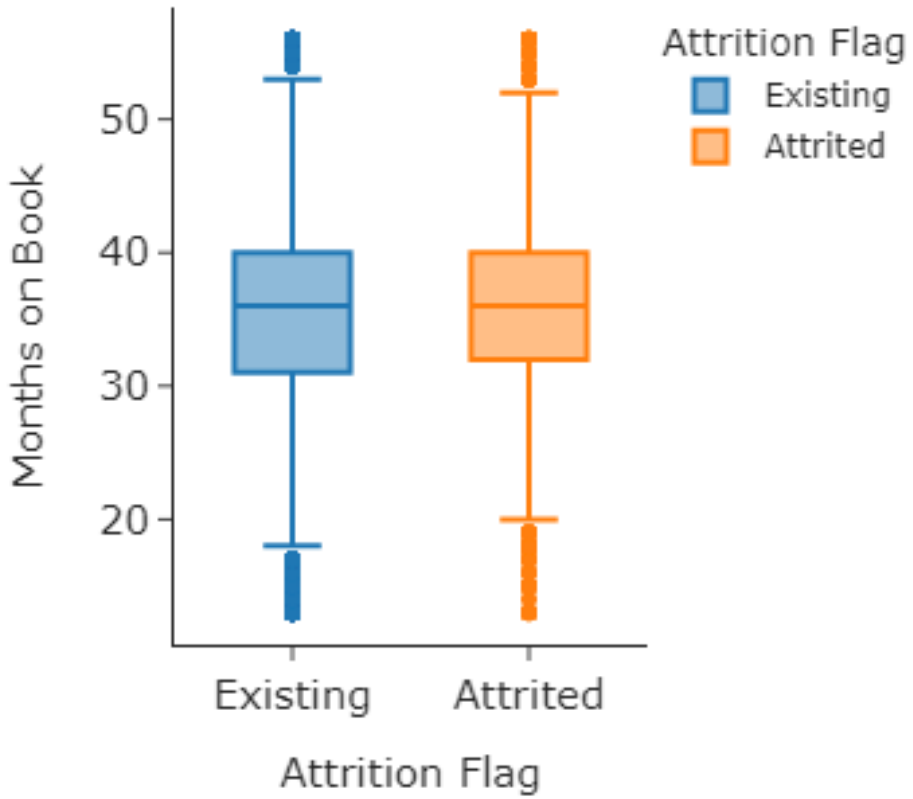
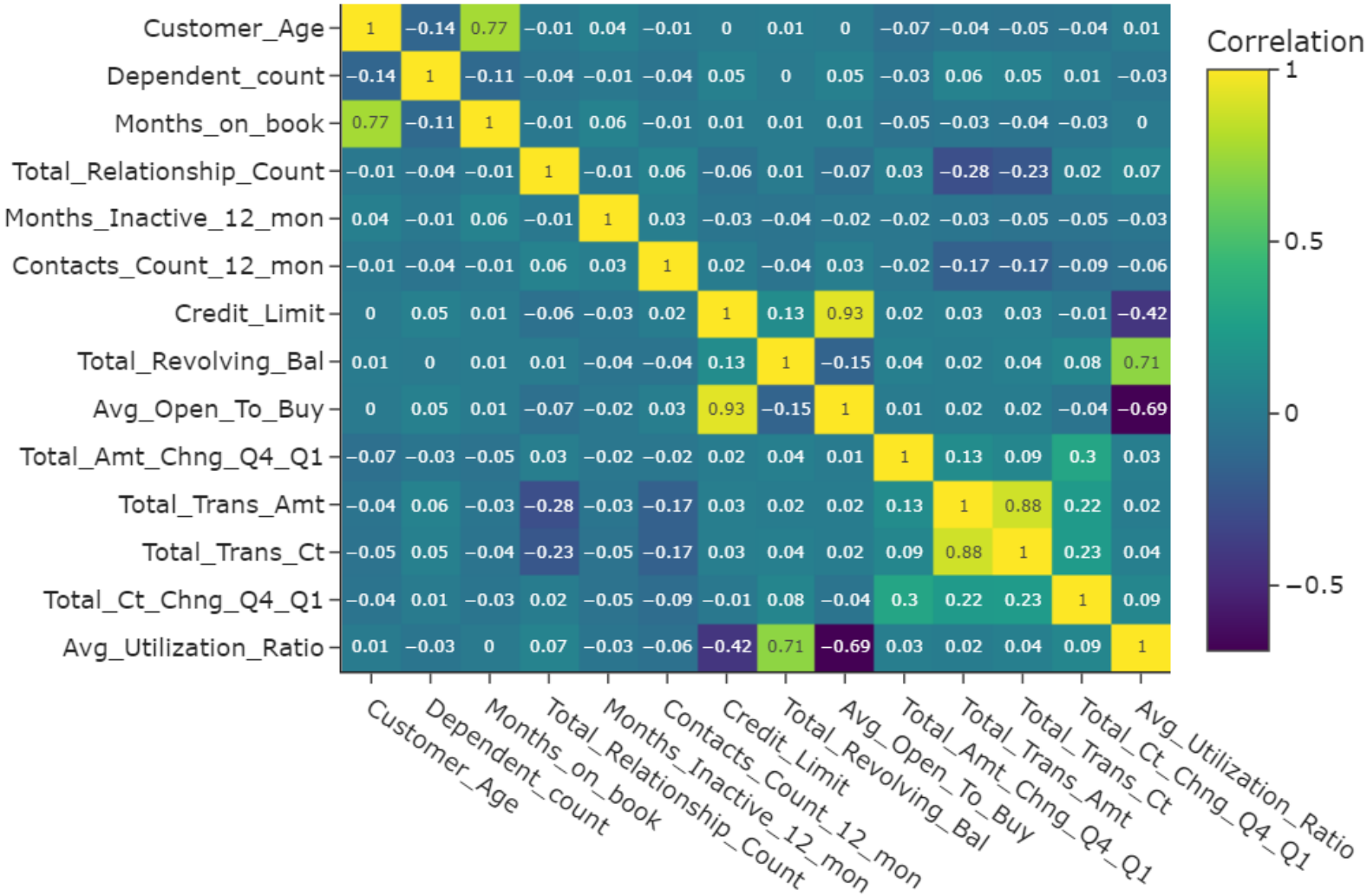


Figure: Box Plot of Months_on_book

Correlation Structure



Feature Selection

Feature selection is crucial for several reasons, particularly in the context of data analysis and machine learning:

- Enhances model's performance by reducing overfitting
- Faster training by avoiding the curse of dimensionality
- Interpretability

Categorical Variables Selection

- Our approach addresses class imbalance by utilizing techniques involving Euclidean distance and mean calculation.
- Categorical variables pose a unique challenge. Due to their nature, they cannot be directly used in calculating Euclidean distance and mean.
- To overcome this limitation, we have made the decision to exclude categorical variables from our feature selection process.

Feature Selection & Feature Engineering

Numerical Variables Selection

For numerical feature selection, we employed two techniques:

- **Box Plot Analysis:** This visual analysis helped us identify features that exhibit significant differences between the classes.
- **Correlation Analysis:** To address multicollinearity, we performed correlation analysis. When we identified pairs of highly correlated variables, we made the decision to drop one of the variables to mitigate redundancy and enhance model interpretability.
- Selected variables are:

1. Dependent_count
2. Total_Relationship_Count
3. Months_Inactive_12_mon
4. Contacts_Count_12_mon
5. Credit_Limit
6. Total_Revolving_Bal
7. Total_Trans_Amt
8. Total_Trans_Ct
9. Total_Ct_Chng_Q4_Q1
10. Avg_Utilization_Ratio

Feature Engineering

- Machine learning algorithms are sensitive to the scales of features.
- To mitigate this issue, we standardized every numerical variable.
- The standardized values (z) will be derived using the formula $z = \frac{x-\mu}{\sigma}$, where x denotes the original value, μ signifies the mean, and σ signifies the standard deviation for each individual feature.
- Feature scaling ensures that all features are on a similar scale, preventing certain features from dominating others during model training. This helps algorithms converge faster and produce more accurate results.

Experimental Setup

Data Spilitting

The dataset was divided into two subsets: a training set and a test set. The training set, comprising 80% of the data, was used to train the models, while the remaining 20% constituted the test set for evaluating the models' predictive performance.

Models Utilized

We have employed three distinct models, each chosen for its simplicity and ease of interpretation:

1. Naive Bayes
2. Logistic Regression
3. Decision Tree

Experiment

Model Evaluation Metrics

To effectively measure model performance, two primary evaluation metrics were employed:

1. **Accuracy Score:** It provides an overall assessment of the model's performance by calculating the ratio of correctly predicted instances (both positive and negative) to the total number of instances.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Instances} \tag{1}$$

2. **Recall Score:** Reacall is the model's ability to correctly identify positive instances (True Positives) out of all actual positive instances (True Positives + False Negatives)

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

Methodology

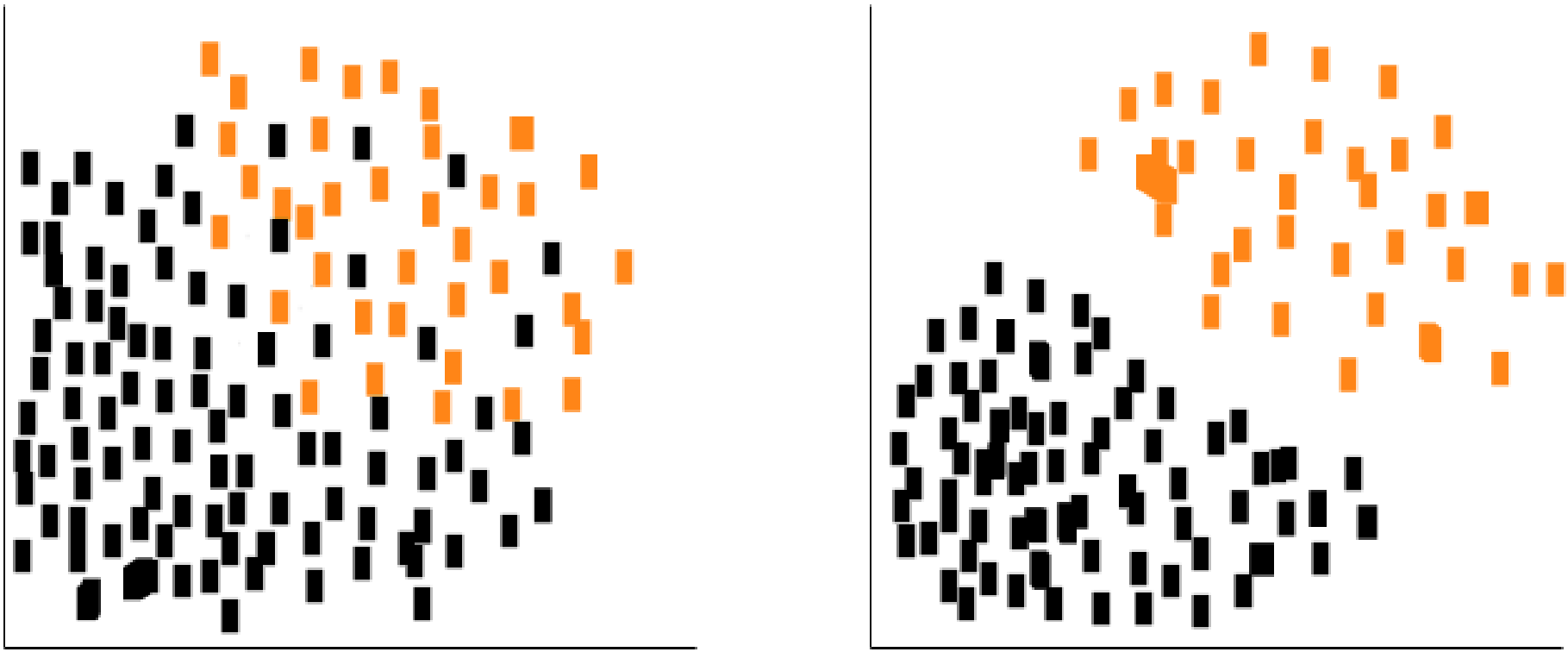


Figure: Dummy Image of Undersampling

Experiment

Undersampling for Equitable Representation

In the context of undersampling, our approach involves a systematic process aimed at achieving dataset balance while enhancing the decision boundary by removing noise. The steps undertaken are as follows:

1. Mean Vector Calculation
2. Distance Calculation
3. Sorting by Distance
4. Targeted Removal

Experiment

Oversampling for Amplified Insights

Within the framework of addressing class imbalance and refining analytical accuracy, our approach encompasses a methodical series of actions designed to amplify the minority class instances.

1. Triple Instance Sampling
2. Mean Value Computation
3. Augmenting Minority Class
4. Iterative Process

Experiment

Hybrid Sampling for Synthesized Prowess

Hybrid sampling comprises augmentation of the minority class while systematically reducing the majority class instances. The methodology unfolds as follows:

1. Augmentation of Minority Class:
   - Triple Instance Sampling
   - Mean Value Computation
   - Augmenting Minority Class
   - Iterative Process
2. Reduction of Majority Class:
   - Mean Vector Calculation
   - Distance Calculation
   - Sorting by Distance
   - Targeted Removal

Results

Model Performance on original Data

Algorithms	Majority Class Recall	Minority Class Recall	Accuracy
Naïve Bayes	0.93	0.59	0.88
Logistic Regression	0.97	0.52	0.9
Decision Tree	0.96	0.76	0.92

Undersampling

Algorithms	Majority Class Recall	Minority Class Recall	Accuracy
Naïve Bayes	0.72	0.79	0.73
Logistic Regression	0.8	0.83	0.81
Decision Tree	0.71	0.94	0.74

Results

Performance after Oversampling

Algorithms	Majority Class Recall	Minority Class Recall	Accuracy
Naïve Bayes	0.87	0.58	0.82
Logistic Regression	0.88	0.78	0.86
Decision Tree	0.94	0.79	0.92

Performance after Hybrid Sampling

Algorithms	Majority Class Recall	Minority Class Recall	Accuracy
Naïve Bayes	0.75	0.73	0.75
Logistic Regression	0.86	0.77	0.84
Decision Tree	0.78	0.87	0.79

Conclusion

- Under-sampling techniques led to notable improvements in the recall of the minority class but raised concerns about potential information loss from the majority class.
- Oversampling, on the other hand, elicited diverse responses from different algorithms, with the introduction of noise and redundancy as potential trade-offs.
- Hybrid sampling emerged as a promising approach, striking a balance between class representation and accuracy. It demonstrated the potential for effectively addressing class imbalance while maintaining competitive model performance.
- Decision Tree model's consistent performance, making it a compelling choice for practical applications in credit card churn prediction.

Thank You