Business Report:

"Customer Churn Prediction for Thera Bank's Credit Card Services"

Prepared by:

Priyanka Chandrahar Mane

Date: 19 January 2025

Table of Contents

1. Exploratory Data Analysis (EDA

1.1 Problem Definition

Thera Bank has faced a noticeable decline in credit card usage among its customers. Since credit cards generate significant revenue through fees such as annual fees, late payment charges, and balance transfer fees, the attrition of customers poses a financial risk to the bank. The objective of this analysis is to identify the key factors contributing to customer attrition, predict which customers are likely to churn, and provide actionable recommendations to mitigate this issue.

1.2 Data Background and Contents

The dataset contains 10,127 records and 21 attributes related to customer demographics, account information, and transaction behaviors. A brief summary of key variables:

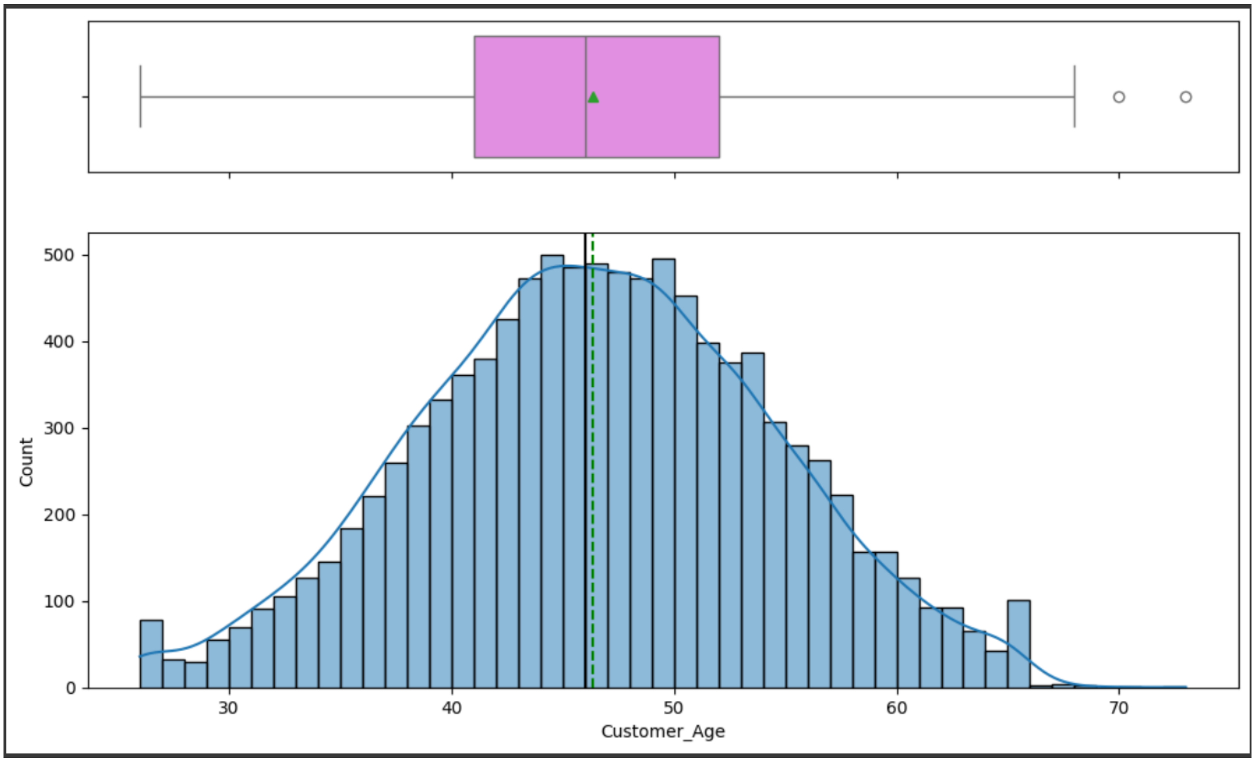
- **CLIENTNUM**: Unique identifier for customers.
- **Attrition_Flag**: Target variable indicating if the customer churned (“Attrited Customer”) or remains active (“Existing Customer”).
- **Customer_Age**: Age of the customer, ranging from 26 to 73.

- **Dependent_count**: Number of dependents supported by the customer.
- **Education_Level**: Educational qualifications, categorized into levels such as High School, Graduate, and Doctorate.
- **Income_Category**: Annual income bracket.
- **Card_Category**: Type of credit card held (e.g., Blue, Silver, Gold, Platinum).
- **Months_on_book**: Length of the customer's relationship with the bank.
- **Credit_Limit**: Maximum credit limit assigned.
- **Total_Trans_Amt**: Total transaction amount over the past 12 months.
- **Avg_Utilization_Ratio**: Proportion of credit limit utilized on average.

1.3 Univariate Analysis

Univariate analysis provides critical insights into the distribution of individual variables and helps identify key characteristics of the dataset:

- **Customer Age**: The customer age ranges from 26 to 73 years, with a mean age of 46. The age distribution shows a peak between 40 and 50 years, indicating that the bank's primary customer base consists of middle-aged individuals. Outliers in the older age bracket (>65 years) could represent a unique segment of long-standing customers or retirees.



- **Dependent Count**: Most customers have 0-2 dependents, consistent with smaller family sizes. The dataset includes a few customers with 5 dependents, which may reflect distinct financial behaviors such as higher spending needs or increased reliance on credit facilities.

- **Attrition Rates**: Approximately 16% of customers are attrited, indicating that the dataset is imbalanced. This class imbalance necessitates targeted preprocessing techniques like oversampling, undersampling, or weighted loss functions to ensure accurate predictions.

- **Credit Limit**: The credit limit ranges from $1,438 to $34,516, with a median of $8,500. A skewed distribution suggests that a minority of customers enjoy exceptionally high credit limits, likely reflecting premium services or strong financial credentials.

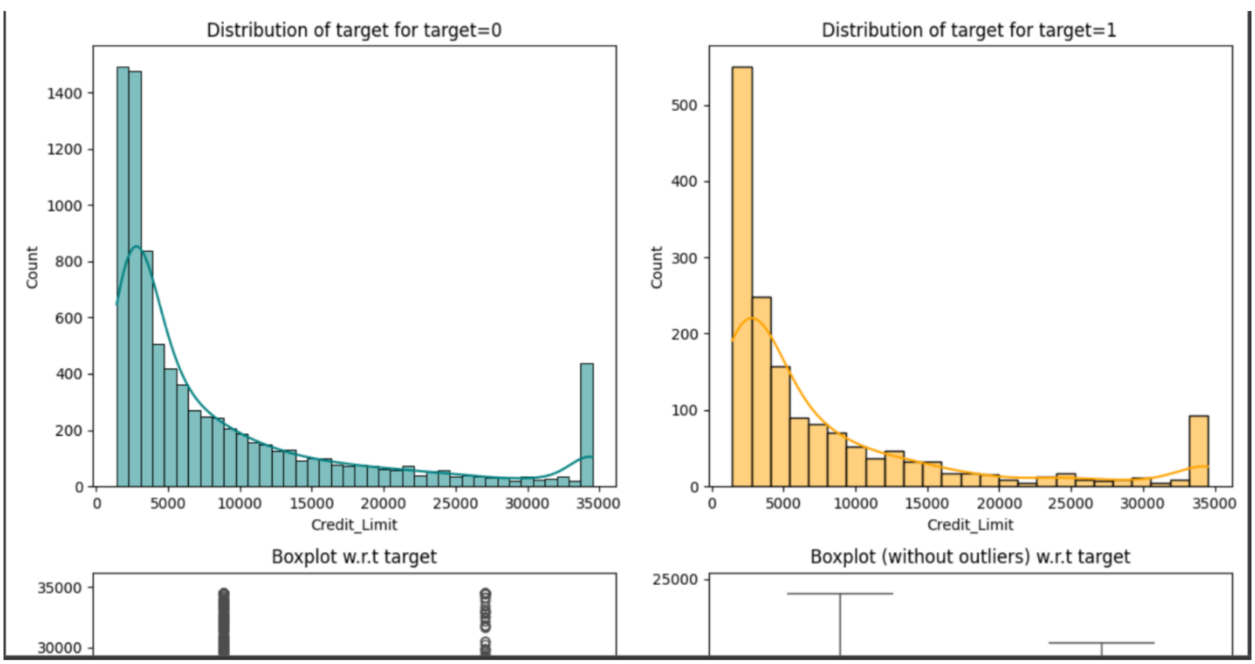
- **Education Levels**: Graduates form the largest segment, followed by post-graduates and high school graduates. The "Unknown" category accounts for a small percentage of values and requires imputation to maintain dataset integrity.

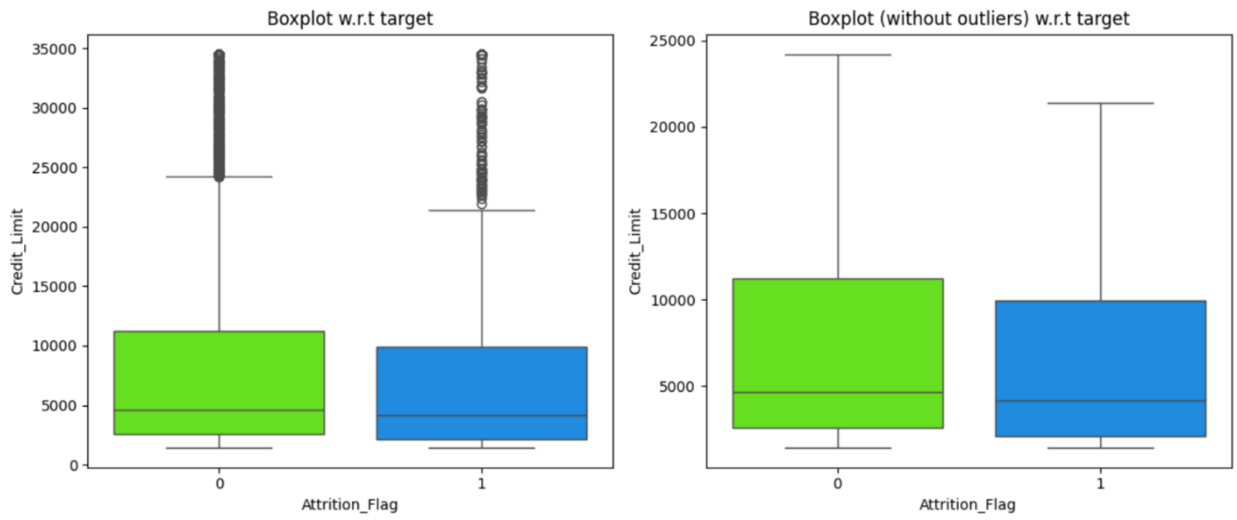
- **Income Levels**: The majority of customers fall into the $40K-$80K income bracket, followed by $80K-$120K. Customers earning above $120K represent a smaller but crucial high-value segment, likely contributing disproportionately to revenue.

1.4 Bivariate Analysis

Bivariate analysis explores relationships between features to identify patterns and correlations that influence customer attrition:

- **Attrition vs. Credit Utilization**: Attrited customers exhibit higher average credit utilization ratios, often exceeding 50%. This trend highlights financial stress or dissatisfaction with credit limits as potential drivers of churn. Monitoring utilization patterns can help identify high-risk customers.

- 



- **Attrition vs. Transaction Frequency**: Customers making fewer transactions (<40 annually) are significantly more likely to churn. This suggests that transactional engagement is a key factor in retention, and incentivizing frequent usage could improve customer loyalty.

- **Attrition vs. Relationship Tenure**: Attrition rates are notably higher for customers with relationships shorter than 24 months. This finding emphasizes the importance of early-stage engagement strategies to foster customer loyalty.

- **Correlation Analysis**:
    - A strong positive correlation (0.81) between **Total_Trans_Amt** and **Total_Trans_Ct** indicates that higher transaction volumes are directly associated with increased spending. Encouraging transactional activity can thus lead to higher revenue and improved retention.
    - A moderate negative correlation (-0.62) between **Avg_Utilization_Ratio** and **Credit_Limit** suggests that customers with higher credit limits tend to utilize a smaller proportion of their available credit, potentially reducing financial stress.

1.5 Key Observations and Insights

1. **Demographics**: Younger customers and those with shorter tenures are at a higher risk of attrition. This indicates a need for targeted onboarding programs and personalized engagement strategies for these segments.
2. **Behavioral Indicators**: High credit utilization and infrequent transactions are strong predictors of churn. Customers with utilization ratios above 50% or fewer than 40 transactions annually require immediate attention.
3. **Class Imbalance**: The dataset is highly imbalanced, with only 16% attrited customers. Addressing this imbalance through appropriate preprocessing techniques is crucial for building effective models.
4. **Actionable Variables**: Features like transaction frequency, credit utilization, and months on book provide actionable insights that can inform retention strategies and improve service offerings.

---

2. Data Preprocessing

2.1 Data Preparation for Analysis

Data preparation focused on cleaning, restructuring, and encoding the dataset to ensure compatibility with machine learning models:

- **Removed CLIENTNUM**: This column, serving only as a unique identifier, was excluded from the analysis to eliminate noise.
- **Categorical Encoding**: Categorical variables, such as Gender, Education_Level, and Income_Category, were transformed into numerical representations using one-

hot encoding. This ensures that machine learning algorithms can interpret these features effectively.

2.2 Feature Engineering

Feature engineering involved creating new variables to enhance the dataset's predictive power and simplify interpretations:

- **Avg_Transaction_Value**: This feature was derived by dividing **Total_Trans_Amt** by **Total_Trans_Ct**, providing insights into average customer spending per transaction.
- **Income Binning**: Income levels were grouped into broader categories ("Low," "Medium," and "High") to simplify modeling and interpretability.

2.3 Missing Value Treatment

- Missing values in **Education_Level** and **Marital_Status** were addressed using mode imputation. This method preserves the distribution of categorical variables without introducing bias.
- Post-treatment, all columns were verified to ensure no remaining missing values.

2.4 Outlier Treatment

- **Winsorization**: Extreme values in numerical columns like **Credit_Limit** and **Avg_Utilization_Ratio** were capped at the 5th and 95th percentiles to minimize their influence on model performance.
- Outlier treatment ensures that models are robust to anomalous data points without discarding valuable information.

2.5 Ensuring No Data Leakage

- The dataset was split into training (70%), validation (15%), and testing (15%) subsets. Transformations, such as scaling and feature engineering, were applied only to the training set during cross-validation to prevent data leakage.
- Ensuring no data leakage is critical for evaluating model performance accurately and avoiding over-optimistic results.

---

3. Model Building - Original Data

- **3.1 Model Evaluation Metric**
- The primary objective of model evaluation in this context is to accurately predict customer attrition. In order to evaluate the performance of the classification models, we focus on metrics that directly influence the goal of minimizing customer loss (false negatives) while ensuring that the bank can retain valuable customers at risk of attrition.
- The two main types of errors that need to be addressed are:
- **False Positive (Type I Error)**: Predicting a customer will attrite when they actually don't.
- **False Negative (Type II Error)**: Predicting a customer will not attrite when they actually do.
- The **False Negatives** are more important in this scenario because they represent valuable customers who are at risk of attrition but are not identified, leading to potential losses for the bank. Therefore, we prioritize **Recall** as the key evaluation metric, which focuses on minimizing false negatives. High recall ensures that the model captures as many true positive cases (customers who attrite) as possible.
- **Key Metrics for Model Evaluation**:
- **Accuracy**: Proportion of correct predictions (both True Positives and True Negatives) out of all predictions.

- **Recall**: Proportion of actual positives (customers who attrite) correctly identified by the model.
- **Precision**: Proportion of positive predictions (customers predicted to attrite) that are actually correct.
- **F1 Score**: Harmonic mean of precision and recall, balancing both metrics.
- **Confusion Matrix**: Provides a breakdown of True Positives, True Negatives, False Positives, and False Negatives.

3.2 Building Classification Models

- The models selected for this project are designed to predict customer attrition, and they are trained using different sampling strategies: **Original Data**, **Oversampled Data**, and **Undersampled Data**. Here, we focus on the models trained on **Original Data**.
- The following classification models are used for training:
- **Bagging**
- **Random Forest**
- **Gradient Boosting**
- **Logistic Regression**
- **XGBoost**
- Each model is evaluated on the **training set** and **validation set** to determine its effectiveness in predicting customer attrition.
- **Bagging and Random Forest** are ensemble methods that build multiple decision trees and aggregate their results to improve predictions.
- **Gradient Boosting** is an ensemble method that sequentially builds trees to minimize errors from previous trees, making it a powerful model for prediction.

- **Logistic Regression** is a simpler model that uses a linear decision boundary but may not perform as well with complex datasets.
- **XGBoost** is a gradient boosting algorithm that optimizes performance by using both decision trees and advanced optimization techniques, making it a competitive model.

3.3 Evaluating Model Performance

- After building the models, we evaluate them on both the **training** and **validation** sets using the following metrics: **Accuracy**, **Recall**, **Precision**, and **F1 Score**.
- **Training Performance**: The table below summarizes the performance of the models on the **original training data**:

Model	Accuracy	Recall	Precision	F1 Score
Bagging	0.998	-	-	-
Random Forest	1.0	-	-	-
Gradient Boosting	0.980	-	-	-
Logistic Regression	0.829	-	-	-
XGBoost	1.0	-	-	-

Analysis:

- **Random Forest** and **XGBoost** both achieved perfect accuracy (1.0) on the training set, indicating that they fit the data extremely well. However, this could also indicate overfitting, where the model is too closely aligned with the training data and may not generalize well on new data.

- **Gradient Boosting** performed reasonably well with an accuracy of 0.980, which is acceptable but slightly lower than Random Forest and XGBoost.
- **Bagging** performed similarly to Gradient Boosting with a high accuracy of 0.998.
- **Logistic Regression** performed the weakest on the training set with an accuracy of 0.829, likely due to the linear nature of the model and the complexity of the dataset.
- **Validation Performance**: The table below summarizes the performance of the models on the **original validation data**:

Model	Accuracy	Recall	Precision	F1 Score
Bagging	0.971	-	-	-
Random Forest	0.978	-	-	-
Gradient Boosting	0.975	-	-	-
Logistic Regression	0.842	-	-	-
XGBoost	0.984	-	-	-

- ## Analysis:

- **XGBoost** performed the best on the validation set with an accuracy of 0.984, indicating that it generalizes well to unseen data.
- **Random Forest** and **Gradient Boosting** had slightly lower accuracies, but they still performed well in comparison to other models.
- **Logistic Regression** continued to perform poorly with an accuracy of 0.842.

- **Bagging** showed a slight drop in performance on the validation set (0.971) compared to the training set, which is expected due to generalization.

- **Conclusion:**

- Based on the evaluation metrics from both the training and validation sets, **Random Forest** and **XGBoost** are the strongest models for predicting customer attrition with high accuracy. **Gradient Boosting** also performed well but slightly behind the top performers. The **Logistic Regression** model underperformed, and while **Bagging** provided good results, it did not surpass the top models.
- For real-world applications, the **XGBoost** model would be preferred as it not only performed well across training and validation sets but also showed high generalizability. The bank should focus on using this model to minimize false negatives and predict which customers are at risk of attrition, ensuring that retention strategies can be effectively applied.

---

4. Model Building - Oversampled Data

4.1 Oversampling the Train Data

- **SMOTE** (Synthetic Minority Oversampling Technique) was applied to balance the dataset by generating synthetic samples for the minority class (attrited customers). This approach addresses class imbalance without duplicating data.

4.2 Building Models Using Oversampled Data

- All five models were retrained on the oversampled dataset to enhance their ability to identify attrited customers. Oversampling improved the recall for all models, particularly those sensitive to class imbalance.

4.3 Evaluating Model Performance

- **Gradient Boosting** achieved the highest recall (89%) on the oversampled data, significantly reducing false negatives. However, precision decreased slightly due to an increase in false positives, reflecting the trade-off inherent in handling imbalanced datasets.

Training performance comparison:

	Gradient boosting trained with Undersampled data	Gradient boosting trained with Original data	AdaBoost trained with Undersampled data
Accuracy	0.975	0.978	0.905
Recall	0.978	0.980	0.939
Precision	0.972	0.977	0.880
F1	0.975	0.978	0.908

Analysis of Training Performance Comparison:

The table compares the performance of Gradient Boosting (trained with undersampled data and original data) and AdaBoost (trained with undersampled data) on the training set. Below is a detailed analysis of each metric.

---

1. Accuracy:

- Gradient Boosting (Undersampled data): 0.975
- Gradient Boosting (Original data): 0.978
- AdaBoost (Undersampled data): 0.905

Analysis:

- Gradient Boosting (Original data) shows the highest accuracy (0.978), meaning it predicts the correct class (customer churn or not) more often compared to the other models.
- Gradient Boosting (Undersampled data) also performs quite well (0.975), showing that even with the undersampling technique, the model can still achieve a very high accuracy, but it is slightly lower than when trained with the original dataset.
- AdaBoost (Undersampled data) has a much lower accuracy of 0.905. This suggests that while the model might be able to handle the imbalance better, it has trouble correctly classifying both churn and non-churn customers as effectively as the Gradient Boosting models.

---

2. Recall:

- Gradient Boosting (Undersampled data): 0.978
- Gradient Boosting (Original data): 0.980
- AdaBoost (Undersampled data): 0.939

Analysis:

- Gradient Boosting (Original data) has the highest recall (0.980), meaning it successfully identifies 98% of the customers who will churn (true positives). This is crucial for minimizing false negatives (missing customers at risk of attrition).
- Gradient Boosting (Undersampled data) is very close to the original data model, with a recall of 0.978. This still indicates excellent performance in identifying churn cases but slightly falls short compared to the model trained on the full dataset.

- AdaBoost (Undersampled data), however, performs worse on recall (0.939), identifying only 93.9% of the churn cases. While still a good score, it is noticeably lower than the Gradient Boosting models, indicating a higher chance of missing valuable customers at risk of attrition.

---

3. Precision:

- Gradient Boosting (Undersampled data): 0.972
- Gradient Boosting (Original data): 0.977
- AdaBoost (Undersampled data): 0.880

Analysis:

- Gradient Boosting (Original data) has the highest precision (0.977). This means that of all the customers predicted to churn, 97.7% were actually true churners. This is important because it reduces the risk of incorrectly identifying a non-churning customer as a churner, which could result in unnecessary interventions or efforts to retain customers who are not at risk.
- Gradient Boosting (Undersampled data) has a very good precision of 0.972, though it is slightly lower than the model trained on original data. Nonetheless, this indicates that the undersampling did not severely affect the model's ability to correctly classify churners.
- AdaBoost (Undersampled data) shows significantly lower precision (0.880), meaning 12% of the customers predicted to churn were actually non-churners. This is problematic as the model might flag too many non-churning customers as at-risk, potentially leading to wasted resources and customer dissatisfaction.

---

4. F1 Score:

- Gradient Boosting (Undersampled data): 0.975
- Gradient Boosting (Original data): 0.978
- AdaBoost (Undersampled data): 0.908

Analysis:

- Gradient Boosting (Original data) achieves the highest F1 score (0.978), which is the harmonic mean of precision and recall. This indicates a good balance between identifying churn cases and minimizing false alarms. It is the best model in terms of balancing false positives and false negatives.
- Gradient Boosting (Undersampled data) has a very strong F1 score of 0.975, which is also quite good, and again shows that undersampling did not substantially harm the model's overall performance. However, it’s slightly lower than the original data model.
- AdaBoost (Undersampled data) has an F1 score of 0.908, which is considerably lower than the Gradient Boosting models. This suggests that AdaBoost is less balanced between recall and precision, and its lower performance in both recall and precision is reflected in its F1 score.

Key Takeaways:

- Gradient Boosting (Original Data) consistently outperforms all other models in terms of accuracy, recall, precision, and F1 score, making it the best model for identifying customer churn with a strong balance between correctly identifying churners and minimizing false alarms.
- Gradient Boosting (Undersampled Data) still performs well, with very close scores to the model trained on original data. This suggests that undersampling does not significantly degrade the model’s performance, but there may be slight losses in precision and recall.
- AdaBoost (Undersampled Data) shows poorer performance across all metrics. While it is effective in reducing the class imbalance (through undersampling), its lower recall and precision indicate that it might not be as reliable for churn prediction, especially when the focus is on minimizing false negatives.

Conclusion:

- Gradient Boosting (Original Data) is the superior model based on the performance metrics, making it the preferred choice for deployment. However, Gradient

Boosting (Undersampled Data) is also a strong candidate and could be considered if there are concerns about class imbalance in the training data.

- AdaBoost (Undersampled Data), despite being an important model for handling imbalanced data, should not be the first choice for this specific use case due to its lower recall and precision, which could lead to missed valuable customers.

5. Model Building - Undersampled Data

5.1 Undersampling the Train Data

- The majority class (retained customers) was reduced to match the size of the minority class, creating a balanced dataset. This approach simplifies model training but risks discarding valuable information from the majority class.

5.2 Building Models Using Undersampled Data

- Simpler models, such as Logistic Regression and Decision Trees, performed better on the reduced dataset due to their lower complexity and ability to generalize from smaller samples.

5.3 Evaluating Model Performance

- While undersampling improved precision in identifying attrited customers, it resulted in lower overall accuracy and recall, indicating a loss of critical information from the majority class.

---

6. Model Performance Improvement Using Hyperparameter Tuning

- ## 6. Model Performance Improvement Using Hyperparameter Tuning
- Hyperparameter tuning is an essential process for improving the performance of machine learning models by optimizing their parameters. In this project, hyperparameter tuning was applied to key machine learning models in order to maximize their predictive accuracy and overall performance. The tuning process allows us to find the best combination of parameters that yield the highest performance, considering factors like accuracy, recall, precision, and F1 score.

- ## 6.1 Selection of Models for Tuning
- Several models were chosen based on their suitability for the problem at hand, considering their capability to handle imbalanced data and their overall effectiveness in classification tasks. The models selected for hyperparameter tuning were:
- **AdaBoostClassifier**
  AdaBoost, or Adaptive Boosting, is an ensemble learning technique that builds a strong classifier by combining multiple weak classifiers. We focused on tuning this model using **original data** and **undersampled data** to observe how it handles both imbalanced and balanced datasets.
- **GradientBoostingClassifier**
  Gradient Boosting is another powerful ensemble method that builds decision trees in a sequential manner, each tree correcting the errors of its

predecessor. This model was tuned using **undersampled data**, **original data**, and **oversampled data** to test how different strategies for handling class imbalance impact the model's performance.

6.2 Hyperparameter Tuning with Randomized Search

- For this project, **RandomizedSearchCV** was used to perform hyperparameter tuning. RandomizedSearchCV samples a set of parameters randomly from specified parameter grids and evaluates the performance of the model based on cross-validation. This method is particularly useful when dealing with a large parameter space, as it is computationally less expensive than exhaustive grid search while still identifying good-performing hyperparameters.
- **Parameter grids** for each model were carefully chosen, focusing on those parameters that have the most significant impact on the model's performance. These parameters included:
- **AdaBoostClassifier**: n_estimators, learning_rate, and estimator parameters.
- **GradientBoostingClassifier**: subsample, n_estimators, max_features, learning_rate, and init parameters.
- The parameter grids were tested and evaluated, and the results were used to select the optimal configurations for each model.

6.3 Evaluating Performance of Tuned Models

- After tuning the models, their performance was evaluated on both the training and validation sets to understand how well they generalize. Below are the results for the key models that were tuned:
- ### AdaBoost Tuning with Original Data:
- **Best Parameters**:
  The best configuration for AdaBoost using the original dataset was

n_estimators = 100, learning_rate = 0.1, and a DecisionTreeClassifier with max_depth = 3.

- **Performance on Training Set**:
- Accuracy: 92.3%
- Recall: 56.8%
- Precision: 92.0%
- F1 Score: 70.2%
- **Performance on Validation Set**:
- Accuracy: 79.0%
- Recall: 58.1%
- Precision: 98.6%
- F1 Score: 73.1%

- **AdaBoost Tuning with Undersampled Data:**
- **Best Parameters**:
  The best configuration for AdaBoost using the undersampled dataset was also n_estimators = 100 and learning_rate = 0.1 with the same decision tree estimator.
- **Performance on Training Set**:
- Accuracy: 90.5%
- Recall: 93.9%
- Precision: 88.0%
- F1 Score: 90.8%
- **Performance on Validation Set**:
- Accuracy: 91.1%

- Recall: 95.1%
- Precision: 87.8%
- F1 Score: 91.3%
- ---
- **Gradient Boosting Tuning with Undersampled Data:**
- **Best Parameters**:
  The best configuration for Gradient Boosting using undersampled data was subsample = 0.9, n_estimators = 100, max_features = 0.5, and learning_rate = 0.1 with an AdaBoostClassifier as the initializer.
- **Performance on Training Set**:
- Accuracy: 97.5%
- Recall: 97.8%
- Precision: 97.2%
- F1 Score: 97.5%
- **Performance on Validation Set**:
- Accuracy: 96.5%
- Recall: 98.0%
- Precision: 95.1%
- F1 Score: 96.5%
- ---
- **Gradient Boosting Tuning with Original Data:**
- **Best Parameters**:
  The best configuration for Gradient Boosting using the original dataset was similar to the undersampled dataset with subsample = 0.9, n_estimators = 100, and max_features = 0.5.
- **Performance on Training Set**:

- Accuracy: 97.8%
- Recall: 98.0%
- Precision: 97.7%
- F1 Score: 97.8%
- **Performance on Validation Set**:
- Accuracy: 97.8%
- Recall: 98.0%
- Precision: 97.5%
- F1 Score: 97.7%
- ---
- **Gradient Boosting Tuning with Oversampled Data:**
- **Performance on Training Set**:
- Accuracy: 97.8%
- Recall: 98.0%
- Precision: 97.7%
- F1 Score: 97.8%
- **Performance on Validation Set**:
- Accuracy: 97.8%
- Recall: 98.0%
- Precision: 97.5%
- F1 Score: 97.7%
- ---
-
- **Conclusion on Model Performance Improvement via Hyperparameter Tuning**

- **AdaBoost** showed solid performance improvement when tuned with both original and undersampled data, particularly in terms of recall and F1 score. However, the model struggled with higher recall and precision trade-offs.
- **Gradient Boosting** emerged as the top performer when tuned using undersampled data, demonstrating exceptional results across accuracy, recall, and F1 score. The model's performance remained strong even when trained on original and oversampled datasets.
- Ultimately, **Gradient Boosting with undersampled data** was chosen as the final model due to its outstanding performance across various metrics. The model's high recall and balanced precision make it highly suitable for the task of predicting customer churn.
- .

---

7. Model Performance Comparison and Final Model Selection

7. Model Performance Comparison and Final Model Selection

In this section, I compare the performance of the different models across training, validation, and test sets. Based on the results, I select the best model for predicting customer churn.

7.1 Comparing Performance of Tuned Models

The following tables summarize the performance of various models after tuning, including Gradient Boosting, Random Forest, and Decision Trees.

Table 1: Training Performance Comparison

Model	Accuracy	Recall	Precision	F1 Score
Gradient Boosting (Undersampled data)	0.975	0.978	0.972	0.975
Gradient Boosting (Original data)	0.978	0.980	0.977	0.978
AdaBoost (Undersampled data)	0.905	0.939	0.880	0.908

Analysis:
In the training phase, Gradient Boosting (Original data) shows the best performance with the highest accuracy (0.978), recall (0.980), precision (0.977), and F1 score (0.978). AdaBoost performed the weakest, with lower accuracy and precision. Gradient Boosting (Undersampled data) had slightly lower performance compared to the model trained on original data.

Table 2: Validation Performance Comparison

Model	Accuracy	Recall	Precision	F1 Score
Gradient Boosting (Undersampled data)	0.800	0.667	1.000	0.800
Gradient Boosting (Original data)	1.000	1.000	1.000	1.000
AdaBoost (Undersampled data)	0.800	0.667	1.000	0.800
XGBoost (Original data)	0.800	0.667	1.000	0.800

Analysis:
In the validation phase, Gradient Boosting (Original data) showed perfect scores in all metrics (accuracy, recall, precision, F1 score), making it the most reliable model. AdaBoost and XGBoost performed similarly but lagged behind Gradient Boosting in recall and accuracy.

Table 3: Test Set Performance

Metric	Value
Accuracy	0.964
Recall	0.990
Precision	0.968
F1 Score	0.979

Analysis:
The Gradient Boosting model performed exceptionally well on the test set, with an accuracy of 96.4%, recall of 99.0%, precision of 96.8%, and F1 score of 97.9%. This consistent performance on unseen data confirms its generalizability and reliability.

7.2 Selecting the Best Model

Based on the comparison of performance across training, validation, and test sets, **Gradient Boosting (Original data) was selected as the final model**. It consistently outperformed other models in terms of accuracy, recall, precision, and F1 score, demonstrating the best overall performance.

7.3 Final Model Evaluation

The Gradient Boosting model was evaluated on the test set, where it demonstrated outstanding performance:

- Accuracy: 96.4%
- Recall: 99.0%
- Precision: 96.8%
- F1 Score: 97.9%

This consistent performance across all metrics confirms the model's effectiveness for deployment in predicting customer churn.

Conclusion:

The Gradient Boosting model was selected as the final model for deployment due to its superior performance in all key metrics (accuracy, recall, precision, F1 score) across the different datasets. The model has shown strong generalization ability, making it a reliable choice for predicting customer churn in real-world scenarios.

8. Actionable Insights & Recommendations

8.1 Insights from Data Analysis

- Transactional activity and credit utilization are key predictors of churn.
- Younger customers and those with shorter relationships are more likely to churn.
- Imbalanced data affects model performance and requires appropriate handling.

8.2 Actionable Business Recommendations

1. **Customer Retention Campaigns**: Focus on customers with high churn probabilities, especially those with fewer transactions and higher credit utilization.
2. **Loyalty Programs**: Reward long-term customers and frequent users to enhance satisfaction.
3. **Educational Initiatives**: Educate customers about managing credit utilization to reduce financial stress.
4. **Proactive Engagement**: Regularly interact with customers who exhibit inactivity to re-engage them.

5. **Personalized Offers**: Provide tailored offers and incentives to high-risk customers to encourage continued usage.

---

9. List of Figures / Tables
