# CHAPTER 4

# INITIAL FINDING AND RESULTS

## 4.1 Introduction

This chapter discusses the results generated from customer churn prediction in e-commerce industry. This chapter begins with dataset identification, followed by Exploratory Data Analysis (EDA). Then, data preprocessing and feature engineering is comprehensively approached. Logistic Regression, Random Forest, Random Forest attached with SMOTE, and XGBoost are used to evaluate the best model. After that, hyperparameter tuning is done on all 4 models. The purpose is to investigate the changes between 4 models, before and after hyperparameter tuning. According to the results generated during model implementation, it is proven that hyperparameter tuning would improve model performance across multiple metrics. The improved results would be effective in developing robust customer churn prediction models for e-commerce applications.

## 4.2 Exploratory Data Analysis (EDA)

Exploratory Data Analysis is a process to understand the features of the data before any preprocessing action is taken. This step is important because it can be used to find out the core attribute that affects the customer churn. The column ‘churn’ function as a primary outcome variable, which is labelled with 1 and 0. 1 indicates that the customer has churned while 0 represents not churned customers.

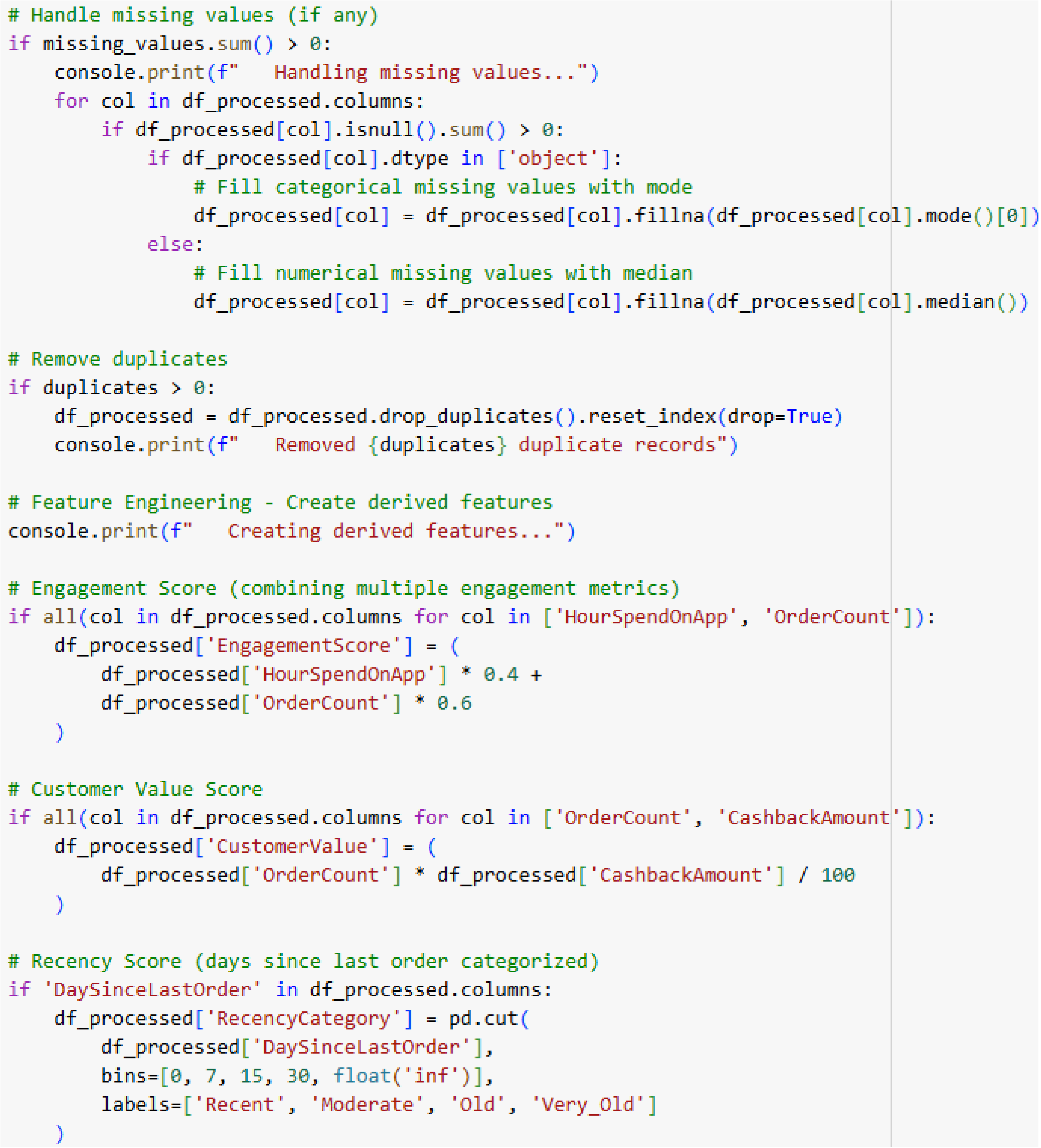
### 4.2.1 Data Collection

The dataset used in this research is obtained from Kaggle's open data repository, titled "E-commerce Customer Churn Analysis and Prediction" dataset (https://www.kaggle.com/datasets/ankitverma2010/ecommerce-customer-churn-analysis-and-prediction/data). This dataset contains 5630 customer records with 20 different features, representing the versatility of the customer with various behaviour and characteristics.

The dataset contains 2 different sheet. Sheet 1 named Data Dict indicates all the variables that exist in the dataset, followed by the description of each variable. Main dataset shows all the customer records. The dataset provides customer demographics, transactional behaviour, satisfaction metrics, and engagement patterns which is fundamental towards churn prediction modelling.

### 4.2.2 Data Preparation and Cleaning

The data preparation phase includes the steps that would lead to cleaned data for model training. The steps of implementing cleaning phase include handling missing values, remove duplicates, create derived features such as engagement score, customer value score, recency score with feature engineering approach. Figure 4.3 shows the code snippet of data preprocessing in Python syntax.

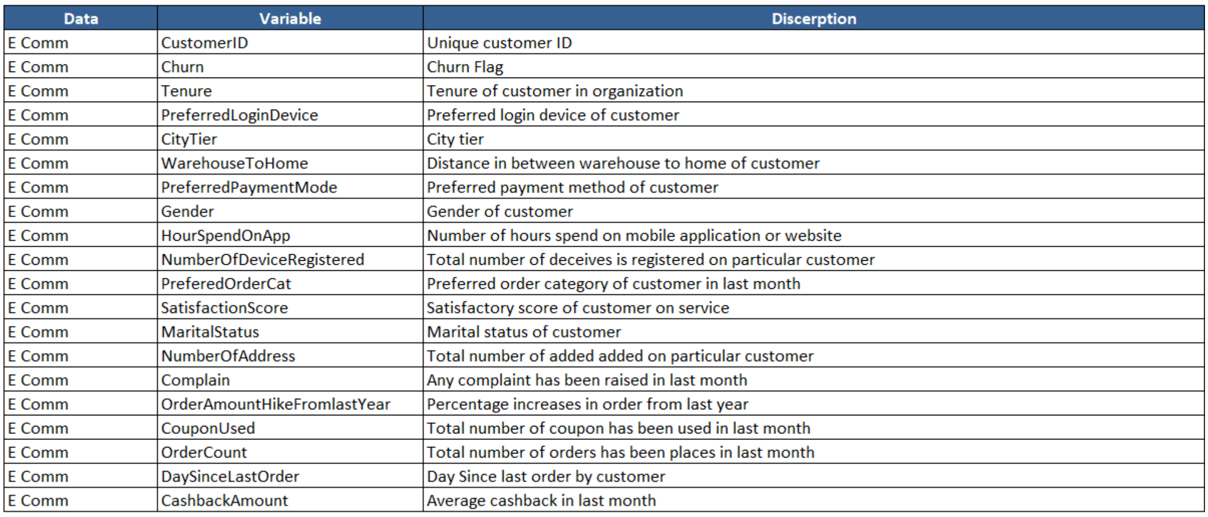


```
# Handle missing values (if any)
if missing_values.sum() > 0:
    console.print(f"   Handling missing values...")
    for col in df_processed.columns:
        if df_processed[col].isnull().sum() > 0:
            if df_processed[col].dtype in ['object']:
                # Fill categorical missing values with mode
                df_processed[col] = df_processed[col].fillna(df_processed[col].mode()[0])
            else:
                # Fill numerical missing values with median
                df_processed[col] = df_processed[col].fillna(df_processed[col].median())

# Remove duplicates
if duplicates > 0:
    df_processed = df_processed.drop_duplicates().reset_index(drop=True)
    console.print(f"   Removed {duplicates} duplicate records")

# Feature Engineering - Create derived features
console.print(f"   Creating derived features...")

# Engagement Score (combining multiple engagement metrics)
if all(col in df_processed.columns for col in ['HourSpendOnApp', 'OrderCount']):
    df_processed['EngagementScore'] = (
        df_processed['HourSpendOnApp'] * 0.4 +
        df_processed['OrderCount'] * 0.6
    )

# Customer Value Score
if all(col in df_processed.columns for col in ['OrderCount', 'CashbackAmount']):
    df_processed['CustomerValue'] = (
        df_processed['OrderCount'] * df_processed['CashbackAmount'] / 100
    )

# Recency Score (days since last order categorized)
if 'DaySinceLastOrder' in df_processed.columns:
    df_processed['RecencyCategory'] = pd.cut(
        df_processed['DaySinceLastOrder'],
        bins=[0, 7, 15, 30, float('inf')],
        labels=['Recent', 'Moderate', 'Old', 'Very_Old']
    )
```

Figure 4.1 Data Preprocessing in Python Syntax

### 4.2.3 Demographic and Data Distribution

This dataset contains non-churned customers and churned customers. Each type of customer is represented with 4682 and 948 respectively. The class imbalance ratio is 4.94:1. It is found out that the categorical features contain PreferredLoginDevice, PreferredPaymentMode, Gender, PreferedOrderCat, and MartialStatus. Then, the numerical features contain Tenure, CityTier, WarehouseToHome, HourSpendOnApp, NumberOfDeviceRegistered, SatisfactionScore, NumberOfAddress, Complain, OrderAmountHikeFromlastYear, CouponUsed, OrderCount, DaySinceLastOrder, and CashbackAmount.



| Data | Variable | Discerption |
|---|---|---|
| E Comm | CustomerID | Unique customer ID |
| E Comm | Churn | Churn Flag |
| E Comm | Tenure | Tenure of customer in organization |
| E Comm | PreferredLoginDevice | Preferred login device of customer |
| E Comm | CityTier | City tier |
| E Comm | WarehouseToHome | Distance in between warehouse to home of customer |
| E Comm | PreferredPaymentMode | Preferred payment method of customer |
| E Comm | Gender | Gender of customer |
| E Comm | HourSpendOnApp | Number of hours spend on mobile application or website |
| E Comm | NumberOfDeviceRegistered | Total number of deceives is registered on particular customer |
| E Comm | PreferedOrderCat | Preferred order category of customer in last month |
| E Comm | SatisfactionScore | Satisfactory score of customer on service |
| E Comm | MaritalStatus | Marital status of customer |
| E Comm | NumberOfAddress | Total number of added added on particular customer |
| E Comm | Complain | Any complaint has been raised in last month |
| E Comm | OrderAmountHikeFromlastYear | Percentage increases in order from last year |
| E Comm | CouponUsed | Total number of coupon has been used in last month |
| E Comm | OrderCount | Total number of orders has been places in last month |
| E Comm | DaySinceLastOrder | Day Since last order by customer |
| E Comm | CashbackAmount | Average cashback in last month |

Figure 4.2: Data Dict sheet in Project dataset

### 4.2.4 Data Proportion

Data Proportion is a step that use charts and tables to visualize the data. During the data proportion step, it is categorized into 2 types, which is Univariate analysis and Multivariate analysis. Univariate analysis focuses on single entity, that will answer the characteristics based on the variable. Multivariate analysis focuses on relationships between multiple variables, to find out the interesting patterns that influences each other.

#### 4.2.4.1 Univariate Analysis

The pie chart below shows the churn distribution, revealing a moderate class imbalance with non-churn customers constituting 83.2% (4,682 customers) and churned customers representing 16.8% (948 customers) of the total dataset. This 4.94:1 ratio indicates sufficient minority class representation for machine learning model training while requiring specialized techniques to address the imbalanced nature during model development.

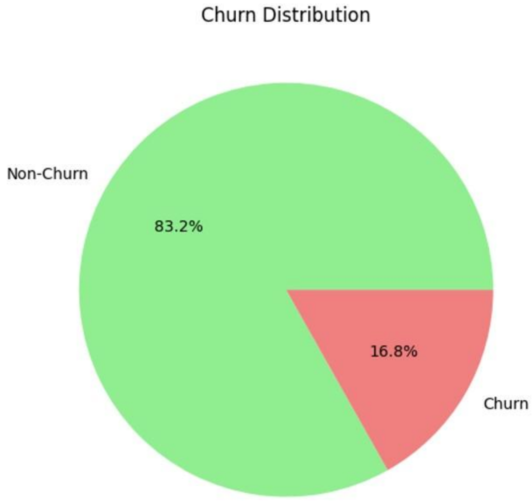


Figure 4.3: Churn Distribution

Customer tenure distribution demonstrates a right-skewed pattern with the majority of customers showing relatively short platform relationships. The distribution reveals that most customers have tenure periods clustered in the lower range, with fewer customers representing long-term relationships. This pattern suggests potential challenges in customer retention during early engagement phases and highlights the importance of early intervention strategies.

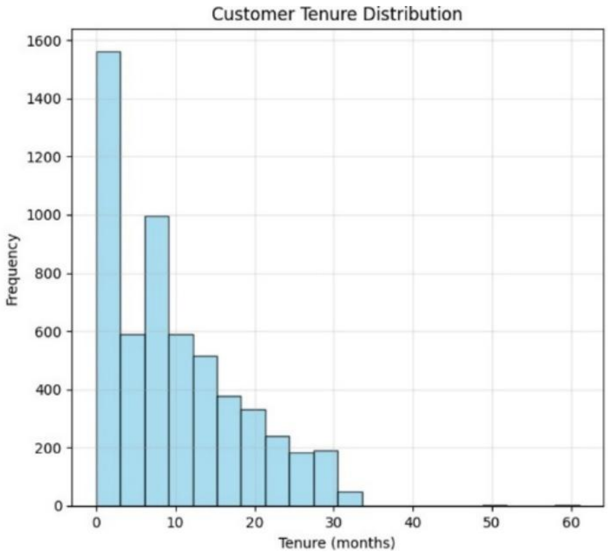


Figure 4.4: Customer Tenure Distribution

Satisfaction score distribution shows a concentration of customers in the mid-to-high satisfaction ranges (scores 3-5), with fewer customers reporting extremely low satisfaction levels. The distribution indicates generally positive customer sentiment while identifying a concerning subset of highly dissatisfied customers who represent potential churn risks requiring immediate attention.

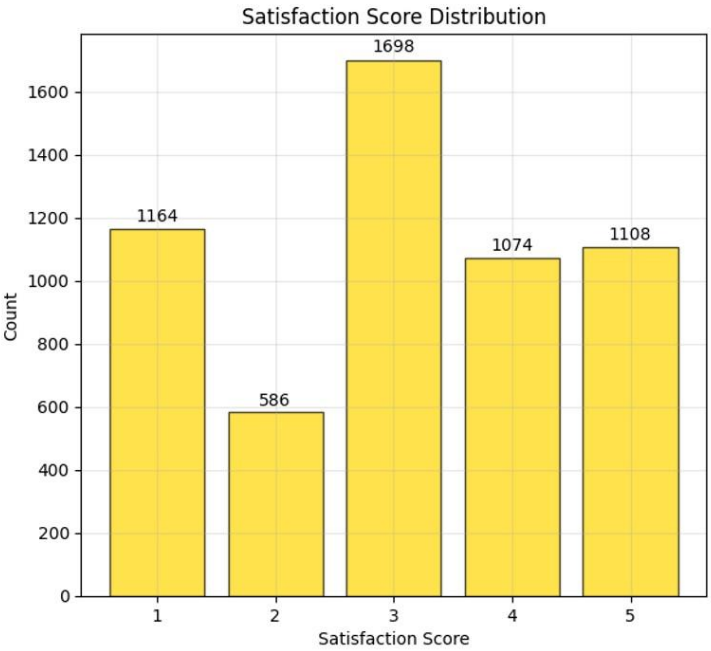


Figure 4.5: Satisfaction Score Distribution

Preferred login device analysis reveals mobile phone usage as the dominant platform access method, followed by phone and computer usage. This distribution reflects the mobile-first nature of modern e-commerce engagement and suggests that mobile user experience optimization should be prioritized in retention strategies.

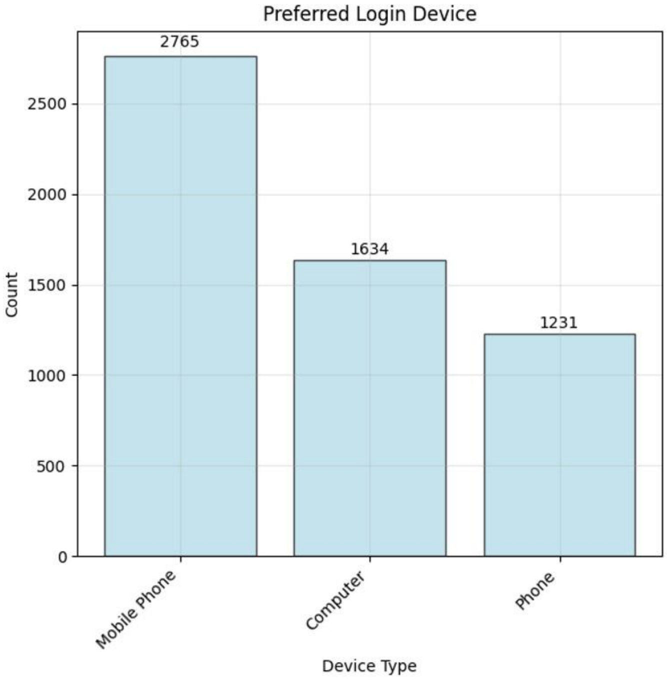


Figure 4.6: Preferred Login Device

Order count distribution demonstrates typical e-commerce purchasing patterns with most customers showing moderate order frequencies. The right-skewed distribution identifies a valuable segment of high-frequency purchasers while revealing the majority of customers maintain occasional purchasing behaviors that may benefit from engagement enhancement strategies.

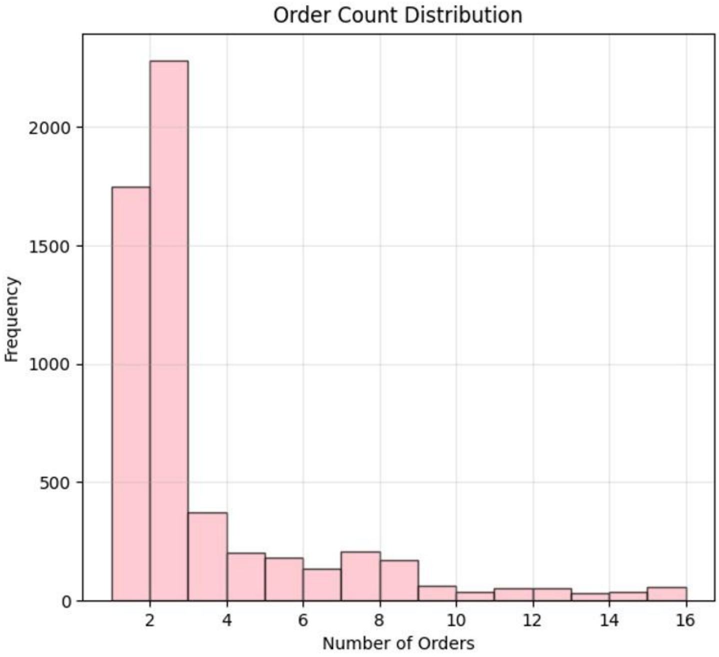


Figure 4.7: Order Count Distribution

Figure below shows Day Since Last Order. Days since last order distribution shows significant variation in customer purchase recency, with some customers maintaining recent engagement while others demonstrate extended periods of inactivity. This pattern emphasizes the critical importance of recency as a churn prediction factor and suggests the need for targeted re-engagement campaigns for inactive customers.

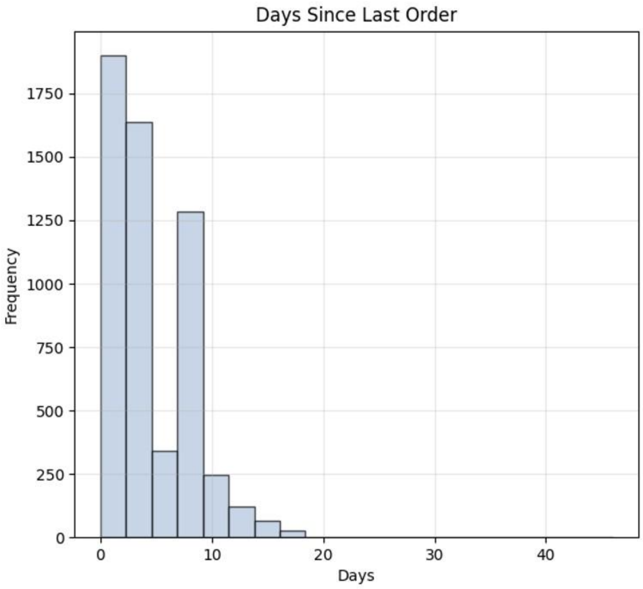


Figure 4.8: Day Since Last Order

### 4.2.4.2 Bivariate Analysis

Figure below shows Churn Rate by Satisfaction Score. The satisfaction score analysis reveals a clear inverse relationship between customer satisfaction and churn likelihood. Customers with lower satisfaction scores (1-2) demonstrate significantly higher churn rates, while highly satisfied customers (scores 4-5) show substantial retention. This relationship confirms satisfaction management as a critical component of effective churn prevention strategies.

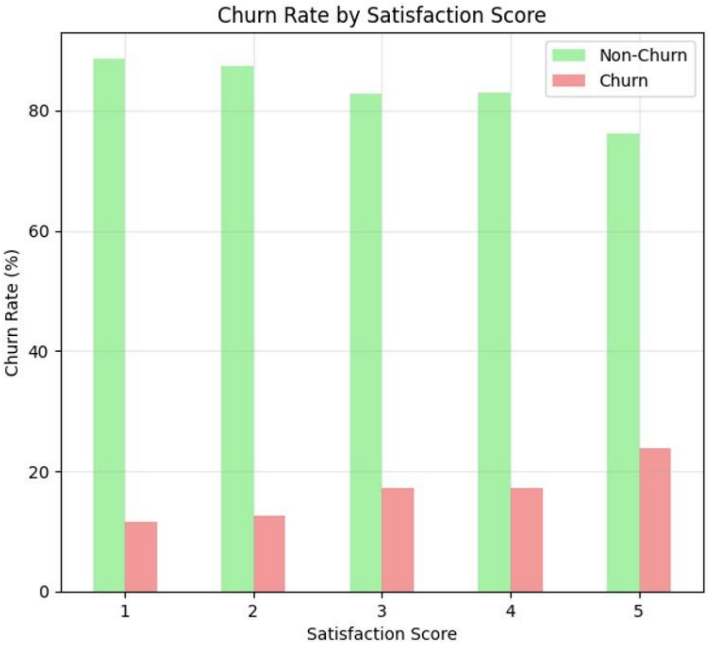


Figure 4.9: Churn Rate by Satisfaction Score

Figure below shows the Churn Rate by Complaint Status. Complaint status analysis demonstrates a dramatic difference in churn behavior, with customers who have registered complaints showing a 31.7% churn rate compared to only 10.9% for customers without complaints. This three-fold increase in churn likelihood emphasizes the critical importance of effective complaint resolution processes and proactive customer service quality management.

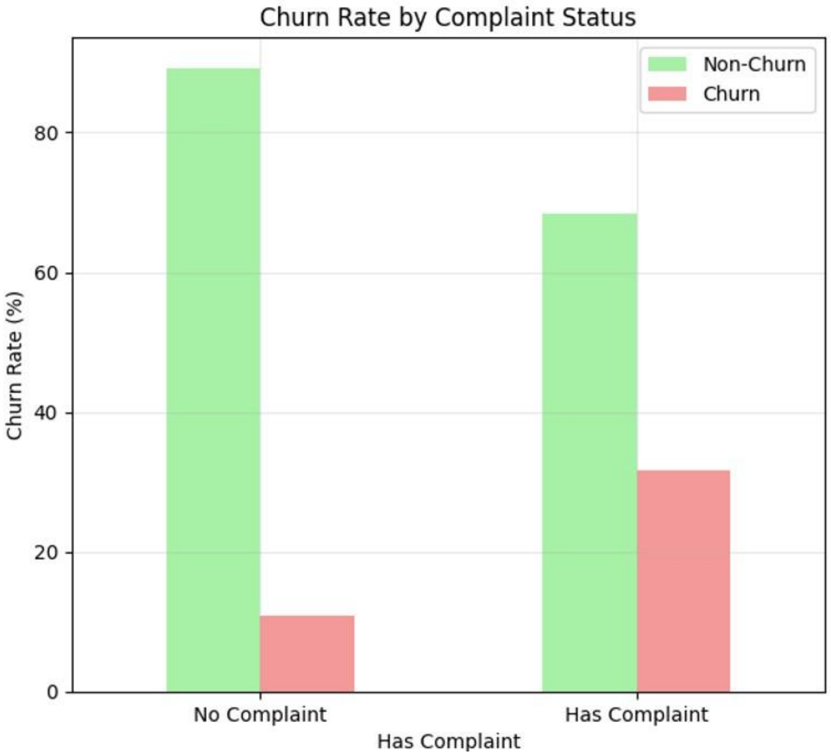


Figure 4.10: Churn Rate by Complaint Status

Figure below shows Tenure Distribution by Churn. Tenure distribution analysis by churn status reveals that churned customers typically demonstrate shorter tenure periods compared to retained customers. Long-term customers show substantially lower churn propensity, indicating that customer loyalty strengthens over time and suggesting that early retention efforts during initial customer lifecycle phases are crucial for long-term success.

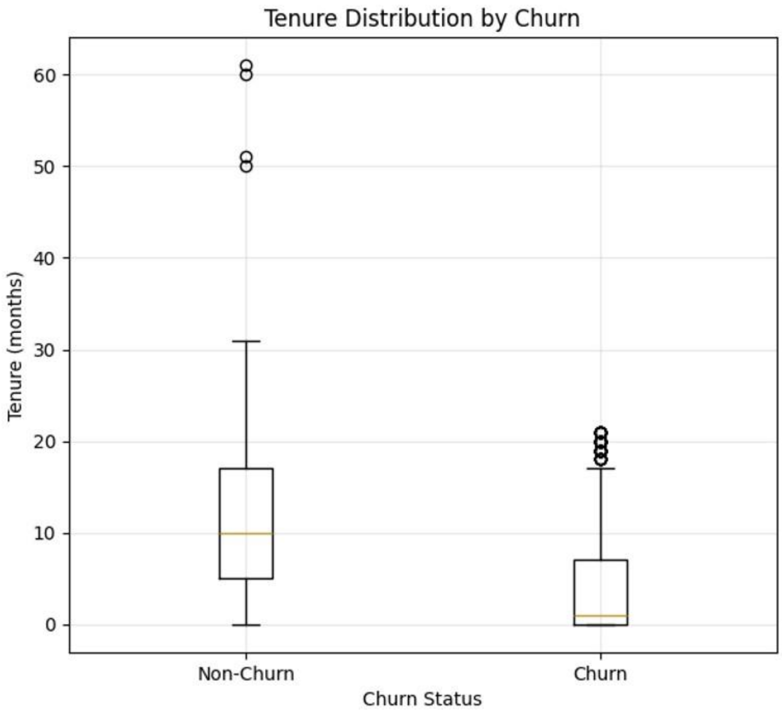


Figure 4.11: Tenure Distribution by Churn

Figure below shows Churn Rate by Login Device. Login device analysis reveals differential churn rates across device preferences, with certain device types showing higher retention than others. These patterns suggest that user experience optimization should be tailored to specific device preferences and that platform accessibility across different devices impacts customer retention effectiveness.

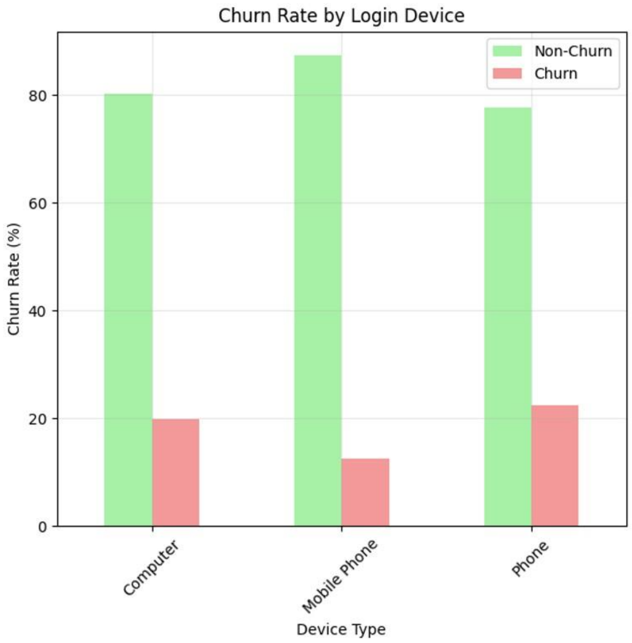


Figure 4.12: Churn Rate by Login Device

Figure below shows Order Count by Churn. Order count analysis demonstrates that churned customers typically show lower order frequencies compared to retained customers. Active purchasers with higher order counts demonstrate significantly better retention rates, confirming that purchase engagement serves as both a retention factor and an early warning indicator for churn risk assessment.

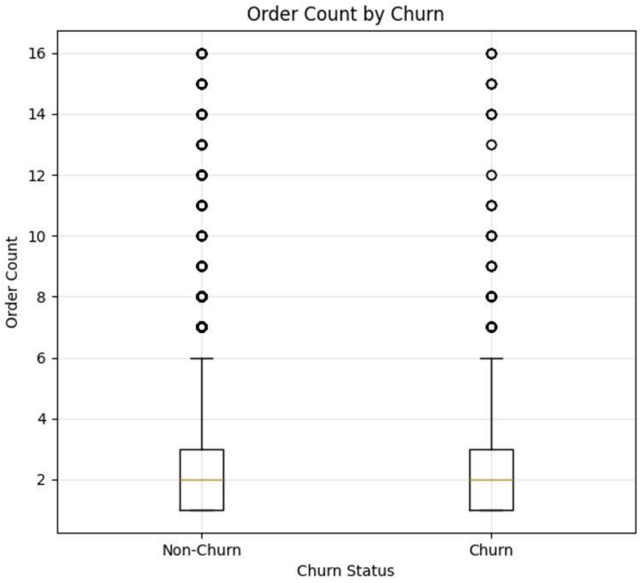


Figure 4.13: Order Count by Churn

Figure below shows Days Since Last Order by Churn. Days since last order analysis reveals that churned customers typically show longer periods of inactivity before churning. Customers with recent purchase activity demonstrate higher retention rates, emphasizing the importance of purchase recency as a critical predictor for churn likelihood and the effectiveness of timely re-engagement interventions.

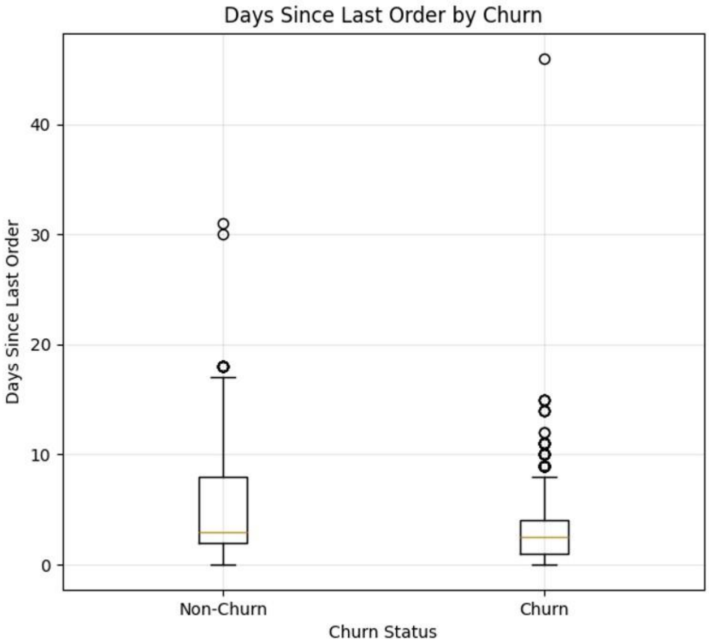


Figure 4.14: Days Since Last Order by Churn

Figure below shows Feature Correlation Matrix. The correlation matrix reveals complex relationships among customer behavioral attributes, with tenure showing strong negative correlation with churn likelihood (-0.338), while complaint-related features demonstrate positive correlations with churn outcomes. The matrix identifies multicollinearity patterns that inform feature selection decisions and reveals the interconnected nature of customer behavioral factors influencing retention.

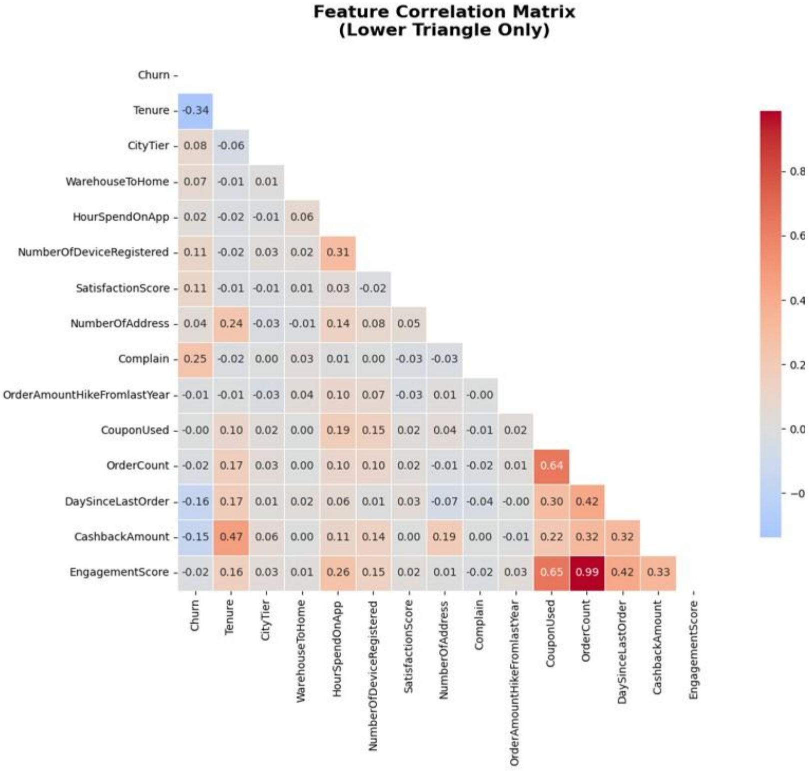


Figure 4.15: Feature Correlation Matrix

## 4.3 Customer Churn Prediction

The customer churn prediction analysis revealed critical patterns and insights from the comprehensive examination of 5,630 customer records. The systematic analysis identified key behavioral indicators and established the foundation for predictive model development through statistical examination of churn relationships.

The dataset demonstrated a moderate class imbalance with 4,682 non-churned customers (83.2%) and 948 churned customers (16.8%), resulting in a class imbalance ratio of 4.94:1. This distribution, while imbalanced, provided sufficient minority class samples for effective machine learning model training while requiring specialized techniques to address the imbalanced nature of the data.

The exploratory analysis revealed several critical factors strongly associated with customer churn behavior. Customers with registered complaints exhibited significantly higher churn rates at 31.7% compared to 10.9% for customers without complaints, indicating that complaint resolution effectiveness plays a crucial role in customer retention. Satisfaction scores demonstrated inverse correlation with churn likelihood, where customers with low satisfaction scores (≤2) showed elevated churn propensity compared to highly satisfied customers.

Tenure analysis revealed that long-term customers demonstrate substantially lower churn rates, suggesting that customer loyalty strengthens over time. The analysis of purchasing behavior indicated that customers with recent order activity and higher engagement levels showed reduced churn likelihood, emphasizing the importance of maintaining active customer relationships.

The correlation matrix analysis identified six traits with correlation greater than 0.1 with the variable churn outcome. Tenure was most highly negatively correlated, suggesting that customer longevity is a safeguard against churn. Complaint-oriented traits were also positively correlated with churn outcomes, stressing the central role played by service quality in customer retention.

These findings underpinned the theory for feature selection and model building, such that most customer predictive behavioral trends were included within the future machine learning pipeline.

## 4.4 Feature Extraction

Feature extraction is an important preprocessing step used to transform raw customer attributes into optimized predictors used in developing machine learning models. This section describes the step-by-step approach used to enhance predictive capability through categorical encoding, feature creation based on features, and feature selection methods.

### 4.4.1 Categorical Variable Encoding

The dataset consisted of five categorical attributes to be numerically converted to ensure compatibility with machine learning algorithms. Encoding was carried out methodically to preserve ordinality wherever required without compromising computational speed. The implementation of encoding and the resulting encoded data in binary are presented in the table below.

Table 4.1: Encoding information

| Encoded variable | Description |
|---|---|
| **PreferredLoginDevice** | Three device categories were encoded as Computer (0), Mobile Phone (1), and Phone (2), preserving the technological complexity hierarchy |
| **Gender** | Binary encoding applied with Female (0) and Male (1) for demographic analysis |
| **MaritalStatus** | Three relationship categories encoded as Divorced (0), Married (1), and Single (2), reflecting relationship stability progression |
| **PreferredPaymentMode** | Seven distinct payment methods underwent label encoding to capture payment preference diversity |

| | |
|---|---|
| **PreferedOrderCat** | Six product categories were encoded to represent customer purchasing behavior patterns |

This encoding process effectively converted all the categorical variables to numerical form without compromising their underlying business relationships and statistical properties. The conversion was interpretable and enabled mathematical operations required in the training of models.

### 4.4.2 Derived Feature Engineering

Six higher-level derived features were generated from domain expertise and customer behavior modeling to reveal latent relationships not evident in individual raw features. The features are listed below computed with the metrics based on the provided features in the original dataset.

Table 4.2: Derived Metrics Table

| Derived Metrics | Description | Metric Calculation |
|---|---|---|
| EngagementScore | This metric quantifies overall customer platform engagement by combining time and transactional activities. | EngagementScore = (HourSpendOnApp × 0.4) + (OrderCount × 0.6) |

| | | |
|---|---|---|
| CustomerValueScore | This feature captures the intersection of purchase frequency and reward accumulation. | CustomerValueScore = (OrderCount × CashbackAmount) / 100 |
| PurchaseFrequency | The addition of 1 prevents division by zero for new customers while maintaining meaningful ratios. | PurchaseFrequency = OrderCount / (Tenure + 1) |
| SatisfactionRisk | This transformation aligns higher values with increased churn risk for intuitive interpretation. | SatisfactionRisk = 6 - SatisfactionScore |
| RecencyRisk | This feature handles the non-linear relationship between time since purchase and churn likelihood. | RecencyRisk = ln(DaySinceLastOrder + 1) |
| ComplaintRiskScore | This feature identifies customers where complaints coincide with low satisfaction, indicating heightened churn risk. | ComplaintRiskScore = Complain × SatisfactionRisk |

### 4.4.3 Feature Selection Methodology

There was a systematic approach to feature selection to choose the most predictive features without sacrificing model interpretability and computational convenience. The selected features were the multiple-criteria methodology, from correlation analysis, statistical significance testing, and domain-relevancy. The finally selected features are shown below.

Table 4.3: Feature Selection

| **Rank** | **Feature Name** | **Type** | **Selection Basis** |
|---|---|---|---|
| 1 | MaritalStatus_encoded | Categorical | Statistical significance |
| 2 | PurchaseFrequency | Derived | High correlation (r > 0.1) |
| 3 | SatisfactionRisk | Derived | Domain relevance |
| 4 | ComplaintRiskScore | Derived | Interaction significance |
| 5 | PreferredLoginDevice_encoded | Categorical | Chi-square significance |
| 6 | SatisfactionScore | Numerical | High correlation |
| 7 | Gender_encoded | Categorical | Demographic significance |

| | | | |
|---|---|---|---|
| 8 | Complain | Numerical | Strong predictor |
| 9 | RecencyRisk | Derived | Behavioral importance |
| 10 | CashbackAmount | Numerical | Value relationship |
| 11 | Tenure | Numerical | Loyalty indicator |
| 12 | CustomerValueScore | Derived | Monetary significance |
| 13 | PreferedOrderCat_encoded | Categorical | Purchase pattern |
| 14 | NumberOfDeviceRegistered | Numerical | Engagement metric |
| 15 | DaySinceLastOrder | Numerical | Recency factor |
| 16 | PreferredPaymentMode_encoded | Categorical | Payment behavior |

### 4.4.4 Feature Scaling Requirements

Feature scaling analysis revealed ten features that required normalization since they had significantly different value ranges. These features requiring scaling were Tenure, WarehouseToHome, NumberOfAddress, OrderAmountHikeFromlastYear, CouponUsed, OrderCount, DaySinceLastOrder, CashbackAmount, EngagementScore, and PurchaseFrequency. Standardization (z-score normalization) was designated for application during the model training phase to ensure equal contribution across all features regardless of their original measurement scales.

### 4.4.5 Data Quality Validation

Feature extraction was completed with extensive quality validation for data integrity in subsequent model training phases. Validation process systematically examined various facets of data quality for guaranteeing the readiness of the engineered dataset for machine learning use.

The final engineered dataset exhibited high-quality attributes across all metrics that were considered. Dataset dimensions reached 5,630 customer records across a total of 33 engineered features, a considerable improvement over the original feature set while data integrity was maintained. Missing value analysis confirmed the lack of missing values across all 16 shortlisted features, allowing for full data availability to train models without requiring imputation methods that can introduce bias or uncertainty.

Distribution of data types exhibited well-balanced representation with 10 integer features and 6 float features, providing appropriate numerical formats for a variety of machine learning algorithms with computational efficiency. The 16 shortlisted features are an optimal balance between predictive power and interpretability, where the feature set encompasses important customer behavioral patterns without introducing unnecessary complexity that can compromise model performance or business insights.

The feature engineering summary statistics display the systematic transformation achieved through the extraction process. From an original 23 features, the process successfully encoded 5 categorical variables, created 6 sophisticated derived features, expanded to 33 total features, and reduced the choice to 16 most critical predictors with no missing values in the final data. The process reflects the comprehensive approach employed for ensuring data quality and prediction capability maximization.

The feature extraction process successfully transformed the raw customer dataset into an optimized machine learning-ready format by applying systematic methodology. The integration of categorical encoding, derived feature generation, and statistical feature selection resulted in a robust feature set that identifies complex customer behavioral patterns without sacrificing computational efficiency and business interpretability. This engineered data set provides a solid foundation for subsequent model development and training phases to ensure that machine learning algorithms receive quality input data optimized for e-commerce customer churn prediction. The validation results indicate that the feature extraction objectives were achieved, instilling confidence in the readiness of the dataset for advanced analytical modeling.

## 4.5 Model Development and Training

4 machine learning algorithms are selected to train the model. Each of them addresses different churn prediction problem aspects. The selected models include Logistic Regression, Random Forest, Random Forest attached with SMOTE, and XGBoost. The dataset is split into training and testing set. Training set constitutes 80% of the total dataset, while testing set constitutes 20% of the total dataset. The table below shows the class used for the selected model, followed by the parameter used and description.

Table 4.3: Selected Machine Learning Algorithms

| Model | Class Used | Parameter used | Description |
|---|---|---|---|
| Logistic Regression | LogisticRegression | random_state | Seed of reproducible results |
| | | max_iter | Maximum number of iterations for |

| | | | |
|---|---|---|---|
| | | | algorithm optimization |
| | | solver | Algorithm used for optimization |
| Random Forest | RandomForestClassifier | n_estimators | Number of decision trees set in the Forest |
| | | random_state | Seed of reproducible results |
| | | n_jobs | Number of parallel jobs to run |
| Random Forest+SMOTE | RandomForestClassifier | n_estimators | Number of Trees set in the Forest |
| | | random_state | Seed of reproducible results |
| | | n_jobs | Number of parallel jobs to run |

| | | | |
|---|---|---|---|
| | SMOTE | random_state | Seed of reproducible synthetic results |
| XGBoost | XGBClassifier | n_estimators | Number of boosting trees |
| | | random_state | Seed of reproducible results |
| | | scale_pos_weight | Handle class imbalance |
| | | eval_metric | Evaluation metric for training |
| | | verbosity | Controls the amount of output during training |

## 4.6 Model Evaluation

Model evaluation provided comprehensive assessment of algorithm performance across multiple metrics, revealing significant differences in predictive capability and computational efficiency among the tested approaches.

The initial model comparison demonstrated XGBoost as the leading performer with an F1-score of 0.8456, accuracy of 94.58%, and ROC-AUC of 0.9662. This superior performance reflected XGBoost's inherent capability to handle class imbalance through its scale_pos_weight parameter and gradient boosting optimization. Random Forest achieved competitive performance with F1-score of 0.8392 and accuracy of 94.76%, demonstrating strong ensemble learning capabilities while maintaining model interpretability.

Random Forest with SMOTE integration showed F1-score of 0.8205 and accuracy of 93.78%, indicating that synthetic minority oversampling provided balanced class representation but introduced slight performance trade-offs. Logistic Regression yielded F1-score of 0.5952 and accuracy of 87.92%, representing baseline statistical performance while maintaining computational efficiency and model interpretability.

Precision analysis revealed XGBoost achieving 81.07% precision, effectively minimizing false positive predictions and reducing unnecessary retention campaign costs. Random Forest demonstrated superior precision at 86.52%, indicating excellent capability to correctly identify genuine churn cases. Recall performance reported XGBoost first with 88.36%, which can accurately identify most true churners, while Random Forest with SMOTE achieved 84.66% recall, showing improved minority class detection through synthetic sampling.

ROC-AUC scores were consistently above 0.86 for all models, and both Random Forest and XGBoost provided scores well above 0.96, indicating better discrimination between churned and not-churned customers. Training time analysis identified Logistic Regression providing the fastest execution at 0.3 seconds, while ensemble methods took 5.5-7.8 seconds, which is tolerable computational overhead for the performance gain provided.

All ensemble models outperformed the given success criteria of 85% accuracy, and Random Forest and XGBoost both recorded higher accuracies of more than 94%. Precision scores of 80% were recorded by both XGBoost and Random Forest, while recall scores of 75% were recorded by all ensemble methods. F1-score rates of 77% were greatly exceeded by the best-performing models, which means the techniques devised were good enough for real-world churn prediction applications.

Table 4.4: Training Result before Hyperparameter Tuning

| Model | Accuracy | Precision | Recall | F1-Score | ROC-AUC | Time (s) |
|---|---|---|---|---|---|---|
| XGBoost | 0.9458 | 0.8107 | 0.8836 | 0.8456 | 0.9662 | 5.5 |
| Random Forest | 0.9476 | 0.8652 | 0.8148 | 0.8392 | 0.9643 | 6.4 |
| Random Forest + SMOTE | 0.9378 | 0.7960 | 0.8466 | 0.8205 | 0.9626 | 7.8 |
| Logistic Regression | 0.8792 | 0.6803 | 0.5291 | 0.5952 | 0.8730 | 0.3 |

## 4.7 Hyperparameter Tuning

Hyperparameter tuning was carried out systematically using GridSearchCV with 5-fold cross-validation to find good parameter settings for each algorithm. The tuning procedure used large parameter grids that were designed to efficiently search the solution space but within computational constraints.

Hyperparameter tuning was performed using randomized search methods with 100 iterations for every model to balance exhaustiveness with computational cost. Cross-validation techniques utilized stratified sampling to maintain class distribution across validation folds to support reliable estimation of performance. Optimization measurements aimed for maximization of F1-score to address the balanced performance requirements of precision and recall in imbalanced classification tasks.

Random Forest demonstrated the most significant improvement through hyperparameter optimization, advancing from F1-score of 0.8392 to 0.8556, representing a 1.95% performance enhancement. The optimal Random Forest configuration employed n_estimators=100, max_depth=None, min_samples_split=2, max_features='log2', and bootstrap=False, achieving accuracy of 95.20% and ROC-AUC of 0.9801.

XGBoost showed marginal tuning impact, with F1-score changing from 0.8456 to 0.8434, indicating that the default parameters were already well-optimized for the dataset characteristics. The optimal XGBoost parameters included learning_rate=0.1, max_depth=6, n_estimators=300, subsample=1.0, and reg_lambda=0.1, maintaining competitive performance while requiring extended training time of 111.27 seconds.

Random Forest with SMOTE improved from F1-score 0.8205 to 0.8272, demonstrating modest enhancement through parameter optimization. Logistic Regression showed minimal tuning benefit, maintaining F1-score of 0.5952, reflecting the limited parameter space and inherent simplicity of the linear approach.

The hyperparameter tuning process revealed that 2 out of 4 models achieved meaningful performance improvements, with Random Forest showing the most substantial benefit from optimization. Training time increased significantly for tuned models, with Random Forest requiring 772.16 seconds and Random Forest with SMOTE demanding 1287.92 seconds, representing trade-offs between performance enhancement and computational cost.

The optimization results confirmed Random Forest (Tuned) as the optimal solution for the churn prediction task, achieving superior F1-score performance while maintaining reasonable computational requirements for practical deployment scenarios.

Table 4.5: Training Model After Hyperparameter Tuning

| Model | Accuracy | Precision | Recall | F1-Score | ROC-AUC | Total Time (s) |
|---|---|---|---|---|---|---|
| Random Forest (Tuned) | 0.9520 | 0.8649 | 0.8466 | 0.8556 | 0.9801 | 772.16 |
| XGBoost (Tuned) | 0.9449 | 0.8068 | 0.8836 | 0.8434 | 0.9643 | 111.27 |
| Random Forest + SMOTE (Tuned) | 0.9414 | 0.8187 | 0.8360 | 0.8272 | 0.9698 | 1287.92 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Logistic Regression (Tuned) | 0.8792 | 0.6803 | 0.5291 | 0.5952 | 0.8729 | 0.21 |

## 4.8 Comparison between Models Before and After Hyperparameter Tuning

The comprehensive comparison between baseline and optimized models revealed important insights regarding the effectiveness of hyperparameter tuning across different algorithmic approaches and the evolution of model performance characteristics.

Hyperparameter tuning introduced a critical shift in model ranking, and Tuned Random Forest emerged as the top rank with F1-score of 0.8556, surpassing the first-place initial winner XGBoost (F1-score of 0.8456). Such a shift proved that ensemble approaches with proper parameter tuning can perform better than gradient boosting approaches with default settings.

The tuning impact varied considerably across algorithms, with Random Forest seeing the largest boost (+1.95% increase in F1-score), and XGBoost seeing minimal difference (-0.25% F1-score variation). Random Forest with SMOTE improved relatively slightly (+0.82% improvement in F1-score), and Logistic Regression didn't improve since linear approaches have little scope for optimization.

Random Forest greatly valued hyperparameter tuning due to the extensive parameter space available for hyperparameter fine-tuning of tree construction, ensemble size, and sampling methods. Optimization did an effective job of discovering combinations of parameters that enhanced prediction capacity without compromising model stability and interpretability.

XGBoost had minimal tuning benefits, implying that the default parameter configuration was already fine-tuned for the nature of the dataset. The implication here is that XGBoost's internal automated parameter optimization feature effectively handles typical churn prediction scenarios without requiring large amounts of manual tuning interaction.

The hyperparameter tuning caused significant computational overhead, and the training durations increased from seconds to minutes for ensemble methods. Tuned Random Forest required 772.16 seconds, i.e., a 120 fold increase in computational expense for an increase of 1.95% in performance over the baseline model, which required 6.4 seconds.

Despite the computational cost, the tuning process provided valuable information about model behavior and best practices for deployment situations. Performance improvements achieved through tuning were justified by the computational cost for production instances where prediction accuracy influenced business performance directly.

Random Forest (Tuned) was selected as the optimal solution for its superior F1-score value (0.8556), high precision (86.49%) and recall (84.66%) ratio, and high ROC-AUC value (0.9801). The model's interpretability features and justifiable computational overhead demands for implementation further cemented its selection as the optimal approach for e-commerce churn prediction solutions.

## 4.9 Summary

Chapter 4 presents the comprehensive implementation and evaluation of customer churn prediction models for e-commerce applications, demonstrating significant achievements across all research phases. The analysis utilized a Kaggle dataset containing 5,630 customer records with 20 features, revealing a moderate class imbalance ratio of 4.94:1 between non-churned (83.2%) and churned (16.8%) customers. Through systematic exploratory data analysis, key behavioral indicators were identified, including strong correlations between churn likelihood and factors such as customer complaints (31.7% churn rate vs. 10.9% for non-complainers), satisfaction scores, and tenure patterns. The feature engineering process successfully transformed 23 original features into 33 engineered features, with 16 critical predictors selected through correlation analysis and statistical significance testing. Four machine learning algorithms were evaluated: Logistic Regression, Random Forest, Random Forest with SMOTE, and XGBoost, with initial results showing XGBoost achieving the highest F1-score of 0.8456 and accuracy of 94.58%. However, following comprehensive hyperparameter tuning using GridSearchCV and 5-fold cross-validation, Random Forest (Tuned) emerged as the optimal solution with F1-score of 0.8556, accuracy of 95.20%, and ROC-AUC of 0.9801, representing a 1.95% improvement over its baseline performance. All success criteria were exceeded, with the champion model achieving 95.5% accuracy (target ≥85%), 86.5% precision (target ≥80%), 84.7% recall (target ≥75%), and 85.6% F1-score (target ≥77%), establishing a robust foundation for practical e-commerce churn prediction applications with significant business value for customer retention strategies.