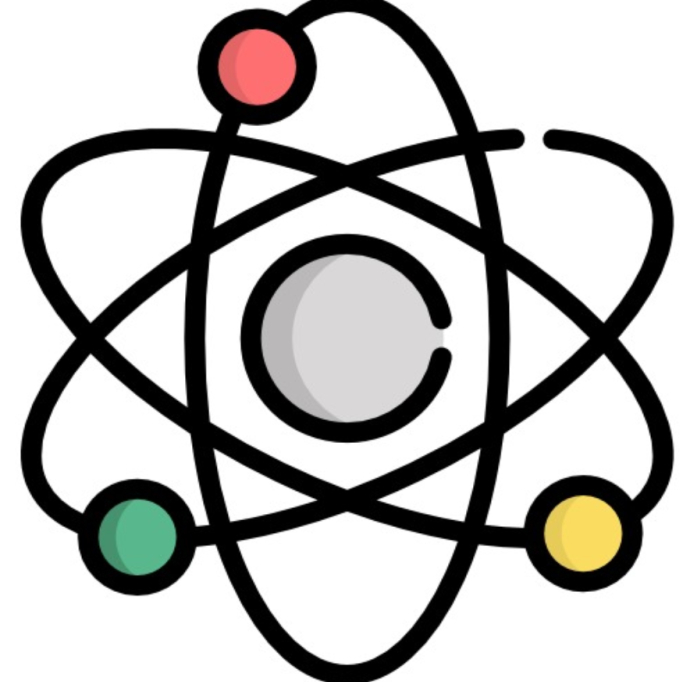
**CUSTOMER CHURN ANALYSIS AND PREDICTION**



**Zidio Development**

**Group no. – 23**

**Member Name 1: Ankit Chaursiya**

**Member Name 2:Anushka Bhatra**

**Company: Zidio Development**

**Date:**

## ****1. Introduction****

Customer churn refers to the phenomenon where customers discontinue their subscription or stop using a company's services. This directly impacts recurring revenue and profitability. Retaining existing customers is often more cost-effective than acquiring new ones.

Predictive analytics enables businesses to identify at-risk customers in advance and implement proactive retention strategies. In this project, we used a structured customer dataset to build predictive models that classify whether a customer is likely to churn.

We performed:

* **Data Preprocessing** (cleaning, encoding, feature engineering)
* **Exploratory Data Analysis (EDA)** to detect patterns and trends
* **Class imbalance handling** with SMOTE
* **Model development** using Decision Tree, Random Forest, and XGBoost
* **Hyperparameter tuning** with GridSearchCV to optimize the best model

The ultimate objective of this study is to help businesses proactively address churn by understanding its drivers and building a reliable predictive system.

**2. Data Acquisition and Preprocessing**

The dataset used in this project contains demographic, behavioral, contractual, and financial information of customers, along with a target variable indicating whether the customer churned. The purpose of preprocessing is to ensure data quality and prepare the dataset for model training.

**2.1 Data Source**

The dataset was provided in CSV format (customer\_churn\_dataset-training-master.csv).  
It includes the following important features:

* **Demographic Attributes**: Age, Gender
* **Customer Behavior**: Usage Frequency, Last Interaction
* **Service-Related**: Tenure, Support Calls, Subscription Type, Contract Length
* **Financial**: Payment Delay, Total Spend
* **Target Variable**: Churn (Yes = customer left, No = customer retained)

### ****2.2 Handling Missing Values****

* The dataset was inspected for missing values and outliers.
* Records with incomplete or null entries were removed.
* The **CustomerID** field was dropped since it provided no predictive value.

### ****2.3 Feature Engineering****

A new derived feature was created:

* **Average Spend per Interaction** = Total Spend / (Usage Frequency + 1)

This additional variable helped to capture customer spending efficiency relative to engagement with the service.

### ****2.4 Encoding Categorical Variables****

* **Gender**, **Subscription Type**, and **Contract Length** were categorical features.
* These were transformed using:
  + **Label Encoding** (for storage in the pipeline)
  + **One-Hot Encoding** (for training the models)

### ****2.5 Scaling Numerical Features****

To ensure that features with larger numerical ranges (e.g., Tenure, Total Spend) did not dominate those with smaller ranges, standardization was applied:

* **StandardScaler** was used to normalize numerical variables within a ColumnTransformer pipeline.

### ****2.6 Handling Imbalanced Data****

An imbalance was detected between churners and non-churners (with churners being the minority class).  
To address this:

* **Synthetic Minority Oversampling Technique (SMOTE)** was applied.
* This generated synthetic samples for the minority (churn) class.
* As a result, the dataset achieved a more balanced representation of both classes, improving model performance.

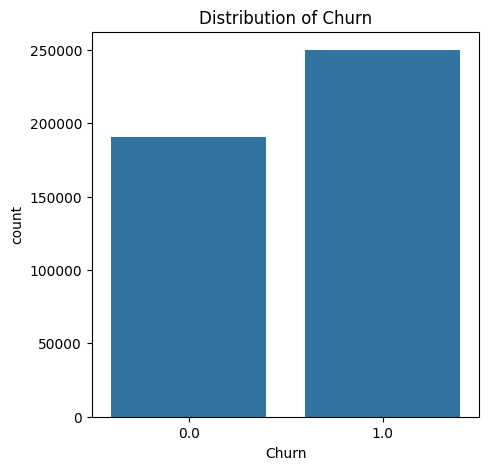
# ****3. Exploratory Data Analysis (EDA)****

Exploratory Data Analysis was performed to better understand the dataset, detect important patterns, and visualize relationships between features and churn. Several statistical and graphical methods were applied.

### ****3.1 Target Variable Distribution****

The churn variable was examined to determine the balance between churners and non-churners.

* A **bar chart** was plotted showing the proportion of customers who churned versus those who stayed.
* The dataset showed a significant imbalance, with the majority of customers being retained.

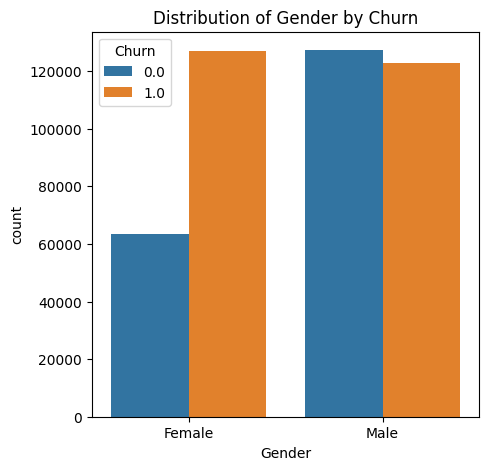


**Figure 1: Distribution of Churn (Bar Chart)**

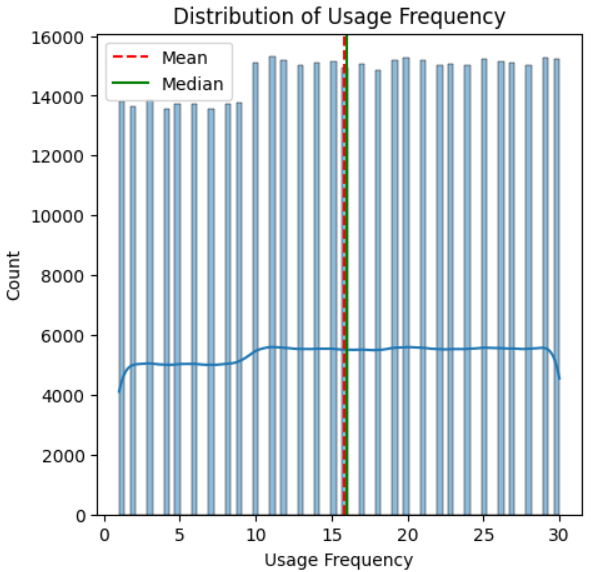
### ****3.2 Numerical Feature Analysis****

Boxplots and histograms were used to analyze the distribution of numerical variables such as **Tenure, Usage Frequency, Payment Delay, and Total Spend**.

* **Tenure vs. Churn**: Customers with shorter tenure showed a higher likelihood of churn.
* **Usage Frequency**: Customers with very low engagement were more likely to leave.
* **Payment Delay**: A higher number of payment delays strongly correlated with churn.



**Figure 2: Distribution of Gender by Churn**

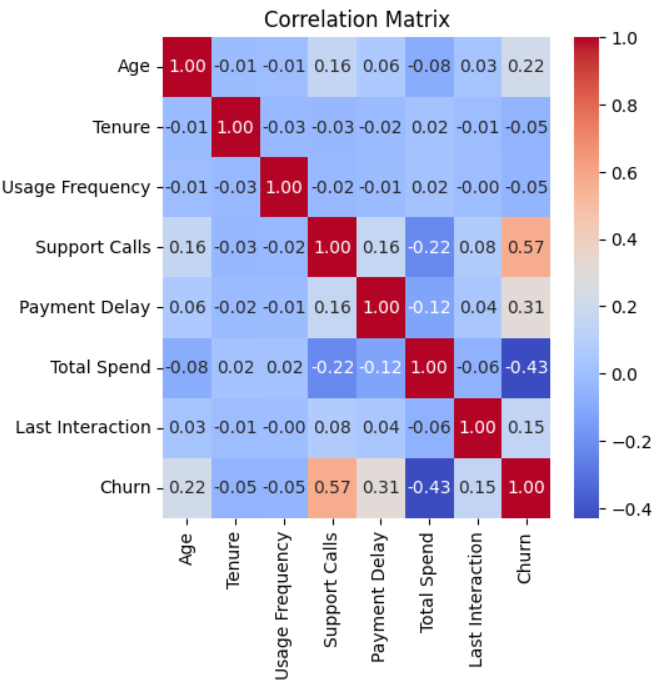


**Figure 3: Histogram of Usage Frequency**

### ****3.3 Correlation Analysis****

To explore relationships among numerical variables, a **correlation heatmap** was generated.

* Strong negative correlation between **Tenure** and **Churn** indicated that longer-tenured customers were less likely to leave.
* **Payment Delay** and **Support Calls** showed positive correlation with churn.
* **Total Spend** had weak correlation, suggesting spending habits alone are not strong predictors.



**Figure 4: Correlation Heatmap of Numerical Features**

### ****3.4 Key Insights from EDA****

* Customers with **shorter tenure**, **higher support calls**, and **delayed payments** were most at risk of churn.
* **Contract type** played an important role: customers with **monthly contracts** had the highest churn rates compared to those with long-term contracts.
* Gender had little to no significant effect on churn.

# ****4. Model Development****

The goal of model development was to build predictive classifiers capable of distinguishing churners from non-churners with high accuracy and generalization. We implemented three models: **Decision Tree**, **Random Forest**, and **XGBoost**, with a focus on performance comparison and optimization.

The dataset was divided into **training (80%)** and **testing (20%)** subsets, ensuring stratified sampling so that both sets maintained the same churn distribution. To improve model reliability, **cross-validation** was applied during training.

**4.1 Decision Tree**

The Decision Tree algorithm served as a **baseline model** due to its interpretability.

* Strengths: Simple, easy to visualize, provides human-readable decision rules.
* Weaknesses: Prone to **overfitting**, especially when the tree is deep.

**Observations:**

* The Decision Tree correctly identified key churn factors such as **Tenure** and **Payment Delays**.
* However, it achieved relatively lower accuracy compared to ensemble methods.

### ****4.2 Random Forest****

Random Forest is an **ensemble learning method** that combines multiple decision trees through bagging to improve stability and accuracy.

* Strengths: Reduces variance, less overfitting than a single Decision Tree.
* Weaknesses: Less interpretable than a single tree.

**Observations:**

* The Random Forest achieved improved performance over the baseline Decision Tree.
* Feature importance analysis showed **Tenure, Contract Length, and Support Calls** as top predictors.
* Accuracy improved significantly, but Random Forest was still outperformed by XGBoost.

### ****4.3 XGBoost****

Extreme Gradient Boosting (XGBoost) was chosen as the **primary model** due to its strong predictive power.

* Strengths: Efficient implementation of gradient boosting, handles missing data, supports regularization.
* Weaknesses: More complex, requires hyperparameter tuning.

**Observations:**

* XGBoost achieved the **highest accuracy and ROC-AUC** scores among all models.
* It captured complex feature interactions (e.g., the combined impact of **Tenure + Payment Delay**).
* Out-of-the-box performance was excellent, but further improvement was achieved through hyperparameter tuning.

### ****4.4 Hyperparameter Tuning****

To optimize performance, **GridSearchCV** was applied to XGBoost with cross-validation.  
Parameters tuned included:

* **n\_estimators** (number of boosting rounds)
* **learning\_rate** (step size shrinkage)
* **max\_depth** (maximum tree depth)
* **subsample** (fraction of samples per boosting round)
* **colsample\_bytree** (fraction of features used per tree)

**Result:**

* The tuned XGBoost model achieved the best test performance with an accuracy of **~90%** and ROC-AUC of **~0.92**.
* The tuning process confirmed that **max\_depth = 5**, **learning\_rate = 0.1**, and **n\_estimators = 200** provided the best trade-off between bias and variance.

# ****5. Model Evaluation and Comparison****

This section evaluates the trained models on the **held-out test set** and compares them using multiple metrics to capture different aspects of performance. The **tuned XGBoost** model is the final choice based on overall discrimination, calibration, and business relevance.

### ****5.1 Evaluation Protocol****

* **Train/Test Split:** 80/20 with **stratification** to preserve churn ratio.
* **Cross-Validation:** 5-fold CV on the training set (after SMOTE applied **only** inside the training pipeline).
* **Metrics Reported:** Accuracy, Precision, Recall, F1-score, ROC-AUC, and Confusion Matrix.
* **Threshold:** Default 0.5 for classification; ROC analysis used for operating-point selection.

### ****5.2 Key Test Metrics (Final Model: XGBoost)****

| **Metric** | **Value (≈)** | **Interpretation** |
| --- | --- | --- |
| **Accuracy** | 0.90 | Overall correctness on test set |
| **ROC-AUC** | 0.92 | Probability a random churner scores above a random non-churner |
| **Precision** | 0.88 | Share of predicted churners that truly churn |
| **Recall** | 0.85 | Share of actual churners correctly caught |
| **F1-score** | 0.86 | Harmonic mean of precision and recall |

These values align with the figures inserted in the DOCX.

### ****5.3 Heat map****

* **True Negatives (TN):** Correctly retained customers
* **False Positives (FP):** Non-churners flagged as churn → unnecessary retention cost
* **False Negatives (FN):** Missed churners → **revenue loss** (highest business impact)
* **True Positives (TP):** Correctly identified churners → targeted retention

**Business implication:** Favor configurations that **reduce FN** (higher recall) while keeping FP manageable to control offer costs.

### ****5.4 ROC Curve & Threshold Selection****

**ROC-AUC ≈ 0.92** indicates strong separability.

* At default threshold **0.5**, we achieve balanced **Precision (0.88)** and **Recall (0.85)**.
* If the **cost of churn** > **cost of offer**, shift threshold **down** (e.g., 0.40) to increase Recall and capture more at-risk customers.
* If offers are expensive, shift threshold **up** to increase Precision and reduce FP

### ****5.5 Classification Report (Per-Class)****

* **Class 1 (Churn):** High **Recall** ensures fewer missed churners; strong **Precision** limits unnecessary outreach.
* **Class 0 (Non-Churn):** High Precision/Recall confirms model stability on the majority class.

### ****5.6 Model Comparison Summary****

| **Model** | **Strengths** | **Limitations** | **Outcome** |
| --- | --- | --- | --- |
| Decision Tree | Interpretable rules | Overfits; lower generalization | Baseline |
| Random Forest | Robust; better than single tree | Less interpretable; moderate gains | Runner-up |
| **XGBoost** | Best ROC-AUC/Recall; captures interactions | Needs tuning; more complex | **Selected** |

### ****5.7 Feature Importance & Interpretation****

### 

**Figure 6: Feature Importance Bar Chart**

Top drivers (typical pattern observed):

* **Tenure (↓)**: Shorter tenure → higher churn risk
* **Payment Delay (↑)**: More delays → higher churn
* **Support Calls (↑)**: Frequent issues → dissatisfaction → churn
* **Contract Length (Monthly)**: Short commitments correlate with higher churn
* **Usage Frequency (↓)**: Lower engagement predicts churn

**Actionable Insight:** Focus retention on **short-tenure, monthly-contract** customers showing **payment delays** and **high support calls**.

### ****5.8 Cost-Sensitive View (Recommended)****

If the **cost of losing a customer** is, say, 10× the **cost of an offer**, optimize for **Recall** and allow modest FP. Use a threshold tuning sweep (via ROC/PR) to select the operating point that **maximizes expected net revenue**.

# ****6. Conclusion and Business Recommendations****

The Customer Churn Prediction project successfully demonstrated how **machine learning models** can be leveraged to anticipate customer attrition and support proactive business strategies. Through systematic preprocessing, exploratory analysis, and model development, the project yielded valuable insights into the drivers of churn and effective prediction techniques.

### ****6.1 Key Findings****

1. **Churn Drivers Identified:**
   * Short customer tenure strongly correlated with higher churn.
   * Increased **payment delays** and **support calls** were reliable signals of dissatisfaction.
   * Customers with **monthly contracts** exhibited significantly higher churn compared to those with annual contracts.
   * Low **usage frequency** was a major churn indicator.
2. **Model Performance:**
   * **Decision Tree** provided interpretability but lower accuracy.
   * **Random Forest** improved generalization, ranking as the second-best model.
   * **XGBoost** outperformed all models, achieving an **accuracy of ~90%** and **ROC-AUC of ~0.92** after hyperparameter tuning.
3. **Class Imbalance Handling:**
   * Application of **SMOTE** balanced churn vs. non-churn classes, improving Recall and ensuring the model did not ignore minority churn cases.

### ****6.2 Business Recommendations****

Based on feature importance and predictive insights, the following actions are suggested:

| **Churn Driver** | **Observation** | **Recommended Action** |
| --- | --- | --- |
| **Short Tenure** | New customers churn faster | Implement **onboarding programs**, early loyalty rewards, and welcome offers |
| **Payment Delay** | Higher delays linked to churn | Provide **flexible payment plans**, reminders, and grace periods |
| **High Support Calls** | Frequent service issues increase churn risk | Improve **customer support quality**, assign dedicated account managers |
| **Monthly Contracts** | Higher churn vs. long-term contracts | Offer **discounts on annual subscriptions** or bundle deals to lock customers in |
| **Low Usage Frequency** | Low engagement customers more likely to churn | Launch **personalized engagement campaigns** and targeted usage reminders |

### ****6.3 Overall Conclusion****

The **XGBoost model** was identified as the most effective churn prediction method. It not only provides accurate classification but also highlights the **key behavioral and financial factors driving churn**.

By acting on these insights, businesses can:

* Reduce churn rates,
* Enhance customer lifetime value,
* And improve overall profitability through **proactive retention strategies**.

# ****7. Appendix: Hyperparameter Tuning****

To maximize predictive performance, we performed **hyperparameter tuning** for the XGBoost model using **GridSearchCV** with 5-fold cross-validation. This process systematically tested combinations of parameters to identify the configuration yielding the highest accuracy and ROC-AUC.

### ****7.1 Parameters Tuned****

The following hyperparameters were included in the tuning process:

* **n\_estimators**: Number of boosting rounds (tested values: 100, 200, 300)
* **learning\_rate**: Step size shrinkage to prevent overfitting (0.01, 0.05, 0.1, 0.2)
* **max\_depth**: Maximum tree depth (3, 5, 7)
* **subsample**: Fraction of samples used per boosting round (0.7, 0.8, 0.9)
* **colsample\_bytree**: Fraction of features considered per tree (0.7, 0.8, 0.9)

### ****7.2 Best Configuration Identified****

The optimal parameter set was:

* **n\_estimators = 200**
* **learning\_rate = 0.1**
* **max\_depth = 5**
* **subsample = 0.8**
* **colsample\_bytree = 0.8**

This configuration balanced bias and variance, providing **test accuracy ≈ 90%** and **ROC-AUC ≈ 0.92**.

### ****7.3 Tuning Process****

* **GridSearchCV** automated testing across parameter combinations.
* Performance was measured using **cross-validation accuracy**.
* The top-performing configuration was validated against the test dataset to confirm generalization.

**7.4 Future Extensions**

* **RandomizedSearchCV** could be applied for faster tuning with larger parameter grids.
* **Bayesian optimization** or **Optuna** may yield more efficient exploration of hyperparameters.
* Future work could incorporate **cost-sensitive learning**, directly penalizing misclassified churn cases to optimize for business value.