Customer Churn Prediction in Telecom Industry

Karthika Vellingiri

Data Science Department, Bellevue University

DSC 550 – T303 Data Mining

Professor Brett Werner

28 February 2025

# Introduction

Customer churn is a significant challenge in the telecom industry, where retaining existing customers is often more cost-effective than acquiring new ones. High churn rates lead to increased costs for customer acquisition and lost revenue. This project focuses on developing a churn prediction model to help telecom businesses reduce churn, improve profitability, and maintain a competitive edge in a rapidly evolving market. The results aim to offer actionable insights that drive informed, data-driven decision-making across the organization.

Addressing customer churn is crucial because it directly affects a company's profitability and long-term sustainability. Customer churn negatively impacts profitability, as acquiring a new customer can cost **five times more** than retaining an existing one. Additionally, a high churn rate signals dissatisfaction, leading to brand reputation challenges. An effective churn prediction model provides telecom companies with actionable insights to:

* Improve customer retention strategies.
* Reduce revenue loss by addressing customer pain points early.
* Optimize marketing efforts by focusing on high-risk customers.

To gain buy-in from executives and decision-makers, this project should be positioned as a **data-driven initiative that directly impacts revenue growth and cost efficiency**. By identifying at-risk customers early, the company can implement personalized interventions (discounts, loyalty programs, better customer support) to prevent churn. This **reduces marketing and acquisition costs** while improving overall customer satisfaction.

The dataset used for this project was obtained from Kaggle, specifically the Telco Customer Churn dataset, [Source: <https://www.kaggle.com/datasets/blastchar/telco-customer-churn> ]. This dataset contains customer demographic information along with churn labels that indicate whether a customer has left the service. The dataset includes features such as customer tenure, monthly charges, payment method, and service usage.

# Milestones Overview

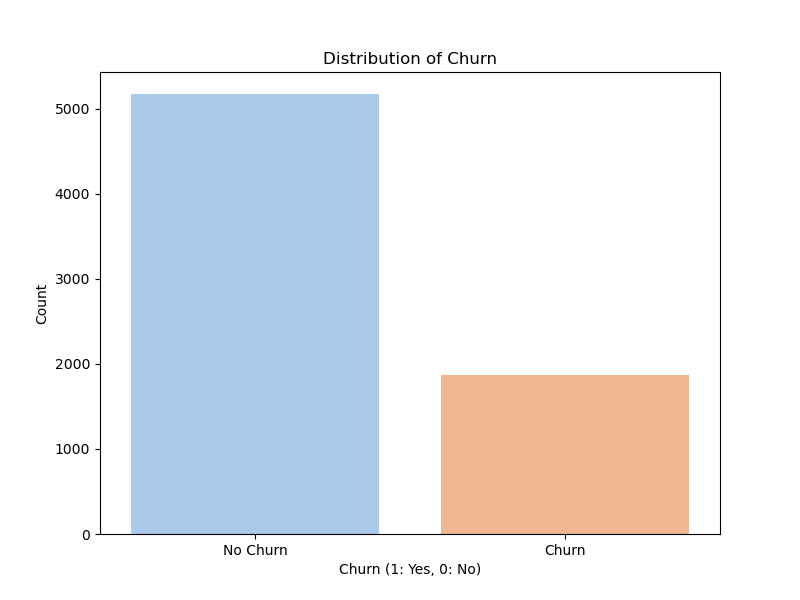
## Milestone 1: Exploratory Data Analysis (EDA)

In this milestone, the focus was on understanding the dataset and preparing it for further analysis. The dataset contains customer information such as demographic details, service usage patterns, and account features. The goal was to uncover trends, patterns, and insights that could help in predicting customer churn.

### Visualisations

#### Distribution of Target Variable (Churn)

A bar chart was created to illustrate the imbalance in the target classes, showing a higher proportion of non-churned customers compared to churned customers. This imbalance could impact model performance and may necessitate techniques such as resampling or class-weight adjustments to enhance predictive accuracy.



#### Feature Relationships

Features like tenure and monthly charges show a strong correlation with churn. Gaining an understanding of these relationships helps guide feature engineering and the selection of appropriate models moving forward.

##### Monthly Charges vs. Churn

A chart of a number of rectangular objects

AI-generated content may be incorrect.

Customers who churn tend to have higher monthly charges, as shown in the **Monthly Charges vs. Churn** box plot. This insight could be valuable in targeting high-value customers for retention efforts.

##### Tenure vs. Churn

The **Tenure vs. Churn** box plot reveals that long-term customers (with higher tenure) are less likely to churn. This suggests that improving customer retention strategies for newer customers could be vital in reducing churn.

A diagram of a chart

AI-generated content may be incorrect.

The analysis of these graphs establishes a solid foundation for developing hypotheses about customer churn, which can be validated using predictive models in the subsequent stages of the project.

### Insights Gained

The dataset exhibited several imbalances in both the target variable and certain features, which influenced the choices made during preprocessing. Additionally, key patterns, such as a potential correlation between Tenure and Churn, were uncovered.

## Milestone 2: Data Preparation

This phase involved cleaning and preprocessing the data to ensure it is ready for model building. These data preparation steps are essential in ensuring the dataset is well-structured and ready for building effective predictive models.

### Data Preparation Steps

#### Missing Data Handling

By addressing missing values through imputation or removal, we ensure that the dataset doesn't contain gaps that could lead to inaccurate or biased model predictions. Handling missing data also helps in retaining valuable information, preventing data loss.

#### Feature Engineering

New features like AvgMonthlySpend and TenureGroup were created to enhance the dataset. AvgMonthlySpend was computed as the ratio of TotalCharges to Tenure, and TenureGroup categorized customers into different tenure brackets.

#### Encoding Categorical Variables

Encoding categorical variables like 'gender' and 'contract type' into numerical values using one-hot encoding ensures that machine learning algorithms can interpret them correctly, as they typically require numerical input. This step transforms qualitative data into a quantitative format suitable for model processing.

#### Scaling Numerical Features

Min-Max Scaling was applied to numerical features like Monthly charges and Tenure. This transformation scales the features to a fixed range, typically between 0 and 1, ensuring that all features have the same scale. This is crucial for models that are sensitive to feature magnitudes, like K-Nearest Neighbors and Logistic Regression, as it prevents variables with larger ranges from dominating the model's performance.

#### Train-Test Split

The data preprocessing steps are integrated into a **pipeline**, ensuring consistent transformations for both training and test sets. While the code does not explicitly include the train-test split, it would typically follow after preprocessing to evaluate the model’s generalization to unseen data. The pipeline ensures that all transformations, such as scaling and encoding, are applied consistently across both training and test datasets.

### Outcome

The dataset was cleaned, transformed, and prepared for training, with no missing values and all features in the appropriate format. These steps help prepare the data in a way that minimizes bias, enhances model accuracy, and ensures robust performance when evaluated on new, unseen data.

## Milestone 3: Model Building and Evaluation

In this milestone, the goal was to select, build, and evaluate a predictive model for customer churn using the prepared dataset. We chose **Logistic Regression** as the baseline model due to its suitability for binary classification problems, interpretability, and efficiency. The following steps were carried out:

### Model Selection

**Logistic Regression** was selected as the model for the following reasons:

* It is a commonly used and well-understood algorithm for binary classification tasks.
* It works well for problems where the decision boundary is approximately linear, such as predicting customer churn.
* It is computationally efficient and easy to interpret, making it ideal for understanding how different features affect churn prediction.

### Model Training

After preprocessing the data, the dataset was split into training (80%) and test (20%) sets using the train\_test\_split function. This ensures that the model is trained on one subset of the data and evaluated on an unseen subset to assess its generalization ability. The Logistic Regression model was then trained using the training set, which had undergone necessary preprocessing steps, including handling missing values, encoding categorical variables, and scaling numerical features. The model learned the relationship between the features (such as tenure, monthly charges, contract type, etc.) and the target variable Churn.

### Model Optimization

To optimize the model, we performed hyperparameter tuning using **Grid Search Cross-Validation** to find the best value for the regularization strength parameter (C). This process helps to balance the model's complexity and avoid overfitting, ultimately improving its generalizability.

### Model Evaluation

The evaluation of both the baseline and hyperparameter-tuned models reveals important insights into their performance, especially in terms of predicting churn.

#### Baseline Model Evaluation

* **Accuracy**: The baseline model achieved an accuracy of 80.7%, indicating that it correctly predicted the churn status for most customers. However, accuracy alone can be misleading in the context of imbalanced classes.
* **Confusion Matrix:** From the confusion matrix, we observe that the model accurately predicted 955 customers who did not churn but missed 191 churned customers (false negatives). The model also predicted 81 non-churned customers as churned (false positives).

A diagram of heatmap

AI-generated content may be incorrect.

* **Precision & Recall**: The baseline model showed a precision of 83.3% for "Not Churn" customers, meaning that the majority of customers predicted as "Not Churn" indeed stayed. However, the recall for the "Churn" class was 48.8%, suggesting that many actual churns were missed by the model. This highlights the model’s limited ability to identify at-risk customers effectively.
* **F1-Score**: The F1-score of 57.23% for the "Churn" class reflects the imbalance between precision and recall. While precision is high, recall lags, suggesting that the model is focusing more on not missing non-churn customers.
* **ROC AUC**: The AUC of 0.8520 indicates that the model has good discrimination ability between churned and non-churned customers across various thresholds, though there is still room for improvement.

A graph with a line drawn on it

AI-generated content may be incorrect.

#### Hyperparameter-Tuned Model Evaluation

* **Confusion Matrix:** In the confusion matrix for the tuned model, we observe that the model predicted 961 non-churned customers correctly but missed 198 churned customers (false negatives). It also predicted 75 non-churned customers as churned (false positives).

A graph with numbers and a chart

AI-generated content may be incorrect.

* **Accuracy**: The tuned model maintained a similar accuracy (80.6%) to the baseline model, suggesting that hyperparameter tuning did not significantly affect the overall correctness of predictions.
* **Precision & Recall**: The precision for the "Churn" class increased slightly to 70%, indicating a stronger focus on identifying churned customers. However, the recall decreased slightly to 46.9%, meaning the model missed more churn cases compared to the baseline. This trade-off suggests that the tuning adjusted the model to be more conservative in predicting churn, prioritizing fewer false positives.
* **F1-Score**: The F1-score for the "Churn" class dropped to 56.18%, which reflects the decrease in recall, although precision improved.

**ROC AUC**: The AUC for the tuned model was very similar to the baseline (0.8462), maintaining the model's ability to distinguish between churned and non-churned customers, with little impact from the hyperparameter tuning.

A graph with a green line

AI-generated content may be incorrect.

### Comparison & Insights

The performance of the baseline Logistic Regression model was compared with the hyperparameter-tuned version:

* **Accuracy**: Both models showed similar performance, with the baseline model achieving an accuracy of 80.7% and the tuned model slightly lower at 80.6%.
* **Precision and Recall for Churn**: The tuned model demonstrated a slight improvement in precision (0.7000) but a small decrease in recall (0.4692), suggesting a trade-off where the model became more precise but missed some churn cases.
* **ROC AUC**: The AUC scores for both models were nearly identical, indicating strong and stable performance in discriminating between churned and non-churned customers.
* **Impact of Hyperparameter Tuning**: The tuning process improved precision but slightly reduced recall. However, the overall effect on model performance was modest, and both models were effective at distinguishing between churn and non-churn customers.
* **Confusion Matrix**: Comparing both models, we see that the tuned model slightly improved the True Positives (175 vs. 182) but also led to more False Negatives (198 vs. 191). The baseline model performed better in detecting churned customers, but the tuned model improved the precision for the "Churn" class by reducing the False Positives (75 vs. 81). This reflects the model's trade-off between precision and recall.

# Conclusion

The analysis and model development revealed that the baseline Logistic Regression model achieved an accuracy of **80.7%**, but with a **recall of 48.8% for churn**, indicating that a significant number of churn cases were missed. Hyperparameter tuning led to a slight improvement in **precision (70%)**, but at the cost of reducing **recall (46.9%)**. Both models exhibited strong **ROC AUC scores (~0.85)**, demonstrating good overall discrimination between churned and non-churned customers.

However, the model is not yet suitable for deployment due to its low recall. Further improvements are necessary, including adjusting decision thresholds, experimenting with alternative models such as Random Forest and XGBoost, and addressing data imbalance through oversampling or other techniques. Key challenges include effectively handling data imbalance, adapting to dynamic customer behavior, improving feature selection, and ensuring model interpretability for stakeholders.

With further refinements, the model has the potential to become a valuable tool for minimizing churn and improving profitability. Integrating the model into real-time systems and optimizing the trade-off between retention costs and prediction accuracy will be crucial for long-term effectiveness. Continuous monitoring and regular updates will be essential to maintaining performance and maximizing impact post-deployment.