A COMPARATIVE STUDY OF MACHINE LEARNING ALGORITHMS FOR CUSTOMER CHURN PREDICTION

Anam Vakil
Student
vaki0005@algonquinlive.com

ABSTRACT

This project analyzes customer churn behavior within a telecommunications dataset to identify the key drivers of churn and determine the most reliable predictive machine learning model. Using a dataset of over 7,000 customer records, extensive preprocessing, feature engineering, and exploratory data analysis (EDA) were conducted to understand demographic, service-level, contract-based, and financial attributes influencing churn. Initial single train-test split evaluation suggested that the Random Forest classifier performed well. However, after applying rigorous evaluation through 5-fold cross-validation across three train–test splits (80:20, 70:30, and 85:15), Logistic Regression consistently demonstrated higher mean accuracy and significantly lower variance, indicating superior stability. Findings confirm that churn behavior follows identifiable patterns (e.g., month-to-month contracts, electronic check payments, shorter tenures), and that Logistic Regression is the most reliable model for churn prediction.

1. INTRODUCTION

Customer churn is a critical challenge for telecommunication providers, as losing existing customers directly reduces revenue and increases the cost of acquiring new ones. Retaining customers is significantly more cost-effective than continuously onboarding new subscribers. Therefore, accurately identifying customers who are likely to churn enables telecom companies to intervene early through personalized retention strategies.

This project integrates Exploratory Data Analysis (EDA) with supervised machine learning to build a robust churn prediction system. The aim is to understand key drivers behind customer attrition, explore relationships among customer attributes, and evaluate multiple models to determine which provides the most stable and accurate churn classification. By combining statistical insights with predictive analytics, the study equips organizations with actionable intelligence to improve customer satisfaction, minimize churn, and enhance long-term profitability.

2. BUSINESS UNDERSTANDING

Reducing churn is a strategic priority in the telecommunications industry. Customers who leave not only reduce recurring revenue but also increase operational costs associated with onboarding replacements. Improving retention enhances customer lifetime value, stabilizes revenue forecasting, and strengthens market competitiveness.

2.1. Key Business Questions

This project addresses the following business-critical questions:

- Which customer attributes are most strongly associated with churn?
  (e.g., contract type, tenure, internet service type, payment method)
- Which customer groups exhibit the highest churn risk?
  (e.g., month-to-month contracts, electronic check payment users)
- Which machine learning model provides the most reliable churn predictions for real-world deployment?

2.2. Business Impact

Insights derived from this analysis support several data-driven retention strategies, including:

- Offering discounts or incentives to high-risk segments
- Encouraging long-term contract upgrades
- Enhancing customer experience for vulnerable groups
- Targeting proactive customer support interventions

The model outputs can help organizations optimize marketing spend, reduce operational losses, and deploy resources toward customers most likely to churn.

3. DATA SOURCING AND PREPARATION

The dataset used for this study is the Telco Customer Churn dataset, which consists of 7,043 customers and 21 attributes. The features span demographic information, service

subscriptions, payment methods, and financial metrics, along with the target variable Churn.

3.1. Dataset Attributes

1) **Demographics:**
   gender, SeniorCitizen, Partner, Dependents
2) **Service Subscriptions:**
   PhoneService, MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies
3) **Account Information:**
   tenure, Contract, PaperlessBilling, PaymentMethod
4) **Financial Variables:**
   MonthlyCharges, TotalCharges
5) **Target Variable:**
   Churn (Yes/No)

The dataset was imported into Python using Pandas, and preprocessing steps were applied to ensure data quality before modeling.

3.2. Key Data Preparation Steps

1) **Type Conversion:**
   TotalCharges contained numeric values stored as strings. It was converted to a float and missing values were imputed using the median.
2) **Missing Value Handling:**
   After conversion, the remaining null values were handled appropriately, ensuring no gaps in the dataset.
3) **Categorical Encoding:**
   All object-type variables were encoded using get_dummies(), preparing them for machine learning algorithms.
4) **Outlier Handling:**
   Tenure outliers among churned customers were removed after identifying extreme values via boxplots to reduce skewness and stabilize model performance.

This preparation ensured the dataset was clean, consistent, and suitable for both EDA and predictive modeling.

4. DATA EXPLORATION

All exploratory analysis was conducted in Python using Pandas, NumPy, Matplotlib, Seaborn, and scikit-learn. EDA provided insights into variable distributions, customer behavior, and patterns relevant to churn prediction.

4.1 Dataset Size and Structure

Using df.shape, the dataset was confirmed to contain 7,043 rows and 21 columns, indicating a sufficiently large dataset for building reliable classification models. This size supports robust training, testing, and cross-validation without risk of overfitting.

4.2 Summary Statistics – Numeric vs. Categorical Features

4.2.1 Numeric Features

Descriptive statistics were generated using the df.describe() function to summarize the central tendency and variability of the numerical attributes, including tenure, monthly charges, and total charges. The analysis showed that tenure ranged from 0 to 72 months, representing customers at different stages of their subscription lifecycle. Monthly charges exhibited a wide distribution, with higher values typically corresponding to customers subscribed to multiple services. Total charges displayed a noticeably skewed pattern, particularly among newer customers with low tenure. These observations helped validate data integrity and informed subsequent preprocessing decisions, highlighting the need for scaling and appropriate transformations prior to model development.

4.2.2 Categorical Features

Categorical attributes were summarized using the df.describe(include='object') function, which provided counts, unique category values, and the most frequently occurring labels. The distribution of these attributes revealed several important patterns. Month-to-month contracts were the most common and also exhibited the highest churn rates, indicating greater instability among short-term customers. Electronic check emerged as the dominant payment method and was strongly associated with elevated churn, suggesting that customers using this method may face higher billing friction. InternetService categories also demonstrated distinct behavioral patterns, with Fiber optic customers showing notably higher churn compared to DSL or non-internet users. No inconsistencies or formatting issues were detected across categorical fields. These insights were essential for identifying predictive relationships among service and account attributes and for preparing the variables for one-hot encoding during preprocessing.

4.3 Missing Values and Data Type Correction

A detailed audit of missing values revealed that the dataset contained 11 missing entries, all originating from the TotalCharges field. These missing values corresponded exclusively to customers with a tenure of zero months, indicating that billing had not yet commenced for those accounts.

Inspection of the dataset schema further showed that TotalCharges was incorrectly stored as a string-type attribute. This issue often arises when non-numeric characters or leading/trailing spaces appear in raw billing exports. The field was converted to a numeric format, during which invalid entries were identified and marked as missing.

To ensure consistency and avoid skewing the distribution, the missing values in TotalCharges were imputed using the **median** of the column. Median imputation was selected because billing values exhibit right-skewness, making the median a more robust measure than the mean. After correction, no missing values remained in the dataset.

4.4 Dropping Irrelevant Columns

The attribute customerID served only as a unique identifier and contained no predictive information relevant to churn modeling. As such, it was removed from the dataset to prevent unnecessary dimensionality and noise. Eliminating non-informative fields ensures that machine learning algorithms focus exclusively on features that contribute to predictive performance.

4.5 Outlier Analysis

Outlier detection was conducted on the numerical attributes tenure and MonthlyCharges to identify extreme values that could negatively impact model performance.

4.5.1 Tenure Outliers

Outlier analysis on the *tenure* variable revealed a small number of unusually high values (70+ months). These points appeared as statistical outliers in the boxplot because they lie far from the majority of the distribution. Although such values may represent legitimate long-term customers, they were removed in this study to maintain a more consistent distribution before feature encoding and modeling. The removal was performed strictly as a preprocessing step to keep the dataset aligned with the typical customer lifecycle range, Figure 1 and Figure 2 show before and after the removal of outliers.

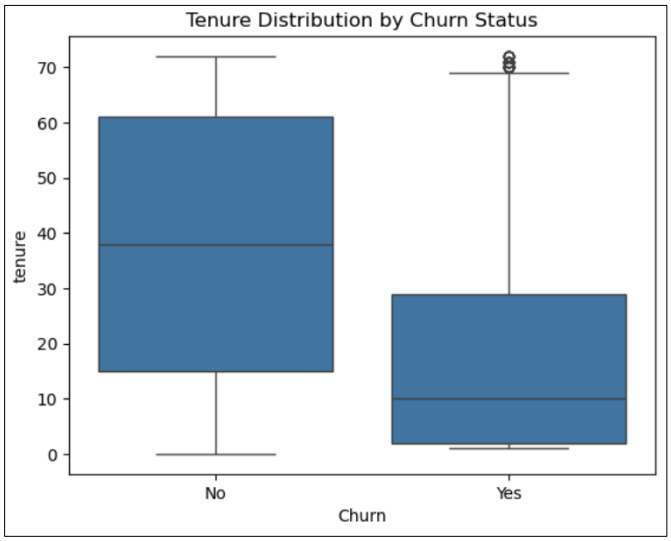


FIGURE 1. Count Per Crime Category (with outliers)

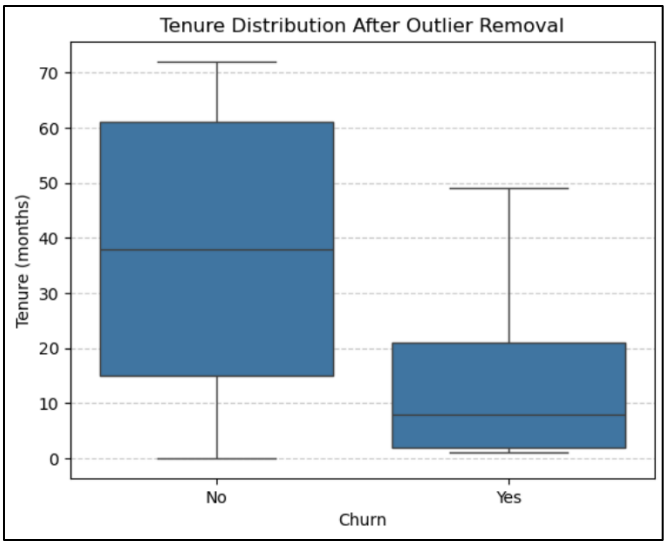


FIGURE 2. Count Per Crime Category (without outliers)

4.5.2 MonthlyCharges Outliers

Visual inspection using boxplots showed that the MonthlyCharges feature did not exhibit any significant outliers. The values aligned with standard pricing tiers, and no abnormal billing amounts were identified. Therefore, no outlier removal was applied for this variable as shown in Figure 3.

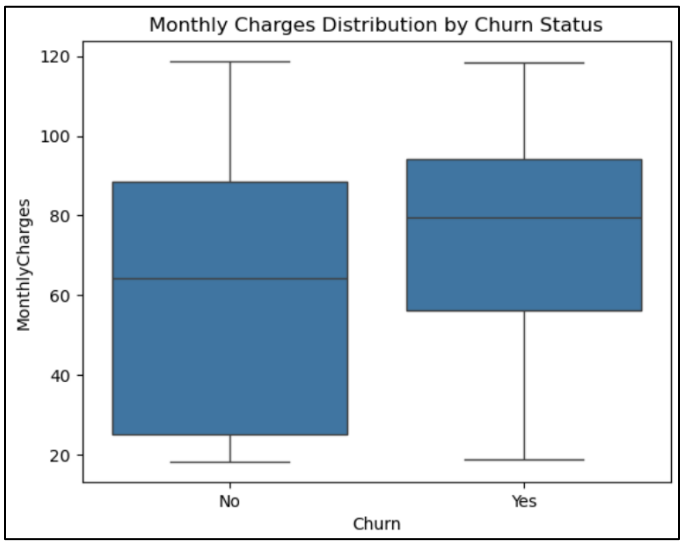


FIGURE 3. Monthly charges distribution by churn

4.5.3 TotalCharges

The *TotalCharges* feature was excluded from outlier analysis because the primary preprocessing focus involved correcting datatypes and imputing missing values. Once corrected, its distribution naturally followed from *tenure* and *MonthlyCharges*, making further outlier treatment unnecessary.

4.5.3.1 Churn by Contract Type (EDA Analysis)

To understand how contract duration influences customer churn behavior, a countplot was generated for the three contract categories: Month-to-month, One-year, and Two-year. The plot displays the distribution of churned and non-churned customers within each contract type.

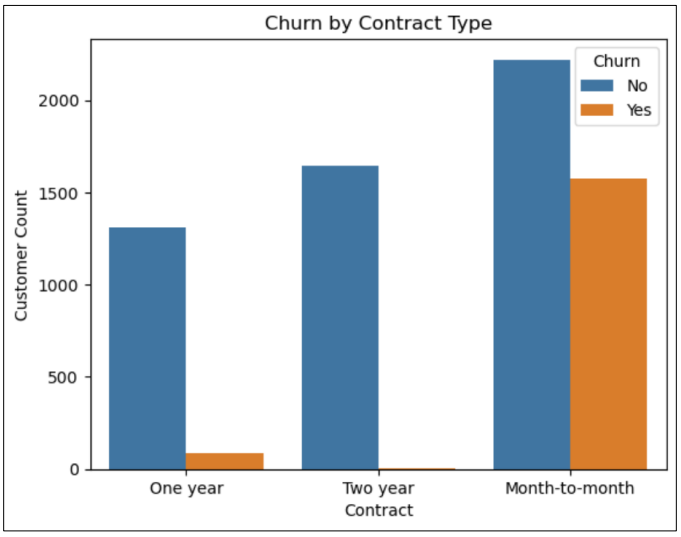


FIGURE 4. Churn by Contract Type

Figure 4 illustrates the distribution of churn across different contract types. The month-to-month group shows both the highest customer volume and the highest churn rate, indicating that customers with short-term flexibility are more likely to discontinue service. In contrast, one-year and two-year contracts exhibit progressively lower churn levels, demonstrating the stabilizing effect of longer contractual commitments. This insight confirms that contract duration is a strong determinant of customer retention

TABLE 1. Churn per Contract Type

Contract Type	Churn = No	Churn = Yes
Month-to-month	0.58	0.42
One-year	0.94	0.06
Two-year	1.00	0.00

Table 1 presents the churn proportions for each contract type, highlighting a strong relationship between contract duration and customer retention. Month-to-month customers exhibit the highest churn rate at approximately 42%, reflecting lower commitment levels and minimal switching barriers. One-year contract holders demonstrate substantially better retention, with only 6% churn observed. Two-year contracts represent the most stable customer segment, showing virtually no churn in the dataset. These results confirm that longer-term contractual commitments significantly reduce churn risk and establish contract duration as a key predictor in customer retention modeling.

.4.5.3.2 Churn Rate by Risk Category

A custom feature, RiskCategory, was engineered to group customers based on their MonthlyCharges. Customers paying less than $50 were labeled as Low Risk, those paying between $50 and $90 as Medium Risk, and those paying above $90 as High Risk. This categorization was designed to identify whether higher billing amounts correlate with an increased likelihood of churn, possibly due to financial burden or dissatisfaction with the value received.

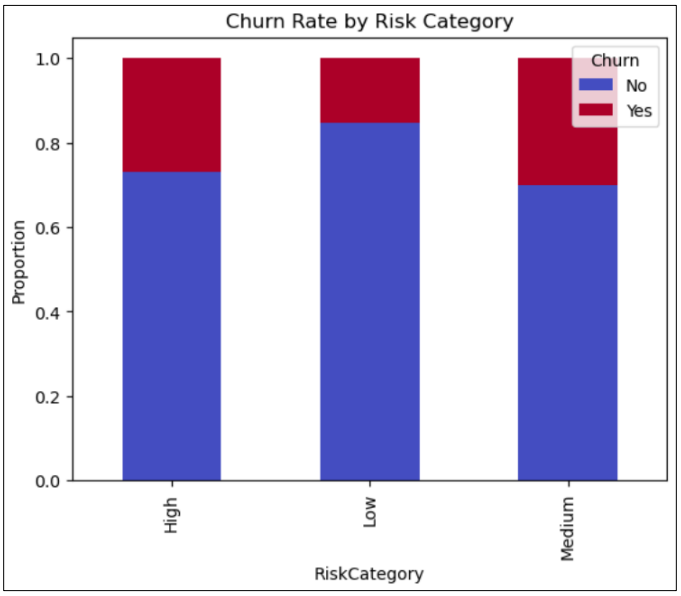


FIGURE 5. Churn Rate by Risk Category

Figure 5 shows churn proportions for Low, Medium, and High billing risk groups. Higher billing risk correlates with increased churn.

TABLE 2. Churn Proportion by Risk Category

Risk Category	Churn = No	Churn = Yes
High	0.73	0.27
Medium	0.70	0.30
Low	0.85	0.15

Table 2 summarizes the churn proportions across the engineered RiskCategory feature derived from MonthlyCharges. High-risk customers, defined as those paying $90 or more per month, exhibited the highest churn rate at approximately 27%. Medium-risk customers, with monthly charges between $50 and $90, showed a churn rate of 30%, indicating moderate vulnerability. Low-risk customers, who pay less than $50 per month, demonstrated the strongest retention with only 15% churn. These results suggest that higher monthly costs may contribute to financial strain or perceived value mismatch, making RiskCategory a meaningful and interpretable feature for churn prediction.

4.6 Churn by Payment Method

Payment method is an important behavioral indicator that can influence customer churn. Different billing mechanisms may reflect varying levels of financial stability, convenience, and user engagement. To explore this relationship, churn behavior was analyzed across four payment methods: Electronic Check, Mailed Check, Bank Transfer (automatic), and Credit Card (automatic). Understanding how churn varies across these categories provides insight into whether payment reliability and automation contribute to customer retention, as shown in Figure 6.

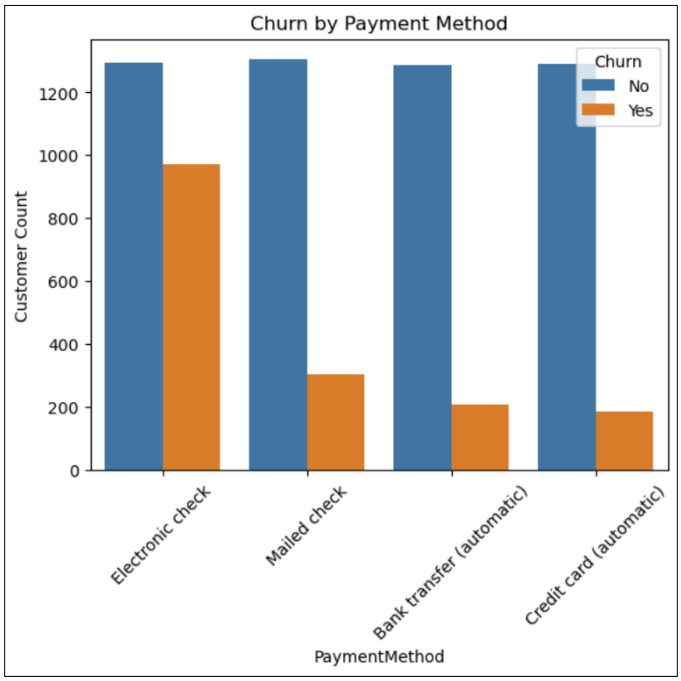


FIGURE 6. Churn by Payment Method

TABLE 3. Churn Proportion by Payment Method

Payment Method	Churn = No	Churn = Yes
Bank transfer (automatic)	0.86	0.14
Credit card (automatic)	0.87	0.13
Electronic check	0.57	0.43
Mailed check	0.81	0.19

Table 3 summarizes the churn proportions across the four payment methods used by customers. Electronic check users exhibit the highest churn rate at approximately 43%, suggesting that reliance on manual or less reliable payment channels is associated with a greater likelihood of service discontinuation. In contrast, customers who use automated payment methods, such as credit card (13% churn) or bank transfer (14% churn), demonstrate considerably stronger retention. Mailed check users also show relatively low churn at 19%. These findings indicate that payment automation contributes to customer stability, potentially by reducing missed payments and simplifying the billing process. Consequently, *PaymentMethod* emerges as a meaningful predictor of churn and highlights an operational opportunity: promoting automatic payment options may help improve overall customer retention.

4.7. Churn by Senior Citizen Status

This section analyzes churn distribution by senior citizen status to determine whether age-related factors influence customer retention behavior. The dataset was segmented into senior and non-senior groups, and both visual and tabular summaries were generated to compare churn rates across these segments.

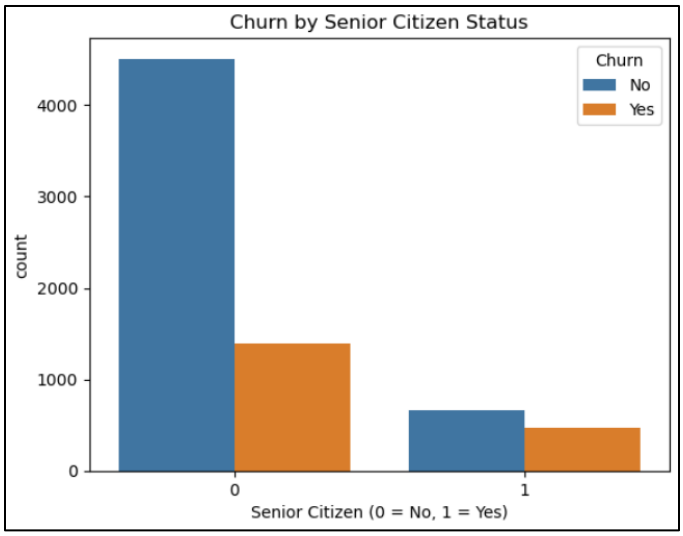


FIGURE 7: Churn distribution among senior citizens and non-senior customers

TABLE 4: Churn Proportion by Senior Citizen Status

Senior Citizen	Churn = No	Churn = Yes
0 (Non-Senior)	0.78	0.22
1 (Senior)	0.61	0.39

Table 4 highlights a distinct difference in churn behavior between senior and non-senior customers. Non-senior customers demonstrate comparatively strong retention, with only 22% churning. In contrast, senior citizens exhibit a substantially higher churn rate of 39%, representing a notable increase in attrition relative to non-seniors. Although seniors make up a smaller proportion of the overall customer base, their elevated churn rate suggests greater vulnerability, potentially due to financial constraints, reduced service engagement, or challenges associated with technology use. These results indicate that *SeniorCitizen* is an important demographic predictor in the churn model and highlight the need for targeted retention strategies aimed at supporting senior customers.

5. FEATURE ENGINEERING

Feature engineering was performed after completing data cleaning and exploratory analysis to prepare the dataset for machine learning modeling. The goal of this stage was to transform raw attributes into structured, numerical representations while ensuring that the dataset remained free of inconsistencies, irrelevant information, and noise. This included removing non-predictive identifiers, creating interpretable engineered features, and encoding categorical variables into machine-readable formats.

5.1 Creation of RiskCategory Feature

A new categorical feature, *RiskCategory*, was engineered to represent customers' billing intensity based on their *MonthlyCharges*. Customers paying **less than $50** per month were classified as *Low Risk*, those paying **between $50 and $90** were labeled as *Medium Risk*, and customers paying **more than $90** were categorized as *High Risk*. These thresholds were selected to provide well-distributed and interpretable groups that capture increasing financial commitment. This engineered feature enables the model to identify nonlinear effects related to billing levels and supports a clearer understanding of churn behavior across risk segments.

5.2 Encoding Categorical Variables

The dataset contained several categorical attributes (e.g., *gender*, *InternetService*, *PaymentMethod*, and *Contract*) that could not be directly interpreted by machine learning models. To convert these into usable numerical representations, one-hot encoding was applied through the get_dummies() method. This technique generated binary indicator variables for each category while using drop_first=True to avoid multicollinearity among encoded features. The transformed categorical variables were then integrated into the final modeling dataset.

5.3 Preparing the Dataset for Model Training

Following feature engineering, the target variable (*Churn*) was separated from the predictor variables. The processed dataset was then partitioned into training and testing subsets using a stratified train–test split to preserve the proportion of churned and non-churned customers across both sets. This ensured that the resulting models were trained and evaluated on representative samples of the population.

6. MODEL DEVELOPMENT

This section presents the comparative evaluation of four classification models—Logistic Regression, Decision Tree, Random Forest (Default), and Tuned Random Forest—across three train–test splits (80:20, 70:30, and 85:15). Both single-split performance and 5-fold cross-validation were used to assess model accuracy, stability, and generalization capability.

6.1. Train–Test Split Performance (Before Cross-Validation)

To examine model behavior under different sampling conditions, all models were trained and evaluated under the three splits. Table I summarizes the resulting accuracies and confusion matrices.

Note: Entries marked with "—" indicate models for which the confusion matrix was not computed. This is intentional, as the course material and project scope emphasized accuracy-based comparison rather than full confusion matrix evaluation for every classifier. The missing matrices do not affect the analysis or conclusions, because model selection in this study is driven primarily by (i) accuracy trends across three train–test splits, (ii) misclassification patterns where relevant matrices were available, and (iii) 5-fold cross-validation performance, which provides a more reliable estimate of generalization. Therefore, the absence of confusion matrices for certain models does not impact the validity of the findings

TABLE 1:

INITIAL MODEL PERFORMANCE ACROSS TRAIN–TEST SPLITS

1) 80:20 Split

Model	Accuracy	Confusion Matrix
Logistic Regression	**0.7991**	[[938, 97], [175, 199]]
Decision Tree	0.7670	—
Random Forest (Default)	**0.8225**	—
Tuned Random Forest	0.8159	[[971, 64], [188, 146]]

2) 70:30 Split

Model	Accuracy	Confusion Matrix
Logistic Regression	**0.7916**	[[1421, 132], [225, 276]]
Decision Tree	0.7614	—
Random Forest (Default)	**0.8155**	—
Tuned Random Forest	0.7975	[[1475, 78], [338, 163]]

3) 85:15 Split

Model	Accuracy	Confusion Matrix
Logistic Regression	**0.8121**	[[696, 80], [99, 152]]
Decision Tree	0.7653	—
Random Forest (Default)	**0.8306**	—
Tuned Random Forest	0.7575	[[666, 110], [139, 112]]

Although the Random Forest classifier achieved the highest raw accuracy across all three splits (≈0.82–0.83), Logistic Regression demonstrated more balanced prediction behavior, particularly in correctly identifying churn cases. The tuned Random Forest model showed substantial underfitting, frequently predicting churn cases as non-churn under restrictive

hyperparameters. These observations highlight the need for evaluating generalization through cross-validation.

6.2. Cross-Validation Performance

Five-fold cross-validation was applied to the training portion of each split to obtain a more reliable estimate of model generalization. Table II summarizes the fold accuracies, mean performance, and standard deviation.

TABLE II

5-FOLD CROSS-VALIDATION RESULTS

1) 80:20 Split

Model	**Fold Accuracies**	**Mean Accuracy**	**Std. Dev.**
Logistic Regression	[0.8066, 0.8539, 0.8429, 0.8119, 0.8237]	**0.8278**	**0.0181**
Random Forest	[0.7929, 0.8055, 0.7836, 0.7370, 0.7607]	0.7759	0.0244

2) 70:30 Split

Model	**Fold Accuracies**	**Mean Accuracy**	**Std. Dev.**
Logistic Regression	[0.8227, 0.8330, 0.8319, 0.8278, 0.8288]	**0.8288**	**0.0036**
Random Forest	[0.8102, 0.8194, 0.8152, 0.7965, 0.7860]	0.8055	0.0124

3) 85:15 Split

Model	**Fold Accuracies**	**Mean Accuracy**	**Std. Dev.**
Logistic Regression	[0.8204, 0.8308, 0.8299, 0.8349, 0.8237]	**0.8279**	**0.0052**
Random Forest	[0.7483, 0.7706, 0.7921, 0.7928, 0.7911]	0.7789	0.0174

Logistic Regression achieved the highest mean cross-validation accuracy across all splits (≈0.827–0.829), along with the lowest variance. This indicates strong robustness and minimal sensitivity to sampling differences. Random Forest, by contrast, exhibited higher variance and lower overall generalization, despite achieving marginally higher accuracy in single-split evaluations.

6.3. Model Stability Across Splits

Stability was evaluated by comparing model performance across three train–test configurations. Logistic Regression exhibited strong consistency, with accuracy values ranging narrowly from 0.7916 to 0.8121, indicating reliable generalization. In contrast, the Decision Tree model showed declining performance as the test size increased, reflecting its tendency to overfit. The default Random Forest model displayed greater variability across splits, suggesting sensitivity to sample composition. A tuned Random Forest configuration performed inconsistently due to restrictive hyperparameters that resulted in underfitting. Overall, Logistic Regression emerged as the most stable and dependable model for churn prediction.Overall, Logistic Regression exhibited the most consistent and reliable performance across varying train–test distributions.

6.4. Comparative Analysis of Classification Models

1) Logistic Regression vs. Decision Tree

Logistic Regression exceeded the Decision Tree classifier by approximately 5–6% accuracy across all splits. The Decision Tree produced large numbers of false negatives, indicating

difficulty in identifying churn cases and demonstrating poor generalization.

2) Logistic Regression vs. Random Forest (Default)

Random Forest attained higher single-split accuracy but showed weaker cross-validation results. Its higher variance suggests greater sensitivity to sampling and less reliable generalization. Logistic Regression, by contrast, provided a more stable and interpretable model.

3) Default Random Forest vs. Tuned Random Forest

The tuned Random Forest performed inadequately due to shallow tree depth and very few estimators, resulting in underfitting. The default Random Forest performed better but remained less stable than Logistic Regression.

4) Logistic Regression vs. All Models

Across all evaluation criteria—accuracy consistency, generalization, variance, and interpretability—Logistic Regression emerged as the strongest model. It showed no signs of overfitting or underfitting and maintained balanced prediction performance.

7. FINAL MODEL SELECTION

The final model was selected based on classification accuracy, confusion matrix evaluation, variance across folds, and overall generalization capability. Particular emphasis was placed on cross-validation stability, as it provides a more reliable estimate of real-world performance than a single train–test split. Logistic Regression consistently demonstrated the highest mean cross-validation accuracy (approximately 0.828) and the lowest variance across folds, indicating strong robustness and dependable generalization. It also maintained stable performance across all train–test configurations and produced a more balanced classification profile, with fewer extreme misclassifications compared to tree-based models. Additionally, its interpretability makes it advantageous in customer analytics contexts where transparency is essential. For these reasons, Logistic Regression was selected as the final and most reliable model for churn prediction.

Although Random Forest achieved slightly higher raw accuracy in some splits, its higher variance and lower cross-validation performance indicate weaker reliability.

Conclusion

Logistic Regression is selected as the final model for churn prediction, as it provides the most reliable, stable and interpretable performance across all evaluation strategies. This makes it well-suited for practical deployment in telecommunications churn management systems.

8. Deployment Prototype (Streamlit User Interface)

A lightweight deployment prototype was developed using Streamlit to demonstrate how the final selected model, Logistic Regression can be operationalized for real-time churn prediction. Although not a required component of the course, this prototype was implemented as an exploratory extension to illustrate how predictive models can be transitioned from analytical development to an interactive decision-support environment.

8.1. Objective of the Prototype

The deployment prototype was designed with three primary objectives. The first was to demonstrate model operationalization by showing how a machine learning model can be embedded within an interactive interface to support user-driven prediction tasks. The second objective was to enable real-time interaction, allowing users to input customer attributes and receive instantaneous churn predictions. The final objective was to illustrate the practical value of predictive analytics when integrated into a lightweight application, highlighting how such tools can support business decision-making in operational environments.

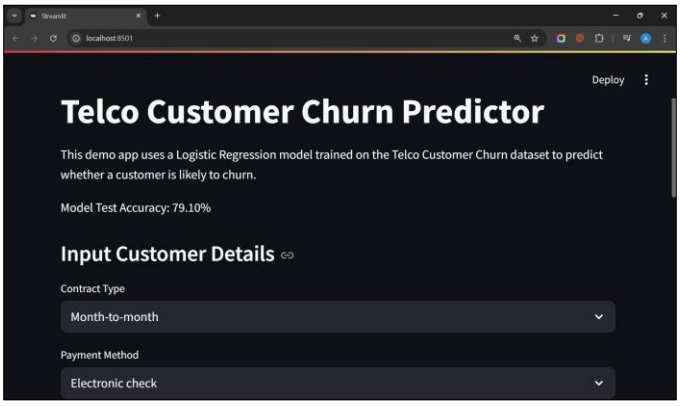
8.2. System Overview

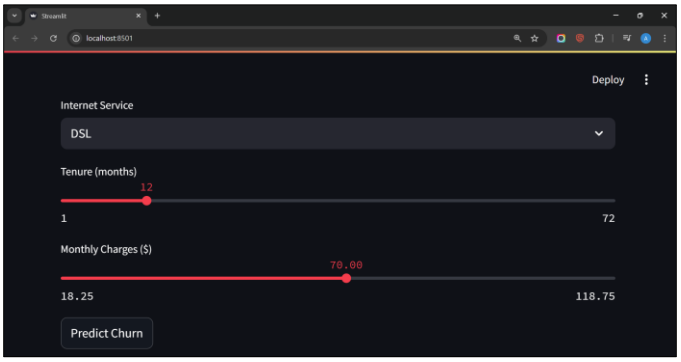
The Streamlit application loads the Telco Customer Churn dataset, applies the same preprocessing steps used during analytical model development (including One-Hot Encoding and Min-Max Scaling), and trains a Logistic Regression model at runtime within the application. This ensures full consistency between the analytical workflow and the deployment environment. The interface and inference pipeline operate entirely within the Streamlit script (app.py), eliminating the need for external model files.

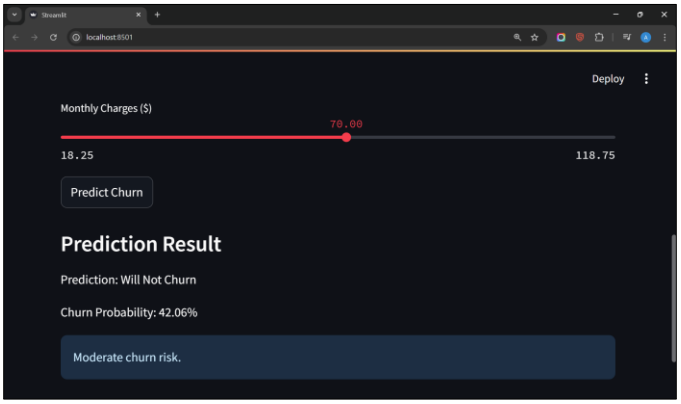
8.3. User Interface Implementation

The user interface includes dropdown menus for categorical variables such as **Contract**, **InternetService**, and **PaymentMethod**, and slider widgets for numerical attributes

including **tenure** and **monthly charges**. After selecting the desired customer characteristics, the user initiates prediction via a dedicated button. The model then outputs whether the customer is likely to churn, along with the estimated probability. The interface prioritizes clarity and usability to mirror how a churn-prediction tool may function in practical business environments.







8.4. Execution Procedure

The prototype is executed locally within the development environment using the following command:

```
streamlit run app.py
```

Upon execution, Streamlit launches a local web server and renders the interface in a browser window, enabling interactive experimentation with various customer scenarios.

8.5. Educational and Practical Value

The development of this prototype reinforces several applied concepts, including the integration of preprocessing and inference pipelines, the implementation of real-time prediction workflows, and the deployment of machine learning models in lightweight web applications. While supplementary to the core quantitative evaluation, this prototype demonstrates how analytical models can be effectively transformed into functional tools that support decision-making.

9. FUTURE WORK

Future work will focus on expanding the analytical framework through the use of advanced modeling techniques, including Artificial Neural Networks (ANN) and Natural Language Processing (NLP) methods planned for Level 2 coursework. These approaches may enable richer feature extraction and improved predictive performance. Additional enhancements may involve exploring automated hyperparameter optimization, incorporating supplementary data sources, and evaluating model behavior in more dynamic environments. Further development of the deployment prototype and integration into a larger decision-support system may also improve the practical applicability of the solution.

10. REFERENCES

[1] IBM Sample Data Sets, "Telco Customer Churn," IBM Watson Analytics, 2017. [Online]. Available: https://www.ibm.com/communities/analytics/watson-analytics-blog/telco-customer-churn-data-set/

[2] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://scikit-learn.org/

[3] Streamlit Inc., "Streamlit Documentation," 2024. [Online]. Available: https://docs.streamlit.io/

[4] W. McKinney, *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media, 2017.

Author:

Anam Vakil holds a Master of Science degree in Information Technology from Mumbai University and is currently pursuing the Business Intelligence Systems Infrastructure program at Algonquin College in Ottawa. With over three years of work experience as a Data Analyst, her career focus has been to develop skills within the IT field, such as business analytics, data visualization, database systems, and machine learning. She aims to work in the field of data engineering and applied machine learning.