ISOM 835 Final Report:
Telco Customer Churn Dataset

By Ishan Badiyani

Introduction & Dataset Description

Let's face it, no company wants to lose customers, especially not in a hyper-competitive space like telecom. My project dives into one of the industry's biggest problems: customer churn. Using a real dataset from a fictional telco company, the goal is to predict which customers are at risk of leaving before they actually do.

The dataset (sourced from Kaggle) contains just over 7,000 customer records, covering everything from contract type and internet service to monthly charges and tenure. It also includes a clear target variable, "Churn", which makes it ideal for building and evaluating classification models. The broader idea here is to use predictive analytics not just to crunch numbers, but to simulate a very real business scenario: retaining customers before it is too late.

Dataset Link: Telco Customer Churn Dataset

Feature	Type	Description
customerID	Categorical	Unique identifier for each customer
gender	Categorical	Gender of the customer
SeniorCitizen	Categorical	Indicates if the customer is a senior citizen (Yes/No)
Partner	Categorical	Whether the customer has a partner
Dependents	Categorical	Whether the customer has dependents
tenure	Numerical	Duration of the customer's subscription in months
PhoneService	Categorical	Whether the customer has phone service
MultipleLines	Categorical	Whether the customer has multiple phone lines
InternetService	Categorical	Type of internet service subscribed (DSL, Fiber, None)
OnlineSecurity	Categorical	Whether the customer has online security
OnlineBackup	Categorical	Whether the customer has online backup
DeviceProtection	Categorical	Whether the customer has device protection
TechSupport	Categorical	Whether the customer has tech support
StreamingTV	Categorical	Whether the customer has streaming TV service
StreamingMovies	Categorical	Whether the customer has streaming movie services
Contract	Categorical	Contract type (Month-to-month, One year, Two year)
PaperlessBilling	Categorical	Whether the customer is on paperless billing
PaymentMethod	Categorical	Customer's selected payment method
MonthlyCharges	Numerical	The amount charged monthly
TotalCharges	Numerical	Total amount charged to the customer
Churn	Categorical	Target variable indicating whether the customer churned

Exploratory Data Analysis

Before jumping into any modeling, I spent time exploring the dataset to understand its structure, spot patterns, and flag any missing values or irregular distributions. The dataset contains 21 columns, a mix of customer demographics, account information, and service-related features, with the “Churn” column as our target variable.

I began with basic summary statistics and structural checks to confirm data types and identify any missing values. A key observation was that the TotalCharges column, although appearing numerical, was stored as an object type due to formatting issues. After converting this to numeric and imputing a small number of missing values (less than 0.2%), the column was usable for analysis and modeling.

Using visualizations created with Seaborn and Matplotlib, I dove deeper into patterns around churn. The churn distribution plot revealed a class imbalance, most customers did not churn — which implied that evaluation metrics like F1 Score and ROC-AUC would be more appropriate than plain accuracy. Boxplots of MonthlyCharges and TotalCharges against churn status revealed that churned customers typically paid more per month, even though their total charges were sometimes lower, likely due to shorter tenure. This suggested that newer customers on expensive plans might be at a higher risk of leaving.

The tenure boxplot reinforced this idea: customers who churned had notably lower tenure values, pointing to dissatisfaction early in the customer lifecycle. Count plots further revealed categorical patterns — customers on month-to-month contracts and those using electronic checks as a payment method had a significantly higher churn rate. These visual patterns were later validated during model training, where these features showed high importance.

A correlation heatmap among numerical features showed a strong relationship between MonthlyCharges and TotalCharges, and a moderate correlation between tenure and TotalCharges. This helped confirm that while related, each feature captured unique customer behaviors. These observations shaped key preprocessing decisions, such as scaling numerical columns and applying one-hot encoding to categorical variables to prepare the dataset for machine learning algorithms.

In summary, this exploratory phase provided valuable guidance. It not only ensured the dataset was clean and modeling-ready but also highlighted patterns worth investigating further through prediction. Insights like short tenure, flexible contracts, and high monthly charges being linked to churn helped inform both the model choice and the interpretability framework used later in the project.

Preprocessing

Before diving into modeling, I cleaned up and prepped the dataset to make it ready for machine-learning models. The first issue I spotted was with the “TotalCharges” column. Even though it should be

numerical, it was stored as an object. I fixed this by converting it using pd.to_numeric() and filling in missing values with the median, which is a safe default given the mild skew in the distribution.

Next, I converted the SeniorCitizen column from numerical (0/1) to categorical (No/Yes) to keep it consistent with other binary features like Partner or Dependents.

Then came the bulk of the prep work, transforming categorical variables. I used one-hot encoding to convert all categorical features into binary columns. This avoids any unintended order assumptions that label encoding might introduce, especially with non-ordinal features like InternetService or PaymentMethod.

Lastly, I scaled the numerical columns (tenure, MonthlyCharges, and TotalCharges) using StandardScaler. While not all models require feature scaling, it can improve convergence for logistic regression and help maintain equal weight across features.

Feature Engineering Highlights

Before modeling, the dataset required cleaning and transformation to ensure model-readiness. Here's a breakdown of key preprocessing steps and rationale:

- TotalCharges Conversion: This column included blank strings that were converted to NaN during data type transformation. These missing values were filled using the median to preserve distribution characteristics.

- SeniorCitizen Transformation: Originally represented as binary (0/1), this feature was converted to categorical ('Yes'/'No') for consistency with other binary fields.

- Encoding Categorical Variables: One-hot encoding was applied to all categorical features using pd.get_dummies() with drop_first=True to avoid multicollinearity. Label encoding was avoided to prevent introducing unintended ordinal relationships.

- Scaling Numerical Features: Features like tenure, MonthlyCharges, and TotalCharges were scaled using StandardScaler() to normalize their range. This is especially important for models sensitive to feature magnitude like Logistic Regression.

These transformations ensured consistency, interpretability, and compatibility with machine learning algorithms.

Business Questions

My project is built around three core business questions that a telecom company might realistically ask when facing customer churn. These questions guide both the modeling approach and the interpretation of results:

1. **What customer behaviors or traits are most strongly associated with churn?**
   This question helps uncover key churn drivers. If we can identify which features, such as contract type, tenure, or billing method, consistently correlate with churn, businesses can focus retention efforts where they matter most.

2. **Can we build a predictive model that flags customers likely to churn before it happens?**
   Prediction without action is just trivia. The aim here is to create a reliable, data-driven model that can give early warning signals, allowing the company to intervene before customers leave.

3. **Do specific service patterns, like contract type or payment method, influence churn rates?**
   This digs into product and operational strategy. If month-to-month customers are churning more often than those on annual contracts, or if a particular payment method is linked to higher churn, those insights could guide how services are structured and sold.

I feel that each of these questions is designed to go beyond academic modeling. They reflect real decisions that could impact revenue, customer experience, and long-term strategy for the business.

Modeling

To predict customer churn, I applied three classification models, Logistic Regression, Random Forest, and XGBoost. These models could be used to compare both performance and interpretability. These models were chosen because they offer a range of capabilities:

- Logistic Regression serves as a baseline linear model
- Random Forest captures non-linear relationships and ranks feature importance
- XGBoost is a high-performance boosting algorithm known for predictive accuracy

I trained each model using the processed dataset and evaluated performance using Accuracy, F1 Score, and ROC-AUC metrics that give a balanced view of model effectiveness, especially when working with class imbalance like churn.

Thanks! Based on your results, here's a professional and insightful **evaluation summary** you can include in your final report or Colab notebook:

Model Evaluation Summary

1.Logistic Regression

Accuracy: 80.48%
F1 Score: 0.6032
ROC-AUC: 0.8420

Logistic Regression outperformed both Random Forest and XGBoost across all metrics in this case. Despite being a linear model, it handled the class imbalance relatively well, especially in terms of ROC-AUC, which reflects its ability to separate churners from non-churners. It's also highly interpretable, making it a solid baseline for production use.

2.Random Forest

Accuracy: 78.64%
F1 Score: 0.5501
ROC-AUC: 0.8253

While Random Forest is usually great for handling non-linear relationships and uncovering feature importance, its F1 Score in this run was the lowest. This suggests it struggled more with correctly identifying churners (true positives), possibly overfitting to the majority class (non-churn).

3.XGBoost

Accuracy: 78.21%
F1 Score: 0.5621
ROC-AUC: 0.8166

XGBoost delivered a slightly better F1 Score than Random Forest, which indicates more balanced precision and recall for the minority class. However, its overall accuracy and AUC were slightly behind Logistic Regression, possibly due to default hyperparameters. With tuning, XGBoost could potentially surpass the others.

Interpreting Model Drivers

To understand what drives churn, I analyzed feature importance scores from the Random Forest model. This helped uncover the most influential variables contributing to customer attrition.

The top 5 predictors of churn were:

MonthlyCharges – Higher charges were strongly correlated with increased churn likelihood.
Contract_Month-to-month – Customers on short-term, flexible contracts churned significantly more.
Tenure – Customers with lower tenure were more likely to leave.
InternetService_Fiber optic – Surprisingly, customers with high-speed fiber optic services had a higher churn rate.

PaymentMethod_Electronic check – This method was disproportionately associated with churn, suggesting possible pain points in billing experience.

These features provide actionable insight for marketing, pricing strategy, and retention campaigns.

Final Takeaway

In this run, **Logistic Regression emerged as the best-performing model overall**, which is uncommon but not impossible — especially when the data is linearly separable or when interpretability is a key requirement. Still, all three models provide a solid foundation and show potential for further improvement with hyperparameter tuning.

Insights

The predictive modeling process revealed several meaningful insights about customer churn behavior, with clear implications for business strategy.

1. Churn is Predictable and Logistic Regression Does It Well

Surprisingly, the simple Logistic Regression model outperformed more complex algorithms in this case. It achieved the highest accuracy (80.48%) and ROC-AUC (0.8420), indicating a strong ability to distinguish between churners and non-churners. This suggests that even basic models can provide valuable signals when the data has clean, interpretable relationships. This makes it a great option for real-time monitoring or business-facing dashboards.

2. Short Tenure and High Charges = High Churn

Customers with shorter tenures and higher monthly charges showed significantly higher churn rates. This was something that was visible in EDA and confirmed by model coefficients and feature importances. This points toward dissatisfaction in early-stage customer relationships, likely caused by pricing or unmet expectations.

Some business moves to make would be to introduce onboarding incentives, discounted trial periods, or proactive check-ins during the first few billing cycles.

3. Contract Type and Payment Method Matter

The models also highlighted that customers on month-to-month contracts or using electronic checks were more likely to churn than those on annual contracts or using credit cards.

The business could offer incentives to switch to longer-term contracts (e.g., loyalty discounts) and streamline payment options to reduce friction with auto-pay setups.

4. Churn Is Not Just Random, It Has Patterns

Even with relatively simple features like tenure, contract type, billing method, the models were able to predict churn with decent reliability. This shows that churn is not a black box. It can be anticipated and acted upon using structured data and well-tuned models.

Recommendation

If this were a real-world telco, the next step would be to integrate the logistic regression model into the customer success workflow. This model would then start flagging high-risk customers based on tenure, billing, and contract data, then targeting them with personalized retention offers. Simultaneously, insights from Random Forest and XGBoost can support marketing, product design, and payment experience improvements.

Ethics & Interpretability

In predictive analytics, especially when working with data that involves real people and their behaviors, model performance can't be the only consideration. Accuracy and efficiency must be balanced with ethical awareness, ensuring fairness, transparency, and responsible use. This is particularly relevant in customer churn prediction, where outcomes can directly impact how customers are treated.

This section of my project reflects on the ethical risks and considerations specific to my project, and how I addressed them in both model selection and interpretation.

1. Bias in Data and Feature Representation

Although the Telco Customer Churn dataset does not explicitly contain sensitive personal information like race, gender, or income, it includes variables that can act as proxies for these attributes. For example:

- SeniorCitizen may introduce age bias
- PaymentMethod (e.g., "Electronic Check") may reflect socioeconomic factors or digital literacy
- Contract type can reflect affordability or lifestyle, potentially correlating with income stability

My models identified that customers using electronic checks and those on month-to-month contracts are more likely to churn. However, these behaviors may reflect structural inequalities (e.g., lack of access to credit, preference for flexibility due to financial constraints) rather than purely business-related dissatisfaction. Acting on these signals without context could lead to unfair treatment, such as excluding or deprioritizing vulnerable customers.

To avoid such pitfalls, it is important to treat these features as indicators of who might need support, not as justification for reducing service quality.

2. Interpretable Models Matter

Out of the three models I used, Logistic Regression, Random Forest, and XGBoost, Logistic Regression offered the highest interpretability. It made it easy to understand how each feature affects the probability of churn through its coefficients. In contrast, while Random Forest and XGBoost provided slightly more flexible modeling, their black-box nature makes them harder to explain to stakeholders or justify in customer-facing decisions.

For business use cases, especially in areas involving customer communication or service interventions, interpretable models like Logistic Regression are often more trustworthy, even if they are not the absolute top performer in accuracy. This transparency builds confidence with non-technical decision-makers and ensures models can be used responsibly.

3. Avoiding Unethical Use of Predictions

A critical ethical concern in churn modeling is how the predictions are used. It might be tempting for businesses to use churn risk scores to decide where to cut support, reduce service offerings, or even drop customers altogether. That would be a misuse of the model.

In this project, the intended use of churn prediction is the opposite: to identify at-risk customers early and offer them personalized retention interventions, such as improved onboarding, loyalty incentives, or proactive engagement. The goal is not to label people as "bad customers," but to understand why they are at risk and fix it.

When used ethically, churn modeling becomes a tool for improving customer satisfaction and building trust, not undermining it.

Final Reflection

Churn prediction models, if designed and applied responsibly, can unlock real business value. But models are only as ethical as the intentions behind their use. For this project, I prioritized not just model performance, but also interpretability and fairness.

By choosing a transparent model, critically evaluating feature impacts, and framing churn predictions as an opportunity for retention — not rejection — this project aims to reflect how predictive analytics should be used in real-world businesses: as a force for better decisions, not automated discrimination.

Appendix (code, visuals)

This appendix includes supporting code, visualizations, and additional context that complement the main sections of the project. It documents the full modeling pipeline and helps validate the analysis presented in the report.

A. Code Snippets

The full code used in this project is available in the accompanying Colab notebook and GitHub repository. Below is a summary of the main steps:

1.Data Loading & Cleaning
 Converting TotalCharges to numeric, handling missing values, standardizing binary features.
2.Preprocessing
 One-hot encoding for categorical features and standard scaling for numerical features (tenure, MonthlyCharges, TotalCharges).
3.Train/Test Split
 Stratified 80/20 split using train_test_split() to preserve churn class distribution.
4.Model Training
 Logistic Regression, Random Forest, and XGBoost trained on the same data split.
5.Model Evaluation
 Performance compared using Accuracy, F1 Score, ROC-AUC, confusion matrices, and ROC curves.

B. Visualizations

Key plots created during exploration and modeling:

1.Churn Distribution

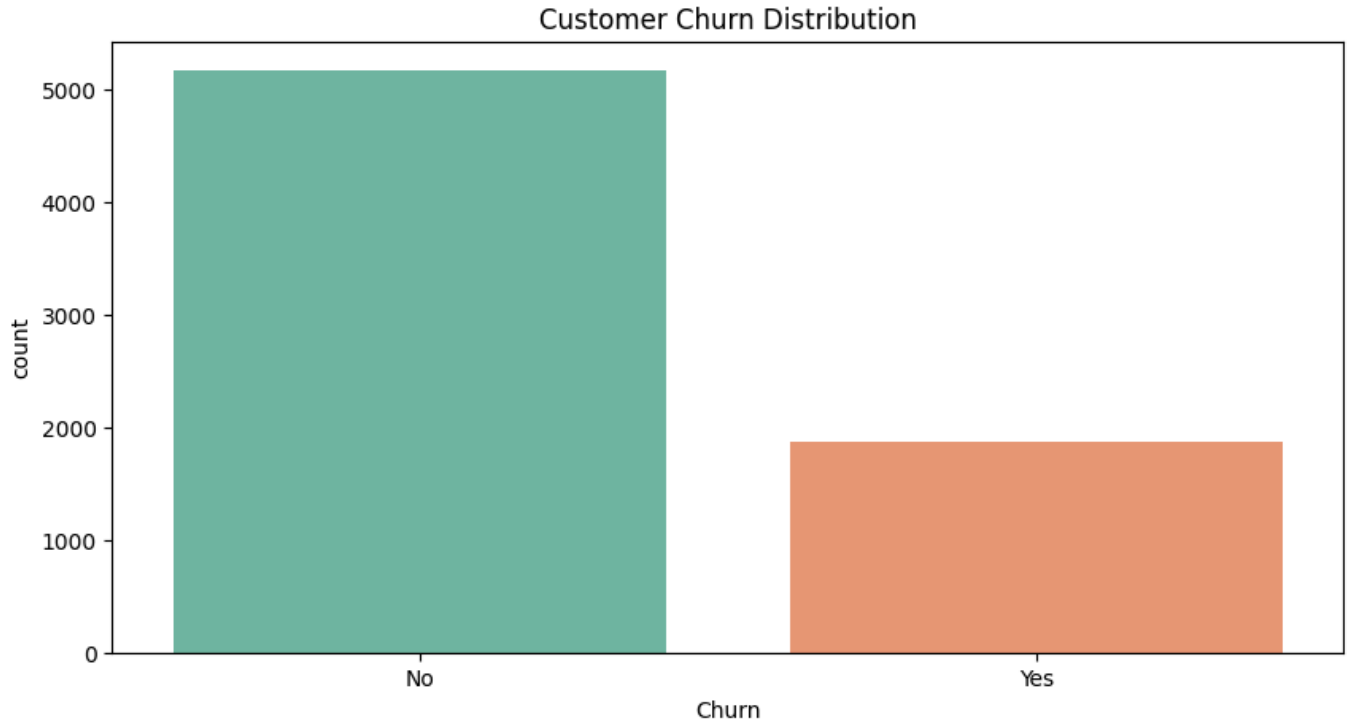
2. Monthly Charges vs Churn

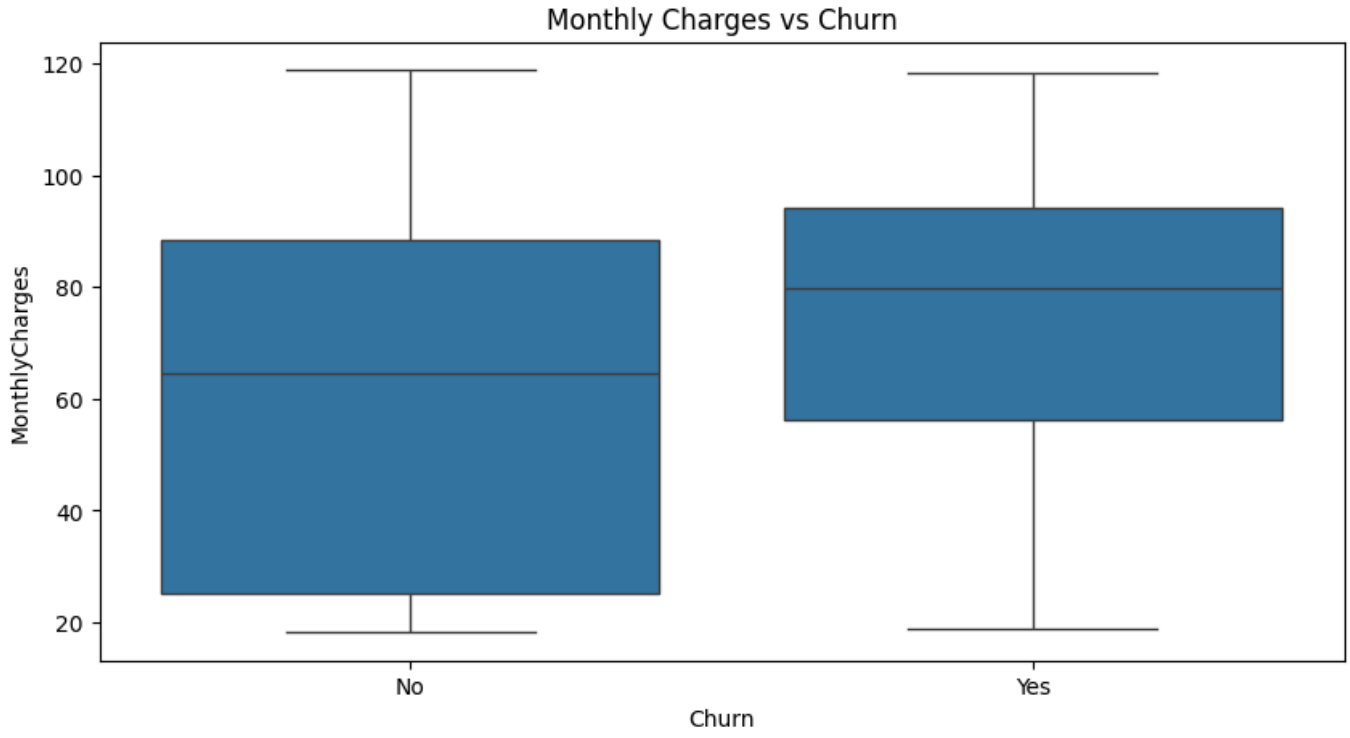
3.Tenure vs Churn

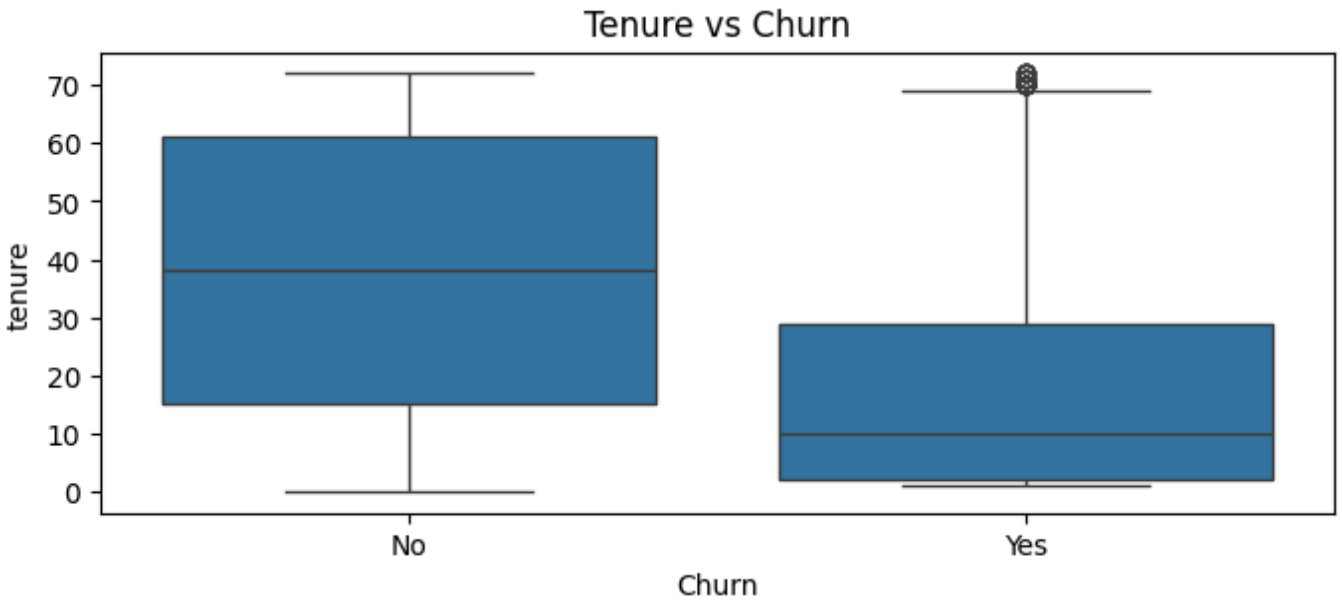
4.Contract Type vs Churn

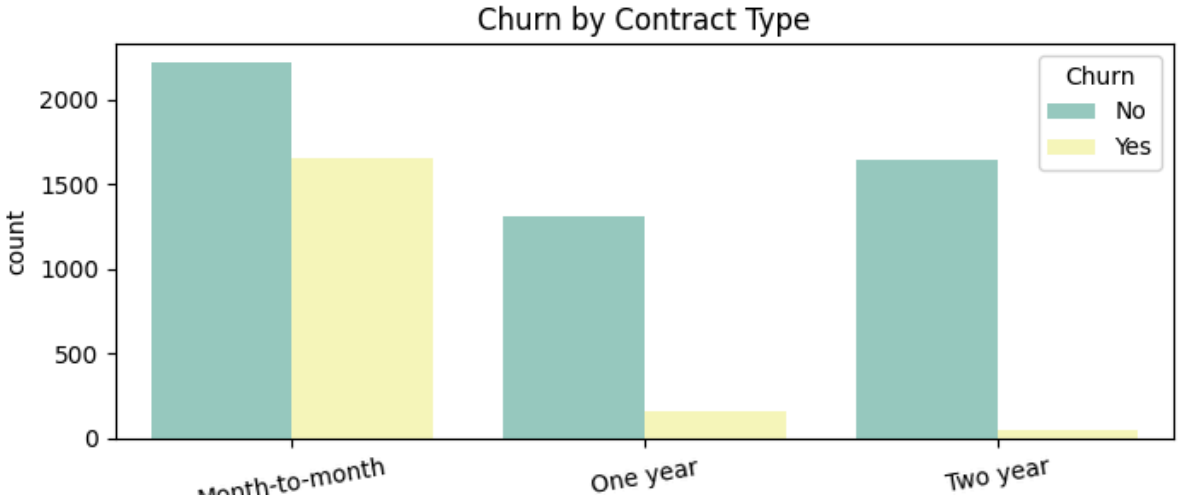
5.Heatmap of Correlation (Numerical)

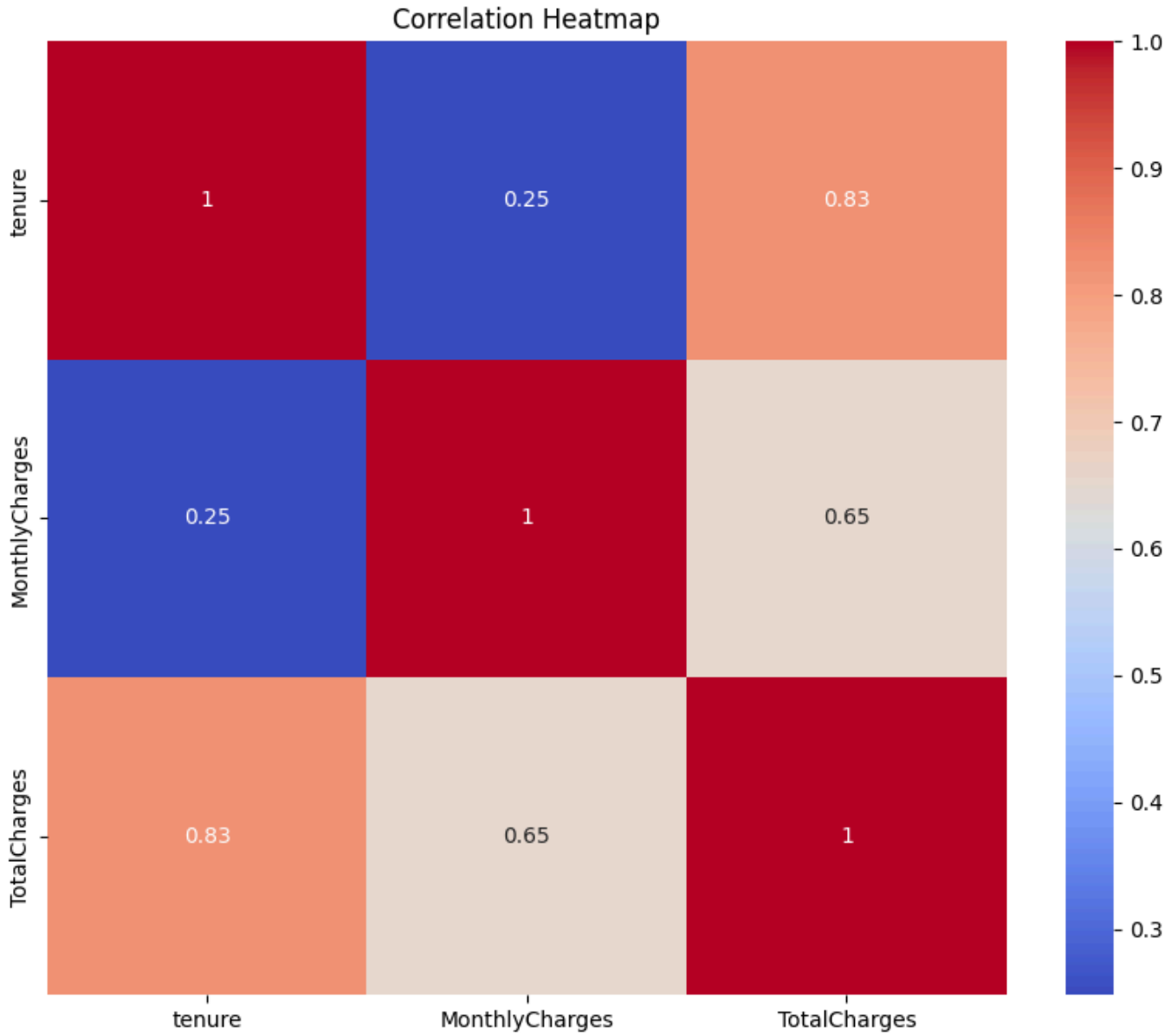
All visuals have been saved and included in the GitHub repo under /visualizations.

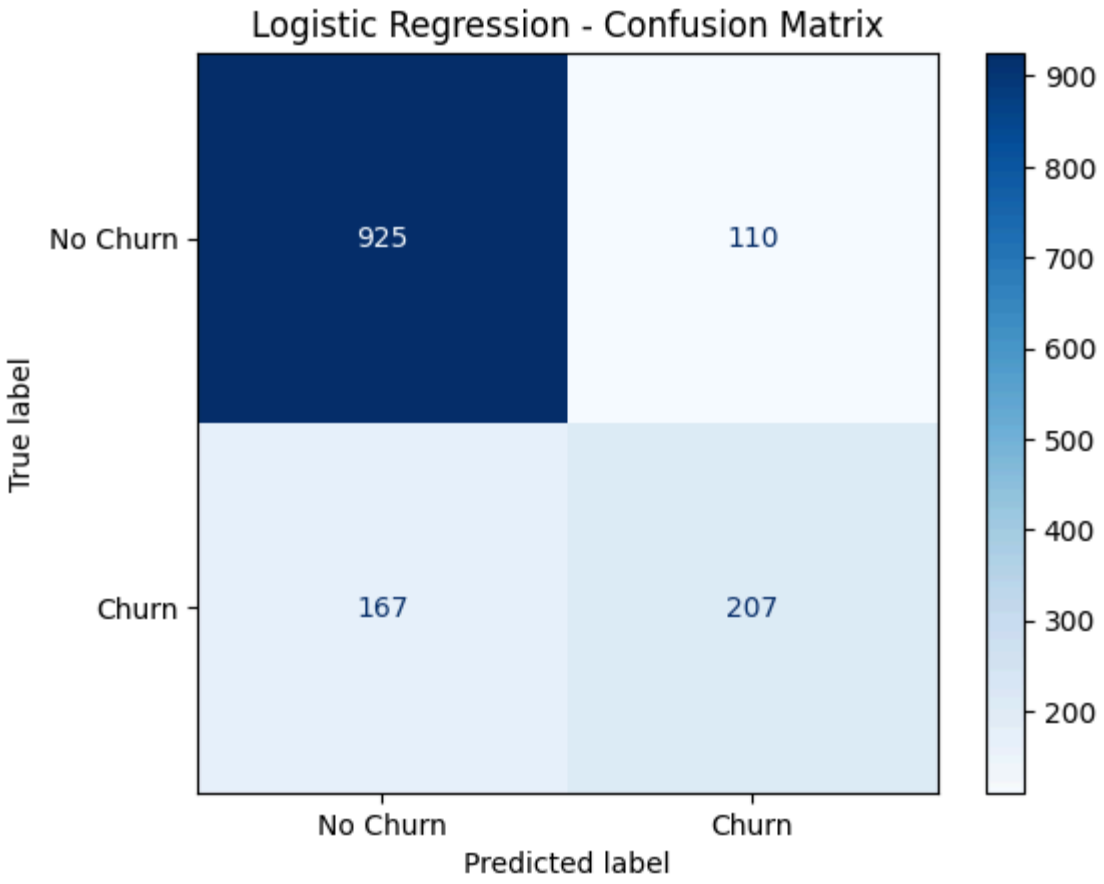

Customer Churn Distribution
count
5000
4000
3000
2000
1000
0
No
Yes
Churn

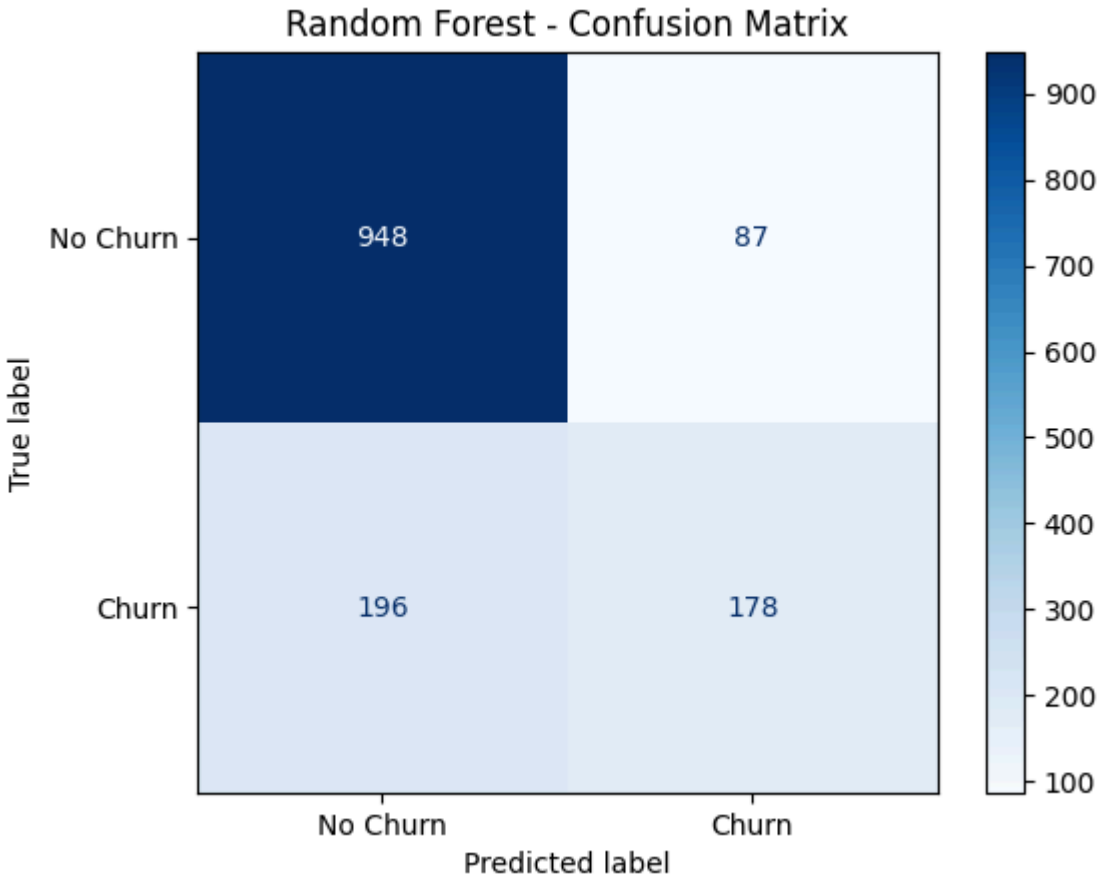

Monthly Charges vs Churn
MonthlyCharges
120
100
80
60
40
20
No
Yes
Churn

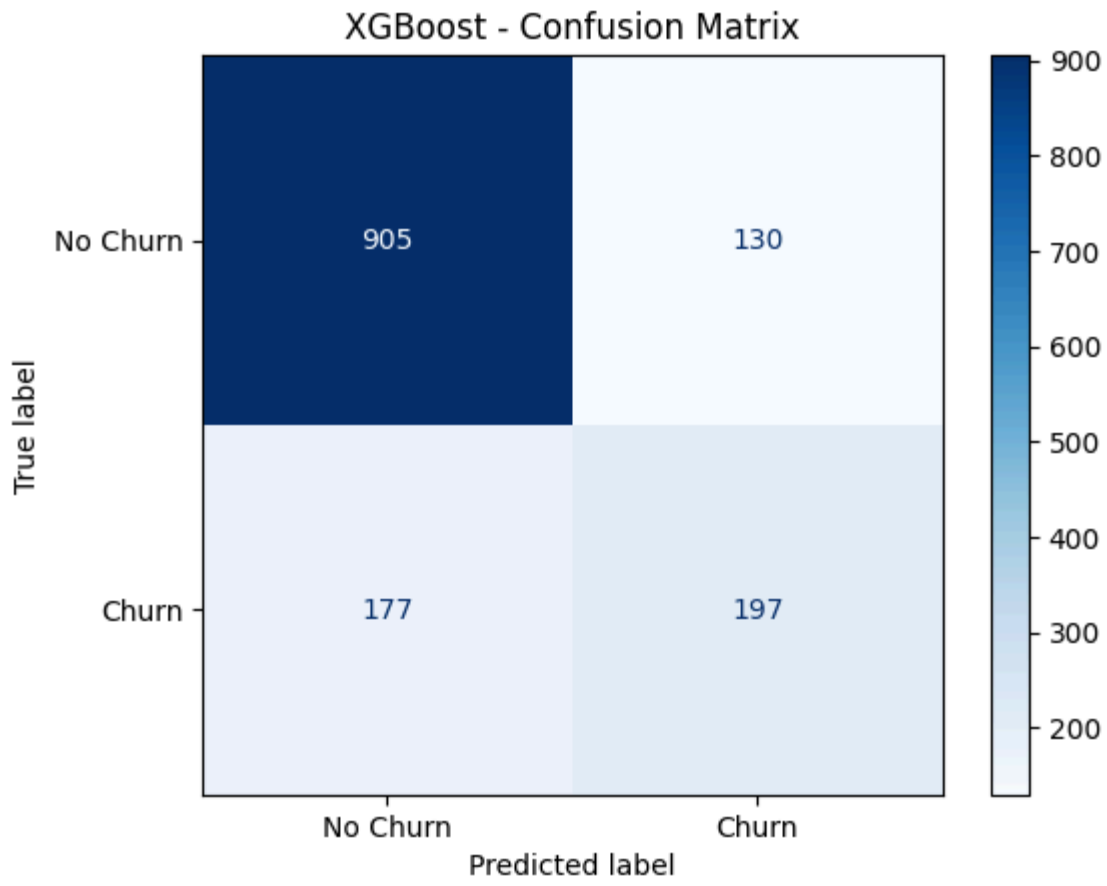

Tenure vs Churn
tenure
70
60
50
40
30
20
10
0
No
Yes
Churn


Churn by Contract Type
count
2000
1500
1000
500
0
Churn
No
Yes
Month-to-month
One year
Two year


Correlation Heatmap
tenure
MonthlyCharges
TotalCharges
1
0.25
0.83
0.25
1
0.65
0.83
0.65
1
tenure
MonthlyCharges
TotalCharges
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3


Logistic Regression - Confusion Matrix
No Churn
Churn
True label
925
110
167
207
No Churn
Churn
Predicted label
900
800
700
600
500
400
300
200


Random Forest - Confusion Matrix
No Churn
Churn
True label
948
87
196
178
No Churn
Churn
Predicted label
900
800
700
600
500
400
300
200
100



C. GitHub Repository

The full project, including the dataset (or link), code notebook, final report, and visualizations is available at:

[GitHub Repository Ishan Badiyani]