Customer Churn Prediction Using Machine Learning

CSCE 5215 Project
Dr. Tymoshchuk

Group 2

Amrit Adhikari amritadhikari@my.unt.edu
Son Dang sondang@my.unt.edu
Bob Jack robertjack@my.unt.edu
Crista Kingsley cristakingsley@my.unt.edu

July 12, 2025

Project Code Link:
https://colab.research.google.com/drive/1kaDfLKQxFsSzpQJh3utq_G2WZgO6J1aF

Section 1: Project Proposal

Chosen Topic

Customer Churn Prediction #3

Group Members and Work Split

Team Member	***Task***
Amrit Adhikari amritadhikari@my.unt.edu	Load data, clean values, encode features, and perform exploratory data analysis
Son Dang sondang@my.unt.edu	Train model with logistic regression and decision tree; apply hyperparameter tuning
Bob Jack robertjack@my.unt.edu	Train and tune Random Forest/XGBoost and compare model performance
Crista Kingsley cristakingsley@my.unt.edu	Build confusion matrix(ces), receiver operator characteristic (ROC) curve, feature importance plots, monitor and assist other members work progress

Each member is responsible for adding their own contributions to the report and slide presentation, providing their respective source code, and collaborating on shared sections of the report such as introduction and conclusion.

Problem Statement

Our objective is to predict whether a Telco customer will cancel (churn) by learning from the information the company already collects about each account. The dataset gives us a mix of service-related signals, such as whether the customer currently has Online Security protection, what type of Internet Service they use, whether they have Multiple Lines or Phone Service, and how long they've been with the company (tenure)—along with a few personal or household indicators, including Dependents, Partner status, and gender. Every customer in the dataset will have a unique id called `customerID`. We will hold on to this ID as a reference, but we will not add it to the model. We want to explore the difference in customer traits and behaviors and how that relates to churn. Companies can use this model to detect people that are likely to cancel services and

subscriptions and enable them to take action to retain those customers. Additionally, they could use the model's results to estimate projected yearly losses from churn.

Proposed Method(s) and Dataset

The dataset we will use for this project comes from Kaggle called Telco Customer Churn. This publicly accessed dataset includes customer attributes and a Churn target column. This target is an indicator of whether a customer is likely to cancel their services. There are 21 feature columns and over 7000 rows in this dataset.

To get a baseline for us to understand churn, we'll try out some simpler models like Logistic Regression and Decision Trees. After that, we'll move on to other techniques like Random Forest and XGBoost. These are better at picking up complex patterns in the data.

Python will be used to write source code, using libraries like scikit-learn and XGBoost for the machine learning components. For data analysis and showing our results, we'll use tools like pandas, NumPy, Seaborn, and Matplotlib to spot trends and visualize how the models are performing.

Evaluation Approach

We will evaluate our models using classification metrics that are suited for churn prediction, which often involves imbalanced classes. Key metrics will include:

- Accuracy - to check for general performance overview
- Precision - to measure the correctly predicted churns (minimizing false positives)
- Recall - to evaluate the actually correct churn cases predicted (minimizing
- false negatives)
- F1-Score - to show the balance between precision and recall
- ROC-AUC - to give an overall separation between churn and non-churn customers

We will include confusion matrices to help visualize prediction outcomes. Also, ROC curves will be added to support model comparisons. We will apply cross-validation against our train/test splits to verify the reliability in our results and double check for overfitting.

If class imbalance is evident, we may add a resampling technique and/or some class weights to improve the recall values for churned customers.

Anticipated Challenges

- Handling missing or inconsistent data fields
- Correcting class imbalance from churn outcomes
- Correctly encode categorical variables
- Prevent model overfitting
- Selecting meaningful evaluation metrics

Planned Deliverables

- Google Colab notebook containing code for data preprocessing, modeling, and evaluation
- Summary report (PDF) detailing the methodology, results, and business implications
- Presentation slides to summarize the project workflow and findings
- Video presentation link (Optional)

Section 2: Final Results

Project Overview and Problem Formulation

We set out to predict customer churn using machine learning by following the workflow laid out in our initial project proposal. As a team, we explored and preprocessed the Telco Customer Churn dataset. After data preparation concluded, we applied several classification models (Logistic, Regression, Decision Tree, Random Forest, and XGBoost). Then, we tuned their hyperparameters to get the most out of each model. Following tuning, we evaluated their performance using metrics like precision, recall, F_1 score, and ROC-AUC. We also searched for the features that contributed the most to predicting churn with the goal of forming our results aimed at businesses who want to explore customer retention strategies.

Customer churn happens when someone cancels or stops using a service. In telecom, this directly affects revenue for many companies. The general idea is to keep existing customers by offering incentives for them to stay. Acquiring a new customer can cost five to 25 times more than retaining an existing one (Gallo, 2014). So, there is a lot of importance placed on predicting customer turnover. Ofek & Sarvary (2016) emphasize that winning back existing customers can be a powerful strategy. Early detection of churn gives companies an advantage to take necessary action to apply an acceptable strategy to retain their revenue stream.

In this project, we want to apply machine learning to resolve the following question:

Hypothesis: *Can we figure out which consumers will cancel company services based on their account details and usage habits?*

Solution:

- Build machine learning models that can classify customers into churn or no-churn
- Evaluate the models' performance on unseen data
- Dig into which features matter most, so companies can utilize the results to assist with retention strategies

Dataset and Feature Engineering

We used the well-known Telco Customer Churn dataset from Kaggle, which includes about 7,000 customer records (blastchar, 2017). It covers everything from demographics to service usage and billing history, giving us a solid foundation to build upon. The dataset includes a variety of categorical features like gender, contract type, and payment method, alongside numerical attributes such as tenure, monthly charges,

and total charges. That leaves us with a binary target variable, churn. The resulting value informs us whether the customer stays or drops their service.

Initially, we loaded the Telco Customer Churn dataset from Kaggle. Using pandas `DataFrame`, we examined the structure to better understand the shape and data types. Also, we evaluated the number of columns that the set used and how many rows of data it contained. This allowed us to place everything into perspective to understand the larger picture. From this, we quickly spotted the `customerID` column. It was only being used as an identifier and had no bearing on churn prediction, so it was dropped from the set.

Our next action was to clean the remaining data. The `TotalCharges` column needed to be converted to numeric values, but the column had intermittent blank spaces or other formatting problems. `NaN` values were added to each incident to allow the column to be processed. After this, we planned to handle these missing values appropriately. We also transformed the target column `Churn` from its original `'Yes'` and `'No'` strings into binary values `1` and `0`, which makes it easier to use in machine learning models.

After preparing the data types, we did some basic inspection for missing values by using `isnull().sum()` to count how many `NaN`s were in each column. This revealed that most columns were clean, with missing data mainly appearing in `TotalCharges` after the conversion step. We also checked the unique values in each column, especially for categorical features, to get a sense of what kinds of options were present (like contract types or internet services).

Categorical features were encoded into numeric forms using label encoding to make them suitable for machine learning algorithms. We adjusted numeric features to have a mean of *zero* and a standard deviation of *one*. The models could then train better at the same scale. Since the data set had far more non-churners than churners, we needed to handle this imbalance. To adjust the model, we set `class_weight` to `'balanced'` for both groups to be evaluated evenly.

Exploratory Data Analysis (EDA)

We began our exploration by looking at how churn is spread across the dataset. It quickly became clear that most customers did not churn, meaning there's a strong class imbalance. This is important to note because if we're not careful, any predictive model might just learn to always predict "no churn," ignoring the actual churn cases.

Next, we looked at how different features are linked to churn. The `tenure` data showed that customers who had only been with the company for a short time were much more

likely to leave. This points to the first few months being a critical period for keeping customers. Those who stayed longer tended to be more loyal and less likely to churn. Similar patterns have been observed in other telecom churn studies. For example, Mazuik (2021) found that customers with low tenure and month-to-month contracts were more likely to churn.

When we examined monthly charges, we did see that customers with higher bills were somewhat more likely to churn, though this trend wasn't as strong as what we found with tenure. Looking at contract types gave us even clearer insights: customers on month-to-month contracts had noticeably higher churn rates compared to those with one or two-year agreements. This suggests that longer contracts helped in retaining customers. We also initially saw that the type of internet service a customer had made a difference, with some services linked to higher churn than others.

Overall, these patterns gave us a good sense of which factors might be most important when we try to predict churn. Companies can focus on these factors to improve customer retention.

Model Development and Evaluation

To create a solution to our project's hypothesis, we established a structured pipeline to build and evaluate our churn prediction system. The setup was: model selection, training, validation, and performance comparison. We felt this would help us focus on identifying the customers who are likely to cancel their services from the telecom provided data.

We began by splitting the dataset into training and testing sets using an 80/20 ratio. We can train the model with a large portion of the available data, then use the remaining portion to perform an unbiased evaluation. To retain the original churn rate across both sets in the split, we applied stratification to keep churned and non-churn proportion intact.

To make our results more reliable and avoid depending on having only one train-test split, we used Stratified K-Fold Cross-Validation (Scikit-learn developers, n.d.-b). The original churn ratio was then kept consistent throughout each fold. This becomes very important when dealing with a class imbalance that is present in our dataset. We performed five folds that rotated the training, and validation sets to improve the average performance with the different partitions.

We started with two simple models to establish performance baselines:

- Logistic Regression was easy to understand and fast to train but struggled with non-linear patterns in the data.
- Decision Tree helped capture basic splits in the data but was prone to overfitting.

These two models were used to give us baseline results for the churn data. This gave us a baseline for comparing more complex algorithms.

For more powerful modeling, we implemented:

- Random Forest, which was used to bring a better balance to accuracy and reliability. It handles noise better and won't overfit like a decision tree.
- XGBoost, which works by repeatedly building models and correcting errors from the previous iterations (Chen & Guestrin, 2016).

Both advanced models helped us get better tuning and gain a better understanding of what factors affect customer churn.

To evaluate how well our models performed (even with imbalanced data), we used some different classification metrics to give us a clearer picture:

- Accuracy was used to get an overall understanding of correctness.
- Precision showed how many of our predicted churns were confirmed.
- Recall indicated how many of the real churn cases we were able to catch.
- F_1 Score balanced precision and recall keeping one from being favored over the other.
- ROC-AUC measured how well the model could separate churners from non-churners across different thresholds.

Combining these metrics gave us a better understanding of how each model performed. We can highlight the trade-offs between catching more churners versus avoiding false alarms.

With the four combined models, we could begin to analyze which inputs had the biggest influence on churn predictions. After training the models, we compared their performances with our chosen metrics. We placed more emphasis on recall and F_1 due to their relevance to imbalanced classification problems. From reviewing the general results, we determined which model deserved deeper evaluation.

To find the best possible parameter combination for each model, we used `GridSearchCV` with cross-validation (Scikit-learn developers, n.d.-a). This function found the optimal setup for each model to work with the data. This was applied to all four models so there would be a fair comparison of each model when we evaluated the

results. We adjusted the Logistic Regression model's regularization strength to avoid overfitting. The Decision Tree model needed adjustments controlling depth and minimum samples per split. This prevented the tree from just memorizing training data. With Random Forest, we tuned `n_estimators`, `max_depth`, and `min_samples_split` to balance between bias and variance. Lastly, for XGBoost, we focused on `learning_rate`, `n_estimators`, and `max_depth`, since it's more sensitive to those settings. Doing this helped each model generalize better and gave us a more honest picture of how they perform beyond just the training set.

After training and tuning each model, we evaluated their performance using our full set of classification metrics to see which one handled the churn prediction task best.

Results and Discussion

After training and tuning all four models with the best parameters, we evaluated their performance using the test set from the dataset split. The metrics used were accuracy, precision, recall and F_1 score. There was variance in how each of the four models dealt with churn prediction. This was even more evident when it came to identifying customers that were at risk of dropping their services.

Model	**Accuracy**	**Precision**	**Recall**	**F_1 Score**
Logistic Regression (Churn)	0.79	0.63	0.48	0.55
Logistic Regression (Non-Churn)		0.83	0.90	0.86
Decision Tree (Churn)	0.78	0.58	0.54	0.56
Decision Tree (Non-Churn)		0.84	0.86	0.85
Random Forest (Churn)	0.73	0.49	0.78	0.60
Random Forest (Non-Churn)		0.90	0.71	0.80
XGBoost (Churn)	0.77	0.54	0.70	0.61
XGBoost (Non-Churn)		0.88	0.79	0.83

Table 1. Precision, Recall, and F_1 score for churn and non-churn classes across all models shows how each model performed on both classes, with a focus on identifying churn (class 1) accurately.

Logistic Regression and Decision Tree gave us a decent baseline to start our evaluation. They both struggled with recall in the churn class. Logistic Regression only achieved a 0.48 recall score on the churn class. Decision Tree did a bit better in all the scores, but it missed a large number of true customer churns.

Random Forest made better gains in recall by scoring 0.78. Although, this increase came at the cost of precision. It was flagging more customers for being at risk for churn when they were not. The best overall balance came from XGBoost. It had improvements in both recall and F_1 score without having losses in accuracy. Out of the four models, we feel this one was the most dependable. Table 1 shows these performance metrics from the model evaluation.

After looking at these results, we notice that the models performed much better in the majority class (non-churn). Precision and recall were consistently higher for this group across all four models. Our main goal was to only identify the churner class, so we only focused on evaluating the values in `Churn`. This is the point where we begin to see gaps in the results. Logistic Regression only caught 48% of churners, yet XGBoost improved recall and held a higher F_1 score. Random Forest was closer to XGBoost than the baseline models, but it fell short in points across the board to XGBoost.

Next, we wanted to visualize how each model made its predictions on the test set. We ran confusion matrices for all four to get each model's correct and incorrect predictions in two categories: churners (1) and non-churners (0). From Table 2, we can see that false negatives are critical in this problem. They represent customers who were predicted to stay but left instead. From a business standpoint, mis-identifying a customer who will churn is much worse than flagging a customer who won't leave. So, these false negatives would be the situations where a company did not take any action and lost that customer, and in effect lost revenue.

Model	**True Pos.**	**False Pos.**	**True Neg.**	**False Neg.**
Logistic Regression	178	106	935	190
Decision Tree	200	146	895	168
Random Forest	287	297	744	81
XGBoost	257	216	825	111

Table 2. Confusion matrix results for all models display true positives, false positives, true negatives, and false negatives for each model, highlighting differences in churn prediction accuracy.

Again in Table 2, we see that Logistic Regression and Decision Tree had high false negative counts (190 and 168, respectively). This lowered their recall scores and reduced reliability in detecting true churn cases. Random Forest did the best out of all the models by catching more churners by having a false negative count of 81. Although, this came at a cost by having the highest count of false positives at 297. This means that many more customers were flagged as potential churners who were not. This could lead a company to spend more than necessary on incentives to keep their revenue stream.

XGBoost had the best balance out of all four models. At 111 false negatives, it fared better than the baseline models but did not do as well as Random Forest. However, itkept its false positives lower at 216. This gave XGBoost a much better recall score than Random Forest, making it a much better predictor for this problem all around than the other models. This supports the idea that XGBoost is a good match for a structured and tabular problem like churn (Chen & Guestrin, 2016). To figure out how each model compares in separating churners from non-churners, we plotted their Receiver Operating Characteristic (ROC) curves and compared the Area Under the Curve (AUC) values.

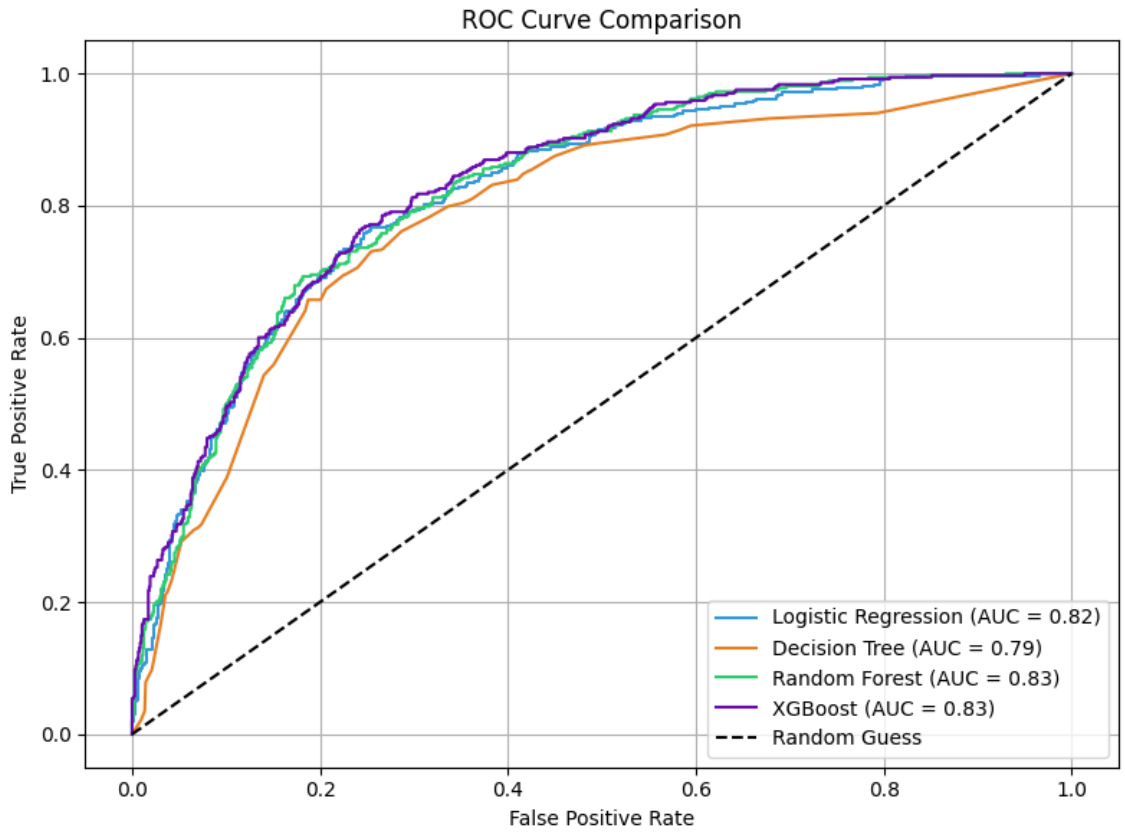

Figure 1. ROC Curve Comparison for All Four Models. XGBoost and Random Forest showed the highest AUC, indicating stronger overall ability to distinguish churners from non-churners.

Figure 1 shows the ROC curve with all models against the random guess line. Both XGBoost and Random Forest had the best ROC scores at 0.83, followed by Logistic Regression at 0.82, and Decision Tree in the last spot with 0.79.

The ROC curve plots the true positive rate (recall) against the false positive rate at various thresholds. The ideal model that could achieve a perfect separation would be up against the upper-left corner of the graph in Figure 1. This would give an AUC score of 1.0. A model that does no better than a random guess, would be near the center line and have an AUC score of 0.5.

In this problem, all four of the models were significantly better than random guesses. The higher curves for Random Forest and XGBoost indicate their ability to separate the two churn classes. This matches up with the values we saw from the evaluation metrics. The advanced models had better recall and F_1 scores while keeping a competitive accuracy score. Again, we see XGBoost performing better than the baseline models and closely edges out Random Forest.

We looked at feature importance for us to see what columns in our data have the most impact in predicting churn. The two best-performing models, Random Forest (Figure 2)

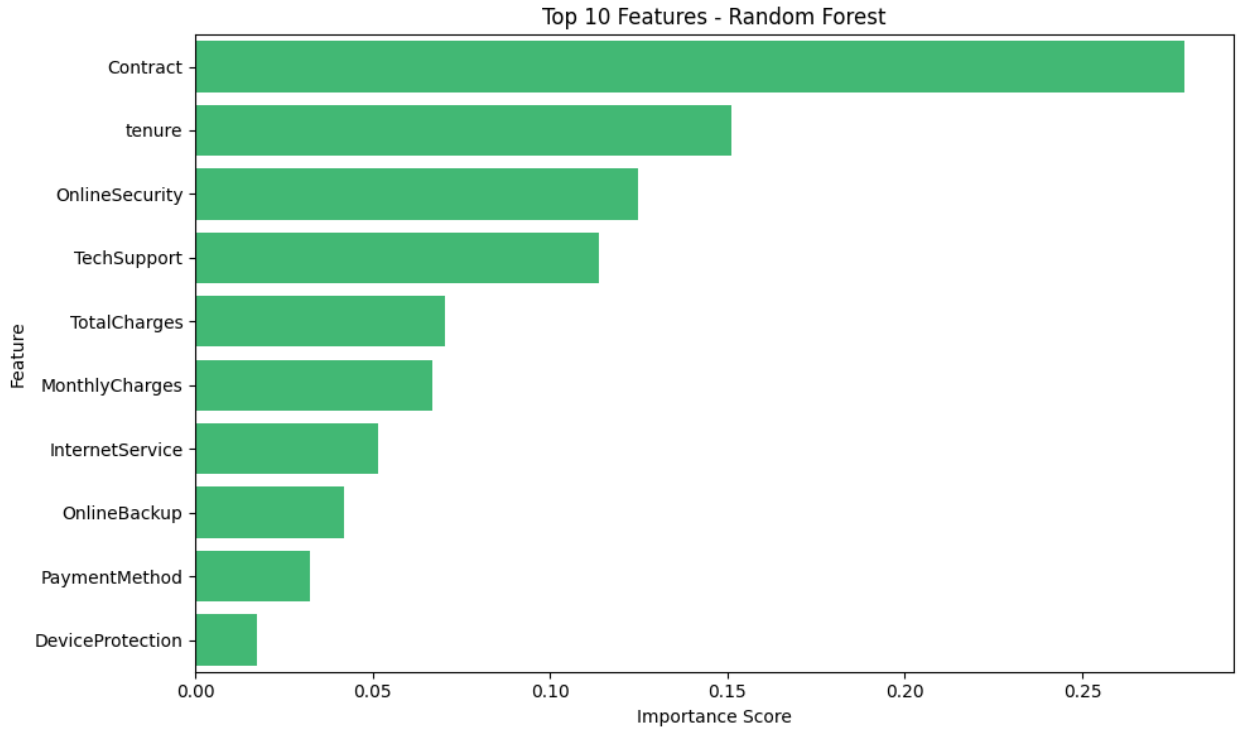

Figure 2. Top 10 Most Important Features According to Random Forest. Contract type and tenure were the strongest predictors of churn, followed by support and billing-related services like OnlineSecurity and MonthlyCharges.

and XGBoost (Figure 3), have built in measures dealing with feature importance. We can rank input variables to see how they contribute to detecting churn. Due to the performance gap in the advanced models and the baseline, we are focusing on highlighting the features for these two for brevity.

Both models show `Contract` being the most predictive feature from the data. This matches up logically because customers that have month-to-month contracts would be more likely to cancel than customers with long-term contracts. In Figure 2, you see another strong predictor, `tenure`. This feature is the duration that a customer has been with the company. This seems to be in line with real-world patterns. Customers with month-to-month contracts may be only evaluating the company's service, whereas a long-term customer may have issues with breaking a service contract.

The other top-ranked features in Figure 3 include `OnlineSecurity`, `TechSupport`, and `TotalCharges`. These may show that customers who feel the security and tech support should be significant enough for the value of the service they are paying for.

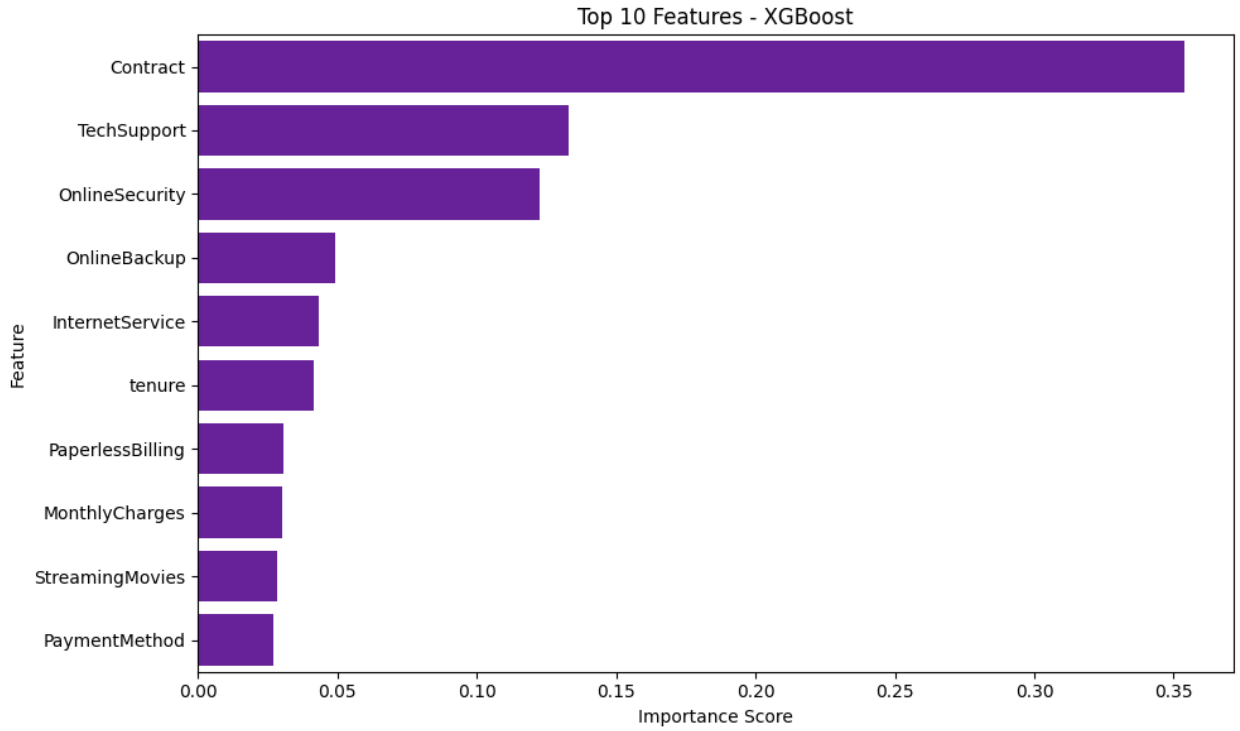

Figure 3. Top 10 Most Important Features According to XGBoost. Contract type remained the most influential feature, with TechSupport, OnlineSecurity, and OnlineBackup also playing key roles in identifying churn risk.

Otherwise, they will leave. Feature importance can show us the areas that a company can focus their attention on getting the most out of their investment to retain customers.

The four models used had their own strengths and trade-offs. Logistic Regression was fast, lightweight, and easy to interpret. It did not have good performance with churn recall compared to the complex models. The Decision Tree improved slightly over Logistic Regression, but it needed to be properly tuned to avoid overfitting. We saw a big jump in recall with Random Forest over the baseline models. This was the first model to show that it was more reliable in spotting customer churn. Although XGBoost did better across the board with a better overall balance of precision, recall and F_1 score. Hyperparametric tuning was likely the factor in this difference.

For applying these results to prospective businesses, this type of model can help companies use these predictions and reach out to customers that are flagged as risks. Whether they offer incentives, personalized support, or contract adjustments, they can address turnover and try to retain their revenue stream.

There are some limitations in this project to reliably detect churn. We worked with a dataset that was relatively small and not too complex. There could be data that would involve customer support call logs, surveys, or competition service pricing that could be added to improve our predictions. We also want to keep watch on how false positives are flagged. Companies will waste money and resources on chasing these customers who are wrongly identified. This is a trade-off that a company will need to choose when they decide the level of aggressiveness in strategy for retaining customers.

Conclusion

In this project, we built a model that predicts whether a customer will churn or not using their account information and usage data. After preprocessing, encoding, and addressing class imbalance, we trained our four models: Logistic Regression, Decision Tree, Random Forest, and XGBoost. The baseline models were simple and easy to interpret, but they struggled to predict customers who were likely to churn. Random Forest greatly improved in its recall score, but XGBoost outperformed all models by balancing recall, precision, and F1 score more effectively than the rest. A telecom company could reliably use the XGBoost model to find customers who are likely to drop services and make efforts to retain them.

Deployment of these models does have trade-offs. The advanced models are harder to interpret. If transparency is an issue, then this could be a potential challenge. Future improvements could include adding more data sources like customer support logs, or satisfaction surveys to improve predictions. Integration into a dashboard would help

companies submit data easier and be able to work with the results in formatting that is suitable for them to best retain customers and reduce revenue loss. From the work done in this project, we were able to reliably identify which consumers were subject to churn with the XGBoost model, and we identified the major factors contributing to churn from the used dataset.

References

blastchar. (2017). *Telco Customer Churn* [Data set]. Kaggle. https://www.kaggle.com/datasets/blastchar/telco-customer-churn

Chen, T., & Guestrin, C. (2016). *XGBoost: A scalable tree boosting system*. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). Association for Computing Machinery. https://doi.org/10.1145/2939672.2939785

Gallo, A. (2014, October 29). *The value of keeping the right customers*. Harvard Business Review. https://hbr.org/2014/10/the-value-of-keeping-the-right-customers

Maziuk, M. (2021, January 24). *Customer churn analysis: How to identify dissatisfied customers*. MaryiaMaziuk.com. https://maryiamaziuk.com/customer-churn/

Ofek, E., & Sarvary, M. (2016, March). *Winning back lost customers*. Harvard Business Review. https://hbr.org/2016/03/winning-back-lost-customers

Scikit-learn developers. (n.d.-a). *sklearn.model_selection.GridSearchCV*. Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

Scikit-learn developers. (n.d.-b). *sklearn.model_selection.StratifiedKFold*. Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html