Churn Prediction Using Subscription-based Service Customer Churn Data

Yaqi Liu
DSI, Brown University
GitHub Repository (Imamberr, 2024)

---

1. Introduction

Customer churn, where customers stop engaging with a company, presents major challenges across industries. As acquiring customers is costly, retaining them is crucial. Predicting churn allows businesses to implement proactive strategies, like targeted promotions or personalized support, to reduce churn rates. My object through this project is to develop a binary classifier to predict whether a customer will churn for a subscription based service. This study uses an iid Kaggle dataset with 243,781 observations and 20 features, including numerical, categorical, and ordinal data related to customer behavior with no missing value. (Safrin03, n.d.)

A key challenge in churn prediction is class imbalance, where non-churning customers far outnumber churners. Previous research highlights the effectiveness of machine learning in tackling this issue. Umut Berke Koç achieved F1 scores of up to 0.42 using XGBoost and Random Forest models, emphasizing the importance of non-linear approaches. (UmutBerkEkoç,2024) Pedram A. Darestani reported AUC scores of 0.85 for logistic regression and 0.90 for decision trees in the telecom industry, showcasing the value of feature engineering and advanced techniques.(Abdolahi, 2024)

Building on these findings, this project evaluates Logistic Regression, Random Forest, Support Vector Machines, and XGBoost to predict churn. The goal is to identify the best-performing model while gaining insights into churn drivers. The F1 score is used for evaluation, given its suitability for imbalanced datasets. This study also applies robust cross-validation and explores feature importance using interpretability tools like SHAP, contributing to the existing research.

---

2. Exploratory Data Analysis

The target variable, Churn, indicates whether a customer has discontinued their subscription, with a churn rate of 18% (Fig 1), highlighting significant class imbalance.

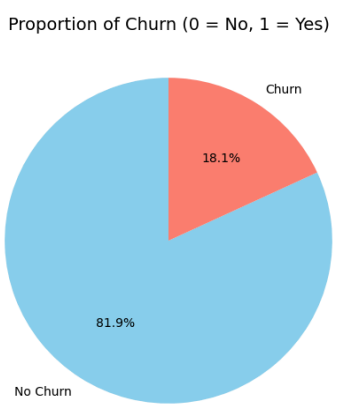

Fig 1. Pie Chart of churn rate proportion

Column_name	Column_type	Data_type	Description
21 unique values	Feature 90% Identifier 5% Other (1) 5%	string 48% float 29% Other (5) 24%	21 unique values
AccountAge	Feature	integer	The age of the user's account in months.
MonthlyCharges	Feature	float	The amount charged to the user on a monthly basis.
TotalCharges	Feature	float	The total charges incurred by the user over the account's lifetime.
SubscriptionType	Feature	object	The type of subscription chosen by the user (Basic, Standard, or Premium).
PaymentMethod	Feature	string	The method of payment used by the user.
PaperlessBilling	Feature	string	Indicates whether the user has opted for paperless billing (Yes or No).
ContentType	Feature	string	The type of content preferred by the user (Movies, TV Shows, or Both).
MultiDeviceAccess	Feature	string	Indicates whether the user has access to the service on multiple devices (Yes or No).
DeviceRegistered	Feature	string	The type of device registered by the user (TV, Mobile, Tablet, or Computer).
ViewingHoursPerWeek	Feature	float	The number of hours the user spends watching content per week.

AverageViewingDuration	Feature	float	The average duration of each viewing session in minutes.
ContentDownloadsPerMonth	Feature	integer	The number of content downloads by the user per month.
GenrePreference	Feature	string	The preferred genre of content chosen by the user.
UserRating	Feature	float	The user's rating for the service on a scale of 1 to 5.
SupportTicketsPerMonth	Feature	integer	The number of support tickets raised by the user per month.
Gender	Feature	string	The gender of the user (Male or Female).
WatchlistSize	Feature	float	The number of items in the user's watchlist.
ParentalControl	Feature	string	Indicates whether parental control is enabled for the user (Yes or No).
SubtitlesEnabled	Feature	string	Indicates whether subtitles are enabled for the user (Yes or No).
CustomerID	Identifier	string	A unique identifier for each customer.
Churn	Target	integer	The target variable indicating whether a user has churned or not (1 for churned, 0 for not churned).

Table 1. Data descriptions

The following correlation heatmap revealed that **AccountAge** has the strongest negative correlation with churn, showing customers with longer account durations are less likely to churn. **AverageViewingDuration**, the second-highest negative correlation, suggests that longer viewing sessions also reduce churn likelihood.

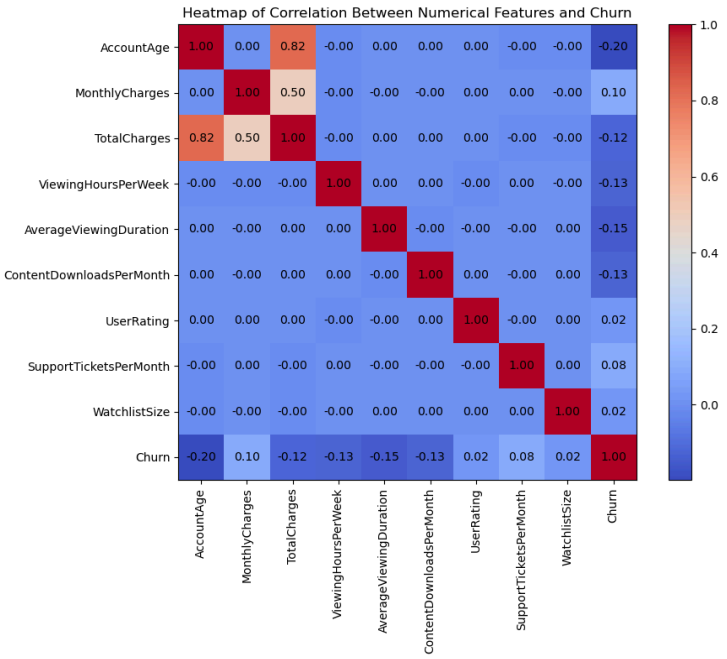

Fig 2. Correlation heatmap of Numerical Features with target variable Churn

To further explore the relationship between `AccountAge` and churn, a box plot was generated to compare the distribution of account age for churned and non-churned customers. This visualization (Fig 3) confirmed a distinct pattern: churned customers tend to have significantly lower account ages compared to those who remain subscribed. This emphasizes the importance of customer retention strategies during the early stages of the subscription lifecycle.

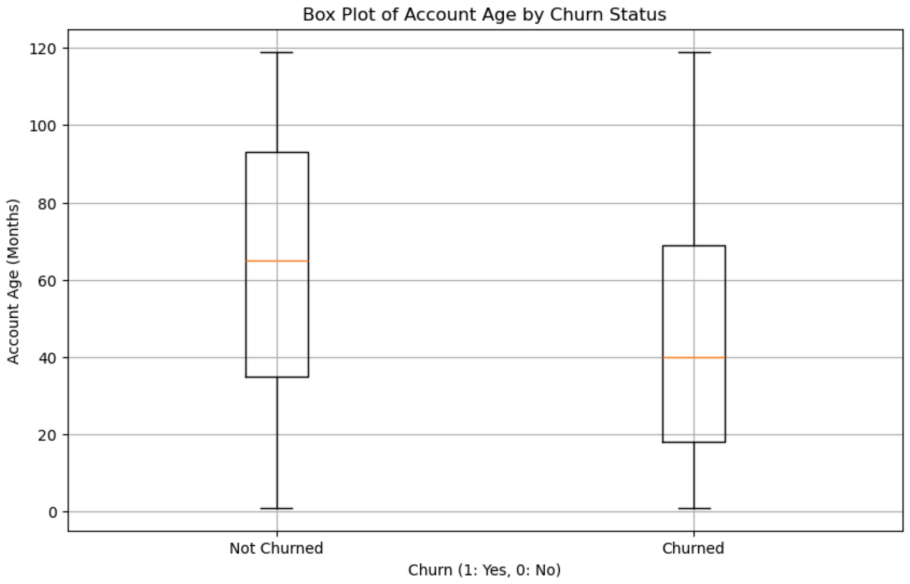

Fig 3. Box Plot of Account Age by Churn Status

The next figure (Fig 4.) shows the relationship between customer behavior and churn. It shows that customers with "Basic" subscriptions and those using "Electronic Check" or

"Mailed Check" payment methods have the highest churn rates. In contrast, "Premium" subscriptions and automated payments like "Bank Transfer" and "Credit Card" have the lowest churn rates, highlighting subscription tiers and payment methods as key churn predictors.

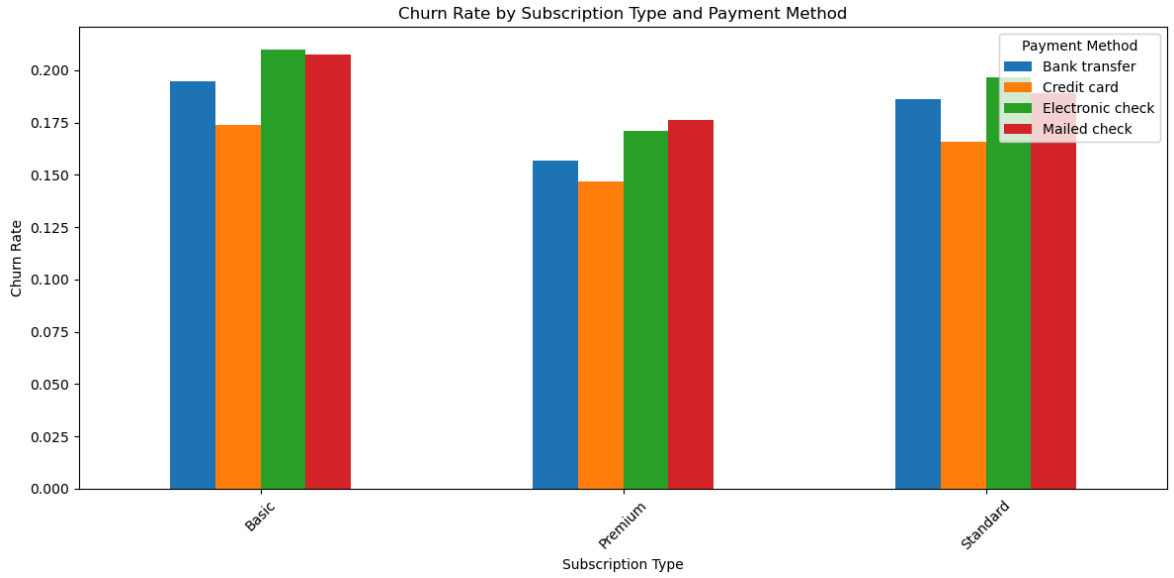

Fig 4. Churn Rate by Subscription Type and Payment Method

---

3. Methodology

This section outlines the splitting strategy, data preprocessing, machine learning pipeline, evaluation metrics, and model selection. To measure the uncertainties due to data splitting and the non-deterministic nature of certain ML algorithms, we evaluated each model over five different random states. The primary objective is to develop a robust churn prediction pipeline and evaluate its performance using multiple ML algorithms while addressing the challenges of class imbalance and model uncertainty.

3.1 Splitting Strategy

The dataset was split into a test set (20%) and "other" set (80%), using stratified sampling to preserve the churn class proportions. The test set was reserved for final evaluation, while the "other" set was split into training (60%) and validation (20%) sets using 5-fold Stratified Cross-Validation, keeping the proportion of churn and non-churn customers across folds.

3.2 Data Preprocessing

Feature preprocessing was performed using a `ColumnTransformer` pipeline to handle the diverse data types present in the dataset. The preprocessing pipeline was designed to

handle the diverse feature types present in the dataset. Numerical features were standardized using a `StandardScaler`, ensuring that they were on a uniform scale with zero mean and unit variance. The ordinal feature, `SubscriptionType`, was encoded using an `OrdinalEncoder` with a predefined order (`Basic`, `Standard`, `Premium`), capturing the natural ranking of subscription levels, while categorical features were transformed into numerical representations using a `OneHotEncoder`.

3.3 Addressing Class Imbalance

The dataset had a significant class imbalance, with most customers not churning. To address this, different techniques were applied across models. For Logistic Regression, Random Forest, and SVM, the parameter `class_weight='balanced'` was used to automatically adjust class weights inversely to their frequencies in the dataset. For XGBoost, the parameter `scale_pos_weight` was set to the ratio of the negative class to the positive class, increasing the weight of churned customers. These approaches ensured the models focused on the minority class (churn) during training, reducing bias toward the majority class and improving the prediction of churned customers.

3.4 Hyperparameter Tuning and Cross-Validation

The preprocessing pipeline, model, and hyperparameter tuning were integrated into a `Pipeline` object, ensuring that all steps were applied consistently during both training and validation. Once the best hyperparameters were identified for a model, the entire pipeline was retrained on the training set. 5 random states were iterated to calculate the average performance and standard deviation for each model, providing reliable estimates of their generalization capabilities.

Each model underwent hyperparameter tuning using `GridSearchCV` to identify the optimal parameters that maximize the F1 score. The F1 score was chosen as the primary evaluation metric because of its ability to balance precision and recall, which is particularly important for imbalanced datasets like this one. In the churn prediction context, precision measures how many predicted churners are actual churners, while recall captures how many of the actual churners are identified. Optimizing for the F1 score ensures the model is not overly biased toward either false positives or false negatives, striking a balance between these metrics to deliver robust performance.

The grid search was integrated with the Stratified K-Fold CV to evaluate each hyperparameter configuration across multiple folds. The parameters tuned for each model included:

Model	Tuned parameters

L1-Logistic Regression	C: [0.001, 0.01, 0.1, 1, 10, 100, 1000]
L2-Logistic Regression	C: [0.001, 0.01, 0.1, 1, 10, 100, 1000]
Random Forest	max_depth: [None, 1, 3, 5, 10, 20], max_features: [0.3,0.5,0.7,1.0]
SVM	C: [0.001, 0.01, 0.1, 1, 10, 100, 1000], kernel: ['linear', 'rbf'], gamma: ['scale', 'auto', 0.01, 0.1, 1]
XGBoost	n_estimators: [50, 100, 200, 500], max_depth: [1, 3, 5, 7, 10], Learning rate: [0.01, 0.1, 0.2]

Table 2. Tuned hyperparameters ranges for ML algorithms

For each model, a range of thresholds (from 0.1 to 0.9) was tested to determine the decision boundary that maximized the F1 score. This approach ensured that predictions were finely tuned to the problem's class imbalance.

---

4. Results

4.1 Model Performance Compared to Baseline

The baseline F1 score, calculated based on the class distribution of 22% churners and 78% non-churners, was 0.18. All tested models significantly outperformed this baseline, with their average F1 scores ranging from 0.4162 (Random Forest) to 0.4489 (XGBoost).

The comparison of model performances is summarized in Table 3 and visualized in Figure 5. XGBoost achieved the highest average F1 score (0.4489 ± 0.028) which is the most predictive ML algorithm, followed closely by Logistic Lasso (0.4469) and Logistic Ridge (0.4466). Random Forest was the least predictive model with an average F1 score of 0.4162, while SVM performed moderately well with an F1 score of 0.4440. Standard deviations are low across all models.

Model	Average F1 Score	Avg Std Dev	Best Params
logistic_lasso	0.4469	0.029	C: 0.1
logistic_ridge	0.4466	0.03	C: 0.01
random_forest	0.4162	0.026	max_depth: 5, max_feature: 0.7

svm	0.4440	0.026	C: 10, gamma: 'auto', kernel: 'rbf'
xgboost	0.4489	0.028	learning_rate: 0.1, n_estimators: 200, max_depth: 3

Table 3. Performance Summary of Machine Learning Models with Optimized Hyperparameters

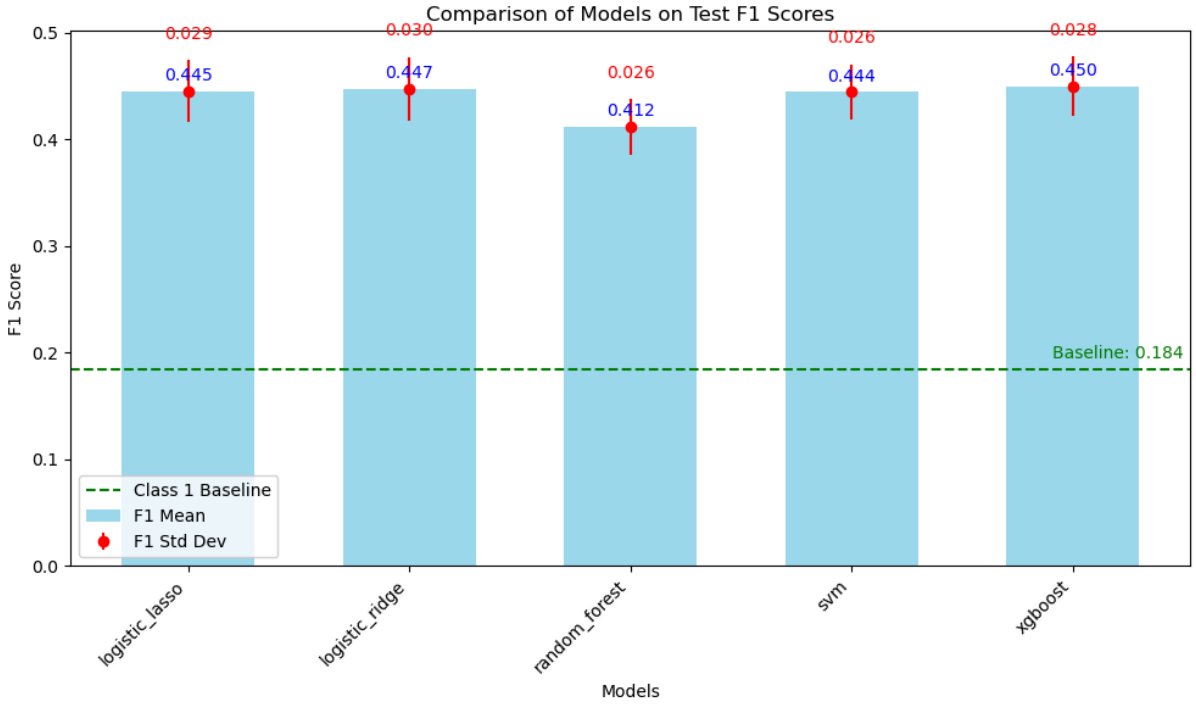

Fig 5. Comparison of Models Based on Test F1 Scores and Standard Deviations

The confusion matrix (Fig 6.) for the XGBoost model at the best threshold (0.50) shows that it correctly predicts 68% of churn cases (highest true positives rate across all models), with a lowest false negative rate (32%) across all models, indicating a balanced performance in handling both classes.

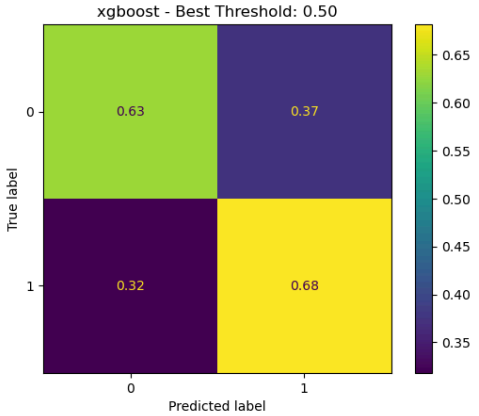

Fig 6. Confusion Matrix for XGBoost Model at Best Threshold (0.50)

4.2 Global Feature Importance

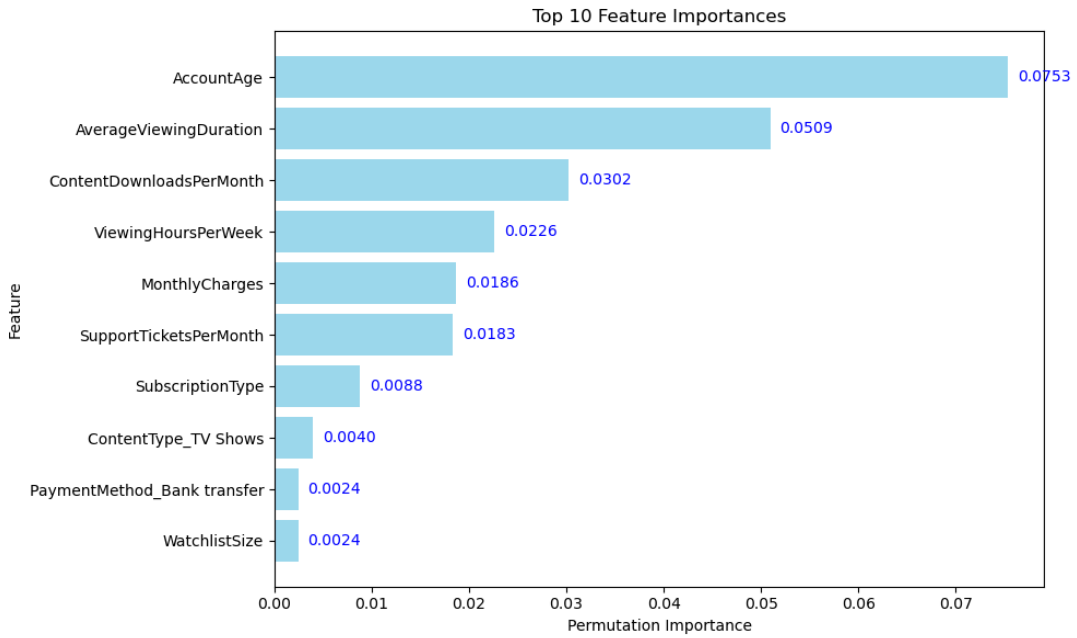

Fig 7. Permutation importance plot

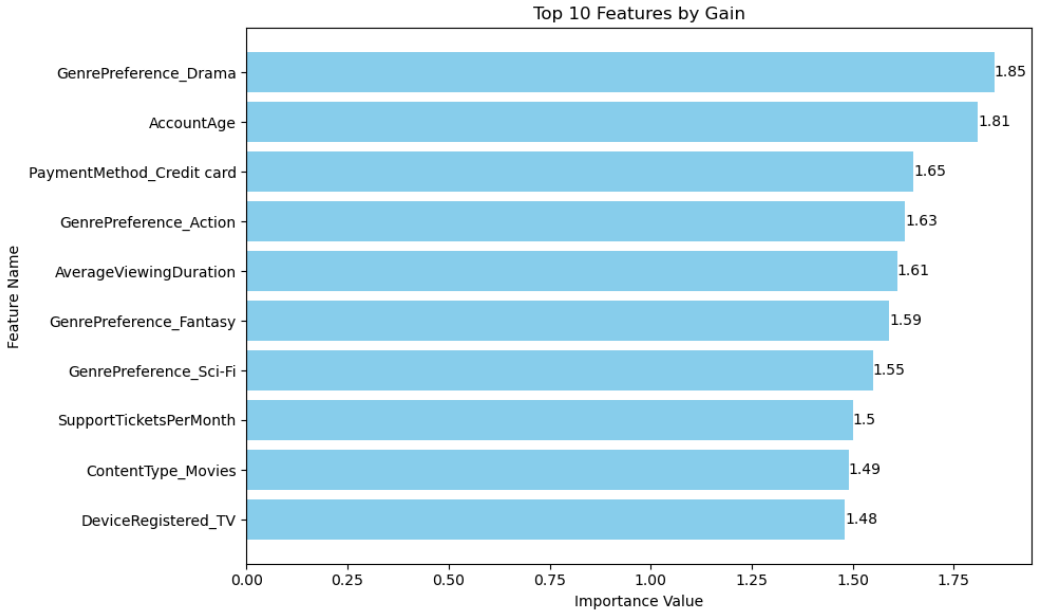

Fig 8. Feature Importance Plot by Gain for XGBoost

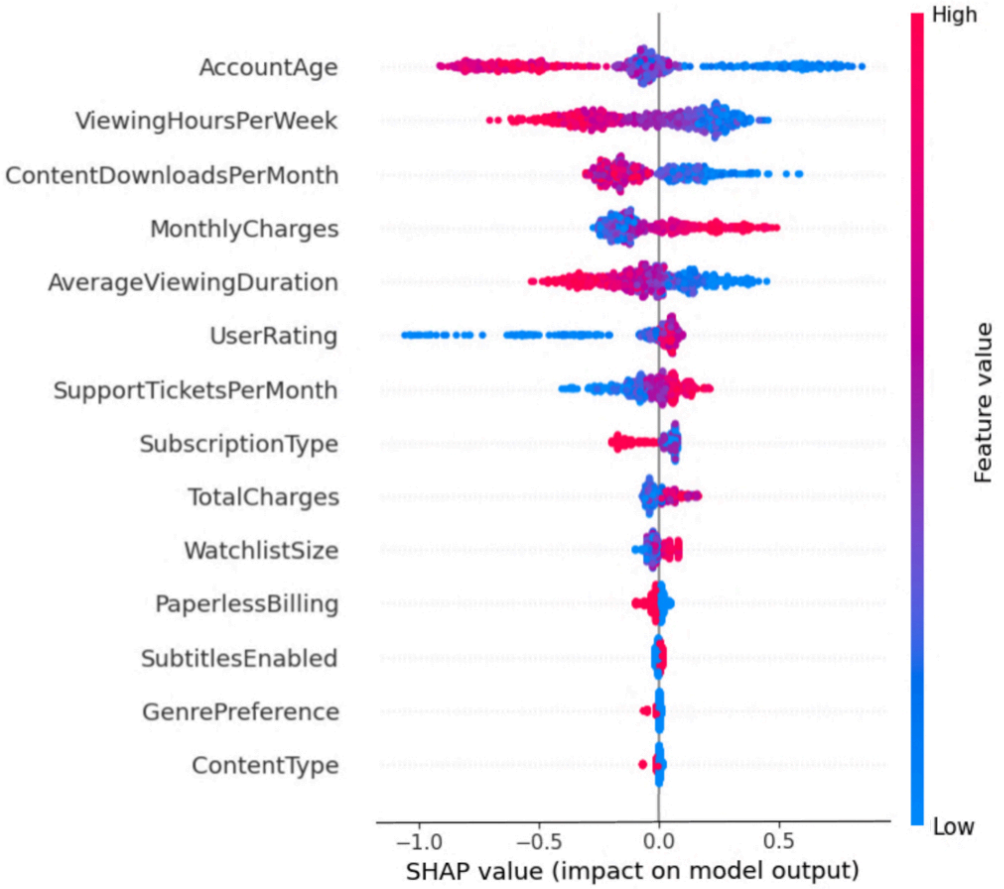

Fig 9. SHAP Summary Plot: Feature Impact on Model Predictions

Permutation importance, gain-based feature importance, and SHAP values consistently highlighted key features driving churn predictions, aligning well with the practical implications of the churn problem. Across all methods, AccountAge was the most significant feature, highlighting that customers with longer subscriptions are less likely to churn, emphasizing the need for retention strategies targeting newer subscribers. AverageViewingDuration and ViewingHoursPerWeek were also critical, reflecting that higher engagement reduces churn likelihood. These insights underscore the importance of promoting relevant content and personalized recommendations to boost user activity.

On the other hand, features like WatchlistSize, GenrePreference and ContentType_TV Shows consistently showed low importance across all methods, which is surprising given their connection to content engagement. This suggests that broader behavioral patterns, such as overall viewing consistency and account longevity, are more predictive of churn than specific content-related metrics. It may also reflect the possibility that simply having a large watchlist does not directly translate into actual engagement with the service.

An interesting finding is that the top three highly correlated features from the heatmap ("ViewingHoursPerWeek," "AccountAge," and "AverageViewingDuration") align with the top three most important features in the permutation importance plot. This consistency validates

the robustness of the feature importance analysis. Additionally, features related to customer preferences, such as "GenrePreference" categories and "PaymentMethod," repeatedly appeared as significant contributors, indicating their relevance to churn prediction.

4.3 Local Feature Importance

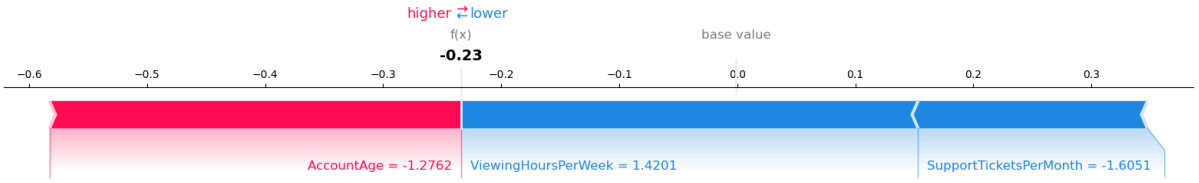

Fig 10. SHAP Force Plot for index 150

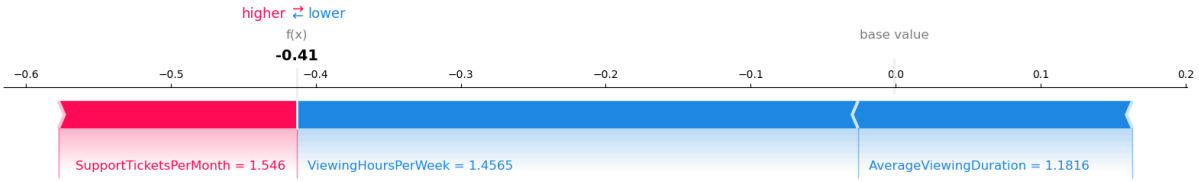

Fig 11. SHAP Force Plot for index 250

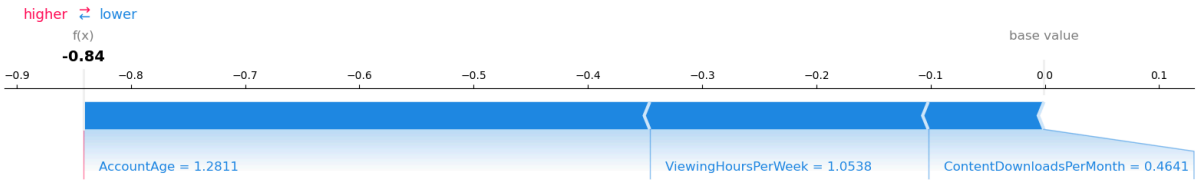

Fig 12. SHAP Force Plot for index 500

The SHAP Force Plot for index 150 (Fig 10) , with a prediction of -0.23, suggests the customer is not likely to churn. AccountAge contributes significantly to this, showing that a shorter account age increases churn risk. SupportTicketsPerMonth also negatively impacts the prediction, indicating potential dissatisfaction. ViewingHoursPerWeek provides a negative influence, suggesting that weekly engagement mitigates churn risk.

In the SHAP Force Plot for index 250 (Fig 11), the prediction of -0.41 indicates a likelihood of not churn. SupportTicketsPerMonth is the strongest positive factor, highlighting dissatisfaction as a major issue. ViewingHoursPerWeek and AverageViewingDuration

negatively impact the prediction, showing that consistent engagement and long viewing sessions help reduce churn risk.

The SHAP Force Plot for index 500 (Fig 12), with a prediction of -0.84, suggests this customer is highly unlikely to churn. AccountAge plays the most significant role, strongly reducing churn risk. ViewingHoursPerWeek and ContentDownloadsPerMonth also negatively influence the prediction, reflecting sustained weekly engagement and moderate interaction with downloadable content.

---

5. Outlook

There are several opportunities to improve predictive power and interpretability. Creating interaction features, such as combining "AccountAge" with "MonthlyCharges," and generating time series features, like trends in "ViewingHoursPerWeek," could capture more nuanced patterns in customer behavior. Additionally, incorporating ensemble methods, such as stacking or blending, could combine the strengths of different models to achieve better predictions.

Finally, collecting additional data, such as customer feedback, satisfaction ratings, or time-based behavioral trends, could provide a deeper understanding of churn drivers. Integrating external data sources, such as social media engagement, might also reveal hidden patterns. These enhancements would improve accuracy and actionable insights, enabling effective strategies for customer retention.

---

6. References

1. Imamberr. (2024, December 15). *DATA1030 Final Project*. GitHub. Retrieved December 15, 2024, from https://github.com/imamberr/DATA1030_final_project.git
2. Safrin03. (n.d.). *Predictive Analytics for Customer Churn Dataset*. Kaggle. Retrieved December 15, 2024, from https://www.kaggle.com/datasets/safrin03/predictive-analytics-for-customer-churn-dataset?select=data_descriptions.csv
3. UmutBerkEkoç. (2024). *Predictive Analytics for Customer Churn*. Kaggle. Retrieved December 15, 2024, from https://www.kaggle.com/code/umutberkekoc/predictive-analytics-for-customer-churn
4. Abdolahi, P. (2024). *Churn Prediction*. Kaggle. Retrieved December 15, 2024, from https://www.kaggle.com/code/pedramabdolahi/churn-prediction