How Show Recommendations In Netflix Affects User Churn Rate

eCommerce

Capstone Project

AIM Artificial Intelligence and Machine Learning

Adrian Centeno

Part I.

What is the business problem?
What ML task is needed for this?
What are your metrics to define success?

Netflix is known to be one of the biggest streaming platforms currently in the world. Popular enough that you would hear a lot about it and see it in all medias, news, and even amongst friends and families as they give you recommendations, reviews, and feedback to all the shows, movies, and even games that are available in its application. But how does Netflix become big? What do they do right that makes their users come back time and time again to see what's the new movie released? Or if the new season for their favorite series has already been released?

The business problem I came up with is how a platform like Netflix can sustain its viewers and users consistently. How can Netflix continue being one of the top platforms even with multiple other streaming platforms out there, even released before the time of Netflix. The first thing that came into mind, is whenever we talk about Netflix it would always seems to recommend the right shows, catered and personalized for each person. Each family member can setup their own account and based from what each person watches, every family member gets recommended different movies and shows. So for this business problem, I would like to focus on the Binary Classification task, and see how better show recommendations help keep subscribers coming back for more.

For this we would have to measure using AUC-ROC, our primary metric and the secondary metrics being Accuracy, Precision, Recall, and the F1-Score. AUC-ROC is ideal for it measures the model's ability to discriminate between the two classes of active and inactive users across the given thresholds.

We are focusing on the churn prediction of the company. Hence what is important is to have balance amongst the classes that are being compared to have accurate data. Since usually there is a smaller group of people who would unsubscribe, it is easy to fall for class imbalance. This is why finding the ROC-AUC is useful to measure the model's ability to differentiate between those churned and those who stays. The other secondary metrics like accuracy is for aiming on how correct the model will be. Precision is for how much of those who churn actually do churn, and recall measures how many of those who churned are correctly identified. Lastly F1-Score is to balance precision and recall. It captures both precision and recall and captures their relation in its own metric. Altogether these metrics are important for a good framework to churn prediction.

Part II.

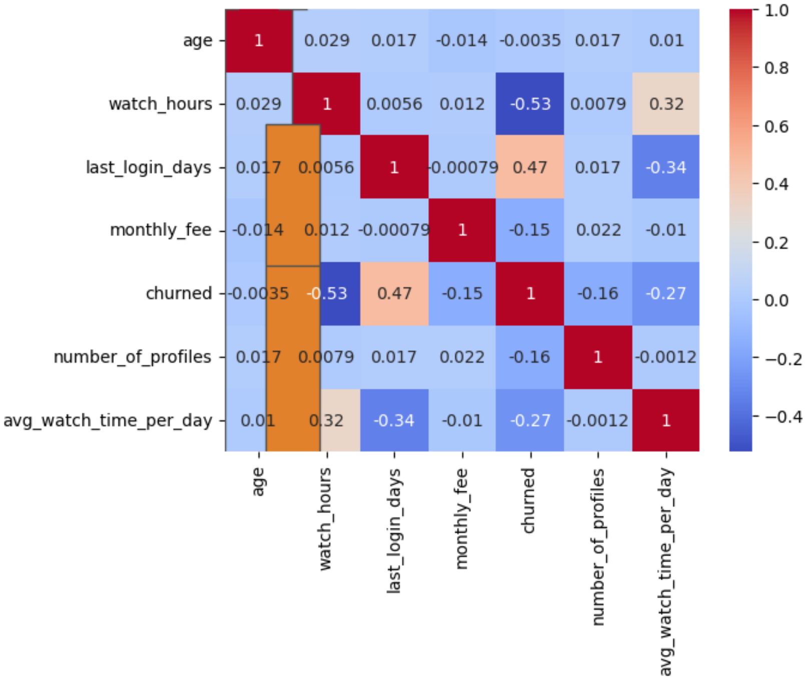
Finding the right datasets.
Provide a Data Dictionary.

As for the dataset I was able to pick up from Kaggle website a dataset that already includes Netflix Customer Churn Dataset.

Wadood, A. (2025, July 5). Netflix customer churn dataset(upvote if you like). Kaggle. [*https://www.kaggle.com/datasets/abdulwadood11220/netflix-customer-churn-dataset/data*](https://www.kaggle.com/datasets/abdulwadood11220/netflix-customer-churn-dataset/data)

Part III.

Cleaning, exploring, transforming the data
EDA
Feature Engineering Report with code & Justifications



EDA or exploratory data analysis is important for it shows the characteristics of the dataset as we identify the found patterns that shows how related each variable is to each other. In this EDA a histogram with KDE was made to help understand the distribution of watch hours among users. A box plot is used to compare the distribution of watch hours to those who churned and those that did not. Lastly the correlation heatmap is for all numerical features important to identify the relationships of each variable to one another.

Feature engineering is an important process for creating new features from the data given to help improve the performance of your machine model. In the model there were three new features engineered to capture the user engagement important for the business problem we are trying to solve.

Engagement_intensity = *(watch_hours/last_login + 1)*: This feature was made to represent the how much active the user is based on their watch time in relation to their last login. Higher values mean more watch time.

Heavy_user = would represent as a binary feature. A user would be classified “heavy” if their watch_hour is more than the median of watch_hours of all users.

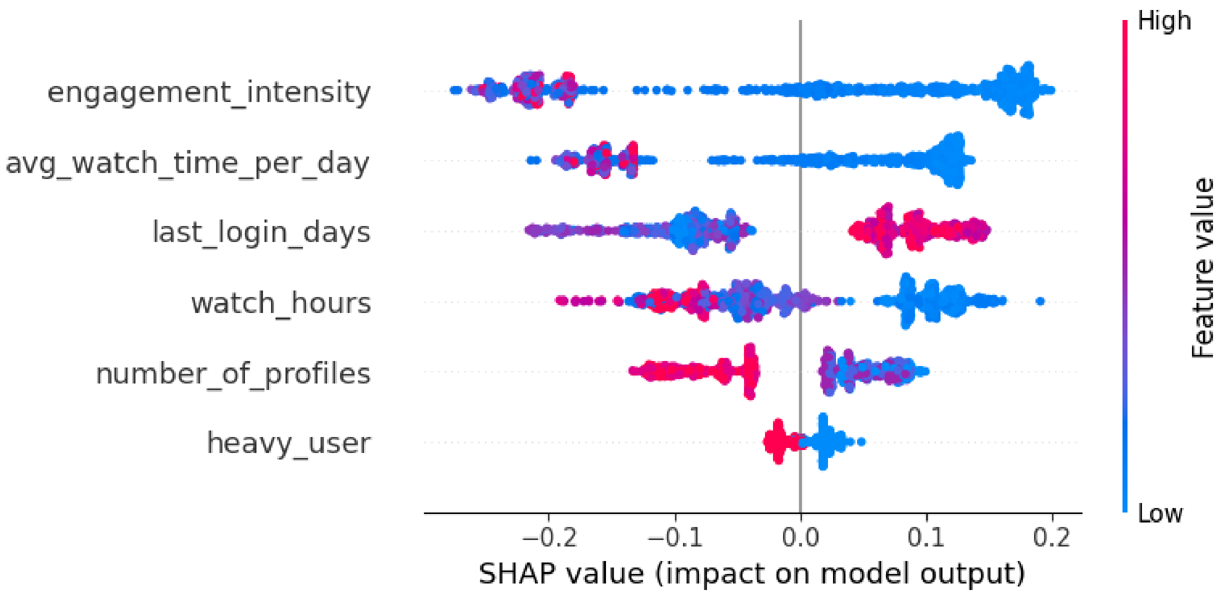
Low_engagement = is also a binary feature that will notice users who have watch_hours and last_login_days that are below 25% (the 25 percent of users who rarely use the app).

Part IV & V.

Use and experiment with appropriate models
Evaluation and comparison of models following given metrics
Understanding and explaining the model
Biases & Fairness Auditing

For this part four classification models were used: Logistic Regression, Decision Tree, and Random Forest to determine Netflix user activity. As observed amongst all four models, Random Forest is the best performing with the highest ROC-AUC.

Model	**Result**
Logistic Regression	0.94569
Decision Tree	0.96431
Random Forest	**0.97184**



The SHAP Summary Plot reveals which features have the most impact to the least impact to the model's output. In this case, engagement_intensity has the most impact, while heavy_user has the least. SHAP value explains that the higher it is the higher probability for churning, while the lower it is shows a lower probability to churn. Lastly the red colored dots represents feature value (red for high and blue for low).

For the Fairness metrics the following was observed:

- **Demographic Parity** measures the proportion of positive predictions (e.g., 'active') for each group. Ideally, this proportion should be similar across groups, meaning the model's positive prediction rate is independent of the sensitive attribute. As per my finding the predicted churn rates are relatively similar across the different age groups. This suggests that the model's prediction of churn is reasonably balanced across the age categories.

age_group	pred_churn
Youth	53.24%
Adult	56.04%
Middle-aged	50.99%
Senior	55.77%

- **Disparate Impact** assesses whether a selection rate for a protected group is substantially different from that of a reference group. It's often calculated as the ratio of the positive outcome rate for the protected group to the positive outcome rate for the reference group. A significantly different ratio above or below 1.0 (below 0.8 or above 1.25) would show potential disparate impact, in other words the model is treating other groups differently. As per result of the fairness metric the disparate impact of the female group (protected) to the male group (reference) is 1.0586, showing fair results and the groups are not treated differently.

- **Equalised Odds** is a fairness metric that requires a model to have equal True Positive Rates (TPR) and equal False Positive Rates (FPR) across different protected groups. In other words the model should correctly identify positive cases (e.g., churned customers) at the same rate, and incorrectly identify negative cases (e.g., non-churned customers as churned) at the same rate, regardless of the group. TPR also known as recall are positive cases correctly identified by the model, while FPR are negative cases incorrectly identified as positive by the model. Ideally we would want the TPR and FPR results to be close across all groups to achieve equalized odds.

gender	TPR	FPR
Male	0.9349	0.1348
Female	0.9672	0.0869
Other	0.9272	0.1582

Based on the results there are slightly noticeable differences amongst the TPR and FPR across the gender groups. This signifies the model seems to have a slightly lower rate of correctly identified churn for the Other group and a higher rate of incorrect predicted churn for this group comparing to females.

Mitigations:

Although the fairness audit did not reveal severe violations across demographic groups, the following mitigation strategies can be proposed ensuring a fair and robust churn prediction model that can be ethically deployed over time.

Firstly, a group-specific decision threshold adjustment can be applied if unequal false positive emerge across sensitive groups. Properly calibrating classification thresholds separately for different demographic segments can help achieve balanced error rates without retraining or altering core model parameters.

For addressing potential bias from unbalanced data distributions, we can use instance reweighting. Reweighting gives higher importance to the smaller disadvantaged groups, allowing the learning algorithm to better capture their behavior without oversampling or generating synthetic data. It preserves the original data distribution and reduces risk of the majority groups dominating the model learning.