Churn prediction for Amazon Prime Video

Capstone Project for Machine Learning Engineer Nanodegree

Dmitry Fadeev
Munich, Germany
https://github.com/fadeetch
https://www.linkedin.com/in/dmitryfadeeff/

April 10, 2020

Abstract

Any business is naturally interested in understanding how its customers behave. Amazon Prime Video is no exception. Once initial findings are uncovered, the next step is to predict who of the customers might churn in the near future. This project fulfills two goals: a) analyses datasets from customers in Germany and the UK in order to extract valuable insights into customer behaviour and b) builds a binary classification model that predicts which customers are inactive (churned) within 6 months.

Keywords: Binary classification · Churn prediction · Ensemble

1. Problem Statement

One of the Amazon Prime Video businesses is rent-or-buy business - transactional video on demand (TVOD). It means customers come to the storefront, select a title they like and buy or rent it individually. In contrast, SVOD (subscription video on demand) business model works exactly like the one of Netflix: a customer pays a monthly / annual subscription and watches a selection of movies available for this subscription. The key selling point for TVOD is its vast selection, which means it offers a far richer set of titles than SVOD does.

The problem is - as in every business - that some customers might churn. Economically, it is common in e-commerce that the cost of acquisition is far higher than the cost of retention. Which means - it is an important business problem to prevent churn. In more technical terms, it is important for decision makers to understand which features signal a customer becoming churn and identify those customers. Given these insights, marketing can build campaigns to target those customers who are expected to churn in order to prevent their inactivity thus keeping engagement at sustainable levels.

In essence , this is a classification problem where the model takes a set of features (X) as input and produces the expected binary output (Y). An input X contains the following features:

- Numerical:
    - Number of transactions during customer's lifetime and in the last year
    - Average sales price (asp) for transactions that a customer made in full lifetime and last year
    - Average revenue per user on monthly (arpu monthly) basis during lifetime
    - Frequency of transactions in customer's lifetime
    - Tenure of the customer (what is the time since the first transaction)
- Categorical
    - Has customer predominantly made transactions during promotions or bought at full price
    - What is customer`s preference for content age (catalog vs. new releases)
    - What is customer`s preference for content type (buying or renting, movies vs TV)
    - Given customer`s performance in the last year, is she engaged or moderate consumer (reference below to the engagement class)
    - For each engagement class for the last year, how was customer's behaviour in the previous lifetime - before last twelve months (referred to as hidden class)

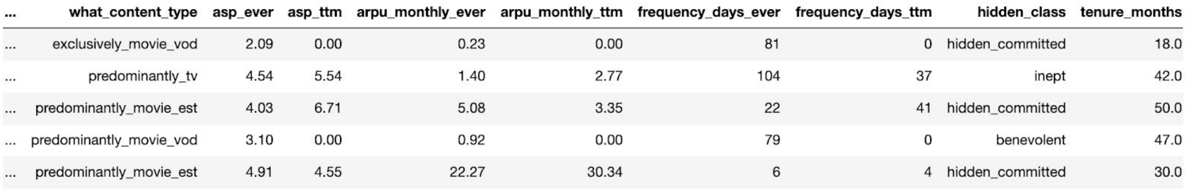
Output variable Y is a binary value signaling one of the two states: 0 if a customer was inactive in the last 6 months, implying that she did not make any transactions, or 1 if a customer made at least 1 transaction in the last 6 months.

2. Datasets and Inputs

Datasets were obtained from the Amazon Data Warehouse (DWH) and contain anonymized summary statistics on around 500k customers in each DE and the UK. The files are the following:

- DE_subset.csv
- UK_subset.csv

Each file has 22 columns and around 500k rows. One row represents summary statistics on a customer level. Target variable is also provided and identifies two states: state 0 (inactive) and state 1 (active).
AS mentioned above, the dataset has both numerical and categorical variables. Sample of the data is shown below:



...	what_content_type	asp_ever	asp_ttm	arpu_monthly_ever	arpu_monthly_ttm	frequency_days_ever	frequency_days_ttm	hidden_class	tenure_months
...	exclusively_movie_vod	2.09	0.00	0.23	0.00	81	0	hidden_committed	18.0
...	predominantly_tv	4.54	5.54	1.40	2.77	104	37	inept	42.0
...	predominantly_movie_est	4.03	6.71	5.08	3.35	22	41	hidden_committed	50.0
...	predominantly_movie_vod	3.10	0.00	0.92	0.00	79	0	benevolent	47.0
...	predominantly_movie_est	4.91	4.55	22.27	30.34	6	4	hidden_committed	30.0

Distribution of output values reveals no real class imbalance: in Germany, 60% are in state 1, while in the UK - around 48%, which is in line with a general finding that on average German customers are more engaged consumers of rent-or-buy offers than British ones.

Another important characteristic is the distribution of customer count by engagement classes (categorical variable that scores how many transactions were made in the last year): in Germany, 51% are moderate, 30% are one-time and the rest are engaged, while in the UK 44% are moderate and 39% are one-time with rest being engaged.

Moreover, pairplots did not exhibit large numbers of outliers which might have skewed results. Descriptive statistics on the set shows that there is a strong difference in specific features split by output states which implies that these variables might be useful in separating the classes. For example, in DE, the average lifetime number of transactions for customers in state 1 is 31 and in state 0 just 7, while in the UK the numbers are 22 and 5, respectively. Also, in Germany average lifetime monthly revenue is 4.8 EUR for state 1 and 0.9 EUR for state 0, while in the UK it is 3.6 GBP and 0.62 GBP. On other hand, lifetime average sales price is quite close for both states: in Germany 5.0 EUR for state 0 and 4.88 EUR for state 1, while in the UK 4.27 GBP and 4.31 GBP. These differences or similarities in means for specific features in both classes serve as a starting point for identifying features that have the most predictive power for this classification problem.

3. Solution Statement

Solution is represented by a classification model, which predicts whether a customer is in state 0 (inactive) or state 1 (active).

4. Benchmark Model

The field “Target” is contained in each of the datasets. The model does the classic train / test split. Therefore, it is possible to see what the overall accuracy of the model is. Also, simple logistic regression serves as a model with benchmark performance. More robust algorithm - random forest classifier scores against benchmark model.

5. Evaluation Metrics

Evaluation of the model is summarized in the ROC curve and area under the curve metric (AUC). Both benchmark and more robust models are evaluated based on AUC. Also, classification report and confusion matrix provides additional details for both models.

6. Project Design

a. Exploratory Data Analysis

This step builds distributions by specific values to get a feeling for data at hand. In particular, a researcher might be interested in understanding what is the average monetary spend of a customer depending on an engagement class. Engagement class is part of a customer segmentation model which is beyond the current project. However, in a nutshell, segmentation model scores customer behaviour in the *last twelve months* and depending on the number of transactions a customer made, assigns various labels: one-time, moderate or engaged. Also, the dataset contains other categorical variables such as content type or content age preference which might play a role in the subsequent classification model.

b. Preprocessing

i. Identify missing values

This step checks missing values in a dataframe and encodes missing values with zeros.

ii. Avoid multicollinearity

This step builds a correlation matrix and aims at identifying features with relatively high correlation. In order to avoid multicollinearity, some features are removed.

iii. Build pairplots for the feature set

This step explores how data is distributed for each quantitative feature.

iv. One-hot encoding

This step transforms categorical values into so-called one-hot-encodings so that the data can be fed into the classification algorithm.

c. Modeling

i. Split into train and test sets

This step applies functions to split the dataset into train and test subsets. The model uses 33% of the data for the test size.

ii. Fit algorithms for training sets

This step uses a library to train the algorithms. In case of convergence problems, explicit maximum number of iterations should fix the issue.

iii. Score for test sets

This step makes predictions on the test set and creates an object that contains predicted binary values which are then used for the evaluation step.

iv. Check evaluation metrics for the models

This step performs evaluation, prints accuracy, confusion matrix and ROC curve. In this step a researcher can also compare the benchmark model and other models and recommend the one with higher AUC as the final solution to the given problem.

v. Feature importance

This step seeks to make the model interpretable and figure out what is the subset of features that have the most significant outcome.

A potential next step might be to test other classification models on a set of features that exhibit the largest predictive power and compare their performance with given models. However, on a given dataset, in both UK and DE, the results with random forest classifier already provide AUC of 95%, therefore, the project did not consider any other models yet.

References:

F .Kruber, J. Wurst, “Unsupervised and Supervised Learning with the Random Forest Algorithm for Traffic Scenario Clustering and Classification”; URL https://arxiv.org/pdf/2004.02126.pdf

T. Alasalmi, J. Suutala, “Beer Classifier Calibration for Small Data Sets”, URL https://arxiv.org/pdf/2002.10199.pdf

E. Admasu, A. Teklay, “Student Performance Prediction with Optimum Multilabel Ensemble Model”; URL https://arxiv.org/pdf/1909.07444.pdf