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 15, 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. Definition

a. Project Overview

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 rent-or-buy business is customer-facing one. In other words, it generates revenue from sales of digital video content to individual customers. The key factor of such a business is retention. Put simply, what percentage of your existing customers stay active (make at least 1 transaction) during the next X months. Also, the complement of retention is churn. Therefore, to stay sustainable, a business has to minimize churn.

Why is it important to keep churn low? Economically, it is common in e-commerce that the cost of acquisition is far higher than the cost of retention. Which means - it is one of the key business problems 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.

b. **Problem Statement**

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.

c. **Metrics**

The standard metric for binary classification problems is the ROC curve and the AUC, respectively. ROC (receiver operating characteristic) curve plots True positive rate (also known as probability of detection, or sensitivity, or recall) against False positive rate (also known as probability of false alarm, and calculated as 1 - specificity) at various thresholds. In simple terms, the ROC curve allows to create a snapshot of the confusion matrix at various thresholds, which results in a visual generalization of the classification model performance. Normally, AUC (area under the curve) is compared with the horizontal line shown in red in the figure 1 below which has an area of 0.5. Generally, the goal is to maximize AUC and raise it as close to 1.0 as possible.

Also, sometimes researchers use precision instead of false positive rate since precision does not incorporate true negative and this metric might be better suited for model performance analysis in case of imbalanced classes. Since the current model does not have this issue (explained in the Analysis section), the ROC sticks with false positive rate.

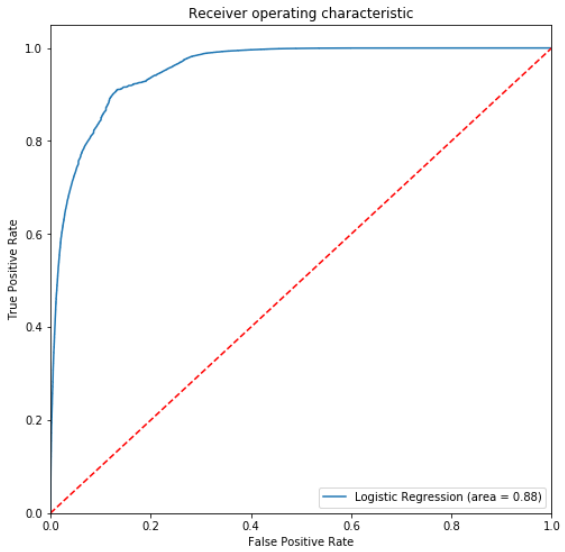

Figure 1: Example of the ROC curve

2. Analysis

a. Data Exploration

Datasets were obtained from the Amazon Data Warehouse (DWH) and contain anonymized summary statistics on around 500k randomly selected 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. Put differently, each row is one unique customer and it provides information on the following:

- ❏ How many units did a customer made in the lifetime and in the last twelve months
- ❏ How much revenue a customer generated in lifetime and in the last twelve months
- ❏ What kind of preferences a customer has for promotional campaigns
- ❏ What does a customer like in terms of content age (catalog vs new releases)
- ❏ What does a customer like in terms of content type (movie vs TV)
- ❏ What is average sales price at which customer made transactions: lifetime and in the last twelve months
- ❏ What is the average monthly revenue that a customer generated: lifetime and in the last twelve months
- ❏ What is a customer`s frequency of transactions: lifetime and in the last 12 months
- ❏ What is a customer`s hidden class (engagement before last 12 months)

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

b. Exploratory visualization

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.

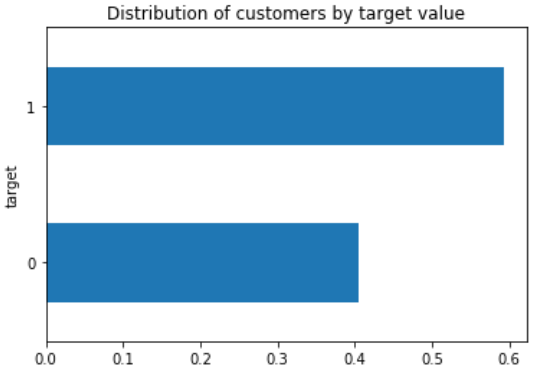

Figure 2: Distribution of customers by target in Germany

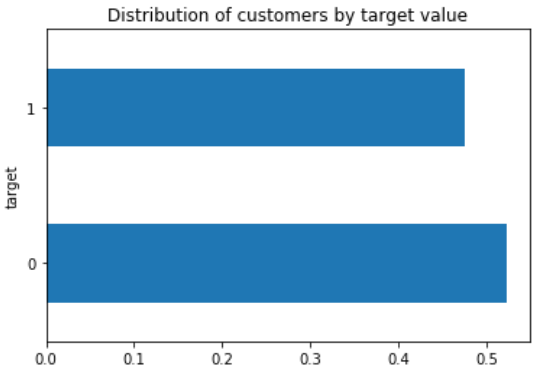

Figure 3: Distribution of customers by target in the UK

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. 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.

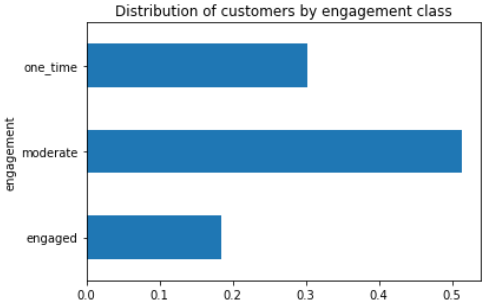

Figure 4: Distribution by engagement class in Germany

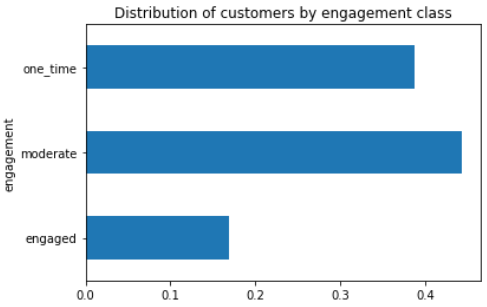

Figure 5: Distribution by engagement class in the UK

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.

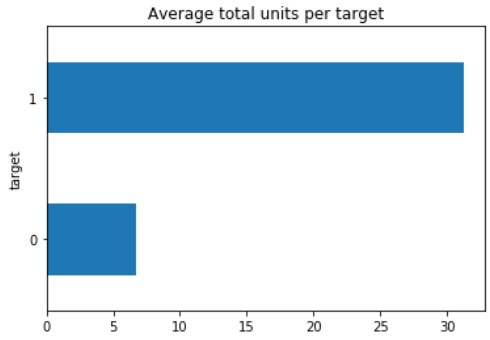

Figure 6: Average number of lifetime units in Germany

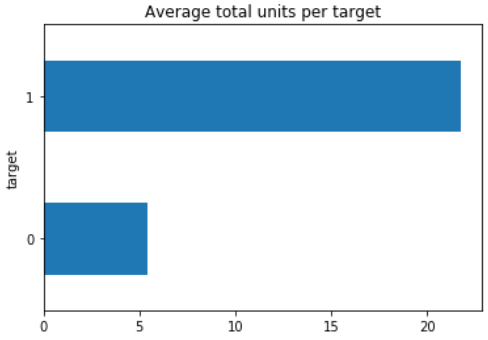

Figure 7: Average number of lifetime units in the UK

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.

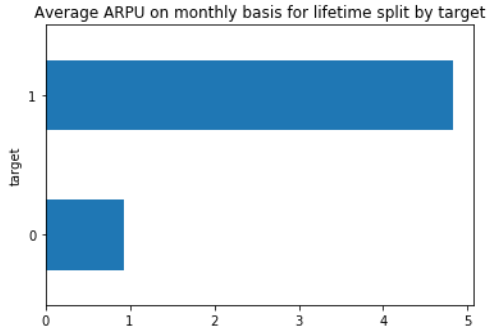

Figure 8: Average monthly revenue in Germany

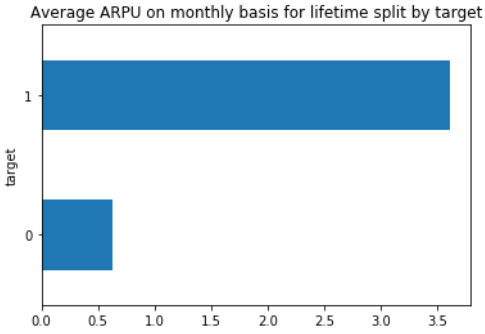

Figure 9: Average monthly revenue in the UK

On the other hand, some of the metrics are quite close for both states. For example, lifetime average sales price is 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.

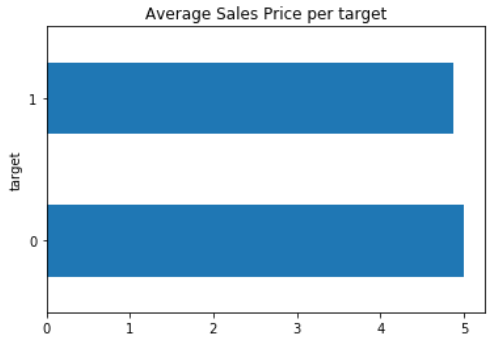

Figure 10: Average lifetime sales price in Germany

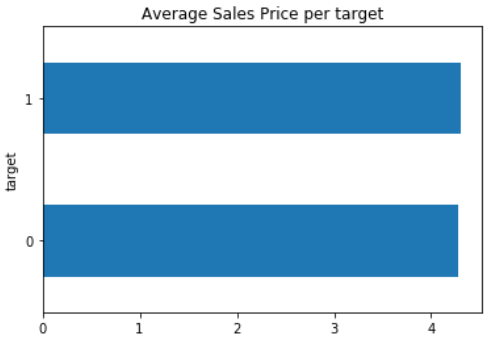

Figure 11: Average lifetime sales price in the UK

Moreover, tenure in months (time since first transaction) does not seem to differ too much between the states. In Germany, average tenure for state 0 is 34 months, while for state 1 it is 30 months. For the UK, tenure for state 0 is, on average, 36 months and for state 1: 26 months.

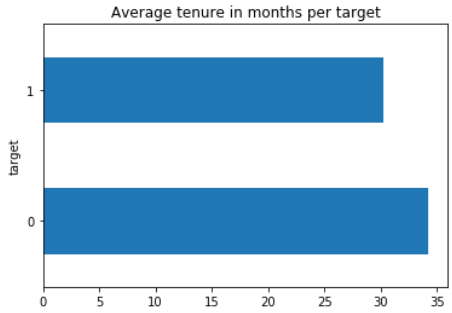

Figure 12: Average tenure in Germany

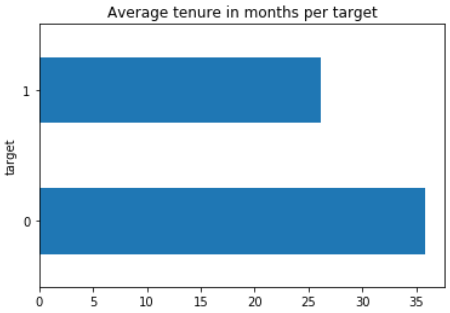

Figure 13: Average tenure in the UK

What is also interesting is the fact that the timeframe for statistics plays a role. Average sales price for lifetime does not seem to differ a lot between the two states. However, if the average sales price is calculated only for the last 12 months (shortened to TTM), the difference is much clearer. In Germany, the average sales price for state 1 is 4.96 EUR, while for state 0 the metric is at 1.79 EUR. Similarly, in the UK the numbers are 4.36 GBP for state 1 and 1.06 GBP for state 0.

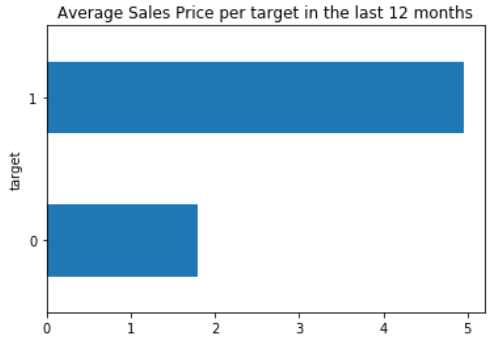

Figure 14: Average TTM sales price in Germany

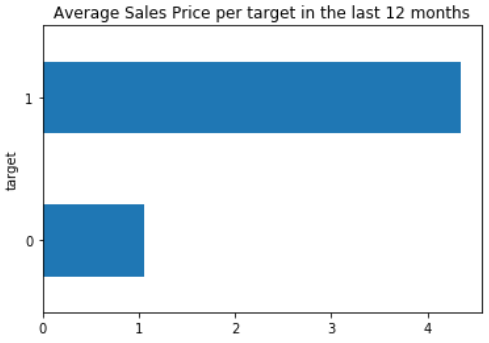

Figure 15: Average TTM sales price in the UK

Lastly, in order to have an even better grasp of the data, it is advisable to check multiple layers. For example, what are the average lifetime units for various categorical variables ? In particular, is there a difference in lifetime units when breaking down customer base by the kind of promotion they participate in.

In Germany, it shows that customers who are actively participating in promotions, in *each state transact on average more than non-promotional users*. PMD promotions mean Prime Member Deals, ie promotions for which only those customers are eligible who also possess Prime subscription. Vendisto promotions are just studio specific promotions and are applicable to all customers irrespective of the fact whether they have or have not a Prime subscription.

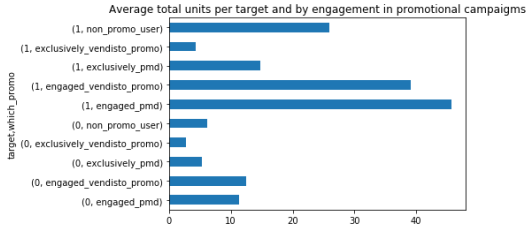

Figure 16: Preference for promotions in Germany

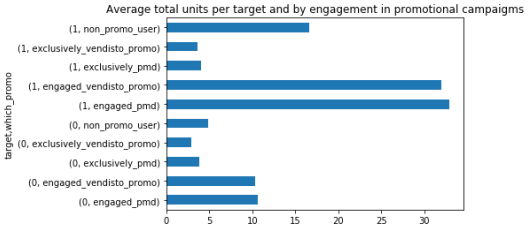

Figure 17: Preference for promotions in the UK

SImilarly, a multilayer plot with average units can be built for other categorical variables such as preference for content age or content type and hidden class. In both countries, on average, those customers who prefer catalogue, make more transactions.

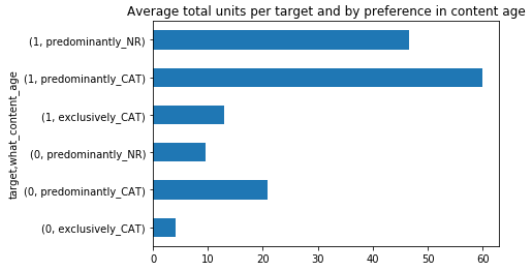

Figure 18: Preference for content age in Germany

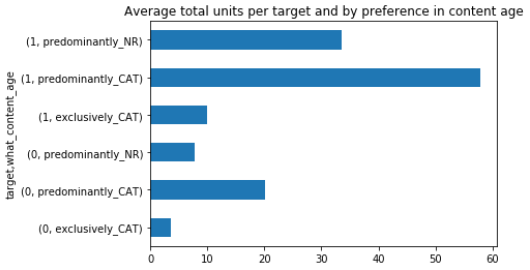

Figure 19: Preference for content age in the UK

Also, both countries exhibit the same pattern with regard to preferences for content type: those who rent (video on demand, VOD) on average transact more than those who buy into their libraries (electronic sales through, EST).

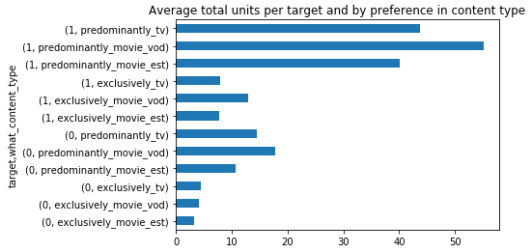

Figure 20: Preference for content type in Germany

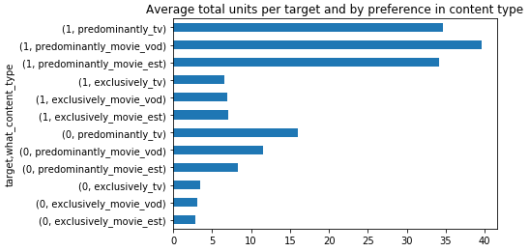

Figure 21: Preference for content type in the UK

Finally, data from both countries imply that those customers who showed strong buying patterns in the period before last 12 months (hidden committed), are transacting more than the other classes.

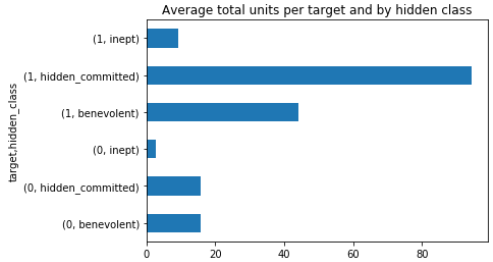

Figure 22: Hidden class in Germany

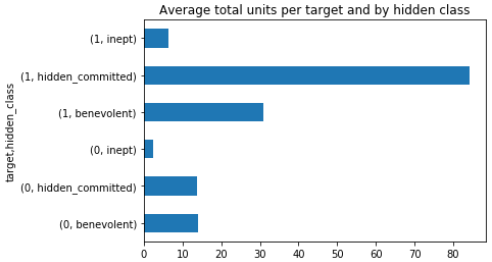

Figure 23: Hidden class in the UK

c. Algorithms and Techniques

Since the churn prediction is basically a binary classification problem and the dataset has labels for the target variable - it is a classic example of the supervised learning. Therefore, the current paper attempts to implement standard classification algorithms such as logistic regression and random forest classifier.

d. Benchmark

One of the foundational and most used models for classification problems is logistic regression. This paper considers logistic regression the benchmark against which more complex algorithms, namely random forest classifier will be compared. The benchmark metric is the AUC and since both logistic regression and random forest classifier allow for creation of ROC curve, it will be easy to compare which model has a higher area under the curve (AUC). The one with a higher AUC wins.

3. Methodology

a. Data preprocessing

The first step in data preprocessing is to check data types of columns, especially timestamps might be having object data type and have to be casted to timestamps if they are a basis for new fields.

```
encrypted_customerID              object
total_units                        int64
total_revenue                    float64
total_units_ttm                    int64
total_revenue_ttm                float64
adj_units_yearly_ever              int64
adj_revenue_yearlu_ever          float64
adj_units_yearly_before_ttm        int64
adj_revenue_yearly_before_ttm    float64
which_promo                       object
engagement                        object
what_content_age                  object
what_content_type                 object
asp_ever                         float64
asp_ttm                          float64
arpu_monthly_ever                float64
arpu_monthly_ttm                 float64
frequency_days_ever                int64
frequency_days_ttm                 int64
hidden_class                      object
tenure_months                    float64
target                             int64
dtype: object
```

Figure 24: Data types in the dataset

Since no such problems arise with timestamps in the dataset, next step is to check missing values using command

```
df.apply(lambda x: sum(x.isnull()), axis=0)
```

There are missing values in a field arpu_monthly_ever; this issue is remedied by filling missing values with zeros.

Next step proceeds to feature engineering. The goal is to eliminate or at least minimize multicollinearity. In order to figure out which features have high correlation, a correlation matrix is built (see figure below). After some features are dropped, the pairplot is created to double check abnormalities. Since no large skews are identified, the final step is to process categorical variables using one-hot-encoding. It implies that categorical variables are transformed into numerical values using binary dummy variables. This results in extension of the number of dataset columns but the step is absolutely necessary in order for the algorithm to be able to consume the data.

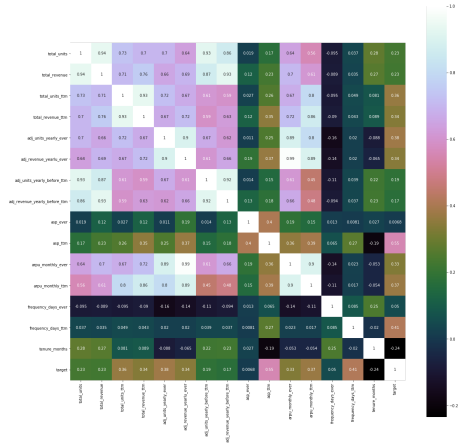

Figure 25: Correlation matrix in the UK

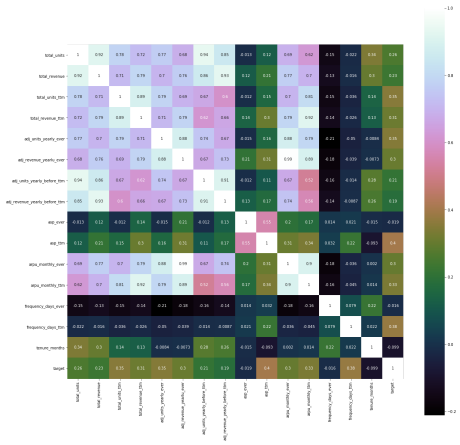

Figure 26: Correlation matrix in Germany

b. **Implementation**

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 scikit learn 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 we can also compare the benchmark model and other models and recommend the one with higher AUC as the final solution to the given problem.

For recap, we started off with a logistic regression which produced the following default confusion matrices for both countries:

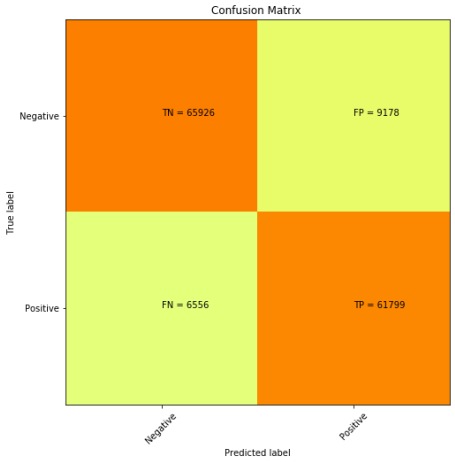

Figure 27: Confusion matrix for log reg in the UK

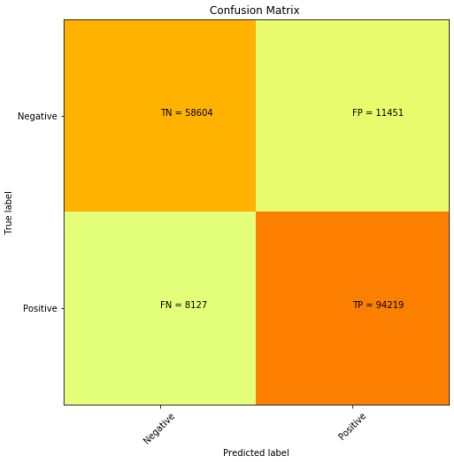

Figure 28: Confusion matrix for log reg in Germany

Correspondingly, the ROC curves are already achieving almost 90% AUC in both countries:

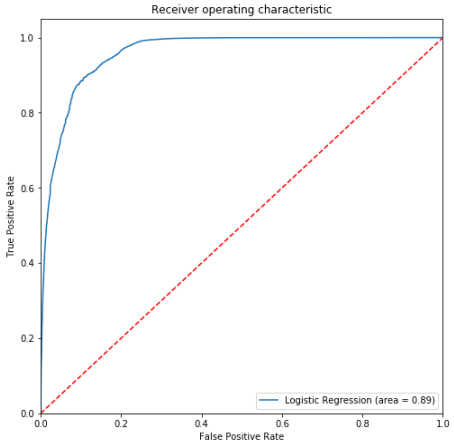

Figure 29: ROC curve for log regression in the UK

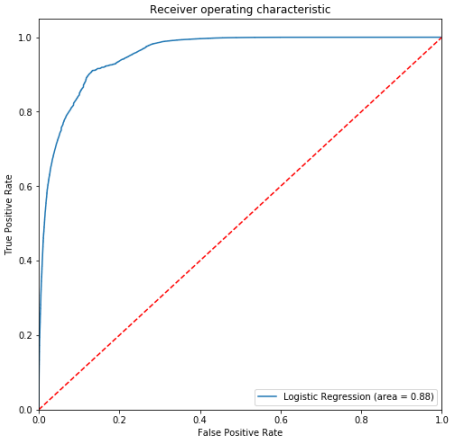

Figure 30: ROC curve for log regression in Germany

Logistic regression served as a benchmark model. Next, we are running a random forest classifier (rfc) to check whether we can improve overall accuracy and area under the curve. Random forest classifier delivers the following confusion matrix and ROC curve.

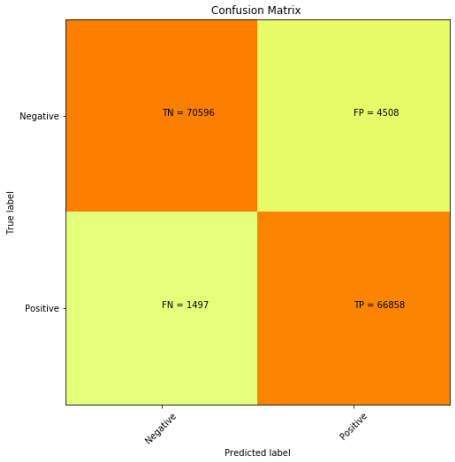

Figure 31: Confusion matrix for rfc in the UK

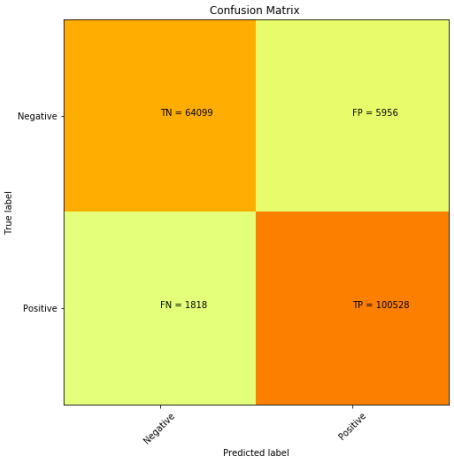

Figure 32: Confusion matrix for rfc in Germany

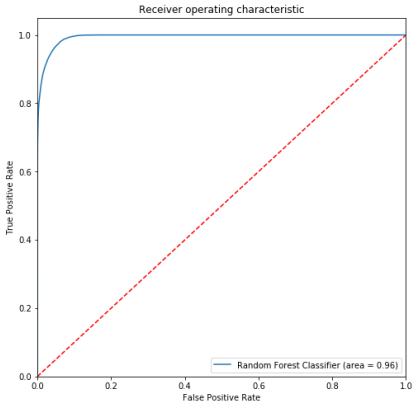

Figure 33: ROC curve for rfc in the UK

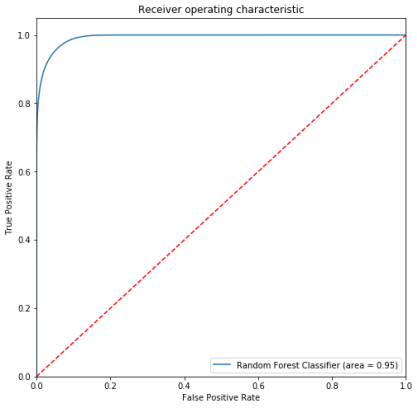

Figure 34: ROC curve for rfc in Germany

What we see is that AUC achieved 0.96 on the test set in the UK and 0.95 in Germany, implying that with the new model - random forest classifier - we raised the AUC from around 90% to 95%. However, in the next step we can check a subset of features and potentially slightly improve model performance.

c. Refinement

At this stage we are trying to see whether a subset of features can do an even better job. We selected a random forest classifier since it is considered a powerful model for binary classification which also provides some insight into feature importance. This latter aspect is extremely important for business users since they are always interested in understanding the model and why it works. In particular, they are curious about a subset of features that might have the most predictive power.

We build feature importance graphs for both countries. Basic idea behind this exercise is to check by how much overall model performance would decrease if we removed some of the features.

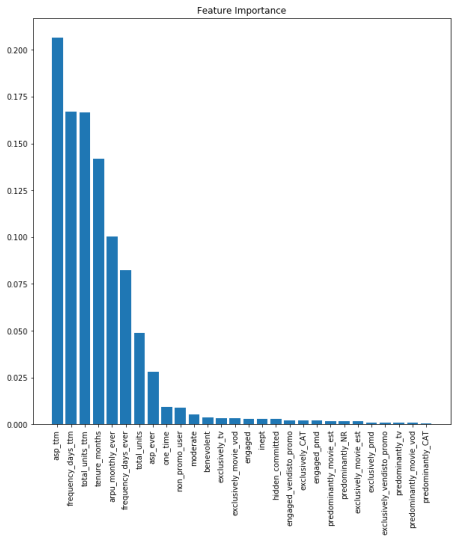

Figure 35: Feature importance in the UK

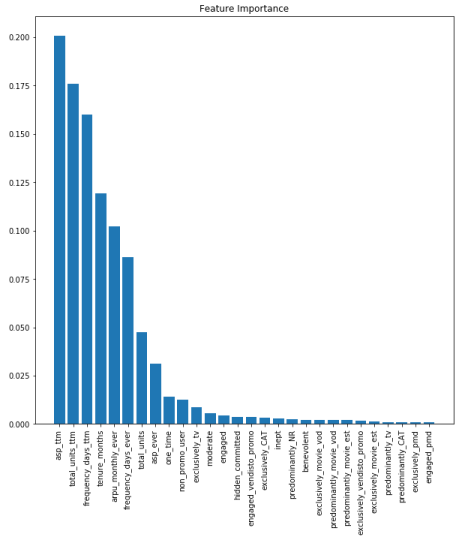

Figure 36: Feature importance in Germany

Next, we make a subset of only the most relevant features: ['asp_ttm','frequency_days_ttm','total_units_ttm', 'tenure_months','arpu_monthly_ever','frequency_days_ever','total_units','asp_ever','target']. And re-run the random forest classifier. The final result is summarized in the following ROC curve

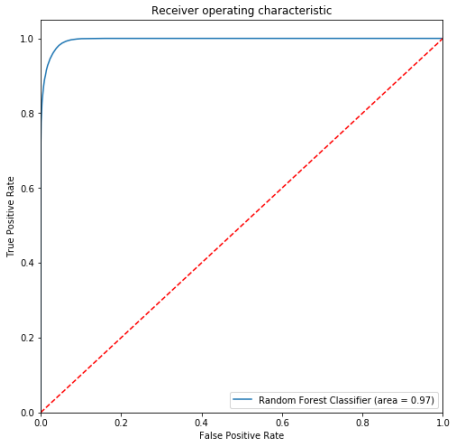

Figure 37: ROC curve for subset rfc in the UK

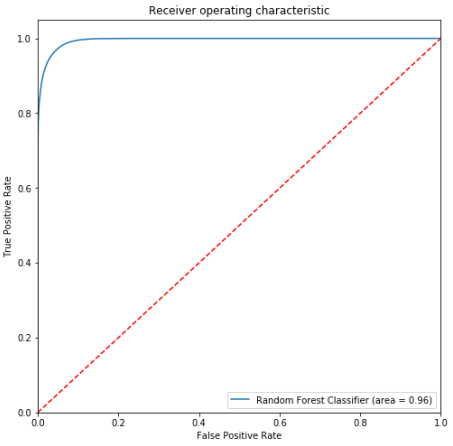

Figure 38: ROC curve for subset rfc in Germany

4. Results

a. Model evaluation and validation

So far, we have checked the benchmark logistic regression model, default random classifier model and a random classifier with the most important features. The latter serves as the final model achieving AUC of 97% in the UK and 96% in Germany. Here is the classification report from the final model.

Overall accuracy: 0.9667709938031075

	precision	recall	f1-score	support
0	0.99	0.95	0.97	75104
1	0.95	0.98	0.97	68355
accuracy			0.97	143459
macro avg	0.97	0.97	0.97	143459
weighted avg	0.97	0.97	0.97	143459

Figure 39: Classification report in the UK

Overall accuracy: 0.9634166855180655

	precision	recall	f1-score	support
0	0.98	0.93	0.95	70055
1	0.95	0.99	0.97	102346
accuracy			0.96	172401
macro avg	0.97	0.96	0.96	172401
weighted avg	0.96	0.96	0.96	172401

Figure 40: Classification report in Germany

b. **Justification**

Evaluation of the model was 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. Classification report for the final model was presented in the paragraph above, while the classification report for the benchmark is shown below.

Overall accuracy: 0.8903240647153542

	precision	recall	f1-score	support
0	0.91	0.88	0.89	75104
1	0.87	0.90	0.89	68355
accuracy			0.89	143459
macro avg	0.89	0.89	0.89	143459
weighted avg	0.89	0.89	0.89	143459

Figure 41: Classification report in the UK

Overall accuracy: 0.886439173786695

	precision	recall	f1-score	support
0	0.88	0.84	0.86	70055
1	0.89	0.92	0.91	102346
accuracy			0.89	172401
macro avg	0.88	0.88	0.88	172401
weighted avg	0.89	0.89	0.89	172401

Figure 42: Classification report in Germany

Comparison of the two - benchmark and the final model clearly shows the superiority of the final model across all of the metrics: accuracy, precision, recall. Analysis performed in the paper shows that the binary classification model can solve the underlying problem of churn prediction quite well, also it provides some form of interpretation as to which features play the most important role giving the end users some information on which metrics they should pay attention to in order to minimize churn.

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