# Santandar Product Recommendation Competition

*by Sarah Huang*

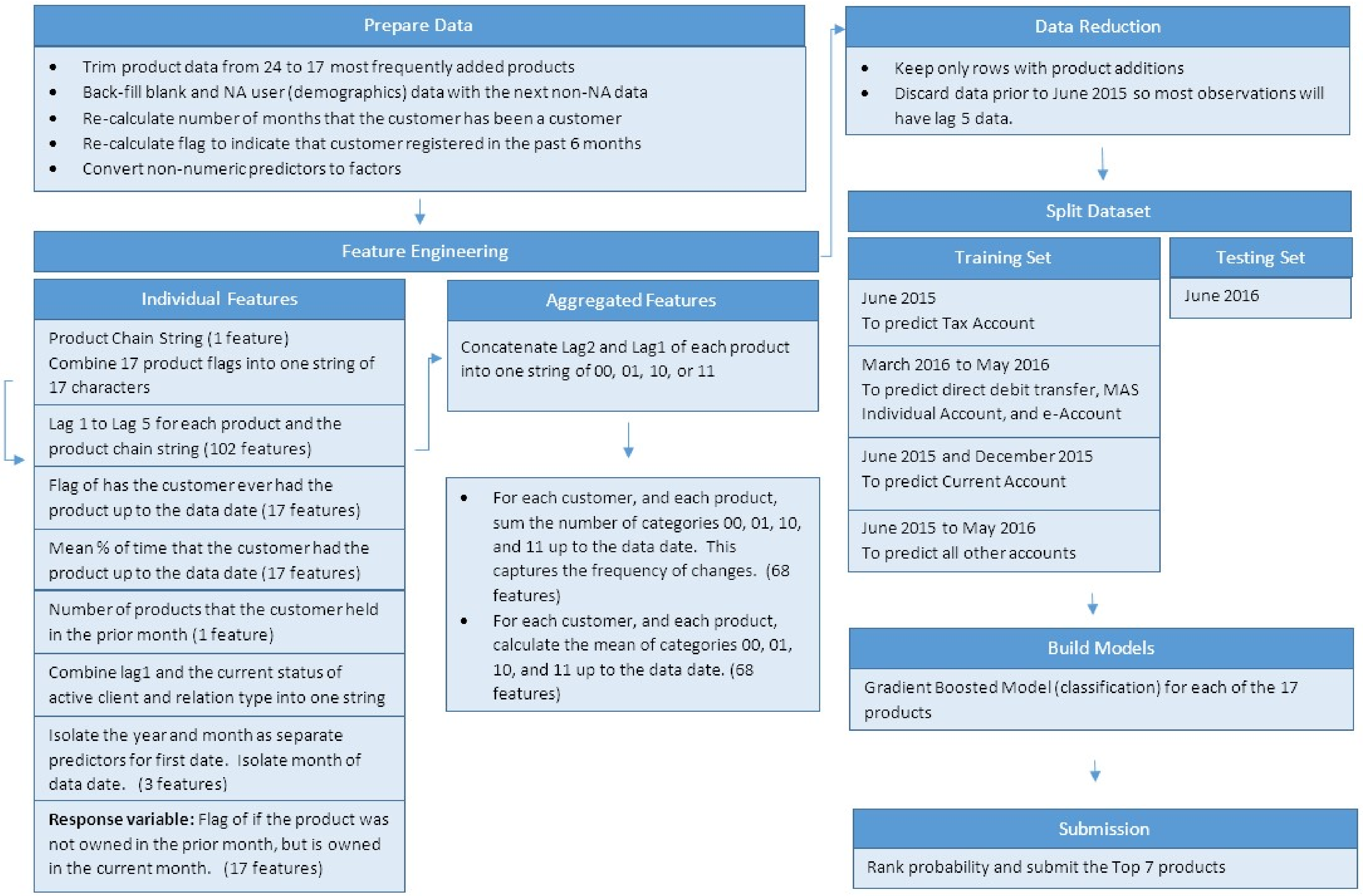
## Overview

The Santandar Product Recommendation Competition is hosted by Kaggle. The goal of the challenge is to predict which of the 24 products Santandar's existing customers will likely to add in the next month based on their past behavior and that of similar customers. The training data consists of approximately 13.6 million of rows of data, with a timeframe spanning across from January 2015 to May 2016. The user data consists of 24 predictors including demographics data such as age, province of residence, and sex. The product data consists of flags to indicate ownership for the 24 products for the respective months. The objective is to recommend the top 7 products that the customer is likely to add to each of the nearly 1 million customers in the test set. The scoring matrices is based on mean average precision at 7 (MAP@7 criterion). Higher score is rewarded if actual outcome is predicted earlier in the list of recommendations. Since new products are added for about 3.5% of customers, the perfect score would be close to 0.035. The winner of the competition scored 0.031409. The model approach and the script provided scored 0.0307548.

My approach to solve this problem is to predict the probability of a product being added using separate gradient boosted model for each product. The top 7 products are chosen based on the highest 7 probabilities. Products are being modeled for different time periods based on its traits. The predictive model is run on 17 of the 24 products to save time and to cut down on the training set size. The products that are rarely added are skipped. The exploratory analysis section will detail the reasoning behind this approach. My code is posted on Github.

**Modeling Strategy**

The flow chart below outlines the modeling strategy:



**Prepare Data**

- Trim product data from 24 to 17 most frequently added products
- Back-fill blank and NA user (demographics) data with the next non-NA data
- Re-calculate number of months that the customer has been a customer
- Re-calculate flag to indicate that customer registered in the past 6 months
- Convert non-numeric predictors to factors

**Feature Engineering**

**Individual Features**

Product Chain String (1 feature)
Combine 17 product flags into one string of 17 characters

Lag 1 to Lag 5 for each product and the product chain string (102 features)

Flag of has the customer ever had the product up to the data date (17 features)

Mean % of time that the customer had the product up to the data date (17 features)

Number of products that the customer held in the prior month (1 feature)

Combine lag1 and the current status of active client and relation type into one string

Isolate the year and month as separate predictors for first date. Isolate month of data date. (3 features)

**Response variable:** Flag of if the product was not owned in the prior month, but is owned in the current month. (17 features)

**Aggregated Features**

Concatenate Lag2 and Lag1 of each product into one string of 00, 01, 10, or 11

- For each customer, and each product, sum the number of categories 00, 01, 10, and 11 up to the data date. This captures the frequency of changes. (68 features)
- For each customer, and each product, calculate the mean of categories 00, 01, 10, and 11 up to the data date. (68 features)

**Data Reduction**

- Keep only rows with product additions
- Discard data prior to June 2015 so most observations will have lag 5 data.

**Split Dataset**

**Training Set**

June 2015
To predict Tax Account

March 2016 to May 2016
To predict direct debit transfer, MAS Individual Account, and e-Account

June 2015 and December 2015
To predict Current Account

June 2015 to May 2016
To predict all other accounts

**Testing Set**

June 2016

**Build Models**

Gradient Boosted Model (classification) for each of the 17 products

**Submission**

Rank probability and submit the Top 7 products

The challenges in this contest are to capture the historical behavior, seasonality, as well as to deal with the size of the dataset on a laptop. To reduce the size of the dataset, only observations that corresponds with product additions are included in the modeling process. As part of feature engineering, lag predictors going back 5 months are created. Data without 5 months of lag data are excluded, therefore only data from June 2015 to May 2016 remains. In addition, only 17 of the 24 products are kept for analysis since 7 of the products are extremely rare. This data reduction strategy effectively reduces the data from approximately 13.6 million rows of training data to 426k rows of training data. The test set for submission has about 1 million customers.

The following features are created to capture product patterns and historical behavior:

- a predictor that concatenates 17 product flags into one string (product chain) (1 feature)
- lag 1 to lag 5 for each of the 17 products and the product chain (102 features)
- flag of whether the customer has ever owned the product up to the data month for each product. (17 features)

  Ex: for April 2016 data, the flag will be "1" if the customer has ever held the product from January 2015 until March 2016.

- mean (% of time) that the customer owned the product from January 2015 up to the data month. (17 features)

  Ex: a customer who has used credit card for 5 of the months since January 2015 to August 2015 will have a mean of 0.625 for the September 2015 data date.

- number of products that the customer held in the prior month (1 feature)
- capture change in relation type and active client from prior month by concatenating current and lag1 status into one string (2 features)
- capture frequency of historical product behavior by concatenating lag2 and lag1 product flag, and then for each product up to the data date, sum the number of times each pattern occurs. The categories are from 0 to 1, from 0 to 0, from 1 to 0, and from 1 to 1. (68 features)

  01 means the customer added the product last month, 00 means the customer did not own and did not add the product last month, 10 means the product was deleted last month, 11 means that the customer owns the product for both last month and the month before.

  The idea of this feature is from a forum discussion in the Kaggle site.

- for each product up to the data date, calculate the mean (% of time) that the product pattern changes from 0 to 1, from 0 to 0, from 1 to 0, and from 1 to 1. (68 features)
  Ex: a customer who has a debit transfer every other month will have a mean of 0.5 for the categories "01" and "10", and have mean of 0 for "11" and "00".

To capture seasonality, date variables are isolated into Year and Month. The data date is isolated into data month. The inception date of the customer account is isolated into inception year and inception month.

Some products exhibit seasonality and trend, which is discussed in detail in the exploratory analysis section. Due to time-varying characteristics of some products, different products are trained using different periods of data to match the pattern in the testing set (June 2016). Specifically, tax is predicted with June 2015 data, MAS individual account and e-accounts are predicted with most 3 recent months (March to May 2016), current account is predicted using June 2015 and December 2015 data, while the remaining products are predicted using June 2015 to May 2016 data. January 2015 to May 2015 data are discarded since those data will not have complete lag1 to lag5 information.

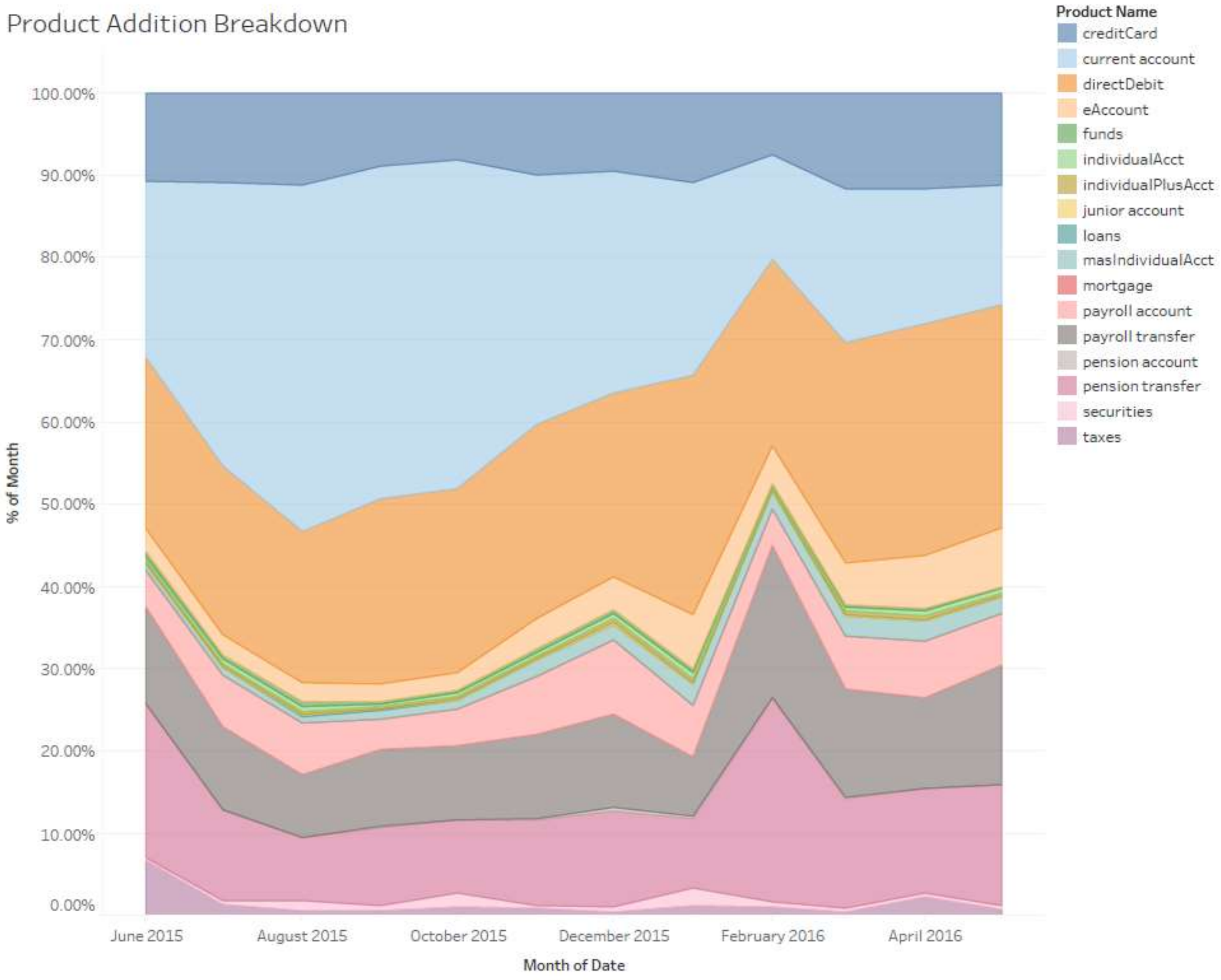
Upon completion of data cleaning and transformations, there are 321 features in total.

A gradient boosted model is run on each of the 17 products separately. The goal of each model is to predict the probability of the customer adding the product in June 2016. The R interface of the H2O platform is used to run the model. The benefit of H2O is that the core code is written in Java and the web UI provides convenient analytics tools. The probability of the 17 products are then added to a matrix and then ranked. The top 7 products are recommended for each customer.

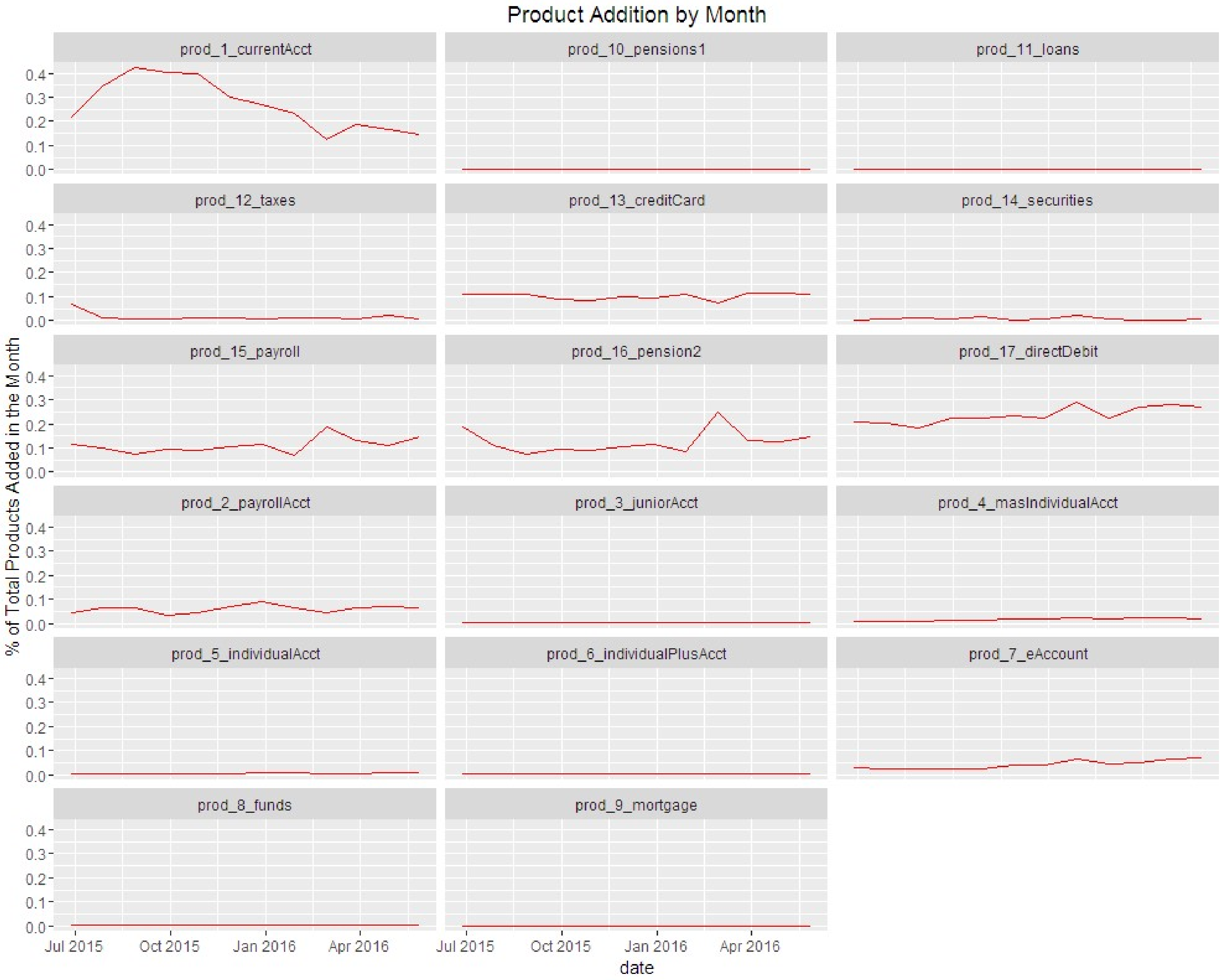
Modeling was also done on a multinomial approach with similar results. However, it was more helpful to review the variable importance of separate models rather than just one model, therefore the multinomial approach was abandoned.

## Exploratory Analysis

The chart below provides an overview of product addition distribution across months.  From the chart, one can see that some products remain relatively flat across months, some increased in popularity, some had spikes in certain months, and some moves in correlation with other products.  The bigger area means product makes up a bigger percentage of the products that are being added.  Current account, direct debit, credit card, payroll transfer, and pension transfer make up a big bulk of the additions, while many of the other products are rarely added

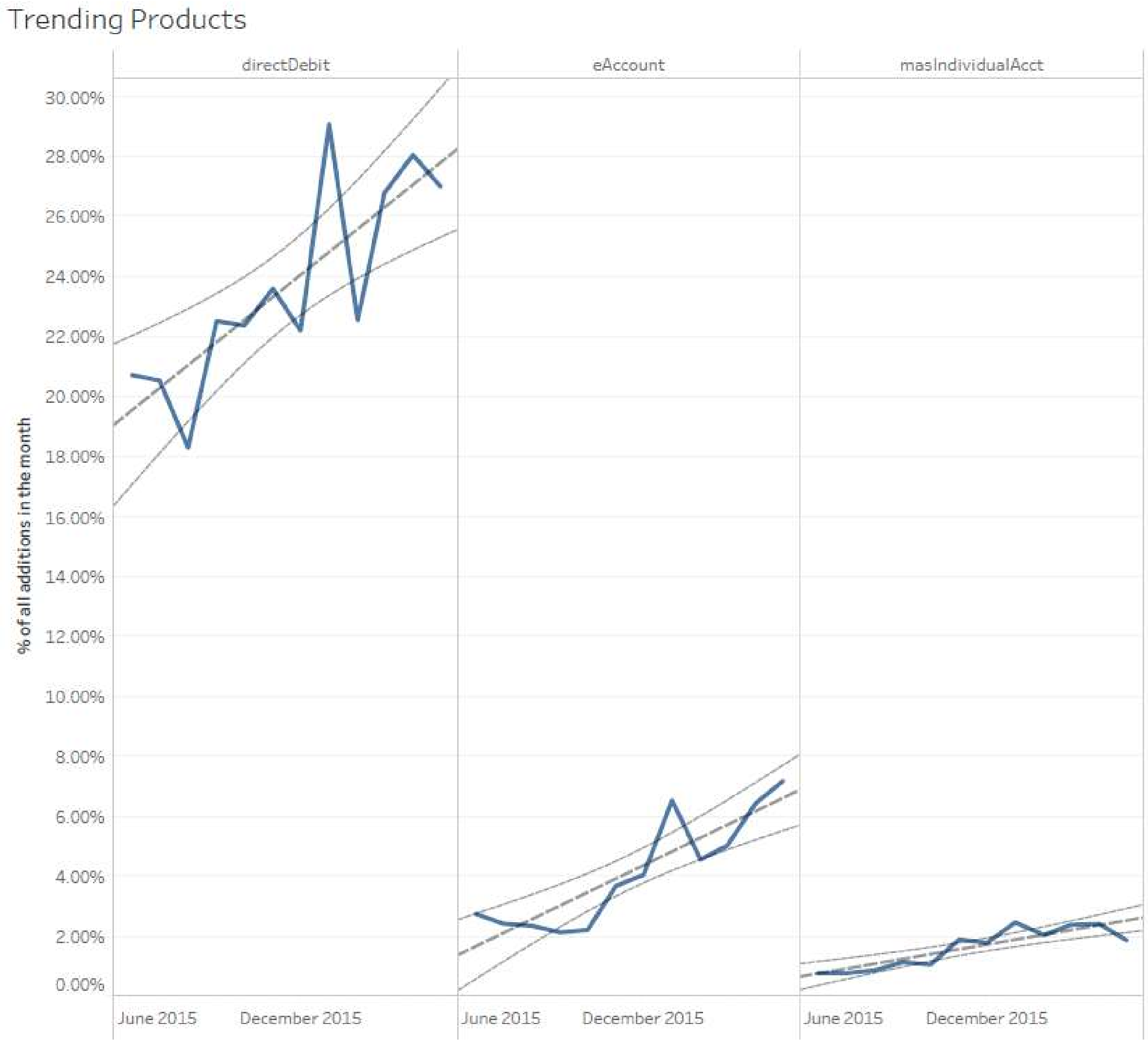


The chart below shows similar information with a different view:

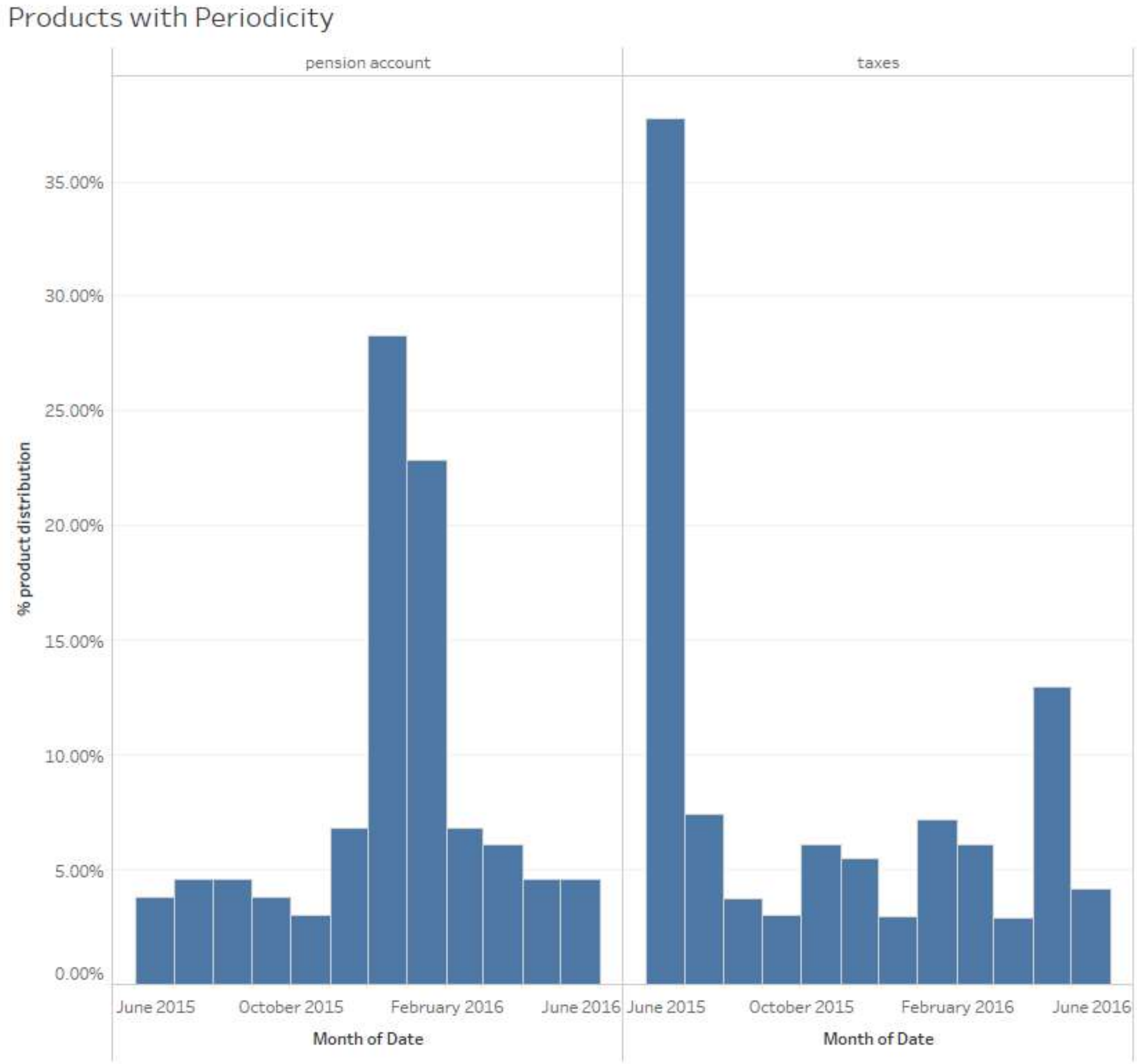


The products could be categorized into 3 categories. The first one is trending product, where positive or negative trends are exhibited over time. The second category is products that act differently in different months. The third category is products that do not exhibit trend nor periodicity.

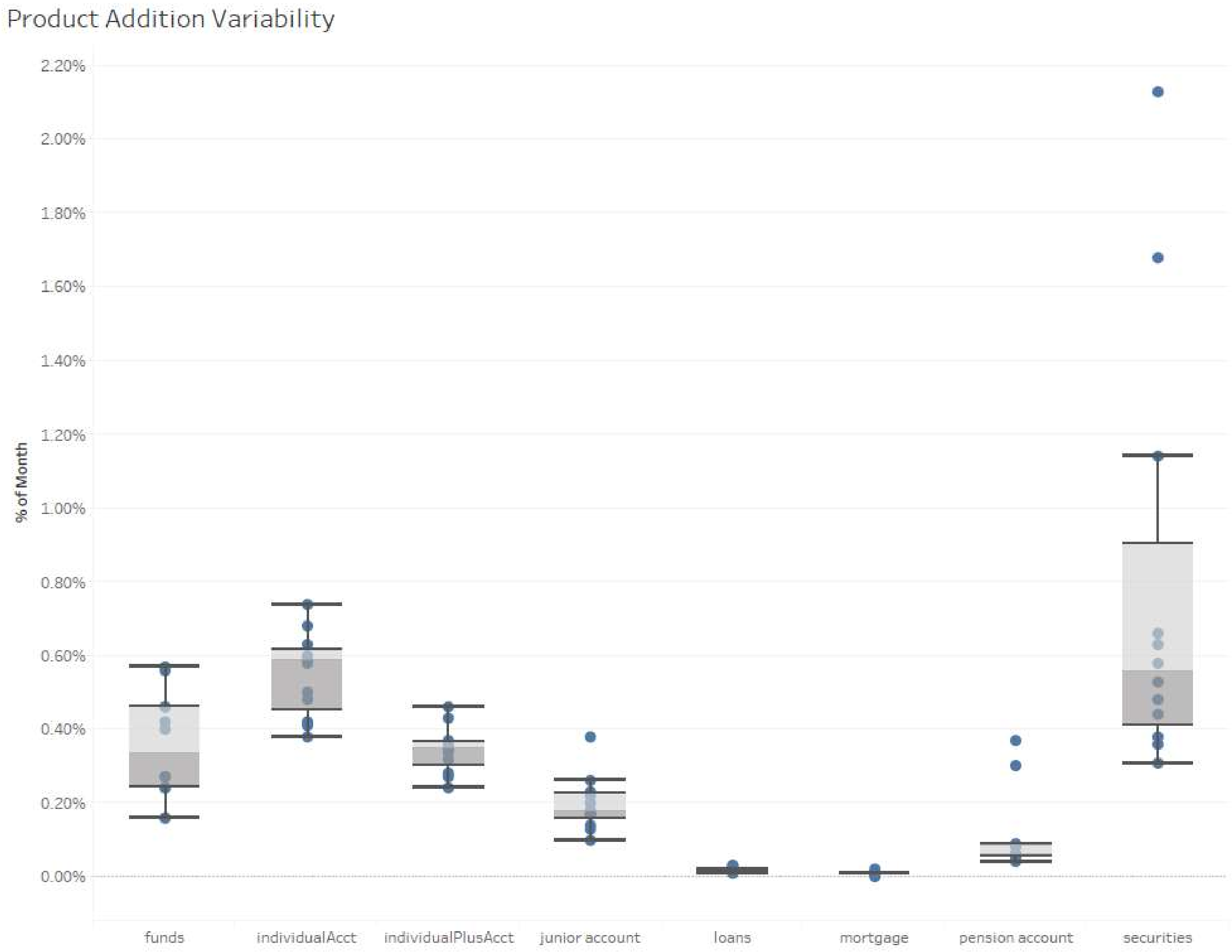
**Trending Products:** The chart below shows each product as a percentage of all the products added in the month. Direct debit, e-account, and MAS individual account exhibit positive trend over time, which indicates that more recent observations may be more meaningful in making predictions for the future months. The prediction of these three products are based on the data of the most recent three months.



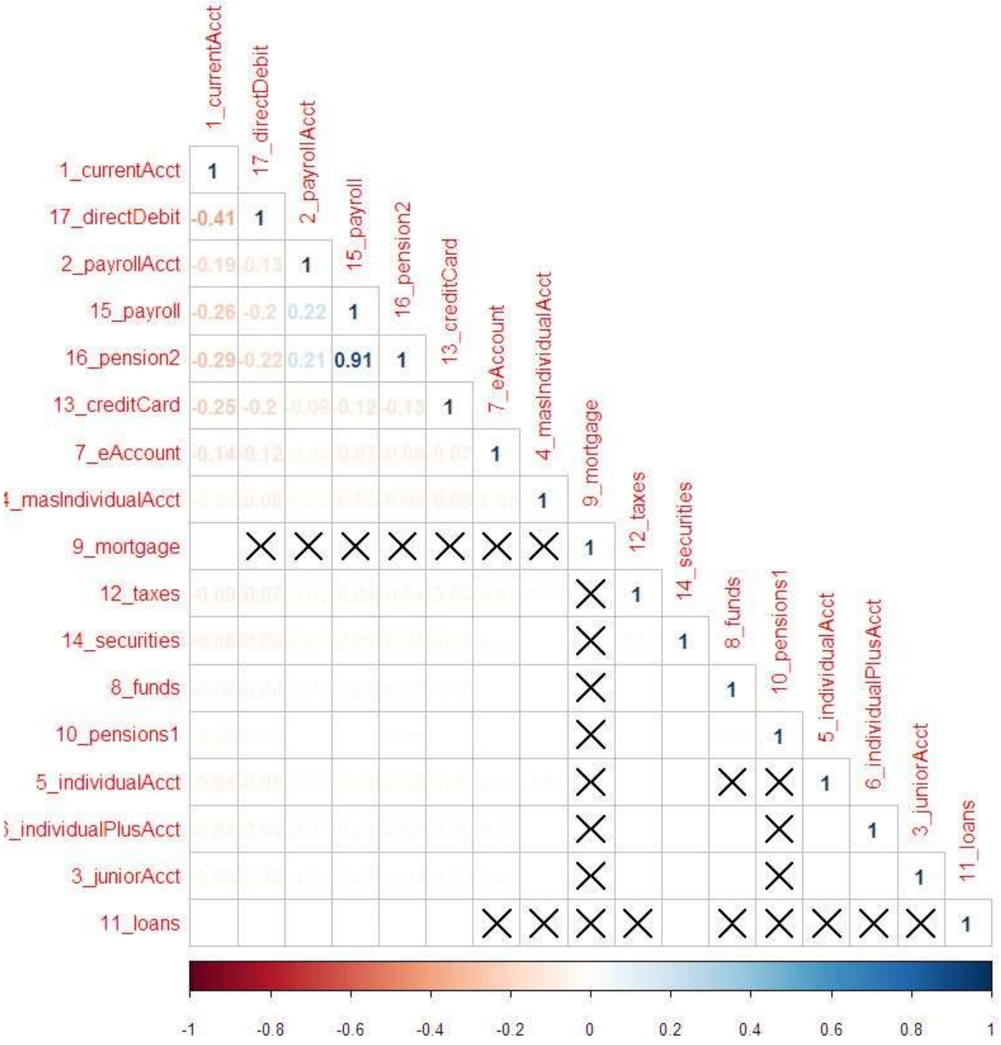
**Products with Periodicity:** The chart below shows the distribution of product addition for each month. A product not exhibiting periodicity will have even distribution across different months. Pension account and tax product exhibits periodicity, in which that product addition is not evenly distributed across different months. For all the tax product added, 37% occurred in June. An explanation for this occurrence might be that June is end of tax season in Spain. Approximately 50% of pension accounts are added in December and January, coinciding with yearend. Since the competition is to predict product addition in June, the June data is used to predict the tax product.

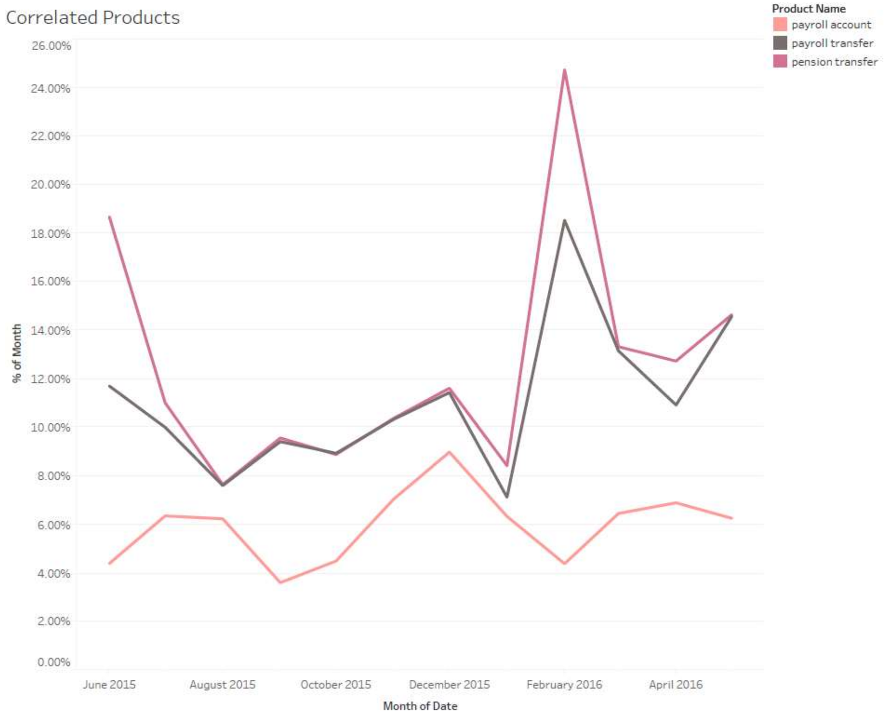


**Stable Products:** The chart below shows the distribution of each product as the % of total products added for the month. Many of the products, such as individual account, loans, and mortgages, not only makes up as a very small percentage of products that are added for the month, but also the variability across months remain about the same every month.

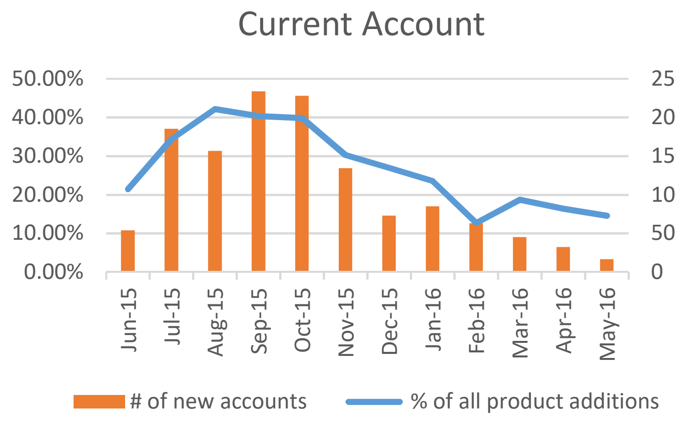


**Correlated Products**: The chart below shows the correlation of product additions. Correlation that are below 95% confidence interval are crossed out. Payroll transfer (15), and pension transfer (16) has correlation of 0.91, which means they are frequently bought together. Those two type of products are also correlated with the Payroll account (2), with correlation of 0.22 and 0.21. In addition, a customer cannot have the payroll transfer product unless the customer has the pension transfer product. Of the customers that has the pension transfer product, 91.67% also has the payroll transfer product. In the modeling process, if the customer did not own both payroll transfer and pension transfer product in the prior month, and the probability of payroll transfer is greater than pension transfer, the probability is adjusted so that the probabilities are switched between the two. This adjustment improves the MAP score by 0.00003. The insight of switching probability was learned from a post in the Kaggle forum.



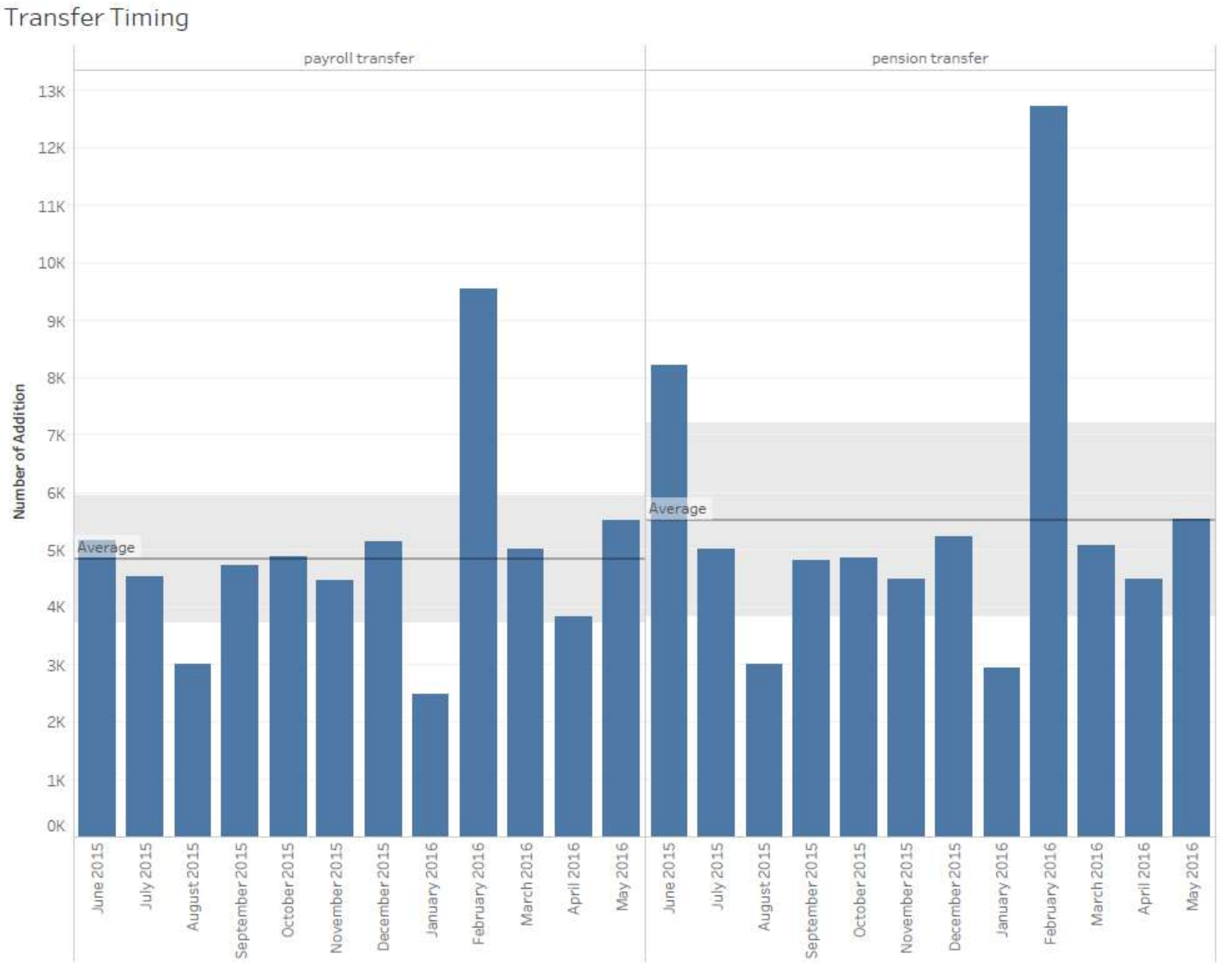


**Anomalies Explained (Current Account):** The line in the chart below shows the % of current account relative to all the products added in a certain month. In June 2015, current account makes up 21.49% of all the products added. The percentage reached as high as 42% in August 2015. In fact, the surge of additions from July 2015 to October 2015 coincides with the surge in new accounts openings in the same period. The fluctuation proved to be challenging in modeling. Using different months to make prediction changes the accuracy dramatically. For my case, the MAP score varies from 0.0271467 to 0.0306891 holding all else constant. The final model was run with June 2015 and December 2015. June 2015 was chosen because it is in the same month as the testing set. December 2015 was chosen because current account as a percentage of all the products added in the month is closest to the percentage in the testing set, which is found to be around 27.6%.



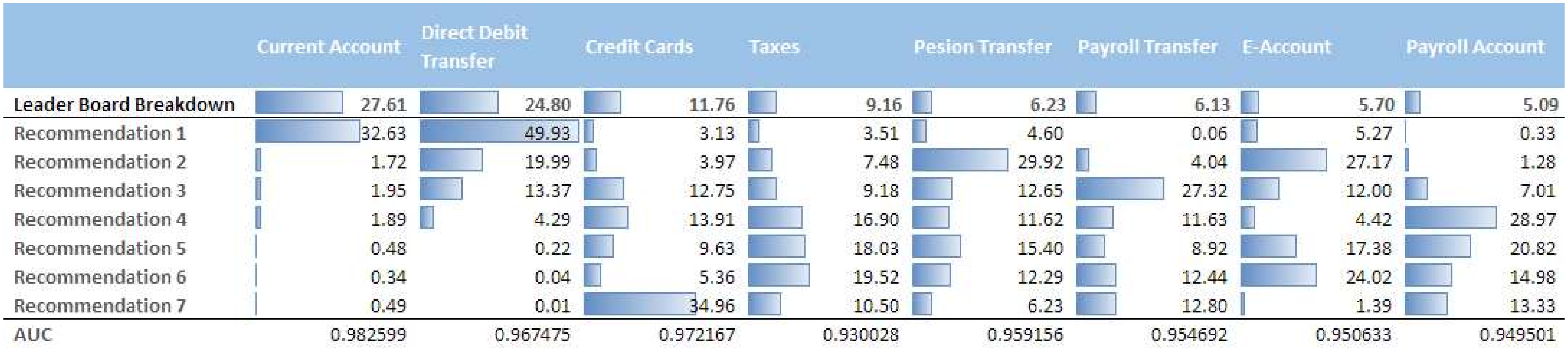
| date | Current Account |
|---|---|
| Jun-15 | 21.5% |
| Jul-15 | 34.4% |
| Aug-15 | 42.2% |
| Sep-15 | 40.4% |
| Oct-15 | 39.9% |
| Nov-15 | 30.2% |
| Dec-15 | 27.0% |
| Jan-16 | 23.5% |
| Feb-16 | 12.7% |
| Mar-16 | 18.7% |
| Apr-16 | 16.4% |
| May-16 | 14.6% |
| Jun-16 | 27.6% |
| Std Dev | 10.5% |
| Mean | 26.8% |
| Median | 25.2% |

**Anomalies Explained (pension transfer and payroll transfer):** The unusual spike of addition for pension transfer and payroll transfer is probably due to timing of transaction. There was an unusual dip in January 2016, possibly due to holiday, and the dip is made up in February 2016. The other months are relatively flat.



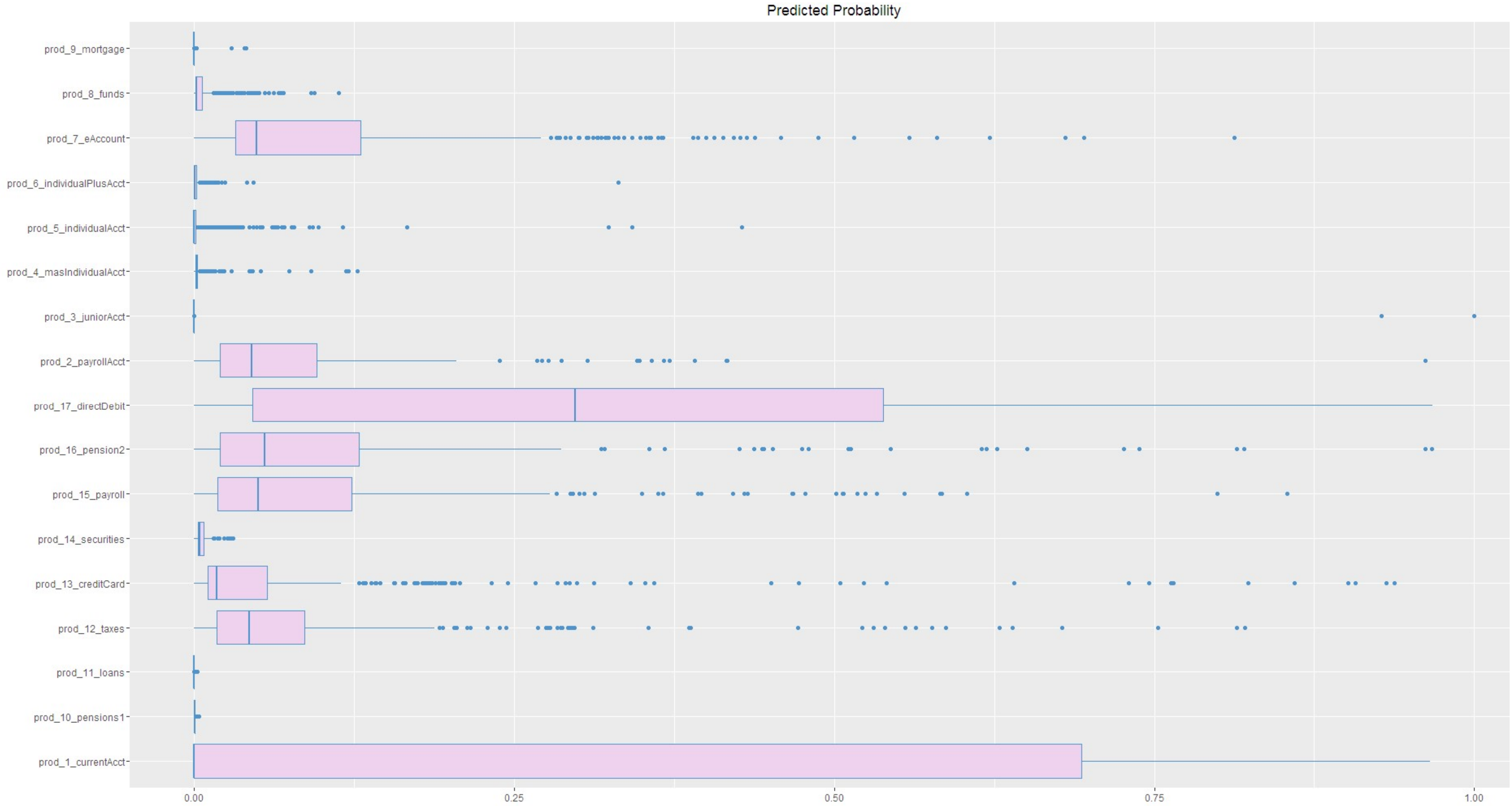
**How did the predictive models do?**

The product that is the most frequently added in the test set is current account, which accounts for around 27.6% of all products added in June 2016. The second most common product added in the test set is direct debit transfer. In general, the most frequently added products in the test sets are also the most frequently recommended products in the models. The accuracy of recommendations could benefit from more accurately recommending e-account. The likelihood of adding e-account seem to be over-recommended by the models.



| | Current Account | Direct Debit Transfer | Credit Cards | Taxes | Pesion Transfer | Payroll Transfer | E-Account | Payroll Account |
|---|---|---|---|---|---|---|---|---|
| Leader Board Breakdown | 27.61 | 24.80 | 11.76 | 9.16 | 6.23 | 6.13 | 5.70 | 5.09 |
| Recommendation 1 | 32.63 | 49.93 | 3.13 | 3.51 | 4.60 | 0.06 | 5.27 | 0.33 |
| Recommendation 2 | 1.72 | 19.99 | 3.97 | 7.48 | 29.92 | 4.04 | 27.17 | 1.28 |
| Recommendation 3 | 1.95 | 13.37 | 12.75 | 9.18 | 12.65 | 27.32 | 12.00 | 7.01 |
| Recommendation 4 | 1.89 | 4.29 | 13.91 | 16.90 | 11.62 | 11.63 | 4.42 | 28.97 |
| Recommendation 5 | 0.48 | 0.22 | 9.63 | 18.03 | 15.40 | 8.92 | 17.38 | 20.82 |
| Recommendation 6 | 0.34 | 0.04 | 5.36 | 19.52 | 12.29 | 12.44 | 24.02 | 14.98 |
| Recommendation 7 | 0.49 | 0.01 | 34.96 | 10.50 | 6.23 | 12.80 | 1.39 | 13.33 |
| AUC | 0.982599 | 0.967475 | 0.972167 | 0.930028 | 0.959156 | 0.954692 | 0.950633 | 0.949501 |

## What is the distribution of probability for each product?

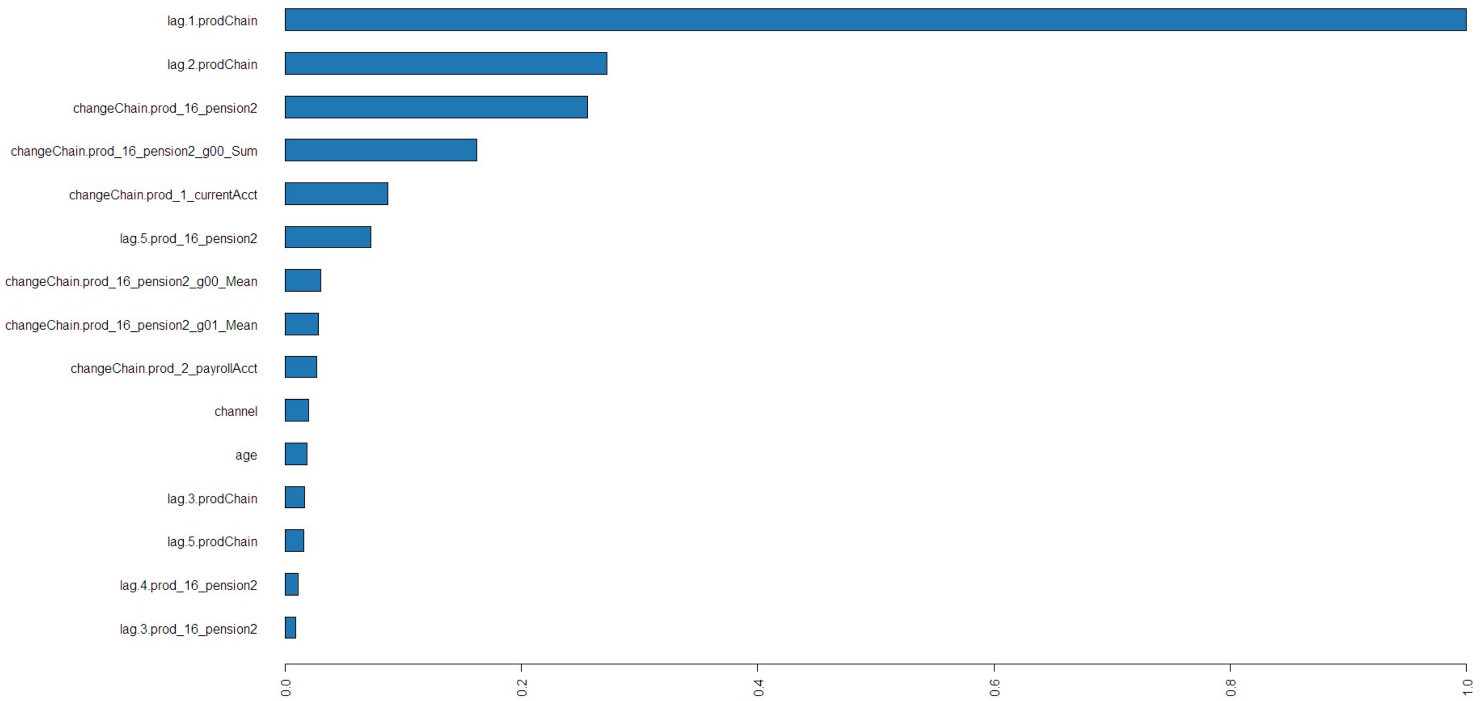


## What could be done next?

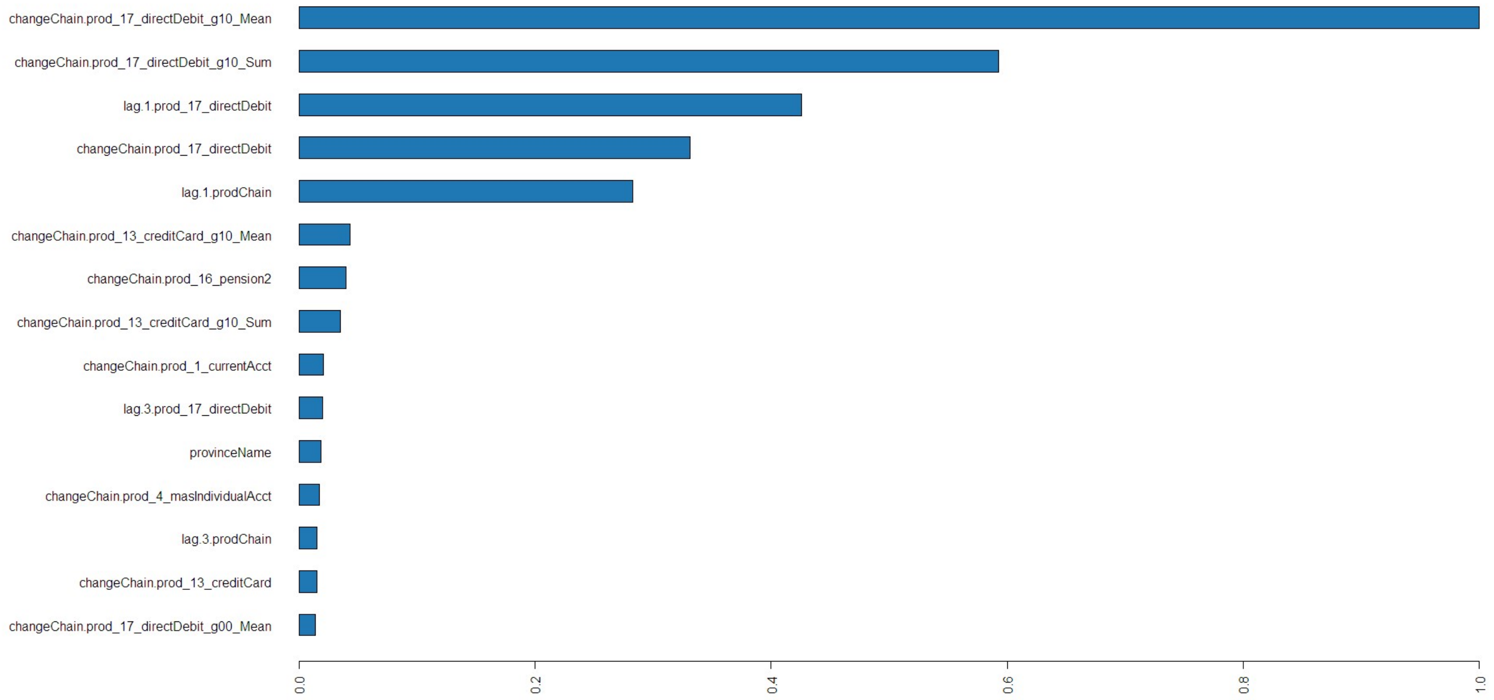
Upon reviewing the variable importance of each product, the predictors that contributes the most to the model predictions are usually the consumer's history of that product. However, e-account is an exception, being predicted by the MAS individual account. It will be interesting to see if the correlation between the two are spurious, or has a fundamental reason behind it. It will also be interesting to review the precision and recall of each model to see if manual adjustments to probability based on accuracy of the models will improve score.

## Variable Importance

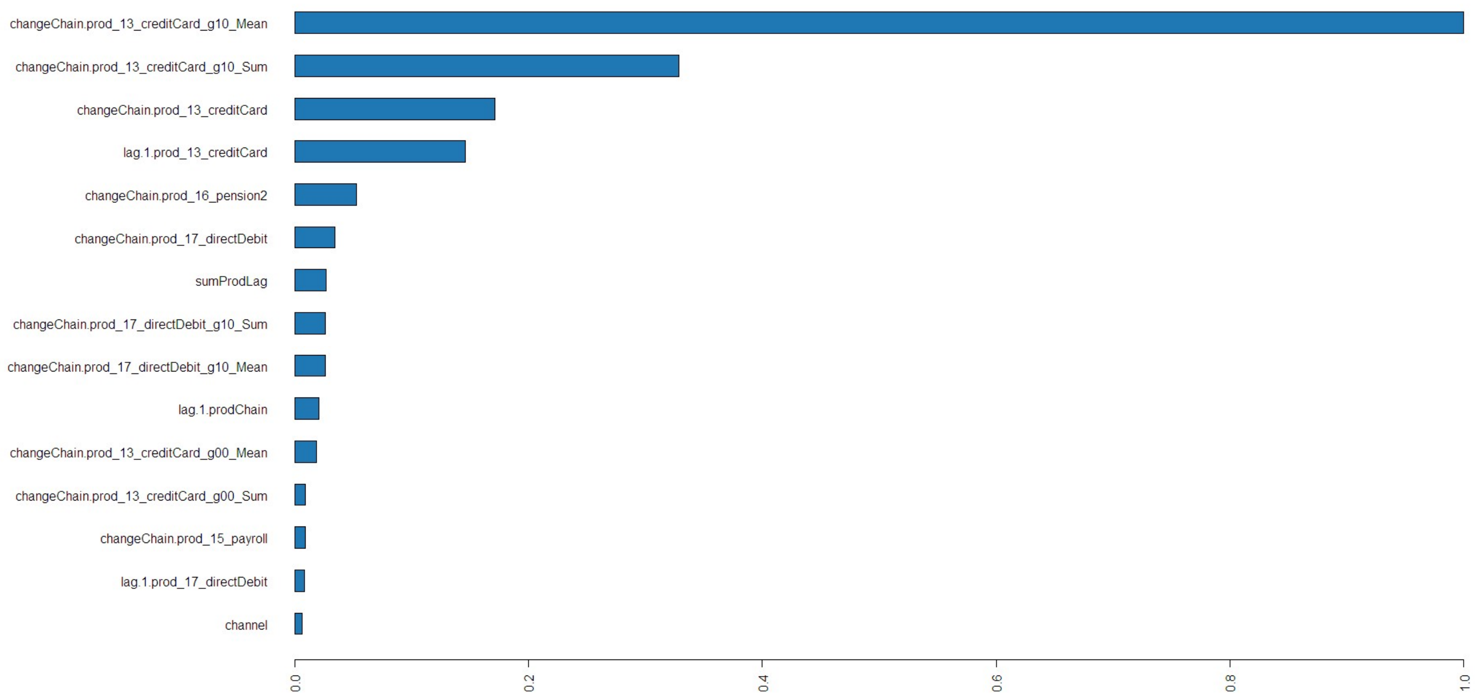
### Current Account



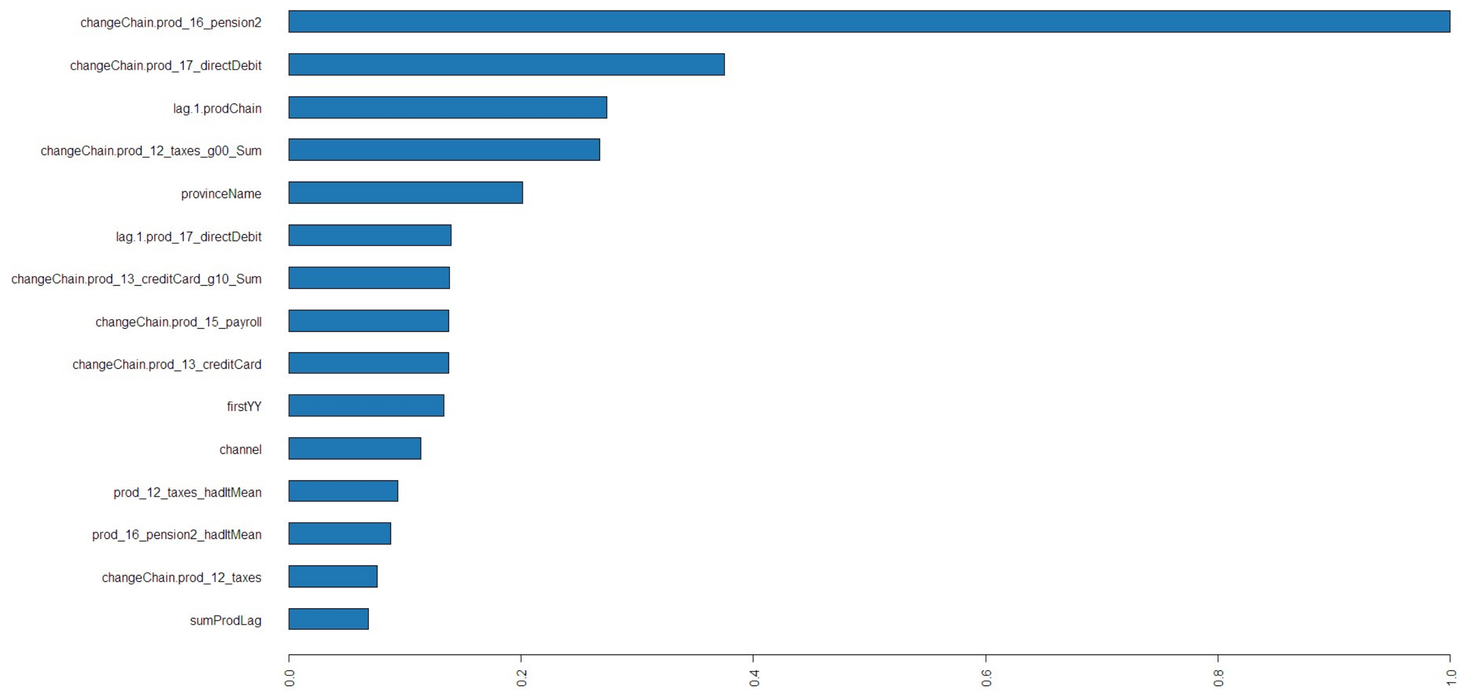
### Direct Debit Transfer



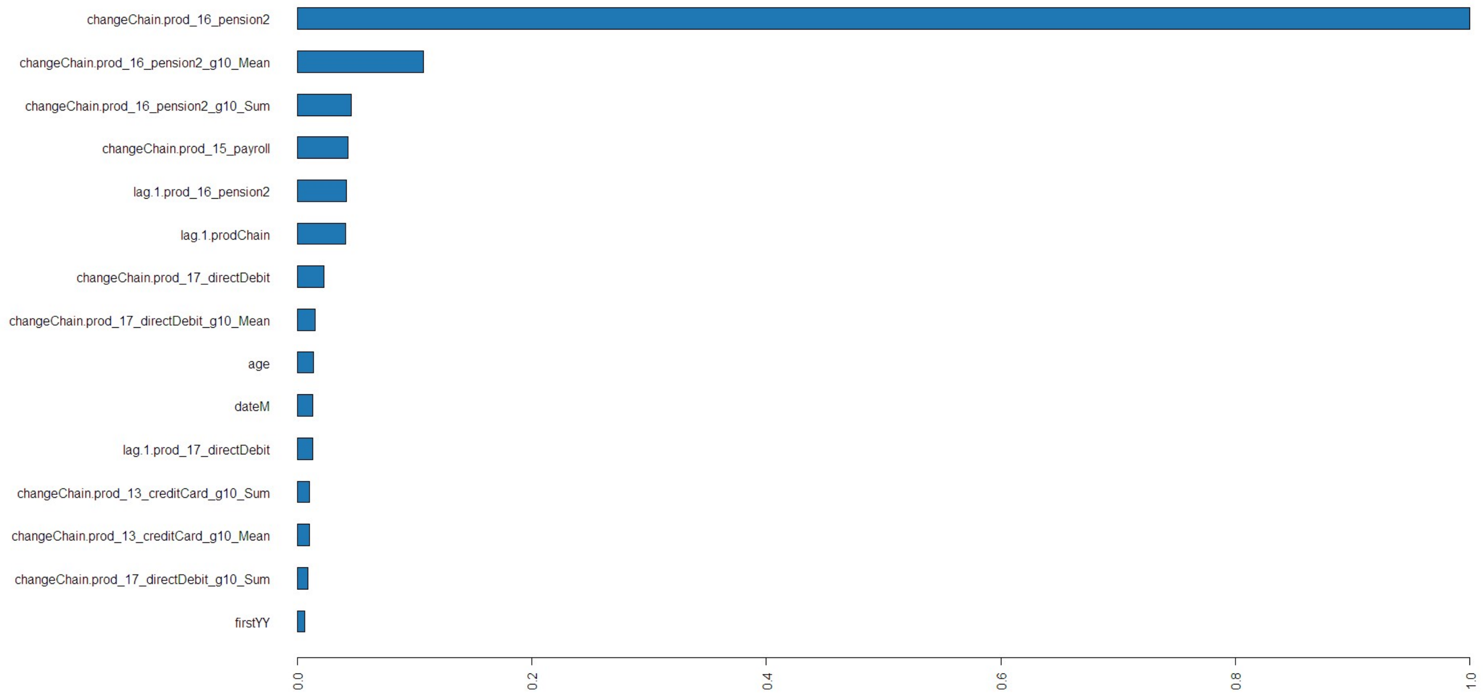
## Credit Cards



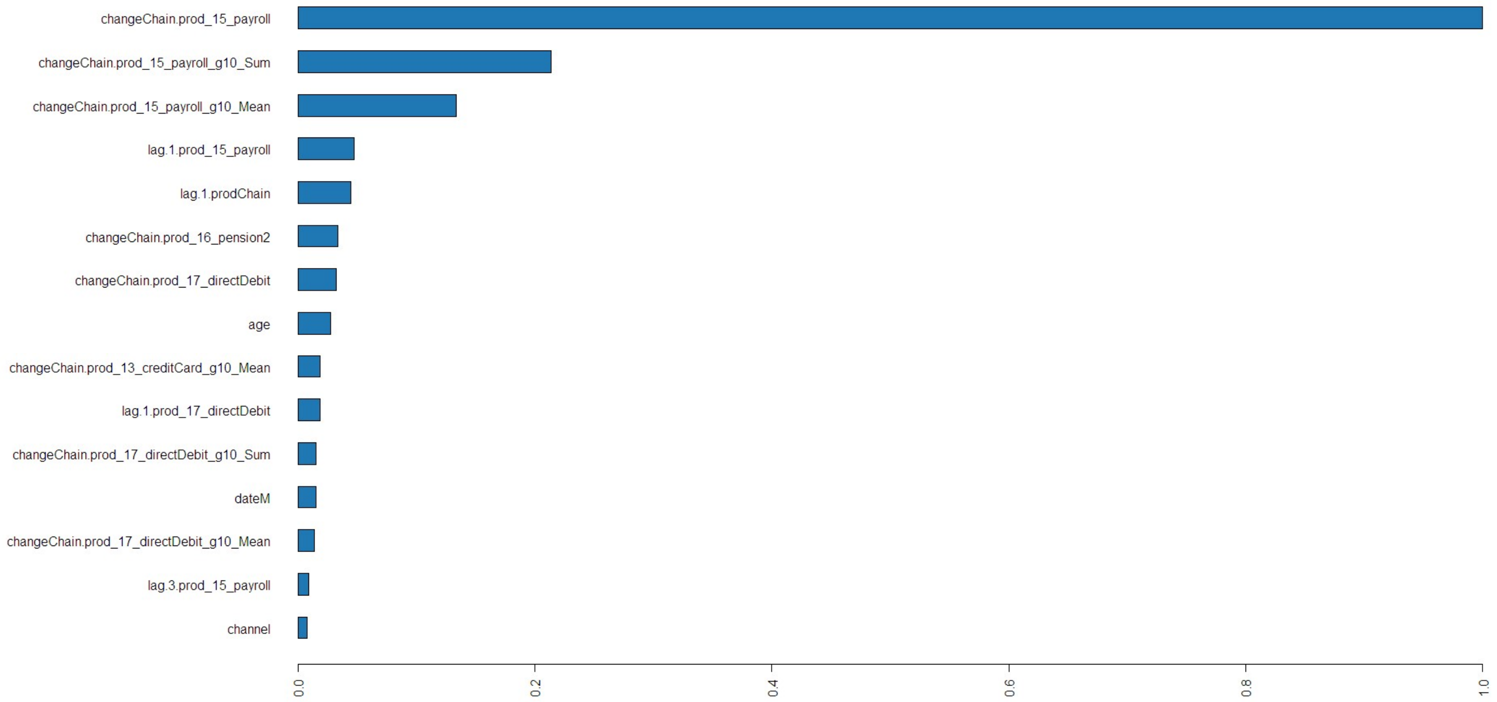
## Tax



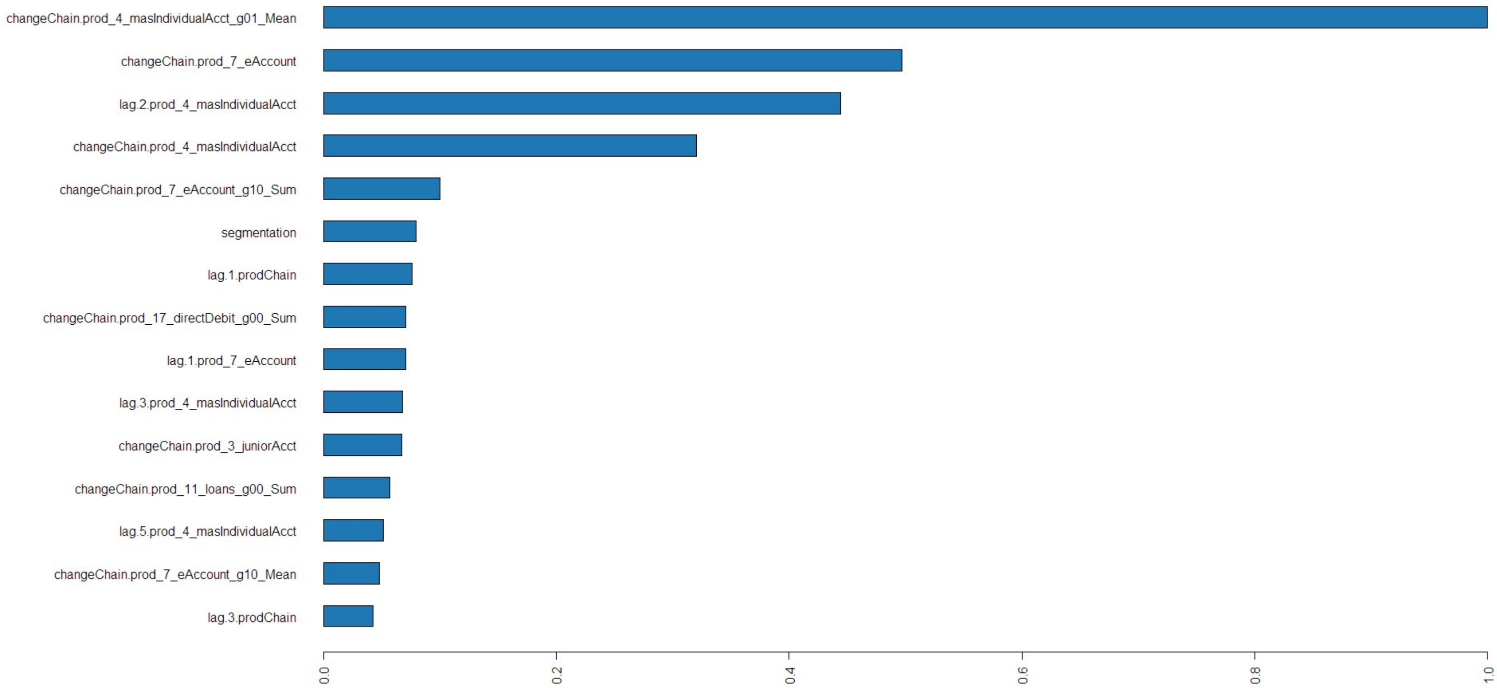
## Pension Transfer



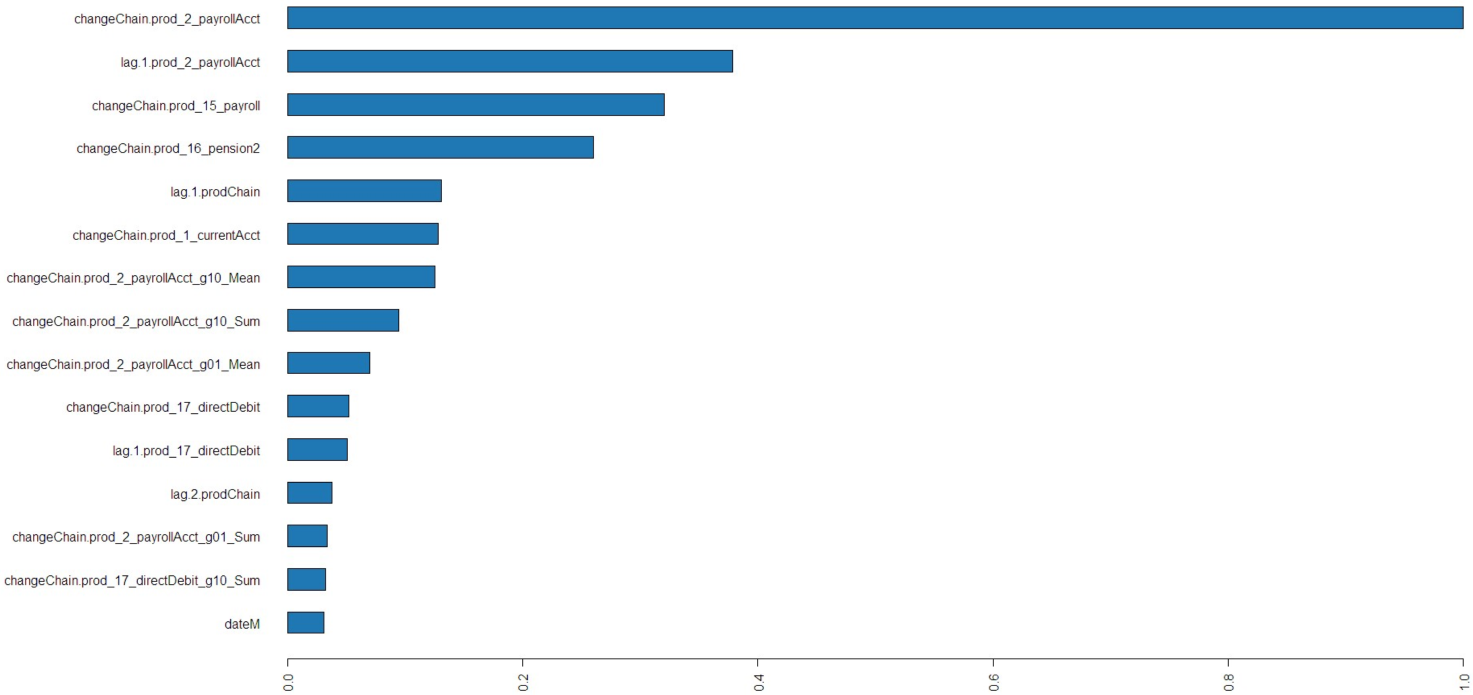
## Payroll Transfer



## E-account



## Payroll Account


changeChain.prod_2_payrollAcct
lag.1.prod_2_payrollAcct
changeChain.prod_15_payroll
changeChain.prod_16_pension2
lag.1.prodChain
changeChain.prod_1_currentAcct
changeChain.prod_2_payrollAcct_g10_Mean
changeChain.prod_2_payrollAcct_g10_Sum
changeChain.prod_2_payrollAcct_g01_Mean
changeChain.prod_17_directDebit
lag.1.prod_17_directDebit
lag.2.prodChain
changeChain.prod_2_payrollAcct_g01_Sum
changeChain.prod_17_directDebit_g10_Sum
dateM