Transaction Data Analysis and customer loyalty score prediction

XIAOCHEN JIN

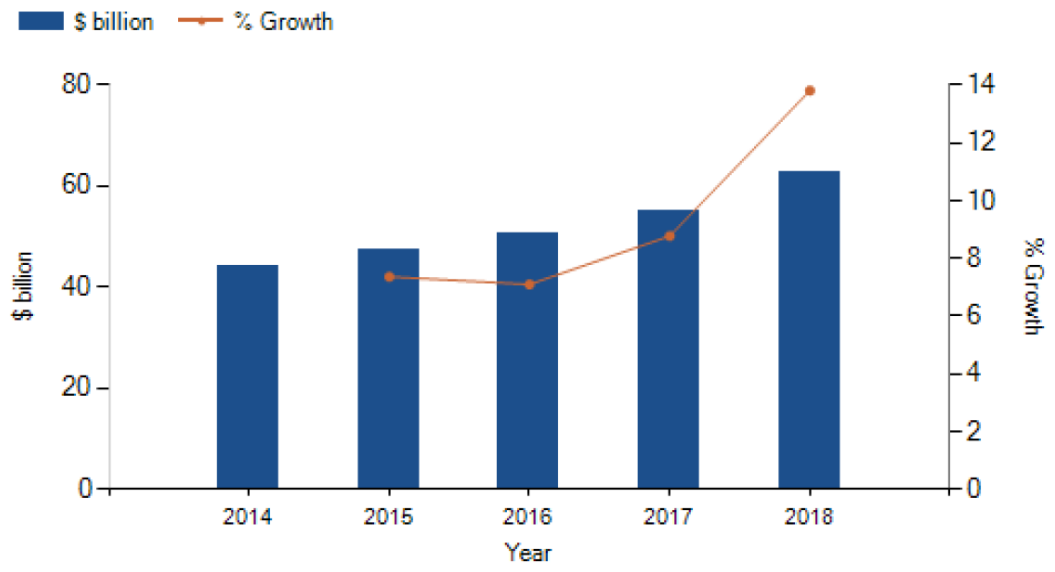
2019/08/22

Background

- Customer loyalty has long been an issue that attract credit card companies' attention
  - it is much more costly to attract new customers
  - Features relate with customer loyalty
- Challenge of maintaining loyal customers
  - Issue for multiple card holders
  - Low switching cost and 0% balance transfer

Credit Card Industry in Brazil

Figure 1: Brazil credit cards market value: $ billion, 2014–18



SOURCE: MARKETLINE

MARKETLINE

- Market is expanding
  - Continue to grow in 5 years
  - Expect to reach 94.1 billion dollars in 2023
- Elo as an emerging competitor in this market, also expects to expand its own share of profit

Data Overview

Data is constructed by five files in total:

- Training data:
  - Target value, basic credit card information
- Historical transaction data:
  - Transaction history in the past 13 months
- More recent transaction history data:
  - Transaction history in the past 2 months
- Merchant information data:
  - Various detailed information related with merchants
- Test data:
  - Same features as training data expect target is removed

Data exploration

- How big is the dataset?
  - The largest data file among the five ones is historical transaction data, which includes 29,112,361 lines of records (2.65G in size). There are 325,540 unique credit card ID and 326,312 unique merchandize ID in historical transaction data file.

- Train vs Test
  - Train and test file share very similar distribution in their variables such as card activation date and censored features

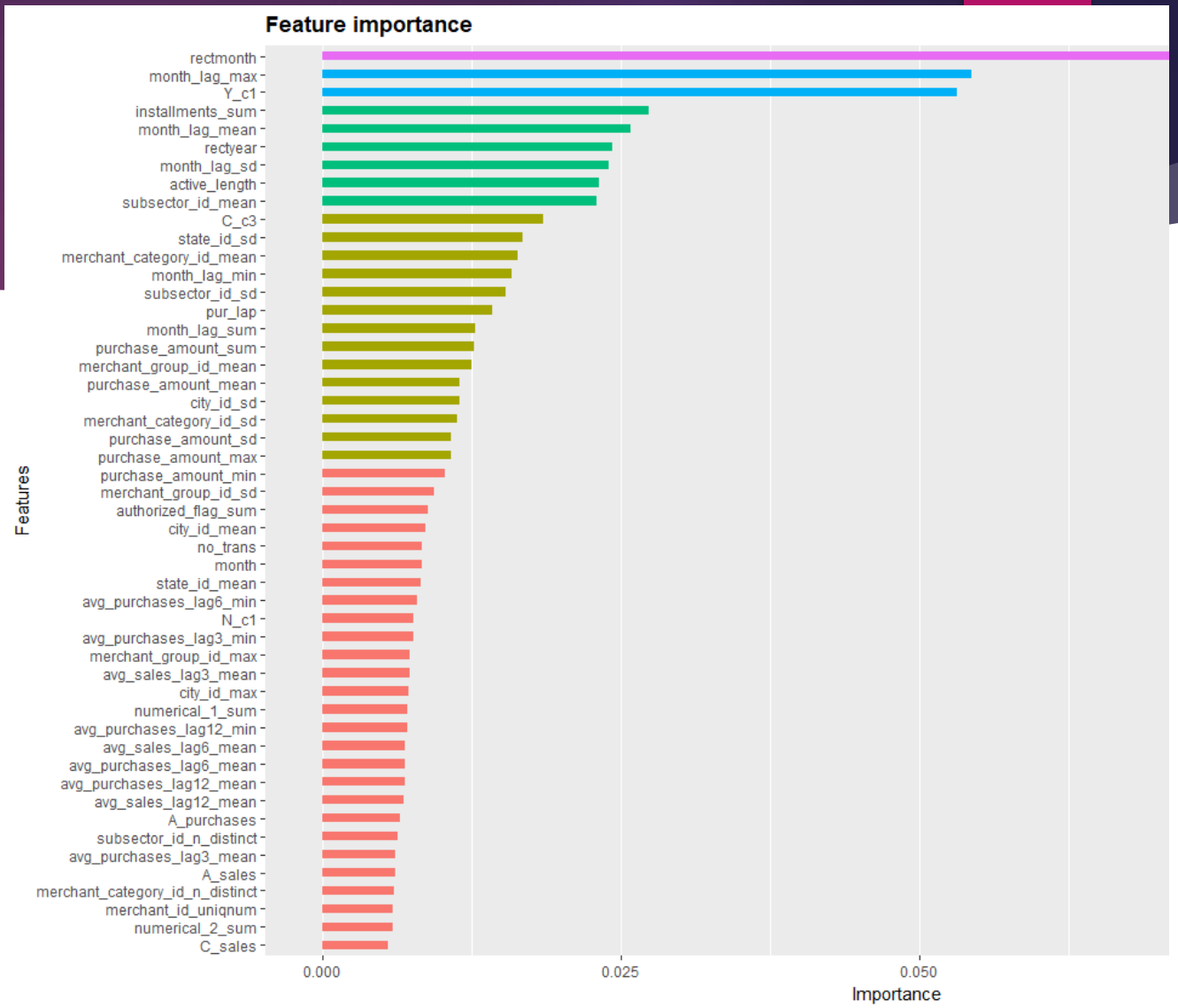
Data Aggregation

Data Aggregation

- First is the file combination.
  - Join train data, merchant data and two transaction data files together based on card ID in train set
- Second is feature creation and aggregation.
  - In the process of joining, data such as transaction records need to be aggregated in the level of card ID
  - Methods such as sum, mean, standard deviation were applied to numerical variables
  - Categorical variables are processed by calculating summarized statistical values on number of occurrence for each level
- The third step was mapping this aggregated data set back to train file according to card ID.
  - The final data set ended up with 137 variables in total

Feature Importance

- Generated feature importance graph from XGBoost model
- The most important features are most recent purchase month, maximum number of month lag between transactions, and number of Yes in category_1



Feature Importance

- Simple linear regression results

	Estimate	Standard Error	t-value	P-value
Authorized_flag_sum	1.728485e-02	3.326174e-03	5.196616	2.056133e-07
Y_C1	-3.531188e-02	7.336123e-03	-4.813425	1.498128e-06
N_C1	-3.287560e-02	7.496519e-03	-4.385448	1.165364e-05
Avg_pur_lap	2.980557e-02	8.531616e-03	3.493544	4.780305e-04
Month_lag_sd	-2.626114e-01	7.870639e-02	-3.336596	8.501741e-04
Month_lag_max	-4.156970e-01	1.263291e-01	-3.290588	1.002093e-03
Month_lag_mean	-1.079141e-01	3.484160e-02	-3.097277	1.956681e-03
Merchant_category_id_max	-2.104776e-03	7.371669e-04	-2.855223	4.306572e-03
Merchant_category_id_min	1.361909e-03	5.445392e-04	2.501030	1.239385e-02
Merchant_category_id_sd	2.573909e-03	1.083403e-03	2.375763	1.752522e-02
State_id_mean	2.027150e-02	9.027418e-03	2.245548	2.474748e-02
A_c3	5.745411e-02	2.728081e-02	2.106027	3.521863e-02
Category_1_uniqnum	-2.344199e-01	1.132406e-01	-2.070105	3.845962e-02
Merchant_group_id_max	-4.510188e-06	2.194619e-06	-2.055112	3.988550e-02
Merchant_group_id_sd	1.233493e-05	6.253304e-06	1.972546	4.856569e-02

Non-informative predictors removal

Multicollinearity Issue

Feature Selection

Feature Selection: Non-informative predictors removal

- Non-informative predictors refers to predictors with low variation and limited ability in offering new information
- Two criteria:
  - Frequency of the most frequent unique value over the total occurrence
  - The ratio between most frequent value and second frequent value in one variable
- Predictors are considered non-informative if the most frequent value occurred more than 70% of the total times and 20 times more frequent than the second frequent value

Selected Non-informative Features

- Column 1 represents the rate of occurrence of the most frequent value in each feature
- Column 2 represents the ratio between most frequent value and second most frequent value in each feature

Column 1	Column 2	Variable Names
0.8597047	20.91938	merchant_group_id_min
0.9497318	23.11003	active_months_lag3_min
0.9056791	22.03808	active_months_lag6_min
0.9588990	23.33309	active_months_lag3_max
0.9588494	23.33189	active_months_lag6_max
0.9548280	23.23403	active_months_lag12_max
0.9497318	13697.64286	active_months_lag3_mean
0.9056791	2438.29333	active_months_lag6_mean
0.9264401	18706.40000	active_months_lag3_sd
0.8824368	4143.69767	active_months_lag6_sd
0.9497368	23.11015	active_months_lag3_n_distin
0.9163567	22.29790	numerical_1_min
0.9420009	22.92191	numerical_2_min
0.8931591	31.22299	C2_ 2
0.8440498	36.06942	C2_4

Feature Selection: Multicollinearity Issues

- Multicollinearity refers to the issue that predictors are correlated with each other, which affects their ability in predicting the target value
- I used the following algorithm provided by Kuhn and Johnson (2013) to remove predictors with multicollinearity issue :

1. Calculate the pairwise correlation matrix of the predictors.

2. Determine the two variables (A and B) that has the largest absolute pairwise correlation.

3. Calculate the average correlation between variable A and other variables, repeat the same process with variable B.

4. Compare the average correlation value for A and B, remove the predictor with larger correlation.

5. Repeat steps 2-4 until no absolute correlation is above the threshold.

- Set threshold as 0.8, 29 variables are considered having multicollinearity

Distribution of Target value

- The distribution of target follows a gaussian like distribution
- With center around 0 and standard deviation 3.85
- It has a long left tail with minimum number as -33.21928
- Which determines the analysis plan of building separate models for outlier target values

Analysis Plan Determination

- Adding feature inspired from Kaggle kernel
  - Average monthly purchase amount ratio
  - By aggregating transactions for each card by month, then calculating purchase amount ratio across months and averaging the results by card in the end.
- Three sets of cross validation
  - 5 times 80% data partition
  - 5 folds cross validation
  - 10 folds cross validation
- Three sets of training data
  - Full set with all features (138 features)
  - Without non-informative features (123 features)
  - Without non-informative and multicollinear features (94 features)
- Build separate models for outlier values and create ensemble prediction with signed weight

Method

Parametric method:

- GLM with elastic net
- Ridge Regression

Non-parametric method:

- Random Forest
- Extreme Gradient Boosting

Parametric method: GLM with elastic net

- Basic Formula(Hastie and Qian, 2014):

$$\min_{\beta_0,\beta} \frac{1}{N} \sum_{i=1}^{N} w_i l(y_i, \beta_0 + \beta^T x_i) + \lambda[(1-\alpha) \|\beta\|_2^2 /2 + \alpha \|\beta\|_1].$$

- Hyperparameter tuning selection:
  - α grid search 10 values from 0.3 or 0.7
  - λ grid search 10 values from 0.001 to 0.01

Parametric method: Ridge regression

- Basic Formula (James et al., 2013):

$$\hat{\beta}^{ridge} = \underset{\beta}{argmin} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\} = RSS + \lambda \sum_{j=1}^{p} \beta_j^2$$

- Hyperparameter tuning selection
  - Set α to 0
  - λ grid search 20 values from 0 to 0.2

Nonparametric method: Random Forest

- Advantage of using Random Forest
- Basic algorithm(Hastie et al., 2009):

1. For decision tree b from 1 to B:

   a. Create bootstrap sample Z* of size N from training set

   b. Build decision tree T_b on each set of bootstrap samples by repeating the following steps until minimum size of terminal node n_{min} is reached:

      i. Select *m* number of features at random from *p* features

      ii. Split the best split-point from *m* into two nodes

2. Output the ensemble tree $\{T_b\}_{B1}$

- Hyperparameter selection
  - number of variables that randomly selected in each split (*m*)
  - minimum size of terminal node (n_{min})
  - total number of trees in the ensemble (*B*)
  - *m* from 20 to 40
  - n_{min} equals 5
  - B equals 500
- Parallelization

Nonparametric Method: XGBoost

- Basic Algorithm for boosting (James et.al., 2013):

1. Set initial prediction of Y as F_0, which equals the value that minimize the loss function residual defines as $y_i - f_0(x)$
2. For m from 1 to M:

   a. For i from 1 to N:

      i. Fit tree model h_m on residuals and get new prediction $f_m = f_{m-1} + h_m$

      ii. Compute updated residuals $y_i - f_m(x)$
3. Repeat the previous steps until the residual $y_i - f_m(x)$ stop decreasing

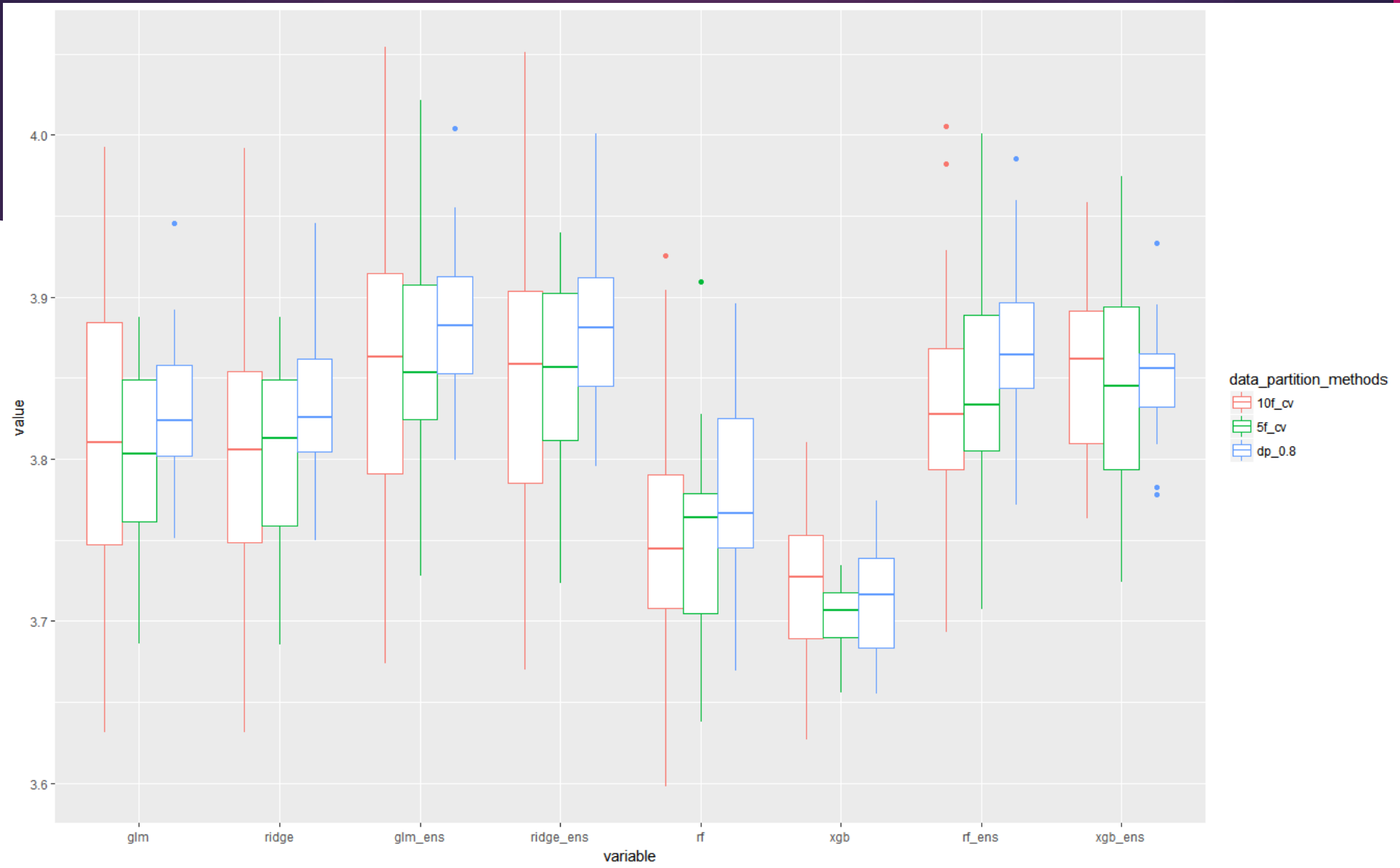
- Gradient boosting

Nonparametric Method: XGBoost

- **Hyperparameter tuning selection:**
  - Optimal number of iterations *M* equals 1000
  - learning rate η equaled to 0.01, which is small enough to balance iteration number
  - L1 regulation parameter α equaled 0
  - L2 regulation parameter λ equaled 0 or 5
  - minimum split loss *gamma* was 0 or 1
  - maximum depth of each tree was grid searched from 5, 15, 25 levels
  - minimum child weight is controlled as 5 or 10
  - 80% of the total number of training set were selected at random
  - 70% of the column features were selected
  - 60% of the column features were selected

Results

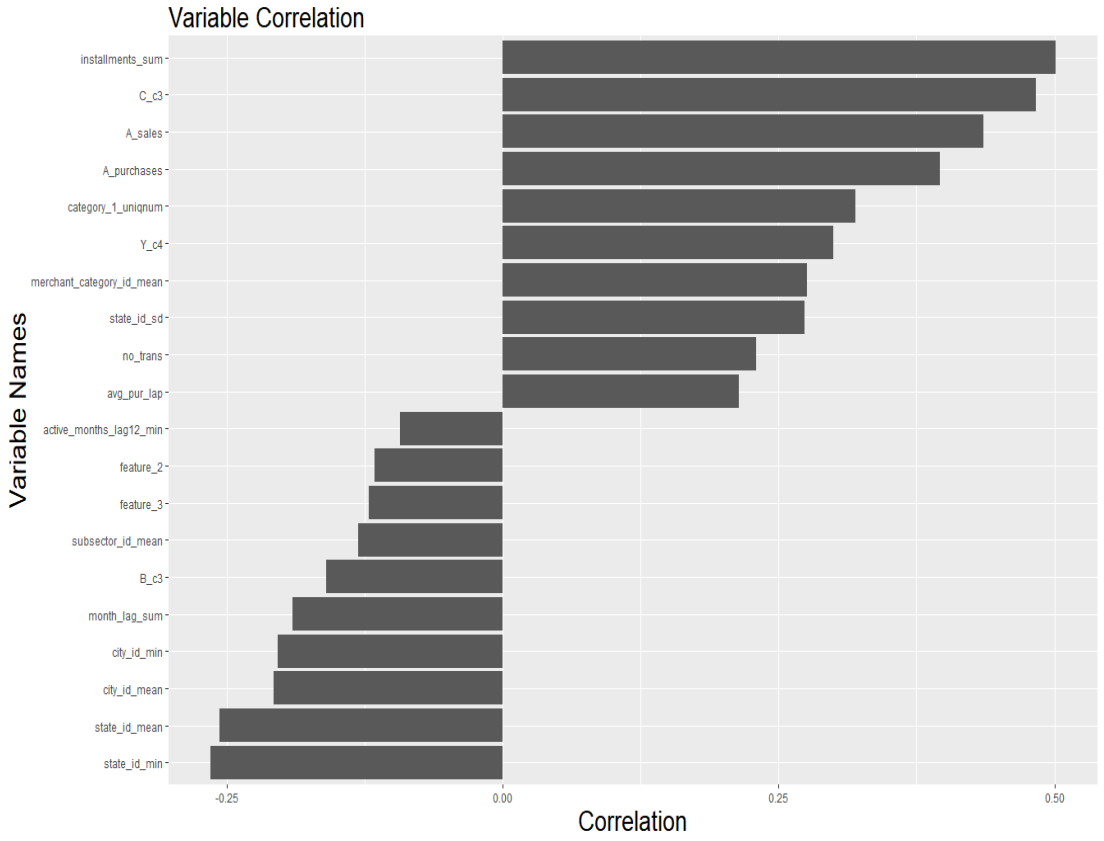
- The result shows that the model with best performance is XGBoost
- The performance between ridge regression and GLM were very similar with each other
- There were slightly difference between different cross-validation splits
- Model outliers separately did not improve the prediction accuracy
  - Most weight equals to 0, which means that only non-outlier predictions were used
  - Ensemble models had lower variance in RMSE for predictions

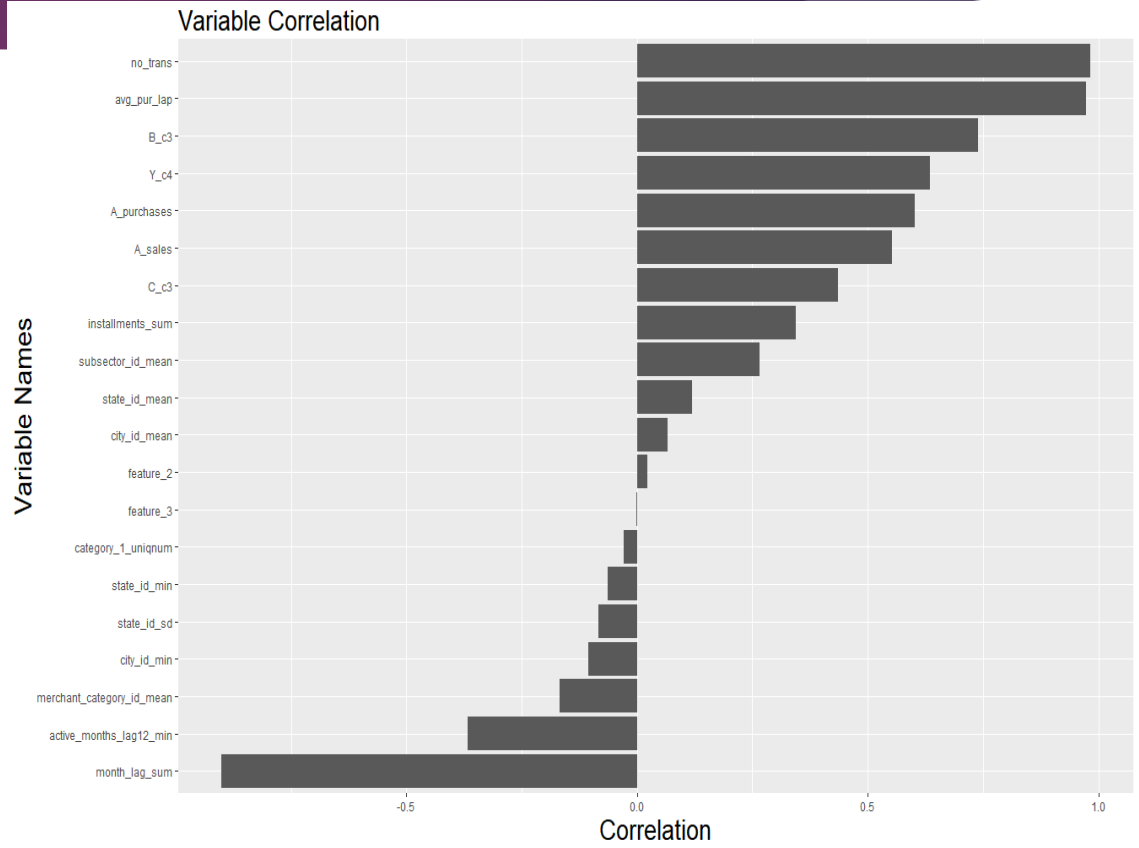

4.0
3.9
3.8
3.7
3.6
value
glm
ridge
glm_ens
ridge_ens
rf
xgb
rf_ens
xgb_ens
variable
data_partition_methods
10f_cv
5f_cv
dp_0.8

Censored data exploration

- Censored numerical variables
  - Numerical 1 and Numerical 2 from Merchant dataset
  - Based on a kernel discussion, the author used genetic programming to transform values in the features to price like form
  - 40% Values end with 99,00,50 and matches with real life expense in Brazil
- Censored categorical variables
  - Three censored categorical variable in merchant data set and three in transaction data set
  - Focus on Category_1 and Category _3
  - Correlations were found between levels in the two categories

Category_1 yes and no





Limitations and future study suggestions

- Feature selection process
  - PIMP with random Forest (Altmann et al, 2010)
- Revision on outlier detection methods
- More investigation on anonymous feature
  - Looking for similar data set
  - Matching data with more customer information to achieve more insights on customer behaviors

References

- Altmann, A., Tolosi, L., Sander, O., Lengauer, T. (2010). Permutation importance: a corrected feature
- importance measure, *Bioinformatics*, 26(10), 1340-1347.
- Hastie, T., & Qian, J. (2014). Glmnet vignette. Retrieved from http://www.web.stanford.edu/~hastie/Papers/Glmnet_Vignette.pdf. Accessed September 20, 2016.
- Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*. 2nd ed. New York: Springer.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: With applications in R*.
- MarketLine Industry Profile: Credit Cards in Brazil. (2019). *Credit Cards Industry Profile: Brazil*, N.PAG. Retrieved from http://search.ebscohost.com.ezp2.lib.umn.edu/login.aspx?direct=true&AuthType=ip,uid&db=buh&AN=134988260&site=ehost-live