

The Kaggle logo, featuring the word "kaggle" in a blue, lowercase, sans-serif font. Below the text is a blue, low-poly geometric shape resembling a crystal or a stylized mountain.

Kaggle Competition:

Elo Merchant Category Recommendation

PREDICT LOYALTY OF ELO CARD HOLDER THROUGH PURCHASE BEHAVIOR

Project Summary Report

Big Data Management
System & Tools

YUNFEI BAI

Introduction:

Elo is one of the largest payment brands in Brazil (<https://cartaoelo.com.br/>). It has many valuable data about their customer's purchase behaviors and Elo has built machine learning models to understand the most important aspects and preferences in their customers' lifecycle, from food to shopping. However, so far none of them is specifically tailored for an individual or profile. The purpose of this kaggle competition is to develop algorithms to uncover signal in customer loyalty, helping Elo reduce unwanted campaigns and create right experience for customers.

Objectives

Why this relates to big data? The reason applying big data approach to this competition is not only because this is for big data class project, but mostly because of the nature of the datasets. The data comes with five csv files, "historical_transactions.csv", "new_merchant_transactions.csv", "merchant.csv", "train.csv" and "test.csv". The total data volume is over 3GB, with more than 0.3 million customers' over 30 million transactions. It even cannot be read in python as a whole file in a 8GB RAM PC, you could imagine how slow it will be for the analytic works on a regular PC. Then it is quit nature to employ Hadoop and Spark platform to store and analyze it. I use Databrick as the platform for this solution. For community edition, I cannot upload a single file for more than 2GB, so I have to split biggest file, "historical_trasaction.csv" into two chunks to upload them into Databrick.

This is my first kaggle competition and happen to be my first big data analysis challenge with kaggle. Therefore, the purpose of this project is to participate this competition by employing what I have learned in this big data class, using scala, Spark MLlib pipeline and potential other advanced algorithms to predict the loyalty score for Elo customer base.

Feature Engineering and Exploratory Data Analysis:

After review the datasets for this project, the most relevant data is customer transactions. This consists of two part, historical and new transactions. First step is to combine these two datasets into a huge transaction dataset including the entire customer base. In addition, based this dataset, I will aggregate and derive more features. There all some null values existing in three variables, in Fig1. Therefore, in the second step, I filled those null columns with default values and also transform one variable "purchase_date" into "days_to_today" as a numerical feature.

category_3	merchant_id	category_2
30841311	30910695	28310783
null	null	2.1947917865782802
null	null	1.5316501657895172
A	M_ID_000025127f	1.0
C	M_ID_ffffc28eaa	5.0

Fig.1 Three columns that have null values.

In the third step, I aggregated this overall transaction table into card_holder level and engineered out 19 numerical and 7 categorical features. The details of the processing and feature engineering can be found in my databricks notebook (Appendix 2). All these features, overall, categorized customers' purchase behaviors from many aspects, covering purchase date, amount, installments, location (city/states), etc.

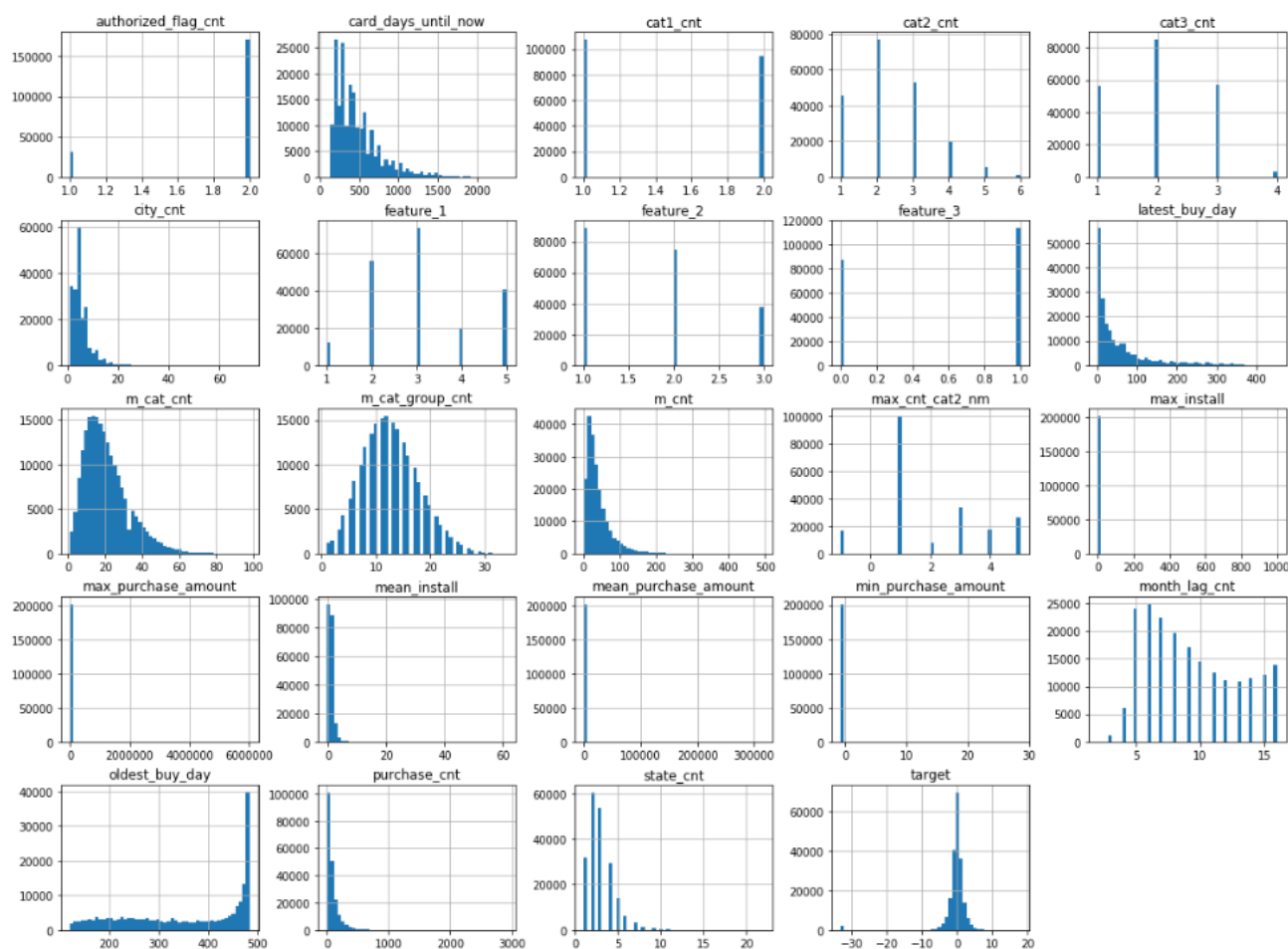


Fig.2 Histogram of engineered variables in dataset

In descriptive analysis, I further plot histogram of most engineered numerical variables in the dataset in Fig 2. Most continues-like numerical variables like card_days_until_now, m_cat_cnt, have decent distribution. Only some discrete variables have extreme skew distribution. Future we plot out correlation heat map for variables to check relationship between given two variables from dataset. In the Fig.3, some variables like purchase_cnt, m_cnt are highly correlated with each other, which suggests we may consider taking one out in the modeling step. For our target variable, however, we can see none of the independent variables is highly correlated to it; in fact, the correlation to the target variable is very low, which is interesting.

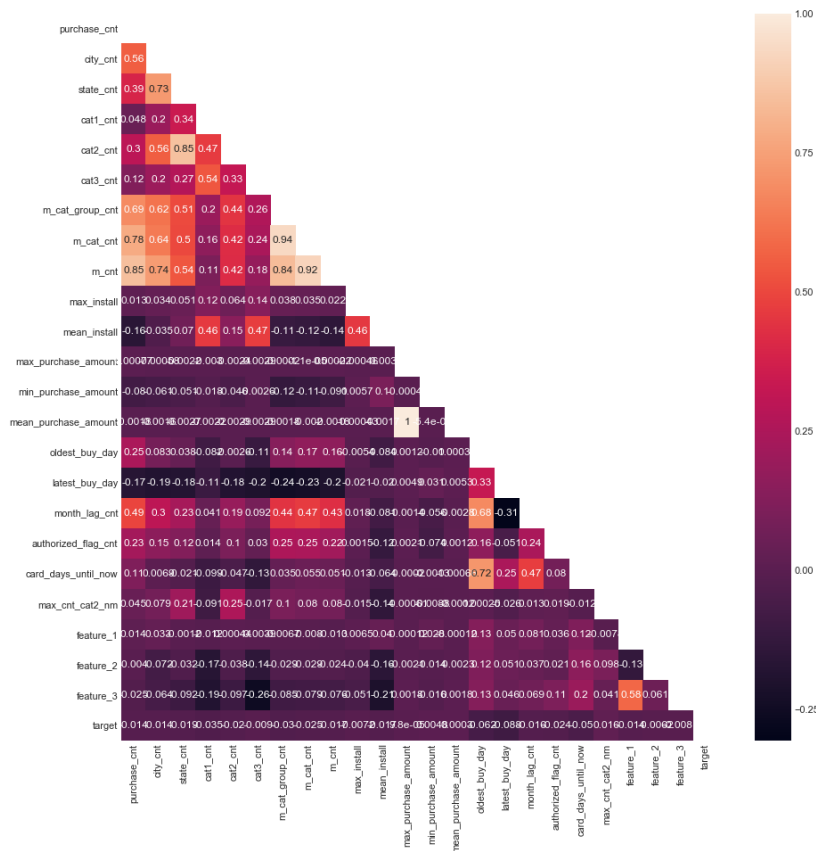


Fig.3 Correlation heatmap

Even appears to be a low correlation between target and independent variables, I further plotted cluster heatmap, in Fig 4. to reveal some cluster correlation out of these weak correlations. From the graph, the min_purchase_amount, feature_2, feature_3 and max_cnt_cat2_nm are more likely to correlate to target value.

After the exploration of the engineered variables, I then built a Spark MLlib pipeline to encode (one-hot encoding) categorical variables and finally ensembled all numerical and categorical variables into a single feature vector. I then split the data into 70% and 30% for training and testing purpose. This whole process in this section is also illustrated in Fig.5.

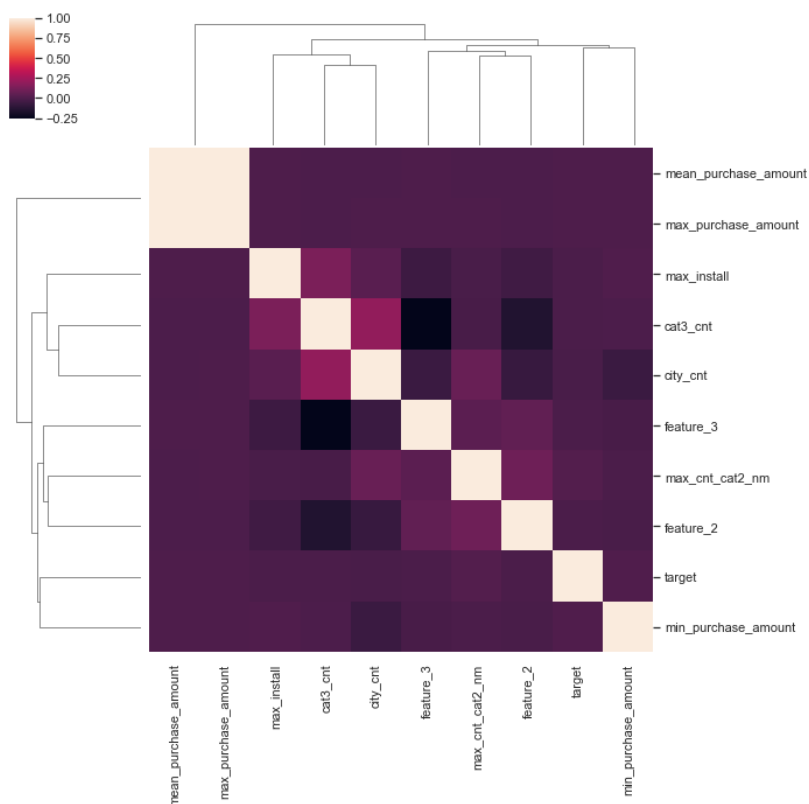


Fig.4 Cluster heatmap of variables related to target value

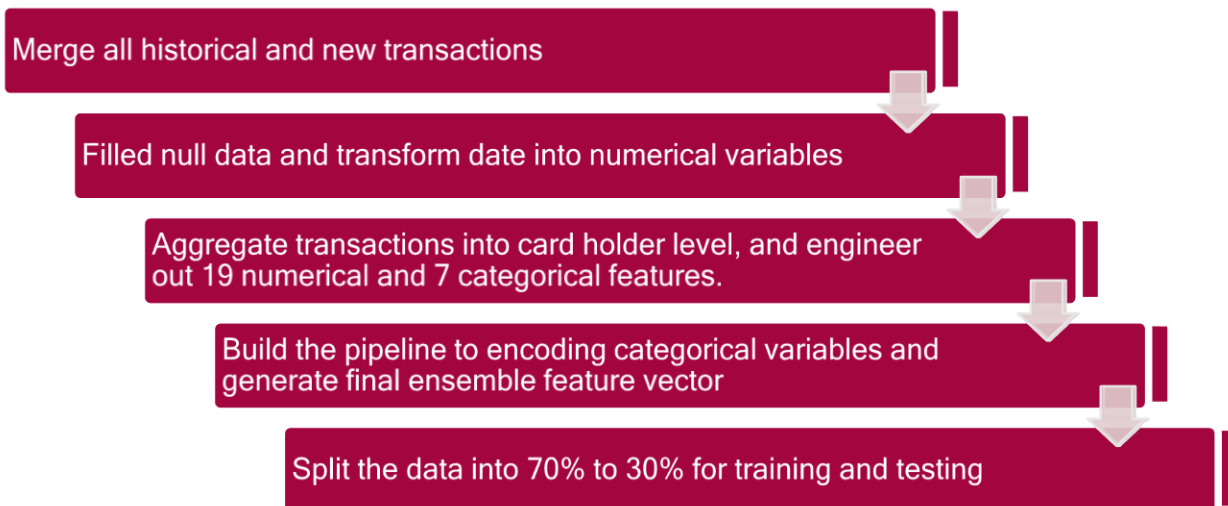


Fig.5 Feature engineering process.

Prediction Regression Models:

The ensemble feature vector is fitted into four algorithms, multiple linear regression, random forest regression and gradient boosting regression and XGBoost tree regression. The baseline models then are used to predict on the validation dataset (the 30%) to give us testing scores for these four algorithms.

Surprisingly, of all four models, XGBoost, the most advanced decision tree model for many other competitions is giving worst score in terms of RMSE. Linear regression is slightly better but still give a general poor score. Both the random forest regression and gradient-boosted tree regression give better testing scores of the four. Then in submission Leader Board on kaggle, I submitted these four models into four entries to compare them in terms of the final results. Both the XGBoost and linear regression give the poor scores as expected, aligning with the testing scores. On the other side, as well as expected, both random forest regression and gradient-boosted regression give better score, and of the two, gradient-boosted gives the lowest RMSE, which is best score of the four algorithm.

	Linear Regression	Random Forest Regression	Gradient-boosted Tree Regression	XGBoost Tree Regression
Testing Scores	RMSE : 3.748	RMSE : 3.699	RMSE : 3.702	RMSE : 3.778
LeaderBoard Submission Scores	RMSE : 3.900	RMSE : 3.851	RMSE : 3.838	RMSE : 3.938







Fig.6 Final score comparison between different regression models.

Conclusion:

This is my first kaggle competition, also happening to be my first big data analysis competition. A very careful data cleaning, feature engineering and detailed descriptive analysis for Elo Merchant Category Recommendation dataset was performed. With Spark MLlib pipeline and Spark XGBoost library, I finished a thoroughly model exploration among in total four regression models: multiple linear regression, random forest regression and gradient boosting regression and XGBoosting tree regression. Compared among these algorithms, both gradient boosting regression and random forest regression give us fairly good prediction. The multiple linear regression give us a fairly poor score, and surprisingly, the more advanced tree model, XGBoost approved to perform worst in this dataset.

Extra comment:

My current score is around 50% of all participants. Among the discussion in the competition board, the best scores so far are around 3.67, and are mostly coming from 5 fold lightGB model.

560	new	Geekdreams		3.835	6	4d
561	new	chanchino		3.837	2	6d
562	▼ 188	Tee Ming Yi		3.838	16	3d
563	new	Yunfei Bai		3.838	4	now
Your Best Entry ↑ You advanced 10 places on the leaderboard! Your submission scored 3.838, which is an improvement of your previous score of 3.851. Great job! Tweet this!						
564	▼ 346	Mikhail Novikov		3.838	1	10d
565	▼ 197	I'll be on the LB one day		3.839	6	1d

Appendix:

1. Elo competition on kaggle:

<https://www.kaggle.com/c/elo-merchant-category-recommendation>

2. My databricks notebook for this project:

<https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/7792953078525217/2896759813459456/4057496721065160/latest.html>

3. Github link for the documents:

<https://github.com/yunfeibai123/3252-Class-Project>