Olist Recommender System

SI 671

Students: Peiyan Wang, Yue Ma, Zhengyang Zhao

Advisor: Prof. Paramveer Dhillon

Introduction

More and more shopping and transaction processing move online with the development of e-commerce. Based on items and users' profiles, users' purchase history, and other information, recommender systems can help users to get personalized recommendations, increase sales, and enhance their shopping experience.

Objective

We focus on the users' ratings on the products to predict the rating of an unseen product for a user.

Methodology

Data Preprocessing

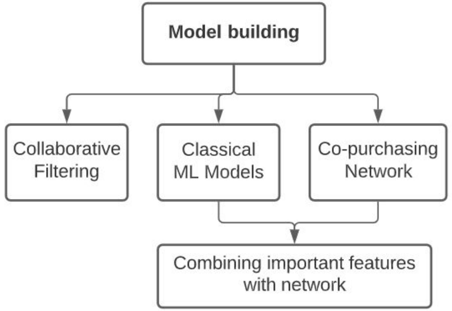
- Dataset combining
- Data quality check
- Matrix construction
- Network construction
- Feature extraction and engineering

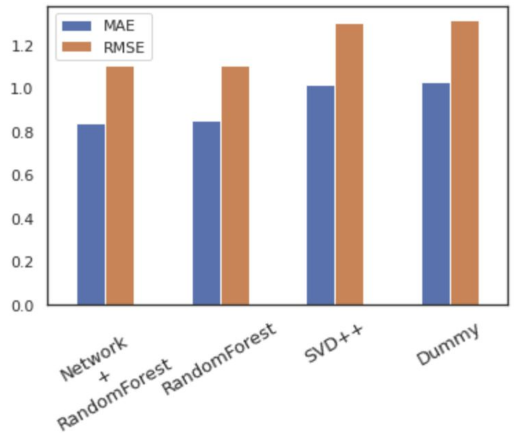
Algorithms

- Collaborative filtering
- Classic machine learning
- Network analysis + Machine learning

Model Performance

- Implemented collaborative filtering algorithms to predict ratings based on customer and product information, including SVD, SVD++,KNN, NMF, and Co-clustering, with best MAE of 1.01.
- Implemented classical ML algorithms including MLP, Ordinal Logistic, KNN, RandomForest, XGBoost, AdaBoost, GBDT, Linear Regression, Ridge, Lasso models. Random Forest Regressor has the best MAE of 0.85.
- Implemented Network and added extra features for model training with best MAE of 0.84 from Random Forest Regression model





Result and Conclusion

- The baseline Dummy Regression Model's MAE is 1.03.
- The best MAE score of classical machine learning models is 0.85 with Random Forest Model.
- The best performance is combining network features and Random Forest Model, with MAE of 0.84.
- The reason we don't get much improvement from collaborative filtering is that the range of ratings are relatively small(1-5), and over 80% ratings are over 4, indicating imbalanced dataset.

Olist Recommender System

Ma, Yue
School of Information
University of Michigan
maamber@umich.edu

Wang, Peiyan
School of Information
University of Michigan
peiyanw@umich.edu

Zhao, Zhengyang
School of Information
University of Michigan
zzyang@umich.edu

Abstract

E-commerce is going wild around the world. Building a recommender system would help the company to provide customers more related options for purchasing, and ultimately, increase the sales income of the company. The main objective of the project is to predict review scores with an optimal prediction model. In this project, matrix methods, classical machine learning and a combined model with network analysis are utilized to predict the customers' review scores. We found that the most important features that contribute to the prediction results are mainly focused on whether orders are delivered on time and the freight value. SVD++ model, classical Random Forest Regressor, Network combined Random Forest Regressor are the best models from the three approaches. Among them, the combined model has the best performance with MAE as low as 0.84, increasing by 18% from the dummy regressor.

1 Introduction

More and more shopping and transaction processing move online with the development of e-commerce. Companies have built and are building recommender systems to provide users more options to review, in order to increase the likelihood of the user purchasing the recommended item(s). Based on products/items, users' profiles, users' purchase history, and other information collected by e-commerce, a recommender system can help users to get personalized recommendations, and benefit the company as well, including increasing sales outcome, and providing users an enhanced shopping experience. Thus, a dataset that includes all of the information mentioned above is required.

We wonder how we can perform data manipulation, analysis, feature analysis, and model building on an ecommerce dataset. Therefore, we found an ecommerce trailer in Brazil named Olist has a dataset that meets all of our needs so that we can utilize multiple approaches to conduct prediction of product/item ratings. Additionally, based on users' behavior on an e-commerce platform, we can do comparison between the performance of different recommender algorithms.

2 Related works

Because of massive data generated from social media, e-commerce of businesses over the last decades, recommendation systems have appeared in various industries after the penetration of internet services.Collaborative filtering and machine learning techniques are used for suggesting products of interest to users.(V. Subramaniyaswamy,2017). Many companies are using recommender systems, such as Amazon (Linden et al., 2003), Netflix (Koren, 2009a), Google (Das et al., 2007), and Facebook (Shapira et al., 2013).

Currently, recommender systems are categorized into content-based systems and collaborative filtering systems. Content-based systems use content to build models. Collaborative filtering systems are based on customer ratings of products regardless of the content. Collaborative filtering is more

Preprint. Under review.

popular in businesses because of their simplicity and high performance levels (Bobadilla et al., 2013, Shi et al., 2014, Su and Khoshgoftaar, 2009).

User-based collaborative filtering systems recommend products based on similar customers, which is computed by theri ratings of products that both have rated (Konstan et al., 1997, Shardanand and Maes, 1995). Item-based collaborative filtering systems utilized similarities between products, which is computed by cosine similarity and conditional probability based similarity (Karypis, 2001).

In this paper, we would like to utilize collaborative filtering methods as well as hybrid approach of recommender methods.

3 Dataset

The datasets we used are the real commercial data from Olist, the largest department store in Brazilian marketplaces. The datasets have information of 100k orders from 2016 to 2018 made at multiple marketplaces in Brazil. The datasets consist of nine interactive tables and are deposited in Kaggle. The relationship between the 9 tables are shown in figure 1. In this project, we mainly used tables containing the information of payments, orders, reviews, customer, items, products, and categories name.

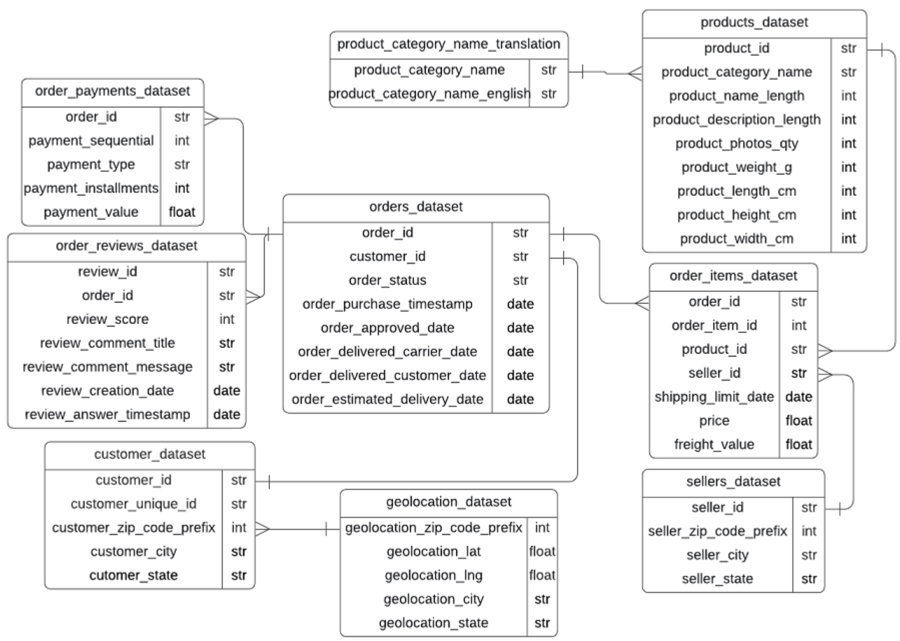


Figure 1: Relationship between the tables of Olist's commercial dataset.

4 Methodology

For our project, we focus on the users' ratings on the products to predict the rating of an unseen product for a user. In order to figure out the best recommendation algorithm, we tried different approaches including classical Machine Learning algorithms, network analysis, and collaborative filtering. We would like to compare the performances of different algorithms and approaches with the baseline model to figure out the best recommendation algorithm.

4.1 Data Pre-processing

For those three approaches, we utilized different pre-processing steps.

4.1.1 Collaborative Filtering

In order to use collaborative filtering to predict the review rates of one product by one customer, we mainly focus on customer_id, product_id, and review score. After reading in the csv files and merging the tables, we check the data quality, including completeness(no missing value, so no further processing), uniqueness,and conformity.

Firstly, we removed the duplicate products in the same order(such as 3 pens in an order), which can't provide more information. Secondly, we utilized the average reviewing score as the product's reviewing score by one customer, because one customer may buy and rate the same product multiple times.

4.1.2 Classical ML Method

We combined 6 tables for rating prediction including order_reviews, orders, order_items, order_payments, products, and product_category. Besides the existing features, we also add additional features that may affect the ratings. 'Actual_delivery_duration' and 'estimated_delivery_duration' are the actual and estimated delivery time consumption, respectively. 'Diff_estimated_actual_delivery' is the difference between the estimated and actual delivery time. 'Is_late' records if the product is delivered to the customer in time. 'If_comment' means if the customer added a comment to the product. We also added other features such as the total freight value (tot_freight_value), number of products ('num_of_products'), number of sellers ('num_of_sellers') , number of reviews ('tot_reviews') of one order, times of instalments ('payment_times'), the number of the orders containing the product ('product_tit_orders').

Additionally, we drop the features that are not useful for prediction such as 'order_id','product_id', 'payment_type', etc. We also drop the samples with missing values. For the categorical variables, we used LabelEncoder to transform them into numerical variables. Now, the dataset is ready for machine learning prediction.

Since some machine learning models are sensitive to the scale of the values, we used MinMaxScaler and StandardScaler to normalize the dataset.

4.1.3 Network Analysis

Within the combined datasets, we first subtracted orders with more than one product/item and removed orders having multiple same items. We then treated these linked items as nodes and built a network dataset with these nodes while without directions at the same time. In order to have more understanding on this newly built network dataset, we used networkx to do some analyses. We calculated clustering coefficients for nodes, degree centrality, closeness centrality, betweenness centrality, and degree for nodes. Then we treated these calculated values as our new features for the combined dataset, and for those items that don't have any links with other items will have zeros for these network features.

Additionally, we kept some original features that are critical for prediction, such as order duration, item prices, and did modification of these features to realize feature extraction.

4.2 Model Building

4.2.1 Collaborative Filtering Recommender System

1. We built the initial recommender engine using SVD, SVD++,KNN, NMF, and Co-clustering. Among those algorithms, SVD++ has the lowest MAE and RMSE score without tuning the parameters.
2. Tuned parameters step by step. However, turning the parameters doesn't improve the performance a lot.
3. Utilized RMSE/MAE and cross validation to evaluate the model.
4. The best result among those collaborative filtering algorithms is.

The reason we don't get much improvement from tuning hyperparameters is that the range of ratings are relatively small(1-5), and over 80% ratings are over 4, indicating imbalanced dataset.

4.2.2 Classical Machine Learning Models

We split the dataset into train_validation and test dataset with test size as 0.2. The train_validation dataset was split into train and validation dataset with the same test size.

To build a baseline, we trained the dummy regressor with mean ratings as predictions. Besides, we trained and validated the performance of 10 machine learning models including KNN, XGBoost, AdaBoost, GBDT, RandomForest, LinearRession, Lasso, Ridge, MLP, and Ordinal Logistic Regression models.

4.2.3 Combining Network features and Machine Learning Models

For combination of network features into machine learning models, we didn't take all original features, instead, we selected most relevant features, like product price, customer location, delivery duration, and added modified features, like whether the delivery is late, and whether the order has a review. Overall, we took 14 features for model training, and applied 10 machine learning models including KNN, XGBoost, AdaBoost, GBDT, RandomForest, LinearRession, Lasso, Ridge, MLP, and Ordinal Logistic Regression models.

4.2.4 Hyper-parameter Tuning

We selected two models, RandomForest and GBDT regressors, to tune the parameters. GridSearchCV is used. 'n_estimators' and 'min_samples_leaf' are tuned for Random Forest Regressor. 'n_estimators', 'min_samples_leaf', and 'max_depth' are tuned for the GBDT regressor.

4.3 Evaluation Method

MAE and RMSE are used as evaluation metrics to measure the performance of models. For classical machine models, we built a dummy regressor as a baseline to compare with other models.

5 Analysis of results

5.1 Most of review scores are over 4

We randomly selected four groups of 1000 samples from our pre-processed datasets to grasp the essence of the distribution of the review score among customers. By encoding samples with Altair library, we found that around 80% of customers rate their products with review scores over 4.

5.2 In-time delivery and actual delivery time consumption are important features for prediction

We extracted and added features by feature engineering and measured the absolute correlation between the features and customer ratings. As shown in figure 3, in-time delivery (is_late) and delivery time consumption (actual_delivery_duration) are top 2 important features to predict customer ratings. Late delivery and long time consumption lead to low ratings. In practice, we could recommend the Olist company to reduce the delivery time and send the products on time to increase the customer satisfaction based on our analysis.

5.3 Random Forest Regressor has the best performance among 10 classical machine learning models

After the feature engineering, we trained ten classical machine learning models and one dummy regressor to predict the customer ratings. Figure 4 indicated the performances, measured by MAE and RMSE, of the regression models. The Random Forest Regressor has best performance with MAE as 0.86 and RMSE as 1.11. MAE increased by 18% and RMSE increased by 16% compared to the dummy regression model's performance.

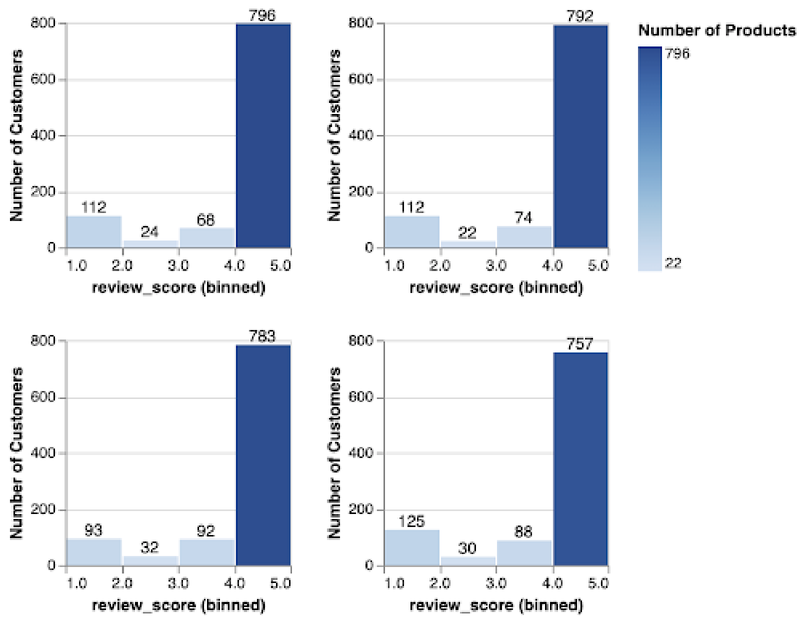


Figure 2: The distribution of reviewing score in randomly selected four groups of 1000 samples.

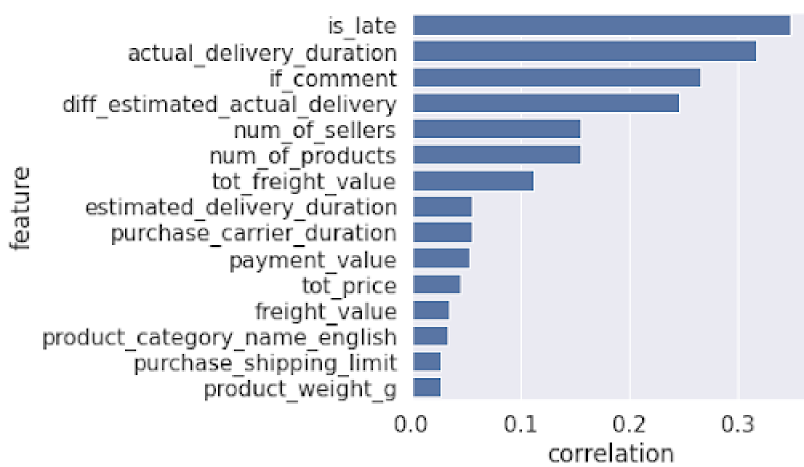


Figure 3: The correlation between top 15 features and customer rating prediction.

5.4 Adding extra features calculated based on the dataset itself can improve the prediction model performance

With different methods of feature engineering, the performance of the same prediction model could also be different. For our feature engineering with network analysis, same as classical feature engineering, the best prediction model is Random Forest, based on the illustration of figure 5. The Random Forest Regressor has the best performance with MAE as 0.84 and RMSE as 1.10. Compared to the dummy regression model, MAE decreased by 19%, RMSE increased by 17%. After tuning hyperparameters on the Random Forest Regressor, MAE decreased by 21% to 0.82, RMSE decreased by 19% to 1.08.

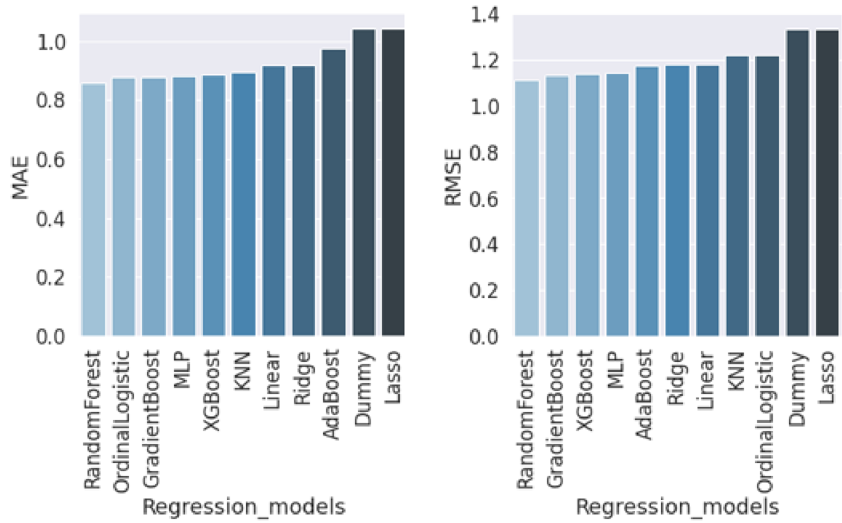


Figure 4: MAE and RMSE scores of the classical machine learning models predicting customer ratings.

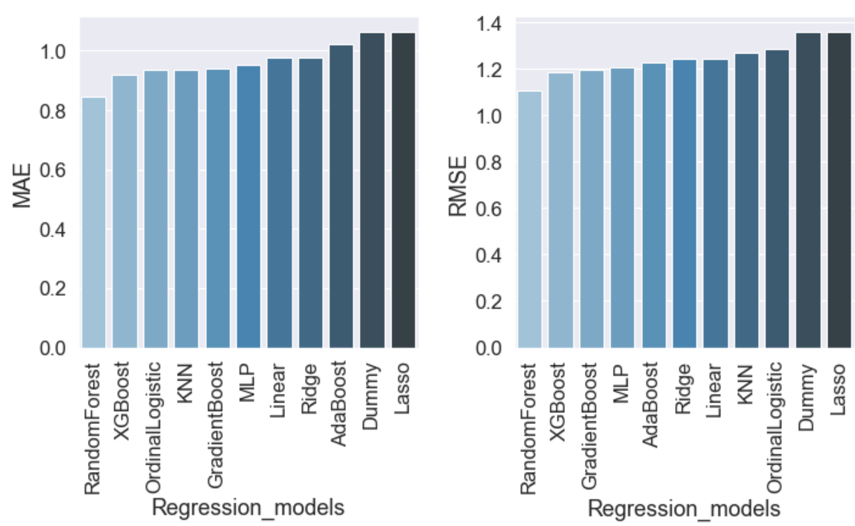
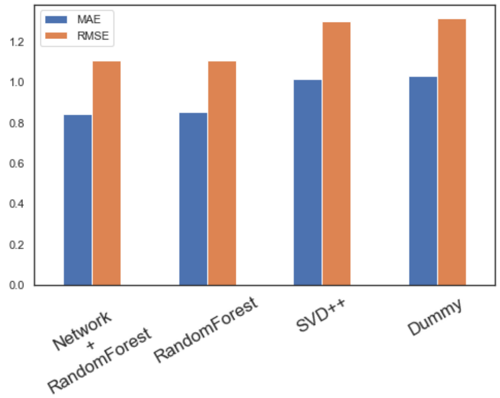


Figure 5: MAE and RMSE scores of the machine learning models collaborated with network analysis predicting customer ratings.

6 Discussion and Conclusion



Based on the chart, we can see that combining network features with Random Forest Model leads to the lowest MAE and RMSE. The reason we don't get much improvement from collaborative filtering is that the range of ratings are relatively small(1-5), and over 80% ratings are over 4, indicating imbalanced dataset. In the real-world, the ratings from customers probably are not balanced and cleaned, so we would like to consider the customer's co-purchasing preference at the same time. Collaborative filtering is not necessarily the best method to predict the ratings from customers, the best model will be based on real-world situations.

If we have longer time, we will try the following steps:

(1) We will also try content-based filtering, using features including product categories, price, freight to predict the customers' review ratings.

(2) We would like to build a User Interface to recommend potential products to customers.

(3) Based on network analysis, we could solve the cold start problem by recommending popular products to new customers.

(4) Because around 80% of rating scores are higher than 4, the distribution of rating scores is extremely left-skewed, which may cause high residual. We plan to upsample the records with ratings lower than 4 or downsample the records with ratings higher than 4 to build a balanced dataset. Or we could use log transformation for ratings to adjust the distribution.

References

[1] A Personalised Movie Recommendation System Based on Collaborative Filtering."International Journal of High Performance Computing and Networking, www.inderscienceonline.com/doi/pdf/10.1504/IJHPCN.2017.083199.

[2] Linden et al., 2003 G. Linden, B. Smith, J. York Amazon.com recommendations: item-to-item collaborative filtering IEEE Internet Comput., 7 (2003), pp. 76-80

[3] Koren, 2009a Y. Koren The BellKor Solution to the Netflix Grand Prize NetFlix, Los Gatos, CA (2009)

[4] Das, A.S., Datar, M., Garg, A., Rajaram, S., 2007. Google news personalization: scalable online collaborative filtering. In: Proceedings of the 16th International Conference on World Wide Web. ACM Press, New York, NY, pp. 271–280.

[5] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez Recommender systems survey Knowl.-Based Syst., 46 (2013), pp. 109-132

[6] Karypis, G., 2001. Evaluation of item-based top-N recommendation algorithms. In: Proceedings of the 10th International Conference on Information and Knowledge Management. ACM Press, New York, NY, pp. 247–254.

[7] J.A. Konstan, B.N. Miller, D. Maltz, J.L. Herlocker, L.R. Gordon, J. Riedl GroupLens: applying collaborative filtering to Usenet news Commun. ACM, 40 (3) (1997), pp. 77-87

[8] B. Shapira, L. Rokach, S. Freilikhman Facebook single and cross domain data for recommendation systems User Model. User-Adap. Inter., 23 (2013), pp. 211-247

[9] Thorat, Poonam B., et al. "Survey on Collaborative Filtering, Content-Based Filtering and Hybrid Recommendation System: Semantic Scholar." Undefined, 1 Jan. 1970

[10] Bobadilla, Jesus, et al. "A framework for collaborative filtering recommender systems." Expert Systems with Applications 38.12 (2011): 14609-14623.

Appendix

According to the rubric of final project report, we discussed in the following corresponding part:

Problem Statement Motivation -> Introduction

Design of the methodological approach to solve the problem - Methodology

Design and thoroughness of experiments -> Methodology

Analysis of results -> Analysis of results

Discussion and Conclusion -> Discussion and Conclusion