Işıksu Ekşioğlu

Problem and Solution

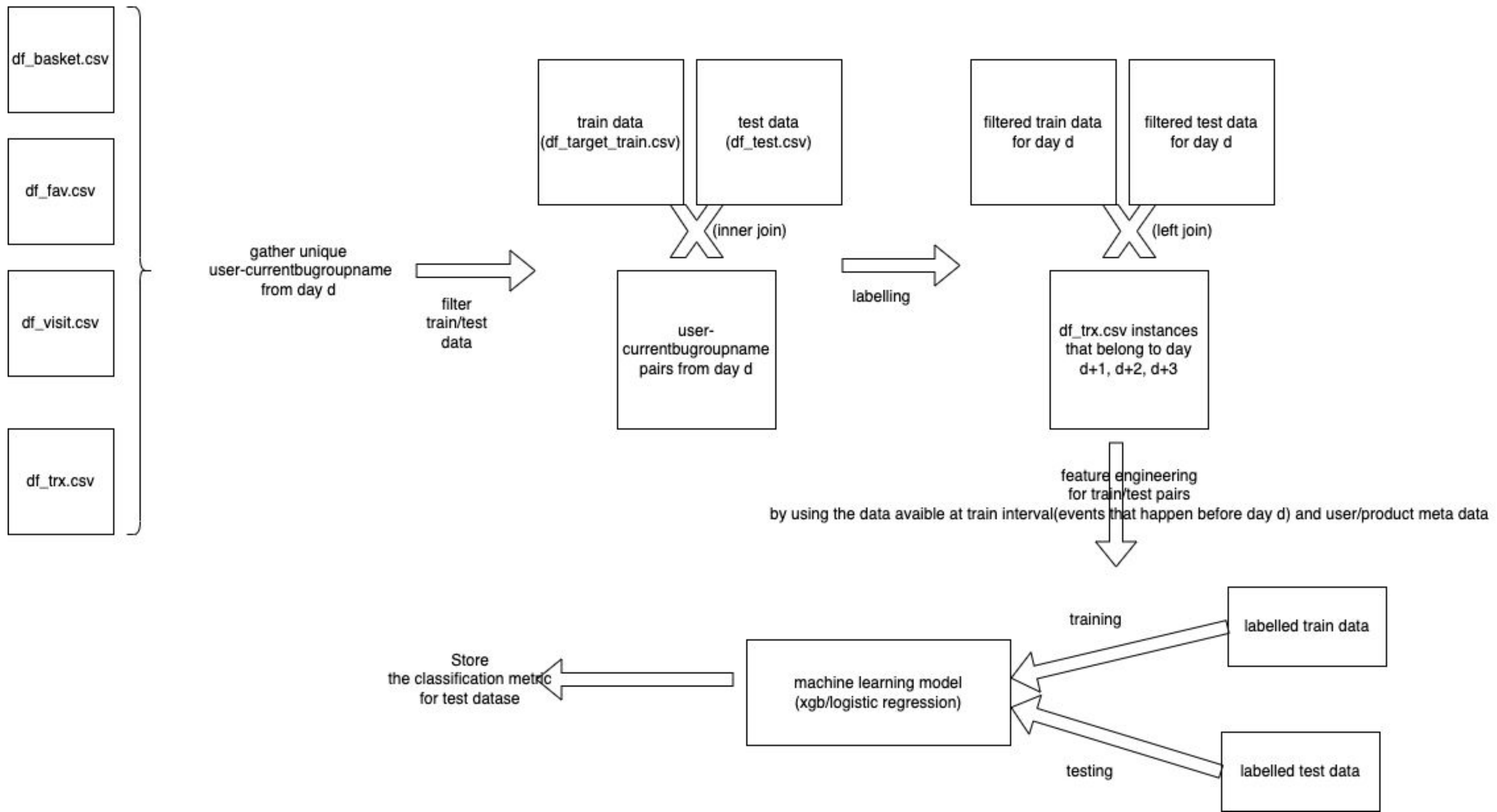
- Problem:
    - Build a model(s) that can predict the probability of a customer to order a product from the given currentbugroupname in the next 3 days (test period)
        - My paraphrasing: Given user-currentbugroupname from day d, predict whether the user in the pair will purchased an item from the currentbugroupname in day d+1, d+2, d+3
        - It is a binary classification problem

- Solution
    - Create binary classification dataset from user-currentbugroupname pairs in train/test dataset for each day in the data collection
    - Train/test machine learning model to solve the binary classification problem
    - Store classification metric results for test split for each day
    - Present aggregation of classification metrics from each day as result

Methology

- Train and test dataset include all user-currentbugroupname interaction pairs from October 2020
    - Prediction algorithm for test instances that belong to day d:
        - Create binary classification dataset from user-currentbugroupname interaction pairs for day d
            - Gather pairs for day d from train/test datasets (Gather train/test instances)
            - Label pairs by using df_trx.csv as follows:
                - For a user-currentbugroupname pair from day d
                    - Label it as positive if there is a record for this pair at d+1, d+2, d+3 in df_trx.csv
                    - Label it as negative if there isn’t any record for this pair at d+1, d+2, d+3 in df_trx.csv
            - Make feature engineering for user-currentbugroupname pairs from day d
        - Train machine learning model with created train dataset
        - Test trained machine learning model with created test dataset
    - Repeat the above procedure for each day d from October 2020
    - Store classification metrics for test dataset for each day d from October 2020
    - Present mean and standard deviation of classification metrics as result

Prediction algorithm for Day d

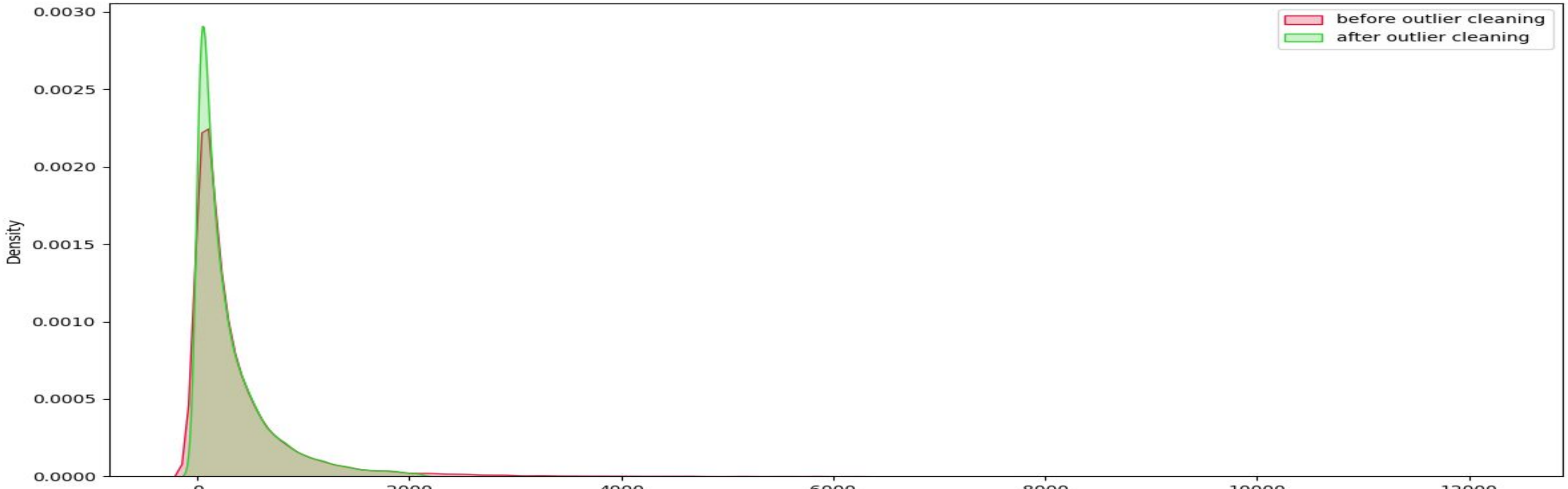

df_basket.csv
df_fav.csv
df_visit.csv
df_trx.csv
gather unique user-currentbugroupname from day d
filter train/test data
train data (df_target_train.csv)
test data (df_test.csv)
(inner join)
user-currentbugroupname pairs from day d
labelling
filtered train data for day d
filtered test data for day d
(left join)
df_trx.csv instances that belong to day d+1, d+2, d+3
feature engineering for train/test pairs by using the data avaible at train interval(events that happen before day d) and user/product meta data
training
labelled train data
testing
labelled test data
machine learning model (xgb/logistic regression)
Store the classification metric for test datase

Preprocessing

- Drop duplicates from all the datas available
- Drop instances if any of the userid, contentid , date, currentbugroupname is nan from all the datas available
- Eliminate instances from df_trx data if their quantity/price features 0 or negative
- Eliminate instances from df_basket data if their addtobasket_count feature 0 or negative
- Eliminate instances from df_fav data if their fav_count features 0 or negative
- Eliminate instances from df_visit data if their productdetailcount features 0 or negative
- Eliminate instances from df_demo if their age 0 or negative
- Fill the nan values in age/gender columns of df_product data with most frequent elements
- Delete the instances from event datasets if their userid is nan or contentid’s currentbugroupname is nan
- Detect users that have extreme behaviour (outlier analysis)

Outlier Analysis

- Each users' addtobasket, fav, visit, purchased record counts were calculated
- using the z-score formula to remove all those users which surpasse 3 standard deviations in the Standard Normalized Distribution
- Plots that show outlier elimination effect and distribution difference between normal users and extreme users can be found in the given link
    - Link: https://github.com/noeldar/trendyol_case/blob/main/preprocess.ipynb
    - As can be seen there is a significant difference at records' counts between normal users and extreme users. Below plot can be given as an example that shows eliminating outliers handles extreme cases



Feature Engineering

- Feature names and their explanations:
  - Same_gender: binary feature that shows whether busgroup contains items that have the same gender with the user in the user-busgroup pair
  - Same_age: binary feature that shows whether busgroup contains items that have the same age category with the user in the user-busgroup pair
  - Price_ratio: user avg price in training interval / busgroup avg price in training interval for a user-busgroup pair
  - User_busgroup_avg_productdetailcount_ratio: User_busgroup pair avg productdetailcount in training interval / the busgroup avg productdetailcount in training interval
  - User_busgroup_avg_fav_count_ratio: User_busgroup pair avg fav_count in training interval / the busgroup avg fav_count in training interval
  - User_busgroup_avg_addtobasket_count_ratio: User_busgroup pair avg addtobasket_count in training interval / the busgroup avg addtobasket_count in training interval
  - User_busgroup_avg_quantity_ratio: User_busgroup pair avg quantity in training interval / the busgroup avg quantity in training interval
  - Unique_userid_fav_ratio: unique userid count that favored items that are in busgroup/unique userid count that visit busgroup
  - Unique_userid_basket_ratio: unique userid count that added items to basket that are in busgroup/unique userid count that visit busgroup
  - Unique_userid_trx_ratio: unique userid count that purchased items that are in busgroup/unique userid count that visit busgroup
  - Basket_records_ratio: record count that added items to basket that are in busgroup/record count that visit busgroup
  - Fav_records_ratio: record count that favored items to basket that are in busgroup/record count that visit busgroup
  - Trx_records_ratio: record count that purchased items to basket that are in busgroup/record count that visit busgroup

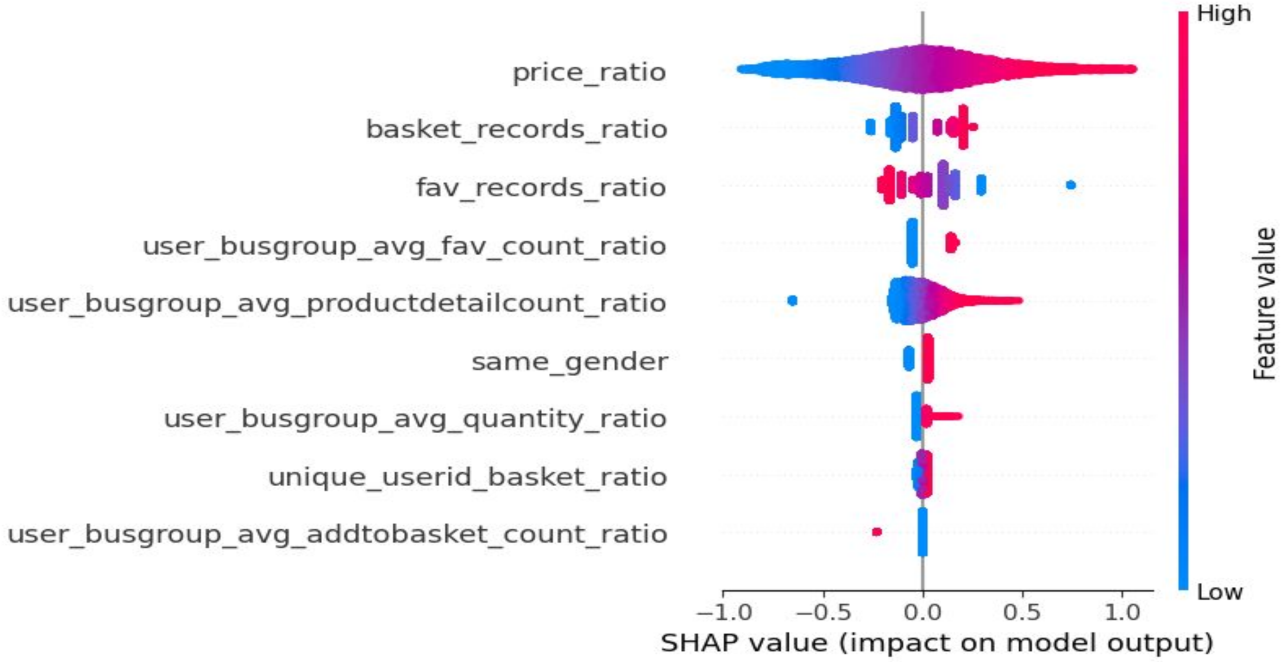
Models and Results

- Models:
  - Logistic regression
  - XGBoost
  - Experimented with linear and non-linear model
- Results

	Logistic Regression	XGBoost
AUC	0.646	0.593
F1	0.841	0.840
Avg Precision	0.162	0.14

Future Importance Analysis

- Feature Importance calculated by SHAP method



Insights

- According to the feature importance analysis, the situations that most affect the purchase of items from the busgroup are the popularity and price of the busgroup.

- Features showing users' interest in bus groups did not affect the prediction much. Therefore, the interests of the users in the discussed dataset vary according to the characteristics of the busgroup.

- We have a low average precision score but a high F1 score
    - it may indicate that our models are correctly identifying true positives (leading to a high F1 score) but is also generating a high number of false positives, which lowers the average precision score.

Future Work

- Low average precision score needs to be handled
    - Analysing probability distributions of false positives and adjust models’ thresholds
    - Neural networks models can be used since different combination of the generated features and upsampling didn’t increase the metric
        - Both trained models aren’t successful enough to predict positive class instances
- Addition of search term dataset as feature to understand users’ interests