Elo Merchant Category Recommendation

This project is intended to help understand customer loyalty and build a recommendation engine with discount from credit card provider

Introduction

ELO, one of the largest payment brands in Brazil, has built partnerships with merchants in order to offer promotions or discounts to cardholders.

Data is at https://www.kaggle.com/c/elo-merchant-category-recommendation/data

Problem Statement –

Build machine learning model to predict loyalty score for card id's in test dataset and tailor recommendations for an individual or profile, create right customer experiences. This will help client reduce unwanted campaigns.

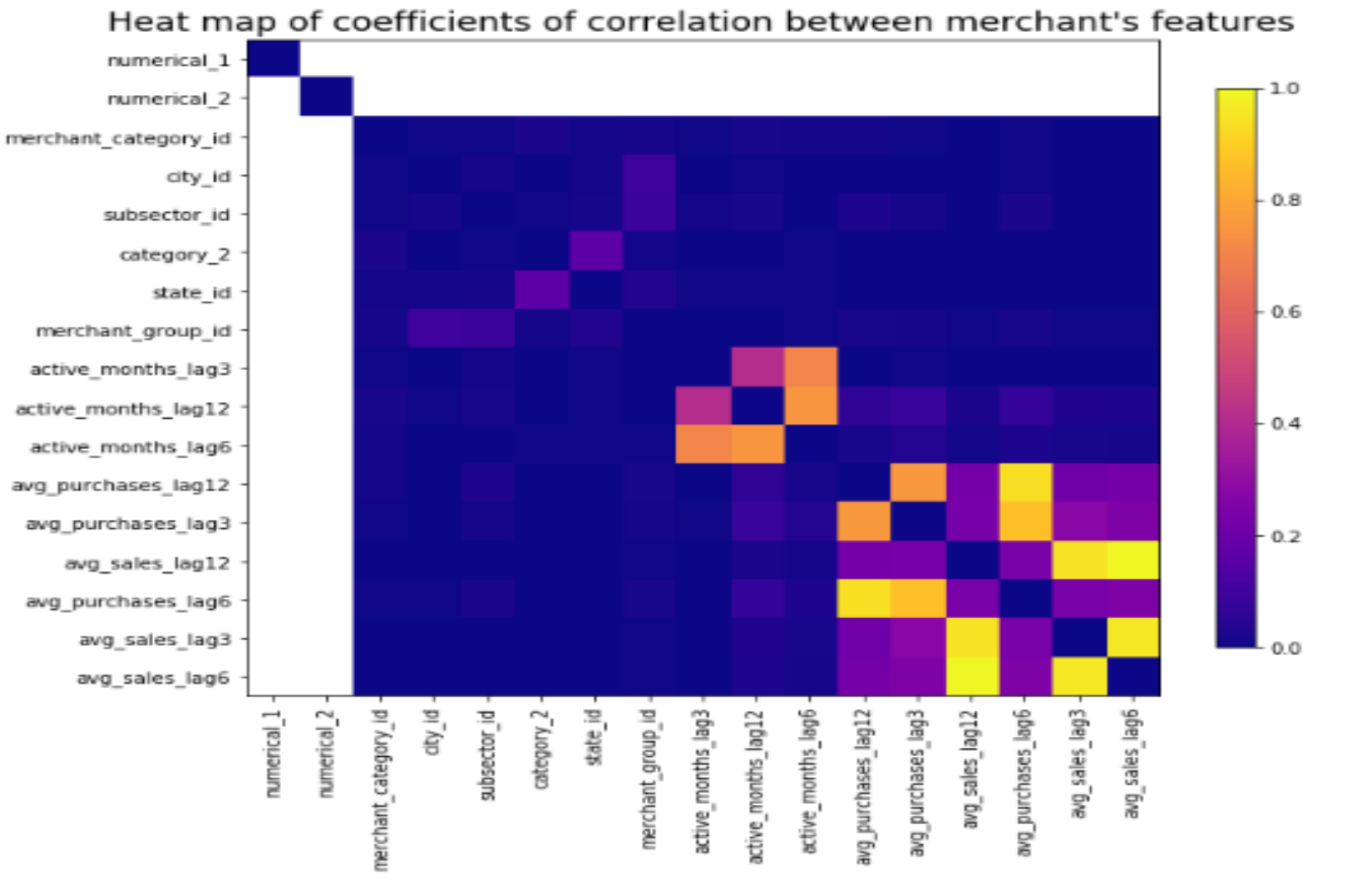
Objective

Steps performed in this project are outlined below

- **Clean the data –** Data cleaning techniques impute missing data, 3-sigma impute for outliers
- **Perform EDA -** Visual insights into data and correlation
- **Perform Feature Engineering -** To create Features which will help to increase predictive power of Machine Learning Algorithms
- **Build Machine learning Model -** Machine learning algorithms used and methods applied to predict the model
- **Conclusion and recommendations**

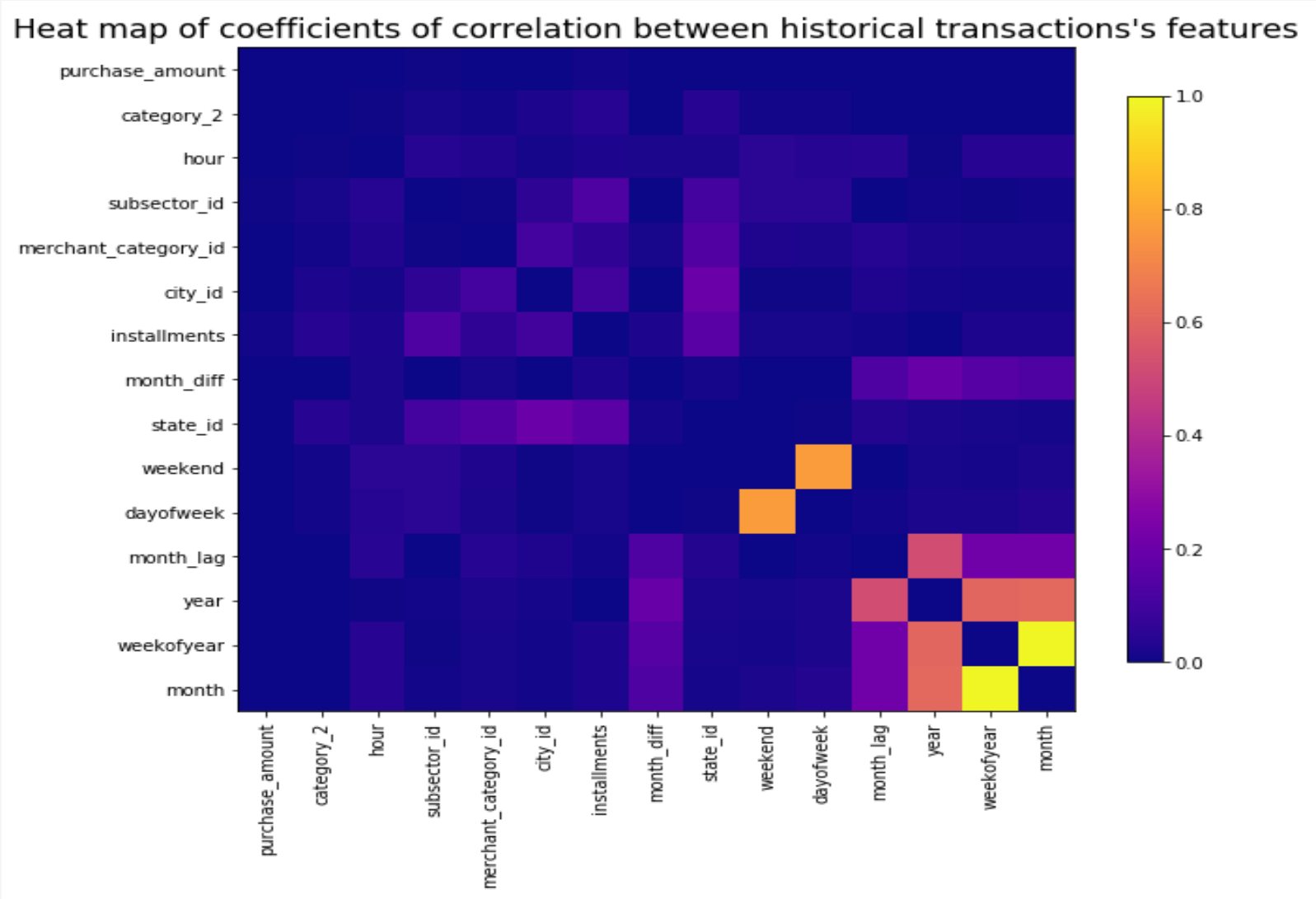
EDA

- There is no corelation numerical_1 and numerical_2 feature.
- There is correlation between avg_sales and avg_purchases of 3, 6 an 12 month.



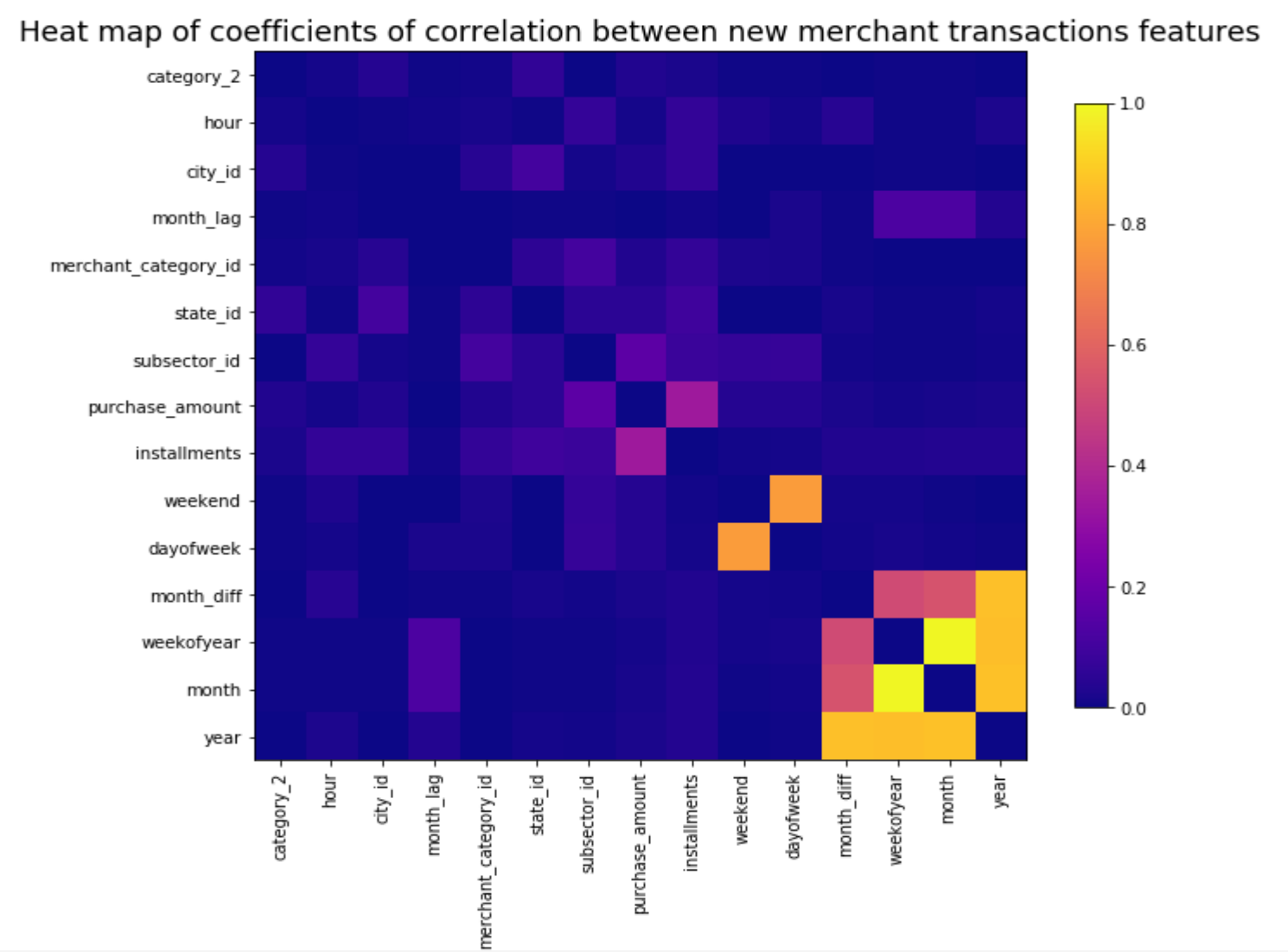
EDA

There seems to be no correlation between features.



EDA

There seems to be a correlation purchase amount and number of installments.



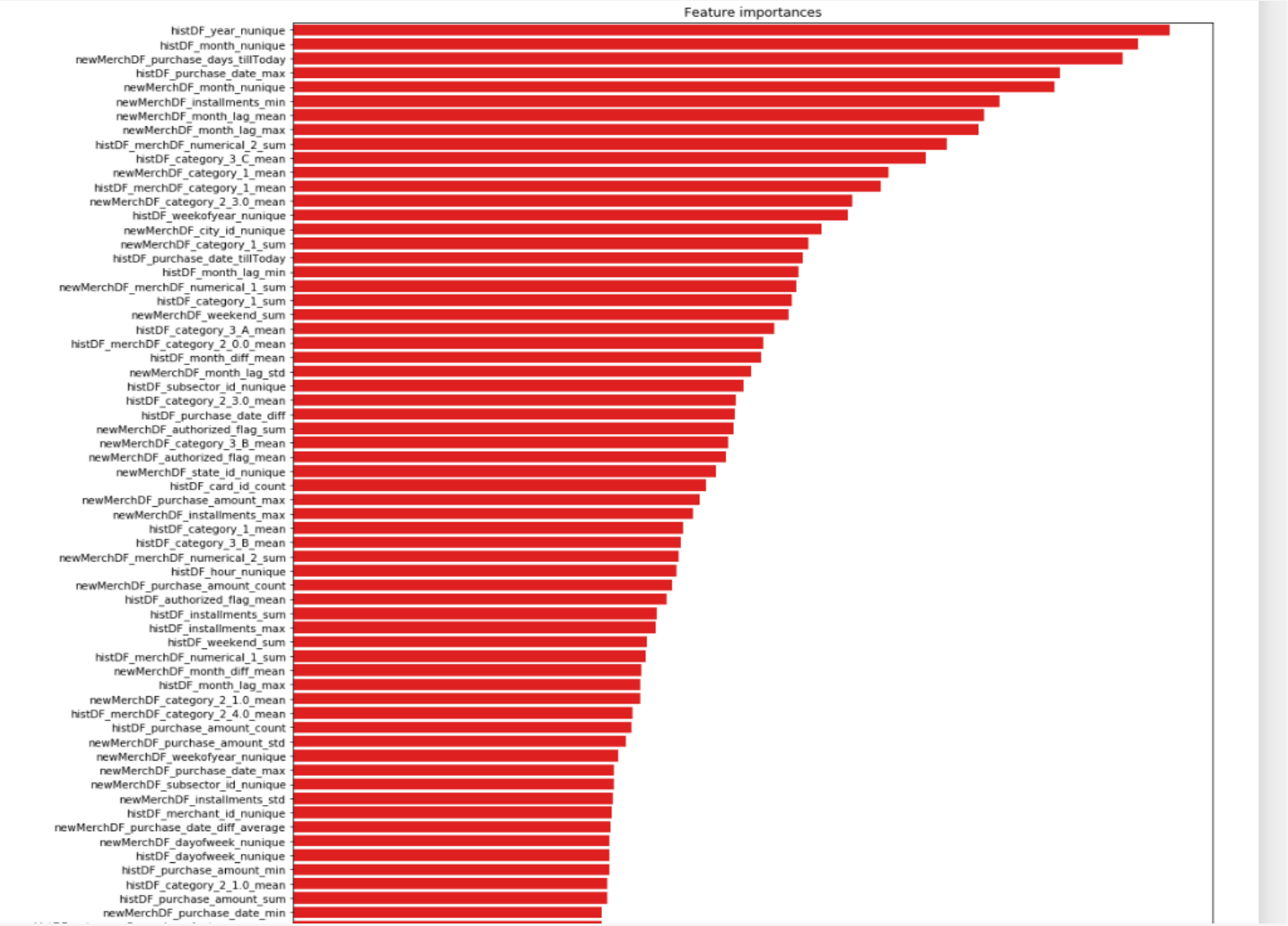
Feature engineering and Machine Learning Model

General process followed for featuring engineering is

1. One hot encoding is applied to categorical features to **merchant.csv**, **historical_transactions.csv and new_merchant_transactions.csv.**
2. Categorical features and anonymized in **merchant.csv** are merged to **historical_transactions.csv** and **new_merchant_transactions.csv**
3. Aggregate functions (mean, count, sum, nunique) are applied to datasets **historical_transactions.csv and new_merchant_transactions.csv** by grouping by card_id.
4. Datetime features are added to aggregated Data Fames.
5. Aggregated Data Fames are merged with train and test data
6. Datetime features are added to merged **train** and **test** data frame and outlier feature is added to **train** data frame to handle outliers.
7. Training data is trained on **XGBOOST** ML algorithm
8. **RandomizedSearchCV** is used for tuning **XGBOOST** algorithm hyperparameters
9. **Tuned Hyperparameters** are **n_estimators** - 100, **max_depth** - 8, **min_child_weight** - 32, **gamma** – 0.2, **colsample_bytree**- 0.2, **colsample_bylevel** – 0.6
10. **RMSE** is used as evaluation metric, is calculated on target and values predicted from train dataset, which is **3.38569**
11. Feature importance is generated on the trained model.

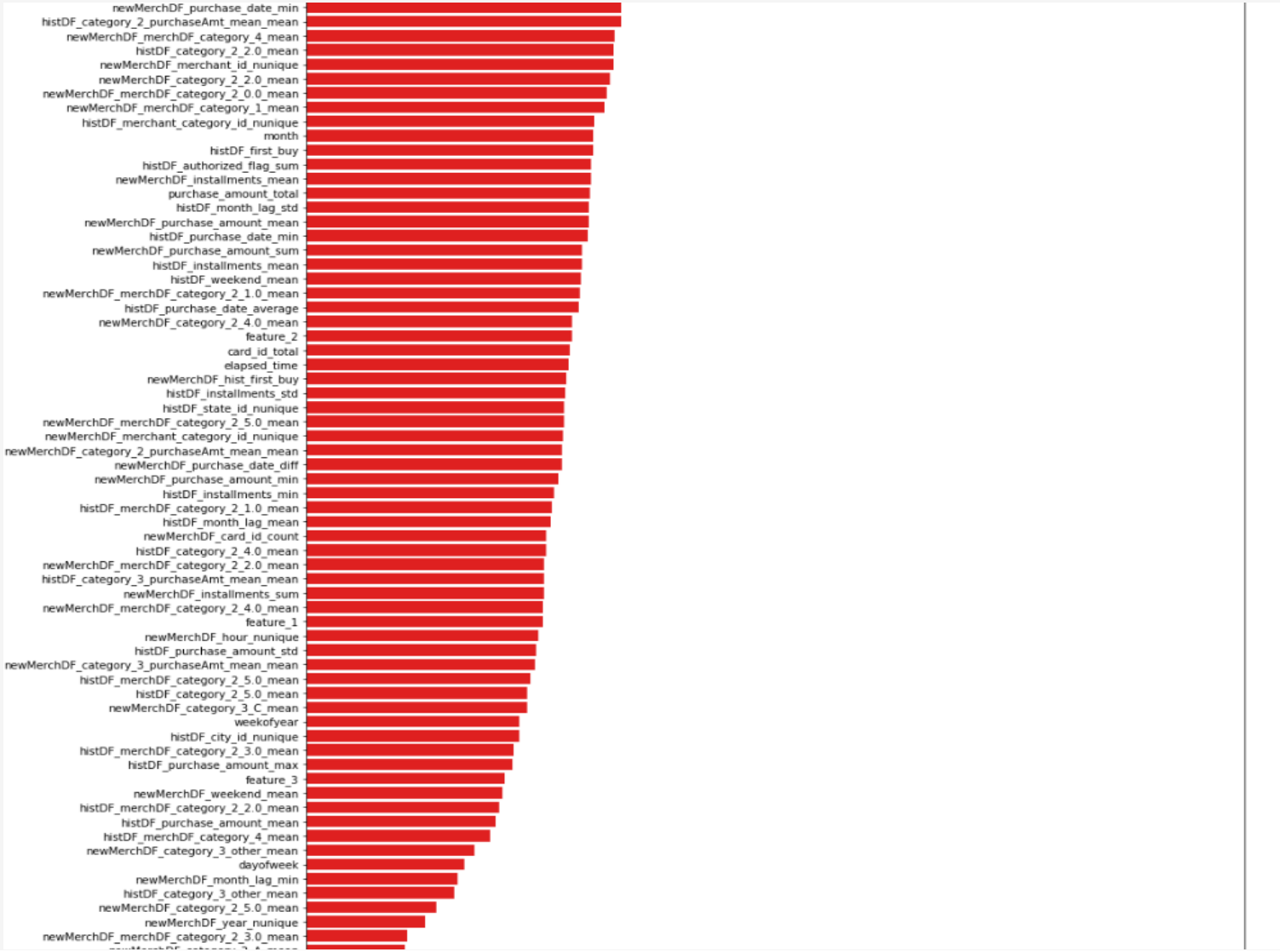
Machine Learning Model

Feature Importance



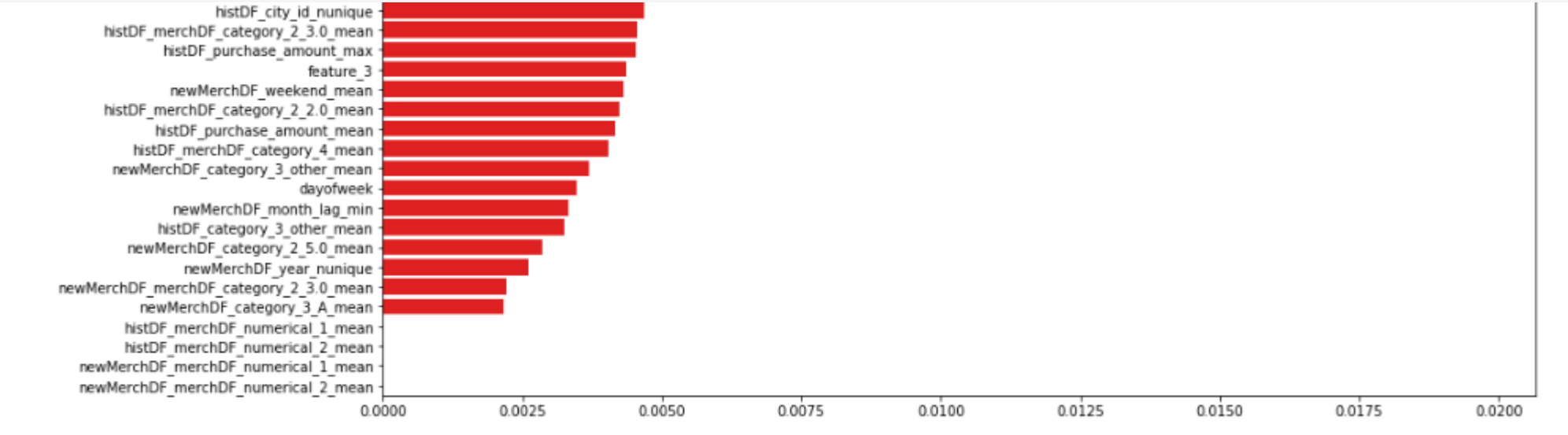
Machine Learning Model

Feature Importance



Machine Learning Model

Feature Importance



Conclusion and Recommendation

Top five features impacting model impacting loyalty score

1. **histDF_year_nunique** -- number of unique year in a card ID transactions in Historical transactions dataset
2. **histDF_month_nunique** -- number of unique months in a card ID transactions in Historical transactions dataset
3. **newMerchDF_purchase_days_tillToday** -- number of purchase days from last purchase date in new merchant transactions dataset
4. **histDF_purchase_date_max** -- Most recent purchase date of card ID in Historical transactions dataset
5. **newMerchDF_month_nunique** -- number of unique months in a card ID transactions in new merchant transactions dataset

Recommendation -

1. If the loyalty score of a card is low, then discount in top important category can sent to card holder.
2. Loyalty score can be monitored monthly and if the loyalty score decrease then a discount in most important category can set to card holder.