PMI Take Home

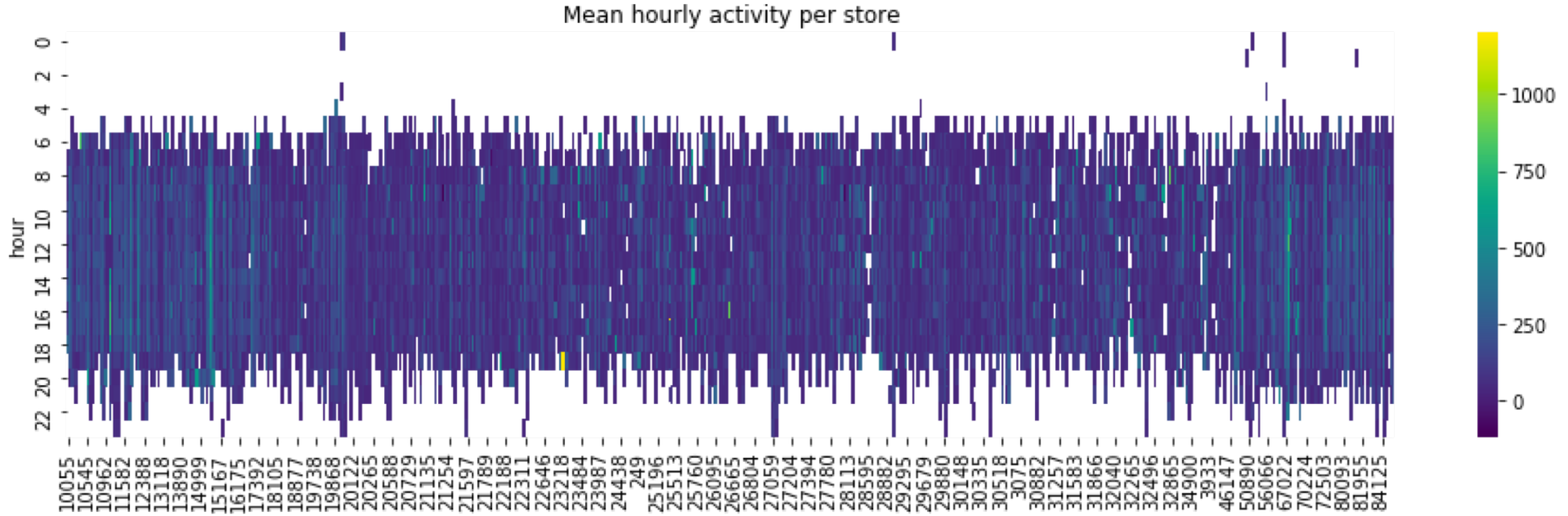
Oleg Kim

Target definition

- Based on **sales volume** data
  - Time series for each store
  - Varied time delta but **1-hr** is most common

- **Duplicated stores** were removed
  - Only a single store was duplicated

- Most stores are only active during day time and inactive on Sundays
  - Most activity is **consistent** across different stores

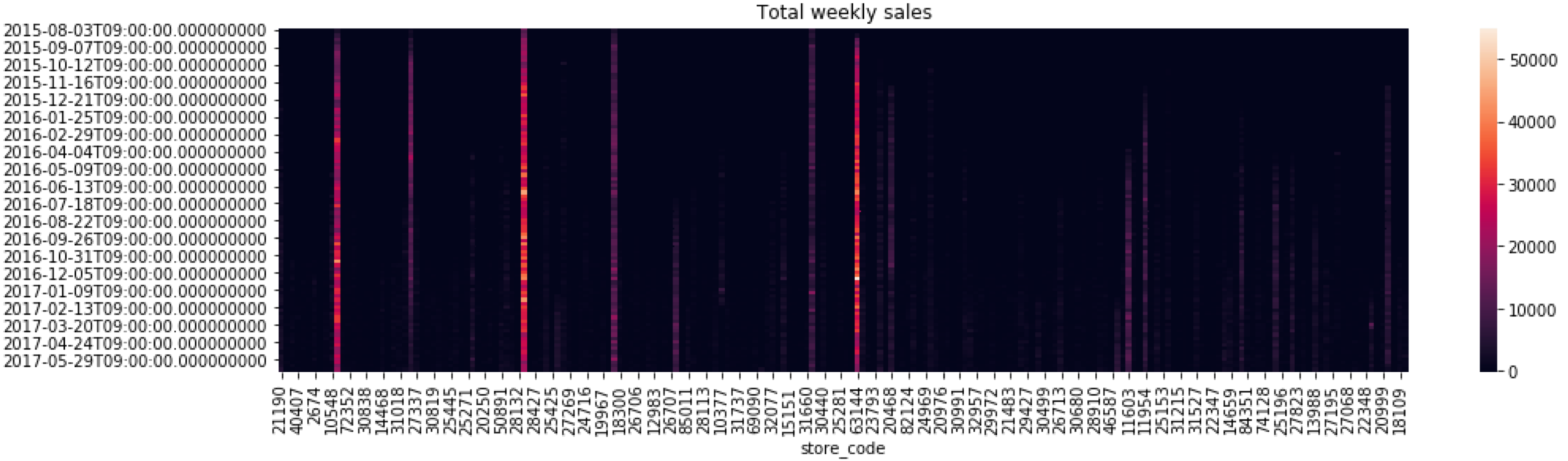


Target definition

- Aggregate sales volume using **total weekly amount**
  - Smooths out variation due to evenings and weekends

- Highly skewed distribution
  - Use **inverse hyperbolic sine transform**

- **Disregard negative** sales
  - Only 8 stores with negative sales



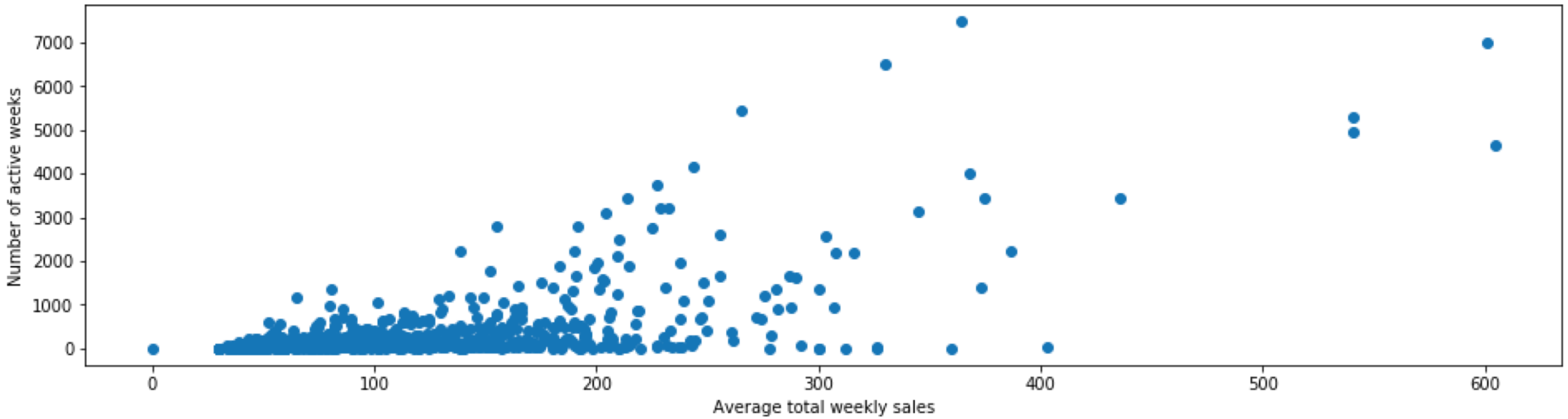
Target definition

Final target: average total weekly amount over all active weeks

- Treats old and recently open stores equally

Potential work

- Use different time scales than week
- Incorporate day and weekday information
- Keep negative sales volumes
- Use different transform
- Use weighted average
- Have a cutoff point



Features definition

- Features for each store are stored in **JSON** format

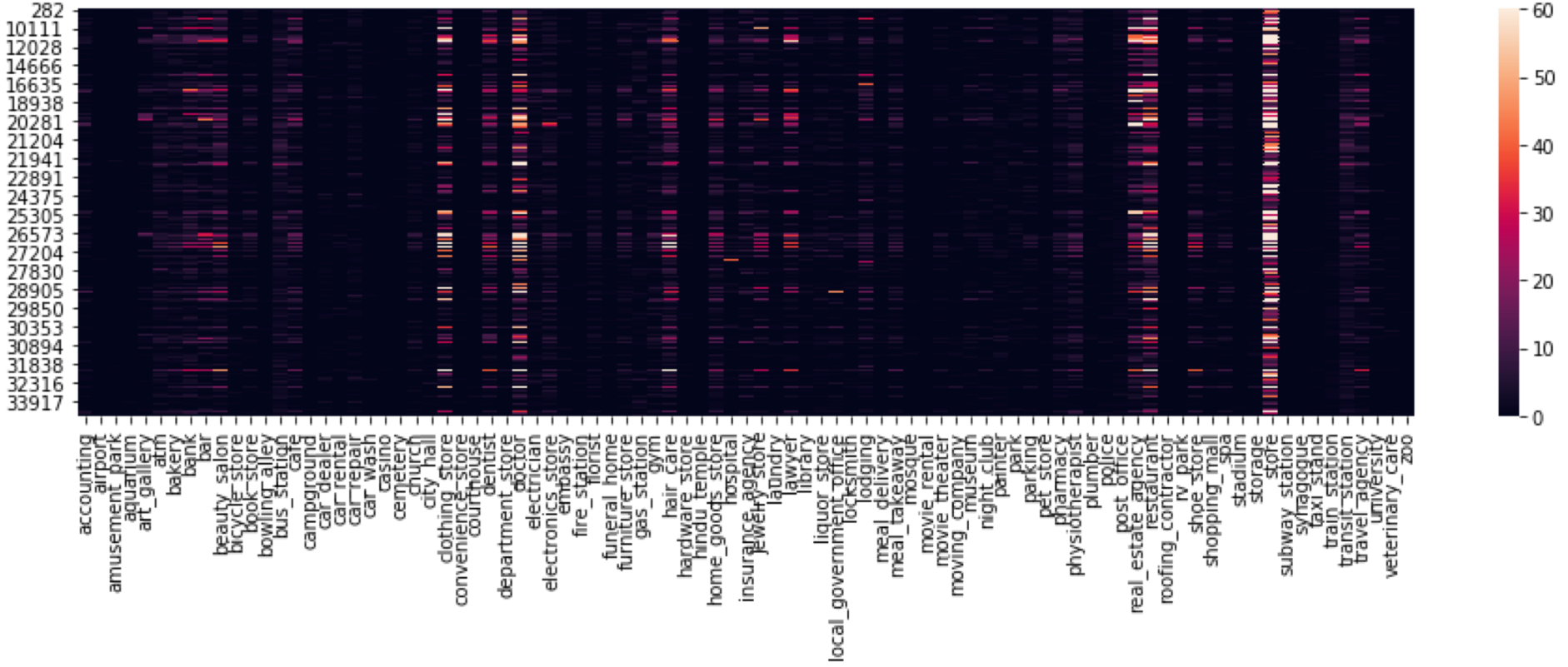
- Existing information:
  - address,
  - geocoordinates,
  - opening hours,
  - ratings,
  - reviews

- Variable number of features for each store (nature of **nested data structure**)

```
[... list of]
    {
        store_code: Int,
        surroundings: [... list of]
            {
                String: [... list of]
                    {
                        address_components: [... list of]
                            {
                                long_name: String,
                                short_name: String,
                                types: [... list of String]
                            },
                        formatted_address: String,
                        icon: String,
                        international_phone_number: String,
                        latitude: Float,
                        longitude: Float,
                        name: String,
                        opening_hours:
                        {
                            open_now: Boolean,
                            periods: [... list of]
                               {
                                   [open, close]:
                                       {
                                           day: Int,
                                           time: String
                                        }
                               }
                            weekday_text: [... list of String]
                        }
                        place_id: String,
                        rating: Float,
                        reviews: [... list of]
                            {
                               author_name: String,
                               author_url: String,
                               language: String,
                               profile_photo_url: String,
                               rating: Int,
                               relative_time_description: String,
                               text: String,
                               time: Int
                            }
                        types: [... list of String],
                        user_ratings_total: Int
                    }
            }
    }
```

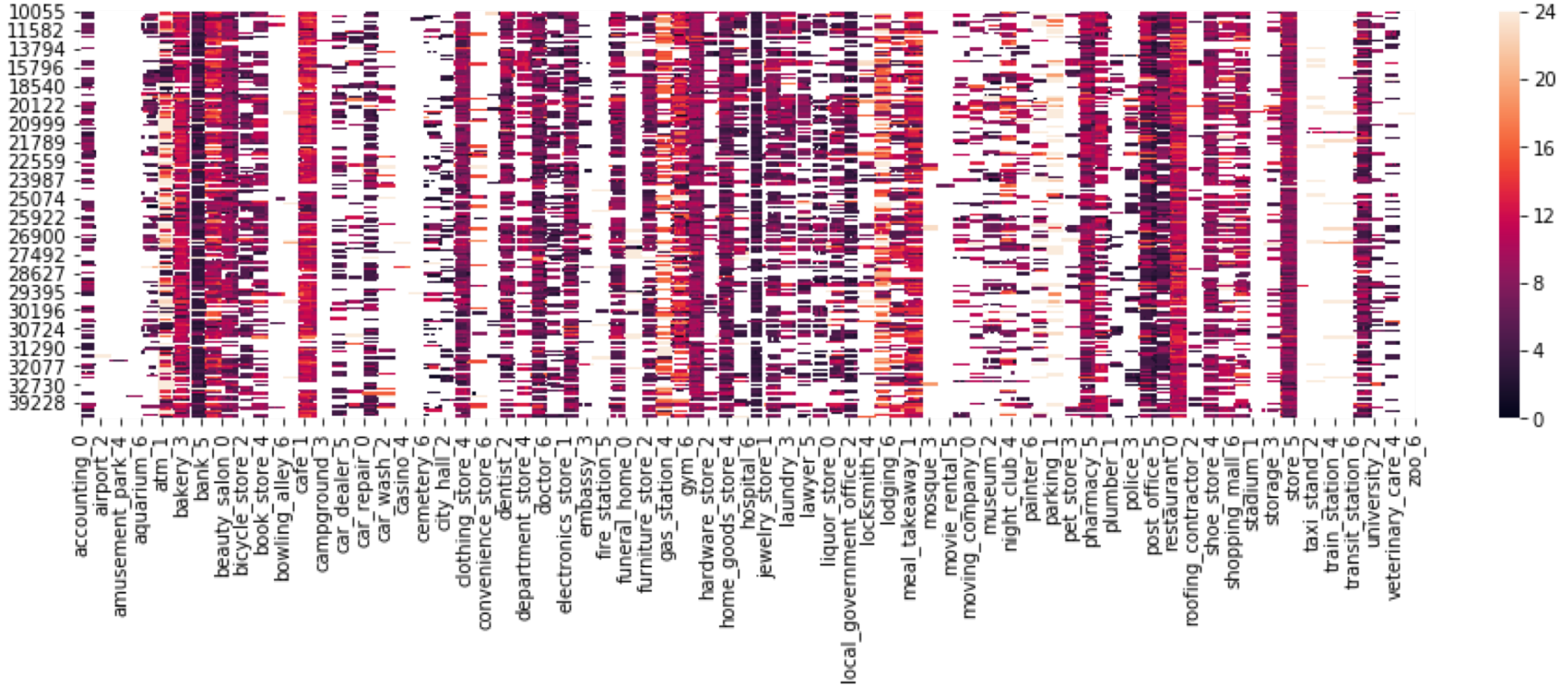
Features definition

- **Many amenities are absent** for many stores
- Need to **impute null values** meaningfully for each amenity type



Features definition

- Features used for each amenity type:
  - Number of amenities
  - Mean of latitude
  - Mean of longitude
  - Mean total time open during each day of the week
  - Mean rating



Features definition

Other possibilities

- Location
  - Calculate a proxy for store location using all amenities
  - Extract min, max and std in addition to the mean
  - Do a graph analysis of locations

Features definition

Other possibilities

- Opening times
  - get opening times during different parts of day (morning, day, evening, night, lunch, dinner)
  - get information of being open during a each hour of the day
  - get opening and closing times on certain days (e.g. weekends, holidays)
  - get open activity during different week and months
  - get raw opening and closing times
  - Extract std, min and max of total time open

Features definition

Other possibilities

- Reviews
  - NLP on the text of reviews
  - NLP on text geographical data, such as street and city names
  - process individual review to get a better measure of rating (e.g. taking into account time of reviews and spread of ratings)

Features definition

Other possibilities

- Combine features
  - use location of the amenity when calculating opening hours duration
  - group different amenity types (e.g. shoe store and clothing store) and do feature engineering for entire group
  - don't aggregate and instead set each amenity as a separate feature. In this case, ordering of amenities will need to be decided and different number of amenities for different stores needs to be handled.

Modeling

- **Three models** were tried
  - Regression
  - Classification
  - Ranking

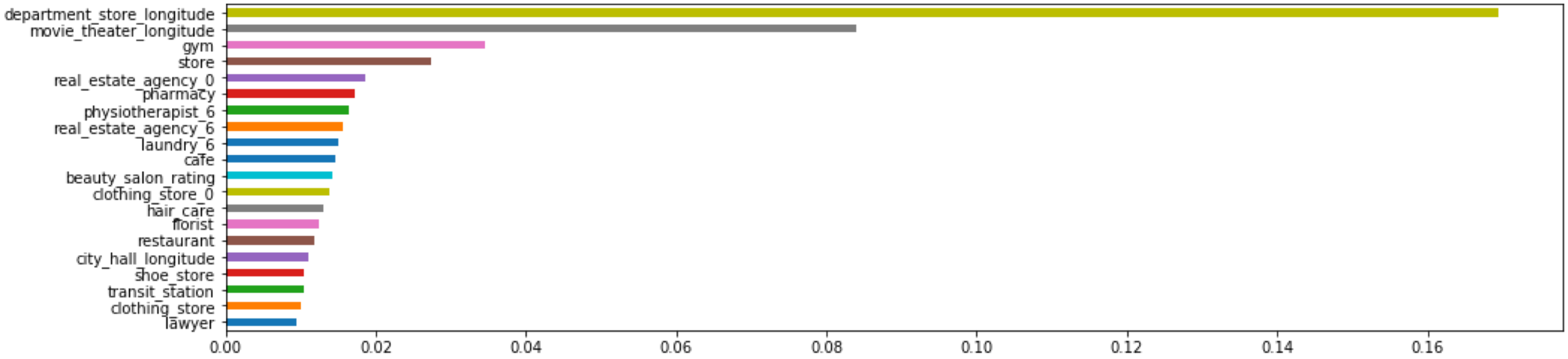
- **Preprocessing**
  - Remove features with low variance
  - Remove features with high number of nulls
  - Null imputation:
    - Number of amenities is 100% complete
    - Mean time open is imputed with 0
    - The rest are imputed with average

- **Model assessment**
  - Train-test split

Regression

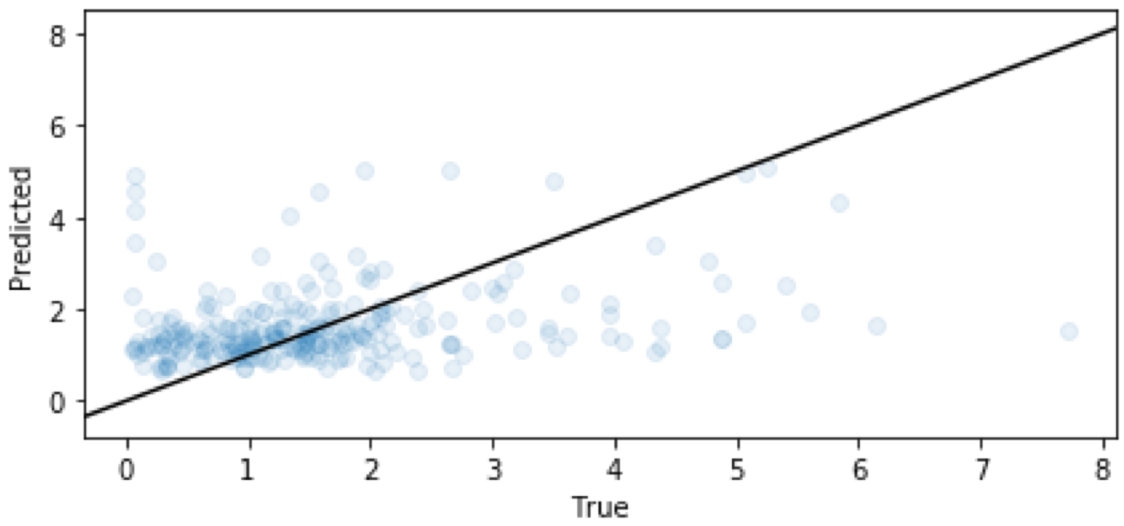
Data is highly ordinal

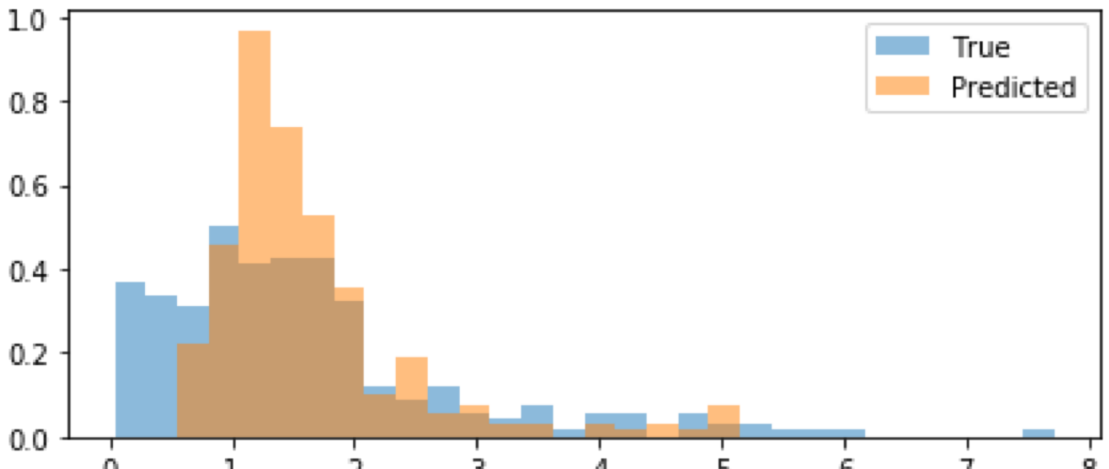
Algorithm used: **Random Forest**



Regression

MSE: 1.69 (random: 4.21)



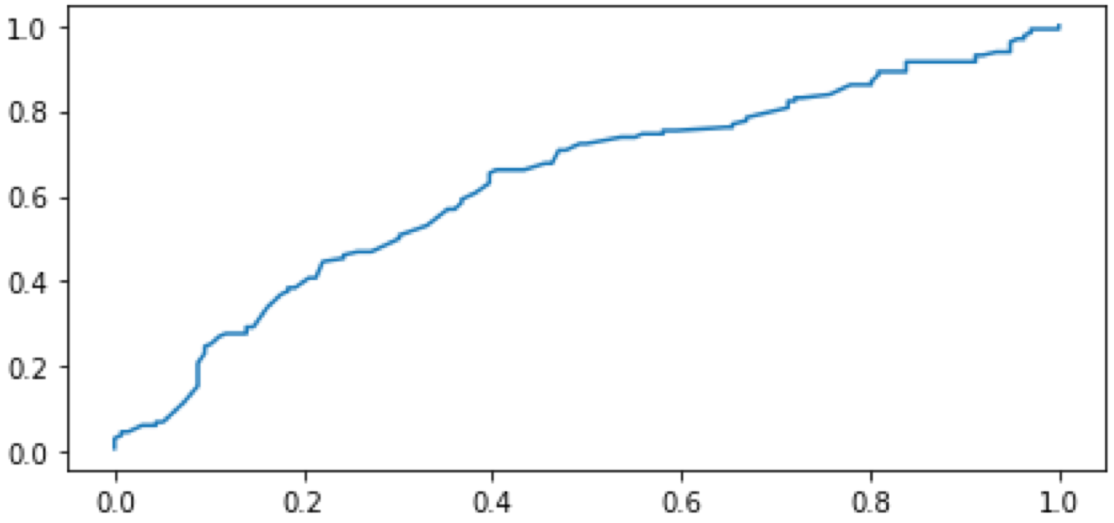


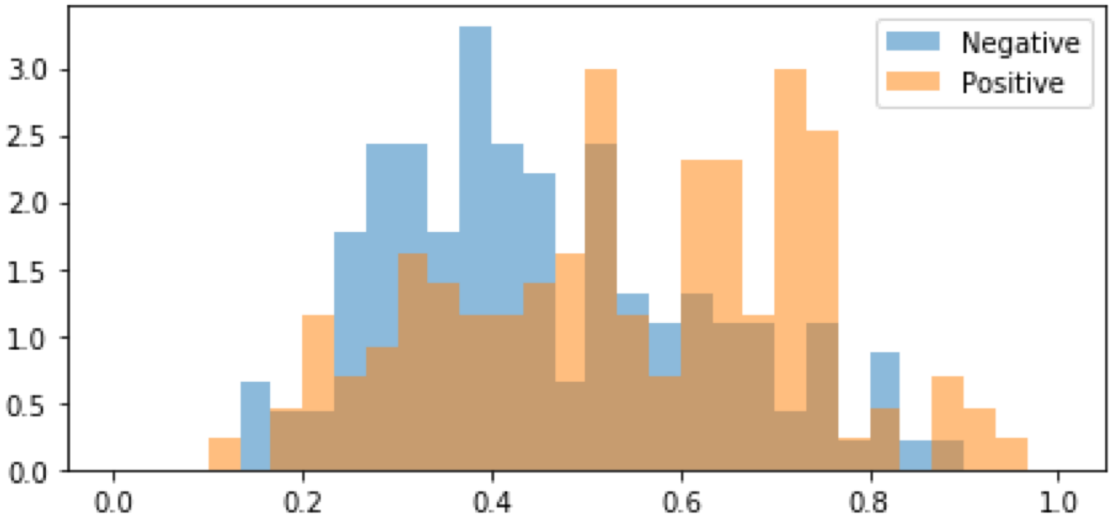
Classification

Data is highly ordinal

Algorithm used: **Random Forest**

ROC AUC: .633





Ranking

- Ranking is important in case rank matters more than absolute value
- Algorithm used: **LambdaMART**
- Metric used: **nDCG**
- **Comparison**
  - RF regressor: 0.231
  - RF classifier: 0.156
  - LambdaMART: 0.198

Potential work

Potential work

- investigate effects of different random seeds
- different threshold for standard deviation and null counts to discard
- different imputation of nulls
- more rigorous feature selection (backward selection, genetic algorithms)
- dimensionality reduction (PCA, MCA)
- more rigorous hyperparameter selection (grid search, random search, bayesian optimization)
- removal of outliers (kernel density estimation, clustering techniques)
- data normalization
- different definition of classes (extreme quantiles, multiclass)

Production

- more robust checks for data validity
- preprocessing steps to be written as pipeline components, because they rely on saved parameters when transforming unseen data