The patterns and forecasting analysis of grocery sales for Corporación Favorita retailer

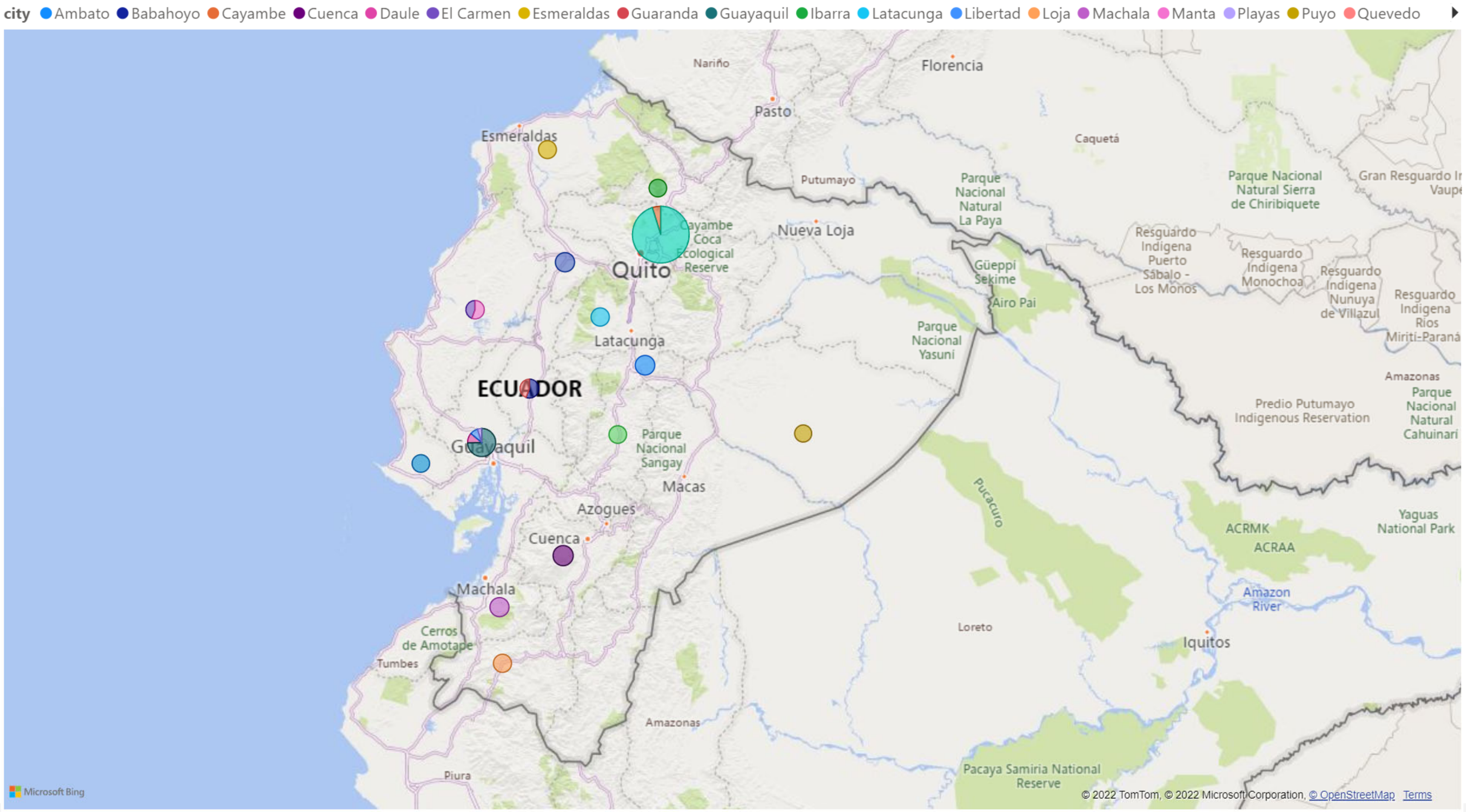
Dr. Xuezhi Zeng

Content

- Problem summary
- Exploratory Data Analysis
- Forecasting of grocery sales
- What customer insights are provided by the analysis
- Future improvement

About the retail company

Corporacion Favorita is a grocery chain in Ecuador with over 100 stores. They are holding a Kaggle competition to develop a model for predicting unit sales of items for each of their stores to improve inventory management



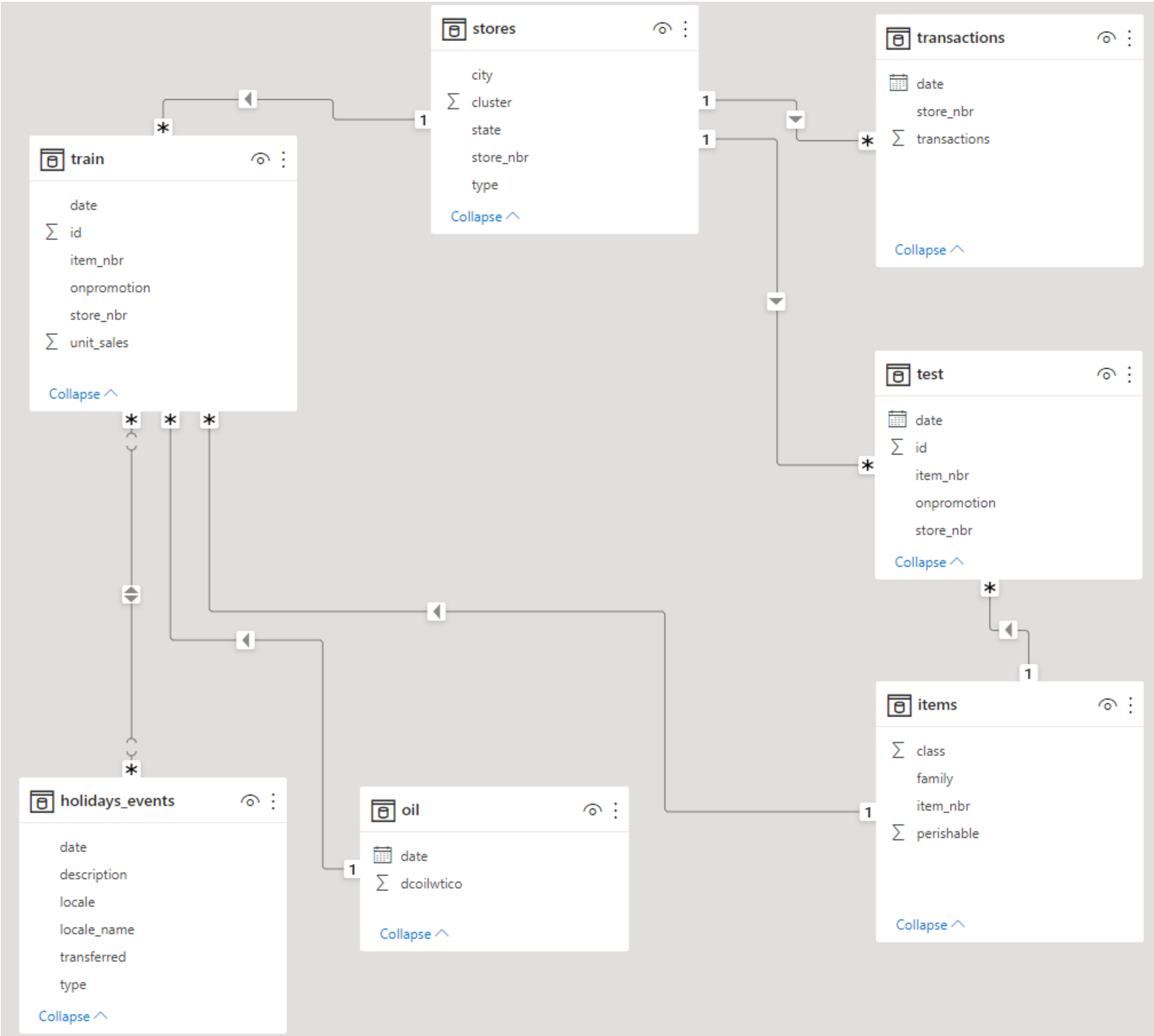
Problem summary

They encounter the following two problems:

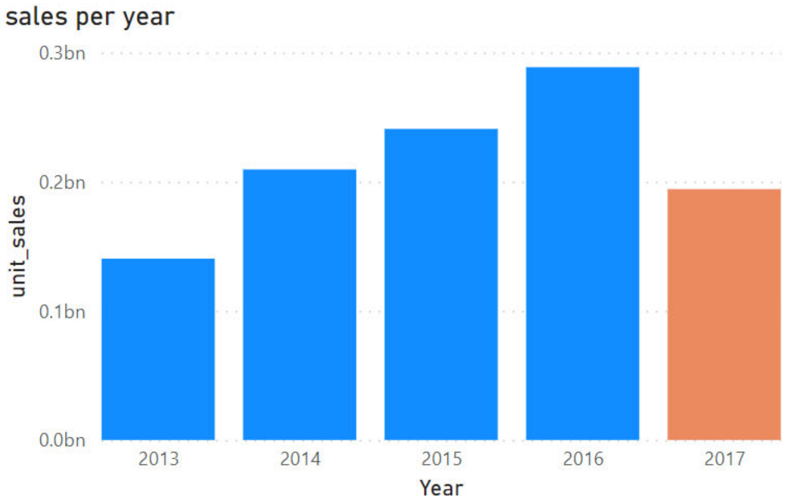
- Stocking-out: A situation in which an item is out of stock
  - popular items quickly sell out, leaving money on the table and customers fuming
  - causes an increased risk of lost sales, since customers are more likely to look elsewhere for the necessary items, which in-turn can be a huge opportunity loss and customer retain for the retailer

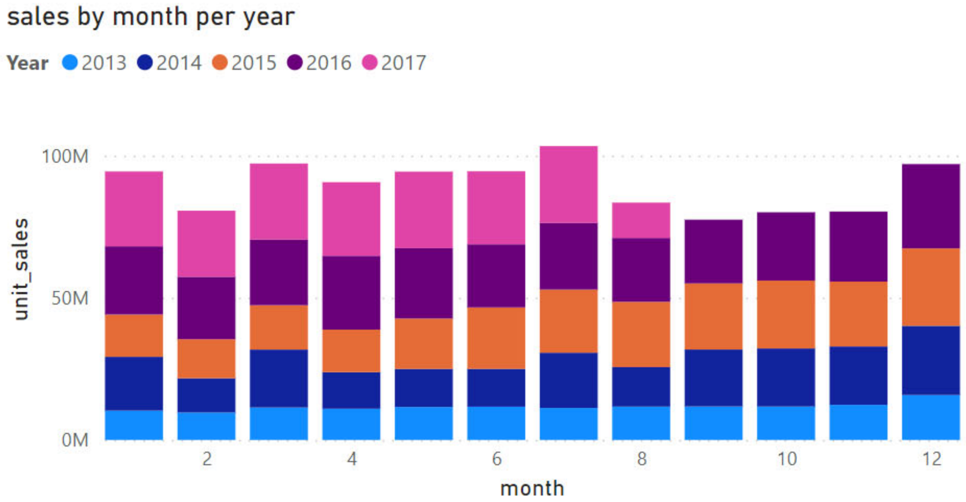
- Overstocking: A situation in which an item is present in more quantity than is necessary or required
  - potentially cost drastic amounts of money as products just sit on shelves where they are not being utilized
  - depending on their type (e.g., perishable or fragile) they can get spoiled and damaged which would cause management cost and a total loss for the retailer

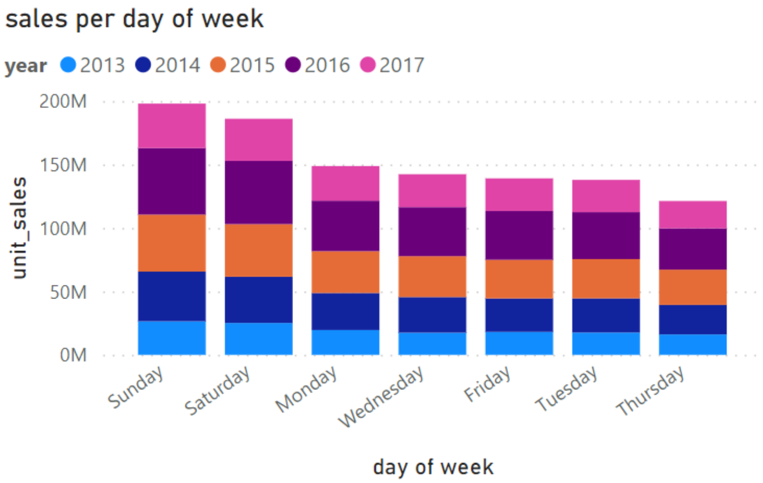
Data model

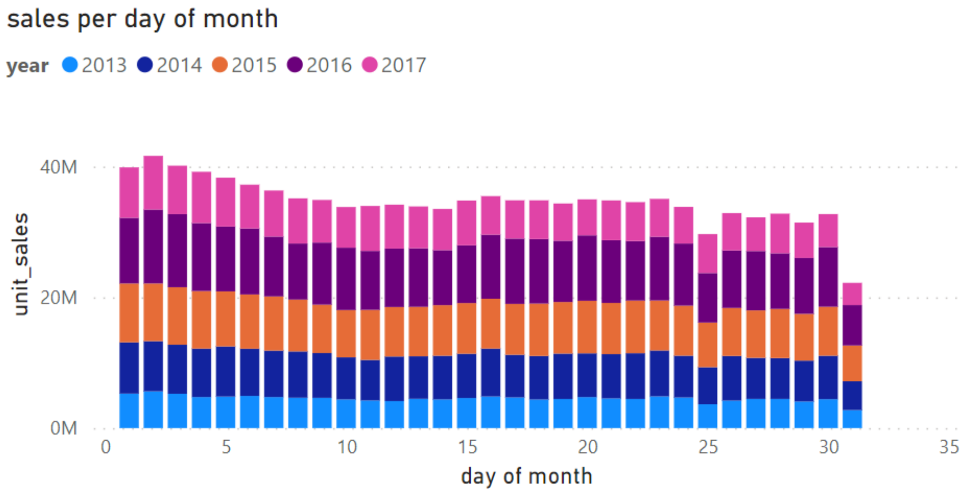

stores
city
cluster
state
store_nbr
type
Collapse
transactions
date
store_nbr
transactions
Collapse
train
date
id
item_nbr
onpromotion
store_nbr
unit_sales
Collapse
test
date
id
item_nbr
onpromotion
store_nbr
Collapse
items
class
family
item_nbr
perishable
Collapse
holidays_events
date
description
locale
locale_name
transferred
type
Collapse
oil
date
dcoilwtico
Collapse
1
*

Exploratory data analysis: Overall sales varying across years, months, day of week, and day of month

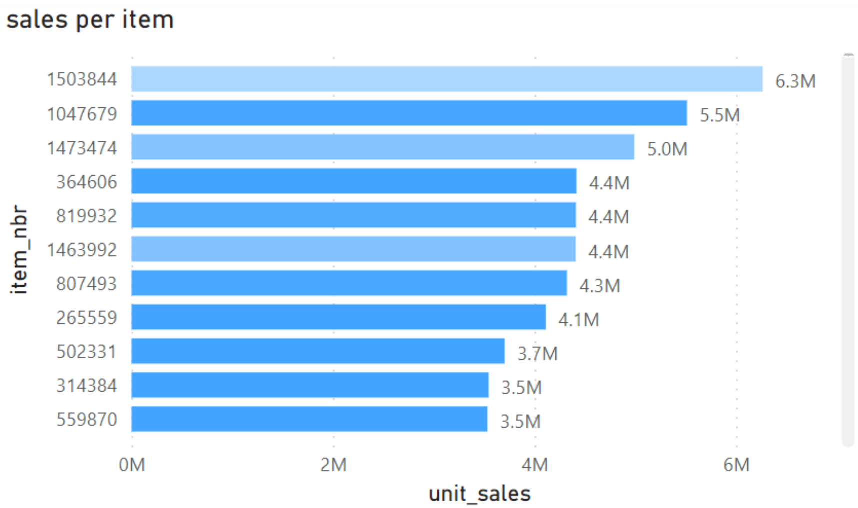


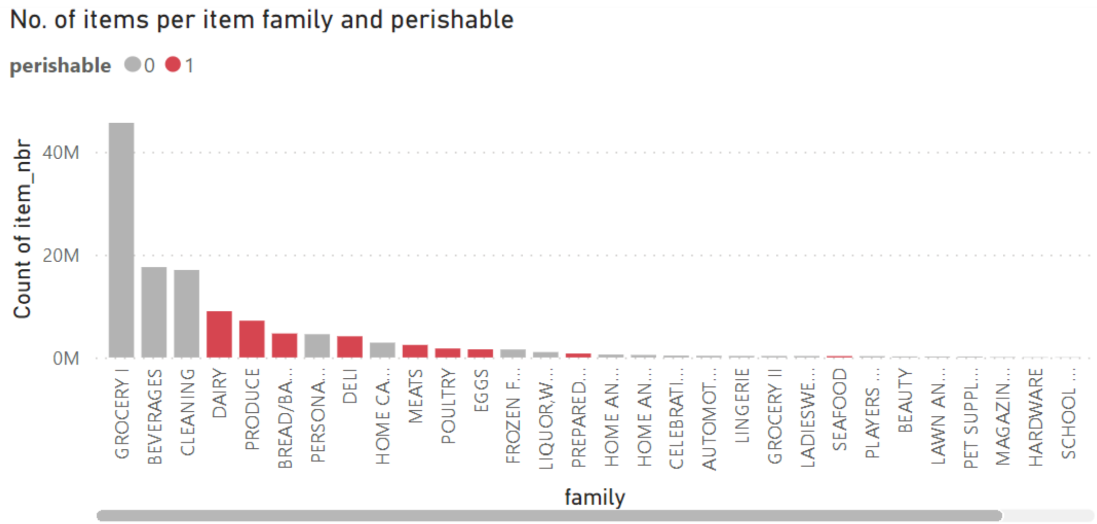


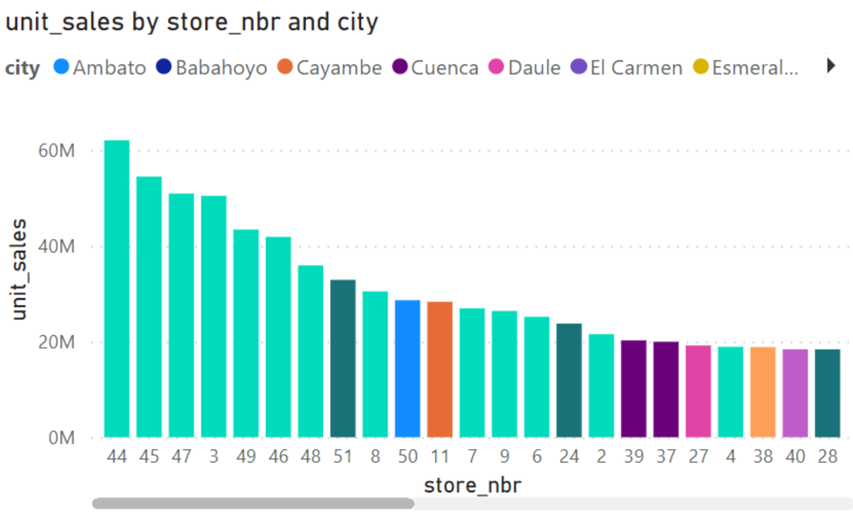


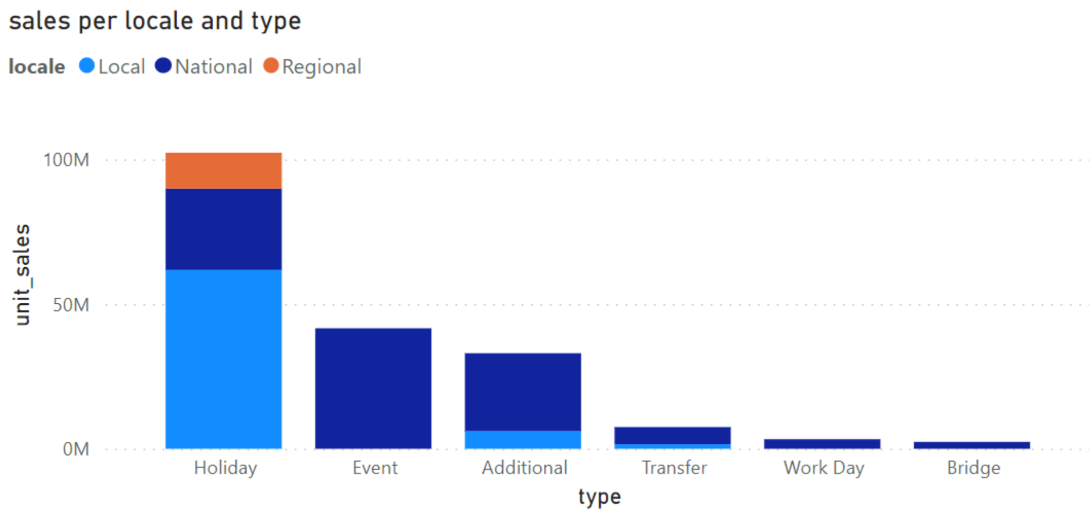


Exploratory data analysis: the sales categorized under items, stores, and locales

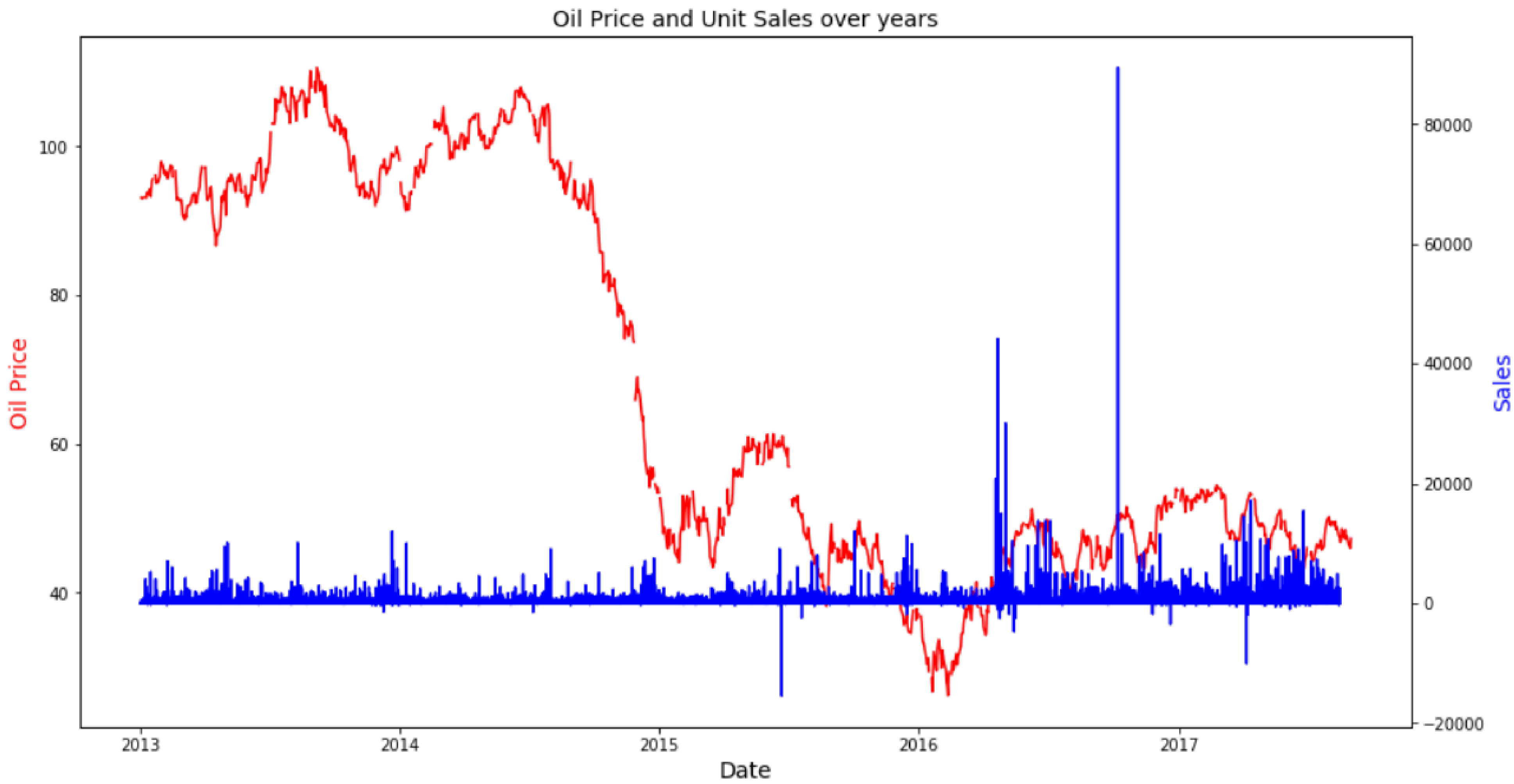




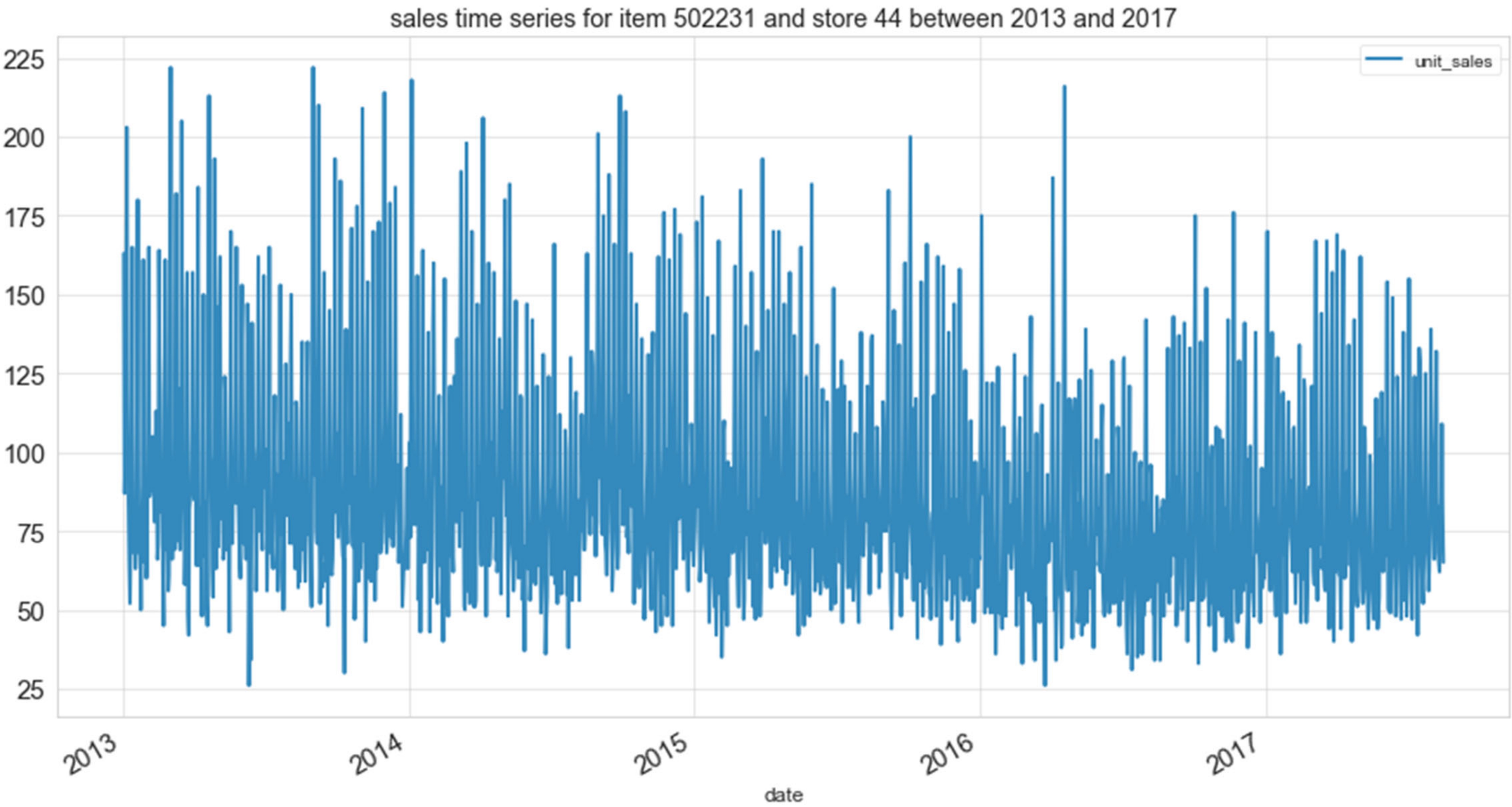




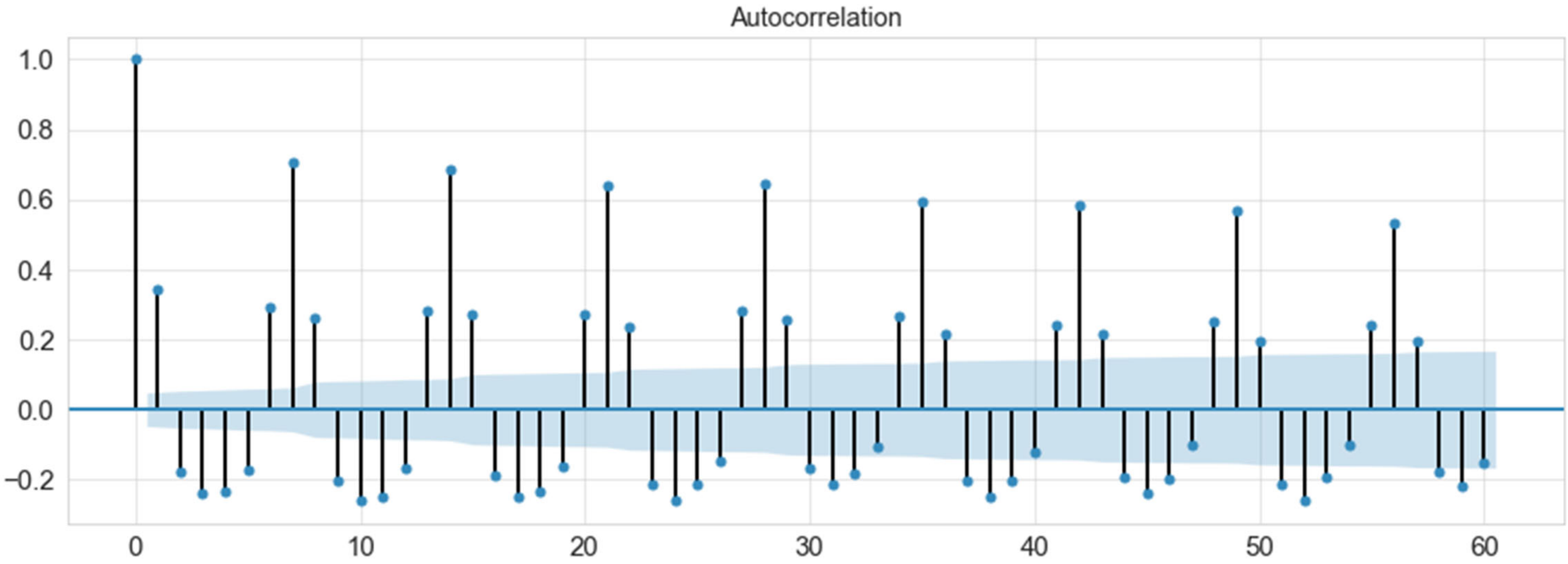
Comparing the trends between oil price and overall sale over years



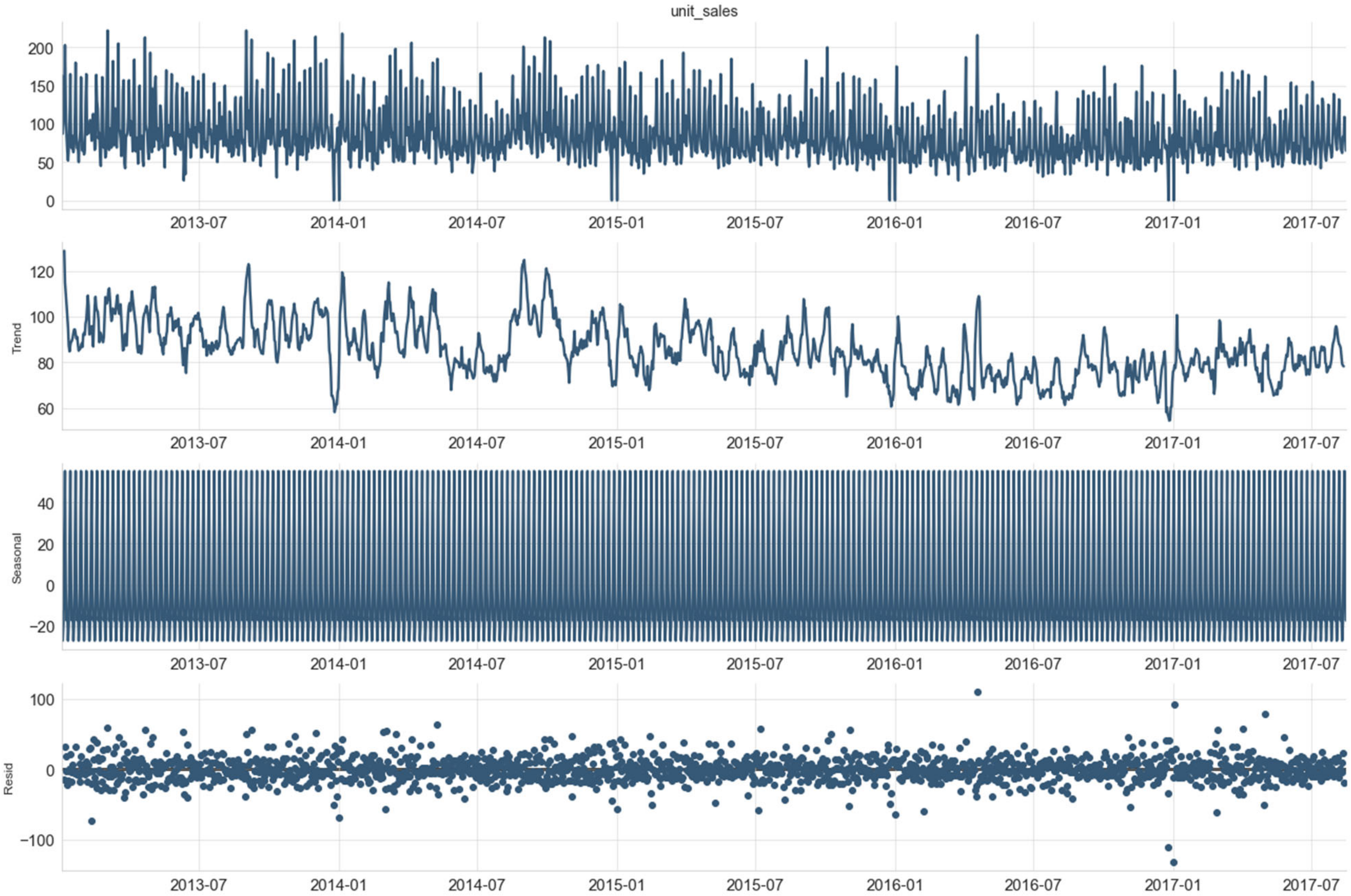
Example case: Sales of item 502231 from store 44 between 2013 and 2017



Example case: Autocorrelation test



Example case: Decomposing trend, seasonality and noise



Example case: Stationarity test

	Value	module
name	Item 502331 in Store 2	nan
ADF_Statistic	-5.105800	adfuller
p-value	0.000014	adfuller
num_lags_used	22	adfuller
n_observations_used	1664	adfuller
IC_for_best	14861.128981	adfuller
1%	-3.434286	adfuller
5%	-2.863278	adfuller
10%	-2.567696	adfuller
likely stationary	True	p-value < 0.05

Pre-processing

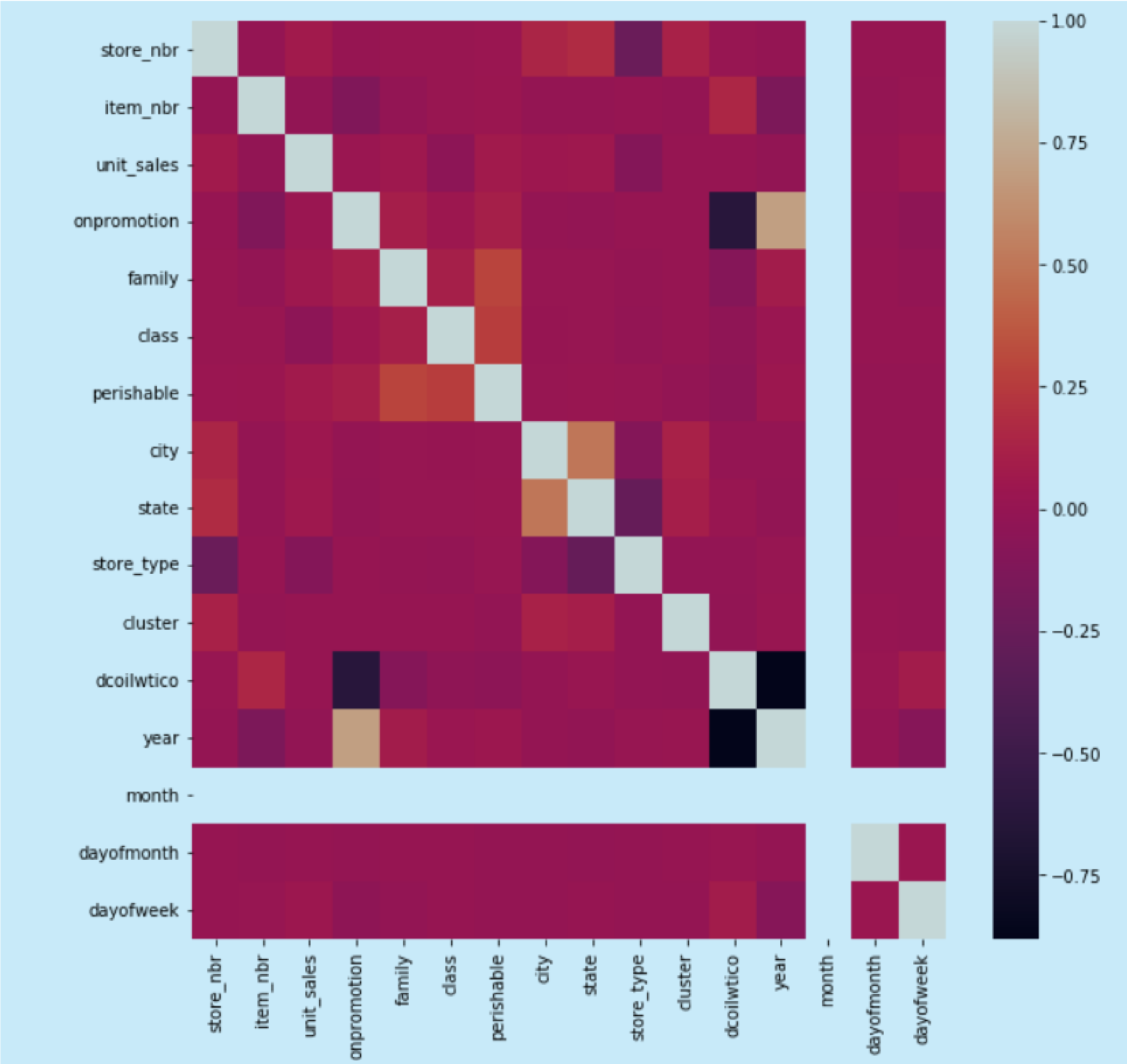
- Selected train data with month in August and day > 15
- Introduced extra features (year, month, day of month, day of week)
- Filled in missing values for onpromotion with -1
- Filled in missing values for oil price using moving average

Feature representation

- **Continuous features:**
  - unit_sales
  - dcoilwtico
- **Categorical features:**
  - store_nbr
  - item_nbr
  - onpromotion
  - family
  - class
  - perishable
  - city
  - state
  - store_type
  - cluster
  - year
  - month
  - dayofmonth
  - dayofweek

Label encoding

Correlation matrix between these features



Evaluation metrics

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2}{N}}$$

$$MAE = \frac{\sum_{i=1}^{N}\left|\hat{y}_i - y_i\right|}{N}$$

$$R^2 = \frac{\sum_{i=1}^{N}\left(\hat{y}_i - \bar{y}\right)^2}{\sum_{i=1}^{N}\left(y_i - \bar{y}\right)^2}$$

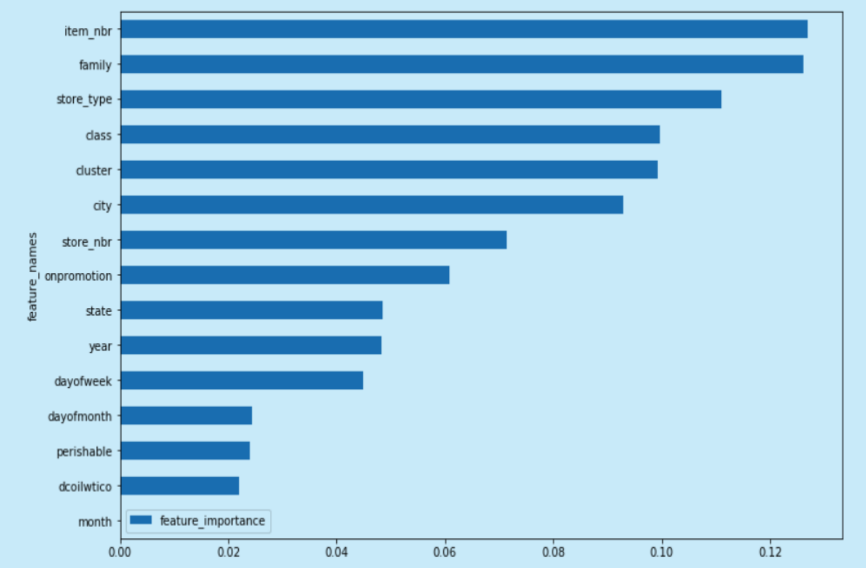
$$NWRMSE = \sqrt{\frac{\sum_{i=1}^{n} weights_i * \left(\hat{y}_i - y_i\right)^2}{\sum_{i=1}^{n} weights_i}}$$

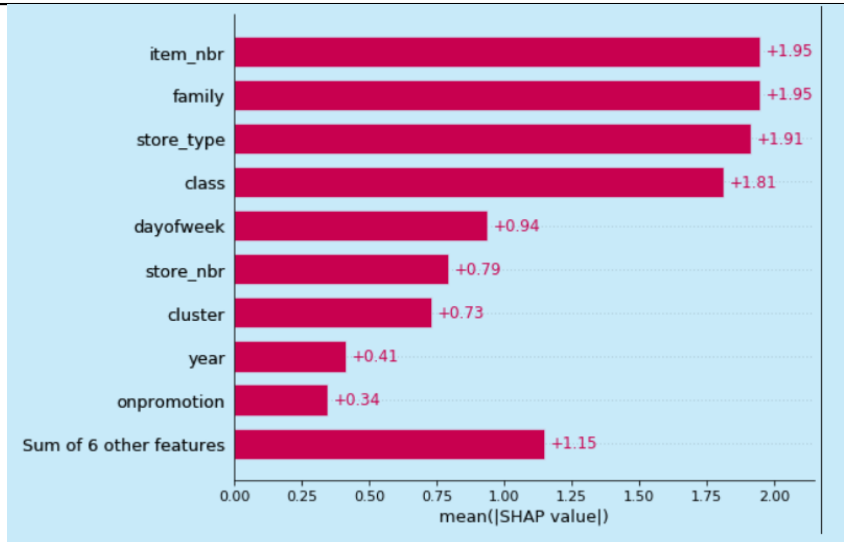
$$NWRMSLE = \sqrt{\frac{\sum_{i=1}^{n} weights_i * \left(log(\hat{y}_i + 1) - log(y_i + 1)\right)^2}{\Sigma_{i=1}^{n} weights_i}}$$

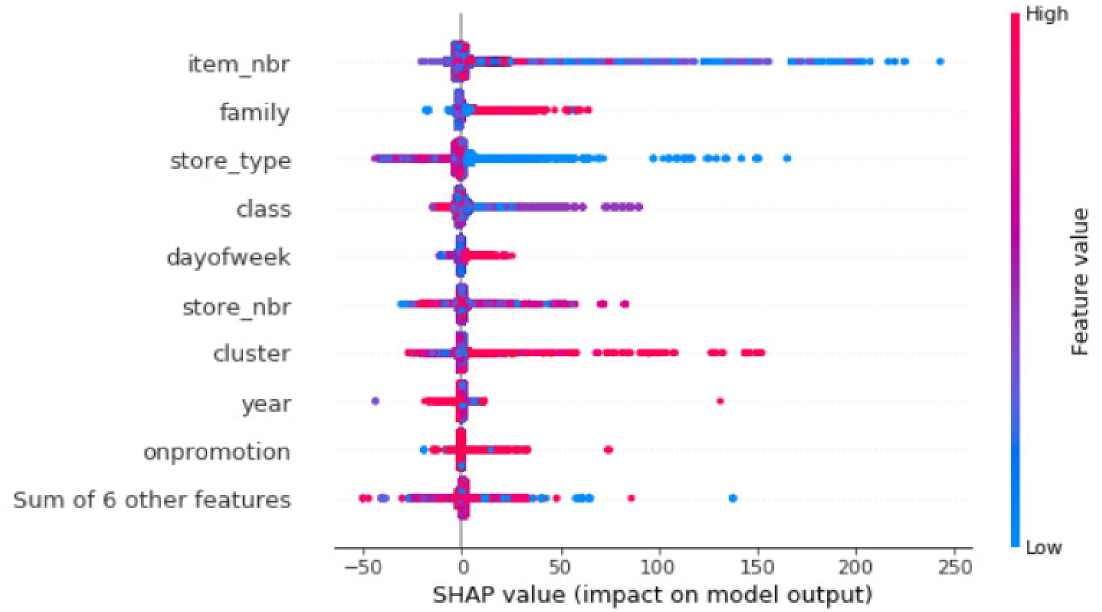
Model evaluation (1)

Forecasting models	Metrics				
	RMSE	MAE	R^2	NWRMSE	NWRMSLE
Linear regression	Train:16.745 Test: 16.534	Train:6.972 Test: 6.961	Train:0.03 Test: 0.03	Train:17.059 Test:16.814	Train:0.948 Test: 0.948
XGBRegression	Train:12.640 Test: 12.471	Train:5.150 Test: 5.162	Train: 0.447 Test: 0.448	Train:12.856 Test: 0.6515	Train:0.734 Test: 0.736
Random Forest regression (estimators = 10, 'max_depth = 5, min_samples_leaf = 3)	Train:16.255 Test: 16.074	Train:6.707 Test: 6.695	Train: 0.086 Test: 0.083	Train:16.541 Test: 16.331	Train:0.906 Test: 0.906

Model interpretability







Embedding representation

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
Food	0.04	0.01	0.02	0.01	0.95	0.97
⋮						

Source: from Andrew Ng's AI course

Amazing results using embedding in text

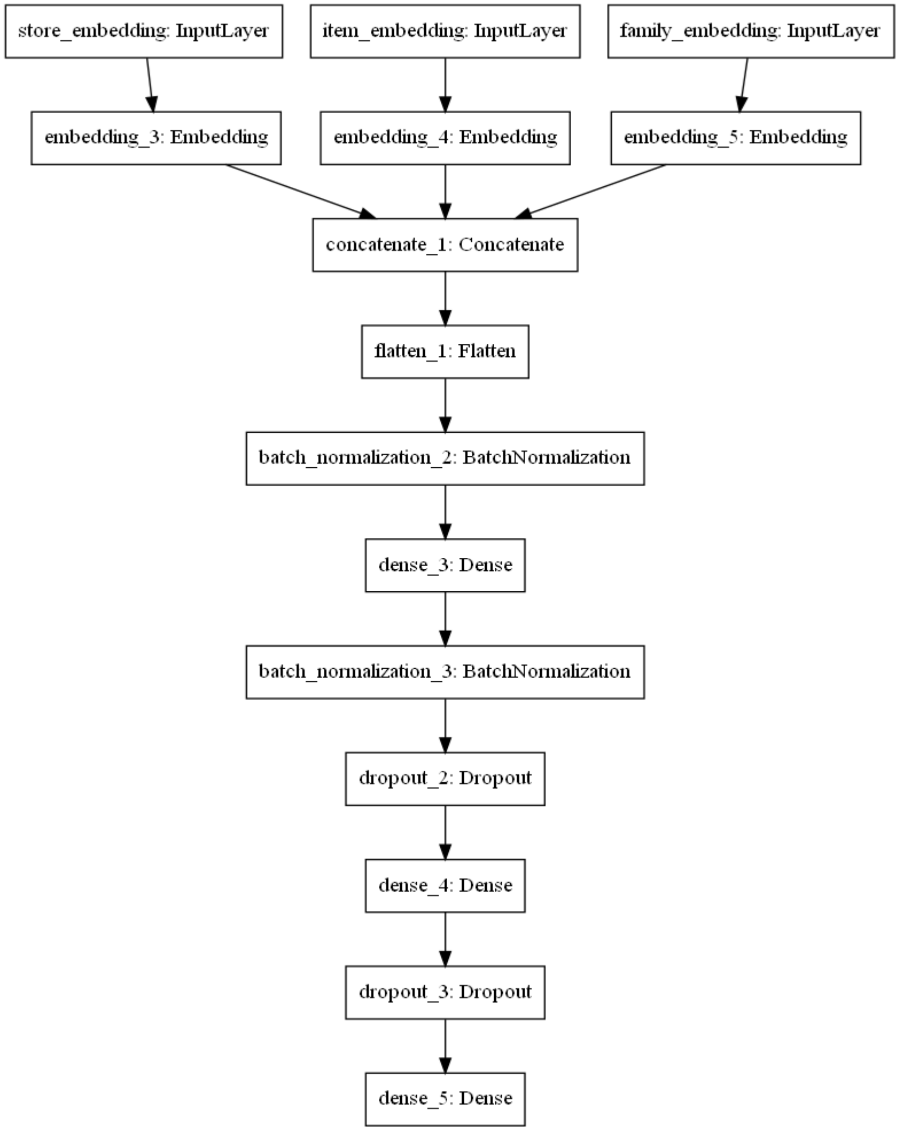
- Rome - Italy + Australia = ?

  Rome: Italy = Canberra: Australia

- Women - Breast cancer + Men = ?

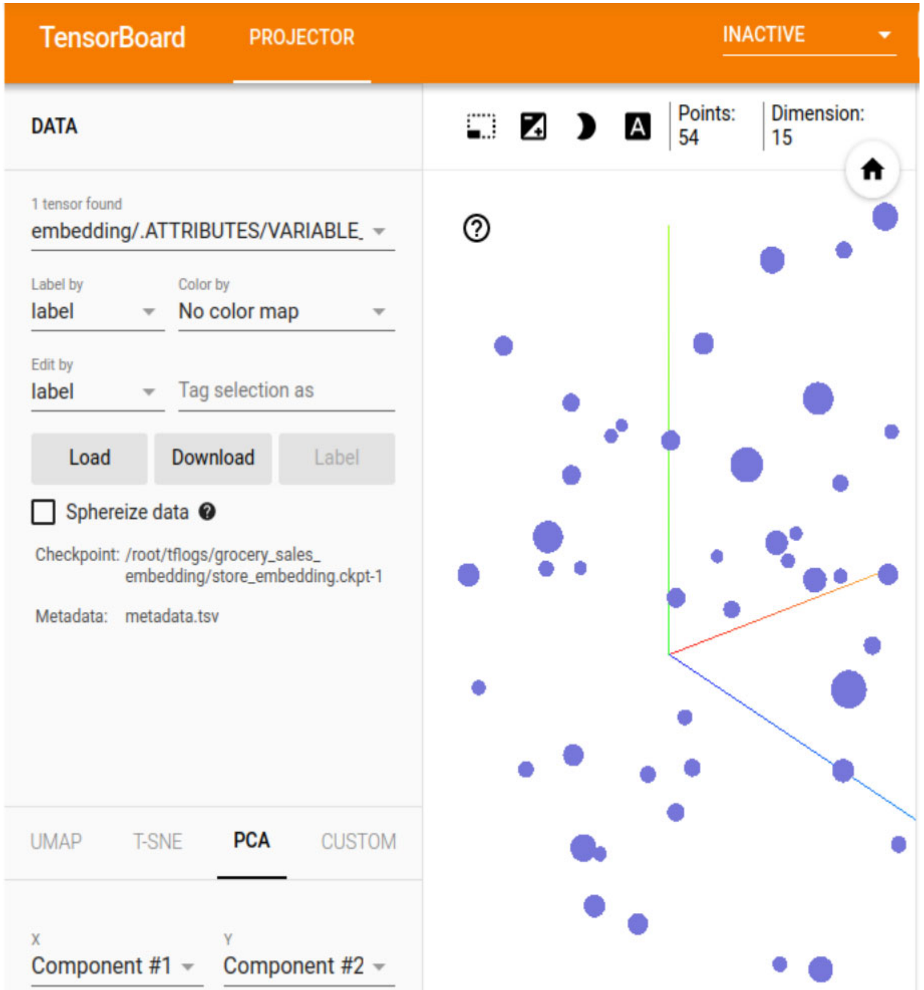
  Women: Breast cancer = Men: Prostate cancer

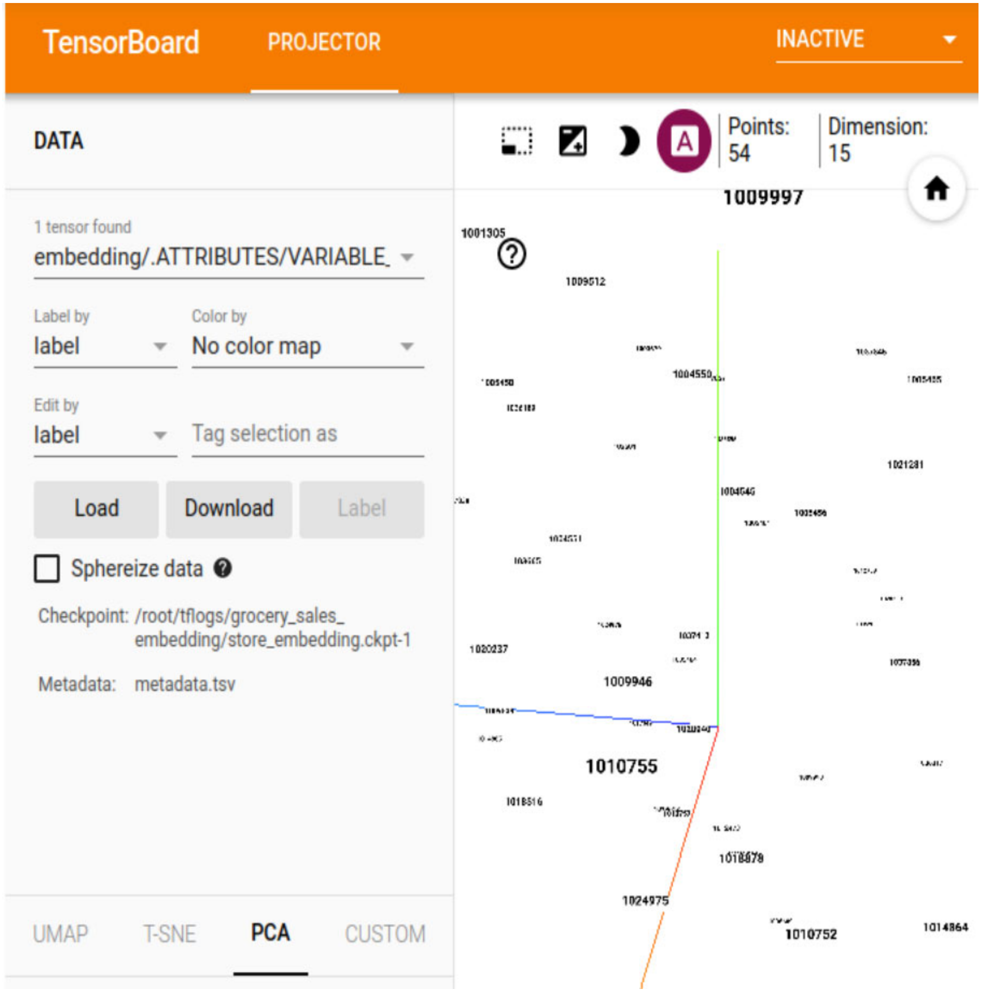
Embedding neural network structure




```
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
store_embedding (InputLayer)    [(None, 1)]          0
__________________________________________________________________________________________________
item_embedding (InputLayer)     [(None, 1)]          0
__________________________________________________________________________________________________
family_embedding (InputLayer)   [(None, 1)]          0
__________________________________________________________________________________________________
embedding_3 (Embedding)         (None, 1, 20)        1080        store_embedding[0][0]
__________________________________________________________________________________________________
embedding_4 (Embedding)         (None, 1, 100)       410000      item_embedding[0][0]
__________________________________________________________________________________________________
embedding_5 (Embedding)         (None, 1, 15)        495         family_embedding[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 1, 135)       0           embedding_3[0][0]
                                                                 embedding_4[0][0]
                                                                 embedding_5[0][0]
__________________________________________________________________________________________________
flatten_1 (Flatten)             (None, 135)          0           concatenate_1[0][0]
__________________________________________________________________________________________________
batch_normalization_2 (BatchNor (None, 135)          540         flatten_1[0][0]
__________________________________________________________________________________________________
dense_3 (Dense)                 (None, 100)          13600       batch_normalization_2[0][0]
__________________________________________________________________________________________________
batch_normalization_3 (BatchNor (None, 100)          400         dense_3[0][0]
__________________________________________________________________________________________________
dropout_2 (Dropout)             (None, 100)          0           batch_normalization_3[0][0]
__________________________________________________________________________________________________
dense_4 (Dense)                 (None, 50)           5050        dropout_2[0][0]
__________________________________________________________________________________________________
dropout_3 (Dropout)             (None, 50)           0           dense_4[0][0]
__________________________________________________________________________________________________
dense_5 (Dense)                 (None, 1)            51          dropout_3[0][0]
==================================================================================================
```

Embedding visualization in Tensorboard

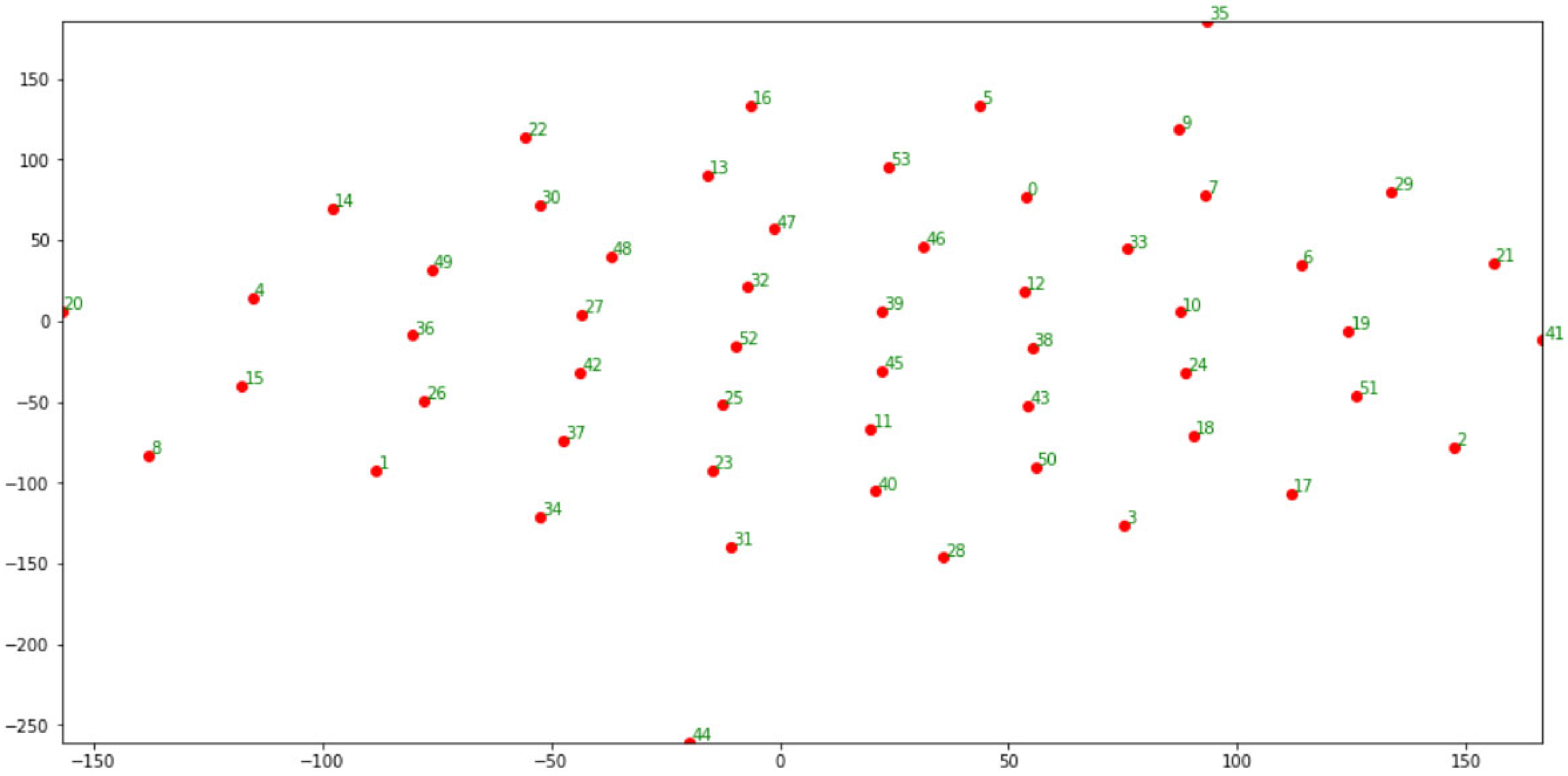




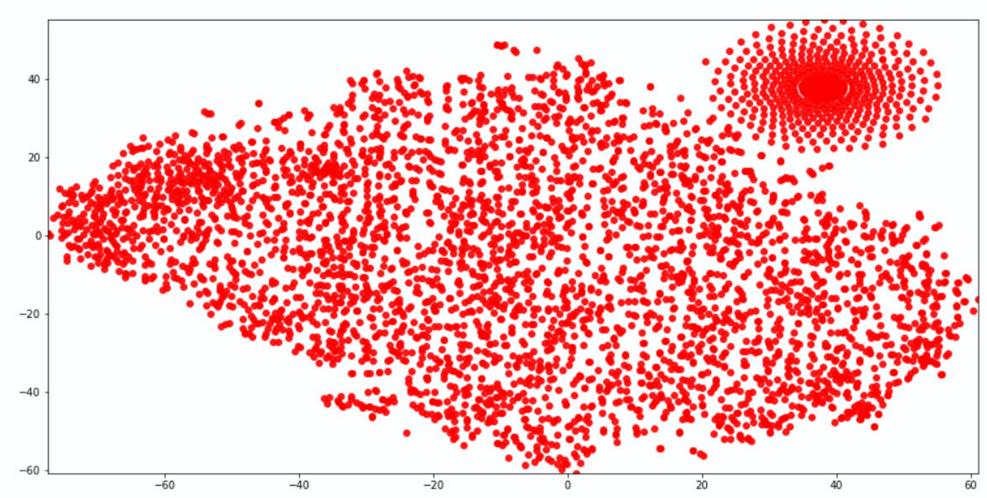
Model evaluation (2)

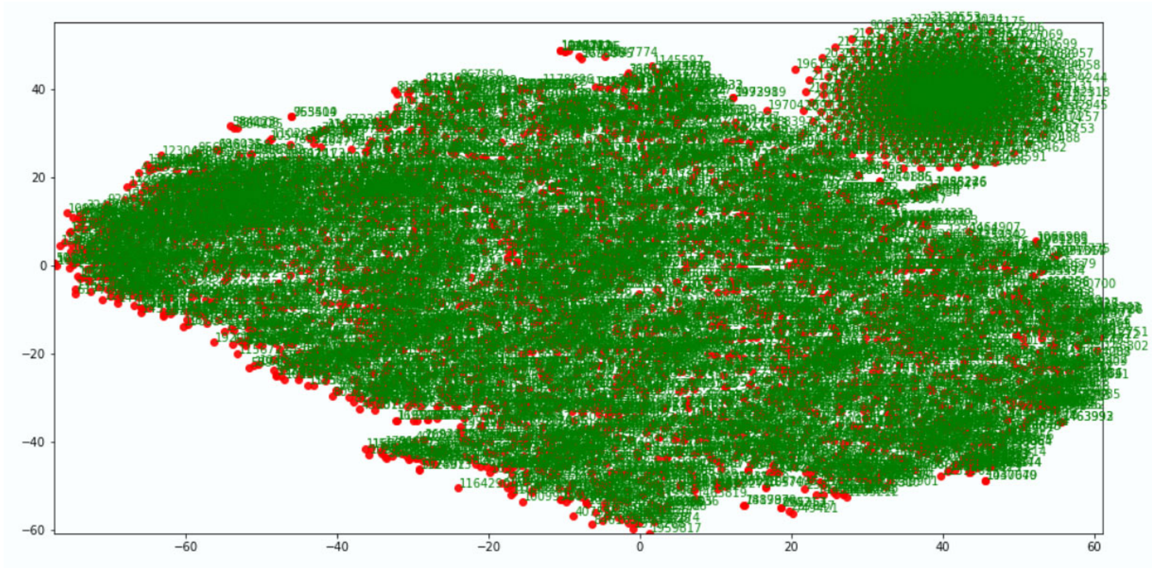
Forecasting models	Metrics				
	RMSE	MAE	R^2	NWRMSE	NWRMSLE
Linear regression	Train:16.745 Test: 16.534	Train:6.972 Test: 6.961	Train:0.03 Test: 0.03	Train:17.059 Test:16.814	Train:0.948 Test: 0.948
XGBRegression	Train:12.640 Test: 12.471	Train:5.150 Test: 5.162	Train: 0.447 Test: 0.448	Train:12.856 Test: 0.6515	Train:0.734 Test: 0.736
Random Forest regression (estimators = 10, 'max_depth = 5, min_samples_leaf = 3)	Train:16.255 Test: 16.074	Train:6.707 Test: 6.695	Train: 0.086 Test: 0.083	Train:16.541 Test: 16.331	Train:0.906 Test: 0.906
Embedding	Train: 10.994 Test: 11.148	Train: 4.222 Test: 4.248	Train: 0.585 Test: 0.559	Train: 11.118 Test: 11.270	Train: 0.626 Test: 0.628

The embedding result of store_nbr

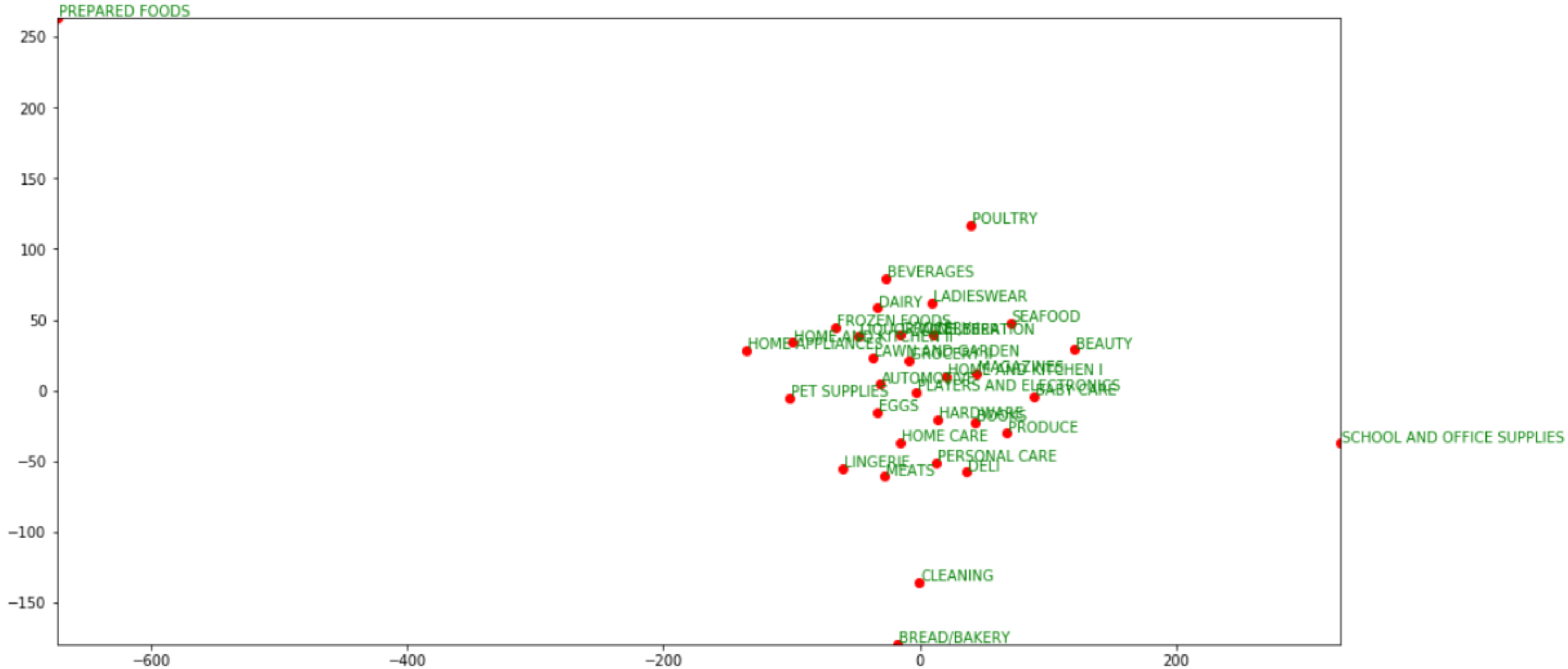


The embedding result of item_nbr





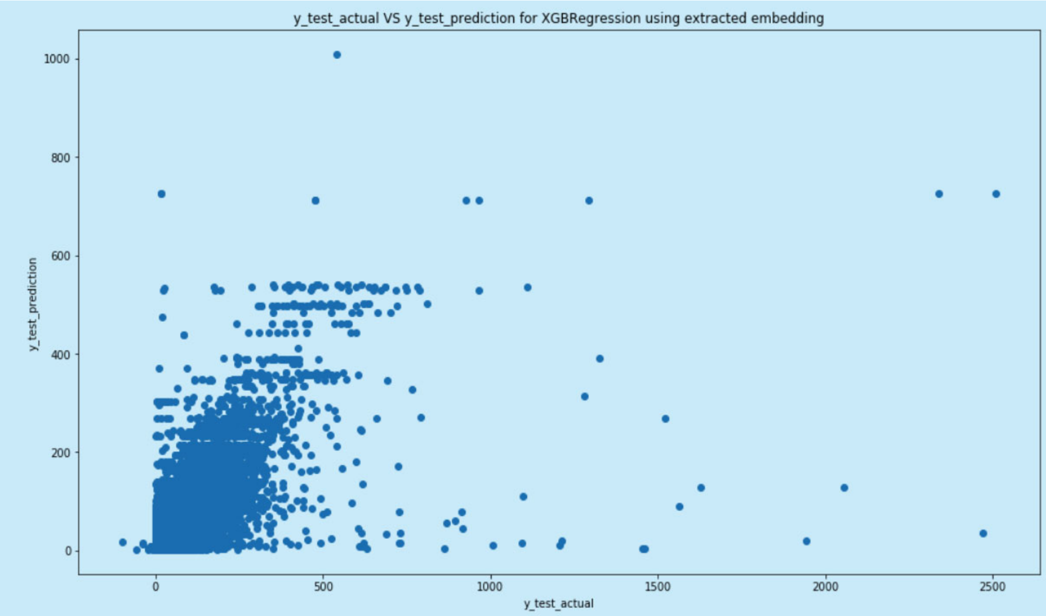
The embedding result of item’s family

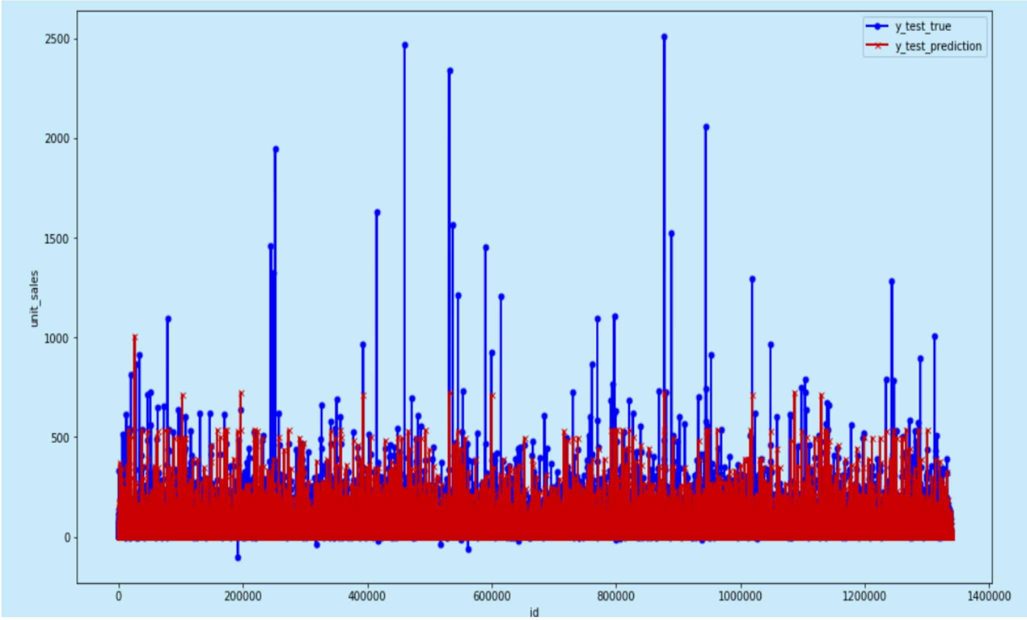


Model evaluation (3)

Forecasting models	Metrics				
	RMSE	MAE	R2	NWRMSE	NWRMSLE
Linear regression	Train:16.745 Test: 16.534	Train:6.972 Test: 6.961	Train:0.03 Test: 0.03	Train:17.059 Test:16.814	Train:0.948 Test: 0.948
XGBRegression	Train:12.640 Test: 12.471	Train:5.150 Test: 5.162	Train: 0.447 Test: 0.448	Train:12.856 Test: 0.6515	Train:0.734 Test: 0.736
Random Forest regression (estimators = 10, 'max_depth = 5, min_samples_leaf = 3)	Train:16.255 Test: 16.074	Train:6.707 Test: 6.695	Train: 0.086 Test: 0.083	Train:16.541 Test: 16.331	Train:0.906 Test: 0.906
Embedding	Train: 10.994 Test: 11.148	Train: 4.222 Test: 4.248	Train: 0.585 Test: 0.559	Train: 11.118 Test: 11.270	Train: 0.626 Test: 0.628
Linear Regression using embedding features	Train: 10.637 Test: 11.058	Train: 4.214 Test: 4.250	Train: 0.609 Test: 0.566	Train: 10.722 Test: 11.182	Train: 0.617 Test: 0.620
XGBRegression using embedding features	Train: 9.692 Test: 10.551	Train: 4.071 Test: 4.128	Train: 0.675 Test: 0.605	Train: 9.714 Test: 10.634	Train: 0.608 Test: 0.611
Random Forest regression (estimators = 10, 'max_depth = 5, min_samples_leaf = 3) using embedding features	Train: 10.518 Test: 10.982	Train: 4.232 Test: 4.270	Train: 0.617 Test: 0.572	Train: 10.601 Test: 11.103	Train: 0.621 Test: 0.623

The prediction outcome for XGBRegression using embedding features





What customer insights are provided by this analysis (1)

- The highest sale month of the year is December
- The lowest sale month of the year is February
- Sunday has the highest sale of the week, followed by Saturday
- Thursday has the least sale
- There is no relation between unit_sales and oil price
- A clear weekly periodicity of unit_sales time series for item 502231 from store 44 between 2013 and 2017 is observed
- The stationarity test (p value < 0.05) indicating the unit_sales time series for item 502231 and store 44 between 2013 and 2017 is stationary

What customer insights are provided by this analysis (2)

- Forecasting model using random forest indicates transaction, family, class, and item_nbr are most important predicting features

- Forecasting model using embedding representation achieves a decent performance

- Embedding feature (i.e., item_nbr and family embedding) can capture the hidden relationship between different items

- A large number of items with high item_nbr which are all unlikely to have a high unit sales. However, for all items with moderate item_nbr, they are more likely to have a high unit_sales

- Promotion has positive impact on predicted unit_sales

Future improvements (1)

- Data engineering
  - Spark/GPU
  - Elastic search
  - Knowledge graph

- Feature engineering
  - To include and construct complicated features such as mean, min, max, skewness, rolling over a sliding window, time series lags etc.,
  - To introduce more features such as unit_price of each item and product review/comments (e.g., positive, neural, or negative)

Future improvements (2)

- Forecasting models
  - Sequence model (e.g., LSTM), time series forecasting techniques (ARIMA, SARIMA, SARIMAX, Prophet etc.,)

- A new evaluation metric
  - Formula

$$Mfinancialloss = \frac{\sum_{i=1}^{N}\left[R_i * |\hat{y}_i - y_i|_{(\hat{y_i} < y_i)} + C_i * |\hat{y}_i - y_i|_{(\hat{y_i} > y_i)}\right]}{N}$$

  - where R_i denotes revenue loss per unit per item when under-estimating (stocking-out), and C_i denotes management cost incurred per unit per item when over-estimating (over-stocking)

Thank You!