Predict Future Sales

Manan Parasher (301350098), Ruchita Rozario (301359835), Slavvy Coelho (301359836)

Problem Statement

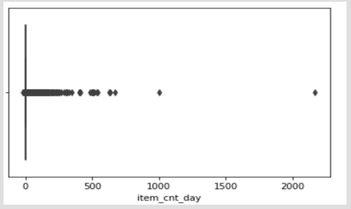
Sales forecasting uses past figures to predict short-term or long-term performance of the organization. We have developed a project that can predict total sales for every product and store in the next month.

Data Inspection

- Checked the testing and training dataset. Inspected the size, data types and types of columns.
- Inspected data for null values and missing data.

Data Pre-processing

- Removed the outliers for item price and item day count.
- Filtered the negative values and filled those values by averaging.



	shop_name	shop_id	shop_city	shop_type
0	орджоникидзе фран	0	якутск	unkown
1	тц центральный фран	1	якутск	тц
2	тц мега	2	адыгея	тц
3	трк октябрькиномир	3	балашиха	трк
4	тц волга молл	4	волжский	тц

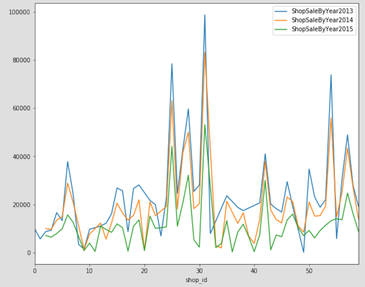
- Cleaned the data by removing unwanted characters from shop names and item names.
- Dropped duplicated shops from training data.

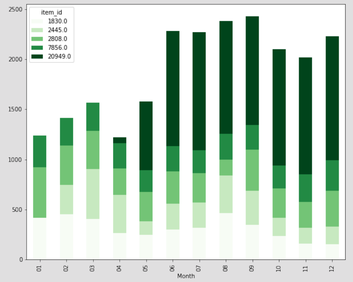
Exploratory Data Analysis

Performed EDA to discover potential patterns in the data. These patters are useful in handling the data more efficiently.

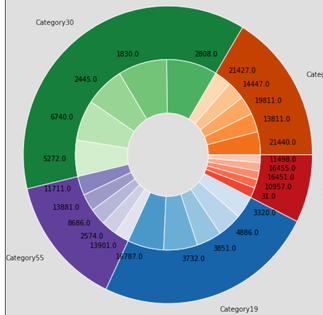
Checked shop wise sales trend.

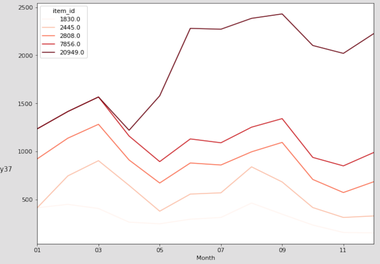
Grouped the data on item id to find patterns.





Checked sales trend on the basis of item categories and shop id.





Clustering

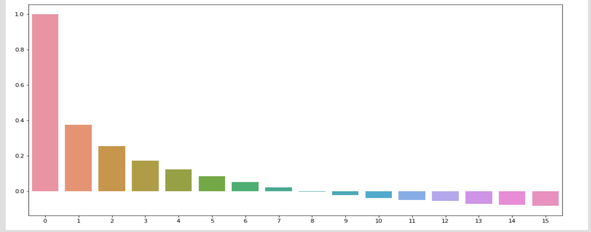
Used k-means Clustering to cluster the shops on different basis.

Transactions for shop
Categories by shop
Items they hold

Revenue by shop
Quarterly Sale
Increase in sales

Feature Engineering

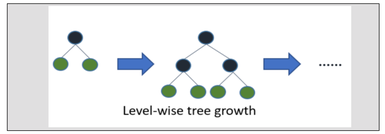
Described the autocorrelation between an observation and another observation at a prior time step using ACF.



Graph shows that the features 1, 2, 3, 4, 5, 10, 11, 12 should be used as lag features for item id.

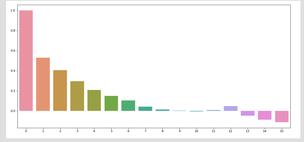
LightGBM Model

Light GBM is a gradient boosting framework that uses tree based learning algorithm. Light GBM grows tree leaf-wise while other algorithm grows level-wise. It will choose the leaf with max delta loss to grow.



Feature Engineering

- We have used Autocorrelation function (ACF) to determine how much the previous values of the features affect the prediction of the current one. Created lag features for the following:
- Shop/item pairs
- Items
- Shops
- Categories



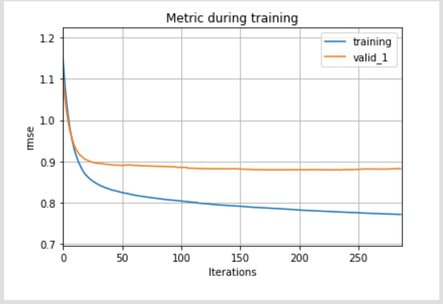
Data Preparation

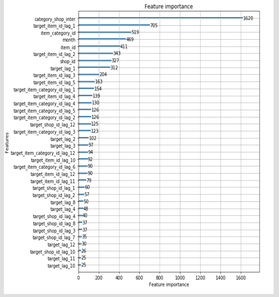
- Added lag and date features to the training data.

	shop_id	item_id	date_block_num	target	item_category_id	target_lag_1	target_lag_2	target_lag_3	target_lag_4	target_lag_8	...	targe
4456026	2	27	12	0.0	19	0.0	0.0	0.0	0.0	0.0	...	
4456027	2	30	12	0.0	40	0.0	0.0	0.0	0.0	0.0	...	
4456028	2	31	12	0.0	37	0.0	0.0	0.0	0.0	0.0	...	
4456029	2	32	12	1.0	40	0.0	0.0	0.0	0.0	0.0	...	
4456030	2	33	12	1.0	37	1.0	2.0	0.0	0.0	0.0	...	
4456031	2	34	12	0.0	40	0.0	0.0	0.0	0.0	0.0	...	
4456032	2	36	12	0.0	37	0.0	0.0	0.0	0.0	0.0	...	
4456033	2	37	12	0.0	40	0.0	0.0	0.0	0.0	0.0	...	
4456034	2	39	12	0.0	41	0.0	0.0	0.0	0.0	0.0	...	
4456035	2	40	12	0.0	57	0.0	0.0	0.0	0.0	0.0	...	

Model Training and Inference

- Used predefined estimators and early stopping and trained the model.
- MSE between validation and training showed model needs to be improved.
- Performed hyper-parameter tuning to get better results.
- Tweak the values of leaves and max depth to improve Accuracy.
- Feature importance graph showed that “category_ shop_id” lag feature was the most useful.





LSTM Model

A Recurrent Neural Network (RNN) is a type of neural network well-suited to time series data. RNNs process a time series step-by-step, maintaining an internal state summarizing the information they've seen so far. We have used a special RNN layer called Long Short Term Memory (LSTM)

Data Processing

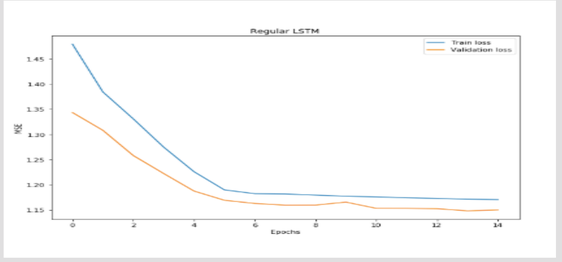
- Aggregated the data by month as the data was given on a daily basis.
- Visualized the distributions of the test set and filtered the “item_cnt” values between 0 and 20.
- Used "item_cnt" feature to perform univariate time series prediction using pivot function.
- Replaced the missing “shop_id” and “item_id” values with 0.

Model Preparation

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 33, 64)            16896
_________________________________________________________________
dropout (Dropout)            (None, 33, 64)            0
_________________________________________________________________
lstm_1 (LSTM)                (None, 32)                12416
_________________________________________________________________
dropout_1 (Dropout)          (None, 32)                0
_________________________________________________________________
dense (Dense)                (None, 1)                 33
=================================================================
Total params: 29,345
Trainable params: 29,345
Non-trainable params: 0
_________________________________________________________________
```

Model Training

- Trained the model for 20 epochs. Training Vs Validation loss showed that model worked good.



Model Inference

- Using dropout rates to over come model over-fitting was effective in increasing the accuracy of the model.

Ensemble Model

- Combined the LightGBM and LSTM model to create an ensemble model.
- Achieved a score of 0.88904 using the above model.