CMPE257 Project  
Group 2:

**Predict Future Sales**

**Team members:**

1. **016043755 – Sushmitha Dhummi Thrilochana**
2. **015215655 - Pratibha Awasthi**
3. **015239549 - Ishaan Bhalla**
4. **015016818 - Sai Sri Govind Chennu**

**Table of Contents**

[Introduction 3](#_Toc103547758)

[Data Description 3](#_Toc103547759)

[Objective 4](#_Toc103547760)

[Literature Review and related work: 4](#_Toc103547761)

[Other related published work: 5](#_Toc103547762)

[Data Exploration and Data Pre-processing: 6](#_Toc103547763)

[Data Exploration: 6](#_Toc103547764)

[Visualizing Data 9](#_Toc103547765)

[Data cleaning: 12](#_Toc103547766)

[Problem Statement and Methods applied 14](#_Toc103547767)

[Prophet 14](#_Toc103547768)

[Observations: 17](#_Toc103547769)

[Advantages of Prophet: 18](#_Toc103547770)

[Feedback & Experience using Prophet: 18](#_Toc103547771)

[Long Short-Term Memory (LSTM) 19](#_Toc103547772)

[Why LSTM model for Time series data? 19](#_Toc103547773)

[Steps to create LSTM model for time series data 19](#_Toc103547774)

[Test Results: 21](#_Toc103547775)

[LightGBM 22](#_Toc103547776)

[Steps for building model: 22](#_Toc103547777)

[XGBoost: 25](#_Toc103547778)

[Time Series Data Preparation 25](#_Toc103547779)

[XGBoost Training 25](#_Toc103547780)

[Feature importance 26](#_Toc103547781)

[Results: 27](#_Toc103547782)

[Kaggle Submissions: 27](#_Toc103547783)

[Future Scope: 27](#_Toc103547784)

[Glossary: 27](#_Toc103547785)

[References: 28](#_Toc103547786)

# Introduction

Sales forecasting is a frequent application of Machine Learning. Businesses can use this forecasting to identify benchmarks, determine incremental impacts of new initiatives, plan resources in response to expected demand, and project future budgets. This report provides detailed Machine Learning based solutions to the Kaggle competition - [Predict Future Sales](https://www.kaggle.com/c/competitive-data-science-predict-future-sales). We have implemented different Machine Learning models to produce forecasted output for the given dataset and concluded the best results with Kaggle ranking.

# Data Description

The dataset provided to us is a time-series daily historical sales data. It consists of **2,170 items** sold by **60 shops** between **January 2013 to October 2015**.

The data set consist of 6 csv extension files which are given below with their descriptions:

1. **sales\_train.csv:** This is the training set which consist of historical data from January 2013 to October 2015.
2. **test.csv:** This is the test set. We are expected to forecast the sales for these given shops and products for **November 2015.**
3. **sample\_submission.csv:** This file exhibits the correct format expected.
4. **items.csv:** supplemental information about the items categories.
5. **shops.csv**: supplemental information about the shops.

The Data fields present in these with their descriptions are given below:

1. **ID** - an Id that represents a (Shop, Item) tuple within the test set
2. **shop\_id** - unique identifier of a shop
3. **item\_id** - unique identifier of a product
4. **item\_category\_id** - unique identifier of item category
5. **item\_cnt\_day** - number of products sold. You are predicting a monthly amount of this measure
6. **item\_price** - current price of an item
7. **date** - date in format dd/mm/yyyy
8. **date\_block\_num** - a consecutive **month number**, used for convenience. January 2013 is 0, February 2013 is 1 , . . . , October 2015 is 33
9. **item\_name** - name of item
10. **shop\_name** - name of shop
11. **item\_category\_name** - name of item category

The dataset can be download from [here](https://www.kaggle.com/c/competitive-data-science-predict-future-sales/data).

# Objective

The competition requires us to predict the future total sales to happen in the next month (Nov 2015) for every item and store in the test file.

The submissions are evaluated by **root mean squared error (RMSE)** with true target values clipped into [0,20] range.

# Literature Review and related work:

**Why predict future sales?**

Sales forecasting is a technique that uses historical sales data as inputs to make informed predictions about the direction of future trends.

* **Manage supply chain efficiently:** Knowing future consumer trends allow business’ sales operations align their supply chain activities efficiently like material purchases, inventory stocking, warehouse capacity plans, hiring and handle the market demand most efficiently with improvised decision making.
* **Make higher Revenue:** It enables companies to focus their sales team on high-profit sales opportunities resulting in higher revenue.
* **Incorporate right changes:** It enables companies to incorporate the right changes like pricing, marketing, product changes, locations, hiring etc for improved business outcomes.

**Different Sales Forecasting Techniques:**

1. **Qualitative Methods:**

* **Market Research:** A systemic process of actively surveying or interviewing potential customers to determine the interest of service or product.
* **Delphi Method:** A panel of experts is interviewed by a sequence of questionnaires enabling forecaster to have all information for forecasting.
* **Visionary Forecast**: It is a non-scientific method where a 'visionary' or 'futurologist' attempt to forecast through subjective opinion, guesswork and imagination.

1. **Time Series Analysis and Projection:**

* **Moving Average:** It is technical indicator that investors and traders use to determine trend direction, and seasonal irregularities. It is calculated by adding up data points during specific period and dividing by number of time periods.
* **Exponential smoothing:** This is similar to moving average except that more recent data points are given more weight. Applied mainly for production and inventory control.
* **Trend Projection:** This technique fits a trend line to a mathematical equation and then projects it into the future by means of this equation. It is typically used to forecast new-products and long-term sales.

1. **Casual methods:**

* **Regression model:** This functionally relates sales to other economic or internal variables to estimate an equation using least-square error technique. It is good for short-term predictions.
* **Life-cycle Analysis:** The product acceptance by various groups is analysed to forecast product growth rates.

## Other related published work:

1. [Sales prediction using machine learning approaches](https://aip.scitation.org/doi/abs/10.1063/5.0068655) [AIP Conference Proceedings 2387, 140038 (2021)]
2. "[Intelligent Sales Prediction Using Machine Learning Techniques](https://ieeexplore.ieee.org/document/8659115)," S. Cheriyan, S. Ibrahim, S. Mohanan and S. Treesa, 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE), 2018;

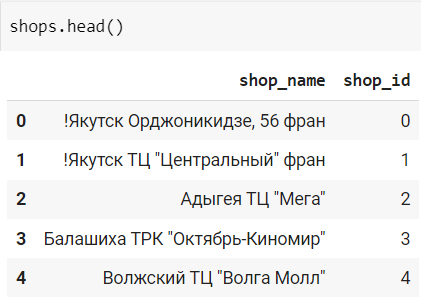
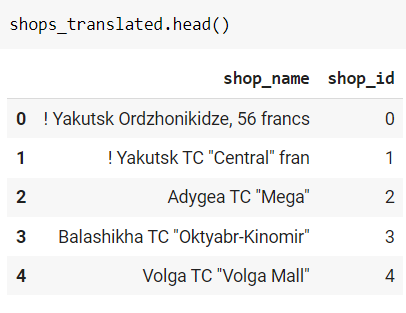
# Data Exploration and Data Pre-processing:

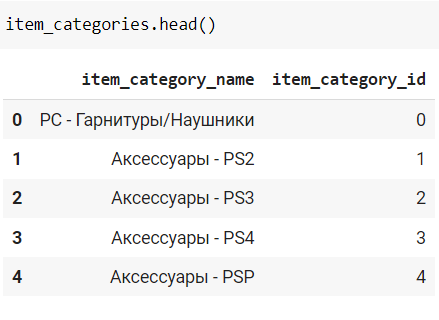
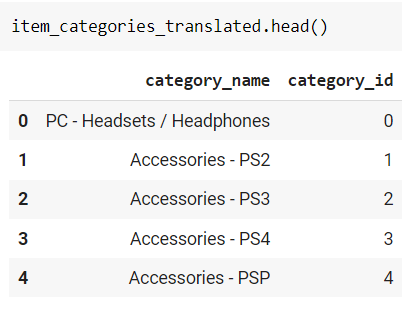
## Data Exploration:

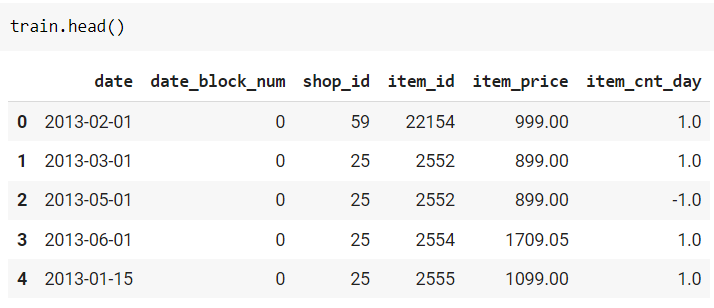
The very first step in data analysis is to explore and visualise the unstructured data to uncover patterns, characteristics, and points of interest. It creates a broader picture of important trends and points that require further study. It also gives us an idea of the amount of cleaning required in the data. Given below are some glimpses of data set.

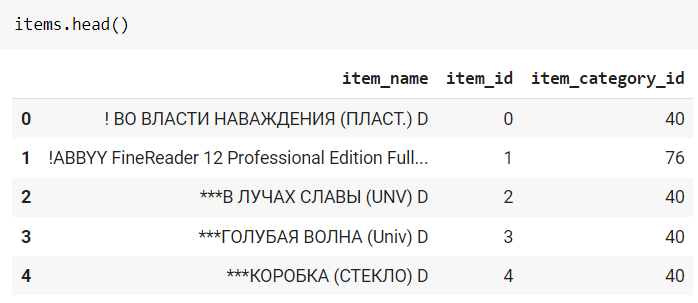
We translated the original dataset in Russian language to English for better understanding and to see if there is any scope for feature extraction.

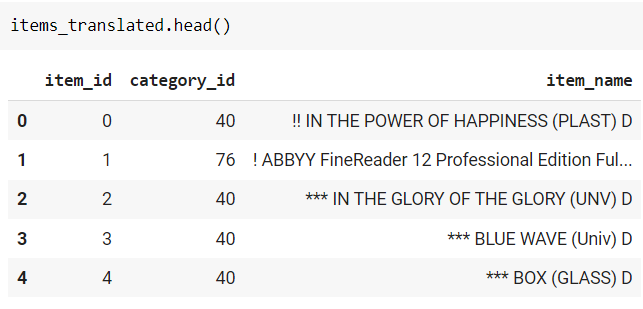
**Original Dataset in Russian language Translated dataset in English**

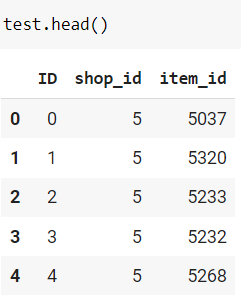
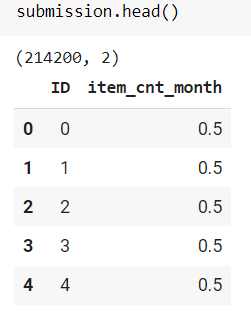
 

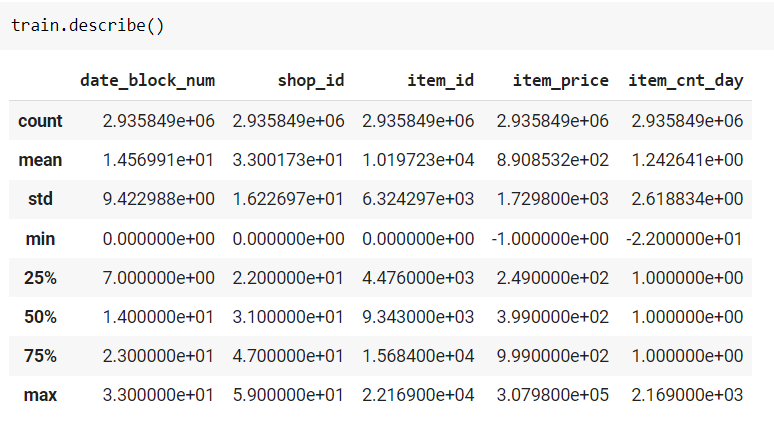
 



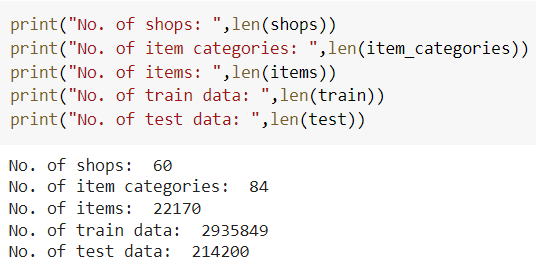




****

**Total Number of samples in each set:**



**Number of Unique items in Train and Test set:**

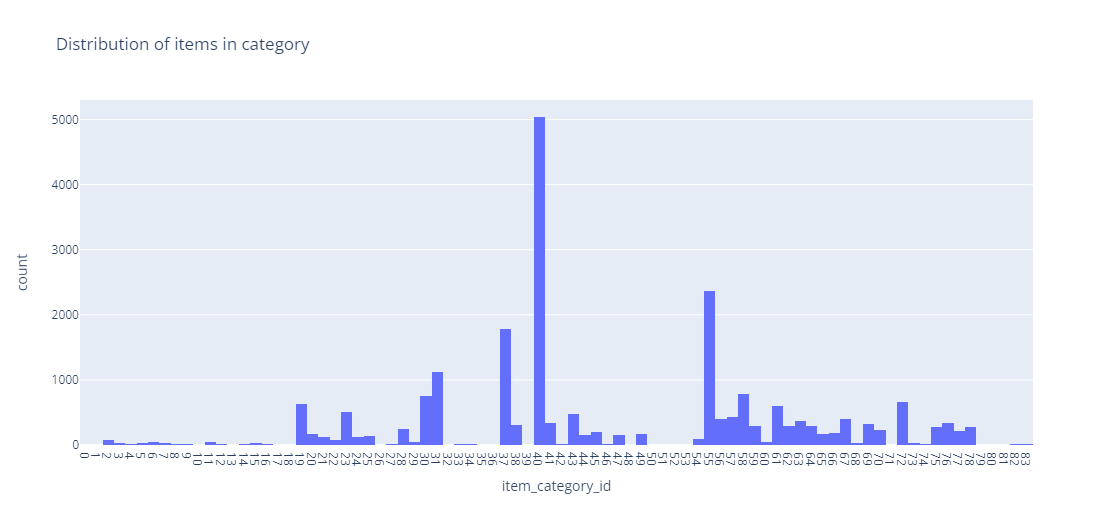
|  |  |  |
| --- | --- | --- |
| Dataset | Train set | Test Set |
| Shops | 60 | 42 |
| Items | 22,170 | 5100 |
| Item Category | 84 | - |
| Total Samples | **2,935,849** | **214,200** (42 x 5100) |

From exploration we can conclude that we only need to forecast sales for 5,100 items for 42 shops. Hence, we may not include all shops and items to reduce computing resources required to train models.

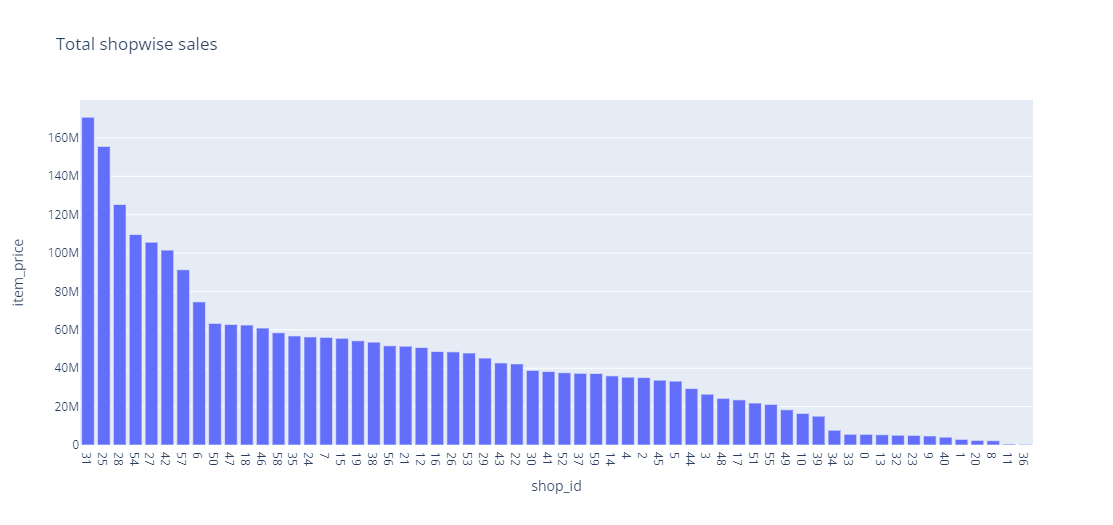
# Visualizing Data

For data visualization, we have mainly used Plotly, Seaborn and Matplotlib python visualization libraries.

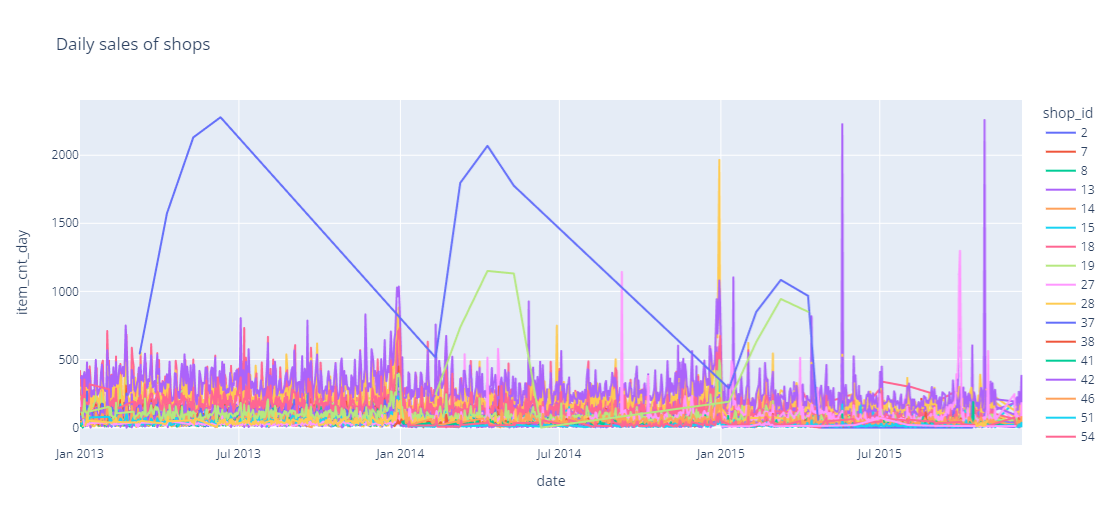
**1. Distribution of items in each category:** We can see the distribution of items among 83 categories. Item category 40 has highest number of items.



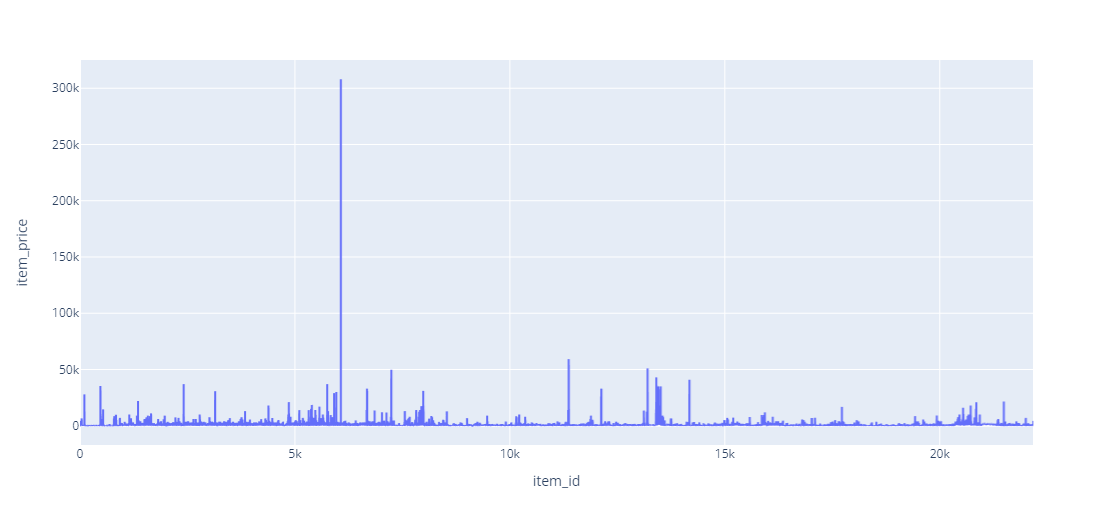
1. **Total sales made by each shop over the span of 34 months.**



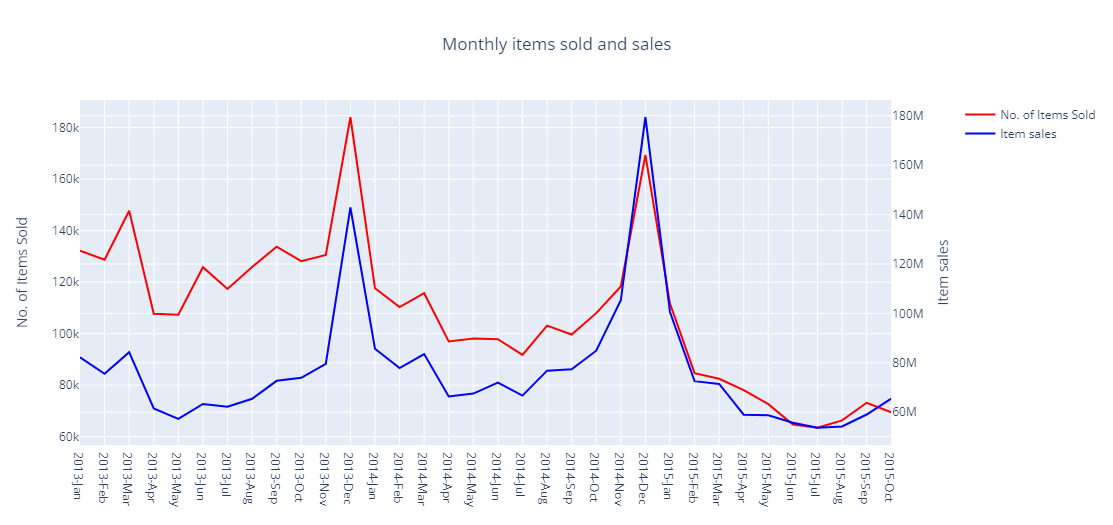
1. Below plot shows **the total number of items sold by each shop on a day** over the span of 34 months. We can see that shop no. 9 opens sporadically and makes huge sales when opened. We also see peaks in the year ends sales.



1. **Plot of item prices of each item:** We can most items are pretty much in same price range except for one item (6606) which is way high. This is clearly an outlier.



1. **Below plot shows the total sales and number of items sold in a month:** We can see there is **seasonality** in the sales trend. The sales seem to peak in the year end, and then follows a decreasing trend.



# Data cleaning:

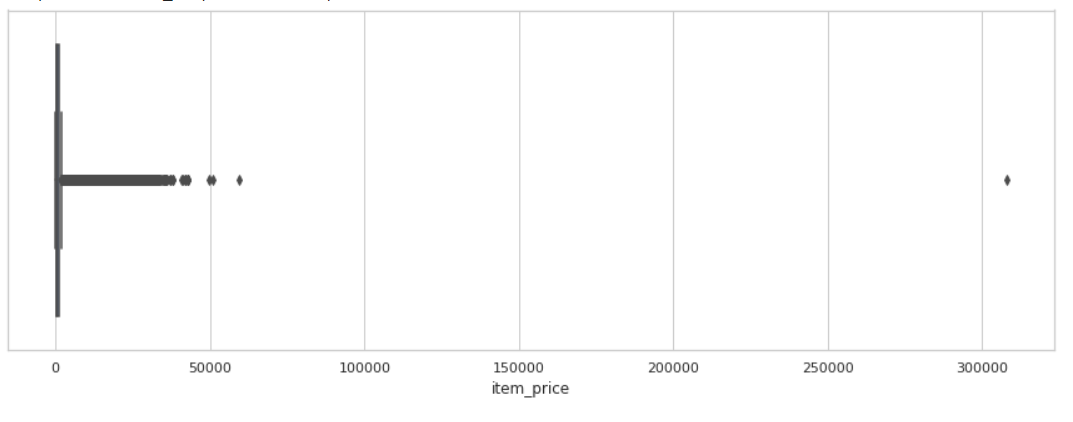
Data in its true form is raw and not usable. It needs to be cleaned and produced in a form that is more readable and usable. The practice of modifying or altering data in order to make it more understandable and structured is known as data manipulation. It enhances the quality of the data for future modelling purposes. Following are some of the steps performed to clean the data

From above plots, there are clearly some outliers that needs to be treated.

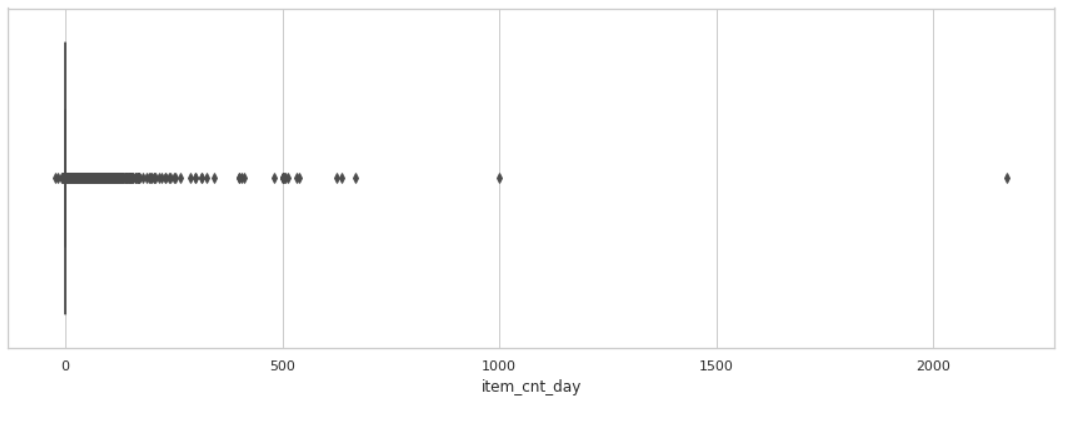
An outlier is an observation or value that lies an abnormal distance from other values in a given sample. These are stranglers that can be extremely high values or extremely low values. This can be variability in the measurement or it can sometimes indicate an error during experiment. Usually, outliers can lead to misleading interpretations and hence are advised to be removed before training a model.

Detecting outliers using box plot.

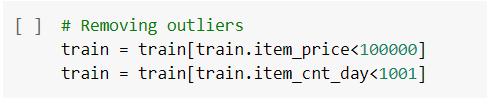
**1**. **Box plot of item\_price feature:** We see there is one particular item having price above 300k, far away from rest of the sample.



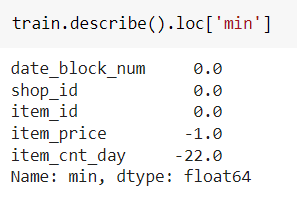
1. **Box plot of item\_cnt\_day feature:** We see there is one sample with item count more than 2000.



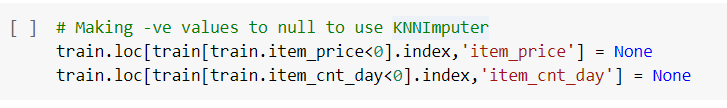
We simply remove these outlier samples as they can skew the training considerably.

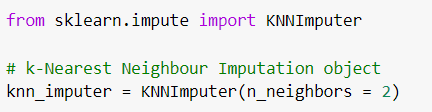


1. **Handling negative values:** We see train data has samples with negative item\_price and negative item count.

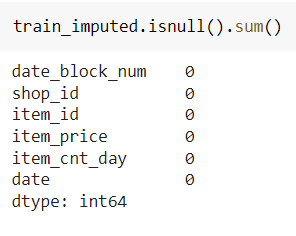


Since item price and count is not fixed and vary with months, we have handled them by making the negative values to null and imputing them using Scikit-learn’s **KNNImputer** that uses K-Nearest Neighbour algorithm to assign null values with values of it’s closes neighbouring sample.





1. **Handling null values:** This dataset has no null values.

****

# Problem Statement and Methods applied

Our task is to forecast the total amount of products sold in every shop for the test set. We had applied following models:

1. Prophet
2. LSTM
3. ARIMA
4. LightGBM
5. XGBoost

## Prophet

Facebook Prophet is an open-source local Bayesian structural time series model. It is an additive model, that is, non-linear trends are fit yearly, weekly and daily seasonality including the holiday effects.

Simplified more, in order to provide final forecasted output, it parallelly computes and considers following parameters:

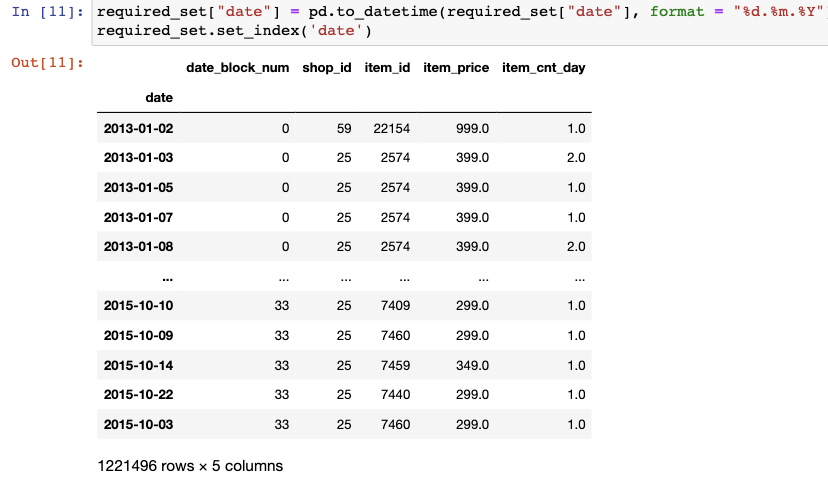
1. Linear logistic growth curve trend,
2. Yearly seasonal component,
3. Weakly seasonal component, and
4. List of important holidays.

**Step 1:** The necessary packages, libraries and dataset was installed (included in above section).

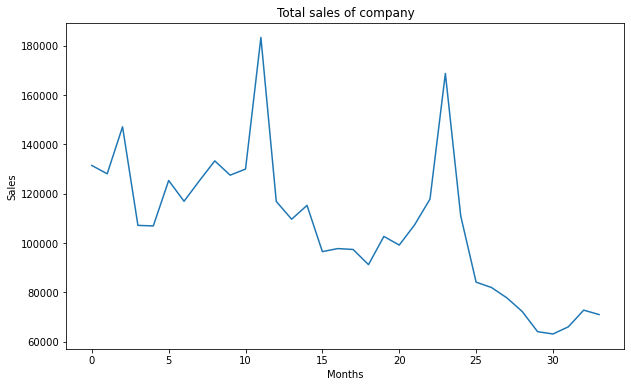
**Step 2:** Later, the dataset was visualised to get a overall idea of the data (included in above section).

**Step 3:** Outliers were removed from the train data (included in above section).

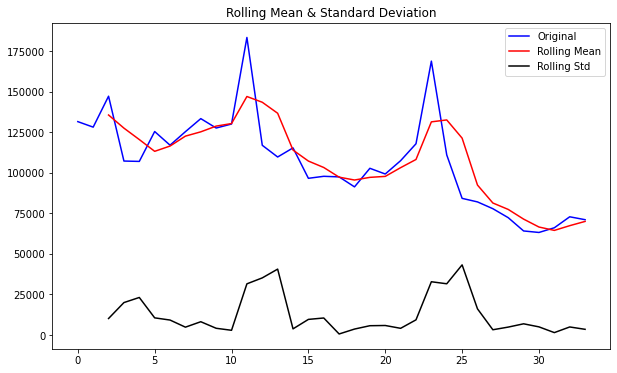
**Step 4:** Changed the date column from object type to date time series type to be used as input.



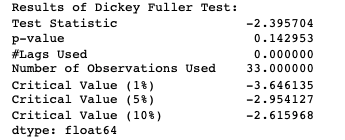
**Step 5:** Group by the total month sales (34 months)



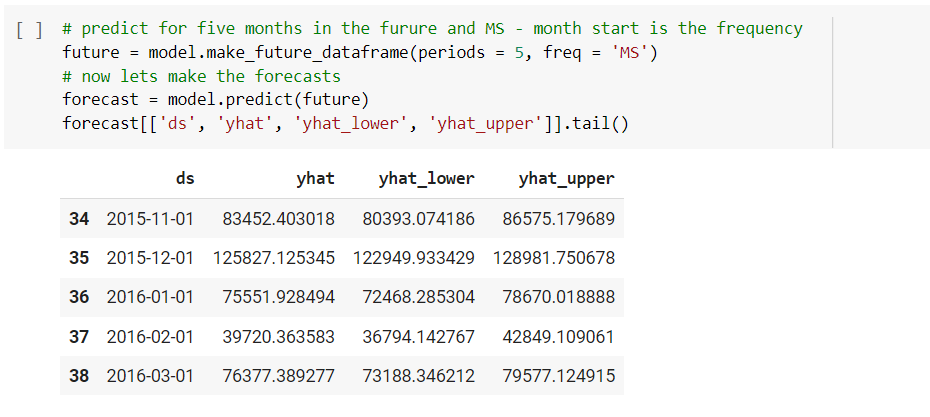
**Step 6:** Calculated the rolling mean and rolling standard deviation and plotted the graph.



**Step 7:** Performed Augmented Dickey-fuller test to check the stationarity.

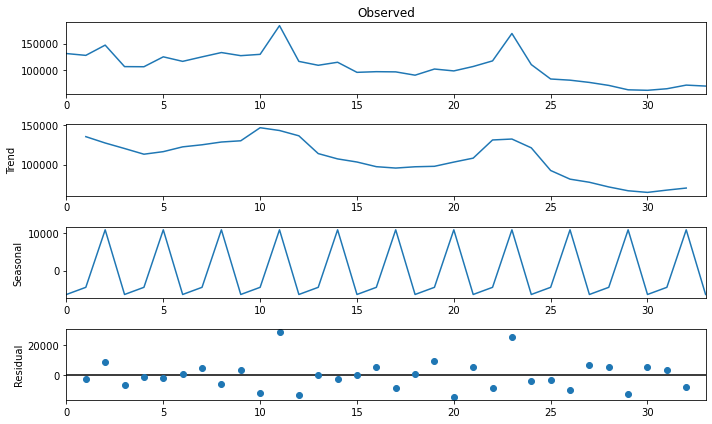


**Step 8:** Prophet requires input as a pandas data frame having only 2 columns - 'ds' and ’y’. ‘ds’ column should have the date series and ‘y’ should have corresponding item count values.

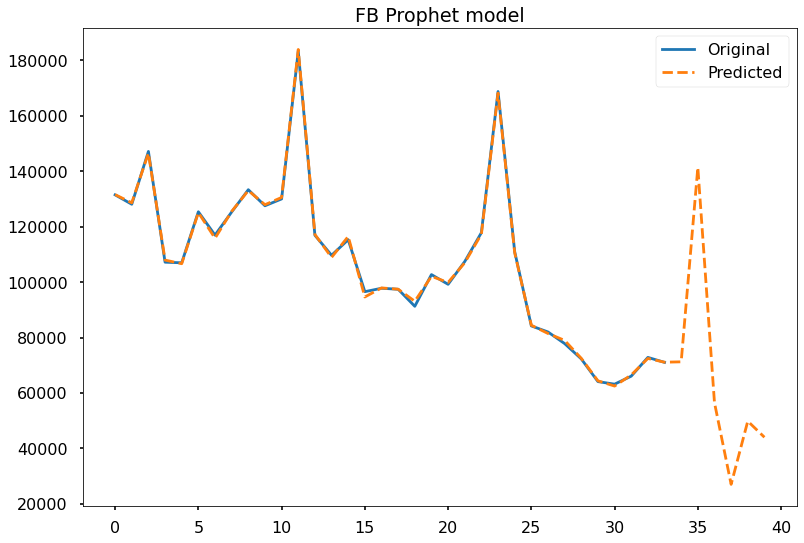


### **Observations:**

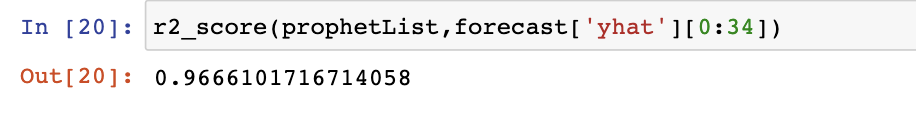
In the graph below x axis represents the month whereas y axis represents the item\_cnt or sales. We can observe that the sales decreases with time but peaks seasonally.

As mentioned, the Facebook’s prophet is an additive model. It means it takes in account all: the trend, the seasonality and the residual to produce its forecasting output 

After training the model, the graph is plotted to represent the prediction in comparison to original. In the below graph, orange dotted line shows the prediction with blue in contrast showing the original sales, where, x axis represents the months and y represents the sales.



The r2 score is the measurement of the performance of a regression-based learning model. It measures the amount of variance in the prediction.



The r2 percentage observed here is 96.6%

### Advantages of Prophet:

1. It requires less hyperparameter tuning but only for business time series as it was specifically designed for business time series prediction.
2. It claims to be accurate and fast because it uses the stan platform for computation and predictions.
3. It claims to be easy to use for people with no prior knowledge.
4. It can handle outliers and other data issues itself. It fits them in the trend.

### Feedback & Experience using Prophet:

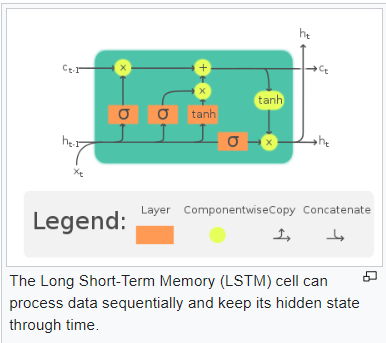
It took us 2-3 hours to install its packages and libraries. Because not a lot of information is available, even on Meta’s official document, it get difficult to debug the issues faced during installation. Among the group of 4 (with 4 systems), we could only be able to install it in one of the systems.

During the execution of code, the Kaggle competition required to submission of limited shops whereas we forecasted for all the shops. In order to derive for the desired shops, we plotted and extracted in a matrix. The execution of the code took more than 12 hours and eventually led to crashing of the chrome. We couldn’t make the submission for prophet over the Kaggle. Hence, we feel that prophet is tedious to use and require powerful GPUs and/or CPUs. But it provides with good result.

## Long Short-Term Memory (LSTM)

LSTM networks are extension of RNN that was mainly developed to deal with vanishing gradient problem that can be encountered when training traditional RNNs. However, LSTM can still suffer the exploding gradient problem.

RNNs with LSTM units categorize data into short-term and long-term memory cells. This enables RNNs to figure out which data is important and should be remembered and looped back into network and which data can be forgotten. LSTM use 3 gates called input, output and forget gate to determine this. The gates are analog form of sigmoid ranging from 0 to 1.



<https://en.wikipedia.org/wiki/Long_short-term_memory>

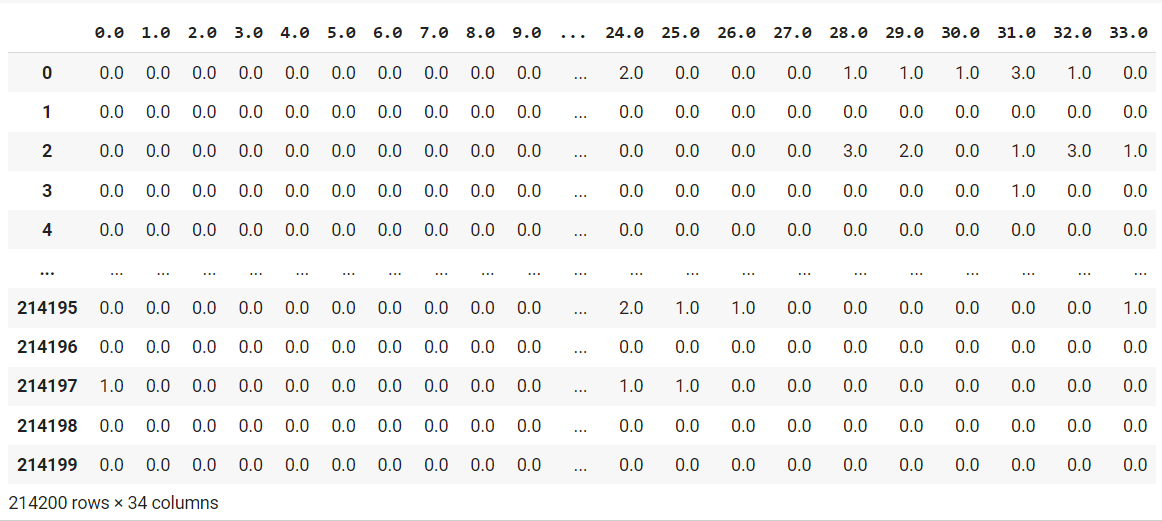
### Why LSTM model for Time series data?

* The LSTM can capture both long-term seasonality, such as a yearly pattern, and short-term seasonality, such as weekly patterns.
* The different gates within LSTM improve its ability to capture nonlinear relationships for forecasting. Causal factors have a nonlinear impact on demand. When these variables are included as input variables, the LSTM can learn the nonlinear relationship for forecasting.
* The LSTM could take inputs with different lengths. This feature is especially useful when LSTM is used to build general forecasting models for specific customers or industries.

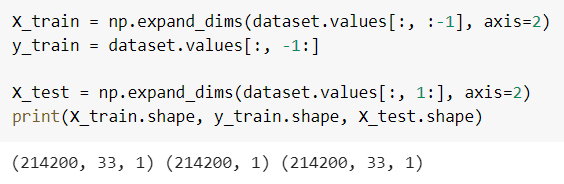
### Steps to create LSTM model for time series data

**Step 1: Explore and clean** the data as mentioned in the data cleaning part of the report.

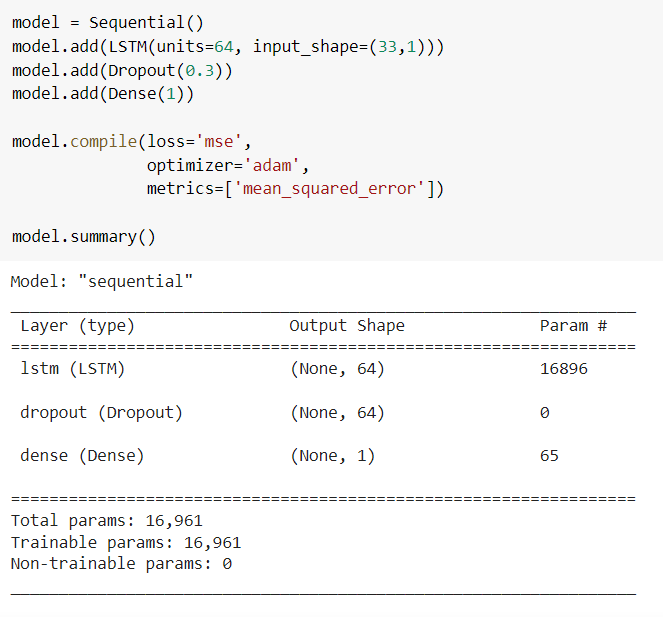
**Step 2:** **Create (Item, Shop) vs month matrix** having total monthly item as values from train set for only those shop item pairs in test data.



**Step 3:** **Split data** to train and test and **reshape** the array.



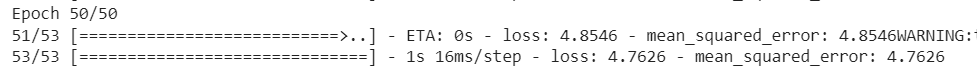
**Step 4:** Build and train model.

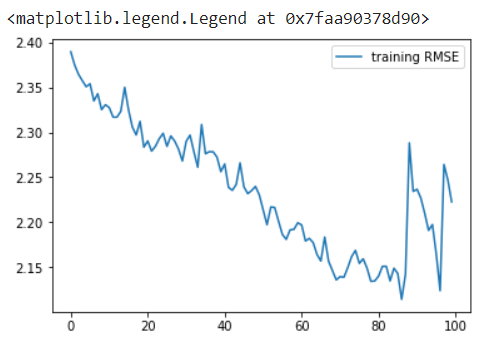




### Test Results:

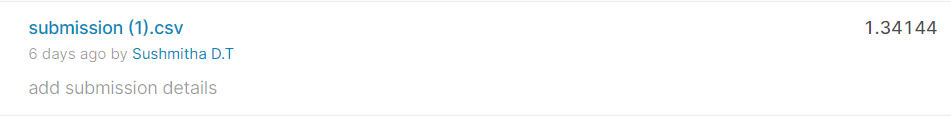
We obtained MSE score 4.7.





**Step 5: Predict for test dataset** provided in the Kaggle competition and submit the prediction in right format.

Upon making submission to Kaggle competition for the Kaggle test set, we got not so great **1.34** RMSE score placing us in **11k rank**. We drop the model for this dataset and explore Gradient Boosting algorithms – LightGBM and XGBoost for better accuracy.



## LightGBM

Since Prophet model would not scale well for large dataset with many multiple time series, we decided to employ algorithm that is fast and scale easily for large dataset while giving good accuracy. LightGBM fits perfectly for this requirement.

* **Light GBM** is a fast, distributed, high-performance gradient boosting framework based on decision tree algorithm. It is designed to be distributed and efficient with the following advantages:
* Faster training speed and higher efficiency:
  + Light GBM uses **leaf wise splitting** over depth-wise splitting which enables it to converge much faster
* Lower memory usage.
* Better accuracy.
* Support of parallel, distributed, and GPU learning.
* Capable of handling large-scale data.

### Steps for building model:

**Step 1: Explore and clean** the data as mentioned in the data cleaning part of the report.

**Step 2: Perform feature Engineering**

1. **Translate the language:** Since the original data is in Russian language, we need to convert to English in order to understand the data and determine which part of features can be extracted.

There are many online tools that can be used for translating language in CSV. We used [Aspose](https://products.aspose.app/cells/translation/csv#:~:text=Translate%20CSV%20online,-Free%20online%20CSV&text=Upload%20your%20CSV%20files%20to,a%20download%20link%20to%20email.) tool

1. **Inspect and extract meaningful features**. Below are some of the features extracted.
2. **group**, **group\_id** from **item\_category\_name** feature
3. **shop\_city** from **shop\_name**
4. **ItemNameFirst4**, **ItemNameFirst6**, **ItemNameFirst11** from **Item\_name** feature.
5. **revenue** from **item\_price** feature
6. **first\_sale\_day**, **week\_day** from **date** feature
7. **shop\_total\_sales** from sum of group by of **shopid**, **weekday**
8. **shop\_day\_sales** from sum of group by of **shopid**
9. **sale\_day\_quality** from **shop\_total\_sales/** **shop\_day\_sales**

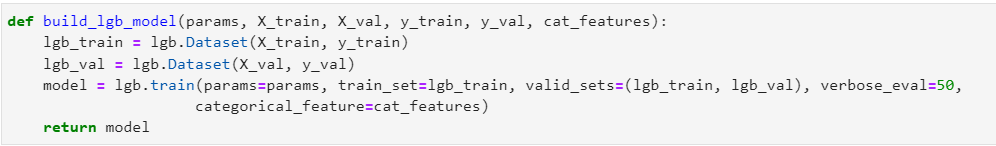
**Step 3:** **Create lag features** by shifting all the sales values by 1.

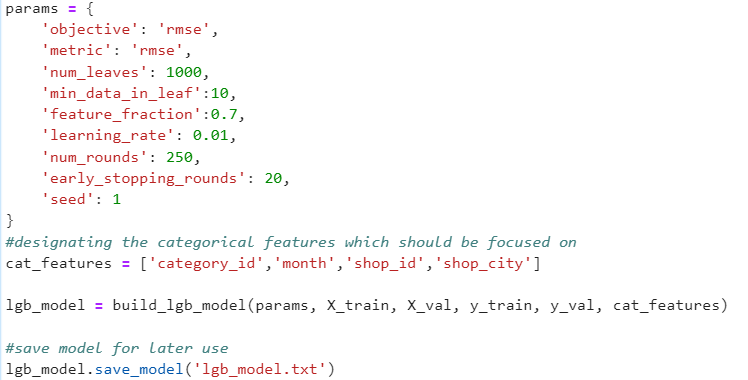
Lags are extremely useful in time series analysis due to a phenomenon known as autocorrelation, which is the tendency of values within a time series to be correlated with previous copies of itself. One advantage of autocorrelation is that it allows us to identify patterns within time series, which aids in determining seasonality, or the tendency for patterns to repeat at regular intervals.

**Step 4:** **Create rolling means features** by to capture the seasonality in the sales through the year.

**Step 5: Feature encode:** Select and encode the required features appropriately using one hot or label to train the model.

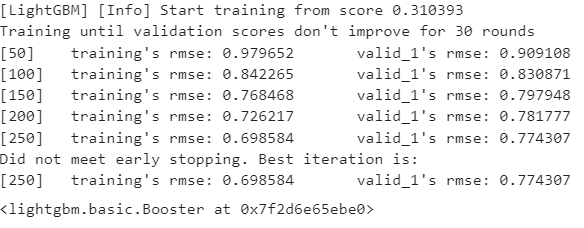
**Step 6: Build model:** Perform grid search to tune parameters and fit the model to train data.





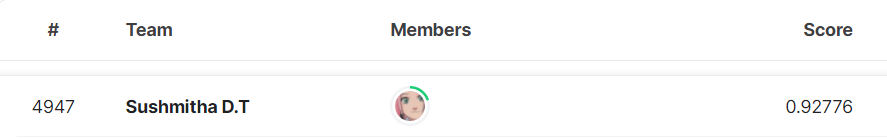
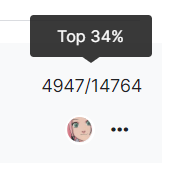
**Training Results:**

Training for 250 iterations took about 1 hour and gave descent test RMSE score of **0.77**.



**Step 7: Predict for test dataset** provided in the Kaggle competition and submit the prediction in right format.

Upon making submission to Kaggle competition for the Kaggle test set, we got **0.92** RMSE score placing us in **4945th rank** which is **top** **34%.**



## XGBoost:

XGBoost is an efficient implementation of gradient boosting for classification and regression problems.

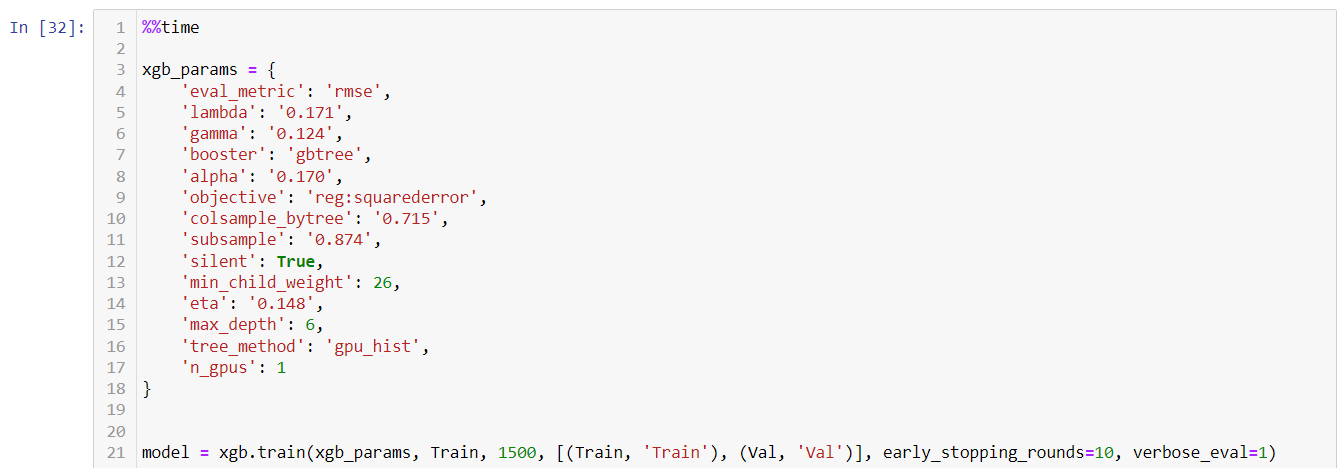
It is both fast and efficient, performing well, if not the best, on a wide range of predictive modelling tasks and is a favourite among data science competition winners, such as those on Kaggle.

XGBoost can also be used for time series forecasting, although it requires that the time series dataset be transformed into a supervised learning problem first.

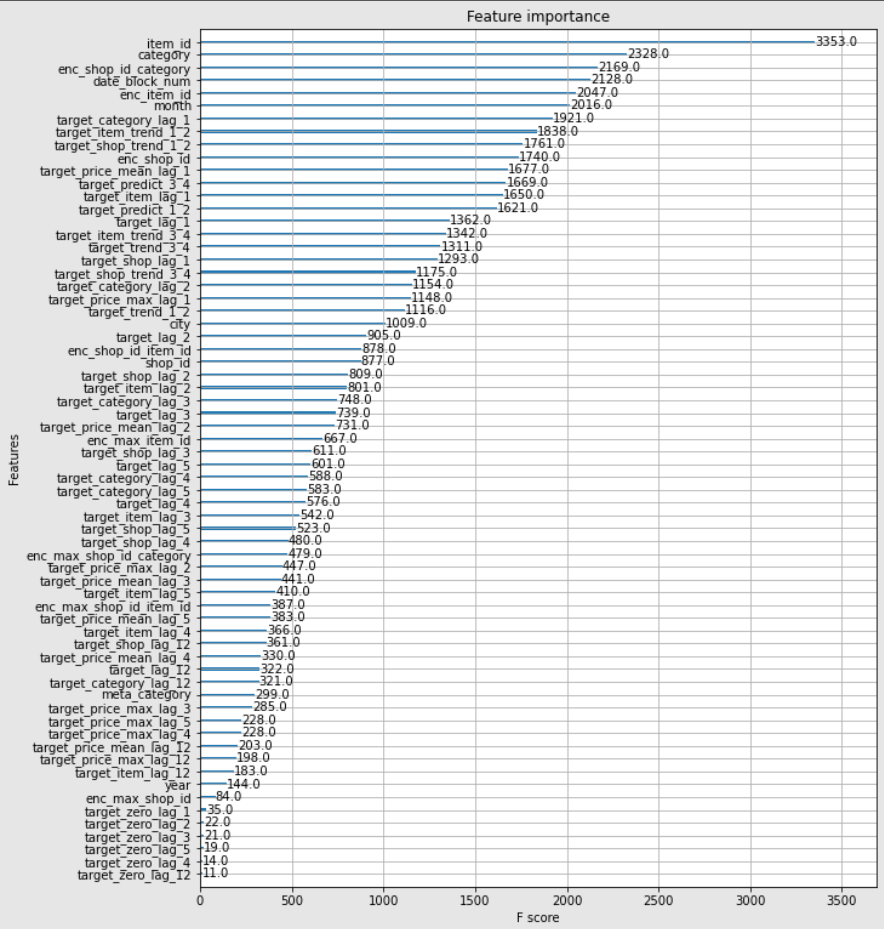
### Time Series Data Preparation

1. Adding new features: Performed feature extraction and added new columns - meta category, category and city to the train dataset
2. Lag features: Given a sequence of numbers for a time series dataset, we can restructure the data to look like a supervised learning problem. We can do this by using previous time steps as input variables and use the next time step as the output variable.
3. Mean encoding features
4. Features Interaction: We create new columns of target trends based on lag features.

### XGBoost Training



### Feature importance

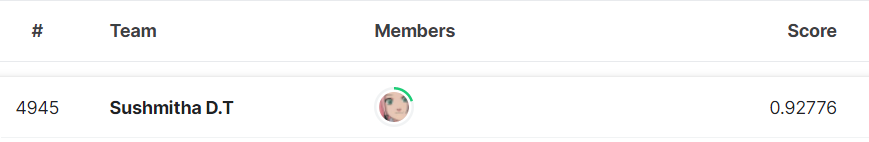


# Results:

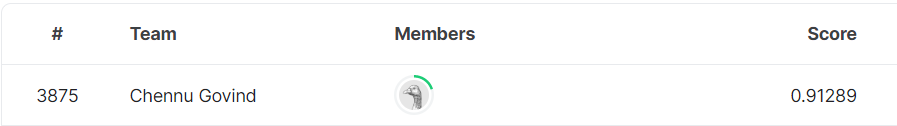
|  |  |  |
| --- | --- | --- |
| Model | Result | Kaggle Ranking  Out of 14,764 |
| Prophet | Takes lot of time to train for whole dataset with multiple time series. Best to predict for few shops and items |  |
| LSTM | Test RMSE: **1.24099** | **11,400s** |
| LightGBM | Test RMSE: **0.92776** | 4945 (top 34%) |
| ARIMA | Test AIC: **438.095** | **6789** |
| XGBoost | Test RMSE: **0.77729** | **3875** |

## Kaggle Submissions:

1. LightGBM:



1. XGBoost: Best score



# Future Scope:

There is a lot of scope to improve the prediction and get better Kaggle raking. We can tune model’s hyperparameters and try different periods of lags to bring down the RMSE score and make better predictions.

# Glossary:

1. Time Series Dataset: Time series data also known as time-stamped data is a sequence taken at successive equally spaced points in time. Example, counts of sunspots
2. Bayesian Model: Statistical model that uses probability to represent all uncertainty regarding input and the output considering that the features are independent to each other. It is often used in forecasting time series data based on an additive and is based on bayes’ theorem.

Formula for bayes’ theorem:

P(A|B) = P(B|A)(PA)

P(B)

Where:

P(A|B) – the probability of event A occurring, given event B has occurred

P(B|A) – the probability of event B occurring, given event A has occurred

P(A) – the probability of event A

P(B) – the probability of event B

1. Trends: The increasing or decreasing value in the series. It suggest the is the overall direction of the data.
2. Seasonality: The repeating short-term cycle in the series. It is a is a periodic component.
3. Residual: It is the left over of when the trend and seasonality are removed. It can be called as random fluctuations.
4. Rolling Mean: Rolling mean or rolling average is a calculation where a series of averages are created for different subsets of the complete data set. It is used to analyse the data points.
5. Rolling Standard Deviation: It is a calculation that gives no prediction but serves as a confirming indicator of market direction.
6. Dickey fuller test:

# References:

1. <https://www.kaggle.com/competitions/competitive-data-science-predict-future-sales/overview>
2. <https://digitaltesseract.com/facebook-prophet-an-overview/#:~:text=Facebook's%20Prophet%20is%20accurate%20and,other%20data%20issues%20by%20itself>.
3. <https://research.facebook.com/blog/2017/02/prophet-forecasting-at-scale/>
4. <https://facebook.github.io/prophet/>
5. <https://towardsdatascience.com/implementing-facebook-prophet-efficiently-c241305405a3>
6. <https://neptune.ai/blog/arima-vs-prophet-vs-lstm>
7. <https://www.youtube.com/watch?v=OaTAe4W9IfA&ab_channel=Stan>
8. <https://statmodeling.stat.columbia.edu/2017/03/01/facebooks-prophet-uses-stan/>
9. <https://corporatefinanceinstitute.com/resources/knowledge/other/bayes-theorem/>
10. <https://dziganto.github.io/python/time%20series/Introduction-to-Time-Series/#:~:text=Trend%2C%20as%20its%20name%20suggests,Residuals%20are%20random%20fluctuations>.
11. <https://machinelearningmastery.com/decompose-time-series-data-trend-seasonality/>
12. <https://en.wikipedia.org/wiki/Moving_average>