Predicting future sales

Clinton Aboagye
Jacob Appia

Predictive Analytics Project
April 2021

Department of Mathematics

Title:	Predicting future sales
Type:	Predictive Analytics Competition
Date:	April 2021
Authors:	Clinton Aboagye Jacob Appia
University:	Illinois State University
Department:	Department of Mathematics

Summary

The goal of this project is to predict the number of unique objects sold in a shop over a period of 1 month using about 3 years' worth of data. Given that the list of shops and products slightly changes every month, we create a robust model capable of handling these situations.

We will use the root mean squared error to test our models accuracy to pick the best model to use.

Preface

This project was submitted in fulfilment of the Illinois State University Annual Statistical Competition on April 1, 2021

Clinton Aboagye
Jacob Appia

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In our modern society, many retail business success is determined by the ability to meet the demand of their consumers. Major retailers are faced with the obstacle of how to meet these demands more effectively and in a timely manner.[Kha] A retailer may operate a lot of shops in different locations, and each shop replenish their stock level based on employees decisions which are influenced by events such as seasons and market trends. Occasionally, this retail shops overstock products than demanded which results in a financial burden to the retailer since the business revenue gets tied up in unsold stock. Similarly, not stocking enough products to meet the demand of the consumers put the retailer at risk of losing potential sales and consumers to competitors.

However, the ability to forecast sales accurately ensures a retailer has enough supplies to meet the consumers demand at all time. Forecasting is a technique that uses historical data and events to build estimates about future trends, potential disasters, and the overall behavior of any subject.

[BK] For the purposes of sales forecasting, statistical methods such ARIMA, regression models, Box-Jenkins model, Holt-Winters model or exponential smoothing are mostly used. These models are usually linear in nature but most real world sales data are not linear hence more advanced techniques are needed to improve the forecasting performance.

In the next section we describe the data and the various statistical techniques to be used for forecasting, and the approach to achieve this. Some of the techniques include dynamic regression models, multiple linear regression, bagging, boosting and artificial neural networks. In section 3, we prepare the data for analysis. This process involves cleaning the data, handling outliers, exploring the data

for possible pattern present and restructuring the data in a suitable format. We finally perform the main data modelling in section 4.

Chapter 2

Data and Methods

In the previous Chapter, we introduced the topic of predicting sales and the various considerations to be made in doing so. In this Chapter, we talk about our specific data to be used as well as the models we will use in our predictions.

2.1 Data

The data used in performing this analysis was retrieved from kaggle.com. It was originally provided by one of the largest Russian software firms- 1C Company.
The following data was provided:

- Sales_train.csv- This is the training data set. It contains daily historical data from January 2013 to October 2015. It contains 2,935,849 observations. This a glimpse of the data.

	date	date_block_num	shop_id	item_id	item_price	item_cnt_day
1	02.01.2013	0	59	22154	999.00	1
2	03.01.2013	0	25	2552	899.00	1
3	05.01.2013	0	25	2552	899.00	-1
4	06.01.2013	0	25	2554	1709.05	1
5	15.01.2013	0	25	2555	1099.00	1
6	10.01.2013	0	25	2564	349.00	1

Figure 2.1: Glimpse of Train.csv

- Test.csv- This is the set on which we make our final predictions. It contains shop_id and item_id for the month of November

2015. We will make predictions on the number of the products that will be sold. The specific feature we will predict is the item_cnt_day which we will discuss more about shortly. We load it into R as Prediction.set. This is a glimpse of the data.

	ID	shop_id	item_id
1	0	5	5037
2	1	5	5320
3	2	5	5233
4	3	5	5232
5	4	5	5268
6	5	5	5039

Figure 2.2: Glimpse of Test.csv

- Items.csv- This gives a description of the items sold in the shops in terms of their categories and names. It has 3 columns- item_id, category_id and item_name.

	item_id	category_id	item_name
1	0	40	!! IN THE POWER OF HAPPINESS (PLAST) D
2	1	76	! ABBYY FineReader 12 Professional Edition Full [PC, Digital ...
3	2	40	*** IN THE GLORY OF THE GLORY (UNV) D
4	3	40	*** BLUE WAVE (Univ) D
5	4	40	*** BOX (GLASS) D
6	5	40	*** NEW AMERICAN GRAPHICS (UNI) D

Figure 2.3: Glimpse of items.csv

- Item_categories.csv- This data set gives supplemental information about the item categories. From the data, there are 84 such categories.

	category_name	category_id
1	PC - Headsets / Headphones	0
2	Accessories - PS2	1
3	Accessories - PS3	2
4	Accessories - PS4	3
5	Accessories - PSP	4
6	Accessories - PSVita	5

Figure 2.4: Glimpse of item categories

- Shops.csv- This data set shows the shop names and shop id's used in the data analysis. A preliminary glance at the data reveals 60 such shops.

	shop_name	shop_id
1	! Yakutsk Ordzhonikidze, 56 francs	0
2	! Yakutsk TC "Central" fran	1
3	Adygea TC "Mega"	2
4	Balashikha TC "Oktyabr-Kinomir"	3
5	Volga TC "Volga Mall"	4
6	Vologda SEC "Marmelad"	5

Figure 2.5: Glimpse of the Predictive.set

Below is the description of the features in the data set:

- shop_id- Unique identifier of a shop. There are 60 unique shops.
- item_id- Unique identifier of a product. There are 22170 unique

items. There are 373 new items in the Test.csv data which are not in Train.csv.

- item_category_id- Unique identifier of item category. There are 84 unique categories.
- item_cnt_day - Number of products sold. We are predicting a monthly amount of this measure for the month of November.
- item_price - Current price of an item.
- date - Date in format dd/mm/yyyy
- date_block_num - A consecutive month number, used for convenience. January 2013 is 0, February 2013 is 1,..., October 2015 is 33
- item_name - Name of item
- shop_name - Name of shop
- item_category_name - Name of item category

2.2 Methods

The main statistical methods to be used here include multiple linear regression and XGBoost model.

2.2.1 Multiple Linear Regression Model

[Lac] Multiple linear regression attempts to model the relationship between two or more explanatory variables and a response variable by fitting a linear equation to observed data. Every value of the independent variable x is associated with a value of the dependent variable y. The population regression line for p explanatory variables $x_1, x_2, ..., x_p$ is defined to be

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p + \epsilon$$

ϵ is assumed to follow a Normal distribution with mean, $\mu, 0$ and variance, σ^2. The equation above describes how the mean response, y, changes with the explanatory variables. The observed values for y vary about their means y and are assumed to have the same standard deviation . The fitted values $\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_p$ estimate the parameters 0, 1, ..., p of the population regression line.
This type of regression is not best suited for time series analysis because it assumes that the predictor variables are inherently uncorrelated, especially with the response variable. Given that we will be using lagged values of our response variable as predictors, we only include it to see examine its performance.

2.2.2 XGBoost

Here I explain XGBoost, one of the models I used in my prediction. XGBoost is a relatively new algorithm used in machine learning. It stands for Extreme Gradient Boosting. It builds on the gradient boosting algorithm and can thus be used in both regression and classification tools.
[VM] We can think of XGBoost as gradient boosting on 'steroids'. It is a perfect combination of software and hardware optimization techniques to yield superior results using less computing resources in the shortest amount of time. XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework.
The XGBoost algorithm was developed as a research project at the University of Washington by Tianqi Chen and Carlos Guestrin. XGBoost and Gradient Boosting Machines (GBMs) are both ensemble tree methods that apply the principle of boosting weak learners using the gradient descent architecture. However, XGBoost improves upon the base GBM framework through systems optimization and algorithmic enhancements. Like many ensemble methods, it can handle correlation among variables very well.
One of its disadvantages is that it can only handle numerical data. Thus we will include the shop ID and item ID features as numerical features. All our categorical features will need to formatted as nu-

merical by first converting them to factors and converting them as numerical values.
In this project, we use XGBoost because of 2 main reasons:

- Execution Speed.
- Model Performance.

Because of XGBoost's unique build, it can run complex data sets with relative ease and high level of accuracy. Given that we have more than 2 million observations in our Training data alone, the XGBoost model comes really handy in here.

2.3 Step-by-step approach

- Import data into R
- Clean data by removing all NA's and outliers.
- Pre-process shops data and categories data in Excel.
  We pre-process the shops data to get shop_city, shop_name and shop_type. We pre-process the categories data to get the item_category and item_subcategories.
- Add features:
  We add features like revenue, price and average total sales per month in every specific shop/specific shop type.
- Add lag features:
  We decide to use lagged sales figures from 3 prior months to help predict sales in the current month.
- Preparing our training and testing set
  We use date block 0-32 (Jan 2013- October 2015) as training data set and date block 33 as the testing data set. we will compute the root mean squared error based on this testing data set.

- Fitting the model
  We go on to fit the model and compute the root mean squared error.
- Display the important features in the model.
- Drop unimportant features and refit the model computing the root mean squared error.
- Making sale predictions for date block 34(November 2015) using full data set.

Chapter 3

Preliminary Exploration before Modelling

In the previous Chapter, we talked about our data, methods and assumptions to be made in doing this analysis. In this Chapter, we'll talk more about the preliminary analysis done before the modelling process.

3.1 Data Exploration with graphs

Figure 3.1 below shows the total sales per month formatted in terms of shops from the least per month to the most per month.
We see that shop 31 has the highest sales of all 60 shops while shop 36 has the lowest sales.

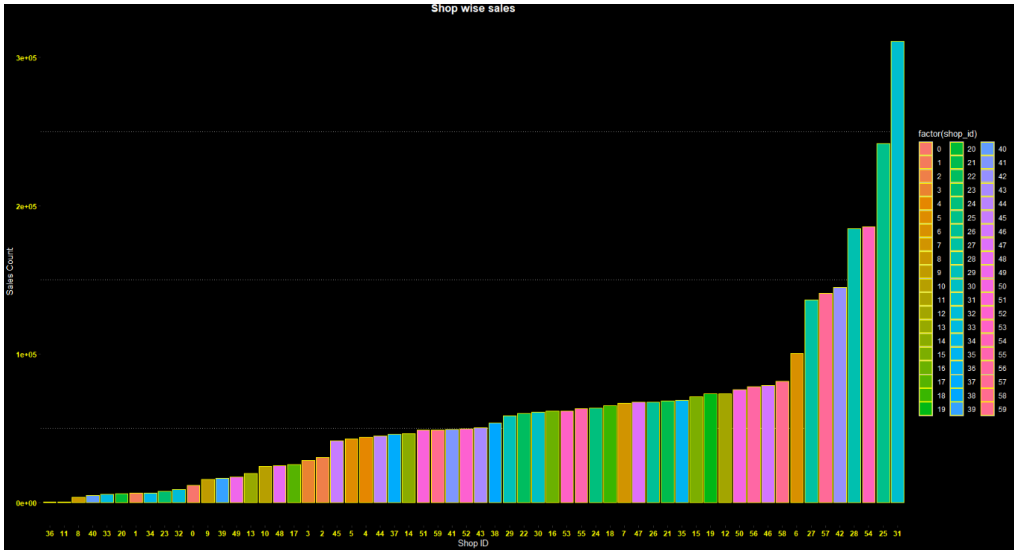


Figure 3.1: Total sales per shop

In terms of item categories, Figure 3.2 below shows the item category

which had the most sales. Item Category 10 has the least sales while item Category 40 had the most sales.

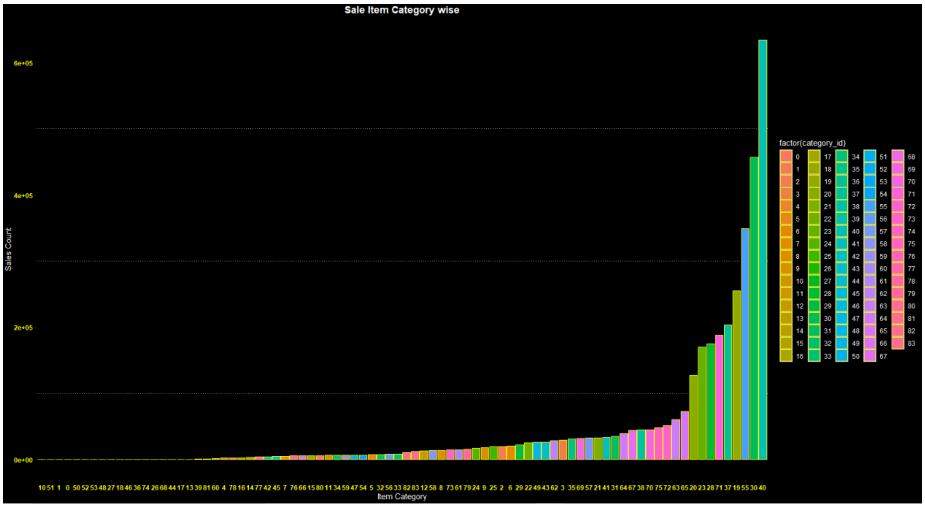


Figure 3.2: Total sales per category

Figure 3.3 shows the shop with the most items in decreasing order of magnitude. From the plot, shop 25 has the most items in it whilst shop 36 has the least items.

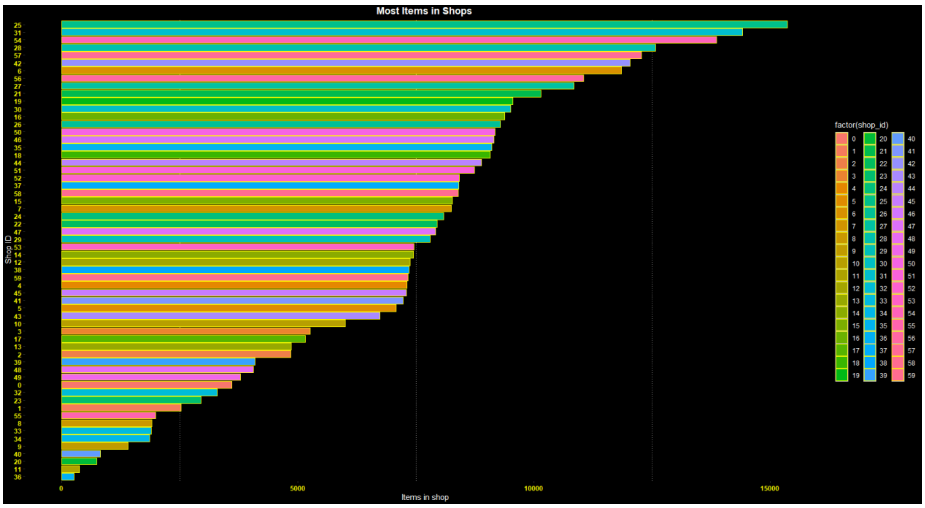


Figure 3.3: Shop with most items

Figure 3.4 shows which item category has the most items. We observe that item category 40 has the most number of items whilst category 10 has the least number.

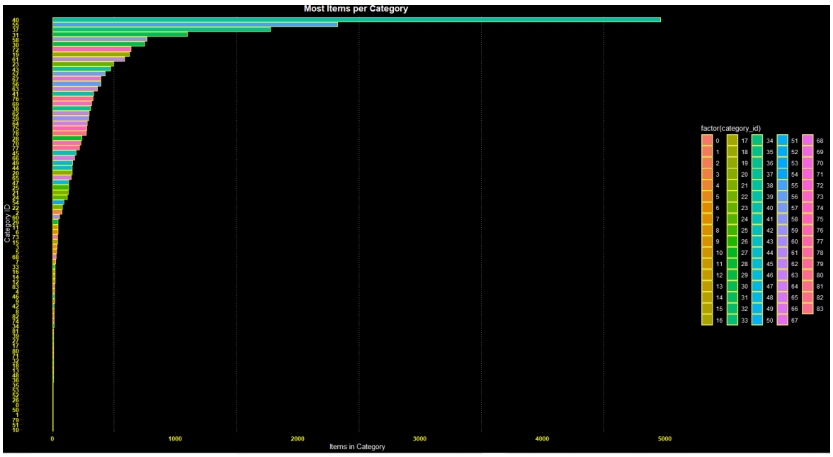


Figure 3.4: Item category with most items

From Figure 3.5, we obtain that item ID 20949 is the most popular in most shops with the highest sales being from shop 31. We note that the most sold item in shop 36 is also item 20949 with 16 sales.

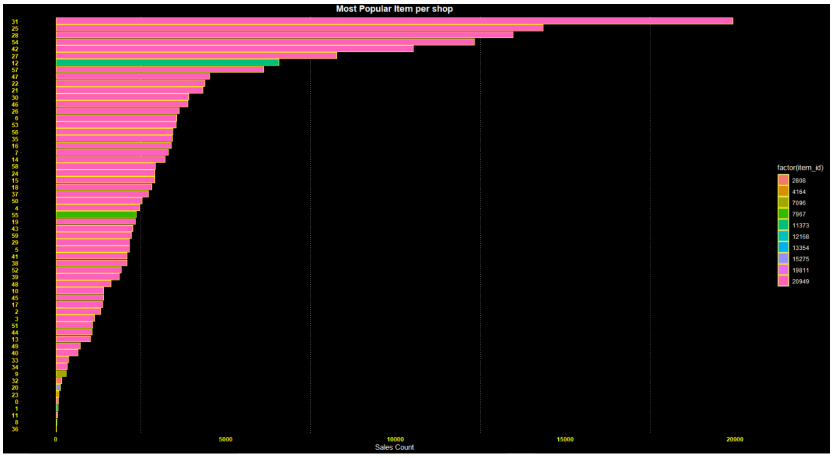


Figure 3.5: Most popular(sold) item per shop

In Figure 3.6, we obtain that shop 25 has the most assortment of item categories while shop 20 has the least assortment. This may be because shop 25 is the largest shop.

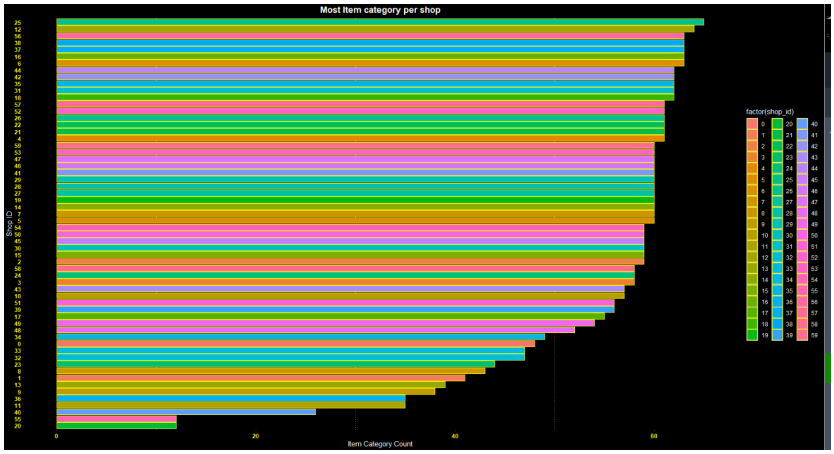


Figure 3.6: Shop with most assortment of item categories

Figure 3.7 shows the most popular item category per shop. In shop 31, the most popular item category patronised is from category 40 with 76069 sales. In shop 36, the most popular item category is from category 55 with 47 total sales.

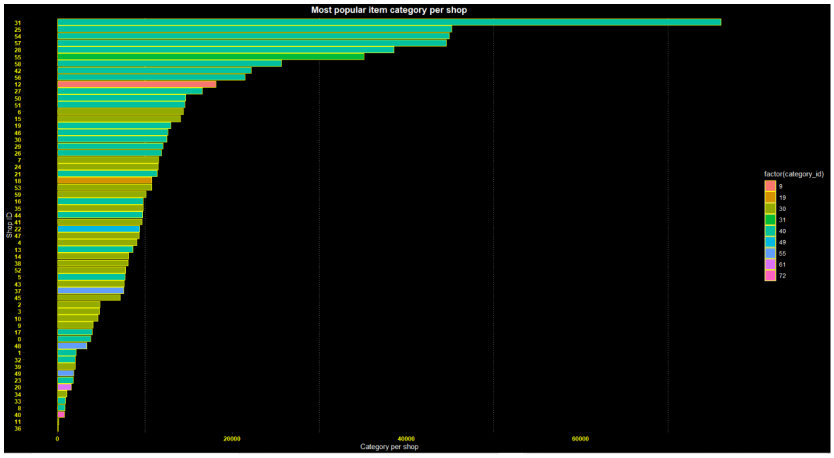


Figure 3.7: Shop with most assortment of item categories

What we have below in Figure 3.8 is the highest grossing item category. It shows the item category which generates the most revenue. Item category 19 generated the most revenue with more than 400 million. Item category 50 generated the least revenue.

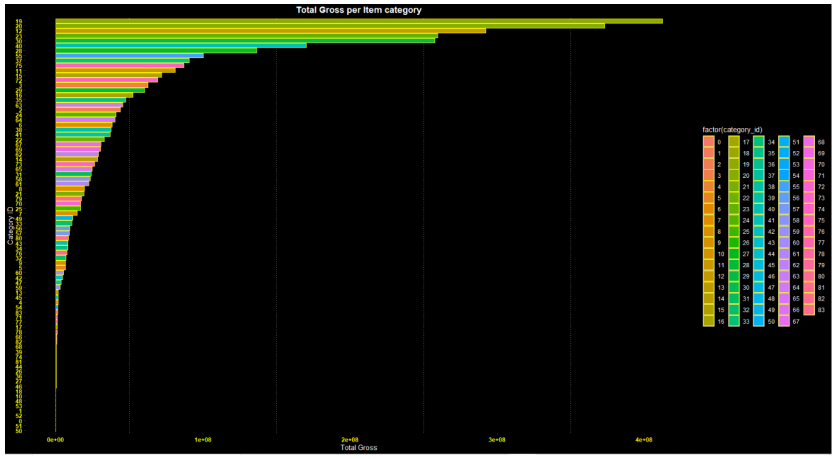


Figure 3.8: Most grossing item category

In Figure 3.9, we have the item which had the most revenue in any item category. This turns out to be item 6675 from category 12.

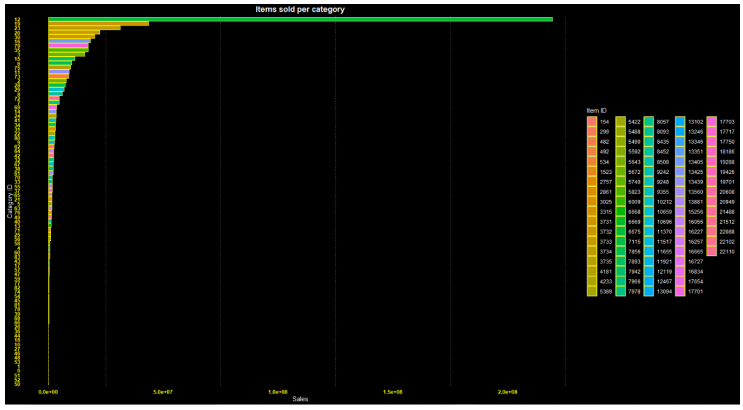


Figure 3.9: Most grossing item

The figure below is an especially interesting one, It shows how rev-

enue changes per day in all 12 months. The months have been formatted as line graphs. Thus one can notice that the most revenue was made on the 29th Day in November. A cursory search online showed that 29th November of 2013 was Black Friday in Russia. This explains the spike quite well.

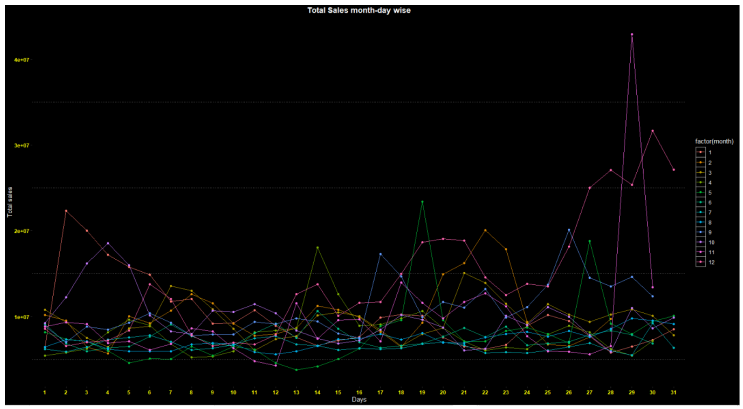


Figure 3.10: Total revenue generated every day per month

We end our exploration here with a graph of the daily items sold from January 2013 till October 2015. [1]

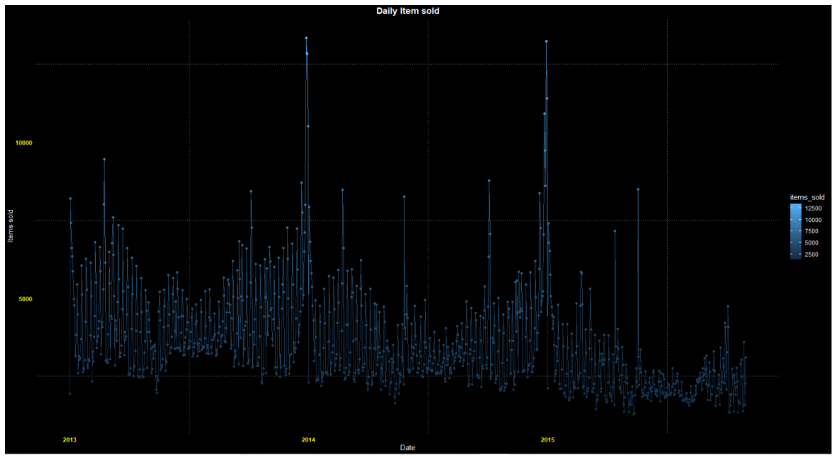


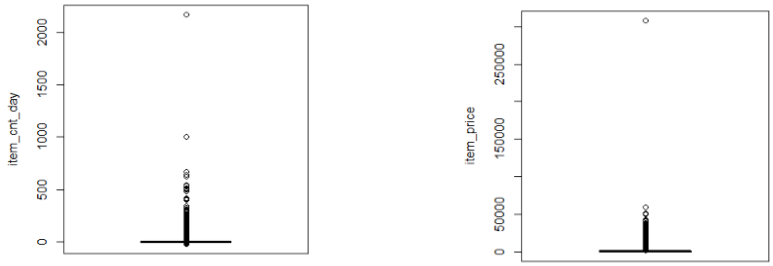
Figure 3.11: Daily Total sales from Jan 2013-Oct 2015

[1] We got help with the codes for running these awesome graphs from https://www.kaggle.com/jeetranjeet619/predict-future-sales-r

3.2 Cleaning the data for use

3.2.1 Removing obvious outliers and negative values from Train.csv

The box plots below show the distribution of the item prices and the quantity bought daily over the period from January 2013 to October 2015. We can see that there seem to be outliers in both plot. So what we will do first is to remove the item that sold more than 1000 in one day and the item with price greater than 300,000. These are obvious outliers which can be due to opening events and the like.



(a) Item count

(b) Item price

Figure 3.12: Box plot of item price and count

Next we remove negative price values as they may imply refunds. We also set any negative item counts to zero.

3.2.2 Cleaning the shops.csv data

Some shops seem to be duplicates of each other. For example, shop ID 0 and shop ID 57 seem to be the same. They have names "!Yakutsk Ordzhonikidze, 56 francs" and "Yakutsk Ordzhonikidze, 56". Several reasons could account for this. For example, re-opening or moving the store location on the same street or shopping center. As a result, we will merge such shops.
We notice that the shops.csv data has the column shop_id formatted in the following form: Shop city | Shop type | Shop name.
We used Excel to format this data into the various sections. Not all of them had this formatting though, and this led to some NA values being generated. We went ahead to create shop categories for shops types which are more than 5 in number. For example, we notice that shop type TC appeared at least 5 times and thus it will be made a shop category and added in as a feature to help with the prediction.
For all the shop types which appear less than 5 times(or those with no types), we group them into a category called Other. Figure 3.2 is a snippet of what our data looks like after transforming it. We had 4 total subcategories: TC, TEC, Shoppingcenter and Other.

	shop_name	shop_id
1	! Yakutsk Ordzhonikidze, 56 francs	0
2	! Yakutsk TC "Central" fran	1
3	Adygea TC "Mega"	2
4	Balashikha TC "Oktyabr-Kinomir"	3
5	Volga TC "Volga Mall"	4
6	Vologda SEC "Marmelad"	5
7	Voronezh (Plekhanovskaya, 13)	6
8	Voronezh SEC "Maksimir"	7
9	Voronezh SEC City-Park "Grad"	8

(a) Before transforming shops data

	shop_city	shop_type	shop_name	shop_id
1	Yakutsk	Other	Ordzhonikidze56francs	0
2	Yakutsk	TC	Centralfran	1
3	Adygea	TC	Mega	2
4	Balashikha	TC	Oktyabr-Kinomir	3
5	Volga	TC	Volga Mall	4
6	Vologda	SEC	Marmelad	5
7	Voronezh	Other	Plekhanovskaya,13	6
8	Voronezh	SEC	Maksimir	7
9	Voronezh	SEC	City-Park"Grad"	8

(b) After transforming shops data

Figure 3.13: Shops data transformation

3.2.3 Cleaning the Category data

We notice also that the category_name in Categories.csv has been formatted to have type and subtype in its name. We will reformat this data to obtain the types and subtypes. Figure 3.3 shows our results.

	category_name	category_id
1	PC - Headsets / Headphones	0
2	Accessories - PS2	1
3	Accessories - PS3	2
4	Accessories - PS4	3
5	Accessories - PSP	4
6	Accessories - PSVita	5
7	Accessories - XBOX 360	6
8	Accessories - XBOX ONE	7

(a) Before formatting categories

	Type	Subtype	category_id
1	PC	Headsets / Headphones	0
2	Accessories	PS2	1
3	Accessories	PS3	2
4	Accessories	PS4	3
5	Accessories	PSP	4
6	Accessories	PSVita	5
7	Accessories	XBOX 360	6
8	Accessories	XBOX ONE	7

(b) After formatting categories data

Figure 3.14: Formatting the Categories data

3.2.4 Investigating the Items data

This data shows the item_id, category_id and item_name. We make no changes to this data. We will, however, use its information to make some changes to our training data.

3.2.5 Investigating the test data

The test data set has 42 unique shops and 5100 items. This makes a total of 214200 observations. Our aim in this project is to predict the quantities of these items bought in each of the 42 shops. Note that whereas there were 21807 unique items in the training data set, there were 5100 items in the testing data set. We check to find whether all the items in the testing data set are in the training data set. Our analysis shows that there were 363 new items in the test data set that were not in the earlier data set. For these 363 new

items we will have to investigate how such situations were handled by past data. We will then make predictions using that background.

3.3 In-depth investigation of Train.csv and Test.csv

3.3.1 Train.csv

As discussed earlier, our data shows daily sales amounts. We are going to convert these daily amounts to monthly amounts because we need to make our predictions for sales in monthly measures. We use the tidyverse command in R to do this. For our convenience, the data has been time formatted such that 0 represents January 2013, 1 represents February 2013 and so on and so forth till 33 which is October 2015.
We add revenue (as Price*Quantity) to the Train.csv data. We notice that the items_id in Items.csv corresponds to the items_id in the Train data so using the items_id as our point of reference, we add the category_id to the Train.csv data. Then using the category_id as our point of intersection, we add Type and Subtype from our category_data to the Train.csv.
Figure 3.4 shows both our before and after Train.csv transformation. One notices that before transformation, we had only 4 useful features: date_block_num, shop_id, item_id and item_price. After transformation and formatting, we have these features added;

- Revenue - This is the product of item_price and item_cnt_month.
- category_id- This was originally in the items.csv file but we decided to add it to our Train.csv and use it in training our model because we believe it will add valuable insights to our data.
- Type - This factor variable has 9 different levels. It simply shows the type of each of items. These may be gifts, games, music etc.

- Subtype- This factor variable has 61 unique levels. It shows specific type of item. For example, a gift item may be a bag, a card/sticker or a game item may be standard edition or limited edition etc.
- shop_city- This simply shows the city in which our shop is situated. This may be Kazan, Moscow etc. Since our data was from a Russian company, our cities are Russian cities.
- shop_type - This also simply shows the type of shop from which the item was purchased. This may be TC, shopping center, SEC or Other. There are only 4 such designations.

	date_block_num	item_price	shop_id	item_id	item_cnt_month	revenue
1	0	9.0	13	13598	18	162.0
2	0	10.0	27	13598	4	40.0
3	0	10.0	31	13598	72	720.0
4	0	10.0	42	13598	55	550.0
5	0	13.0	1	13344	1	13.0
6	0	13.0	1	13345	16	208.0
7	0	13.0	10	13344	15	195.0
8	0	13.0	10	13345	1	13.0
9	0	13.0	13	13345	42	546.0
10	0	13.0	51	13344	35	455.0
11	0	13.0	51	13345	74	962.0
12	0	14.0	0	13354	38	532.0
13	0	14.0	1	13354	8	112.0
14	0	14.0	10	13354	9	126.0

(a) Train.csv

	date_block_num	item_price	shop_id	item_id	item_cnt_month	revenue	category_id	Subtype	Type	shop_city	shop_type
1	0	9.0	13	13598	18	162.0	71	Bags, Albums, Mousepads	Gifts	Kazan	TC
2	0	10.0	27	13598	4	40.0	71	Bags, Albums, Mousepads	Gifts	Moscow	TC
3	0	10.0	31	13598	72	720.0	71	Bags, Albums, Mousepads	Gifts	Moscow	TC
4	0	10.0	42	13598	55	550.0	71	Bags, Albums, Mousepads	Gifts	St.Petersburg	shoppingcenter
5	0	13.0	11	13344	15	195.0	82	Others	Other	Zhukovskyst.	Other
6	0	13.0	11	13345	1	13.0	82	Others	Other	Zhukovskyst.	Other
7	0	13.0	13	13345	42	546.0	82	Others	Other	Kazan	TC
8	0	13.0	51	13344	35	455.0	82	Others	Other	Tyumen	TC
9	0	13.0	51	13345	74	962.0	82	Others	Other	Tyumen	TC
10	0	13.0	58	13344	1	13.0	82	Others	Other	Yakutsk	TC
11	0	13.0	58	13345	16	208.0	82	Others	Other	Yakutsk	TC
12	0	14.0	11	13354	9	126.0	82	Others	Other	Zhukovskyst.	Other
13	0	14.0	50	22010	1	14.0	30	Standard Editions	Other	Tyumen	SEC
14	0	14.0	51	13354	116	1624.0	82	Others	Other	Tyumen	TC
15	0	14.0	57	13354	38	532.0	82	Others	Other	Yakutsk	Other
16	0	14.0	58	13354	8	112.0	82	Others	Other	Yakutsk	TC

(b) After formatting Train.csv

Figure 3.15: Formatting the Train.csv file

3.3.2 Test.csv

We now format our Test.csv to have a form comparable to our Train.csv. The process is similar to what we did in our Train.csv. After formatting, there are 3 variables in our Train.csv which are not in our Test.csv. These are item_cnt_month, revenue and item_price. Item_cnt_month is what we are trying to predict. Item_price was not given in our test.csv.

Chapter 4

Fitting our models

4.1 Multiple Linear Regression Model

We start our model analysis with the multiple linear regression model. After fitting this model, we obtain a root mean squared error of 292 on our testing data set. All the predictors used here were statistically significant.

4.2 XGBoost Model

We fit our XGBoost model and plot the relative importance plot. This is Figure 4.1 as shown below. On the test data set, we obtained a root mean squared error of approximately 242.

We observe that item_price is the most useful predictor for item sales. This is closely followed by revenue. Shop_type is the least useful predictor according to this model.

We now tweak our XGBoost model to improve its performance. In our earlier fit, we formatted all our variables as numeric. What we will do now is maintain some of these variables as factors by using one-hot encoding. What this does is convert a factor variable with n-levels to n separate variables with 1 indicating True and 0 indicating False. For example, the variable shop_type with 4 levels will be discarded leaving its each of its four levels as variable on their own. If a variable belongs to shop_type TC, it records 1 there and 0 at all other places. We do this hot encoding for only the shop_type and category type variables since they have only 4 and 9 factors respectively. The other factor variables will be kept as numeric.

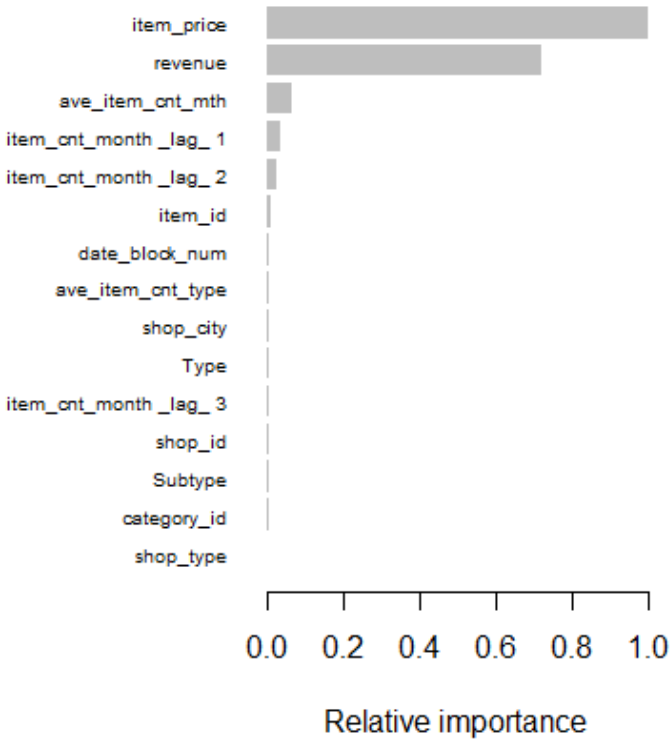


Figure 4.1: Relative Importance of Variables

We weren't able to substantially reduce the root mean squared error. It only reduced by only approximately 1 point. The significant variables in predicting sales turn out to be the same as earlier.

4.3 Conclusion

From observing our different models, it is clear that the XGBoost model has the least root mean square error. The result is summarized in the table below.
We attach an Excel file which shows our final predictions for the prediction set. This is our final deliverable.

We see that price and revenue are the most important predictors of how much is purchased.
Final considerations to improve our XGBoost model is hyperparameter tuning. This can substantially make a model better.

References

[BK] Karimi H.R. Thoben K. Lütjen M. Teucke M. Beheshti-Kashi, S. A survey on retail sales forecasting and prediction in fashion markets.

[Kha] A. Khaled. Optimizing retailer revenue with sales forecasting ai.

[Lac] Michelle Lacey. Multiple linear regression.

[VM] Venkat A. Setty Vishal Morde. Xgboost algorithm: Long may she reign!