Final Project Report

Team member: Dan He, Minqi Liao, Ming Chen

Introduction

In this project, we are trying to build a predictive model for one of the largest Russian software firms to predict the total amount of sales for every product and store in the next month. This model can be used effectively to predict sales amount by month for stores. By accurately predicting sales amount, stores can ensure that their products are in sufficient stock, which can bring a better experience for their customers and lead to a higher sales revenue in the long run.

The dataset contains 11 variables in total (including the decision variable). We will first clean our dataset by excluding the outliers and non-values, then, we will interpret our data through visualization. After that, we will separate our full dataset into training data and test data, and then run multiple modelling analysis with our training data and make predictions in our test data. We will compare our results using RMSE with models we created. Finally, we will conclude our results and make suggestion at the end of the report.

Data preparation

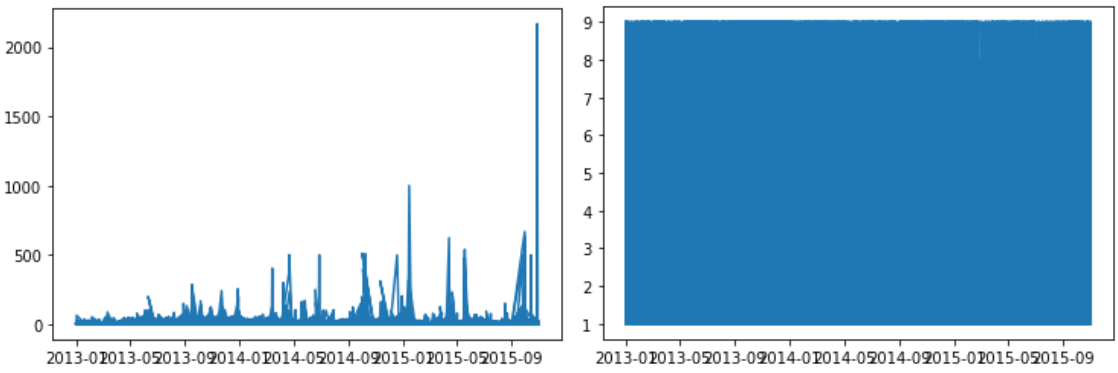
We will use 5 files in our data anlysis, including *sales_train.csv, test.csv, items.csv, item_categories.csv,* and *shops.csv*. The *sales_train.csv* dataset is the main file we will use to clean our dataset.

After loading our data to Python, we will use the "describe" function to outline a breif data summary of our dataset.

	date_block_num	shop_id	item_id	item_price	item_cnt_day
count	2935849.000000	2935849.000000	2935849.000000	2935849.000000	2935849.000000
mean	14.569911	33.001728	10197.227057	890.853233	1.242641
std	9.422988	16.226973	6324.297354	1729.799631	2.618834
min	0.000000	0.000000	0.000000	-1.000000	-22.000000
25%	7.000000	22.000000	4476.000000	249.000000	1.000000
50%	14.000000	31.000000	9343.000000	399.000000	1.000000
75%	23.000000	47.000000	15684.000000	999.000000	1.000000
max	33.000000	59.000000	22169.000000	307980.000000	2169.000000

As we can see from the table (in our code), some of the columns might seem unusual compare to our common sense. For example, both the item price and the item sales amount have some negative values. Since negative values can hurt multiplocative seasonality diagnostics, we should remove these negative values from our dataset. Besides, as we can see from the plots shown below, the first plot shows the amount of item sold per day, it's reasonable to see the amount of

item vary throughout the year. However,some of . At first, we were thinking about using the outlier calculation formula to drop data that's over 3 standard deviation from the mean. However, if we use this method, our plot will change from the first plot to the second plot shown below. In this case, our data cannot well represent the overall time trend of the amount of items. Therefore, we choose to drop the largest item count from our dataset (which is over 2,000) in order to not only keep the time trend of our data, but also increase the accuracy of our prediction in our model analysis.



Besides item count, we also find that there's an outlier in item price column. As we can see from the table, the largest item price is over 300k, while the average item price is around 890. Thus, we decide to drop this value from our dataset to decrease the influence of outliers.

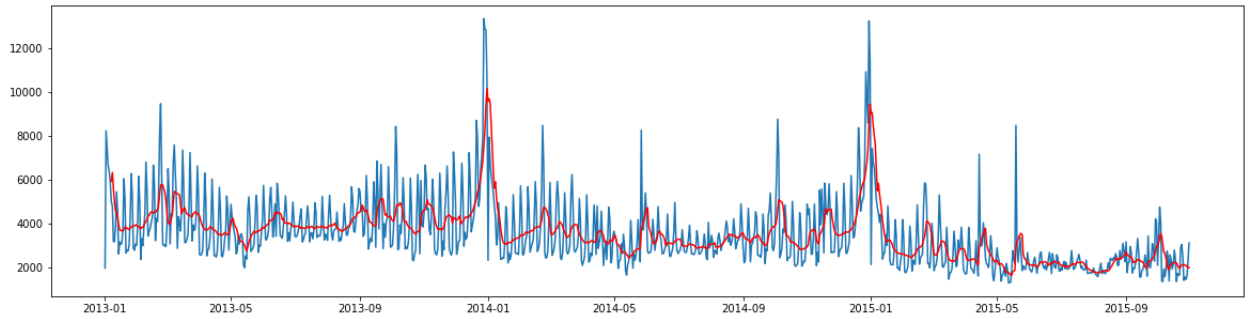
After cleaning the outliers, we checked if our dataset has any null values. As the table in our code shown, we do not have null values, so we don't have to replace any null values to zeros.

Since we want to keep our prediction values in the range of [0,20], we will use clip function for both our training data and test data. We will add clip function both before and after building the model to represents consistency. This can help us limit the effect of outliers and lower the RMSE.

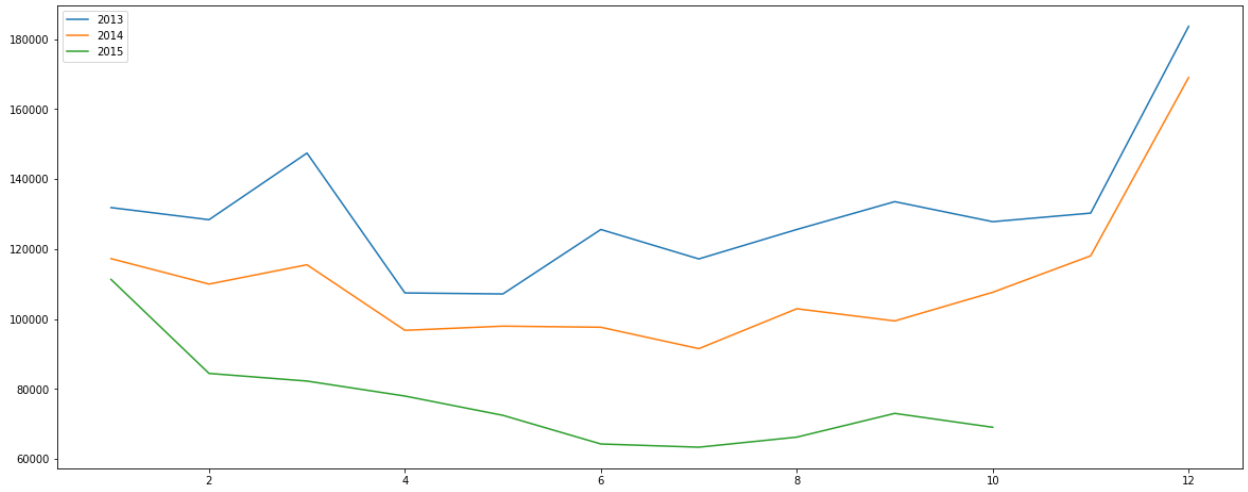
Data Interpretation

After combined our data files, we made several graphs to visualize the relationship between variables (for example, date and item count etc.). These graphs are listed below.

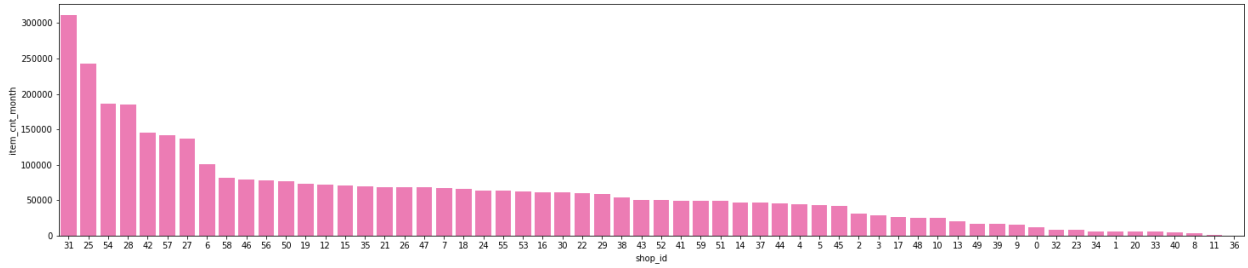
First, we created a line chart to show the time trend of the sum of item count per day (of all store). The blue line represents each day's item count, while the red line represents a smoothed line of overall trend. As we can see from the graph, there is an increase in sales in January (each year), meanwhile, a decrease of sales will occur around May in each year. These changes might relate to national holidays. For example, people are expected to purchase new items during the new year festival, which may lead to an increase in sales.



Besides, we also want to explore the difference of sales in different years. Therefore, we created a line chart to show the total amount of sales by month and separated each year in different colored lines. From this graph, we could interpret that from 2013 to 2015, the number of total sales is decreasing. Besides, we can find that there are some seasonal peaks around March, April and December, which is corresponding to the findings in the previous graph.

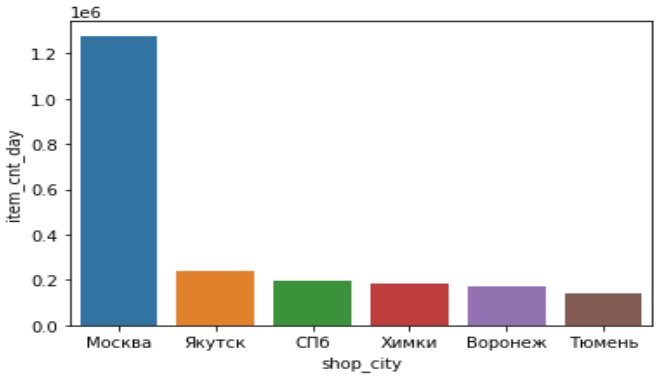


The third graph mainly aim to show the amount of item count by month for each store through a bar graph. The shop id is listed below. The shop 31 has the highest number of sales, which is over 300k; while the shop 36 has the lowest amount of sale, which is about zero. This might because some shops might close due to business problems.



The last bar graph shows the item count per day by city. When we look at the shop name, we notice the shop name is combined by 2 parts: city name and shop name. Since we have big cities and small cities, we would expect different number of sales in different cities, and this can be used as one of the features to predict our outcome. Thus, it would be benefit for our data model if

we separate city name from the shop name and use city id to represent different cities. Like the graph shown, the city Moscow has the largest amount of item sold comparing to the rest of cities. This makes sense because Moscow is the capital and largest city of Russia.



Modelling Analysis

1): Feature Selection

Before we start creating our models, we want to include enough features that could represent some kind of relationship with the outcomes. By looking at the dataset we have, most of the variables are IDs, which are just numbers that does not have specific meanings. We should treat these IDs as categories instead of values to make our prediction more valid. In this case, we decided to transform one of our variables "*item category id*" into dummy variables to represent different categories of each item. Besides, since we did create a new variable called "city code", we also want to transform this variable into dummy variables because different cities will influence the amount of item sold (our outcomes). We also included price mean and price median as our features since these numbers are meaningful and may affect outcomes. We want to use part of our training data as test data to validate our prediction. We choose to use last month's data (date block number = 33) because by using a part of continuous dataset, we can keep the time trend; moreover, using 1 month's data as test data can keep our training data remain large and representative. With these settings, we are ready to leverage machining learning tools to create our predictive models.

2): Modelling

Firstly, we decide to run a linear regression because we have enough number of dummy variables. The RMSE value for linear regression is about 2.318. After it, we think it's possible to improve our model by using lasso regression by applying different parameter values. After running our model, we found that the best model would have a parameter equals to 0.0001, which ends up with the lowest RMSE. We choose this parameter as our tuning parameter, which gives us a RMSE of 2.3168. Lastly, we decide to run a random forest analysis. We use the dataset without dummy variables in our random forest analysis. The result from random forest is about 6.68, which is higher than lasso regression results. In this case, we decide to choose lasso regression model as our predictive model for the final test dataset.

Prediction and Advantages

After created the lasso regression model, we applied it to our final test dataset. The RMSE of the test dataset is about 5.45150. The result is within our expectation because we would expect some level of increase in RMSE since some features in test data might not be included in our training dataset. There are some advantages of our modelling. Firstly, it's easy to understand and beginner friendly. Even using linear regression approach can reach to a low RMSE value by leveraging dummy variables. Secondly, our model included meaningful variables instead of just using id to predict. This would increase our prediction accuracy.

Challenges and Weaknesses

There are some challenges and weaknesses of our modelling process. For challenges, we found it's pretty difficult to directly train our model using just item id and shop id when we first start analyzing our dataset, since these numerical information does not have specific meanings with our outcomes. In this case, we choose to transform some of these variables into dummy variables to represent its category. This approach works well in linear and lasso regression, but not in random forest. Even though we reached to a RMSE of 2.3168, our final predict ends up with a much higher RMSE, which means our model still needs to be improved.

Recommendations and Future Directions

In order to improve our predictive model, we can leverage XGBoost and LGBMRegressor to apply level-wise tree growth and leaf-wise tree growth approaches to train our dataset.

Reference

https://github.com/KubaMichalczyk/kaggle-predict-future-sales.git