Capstone 2- Predicting Future Sales- Final Submission Report

Davy Gia Nguyen

**Background**

My main goal for this capstone project is to apply and enhance my data science skills from the Springboard data scientist program by predicting total sales for every product and store by analyzing several time-series datasets consisting of daily sales data from 1C Company, one of the largest Russian software firms.

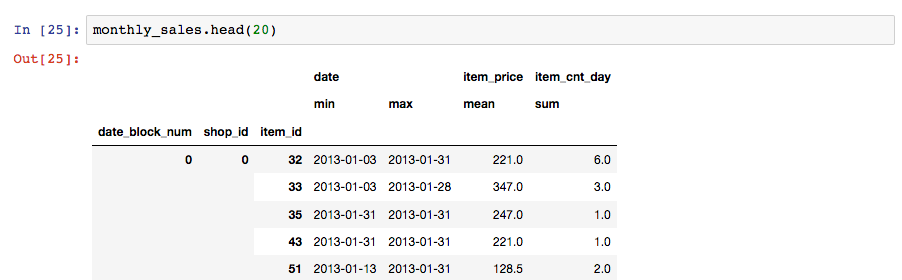
1C is an enterprise software company with integrated systems and programs intended for automation of daily enterprise activities. For instances, 1C provide solutions for various business tasks of economic and management activity including management and business accounting, human resources management, customer relationship management, supplier relationship management, material requirements planning, and other similar solutions.

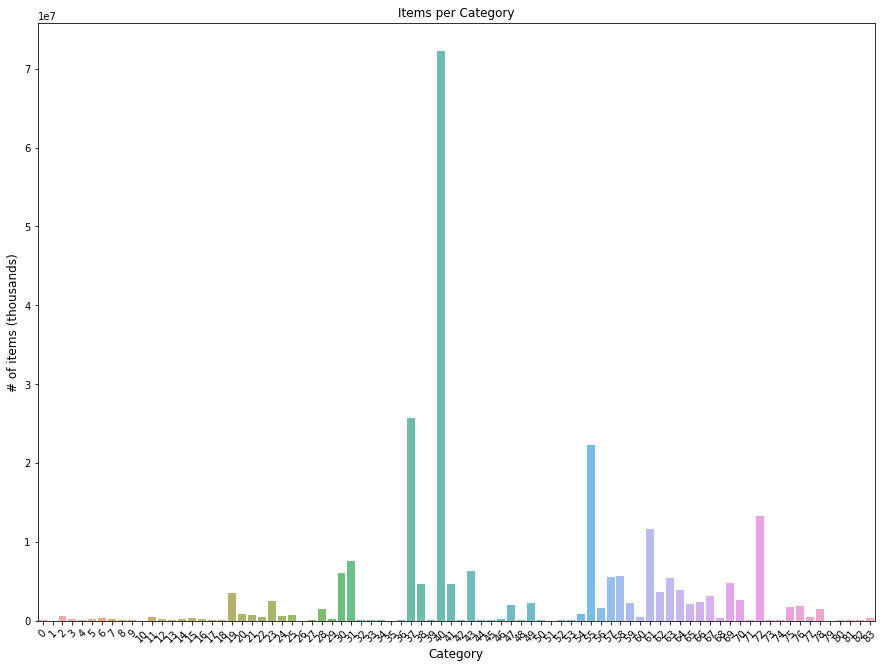
I hope to gain the business insights and data intelligence that a project like this would require so I can assist future clients with similar goals. Since sales prediction and forecasting are paramount to almost all kinds of businesses, I hope to bring my analysis and insights to a level that would help a similar client have full confidence in making similar crucial executive decisions regarding future sales predictions.

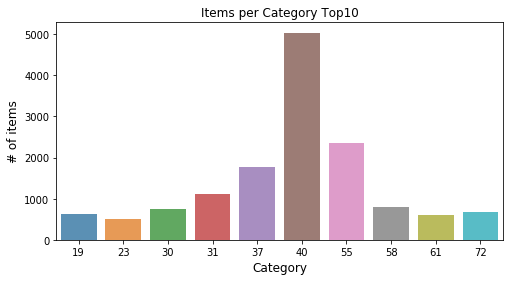
**Exploratory Data Analysis**

The time-series data from 1C Company is composed of 5 datasets with all the items data, item categories data, sales training data, test data, and shops data. Initial exploratory data analysis provide a few understandings on how to find our solution of predicting future sales of the next month for each store and item combinations. Some of these findings include that the data shows us there are 84 unique item categories, 60 stores, and 22,170 items. Our test set has 214,200 entries and our sales training data has 2,935,849 entries.

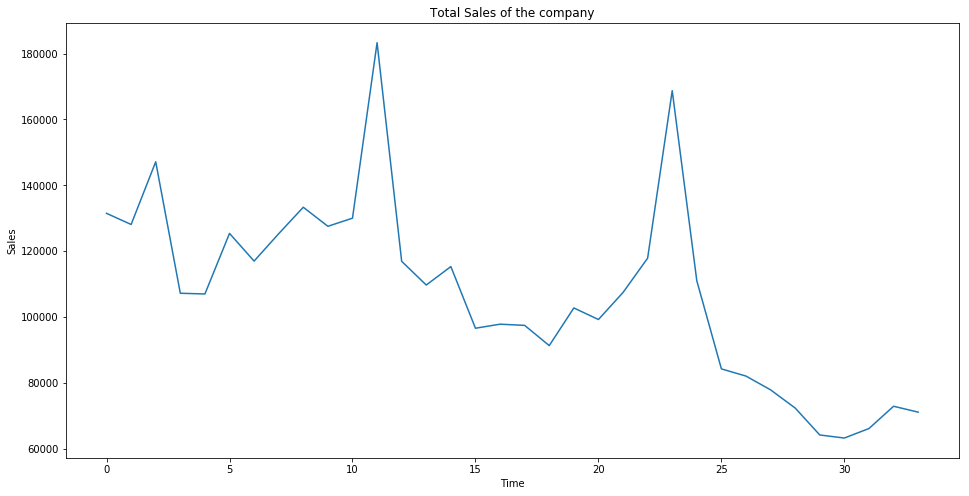
We then wrangled our data into a new dataframe, monthly\_sales, by rearranging our sales training dataset to get a better idea of the provided daily historical sales data:

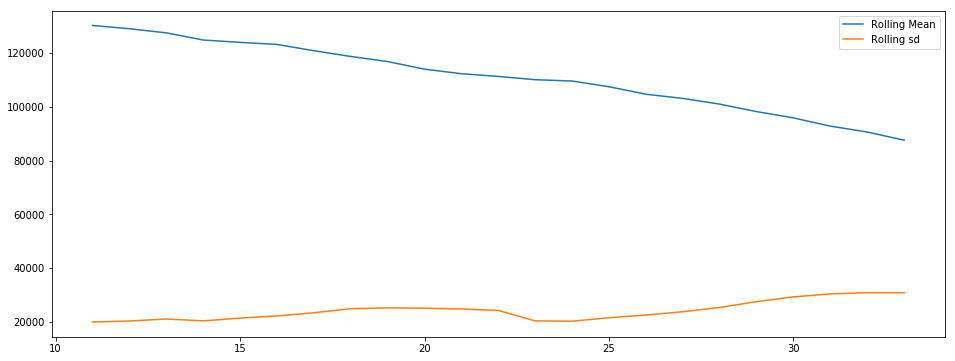


This step narrows down our data to 1,609,124 entries. We then plot the number of items for each item category. 



Since our goal is to predict futures sales for the next month using a store-item combination, sales over time of each combination is in a time-series, so we can gain some insights by first computing and plotting the total sales per month for the entire company as a function of time before diving into each shop and item.



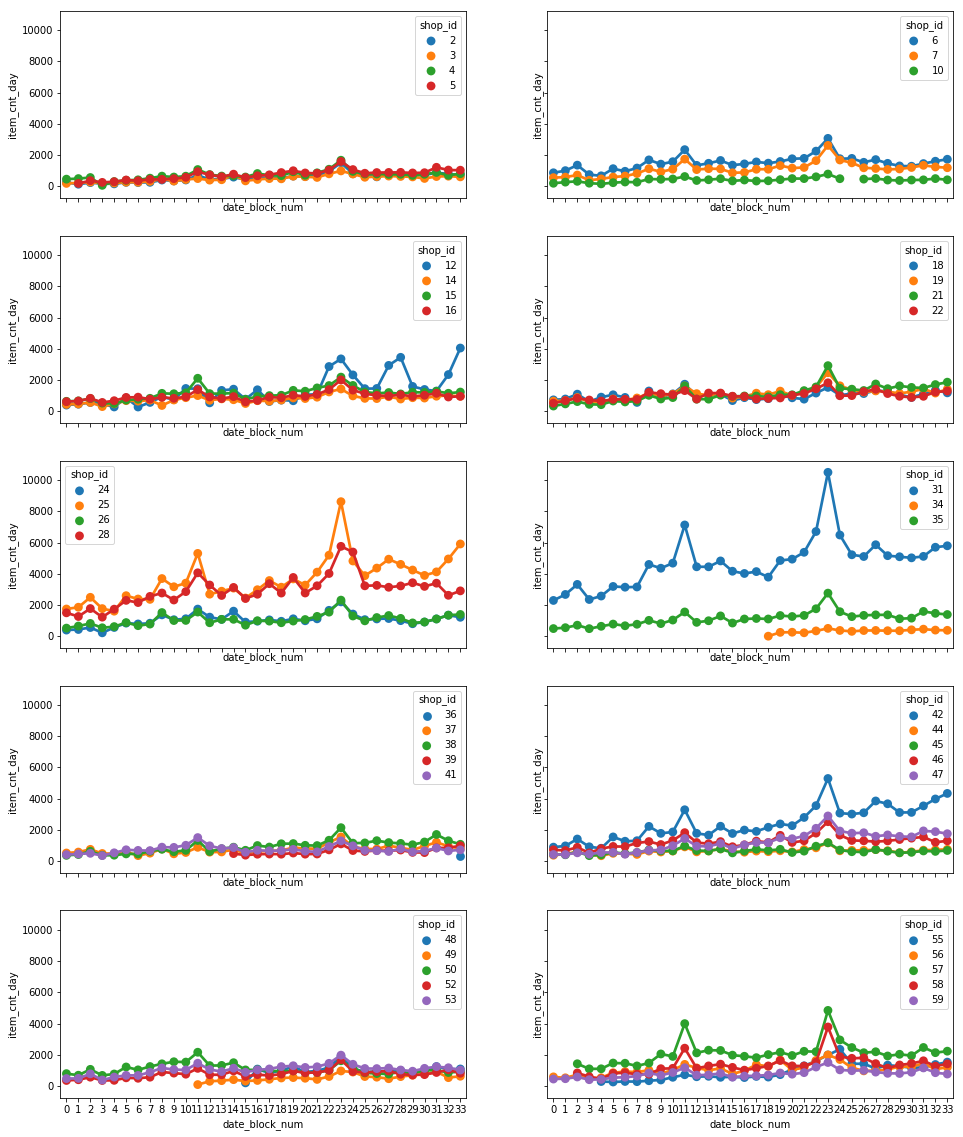


We observe that within the 34 months of our collected sales data, total sales have an overall decreasing trend and that the peaks and valleys indicate a seasonal sales cycle with higher sales during the holiday season every November and December, which indicates that our future sales prediction should increase as it approaches month 35 and 36. Therefore, we can reasonably hypothesize that our prediction for the total sales for every product and store in the next month should follow this pattern and be higher than its preceding months but its peak should be lower than the peaks in month 11 (highest in 2013) and month 22 (second highest).

We also visualize sales for all 34 months for each shop and item categories. We find that the trends are similar to our total sales trends but most recorded sales data are actually constant overtime with a few exceptional shops and item categories. Furthermore, we ensured our heavily-skewed data has stationarity with a constant mean, standard deviation, and covariance by performing the Augmented Dicky Fuller Test (ADF) since most statistical modeling and machine learning methods assume or require time series data to be stationary.

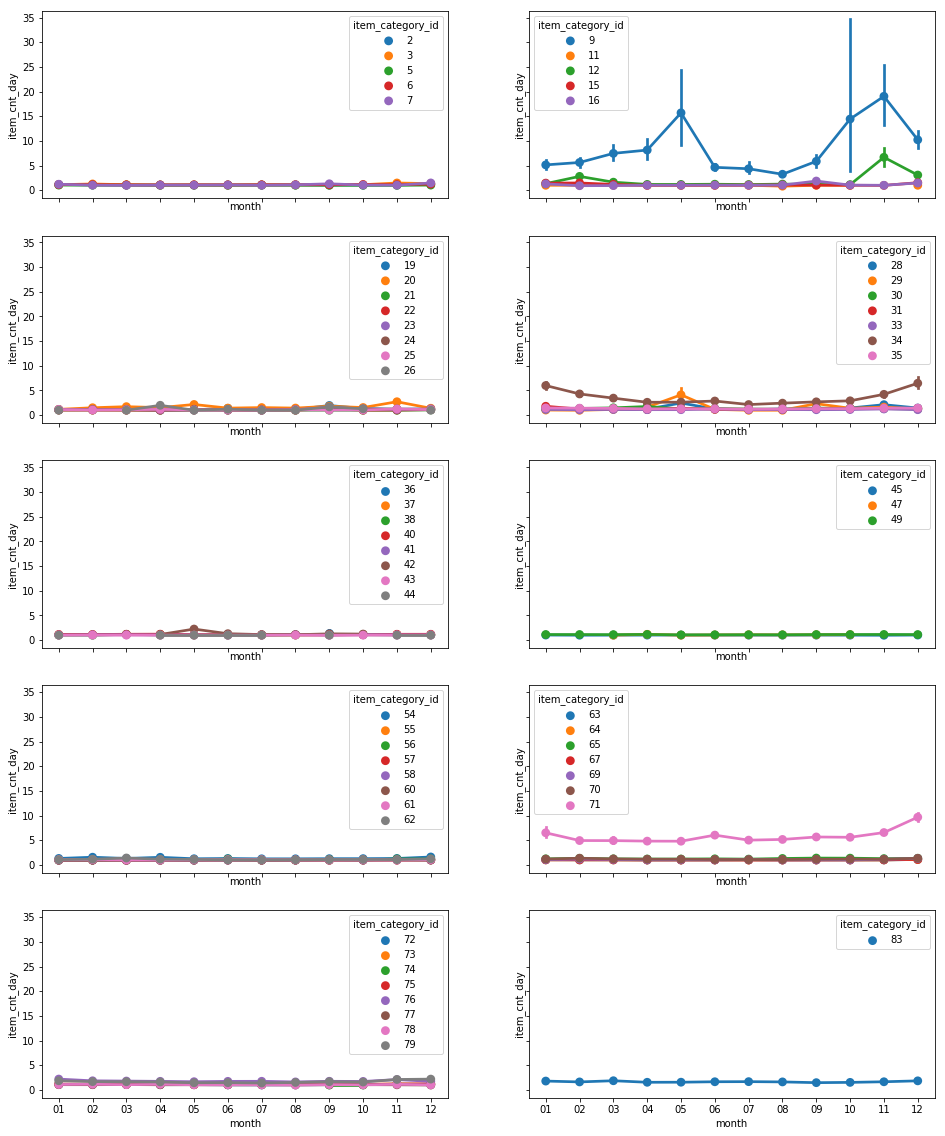
After our EDA, we proceed to research the best modeling methods for our problem of predicting future sales. We then gathered more knowledge from various resources including Kaggle kernels, various data science blog, and Github repositories including *luliu31415926*’s, who ranked 6th on Kaggle's public leaderboard for the same 1C competition and also *entron*’s, the winner of a similar competition called Rossman Sales. Combining our EDA insights and research findings, we decided to employ a recurrent neural network model call long short-term memory (LSTM). Each LSTM network unit will be responsible for ‘remembering’ values over arbitrary time intervals to help us with our forecast. We hope that the model is robust enough to tackle situations for our nuanced datasets that have changes for shops and items from month to month.

We see below the data exploration visualizations for all shops from the EDA notebook:

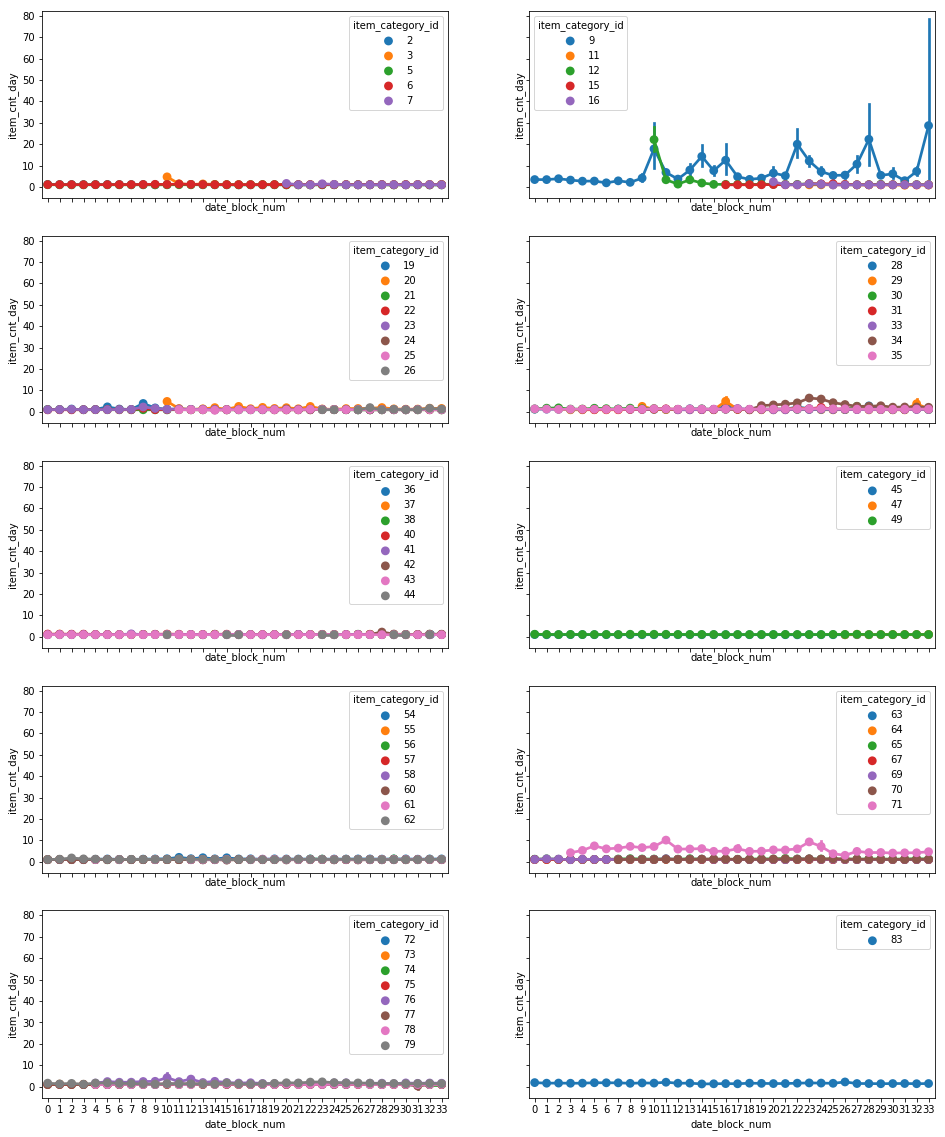


We note again the increases towards the end of each year around November indicating higher sales during the beginning of each holiday season. Thus, we will add the categories of month and year so our neural network will recognize this pattern.

Plots of historical monthly sales for each item category:



We recognize from the display above that only a few categories such as 8, 9 and 71 do better overall and vary significantly on a monthly basis in all 34 months while most other categories have constant monthly sales. Below, we will plot the trends of sales for all 34 months for each item category.

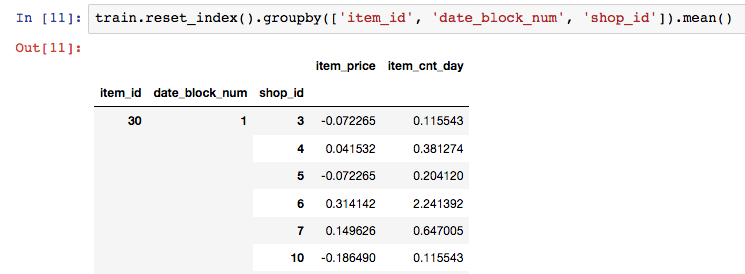


We see above again that generally, most stores actually have a fairly constant trend with a few exceptions with a similar seasonal trend as our previous total sales plot for the entire company. Thus, because of the large numbers of items, we will skip the tedious task of looking at the trends for each item and just focus on the monthly and yearly trends for each of the 60 shops and 84 item categories.

Since there are close to 3 million items in our sales training data and the sale data from early 2013 is unlikely to have high predictive power for Novembmer 2015 sales prediction, it will be more efficient if we examine the patterns of how each shop and item category vary over time while narrowing our data to the months leading up to November (July-October) in 2013 and 2014.

**Recurrent Neural Network**

We then build our recurrent neural network units for the training and prediction tasks. Our long short-term memory (LSTM) networks are composed of units that each contain a cell, an input gate, an output gate, and forget gate. Each cell is responsible for remembering values over arbitrary time intervals, which should help us with our time-series data that contain changes in shop and item categories from month to month. Since we are using a narrower sequence for our machine learning, we will have to fill in the resulting empty values with the closest past record due to the price of each item varying slightly by store location and month. Our data is now significantly smaller from 3 million rows to 600,159 leading to a reasonable 30% test set fo 214,200. Additionally, we scale the data to speed up our learning.



After building our LSTM model, we will now finally able to evaluate all of our operations and analysis so far by computing the root mean squared error for our model above using 13 epochs. Our resulting scores are:

Train Score: 2.52 RMSE  
 Test Score: 1.45 RMSE

We observe that our model perform fairly well with an average error of 2.52 for the training data set and 1.45 for the test set. We will next incrementally fit the LSTM model on our validation set and create a submission csv for our Kaggle competition (scores for 1 epoch are 3.85 RMSE train score and 4.29 RMSE test score.

Although our RMSE is fairly good, my research findings did indicate that if we can generate an ensemble of five different machine learning models including XGBRegressor, Random Forest, Linear Regression, simple neural network, and embedding neural network, this would lead to significant improvement over any single model.

Sources:

<https://github.com/luliu31415926/kaggle_1c_company_predict_sales><https://github.com/cbenge509/1C_Regression><https://www.kaggle.com/jagangupta/time-series-basics-exploring-traditional-ts><https://www.kaggle.com/the1owl/playing-in-the-sandbox><https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/><https://www.kaggle.com/minhtriet/a-beginner-guide-for-sale-data-prediction>