Problem/Overview:

Individual retail stores based in population dense areas are always out-of-stock or running low on many of their more popular products. The stem of this issue occurs are three different levels of the retail experience: the micro-level and macro-level of the consumers (individual retail stores and corporate), and the supplier (production company). Often when customers are faced with the “out-of-stock” reply for their desired products, they become deterred to try again at the same store later, or, even worse, to completely give up on physical stores to make their purchases. This is a large cause for the movement of physical shopping to online shopping where they can easily get the status of all products at a moment’s notice, and the purchases are made and delivered at the customers’ convenience.

The loss of physical customers usually results in the loss of loyal customers as most online customers will purchase their desired item from the cheapest, reliable source available, which is usually the larger, more well-known corporations that can afford to cut their sale prices. This causes many local or smaller corporations to have increased churn rates, and eventually go out of business.

My objective for this project is to experiment with a sample of retail data, specifically from 1C Company, a Russian software firm. I will analyze the data and generate a predictive model that can reliably forecast the future sales of their products for up to a month.

Potential Clients:

The potential clients for this project include the individual retail stores, the larger corporations, and the production companies. Individual retail stores would utilize this data for two main purposes. The first is to determine which products are still desired and which ones should be replaced to bring in potentially larger profit margins. The second is to help managers keep track of the influx and efflux of each product to more efficiently maintain stocked inventory, thus preventing reduced profits if the customer chooses to buy the product elsewhere. The short-term loss of customers usually has long-term results as poor customer experience can affect customer-retention rates.

Corporate retail stores would use this data for similar reasons as individual retail stores, but at a larger magnitude. However, poor customer experience here has a larger affect as it affects customer loyalty and the reception of corporate reliability.

Production companies could apply this data to better prioritize their manufacturing process to focus their efforts on the production of items that have higher sale value. If expanded with geographical information, production companies can focus their shipments to areas where their products are selling well.

Data Wrangling:

This dataset was provided by the 1C Company to a Kaggle Competition several months ago. The time-series dataset consists of daily sales data ranging from Jan. 2013 to Oct. 2015. It consists of csv files containing information about over twenty thousand individual products, multiple product categories, and sixty individual stores. The time series data contains approximately a dozen attributes, consisting of a mixture of both categorical and quantitative information.

Working with a Kaggle Dataset, most of the data was already cleanly organized. This includes data being organized into 3 categorical datasets: items.csv, item_categories.csv, and shops.csv. The training and testing data were also cleanly provided in sales_train_v2.csv and test.csv. After all the datasets were properly imported into Jupyter Notebook, the first step was to use the .head() and .info() parameters of dataframes to see how the data was organized.

Cleaning the Dataset and Dealing with Missing Values:

Immediately I realized that the date column of the training data had a date column, but the dtype was made up of strings. In addition, the string format of the date (DD.MM.YYYY) did not allow me to use parse parameter. As a result, I first defined a function that restructured the string format (YYYY.MM.DD) to an easier to understand and organize format. With a properly restructured string format, Pandas was able to easily parse the date column and convert all entries into datetime objects. I also did a quick search for null values and found none so no steps were necessary to address null values.

Locating and Dealing with Potential Outliers:

Following that, I used the describe parameter to do a quick search for outliers in the dataset. Immediately, I noticed potential outliers in the item_price and item_cnt_day columns. In the item_price column, there was a single negative value (-1.0) and a single extreme value (307980.0) that was multitudes higher than the second largest. Since there was only 1 entry of each, I felt it was safe to call these outliers and remove them from the analysis. In addition, it didn’t make logical sense to sell items are a negative price or at that high of a price. For the item_cnt_day column, there were more than 7300 entries that were negative. It was highly likely that these are due to returns of an item, but no information was provided and no additional information could be gathered. Since there are over 3 million datapoints in our dataset, 7300 is very small percentage that shouldn’t affect the model too much. Nonetheless, I chose to make two datasets, 1 removing the negative entries, and 1 keeping them. I will apply my model to both datasets to check for performance.

Other:

After exploring the three categorical datasets, I realized that the item, item category, and shop names all contained a large amount of Russian characters. There was also no way to extract only the English characters from the columns and it would be costly to translate in a dataset this large, so I chose to exclude those columns from my analysis. This is a valid choice because all items, item categories, and shops have their own respective unique IDs.

Exploratory Data Analysis and Data Story:

The analysis started with a time-series plot of the total monthly sales from Jan. 2013 to Oct. 2015 for three randomly chosen shops. With this data, I hoped to see any obvious trend as well as any seasonal or quarterly patterns within the dataset. From the plot, I observed a general decline in sales over time (2013 > 2014 > 2015), in addition to an annual peak on December of 2013 and 2014. This peak overlaps with several holidays such as Christmas and New Years which could be the cause for the reoccurring peaks.

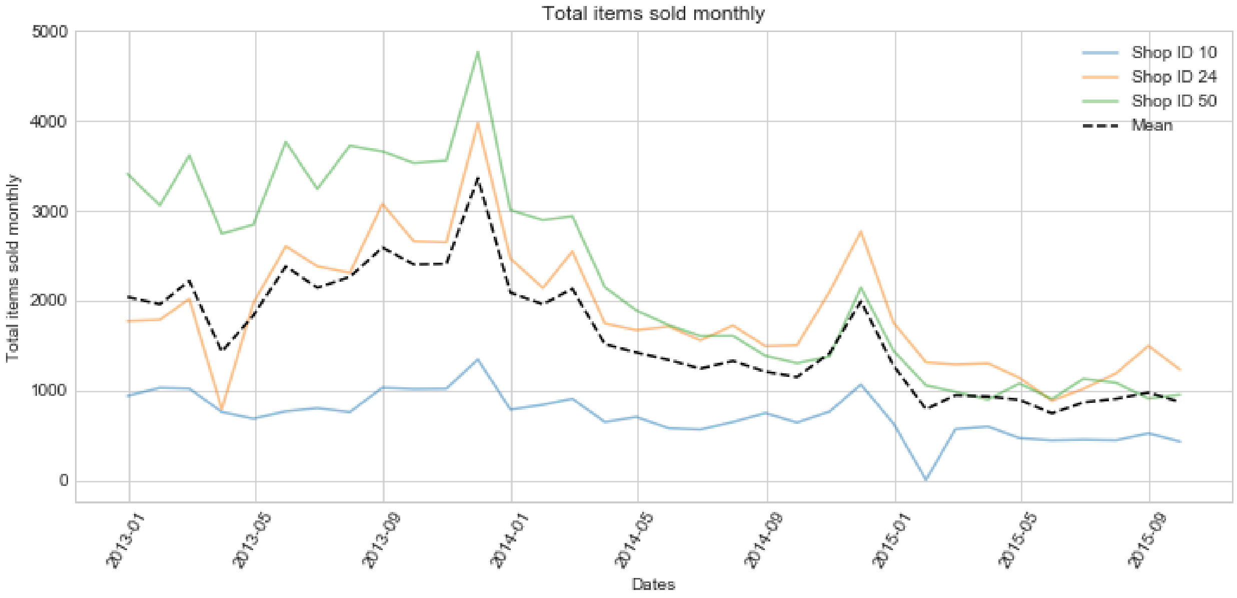

Figure 1. The time-series plot shows an apparent decline in overall sales over time. However, there is a reoccurring peak at December in both 2013 and 2014, which suggests that there is likely a seasonal event that promotes sales around that time.

To follow-up with this trend, I manipulated the data a bit to only utilize data from the two years with a full set of data, 2013 and 2014, and used a combination of a histogram and bar plot. The two plots showed increased total entries and total items sold in the month of December which reflect the same pattern as seen in the time-series plot.

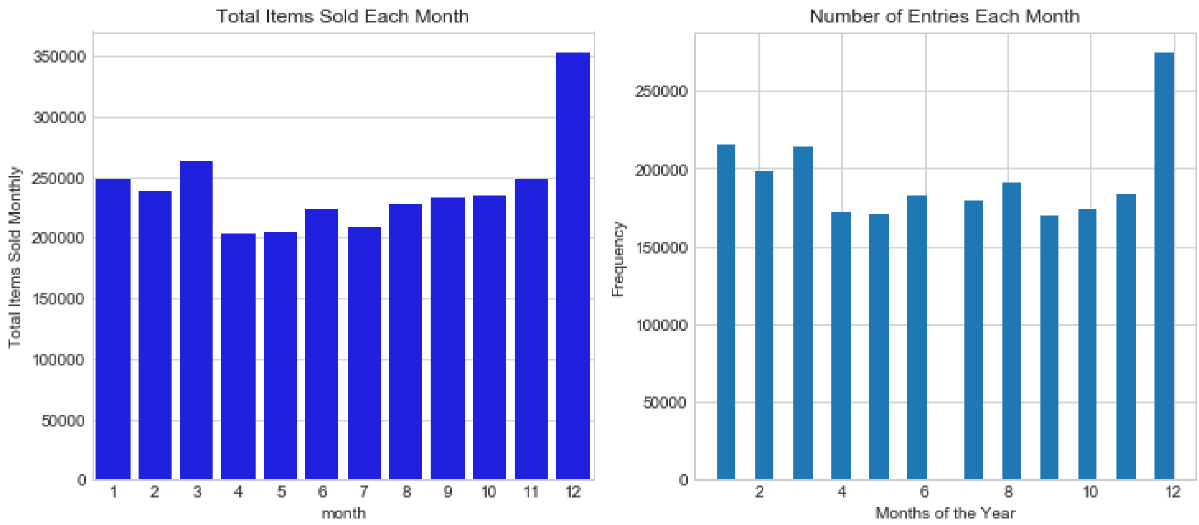

Figure 2. The histogram and bar plot of data from 2013 and 2014 show increased total entries and total sales during the month of December. This pattern is reflective of the pattern seen in the previous time-series plot.

To create the model, it was important to understand how each shop performed relative to each other. Using a combination of both a histogram and bar plots, a quick outline of shop performance was available. This quick outline showed some shops significantly outperforming others (such as shops with IDs 25 and 31), but mainly many shows that performed at the same low level.

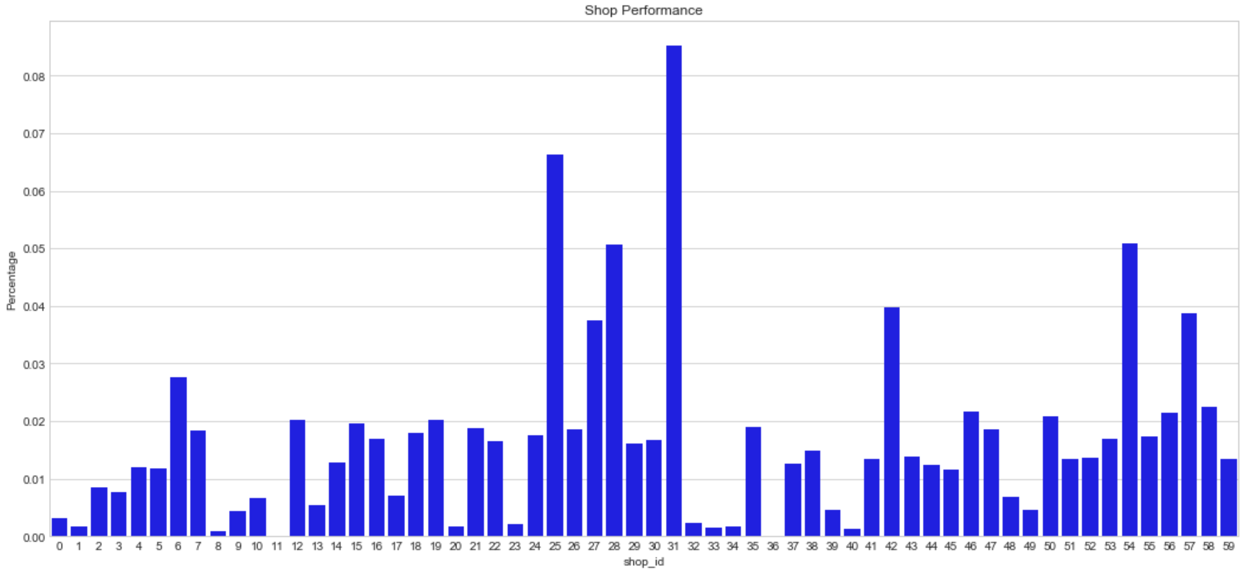

Figure 3. The normalized bar plot shows performance of shops relative to each other. Key points of the plot are the strong performing shops and the many shops at similarly low performance.

Similarly to shop IDs, it was important to understand how each item category ID performed relative to each other. For this plot, it was also noted how many shows show simiarly low performance.

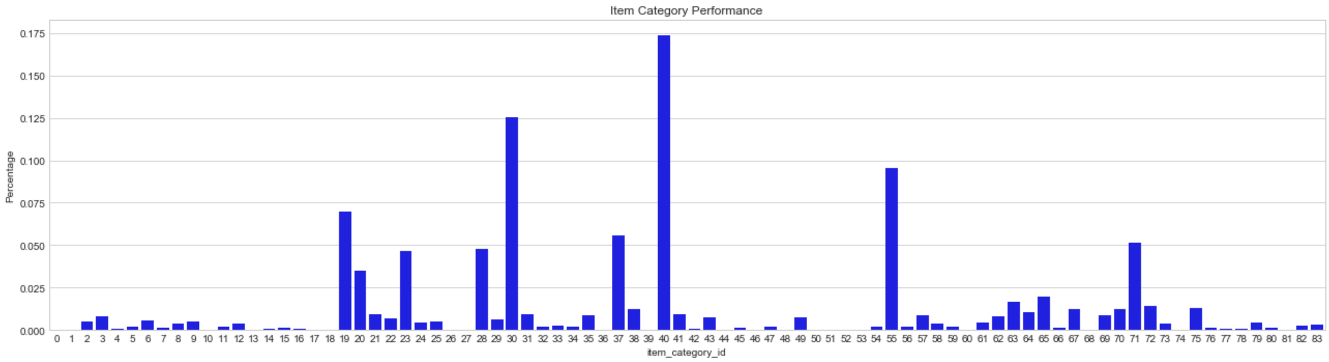

Figure 4. The normalized bar plot of item categories shows relative performance. Key points to notice are the strong performing shops and similarly performing item categories.

Inferential Statistics:

After exploring the datasets, there are a couple of things we can note. The most relevant is that there are nearly 22,000 unique item ids. Considering that we need to use dummy variables to apply categorical data into our linear regression model, it is impractical and too computationally intensive to work with that many values. However, these item ids are also separated into item categories, which have 84 unique ids. At this point, we must acknowledge that by using item category ids to replace individual item ids, we are allowing our model to assume that all items in each category do not have any statistical differences from each other. This assumption can be fatal as performance can vary heavily in each category.

The main variables we will be working with to build the model are item category ids and shop ids. Both these variables are categorical data and contain 84 and 60 unique IDs respectively. To reduce the number of dummy variables we need to create and work with, we created groups for categories and shops that performed at approximately the same level. The result for the item categories was 6 groupings and 11 standalone categories to give us a total of 17 values to work with. For the shops, there were 5 groupings and 8 standalone shops to give us a total of 13 values to work with.

To validate the groupings for both variables, we utilized two-sample t-tests. In our t-tests, the null hypothesis was that there was no statistical difference between our groupings, and alternative hypothesis being there was statistical differences. Using a significance threshold of 0.05, we were able to validate our groupings. Due to a t-score strongly supporting statistical differences, I did not feel it was necessary to obtain further support via the frequentist approach.