Data Capstone Project 1

Eric Huynh

April 2 Cohort

Problem/Overiew:

Individual retail stores based in population dense areas are always out-of-stock or 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, may 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 consumers will purchase their desired item from the cheapest, most 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 effect 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:

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 three tables 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. The total size of the described data set is nearly three million recorded entries.

Approaching the Problem:

The first focus will be to explore how each variable correlates with the items sold per day, then aggregate this data over each month to reflect the format of the desired project product. This initial association study will be limited to individual retail stores, then expanded to the macro-level by applying it to larger samples of stores. Another approach could focus on individual products then expanded to more products. However, because there could be heavy deviations between the performance of individual products, this method could make the correlations confusing.

Next, I will focus on cleaning the data. This involves thoroughly checking the data for any wrongly inputted entries, such as negative values in certain variables and null values). Any necessary missing data will be filled in appropriately. If there are any incorrect data types, such as floats casted as strings or, more commonly, datetime objects casted as strings, the cleaning will address that.

The last part after getting a better understanding of the variable associations and domain is to dive into the data and begin an in-depth visual and statistical analysis.

Data Wrangling:

As a Kaggle Dataset, most of the data was already cleanly organized with the data being organized into 3 categorical datasets: items.csv, item_categories.csv, and shops.csv. The training and testing data were also provided in two separate files: 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(), .info(), .describe() functions 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 was casted as the data type “object”, which suggested that it was being read as strings. In addition, the string format of the date (DD.MM.YYYY) did not allow me to use parse parameter of the read_csv function. As a result, I first defined a function that restructured the string format (YYYY.MM.DD) to an accepted date-parsing 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 in the training data and determined that there were no missing values in the entries so no steps were necessary to address null values.

However, when comparing the training and test data, I found that both sets of data used two different, but overlapping, sets of item IDs. Since the training data did not have a couple of the item IDs used in the test data, additional item IDs were added to the training data and filled in using the mode of each respective column. This was a necessary step because the model could not predict for item IDs that did not appear in the training data. By doing so, this allowed me to create item groupings for items not previously mentioned in the training data. Another approach would have been to assign the new item IDs into groups by their category IDs.

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 did not make logical sense to sell items at 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 supplementary information was provided, and no additional information could be gathered. Since there are nearly 3 million datapoints in the dataset, 7300 is very small percentage that should not affect the model too much. Nonetheless, I chose to make two datasets, 1 removing the negative entries, and 1 keeping them. I will be applying my model to the dataset with the negative entries. Provided the time, I will apply my model to the non-negative dataset 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. However, I acknowledge that using the names could be a preferable method for making the groupings as products with the same function would perform similarly in sales.

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.

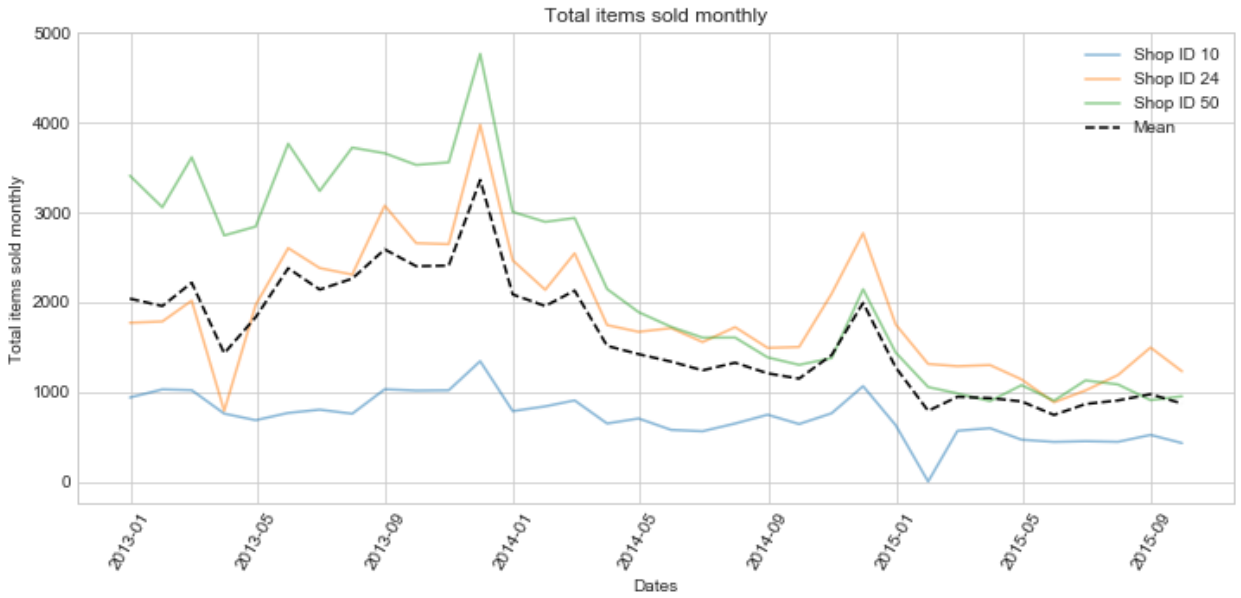

Figure 1. The time-series plot shows an apparent decline in overall sales over time. However, there is a reoccurring peak at December in both 2013 and 2014, which suggests that there is likely a seasonal event that promotes sales around that time.

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.

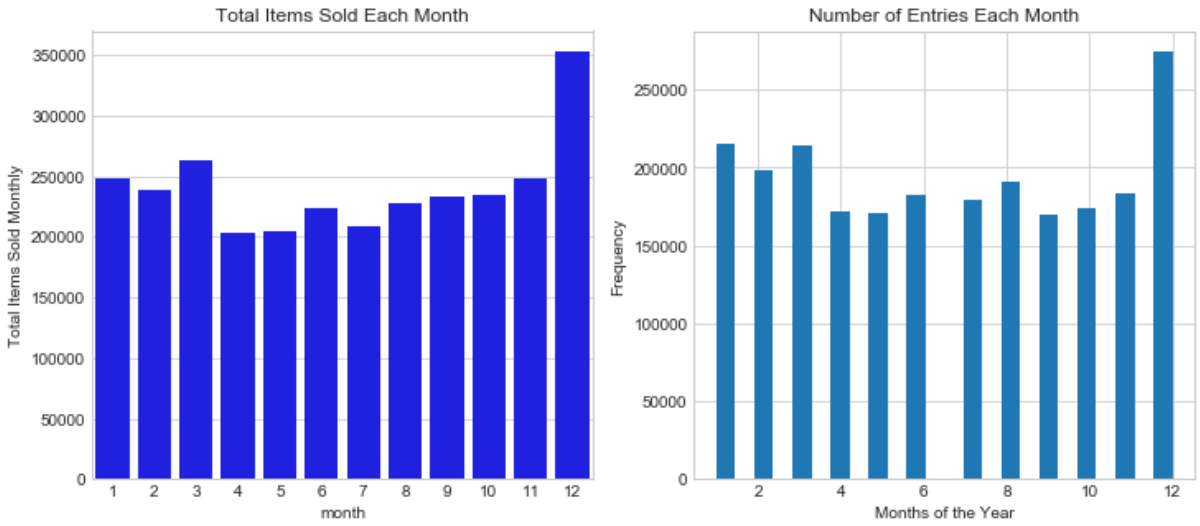

Figure 2. The histogram and bar plot of data from 2013 and 2014 show increased total entries and total sales during the month of December. This pattern is reflective of the pattern seen in the previous time-series plot.

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.

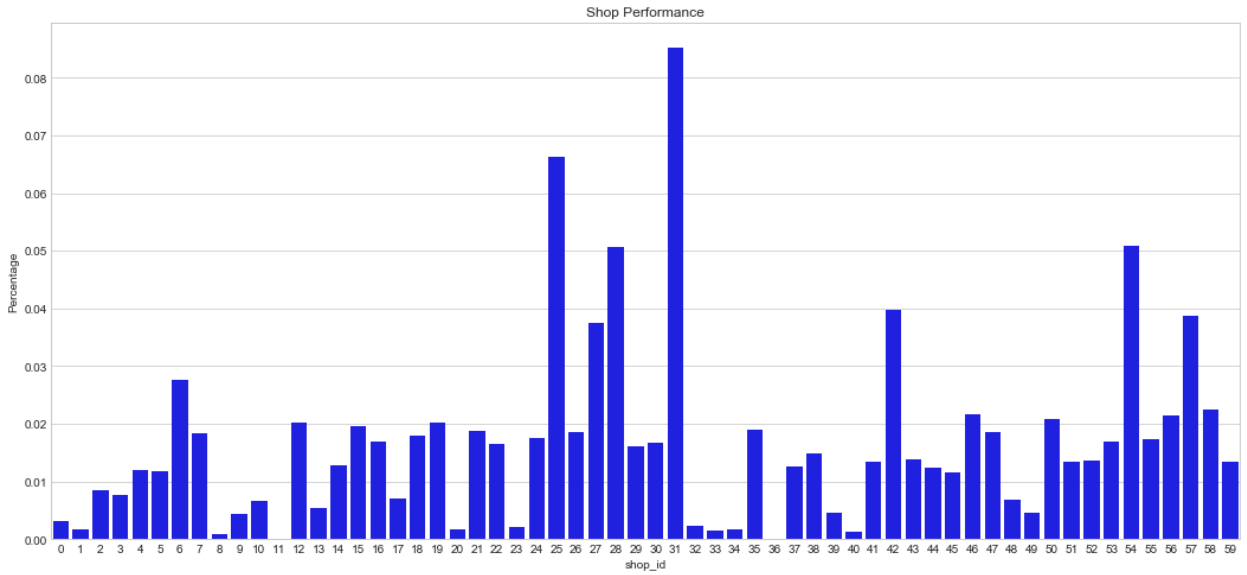

Figure 3. The normalized bar plot shows performance of shops relative to each other. Key points of the plot are the strong performing shops and the many shops at similarly low performance.

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 shops show simiarly low performance.

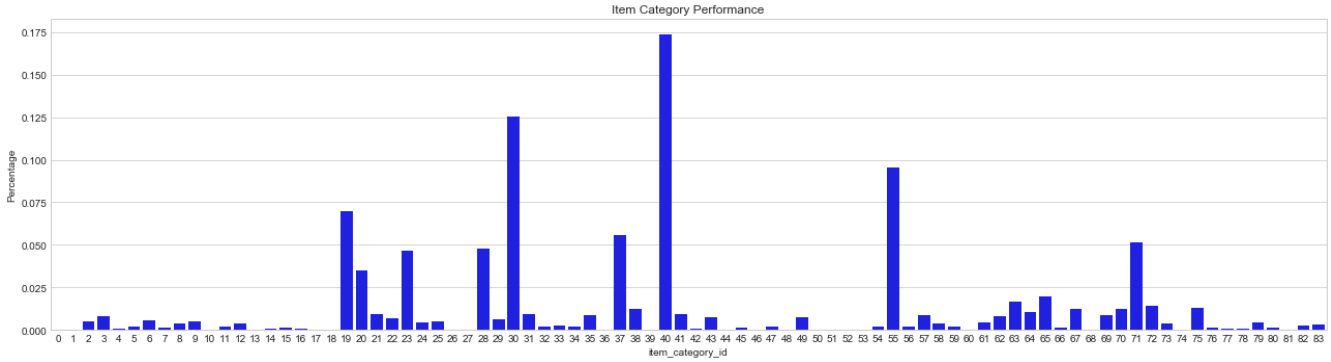

Figure 4. The normalized bar plot of item categories shows relative performance. Key points to notice are the strong performing shops and similarly performing item categories.

Inferential Statistics:

After exploring the datasets, there are a couple of things I could note. The most relevant is that there are nearly 22,000 unique item IDs. Considering that I needed to use dummy variables to apply categorical data into the 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, I acknowledged that by using item category IDs to replace individual item IDs, I am allowing the model to assume that all items in each category do not have any statistical significant differences from each other. This assumption can be fatal as performance can vary heavily in each category. An additional alternative I used was to construct groups of item IDs, grouping together similarly performing items.

The main variables I worked with to build the model were 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.

With the alternative, I constructed a total of 24 groups of item IDs that consisted of item IDs that performed similarly in total sales.

To validate the groupings for both variables, I utilized two-sample t-tests. In the t-tests, the null hypothesis was that there were no statistical differences between the groupings, and alternative hypothesis being there were statistical differences. Using a significance threshold of 0.05, I was able to validate the 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.

Building the Model:

I built a simple supervised linear regression model of the data using 4 variables: item IDs, shop IDs, year, and month. I experimented with a variation of the item IDs and shop IDs, along with manually inputted values for the year and month variables to reflect November 2015 (year=2015, month=11). Prior to model experimentation, I built the model using grouped category IDs and grouped shop IDs by grouping similarly performing unique IDs.

After careful experimentation, I realized that the shop and item category groupings yielded very poor results. In addition, I realized that it was computationally feasible to work with more dummy variables, which I did by completely removing or increasing the number of groupings. As a result, I attempted multiple different approaches including removing both original groupings and grouping similarly performing item IDs instead. I generated 24 groups for similarly performing item IDs.

Preprocessing:

Since I worked with four categorical variables, I generated dummy variables to use in both our training and testing. However, I must note that there was a choice to use the date as a continuous (time-series) variable. I chose to apply the date as a categorical variable instead

because I believed that, as a time-series, the date poorly reflected the yearly cyclical trends that I observed in my initial analysis. I may choose to return to this point in the future to retrain the model using the data as a time-series.

To appropriately train and test the model, I split the data into train and test data sets. Twenty percent of the dataset was allocated to the test data set with the randomly chosen random_state 21 to ensure reproducibility. I applied the month variable to the stratify parameter to ensure that the months were equally represented during the training.

Variable Experimentation & Accuracy Measurements:

Thus far, I have attempted a total of 5 variations of the model variables. The first model used the generated shop and item category groupings. Using the double groupings, I observed that many of the predicted values, in both training and test sets, were right skewed. In addition, the plotted histograms show that there is an irregular amount of negative predictions. Using seaborn's regression plot, I also observed that the model underestimates the values in its predictions and has a cutoff value for the max predicted values which is not reflected in the real values (Figure 5). Root Mean Squared Error was used as the accuracy metric to reflect the Kaggle competitions method for determining the scores. Calculating the score with our split test data, we achieved a RMSE score of 426.39, whereas Kaggle scored the predictions as 1198.72.

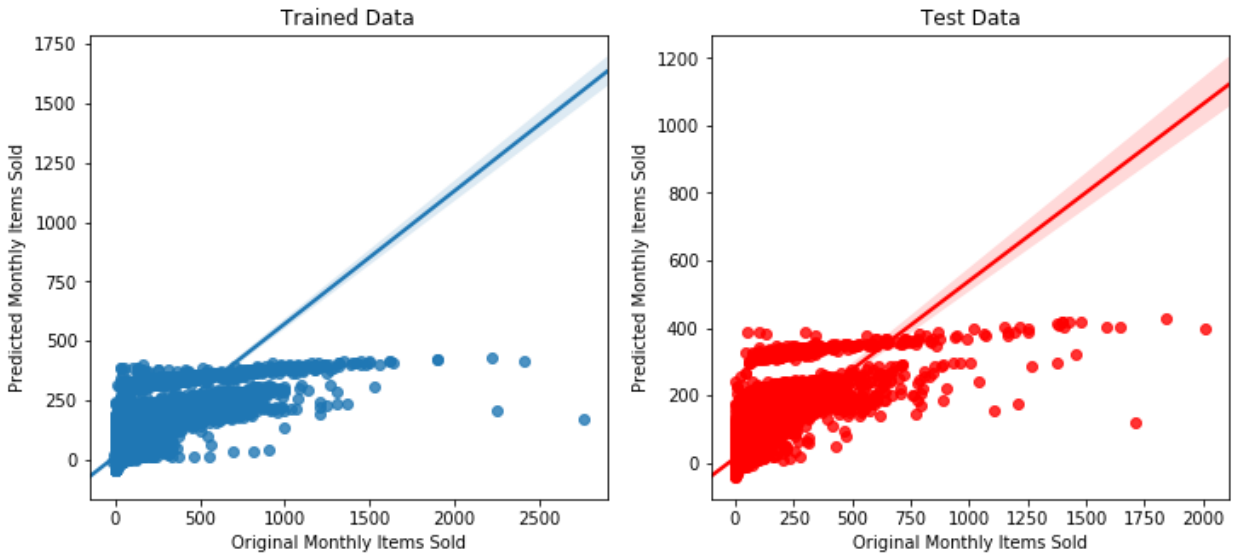

Figure 5. Regression plot of model 1. The model appears to consistently underestimate the values. There are also unexpected max value cutoffs in the predictions that are not in the original.

The second model scrapped the grouped shop IDs, but continued to use the item category groups. The predicted values continue to have a right skew along with a couple of negative predictions. From the regression plots, we observed a less drastic max cutoff value, but similar underestimated predicted values (Figure 6). Calculating the score with our split test data, we achieved a RMSE score of 154.30, whereas Kaggle scored the predictions as 243.86.

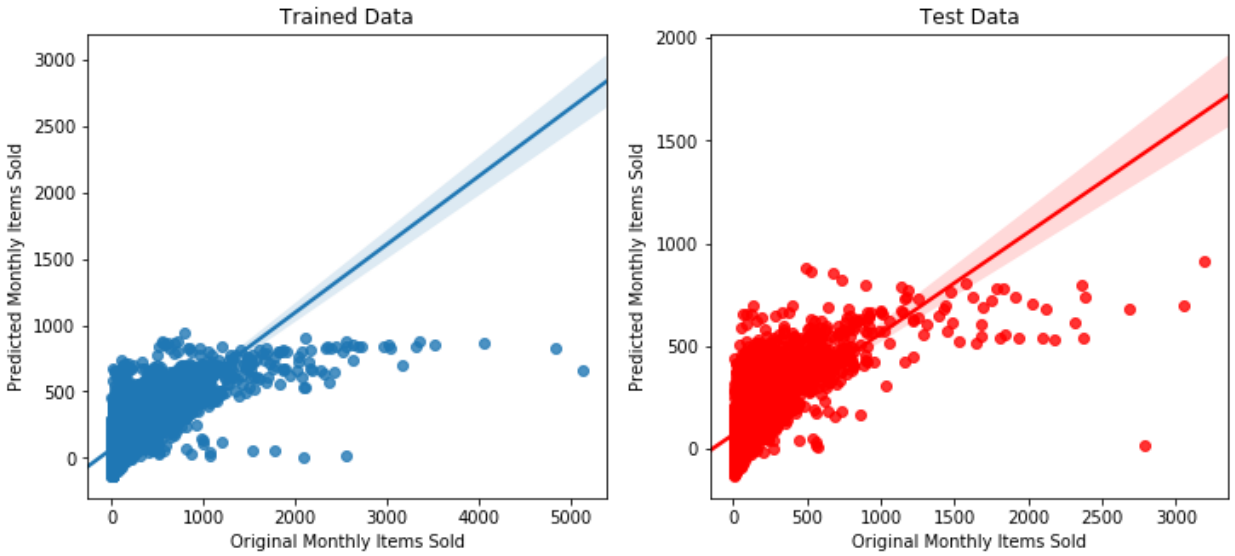

Figure 6. Regression plot of the second model. This is the only model that did not have that irregular y-axis cutoff seen in all the other plots.

The third model scrapped both groups and used the individual shop and item category IDs. The plots follow similar trends to the two previous models (underestimated values along with a right skew). However, the RMSE score for the model was unexpected as the generated RMSE was 90.17 (an improvement from 154.30 from the second model) while the Kaggle scored the prediction as 260.18 (a drop from 243.86).

The fourth model and fifth models used grouped item IDs to train the model. The data for the fifth model was averaged across the time span of the data (the two and a half years were combined). Both scored better than the previous three models but was far higher than the desired score (Figure 3).

Model:	Changed Variable:	Generated RMSE:	Kaggle RMSE:
1	Grouped Shop ID and Item Category ID	426.39	1198.72
2	Scrapped Shop ID	154.30	243.86
3	Scrapped Item Category ID	90.17	260.18
4	Shop IDs and Grouped Item IDs	60.45	152.59
5	Averaged Data and removed year variable	47.46	161.70

Figure 7. Table describing the changes made between each consecutive model. Scores from both Kaggle and split test data are provided as an accuracy measurement.

Noteworthy:

Only the second model had a regression plot that did not have an irregular max cutoff value seen in all the other models (Figure 1 vs. Figure 3).

Future Direction:

Provided time, I would first attempt to add preprocessing steps from the sklearn.preprocessing module to further clean up the data for the model. In addition, I would explore different classifiers or combinations of classifiers to better accommodate the dataset. A possible classifier is the Support Vector Regressor with a polynomial kernel which might help address some of the non-linear aspects of our data. It's also important to optimize the parameters of the classifiers that I choose.

Next, I'd go back to the beginning and attempt different methods of cleaning the data. Especially where I rejected the entries with the negative item price and the extreme value (307980.0). The poor score could also be due to the negative predictions by the model due to the trained values with negative item counts. The negative values for these variables could have a meaning that was not described by the provider.

Another data wrangling approach I would like to attempt is interpreting the data column as a single continuous variable instead of two categorical variables (month and year). By splitting the data into three years for the year variable, this may have left the data for 2014 and 2015 out of the model's analysis. Additionally, I would like to utilize the dataset I constructed with the negative item counts removed to avoid negative predictions by the model. I believe the negative predictions could be a result of these entries overwhelming the positive entries for their respective item IDs.

I would also like to explore the second model for the cause to the irregular max cutoff for model's predicted values. It is an irregular appearance as the max items sold per month in the training data is far greater than the cutoff for the predicted values.