Retail Sales Predictions

Thomas Rauzi

Being able to predict sales is vital for every business. When a company has an accurate idea of how much money they will make, they can predict how much labor and material they will need. Since businesses have to schedule employees and purchase material in advance, they must be able to forecast future demand. Additionally, the client can use the forecast to determine if a future project may be worth exploring. For instance, the company can incorporate the sales projections into a project proposal to determine if the project will pay off.

We use retail sales data from January 1st 2010 to December 7, 2013 to build a simple exponential, double exponential, two seasonal autoregressive moving averages (SARIMA) and long short-term memory models (LSTM) to predict sales four weeks in advance. LSTM provides the most accurate forecast while the simple exponential model was a close second. SARIMA 2, SARIMA 1, and double exponential are third, fourth, and fifth ranked models, respectively.

Data Wrangling

The data is contained in three different datasets: features, stores, and sales from Kaggle (https://www.kaggle.com/datasets/manjeetsingh/retaildataset?select=sales+data-set.csv). Features data sets contains information on external factors that affects a store across time. It contains store number, date, consumer price index (CPI), unemployment rate, temperature, fuel price, unemployment, is a holiday, and five different markdowns. Markdowns are where the store reduced prices in attempt to encourage sales. Holiday variable is a binary variable with True for the week having a holiday and false otherwise. All variables except store number, date, and holiday are set as floats. Store dataset includes information on store number, store type and store size. Store number and type are set to an object while size is an integer. Finally, sales table lists sales by store, department number and week, and it also contains a holiday column. Department number is set to a string like store number.

The data is weekly on 45 stores from January 1st 2010 to December 7, 2013. There are supposedly no missing data points; however, there are periods with negative sales data. Sales data should not be negative, so sales data has a range constrain issue. The number of departments per store ranges between 61 and 79 with a mean of 74 and median of 77. Not all stores and departments have the same number of observations. For example, store 9's department 94 has 74 observations while department 97 only has 9 observations.

Exploratory Data Analysis

For both CPI and unemployment, we are missing one week in march 2013, the whole month of May, June, and July, and one week in October and December. The three-month gap could cause issue when trying to impute the data, so we may only use the data before this missing period. Also, CPI seems to have two distinct groups from the histogram, which is shown in figure 1.

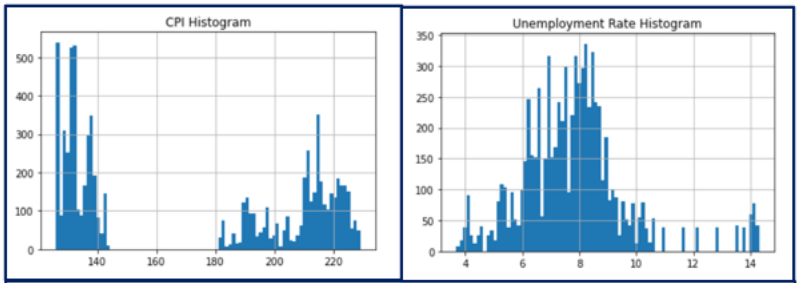


Figure 1: CPI and Unemployment

Each of the three store types seem to correspond to a certain range of store sizes since the boxplots only overlap with the potential outliers. Additionally, the store size has an interesting histogram shape where there are three different groups. The three groups likely coincide with the three types of stores, which suggests we can use the three store types in lieu of using store size. One limitation of store size is that type C only has 6 stores while A and B have 22 and 17, respectively. Boxplots and histograms for store size and type are shown in figure 2.

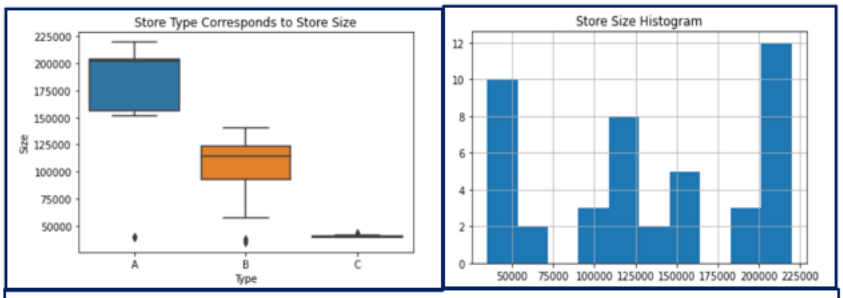


Figure 2: Store Type and Size

median markdowns and median weekly wages have a positive relationship, which is shown in figure 3; furthermore, the relationship depends on the type of store. The strength of the relationship appears constant between all three groups; however, type A stores have both higher markdowns and weekly sales. although weekly sales and markdowns all exhibit positive relationship, markdown 3 and 5 potential have a non-linear relationship.

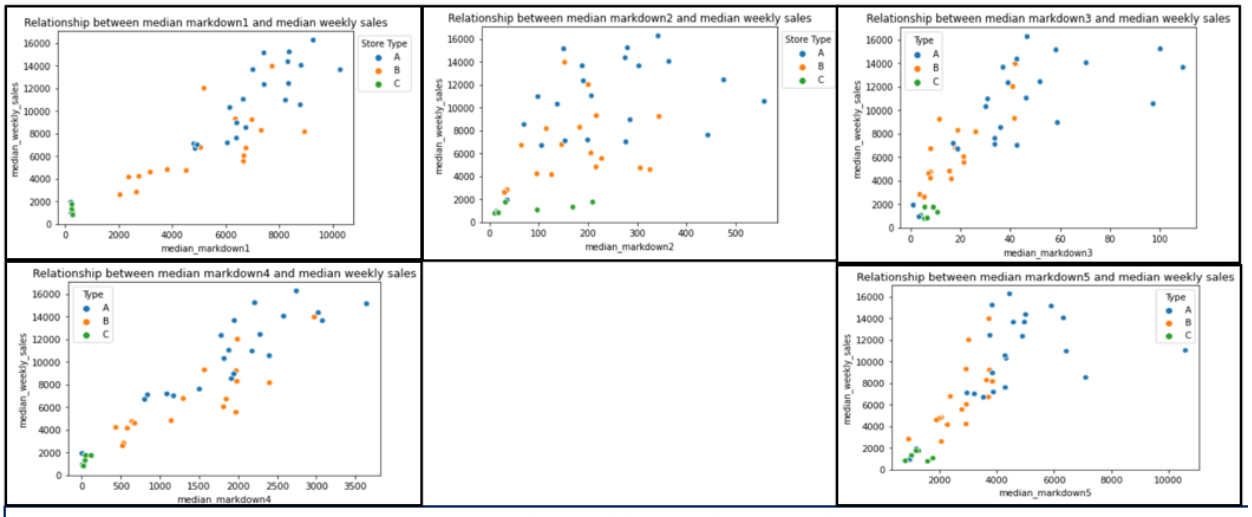


Figure 3: Median Markdowns and Median Sales

As shown in figure 4, store size looks to be directly related to weekly sales, and each of the three store types exhibits a different relationship. Additionally, each appears to have a separate boundary, so

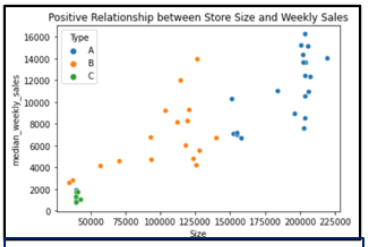


Figure 4: Store Size and Weekly Sales

the relationship between store size and median weekly sales has three groups, which are associated with the three store types. CPI, unemployment, and fuel price appear to have little impact on weekly sales at both the store and weekly level; however, the weekly sales there could be a relationship that is hidden by noise in the time series data.

Preprocessing

Since the average store has 77 departments, it would be too time consuming to model the department level data; furthermore, there is a large variation in the number of observations for each department. To avoid these issues, we will calculate total sales for each store by summing the departments. We replace all negative sales data with zeros before adding up the sales.

Next, we must decide if we will aggregate all of the stores or treat them separately. To answer this question, we begin by looking at the distribution of weekly sales for all stores to get an idea of an average week. Next, we use this information to create a threshold of potential weekly outliers by using 1.5 times the interquartile range (IQR). Using this threshold, we label each week as an outlier or not. Finally, we group all of the weeks by store and count the number of outlier weeks for each store and create the histogram shown in figure 5. Figure 5 indicates the majority of stores have over 120 weeks of outliers. Since each store only has 143 weeks of data, this suggests that these stores just have higher sales. Furthermore, the distributions indicate we would lose valuable information if we aggregated store data, so we should model each store separately.

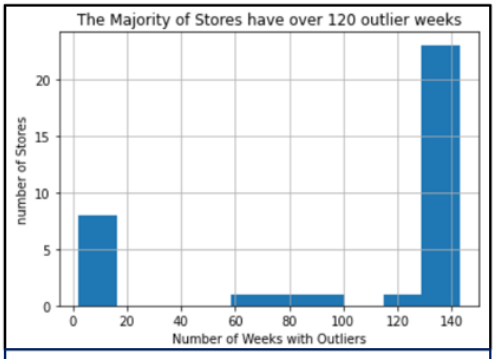
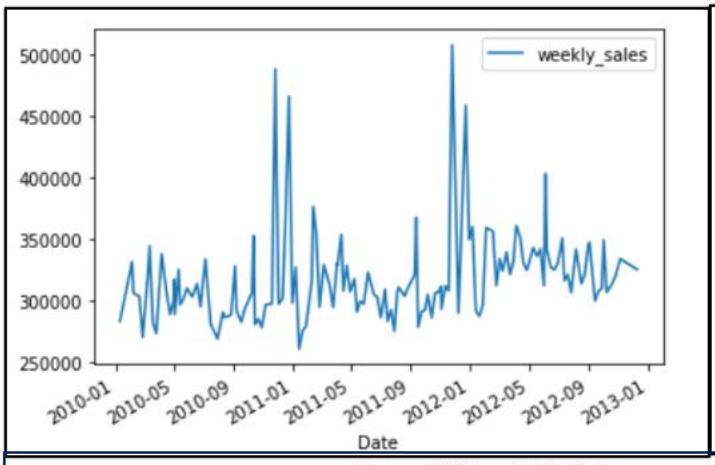


Figure 5: Stores and Number of Outlier Weeks

Having decided to model each store separately, we then divided the data into a train and test set where we used the last 36 weeks of data for the test set and the remaining weeks as the training set. Due to time limitation, we only model one of the stores. We randomly selected store 5.

Modeling

We will use exponential smoothing, double exponential smoothing, seasonal autoregressive moving average (SARIMA), and long short-term memory (LSTM) models. Figure 6 depicts the time series and descriptive statistics for store 5. There does not appear to have any trend, so exponential smoothing is a good starting point.

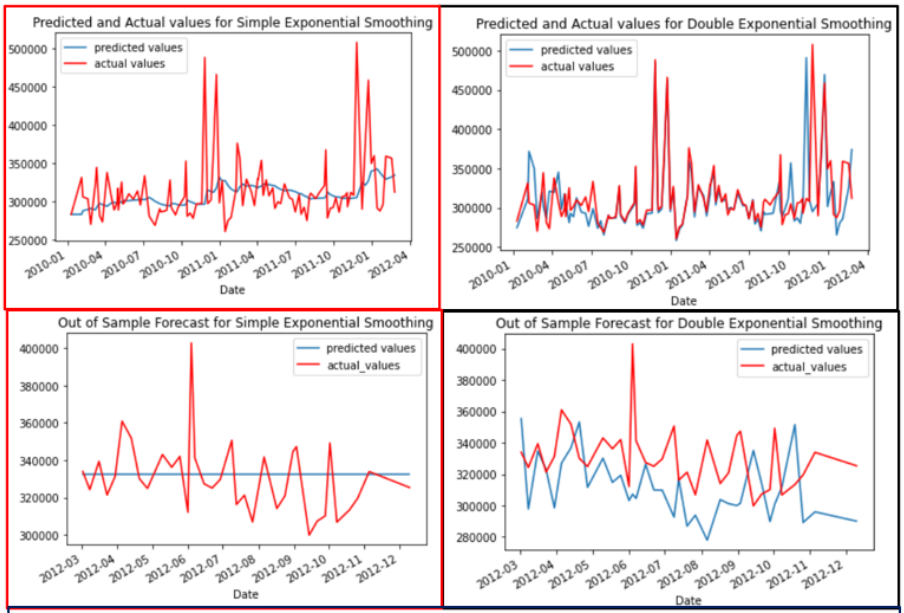


	weekly_sales
count	143.000000
mean	318014.823147
std	37737.946575
min	260636.710000
25%	294696.000000
50%	310338.170000
75%	329863.560000
max	507900.070000

Figure 6: Store 5 Statistics

Simple and Double Exponential Smoothing

Simple exponential smoothing seems to fit the average of the data fairly well. Since there is seasonality in the data, we also use double exponential smoothing where we account for the average, seasonality, and no trend. As shown in the figures below, double exponential smoothing follows the data considerably better in sample, which is shown by the root mean squared error (RMSE) of 34,196 versus 41,527 for simple exponential smoothing. However, the trend is reversed for out of sample where double exponential smoothing has an RMSE of 32,995 compared to 19,205 for the simple exponential model (see figure 7).



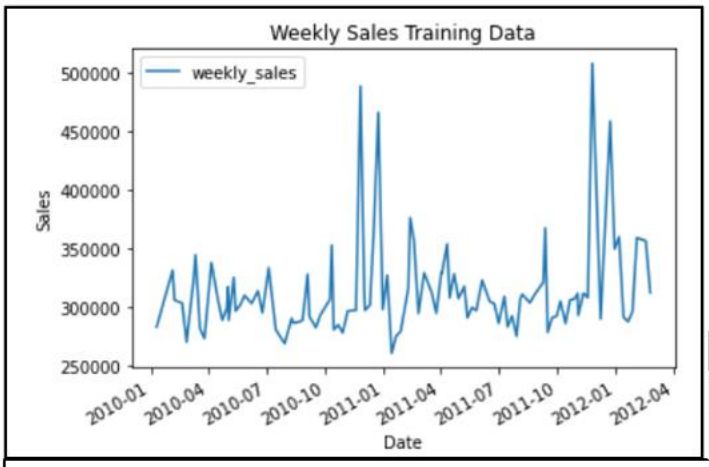
model	in_sample	out_sample
Simple Exponential	41527	19205
Double Exponential	34196	32994

Figure 7: Simple and Double Exponential Models

SARIMA

For SARIMA models the data must be stationary. A stationary time series means the mean and variance are constant and the covariance does not depend on time, but only the length of time between

observations. In other words, the covariance between points at time 10 and point 12 are the same as the covariance for points at time 100 and 102. From figure 8, the time series seems stationary, but to be more objective we conduct an Augmented Dicky Fuller unit root test. The null hypothesis is the data is



ADF Unit Root Test	
Test Statistic	P-value
-8.83	0

Figure 8: Test for stationarity ADF p-value is zero

non-stationary. With a p-value of near zero, we reject the null hypothesis and conclude there is sufficient evidence to say the time series is stationary.

Having determined the time series is stationary, we now determine the model order from the auto correlation function (ACF) and partial auto correlation function (PACF). ACF has a significant lag at 52, which suggest the seasonal period is 52. Since the data is weekly and there are 52 weeks in a year, it makes sense the period is 52. Additionally, there are no significant lags before 52, which suggests there is no autoregressive or moving average term for the non-seasonal part. To determine the seasonal order, usually we would need to plot lags of multiple of 52; however, due to the limited number of observations, we cannot plot a PACF with more than 54 lags. Since PACF decays at multiple of 52 while ACF drops after 52, the seasonal part is MA(1). Thus, the seasonal ARIMA mode is (0,0,0)(0,0,1)52. See figure 9 for the ACF and PACF with 10% significance level bands. Although the first lags are not statistically significant at the 10% level, the lags seem close to the threshold, so we also create a model with a non-seasonal MA term. The second SARIMA model has an order of (0,0,1)(0,0,1)52.

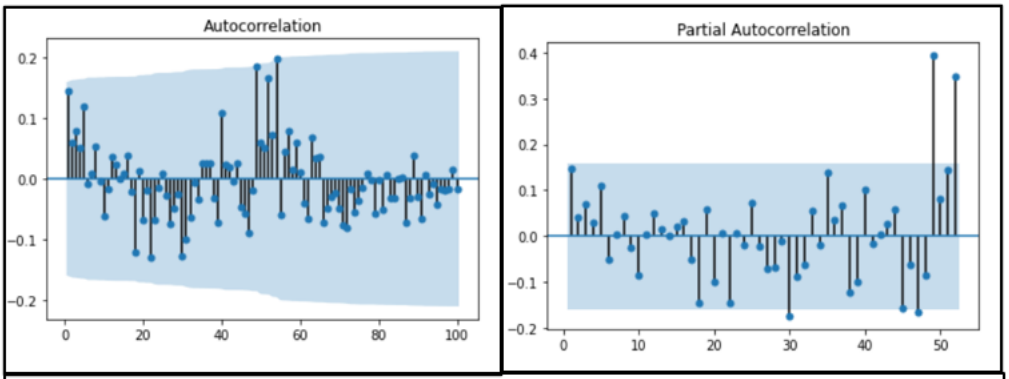
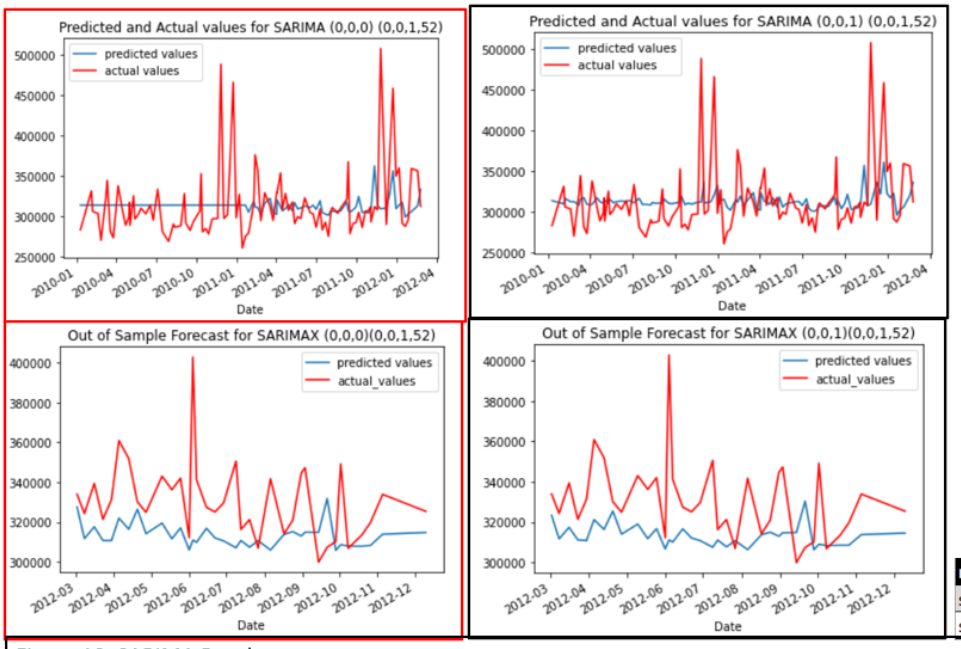


Figure 9: ACF and PACF

Figure 10 shows the actual and predicted values for both in and out of sample for both SARIMA1 (SARIMA (0,0,0)(0,0,1)52) and SARIMA2 (SARIMA (0,0,1)(0,0,1)52). In sample, SARIMA1 has a flat portion in the beginning because the model does not have a non-seasonal part, but the second part seems to follow the data quite well. SARIMA2 predicted values follow the data well over the entire period, which is why it's RMSE is 39,759 versus SARIMAX1's RMSE of 40,068. Additionally, SARIMA2 has a slightly better out of sample RMSE of 26,212 compared to SARIMA1's 26,227. Thus, their out of sample performance is virtually the same, so we would prefer SARIMA1 because we get the same performance with a simpler model.



model	in_sample	out_sample
sarimax1	40068	26227
sarimax2	39759	26212

Figure 10: SARIMA Results

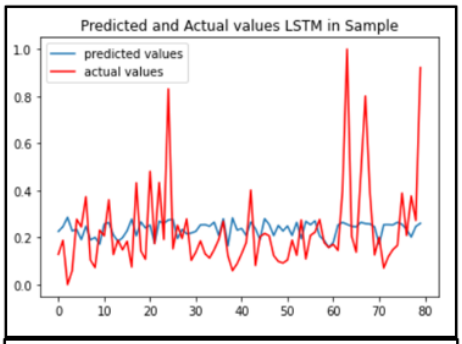
LSTM Model

For the LSTM model we normalized the data and create sequences. We normalize the data to ensure the recurrent neural network converges, and we utilize the SCIKIT Learn min max scaler. After normalizing the data, we create the target variable, which is the retail sales four weeks in the future. Thus, we just shift the training and test data by four weeks. Next, we create a sequence of 24 weeks, which is six months' worth of sales data to predict 4 weeks out. For the training data we create 80 samples or batches with overlapping 24 weeks of data, and the test data has 9 batches. After fitting the model on the normalized training data, we use the inverse transformation method to convert back to dollars.

Through trial and error, we arrived at a model with three LSTM cells where the first two levels have 50 hidden units, and the last unit has only one. Additionally, we include two layers that will drop 20% of the data to avoid over fitting. LSTM architecture is shown in figure 11. The in sample predicted values follows the movements of the actual data, and provides an RMSE of 44,537. Although the out of sample predicted values do not appear to follow the actual data at all, the RMSE is 15,066.

in_sample	out_sample
44537	15066

```
Layer (type)                 Output Shape              Param #
=================================================================
lstm_5 (LSTM)                (None, 24, 50)            10400

dropout_3 (Dropout)          (None, 24, 50)            0

lstm_6 (LSTM)                (None, 24, 50)            20200

dropout_4 (Dropout)          (None, 24, 50)            0

lstm_7 (LSTM)                (None, 1)                 208

=================================================================
Total params: 30,808
Trainable params: 30,808
Non-trainable params: 0
_________________________________________________________________
```



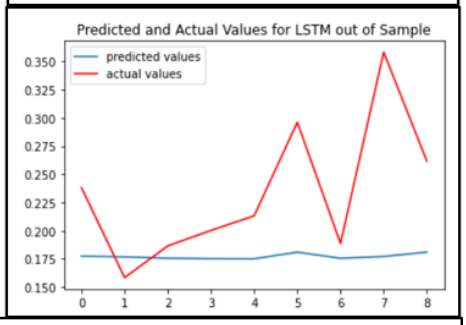


Figure 11: LSTM Results

Summary

Since we are interested in predicting sales data, we will determine the best model by using the out of sample RMSE. Thus, LSTM, simple exponential smoothing, SARIMAX2, SARIMAX1, and double exponential smoothing have RMSE of 15,066; 19,205; 26,212; 26,227; and 32,994, respectively. Since the median sales data for store 5 is 310,338, predictions being off of 15,000 means the predictions are fairly accurate.

It is interesting that for all four models the in-sample RMSE is larger than the out of sample. This suggests that the models did not overfit on the training data. Furthermore, it is interesting that the models switch ranks between in sample and of sample. The models with the highest (worst) error in sample have the lowest (best) error in out of sample, which is shown in table 1. The same can be said the for the 2nd worst in sample as the 2nd best out of sample and so on.

model	in_sample	rank_in_sample	out_sample	rank_out_sample
Simple exponential	41527	2	19205	4
double exponential	34196	5	32994	1
sarimax1	40068	3	26227	2
sarimax2	39759	4	26212	3
LSTM	44537	1	15066	5

Table 1: Model Summary
Larger the rank, the better the model for example rank 5 is better than 1

For future work, we would like to include additional variables into our SARIMAX and LSTM models. Additionally, we can look at structural breaks when testing for stationarity as well as modeling

other stores. Since choosing the model order from the ACF and PACF is subjective, we would like to utilize auto ARIMA functions to determine the model order, and see if it improves model performance.