Jarom Hatch

Final Report:
Retail Data Analysis

Problem Statement

Making decisions with and modeling retail data can often be challenging due to limited or changing history. As we've seen recently, the market can also be very volatile owing to things like supply chain demands, global conflicts, or product shortages. The given dataset contains information on sales, mark downs, holidays, stores, departments, temperature, fuel prices, and unemployment, among others. The sales data covers about two and a half years from February 5th, 2010 to October 26th, 2012. The features data cover this same time period, plus an extra 9 months to July 26th, 2013. The extra 9 months of features data will be critical to implement the model to predict sales for that time.

Using this data, I will implement different machine learning models, choosing one that has the best root mean square error (RMSE) score but also comparing other scoring metrics. Since this is a regression problem, I will specifically look at using linear regression, Lasso regression, Ridge regression, Random Forest regression, and ElasticNet to determine the best machine learning model for this data.

Data Wrangling

The data provided was initially spread out between three csv files called stores, features, and sales. The stores file contains the type and size of the 45 stores included in the history. The features dataset contained 12 columns and 8191 rows of data on the store number, date, temperature, fuel_price, markdowns 1 through 5, CPI (Consumer Price Index), unemployment rate, and is that date a holiday (True or False). Finally, the sales dataset contained the store number, department number, date, the weekly sales for that store and department, and the same holiday data as the features dataset.

In order to get the best results from machine learning algorithms, I needed to clean the data. First, there were numerous NaN values in the features dataset, mainly in the markdown columns. Since this was because no data was present before November 2011 for markdown values, I filled the NaN values with zeros. The remaining NaN values were in both the CPI and Unemployment columns, specifically the last few rows before the date would reset to the next store number. For the CPI values, since they followed a polynomial trend, I used a fitting model to project the CPI values for the missing dates. The unemployment rate had a constant value for every few weeks, so the simplest way to fill those rows was to use a forward fill method using the last data point before the NaN values. I also changed the categorical variable of store type to a continuous variable using an encoding and the dates to ordinal values to work better with the machine learning models. I also found that not every store had the same amount of

departments, so I decided to combine all the weekly sales for every department per store per date together in order to have a single sales value per store per date.

Finally, since the features dataset had extra dates for which I didn't have weekly sales data, I decided to split off those portions of the features dataset to be used with the machine learning model to predict the missing sales values. I then combined the relevant sales, stores, and features datasets to make one dataset with weekly sales data and one without. The dataset with sales data had 6435 rows and 16 columns and the dataset without sales data had 1755 rows and 15 columns.

Exploratory Data Analysis

It was important during this analysis to determine what kind of data was available to predict weekly sales. The data consisted of both continuous and categorical variables with different distributions. Below are the histograms of each feature in the retail sales dataset. The categorical variables are store, date, isholiday, and type (type_B, type_C). The continuous variables were temperature, fuel price, markdown 1 through 5, CPI, unemployment, weekly sales, and size of store. Looking at some of the trends for each feature, there is some preliminary information we can glean. Temperature and unemployment seem to have a semi-normal distribution, fuel price appears to have a bimodal distribution, size doesn't seem to have any appreciable distribution, and weekly sales and the markdown columns are mostly right skewed.

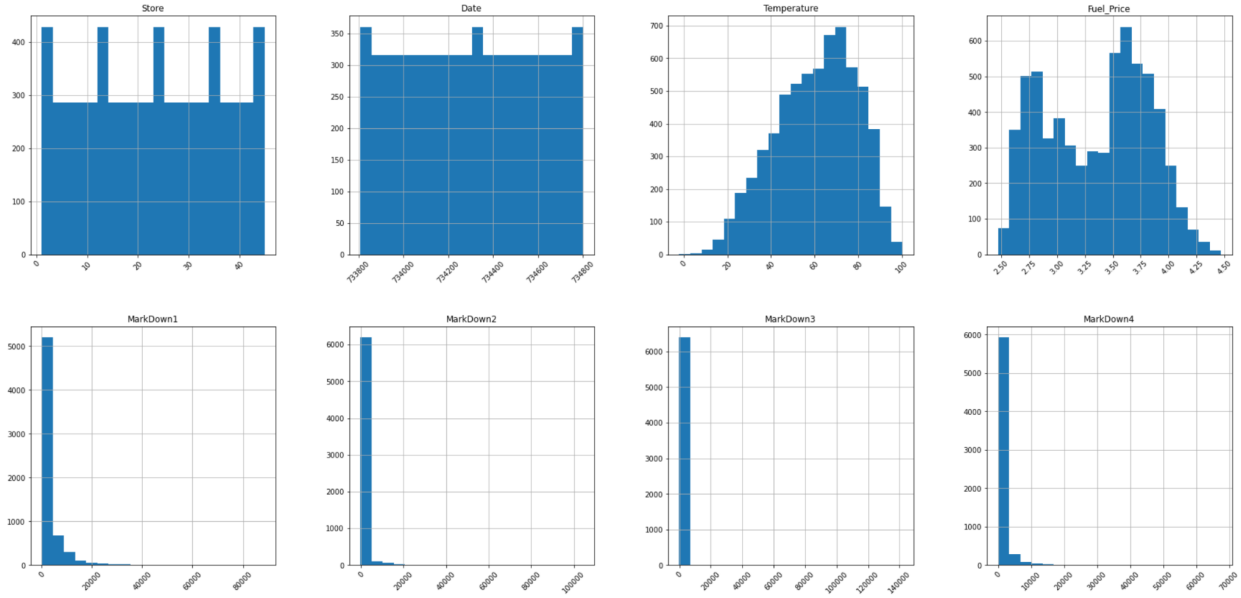

Figure 1A: Histograms of the first 8 features of the retail dataset

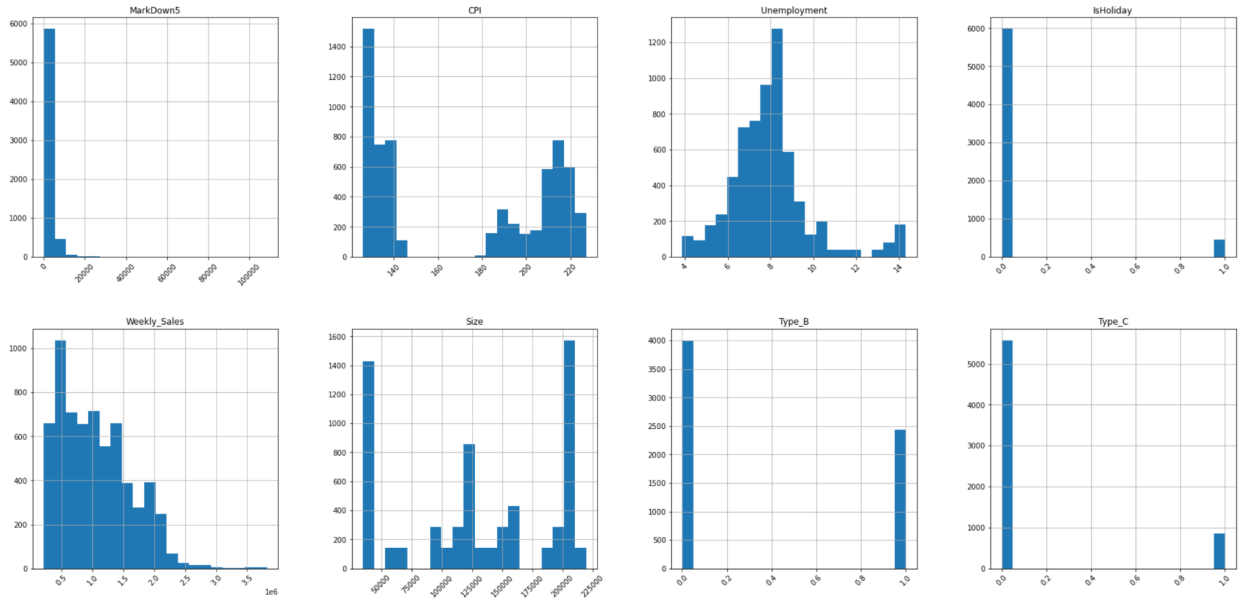

Figure 1B: Histograms of the last 8 features of the retail dataset

Next, I wanted to look into the correlation matrix of continuous variables to determine if there were any preliminary correlations between the different features. Figure 2 below shows the results of this analysis.

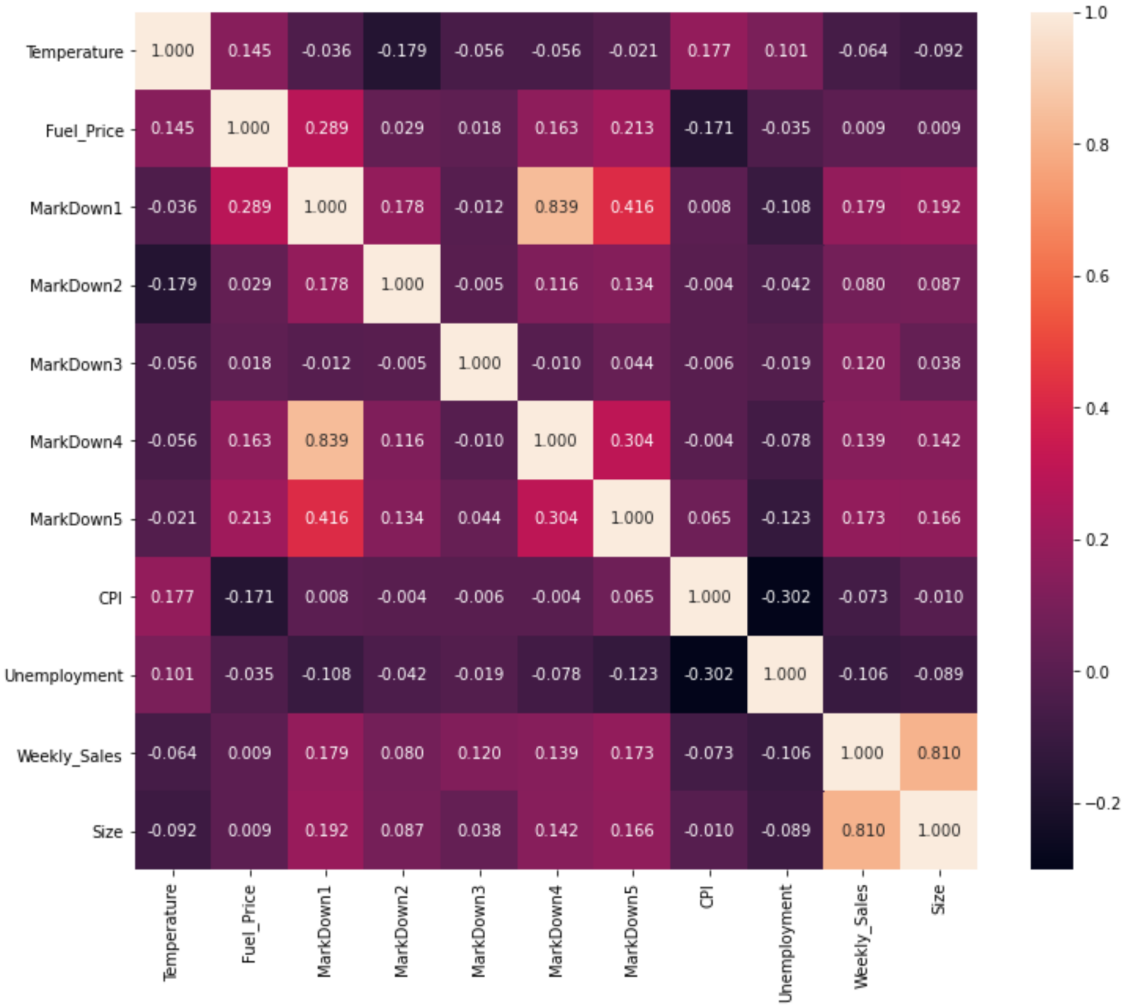

Figure 2: Correlation matrix heatplot of retail data continuous features

Since weekly sales are the target feature, I focused on the correlation between it and the other features. The strongest positive correlation to weekly sales was the store size with a value of 0.810. Fuel price also had a positive correlation with weekly sales, but was extremely close to 0, at 0.009, which indicates its lack of significance. All markdown columns have positive correlations with weekly sales, between 0.080 and 0.179. Although positive, they are very small and may be insignificant. CPI, unemployment, and temperature all had negative correlations with weekly sales, but like the markdown columns, these were also very small values. The largest was unemployment with a value of -0.106. Since they are so small, they could also be insignificant, although the CPI and unemployment being low would make sense for higher sales revenues since people have more purchasing power and more money from employment. Size being the largest contributing factor to sales may indicate that the larger the store, the more items they have available to sell, bringing in more revenue.

Modeling

Before modeling, the retail sales data was split into testing and training sections. I chose a split of 80% training and 20% testing along with a random state variable of 42 for reproducibility. Since some models work well with scaled data, I decided to use a few different types of scaling to determine if they do indeed affect the models' ability to predict sales. I ended up choosing the standard scaler and minimum maximum scaler, both part of the sklearn preprocessing library. The sales data for both the training and test sets was not scaled in accordance with the best practices when setting up machine learning models.

I decided to try out a few different models and iterations using linear regression, lasso regression, ridge regression, elastic net, and random forest regression. I chose these as they are the most common regression models when working with a continuous target variable. After running the standard linear regression on the scaled and unscaled data, the best model returned an RMSE of 301197 and an R^2 value of 0.718. I then plotted the test target values and the predicted target values to get a better idea of how well the model fit the data.

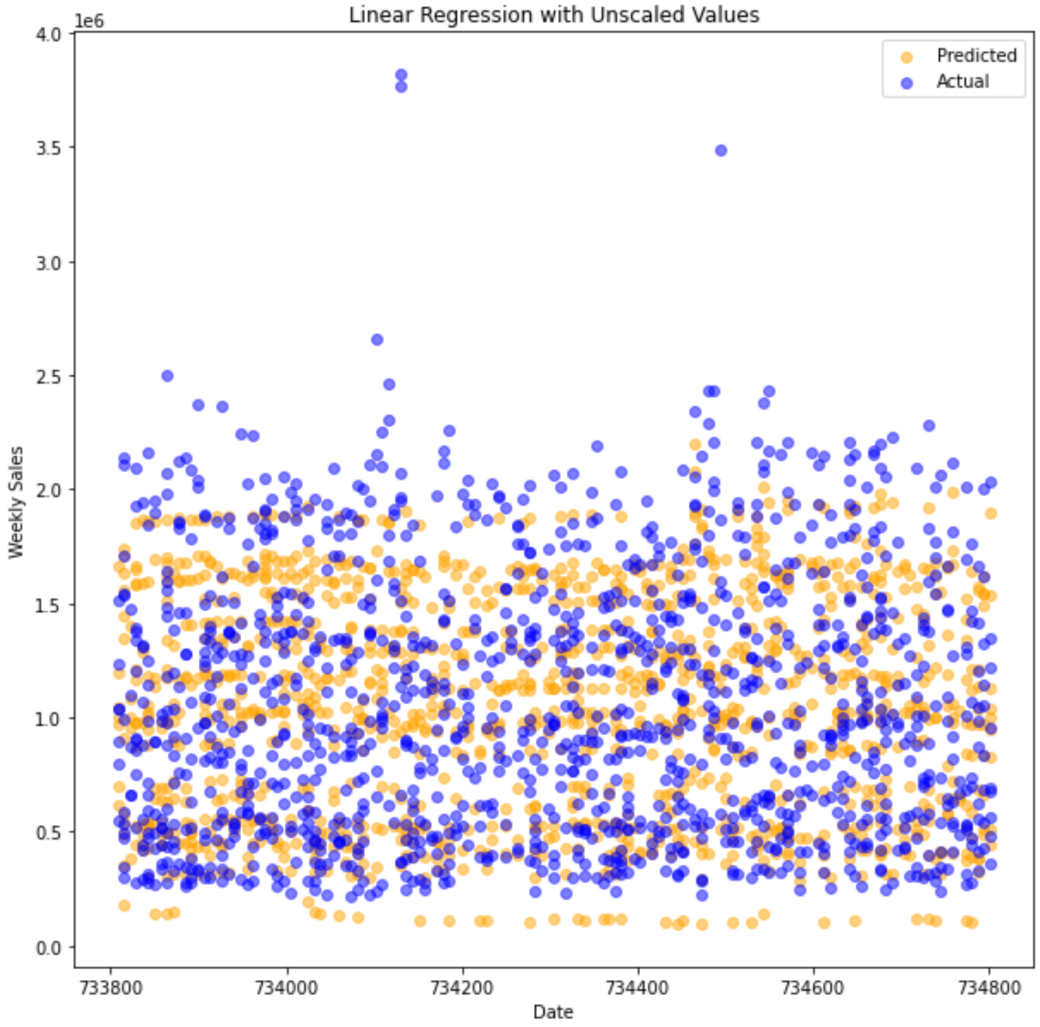

Figure 3: Linear regression plot with predicted and actual values

As can be seen in Figure 3, the predicted data fit the middle range fairly well, but not so much the lower and upper values. Running the lasso, ridge and elastic net models also had similar results to the linear regression models. The random forest model worked much better with the data and produced the results in Figure 4.

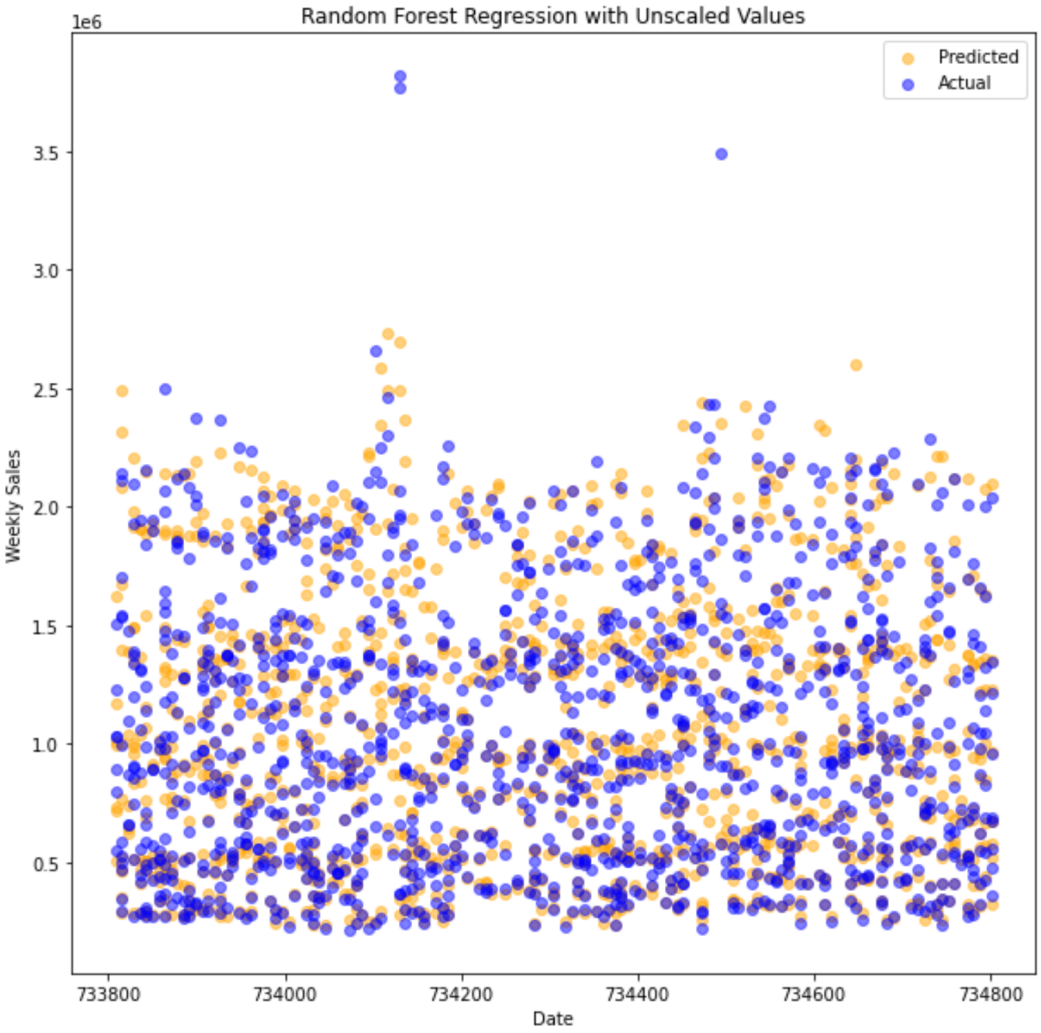

Figure 4: Random forest regression plot with predicted and actual values

The predicted values fit both the middle, lower, and upper ends of the test data much more accurately than the previous models. This model, based on the unscaled data, had an RMSE value of 134777 and an R^2 of 0.944, much better than any of the other models. I did want to make sure that the model was not overfitting the training data as the R^2 value on the training data was 0.993. I decided to run hyperparameter tuning on the random forest model to determine if I could improve the model and see if it was overfitting the data. Using the random search CV function, I found the best parameters for the model and these actually gave an RMSE of 172200 and a predicted R^2 of 0.908. This seemed to actually decrease the accuracy of the model. However, the training R^2 value is 0.916, which is much closer to the testing R^2 value than the model without hyperparameter tuning. Even though the metrics seemed close to the

untuned random forest model, it was still important to plot the data for a visual comparison. As seen in Figure 5, the tuned random forest model fit the data better than the linear, lasso, or ridge models, but had some glaring differences from the original random forest model. For one, there appeared to be bands of data where the values were more clumped together than the actual values. Since the RMSE and visual representations are worse, I will still be moving forward with the original random forest model with the unscaled data.

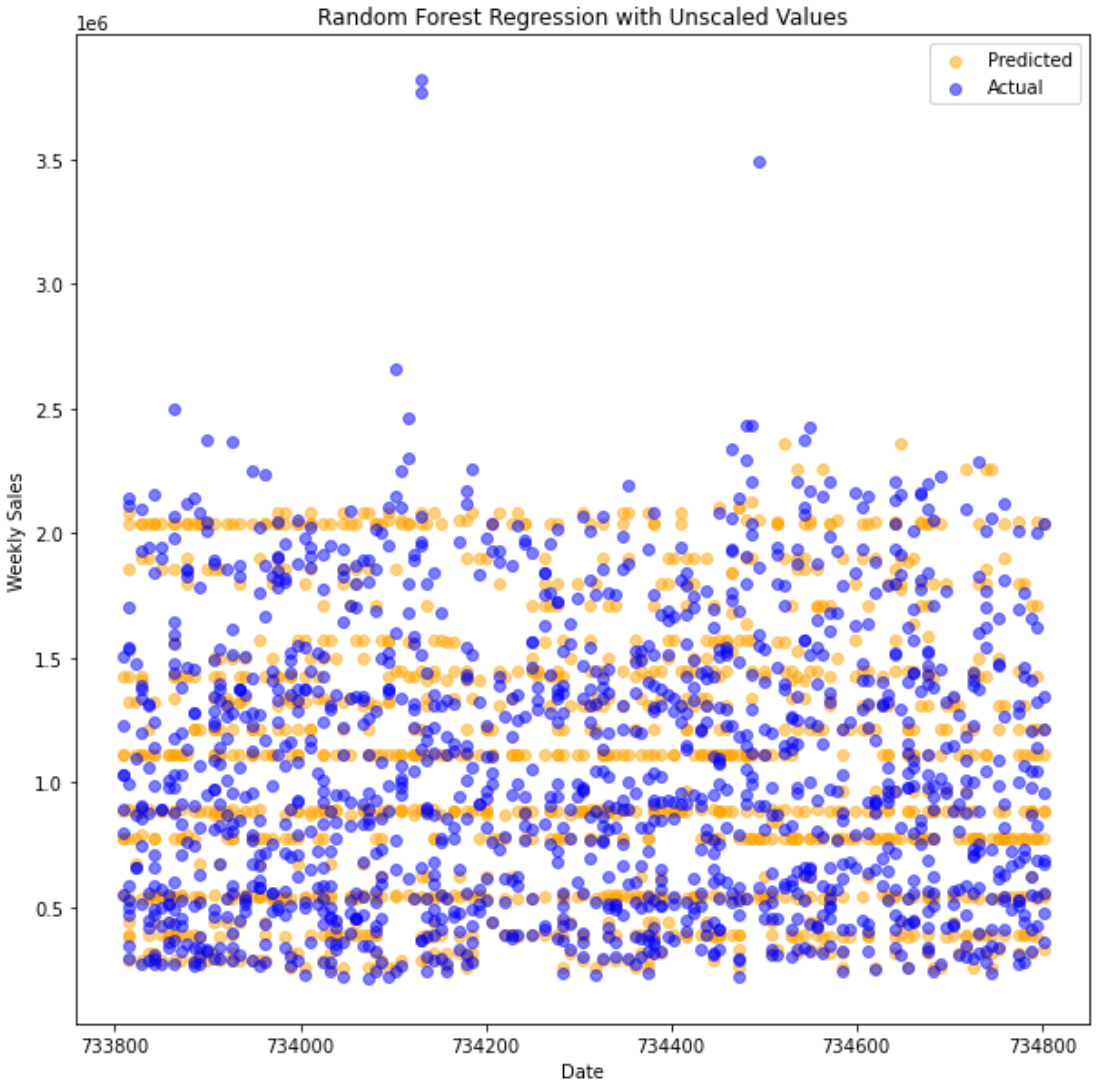

Figure 5: Tuned random forest regression plot with predicted and actual values

Predictions

Once I had chosen the random forest model as the best predictor of future sales data, I needed to then apply it to the provided dataset which had only future features but no sales data. I decided to use the normal random forest model I had first created as it had a better RMSE value and R^2 values for both the training and test data.

I first plotted the predicted future sales values along with the historical sales data to see how well the model handled the new features.

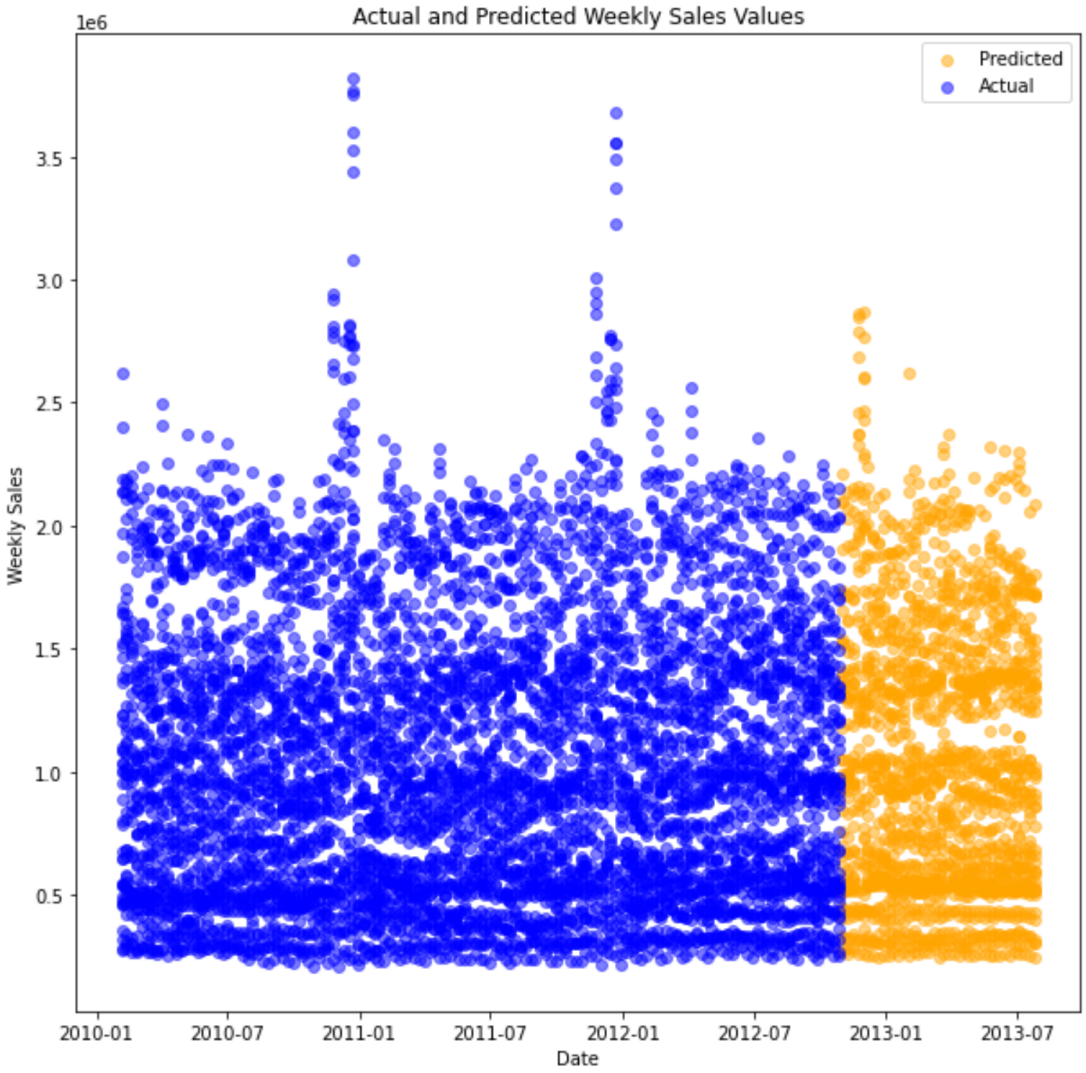

Figure 6: Actual historical sales data plotted with predicted future sales

As shown in Figure 6, the predicted sales values seem to match up very well with the historical data, especially when comparing similar date ranges to the predicted date range. There does seem to be a larger gap in the data around the $1.2 million dollar sales range which seems to

diverge from the historical sales trends. I also wanted to see what the model would predict for just one store's data and in Figure 7, we can see the historical trends and predictions for store 1.

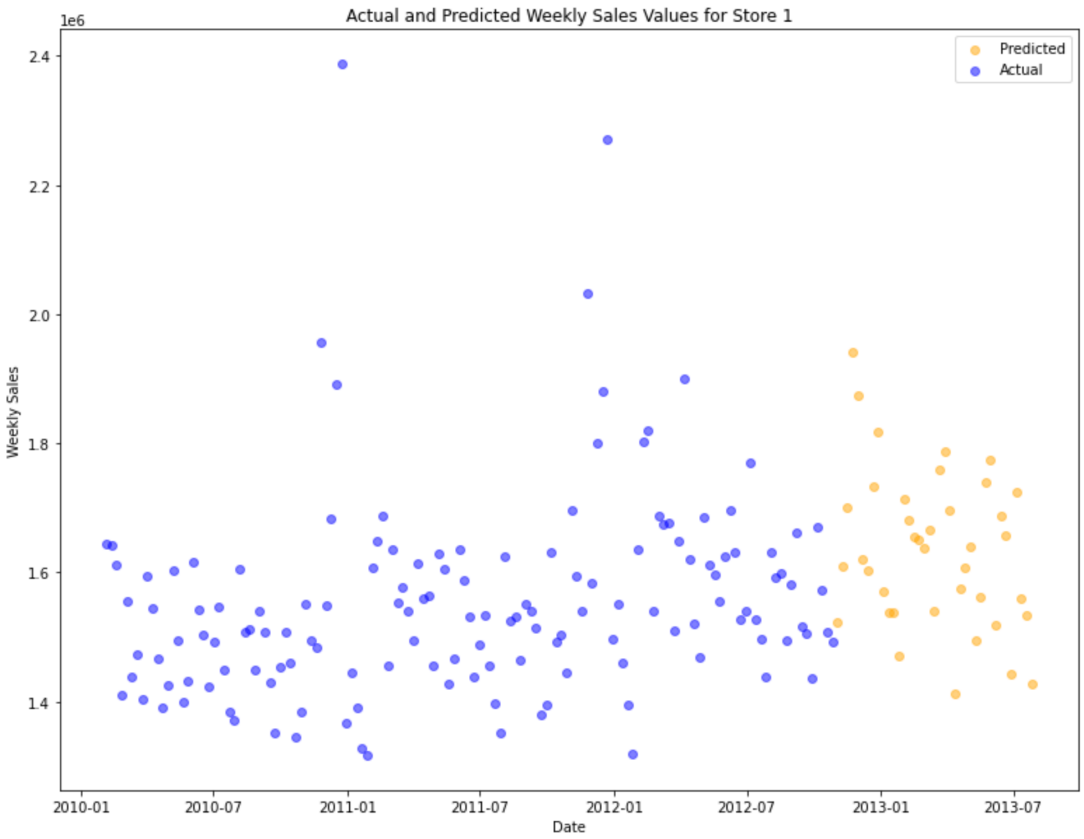

Figure 7: Actual historical sales data plotted with predicted future sales for store 1

It seems that for the most part, the model does an excellent job at predicting the future sales values from the provided features data. It does appear that the model tends to be more conservative, both in the maximum and minimum values it predicts, although the gradual trend in increasing sales over time is reflected in the model.

Summary

Being able to accurately predict sales values in the retail marketplace can be complicated and challenging, but not impossible. Machine learning techniques can be used effectively to process large amounts of data and produce accurate models for future predictions. In this process I first combined the weekly sales for each department per store and date. I then separated the historical data which included sales from the future data without sales information. After applying numerous machine learning models for predicting a continuous variable, like sales, I was able to determine that the best fitting model was a random forest regression using unscaled training and test values. After determining the best model, I could then feed it the future data and predict the weekly sales outcomes. These values closely matched the trends of the historical data, both when looking at all the data as a whole and when separating just store 1 for

closer analysis. I would feel confident in presenting these findings to the retail chain in question and giving them the trained model for any future needs they may foresee. Once the company gets more historical data on weekly sales, they can also retrain the model for more accurate predictions in the future as well.