Firecrackers, Fries, and Forests: A Predictive Model for a new Walmart’s Fourth of July Sales

Client: Walmart Inc.
Authors: Helen Borchart, Kevin Mittal, Sean Conway, Jeff Friedemann

Spring 2019
CSE 160 | Professor Davison
Lehigh University

Executive Summary

Bethlehem is located in what the Department of Agriculture calls a "food desert", a low-income census tract where many residents have low access to a supermarket or large grocery store. An Anthropology professor at Lehigh conducted a survey to determine Bethlehem residents' main concerns regarding access to nutritious food, and the majority cited distance to a grocery store as a major factor in their diet choices. In response to this issue, when Walmart came to Bethlehem city council with a proposal to make a new 100,000 ft^2 discount store in south side, the city embraced it. The store's opening was scheduled for the start of June.

As the planned opening draws nearer, Bethlehem Walmart reached out to our data science team with a concern. While the store was projected to handle June sales without an issue, it was unsure if it would have enough inventory to handle July. Seasonal items such as swimsuits, floaties, and hot dog/hamburger buns were of particular concern. The problem was twofold: purchasing too little inventory or the wrong inventory would drive away business, inhibiting customer loyalty, while ordering too much was impractical due to the store's inability to handle more than 30% excess inventory. Since no data exists on a Bethlehem Walmart, our data science team looked at data from similarly-sized American Walmart stores from February of 2010 to November of 2011 to get a rough idea of the sales by department that this Walmart can expect.

The resulting weekly sales regression model was based off of this same dataset retrieved from Kaggle. We reduced the more than 400,000-entry dataset to stores close to Bethlehem Walmart's size, removing superstores and smaller stores from the set. Second, we focused our dataset specifically on the month of July, as well as a few weeks at the end of June. We also reduced the scope of this project due to computational limitations, and decided to only analyze the top 25 departments with the most variable annual demand that spiked during the July 4th season. We used attributes, such as each store's weekly consumer price index (CPI), gas prices, unemployment rate, average temperature, markdowns, and the prior week's department sales to train our regressors.

This report leverages several data science techniques in order to create the most effective model including Support Vector Machines (SVM), Random Forests, and k-Nearest Neighbor (KNN). These each presented high R^2 values greater than 0.884, while the SVM model presented the highest value at 0.964. Finally, we used cross-validation to ensure our results were accurate, excluding one store from the 17 we chose to train the regressor, and used that store's data on the test set. This was done so that our model had to construct an entirely new sales projection for the tested store.

The Random Forest classifier was the best model out of the three, but this is subject to a few department/store outliers that the other models are not as affected by (as their estimates are more consistently over/underpredictions). However, we would ultimately recommend an ensemble classifier, composed of all three methods (KNN, Random Forest, SVM).

Our model will provide a useful tool to support the new Bethlehem Walmart and will ensure that it has a successful launch and first Fourth of July weekend. If we could improve the model, we would like to implement more time series mechanics (beyond just yesterday's sales figure), and potentially try other models (like Python's Extra Trees, or Neural Networks).

.

Technical Report

Introduction

As a data science team working as consultants for the new Bethlehem Walmart, our goal was to determine how effective prior sales data would be for predicting new sales data. Especially for new stores, it is tough to predict demand and order stock when historical weekly sales data for that particular store is unknown. Analyzing data from stores across the country similar in size to the new one, and in conjunction with location-based factors like gas prices, the CPI, and the unemployment rate to estimate the impact that location plays in the sales should create a model that would be useful for store managers anywhere to understand how much product to purchase and when to do so.

Ultimately, our goal is to provide the 100,000 square foot Bethlehem Walmart with a regression model that will accurately forecast the store's sales for the week of July 4th within a +/- 20% margin of error, 95% of the time. This forecast must be department-specific, and will be used for inventory planning purposes.

Data Understanding

All data was collected from kaggle.com, an online data science hub. It included information on 45 different Walmart stores from different US regions, and up to 98 different departments in each store. The dataset had 422,000 entries, and also included information on the size of each store. Each store was classified as either A, B, or C, where A was the largest and C the smallest. Also included was data on the five different types of price markdowns, as well as each store's weekly sales. Finally, information about the location of each store, including its consumer price index (CPI), gas prices, unemployment rate, and average temperature were included in another dataset. Finally, because each row described the weekly sales of a store, weeks with holidays were identified with the attribute "isHoliday".

Though many of these features do not need a further explanation, we will further define some of the variables we will be using.

- **CPI** - the consumer price index. This is "a measure of the average change overtime in the prices paid by consumers for a market basket of consumer goods and services." (bls.gov)
- **isHoliday** - whether the week contains a special holiday. For convenience, the four holidays fall within the following weeks in the dataset: **Super Bowl** (Mid-February), **Labor Day** (early September), **Thanksgiving**: (late November), **Christmas** (late December).
    a. **Note that July the 4th, Independence Day, is not included.**
- **Unemployment rate -** The unemployment rate is the percentage of unemployed workers in the workforce. By definition, unemployment includes only those without work, still looking.

Data Cleaning

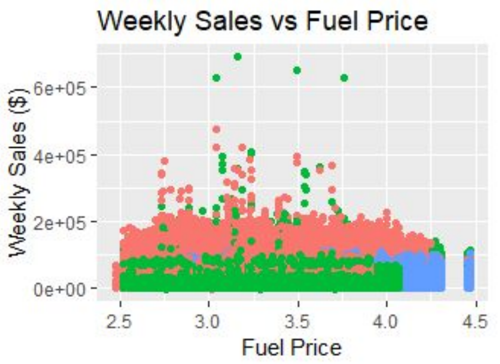
From our raw dataset, we have 421,570 entries based on 45 different Walmarts with 98 different departments. To best make the predictive model, our team performed data cleaning to ensure only the most relevant data would be used to train our regressor. In addition, we wanted to cut the massive dataset down so that training the regressor would not be too computationally expensive. We begin by importing the data with read.csv. We then merged the "stores" dataset with the "features" dataset. By merging, we will be able to examine the specific features of each department of each store for each week.

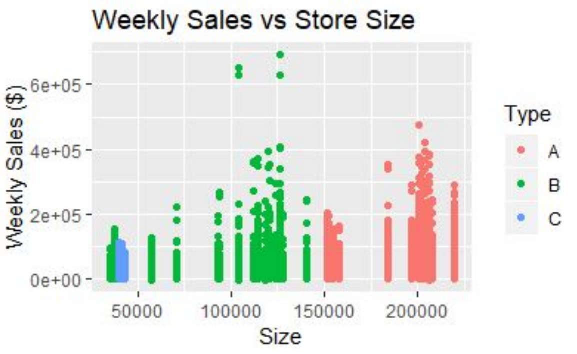
Data Filter

We will be using 14 out of the 16 columns, as we will not include the “IsHoliday”or “Date” attributes. Because the Fourth of July is not classified as a holiday in “IsHoliday,” and we only need to look at the sales for late June and early July, neither were essential.

Data Exploration

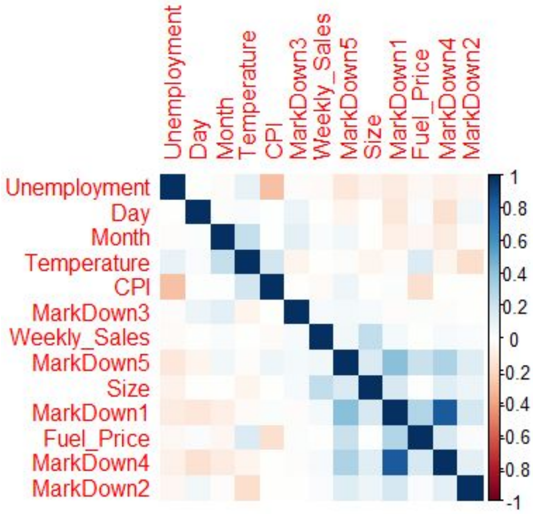
We created scatterplots of each of the store’s attributes in relation to their weekly sales. A few examples are shown below, though the store size plot is especially important:





From the rightmost plot, we deduced that there were 3 types of stores: A (Superstores), B (Standard Walmart), and C (Mini-Walmart), in order of largest to smallest. We filtered out the Superstores & Mini-Walmarts since the proposed Bethlehem Walmart would be of standard size (~100,000 sq ft).

We also investigated how our attributes were correlated with each other. Below, we produced a correlogram, which indicates whether attributes are positively or negatively correlated, and the degree of that correlation. This is measured from -1 for perfect negative correlation, to 1 for perfect positive correlation. Markdown 1 and Markdown 4 showed the most significant positive correlation of the data given, at approximately 0.8. These markdowns may be on products with similar seasonality, or targeting similar consumers.



Missing Data

In order to maintain data integrity, we found any "NA" rows in the markdown columns and replaced them with 0's. After that, we searched for blank values, but all other cells were filled.

Data Engineering

Despite all of the data cleaning that we performed, R Studio continued to crash. We also did not have enough information about the time series nature of the sales predictions as we only were using the attributes initially given in the dataset. To remedy this, we added a column to the data that included the sales of that department in the previous week, giving the model some prior information. We also reduced the scope of this analysis to 17 Type "B" Walmarts stores and 25 of the most highly variable departments around July 4th. We were thus able to reduce the dataset from 400,000+ observations to around 5,000. We had an acceptable computation time of about 5 minutes per model training & testing.

Model Selection

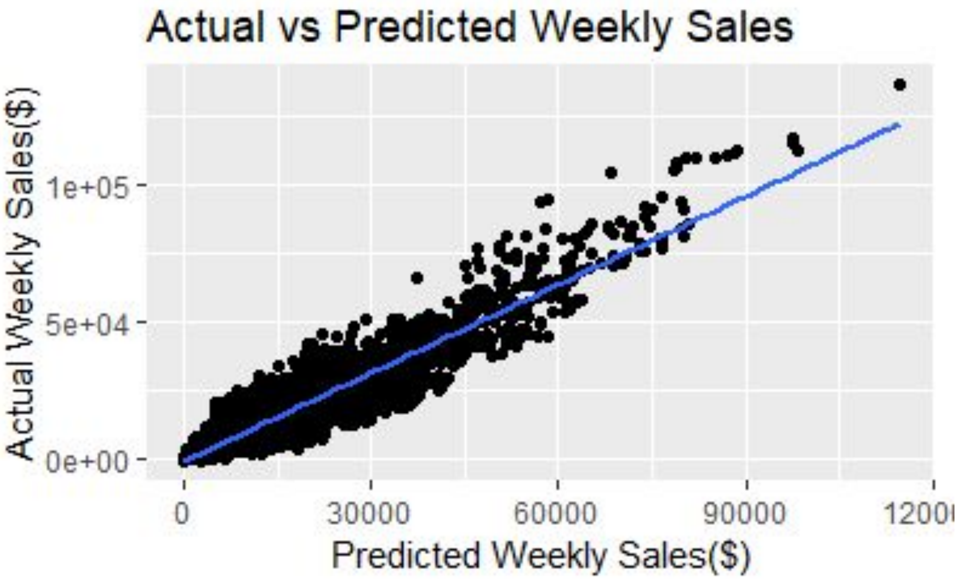
In order to determine which model should be used, we ran several regression methods through K- Fold Cross Validation. The three models are: KNN Regression, Random Forest Regression, and Support Vector Machines. We also created a function that calculated R^2 to measure the error of each model. Although we attempted running Linear Regression on the data, we found it resulted in a poor R^2 value of ~3%, so we did not include this model in our Technical Report in much depth. The likely root of this poor performance is too many attributes.

Model Preparation

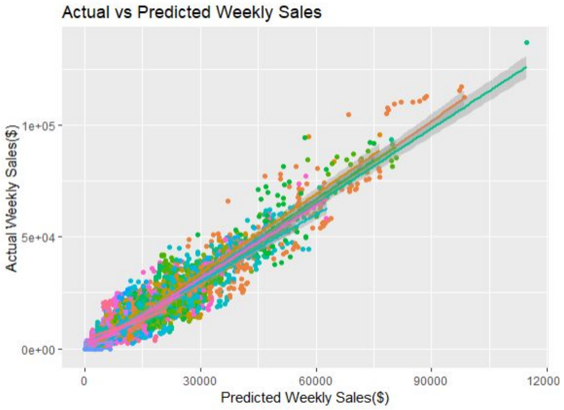
We then ran the training & test data through the K- Fold Cross Validation by creating folds by store number. Therefore, when training data was run, the classifier needed to make a sales forecast for all departments over one month for one store, as opposed to making forecasts for different periods in time for various stores. We looked at the number of unique stores in our dataset (17), and created a fold for each store. In the first fold, we eliminated the first store in the data set. We iterated through every remaining store and then used each as the test set. During each iteration, the stores not selected formed the training set.

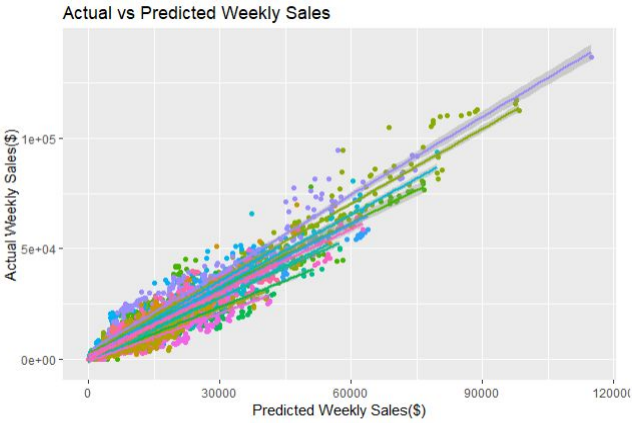
K Nearest Neighbors (KNN)

K Nearest Neighbors (KNN) is an algorithm that can be used for both classification and regression. This algorithm works using 'feature similarity" to predict the values of new data points, depending on how closely it resembles the data points from the training data set. First, the distance between the training point and the new point must be found. Then, a "k" value is used to determine how many neighbors to reference when deciding on a value for a new point. We ran the regressor with the default k = 7. This regression provided us with the lowest R^2 value, .884. Of our three models, though KNN has the lowest R^2 value, it was actually the best predictor for department sales.



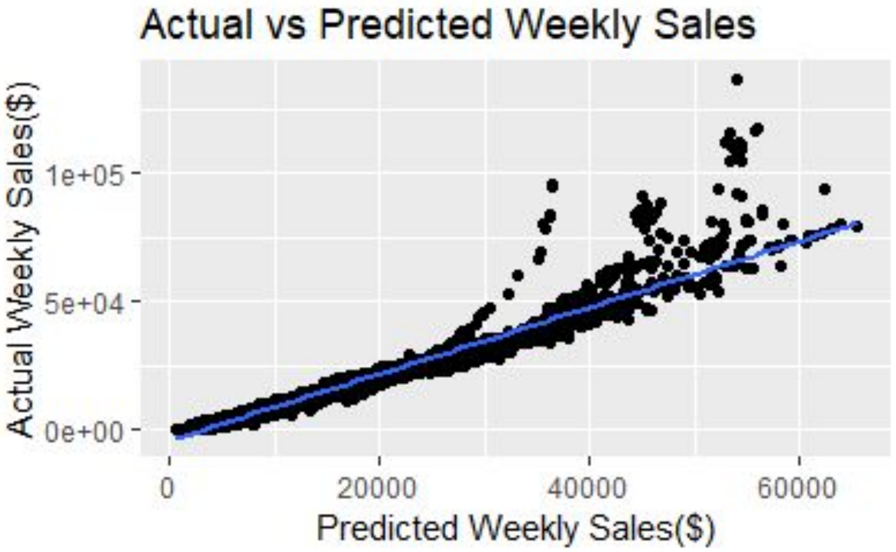
Below on the left, the KNN regressor's predictions are graphed against the actual sales of each department across all stores. On the right, we see the opposite: all predicted weekly store sales are plotted against the actual weekly sales. Evidently KNN works very well when predicting departmental sales, there is little deviance from the actual values, while KNN performs poorly when predicting aggregate store sales. This is likely because of the "grouping" nature of the KNN classifier as the classifier is very good at creating factions, but not aggregation.



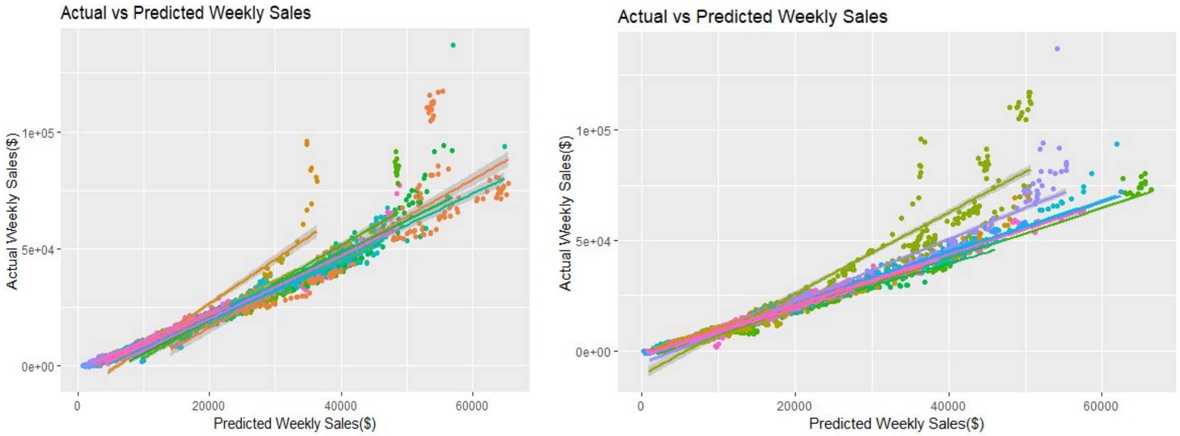


Random Forest Regression

Random Forest Regression is a machine learning model used for predictive analysis. It works through model ensembling, a broad technique that involves multiple predictive models and aggregating the results from each one in order to reduce variance. The trees "vote" on the most likely output. An advantage of using Random Forest is that it is able to showcase the non-linear interaction between the feature variables and the target variables. Using 50 decision trees per run in generating the model, the R^2 value was approximately .926. Running the regression on 150 decision trees did not make a difference in the results - the R^2 model remained around .926.

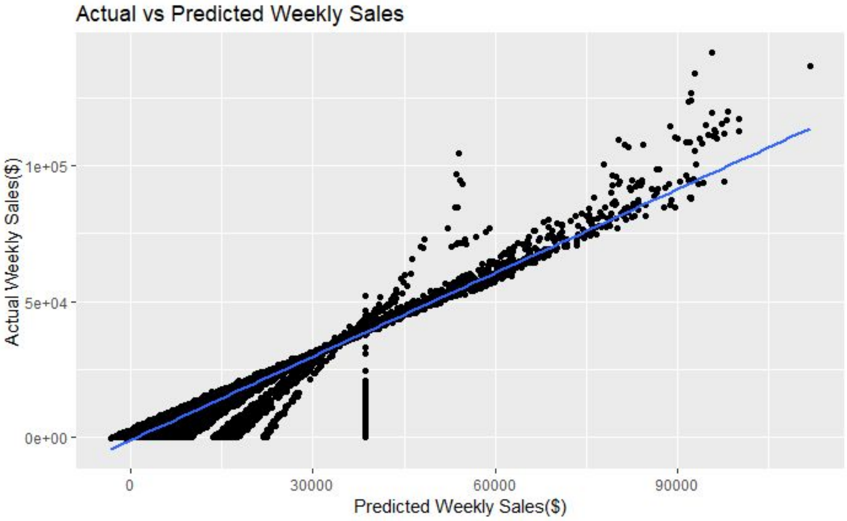


Departmental and store sales are pictured on the left and right, respectively. Overall, the Random Forest model performed well in both instances, though there are outliers in each case. This means that though it will accurately forecast the stores/departments most of the time, though there are a few unusual/edge cases that the Random Forest model does not perform well for. Additionally, when the model does significantly deviate, it tends to severely underpredict actual sales. This would not be optimal for inventory management, as it could create a shortage of something many customers need. We would need more information about what these outlier departments sell to make a decision.

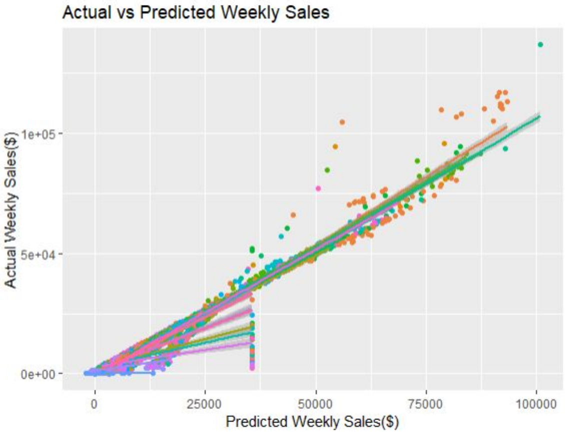


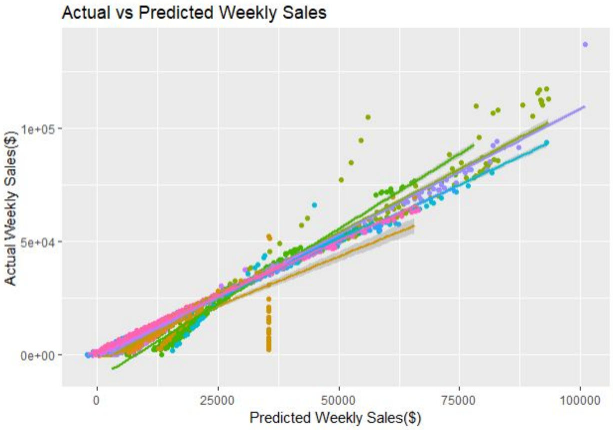
Support Vector Machines

A Support Vector Machine (SVM) is a model formally defined by a separating hyperplane. In other words, given training data, the algorithm will produce output that is an optimal hyperplane. This hyperplane then categorizes the new examples. In a 2D space it looks like a line dividing a plane in two parts where in each class lay in either side. The support vector machine model used on the Walmart data resulted in a very accurate regression with an R^2 value of 0.964. In general it tended to overpredict at the lower range, while its outliers on the higher range had much smaller variances, as can be seen below. Overall performance was very good, though not as effective as our final chosen method, especially because the model was unable to accurately predict the sales of individual departments.



In terms of departmental and store-specific performance, SVM performed very well when aggregating store sales, as seen on the right. However, it performed very poorly when estimating department sales, as seen on the left. Though some departments had fairly accurate predictions, there were 5-6 departments that SVM significantly overestimated. This is not optimal for inventory management, because overestimating demand could be problematic for items that are only sellable for a limited period of time, like produce. Additionally, this could create a storage problem because of limited shelf and square footage.





Analysis

There were several methods we used to analyze the performance of our models. The first was looking at prediction for aggregated weekly sales from each individual store. In this case, KNN tended to overpredict, especially at the larger size stores, while the Random Forest and SVM models tended to be more balanced, though the Random Forest had a few outliers it underpredicted. In terms of weekly sales by department, KNN was very consistent with no outliers, Random Forest was balanced, but had more outliers than KNN, while the SVM model had a higher tendency to overpredict and resulted in large variances between the predicted and actual results.

As far as metrics to indicate success, KNN had an R^2 value of 0.884, Random Forest a value of 0.926 , and SVM a value of 0.964. This explains 66%, 73%, and 81% of the standard deviation, respectively. Overall, the Random Forest was the best model due to its high R^2 value and strong performance in predicting sales for both individual departments and stores. It did however have more than a few number of outliers, for both stores and departments with irregular characteristics, and in those cases, predicted sales significantly underestimated actual sales. It is because of this that our final recommendation is for an ensemble regressor with all 3 methods to result in the most accurate performance over a variety of situations and needs.

Conclusion

Given that weekly sales can be subject to large variances due to unexpected significant events, like an upcoming natural disaster, severe weather conditions, and quickly changing trends, it is important to note that these models should be used a conservative predictor. This would allow flexibility in accounting for slightly over or under results, which would yield the best profit margins for Walmart stores.

Overall however, our results are clear that all of the models were effective in consistently predicting weekly sales. Therefore, we would recommend an ensemble method of SVM, KNN and Random Forest. It should be noted that the SVM model did not effectively predict individual department sales (due to underestimation), though it did do well in predicting aggregate store sales. On the contrary, KNN performed well in terms of departmental sales, but consistently overestimated aggregate sales.

Random forest resulted in a relatively accurate and overall solid model. The model produced was accurate when predicting weekly sales for stores and departments used in the test set, as can be seen by the included R^2 values above. However it is important to note that a high R^2 values does not exclusively determine the best model, as the model is plagued with predictive outliers that it underestimated. If we had to choose one model, we would select Random Forest, provided that we had more information about the outlier departments.

In terms of project improvements, we believe that the model would be more dynamic if we could add larger time-series mechanics to it. For our purposes, we only used one week's sales, but in reality, if the last week used had abnormally high or low sales (i.e. Black Friday), it could have a significant impact on the reliability of the model. Adding a cumulative moving average of 12 or so of the past weeks would bolster the model's performance as a whole.

Ideally, we would have included all departments and stores in the analysis, but because of our computational limitations, we were unable to do so. This would be particularly important for our random forest classifier, as we could see whether more outliers would occur, or if the deviation for stores/departments was just a singular unusual instance.

Lastly, our project would be improved if we were able to try other regressors, including neural networks, or Python's extra trees regressor. The extra trees regressor would be particularly relevant to this project, as it may improve upon the performance of the random forest classifier that we used by accounting for more edge cases. Neural networks, on the other hand, use a weight-based network of nodes that are continually adjusted to improve the model's output. Though these are typically used for classification rather than regression, we believe that due to their high usage in recent years and success within marketing and other domains that they could be useful.