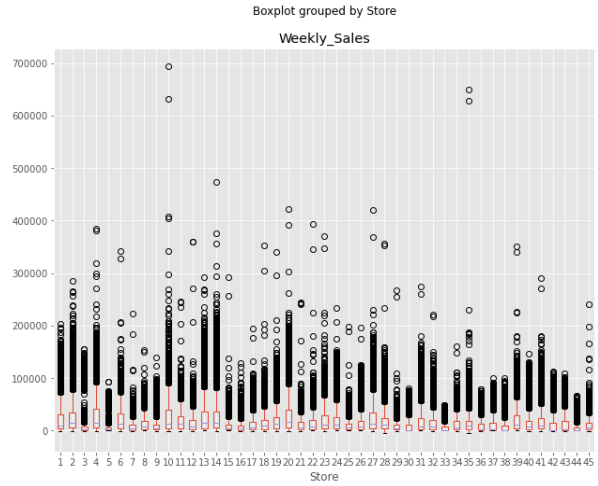
Project 2: Walmart Store Sales Forecasting

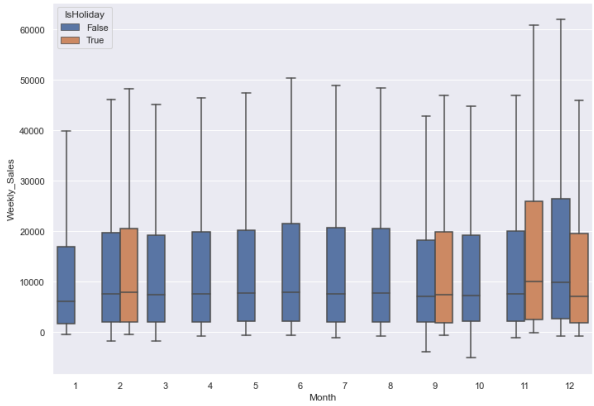
Team Members: Sarah Yang (sarahxy2), Ryan Berry (rpberry2)

November 14, 2021

Introduction

In this project we predicted the weekly sales of 45 different walmart store. The data set used to train and test our model contained the actual weekly sales from February, 2010 through October, 2012. The data set contained the columns - store#, department, date of the week start, the weekly sales in dollars, and if the week contained a holiday or not. As we were exploring our data we found high disparities in store sales between each of the stores as well as between each of the departments. While we also found that there is some correlation between holidays and weekly sales, the most important data points proved to be the department, store, week and year. Our goal was to create a signal representing weekly sales across the weeks of a year that can be applied to future years regardless of store and department.





Data Files

Our raw data sets contained training data in folds. All data sets contained the columns "Store", "Dept", "Date", "IsHoliday". We were given 10 files which represent our 10 folds and an initial training data set that was considered as our initial fold that we trained our data on which contained weekly sales data data for stores and departments from 02/2010-02/2011. We used this initial data set to predict weekly sales for 03/2011-04/2011. We then used our first fold that contained weekly sales data for 03/2011-04/2011, to predict the weekly sales for the next two months of 05/2011-06/2011, and so on till we predicted through the months of 09/2012-10/2012. We were also given a test data set with the same columns, that we used to test and calculate the performance of the model.

Preprocessing

Since the original dataframe has large dimensions (data span 52 weeks) and a lot of NA values, we need to preprocess the data by dealing with these NA values and reduce dimensionality to get a design matrix for a general predictive model. Thus, we chose to use Singular Value Decomposition (SVD) to reduce dimensionality, reduce noise, and also fill in NA values with zero as suggested by Professor Liang.

To perform SVD, we extracted the data and organized it according to each department, then filled in the NA values with zeros. We chose our number of principal components to keep as , and if rows for a particular department is larger than that, we will subtract row mean from each row and compute SVD on the data matrix for that department. Since there were 99 departments, we computed SVD on 99 data matrices corresponding to each department. Then after vertically concatenating the data matrices and adding back the original row means we subtracted from each row, we get a reduced rank version of the original training data. This approach filters out the random noise innate to the smaller singular value components as we just keep the top 8 main principal components, resulting in a smoothed version of the original data. We chose 8 as our hyperparamter to use for SVD after experimenting, as 8 gave us the lowest WMAE score of 1583.

Model Description

We used the results of our SVD computed on each department to fit into a simple linear regression model. The response variable, "Weekly Sales", is modeled using the features "week" and "year." Week is treated as a categorical variable, while year is a numerical variable.

Model Performance Methodology

We used the weighted mean absolute error (WMAE) as a metric to calculate the performance of our model. How we did this entailed looping through the 10 fold we have and running predictions on our test data and the respective fold file containing the actual data to compare our predicted values with. In each fold we calculated the weighted absolute error(WAE) by setting weights of 5 or 1 for rows that are holidays or not, and then summing up the weights multiplied by the absolute value of the actuals subtracted by the predicted values, and divided by the sum of weights. After we got the WAE at each fold, we calculated the mean of the WAE to produce our WMAE value.

We also added a shift trick (original idea comes from the winner of the kaggle competition) to lighten the error associated with fold 5. Since fold 5 includes more holidays than most folds (Thanksgiving and Christmas), the results are weighted higher and are therefore more important to our overall WMAE. Since the sales bulges associated with Christmas mostly happen on the week before (week 51) Christmas week (week 52), we implemented a post-prediction correction. To week 51, we assign 6/7ths of week 51's original prediction plus 1/7th of week 52's prediction.

Results and Runtime

Fold	WMAE	Time (s)
1	1942	9.95
2	1363	9.95
3	1382	10.5
4	1527	12.6
5	2057	19.1
6	1636	13.2
7	1683	13.1
8	1400	14.4
9	1418	13.8
10	1426	14.7
Mean	**1583**	**13.1**
Total	**N/A**	**131.3**

The results and runtime data were gathered from a MacBook Pro 2.6 GHz 4-core Intel Core i7 with 16 GB of RAM.

Conclusion

This assignment taught us to visualize and clean data before applying a model to make predictions. We learned that data, even with primarily categorical features, could be translated into a signal which represents a greater pattern. SVD helped us to reduce the noise/outliers associated with that signal, and simple linear regression allowed us to interpolate new predictions.

Contributions

Ryan worked on exploratory preprocessing and end-to-end testing of the code.
Sarah worked on the SVD method, preprocessing steps, and integrating the final code.
Both of us worked on the performance measurements as well as the visualizations.