Project 2 Report

Paul Holaway (paulch2), Albert Li (xiangl9), & Matthew Schroeder (mas5)

November 3rd, 2022

Statement of Contribution

All group members have contributed to this project. The coding and implementation for this project was done by Matt. Results were summarized by Albert and Paul for each prediction model attempted by Matt. The final report was written by Albert and Paul. Matt also proofread and edited the report before being submitted.

Overview

The goal of the this project was to predict the sales of different departments for forty-five different Walmart stores from February, 2010 through October, 2012. A unique setup was used, with five times the weight was put on holiday weeks when compared with non-holiday ones. Specifically, the holidays were defined as weeks which included the Super Bowl (Week 6), Labor Day (Week 36), Thanksgiving (Week 47), and Christmas (Week 52). We utilized a two-month-ten-split dynamic mechanism when training the model. The initial training data contains data ranging from February, 2010 to February, 2011, while the initial test data contains March, 2010 to May, 2010. Predictions are made on a two-month block. After a set of predictions are made and compared against a two-month test data set, we then calculate the error. The test data is then incorporated into the updated training data. We then use the updated training data to train and make new predictions for the subsequent two months. This process was repeated ten times, for a total of twenty months worth of predictions. The final accuracy of the model is calculated by WMAE (weighted mean average error) as defined below.

$$\text{WMAE} = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} w_i |y_i - \hat{y}_i|$$

w_i represents the week number weight, which is 5 for week 6, 36, 47, and 52 and 1 for the remaining weeks in the year. y_i represents the actual sale total while $\hat{y}_i$ represents the predicted sale total. Even though the original data is a time series data set, we decided to use linear models with appropriate modifications rather than fitting time series models (SARIMA for example), given there is no restrictions on the model we could use. The ultimate goals is to achieve a WMAE lower than 1580, which is achieved by the final model with the **WMAE=1571.709**.

Data Preprocessing

Dataset Introduction

The initial training data set (`train_ini.csv`) contains 421,570 records, with 5 columns.

- `Store`: The number assigned to the store.

- `Dept`: The number assigned to a department in the stores.
- `Date`: The day, month, and year when the sale total was recorded.
- `Weekly_Sales`: The total weekly sales in U.S. Dollars.
- `IsHoliday`: TRUE or FALSE; Whether the week contained one of the holidays.

The range of the data was from February 2nd, 2010 to February 25th, 2011. The initial testing data set (`test.csv`) contains 257,455 with 4 columns (`Store`, `Dept`, `Date`, and `IsHoliday`). The range of the data was from March 4th, 2011 to October 26th, 2012. There are no missing values for both data sets.

Initial Data Setup and Transformation

- `Wk`: We added Week column into all training and test data set by using the week function from `lubridate` package. This allows us to use paired weeks for training and testing. We later converted this into a 52-level factor.
  - **Note**: In the `lubricate` package, there are 53 weeks in 2010 while 52 weeks for others in the data set. Thus, to align the week, we need to minus one for differences among different years while the holiday weeks stay fixed. This is noted and reflected in `mymain.R` lines 16-22.
- `test_current`: Initially contains the date from March 1st, 2011 to May 1st, 2011, which would be updated with later test with two-month block each time. This is the initial testing data set for each fold iteration.
- `tmp_train`: Contains data in 2010 but with ten days earlier and fifteen days later to avoid possible rounding errors caused by week calculation.
- `test_dept` & `test_stores`: Finding unique departments and stores in both testing and training data sets, then getting the intersection between the two to make sure no missing or not-included parameters when running the testing data set on the model built on the training data set.

Modeling Approaches

- Note: All modeling approaches and techniques used for the first time in the modeling process are carried over to all models listed afterwards in this report.

Initial Naive Approach

The first modeling approach that we tried was to simply take the mean of the previous week to predict the future week. We filled in any missing values with 0 as the final tester would do at first, then tried to use the most recent week that has input to replace the missing value if possible. Otherwise, we will use 0 again. The model returned a WMAE over 1900 so it is clear we need to try something fundamentally different.

Linear Regression with Paired Week in Previous Years

The next approach that we tried was a linear regression with the weeks and the year as the independent variables. This leads to substantially less prediction error than the naive approach. With this method, we got our WMAE to 1660.

Square Root of the Absolute Value of the Sales

We noticed there are few negative values in the `week_sale` columns, thus to use square root transformation to normalize the skewed data, we made square root of the absolute value of the response variable. This approach gave improved the WMAE to 1622.

Quadratic Year

To reduce the WMAE, we also tried to increase the power of predictors. Rather than linear regression, we tried quadratic regression on year terms. This approach only reduced the last three folds and the overall WMAE did not change significantly. Based on the previous improvement on the square root of the absolute value of the response variable, we decided to merge these two approaches and made our fourth model. With the quadratic year term plus the square root of the absolute value of the response variable, we were able to reduce the WMAE to 1601.

$\frac{1}{7}$ Shift

To further reduce our WMAE, we decided to implement shifts as post-prediction adjustment to the data (based upon the Kaggle competition winner's strategy). The initial attempt to shift failed, as it returned worse performance for fold 5 and the same for other folds. After a closer look, the our implementation of the winner's shift did not multiply the mean of week 48 to 52 by 1.1. We used $\frac{2}{7}$ shift instead of a multi-level shift based on year (see Campuswire post #964). The results are better, but the WMAE was only 1596. When changing the $\frac{2}{7}$ shift to a $\frac{1}{7}$ shift, the WMAE decreased slightly to 1572. This would then be our final model as we got under the maximum benchmark for this project of 1580.

Table 1: WMAE Using The Final Model

	Fold #1	Fold #2	Fold #3	Fold #4	Fold #5	Fold #6	Fold #7	Fold #8	Fold #9	Fold #10	Average
WMAE	1980.675	1450.024	1423.152	1546.65	1981.607	1631.344	1680.139	1347.62	1348.592	1327.288	**1571.709**

Computer Specifications and Run Time

- MacBook Pro
  - Retina, 13", Early 2015
  - Processor: 2.9 GHz Dual-Core Intel Core i5
  - Memory: 8 GB 1867 MHz DDR3
- Run Time: 23 minutes, 3.5 seconds

Conclusion

For the predictions of the sales for Walmart departments we started with a basic linear regression model that included the week and year of the sales. While it did not do a terrible job, it did not do a great job either. Due to the skewness of the data, we attempted to eliminate it using a transformation. The transformation $\sqrt{|y_i|}$ was the best for eliminating most of the skewness in the data and achieved a better fit. We then added a quadratic term to further increase the accuracy of the fit because even after the transformation, the data still had some skewness and showed some sign of a non-linear relationship. When combining the transformation with a quadratic term for the year, this lead to more significant improvement. So far we had managed to use techniques taught in a basic linear regression course and got the WMAE to 1601. To get the WMAE under the criteria of 1580, we had to use a more advanced technique. A post-prediction adjustment shift was added to adjust for weeks that had significantly larger sales (10% higher was the cutoff we used). The shift we originally used was $\frac{2}{7}$ (shifting 2 days of average sales from one week to the next). This however did not make a significant enough difference in our prediction. We then tried a shift of $\frac{1}{7}$ (shifting 1 day of average sales from one week to the next), and that resulted in a satisfactory decrease in WMAE to under 1580.