Jake Goodman (Net ID: jakeg5) & Michael McClanahan (Net ID: mjm31)

STAT-452: Practical Statistical Learning (Online MCS-DS | Fall 2021)

Project 2: Walmart Store Sales Forecasting

**1. Team Members and Contributions**

The team for this project consisted of two members: Jake Goodman (Net ID: jakeg5) and Michael McClanahan (Net ID: mjm31). Each member contributed equally to the project. Both members performed and equal amount of research to develop the code necessary for the *mymain.R* file. Both members also contributed equally to the project report. Michael created a first draft of the introduction, methods, and results. Jake refined and organized the content for the final draft and added team member contributions, a discussion, and a table of results.

**2. Introduction**

*2.1 Objective*

The goal of this project was to develop and test a linear regression model capable of predicting future weekly sales for all departments of 45 Walmart stores located in different regions using their historical sales data [1]. Performance was evaluated using a weighted mean absolute error (WMAE) for the test dataset for each of ten folds of training and test data. Each fold’s test data was a two-month period of weekly sales totals by store and department. Each fold’s train data consisted of weekly sales by store and department from February 2010 up to the first day in the test data period. Mean absolute error was weighted to be five times higher during weeks in the test data that contained the following holidays: Super Bowl, Labor Day, Thanksgiving, Christmas, prior to which Walmart runs their largest promotional markdown events. Performance was deemed sufficient if average WMAE across all folds was less than 1610.

*2.2 Dataset Description [1]*

The Walmart Store Sales Forecasting data set was obtained from Kaggle, where it was originally posted by Walmart for a prediction competition to recruit new data scientists. The dataset contains historical weekly sales totals (in US dollars) from February 10th, 2010 to November 1st, 2012 for all departments of 45 Walmart stores located in different regions. The data set contained the following fields: 'Store': a de-identified numeric identifier for the store, 'Dept': a de-identified numeric identifier for the department (same across for the same kind of department), 'Date ': the first day of the week, 'Weekly\_Sales ': sales in US Dollars for the week for the department in the store, 'IsHoliday ': a binary indicator for whether the week is one of four special holiday weeks.

**3. Data Pre-processing Procedures**

As is noted above, once obtained from Kaggle, test data was split into 1 of 10 folds, with each fold corresponding to 1 2-month time period from March 2011 through October 2012 (see Results). All data from February 2010 to March 2011 was used, if appropriate, for training during processing for each fold. Additionally, for each fold, any data prior to that of the current fold was added as training data.

*3.1 Singular Value Decomposition (SVD) of Training Data*

During the processing of each fold, singular value decomposition was performed (following instructor guidance [2]) on the training data to determine the top *n* principal component stores and weeks that best summarized trends in sales for each department across stores. This was done to reduce noise in weekly variance per department in the training data, which, in turn, should help prevent over-fitting. To do this, the number of unique departments in the fold’s training data were first determined. Then, for each department, if the number of week + store combinations was greater than *d* (the top number of elements to keep), a matrix of number of stores (*m*) X number of weeks (*n*) was constructed, store means were removed and saved, and SVD was performed by passing the resulting matrix to R’s *svd()* function with the following arguments:

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Value** | **Reasoning** |
| *x* | *m* x n matrix on which SVD is to be performed | This is the matrix on which SVD was performed. |
| *nu* | 8 | This is the number of left singular vectors to be computed. This was set to 8 because we wanted to keep the 8 principal store values for each department. |
| *nv* | 8 | This is the number of right singular vectors to be computed. This was set to 8 because we wanted to keep the 8 principal week values for each department. |

The resulting left (*U*), diagonal (*D*) (with only its first *d* entries, all other values being 0), and right (*Vt*) matrices were then added back to the store means as follows to produce the transformed training data matrix for the department (*T*): T = *UDVt* + store mean. This was then repeated for all other departments and a full, transformed training dataset was returned.

**4. Model Implementations**

4.1 Linear Model Implementation

Before developing the Linear Model, some initial steps were taken to ensure the code could be completed efficiently. To start, unique combos of Store and Department were found and localized to ones that occurred both in the train and test dataset. Then, for both train and test, the necessary samples were identified based on the unique combinations found previously while also having dummy variables created through the use of the *model.matrix* function.

The samples (unique combo of store & dept for given date-range) could then be iterated through while having a linear regression model selected from the training data. The linear model was selected through the use of *lm.fit* on a design matrix that had Store, Department, Year, and Week as the features. The models’ coefficients were then extracted which were then used to make the corresponding predictions that were stored off in a pre-allocated list. This list of predictions was then turned into a table after all the samples were processed which was more efficient than continually concatenating smaller tables. The models’ predictions were then evaluated for correctness by using WMAE. This process was repeated for each of the 10 folds and was conceptually adopted from Instructor guidance[2].

**5. Results**

*5.1 Model Accuracy*

As noted in the table below, our model’s average test WMAE was below the necessary threshold of 1610 across all folds.

|  |  |  |
| --- | --- | --- |
| Fold | Months | Test WMAE |
| 1 | 2011-03 & 2011-04 | 1941.581 |
| 2 | 2011-05 & 2011-06 | 1363.462 |
| 3 | 2011-07 & 2011-08 | 1382.497 |
| 4 | 2011-09 & 2011-010 | 1527.28 |
| 5 | 2011-11 & 2011-12 | 2310.469 |
| 6 | 2012-01 & 2012-02 | 1635.783 |
| 7 | 2012-03 & 2012-04 | 1682.747 |
| 8 | 2012-05 & 2012-06 | 1399.604 |
| 9 | 2012-07 & 2012-08 | 1418.078 |
| 10 | 2012-09 & 2012-10 | 1426.258 |
| Total Average WMAE | | 1608.776 |

*5.2 Processing Runtimes*

Using a Dell Precision 5550 laptop with an Intel Core i7 vPRO 2.70 GHz processor and 32 GB SSD memory, we saw a total runtime, for all 10 folds of data, of 61.18 seconds.

**6. Discussion**

During this project, a linear regression model was developed to be able to predict future weekly sales data based on historical data from different Walmart stores (45 different stores). In addition to the model that was developed, Singular Value Decomposition was used to process the training data to help reduce variance and have more accurate predictions. With this technique, sufficient accuracy was achieved across the 10 folds as a total average WMAE of 1608 was seen, which is less than the 1610 target value.

It is noteworthy that when looking at the WMAE values across the 10 folds, most of the folds (9 of 10) were relatively low had a value no greater than 1.25x the target value. The one exception fold was fold 5 which happened to contain two holiday weeks. Since holiday weeks are weighted higher here, this fold had a higher WMAE value than the rest of the folds. In a potential future iteration of this model, this could be addressed by using a post-prediction adjustment to circularly shift a certain amount of sales (amount TBD) within the effected timeframe (last 5 weeks of year). This would theoretically offset some of the impact seen with having multiple holiday weeks.

**7. References**

[1] Walmart (2014, May). Walmart Recruiting - Store Sales Forecasting: Use historical markdown data to predict store sales. Retrieved November 7, 2021 from https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting.

[2] Liang, F. (2021, October). Instructor guidance for Project 2. Retrieved November 7, 2021 from Campuswire: https://campuswire.com/c/G497EEF81/feed/654