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

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

Project 3: Movie Review Sentiment Analysis

**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. Jake created the Rmarkdown file explaining how the vocabulary was constructed, and documented and discussed the results. Michael wrote the introduction, methods outlining model implementation and training processes, and a discussion of model interpretability.

**2. Introduction**

*2.1 Background, Problem Statement, and Objective*

*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.

*2.3 Customized Vocabulary Construction*

**3. Technical Details**

As is noted above, once obtained from Kaggle, test data was split into 10 folds, with each fold corresponding to one 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. Then, the following additional pre-processing steps were performed prior to model training and testing. These steps were implemented following instructor guidance [2].

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

During the processing of each fold, singular value decomposition was performed on the training data to determine the top *d* principal component stores/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, helped 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 (*d*). 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 (*d*). 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 (*V*) 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.

*3.2 Additional Filtering*

During the processing of each fold, only data corresponding to unique *Store*|*Dept* pairs present in both the training and test data were kept. Once unique combinations were determined, they were saved and used to filter both data sets accordingly.

*3.3 Dummy Coding of Dates*

Each observation in the test and train data corresponded to the total sales for a given department within a Walmart store for a given week. In the data, the week was identified using the date of the first day of the week as a string with formatting YYYY-MM-DD. For the week to be evaluated by a linear regression model, the week number (1-52) was extracted from the date string using the *lubridate* R package and the new 'Week' variable was encoded as a factor with 52 levels. The year was also extracted from the string using the same package and was kept as an integer.

Once preprocessing was complete during the processing of each fold, a linear regression model with 'Weekly\_Sales' as the response and 'Week' and 'Year' as predictors was fit for each Walmart store and department using the training data for that store and department. Similarly, the model was used to generate predictions using observations for that store and department in the test data.

**4. Linear Model Implementation**

Once preprocessing was complete during the processing of each fold, a linear regression model with 'Weekly\_Sales' as the response and 'Week' and 'Year' as predictors was fit for each Walmart store and department that was previously found to be present in both the test and training. Similarly, the model was used to generate predictions using observations for that store and department in the test data.

It is worth noting that each linear model was fit using R’s *lm.fit()* function on a design matrix (built using the *model.matrix()* function) that had 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. To calculate WMAE predicted response values for 'Weekly\_Sales' were compared to observed values using each test data observation’s 'Date', 'Store', and 'Dept' values as a composite key. This process was repeated for each of the 10 folds and was conceptually adopted following 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.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test AUC and Processing Time by Split Number** | | | | | |
| Metric | 1 | 2 | 3 | 4 | 5 |
| Test AUC | 0.9601 | 0.9632 | 0.9629 | 0.9635 | 0.9627 |
| Processing Time (seconds) | 26.9400 | 28.0327 | 27.3694 | 26.8944 | 27.2591 |

*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 5 test/train splits, of 137.06 seconds. This does not include processing time for vocabulary construction. Average processing time for each split (training and prediction/evaluation) was 27.41 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] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). “Learning Word Vectors for Sentiment Analysis.” *The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).*

[2] Liang, F. (2021, November). Instructor guidance for Project 3. Retrieved November 28, 2021 from Campuswire: https://campuswire.com/c/G497EEF81/feed/1148