Capstone 2 Milestone Report 2

Introduction

Walmart is an American retail operation that operates a chain of grocery stores by the same name. Since its opening in 1962, it has expanded its ventures to many other countries, including Mexico, China, and Germany. As of 2019, Walmart is the world's largest company by both revenue and number of employees, at $514.405 billion per year and 2.2 million people, respectively.

Like all companies, Walmart's revenue is impacted by a variety of factors, some of which are included in the Walmart Recruiting dataset on Kaggle https://www.kaggle.com/bletchley/course-material-walmart-challenge#test.csv). Using this dataset, I would like to fit a regression model to predict the weekly sales made given the factors. In this report, I'm going to continue on with the work that I've done in my first report and fit some Machine Learning models.

Feature Engineering

Before fitting a model with our training data, I deleted some features to reduce redundancy and engineered some new features to either restate the information in the deleted columns in a way more interpretable for the model, or to derive predictors that could be helpful for revenue prediction.

The features I engineered are:

1. **Total Markdowns**

As seen in Figure 2, the markdown columns are highly correlated with each other and contribute to multicollinearity among the features. Considering that the total amount of discounts is more likely to contribute to revenue than a specific promotional event, I combined them into one Total Markdowns column, which kept the relevant information about how much discounts there were and reduced redundancy.

2. **Type A, B, and C store dummy variables.**

In order to use categorical variables in regression models, they have to be converted to dummy variables.

3. **Year, Month, Week of Year, and Day**

The year, month, week, and day could have an impact on the revenue made. The original date format won't work with a lot of models and thus, I parsed the date into year, month, week of year, and day columns.

4. **Yearly Median value of each Store-Dept combination**

This provides a baseline value for each store-department combination that the models could modify for each week given the other attributes. I'm using the median instead of the mean because the mean can be biased by extreme values, and thus won't give as clear an indicator of each store-department's average performance.

5. **Lagged Revenue and its difference from Median**

The median provides a baseline for our predictions, but doesn't tell us anything about how well the specific department-store combination is doing at a specific period in time. To account for this, I added a column with the weekly revenue of the department-store combination during the week before. I'm also adding a difference from median to give information on how well the particular combination did last week relative to its median.

6. **Big Holidays**

It appears that a significant portion of weekly revenues > $300,000 are made in the Thanksgiving, Christmas weeks, so I made some dummy variables for them. The week before Christmas also sees higher sales, so I made a dummy variable for that, too.

7. **Department 72 dummy variable**

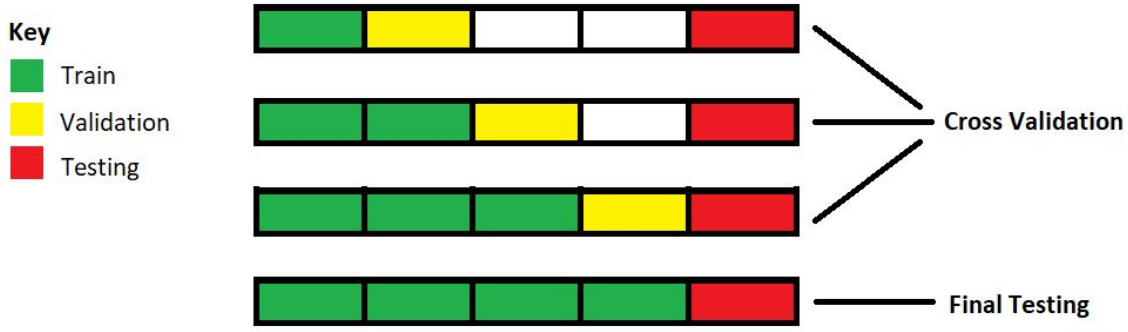
Looking at the EDA, department 72 sees higher sales compared to other departments, and so I created a dummy variable for that.

Machine Learning

After adding and deleting the features described in the “Feature Engineering” section, I split the data in the “training” csv file into the training and testing dataset, with the chronologically latest 20% as the test dataset and the remaining 80% as the training dataset. I split the data this way instead of the usual random split because our data has a time series element to it, and since we’re trying to predict a store’s weekly revenue, we cannot randomly split the dataset without consideration of the time series element.

In the same logic I cross-validated my results not by splitting and testing on 4 equal segments, but on 4 incrementally larger parts and predicting each quarter with all the data that came before. (See Figure 6 for more explanation), which is standard for Time Series Forecasting.

Figure 6. Explanation of cross-validation.



The models I used in this capstone are:

- Ridge Regression
- Lasso Regression
- RandomForestRegressor
- XGBoost
- Support Vector Regressor

For Ridge and Lasso Regression, I used sklearn's RFE (Recursive Feature Elimination), which is a feature selector that recursively selects a smaller and smaller subset of features until the optimal subset of features is reached. Unlike the tree-based regressors, Linear Regression and its related algorithms do not have a built in function that measures each feature's contribution to variance in the target column. RFE should help to reduce the distraction from unimportant features and better accuracy of prediction.

In the models I tried both the unscaled, unmodified dataset and a version of it with standardized features, because for some models standardizing promises better results. The validation results are shown in Table 1.

Table 1. MAE of different models for prediction of weekly revenue

Model	MAE with scaled dataset	MAE with unscaled dataset
Ridge Regression	14979.07	2641.02
Lasso Regression	15045.89	2631.40
RandomForestRegressor	11766.21	2573.75
XGBoost	13107.88	**2224.51**
SVR	13345.28	13431.92

XGBoost trained using the unscaled dataset produced the model with the lowest MAE. Generally, non-linear regressors like RandomForest and XGBoost performed better than linear regressors like Ridge and Lasso, and, in contrast to what was usually expected, scaling actually lowered the performance in all of the algorithms besides SVR. Further investigation is needed to better understand the results.

According to RandomForestRegressor's feature importances on the unscaled dataset (Figure 7), the feature that contributed the most to our results is Lagged Revenue, mostly likely due to its closeness in scale to the Weekly Revenue.

Figure 7. RandomForestRegressor Feature importance on unscaled dataset