1. Team members and contributions

a. Anurag Anand
   - Performed Exploratory Data Analysis
   - Wrote Data Transformation Pipelines
   - Built Gradient Boosted and Random Forest Models

b. Sandip Sonawane
   - Performed Exploratory Data Analysis
   - Features Preprocessing
   - Built SVD + Linear Model

2. Introduction

The goal of this project is to predict the future weekly sales of different walmart stores using the Walmart store sales forecasting dataset available on Kaggle. This is primarily a regression problem with time series data. In this project preprocessing of input data is done to make sure our model does not deviate too much because of outliers or missing values. We built least squares, tree-based regression models (Gradient Boosted Trees and Random Forest Models) to generate forecasts.

3. Data Source

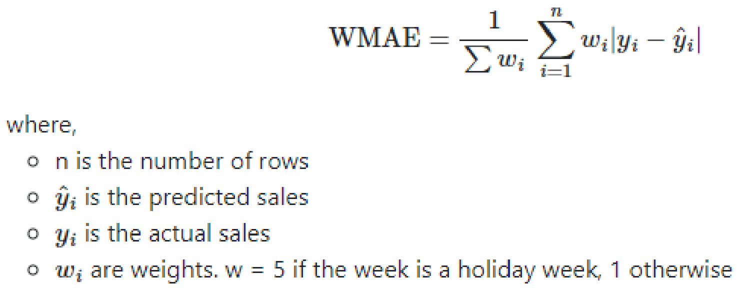
The data is taken from Kaggle. We are provided with historical sales data for 45 Walmart stores located in different regions in the United States. Each store contains a number of departments, and we are tasked with predicting the department-wide weekly sales for each store. For this project, we are only required to use the 5 variables viz. Store, department, date, isholiday and weekly_sales. The description of each variable can be found at this link.

We have the below files to build and train models.

- **train_ini.csv:** 5 columns (“Store”, “Dept”, “Date”, “Weekly_Sales”, “IsHoliday”), same as the train.csv file on Kaggle but ranging from 2010-02 to 2011-02.
- **test.csv:** 4 columns (“Store”, “Dept”, “Date”, “IsHoliday”), in the same format as the train.csv file on Kaggle ranging from 2011-03 to 2012-10 with the “Weekly_Sales” column being removed.
- **Folds:** fold_1.csv, …, fold_10.csv: 5 columns (“Store”, “Dept”, “Date”, “Weekly_Sales”, “IsHoliday”), same as the train.csv file on Kaggle, and one for every two months starting from 2011-03 to 2012-10.

4. Evaluation metric

To measure the performance of regression models, weighted mean absolute error (WMAE) is used as a performance metric.



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

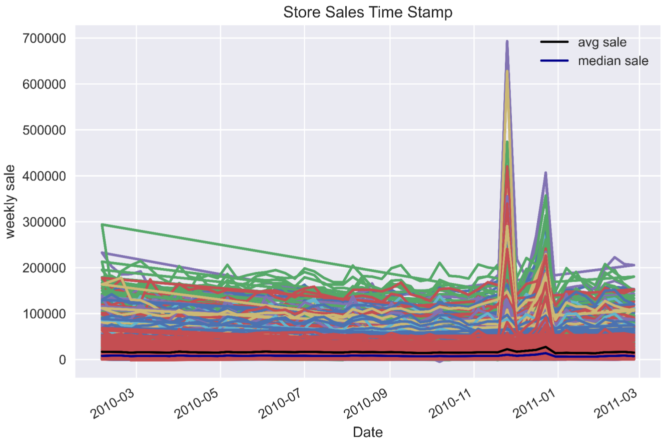
where,

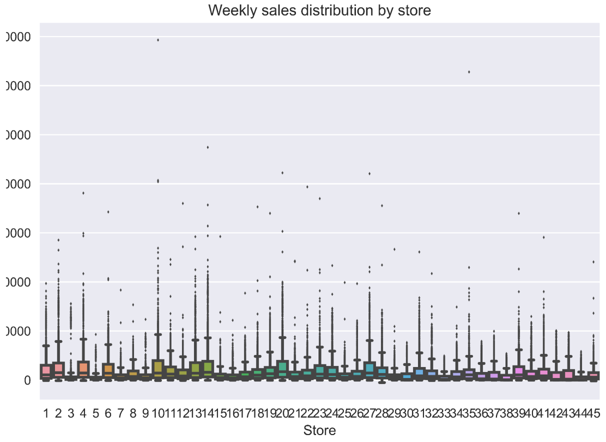
- n is the number of rows
- $\hat{y}_i$ is the predicted sales
- y_i is the actual sales
- w_i are weights. w = 5 if the week is a holiday week, 1 otherwise

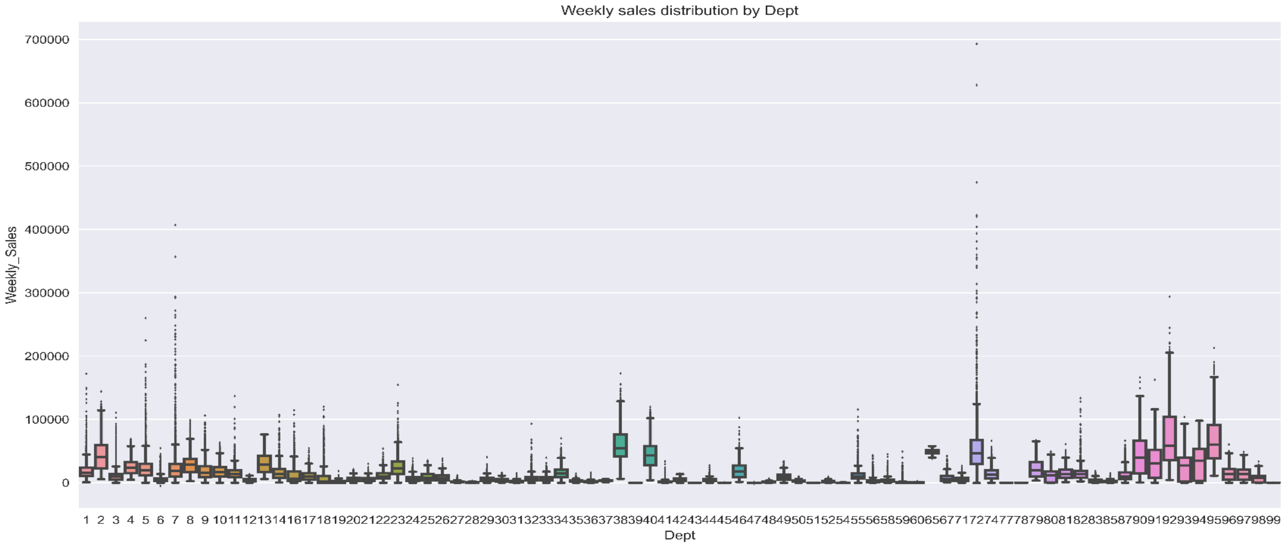
5. Exploratory Data Analysis

For this project, we only have 4 explanatory variables. Below are the observations from exploratory data analysis.

1. There are no missing values.
2. Different stores have different mean weekly sales
3. Some departments have very high mean weekly sales while some departments have close zero mean weekly sales.
4. Sales spike during the holiday season.
5. There are few weekly sales having negative values, we replaced these with absolute values. These values were comparatively small and wouldn’t affect the model much.







6. Data Pre-processing

We created a 'week' variable from date. We confirmed that the holidays fall into the same week for each year when we look into the data of subsequent years. In addition to that, we also created a 'year' variable. Since we were going to try out different modelling methodologies, we took some liberty during the data preparation phase. For the linear model we stuck with year and week variables, but for the tree based models we augmented the data with lag variables of order one (which we call last week sales) and created 2nd, 3rd and 4th order differences of the last week sales.

7. Modeling

7.1 RandomForest

Our First Model was a random Forest Model. In order to arrive at the best model for each fold, we undertook 5 fold cross validation to arrive at the optimal set of hyper-parameters for each fold, then we utilized the list of best parameters to do the prediction for each fold. The parameter search space was *'max_features':['auto'],'max_depth':[20,25,30,35],'min_samples_leaf':[15,20,25,50].*

7.2 Gradient Boosted Trees

Following the Random Forest model we went ahead with Gradient Boosted trees. We followed a similar approach and did a 5 fold cross validation for each fold to arrive at the optimal set of hyperparameters. We used the list of best parameters to do the prediction for each fold. The parameter search space for each GBR tree was *{'max_features':['auto'],'max_depth':[1,3,5],'min_samples_leaf':[75,100,125,150]}*. For all the other parameters we used default settings

7.3 Linear Model

As per prof. Liang's post on what we have tried, we created two new features: 'year' and 'week'. For each department and store combination, we formed the weekly sales data. We made sure that 'week' is a categorical variable since it does not have continuity and for next year, weeks will start from 1.

7.4 SVD + Linear Model

To reduce noise in training data, we performed SVD on as per prof. Liang's post on using SVD with 8 components and then reconstructed the training data with these 8 components. This model was able to achieve an average 10 fold WMAE below 1610.

8. Results

8.1 Random Forest (Foldwise) [ordered from fold 1 to fold 10]:
[1382.695, 1813.732, 3091.999, 1595.443, 5757.098, 2031.333,1620.199, 1395.233, 1352.661, 1546.113] **(Avg WMAE):** 2158.650

8.2 GBR (Foldwise) [ordered from fold 1 to fold 10]:
[1358.266, 1952.12 , 3104.36 , 1595.951, 5939.935, 2155.371, 1529.642, 1386.543, 1291.52, 1455.87] **(Average WMAE):** 2176.95

8.3 Linear Model (Foldwise) [ordered from fold 1 to fold 10]:
[2045.243 1466.912 1449.852 1593.998 2324.496 1677.483 1722.274 1428.212 1443.960 1444.656] **(Average WMAE):** 1659.709

8.4 SVD + Linear Model (Foldwise) [ordered from fold 1 to fold 10]:
[1941.581, 1363.462, 1382.497, 1527.280, 2310.469, 1635.78, 1682.74, 1399.60, 1418.01, 1426.26]**(Average WMAE):** 1608.776

9. Running Time

- The system used for running xgboost model: Legion 5, Win 11, Intel Core i7-10750H CPU @ 2.60GHz. 16 GB ram. It takes 8 min 30 seconds to train 10 different models for SCD + linear model.
- The system used for running the Tree Based Methods: Win 10, Intel Core i5 - 10600K CPU @4.10GHz (12CPUs). It takes 5 min to train 10 different models for both Random Forest and GBR

10. Interesting Findings

- The performance of GBR and Random Forest is very identical and the difference between the two is not significant (The random forest performs marginally better than GBR)
- The most important variable in the tree based methods last weeks sales ~(0.98) across all the folds
- This explains why our prediction goes bad for folds 3 and 5. These two folds have holiday weeks which see an unusual spike in sales and its value is significantly different from previous weeks sales value
- SVD reduces the noise in the data and overcomes the drawback of the tree based methods and gives accurate forecasts

11. Conclusion

The project gave us deep insight into how time series data behaves and potential methodologies to model time series data. We learned to model time series data without using pure-play time-series modeling techniques like ARMA, ARIMA, etc. The solution that we have explored utilizes tree-based and low-rank approximation-based methods to forecast weekly sales. The tree-based methods did not work for two prediction windows. In both the windows, our predictors were not able to capture the seasonality effect. The SVD coupled with a linear model effectively modelled the time series. We also learned how to reduce code running time in the evaluation environment by hard coding the best hyperparameters (found using CV) and changing the hyperparameters dynamically for each fold. We also learned the importance of feature engineering and transforming input variables to improve model performance. In this project, the transformations performed were quite simple yet very effective in reducing average WMAE.

12. References

1. Walmart Recruiting - Store Sales Forecasting. Kaggle https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting/
2. STAT 542 - Project 2. UIUC https://liangfgithub.github.io/F21/F21_Project2.nb.html
3. STAT 542 - Campuswire posts on Project 2. UIUC https://campuswire.com/c/G497EEF81