# Practical Statistical Learning – Fall 2020

## Project 2 (Walmart Stores Sales Forecasting)

**Team Information:**

*Balaji Sathyamurthy (balajis2)*

*Gowri Shankar Ramanan (gsr2)*

**Abstract:**

This paper describes the exploratory study of historical time series sales data for 45 Walmart stores spanning across 99 departments located in different regions and tuning the best algorithms to obtain the best possible results for weekly sales forecasting. The metric used to evaluate performance for this report is Weighted Mean Average Error (WMAE). Each store contains many departments and the time series within the same department are noted to have similar patterns. The training data and relevant information are available at https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting. The overall data showed a 52-week seasonal frequency.

**Goal:**

Our goal is to predict the weekly sales for a series of 2-month period starting from the first period between March 2011 - April 2011, given training data containing weekly sales for various stores and departments from Feb 2010 to Feb 2011. We split the test data into 10 separate folds each representing the 2-month period being forecast. The folds representing the subsequent forecast periods do not overlap with each other. With respect to the dependency structure of the time series, each fold makes forecast based on all the data available until the start period for that fold, which includes the predictions made from the previous fold. So, the last fold (Fold 10) predict for the period Sep 2012 – Oct 2012, based on the data from March 2011 all the way until end of Aug 2012. It is interesting to note that the training data across 10 folds represent 56 weeks of sales data in total, but not all departments have data for all weeks.

**Implementation Details:**

We treat the problem as a time series forecasting problem and omit IsHoliday feature. We evaluated several time series forecasting models including ***naïve*** (Naïve method), ***snaive*** (Seasonal Naïve method) and ***tslm*** (Time Series Regression model) using ***forecast*** package in R. We used ***tidyverse*** for data manipulation and ***lubridate*** for date manipulation. Our code was adapted from various links provided in the references section below, which basically loops over each department and construct a Date-By-Store design matrix separately for each department containing weekly sales data, with each row representing a particular week in the time series (total of 52 rows for 52 weeks in a year period), and each column representing a time series for a particular store (45 columns for 45 stores). The missing sales data is filled with zero value as some stores do not have sales for some weeks. The constructed design matrix is then evaluated against each forecasting model using ***forecast*** R package, to measure the performance at each

fold in terms of Weighted Average Error (WAE). The WAE was found to be the lowest when using ***tslm*** considering both season + trend as predictors for the time series of each store as input. The ***tslm*** automatically creates variables 'trend' and 'season' from the time series characteristics of the data. WMAE is the average of WAE across all 10 folds. Initially, we arrived at WMAE of 1659 employing this model, which was further reduced by overcoming the challenges below.

**Challenges and Solutions:**

The challenge here is that we had to predict sales for Christmas week (Week 52). Unlike most holidays, Christmas happens on a fixed day and therefore its day of the week changes every year. The Fold 5 data was challenging because in 2011, Christmas falls on a Sunday. However, in our train data, we had a Christmas week from 2010 when Christmas falls on a Saturday. The weekly sales were spread starting on Saturday and ending on Friday. We observed there is one pre-Christmas shopping day in week 52 of test data which could increase the weekly sales price in week 52 of test data in fold 5, while this trend is absent in train data.

We learnt that the solution to handle this anomaly is to shift the weekly sales prices in our predictions by 1 day. Based on exploration, we observed that sales are very high around Christmas and Thanksgiving weeks (week number 48 & 52). So, to account for the holiday adjustment, we leveraged and modified the "***shift***" function to shift the weekly sales by 1 day only for Fold 5 if the average weekly sales of the adjacent week are higher than a preset threshold. The threshold is set to 10% to evaluate. The WMAE has dropped to 1623.499 after shifting of weekly sales from week 48 through 52 into the next week for Fold 5, while keeping the total sales of overall 5 weeks constant. We also tried to apply this shift to all other folds to observe that the WMAE has dropped further down by about 1 unit to 1622.3, while increasing the overall prediction time up by 4 more seconds (taking more than ~240 secs). We have chosen to apply the shift only for Fold 5 considering optimal time taken over the tradeoff with relative prediction accuracy gained without increasing much of overall processing cost.

**Results & Performance:**

The WAE (Weighted Average Error) is evaluated for each of the 10 folds of test data, and their mean is calculated to measure the WMAE. Please see **Table A** below for the results at each fold.

**Table A:** WAE and Prediction Time at each Fold

| Fold# | WAE | Prediction Time Taken |
|---|---|---|
| **1** | 2042.401 | 23.79558 secs |
| **2** | 1440.083 | 21.73666 secs |
| **3** | 1434.716 | 22.52397 secs |
| **4** | 1596.988 | 22.73068 secs |
| **5** | 2029.388 | 24.49955 secs |

| | | |
|---|---|---|
| **6** | 1674.185 | 24.09024 secs |
| **7** | 1718.577 | 23.75373 secs |
| **8** | 1420.817 | 24.72175 secs |
| **9** | 1430.801 | 24.59973 secs |
| **10** | 1447.034 | 24.32339 secs |

| **mean(wae)** | **total prediction time taken** |
|---|---|
| 1623.499 | 236.78 secs |

**Evaluation Environment(s) Tech Specs:**

- OS: MacOS Catalina 10.15.7 (19H2)
- Processor: 2.3 GHz Dual-Core Intel Core i5
- Memory: 8 GB 2133 MHz LPDDR3

**References:**

https://liangfgithub.github.io/Example_Code_Project2_Josh.html
Forecasting: Principles and Practice
https://github.com/davidthaler/Walmart_competition_code/blob/master/postprocess.R