Predicting Weekly Sales for Departments of Walmart Stores Based on Historical Sales Data and Location-Specific Factors

1. Authors:

1. Veera Marni vmarni@iu.edu
2. Sam Durham samdurha@indiana.edu
3. Udit Patel udipatel@iu.edu

2. Objectives and significance:

The data set provides data of 45 Walmart stores across various regions and each store again has various departments. Our task is to predict the department wide sales for each department in each store. Walmart also provides several promotional markdowns throughout the year and these markdowns are preceded with prominent holidays (the four largest of which are the Super Bowl, Labor Day, Thanksgiving, and Christmas). These markdown are known to impact sales and our goal here is to predict which department is affected in which store and the extent of impact due to this markdown. The challenge here is to model the effect of markdown on these holiday weeks in the absence of complete historical data. Additional information about location-specific factors (e.g. temperature, fuel prices, unemployment, etc.) are provided that may assist in predicting future sales.

It is important because if we are able to predict the department that are impacted and their extent of impact, we can take steps to avoid problems that might raise in that department such as running out of stock due to excess sales, maintaining extra work force at departments which generally outperform during this period and so on.

Our motivation in doing this is to learn to use our data mining skills to accurately predict which department is going to be mostly affected during the holidays. By doing this we can implement the knowledge and skills that we obtained in this course.

3. Background:

Retail stores are interested in predicting future sales. Accurate predictions of future sales allow stores to maintain an appropriate inventory of goods and employ discounts/promotions that may spur shopping during slow periods. Certain location-specific factors such as weather and fuel prices may also influence consumer spending.

It will likely be important to group stores/departments into categories according to sales patterns, location-specific features, and other features of the data set. The result of this will be helpful in dividing the departments into different categories and when making a prediction this will be helpful to increase the size of the sample data.

New patterns can be found that show how prices of gasoline or weather conditions play a significant role in sales of a department in a particular store at a unique location.

4.Literature Search:

After Walmart has released this dataset many have worked on this particular dataset using different approaches. Moreover this particular dataset has been used in various papers. Here are two decent approaches that caught our eye:

1. Ensemble Learning approach by Sriok(a Kaggle competitor):

This particular user developed a model using 6 different techniques (ARIMA, Unobserved Components Model, Random Forest, KNN Regression, Linear Regression and Principal Component Regression) and finally calculated simple average of all the six models which gave him decent results. Further he improved the accuracy of his results by introducing some different way (not mentioned by the user) to calculate average of all models which he says gave him a good accuracy.

2. Xiv International Symposium New Business Models and Sustainable Competitiveness Symposium Proceedings- Data Analysis and Sales Prediction in Retail Business.

In this paper the authors tried various approaches on the data to aiming to get the best results. Out of all the approaches they tried they concluded that using SVM algorithm on Rapid miner tool gave them the best predictions. In this approach the authors tried to divide the data into smaller fractions at certain criteria (based on departments, holidays, etc.) and then run several iterations to get numbers in predicted sample stable enough.

Some other related works in this scope:

Gicheva et. al. researched the correlation between gas prices and the percent of certain products that are sold at discount price. This research found that consumers' spending habits were influenced by the price of gasoline.

Ma et. al. found that monthly number of shopping trips, expenditure, and purchase volume all decrease significantly as gas prices increase. This research was conducted for a wide range of products and likely provides a more relevant analysis for our particular research.

Murray et. al used a random effects model to investigate the effects of various weather parameters (e.g. temperature, sunlight, humidity, etc.). This study did not find a statistically significant correlation between temperature and the willingness to consume a certain product; however, this research was conducted on a narrow group of consumer products. Furthermore, the laboratory nature of the experiment precluded the need for participants to travel to a store (were simply offered the product). Considering these factors, this research may have limited applicability in our analysis.

Regarding general sales forecasting research, Zhang et. al. used a neural network model to predict seasonal sales, while Divakar et. al. also investigated seasonal sales, including holidays such as Christmas, Thanksgiving, and the Super Bowl.

5. Methods:

Data:

We obtained our data from a Kaggle web competition. Our data consists of weekly, departmental sales for 45 Walmart stores over a period of ~ 2.5 years (2/5/10 - 10/26/12). A summary of the weekly sales data is shown in Table 1 below.

Attribute	Num Attribute Values	Range
Store	45	1 - 45
Dept.	99	1 - 99
Date	143	2/5/10 - 10/26/12
Weekly Sales	continuous	-$5k to $693k
Is a Holiday?	2	YES - NO
Total Num Records	421,570	

Table 1. Summary of store sales data.

Location-Specific Dynamic Factors:

In addition to the sales data for each week, we also have information about various location-specific factors that may have an effect on the sales for a given store. These location-specific factors change on a weekly basis and include the following:

- Temperature
- Fuel Price
- Extent of markdown
- Consumer price index (CPI)
- Unemployment

A summary of the location-specific dynamic data is shown in the table below.

Attribute	Num Attr. Values	Range
Store	45	1 - 45
Date	143	2/5/10 - 7/26/2013
Temperature (F)	Continuous	-7 to 102
Fuel Price		2.47 - 4.47
Markdown 1 - 5		-2781 to 771k
CPI		126 to 229
Unemployment		3.68 to 14.31
Is a Holiday?	2	YES - NO
Total Num Records	8,190	

Table 2. Supplemental Store Data for Given Dates

The data summarized in Table 2 covers both the training time period (2/5/10 - 10/26/12) for which we have sales data and the test time period (11/1/12 - 7/26/13) for which we need to predict sales.

Location-Specific Static Factors:

Several additional store-specific features are provided that do not change over time. These include the following features:

- Store Type (A, B, C)
- Store Size

Attribute	Num Attribute Values	Range
Store	45	1 - 45
Type	3	A - C
Size	continuous	35k to 220k
Total Num of Records	45	

Table 3. Store Data

Holiday Factor:

In addition to location-specific factors, the presence of a holiday may influence the sales for a given week. In this analysis, the weeks that contain the following holidays are indicated as "holiday" weeks for the weekly sales data:

Super Bowl: 12-Feb-10, 11-Feb-11, 10-Feb-12, 8-Feb-13

Labor Day: 10-Sep-10, 9-Sep-11, 7-Sep-12, 6-Sep-13

Thanksgiving: 26-Nov-10, 25-Nov-11, 23-Nov-12, 29-Nov-13

Christmas: 31-Dec-10, 30-Dec-11, 28-Dec-12, 27-Dec-13

Some terminology that might come on your way:

1. Store-Dept doubles is when we are considering a particular department of a particular store. So here we are having 44 stores, 99 departments and hence 44×99 such combinations.
2. Store-Dept-week triplet is when we are considering the sales of a particular week of a particular store-dept double. So here we have 44 stores, 99 department and 44×99×52 such combinations.

Linear Regression Analysis:

Our analysis was performed with Python code. Most of the code relied on Pandas data frames for manipulating the data and performing calculations. At times, we used object-oriented programming in addition to these data frames.

In addition to the historical sales data for store-dept-week triplets, we have location-specific factors such as temperature, gas price, and others (see Table 2 above for full list). Given this data, we hoped to find trends between these factors and the sales values, that would then be useful for predicting sales in future weeks (for which we also have those same factors); therefore, the first step in our analysis was investigate the whether such trends existed. We decided to look for trends within each store-dept double.

For each store-dept double, we gathered all of historical weekly sales data (along with the location specific factors for those weeks). For each store-dept double, this collection of values was represented as a table, where rows corresponded to records (i.e. a particular week) and columns were used to represent

the sales values and each of the location specific factors. The sales data of each table was normalized with the z-score normalization technique:

$$z = \frac{x - \mu}{\sigma}$$

Where,

μ is the average sales for all the weeks of a given store-dept double.

σ is the standard deviation of the sales for all the weeks of a given store-dept double.

We used least-squares regression (numpy and pandas python packages) to fit a line to the data (sales value vs. factor values) of each store-dept double. We determined the correlation between sales and each of the four primary factors: temperature, unemployment, fuel price, and CPI.

Separating Non-Holiday and Holiday Data:

For all of the "averaging" methods (this method, as well as averaging methods described in later sections), the holiday and non-holiday sales values were separated. This means that when we are trying to predict a sales value for a future week, we make sure that sales averages of any historical data was calculated using weeks that are of the same holiday status of that future week. For many of our prediction methods, this means that we store a pair of averages (one corresponding to the holiday weeks, and one corresponding to the non-holiday weeks) during the preprocessing step, prior to making the actual predictions. For all of the regression analyses, we did not separate the holiday and non-holiday weeks (this is listed as an area for improvement in the conclusion).

Weighted Average of Adjacent Weeks for Predicting Future Sales:

In addition to investigating linear regression trends with the location-specific factors, we also implemented a sales prediction algorithm that was based on the average of historical sales data. For a given store-dept-week triplet, the average was computed for the historical sales of relevant weeks (of that same store-dept). Because data was not available for all weeks of each store-dept double, we decided to use an average of weeks that were in a certain range of a given week. Initially, we calculated the week number (1-52) for each date of the records, where a week number of 1 corresponds to the first week of the year, and a week number of 52 corresponds to the last week.

The general method is outlined with the following example: for department 1 of store 1, we may not have any historical sales data for week 4; however, sales data may be available for weeks 3 and 5 of that store.

By using an average of weeks that are within 1 week of the week of interest, we could compute an average for week 4 of that store-dept double. The weeks were assigned a weighting that decreased as the distance from the week of interest increased. An example of this weighting scheme is shown in the table below. We also include am auto weighting method that is discussed later, which turns out to produce the best results.

Week Offset	**-2**	**-1**	**0**	**1**	**2**
Weighting	10	20	40	20	10

Table 4. An example of weightings that may be used for the "weighted average" method.

As shown in the table above, week numbers that are equivalent to the week of interest (i.e. offset = 0) are given the largest weighting. The formula for the weighted average is shown in the pseudocode below.

--------Weighted-Average Algorithm----------

```
weightedProductsSum = 0
weightingsSum = 0

For week in list_of_weeks_within_range:

        weightedProductsSum +=  week.normalized_sales * weightings[week.offset]

        weightingsSum += weightings[week.offset]

return = weightedProductsSum / weightingsSum
```

The error resulting from a division by zero (when list of weeks is empty) is handled appropriately in the calling code. A calculation example is shown using the sales values below and the weightings from Table 4 above.

Week Offset	**Normalized Sales Value**
-1	0.5
-1	0.4
0	1.5
0	1.6

0	1.2

Table 5. Example values for demonstrating the weighted average method.

weightedProductsSum = 0.5*20 + 0.4*20 + 1.5*40 + 1.6*40 + 1.2*40 = 190
weightingsSum = 20 + 20 + 40 + 40 + 40 = 160

Normalized Sale = weightedProductsSum / weightingsSum = 1.18

As indicated above, we would actually run this calculation twice: once for the non-holiday weeks and again for the non-holiday weeks (within range of interest). Upon retrieving the appropriate normalized sales value for a given dept-store-week triplet (also looking at holiday status), this normalized sale value is converted to an absolute sales value with the following formula:

$$Sale = Sale_{norm} * Sale_{std,s,d} + Sale_{avg,s,d}$$

Where Sale(std,s,d) and Sale(avg,s,d) are the standard deviation and average that were stored for the store-dept when the sales values were originally normalized.

Handling Weeks that lacked historical data:

The above solution that uses a range of weeks serves two functions: 1.) it mitigates the variability that arises from averaging a small number of values 2.) it enables an average to be calculated for a week that has no historical data. Although using sales data from a range of weeks (e.g. +/- 1 weeks from week of interest) may allow for predictions of most weeks that lacked historical data, some store-dept-week triplets had neither historical data for that particular week, nor historical data for weeks within the range of weeks. In order to handle these cases, we utilized three additional prediction methods. These prediction methods are listed below, in addition to the "weighted average" method was described above.

Average-Based Prediction Methods:

1. Weighted Average of Adjacent Weeks

As described in the section and pseudocode above, for a given store-dept-week triplet, this method calculates a weighted average from sales values of weeks that are near the week of interest, from the same store-dept.

2. Average from Stores of same type

As noted in Table 2 above, each store was labeled with one of three types: A, B, or C. These types may correspond to a certain pattern of sales, such that stores with the same type have similar sales values. For a given store-dept-week that does not have historical values, one can determine the average of the normalized sales values from each of the stores of the same type, for. This normalized sales value can then be converted to an absolute sales value (i.e. not normalized) with the sales average and sales standard deviation, from the store-dept for which we are predicting the sales.

The below picture shows stores of larger size also have large sales and vice versa.

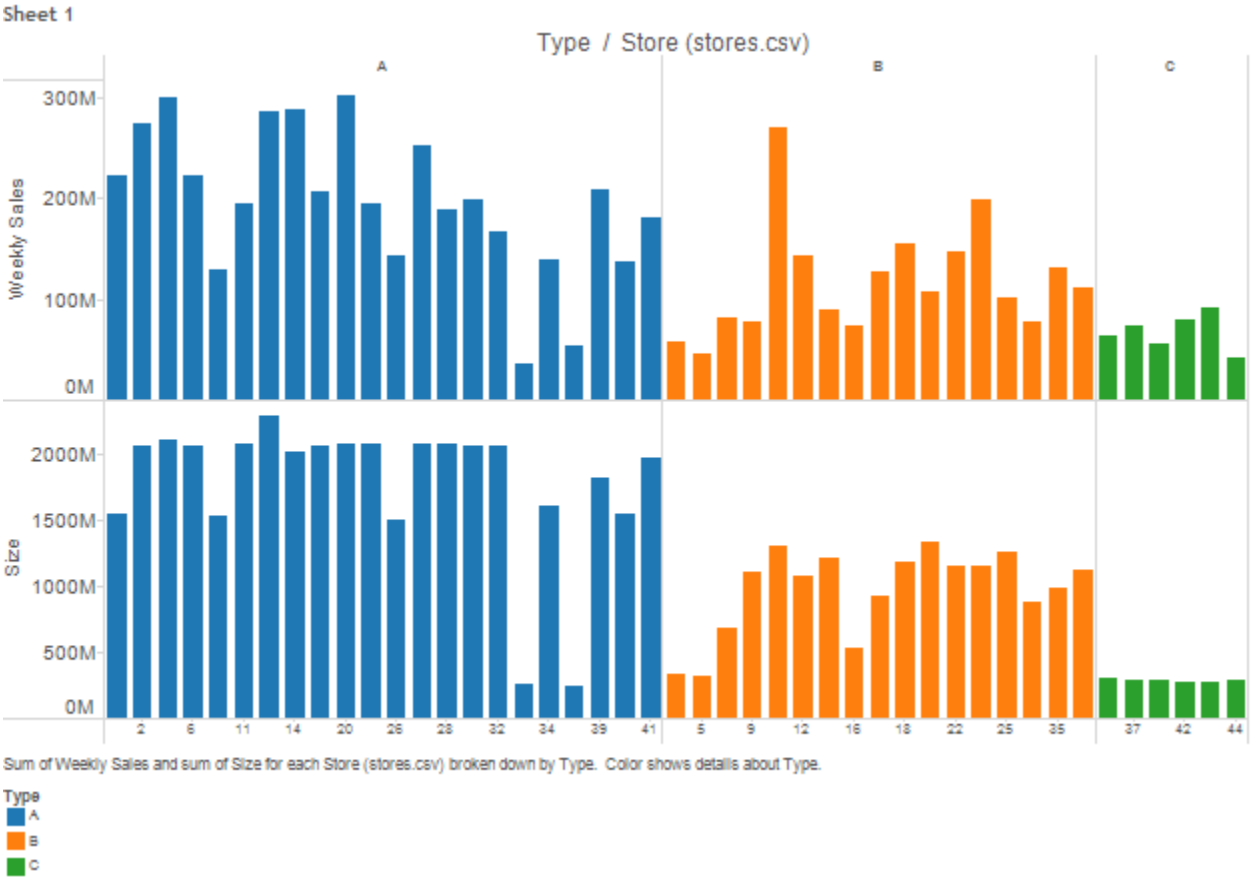

Figure 2. Total Weekly Sales and store size for all stores, grouped by type (A, B, or C).

3. Average from all Stores

This is identical to the method described above, except that the average is calculated from all stores, rather than just stores of the same type. It is expected that the above method would yield a more relevant sales prediction.

4. Average From all departments of Same Store

For a given store-dept-week triplet, this method calculates the average of normalized sale values from the other departments of the same store-week.

Chaining And Averaging Prediction Methods:

In order arrive at a single predicted sales value, one can combine the results of the above methods in two different ways:

1. Highest-Priority Sales Prediction (chaining method)

In this method, one must decide on a ranking for the different prediction methods. After deciding on this ranking, the implemented code initially attempts this most important prediction method and returns the result as the predicted sales. In the case that the first method fails, the second-most important method is attempted and the predicted sales value is returned. Otherwise, the algorithm simply uses the predicted sales value from the first method.

2. Weighted Average of Sales Predictions

This approach requires weighting assignments for all four methods. The algorithm will then run all four of the prediction methods. A weighted average of the results from these prediction methods will be computed using the assigned weightings and the general weighted-average protocol that was outlined in the pseudocode above.

Predicting Sales with Linear Regression Trends:

As mentioned at the beginning for the methods section, for each store-dept double, we calculated the correlation between the four factors (temperature, gas price, unemployment and CPI) and normalized weekly sales. Although the number of "good" correlations was small, we thought that even the mediocre correlations could be somehow utilized. The poor fit could be taken into consideration by assigning a weighting to the regression-predicted sales value that is proportional to the correlation. These regression-based predicted sales values could then be added to the list of four prediction methods that were outlined above. This overall process is described in the following pseudocode:

---------Regression-Based Prediction Algorithm----------

```
maxRegressionWeighting = 10      //This value is selected by programmer
weightedProductsSum = 0
weightingsSum = 0
```

For future_week in list_of_future_weeks:

    For factor in list_of_factors:

        model = get_linear_regression_model(future_week.store, future_week.dept, factor)

        normalizedSale= model.slope * future_week.factor_values[factor] + model.intercept

        weightedProductsSum += normalizedSale* (model.r2 * maxRegressionWeighting)

        weightingsSum += model.r2 * maxRegressionWeighting

return weightedProductsSum / weightingsSum

Thus, we will obtain a single weighted predicted sales from the regression models for all four factors. This value could be added to the list of four methods that was outlined above (with either the sequential or weighted average approach). Alternatively, the predicted sales for each of the four factors could be added to the list of four methods individually, such that the list would now have eight different methods.

Combining All of the Methods:

Several different sales prediction methods were outlined above, along with variations of those methods. In the results section below, we show results and analyses of several of these variations

Evaluation Strategy:

The predicted sales values were tested against the true sales values through the Kaggle competition website. The evalation method is outlined in the following:

This competition is evaluated on the weighted mean absolute error (WMAE):

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

Where,

- n is the number of rows
- $\hat{y}_i$ is the predicted sales

- $\hat{y}_i$ is the actual sales
- $\hat{w}_i$ is weights. w = 5 if the week is a holiday week, 1 otherwise

6.Results:

Linear Regression:

Upon calculating the correlations for each of the store-dept doubles, we produced some example plots in order to visualize the correlation and qualitatively determine the minimum R2 value that may be classified as a "good" correlation (i.e. that would be useful in predicting future sales values). The following three figures show plots of normalized weekly sales vs. Temperature of three representative store-dept doubles. The figures are arrange in order of decreasing correlation.

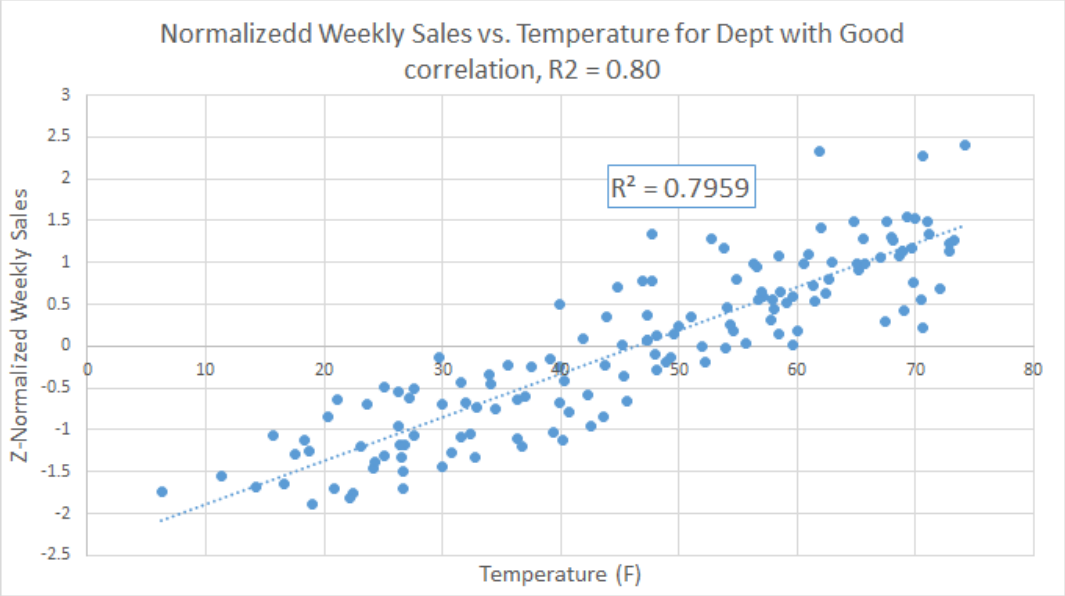

Figure 3. Weekly normalized sales vs. Temperature for a representative store-dept double with good correlation.

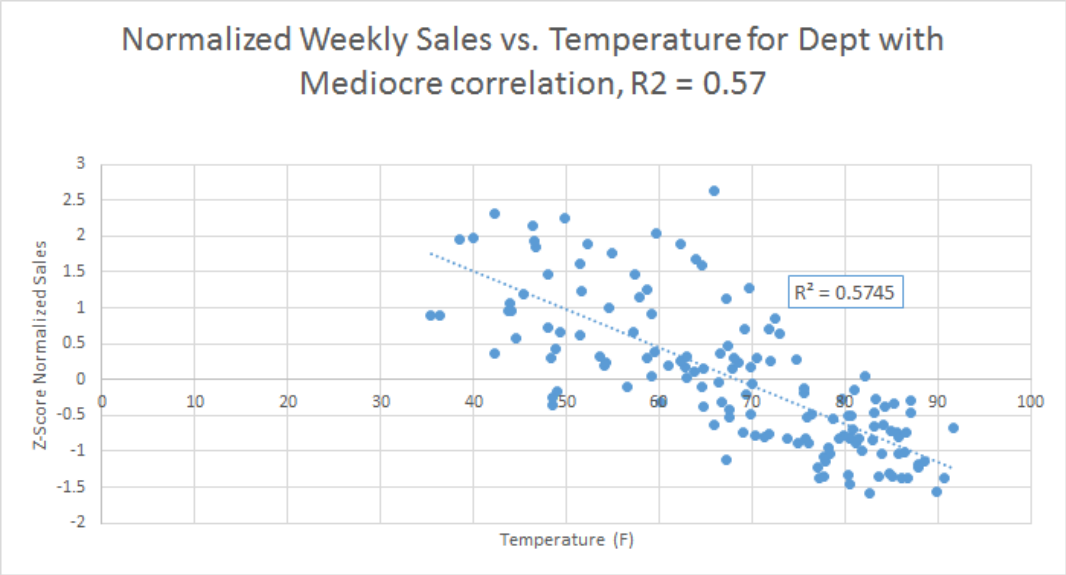

Figure 4. Weekly normalized sales vs. Temperature for a representative store-dept double with medicore correlation.

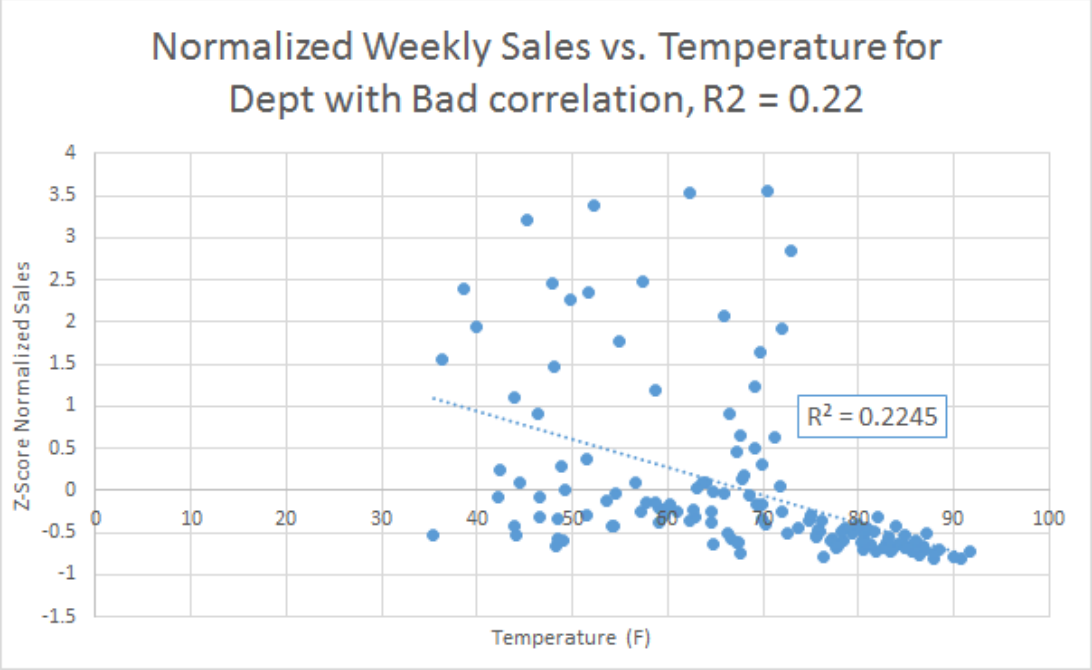

Figure 5. Weekly normalized sales vs. Temperature for a representative store-dept double with bad correlation.

The first figure has a correlation of R2 = 0.8. Visually, a trend is apparent in the graph, and one can imagine that this trend could be helpful in predicting future sales for this store-dept double, for future

weeks when the temperature is known. In the second figure (R2 = 0.57) a trend is still visible, however, the sales values significantly deviate from the best-fit line. In the last figure, although a weak trend is apparent over a select range of temperatures (~75 – 92 F), the sales values are very erratic (no trend apparent) for temperatures below this range. Although the deviation is significant, on average this trend could still provide useful information for predicting future sales, if the extent of correlation is taken into account (e.g. through some weighting scheme, as will be explained later). The numbers of store-dept doubles showing this "good" correlation are displayed in the table below for each of the four factors. Only correlations for store-dept doubles with a minimum number of 20 values were considered. With smaller sample sizes, correlations can too easily appear by chance.

	Doubles with R2 > 0.5	
	Total Num	**% of All Doubles**
Temp	55	1.26%
Unemployment	71	1.63%
Fuel Price	24	0.55%
CPI	91	2.09%

Table 6. Store-dept doubles with a minimum correlation of R2 > 0.5 for the four factors.

As shown in Table 6 above, very few store-dept doubles (relative to the overall number of store-dept double) showed a "good" correlation. Furthermore, of the factors that showed good correlation, it was suspected that the underlying factor of this correlation may actually be the week number (e.g. 1 – 52) of the year. To investigate this possibility, the data of the high-correlation figure (R2 > 0.8) above was replotted as weekly sales vs. "# of Weeks from Mid-Year." This transformation of week number was completed according to the following formula:

$$\#\ of\ Weeks\ From\ Mid\ Year = abs(WeekNum - 26)$$

Where 26 represents the week corresponding to the middle of the year (i.e. ~July 1st), such that January 1st (originally week num = 1) and December 24th (originally week num = 52) would map to adjacent week numbers (25 and 26). The existence of a correlation with this transformed attribute may suggest that the underlying cause of the sales-vs-temp correlation is actually the time of year. This correlation is shown in Figure 6 below.

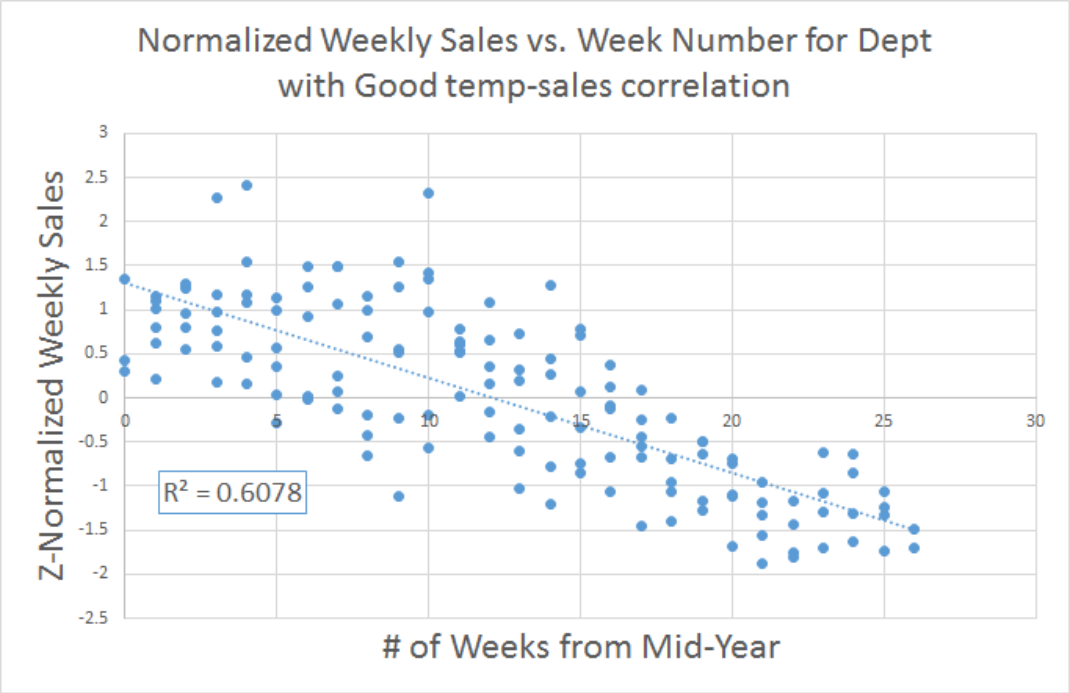

Figure 6. Normalized weekly sales vs. "# of weeks from mid-year" for the store-dept that showed a high correlation for sales-vs-temp (see Figure 3 above).

Although a good correlation exists (R2 = 0.61), it is weaker than the sales-vs-temp correlation (R2 = 0.8). Nonetheless, using the week number may sufficiently represent any temperature-based trends. Additionally, very few number of store-dept doubles showed good correlations relative to the factors, and consequently these trends would likely do very little to assist in predicting future sales; however, as stated above, even these week correlations may provide useful information if the extent of correlation is taken into account. The results of incorporating the regression-based trends into the prediction algorithm will be presented at the end of this Results section (after covering the average-based prediction methods).

Only Using "Weighted Average of Adjacent Weeks" Method:

As mentioned in the Methods section, we initially predicted a future sales value by averaging the historical data for that same week of the store-dept-week triplet of interest; however, historical data was not available for each store-dept-week triplets of the future weeks. Even after using a range of weeks (e.g. +/- 1 weeks from week of interest) to compute the average for a given week, we found that we were still unable to predict sales for all future weeks. The number of missing values is shown in Table 7 below.

	Predicting Average From Historical	Number of Missing store-dept-week	
Method	**Range**	**Holidays**	**Non-Holidays**
1	+/- 1 weeks	105	636
2	+/- 2 weeks	105	493

Table 7. Missing predictions when using a range of weeks for computing average.

As the table shows above, expanding the range to +/- 2 weeks (within the week of interest) allowed us to calculate average sales for more of the future weeks; however, some of the future weeks were still not "covered" by this method. Although these numbers of missing predictions are small as a percentage of all triplets (< 0.3%), and thus unlikely to significantly impact the overall score (i.e. SSE), we still felt that we needed to add additional prediction methods to ensure that all future weeks were covered.

Highest Priority Sales Prediction:

As mentioned in the Methods, an additional three prediction methods were devised in order to make sure that sales values could be predicted for all future weeks. The number of future store-dept-week triplets that were still missing predictions after adding each of the three methods is shown in Table 8 below. In this particular table, method #'s 3, 4, and 5 correspond to the three new methods.

	Predicting Average From Historical	Number of Missing store-dept-weekNum triplets	
Method #	**Range**	**Holidays**	**Non-Holidays**
2	+/- 2 weeks	105	493
3	Add Next: Stores of Same Type Average	31	95
4	Add Next: All Stores Average	30	3
5	Add Next: All Depts of Same Store	0	0

Table 8. Missing predictions after successively adding prediction methods to the algorithm.

The extent to which a prediction method is expected to be accurate is denoted by its ranking (i.e. most accurate method is at the top of the table). Each of the added methods reduces the number of missing predictions, until no missing predictions remain after the last method.

Weighted Average of Sales Prediction:

After implementing the above priority-based method, we thought that perhaps each of the four prediction methods may be able to contribute some useful information, even when a method that was expected to be more accurate was able to successfully predict a sales value. One could predict sales using all four methods, and then calculate a weighted average of these four predictions, where the weighting for each method would be proportional to its expected accuracy. A weighting scheme was assigned for the four methods and predictions were made. The results were submitted through the Kaggle interface, along with the predictions using the "Highest Priority Sales Prediction" method (outlined above). The results are shown in Table 9 below.

				Kaggle Results		
Submission	**Date**	**Prediction Method Type**	**Prediction Method Details**	**Score**	**Rank**	**Kaggle Percentile**
1	4/15/2016	Priority-Based	2, 3, 4, 5 order	3475	301	56.4%
2	4/15/2016	Weighted Average	50, 25, 15, 10 weights	3472	301	56.4%

Table 9. Kaggle results for priority-based and weighted-average prediction methods.

As shown in the table above, the weighted-average method did not yield a significant improvement in the Kaggle score.

Adding Regression Analysis:

As mentioned in the beginning of this Results section, very few store-dept doubles yielded a "good" correlation for the four factors; however, as explained in those results, the mediocre or even sub-mediocre correlations may still be able to contribute some knowledge to the prediction algorithm, if the extent of correlation is somehow accounted for in a weighting scheme. Figure 7 below shows the distribution of correlation values for the four different factors.

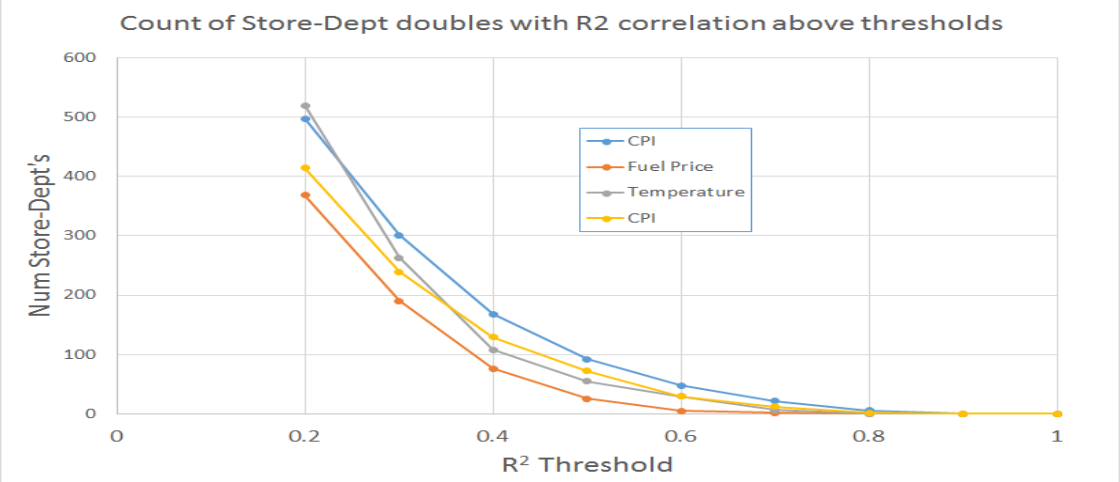

Figure 7. The number of store-dept doubles meeting various minimum R2 thresholds. Total of ~4,455 store-dept doubles.

As shown in the figure above, the number of store-dept doubles that are a above a certain threshold rises sharply as the cutoff threshold is decreased; however, if one is to use the trends of these less-favorable correlations, the weighting for the predicted sale should take into account the correlation of the trend line (as described in the Methods section).

This general approach for incorporating the linear-regression trends is described in the Methods. Each of the four sales predictions (one from each of the four factors) was added individually as a separate method, such that the "Weighted Average of Sales Prediction" method would have a total of eight methods (sales predictions from original four methods, plus sales predictions from the regression model of each of the four factors). The results are shown below with various combinations of weightings and maxRegressionWeight's.

			Kaggle Results		
Trial #	**Prediction Method Type**	**Prediction Methods Details**	**Score**	**Rank**	**Kaggle Percentile**
1	Sequential	2, 3, 4, 5 order	3475	301	56.4%
2	Weighted Average	50, 25, 15, 10 weights	3472	301	56.4%
3	Weighted Average + regression weights	40, 15, 5, 40 and maxRegressionWeight= 5	3934	356	48.5%
4	Weighted Average + regression weights	40, 15, 5, 5 and maxRegressionWeight= 20	3580	318	54.0%
5	Weighted Average + regression weights	50, 25, 10, 10 and maxRegressionWeight= 15	3627	328	52.5%

Table 10. Results for prior trials and the new algorithms that incorporat e regression-based predictions.

For trials that used weighted average prediction type (trials 2- 5), the first four values in the "Prediction Methods Details" indicate the weightings of the first four methods.

The score/rank for the Kaggle submissions did not appear to improve after incorporating this regression-based prediction method.

A different weighted approach:

Formula for the weighted window:

$$if\ i^{th}\ week\ sales = Null: then\ it\ is\ not\ accounted\ in\ the\ summations\ below.$$

$$Total\ Weighted\ sales\ =\ Current\ Week\ sales + \sum_{i=1}^{window\ range}\left(\frac{1}{2(i)} \times i^{th}_{weeksales}\right)$$

$$Total\ weight\ =\ 1 + \sum_{i=1}^{window\ range}\left(\frac{1}{2(i)}\right)$$

$$Weighted\ Sales\ Average\ =\ \frac{\text{Total Weighted sales}}{Total\ weight}$$

Results for various window lengths for 2 different methods					
Considered Range of Weeks Window ("+/-")	Method Type	Number of Missing store-dept WeekNum triplets		Kaggle	
		Holiday	Non-holiDay	Score	Rank
< 2	Average	105	645	3455	285
	Weighted Average	105	645	3430	277
< 3	Average	105	487	3533	314
	Weighted Average	105	487	3448	285
< 4	Average	105	397	3595	320
	Weighted Average	105	397	3467	299
< 5	Average	105	324	3584	318
	Weighted Average	105	324	3464	298
< 6	Average	75	278	3846	353
	Weighted Average	75	278	3394	270
< 7	Average	75	241	3886	355
	Weighted Average	75	241	3398	270

Table 11. Weighted window results for various lengths

Trend Analysis (Best of All):

Having observed that there is trend in the data we thought trend analysis would give a better results. So we changed the weighted model to give higher weights to the closer data from the past compared to data that is far (distance here is a measure of years). This turned out to be the best model of all, which made our position fall below top quartile. However, we agree this to be a naïve approach as we playing with the weights to get the best results. Having said that we also made a note on how to approach these kind of problems in the future. The solution is to perform trend analysis. In this is case it is bit hard to

detect the trends due to the scarcity of both data and given amount time so we have included that as future work. The table below shows the results for this analysis.

Weight Increments	Score	Rank
3	3121.170	209
4	3099.199	204
5	3084.850	198
6	3074.766	193
7	3067.417	191
8	3061.730	188
15	3042.449	183
21	3035.984	180
25	3033.395	178
29	3031.500	176
49	3023.898	150

Table 12: Trend Analysis 1

Observations:

Weighted Models

This Weighted model performance tends to keep increasing with increase in the window length (ranks highlighted red). This might be due to the decrease in missing values in presence of the weighted window.

It turned out that we got the best results with week window range < 7 in the weighted average model. Here we are assuming that this improvement is due to decrease in number missing predictions and due to the weighted approach that keeps the weights of the distant weeks low. In the above weights are assigned so as to have a better results. However, we hope minor adjustments in this formula can give better results.

Trend Analysis

This model outperforms all other models we have built, we believe this kind of approach suits this data sets well as all boils down to detecting the trends based on which we can detect the future.

7.Conclusions:

Three new prediction methods were added to the original "average of adjacent weeks" method in order to be able to make predictions for weeks that did not have sufficient historical data. Complexity was further added to the model by incorporating weighted sales predictions from the linear-regression trends. In general, the Kaggle rank/score worsened as the prediction algorithm was made more complex. Perhaps

this is indicative of some overfitting that is taking place. For example, given the large number of store-dept doubles, perhaps a significant number of "good" correlations formed by chance, rather than forming as a result of some underlying phenomenon that could be reliably used to predict sales for the train data. It was thought the effect of these false correlations would be mitigated by making the "maxRegressionWeight" small (relative to the weightings of the average-based methods, see Table 10 above).

Most of the store-dept-week triplets were covered by the "average of adjacent weeks" method, which is also expected to be the most accurate prediction method; therefore, it was not surprising that adding the additional prediction methods did little to improve the Kaggle score/rank. The best model that we got is the weighted averaging model which gave us a rank of 270 on kaggle

Future Work:

When calculating the correlation between a factor and normalized sales for each store-dept, we did not separate the holiday and non-holiday sales data. Perhaps considering these data points separately would yield more accurate correlations.

Rather than computing the average of stores with the same type, it may be useful to search for stores with similar sales patterns. For example, one could compute a "distance" between two stores with the following equation:

$$D(a,b) = \sum_{w=1}^{52} (S_{a,w} - S_{b,w})^2$$

Where D(a,b) = distance between store a and store b, and S(a,w) is the normalized sales value for store a during week w. The stores that are most similar to a particular store could be used in predicting the average sale for a week where sufficient historical data for that particular week was not available.

We did not investigate the trend between markdown and store sales for the store-dept doubles. Perhaps this information would be useful in predicting sales, especially during holidays when markdowns are often used as a means to spur shopping.

Trend Analysis:

As mentioned in the trend analysis section previously, we hope that a trend analysis is the best approach to work with this type of data sets. We like you to note that our trend analysis model is a bit biased but we do believe minor changes in this will give better results. We can improve this model by seeing trends by

calculating the percentage changes from the across the time axis for the given data and then based on the trend predict the future. Further we can also predict the sales for the stores with missing values by taking into consideration the stores that perform similarly or by mixing the weighted window model with the trend analysis..

8.Individual Tasks:

Note:

a. *All our code is available on GitHub. https://github.com/narayana1043/Data-Mining-Project--Walmart-Weekly-Forecast*
b. *Our codes works in anaconda environment,*
c. *We can make ourselves available if there are any issues*

Sam:

I implemented the four average-based prediction methods. I also implemented the algorithms (priority-based and weighted-average) for computing a single predicted sales value from these four prediction methods. I investigated the existence of correlations between the factors and the sales values of the store-dept doubles. I implemented the regression-based prediction algorithm. I implemented trend analysis for giving the more recent years a higher weighting when averaging historical weekly sales which gave better results comparatively. My code is available with this submission (folder "sams_src")

Veera:

I have implemented a Naive model that works similar to the averaging approach by Sam and performed analysis by altering the adjacent week window length for a range of < 2 to < 7. On observing that this average model failed degrades in performance as the neighboring weeks are treated equally, I performed a weighted average model that automatically assigns weights to its neighbors. In this approach I made sure, the weight of a closer week is given high weight than a far way week. This model turned out to work better. However, having noticed there is some missing data which are influencing the results, I have tried to fill this missing data by taking the departments that perform similar into consideration to fill the null values but did not notice significant improvement.

I tried to implement trend analysis with my other mates and it turned out to improve our results. I felt much can be done on this if we can effectively catch the trend pattern over the years. I feel domain

knowledge also comes into play after detecting the trend and even that influence the predictions. My code is submitted along with others as we have collaborated on Git hub.

Udit:

I have taken up Veera's weighting model and further modified it by changing the weights and did analysis for various window lengths. It is observed that the weighted model performs well up to a window of range of ± 15 and then starts to degrade in performance from then, from the results we have intuition that this is due to overfitting of the model. Experimented with trend analysis to improve the results and noticed that we can only improve the trend if have more data. Catching the trend is difficult in situations where there are only 2 data points and these situations are encountered in this data. Mapped the nulls values to the closer stores values. Closer stores are detected based on their performance as per sales, store type etc.

9. References:

Papers:

1. Djukanovic, Suzana; Milic, Milan; Vuckovic, Milos. Data Analysis And Sales Prediction In Retail Business. Faculty of Organizational Sciences, University of Belgrade, Belgrade, Serbia suzy.89@live.com, milicmil@gmail.com, mvuckovic70@yahoo.com.
2. Gicheva, Dora; Hastings, Justine; Villas-Boas, Sofia; Investigating Income Effects in Scanner Data: Do Gasoline Prices Affect Grocery Purchases? American Economic Review: Papers & Proceedings. 100 (May 2010): pp. 480-484.
3. Murray, Kyle B.; DiMuro, Fabrizio; Finn, Adam; Popkowski, Peter L.; The effect of weather on consumer spending. Journal of Retailing and Consumer Services. 17 (2010) pp. 512-520.
4. Ma, Yu; Ailawadi, Kusum L.; Gauri, Dinesh K.; Grewal, Dhruv. An Empirical Investigation of the Impact of Gasoline Prices on Grocery Shopping Behavior. Journal of Marketing. Vol. 75 (March 2011), 18-35.
5. Zhang, Peter G.; Qi, Min; Neural network forecasting for seasonal and trend time series. European Journal of Operational Research. 160 (2005) pp. 501 - 514.
6. Divakar, Suresh; Ratchford, Brian T.; Shankar, Venkatesh; (2005) Practice Prize Article—CHAN4CAST: A Multichannel, Multiregion Sales Forecasting Model and Decision Support System for Consumer Packaged Goods. Marketing Science 24(3):334-350. http://dx.doi.org/10.1287/mksc.1050.0135.

Websites:

7. Kaggle: https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting
8. Kaggle forum: https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting/forums/t/8055/6-bad-models-make-1-good-model-power-of-ensemble-learning

Prezi: https://prezi.com/adiaelgum2lq/copy-of-walmart-sales-analysis