Capstone Project I Final Report: Walmart Sales in Extreme Weather Days

Zexi Yu

January 25, 2018

Abstract

I will write the abstract when I finish editing this report.

1 Introduction

This is the final report for my first capstone project in Springboard based on my milestone report. Originally from a kaggle competition, this project targets to build a system that will predict item sales in extreme weather events for Walmart using machine learning techniques. All the coding files for this project can be found in [1].

This report will include the following sections:

- Problem Description: In this section, I will define the problem that needs to be solved in this capstone project, point out the potential client, and describe how the result from this capstone project will benefit them.
- Dataset Description: In this section, I will give the source of the datasets used in this capstone project, explaining the datasets' content, as well as the meaning for each column in the datasets.
- Data Wrangling: In order to perform machine learning algorithms, the data must be merged and cleaned. This process is called data wrangling. This subsection describes all the data wrangling steps used in this project.
- Feature Selection: Some algorithms in machine learning, like linear regression, require the features to be linear independent in order to have good performance. Also, some additional features are added in this section to enhance the algorithm performance.
- Initial Observation: In this section, I will show the initial observation of the dataset in order to try to identify useful features for the learning task.
- Learning Method: In this section, I will give the details for the learning method, including:

  - Overall Structure: This subsection describes the general structure of the learning process.
  - Algorithms: This section describes the machine learning techniques used in the project. It includes subsections for each algorithm: SVM, decision tree & random forest, and neural network.

- Result and Performance: The prediction results on testing set are submitted to the kaggle website. The website then gives the root mean squared logarithmic error (RMSLE) and ranking for each result submitted. Then, I evaluate the performance of each algorithm based on its RMSLE score.
- Conclusion: This section concludes the report by reinstating the learning object, the learning algorithms, and their performance.

This project uses python as the coding language. I use Lyx[2] as editing tool for this report.

2 Problem Description

2.1 Problem Definition

As stated above, this project comes from a kaggle competition. Walmart wants to know the correlation between weather condition and its sales record. More specifically, Walmart wants a model that can "accurately predict the sales of 111 potentially weather-sensitive products (like umbrellas, bread, and milk) around the time of major weather events at 45 of their retail locations".

To this end, in this capstone project I would like to accomplish the following objects:

1. A model that will predict product sales in major weather events
2. Evaluate how the weather condition affect the sales

2.2 Potential Benefit for Client

According to the description of the kaggle competition, the model will help Walmart in the following aspects:

- It will help Walmart better manage its inventory, and keep its customers out of rain.
- It will help Walmart better evaluate the effectiveness of its current management against weather event

3 Dataset Description

The data can be found at kaggle in this link: https://www.kaggle.com/c/walmart-recruiting-sales-in-stormy-weather

Four datasets are provided:

- train.csv: a training sales dataset with date, store number, item number and its corresponding sales record.
- test.csv: a testing sales dataset with only date, store number, and item number. The prediction on this dataset is then submitted to kaggle for performance evaluation.
- weather.csv: a weather dataset recording all weather conditions with weather station number, rainfall, wind, temperature, and many other features.
- key.csv: a correlation dataset linking store with its nearest weather station.

After merging those datasets into one dataset, we can see the following columns appeared in the final dataset:

- date: year-month-day format
- store_nbr: Walmart store number
- item_nbr: item number, 117 of them, each number indicates one item, we do not have further information about what precise item would that be.
- units: number of items sold on that day
- station_nbr: weather station number
- tmax, tmin, tavg, depart, dewpoint, wetbulb: temperature max, min, average, departure from normal, average dew point, average wet bulb. in Fahrenheit
- sunrise, sunset: time of sunrise and sunset
- codesum: special code in letters indicating the weather conditions of that day, such as RA as rain, SN as snowing
- snowfall: snow/ice on the ground in inches at 1200 UTC
- preciptotal: 24-hour snowdrop in inches
- stnpressure: air pressure
- sealevel: in meters
- resultspeed: resultant wind speed, miles per hour
- resultdir: resultant wind direction, in degrees

- avgspeed: average wind speed, miles per hour

I will talk about data merging in more details in subsection data wrangling of section learning method.

4 Data Wrangling

As stated above, the data must be merged, interpolated and cleaned before passing to machine learning algorithms.

4.1 Data Merge

The goal of this project is to predict item sales based on weather data. The sales record is stored in train.csv, and weather record is stored in weather.csv. Data from key.csv indicates the corresponding relationship between the store and weather station.

Naturally, the first step for this project would be merging the information from the datasets together based on the information from key.csv file. Given a sales record, the program looks up its corresponding weather station number based on its store number, and append the weather information from the weather station on that day to the sales record.

The code for this part is in preprocessing.py file.

4.2 Data Interpolation

Since the object is predicting sales record in near weather events, it is necessary to highlight data of those events. A weather event is defined as rainy days with 1 inch or more rainfall, or snowy days with 2 inches or more snowfall.

Some content in the data needs to be pre-processed before passing to next stage. For example, in the preciptotal, the record "T" means that there is trace of rainfall on the ground, but the actual rainfall is too small to be measurable. Because this explaination, I replace all "T" with 0.

There are missing data in the numeric features as well. In order to process them, I first order them by store, item, and date, and perform linear interpolation on the result. For such interpolation, pandas provide a convenient function: pd.interpolate(). This function uses linear interpolate by default.

This part of code is also in preprocessing.py file.

4.3 Outliers

Two outliers are found at sales record: the highest and second highest sales record for item 5 is 5568 and 3369, while the third place is only 500. So I excluded these sales out of learning period. Note that the classifiers I use in this project are robust against outliers themselves, but I still exclude them for safety.

This part of code is also in preprocessing.py file.

5 Feature Selection

As stated above, for some algorithms like linear regression or SVM, it is necessary to rule out linear dependent features to ensure good performance. Therefore, I carry out one more step for this. The technique I use is called Variance Inflation Factors (VIF).

5.1 VIF

Some features have strong (linear) correlations between them, and that may affect the performance of some learning methods. So it is best to identify those correlations using variance inflation factor, also known as VIF.

For simplicity, I divided the features into several groups, and perform VIF on each group independently. Within those groups, one feature with VIF score higher than 3.0 is excluded, and VIF is performed again until no score is over 3.0. The groups I divided are:

- temperature related features: tmax, tmin, tavg, depart, dewpoint, wetbulb. All features but tavg are excluded
- rainfall related features: snowfall, preciptotal. The feature snowfall is excluded
- wind related features: resultspeed, resultdir, avgspeed. The features resultspeed and resultdir are excluded.
- other numeric features: stnpressure, sealevel. These features are constant values for same store, therefore, their information is not very meaningful given that store number is already in the features. Therefore, I exclude them without performing VIF.

5.2 Feature Creation

After actually performing learning models, I find the following features useful:

- Sales average for 7 days before the day.
- Sales average for 7 days after the day.
- Markers indicating if the day is within 3 days before a weather event, or within 3 days after a weather event.
- Weekday, month and year, since in my initial observation I find weekday, month and year have good relations with sales record.

Of course, in order to use categorical features such as weekday, month, year, store number, and item number, we need to transform them into dummy/indicator variables. Pandas provide this function by pandas.get_dummies(). However, I later find that the data with all the dummy variables is too large for my laptop to handle, so I propose an alternative route in subsection 7.1.

For simplicity, I won't list in detail how I reach those conclusions here. If you are interested, please see my data storytelling report and inferential statistics report at the following address: [3]. I will give a brief summary of these two report as well as highlighting some interesting points in the following section, however.

It is also worth noting in advance that, as you will see in the result section, the performance of SVM is actually better if all the numeric weather features are excluded and only keep the indicators for extreme weather, rain, and snow. In the result section, you will see the performance difference between models with numeric weather features and without them.

6 Initial Observation

Initial observation is important. It can provide me direct impression on how the dataset looks like in terms of how sales change with date, month, year, rainfall and other features, and how these features interact with each other. This is helpful, for it will provide some initial ideas about which features might be useful for the coming learning task, and which learning structure might be useful in order to enhance learning efficiency.

In order to maintain the compact structure of this report, I do not put all my observation here. If you are interested in a full version, you can visit my data storytelling reports 1 & 2 and my inferential statistic report. They are available in the following folder: [3]. From there you can see all my observation and how I test them using hypothesis testing. In this section, my plan is to directly jump to the conclusions I made, and only show part of my hypothesis testing procedure on some interesting points.

Here are the conclusions that still stand after inferential statistics test:

- The sales pattern of normal days is different from the sales pattern of extreme weather period.

- Item 5 sells best during extreme weather period.

- Features that indicating whether the day is a normal day, a day before a bad weather or a day after a bad weather could be useful since observation suggests that even when it is a sunny day, the sales record close to bad weather still differ from normal case.

- For item 5: Year does affect the sales record.

- For item 5: Weekday does affect the sales record.

- For item 5: Rainfall/Snowfall: Since we can indicate major events by event marker (see the part 1 report), whether rain/snow present is useful in predicting item 5 sales, but the amount of rainfall/snowfall does not matter that much.

- For item 5: Temperature: During normal days, The temperature and selling record surely has a correlation in general case. However, this correlation is not linear. Also, the correlation between temperature and selling record during major weather events is less stronger than the one during normal days.

You may notice I paid extra interest towards item 5. The reason for that is because there are 110 items in this dataset, and it is not practical to analyze them one by one. Instead, it would be more efficient to discuss some import items. From the figure 1, I think item 5 is the most import one because:

1. It is the best selling item during extreme weather events. Given our learning target is predicting item sales during extreme weather, the prediction error on item 5 is more likely to be the dominant part in the prediction error.

2. The sales of item 5 changes significantly between normal days and extreme weather event days. Therefore, performing exploratory data analysis (EDA) on this item is more likely to expose the correlation between sales and features, and the interactions between features themselves.

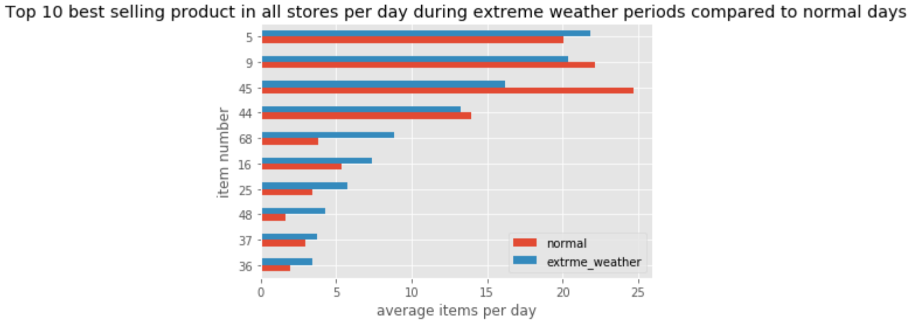

Figure 1: Top 10 hot sales item

As I stated, for simplicity I will only show part of my hypothesis test procedure in this report.

As stated above, the time features such as weekday are helpful towards learning task, for they have impacts on sales. Here I will show one point and how I evaluate it:

- For item 5: Weekday does affect the sales record.

The figure 2 shows what I observed:

I evaluate this claim by calculating the 99% confidence interval of average sales per weekday using bootstrap technique: I resampled the data using bootstrap sampling for 100000 times, calculate the daily average, and come up with 99% confidence interval. There is no interval overlapping between weekends

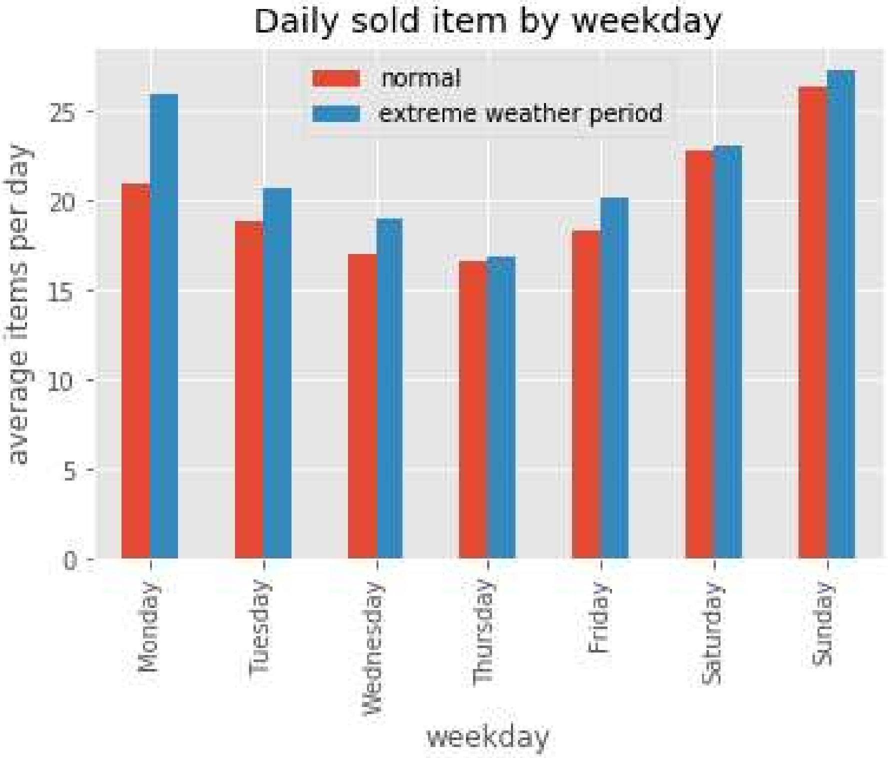

Figure 2: Item 5 sales on each weekday

plus Monday (Saturday, Sunday and Monday) and other weekdays (Tuesday, Wednesday, Thursday and Friday), so one can say that the selling record on weekdays do differ from one another.

Of course, I paid attention to other item as well, here is a claim I made in my inferential statistics report to support the point “The sales pattern of normal days is different from the sales pattern of extreme weather period.”:

- In extreme weather days, the sales record for item 45 dropped significantly.

This claim can be easily observed in figure 1. The daily average sales record for item 45 dropped from 25 in normal days to 16 in extreme weather days. I evaluate this claim by null hypothesis test. The null hypothesis is: “The item 45 sales record in extreme weather events is actually the same as the item 45 sales record in normal days". Then, I use permutation to calculate the p-value. The permutation is performed 100000 times, and the difference between permuted normal data and permuted extreme weather data never surpass the actual value. This means that the p-value is very close to 0, indicating that the null hypothesis does not stand. Therefore, in extreme weather days, the sales record for item 45 did drop.

For other evaluation and observation, please check my reports [3]. But this is the recommendation I made towards features and regressors based on my initial observation:

- Because of the presence of non-linear correlation, linear models, such as linear regression and SVM may not work well.
- Because of the threshold effect of certain features (such as people just stop shopping when they see a tiny bit of rain or snow during normal days), Decision tree and Random Forests may work well.
- There might be a logical correlation between features indicating whether the day is a normal day, a day before a bad weather or a day after a bad weather. Because of this, using neural network on this project might be promising.

7 Learning Method

Predicting the daily sales is a regression problem. But I need to choose appropriate regressors to make it work. Based on the recommendation from last section, the following regressors seem to be promising: support vector machine (SVM), random forest & decision tree, and neural network. We will discuss them in further detail in this section, but there is one problem before we come to that. As mentioned in subsection 5.2, the memory of my laptop (8GB) is too small to train one regressor on the whole data with all the dummy variables/features. Therefore, I need to use an alternative route to train the regressor. The following subsection describes what I propose.

7.1 Overall Structure

Below is a graph showing my workaround for not having enough memory, the general workflow of the learning method.

The basic idea for the learning method is looping over all possible combinations of store and item, and train a regressor for each unique combination. The reason for choosing this structure is:

- As stated, building a learning model for all the data requires a large memory that beyond my laptop's capability.
- There are quite a few store/item combinations with 0 sales record in the training set. For example, none of item 1 is actually sold in store 1. The best explanation for this is that for store 1, there is no storage of item 1 during the time period recorded in the dataset. For this reason, it is more appropriate to build a regressor for each store/item combination, and just predict everything to 0 on the store/item combinations with 0 sales in training dataset. This provide higher efficiency and further shorten the training time.

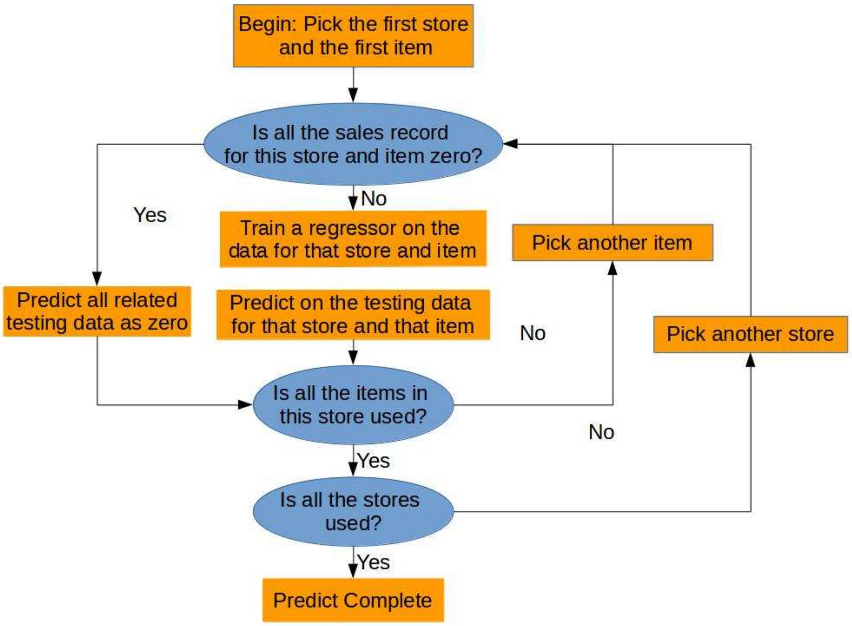
Begin: Pick the first store and the first item
Is all the sales record for this store and item zero?
No
Yes
Train a regressor on the data for that store and item
Pick another item
Predict all related testing data as zero
Predict on the testing data for that store and that item
No
Pick another store
Is all the items in this store used?
No
Yes
Is all the stores used?
Yes
Predict Complete

Figure 3: General structure for regressor training

7.2 Algorithms

This subsection describes the algorithms I choose for the regressor. I choose four algorithms in this case: SVM, Decision Tree, Random Forest, and Neural Network.

7.2.1 SVM

Linear models are the first regressor class I consider to choose, for they cost relatively less computational power, and usually have nice performance if the features and the target are linearly related. Moreover, the results given by linear models are usually easy to explain, the importance of two features can be easily compared by their corresponding coefficient parameters.

In order for SVM to perform better, I use log transformation on the sales record. This method is common in linear models, for it concentrates the data towards normal distribution.

For this project I use the SVR (SVM Regressor) in python scikit-learn package.

7.2.2 Decision Tree & Random Forest

With so many binary indicators and dummy features in this project (such as dummy features for year, month, weekday), it is worth trying decision tree. These two method operate by dividing data based on applying thresholds to features. This behavior is quite similar to the dummy features and indicators, for these features only have 0 or 1 as values. Therefore, I think these two learning method can be a good match for this project.

For this project I use the Decision Tree and Random Forest Regressor in python scikit-learn package.

7.2.3 Neural Network

Neural Network is without any doubt a powerful tool for a learning task with a non-linear structure. It can learn logistic operations like AND, OR and XOR. In my observations I find there could be logistical relations between features. Although the claim is finally rejected (see my inferential statistics report for details), it is still worth trying the neural network.

For this project I use the Deep Neural Network in TensorFlow package.

7.3 The Part I Left Out

Maybe you have noticed which part I missed: parameter tuning. This part means optimize the parameters used by regressors, so that they have better performances than using default parameters. This part is quite important for neural network and have some impact on the performance of SVM, decision tree and random forest. I did not apply this part, for it takes too long for the code to run even in SVM case. Instead, I comment the code so that I can simply uncomment this part of code and run if I want to apply it in the future.

8 Results and Performance

The result is shown in the figure below. The score used here is Root Mean Squared Logarithmic Error (RMSLE). It is also used for ranking in the competition. The numbers listed here come directly from the submission on the Kaggle website, so that they are comparable versus other competitors. The submission page and leaderboard can be found here: [4].

Using its default kernel "rbf", SVM gives better performance than the rest of the algorithm. The performance of SVM without numeric weather feature is better than the one with weather feature. Decision tree and random forest give slightly worse results, and deep neural network does not work so well on this project. I expect this result because there is no parameter tuning for neural netowrk.

The best result given by SVM put me in rank 80/485 among all the participants, and only has 0.009 difference from the top submission. The leaderboard for this competition can be found at here: [4]. If you want to confirm the score,

Algorithm Name	File name	RMSLE score	Training Note
Base Submission	result_0_new.csv	0.11701	result by a simple interpolation
SVM version 1	result_fastsvm.csv_v3	0.12024	trained with numeric weather features
SVM version 2	result_fastsvm.csv_v4	0.10286	trained without numeric weather features
SVM version 3	result_fastsvm.csv_v5	0.10841	trained with linear kernel
Decision Tree	result_tree_ver2.csv	0.11461	None
Random Forest	result_rf.csv	0.13851	None
Deep Neural Network	result_dnn.csv_ver2	0.14245	None

Table 1: Results Table

all you need to do is downloading the result file, and making a submission on this link.

So now we have a series of nice predictors in hand, then what is the most dominant features in this problem? In other words, which factor is most dominant in deciding the sales record for the item? We interpolate this by averaging the coefficients of the SVM regressors using linear kernels, and select top 10 features using their absolute coefficient value (for other kernels do not provide coefficients):

Feature Name	Coefficient Value	Note
Before_Sales	0.556280	Average sales of one week before the day
After_Sales	0.526110	Average sales of one week after the day
weekday_6	0.111950	dummy feature for Sunday
weekday_3	-0.071970	dummy feature for Thursday
weekday_5	0.067471	dummy feature for Saturday
Condition	0.066983	indicator feature marking the weather event day
weekday_2	-0.063852	dummy feature for Wednesday
weekday_1	-0.039919	dummy feature for Tuesday
year_2014	-0.038446	dummy feature for year 2014
SN	0.032191	indicator feature marking a snow day

Table 2: Coef Table

We can see that the most dominant feature for the prediction is the average sales record of one week before the day and the average sales record of one week after the day. Besides that, weekdays seems to be more dominant than other features. Only two weather-related feature make the top 10. One is extreme weather indicator, another is snow indicator.

But is that really the whole picture? I did this by average over all items, there is still possibility for one feature play a dominant role in sales record for a single item, while the same feature does not matter much for other items. In order to see if this is the case, I list the top 5 coefficients for item 5, 9, and 45, the 3 top sellers in the dataset:

Feature Name	Coefficient Value	Note
weekday_6	15.224512	dummy feature for Sunday
After_Sales	13.548483	Average sales of one week after the day
Before_Sales	13.403635	Average sales of one week before the day
weekday_3	-8.166847	dummy feature for Wednesday
weekday_2	-6.777929	dummy feature for Tuesday

Table 3: Top 5 Coef Table For Item 5

Feature Name	Coefficient Value	Note
weekday_6	61.013008	dummy feature for Sunday
year_2014	-39.440597	Average sales of one week after the day
weekday_3	-34.352941	Average sales of one week before the day
year_2012	32.930745	dummy feature for Wednesday
weekday_2	-31.607478	dummy feature for Tuesday

Table 4: Top 5 Coef Table For Item 9

Feature Name	Coefficient Value	Note
year_2014	-118.540682	dummy feature for year 2014
year_2012	97.290682	dummy feature for year 2012
weekday_6	88.562500	dummy feature for Sunday
weekday_3	-67.062500	dummy feature for Wednesday
weekday_5	48.000000	dummy feature for Saturday

Table 5: Top 5 Coef Table For Item 45

It seems that the conclusion holds: The weather for item does not have much influence on item sales, even on the item 45. It is a bit surprising to me since the sales of item 45 dropped by almost 30% during extreme weather days. But, the result is the result.

However, it does not mean that the weekday are always the most dominant features. Take item 51 for example:

Feature Name	Coefficient Value	Note
After_Sales	1.773143	Average sales of one week after the day
month_12	1.227078	dummy feature for December
Before_Sales	1.046728	Average sales of one week before the day
SN	0.533850	indicator feature marking a snow day
month_1	0.227107	dummy feature for January

Table 6: Top 5 Coef Table For Item 51

This item make more sales in December and January, and on a snowy day. This could be something related to low temperature or winter. For this item, weekday no longer play an important role in predicting item sales, all 12 month indicators have higher coefficient than weekdays. Therefore, it is important to analyze each item independently.

9 Conclusion

In this project, I give a model that will predict product sales in major weather events. The model scores 80 out of 405 in the competition. Using Root Mean Squared Logarithmic Error (RMSLE) as evaluation score, it only has 0.009 difference from the best performing submission. Such model will help Walmart better manage their storage, and keep their valued customer out of rain. This model will also help them evaluating their current extreme weather management system. Besides, I look into which features might be more important for predicting sales.

If one wants to predict item sales in general, It turns out that the weather factor may not be that important after all: The two most dominant features for predicting sales in a given day are sales trending (sales average near the day) followed by what day it is in the week. This indicates that most items do not get affected in extreme weather.

However, taking a deeper look into the result, While weekday and average week sales remain to be very influential on item sales, the impact of them are different from item to item. For example, weekdays and years seems to be more important for item 9 sales. Therefore it is worth evaluating the result on item basis.

Based on my prediction result, my recommendation towards Walmart is:

- In general, sales record for past week and the weekday are the most dominant features for predicting one item sales in a given day.

- However the impacts are different from item to item, so it is wise to manage item storage independently. For items related to seasons, month and snowing might be a better indicator than weekday.

References

[1] "Project Code on Github," https://github.com/ZackCode/Capstone1_Walmert_Sales/tree/master/recruiting-competition-practice/code, accessed: 2017-11-07.

[2] "Lyx Home Page," https://www.lyx.org/, accessed: 2017-11-07.

[3] "Project Page on Github," https://github.com/ZackCode/Capstone1_Walmert_Sales/tree/master/recruiting-competition-practice/reports, accessed: 2017-11-07.

[4] "Kaggle Leaderboard," https://www.kaggle.com/c/walmart-recruiting-sales-in-stormy-weather/leaderboard, accessed: 2017-11-07.