Stormy Weather, Stormy Sales

Jonathan Lu and Kara Wong

12/13/2019

Introduction

While we can't control the weather, we can control how we react to it.

Based on a survey conducted by a mobile testing firm, SOASTA, checking the weather is the first thing 45% of participants do in the morning [1]. The weather is the second most commonly first-checked item in the morning, after checking emails. Weather plays an integral part of society, as it affects our daily lives. It determines how we dress, commute, eat, and feel. For example, if we see that a rainstorm is approaching, we would most likely choose to stay indoors rather than going out to run errands. This extreme influence that weather has on us is reflected in the changes in the U.S. gdp numbers. According to the Atomospheric and Environmental Research Center, thirty percent of U.S. gross domestic product is affected by weather one way or the other [2].

As part of an existing drive to be able to predict consumer behaviors, weather is just another factor which retailers want to consider. There are plenty of advantages to analyzing the effect weather has on a business. One of the largest issues which businesses face today are largely inventory based as floor space is expensive. By anticipating and reacting to weather-driven demand, businesses would be better able to maintian necessary stock during times of need, while other times, they would be able to use the space in the store for other, more profitable items. While this knowledge would be useful across all stores in the US, this would have a far larger effect in stores in metropolitan areas as the floor space is worth a lot more per square foot. Being able to anticipate trends in sales would also allow businesses to work around logistic issues such as transportation of goods. While this benefit ties in to the main one of understanding the necessary stock, optimizing logistic issues surounding transportation of goods between locations will also help businesses lower their operating costs and overhead.

While the idea behind an analysis such as this seem to be extremely useful, actual applications of such a model are extremely limited. When considering models for this data, there is a clear trade of between interpretability and accuracy, the less interpretable, the more accurate. Since this analysis is focused mostly on linear models, the interpretability of the coefficients the most useful aspect to this analysis. This is because while predicting how much you should've stocked is relatively useful information, using models such as the ones explored below to predict how much you should stock will be relatively fruitless. This is because in order to "predict" the future using these models, you will need to have access to accurate weather data in the future. Usually weather data is relatively accurate for one-week out forecasts [3], however, through experience, everyone knows that the weather forecast can change from day to day. This level of uncertainty leads to uncertainties in the model, which in turn, makes them less reliable.

The data used in this analysis was provided by Walmart as part of a recruiting event on Kaggle [4]. On Kaggle, they provided training and testing datasets. In addition to those files, they also provided a key to map stores to stations and a collection of relevant weather station data from the NOAA. The data spans 45 different Walmart locations, 20 different weather stations, and over 2 years of data. The objective that

Our objective is to predict the amount of units sold for each product with regards to major weather events.

[1] Morning Routine: What's the First Thing You Do When You Wake Up?

[2] Retail and Supply Chain

[3] Weather Prediction Accuracy

[4] Kaggle: Walmart Recruiting II: Sales in Stormy WEather

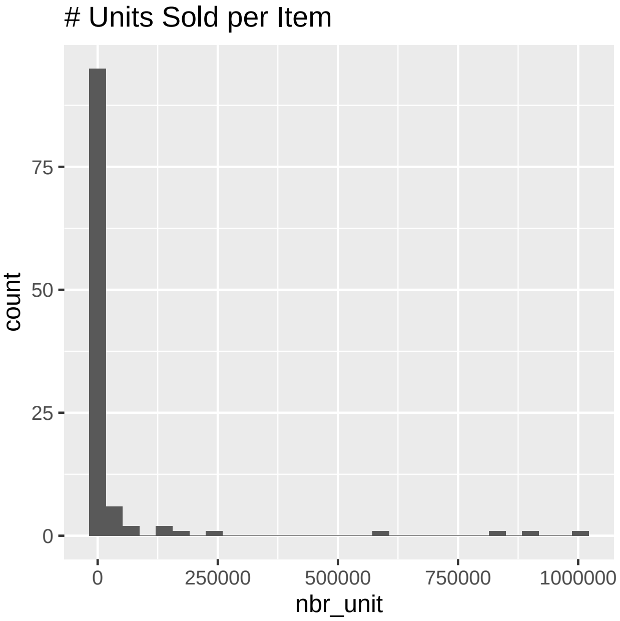
Table 1: Table: Summary of Number of Units Sold per Item Type

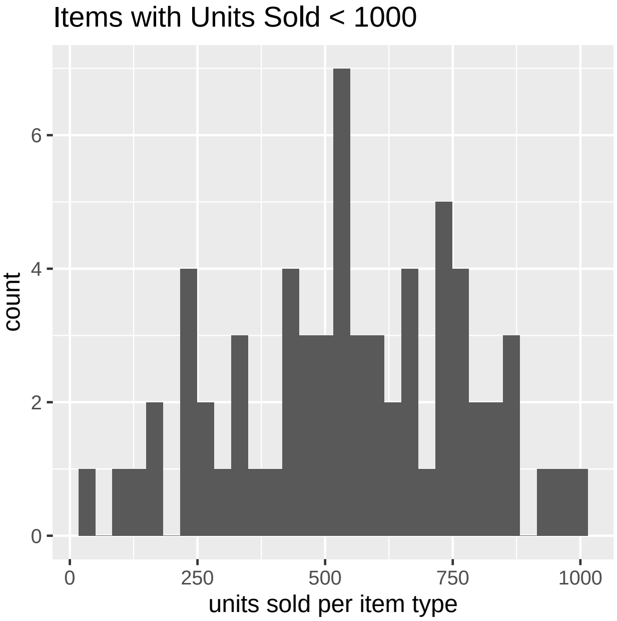
Minimum	First Quartile	Median	Mean	Third Quartile	Maximum
31	526	781	41054	2845	1005111

WHAT DOES THE DATA LOOK LIKE? DESCRIBE BACKGROUND INFO. WHAT MODELS DO WE USE?

Exploratory Data Analysis

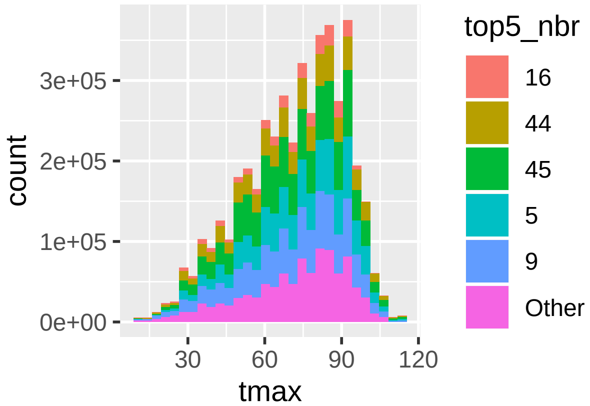
Our first observation when looking into our dataset is that our dataset has a lot of items with low sales. In fact, out of the 111 items, 59% have less than 1,000 sales and 24% have less than 500 sales. On the other hand, the top-five highest selling items all have over 10,000 units sold. Below, we have provided summary statistics for the number of units sold per item type, as well as a graphical representation.

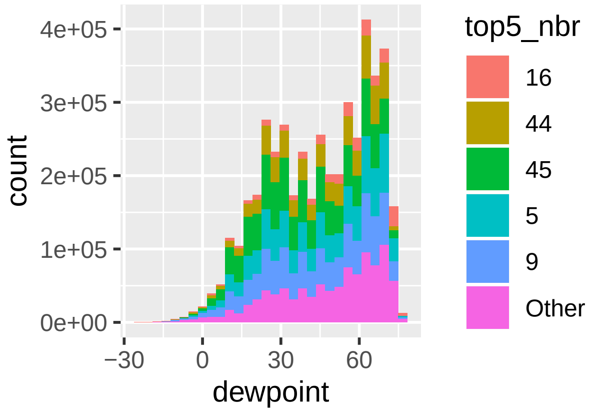

# Units Sold per Item

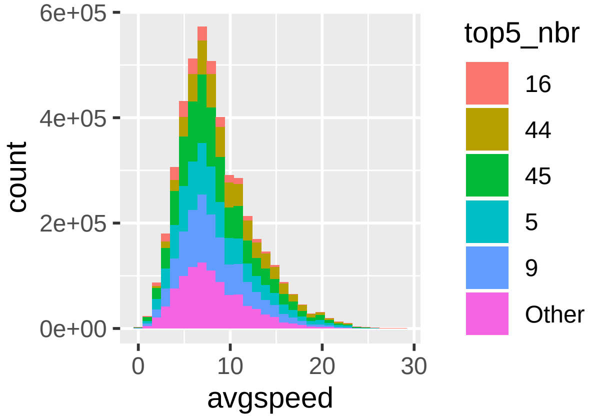

Items with Units Sold < 1000

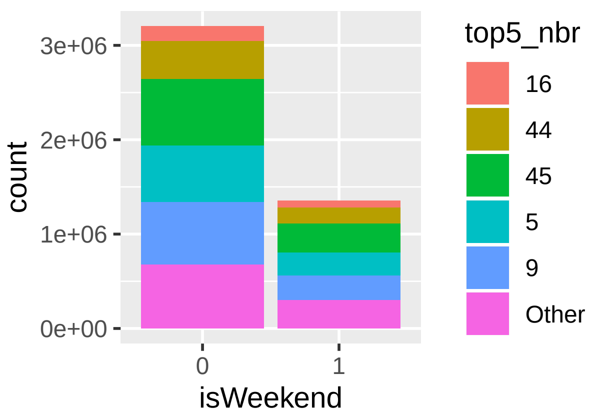
An important thing to note about our dataset is that it does not capture the discrepancies between inventory and demand. The number of units sold may show 0, but this may not necessarily mean that the item was not in demand, as it could just simply have been out of stock or discontinued. This may be an issue when it comes to making predictions.

We then analyze the amount of units sold of these top-five selling items based on the weather-based metrics provided (maximum/min temperature, dewpoint, etc.) The five highest selling items from highest to lowest are items: 45, 9, 5, 44, and 16. From the following plots, we can see that more purchases are made when it's hot and humid, when it's not too windy, and on weekdays. From the plots below, we can also see that there isn't one item type that is driving sales based on these weather metrics, as it seems that all item types' sales follow a similar distribution. More of these plots can be found in the Appendix.

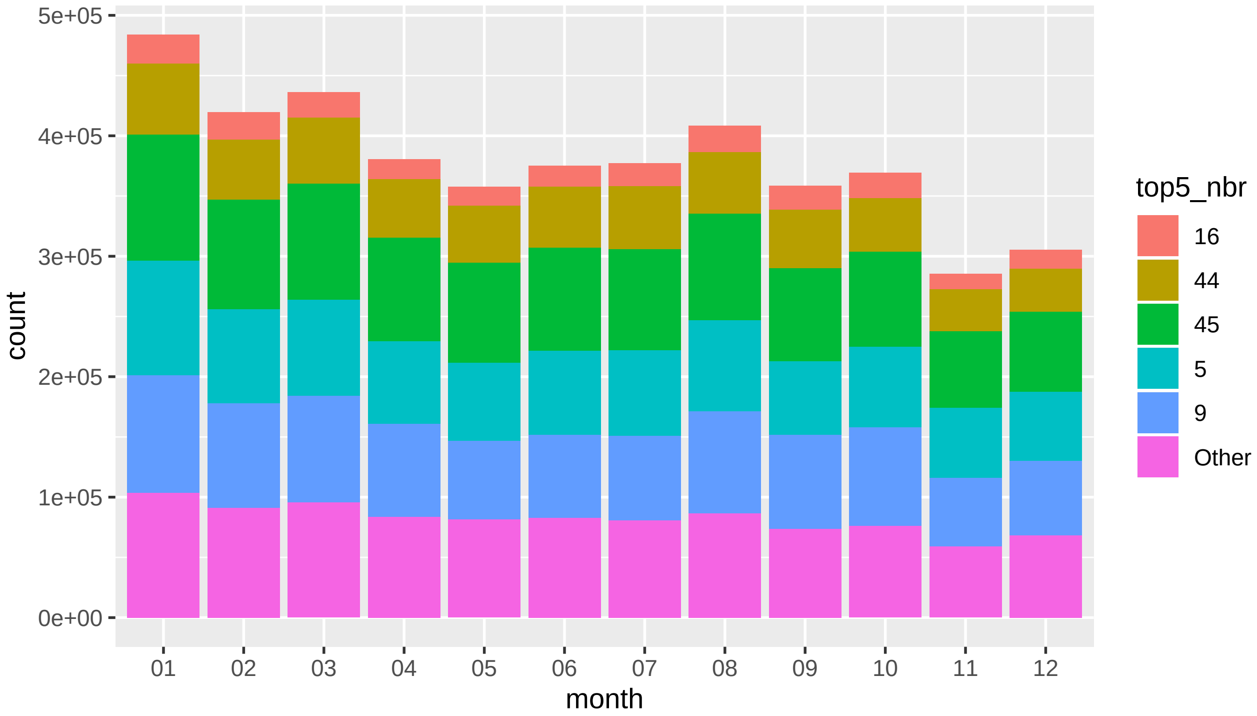




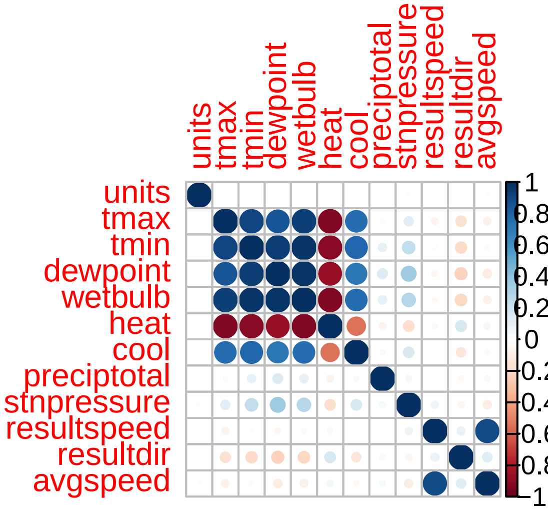




In addition to comparing the amount of sales on weekdays versus weekends, we also looked to see if there were any seasonal differences that could be driving sales. We extracted the month from `date` and provided a graph below that shows how there seems to be a downward trend in sales starting from January.



We've also examined correlations between our variables, and can see that many are correlated. From this, we can already anticipate linear regression model selection.



Linear Regression Model

Before we could build our first linear model, we had to stitch the weather data to the original dataset provided by Walmart. The main bulk of the work when it came to this process was cleaning the weather data to a point where it could be added together.

At an initial glance, we could see that the weather data was extremely incomplete (see Figure 8.) When aggregated to a station level, every single station had some sort of missing data. The variables that were missing the most data were depart, sunrise, sunset, and snowfall. Each of these variables had over 50% of their total data across all stations missing. With this many missing observations, we decided that it would be extremely unreasonable to try and impute data for that many missing observations as we were neither knowledgeable enough about weather, nor knowledgeable enough about the stations themselves to be able to classify these missing observations as anything other than "structurally missing". Therefore, since there was nothing we could do on our own to make these variables complete, we chose to remove them from the data. A variable which was close to the cusp in terms of being "un-imputable" was sea level. However, we quickly noted that sea level was pretty much identical to stnpressure (station pressure) as they both measured approximately the same thing. Since these two variables are pretty much perfectly collinear, we chose to remove sea level as there were far more observations for station pressure. Another variable we removed due to multicollinearity is tavg, or the average temperature of the day. When we looked into it, the value is calculated as the average of the minimum temperature and the maximum temperature. We chose to remove the average because we believed that the more extreme weather experienced, which would be captured better by min and max temperature, the more likely people are to buy weather related gear.

While removing a couple of variables definitely improved on the amount of missing data remaining, almost every single variable and station combination had some number of missing data. However, the amount of data missing at this point was extremely trivial ranging from ~ 0.5% of each station's data. Given that very few observations were missing, we felt relatively comfortable imputing a four day moving average for the missing observations. While this method cleaned up most of the variables for all of the stations except one. Once we aggregated the data to a station level, it became clear that station five had an un-imputable amount of data. Usually in these circumstances, we would remove the station completely, however, since we needed to have matching data for every store in Walmart's initial dataset we had to come up with a way to impute multiple values. To do this, we aggregated the data to a day level and imputed the mean of all the other station's observations at that day.

Before we finish with our weather data, we decided to create additional variables off of the weather data. When looking over the variables provided, one really stood out and that was date. We thought that it would be incredibly naive to think that the actual value of the date would have some predictive power over the number of units sold so instead; we constructed two categorical variables to suplant the use of the date. We decided that whether or not the day was a weekend and what season the day would fall under would be far more predictive as it narrows down the date to certain features related to it. While we thought that seasons might be really interesting as it sort of has to do with the weather, we were a bit worried about whether or not the change in seasons would already be reflected in the temperature data. We ultimately decided to leave it in as some locations such as California, barely have temperature change. We also changed codeSum from a set of strings representing weather events to a categorical variable representing whether or not some weather event happened on the day.

Joining the weather data to the Walmart data was extremely straightforward as all it required was two left joins. The first was to match a station to each store using the provided key. The last join was on the station number and the date, thus giving us a fully merged data set.

When it comes to building the first model, we wanted to keep in as many variables as we could, however, we decided to remove date and station number from model. We chose to remove date because we believed that the actual date would have little to no impact on whether or not someone buys something and instead chose to use our created variables. We also chose to remove station number because the dummy variable that are created to represent each station is perfectly collinear with a certain set of store numbers that the station maps to. The code for the model can be found below.

Since in this model, we treated store number and item number as a categorical variable, there are 170 total variables not including the intercept in this model. When looking at the summary output for this model, it becomes clear that matching store numbers is one of the best ways to predict the amount of unit sales. However, like all categorical variables with an extreme number of factors, not all of the levels are significant. Item number also follows a similar trend in that there are some levels which have coefficients indistinguishable from zero, but most of the levels are irrelevant. We think that this is because most of the items sell less than one a day. Both of these issues will be explored further in the next section of this report.

Diagnostics

Improvements

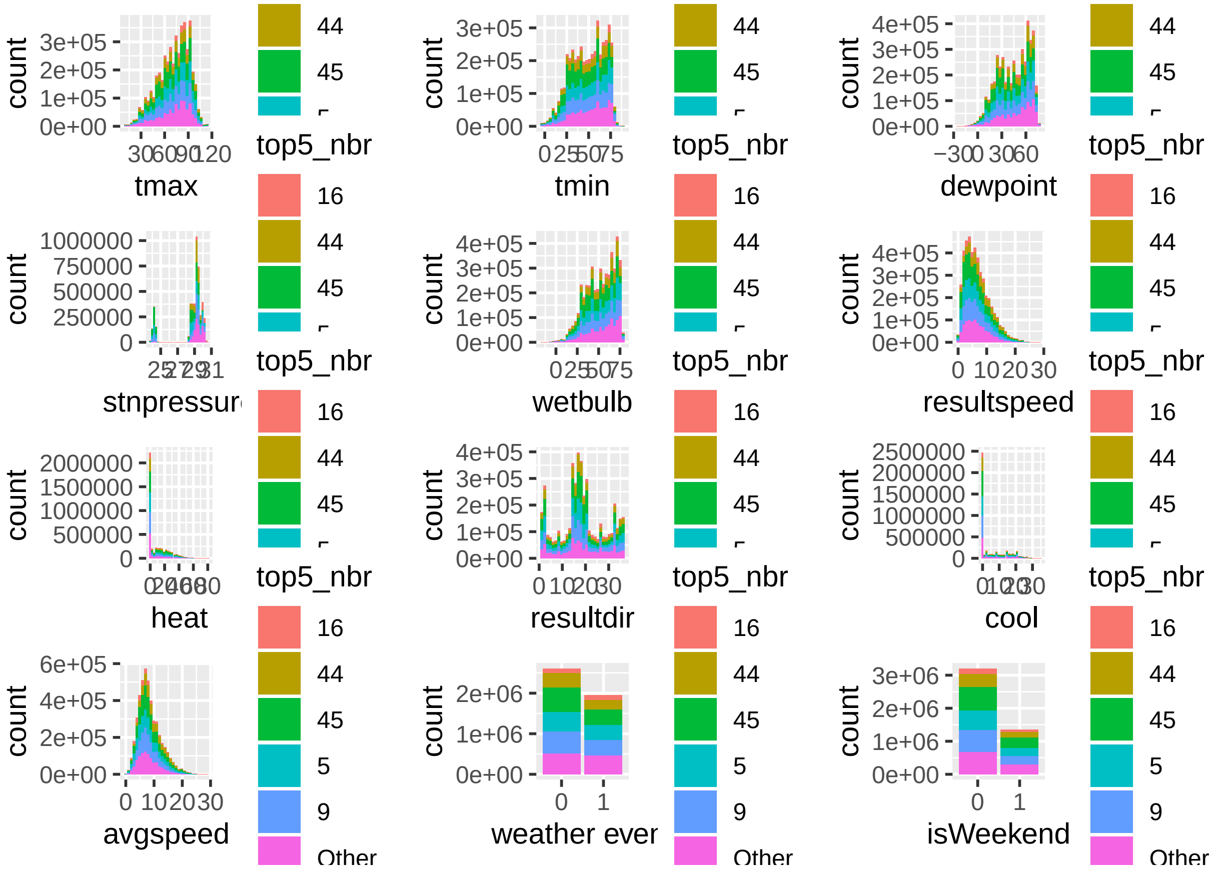
Appendix

include code in the appendix. we will also submit code as a separate attachment

Data Dictionary

- `item_nbr`: categorical variable for each of the 111 types of store items
- `nbr_unit`: integer value of how many units of the corresponding item were sold
- `tmax`: maximum temperature (degrees Fahrenheit)
- `tmin`: minimum temperature (degrees Fahrenheit)
- `dewpoint`: average dew point; the higher the dew point, the greater the amount of moisture in the air
- `wetbulb`: average wet bulb temperature; lowest temperature to which air can be cooled by the evaporation of water into the air at constant pressure
- `heat`: heating degree day (HDD) with season beginning with July; measures the demand for energy needed to heat a building; number of degrees that a day's average temperature is below 65ºF
- `cool`: cooling degree day (CDD) with season beginning with January; measures the demand for energy needed to cool buildings; number of degrees that a day's average temperature is above 65ºF
- `codesum`: binary: 1 if a weather event occurred, else 0.
- `stnpressure`: average station pressure
- `resultspeed`: resultant wind speed; average of all wind speeds at a given place for a certain period
- `resultdir`: resultant direction; average of all wind directions at a given place for a certain period
- `avgspeed`: average wind speed

EDA


count
3e+05
2e+05
1e+05
0e+00
30 60 90 120
tmax
44
45
top5_nbr
3e+05
2e+05
1e+05
0e+00
0 25 50 75
tmin
4e+05
3e+05
2e+05
1e+05
0e+00
−30 0 30 60
dewpoint
16
44
45
1000000
750000
500000
250000
0
25 27 29 31
stnpressur
4e+05
3e+05
2e+05
1e+05
0e+00
0 25 50 75
wetbulb
0 10 20 30
resultspeed
2000000
1500000
1000000
500000
0
0 20 40 60 80
heat
0 10 20 30
resultdir
2500000
2000000
1500000
1000000
500000
0
0 10 20 30
cool
6e+05
4e+05
2e+05
0e+00
0 10 20 30
avgspeed
5
9
Other
2e+06
1e+06
0e+00
0 1
weather ever
3e+06
2e+06
1e+06
0e+00
0 1
isWeekend

References