The Effect of Product Reviews on Company Stock Prices

BUY LOW SELL HIGH PERFORMANCE COMPUTING

JACK BAUMANN, MICHAEL BOLOT, TED MORIN

Technologies

AWS EC2

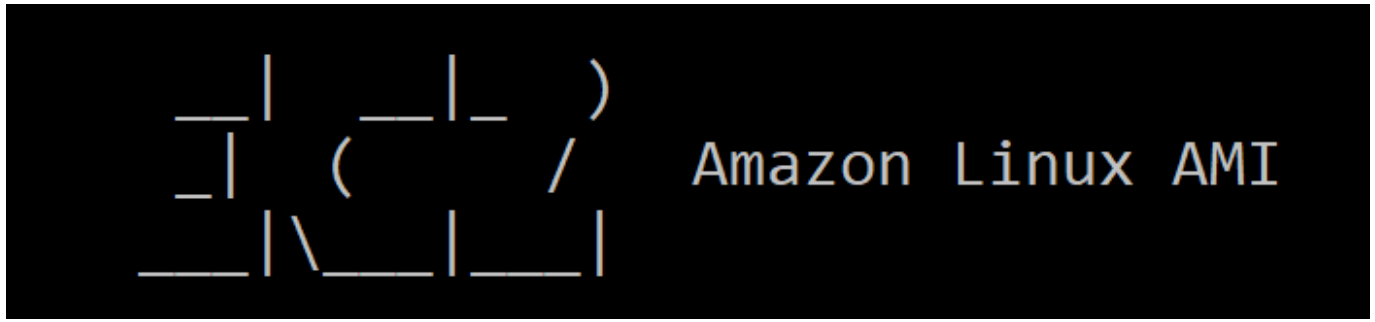
Amazon S3

EMR

Collecting Data

- We collected Amazon user review data from an AWS open data S3 bucket.
- We collected data like daily closing price for specific stock tickers from a Yahoo! Finance query.

- We launched an EC2 instance to gather the data onto.
- Next, we transferred the data to an S3 bucket.



Processing The Data

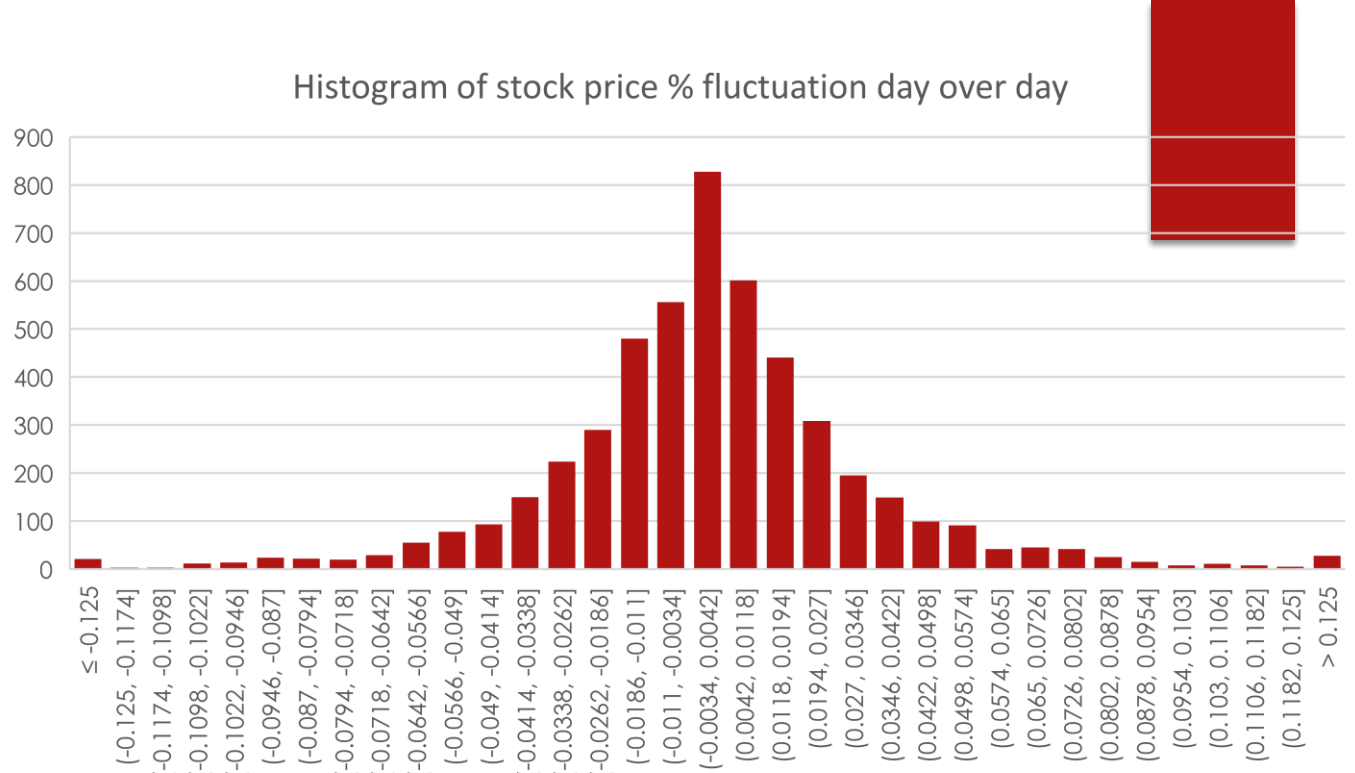
- We spun up an EMR cluster to filter, reduce, and map the data to get only the data we wanted in the format we needed.
- We grabbed the data for 3 million Electronic product reviews from our S3 bucket and processed it with Spark.
- We wrote pyspark scripts to filter out any products without “Lenovo” in their title.
  - This left us with thousands of reviews for Lenovo products.
- We then wrote the data for those products out to a .csv file with the dates and data formatted how we wanted.

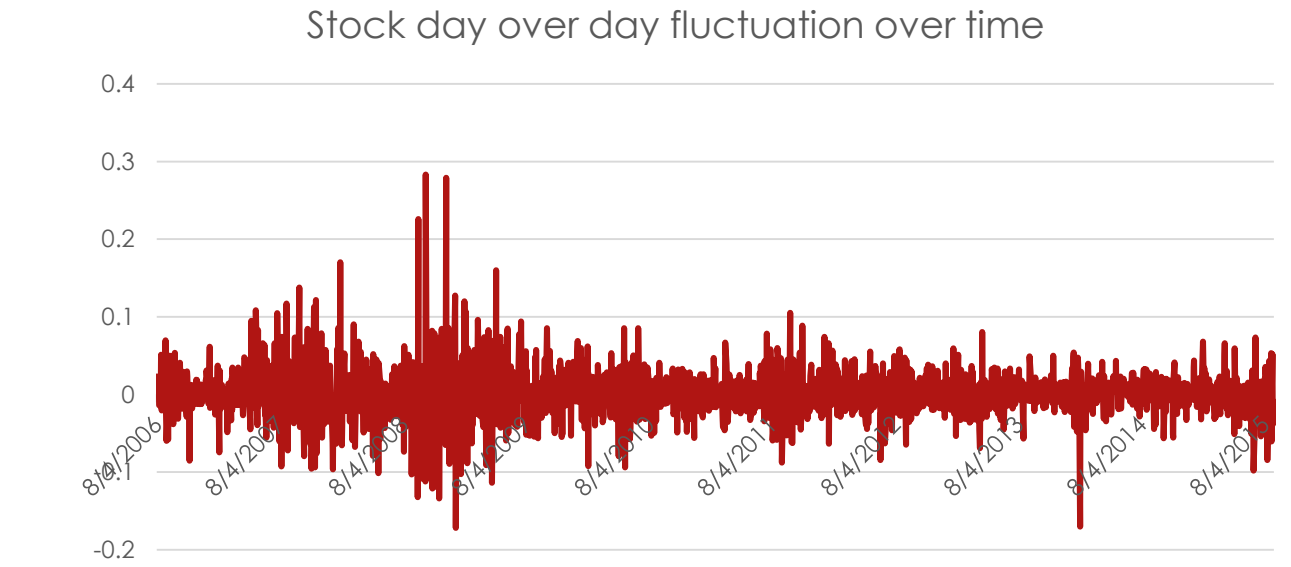
Analyzing the Data

- To analyze the data, we identified the predictors from the review data including score, verified purchaser, verified purchase*score, and vine review*score.
- For each predictor, we computed the average value of that predictor over the last 30 days.
- We then ran a least squares fit from the predictors to the percent change of stock price at market open between that day and the next.

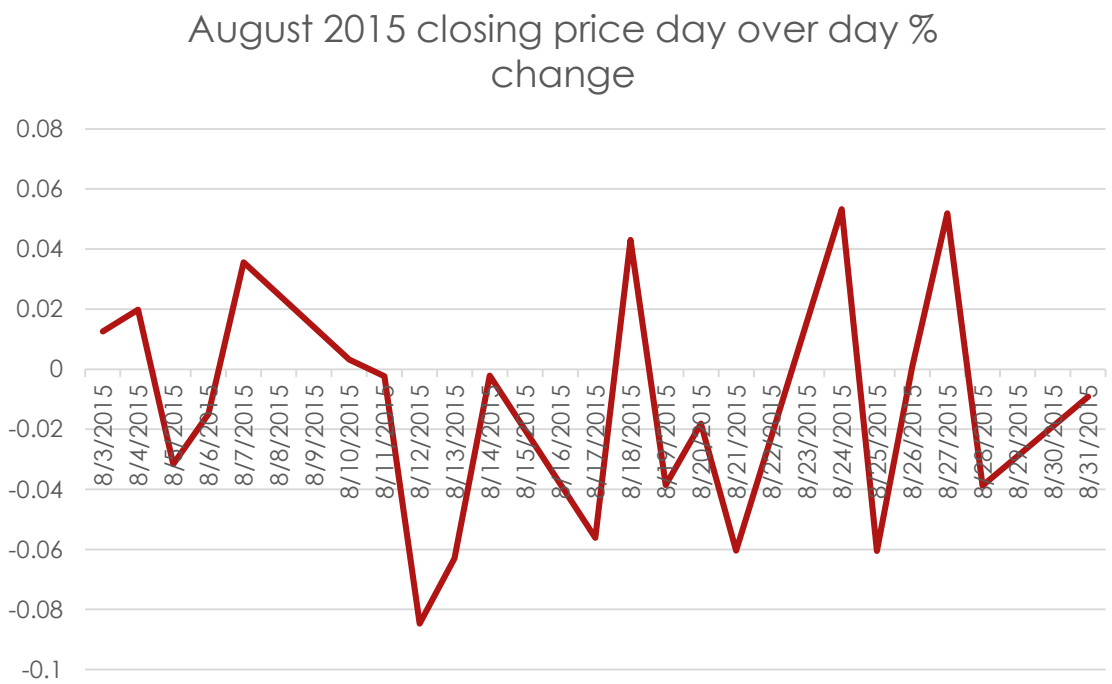
Results

- We chose Lenovo because their stock is not very volatile. Hopefully it will be easier to directly correlate our two factors.





Results


August 2015 closing price day over day % change
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
8/3/2015
8/4/2015
8/5/2015
8/6/2015
8/7/2015
8/8/2015
8/9/2015
8/10/2015
8/11/2015
8/12/2015
8/13/2015
8/14/2015
8/15/2015
8/16/2015
8/17/2015
8/18/2015
8/19/2015
8/20/2015
8/21/2015
8/22/2015
8/23/2015
8/24/2015
8/25/2015
8/26/2015
8/27/2015
8/28/2015
8/29/2015
8/30/2015
8/31/2015


August 2015 Amazon reviews normalized
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
8/1/2015
8/2/2015
8/3/2015
8/4/2015
8/5/2015
8/6/2015
8/7/2015
8/8/2015
8/9/2015
8/10/2015
8/11/2015
8/12/2015
8/13/2015
8/14/2015
8/15/2015
8/16/2015
8/17/2015
8/18/2015
8/19/2015
8/20/2015
8/21/2015
8/22/2015
8/23/2015
8/24/2015
8/25/2015
8/26/2015
8/27/2015
8/28/2015
8/29/2015
8/30/2015
8/31/2015

Results

	Coefficients	[95.0% Conf. Int.]	
Intercept	-1.218e+08	-4.48e+08	2.04e+08
Score	2.435e+07	-4.09e+07	8.96e+07
Score*VP	2.435e+07	-4.09e+07	8.96e+07
Score*VINE	2.435e+07	-4.09e+07	8.96e+07
Score*Helpful	2.435e+07	-4.09e+07	8.96e+07
Review Count	2.435e+07	-4.09e+07	8.96e+07

Issues we need to address

- We currently just filter out products that do not contain “Lenovo” in their title, but ideally we would use the product ASIN (which we have for each review) to match it to the brand Lenovo.
  - This is possible but might require a web scraper which we did not have time to do.
  - With this approach we could discount products with titles like “Headphones for Lenovo T430 laptops”, which are not made by Lenovo themselves.

Where do we go now? Possible future additions to our Hack

We have data and some code to run sentiment analysis on text reviews.

It would be neat to see if the sentiment of a review is more impactful than the score.

We would also love to look at text reviews versus rating reviews and see if text reviews are actually more impactful.

And finally, the most obvious expansion of our project is to compare the trends we saw with Lenovo to other brands.

We think that car companies would also be a very good case study because they are stable and only release one product line a year.

Long term expansions of our Hack

- We initially hoped to scrape for and examine professional reviews from a number of different review websites but had to settle for Amazon user reviews instead.
  - We think we'd see a much stronger correlation if we used professional reviews.
- We have much more data about the Amazon user reviews that we would love to use in the future.
  - We would love to further examine if data like helpfulness, total votes, and vine / verified purchase have more or less sway on stock prices