Capstone Project - Summary Report

Boston University - MSBA Program 2020

Identifying Amazon Influencers and the Effect They Have

Github Link: https://github.com/lunawu007/Capstone-project.git

Authors: Cohort A Team 4: Ke Zang; Di Yao; Luke Towers; Shimiao Li; Yue Wu

Introduction to Business Problem

a. Problem Statement:

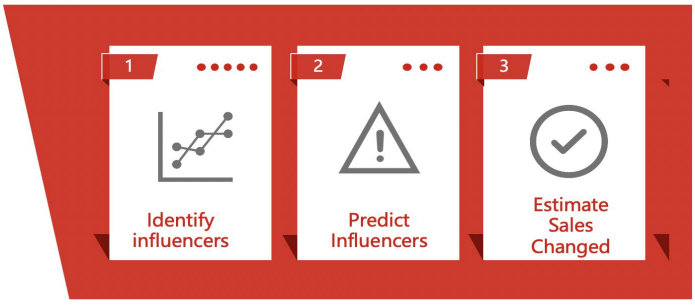
Amazon is all about sales and revenue when it comes to online shopping. With millions of products and millions if not billions of customers Amazon have cemented themselves as the number one online retailer in the US and possibly in the world. On Amazon customers can purchase and review products they buy.

b. Why is it important?

Influencers are users or customers who are paid to review or promote products on different sites and social media platforms such as Instagram, Snapchat and even Amazon. On amazon well liked and respected users are sent products to use, or paid money to leave favourable reviews. We believe that Amazon will want to be able to identify and contact these influencers, and also find out what impact they have on the sale of their products.

The project mainly concerns the following 3 issues:

1) **Identify amazon influencers** from a large data set of 250 million reviews and product descriptions.
2) **Predict influencers** based on variables including the sentiment of review text and length of reviews along with other variables.
3) Estimated the **sales change** based on an influencers impact on a product.



Introduction to Dataset

***a. Data link*:**https://nijianmo.github.io/amazon/index.html

This Dataset is an updated version of the Amazon review dataset released in 2014. For each one specific category, we mainly focus on three related datasets: 5-core, ratings-only and metadata.

b. Dataset Entailed:

Complete review data: To request for the complete review data as well as the per-category files, you will need to complete this form and it is including the following segments:

- raw review data (34gb) - all 233.1 million reviews
- ratings only (6.7gb) - same as above, in csv form without reviews or metadata
- 5-core (14.3gb) - subset of the data in which all users and items have at least 5 reviews (75.26 million reviews)
- Per-category data - the review and product metadata for each category.
- K-cores (i.e., dense subsets): These data have been reduced to extract the k-core, such that each of the remaining users and items have k reviews each.
- Ratings only: These datasets include no metadata or reviews, but only (user,item,rating,timestamp) tuples. Thus they are suitable for use with mymedialite (or similar) packages.

c. Data Summary

The original dataset includes 20-30gbs, which is too big, so we decided to focus on one subsidiary to build a model, ideally the model should be able to be applied to other categories. We finally decided to focus on the subsidiary “Cell Phone and Accessories”, which includes a review dataset that has 1,128,437 rows and 12 columns with 48,186 products. The Top 1 reviewed product is Anker 24W Dual USB Car Charger.

Exploratory data analysis- EDA

a. Data Cleaning

We first stripped out N/As and unused columns in the dataset. Also, converted string to date format for review time column.

b. Segment Decision

We randomly choose a sizable (over 1million rows) segment as our project focus, which is "Cell Phone and Accessories".

c. Adding Column of 'Influencer'

By analyzing the effect of influencer, we first added a column for influencer by descending each segment review's annual helpful votes and then selected top 500 customers as "influencers" while we also randomized 500 other users as "non-influencers".

d. Initial Exploratory

Create a histogram to count how many reviews each rating has. From fig.1, we can see almost 70% of the ratings are 5.0, which means customers are generally satisfied with the products. However, we do realize that there are some outliers with review rating 1-2, we would also do further analysis of their reviews to give vendors' better suggestions of improving their products and also check if those reviews are from 'influencers'.

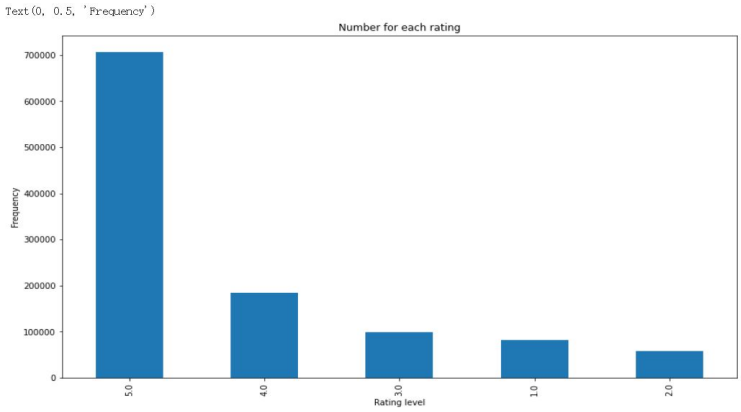
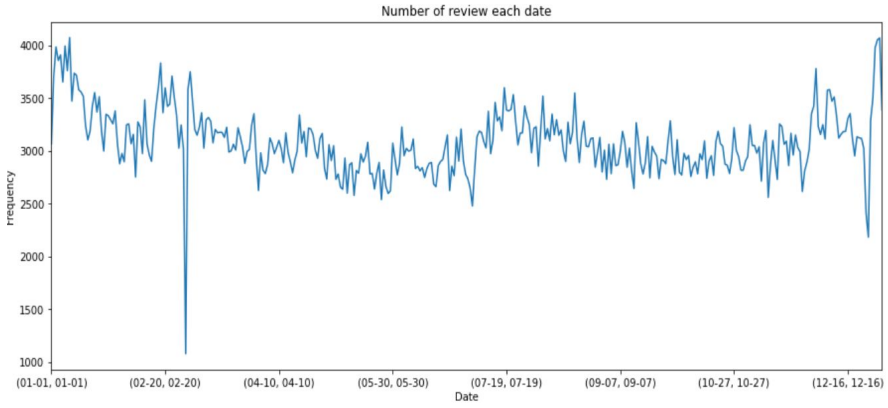


fig.1

We created a time series graph to show which day has more reviews. From fig.2, we can know January and December are the two months with the top number of reviews, which is 4073 and

4069 reviews. It may be because of the holiday effects, Christmas and New Year are around that time. Customers are used to giving reviews four to five days after giving out holiday gifts.



Date	Date	
01-09	01-09	4073
12-30	12-30	4069
12-29	12-29	4052
01-07	01-07	3993
01-03	01-03	3984
		...
05-25	05-25	2537
07-04	07-04	2476
12-24	12-24	2411
12-25	12-25	2182
02-29	02-29	1077

(fig.2)

We also create a table (fig.3) to try to find out the 'influencer' based on how many reviews they did. We can say some customers give lots of reviews and all of them are 5.0. This makes us think that we can't judge whether the influencers only depend on the quantity, because they may be fake reviewers hired by the seller in order to raise the evaluation score of the product.

reviewerID	overall	
A3HFQE02MDPC6I	5.0	306
A2LTYEYGKBYXRR	5.0	200
A3MFRAR121IHTN	5.0	144
A3RDSN4SZKF30	5.0	130
A1FL151ER57VW3	5.0	114
		...
A3KWCPP5TSVIOH	3.0	1
	4.0	1
A3KWIW4P20R68I	4.0	1
A26VFOPF302DGN	4.0	1
AZZYW4Y0E1B6E	1.0	1

reviewerName	overall	
Amazon Customer	5.0	24546
	4.0	5795
	3.0	3080
	1.0	2809
Kindle Customer	5.0	2113
		...
Justlkn	2.0	1
	1.0	1
Justjon	4.0	1
	2.0	1
santi	3.0	1

fig.3

e. Influencers' Effect Exploratory

By comparing influencers and non-influencers' comments, we mainly find 3 primary differences:

1. We can see from fig. 4 that overall clients tended to leave higher average ratings while influencers have a lower average than overall.

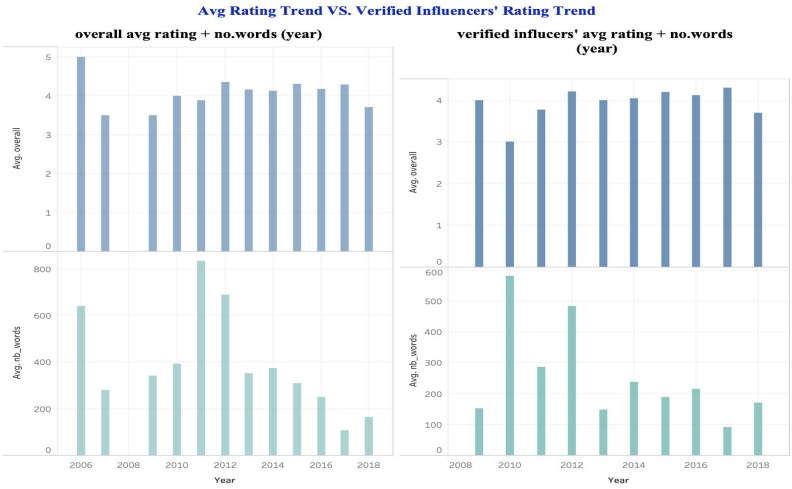


fig.4

2. We can see from below fig.5 that compared with other clients, influencers tend to leave comments with image.

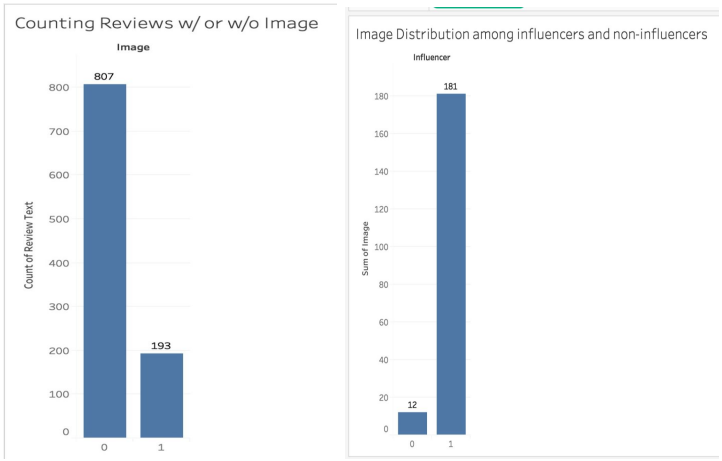


fig.5

From this we can determine that in the future we may have to redefine what characteristics we use to define an influencer. Do their comments just have likes because they have a picture? This measure of likes on comments may not be an effective way to identify influencers.

Solution and Methodology

Identifying influencers

We identified influencers based on the average number of likes per year. We first converted the 'vote' data type from object to numeric and created a new column called 'avg_vote'. 'avg_vote' is calculated by the total number of votes divided by the duration, which is the end of day of our dataset 10/2/2018 minus the review time. Then, divided by 365 days to get the average number per year. Selected top 500 reviewers as our influencer and combined with other random other 500 non-influencers. We generated an influencer dataframe with these 1000 data based on the number of votes.

Sentiment analysis

We utilized text sentiment analysis in order to get an idea about if products have positive reviews or negative. We splitted the influencer dataset into two parts. Ratings higher than 3 star (4 and 5 star) are good reviews, and ratings lower than 3 star (1 to 3 star) are bad reviews. Then we use Word2Vec contexts to analyze the review text to get the emotion score, which is called compound score in word vec 2 algorithm. The compound score is calculated by Word2Vec contexts itself. Generally, we think if people give a product a lower star, it means this customer is angry or not satisfied with the product. Therefore, the emotion of the review text should be negative. However, it is not always true according to the line graph we get. (Figure 6) We can see the peak of the blue line is on the positive side, which means even when customers give lower star ratings to the product, their emotion of review text is still positive. Another thing we find out is the negative and positive emotion texts with the highest scores are from

non-influencer. (Figure 7) Therefore, we think people like somebody's comment are not influenced by the emotion of the commentator.

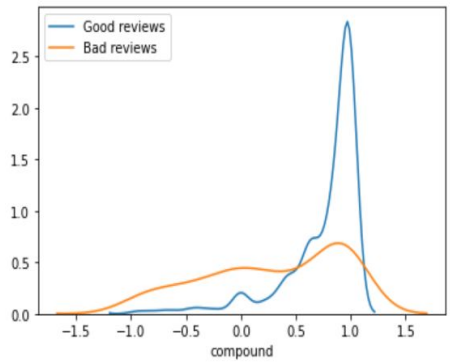


Figure 6 -- The distribution of the reviews sentiments among good reviews and bad ones

	review_clean	pos	influencer
823	nicely build elegant good fit	0.919	0
572	fit perfectly protect well look good	0.852	0
893	good product good price good service	0.744	0
876	excellent packaging awesome look	0.722	0
917	great product nice price	0.697	0
739	beautiful product compact efficient functional	0.691	0
880	cool work great	0.690	0
743	love fun phone super durable	0.661	0
699	great prop easy put away	0.646	0
796	good price quick ship work great	0.636	0

	review_clean	neg	influencer
929	suck waste money	0.655	0
517	screen go bad within day	0.467	0
978	absolutly horrible week use stop work stop rea...	0.456	0
524	super product problem	0.415	0
672	holster clip break real quick	0.412	0
706	week start fall apart quality suck	0.386	0
904	break first time drop phone	0.359	0
642	long hands-free	0.355	0
790	get disgust even return refund phone still bra...	0.329	0
841	place lexus yet	0.306	0

Figure 7 -- highest positive and negative sentiment reviews (with more than 5 words)

Predicting influencers

One of our goals is to build a model to assist Amazon in predicting influencers. In addition to our original influencers dataset, we added several sentiment score variables from our previous WordVec2 results into the dataset, as shown in Figure 8. Thus, we have these 11 variables and 1 binary target (influencer 1 or 0) to build our models:

- Random Forest
- Logistic Regression
- Neural Network

	influencer	overall	verified	image	Month	Weekday	neg	neu	pos	compound	nb_chars	nb_words
0	1	4.0	1	1	7	3	0.080	0.779	0.141	0.9874	3387	574
1	1	5.0	0	0	8	4	0.011	0.829	0.160	0.9993	5203	862
2	1	5.0	1	1	2	4	0.050	0.797	0.153	0.9986	4910	872
3	1	4.0	1	1	4	3	0.058	0.839	0.103	0.9973	7891	1408
4	1	5.0	0	0	9	6	0.037	0.850	0.113	0.9995	10825	1973

Figure 8 -- Influencer DataFrame for Machine Learning

We randomly split the train and test dataset with 75% train and 25% test and ran the Random Forest model in a python notebook, the tree that was formed after this is seen below in figure 9 The model accuracy is 56.5% which is not that good considering random guessing would result in an approximate accuracy of 50%.

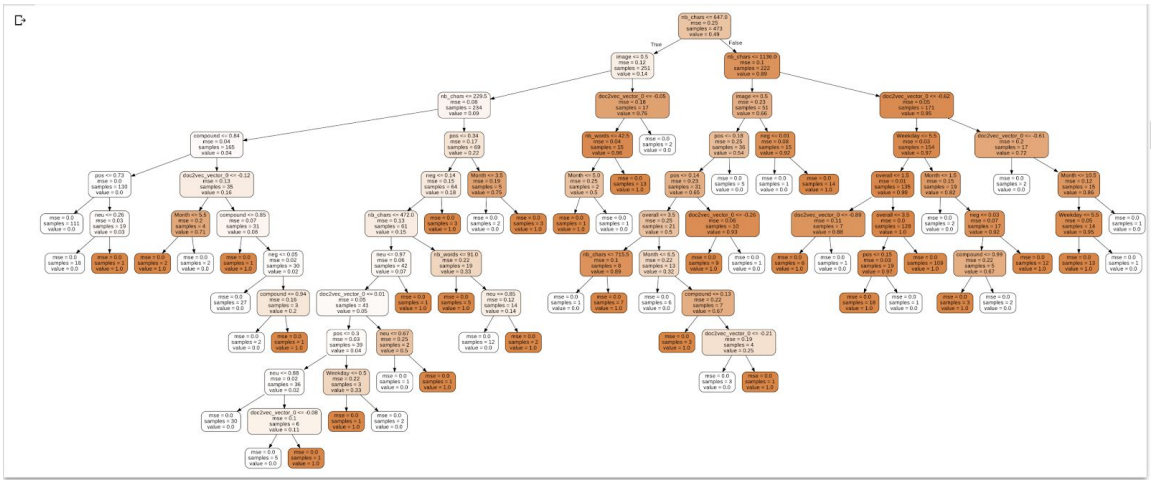


Figure 9 - Random Forest Tree

Variance importance in Figure 10 shows us that the length of the review has a huge impact on whether a review is an influencer or not. Not only this but also whether the review had an image. We found from our EDA that the majority of reviews that were posted by ‘influencers’ had images with them.

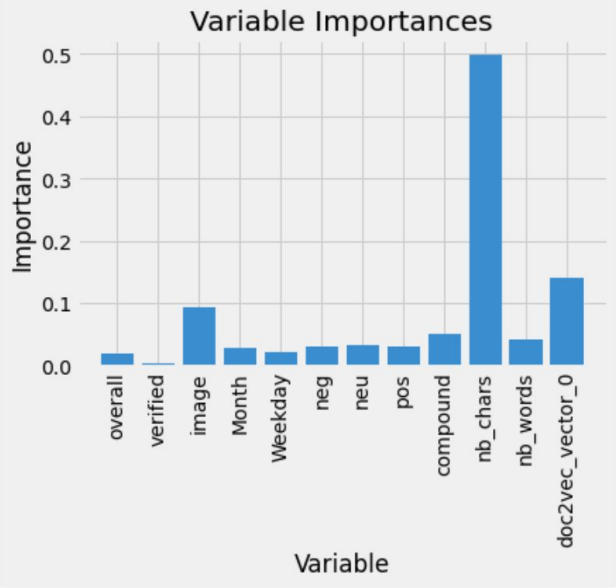


Figure 10 - Variable importance Random Forest

Another approach is Logistic Regression which is a special linear regression for predicting binary classes, so this would be a good fit in predicting whether a reviewer is an influencer or not. The test and training dataset are randomly splitted with a rate of 25%. After training the logistic regression model with 750 data, we tested on the rest, which gave us an 87.6% accuracy. As table Figure 11 showed, if a reviewer attached an image, it’s highly likely that he/she was an influencer. The second strong drive for a reviewer is an influencer is the compound score, which is directly calculated by WordVec2 algorithm. This score is a compounded calculation of both

negative and positive emotions. Another two variables like verified and positive sentiment have some relative drive in predicting someone is not an influencer.

Variable	Coefficiency
overall	-0.14516262427316068
verified	-0.5616851945029869
image	2.649848103295815
Month	-0.012205696996266275
Weekday	-0.04542333598267026
neg	-0.059675822171955155
neu	-0.01226121705080832
pos	-0.49525003414496566
compound	0.2330904391158446
nb_chars	0.002705593172999704
nb_words	-0.0064214398522966475

Figure 11 - Logistic Regression Variable's Coefficiency

We applied a simple logistic regression with sigmoid models in building deep learning. However, the return is only 51% accurate. Lastly, we also tried a linear neural network model. We setted batch size as 100 and learning rate with 0.001. The accuracy of the test data is 72%. The accuracy of this model would be higher if we applied logistic regression instead of linear regression model in the neural network.

Impact on Sales

Since we have identified 500 influencers based on the average vote, we would like to explore the impacts of influencers' on sales: whether customers would tend to purchase after reviewing the comments of influencers.

In terms of quantifying the impact on sales, we have several assumptions.

Assumption 1: We assume that the sales are positively correlated with the number of comments. More customers purchased the product, more comments were left by customers.

Assumption 2: We compare the number of comments 60 days before the influencers' comment date and the number of comments 60 days after. Counting the number of comments is based on the specific product.

Under the two assumptions, we conducted data cleaning and merging procedures and then visualized the number of comments 60 days before the influencers' comment date and the number of comments 60 days after. Horizontal axis is the number of comments 60 days before the influencers' comment date and vertical axis is the number of comments 60 days after.

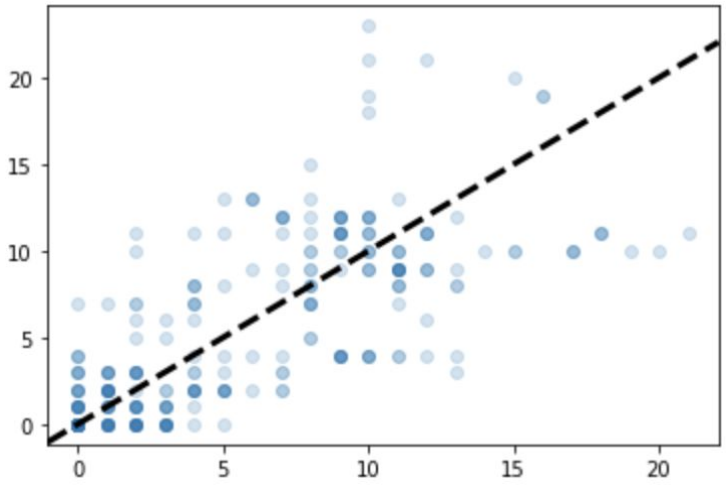


Fig 12

Future statistics description shows that the median value of difference between 60 days before and 60 days after is 0. This may suggest one of two things. Either there is no affect on sales or using the volume of comments is not an effective measure of sales of Amazon products.

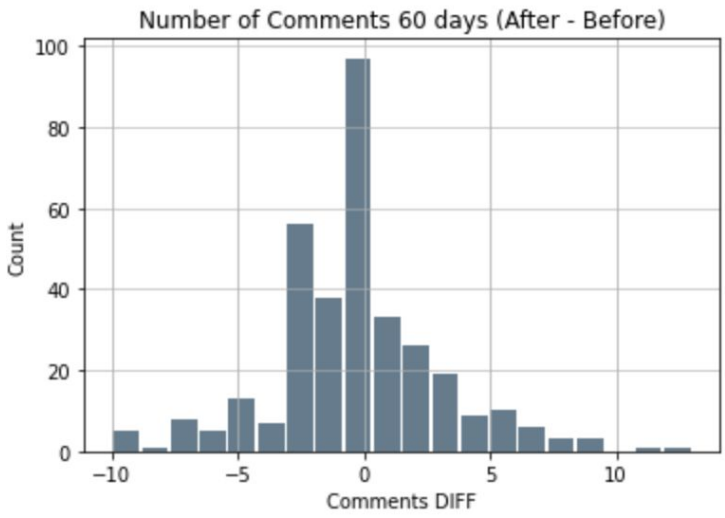


Figure 13

Hence, we can draw the conclusion that the influencers' comments probably do not have a huge impact on the sales. However without the official sales data from Amazon it is hard for us to draw a conclusive conclusion.

Criticism of the results and future work

Criticism

When we do sentiment analysis, the original dataset is too big to analyze. For a regular purely numeric data set the size would be less of an issue, however because we are looking at review text this produces a different problem. When separated into individual words and analysed it makes the dataset far too large to run. Therefore, we only analyzed the influencer dataset, focusing on the emotion of influencers and 500 randomly selected non-influencers.

There are only 5 variables we can use in the beginning, so the number of words has a strong relationship with the influencer, compared with other variables. Therefore, we try to add more

variables in our machine learning model. Even the tokenized word we get from sentiment analysis, however the correlation results still the same.

When we found the influencer, we did not group by the product id. Therefore, it causes some products to have more than one influencer, and some products even do not have an influencer. When we analyze the impact of an influencer, we only keep the products that only have one influencer. So, our dataset becomes more and more smaller which may cause the impact result not to be significant.

Lastly, we determined our "influencer" with numbers of helpful votes. As we explained in the previous section of this paper (section 3), we should also take a deeper look into the "image" variable and see whether image has a more significant impact on whether someone is an influencer or not.

Future Work

1. Build a web scraper utilizing Beautiful Soup to gather more appropriate, unbiased reviews from cell phones and accessories.
2. Re-evaluate the criteria of identifying influencers and probably consider some deep machine learning models to automatically identify.
3. May include several clustering or topic modeling methods to analyze the comments review text.

Conclusion

- The most accurate ML model is the logistic regression model. The best drivers to indicate whether someone is an influencer or not are image, verified or not, and compounded sentiment score.
- People like “influencers” reviews more based on whether they have an image or not and number of words instead of their positive or negative emotion.
- It is clear that without effective sales data of amazon products it is difficult for us to determine whether these “influencers” really affect the sale of products.

Bibliography：

1. Ni, J., 2020. *Amazon Review Data*. [online] Nijianmo.github.io. Available at: <https://nijianmo.github.io/amazon/index.html?nsukey=TruVgRVuAtLRtFtuFSPrNjP78mNVbsQEy1toDKhUDcx66iSlU1mxfS6ZrDGjypuYvrhT1kRjFiu7BwuAOpJvAZ5%2Fn8pj%2FpG%2BfmfRvE5jDKI1m%2B52ZyFMjMxsjvonl1Txz7PTwt01Lv%2FqhV0nM8MyryXGv7fPcuHwtx93trLVMG9Y64Z6xbeM48OuriVPwtY%2BPBJ1eHIjRkbywH8diwK2WA%3D%3D> [Accessed 20 March 2020].
2. https://github.com/barathvaj/Machine-Learning-From-Scratch-CodingChallenge/blob/master/Logistic%20Regression%20From%20Scratch.ipynb
3. https://towardsdatascience.com/random-forest-in-python-24d0893d51c0
4. https://towardsdatascience.com/detecting-bad-customer-reviews-with-nlp-d8b36134dc7e
5. https://machinelearningmastery.com/calculate-feature-importance-with-python/