

**Revision History:**

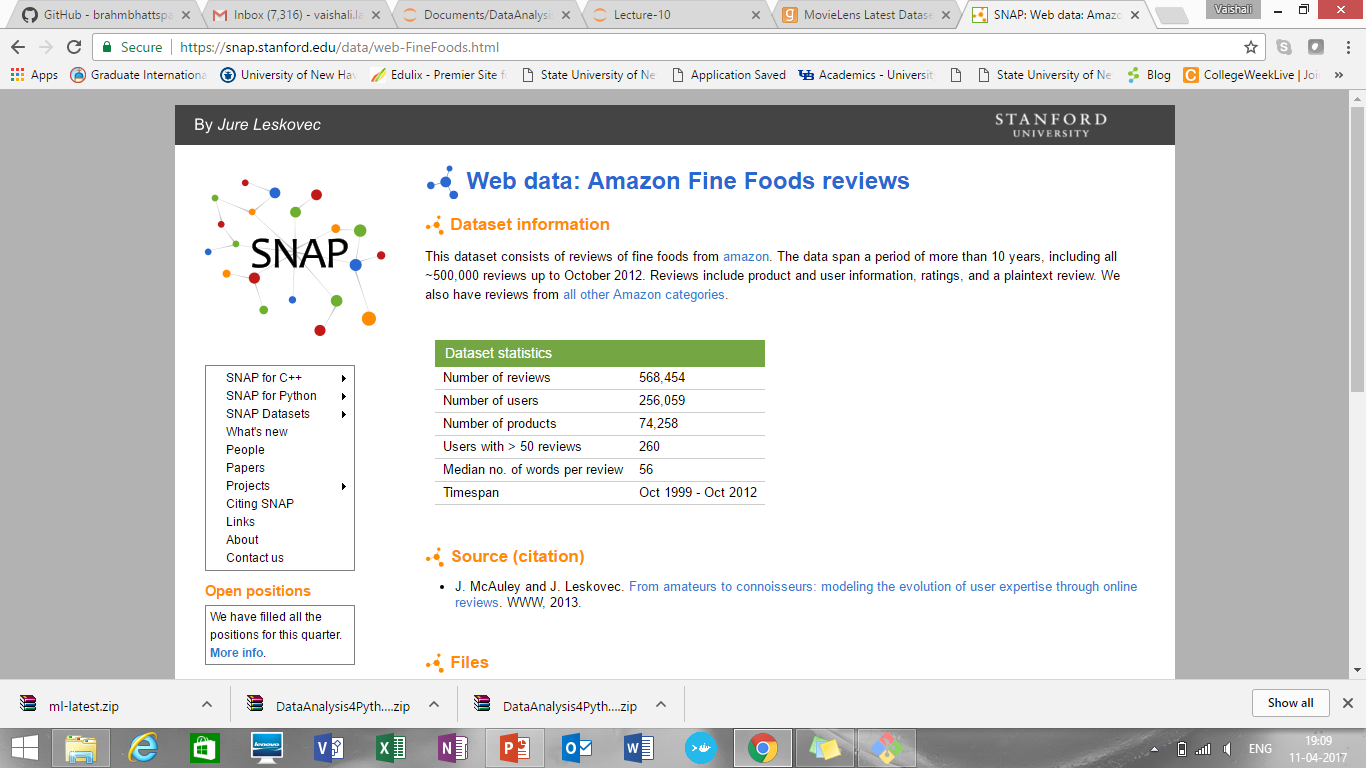
|  |  |  |
| --- | --- | --- |
| **Version** | **Updated Date** | **Comments** |
| 1.0 | 04/27/2017 | Updated for Data Extraction, Data Exploratory and classification (Draft version) |
| 1.1 | 04/27/2017 | Updated for data exploratory |
| 1.2 | 04/28/2017 | Updated for clustering, pipeline, dockerization |
| 1.3 | 04/28/2017 | Updated for Power BI |

**Introduction:**

The focus of this project is, to do sentiment analysis on amazon fine food reviews data which contains more than 5 million reviews. As a part of project, we are performing below tasks:

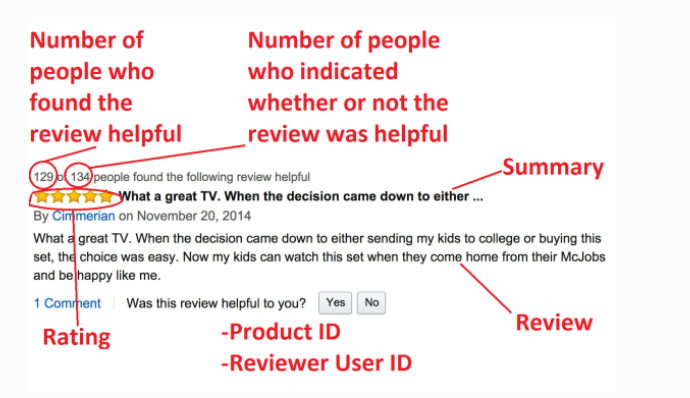
1. Data Extraction and Conversion
2. Cleaning and pre-processing
3. Data Exploratory analysis (using python and Power BI)
4. Sentiment classification and analysis
5. Luigi Pipeline
6. Dockerization
7. Creation of Rest API using Microsoft Azure ML Studio
8. UI deployment

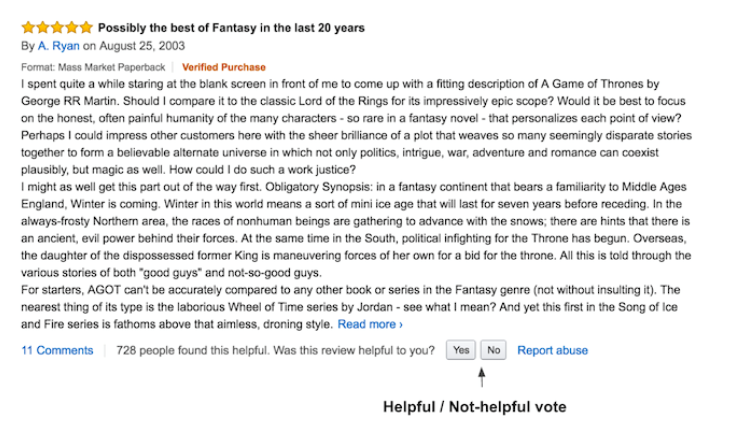
**Little more about our dataset:**



**Columns in dataset:**

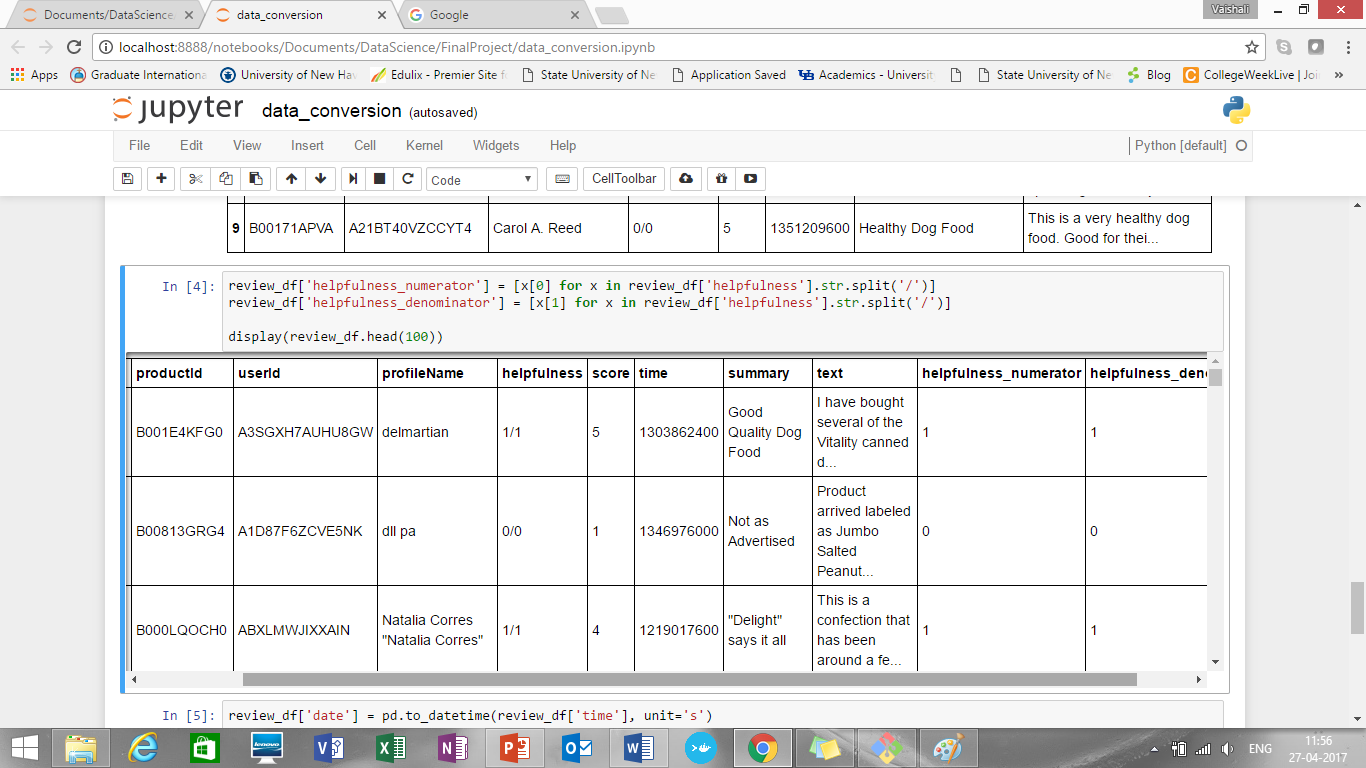
* ProductId - unique identifier for the product
* UserId - unique identifier for the user
* ProfileName- Name of the user
* Helpfulness Numerator - number of users who found the review helpful
* Helpfulness Denominator - number of users who indicated whether they found the review helpful
* Score - rating between 1 and 5
* Time - timestamp for the review
* Summary - summary of the review
* Text - text of the review



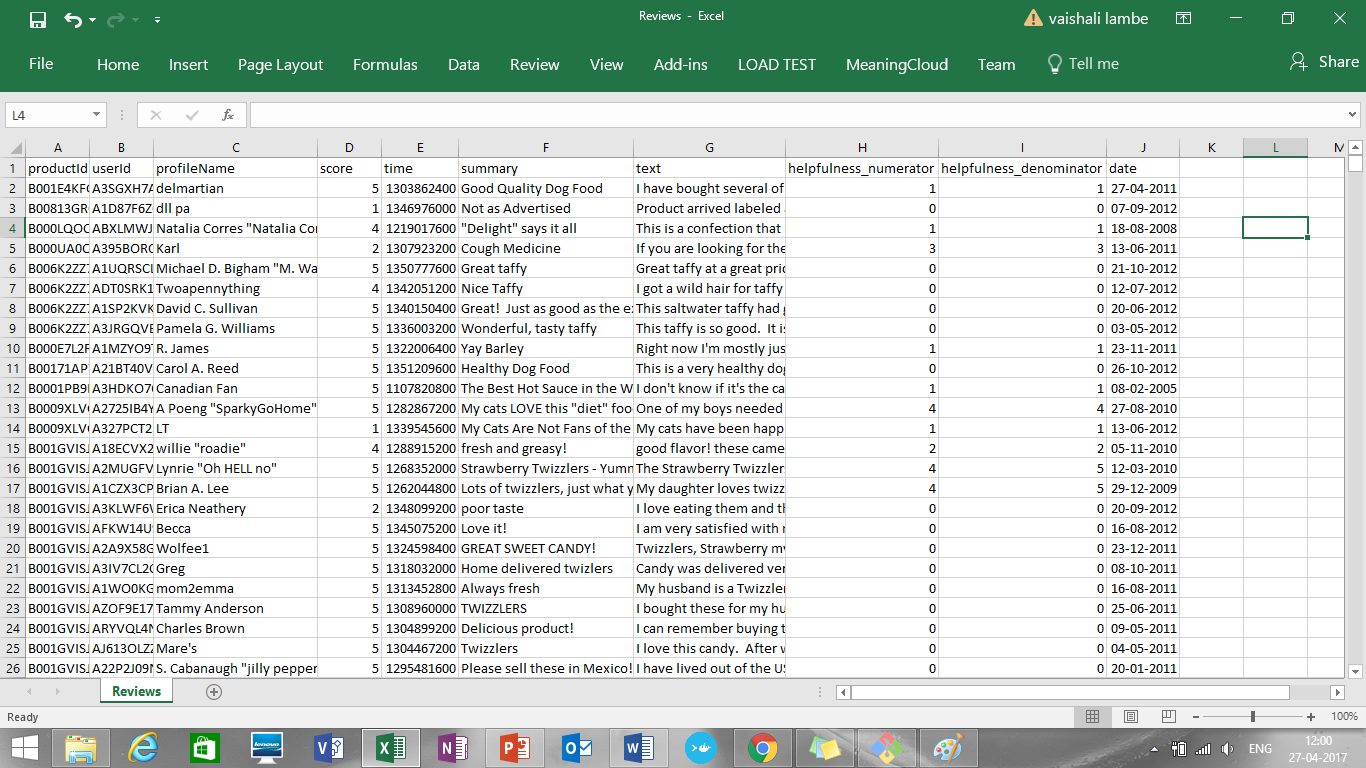


**Data Extraction and Conversion:**

* Automated process to download data from <https://snap.stanford.edu/data/finefoods.txt.gz>
* As data is present in .gz format, decompressed it to get .txt file present
* Then .txt file converted it into .csv file
* To achieve above steps defined several functions like download\_data, decompress\_data, convert\_data.
* Also, helpfulness column divided it into helpfulness numerator and helpfulness denominator before saving it in .csv format
* The output of it was like:



* Added date column as well based on time column
* Output final Reviews.csv file looks like:



**Feature Engineering**

**Defining review helpfulness**

The dataset we used provided the number of helpful and not-helpful votes for each review. Therefore, we defined helpfulness as a binary variable as follows.

helpfulness\_ratio = total\_helpful\_votes / (total\_helpful\_votes + total\_not\_helpful\_votes)

helpfulness = 1 when helpfulness\_ratio >= 0.5 (i.e., helpful), -1 otherwise (i.e., not-helpful)

Defining helpfulness as a binary variable enabled me to set this up as a binary classification problem.

* **Preprocessing Data:**
* Tokenization:

Removing punctuation and converting text to lower case using NLTK’s Porter Stemmer and word tokenize

* Stemming:

the process of reducing inflected (or sometimes derived) words to their word stem, base or root form—generally a written word form

takes morphologically complex words and reduces them to their root morphemes.

For example, the input contains the words "tasty", "tastier", and "tastiest", the suffixes would be stripped off and these forms would be treated as a single stem, "tasti"

* Lemmatization:

removes inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma

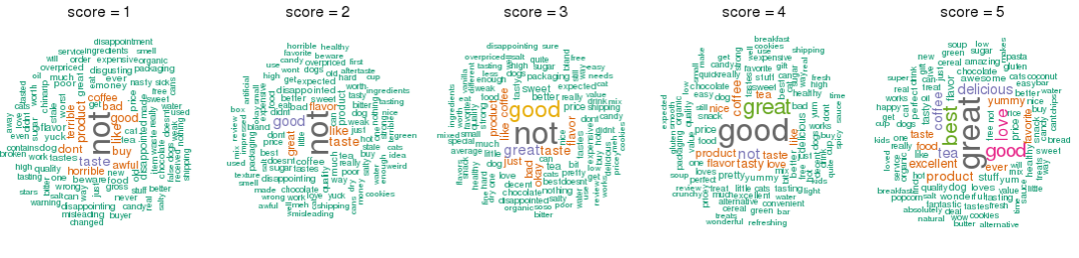
e.g. if support there is word better then it will give me its base form as good or plural form like cats become singular like cat

but if word is “best” it returns best

As lemmatization provides more accurate results over stemming, we have chosen it

* **Exploratory Data Analysis**

**Analysis 1: Word Cloud of Reviews sorted by Score**

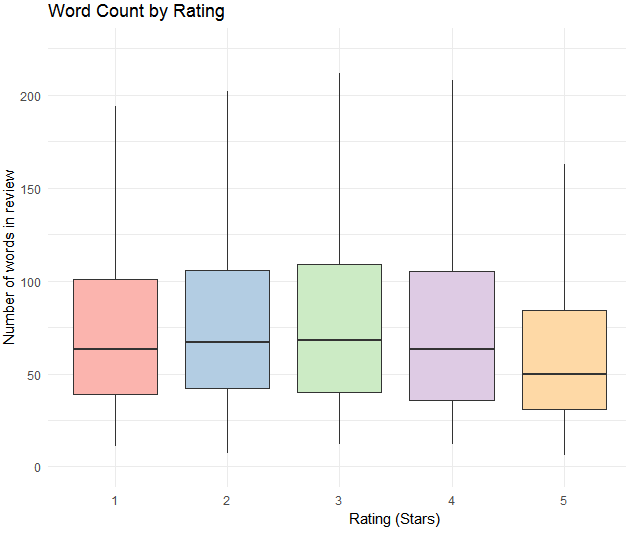
****

We see that the products with rating 1 have negative words such as horrible, awful, disgusting, disappointment, gross, yuck etc. whereas products with rating 5 have positive words in comments such as best, great, love, excellent, yummy, tasty, delicious etc.

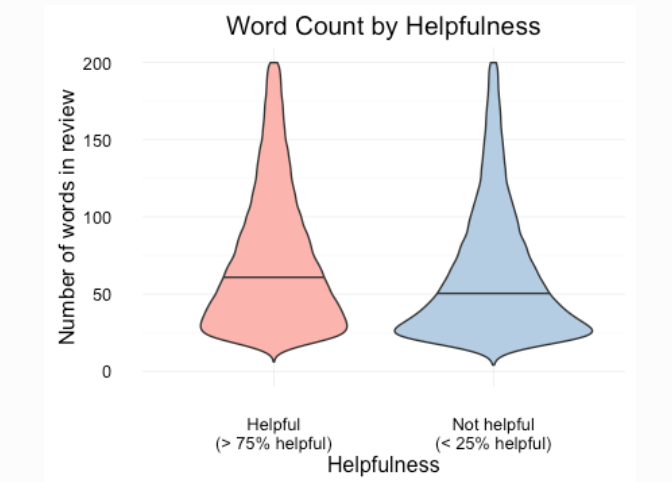
**Analysis 2: Word Count**

One of the most basic characteristics of a review is the number of words it contains. We wanted to see how word count related to the other properties of reviews already discussed, including rating and helpfulness.

**How does word count vary by rating?**



The first question we had regrading word count was how it varied with rating. 5-star reviews had the lowest median word count (53 words), while 3-star reviews had the largest median word count (71 words).

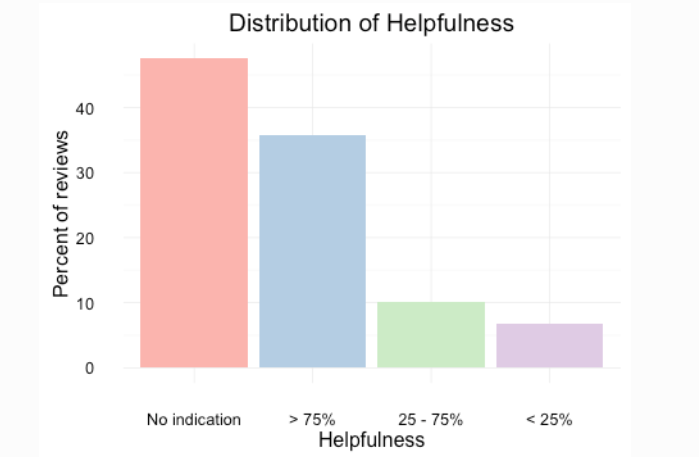
**Analysis3 - How does word count relate to helpfulness?**

The word counts for helpful reviews and not helpful reviews have a similar distribution with the greatest concentration of reviews of approximately 25 words. However, not helpful reviews have a larger concentration of reviews with low word count and helpful reviews have more longer reviews. Helpful reviews have a higher median word count (67 words) than not helpful reviews (54 words).

**Analysis4:**  **Helpfulness**

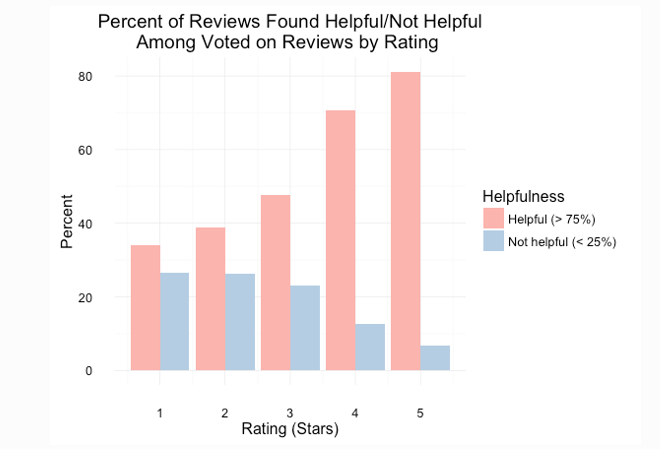
Reviews are voted upon based on how helpful other reviewers find them. The most helpful reviews appear near the top of the list of reviews and are hence more visible. As such, we were interested in exploring the properties of helpful reviews.

**How many reviews are helpful?**

****

Among all reviews, almost half (48%) are not voted on at all. I divided the reviews that were voted upon into three categories: Helpful reviews had more than 75% of voters find the review helpful, unhelpful reviews had less than 25% of voters find the review helpful, and an intermediate group of 25-75% helpfulness. This choice of division seemed to not have a larger impact on results; we will henceforth use this terminology to describe the helpfulness of reviews. Among reviews that are voted on, helpful reviews are the most common.

**Analysis 5: How do ratings affect helpfulness?**

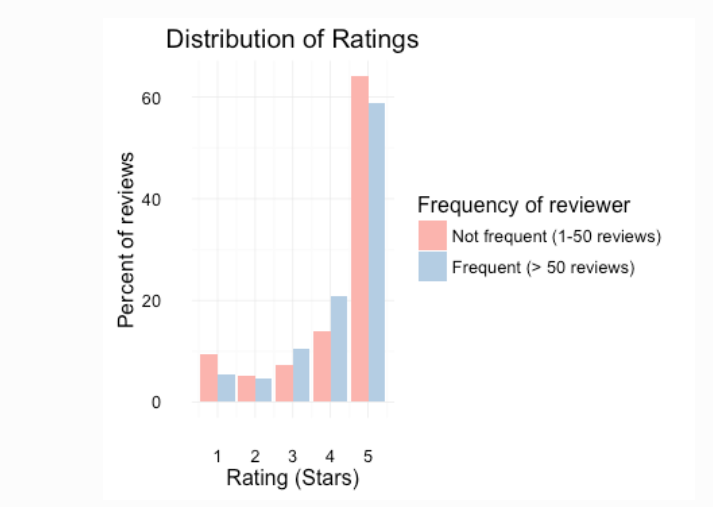
****

For each rating, I looked at the reviews that were voted on and the percent of those reviews that users found helpful or not helpful. As the rating becomes more positive, the reviews become more helpful (and less unhelpful). For 1-star reviews voted upon, 34% were voted helpful, while 27% were found not helpful. For 5-star reviews, 81% were found helpful and 7% not helpful.

**Analysis6: Frequency of reviewers**

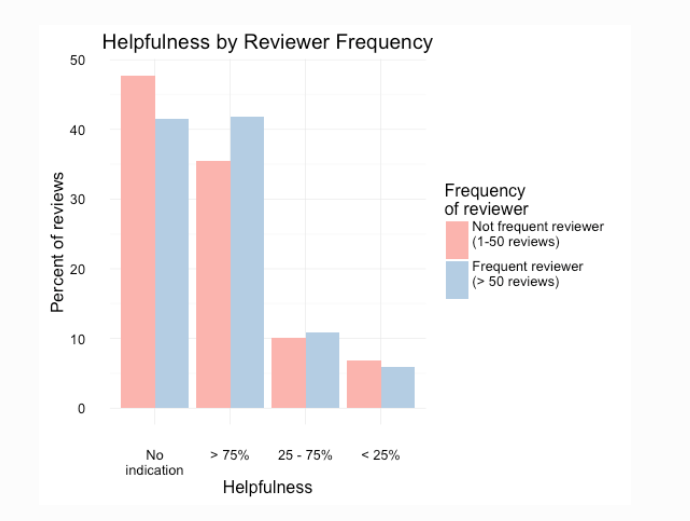
Using User IDs, one can recognize repeat reviewers. Reviewers that have reviewed over 50 products account for over 5% of all reviews in the database. We will call such reviewers frequent reviewers. (The cutoff choice of 50, as opposed to another choice, seemed to not have a larger impact on the results.) I asked: Does the behavior of frequent reviewers differ from that of infrequent reviewers?

**Are frequent reviewers more discerning?**

****

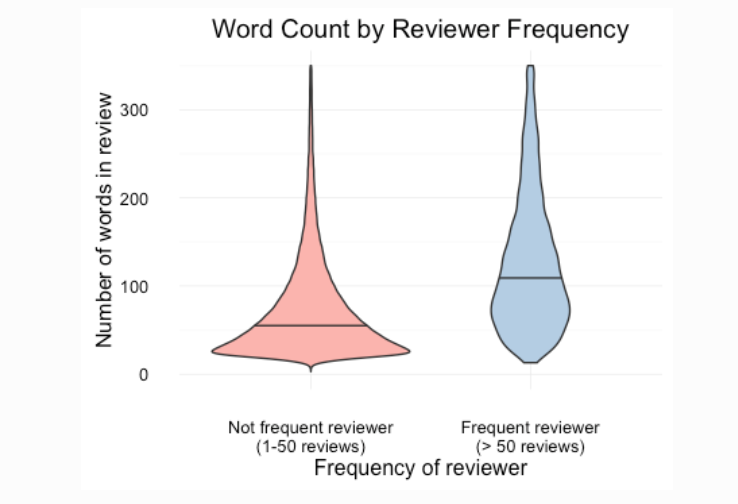
The distribution of ratings among frequent reviewers is like that of all reviews. However, we can see that frequent reviewers give less 5-star reviews and less 1-star review. Frequent users appear to be more discerning in the sense that they give less extreme reviews than infrequent reviews.

**Are frequent reviewers more helpful?**

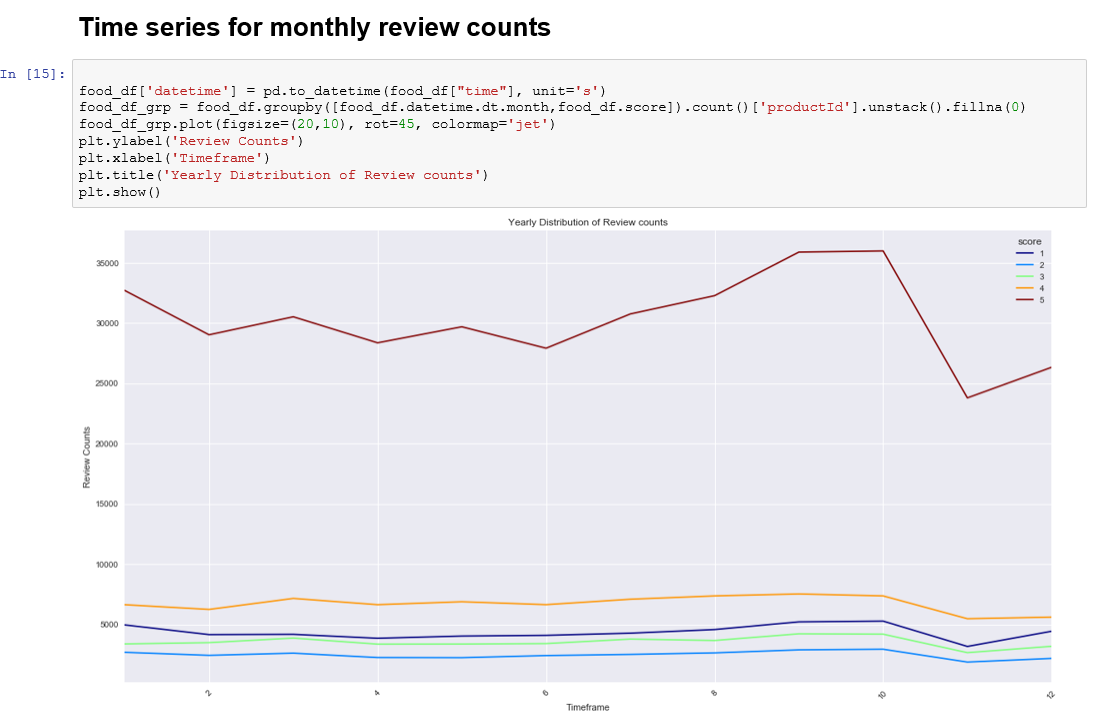
****

The distribution of helpfulness for frequent reviewers is like that of all reviews. However, frequent reviewers are more likely to have their review voted on and when voted on, more likely to be voted helpful, and less likely to be unhelpful.

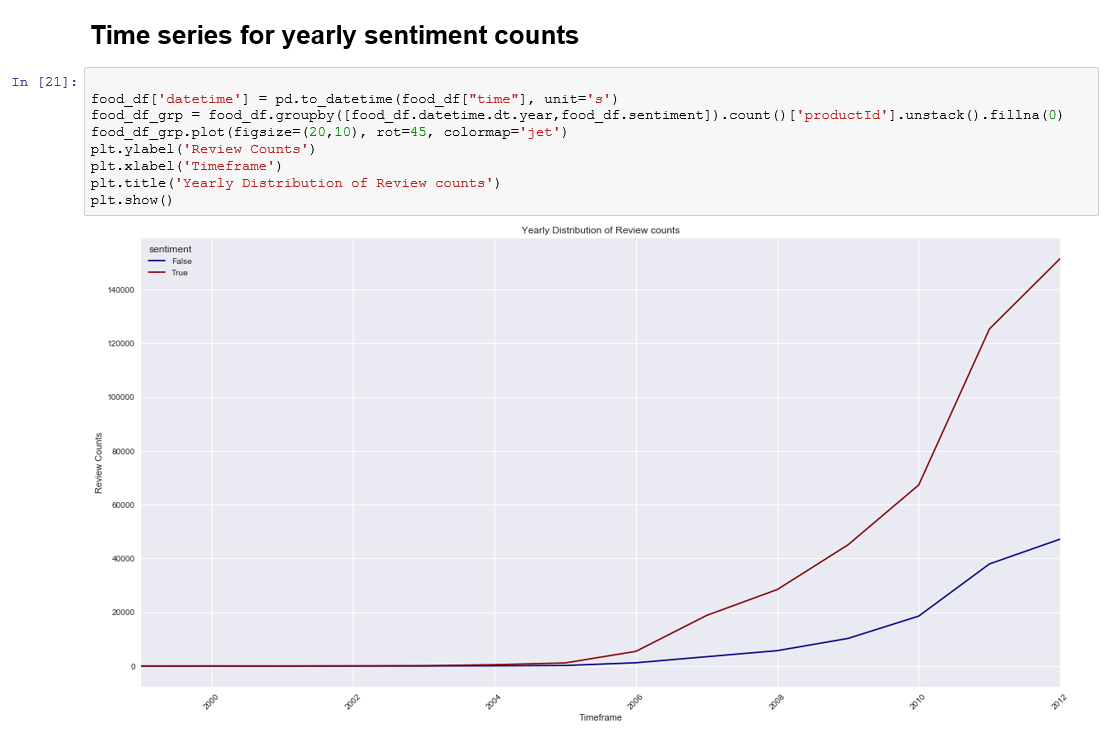
**Are frequent reviewers more verbose?**

****

The distributions of word count for frequent and infrequent reviews shows that infrequent reviewers have a large amount of reviews of low word count. On the other hand, the largest concentration of word count is higher for frequent reviewers than for infrequent reviews. Moreover, the median word count for frequent reviewers is higher than the median for infrequent reviewers.



The total count of monthly reviews of rating 5 are more than other ratings which is ~32000. There was a drop in all the ratings during the month of November.



The label False indicates negative sentiments and label True indicates positive sentiments. The no of positive sentiments is more than that of negative and there has been a sudden increase in number of positive reviews from year 2005 onwards.

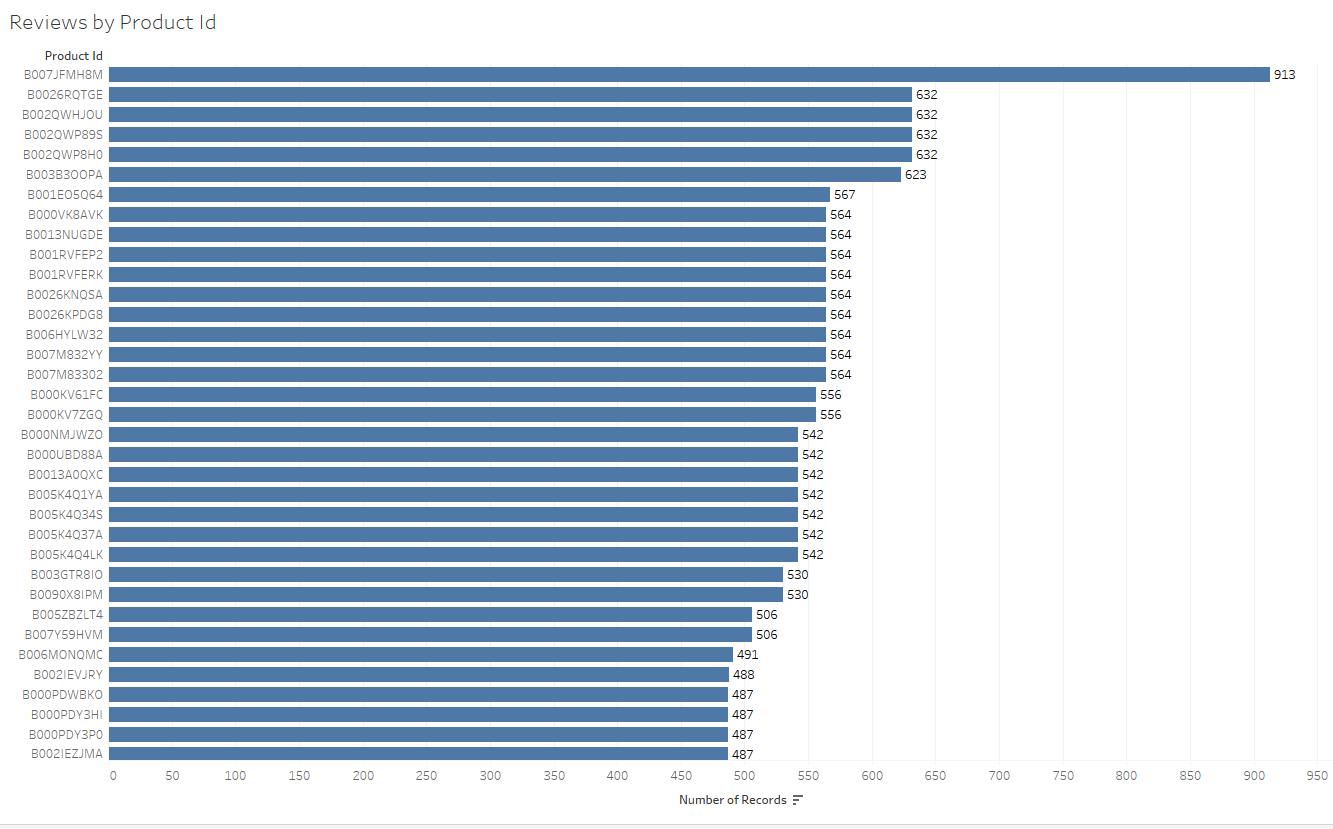
**Conclusions for data exploratory analysis:**

* Positive reviews are very common
* Positive reviews are shorter
* Longer reviews are more helpful
* Despite being more common and shorter, positive reviews are found more helpful
* Frequent reviewers are more discerning in their ratings, write longer reviews, and write more helpful reviews

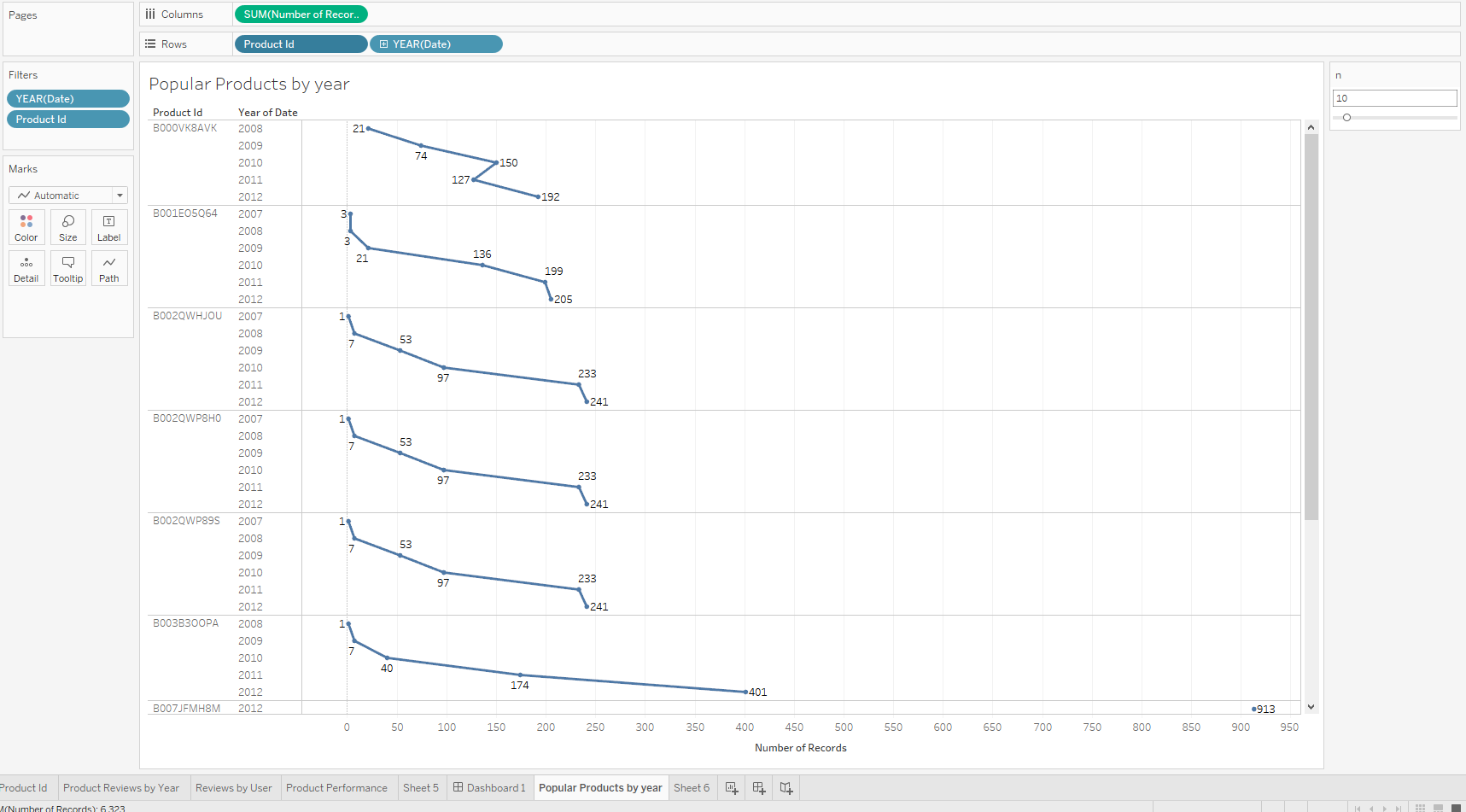
**Data Exploratory using Power BI:**

Product & Reviews

**REVIEWS BY PRODUCT ID**

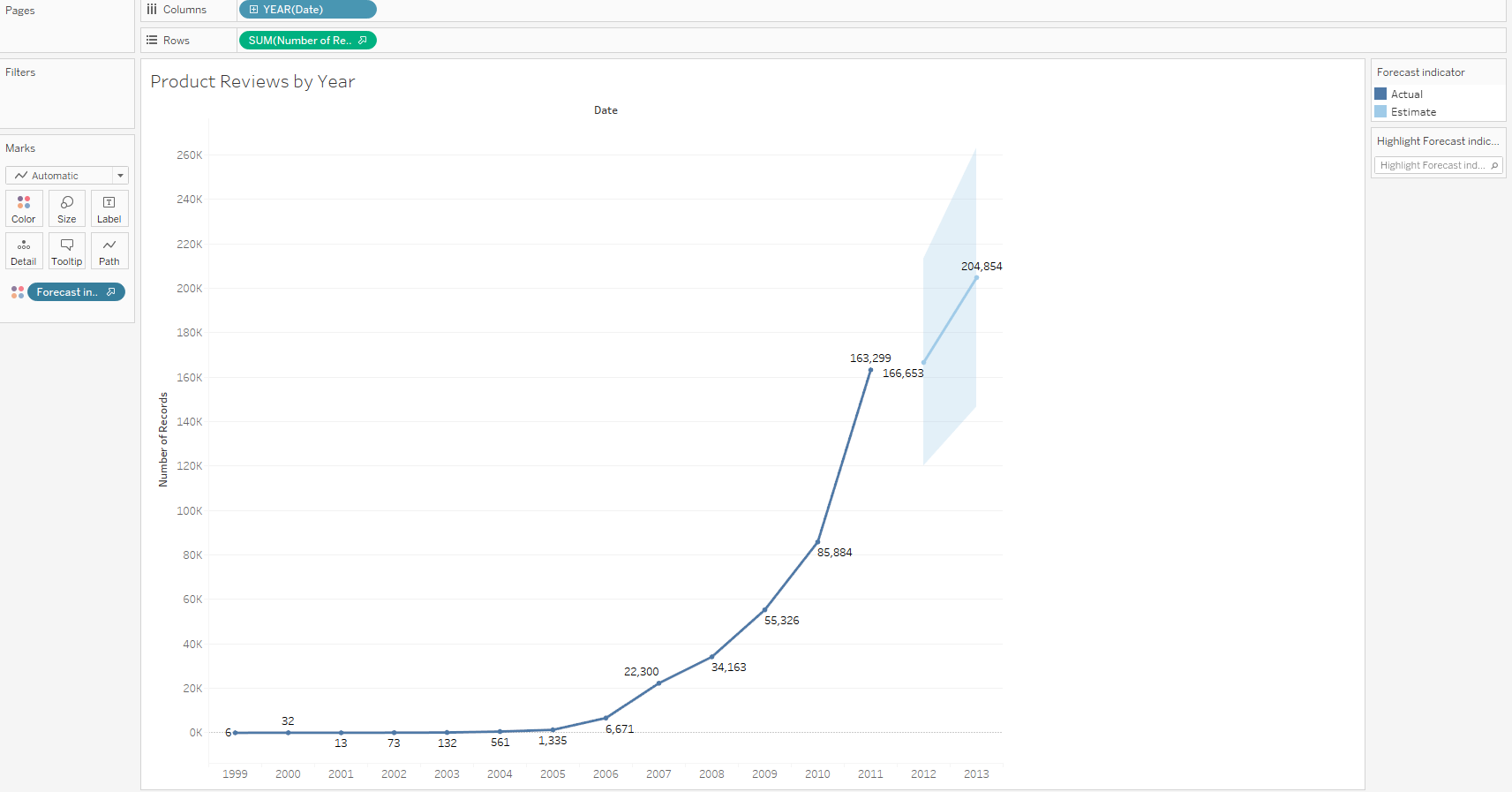
How many products got these reviews So this can tell you what were the popular products 

adding the date filter gives us popular product by years so we can see Top (n) Products as per their Popularity & how have they performed over the years we can increase grainularity of time & see subsequent changes over quarters & months



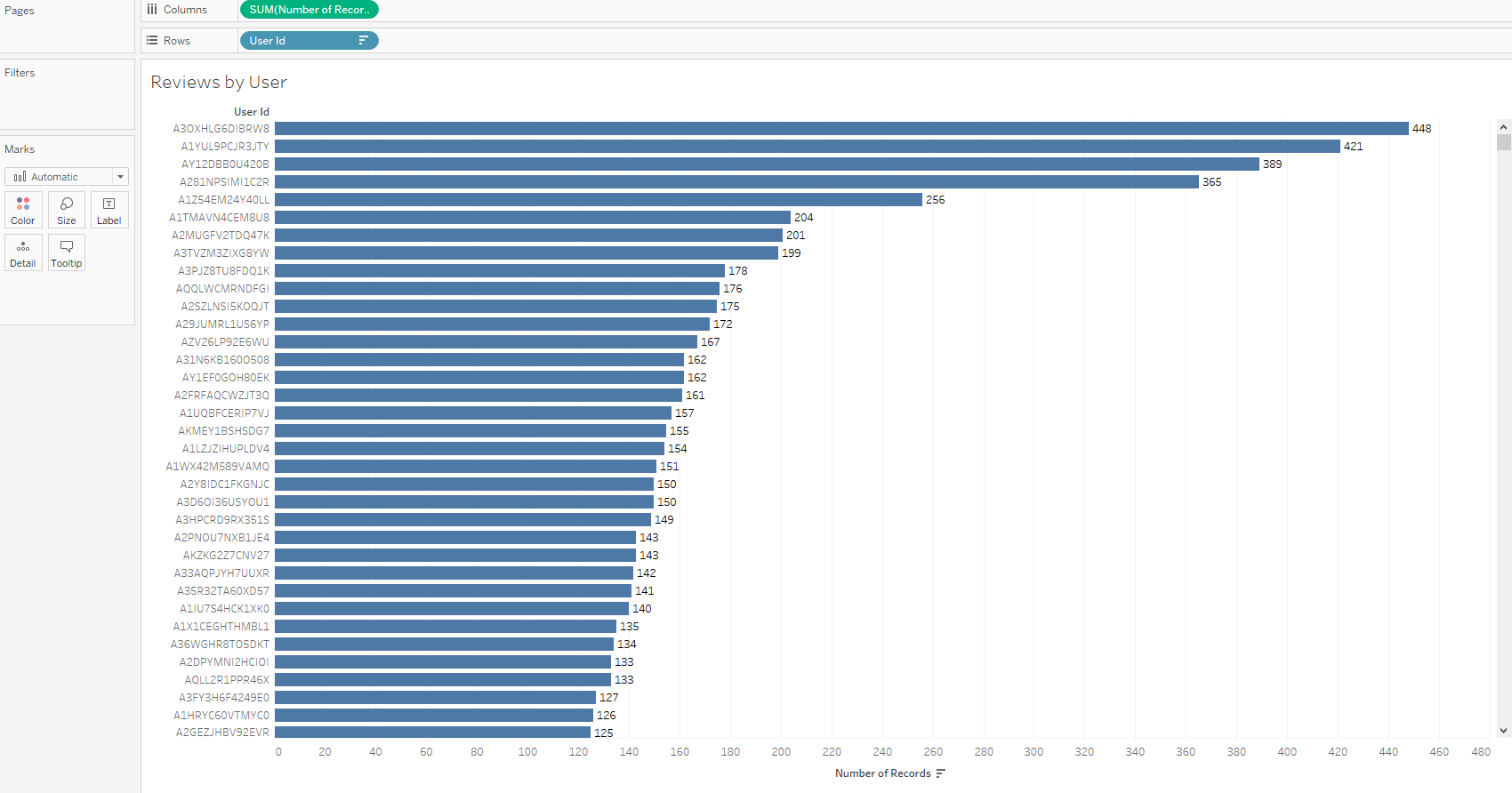
**Product Reviews by Year**

This is different as it shows the forecast of the products & every year no of product reviews increased how overall amazon products are doing as per the sales



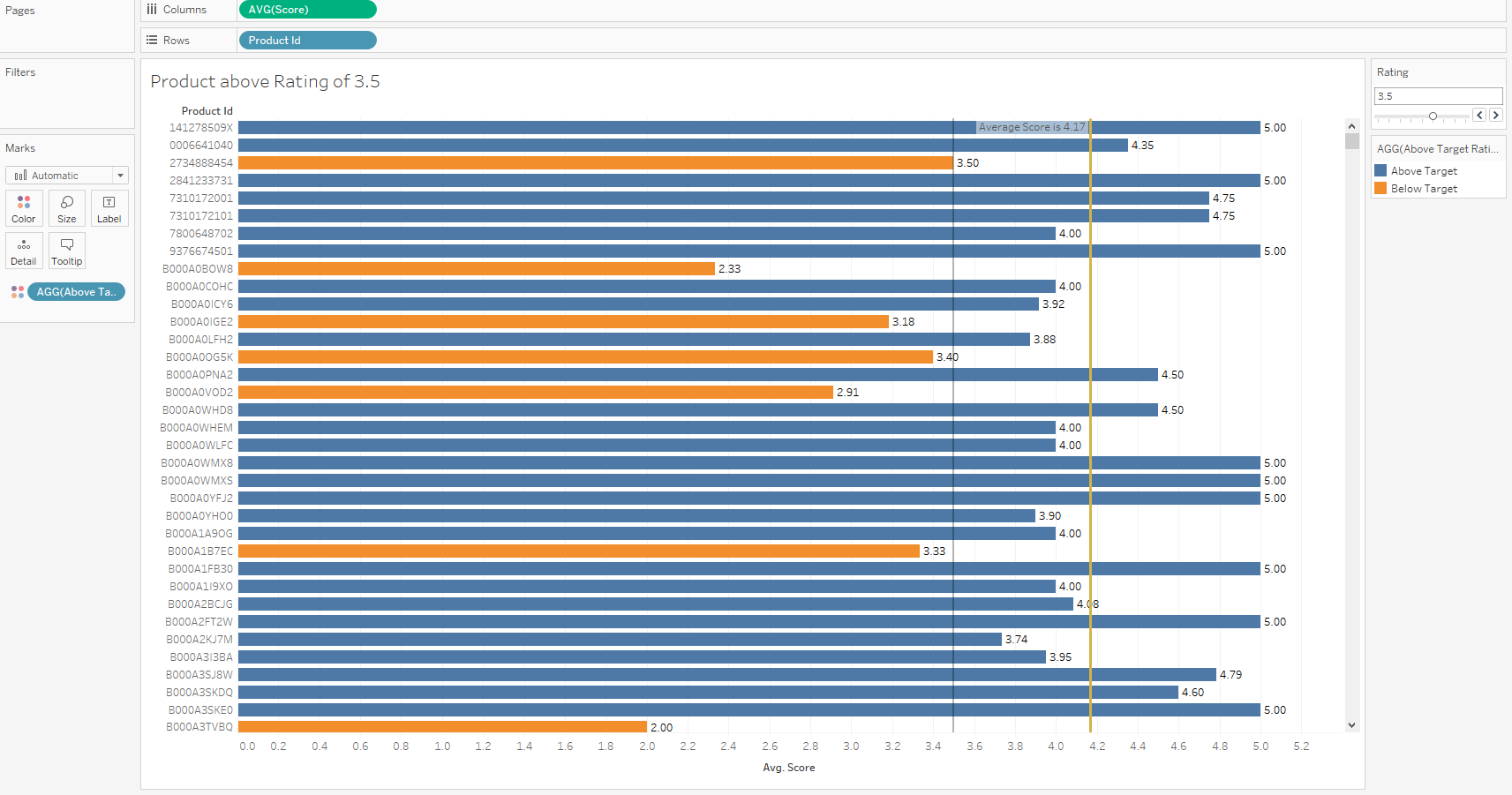
**Reviews done per User**

More reviews mean, more sales by the user, indicating customer loyalty

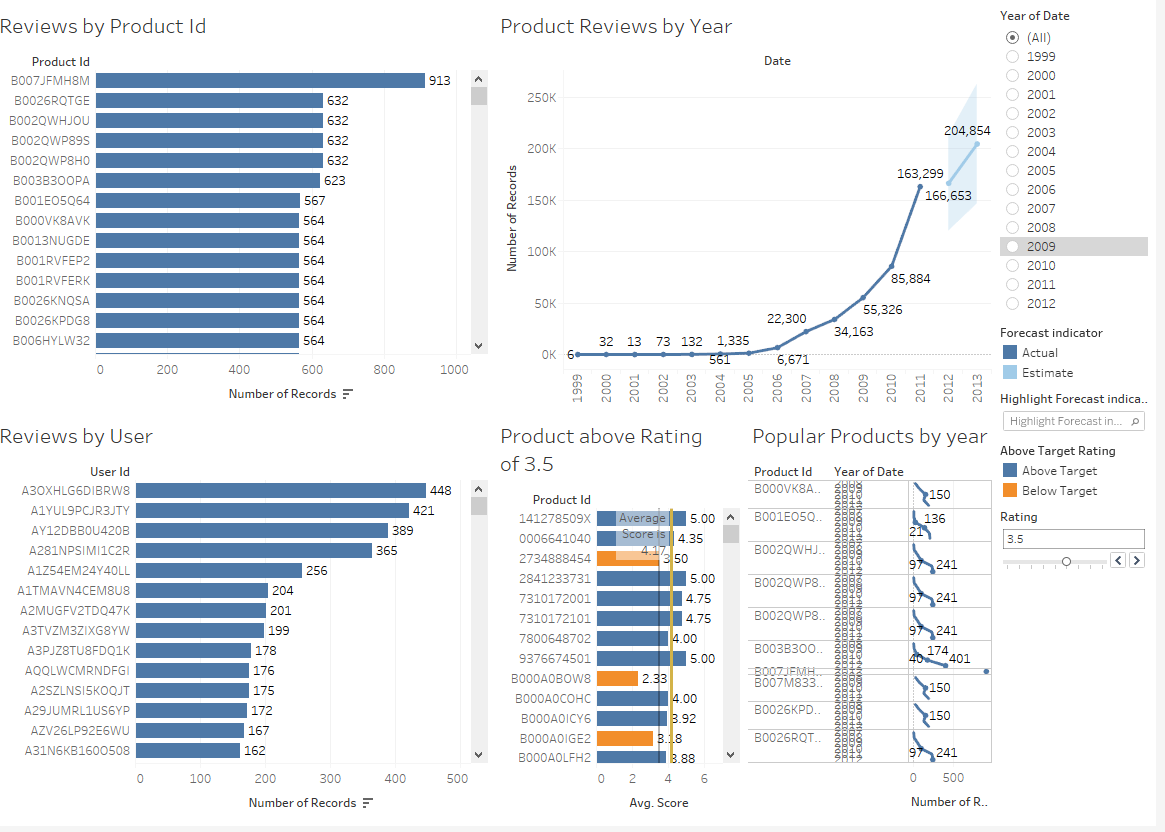


Product above Rating of <Parameters.Rating> it can be any value I have given 3.5

This can be a performance matrix & we can see what products perform above target & which remain below so Amazon can also decide be setting the parameter that which product to associate with & which not based on business strategy

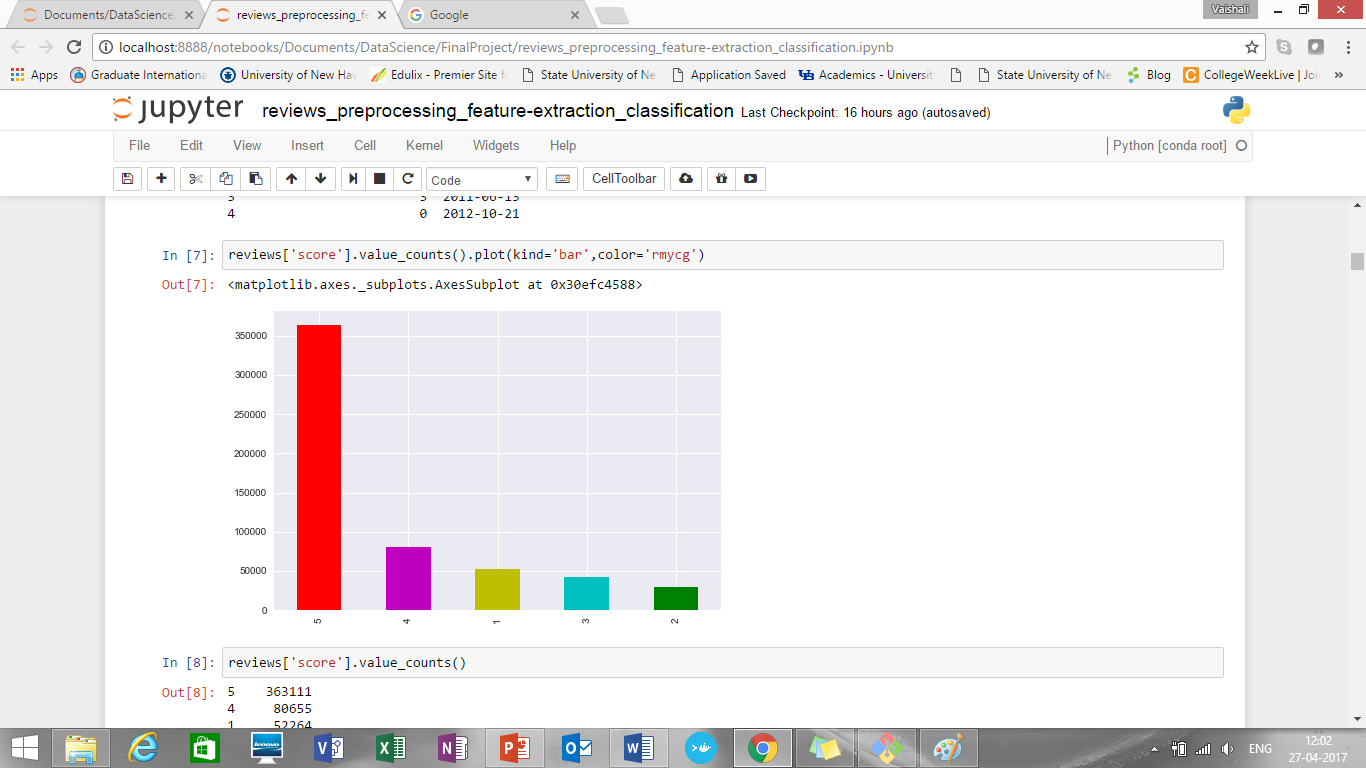


**Dashboard 2**

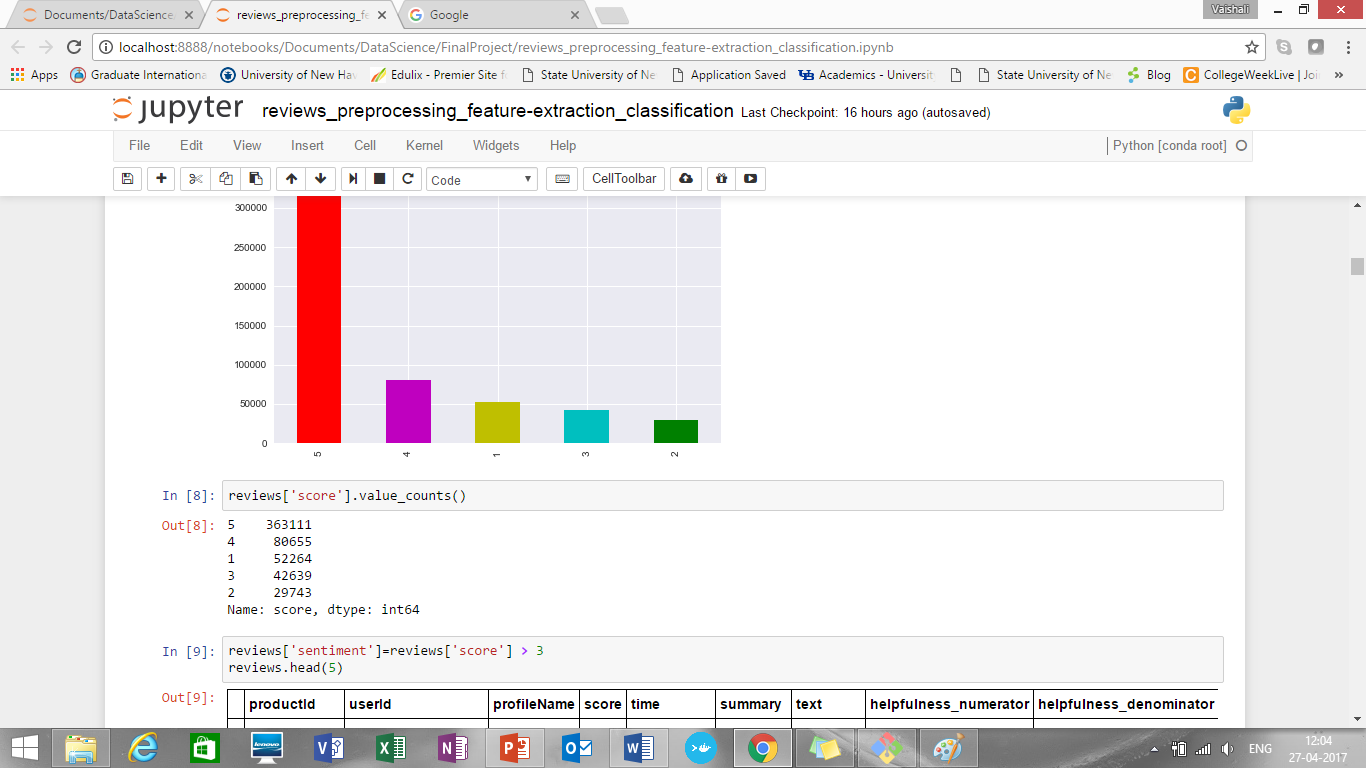


**Sentiment classification and analysis:**

* First checked score and corresponding reviews count.
* Plotted a bar graph



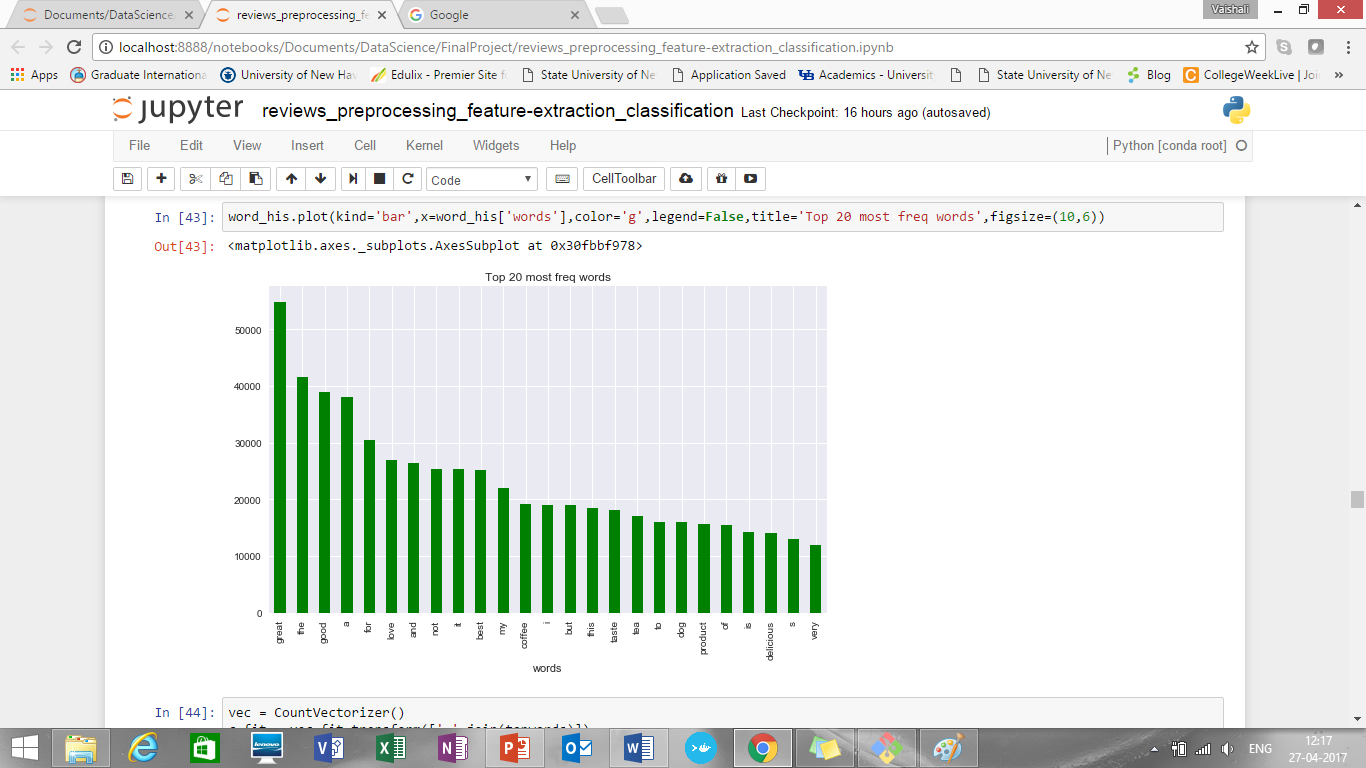
* Score and reviews count found are:



* Approximately 4.44 million reviews having score i.e. ratings 5 and 4
* Rest approximately 1.25 million reviews fall under review score 1, 2, and 3
* So, in percentage,
* Percentage for reviews with score 1, 2 or 3 are 21.93 %
* Percentage for reviews with score 5 and 4 are 78.07 %
* **Approach for sentiment classification:**
* First we divided reviews as positive and negative on the basis on score value.
* Score value > 3 are considered as positive while rest considered as negative
* After tokenizing the training data to find frequency of words we found that total “27084” unique words are present in review text
* When doing text classification the vocabulary of the data set becomes the feature set
* So, we followed feature selection option to reduce the features
* Used most frequent 5000 features/words as feature set
* When we tried to print top most frequent words found that:

[('great', 54849), ('the', 41658), ('good', 38942), ('a', 38104), ('for', 30381), ('love', 27012), ('and', 26424), ('not', 25315), ('it', 25274), ('best', 25203), ('my', 21937), ('coffee', 19164), ('i', 18992), ('but', 18956), ('this', 18423), ('taste', 18104), ('tea', 17030), ('to', 15997), ('dog', 15956), ('product', 15577), ('of', 15451), ('is', 14187), ('delicious', 14020), ('s', 12939), ('very', 12002)]

* Plotted same in graph:



* Then split data into train in test dataset (train = 0.5 and test =0.25)
* Ran three classification algorithms on it:

1. Naïve Bayes
2. Logistic regression
3. Decision trees

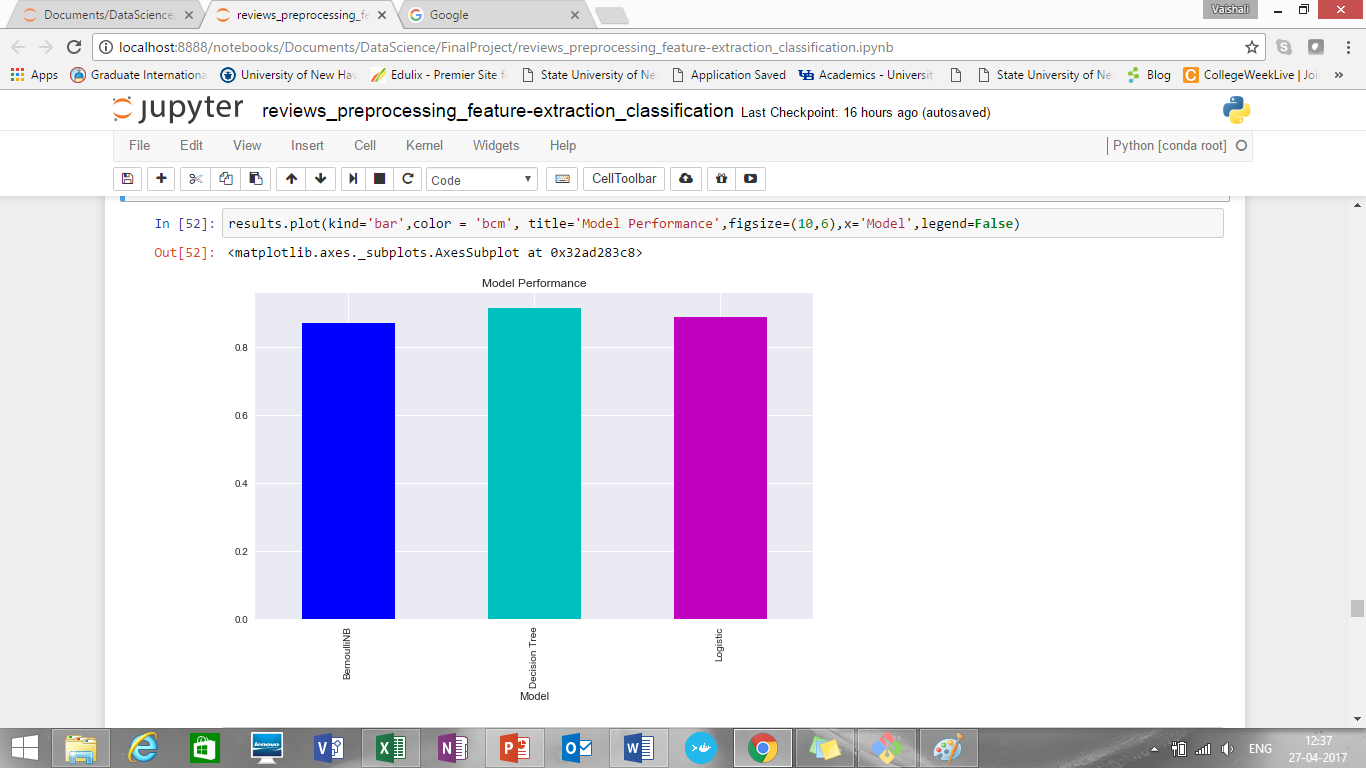
* Summary for accuracy of each model:

|  |  |
| --- | --- |
| **Model** | **Accuracy** |
| Naïve Bayes | 87.01% |
| Logistic Regression | 88.79% |
| Decision Trees | 91.41% |

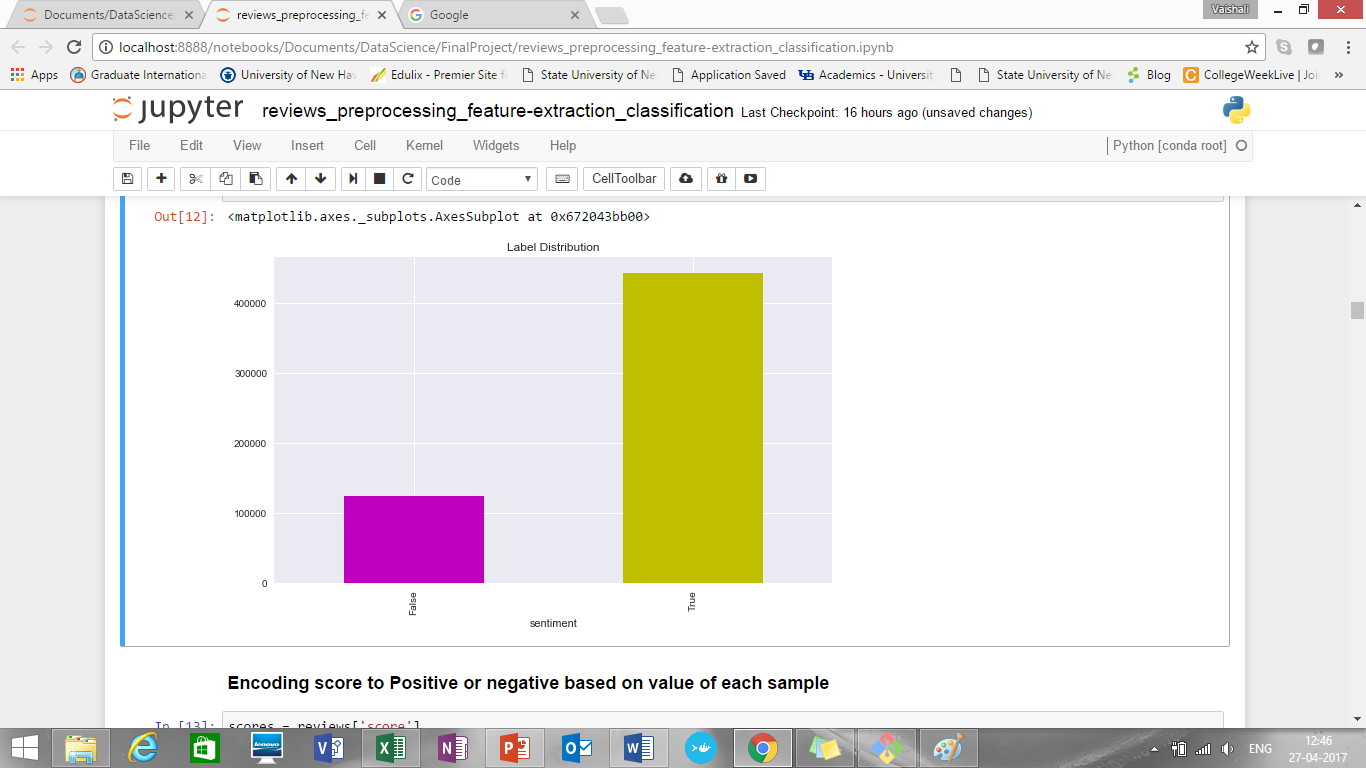
* Classification report (based on feature selection and score classification):

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Classification report for Naïve Bayes** | | |
| precision | recall | f1-score |
| negative | 0.72 | 0.66 | 0.69 |
| positive | 0.91 | 0.93 | 0.93 |
| **average** | **0.87** | **0.87** | **0.87** |
|  |  |  |  |
|  | **Classification report for Decision Tree** | | |
| precision | recall | f1-score |
| negative | 0.82 | 0.77 | 0.8 |
| positive | 0.94 | 0.95 | 0.95 |
| **average** | **0.91** | **0.91** | **0.91** |
|  |  |  |  |
|  |  |  |  |
|  | **Classification report for Logistic Regression** | | |
| precision | recall | f1-score |
| negative | 0.8 | 0.65 | 0.72 |
| positive | 0.91 | 0.95 | 0.93 |
| **average** | **0.88** | **0.89** | **0.88** |

* Plotted graph for each model performance:



* Finally, we can try by running it on our entire feature set. One can guess it will take a lot of time with Decision Tree Classifier.
* We can make use of group of words to get better results. Sometimes sequence of words might have different effect on the prediction. Sequences like "not good" or "not bad" affect the prediction in way different than when used individually
* So, we will use bigrams for it
* First added one more column sentiment label as true and false for positive and negative sentiments
* Plotted it against graph



* Number of positive and negative reviews getting were:

sentiment

False 124646 (negative)

True 443766 (positive)

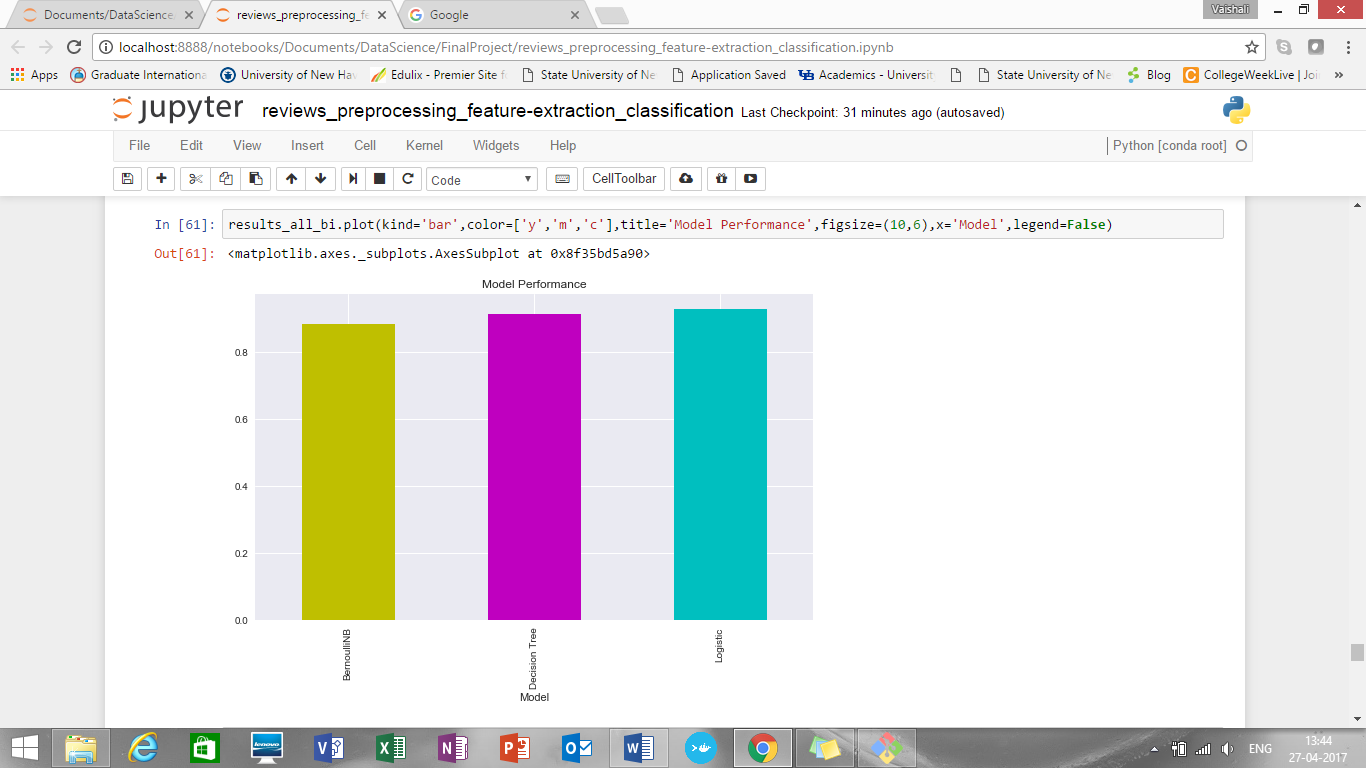
* Ran all three algorithms Naïve Bayes, Logistic Regression and Decision Tree
* Summary for accuracy of each model:

|  |  |
| --- | --- |
| **Model** | **Accuracy** |
| Naïve Bayes | 88.65% |
| Logistic Regression | 92.87% |
| Decision Trees | 91.64% |

* Classification report (based on bigram and sentiment classification):

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Classification report for Naïve Bayes** | | |
| precision | recall | f1-score |
| negative | 0.79 | 0.66 | 0.72 |
| positive | 0.91 | 0.95 | 0.93 |
| **average** | **0.88** | **0.89** | **0.88** |
|  |  |  |  |
|  | **Classification report for Decision Tree** | | |
| precision | recall | f1-score |
| negative | 0.82 | 0.79 | 0.81 |
| positive | 0.94 | 0.95 | 0.95 |
| **average** | **0.92** | **0.92** | **0.92** |
|  |  |  |  |
|  |  |  |  |
|  | **Classification report for Logistic Regression** | | |
| precision | recall | f1-score |
| negative | 0.87 | 0.80 | 0.83 |
| positive | 0.94 | 0.97 | 0.95 |
| **average** | **0.93** | **0.93** | **0.93** |

* Plotted graph for each model performance:



* Further we are populating top 5 Best and top 5 Worst reviews:

**top best reviews by sentiment score**

[1] these cookies were fun to paint but not too tasty to eat the kids did not care much but it might have been more fun if they could have eaten them they really had fun coloring them

[2] i ordered this for a friend at south carolina for christmas i want her to try some italian chocolates lol the item was arrived earlier than expected but nevertheless it was well packed and only melts on her mouth she totally loves itthree different taste and flavors of ferreros guarantees to give three different yummy joy mmmmm

[3] i'm on my first jar of this and am enjoying what it adds to my concoctions vegetarian one pan mix ups of onions garlic vegetables usually tofu rice or pasta or quinoa sometimes olives or capersi may have purchased it at uwajimaya when on salei will be buying many more jars of this mixed spice paste it’s not a paste more liquid with easily identifiable basil leaves i highly recommend this chili paste

[4] excellent shipment and product as well looking forward to purchase again thanks excellent shipment and product as well looking forward to purchase again thanks excellent shipment and product as well looking forward to purchase again thanks

**top worst reviews by sentiment score**

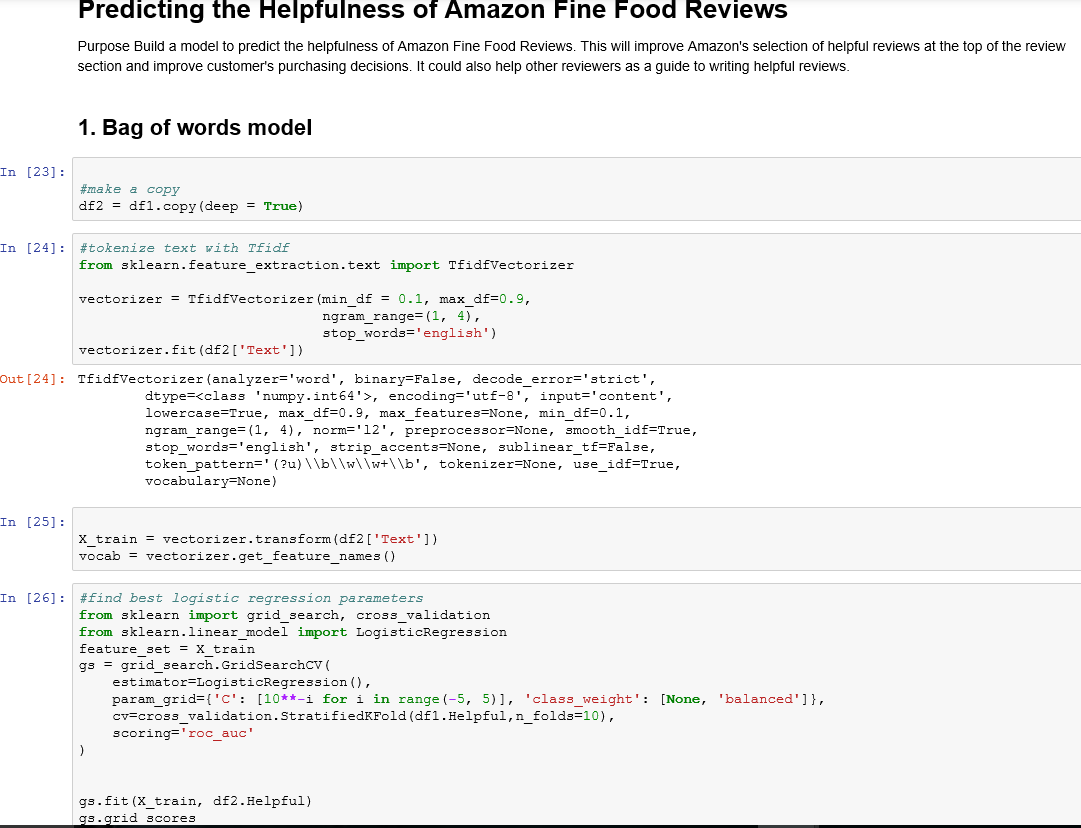
[1] my husband and i were very disappointed in this coffee very weak

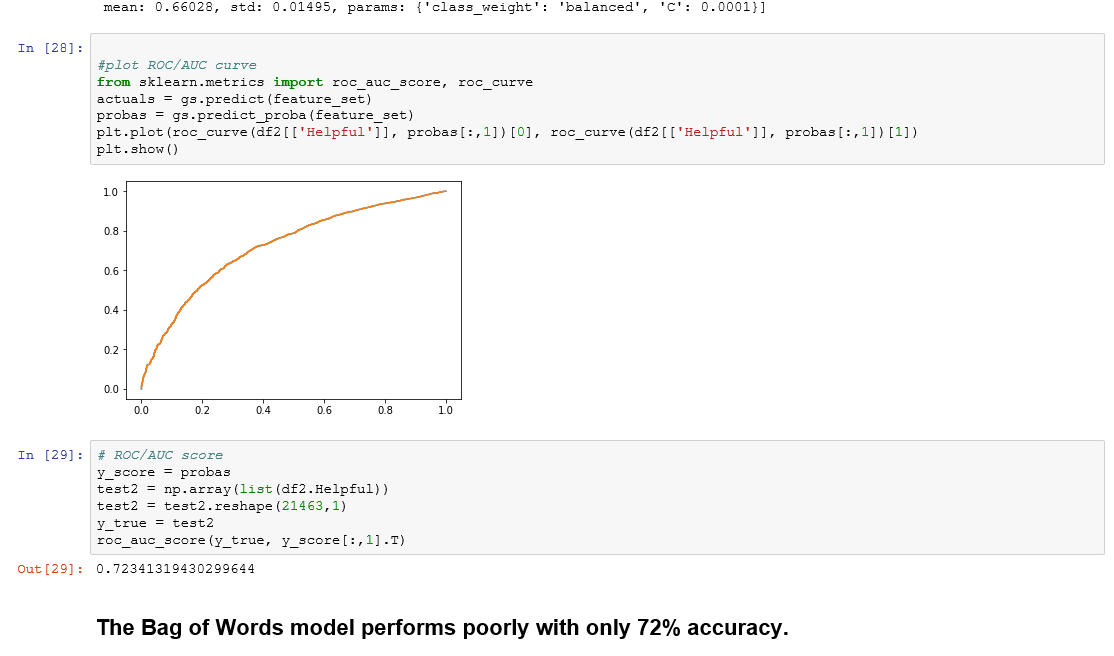
[2] i am a true seattle coffee addict and i have never had a better

[3] no doubt about it this is just the right combination of lobster scampi prawns with a hint of dry white wine & a dash of brandy finish this soup to perfection ingredients water lobster (4%) cod (35%) scampi (25%) concentrated tomato paste modified cornflour white wine prawns (15%) skimmed milk powder butterfat double cream salt yeast extract sugar shrimp powder fish powder vegetable oil vegetable extracts stabiliser (polyphosphates) brandy concentrated lemon juice spices herb and spice extracts with celery\*no artificial colours\*no artificial flavours\*no artificial preservatives information gluten free contains

[4] i will first say that i have only tried making pancakes with gf bisquick my family's experience with these pancakes however was very disappointing and i will not be trying any other recipe i have tried betty crocker's gf brownies and loved them so when i decided to spend a little extra and try gf bisquick i was greatly let down by the result we couldn't even finish them we just threw them out if you can tolerate tree nuts and milk products i recommend trying pamela's pancake mix otherwise if you are looking for great gf pancakes look elsewhere gf bisquick is not worth it in our opinion

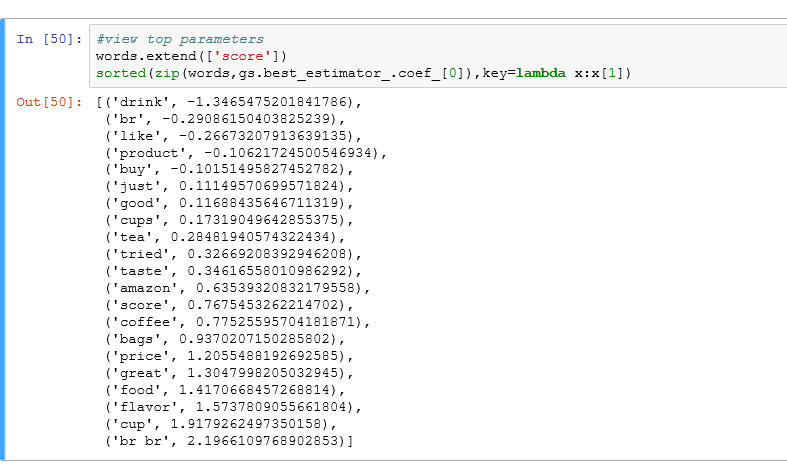
**Predicting Helpfulness of Amazon Fine Food Reviews**







**Logistic Regression to Predict Review Helpfulness with Top Cluster Words**



* Ran all two algorithms Bag of words, K-means clustering and Logistic Regression
* Summary for accuracy of each model:

|  |  |
| --- | --- |
| **Model** | **Accuracy** |
| Bag of Words | 72.34% |
| Logistic Regression | 82.71% |

**Conclusion:**

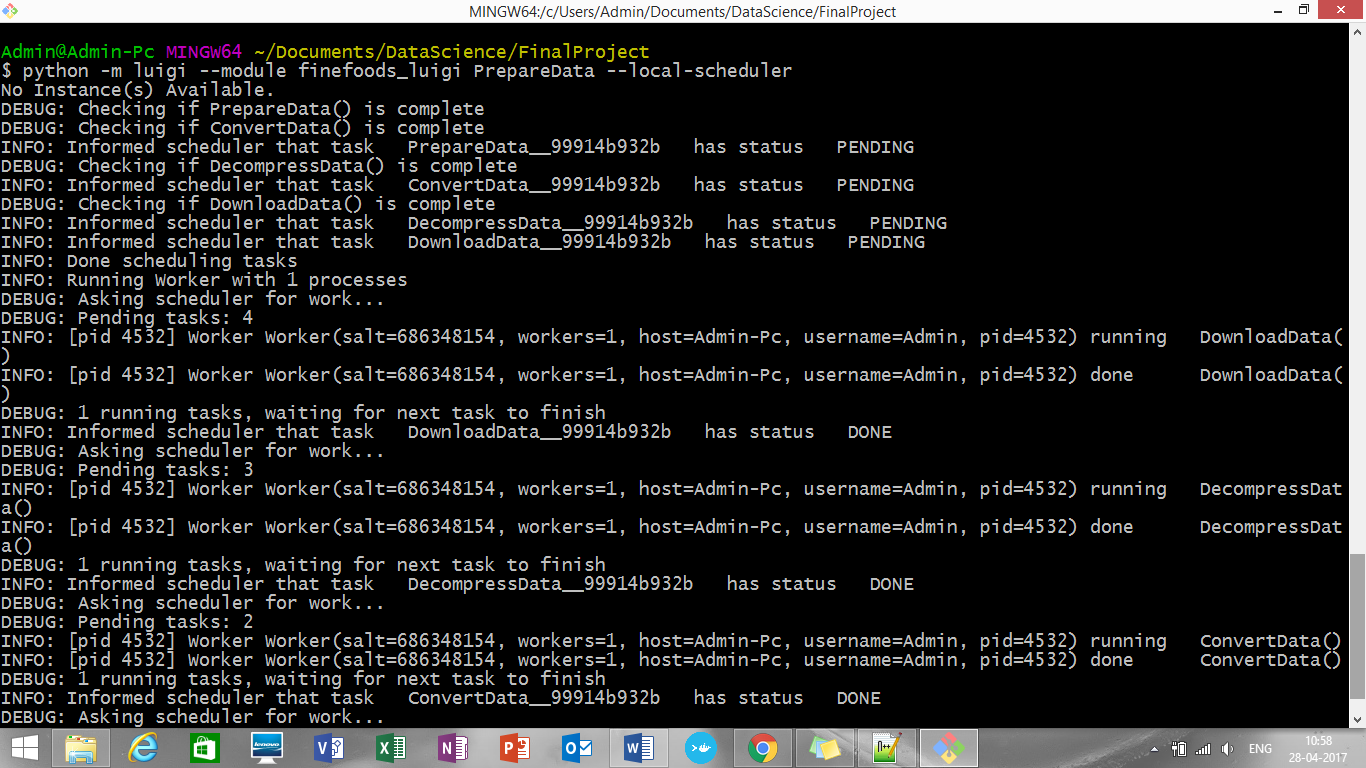
**Recommendations**

Price, Flavour, and Great are the top indicators of a helpful review. This indicates a possible bias among customers to mark a review as helpful when the review is positive. Eating, Like, Don't, Order, Good, and Eat are all negatively correlated with a helpful review, which is difficult to interpret. These may be more common words to remove.

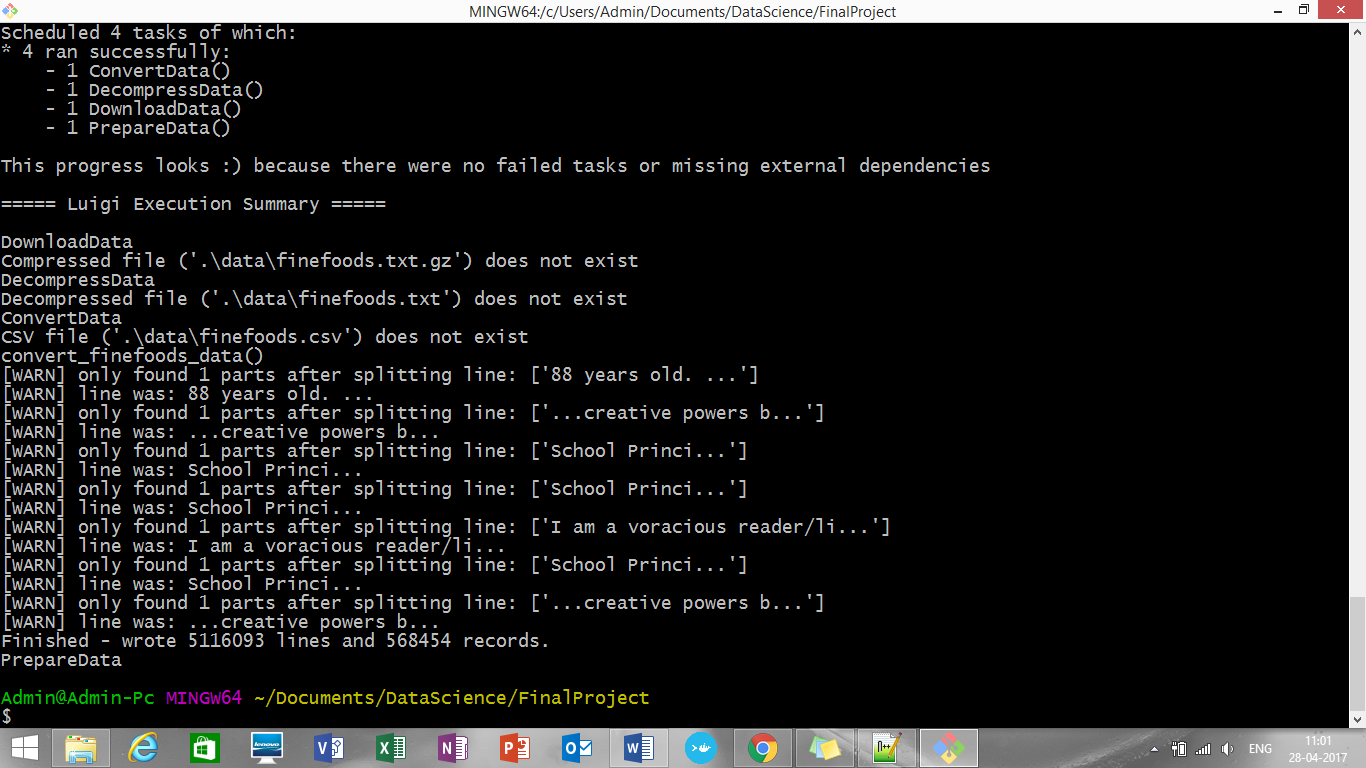
* **Luigi Pipeline:**

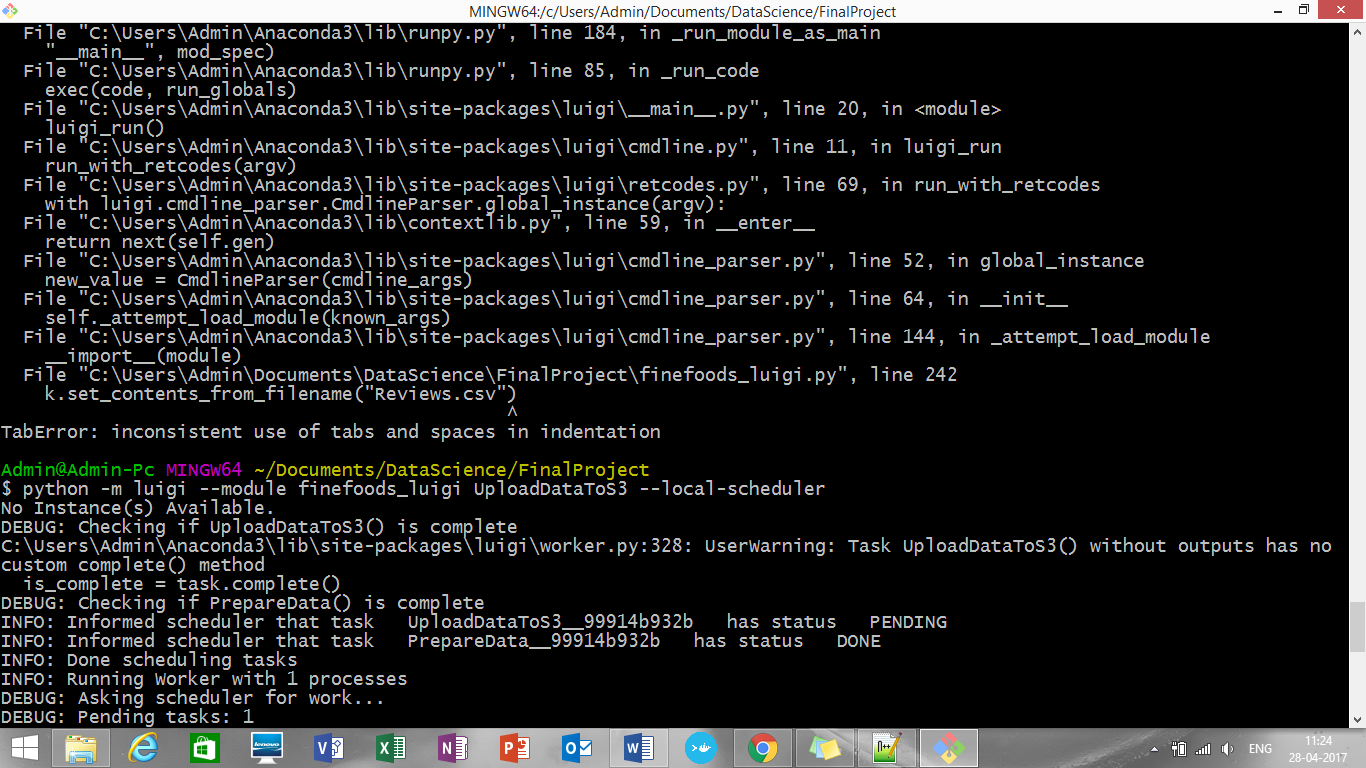
Created luigi pipeline to automate below tasks:

* Download Data
* Decompress Data
* Convert Data
* Prepare Data
* Upload Data to S3 bucket

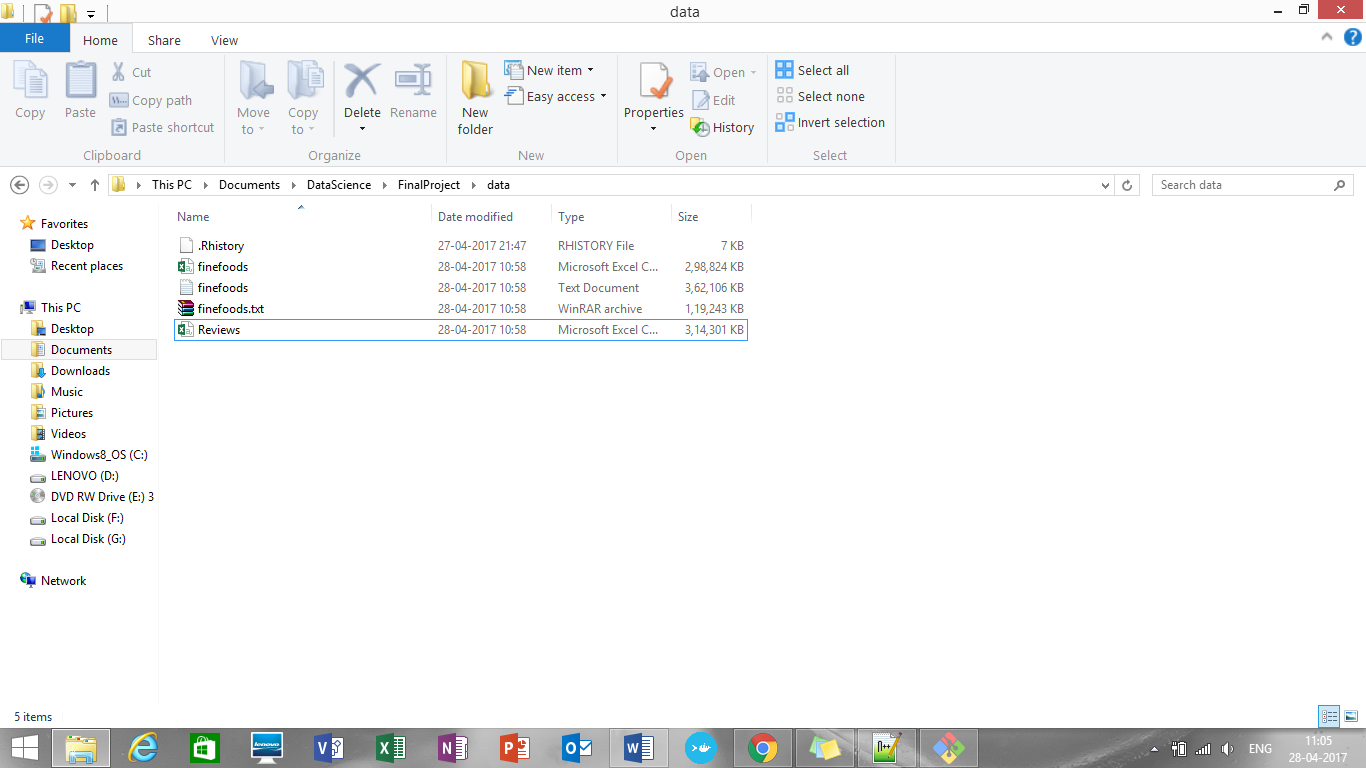








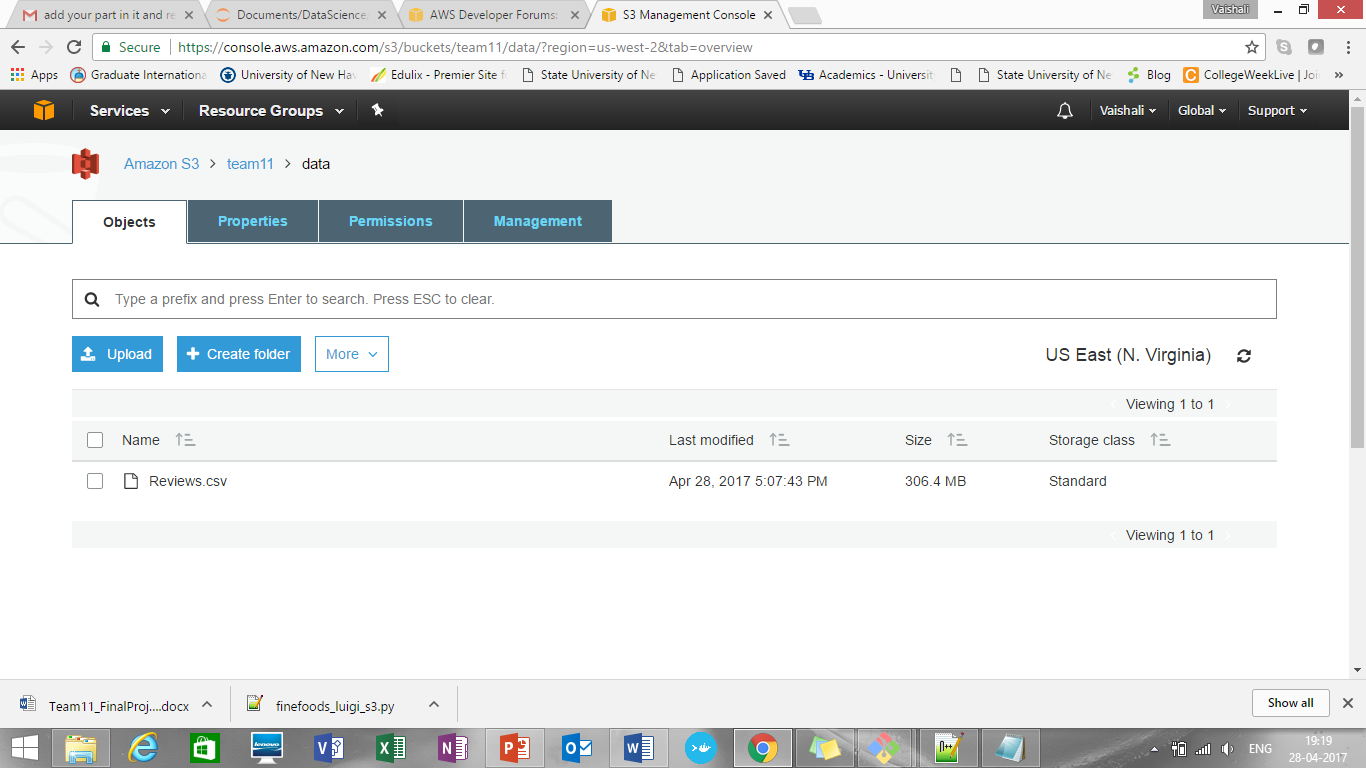
After luigi pipeline run, this is how it is getting saved on local:



As whole data, due to size won’t get uploaded, small data set needs to be uploaded

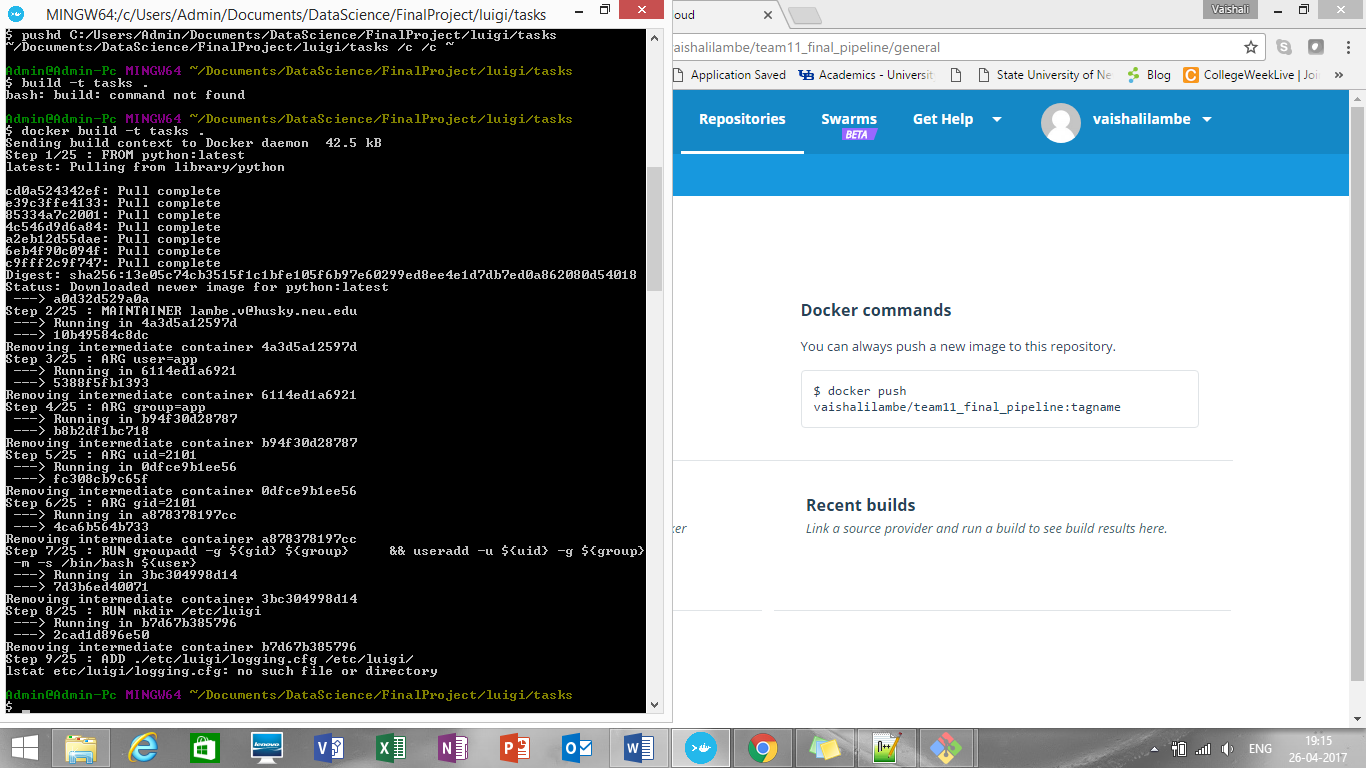
(Refer: <https://forums.aws.amazon.com/thread.jspa?messageID=713676>)

Screenshot after getting uploaded to S3.



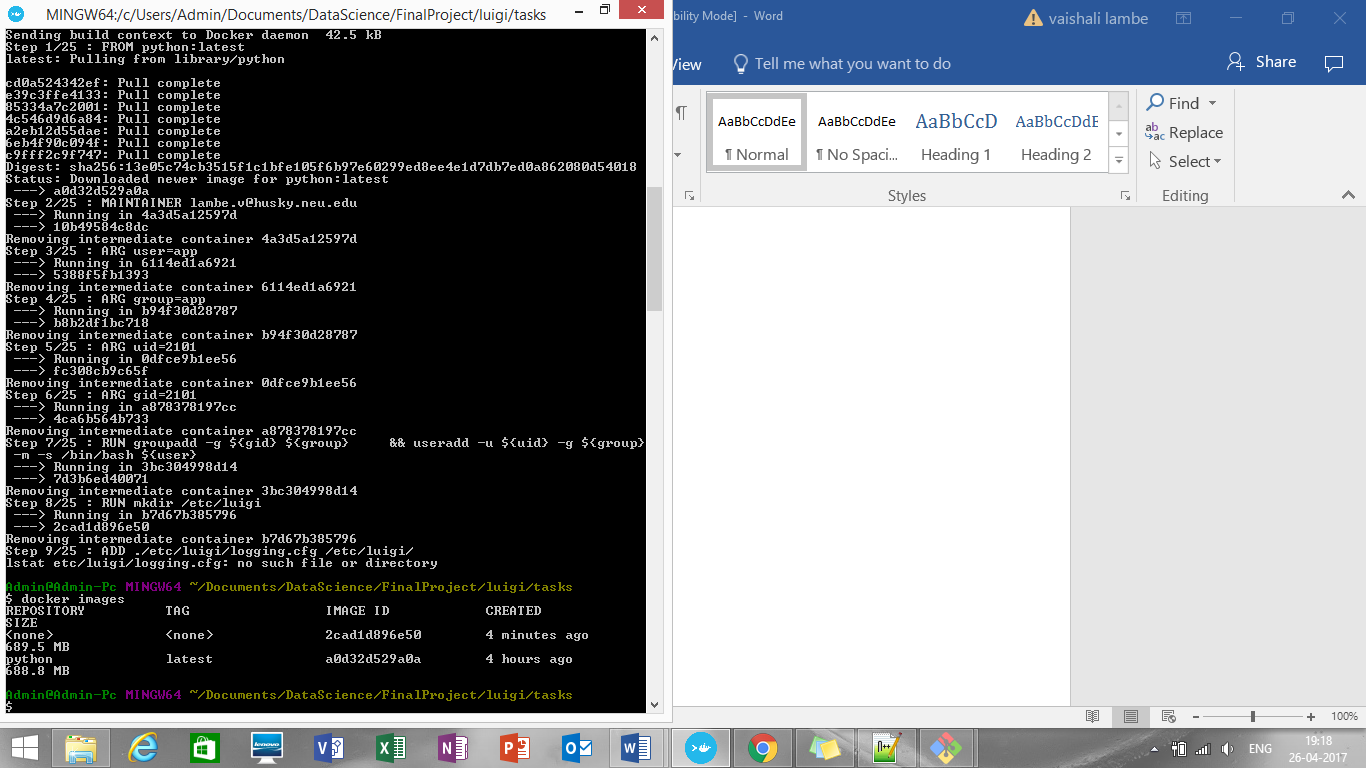
* **Dockerization of a pipeline:**

$ docker build -t tasks .



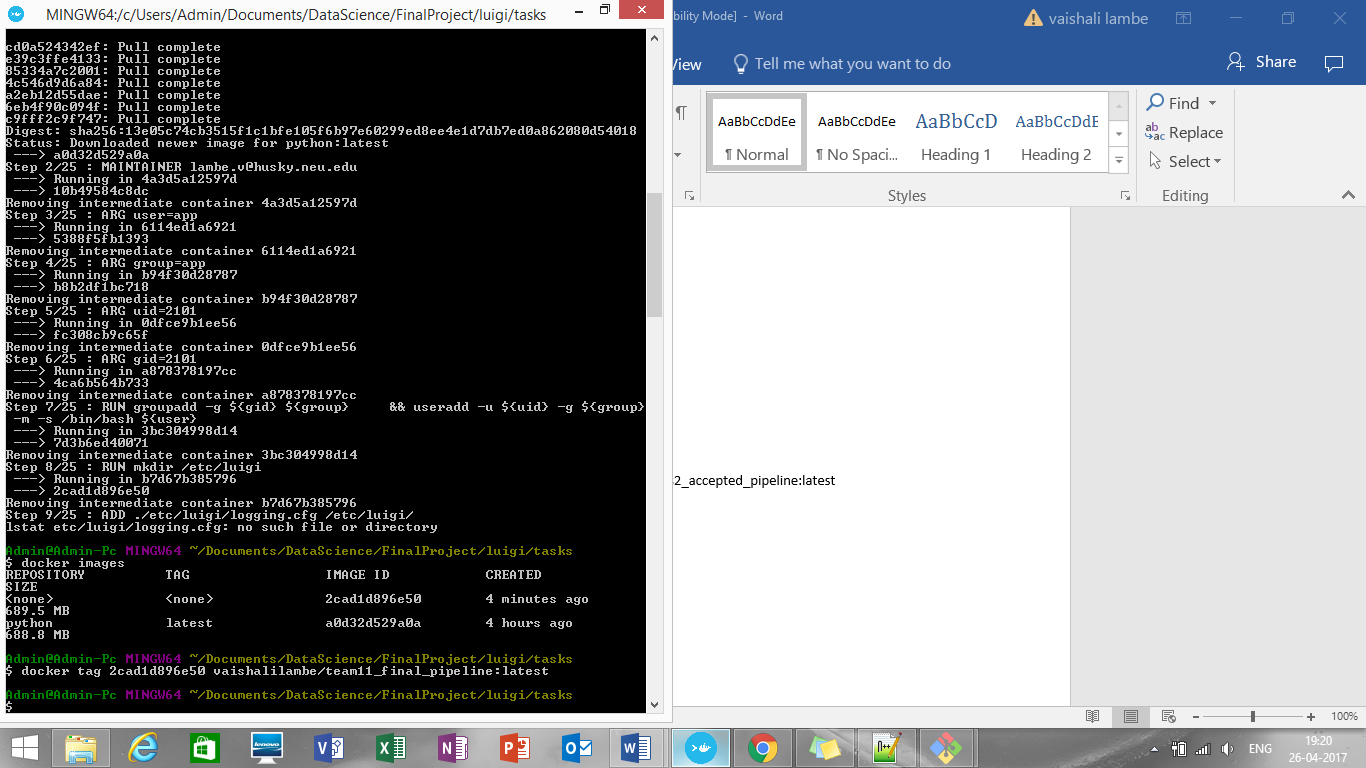
See images created:

$ docker images

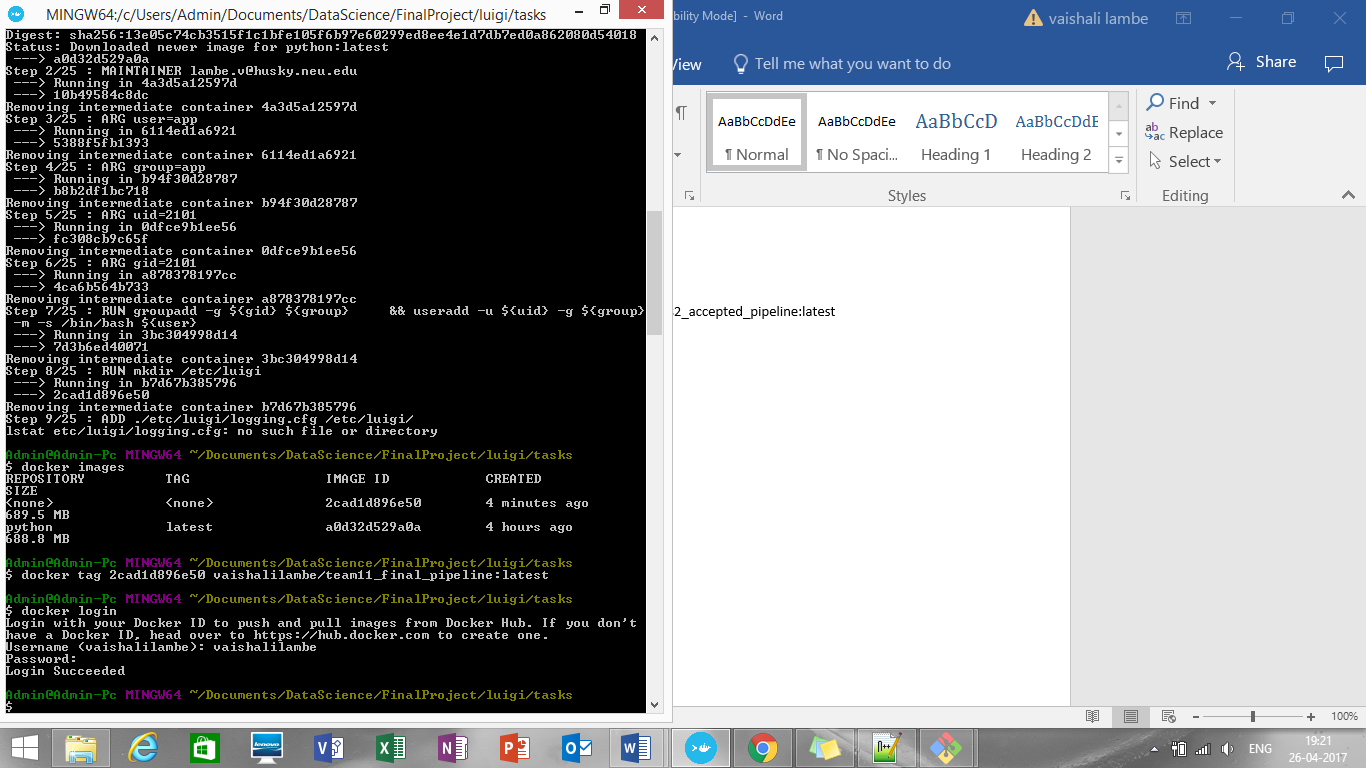


Tag the image

$ docker tag <IMAGE ID> [vaishalilambe](https://hub.docker.com/u/vaishalilambe/)/[team11\_ass2\_accepted\_pipeline](https://hub.docker.com/r/vaishalilambe/team11_ass2_accepted_pipeline/):latest



$ docker login

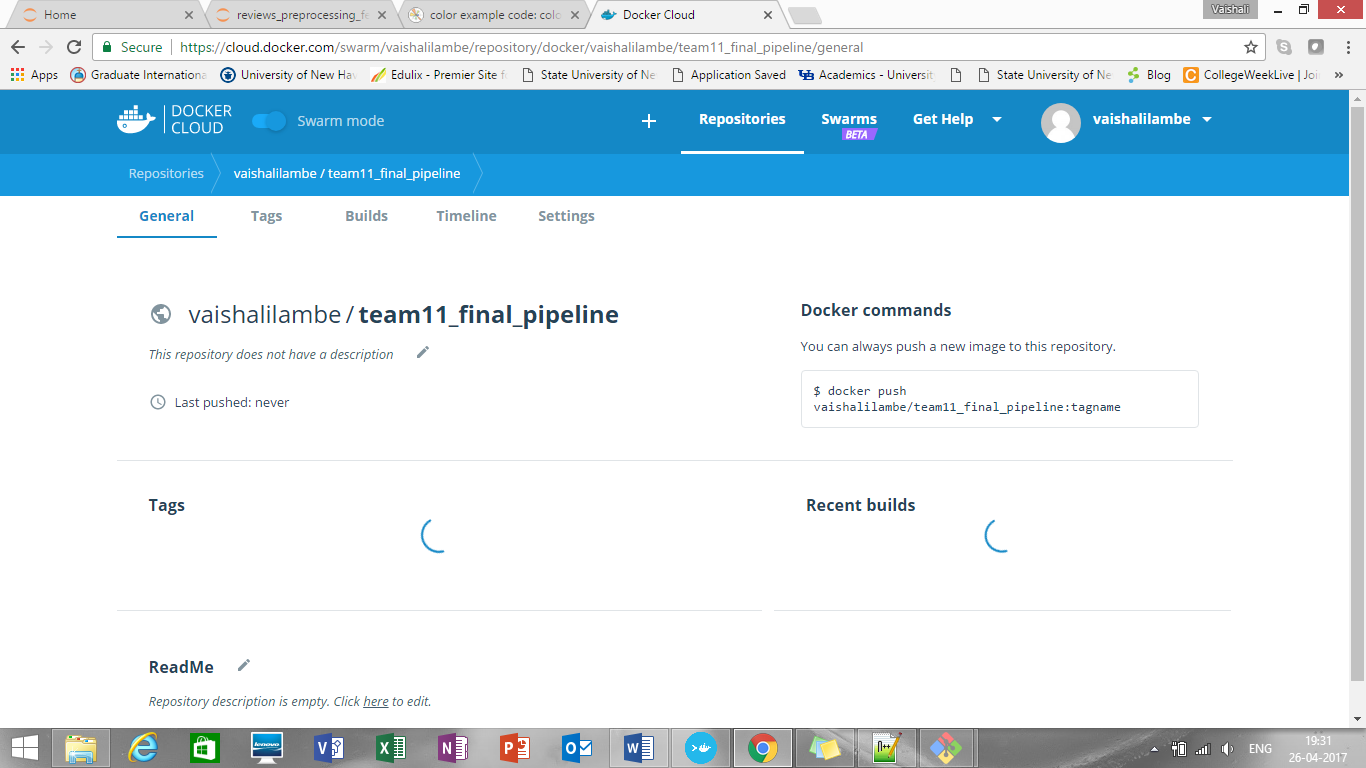


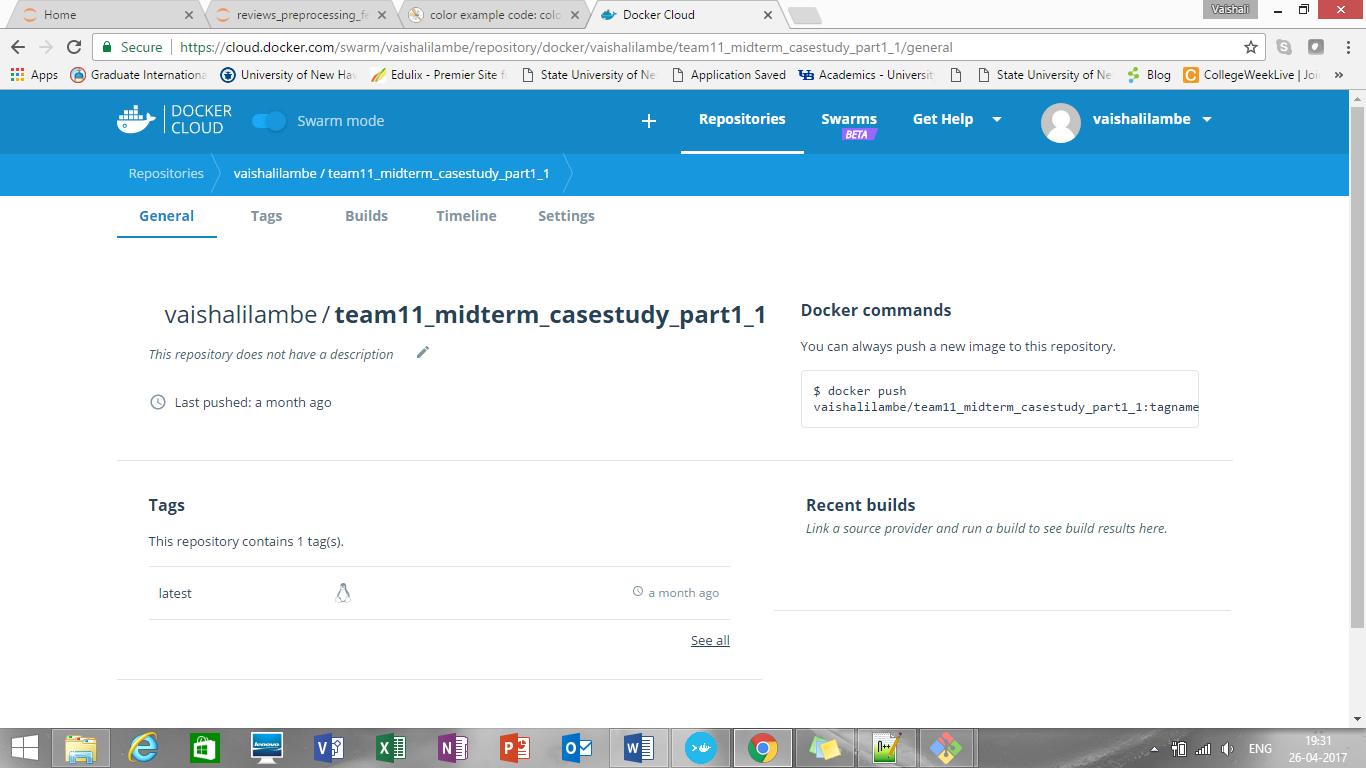
To run the image:

$ docker run vaishalilambe/team11\_final\_pipeline or

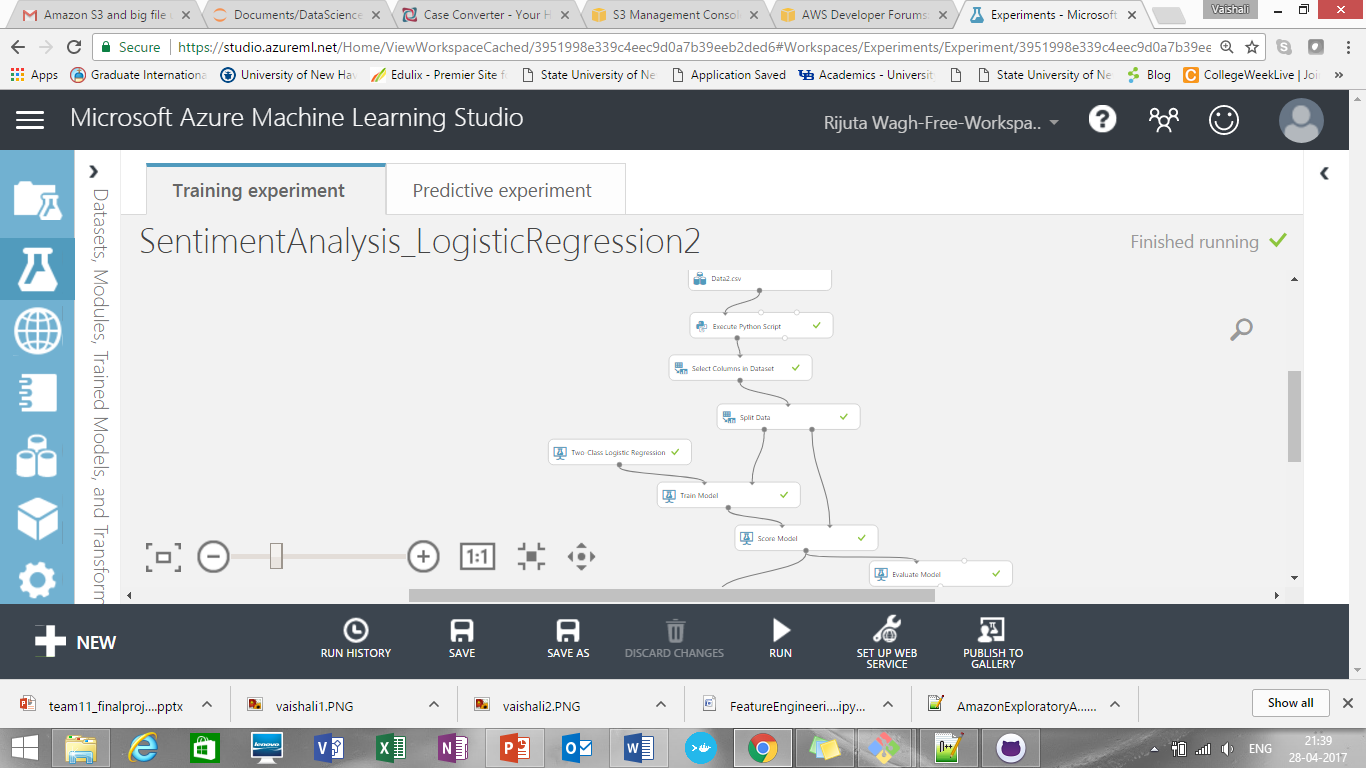
$ docker run -v //c/Users/Admin/Documents/DataScience/FinalProject/data:/home/docker/data vaishalilambe/team11\_final\_pipeline

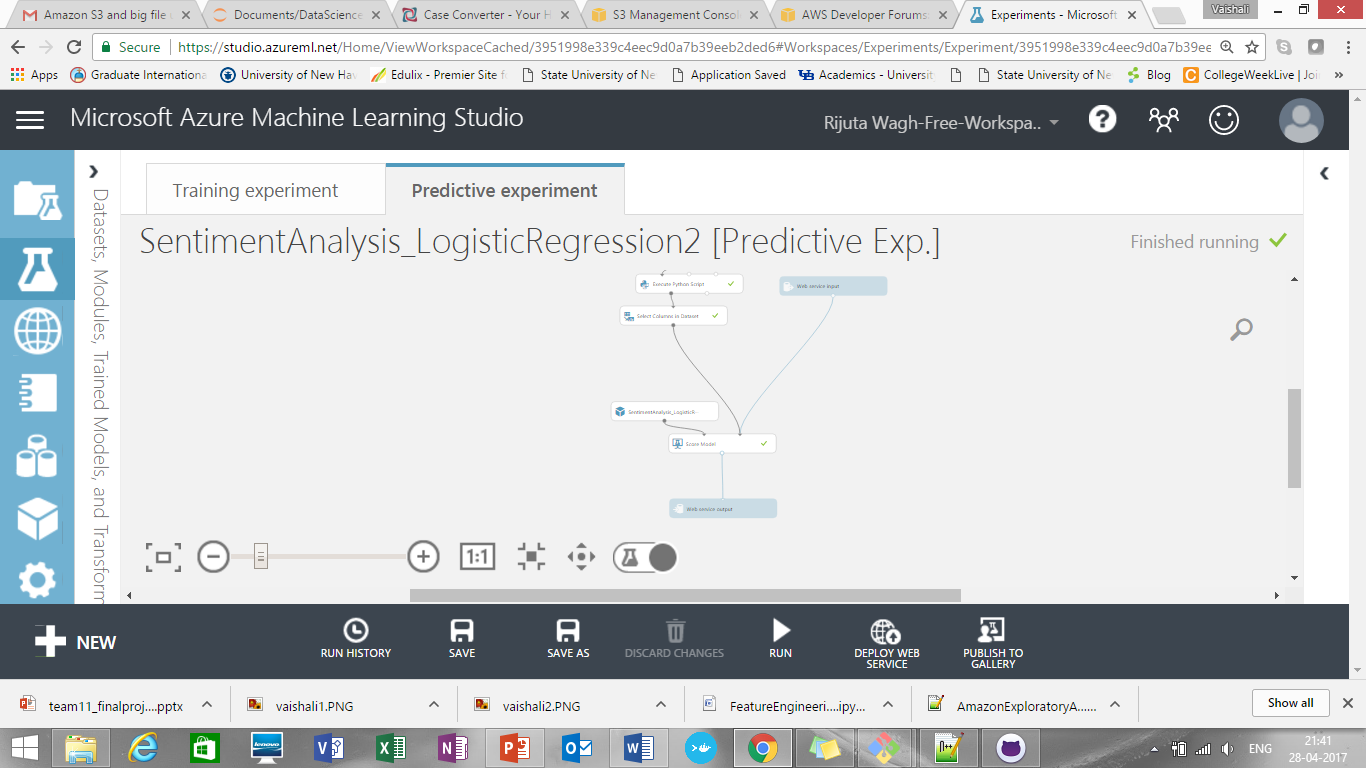
Docker hub:

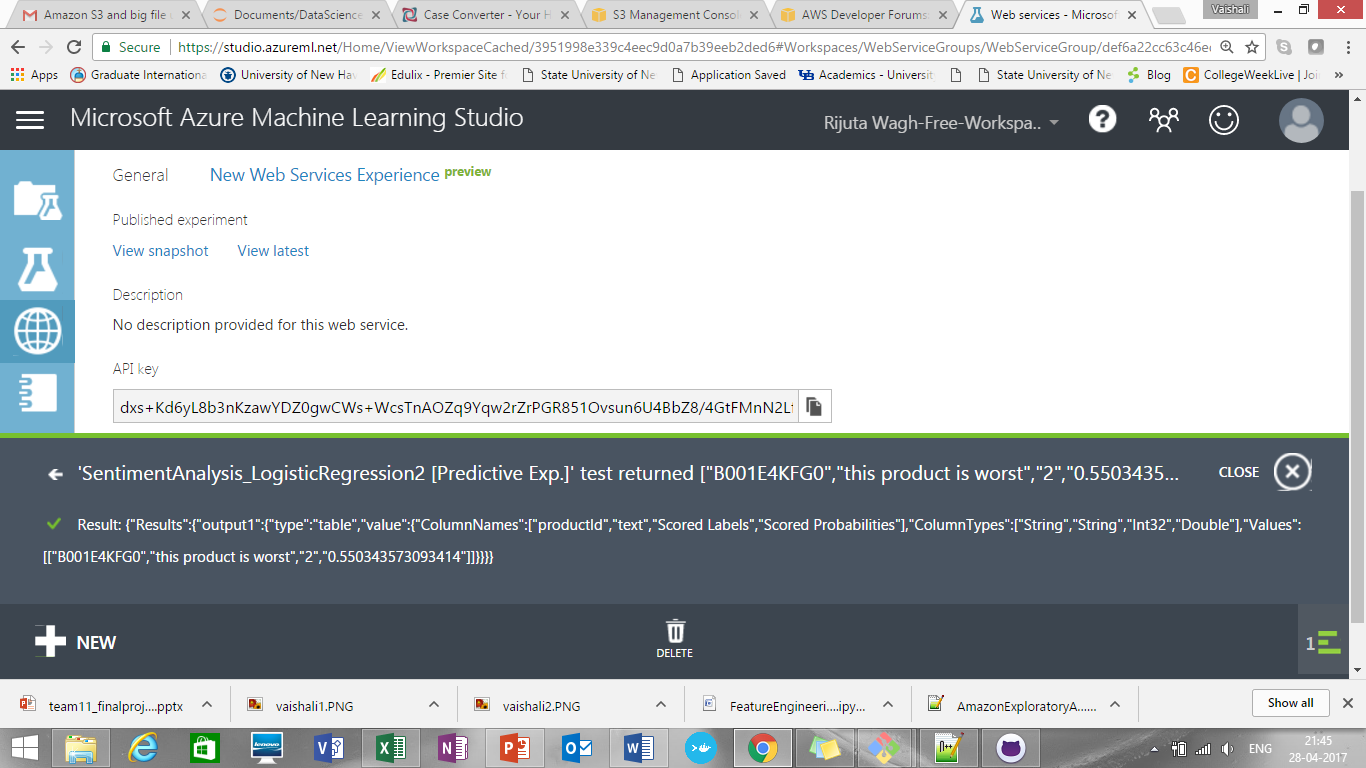




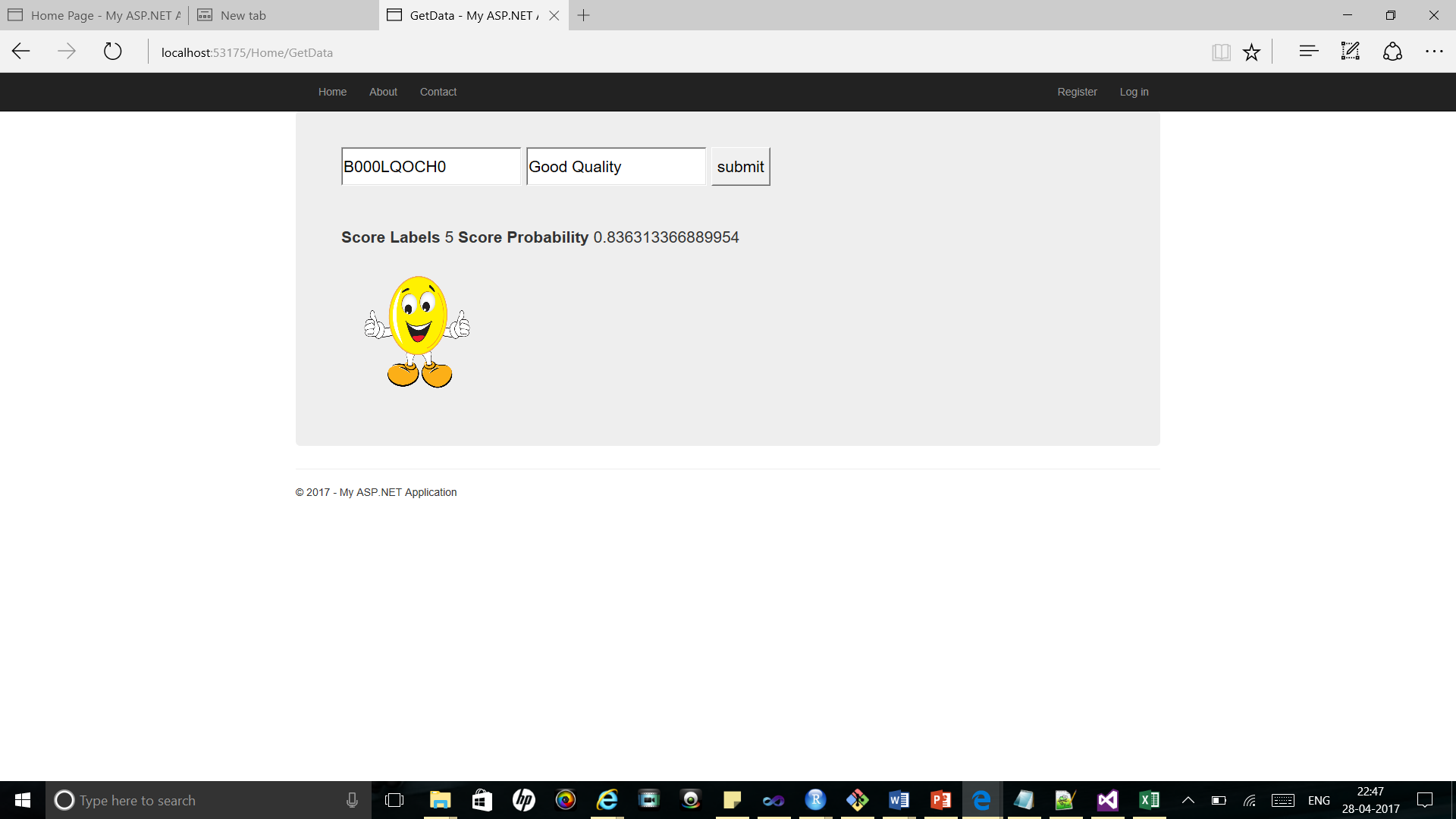
Microsoft Azure ML Studio (Rest API)







After web service deployment output, we are getting at UI side:



Contribution Pie Chart:

**Tasks done:**

**Vaishali and Rijuta:** Rest API, Classification and sentiment analysis, Report documentation, Power Point presentation

**Rijuta:** Data exploratory analysis using python and R, UI, K means clustering-prediction

**Vaishali:** Data Download, pre-processing, Conversion to .csv, Luigi pipeline, Dock erization of pipeline, Activities tracker, Git hub deliverables upload

**Ankur:** Data exploratory analysis using Power BI