Amazon User Review Analysis with UI

Shiva Baghel[#1], Sonal Priya[#2], Harsh Raj[#3], Nirpesh Joshi[#4], Nishant Singh[#5]

[#]*B.Voc (Robotics &AI), Dayalbagh Educational Institute,*
Agra (282005), U.P, India

Notebook Link - https://bit.ly/2WrwYNM (nlp_mod.py)
https://bit.ly/32uT85A(webpage.py)

Abstract

Reviews are the most viewed section in-order to grab the information about a product like the material, efficiency etc. and reviews are more important for Amazon (MNC). In this paper, we classify reviews into categories by using several NLP libraries and algorithms also, extracted companies mentioned. As an addition, we have done sentiment and created a webpage for the ease of understanding. The acquaintance was provided to every customer about the products by this research.

I. Introduction

In today's World online shopping is a need for everyone. Amazon is one of the biggest names among the largest online vendors in the World. Reviews plays an important role while selecting the product. Till date amazon's reviews are only classified on rating (1-5) and the search bar prefers the product name not the review or the category to search a product. So, in this paper we explore some ways which can be used to improve the searching efficiency and ease to understand the review. By performing keyword extraction and Sentiment Analysis.

To do so, we have taken an Amazon review dataset contains about 40,000 reviews and 16 columns containing relevant information about the product. The dataset of related to toys and building material type products. The dataset was messy and reviews are not given as expected, we had extracted the reviews along with their rating and perform several Natural Language Processing (NLP) techniques to clean the reviews like remove punctuation/stop-words, lemmatization, html tags removal.

We worked on 3 model to get the best accuracy. Firstly, we tried Doc2Vec with Logistic Regression. Doc2Vec converts a review into vector so that it is easier to compare with other reviews. Secondly, TF-IDF a very famous approach in NLP. We applied TF-IDF on every review and find their cosine similarity based on the vectors of TF-IDF. Finally, we used Word2Vec model of Google and trained our own model on top of it.

At last we created a webpage to show the results in a better way, so that everyone can see our work in an interactive and friendly manner.

II. Literature Survey

MNCs like Amazon must have considered every aspect of user's desire. If we talk about the searching algorithm and reviews analysis, then the lack of categories-based search and review classification is observed. So, we have researched on this topic that how to can predict the category given a review and search products based on that category.

Till now in amazon review or search bar does not support category-based search. We took a dataset of amazon (basically of toys) and tried various Machine Learning techniques to do so, here is our research-

Before theory part lets understand whole research overview with the help of a flow chart.

The given flow chart includes the steps involved from Data cleansing to Final Model training.

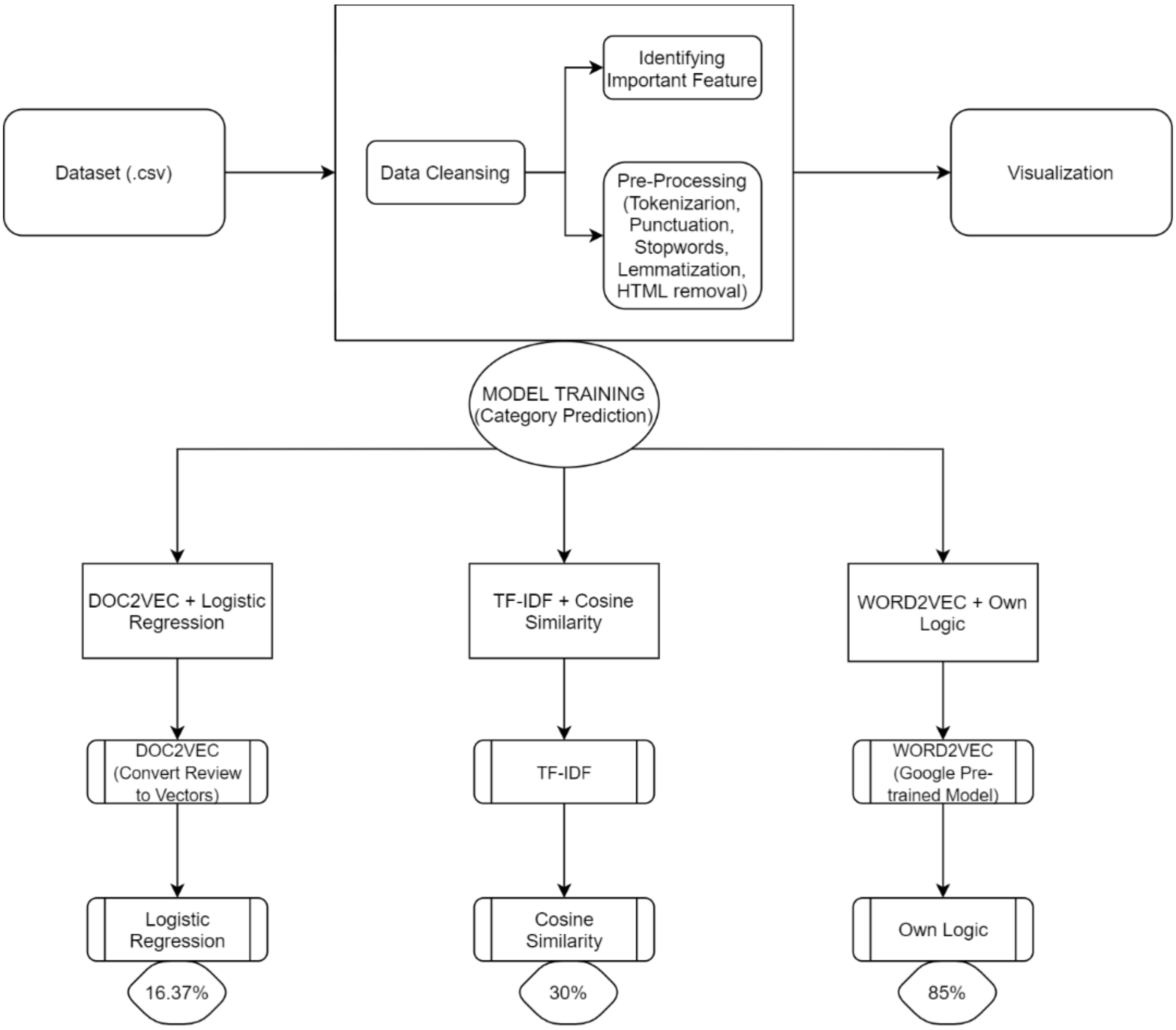

Dataset (.csv)
Data Cleansing
Identifying Important Feature
Pre-Processing (Tokenizarion, Punctuation, Stopwords, Lemmatization, HTML removal)
Visualization
MODEL TRAINING (Category Prediction)
DOC2VEC + Logistic Regression
TF-IDF + Cosine Similarity
WORD2VEC + Own Logic
DOC2VEC (Convert Review to Vectors)
TF-IDF
WORD2VEC (Google Pre-trained Model)
Logistic Regression
Cosine Similarity
Own Logic
16.37%
30%
85%

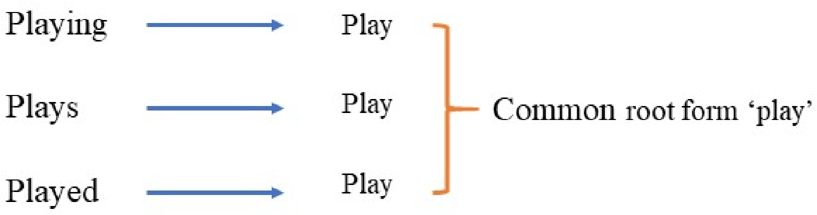
Fig. 1 Process Summary

A. Dataset Insights and feature extraction (pre-processing)-

The dataset is of Amazon with about 40,000 reviews of different toy companies and 16 columns with relevant information.

Now for filtering we identify that some of the columns are not necessary for the future processing they just increase complexity. So, we eliminated those columns.
We extract reviews form the given dataset w.r.t[1] to each manufacturer and its category.
After extraction, data cleaning was performed because we do not want ant impurity in our final feature. For that –

a. Tokenization
b. Punctuation Removal (./?@%#$|)
c. Stop Words Removal (and, or, hi, at)
d. Lemmatization



[1]

e. Html tags Removal – (https://saja, www.dfjdjf.in)are done using several Natural Language

Processing library present in library –
a. NLTK [2]
b. Spacy [3]
After pre-processing we got our processed data and we are ready to perform some NLP and ML techniques.

B. Visualization-

Before training we did some visualization on our data to understand the trend and relationship between columns. Visualization is performed by some python library –

i. Matplotlib [4]

ii. Seaborn [5]

iii. Altair [6]

iv. Plotly [7]

a. Word Cloud (Positive and Negative words) w.r.t manufacturer-

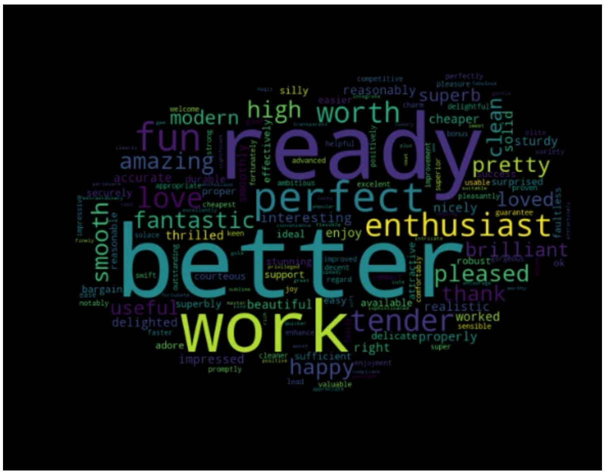


Fig. 2 Postive Words

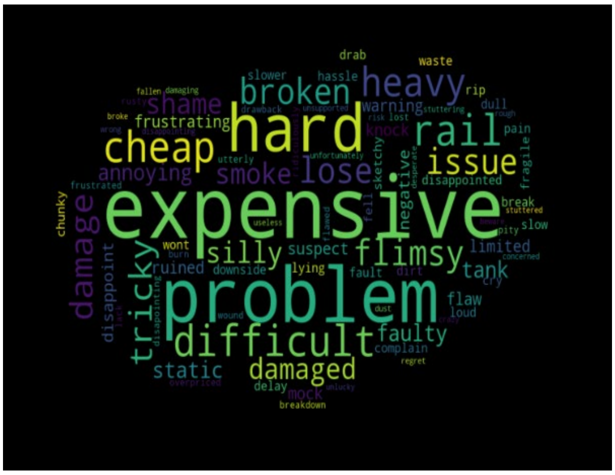


Fig. 3 Negative Words

- From Fig.2, words like “better” and “work”, are used most, that means the toys are in perfect condition and ready to use.
- From Fig.3 “expensive” is most used, means that most of the products are costly.
- Also, from Fig.3 we get that there is some problem with products as words “hard”, “problem”, “difficult”, “flimsy” are more visible.

b. Manufacture v/s price based on category –

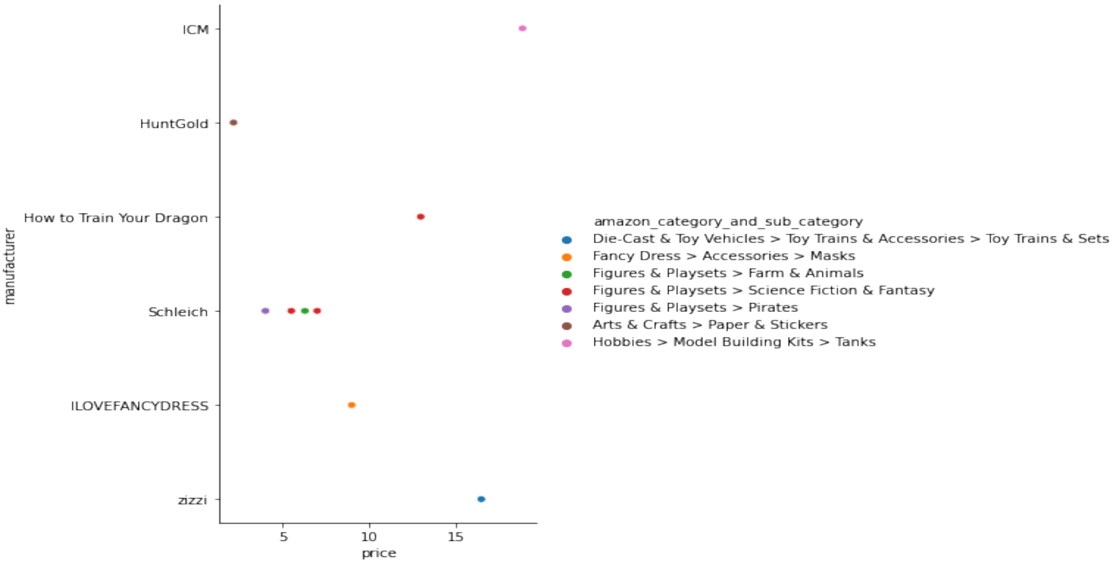


Fig. 4 Top manufacturers

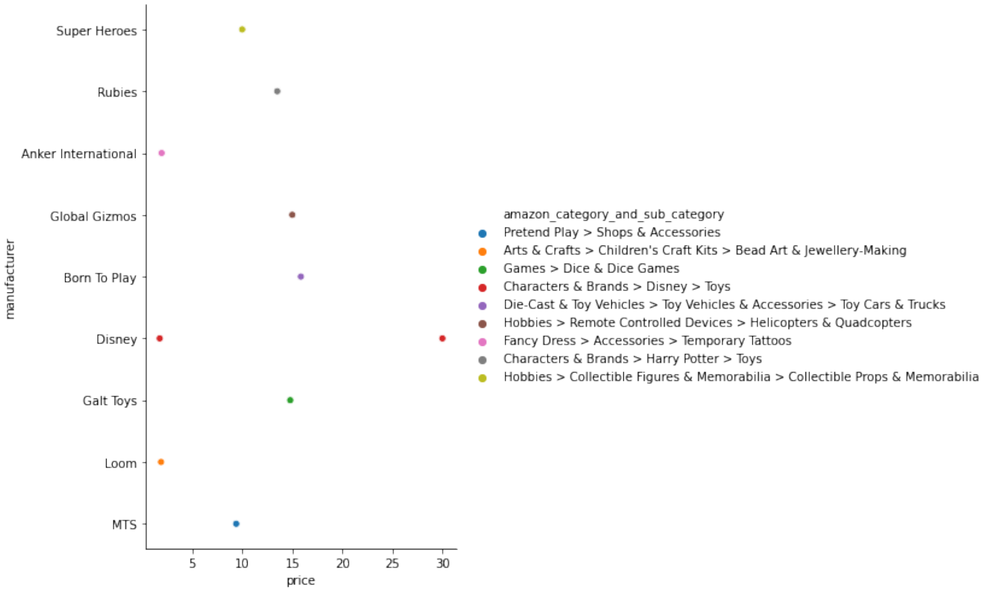


Fig. 5 Least Manufacturers

From the above graph we got to know that-

- Some manufacturer sales more than category of products
- Price difference can be seen b/w products of same category and manufacturer
- Categories like art & craft, hobbies are same in top rated and low rated reviews

c. Manufacturer v/s Rating-

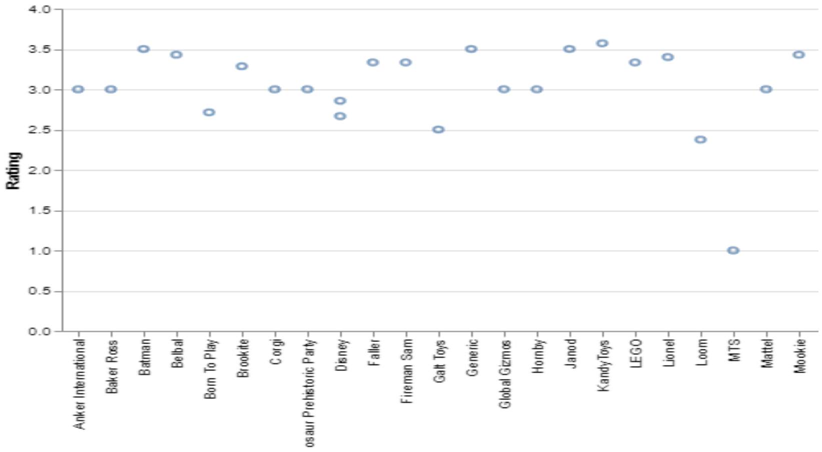


Fig. 6 Ratio for middle manufacturer

- This graph shows (24 manufacturer) the review with rating below 3.5, from 20,000 reviews only these much are negative or low rated. So, the dataset is positive biased.
- Here we got that most of the reviews are rated above 3.5.
- To solve that issue we take a fair ratio of reviews for sentiment and category prediction.

d. Price v/s Rating-

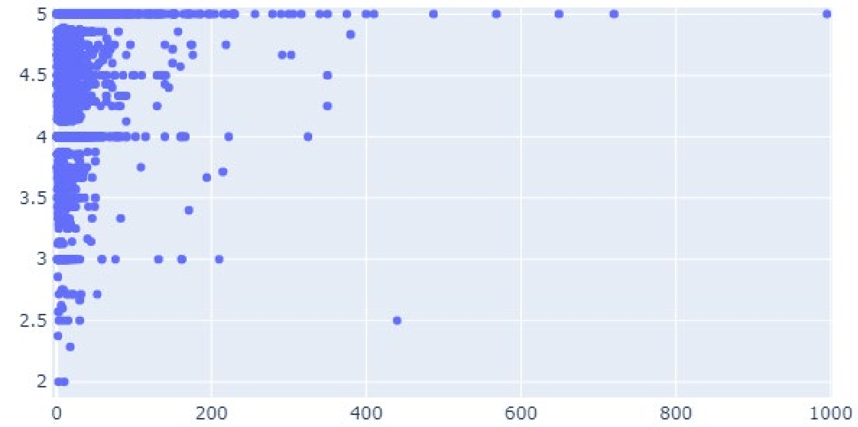


Fig. 7 Price v/s Rating Ratio

- From Fig. 7, It is clearly observed that people buy cheap products with variable rating between 2.5-4.5.
- Most selling products are having price below 100 euro.
- The product with higher price is rated as 5, that means the quality is Good.

e. Most frequent words (Bag of Words)-

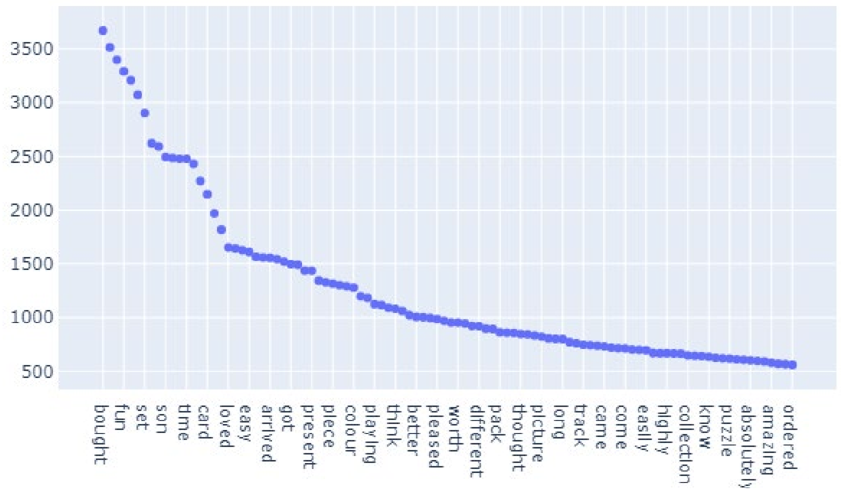


Fig. 8 Most Frequent Words

- Fig. 8 is plotted for top 100 Most occurring words.
- The word "bought", "fun" are seen most frequent among all the reviews.
- It is clear from the word "son" that most of the products are bought by parents for their kids.

f. Manufacturer v/s category based on rating-

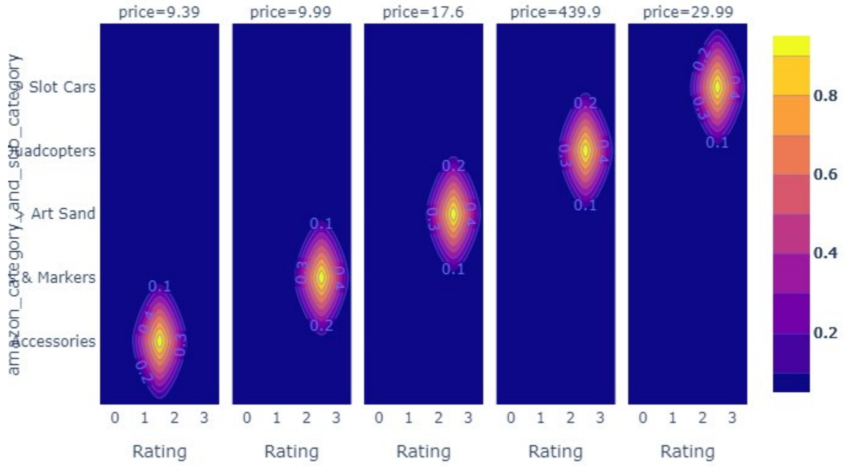


Fig. 9 Density Graph

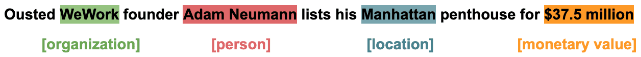
- This density graph shows that the category with higher price and count has higher rating.
- “Cars” and “quadcopters” have highest rating.

C. Entity Detection:

Entity extraction, also known as entity name extraction or named entity recognition, is an information extraction technique that refers to the process of identifying and classifying key elements from text into pre-defined categories.[8]

We used entity extraction to get the company names present in our reviews. There are various approaches and libraries available in python.

i. NLTK – using chunking we can find the entity.

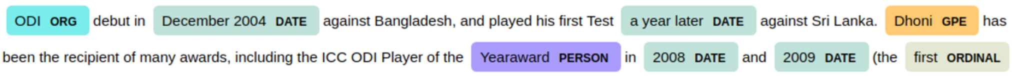


Ousted WeWork founder Adam Neumann lists his Manhattan penthouse for $37.5 million

[organization] [person] [location] [monetary value]

Fig. # NTLK Output[8]

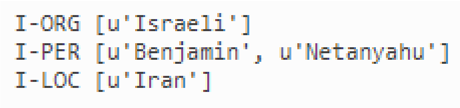
ii. Spacy – displacy will help to find the entity.



ODI ORG debut in December 2004 DATE against Bangladesh, and played his first Test a year later DATE against Sri Lanka. Dhoni GPE has been the recipient of many awards, including the ICC ODI Player of the Yearaward PERSON in 2008 DATE and 2009 DATE (the first ORDINAL

Fig. # Spacy Output [9]

iii. Polyglot –



```
I-ORG [u'Israeli']
I-PER [u'Benjamin', u'Netanyahu']
I-LOC [u'Iran']
```

Fig. # Polyglot Output

- The ORG., PER, LOC, Value are extracted from the given sentences
- We need only ORG. because this contains the Company or ORG name
- We used a hybrid approach, smartly combining all the result from these 3 libraries.

III. METHODOLOGY

After completing the pre-processing and finding trend by visualization we have some image clear in our mind that how to approach the classification. To classify a review into its sub-categories we took 3 approaches that had have mentioned above. Now we see each approach one by one.

1. Doc2Vec with logistic regression –

   Doc2Vec –

   the goal of doc2vec is to create a numeric representation of a document, regardless of its length.

   This algorithm is used to convert document or sentences into vectors and finding similarity between sentences or document. But using only Doc2Vec is a very bad decision.

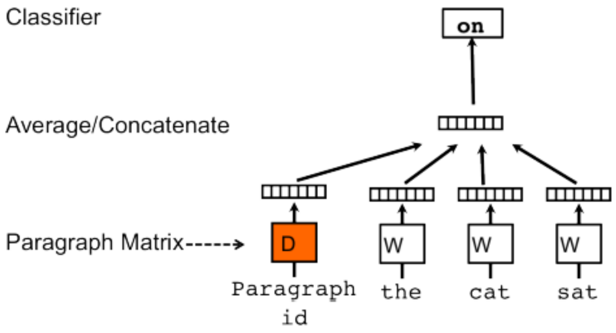
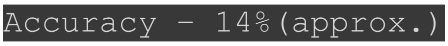


Fig. # Doc2Vec Example [10]

- Fig. # demonstrate that para. Id helps while predicting the perfect word to be fitted in the given set of tokenized words.

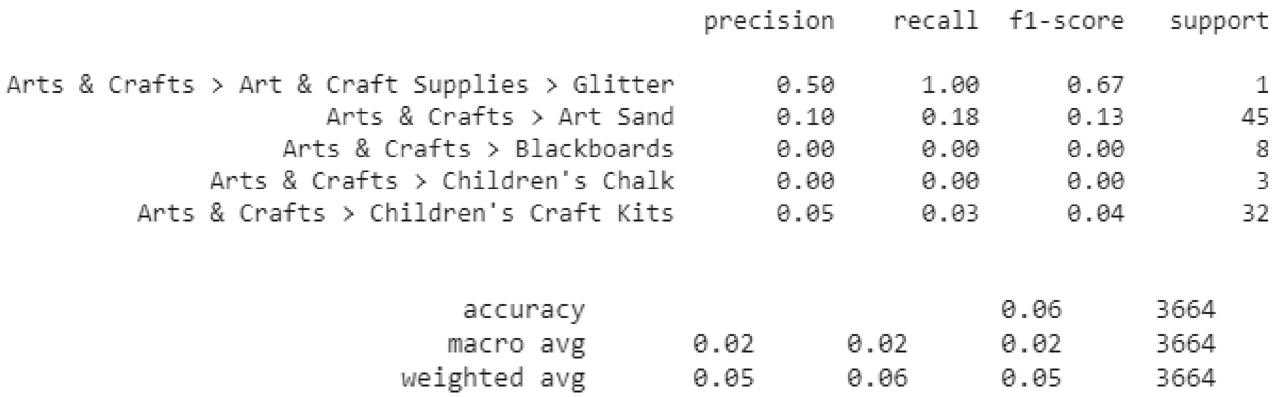


```
Accuracy - 14%(approx.)
```

So, we searched and found that by using Machine Learning algorithms the accuracy can be increased, logistic regression gives the best result-

Multinomial Logistic regression –

- Three or more categories without ordering.
- Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist.



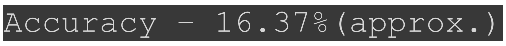
```
                                                precision    recall  f1-score   support

Arts & Crafts > Art & Craft Supplies > Glitter       0.50      1.00      0.67         1
                    Arts & Crafts > Art Sand       0.10      0.18      0.13        45
                 Arts & Crafts > Blackboards       0.00      0.00      0.00         8
            Arts & Crafts > Children's Chalk       0.00      0.00      0.00         3
        Arts & Crafts > Children's Craft Kits       0.05      0.03      0.04        32


                             accuracy                           0.06      3664
                            macro avg       0.02      0.02      0.02      3664
                         weighted avg       0.05      0.06      0.05      3664
```

Fig. # LR result

- From Fig. #, our LR model predicts only some of the category correctly.



```
Accuracy - 16.37%(approx.)
```

2. TF-IDF with Cosine Similarity –

After getting unsatisfactory result from Doc2Vec approach, we tried this

TF-IDF (Term Frequency - Inverse Document Frequent) – is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the TF-IDF weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query.

i. **TF: Term Frequency**, which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (aka. the total number of terms in the document) as a way of normalization:

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).

ii. **IDF: Inverse Document Frequency**, which measures how important a term is. we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

IDF(t) = $\log_e$ (Total number of documents / Number of documents with term t in it). [11]

Cosine Similarity –

Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space.[13]

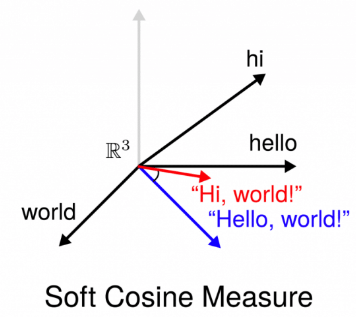


Fig. 10 [12]

- From Fig. 10, the angle between "hi" and "hello" is less than "world", but if we see "Hi, World" and "Hello, World" the angle is very acute, because "world" are same in both the sentence so, only the angle between "hi", "hello" matters, which we can see are same.

This is a very popular approach while solving these types of problems, but in our case, we have a huge number of categories (256) so, many of the reviews of different category seems similar. We can see the result with the help of this heat map-

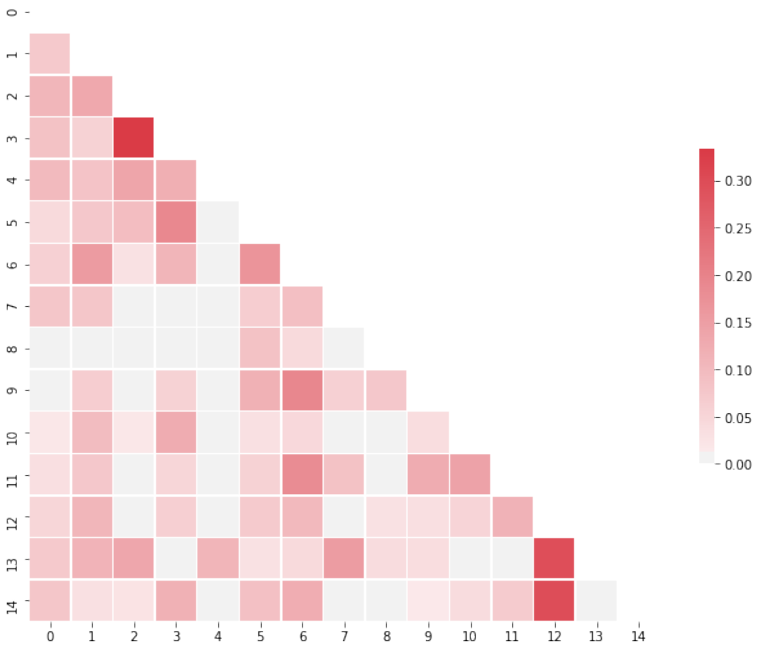


Fig. 11 Heat-Map

Here we can see that some of the reviews are corelated to each other, but the highest similarity score is **30%**. So, we are not happy with this accuracy, therefore we chose to move on to different some other approach.

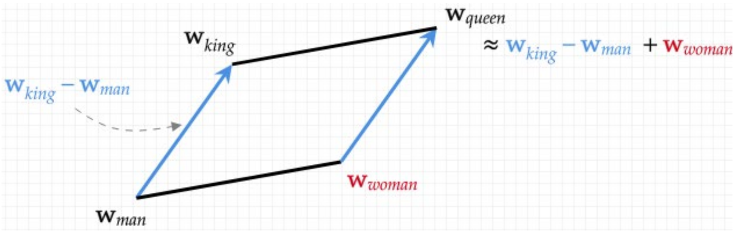
3. Word2Vec with personalized code –

These 2 approaches do not seem very help full. After doing some research, a technique called Word2Vec is found and is very popular nowadays in industries.

Word2Vec – Word2vec is a two-layer neural net that processes text by "vectorizing" words. Its input is a text corpus and its output are a set of vectors: feature vectors that represent words in that corpus. While Word2vec is not a deep neural network, it turns text into a numerical form that deep neural networks can understand.[15]

Word2vec is an algorithm, present in Gensim library of python, is basically an algo. which converts word into vector and links similar type of words in a document. There is a pre-trained Word2Vec model by Google of about 3 million words in 300 dimensions. We trained our data on top this model in-order to get more accurate result.

```
king - man + woman = queen
```



$\mathbf{w}_{queen}$
$\approx \mathbf{w}_{king} - \mathbf{w}_{man} + \mathbf{w}_{woman}$
$\mathbf{w}_{king}$
$\mathbf{w}_{king} - \mathbf{w}_{man}$
$\mathbf{w}_{woman}$
$\mathbf{w}_{man}$

Fig. # Example Word2vec [13]

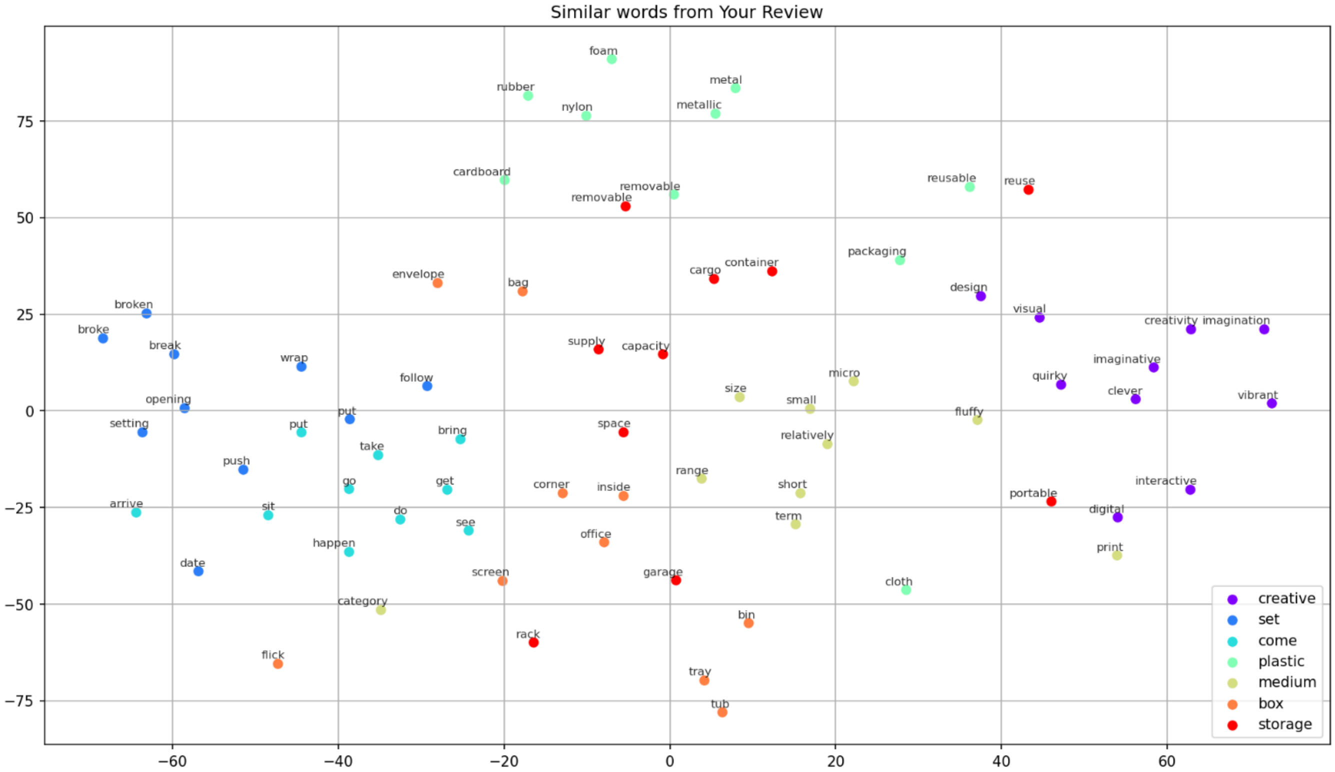


Fig. 12 TSNE graph

After the model training, we just saved the model, and move to write our own personalized classification code for given category and review. To do so we first extracted the company from the given review and matches the top words with their similar words in that review with the reviews of the categories related to the company and return the most relevant category for that review.

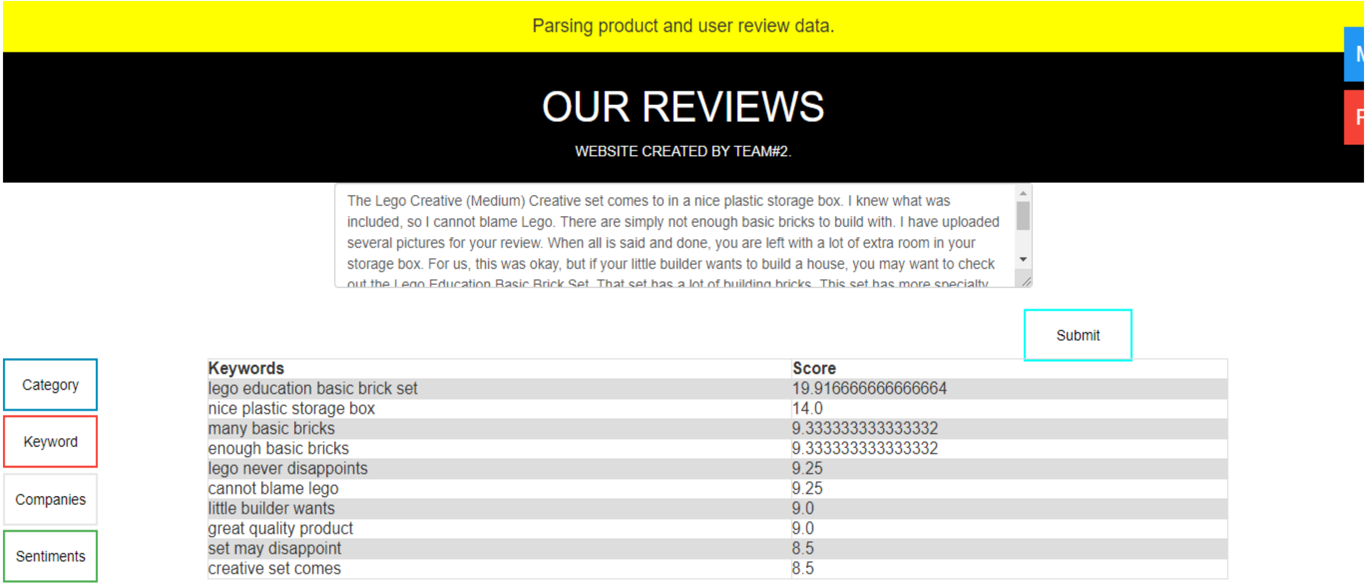
If the company was not given, then we categorize similar categories into a list and then we compare the top words of given review with the category reviews and returns most relevant category.

IV. User Interface

For a better experience and understanding of our model, we decided to create a webpage using Flask (python library). It contains-

i. Text Area

ii. Buttons – each button for different task Category prediction, Company prediction, Keywords, Sentiment.



OUR REVIEWS

WEBSITE CREATED BY TEAM#2.

The Lego Creative (Medium) Creative set comes to in a nice plastic storage box. I knew what was included, so I cannot blame Lego. There are simply not enough basic bricks to build with. I have uploaded several pictures for your review. When all is said and done, you are left with a lot of extra room in your storage box. For us, this was okay, but if your little builder wants to build a house, you may want to check out the Lego Education Basic Brick Set. That set has a lot of building bricks. This set has more specialty

Keywords	Score
lego education basic brick set	19.916666666666664
nice plastic storage box	14.0
many basic bricks	9.333333333333332
enough basic bricks	9.333333333333332
lego never disappoints	9.25
cannot blame lego	9.25
little builder wants	9.0
great quality product	9.0
set may disappoint	8.5
creative set comes	8.5

Fig. 13 Webpage

- Here we can see the beautiful webpage with heading of research, and button that gives relevant result.
- Also, after clicking the button we get a table showing the results with their relevant result

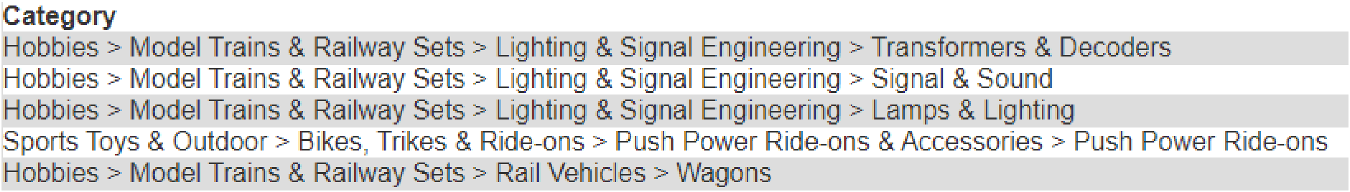
V. Result

The purpose of this research is to provide customer pre-sorted reviews with correct extracted information. If we believe human psychology researches, then information given in points or in form of table is clearer than a paragraph. This is the main reason of classification of reviews and classification on the category is a tough job but once it is done it helps many of the customers to choose the product easily.

The result of our research is as follows-

Word2Vec performs best for classification job and gives very satisfying result in comparison with TD-IDF & Doc2Vec.

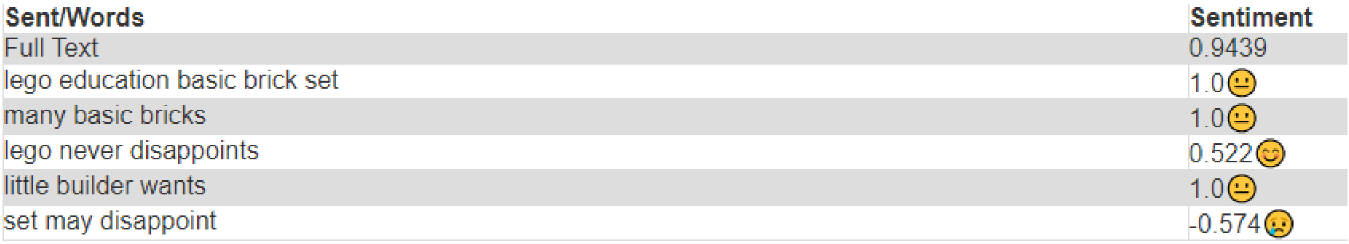
Our model summarizes the review into these tables of category, sentiment, keyword, and companies.



Category
Hobbies > Model Trains & Railway Sets > Lighting & Signal Engineering > Transformers & Decoders
Hobbies > Model Trains & Railway Sets > Lighting & Signal Engineering > Signal & Sound
Hobbies > Model Trains & Railway Sets > Lighting & Signal Engineering > Lamps & Lighting
Sports Toys & Outdoor > Bikes, Trikes & Ride-ons > Push Power Ride-ons & Accessories > Push Power Ride-ons
Hobbies > Model Trains & Railway Sets > Rail Vehicles > Wagons

Fig. 14 Relevant Categories

- From the fig. we can see the predicted category by our word2vec model, these categories are arranged in the decreasing order of relevancy.



Sent/Words	Sentiment
Full Text	0.9439
lego education basic brick set	1.0😐
many basic bricks	1.0😐
lego never disappoints	0.522😊
little builder wants	1.0😐
set may disappoint	-0.574😞

Fig. 15 Sentiment Result

- Sentiment of full text entered and top 5 keywords
- These are the sentiment result using VADER, also for better understanding we added emojis.

VI. CONCLUSION

1. We have taken an amazon dataset of toys containing reviews and other relevant information. After that we extracted only important features from the dataset followed by some feature engineering like –
   a. Tokenization
   b. Punctuation Removal
   c. Stop Words Removal
   d. Emoji Removal
   e. Lemmatization
   f. HTML tags Removal

   After all this done, we have a sorted data-frame containing only important and to the point information.

2. Now we must check if there is some outlier or relationship between columns by data visualization so, the data selection part for training the model is easier. To do so we have used some smart graphs, these smart graphs can be plotted by using some intelligent library of python for visualization like –
   a. Plotly
   b. Altair
   c. Seaborn
   d. Matplotlib

   We got that manufacturer, category, price, and rating having some relation in between.

3. After all that, now comes the main part of extraction the useful information firstly we extracted entity from the given reviews. To do so, we used a hybrid approach like we merged the result coming from all the 3 techniques (NLTK, Spacy, Polyglot).

4. Here comes the main part, category prediction, we have used 3 techniques to predict category –
   a. Doc2Vec with Logistic regression – this gives best result for Doc2vec using some Machine Learning Algorithm.

      Only Doc2Vec will give you accuracy of 14 %
      With logistic regression accuracy 16.37%

   b. TF-IDF with Cosine Similarity – TD-IDF converts word into vectors with its score of occurrences.
      Cosine Similarity – measures the angle between 2 words or sentences and gives back the values. Value between 0.9 – 1.0 is best for any word or sentence.

      Accuracy 30%

   c. Word2Vec with personalized code – Word2Vec is most used technique nowadays in NLP, this technique learns the similar words present in the dataset to a particular word and we can use them for understanding new sentences. We train our model on the top of Google pre-trained model on about 3 million words with 300 dimensions.

      We wrote our own code for the prediction of categories for given manufacturers or ungiven manufacturers.

      Accuracy 85%

VII. REFERENCES

1. From Data camp Image link
2. NLTK library we used - https://www.nltk.org/
3. Spacy - https://spacy.io/
4. Matplotlib - https://matplotlib.org/
5. Seaborn - https://seaborn.pydata.org/
6. Altair - https://altair-viz.github.io/
7. Plotly - https://plotly.com/python/
8. How to Do Named Entity Recognition Python Tutorial - https://monkeylearn.com/blog/named-entity-recognition-python/
9. Image - https://www.google.com/url?sa=i&url=https%3A%2F%2Ftowardsdatascience.com%2Fnamed-entity-recognition-with-nltk-and-spacy-8c4a7d88e7da&psig=AOvVaw234EV8MGKJ__7Q0r4z6058&ust=1595109359265000&source=images&cd=vfe&ved=0CAkQjhxqFwoTCMCWncmj1eoCFQAAAAAdAAAAABAD
10. Image - https://www.google.com/url?sa=i&url=https%3A%2F%2Fmc.ai%2Funderstanding-document-embeddings-of-doc2vec%2F&psig=AOvVaw2edd0T2KXriFNC-O4YU3gY&ust=1595109411797000&source=images&cd=vfe&ved=0CAkQjhxqFwoTCOi9huKj1eoCFQAAAAAdAAAAABAD
11. What does tf-idf mean - http://www.tfidf.com/
12. Image - https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.machinelearningplus.com%2Fnlp%2Fcosine-similarity%2F&psig=AOvVaw2aLjHj3jaHKOTBKCJhNPhg&ust=1595109481891000&source=images&cd=vfe&ved=0CAkQjhxqFwoTCLDQkoOk1eoCFQAAAAAdAAAAABAD
13. Image - https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.ed.ac.uk%2Finformatics%2Fnews-events%2Fstories%2F2019%2Fking-man-woman-queen-the-hidden-algebraic-struct&psig=AOvVaw0LSwnoX8uSLmRi7zQSe4A2&ust=1595109524320000&source=images&cd=vfe&ved=0CAkQjhxqFwoTCMCkz5ik1eoCFQAAAAAdAAAAABAD