BAX-452: Machine Learning

Final Project: Amazon’s Laptop Reviews Analysis using Sentiment Analysis and Topic Modeling

Team

(Sam) Venkata Aravind Sampath Bhagavatula

Sheetal Rajgure

Sneha Guravannavar

Instructor

Jörn Boehnke, Assistant Professor

Table of Contents

Executive Summary:

The objective of this project is to analyze the online product reviews of laptops on Amazon and understand customer's sentiments towards them; we achieved this using word embedding, classification and deep learning techniques. We also aimed to identify the characteristics of each review i.e. what specific aspect each review is talking about, and whether it is being praised (positive sentiment) or criticized(negative sentiment); we achieved this by using a topic modeling technique - LDA (Latent Dirichlet Allocation). We believe that a combination of these techniques would provide actionable insights to the firms. For instance, identifying that 56% reviews talking about 'battery life' are negative would allow the laptop player to understand that their model needs to improve in 'battery life'. Similarly, negative sentiments towards 'delivery service' or 'delivery charges' would help Amazon realize that they need to focus on providing better services and offers.

We explored Logistic Regression and Recurrent Neural Networks to carry sentiment analysis. We achieved 78.2% accuracy through the RNN model and were able to identify 7 characteristics that customers are talking about in these reviews through topic modeling. We also analyzed which characteristics are dominantly good and which are dominantly bad by mapping both technique's results.

Background, Context and Domain Knowledge

The objective is to analyze laptop product reviews on Amazon using different ML techniques. Businesses often want to understand what people are talking about their products and how they can improve to achieve better customer satisfaction, ultimately translating to more business (sales, revenue, transactions, etc.). There are multiple ways firms do this - surveys, feedback, focus groups, review analysis, etc. Text analysis is a huge field and various NLP and ML techniques like clustering, classification, topic modeling, deep learning etc. are employed in the industry to solve business problems like these. Today, every firm has an online presence; Social media has made it convenient for people to express anything. Thus, customers today have various platforms to express their feelings and feedback on products. Therefore, there is a huge amount of text data available, which cannot be parsed manually - we need advanced textual analysis techniques to get actionable insights. Amazon and its competitors as well every other firm who has been talked about online uses text analysis techniques to extract what people are talking about their products. In this project we use a combination of word embeddings, sentiment analysis and topic modeling to achieve our objective, details are explained in subsequent sections.

Data Assessment:

Dataset was sourced from Kaggle, it included 171K Amazon reviews of different product categories. For our scope, we filtered only laptop product reviews which consisted of ~6900 reviews with product name, ratings, price, review (one line review), summary (detailed review) and sentiment (positive, negative or neutral). We majorly used the columns ‘summary’ and ‘sentiment’. Both datasets (one only with reviews and one with all columns) is attached in the zip folder. Snapshot of data in appendix (*3*)

For Data cleaning, we decided to drop the price column since it was not required; we kept other columns for further analysis.

For Data Preprocessing in (a) creating word2vec embeddings, we converted the text lowercase tokens, ignoring tokens that are too long (max length 25 characters) using Gensim python library's preprocessing function. Similarly, in (b) topic modeling, we converted text to lowercase, applied lemmatization to the words so that the root words of all derived words are used, removed stop-words and words with lengths less than 2. This is to remove words that don't add value so as to make computation efficient. We used 'nltk' libraries like WordNetLemmatizer, word_tokenize, stopwords for pre-processing.

Analysis - Methodology

At the outset, we created custom word embeddings using Amazon reviews data and combined it with pre-trained embeddings. A review embedding is the average embedding of all words contained in it. We carried sentiment analysis. We then carried topic modeling to understand the major topic (i.e.cluster) of each review. Stepwise approach is explained as follows:

I. Word Embeddings (Word2Vec)

1. We started with training the word2vec model based on Amazon reviews corpus, let's call it 'custom embeddings'. Using the Gensim libraries available in Python, we trained a word2vec model with vector_size=100 (each word's embedding would be 100 in size i.e. the vector of a word would have 100 dimensions) ,window=10 (while context training the model would consider a surrounding 10 words - 5 ahead 5 behind) ,min_count=2 (words below 2 letters are not considered in the corpus, this is to make the corpus relevant by removing words that don't add value), sg =1 (the method is skipgram), epochs=5 (iterations it takes to train the model, we balanced it to avoid overfitting). After this step, we can check the similarity between two words, the vector form of any word in the corpus, find similar words, etc. As examples, we explained a few in the code as well as in appendix (*4*).
2. Secondly, we downloaded pre-trained word embeddings, let's call it 'pretrained embeddings', the model is already trained on google news corpus and the vector size is 300 (*5*)

3. We know that a review is made up of multiple words and we created embedding and pre-trained embedding for each word. Now we need to find embedding for a review (a vector that captures the essence of the review taking into account all word embeddings in it). For this we take the average of all vectors of the words present in the review. The output is again a vector with 100 dimensions for custom embedding and a vector with 300 dimensions for pretrained embedding. (*6*)
4. Our idea is to combine the two embeddings (concatenation) so that we are able to enhance the pretrained embeddings with more context related to laptop reviews. For example, the word 'black' in pretrained models might have different contexts related to color, race, clothes, etc. However, in our reviews, black refers to the blackness of the laptop screen. Hence combining both would give specific context over and above the pretrained model.

Note: We used custom embeddings with a vector size of 100 for further steps. However the same steps could be extended using the combined vector of size 400, we kept this as future analysis owing to current scope.

II. Sentiment Analysis

1. *Logistic Regression*

   A. We first explored logistic regression to do sentiment analysis using scikit learn libraries in Python. Considering it a classification problem, where a review needs to be categorized into positive, negative or neutral sentiment.

   B. The dataset was split into train and test datasets (80:20). The logistic regression model was trained on the train dataset while the performance was measured using the test dataset since we are interested in calculating the out of sample performance.

   C. Input for logistic regression was the review vector from custom embeddings that we created earlier, while output was probability of belonging to a class. Based on max prob, the inputs were classified as positive or negative or neutral

D. The accuracy of logistic regression was 84% with 16% of false predictions. Confusion matrix (*7*)

2. Recurrent Neural Networks

A. Since this is a text analysis problem, we decided to explore NLP and deep learning technique to achieve the same objective. We built Recurrent Neural Networks using tensorflow (Keras) in Python

B. The structure of the RNN we used is: 1 input layer, 2 hidden layers (1 embedding layer, 1 LSTM - long short-term memory networks) and an output layer with a sigmoid activation function. (*8*)

C. A text vectorizer from Python's tensorflow library was used to vectorize the words - this is to create indexes and vocabulary. For RNN, an important step is to create an embedding matrix that goes as an input to the model. Embedding matrix is nothing but a matrix that represents each word from the vocabulary, with its embeddings along its index. Index comes from vectorizer and embeddings from what we created earlier (custom embeddings). This matrix has words in each review with their corresponding custom word2vec embedding mapped as per index. That's how a neural network identifies which word belongs to which review

D. After defining the RNN structure, we built the model using keras.Model() and fit it with the train data with 2 Epochs.

E. The performance was measured on test data, to get out of sample performance - 78.2% accuracy

III. Topic Modeling

LDA (Latent Dirichlet Allocation): We wanted to extend our sentiment analysis to extract features / characteristics from the reviews to identify laptop characteristics that are classified as good or bad

A. This method uses a clustering algorithm to cluster similar words together. Based on these clusters we can identify a topic/characteristic that each cluster represents and map it to each review. We also have sentiments attached to each review. Using both these data points, we can find out positive sentiment characteristics and negative sentiment characteristics.

B. The first step we did in LDA was preprocessing explained in the Data assessment section. This was to make our corpus efficient by removing redundant words or words that are not adding value

C. Secondly, we use count vectorizer to give an index to each word & review and frequency. Sample output in appendix (*1*)

D.Built an LDA model with number of clusters 7 (we experimented with different numbers of clusters and selected the one that gives us clear and meaningful topics). Input to this model is the vocabulary of vectorizers and output is clusters

E. Now each review has multiple words, which may belong to different clusters. Here we need to find out the corresponding cluster for each review where the maximum of its words belong. The output of LDA gives us a score corresponding to each cluster. The score represents how many words of a review belong to that cluster. Based on which cluster has the maximum score, we assign that cluster to the review. For instance, maximum score for review no 1 is for cluster 7 hence, review 1 belongs to cluster 7 and so on. (example in appendix *9*) Next we add these scores and dominant topic (cluster) to our dataframe

E. In the next step, we extract words and their relevance score. Relevance score represents how strongly a word belongs to the cluster that it is classified into. We arrange the words in each cluster (i.e. topic) in descending order of their relevance score. By looking at the words we identify the topic.(*10*)

Analysis - Results:

The accuracy of logistic regression was 84% with 16% of false predictions. Confusion matrix (*7*)

The performance was measured on test data, to get out of sample performance - 78.2% accuracy

From topic modeling analysis, the resulting topics that we got are-

Topic 1: Battery life

Topic 2: Display

Topic 3: Value for money

Topic 4: Overall product

Topic 5: Service

Topic 6: Wifi

Topic 7: Quality & Performance

The distribution of sentiments for each topic is given in appendix (*2*). We see that, overall dataset has more positive reviews. Customers are talking more about overall quality and performance -- this characteristic got the highest number of reviews followed by display and battery life. Value for money aspect got the least negative reviews. To extend this project, we can plot the graph product wise and identify similar patterns in each product or brand. We can come up with insights like Lenovo has the best battery life, Acer has the best display, etc.

Summary:

For firms looking to analyze reviews, Sentiment analysis alone does not give a deeper picture. It tells about the overall distribution of sentiments. However, what is the context or topic of the review is not known. Hence, we implemented a combination of sentiment analysis techniques and topic modeling. We were able to achieve topic extraction from reviews, word2vec embeddings and sentiment analysis using logistic regression and RNN. Neural networks are better than Logistic regression techniques for text analysis and we would recommend the firm to use deep learning techniques to predict the sentiments. We would also recommend the firm to focus on negative sentiment topics to improve.

Conclusion & Future Scope for Analysis:

Through NLP techniques, we were able to create word embeddings using word2vec, identify the sentiment and extract topics from reviews. This has a huge scope of extension as it is a very deep topic. We can combine the pretrained and custom average review embeddings before sentiment analysis. We can dig deeper into product and brand level to analyze the sentiments and topics. This will give the firm a better picture such as Asus is best known for its gaming features while Lenovo is known for its battery life, or a particular model of HP has low quality display and a particular model of Dell has bad sound quality, etc. We can also dive into 'aspect based sentiment analysis'. A review can talk about multiple things. Example, "Display is good but battery life is bad". The review talks about good sentiment towards display but bad sentiment towards battery life characteristics. We have done some literature review on how to identify multiple sentiments and topics inside one review and it is mentioned in references (*R7* and *R8*). We believe it is a logical step forward for this project.

References:

In addition to class slides, codes and other materials, we referred the below material to aid our project:

1. Gensim Word2Vec Tutorial - Kavita Ganesan

   (https://kavita-ganesan.com/gensim-word2vec-tutorial-starter-code/#.ZByQ5OzMJJV)

2. Topic modeling and sentiment analysis to pinpoint the perfect doctor - Nuo Wang

   (https://blog.insightdatascience.com/topic-modeling-and-sentiment-analysis-to-pinpoint-the-perfect-doctor-6a8fdd4a3904)

3. Sentiment Analysis: Aspect-Based Opinion Mining - An investigation into sentiment analysis and topic modeling techniques - Lowri Williams

   (https://towardsdatascience.com/%EF%B8%8F-sentiment-analysis-aspect-based-opinion-mining-72a75e8c8a6d)

4. Using pre-trained word embeddings in a Keras model - Francois Chollet

   (https://blog.keras.io/using-pre-trained-word-embeddings-in-a-keras-model.html)

5. How to use Natural Language Processing to analyze product reviews? - Gunvant Saini

   (https://towardsdatascience.com/how-to-use-natural-language-processing-to-analyze-product-reviews-17992742393c)

6. The Illustrated Word2vec - Jay Alammar

   (http://jalammar.github.io/illustrated-word2vec/)

7. The power of aspect based sentiment analysis - Kaushik Jagini

   (https://medium.com/swlh/the-power-of-aspect-based-sentiment-analysis-18c3908ac53d)

8. Deep Learning for Aspect-Based Sentiment Analysis - Bo Wang & Min Lu

   (https://cs224d.stanford.edu/reports/WangBo.pdf)

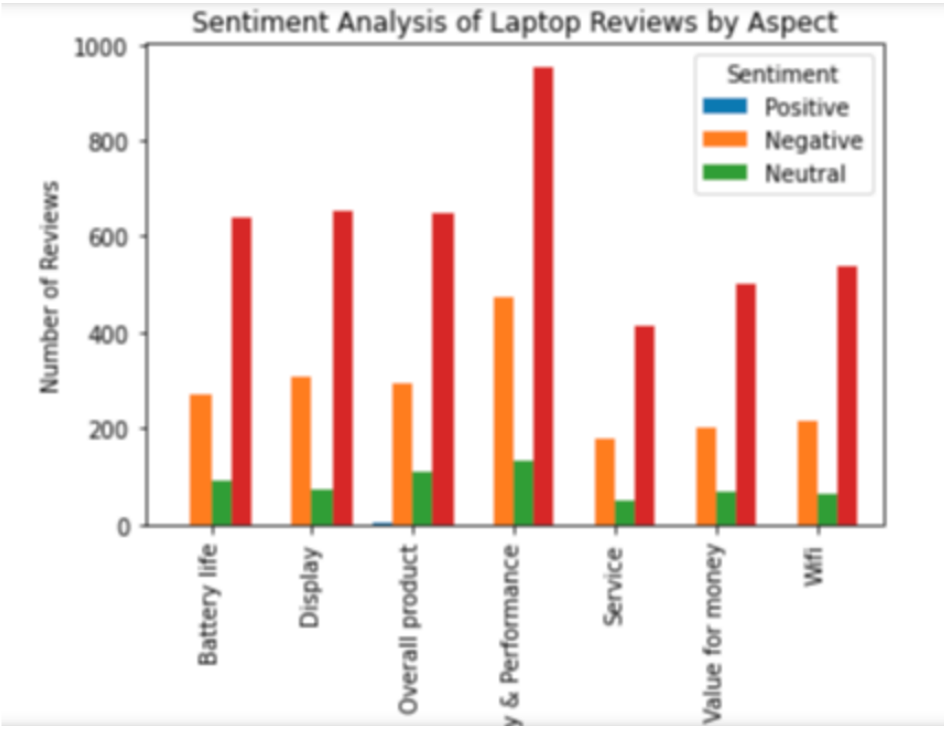
Appendix:

1. Topic Modeling vectorizer output

1st parameter is review index, second is word index / key and third is word frequency in that particular review

```
(0, 22070)    1
(0, 7812)     1
(0, 13671)    1
(0, 11687)    3
(0, 2701)     1
(0, 2448)     1
(0, 13026)    1
(0, 20673)    1
(0, 7012)     1
(0, 8585)     2
(0, 3417)     1
(0, 849)      1
(0, 14539)    1
(0, 19875)    1
(0, 8758)     1
(0, 446)      1
(0, 19509)    1
(0, 17519)    2
(0, 18510)    1
(0, 21608)    1
(0, 10738)    1
(0, 12914)    1
(0, 21592)    1
(0, 22074)    1
(0, 8042)     1
:      :
(6867, 8449)  1
(6867, 331)   1
```

2. Topic Modeling Review Analysis



3.Snapshot of dataset

3.1 in CSV

ProductName	ProductPrice	Rate	Review	Summary	Sentiment
ASUS TUF Gaming A17 with 90Whr Battery Ryzen 5 Hexa	63990	1	Absolute rubbish!	store doesnt open then when i reinstall the windows according to the microsoft customer service loli all function started malfunctioning again like auto changing of gaming modes now doesnt change and when the screen goes of after sometime of non usage it doesnt open when we press any key rather instantly goes to hibernating mode for no reason and i ha	negative
ASUS TUF Gaming A17 with 90Whr Battery Ryzen 5 Hexa	63990	2	Bad quality	the turbo fan is not working at all	negative
ASUS TUF Gaming A17 with 90Whr Battery Ryzen 5 Hexa	63990	5	Classy product	it is very good product but sometimes there is a lag in opening tabs in chrome	negative
ASUS TUF Gaming A17 with 90Whr Battery Ryzen 5 Hexa	63990	4	Wonderful	nicebut no dvd writer so always download the game	negative
ASUS TUF Gaming A17 with 90Whr Battery Ryzen 5 Hexa	63990	1	Horrible	not that much good as i expected	negative
ASUS TUF Gaming A17 with 90Whr Battery Ryzen 5 Hexa	63990	2	Not good	its very tipcial laptop dont give ms office with laptop	negative
ASUS TUF Gaming A17 with 90Whr Battery Ryzen 5 Hexa	63990	2	Not good	cooling fan not working performance is not satisfied	negative
ASUS TUF Gaming A17 with 90Whr Battery Ryzen 5 Hexa	63990	2	Slightly disappointed	extremely poor display color looks artificial and pictures arent sharp pixelated background while watching anime videos i should not have bought this machine first hand	negative
ASUS TUF Gaming A17 with 90Whr Battery Ryzen 5 Hexa	63990	1	Absolute rubbish!	very bad experience passwords went wrong cant sign in epic game so many errors in this lappy worst dont buy	negative
ASUS TUF Gaming A17 with 90Whr Battery Ryzen 5 Hexa	63990	2	Bad quality	i purchased the laptop and with few days the laptop started going blank and i had to perform hard power cycle now the laptop is getting stuck in diagnosing the pc and not booting	negative
ASUS TUF Gaming A17 with 90Whr Battery Ryzen 5 Hexa	63990	1	Absolute rubbish!	very bad experience passwords went wrong cant sign in epic game so many errors in this lappy worst dont buy	negative
ASUS TUF Gaming A17 with 90Whr Battery Ryzen 5 Hexa	63990	3	Good	its ok but not good experience	negative
ASUS TUF Gaming A17 with 90Whr Battery Ryzen 5 Hexa	63990	5	Best in the market!	go for it no doubt	negative
MSI GF63 Thin Core i5 11th Gen - (8 GB/512 GB SSD/Winc	51990	1	Absolute rubbish!	battery problem	negative
Lenovo IdeaPad Gaming Core i5 11th Gen - (8 GB/512 GB	59990	4	Pretty good	everything was good but the sound bit lower	negative
Lenovo IdeaPad Gaming Core i5 11th Gen - (8 GB/512 GB	59990	2	Bad quality	value for money 44k battery backup very poor performance only 1 2 hours only purchasing date 5102022	negative
Lenovo IdeaPad Gaming Core i5 11th Gen - (8 GB/512 GB	59990	5	Terrific	everyone saying that its worth of money but battery problem but in my point of view its electronic device no one can guarantee 100 lets see	negative

3.2 Dataset after data cleaning:

	ProductName	Rate	Review	Summary	Sentiment
6864	ASUS TUF Gaming A17 with 90Whr Battery Ryzen 5...	5	Best in the market!	go for it no doubt	negative
6865	MSI GF63 Thin Core i5 11th Gen - (8 GB/512 GB ...	1	Absolute rubbish!	battery problem	negative
6866	Lenovo IdeaPad Gaming Core i5 11th Gen - (8 GB...	4	Pretty good	everything was good but the sound bit lower	negative
6867	Lenovo IdeaPad Gaming Core i5 11th Gen - (8 GB...	2	Bad quality	value for money 44k battery backup very poor p...	negative
6868	Lenovo IdeaPad Gaming Core i5 11th Gen - (8 GB...	5	Terrific	everyone saying that its worth of money but ba...	negative

3.3 Dataset after review cleaning for topic modeling

	ProductName	Rate	Review	Summary	Sentiment	cleaned_review
0	ASUS TUF Gaming F15 Core i5 10th Gen - (8 GB/5...	1	Horrible	value for money laptop battery backup is low b...	negative	value for money laptop battery backup low that...
1	ASUS TUF Gaming F15 Core i5 10th Gen - (8 GB/5...	1	Worthless	best gaming lappy in low price range with good...	negative	best gaming lappy low price range good graphic
2	ASUS TUF Gaming F15 Core i5 10th Gen - (8 GB/5...	1	Utterly Disappointed	best laptop at this price range pros new ryzen...	negative	best laptop price range pro new ryzen processo...
3	ASUS TUF Gaming F15 Core i5 10th Gen - (8 GB/5...	1	Useless product	delivered without defect and arrived on timebe...	negative	delivered without defect arrived timebest valu...
4	ASUS TUF Gaming F15 Core i5 10th Gen - (8 GB/5...	1	Worthless	this laptop is very nice and good looking perf...	negative	laptop nice good looking performance also good

4. Example of custom word embeddings:

first is similar words, second vector of a word, third similarity

```
In [20]: model.wv.similar_by_word('laptop')

Out[20]: [('lap', 0.8038808107376099),
          ('marvelous', 0.7990869879722595),
          ('flipkartfor', 0.7963902354240417),
          ('prize', 0.7884438037872314),
          ('opinion', 0.7876623868942261),
          ('professional', 0.784482479095459),
          ('goodotherwise', 0.7821691632270813),
          ('performan', 0.7817632555961609),
          ('category', 0.7807818651199341),
          ('itin', 0.7802942395210266)]

In [21]: model.wv.get_vector('display')

Out[21]: array([-0.2125267 , -0.05820439,  0.07371209,  0.27839366,  0.14030728,
               -0.17760277,  0.36471093,  0.4940261 , -0.62416875, -0.18948382,
                0.00794867, -0.24955384,  0.02434595,  0.04067629, -0.0570067 ,
               -0.07151498,  0.05093838, -0.03627001,  0.13717198, -0.7875858 ,
                0.33736297,  0.09448305,  0.1782909 ,  0.01284182, -0.29513294,
               -0.50803906, -0.30869094,  0.36041927, -0.05698458,  0.10190786,
               -0.10267562, -0.06550863,  0.47044995,  0.07716373, -0.2565548 ,
                0.19869976,  0.19983014, -0.23264438, -0.20578265, -0.44895393,
               -0.16036768, -0.13021971, -0.11155153,  0.23314124,  0.07599704,
               -0.33071506,  0.27342582, -0.03238205,  0.35274437, -0.03329845,
               -0.10408803, -0.04607852,  0.07802533,  0.17774816, -0.08299491,
                0.06688396,  0.21433742, -0.28505772,  0.04111367,  0.18173581,
                0.02245242, -0.06139115,  0.08661313,  0.18410699, -0.48372492,
                0.2909853 ,  0.10571846,  0.35606828, -0.5062379 ,  0.1998058 ,
               -0.16355458,  0.02866764, -0.20821112,  0.19169214,  0.01569088,
                0.03121443, -0.05324258,  0.19343399, -0.33407295,  0.06857178,
                0.01802356,  0.35737562, -0.10482359,  0.5292971 , -0.25526297,
               -0.0684476 ,  0.26947117,  0.22609442,  0.15917955,  0.10167849,
                0.35251635, -0.28593954, -0.27280316,  0.1065686 , -0.0670064 ,
```

In [24]:
```
model.wv.similarity("battery","bad")
```
Out[24]: 0.31217515

In [25]:
```
model.wv.similarity("battery","good") #The word battery has more similarity with the word good than bad.We can
#come up with multiple other insigths through this model identifying which word is closer to positive sentiment
#and which to negative
```
Out[25]: 0.49771312

5.Pretrained vector

```
# Load pre-trained vectors
pretrained_wv = api.load('word2vec-google-news-300')
```

6. Mean Vectors

In [31]:
```
#Mean vector of reviews with custom word2vec
import numpy as np

reviews_vector = []
for r in laptop_reader:
    vectors = [model.wv[token] for token in r if token in model.wv]
    if vectors:
        e = np.mean(vectors, axis=0)
        reviews_vector.append(e)
```
```
2023-03-23 12:40:58,300 : INFO : reading file Reviews.csv...this may take a while
2023-03-23 12:40:58,312 : INFO : read 0 reviews
```

In [32]:
```
#Mean vector of reviews with pretrained word2vec
pretrained_reviews_vector = []
for r in laptop_reader:
    vectors = [pretrained_wv[token] for token in r if token in pretrained_wv]
    if vectors:
        e = np.mean(vectors, axis=0)
        pretrained_reviews_vector.append(e)
    else:
        e = np.zeros(vectors)
        pretrained_reviews_vector.append(e)
```
```
2023-03-23 12:41:00,395 : INFO : reading file Reviews.csv...this may take a while
2023-03-23 12:41:00,399 : INFO : read 0 reviews
```

In [ ]:
```
# Combining the mean vectors -- in a vector of 100 + 300 = 400 size
```

7. Confusion matrix of Logistic Regression

	Predicted negative	Predicted neutral	Predicted positive
Actual negative	311	5	72
Actual neutral	39	26	64
Actual positive	34	4	819

8. Structure of RNN

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, None)]            0

 embedding (Embedding)       (None, None, 100)         487200

 bidirectional (Bidirectiona  (None, 128)              84480
 l)

 dense (Dense)               (None, 1)                 129

=================================================================
Total params: 571,809
Trainable params: 84,609
Non-trainable params: 487,200
_________________________________________________________________
```

9. Dominant topic for reviews

	ProductName	Rate	Review	Summary	Sentiment	cleaned_review	Topic1	Topic2	Topic3	Topic4	Topic5	Topic6	Topic7	Dominant_topic
0	ASUS TUF Gaming F15 Core i5 10th Gen - (8 GB/5...	1	Horrible	poor delivery service denied to follow open bo...	negative	poor delivery service denied follow open box p...	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.9881	7
1	ASUS TUF Gaming F15 Core i5 10th Gen - (8 GB/5...	1	Worthless	worst product ever seen	negative	worst product ever seen	0.0179	0.0179	0.0179	0.8927	0.0179	0.0179	0.0179	4
2	ASUS TUF Gaming F15 Core i5 10th Gen - (8 GB/5...	1	Utterly Disappointed	display not working yet	negative	display working yet	0.0238	0.0239	0.0238	0.8569	0.0239	0.0239	0.0239	4
3	ASUS TUF Gaming F15 Core i5 10th Gen - (8 GB/5...	1	Useless product	got it for just 41500 but all excitment went i...	negative	got for just 41500 excitment went vain receive...	0.0042	0.0042	0.0042	0.0042	0.9748	0.0042	0.0042	5
4	ASUS TUF Gaming F15 Core i5 10th Gen - (8 GB/5...	1	Worthless	i brought this laptop for 51490 offer price wa...	negative	brought laptop for 51490 offer price 53990 le ...	0.0014	0.0014	0.9916	0.0014	0.0014	0.0014	0.0014	3

10. Topics as per words in clusters

Topic 1 is related to battery , topic 3 is related to value for money

	topic	relevance_score
battery	Topic1	392.234363
best	Topic1	341.519614
backup	Topic1	277.177650
battery backup	Topic1	252.261289
price	Topic1	231.280304
bad	Topic1	135.788081
range	Topic1	124.640751
best laptop	Topic1	123.724788
price range	Topic1	111.433114
great	Topic1	100.302710

	topic	relevance_score
money	Topic3	426.560058
for money	Topic3	403.199667
value	Topic3	392.225394
value for	Topic3	370.601124
beast	Topic3	77.710530
ram	Topic3	60.250201
big	Topic3	54.148151
come	Topic3	42.855214
best value	Topic3	41.108122
big big	Topic3	39.142857