Fully Automated system to customers' sarcastic opinions recognition from amazon with sale impact and credibility

A Real Time Streaming using the Deep Learning Models/Big Data Vision

PROBLEM STATEMENT

Better consumer than better productivity and better the economy. The rapid development of the Internet, the capacity of users to create contents has created vigorous online communities that deliver a mammon of product information. However, the high volume of reviews that are usually makes it difficult for manufacturer and seller to examine the quality of product by people opinion over products sales.

Currently, product various factors impact the product's sales. We advance further to analyze the sarcasm reviews which would be ironic or sarcastic that impact the sales with sentiment, readability, and product price, discount, reviewCount and listprice. Here, readability means total strength count of characters in reviews, which would impact the product's sales Rank.

The current research impact was encounter by regression model without being considering deep learning model and deep feedforward networks. Further, we extend the existing work and analyze the sarcasm and sentiment about the products with all factors with the augment of deep learning model using tensor flow.

Hypothesis: Do sarcasm in reviews impact the sales with all other products' characteristics more than sentiments?

Hypothesis: Do sentiment or credibility alone impact the products' characteristics?

PROJECT SCOPE

The deep learning model has huge contribution in big data, following the similar trends we will analyze the sentiments, sarcasm of products' reviews using RNN a deep learning model, mxnet deep learning model of feedforward and regression analysis using LM model.

The prediction of sales rank over these product characteristics and reviews characteristics will help to understand the consumer behavior in detail to improve the productivity and product marketing.

Particularly we will analyze the sarcastic reviews separately and in comparison, with sentiment of reviews of the three various categories but here we have chosen IKEA store particularly HOME AND FURNITURE category.

Product outcomes:

1- Distributor, store and Consumer can see his behavior impact of products' sales.
2- The product will allow the researcher to seek a way to see better impact of people opinions in clear vision about product's sales.

INTRODUCTION

Our work of analysis of reviews match the part of speech positive list and negative list to produce sentiment score (SO) and sarcasm to prove the importance of peoples' experience with the goods. The products in this research are chosen from the "Home and furniture" category of Ikea store based on United Kingdom (UK) amazon web site. Henceforth, we will see the better impact of reviews in the vision of customers' opinion over sales rank. In addition to the reviews, other factors of the product are necessarily influential on the sales rank of a product and vice versa.

Important contribution of this project, we investigate the veracity of this theory and quantify the extent to which textual content of each review effects on product sales from Amazon. Prior work has extensively analyzed and classified sentiments in online opinions (Liu, 2005), (Pang, 2008) and explored how *automatic* procedures can be used for obtaining conjoint attributes and levels through the use of natural language processing, statistical clustering methods, they have not examined their economic impact with deep learning model with sarcastic reviews. For example, it has shown that the volume and valence of online product reviews influences product sales such as books and movies (Dellarocas, 2005) (Chevalier, 2006) but this research had no evidence of textual content in these reviews while estimating impact on sales. The author (Ghose, 2007), proposed a log regression model by considering subjectivity and objectivity of the reviews. To the best of our knowledge no prior work on sentiment analysis and sarcasm together of the products' reviews impacts over sales.

SYSTEM DESCRIPTION

SYSTEM REQUIREMENT ANAYSIS

1. Atomic: the requirement is complete by collecting the reviews by external tool.
   Requirement further categorized into sarcastic and sentiment reviews with product's characteristics.
2. Complete: the system will collect requirement into following attribution.

Type	**Variable & Desciption**
Product AND SALES Data	**ASIN :** The product ID in amazon used for product identification, it is non-numberic. e.g. "B00SLN8NFC" **Title:** The product name is non numeric identification for product name.

	Category: It is sub category of product under main category which is give at search in Amazon. e.g. Cooking & Dinning. There are further sub categories of sub categories which is we weed out in clustering analysis.For example, “Tea & Espresso”, “Milk Frothers”,”Handheld Milk Frothers”. **Rating:** rating is the attribute which give value from 1 -5. It is criteria at amazon for product sales ranking. E.g. 4,4.5 and 5 **List price:** It is price of nominated by stores. E.g. 130 **Price:** It is same if list price factor out no discount but different based on discount. E.g. 100 **Discount:** it is percentation of dicount but grab here with actual discounted money on the product. E.g. 30 **SalesRank:** The rank determine the proroity given by customer by most purchased. E.g 23202, 3271 and 1 **ReviewCount:** Total reviews that commented by customers on product.e.g. 100,2 and 1
Reviews **Characteristics**	**Review1 to Reviw8:** It is pure text grab as customers’ comments on the product. E.g. “Overpriced, short lived crazy Halogen Imports.” “This is lovely, very nice to look at and very easy to assemble. Great price. Get short bulbs so they don't stick out of top. Great price and quick delivery “
Review Readibility	**Logcharctercount:** it is length (characters&word&sentences): average length of the reviews 1-8 in all sentences.
Review polarity	**Averagesentscore:** this attribute represent average score of all reviews polarity. High Pos if >+2

	High Neg if<-2

3. Consistent and unambiguous: consumer and store can seek the experience of report in the form of research report with complete tabular results into sarcastic and non-sarcastic categories.
4. Traceable: the requirement is traceable and clearly map in the form of results summary.
5. Validation: the results will be validated by the help of 5-fold test.

ECONOMICAL, SOCIAL AND TECHNICAL CONSTRAINTS

The system will able to contribute in the domain of the economics in the form of better consumer ship. The research project must be published and contribute the results in open community, therefore, keeping this constraint that system outcome is report or research paper, we plan a paper to published right after this project report. We will share the short form of research in slide share.

System will not able to be online or cannot be available in android store, which is biggest constraint or limitation logistically.

Technically on a single machine it will not be possible to run tensor flow with CPU. The process millions of products required cloud platform to run over GPU (the graphic card) based deep learning model. Therefore, we prototype our research to limited deep learning model usage over CPU.

FUNCTIONAL REQUIREMENT

COLLECTING THE REVIEWS AND PRODUCT DETAILS FROM AMAZON.

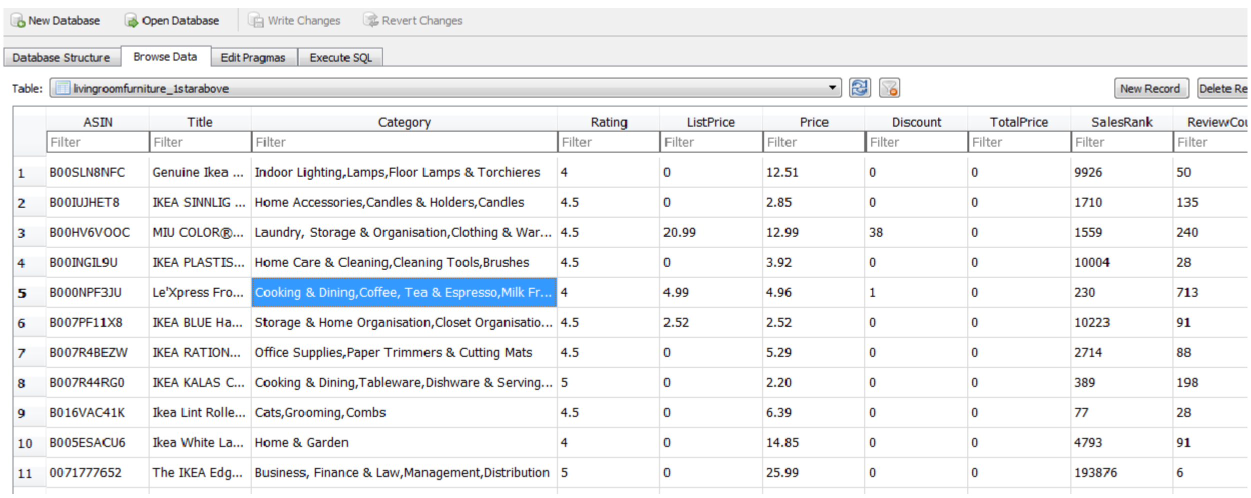
The data is in the form of series of web site attributes from “amazon.co.uk”, these attributes are embedded in web page, and it is almost difficult to grab these product details from web page as show below without help of scrapping tools. To get the required product first we search the product by “Home and Kitchen” category. Then, data filter out by the customers’ review at the rating 4 and above for each of the search category.

SCRAPING:

The data fetch by providing the URL into the scrapping tool **"ZonhasinHunter",** which is a license product. The data fetch all the relevant details and produce all data into complete comma separated file (CSV). Before, grabbing the data, we filtered out data further by special criteria: the data must contain review count greater than 1 and rating must be equal and above then 1 star. Therefore, we concluded that scrapping tools are easy to use , gives less error , user friendly and more reliable in scrapping.

STORAGE AND SQLITE

After scraping process, the work of data pruning initiated. Further, we take comma separated file into SQLite database in the table format. The table contains 25 attributes, which contain top 8 qualitative reviews. Moreover, the irrelevant attributes weed out from the table.



	ASIN	Title	Category	Rating	ListPrice	Price	Discount	TotalPrice	SalesRank	ReviewCou
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	B00SLN8NFC	Genuine Ikea ...	Indoor Lighting,Lamps,Floor Lamps & Torchieres	4	0	12.51	0	0	9926	50
2	B00IUJHET8	IKEA SINNLIG ...	Home Accessories,Candles & Holders,Candles	4.5	0	2.85	0	0	1710	135
3	B00HV6VOOC	MIU COLOR®...	Laundry, Storage & Organisation,Clothing & War...	4.5	20.99	12.99	38	0	1559	240
4	B00INGIL9U	IKEA PLASTIS...	Home Care & Cleaning,Cleaning Tools,Brushes	4.5	0	3.92	0	0	10004	28
5	B000NPF3JU	Le'Xpress Fro...	Cooking & Dining,Coffee, Tea & Espresso,Milk Fr...	4	4.99	4.96	1	0	230	713
6	B007PF11X8	IKEA BLUE Ha...	Storage & Home Organisation,Closet Organisatio...	4.5	2.52	2.52	0	0	10223	91
7	B007R4BEZW	IKEA RATION...	Office Supplies,Paper Trimmers & Cutting Mats	4.5	0	5.29	0	0	2714	88
8	B007R44RG0	IKEA KALAS C...	Cooking & Dining,Tableware,Dishware & Serving...	5	0	2.20	0	0	389	198
9	B016VAC41K	Ikea Lint Rolle...	Cats,Grooming,Combs	4.5	0	6.39	0	0	77	28
10	B005ESACU6	Ikea White La...	Home & Garden	4	0	14.85	0	0	4793	91
11	0071777652	The IKEA Edg...	Business, Finance & Law,Management,Distribution	5	0	25.99	0	0	193876	6

READING AND UNSTRUCTURING

We read comma separated file (CSV) in **R-Studio** into frame and after opting 18 most relevant attributes for analysis, indeed, data categorized as qualitative reviews and quantitative both. The top most 8 customer reviews taken along with other attributes which description is given section.

	ASIN	Title	Category	Rating	ListPrice	Price	Discount	TotalPrice	SalesRank
1	B00SLN8NFC	Genuine Ikea HOLMO Floor Lamp Soft Smooth Relaxi…	Indoor Lighting,Lamps,Floor Lamps & Torchieres	4.0	0.00	12.51	0	0	9926
2	B00IUJHET8	IKEA SINNLIG Scented tealight, Vanilla Pleasure candl…	Home Accessories,Candles & Holders,Candles	4.5	0.00	2.85	0	0	1710
3	B00HV6VOOC	MIU COLORÂ® Drawer Dividers Closet Organizers Br…	Laundry, Storage & Organisation,Clothing & Wardrob…	4.5	20.99	12.99	38	0	1559
4	B00INGIL9U	IKEA PLASTIS - Dish-washing brush, assorted colours	Home Care & Cleaning,Cleaning Tools,Brushes	4.5	0.00	3.92	0	0	10004
5	B000NPF3JU	Le'Xpress Frother	Cooking & Dining,Coffee, Tea & Espresso,Milk Frothe…	[illegible]	4.99	4.96	1	0	230
6	B007PF11X8	IKEA BLUE Hanging Storage With 6 Compartments, for …	Storage & Home Organisation,Closet Organisation Sy…	[illegible]	2.52	2.52	0	0	10223
7	B007R4BEZW	IKEA RATIONELL VARIERA Transparent Drawer mat - Dr…	Office Supplies,Paper Trimmers & Cutting Mats	4.5	0.00	5.29	0	0	2714
8	B007R44RG0	IKEA KALAS CHILDRENS PLATES X 6 NEW	Cooking & Dining,Tableware,Dishware & Serving Pie…	5.0	0.00	2.20	0	0	389
9	B016VAC41K	Ikea Lint Roller+4 Sticky replacement Heads Easily a…	Cats,Grooming,Combs	4.5	0.00	6.39	0	0	77
10	B005ESACU6	Ikea White Lack Side Table K-Deals	Home & Garden	4.0	0.00	14.85	0	0	4793
11	0071777652	The IKEA Edge: Building Global Growth and Social Go…	Business, Finance & Law,Management,Distribution	5.0	0.00	25.99	0	0	193876
12	B007PR8WJW	IKEA Floor Uplighter Light Lamp (1)	Indoor Lighting,Lamps,Floor Lamps & Torchieres	4.5	0.00	13.95	0	0	1924
13	B001R5JXB4	Premier Housewares Four Tier Slatted Wooden Shoe …	Laundry, Storage & Organisation,Clothing & Wardrob…	3.0	24.99	13.75	45	0	1522
14	B00JTERUCY	Ikea Antilop Highchair Cushion & Cover - Reversible …	Bedding & Linens,Bedding,Sheets & Pillowcases,Dec…	4.0	0.00	10.50	0	0	1950
15	B005SSJB68	FRAKTA BLUE LARGE SHOPPING, LAUNDRY BAG SET OF 3	Cooking & Dining,Kitchen Storage & Organisation,Sh…	4.5	0.00	2.49	0	0	2118
16	B007PJORBC	IKEA KALAS CHILDRENS MUGS X 6 NEW	Cooking & Dining,Tableware,Dishware & Serving Pie…	5.0	0.00	2.20	0	0	258
17	B00IBO39DM	IKEA SINNLIG Scented tealight, Crisp Apple Green ca…	Home Accessories,Candles & Holders,Candles	4.5	0.00	2.55	0	0	3271
18	B004PYY350	Premier Houseware 509519 Five Hook Over Door Ha…	Hardware,Hooks,Over Door Hooks	4.5	6.99	4.64	34	0	1565
19	B00938MQR0	Sleek black LED work / desk lamp - fantastic light	Power, Garden & Hand Tools,Hand Tools,Torches,Fla…	4.5	0.00	16.99	0	0	20610
20	B007R4XQT4	IKEA Toilet Brush with Holder WHITE Buy 1 Get 1 FREE	Bathroom,Bathroom Accessories,Toilet Accessories,…	4.5	0.00	2.99	0	0	151

Home Care & Cleaning,Cleaning Tools,Brushes

Before, doing any further work, we derive the attribute semantic oirentation from reviews using semantic funtion in R. The function takes text as input and intergate the positive and negative words for sementic score. Before, computing the semantic score , the pre-processing of text peform cleaning, steming and removal of list of strop word from the text. Then, a dictionary match list of words with the text.

MODELING: DEEP LEARNING / REGRESSION

Finally data prepare for predictive analysis using regression LM, deep learning keras RNN and deep mexnet feed forward network. We comparison of results presented in the form of tabular format and few plots.

FINAL SPECIFICATIONS

PROGRAMMING ENVIRONMENTS

1. RStudio
2. R language
3- SQL Lite

PREDICTIVE ANALYTICS

For Deep Learning:
1. Python in R
2. TensorFlow
3. Keras
4. Anconda
5. MAXNET
6. LM Regresison analysis

SYSTEM ARCHITECTURE

The project is consisted of four steps. We start by collecting data from amazon about ikea products' reviews and product information. Then, we perform the data preparation and feature extraction based on negative score of sentiment to save data into two categories sarcastic and sentiment dataset. These features are used in the prediction process. Finally, we expose the result the result for the user.

The system architecture consists of data science process steps and methodology.

Step 1: Data Collection
Step 2: Data Preparation & Feature Building
Step 3: Data Modeling
Step 4: Data Presentation.

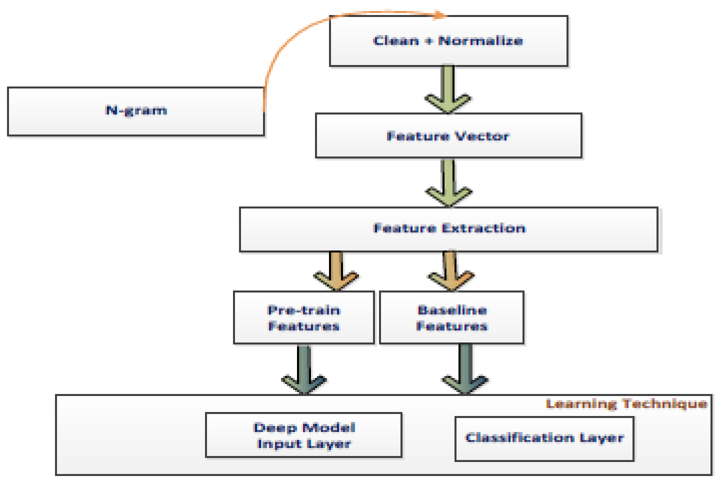


Fig: Methodology for analysis

USE CASE STUDY

The system will perform following functions and displays.

1. It will take read reviews and product information by streaming from tool **ZonhasinHunter (Chinese tool)**.

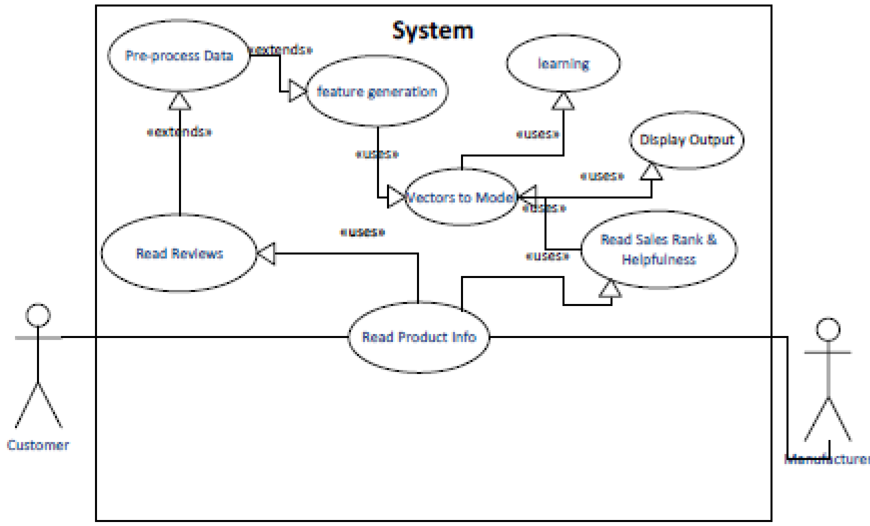
2. Take the input phrase automatically from site

3. Automatically preprocess the data review text, generate features of sarcastic and sentiment dataset with product sales rank, price, discount, pricelist, reviewcount, rating

4. Pass features like recognize sentiment or credibility impact, sarcasm impact and readability of reviews over sales

6. The prediction phase of the Model will output predicted scores until convergence.

8. The result of predictivity and credibility display on the dashboard automatically in the form of plots and graphs.

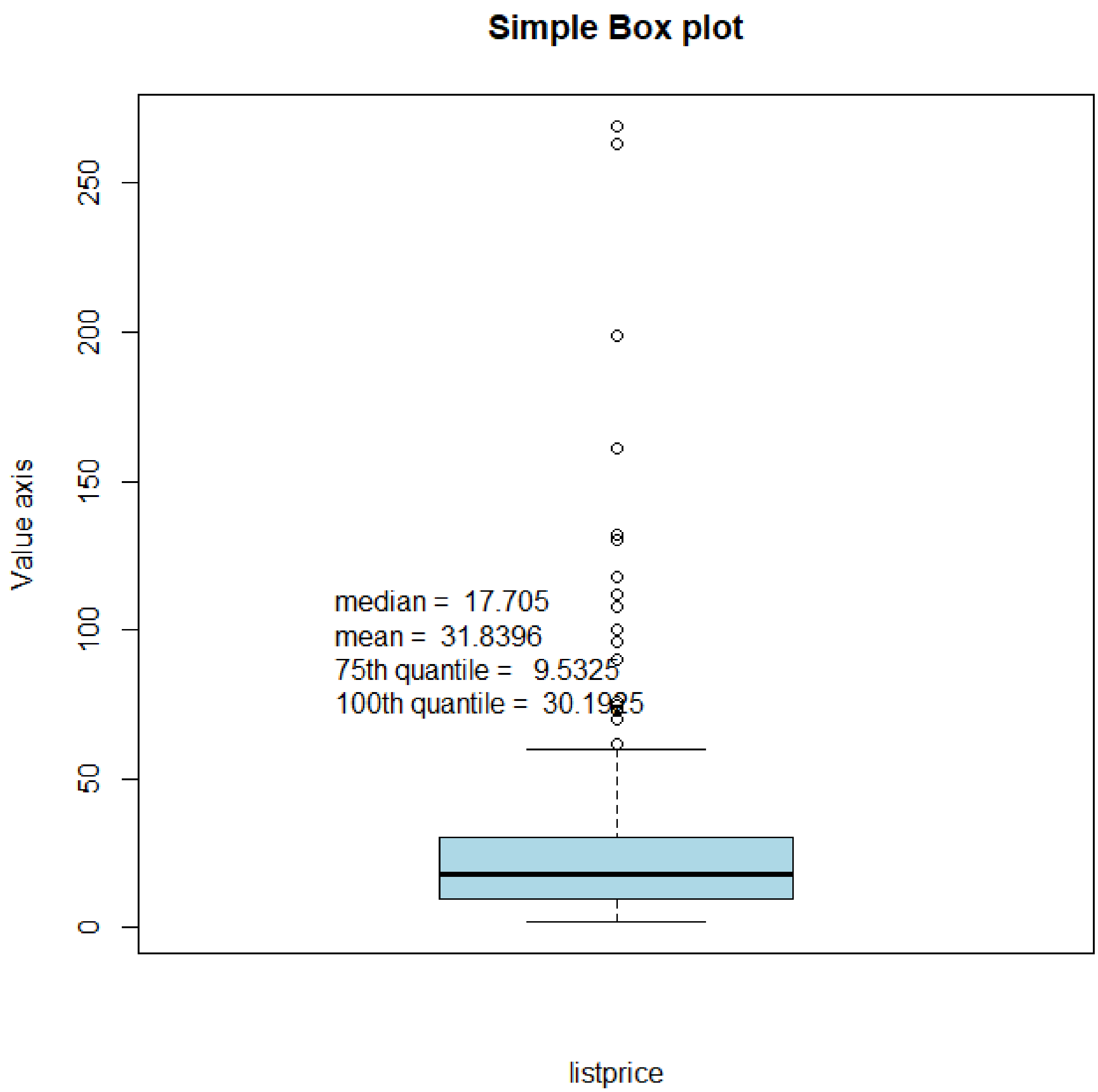


Use Case Diagram

SYSTEM DESIGN

IMPELEMENTATION DETAILS

Descriptive statistics goal is to summarize and show the data in meaningful way.



Not normalized and need to normalize to be prepare for model. We analyze all variables and normalized by log operation to normalize it.

ANALYSING THE SKEW AND OUTLIERS

As shown above fig 1, clearly 3rd quartile have low range values consistent with 1st quartile and 2nd quartile because, range fall fit to median value 25.99, whereas, clearly seen that 4th quartile spread have sign of skewness in box plot. Additionally, we observe the skewness direction by library "e1071", therefore, result shows towards right by the help of "**skewness (duration)"** function, which is high.

Table indicating all results from category "Home and Kitchen"

Attribute	Skewness
List price	2.57: it is high positive indicate more skewness from 3rd quartile to 4th quartile.
Review Count	4.03: it is same result as list price
Price	3.197655: It is same result as review count. That price indicates positive skewness from 3rd quartile to 4th quartile.
Sales Rank	2.474882: It is same result as review count. That price indicates positive skewness from 3rd quartile to 4th quartile.
Rating	-0.1075753: it indicates outliers are in 1st quartile.
Discount	0.07345338: it indicates no skewness in data

Table: Skewness analysis for outliers

Forthcoming, analysis is conducted in the hope that these outliers are not problems, but it is not desirable to lump together along dataset. To normalize the outliers a systematic procedure design is advocated. The elimination of outliers is not realistic because these are not errors but real value. Hence, log of all attributes computed to fit the values within boxplot.

NORMALIZE DATA

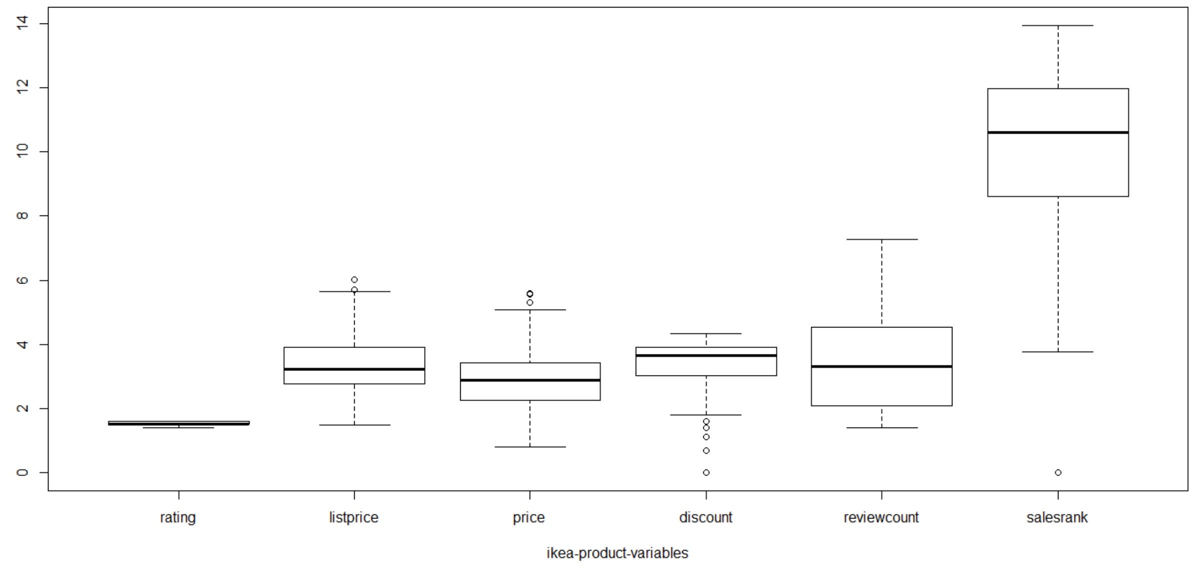


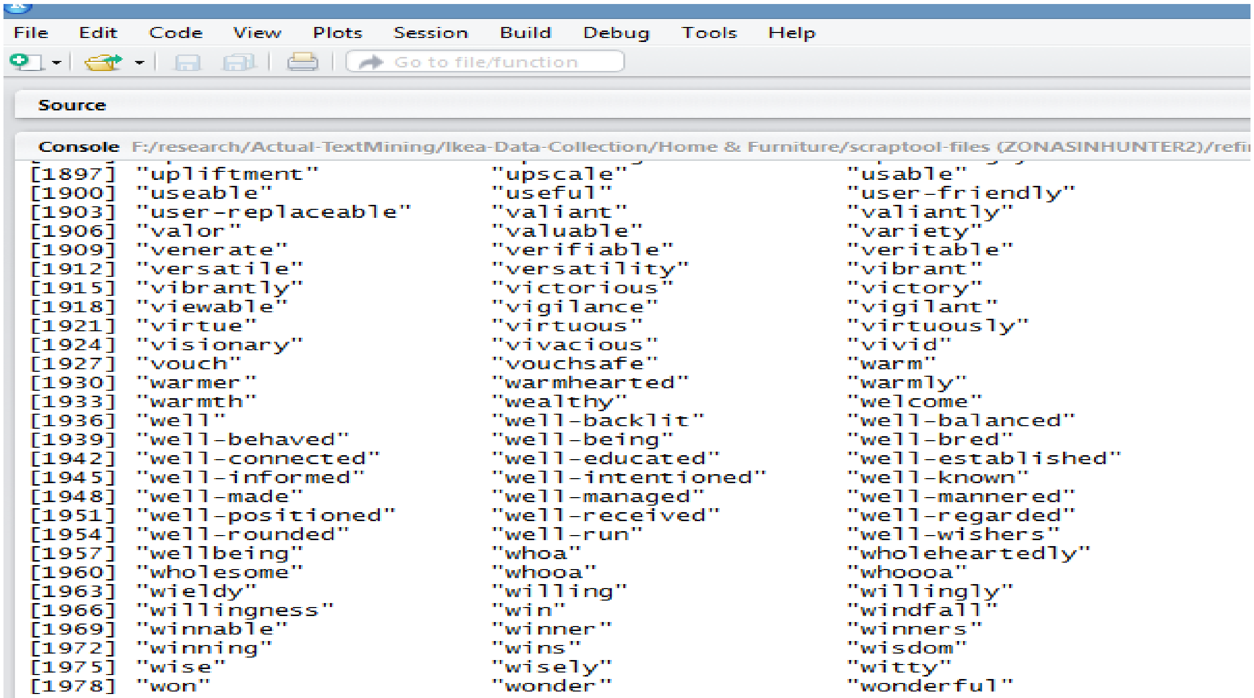
Fig: Boxplot of normalize of all attributes to normalize all the values under "Home and Kitchen"

It is clearly seen the values spread of all attribute outliers has been reduced to minimum. we offered the box plot mention fig.

SENTIMENT SCORE AND SARCASM SCORE

Sentimental analysis is computed in this research for deriving the variable polarity in the form of sentimental score (S0). sentimental analysis will performed by R due to simplicity of its usage in this research. Sentimental variable is having maximum value +2 which extremely positive and -2 which is extremely negative.

Below given sample of 2005 positive words dictonary which match the review positive verbs and count increase by 1. In case of negative match score -1 from total score.



```
File  Edit  Code  View  Plots  Session  Build  Debug  Tools  Help
Go to file/function
Source
Console F:/research/Actual-TextMining/Ikea-Data-Collection/Home & Furniture/scraptool-files (ZONASINHUNTER2)/refi
[1897] "upliftment"          "upscale"            "usable"
[1900] "useable"             "useful"             "user-friendly"
[1903] "user-replaceable"    "valiant"            "valiantly"
[1906] "valor"               "valuable"           "variety"
[1909] "venerate"            "verifiable"         "veritable"
[1912] "versatile"           "versatility"        "vibrant"
[1915] "vibrantly"           "victorious"         "victory"
[1918] "viewable"            "vigilance"          "vigilant"
[1921] "virtue"              "virtuous"           "virtuously"
[1924] "visionary"           "vivacious"          "vivid"
[1927] "vouch"               "vouchsafe"          "warm"
[1930] "warmer"              "warmhearted"        "warmly"
[1933] "warmth"              "wealthy"            "welcome"
[1936] "well"                "well-backlit"       "well-balanced"
[1939] "well-behaved"        "well-being"         "well-bred"
[1942] "well-connected"      "well-educated"      "well-established"
[1945] "well-informed"       "well-intentioned"   "well-known"
[1948] "well-made"           "well-managed"       "well-mannered"
[1951] "well-positioned"     "well-received"      "well-regarded"
[1954] "well-rounded"        "well-run"           "well-wishers"
[1957] "wellbeing"           "whoa"               "wholeheartedly"
[1960] "wholesome"           "whooa"              "whoooa"
[1963] "wieldy"              "willing"            "willingly"
[1966] "willingness"         "win"                "windfall"
[1969] "winnable"            "winner"             "winners"
[1972] "winning"             "wins"               "wisdom"
[1975] "wise"                "wisely"             "witty"
[1978] "won"                 "wonder"             "wonderful"
```

Fig: list of positive words of verb, adjective and noun for Semantic orientation (SO) score

Here, given sample of matched positive and negative words illustration.

> *"Perfect to use when embroidering with my sewing machine. It is so unobtrusive clamping to the table and able to direct exactly where the light is required, especially when threading one of the six needles. Good long lead on it, very light to move and use elsewhere if required a perfect adaptable light. Arrived very quickly and well packed well done Janso 5 stars."*

The underline 7 positive words and their score is 0.7. Polarity this text is having positive score of 0.7 and negative score of 0.3 after deduction, the overall sentiment score is 0.4, which indicate the positivity of text is high, as already indicated that the value more than 0.2 is high positive.

The SO method assigns sentimental score based on hand ranked between -5 and 5 automated dictionaries. Sentimental score (SO) is computed for each review of product as mentioned below.

	ikea_data.ASIN	scores.review2.score	scores.review3.score	X.scores.review4.score	scores.review5.score	scores.review6.score	scores.review7.score	scores.review8.score
1	B004MQSDFW	5	4	0	3	5	6	3
2	B007POZ1O4	2	7	3	0	1	8	5

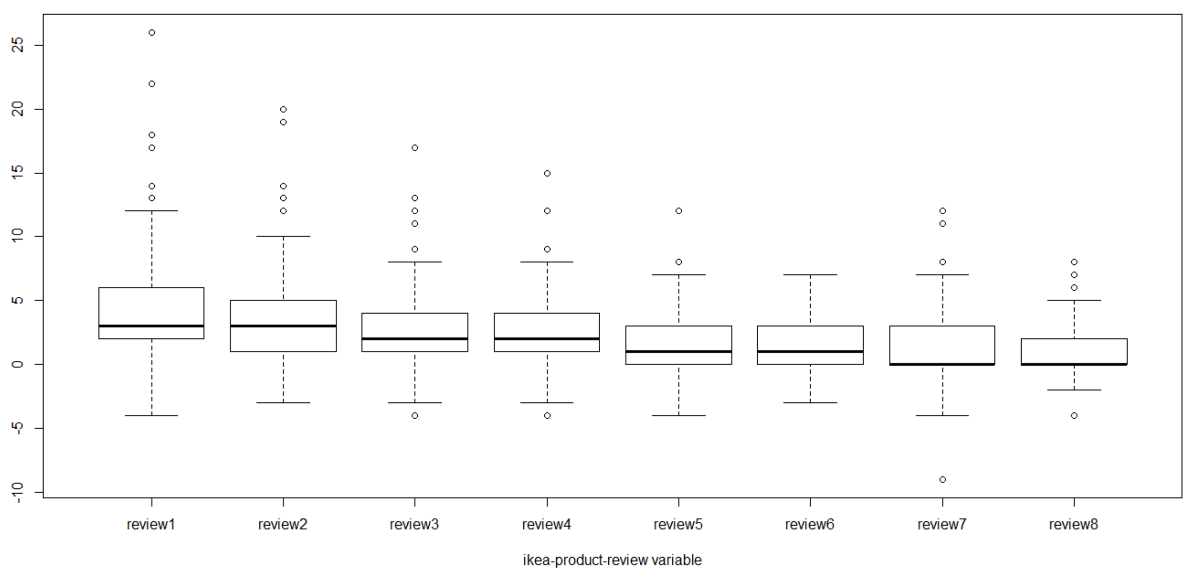
3	B00NX540P0	4	1	4	3	0	6	4
4	B000NPF3JU	1	7	1	3	3	7	3
5	B00HV6VOOC	11	8	5	4	1	5	2
6	B004PYY350	2	0	2	1	3	1	0
7	B00009Y353	11	2	1	15	3	3	6
8	B00CCMAJ8C	3	4	3	3	0	0	0
9	B002PHLZJ6	7	9	2	0	3	6	3
10	B000G1TDME	1	3	6	1	4	6	1
11	B00NIXAQ78	2	5	6	2	1	8	-1
12	B00EICIRU6	2	8	1	3	4	0	0

Figure 1: The above values showing 151 products sentimental values, the values varies between negative, high negative, positive and high positive

The value 8 is high value because is more than +2

-1 value indicate negative sentiment

The box plot of aforemention table represented for all reviews' sentiments of each product. Below given plot showing clearly seen the highest value is 20 which, indicate high positive value because any value greator than +2 is high positve. On the other hand, the value at vertical axis -10 indicate high negative value because any value degraded below -2 is high negative.



Hereafter computed the average sentiment score of all reviews of each product.

FINAL DATASET READY TO PASS IT TO MODEL (POSITIVE/NEGATIVE SENTIMENT)

ikea_dataset1

Filter

	logprice	logdiscount	logreviewcount	loglistprice	logavgcount	logsalesrank	scores.score	scores.review2.score	scores.review3.score	sco
1	1.5892352	3.7376696	6.602588	2.138889	6.100879	6.605298	2	7	3	
2	1.4816045	3.6109179	6.218600	1.944481	5.457988	7.091742	2	0	2	
3	2.9729753	3.1780538	5.442418	3.248435	6.140962	7.489971	11	2	1	
4	0.8458683	4.1108739	3.637586	1.791759	4.768564	7.323831	4	1	4	
5	2.5641798	3.6375862	5.459586	3.044046	5.354225	7.063048	11	8	5	
6	1.5993876	4.0943446	5.971262	2.512846	4.857873	4.595120	1	6	3	
7	2.2762411	3.9318256	4.521789	2.995232	5.596568	8.750208	9	6	6	
8	0.9439059	3.9702919	4.736198	1.704748	4.523146	8.206311	5	0	2	
9	3.8064402	4.1743873	3.828641	4.867534	5.810392	10.181347	4	0	5	
10	4.2483524	4.2766661	4.934474	5.521421	5.370638	7.930925	2	5	6	
11	1.9315214	4.1743873	4.779123	2.993229	6.082219	9.240287	6	1	1	
12	2.2396453	3.2188758	3.784190	2.524928	5.138002	5.598422	9	0	3	
13	1.9154509	4.3438054	6.109248	3.399529	5.616316	0.000000	1	-1	0	
14	4.5652853	3.5835189	5.579730	5.003946	6.176647	9.844003	2	3	1	

SARCASM ANALYSIS AND SCORE

To analyze the sarcasm sentences and reviews, we analyze the sarcastic reviews by following steps.

1. We select the negative reviews of the products.

d1

Filter

	logprice	logdiscount	logreviewcount	loglistprice	logavgcount	logsalesrank	scores.score
15	3.286534	1.386294	4.927254	3.332205	4.444121	5.602119	-2
33	1.238374	3.433987	5.488938	1.607436	5.565478	8.796490	-2
44	3.019937	3.891820	2.197225	3.688629	3.953645	9.553646	-2
62	2.995732	4.219508	4.543295	2.995732	5.181784	13.947888	-1
92	2.875822	2.397895	6.719013	2.995232	5.679319	5.342334	-3
124	1.790091	0.000000	1.609438	1.791759	4.069027	10.556333	-1
142	3.951051	3.871201	3.091042	4.605070	5.514940	13.114533	-4
144	1.163151	4.094345	2.197225	2.079442	3.909520	10.918754	-3

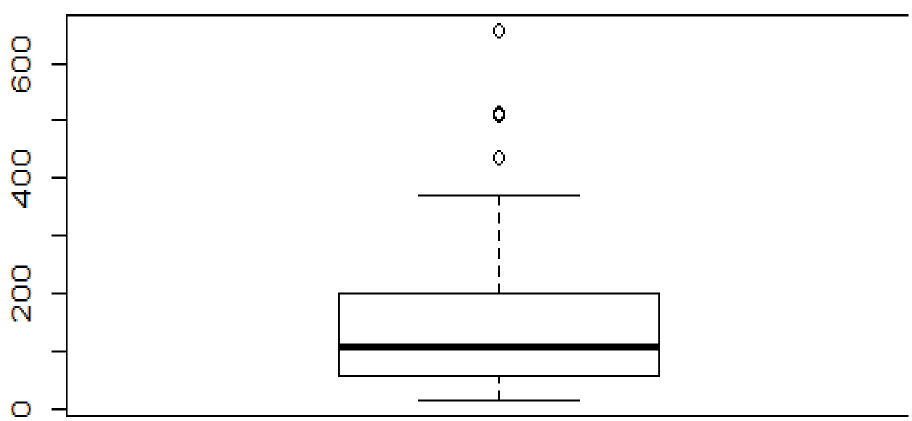
2. The reviews which we select the reviews validated by deepEmoji online https://deepmoji.mit.edu/ which is having confidence level more or equal to medium level for sarcastic and irony.

READABILITY ANALYSIS

Based on research in ***readability***, these metrics are valuable metrics for measuring how tranquil is for a user to read a review. Readability of the product is the count of characters to the number of sentences. Finally, to justification for the cognitive cost required to read a review, we computed the average number of characters per sentence in all the reviews.

Average character count for of all reviews of a product

1	117.75
2	269.5
3	92.125
4	438
5	170.375
6	108.375
7	189.25
8	276.25
9	119.875
10	116.25
11	438
12	44.125
13	73.375
14	85.625
15	185.625
16	260.125
17	332.25
18	509.375
19	48.375
20	513.875
. .	. .
86	137



As illustrated above, the box plot and value spread clearly indicate the average character count of all reviews of the single product.

REGRESSION MODEL AND DEEP MODEL SPECIFICATION

REGRESSION MODEL

These independent variables significance computed by proposed model as mentioned below.

Step 1: Define model LM

$$\log(SalesRank)_k = \propto + \beta1.AvgSentiment_k + \beta2.logRating_k + \beta3.\ logReviewCount_k + \beta4.\ logRead_k + \beta5.\ logPrice_k + \beta6.\ logDiscount_k + \beta7.\ logListPrice_k + U_k \text{ --- } (1)$$

Step 2: Compute the stepwise regression backward.

Step 3: select best model

Step 4: Compute RMSE

MAXNETR NEURAL NETWORK (DEEP LEARNING)

STEP 1: Define training and test data

STEP2: Create data for the model and fully connected hidden layer with number of neurons defined to be one

Step 3: Define regression at output layer

Step 4: Define model with RMSE predefine metric for understanding performance of the model

Step 5: Predict the performance with test data for the trained model

KERAS SEQUENTIAL MODEL DEEP LEARNING

STEP 1: Define Data into matrix and generate sequence model

STEP 2: Define fully connected or dense layer of the model with linear activation function

STEP 3: Define call back of the model has loss function

STEP 4: Loss function define with least square error

STEP 5: Define 500 episode to accuracy

TESTING AND PERFORMANCE EVALUATION

The project is unique in looking at how sarcasm score, *sentimental* score, *readability* and *product & sales characteristics (price, discount, and rating)* in the text of reviews affect product sales. As presented in previous section complete regression, MAXNET Deep Neural Network and KERAS deep learning SEQUENTIAL model analysis.

Model	**Non-Sarcastic + Readability + Product Information**	**Sarcastic + Readability + Product Information**
	RMSE	***RMSE***
Regression LM	3.72	14.72
Deep Learning MAXNET	2.03	5.9
KERAS SEQUENTIAL MODEL	1.26	14.62

Table: The comparison of RMSE of NON-SARCASTIC/SARCASTIC reviews with readability and product information

Therefore, we concluded that due the impact of sarcastic reviews, product price, discount, pricelist, readability and reviewcount impact over Sales Rank of the product is worse than non-sarcastic reviews. The RMSE score is **14.62** with KERAS SQUENTIAL DEEP MODEL, **5.9** with deep learning MAXNET and **14.72** Regression LM.
In comparison It is evident that non-sarcastic review positive effect of sales Rank that increase the productivity as compared to sarcastic. The non-sarcastic, readability, and product information with KERAS sequential model is **1.26** which is best model for sales impact. The second best is deep learning MAXNET and third one is regression LM with **2.03** and **3.72** respectively.

USABILITY AND SOCIAL IMPACT

This is likely to occur when the reviewer clearly outlines the pros and cons of the product, thereby providing sufficient information to the consumer/maker/store contribute the productivity fruitful. Overall, we consider this work a significant step in understanding the factors that affect the perceived economic in the form of customer contribution towards feeling oriented crowdsourcing.

CONCLUSION AND RECOMENDATION

To understand the product information together with sentiments or credibility, readability. The sentiments can critic, ironic and satirical towards product consumer-ship. The results have proven that product information with sentiment without sarcastic has good effect on sales of product because of the results shown more accuracy and less errors in predictivity using RMSE metric.

Further in future that there are many interesting problems that need to be addressed in this area:

- The subjectivity and objectivity analysis of reviews along product & sales characteristics.
- Combine the sentiments (positive, mixed or negative) with subjectivity analysis. Negative reviews may increase sales if the reviews are informative
- The detail sarcastic sentiments like ironic and verbal ironic impact over sales

BIBLIOGRAPHY

Bhatt, A. P. A. C. H. a. G. K., 2015. Amazon Review Classification and Sentiment. *International Journal of Computer Science and Information Technologies,* Volume Vol. 6 (6) , pp. 5107-5110.

Breen, J., 2014. *slides from my R tutorial on twitter text mining#rstats "Things i tend to forget".* [Online] Available at: http://jeffreybreen.wordpress.com/2011/07/04/twitter-text-mining-r-slides/

Chevalier, J. A. a. M. D., 2006. The effect of word of mouth on sales: Online book reviews. *Journal Marketing Research 43.3.(Aug.2006),* 43(3), pp. 345-354.

Das, S. a. C. M., 2007. Yahoo! for Amazon: Sentiment extraction from small talk on the web.. *Management Science,* Volume 53(9), pp. pp.1375-1388.

Dellarocas, C. A. a. Z. M., 2005. *Using online ratings as a proxy of word-of-mouth in motion picture revenue forcasting.* University of Maryland, working paper.

Dellarocas, C. a. Z., 2007. Exploring the value of online product ratings in revenue forcasting: The case of motion pictures.. *Wiley Periodicals, Inc. and JOURNAL OF INTERACTIVE MARKETING,* Volume 21.

Ghose, A. a. I. P., 2007. *Designing Novel Review Ranking Systems: Ranking Systems:Predicting Usefulness and Impact of Reviews.* Minneapolis, Minnesota, USA., ACM 978-1-59593-700-1/07/0008 ...$5.00..

Ghose, A. a. I. P., 2011. Estimating the helpfulness and ecnomic impact of product reviews: Mining Text and reviewer characteristics. *knowledge and Data Engineering, IEEE,* 23(10), pp. 1498-1512.

Hu, M. a. L. B., 2002. *Mining and Summarizing Customer Reviews.* s.l., Proceding of ACM SIGKDD International Conference on Knowledge.

Liu, B. H. M. a. C. J., 2005. *Opinion Observer: Analyzing and Comparing Opinions on the web.* Chiba,Japan, International World wide Web conference (WWW-2005).

Pang, B. a. L. L., 2008. Opinion mining and sentiment analysis.. *Foundations and Trends in Information Retrieval,* Volume 2(1-2), pp. 1-135.

Taboada, M. B. J. T. M. V. K. a. S. M., 2011. Lexicon-based methods for sentiment analysis. *Computational Linguitics,* 37(2), pp. pp. 267-307.

Turney, P., 2002. *Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews.* s.l., In Proceedings of the 40th Annual Meeting, Association for Computational Linguistics, pp. 417-424.

APPENDIX- PROJECT PLAN

2.1 Detail Schedule and Milestone

Milestone1

1. Getting familiar with the Domain

It is important to understand the domain by literature study and domain data. We plan to understand the existing literature first and their methodology. The data collection is real time using API because it is not automated study without knowing the offline nature of data.

Data will be stored in file format .csv and category wise archived.

2. Intensive search about related topics
3. The system high level architecture must be built because this is the step important to pursue further.
4. The system conceptual model with libraries and details.

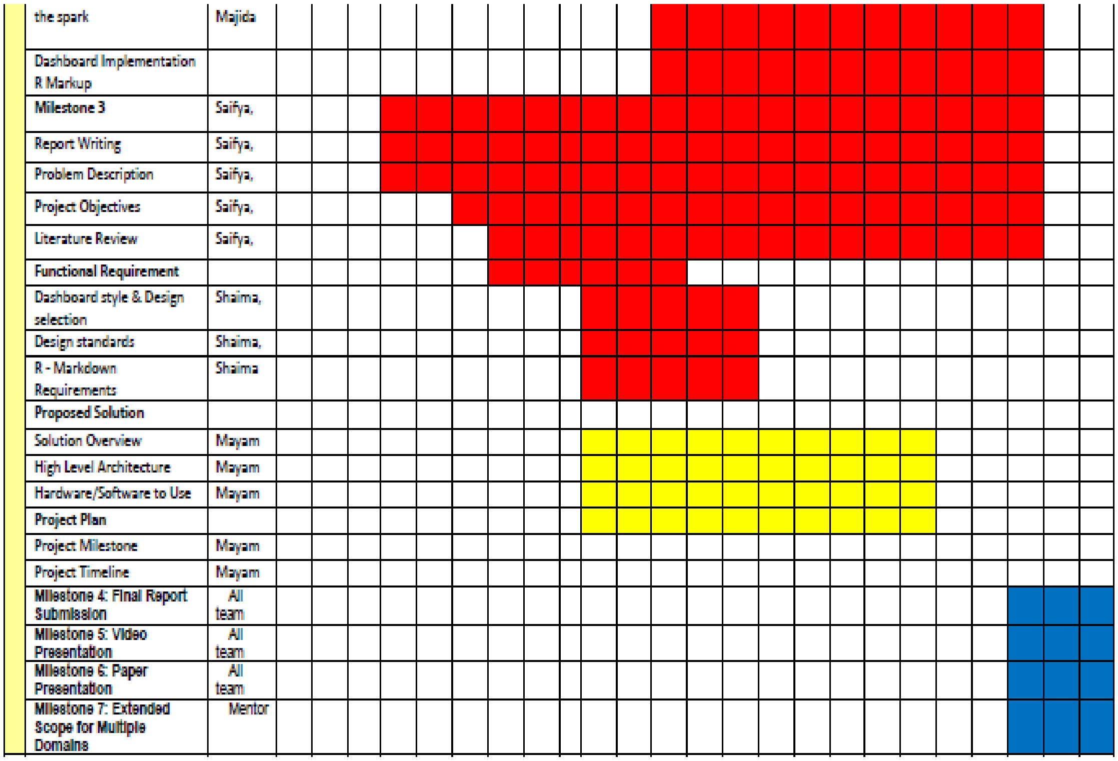
Milestone 2

The dataset will be prepared for pre-processing at this level with the operations of cleaning, normalization and word2vector representation. The word2vector representation is must to seek features in vector further feed to model. The model will produce the features of sentiments, personality and contextual then combined with base model features of sarcasm. The model dimensions are big due to deep learning model so required framework to handle by the spark.

Milestone 3:

As the reports contain many parts where each need detailed description and explanation and the team has 19 weeks to finish it, starting with the problem statement, the description, and the objectives in the 8th week was easy after clearing out and finalizing the project idea. After working on the dashboard implementation as software, the team was able to proceed with the report and write all what is done.

	Resources	W 1	W 2	W 3	W 4	W 5	W 6	W 7	W 8	W 9	W 10	W 11	W 12	W 13	W 14	W 15	W 16	W 17	W 18	W 19	W 20	W 21	W 22	W 23	W 24
Milestone 1	All team																								
Literature Review	Mentor and Team																								
Data Collection	Maryam																								
Streaming API JSON Implementation in R	Maryam																								
Data Preparation for File	Safiya																								
Storage Data by Categories	Safiya																								
System Architecture	Implementation Team																								
Conceptual Model	Saifya, Shaima, Majida																								
Milestone 2																									
Pre-processing	Saifya, Shaima, Majida																								
Date Clearning, normalization, word embedding	Saifya, Shaima, Majida																								
Algorithm Design & Implementation	Safiya																								
Feature extraction of sentiment , sarcasm, contextual	Safiya																								
Pre-train features addition	Safiya																								
Prediction of sarcasm	Saifya, Shaima, Majida																								
Predicted sarcasm on sales and product variables with	Saifya, Shaima,																								



ANTICIPATED RISK	MITIGATION PLAN
The stream might slow down the process streaming product and sales data from Amazon.	To search alternative tool to scraping the data quickly with minimum cost.
Implementation of sarcasm features with the deep learning model.	Required to see existing literature model best for feature extraction.

CODING

DATA PREPARATION & SARCASTIC / SENTIMENTS REVIEWS SCORES & READIBILITY ANALYSIS

```
setwd('C:\\sarcasm\\chapter-6_ResultDiscussion\\Home & Furniture\\Home & Furniture\\Refined-final')
getwd()

#-----Section 02-------------------------------------------
# get data file rerefined-4.csv and put relevant variables in a data frame
#ikea_data <- read.csv("rerefined-4.csv", stringsAsFactors = FALSE)
#ikea_data <- read.csv("lighting_1starabove.csv", stringsAsFactors = FALSE)
```

```
ikea_data <- read.csv("rerefined-4.csv", stringsAsFactors = FALSE)
ikea_data_NUM<-ikea_data

ikea_data_NUM$Available<-NULL
ikea_data_NUM$ASIN<-NULL
ikea_data_NUM$Title<-NULL
ikea_data_NUM$Category<-NULL
ikea_data_NUM$Review1<-NULL
ikea_data_NUM$Review2<-NULL
ikea_data_NUM$Review3<-NULL
ikea_data_NUM$Review4<-NULL
ikea_data_NUM$Review5<-NULL
ikea_data_NUM$Review6<-NULL
ikea_data_NUM$Review7<-NULL
ikea_data_NUM$Review8<-NULL


str(ikea_data)
## boxplot (bottom left) quantile , median, and mode analysis for skewwness varification
boxplot(ikea_data$Price, xlab="listprice", ylab="Value axis", main="Simple Box plot", col="lightblue")
leg1 <- paste("median = ", round(median(ikea_data$Price), digits = 4))
leg11 <- paste("mean = ", round(mean(ikea_data$Price), digits = 4))
lq <- quantile(ikea_data$Price, 0.25)
leg2 <- paste("75th quantile =  ", round(lq,digits = 4))
uq <- quantile(ikea_data$Price, 0.75)
leg3 <- paste("100th quantile = ", round(uq,digits = 4))
legend(x = "right",  paste(leg1, leg11,leg2, leg3, sep = "\n"), bty = "n")


logreviewcount=log(ikea_data$ReviewCount)

logprice=log(ikea_data$Price)
logsalesrank=log(ikea_data$SalesRank)
loglistprice=log(ikea_data$ListPrice)
logdiscount=log(ikea_data$Discount)

logsalesrank[ is.infinite(logsalesrank) ] <- 0 # replace all NA values with 0
logprice[ is.infinite(logprice) ] <- 0
logdiscount[ is.infinite(logdiscount) ] <- 0
loglistprice[ is.infinite(loglistprice) ] <- 0


boxplot(loglistprice,logprice, logdiscount, logreviewcount,logsalesrank,
        names=c( "listprice", "price", "discount", "reviewcount","salesrank" ),
        xlab="ikea-product-variables")


loglistprice=log(ikea_data$ListPrice)
logdiscount=log(ikea_data$Discount)

logsalesrank[ is.infinite(logsalesrank) ] <- 0 # replace all NA values with 0
logprice[ is.infinite(logprice) ] <- 0
logdiscount[ is.infinite(logdiscount) ] <- 0
loglistprice[ is.infinite(loglistprice) ] <- 0
```

```
boxplot(loglistprice,logprice, logdiscount, logreviewcount,logsalesrank,
        names=c("listprice", "price", "discount", "reviewcount","salesrank" ),
        xlab="ikea-product-variables")


qqnorm(ikea_data$Price, xlab = "Theoretical Quantiles: Price" )
qqline(ikea_data$Price, col=2) ## red color

# Reduce the skewness and fit the value within qrtile by mean


# Reduce the skewness and fit the value within qrtile by mean
qqnorm(logprice, xlab = "Theoretical Quantiles: Price" )
qqline(logprice, col=2) ## red color



qqnorm(ikea_data$ReviewCount, xlab = "Theoretical Quantiles: Price" )
qqline(ikea_data$ReviewCount, col=2) ## red color


qqnorm(logreviewcount, xlab = "Theoretical Quantiles: ReviewCount" )
qqline(logreviewcount, col=2) ## red color


qqnorm(ikea_data$SalesRank, xlab = "Theoretical Quantiles: ReviewCount" )
qqline(ikea_data$SalesRank, col=2) ## red color

qqnorm(logsalesrank, xlab = "Theoretical Quantiles: ReviewCount" )
qqline(logsalesrank, col=2) ## red color



qqnorm(ikea_data$Discount, xlab = "Theoretical Quantiles: Discount" )
qqline(ikea_data$Discount, col=2) ## red color

qqnorm(logdiscount, xlab = "Theoretical Quantiles: Discount" )
qqline(logdiscount, col=2) ## red color



library(plyr)
library(stringr)

# function score.sentiment
score.sentiment = function(sentences, pos.words, neg.words, .progress='none')
{
  # Parameters
  # sentences: vector of text to score
  # pos.words: vector of words of postive sentiment
  # neg.words: vector of words of negative sentiment
  # .progress: passed to laply() to control of progress bar

  # create simple array of scores with laply
  scores = laply(sentences,
                 function(sentence, pos.words, neg.words)
```

```
                 {
                  # remove punctuation
                  sentence = gsub("[[:punct:]]", "", sentence)
                  # remove control characters
                  sentence = gsub("[[:cntrl:]]", "", sentence)
                  # remove digits?
                  sentence = gsub('\\d+', '', sentence)

                  # define error handling function when trying tolower
                  tryTolower = function(x)
                  {
                    # create missing value
                    y = NA
                    # tryCatch error
                    try_error = tryCatch(tolower(x), error=function(e) e)
                    # if not an error
                    if (!inherits(try_error, "error"))
                      y = tolower(x)
                    # result
                    return(y)
                  }
                  # use tryTolower with sapply
                  sentence = sapply(sentence, tryTolower)

                  # split sentence into words with str_split (stringr package)
                  word.list = str_split(sentence, "\\s+")
                  words = unlist(word.list)

                  # compare words to the dictionaries of positive & negative terms
                  pos.matches = match(words, pos.words)
                  neg.matches = match(words, neg.words)

                  # get the position of the matched term or NA
                  # we just want a TRUE/FALSE
                  pos.matches = !is.na(pos.matches)
                  neg.matches = !is.na(neg.matches)

                  # final score
                  score = sum(pos.matches) - sum(neg.matches)
                  return(score)
                 }, pos.words, neg.words, .progress=.progress )

  # data frame with scores for each sentence
  scores.df = data.frame(text=sentences, score=scores)
  return(scores.df)
}


# CREATING REVIEW DATA SEPERATELY AND Other quantative data seperately
ikea_reviewdata<-
data.frame(ikea_data$Review1,ikea_data$Review2,ikea_data$Review3,ikea_data$Review4,ikea_data$Review5,ike
a_data$Review6,ikea_data$Review7,ikea_data$Review8)

myvars <- names(ikea_data) %in% c("ikea_data$Review1",
                                  "ikea_data$Review2",
                                  "ikea_data$Review3",
```

```
                  "ikea_data$Review4",
                  "ikea_data$Review5",
                  "ikea_data$Review6",
                  "ikea_data$Review7",
                  "ikea_data$Review8"
)
ikea_dataset1 <- ikea_data[!myvars]
rm(myvars)




# import positive and negative words
pos = readLines("positive_words.txt")
neg = readLines("negative_words.txt")


# apply function score.sentiment
scores = score.sentiment(ikea_reviewdata$ikea_data.Review1, pos, neg, .progress='text')
scores$review2 = score.sentiment(ikea_reviewdata$ikea_data.Review2, pos, neg, .progress='text')
scores$review3 = score.sentiment(ikea_reviewdata$ikea_data.Review3, pos, neg, .progress='text')
scores$review4 = score.sentiment(ikea_reviewdata$ikea_data.Review4, pos, neg, .progress='text')
scores$review5 = score.sentiment(ikea_reviewdata$ikea_data.Review5, pos, neg, .progress='text')
scores$review6 = score.sentiment(ikea_reviewdata$ikea_data.Review6, pos, neg, .progress='text')
scores$review7 = score.sentiment(ikea_reviewdata$ikea_data.Review7, pos, neg, .progress='text')
scores$review8 = score.sentiment(ikea_reviewdata$ikea_data.Review8, pos, neg, .progress='text')
# add variables to data frame
scores$very.pos = as.numeric(scores$score >= 2)
scores$very.neg = as.numeric(scores$score <= -2)

# how many very positives and very negatives
numpos = sum(scores$very.pos)
numneg = sum(scores$very.neg)

# global score
global_score = round( 100 * numpos / (numpos + numneg) )


boxplot(scores$score,scores$review2$score,scores$review3$score,
    scores$review4$score,scores$review5$score,scores$review6$score,
    scores$review7$score,scores$review8$score,
    names=c("review1", "review2", "review3", "review4", "review5","review6","review7","review8" ),
    xlab="ikea-product-review variable")


#average score for each product sentiments
averagrescore<- data.frame(ikea_data$ASIN,scores$score,scores$review2$score,scores$review3$score,+
               scores$review4$score,scores$review5$score,scores$review6$score+
               scores$review7$score,scores$review8$score)
avgscoref<-data.frame(averagrescore)

loglistprice[ is.na(loglistprice) ] <- 0

avgscoref<-data.frame(ID=avgscoref[,1], Means=rowMeans(avgscoref[,-1]))
```

```
#average review word count represent readibility strength of reviewers
library(qdap)

j1<-character_count(ikea_reviewdata$ikea_data.Review1, byrow=TRUE)   # character couting for each row
j2<-character_count(ikea_reviewdata$ikea_data.Review2, byrow=TRUE)
j3<-character_count(ikea_reviewdata$ikea_data.Review3, byrow=TRUE)
j4<-character_count(ikea_reviewdata$ikea_data.Review4, byrow=TRUE)
j5<-character_count(ikea_reviewdata$ikea_data.Review5, byrow=TRUE)
j6<-character_count(ikea_reviewdata$ikea_data.Review6, byrow=TRUE)
j7<-character_count(ikea_reviewdata$ikea_data.Review7, byrow=TRUE)
j8<-character_count(ikea_reviewdata$ikea_data.Review8, byrow=TRUE)


#building frame for char counting of reviews of each product and its mean
countChar<-data.frame(j1,j2,j3,j4,j5,j6,j7,j8)

countChar[ is.na(countChar) ] <- 0 # replace all NA values with 0

averagecountChar<-data.frame(ID=countChar[,0], Means=rowMeans(countChar))

logavgcount=log (averagecountChar$Means)


boxplot(averagecountChar$Means)

logavgscore<-log(avgscoref$Means)
logavgscore


ikea_dataset1<-data.frame(logprice,
                          logdiscount,
                          logreviewcount,
                          loglistprice,
                          logavgcount,
                          logsalesrank,

scores$score,scores$review2$score,scores$review3$score,scores$review4$score,scores$review5$score,scores$revie
w6$score,scores$review7$score,scores$review8$score)
# ikea_dataset1<-data.frame(ikea_data$Price,
#                           ikea_data$Discount,
#                           ikea_data$ReviewCount,
#                           ikea_data$ListPrice,
#                           averagecountChar,
#
ikea_data$SalesRank,scores$score,scores$review2$score,scores$review3$score,scores$review4$score,scores$revie
w5$score,scores$review6$score,scores$review7$score,scores$review8$score)


# ikea_dataset1<-data.frame(logprice,
```

```
#                          logreviewcount,
#                          logavgcount,
#                          logsalesrank
#                         )
d1<-as.data.frame(ikea_dataset1[ikea_dataset1$scores.score<0,1:7],drop=FALSE)
d1
d2<-as.data.frame(ikea_dataset1[ikea_dataset1$scores.review2.score<0,c(1,2,3,4,5,6,8)],drop=FALSE)
d3<-as.data.frame(ikea_dataset1[ikea_dataset1$scores.review3.score<0,c(1,2,3,4,5,6,9)],drop=FALSE)

# j<-merge(d1, d2, by = "row.names", all = TRUE)

# col1 <- c("ab","bc","cd","de")
# col2 <- c(1,2,3,4)
# df1 <- as.data.frame(cbind(col1,col2))
# col1 <- c("ab","ef","fg","gh")
# col3 <- c(5,6,7,8)
# df2 <- as.data.frame(cbind(col1,col3))
# library(plyr)
# d1$ikea_data.Price
# Example <- join(d1,d2,by = "d1$ikea_data.Price", type = "full") #Does not keep col3
# library(dplyr)
# Example <- full_join(df1,df2,by = "d1") #Function not recognised

d3<- scores[scores$review3$score<0,]
d4<- scores[scores$review4$score<0,]

d5<- scores[scores$review5$score<0,]
d6<- scores[scores$review6$score<0,]

d7<- scores[scores$review7$score<0,]

d8<- scores[scores$review8$score<0,]


scores$very.pos
scores$very.neg
```

DEEP LEARNING KERAS RNN, MAXNET AND REGRESSION LM MODEL

```
install_tensorflow()

install.packages("devtools")

devtools::install_github("rstudio/keras")

install_keras()
```

```
install.packages("devtools")

install_github("rstudio/reticulate")
install_github("rstudio/tensorflow")
install_github("rstudio/keras")

library(keras)

library(tensorflow)

library(reticulate)
require(devtools)

py_module_available('keras')
py_module_available('tensorflow')

py_discover_config('keras')

sess = tf$Session()
hello <- tf$constant('Hello, TensorFlow!')
sess$run(hello)
```

```
#mxnet regression version deep learning

install.packages("mlbench")
install.packages("mxnet")
require(mlbench)
cran <- getOption("repos")
cran["dmlc"] <- "https://apache-mxnet.s3-accelerate.dualstack.amazonaws.com/R/CRAN/"
options(repos = cran)
install.packages("mxnet")
install.packages("stringr")
install.packages("stringi")

require(mxnet)
require(stringr)
require(stringi)

train.ind = seq(1, 148, 3)
train.x = data.matrix(ikea_dataset1[train.ind, -4])
train.y = ikea_dataset1[train.ind, 4]
test.x = data.matrix(ikea_dataset1[-train.ind, -4])
```

```
test.y = ikea_dataset1[-train.ind, 4]
x_train<-train.x
y_train<-train.y

x_test<-test.x
y_test<-test.y
```

#the accuracy of non-sarcastic is more than sarcastic one because the rmse is getting down which is error

#after 300 episodes and drastically showing the impact on sales better interms of RMSE

```
# Define the input data
data <- mx.symbol.Variable("data")
# A fully connected hidden layer
# data: input source
# num_hidden: number of neurons in this layer
fc1 <- mx.symbol.FullyConnected(data, num_hidden=1)

# Use linear regression for the output layer
lro <- mx.symbol.LinearRegressionOutput(fc1)

mx.set.seed(0)
model <- mx.model.FeedForward.create(lro, X=train.x, y=train.y,
```

```
                              ctx=mx.cpu(), num.round=500, array.batch.size=20,
                              learning.rate=2e-6, momentum=0.9, eval.metric=mx.metric.rmse)

preds = predict(model, test.x)
## Auto detect layout of input matrix, use rowmajor..
sqrt(mean((preds-test.y)^2))
#-------------------------------------Regression-----------------------------------------------------
library(tidyverse)
library(caret)
library(leaps)
library(MASS)
# Fit the full model
full.model <- lm(ikea_dataset1$logsalesrank ~., data = ikea_dataset1)
# Stepwise regression model
set.seed(123)

train.control <- trainControl(method = "cv", number = 10)

step.model <- stepAIC(full.model,
                tuneGrid = data.frame(nvmax = 1:5),
                trControl = train.control
)

summary(step.model)

predictions <- predict(full.model, ikea_dataset1)
```

```
RMSE(ikea_dataset1$logsalesrank, predictions)

#--------------------------------KERAS RNN-----------------------------------------------

library(keras)

#Prepare mtcars data for Keras

data('mtcars')

mtcarsmat<-as.matrix(ikea_dataset1)

y<-as.numeric(mtcarsmat[,6])

mtcarsxsc<-scale(mtcarsmat)

#Multiple linear regression

modellr <- keras_model_sequential()

modellr %>%

  layer_dense(units = 1, input_shape = dim(mtcarsxsc)[2], activation='linear') %>%

  summary(modellr)

early_stopping <- callback_early_stopping(monitor ='loss', min_delta=0.000001)

modellr %>% compile(

  loss = 'mean_squared_error',

  optimizer = optimizer_adam(lr = 0.001)

)
```

```
historylr <- modellr %>% fit(
  mtcarsxsc, y,
  epochs = 500, batch_size = 1,callbacks = early_stopping
)

plot(historylr)
lrweights <- get_weights(modellr)
plot(lrweights[[1]],xlab="Input variable",ylab="Weight",ylim=c(-1.5,1.5))
summary(historylr$metrics)

plot(historylr)

predictions <- predict(modell1, mtcarsxsc)

r<-RMSE(y, predictions)
```