Paper 261-2013

# Investigating the Impact of Amazon Kindle Fire HD 7” on Amazon.com Consumers Using SAS® Text Miner and SAS® Sentiment Analysis

Srihari Nagarajan, SAS Institute, Hari Harasudhan, Kavi Associates, Goutam Chakraborty,
Oklahoma State University, Stillwater, OK, USA

## ABSTRACT

This paper demonstrates the application of text mining techniques to collect, group and summarize positive and negative opinions on a product. Unfortunately for popular products often there are far too many reviews that make it difficult to read through all reviews and make a decision. For the purpose of this poster, I have developed a tool using ASP.NET to extract (1674) customer reviews for Kindle Fire HD 7” from Amazon.com. On the excel data set thus generated, text mining can be performed to summarize customer comments by grouping related reviews into clusters. The Text Parsing, Text Filter, Text Topic and Text Cluster nodes are used and outputs from every node are discussed. Sentiment Analysis is performed on the data set to develop a model for classifying positive and negative reviews.

## INTRODUCTION

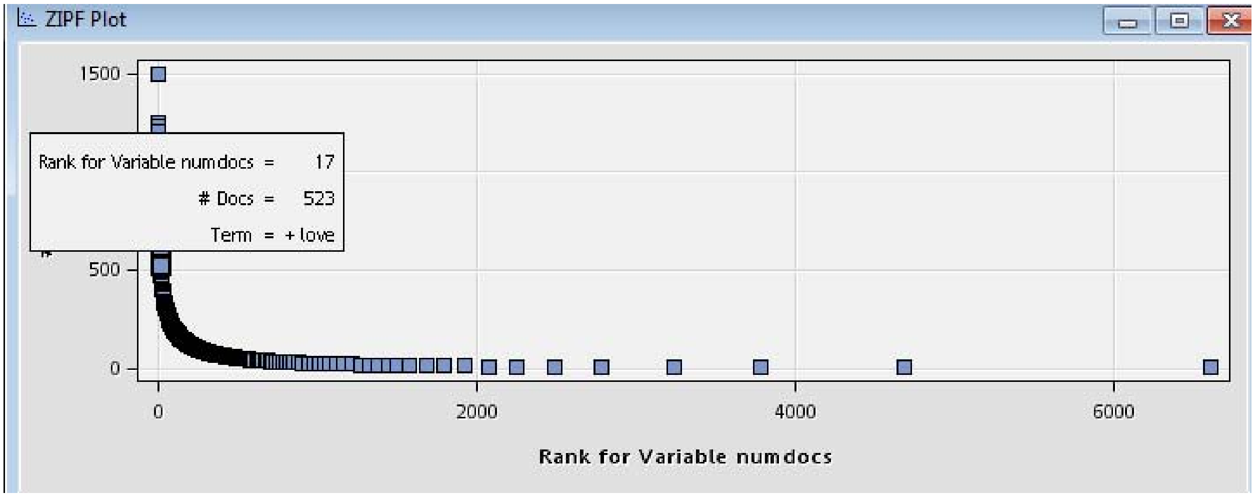
Consumers consider customer reviews posted on ecommerce websites before making a purchase decision. Product manufacturers also consider customer reviews to identify what consumers feel about their products. These reviews help companies identify the strengths and weaknesses of their products and investigate whether the product meets the customer expectations or not. Unfortunately for popular websites and popular products often there are far too many reviews that make it difficult for a prospective buyer to read through all reviews and make a decision. For the purpose of this poster, customer reviews, star rating at review level, comment title, number of comments, number of people who found it useful were extracted using the ASP.NET web crawler tool for Amazon Kindle Fire HD 7”’’. The extracted data for analysis purpose was frozen on October 31, 2012. The tool is designed in such a way that regardless of the number of reviews, it would remove all irrelevant characters from the website and export the text contents alone to an excel worksheet by placing one comment per row. Then, by importing this excel worksheet as a SAS data set, text mining can be performed to summarize customer comments by grouping related reviews into clusters. The Text Parsing, Text Filter, Text Topic and Text Cluster nodes are used and outputs from every node are discussed. Cluster and concept link analysis revealed a great amount of information on what factors concern the users of Kindle Fire HD 7” and topic analysis reveals details on what improvements people were expecting and how this version of Kindle met their expectations. We can also get details on what features most of the people are talking about and the drawbacks of this product that most people are concerned about. Finally we predict the star rating of reviews based on the text topics using Decision Tree algorithm. .

## DATA PREPARATION

Data available in Amazon.com is highly unstructured. This data is extracted by passing the URL of the customer reviews page to the web crawler tool. This tool is written in ASP.NET and is customized to fetch reviews for any available product from a famous e-commerce website. This reusable tool would take the review page URL as the input and would crawl across all review pages for a particular product. The tool filters all irrelevant html tags, empty spaces, special characters and other symbols from the reviews page and extracts just the customer reviews from all pages. This ASP.NET web crawler tool is also available in a GUI interface where the user needs to enter just the URL of the reviews page in a textbox and click the submit button. This tool would extract the review text, star rating, review title, number of customers who found the review useful, number of customers who found the review useless as separate columns of an excel sheet. This excel sheet is then converted into a SAS dataset using SAS Enterprise Guide. This SAS dataset for Amazon Kindle Fire HD 7”” consists of 1674 observations as on Oct 31, 2012.

## TEXT PARSING

The variables in the input data set are converted into SAS data set and are suitable roles. This input SAS data set is now fed into the Text parsing node in SAS Enterprise Miner. Text parsing node was employed to parse the text and analyze the number of terms and documents by frequency. The below ZIPF plot shows that the term ***love*** was found in 523 documents which could be a good sign to the product manufacturer.



Rank for Variable numdocs = 17
# Docs = 523
Term = + love

**Figure 1. ZIPF plot results from the Text Parser Node Results**

The Synonym list should be set to include all identifiable synonyms of terms in the documents. For example Iphone is the parent term for Iphone4 and so on. The Stop list should be set so that the highly frequent terms and low frequency terms are included. These terms can be observed from the interactive filter window in the Text filter node. So this becomes an iterative process. In the Text Filter node, the default option of Check Spelling has been changed from “No” to “Yes”. The Ignore Parts of the Speech option is set to “Abbr, Aux, Conj, Det”. The most frequently used (eg. Do, the, is, and etc.) and the least frequently words used are filtered by preparing a filter data set and exported using the Text Filter node.

This Synonym data set is fed into the Text Parsing node so that these words are parsed in the Parsing node. In the properties panel of Text Parsing node, the Different Parts of Speech option is set to No. A custom Stop List was created based on frequencies of terms extracted by applying the Parsing node one time and then subjectively incorporating terms that are not suitable for this analysis. A custom dictionary dataset was also prepared and fed into the Text Filter node. Once the synonym data set and the custom stop list data sets are fed into the Text Parsing node, the text filter node now automatically filters out the most frequently used and least frequently used words based on the synonym list and cleans up the data accordingly.

The complete text mining sequential diagram shown below represents how the various nodes are connected. Kindle Fire HD 7”_REVIEWS is the SAS format of the excel dataset generated by the web crawler tool.

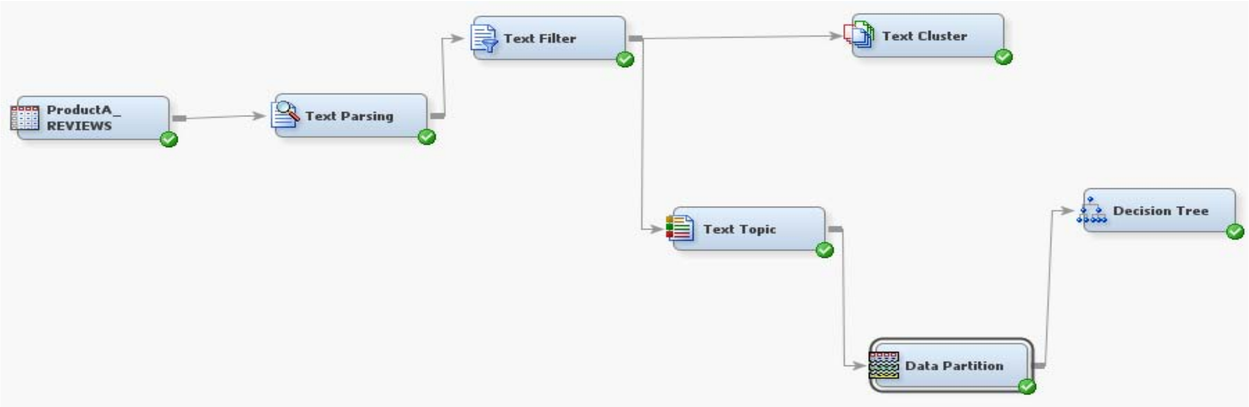
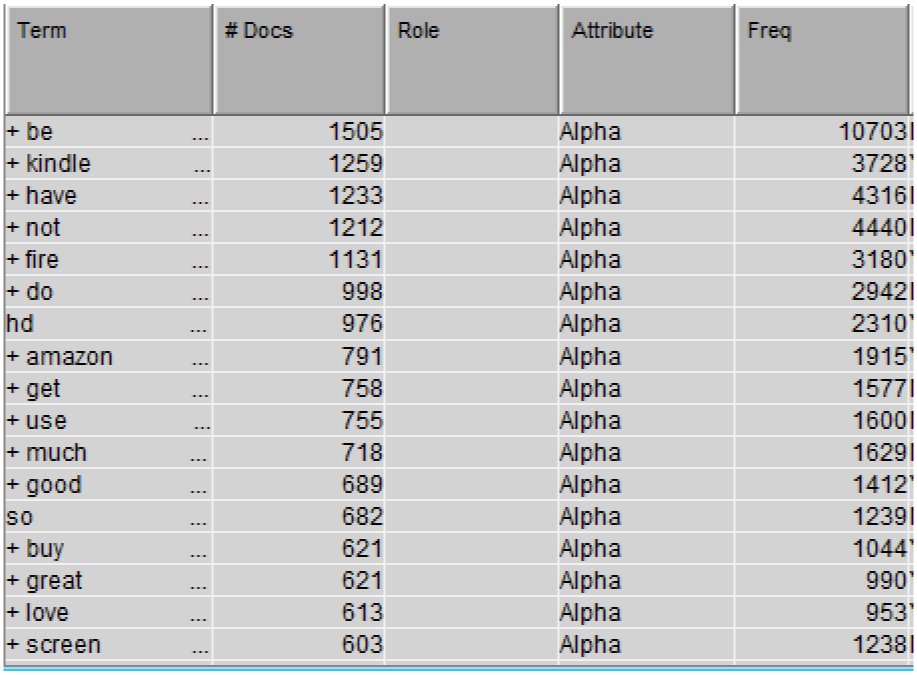


**Figure 2. Complete Text Mining diagram**

The below text parsing output shows the number of terms, their frequencies and the number of documents they are present in.



| Term | # Docs | Role | Attribute | Freq |
|---|---|---|---|---|
| + be ... | 1505 | | Alpha | 10703 |
| + kindle ... | 1259 | | Alpha | 3728 |
| + have ... | 1233 | | Alpha | 4316 |
| + not ... | 1212 | | Alpha | 4440 |
| + fire ... | 1131 | | Alpha | 3180 |
| + do ... | 998 | | Alpha | 2942 |
| hd ... | 976 | | Alpha | 2310 |
| + amazon ... | 791 | | Alpha | 1915 |
| + get ... | 758 | | Alpha | 1577 |
| + use ... | 755 | | Alpha | 1600 |
| + much ... | 718 | | Alpha | 1629 |
| + good ... | 689 | | Alpha | 1412 |
| so ... | 682 | | Alpha | 1239 |
| + buy ... | 621 | | Alpha | 1044 |
| + great ... | 621 | | Alpha | 990 |
| + love ... | 613 | | Alpha | 953 |
| + screen ... | 603 | | Alpha | 1238 |

**Figure 3. Terms window from Text Parser node output**

## TEXT FILTERING

Text Filter node was employed to keep/drop terms that were either too frequent or highly infrequent as these terms were not of much use in grouping topics. This node also gives a concept linkage distribution which helps figure out the relation between different features of the product and how strongly they are related. We can infer that the resolution is related with terms such as love, fantastic, hd, screen, good, video etc. This implies that there is an overall satisfaction with resolution. Similar analysis can be done for every feature that we are interested in.

- By clicking the Interactive Filter Viewer in the Text Filter node, we can find how strongly a term is associated with another term using Concept Links. Concept link diagram of Kindle Fire HD 7" shows that it is mostly associated with *good, love, book, read, new.*
- Most of them are positive comments made by the reviewer.
- Further refining of term *read* shows it is related to terms such as *easy, game, find* suggesting people have made comments on the ease of use of Kindle Fire HD 7".

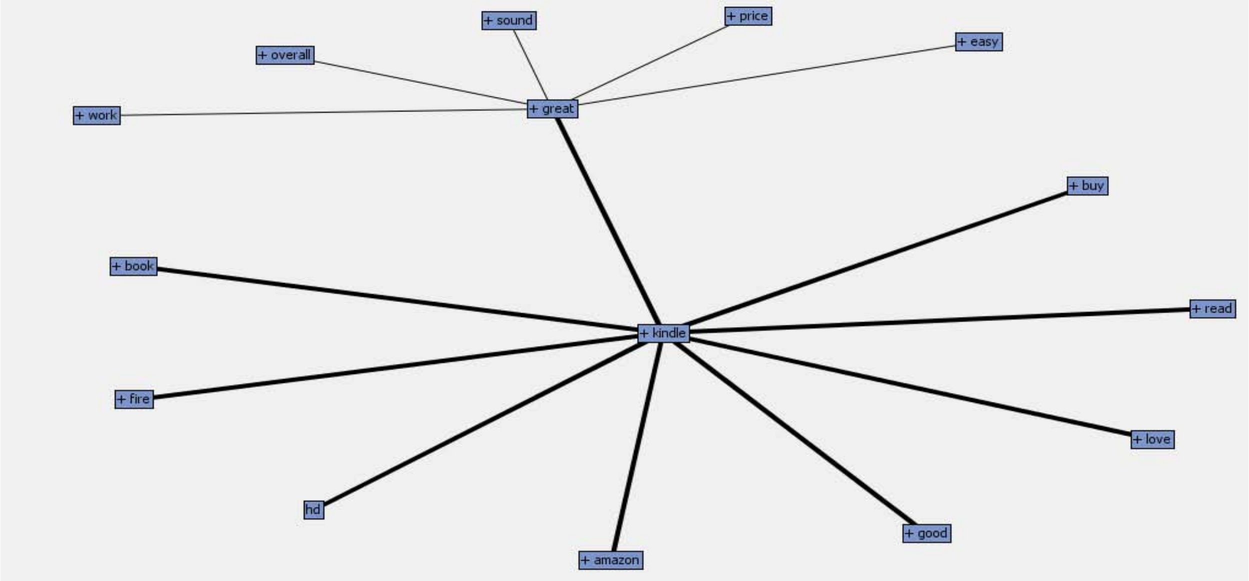


**Figure 4. Concept Link diagram showing relationship between terms**

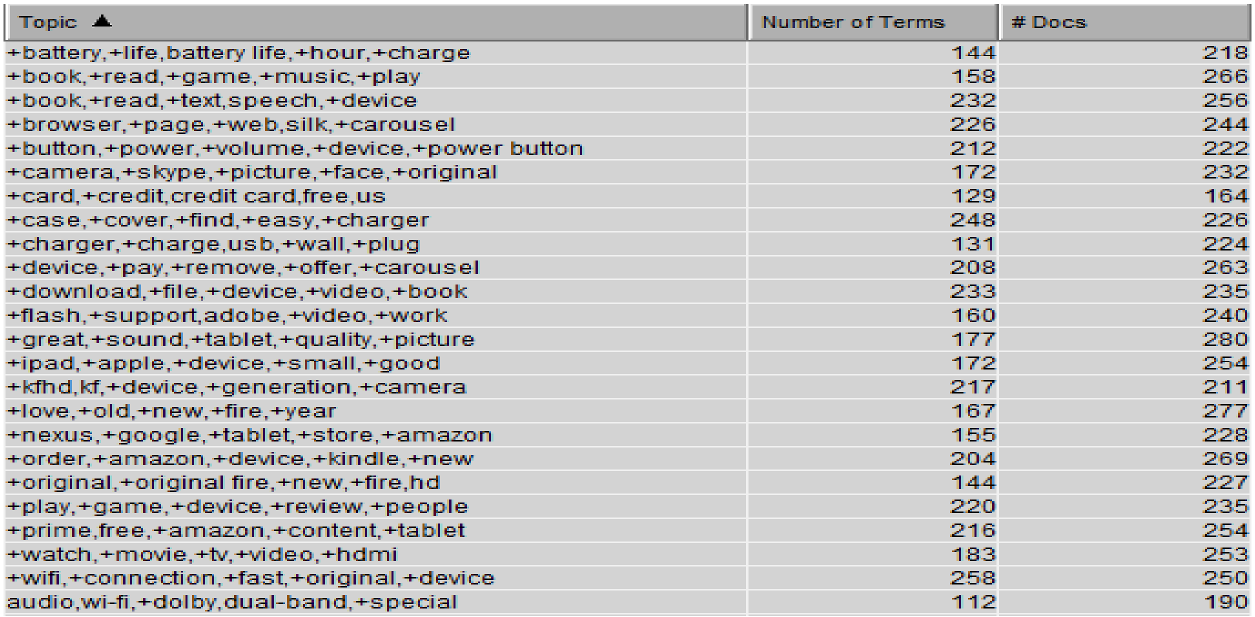
So now we can infer that Kindle is associated with buy, good, book, tablet, love. The thicker the connection line between terms, more is the strength of association between terms. We can infer that a lot of people love reading in this product. A considerable lot of people love to watch movies in this product. Many people feel that this is their new love and a lot of other people feel that this product is so original.

Further refinement could result in finer details such as people finding it easy to read, a lot of people find it better for reading and gaming and so on. By doing a more detailed filtering of terms according to business needs, this term association can be refined further to get more useful results.

## TEXT TOPIC

From the Text Topic node output, we find the terms that are grouped together and their cutoffs. We find that the terms love, buy, Amazon, fire and device occur together in 275 documents. This Text Topic output can be refined further by using a text cluster node to group them into clusters. The Text Topic node performs cluster analysis to group the documents and summarizes the collection by identifying "topics". These topics are generated by combining words that are interesting to analysts.

The below figure represents the topic grouping with the number of documents in which this grouping occurs and the number of terms in these documents. This gives an overview of the combination of terms which in-turn represents the features and feelings of consumers. This can be beneficial to the product makers to identify the features that customers are interested in and the features that consumers are dissatisfied with. This will serve as a good means to enhance their product to increase customer satisfaction.



| Topic ▲ | Number of Terms | # Docs |
|---|---|---|
| +battery,+life,battery life,+hour,+charge | 144 | 218 |
| +book,+read,+game,+music,+play | 158 | 266 |
| +book,+read,+text,speech,+device | 232 | 256 |
| +browser,+page,+web,silk,+carousel | 226 | 244 |
| +button,+power,+volume,+device,+power button | 212 | 222 |
| +camera,+skype,+picture,+face,+original | 172 | 232 |
| +card,+credit,credit card,free,us | 129 | 164 |
| +case,+cover,+find,+easy,+charger | 248 | 226 |
| +charger,+charge,usb,+wall,+plug | 131 | 224 |
| +device,+pay,+remove,+offer,+carousel | 208 | 263 |
| +download,+file,+device,+video,+book | 233 | 235 |
| +flash,+support,adobe,+video,+work | 160 | 240 |
| +great,+sound,+tablet,+quality,+picture | 177 | 280 |
| +ipad,+apple,+device,+small,+good | 172 | 254 |
| +kfhd,kf,+device,+generation,+camera | 217 | 211 |
| +love,+old,+new,+fire,+year | 167 | 277 |
| +nexus,+google,+tablet,+store,+amazon | 155 | 228 |
| +order,+amazon,+device,+kindle,+new | 204 | 269 |
| +original,+original fire,+new,+fire,hd | 144 | 227 |
| +play,+game,+device,+review,+people | 220 | 235 |
| +prime,free,+amazon,+content,+tablet | 216 | 254 |
| +watch,+movie,+tv,+video,+hdmi | 183 | 253 |
| +wifi,+connection,+fast,+original,+device | 258 | 250 |
| audio,wi-fi,+dolby,dual-band,+special | 112 | 190 |

**Figure 5. Topics with number of terms and documents from Text Topic results**

## TEXT CLUSTER

The Text Cluster node groups the terms into clusters where each cluster represents the related terms that occur together.  This view can be useful to product manufacturers in the sense that the related terms are grouped into clusters and the biggest sector in the circle represents the topics that most customers are talking about.

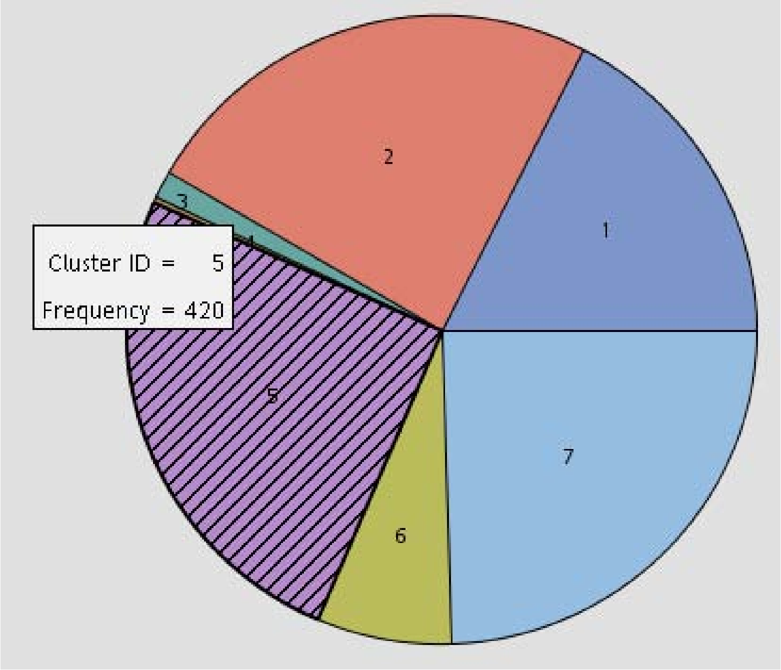
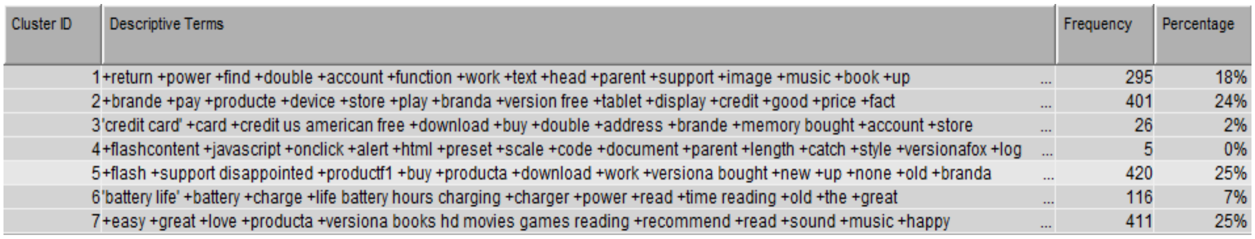


**Figure 6. Cluster diagram from Text Cluster output**

The Text Cluster node produces 7 clusters with each cluster talking about different aspects of Amazon Kindle Fire HD 7".

- Cluster 6 talks about the *battery*, *chargers* and *read time* of the product.
- Cluster 7 talks about some good points about *books*, *movies, games* and *sound.*
- Cluster 5 mainly talks about the problems associated with Kindle Fire HD 7" and has a number of dissatisfied customers.



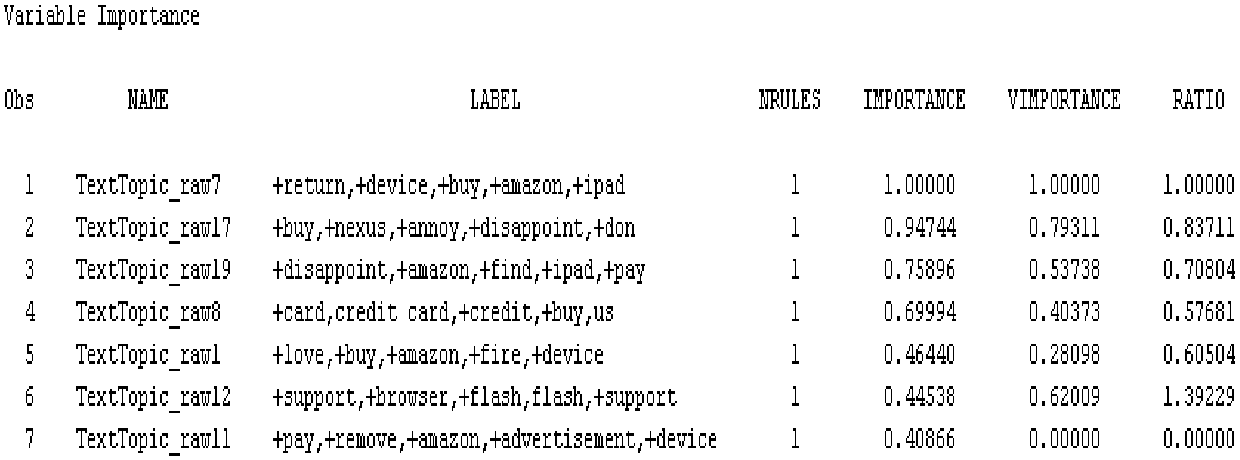
| Cluster ID | Descriptive Terms | | Frequency | Percentage |
|---|---|---|---|---|
| 1 | +return +power +find +double +account +function +work +text +head +parent +support +image +music +book +up | ... | 295 | 18% |
| 2 | +brande +pay +producte +device +store +play +branda +version free +tablet +display +credit +good +price +fact | ... | 401 | 24% |
| 3 | 'credit card' +card +credit us american free +download +buy +double +address +brande +memory bought +account +store | ... | 26 | 2% |
| 4 | +flashcontent +javascript +onclick +alert +html +preset +scale +code +document +parent +length +catch +style +versionafox +log | ... | 5 | 0% |
| 5 | +flash +support disappointed +productf1 +buy +producta +download +work +versiona bought +new +up +none +old +branda | ... | 420 | 25% |
| 6 | 'battery life' +battery +charge +life battery hours charging +charger +power +read +time reading +old +the +great | ... | 116 | 7% |
| 7 | +easy +great +love +producta +versiona books hd movies games reading +recommend +read +sound +music +happy | ... | 411 | 25% |

**Figure 7. Cluster contents from Text Cluster output**

Text Cluster node can be used to find out how the topics are clustered and we can find out the hot topics of discussion in these clusters as shown in the figure above.

## PREDICTION OF STAR RATING

We have partitioned the data for training and validation purposes in a 70:30 ratio using a data. The data partition node is connected to the Text Topic node. The partitioned data is fed to a Decision Tree node to predict the star rating a user is likely to give based on his textual review. The variables used for prediction are the individual topics and the tree is pruned automatically. The decision tree has a predictive accuracy of 87%.



```
Variable Importance

Obs       NAME                       LABEL                     NRULES   IMPORTANCE   VIMPORTANCE   RATIO


  1   TextTopic_raw7    +return,+device,+buy,+amazon,+ipad            1      1.00000      1.00000     1.00000
  2   TextTopic_raw17   +buy,+nexus,+annoy,+disappoint,+don           1      0.94744      0.79311     0.83711
  3   TextTopic_raw19   +disappoint,+amazon,+find,+ipad,+pay          1      0.75896      0.53738     0.70804
  4   TextTopic_raw8    +card,credit card,+credit,+buy,us             1      0.69994      0.40373     0.57681
  5   TextTopic_raw1    +love,+buy,+amazon,+fire,+device              1      0.46440      0.28098     0.60504
  6   TextTopic_raw12   +support,+browser,+flash,flash,+support       1      0.44538      0.62009     1.39229
  7   TextTopic_raw11   +pay,+remove,+amazon,+advertisement,+device   1      0.40866      0.00000     0.00000
```

**Figure 8. Variable importance output from Decision Tree results**

The important predictor variables are the topics shown above. We can infer that the terms shown in each of the topics above are the most important predictors for assignment of star rating for each review. Some of the important terms are *support, call, language, competitor brandf, language, love, buy, great, like* etc.

The figure below represents a partial decision tree. The splitting mechanism occurs based on the important variables which are the topic groups from the Text Topic node.

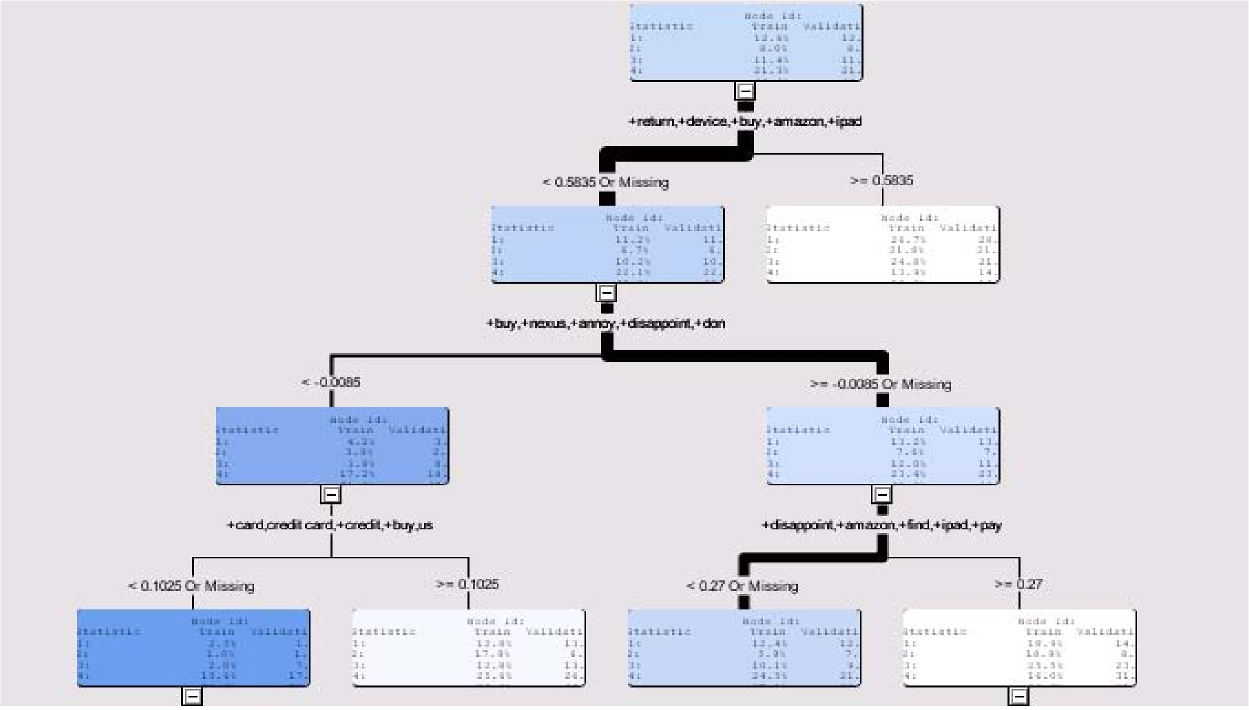

+return,+device,+buy,+amazon,+ipad
< 0.5835 Or Missing
>= 0.5835
+buy,+nexus,+annoy,+disappoint,+don
< -0.0085
>= -0.0085 Or Missing
+card,credit card,+credit,+buy,us
+disappoint,+amazon,+find,+ipad,+pay
< 0.1025 Or Missing
>= 0.1025
< 0.27 Or Missing
>= 0.27

**Figure 9. Decision Tree partial view**

The above pruned decision tree diagram shows the hierarchy of terms used along with their weights used for classifying nodes.

## SENTIMENT ANALYSIS

### DATA PREPARATION

The input SAS data set created from the web crawler output is filtered by classifying observations into positive and negative using the individual star ratings. We have classified reviews with 4 and 5 stars as positive reviews and 1, 2, 3 stars as negative reviews.

Since SAS Sentiment Analysis Studio accepts only text file input, we convert the SAS data sets generated for Text Mining into individual text files using the below SAS Macro code.

```
/*WORK1: START******/
dm "log; clear; output; clear;";
%macro createtxt(numvars);
%do i=1 %to &numvars;
   data _null_;
   obsnum=&i;
   length text $2500.;
      set source POINT=obsnum;
      file &&name&i;
      text=compbl(strip(tranwrd(comment,"",'')));
      put text;
      STOP;
   run;
%end;
%mend;
```

```
data _null_;
    set source nobs=count;
call symput("name"||left(_n_)," 'C:\DestinationFolder"||Id||".txt' ");
  if _n_=1 then call symput("numvars", trim(left(put(count, best.))));
run;
%createtxt(&numvars);

/*WORK1:END*/
```

## MODEL BUILDING

A corpus is created in SAS Sentiment Studio and the positive and negative reviews are fed into the corpus. Since the sample contains around 1100 positive reviews and around 600 negative reviews, we have taken around 600 positive reviews for training and validation purpose and saved the remaining 500 positive reviews for testing purposes.

We now develop a Statistical training model with 70% of the data for the training and 30% for validation. Best Mode was chosen before training the statistical model. For model building purposes we have used Model based Sentiment Analysis approach. To further refine this analysis, we can adopt rule based approach to better define rules based on the product line and business knowledge and finally do a hybrid model which could provide the best results.

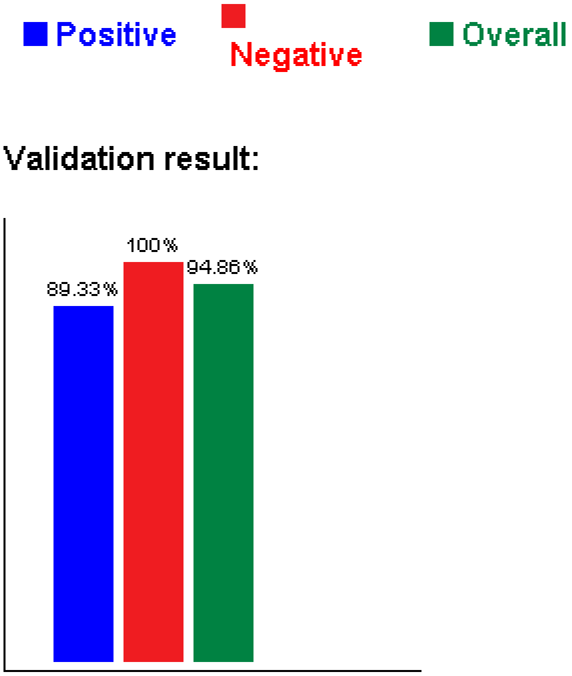


Figure 10. Model Results from SAS Sentiment Miner

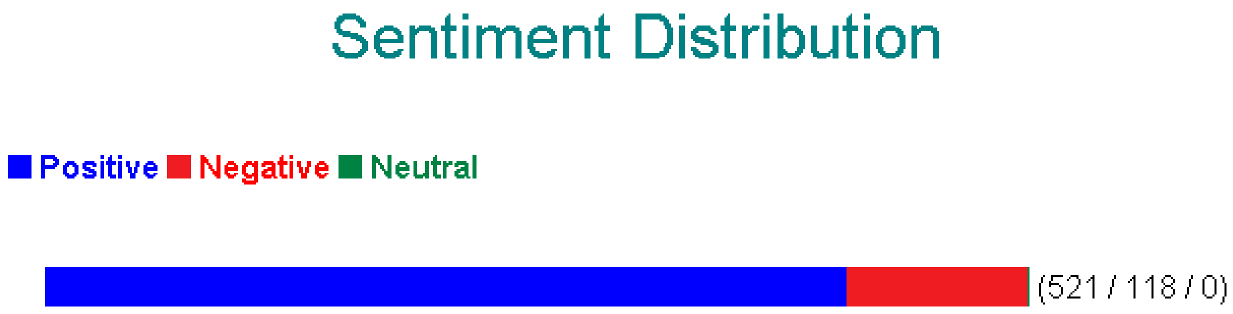
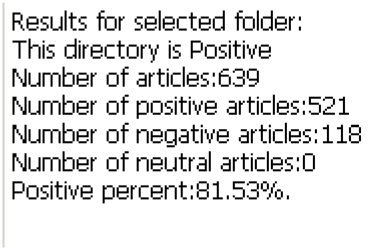


**Figure 11. Sentiment Distribution Results from SAS Sentiment Miner**

As shown above, the model predicts that there is 81.53% positive comments in the test data which is classified as positive. This shows that the model is 81.53% accurate.

## SENTIMENT MODEL RESULTS

The Sentiment Analysis model has an overall accuracy of 94.86% in the validation data. It has a positive accuracy of 89.33% in the validation data.



```
Results for selected folder:
This directory is Positive
Number of articles:639
Number of positive articles:521
Number of negative articles:118
Number of neutral articles:0
Positive percent:81.53%.
```

The same model when applied on testing data reveals that the 81.53% of the positive sentiments are classified as positive. So we can conclude that the model has an accuracy of 81.53% in the test data.

## Appendix:

The below code written in C#.NET is the reusable web crawler tool used to prepare excel dataset from Amazon.com.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Text.RegularExpressions;
using System.Text;
using System.IO;
using System.Xml;
using System.Collections;
using Microsoft.Office.Interop.Excel;
using System.Globalization;
using System.Reflection;

namespace ConsoleApplication2
{
    class Program
    {
        string FinalReview = string.Empty;
        int iter = 1;
        int pageno;
        int i = 2;
        ArrayList[] alTotal = new ArrayList[6];
        ArrayList alReviewArr = new ArrayList();
        ArrayList alStarRating = new ArrayList();
        ArrayList alUseful = new ArrayList();
        ArrayList alUseless = new ArrayList();
        ArrayList alTitle = new ArrayList();
        ArrayList alCommentCnt = new ArrayList();
        string extractedContent = "";
        string url = "http://www.Amazon.com/Kindle-Fire-HD/product-
reviews/B0083PWAPW/ref=pr_all_summary_cm_cr_acr_txt?ie=UTF8&showViewpoints=1";
        static void Main(string[] args)
        {
            //ebay eb = new ebay();
            //eb.ebayCall();
            Program prg = new Program();
            prg.scrapeWebsite(prg.url);
            for(int j=0; j<prg.pageno; j++)
            prg.alTotal = prg.iterWebsite(prg.url);
            prg.writetoExcel(prg.alTotal);

        }
        public void writetoExcel(ArrayList[] Datas)
        {
            Microsoft.Office.Interop.Excel._Application oExcel = new
Microsoft.Office.Interop.Excel.Application();
            GC.Collect();
            Microsoft.Office.Interop.Excel._Workbook oWorkBook;
            Microsoft.Office.Interop.Excel._Worksheet oSheet;
            oExcel = new Microsoft.Office.Interop.Excel.Application();

            oExcel.Visible = false;
            oWorkBook = 
(Microsoft.Office.Interop.Excel._Workbook)(oExcel.Workbooks.Add(Missing.Value));
            oSheet = (Microsoft.Office.Interop.Excel._Worksheet)oWorkBook.ActiveSheet;
            for (int m = 0; m < Datas.Length; m++)
            {
                for (int j = 0; j < Datas[m].Count; j++)
                {
                    for (int k = m; k < m+1; k++)
                    {
                        oSheet.Cells[j + 1, k + 1] = Datas[m][j];
                    }
                }
            }

            string strFile = "Kindle Fire HD 7”FireReviews.xls";
            oWorkBook.SaveAs("C:\\Users\\srihari\\Desktop\\" + strFile, 
Microsoft.Office.Interop.Excel.XlFileFormat.xlWorkbookNormal, null, null, false, false, 
Microsoft.Office.Interop.Excel.XlSaveAsAccessMode.xlShared, false, false, null, null, 
null);
            oWorkBook.Close(null, null, null);
            oExcel.Workbooks.Close();
            oExcel.Quit();
            System.Runtime.InteropServices.Marshal.ReleaseComObject(oExcel);
            System.Runtime.InteropServices.Marshal.ReleaseComObject(oSheet);
            System.Runtime.InteropServices.Marshal.ReleaseComObject(oWorkBook);
            oSheet = null;
            oWorkBook = null;
            oExcel = null;
            GC.Collect();
        }

        public ArrayList[] iterWebsite(string url)
        {
            if (i <= pageno)
            {
                url = url.Replace(url.Substring(url.IndexOf("?ie")), 
"?ie=UTF8&showViewpoints=0&pageNumber=" + i);
                iter++;
                i++;
                alTotal = scrapeWebsite(url);
            }
            return alTotal;
        }

        public ArrayList[] scrapeWebsite(string url)
        {
            int pat = 0;
            string pagetxt;
            int strt = 0;
            int a, b;
            string nodesep, commentcnt;
            string starrating;
            string useful, useless, title;
            string[] col2, col3, col4, col5;
            WebClient wc = new WebClient();
            wc.Headers.Add("HTTP_USER_AGENT", "Web-Scraper-Agent (your-custom-user-agent-
here)");
            try
            {
                // Download the web page content from the URL
                extractedContent = wc.DownloadString(url);
                //Remove CSS styles, if any found
                extractedContent = Regex.Replace(extractedContent, "<style(.|
)*?>*</style>", "");
                //Remove script blocks
                extractedContent = Regex.Replace(extractedContent, "<script(.|
)*?>*</script>", "");
                // Remove all images
                extractedContent = Regex.Replace(extractedContent, "<img(.| )*?/>", "");
                // Remove all HTML tags, leaving on the text inside.
                extractedContent = Regex.Replace(extractedContent, "<(.| )*?>", "");
                // Remove all extra spaces, tabs and repeated line-breaks
                extractedContent = Regex.Replace(extractedContent, "(x09)?", "");
                extractedContent = Regex.Replace(extractedContent, "(x20){2,}", " ");
                extractedContent = Regex.Replace(extractedContent, "(x0Dx0A)+", " ");
                extractedContent = extractedContent.Replace(":nbsp;", "");

                    extractedContent =
extractedContent.Substring(extractedContent.IndexOf("Previous |"),
(extractedContent.LastIndexOf("Help other") - extractedContent.IndexOf("Previous |")));
                    strt = extractedContent.IndexOf("Previous |");

                if (iter < 2)
                {
                    pagetxt = extractedContent.Substring(strt, 30);
                    //extractedContent.LastIndexOf('|')
                    if (pagetxt.IndexOf(';') > 0)
                    {
                        pat = pagetxt.IndexOf(';');
                        pagetxt = pagetxt.Substring((pat + 1), (pagetxt.Length - pat -
1));

                        pat = pagetxt.LastIndexOf('|');
                        if (pagetxt.LastIndexOf('|') > 0)
                            pagetxt = pagetxt.Substring((pat - 2), 2);
                    }
                    else
                    {
                        pat = pagetxt.LastIndexOf('|');
                        pagetxt = pagetxt.Substring((pat - 2), 2);
                    }
                    pageno = Convert.ToInt16(pagetxt);
                }
                string[] c = Regex.Split(extractedContent, "This review is from");
                foreach (string d in c)
                {
                        col2 = Regex.Split(d, "people found the following review");
                        if (d != c[c.Length - 1])
                        {
                            if (d.IndexOf("people found the following review") != -1)
                            {
                                col5 =
Regex.Split((col2[0].Substring((col2[0].TrimEnd().Length - 15), 15).Trim()), " of ");
                                useful = col5[0];
                                alUseful.Add((string)useful);
                                useless = 
Convert.ToString(Convert.ToInt32(int.Parse(col5[1], NumberStyles.AllowThousands)) - 
Convert.ToInt32(int.Parse(col5[0], NumberStyles.AllowThousands)));
                                alUseless.Add((string)useless);
                                col3 = Regex.Split(col2[1], " out of 5 stars");
                            }
                            else
                            {
                                col3 = Regex.Split(col2[0], " out of 5 stars");
                                useful = useless = "0";
                                alUseful.Add((string)useful);
                                alUseless.Add((string)useless);
                            }
                            starrating = col3[0].Substring((col3[0].Length - 3), 3);
                            alStarRating.Add((string)starrating);
                            title = col3[1].Substring(0, col3[1].IndexOf(',')).Trim();
                            alTitle.Add((string)title);
                        }

                    if (d != c[0])
                    {
                        if (col2[0].IndexOf("PermalinkComment&nbsp;Comments") != -1)
                        {
                            col4 = Regex.Split(col2[0], 
"PermalinkComment&nbsp;Comments");
                            commentcnt = col4[1].Trim().Substring(1, 
(col4[1].TrimStart().IndexOf(')') - 1));
                        }
                        else
                            commentcnt = "0";
                        alCommentCnt.Add((string)commentcnt);
                        a = d.IndexOf(":");
                        if (d.IndexOf("Help other customers") != -1)
                        {
                            b = d.IndexOf("Help other customers", a);
                            nodesep = d.Substring(a, (b - a));
                        }
                        else
                            nodesep = d.Substring(a);
                        alReviewArr.Add((string)nodesep);
                    }
                }
            }
            catch (Exception e)
            {
                alReviewArr.Add("Error on downloading: " + e.Message);
            }
            alTotal[0] = alReviewArr;
            alTotal[1] = alStarRating;
            alTotal[2] = alUseful;
            alTotal[3] = alUseless;
            alTotal[4] = alTitle;
            alTotal[5] = alCommentCnt;

            return alTotal;
        }

    }
```

**CONTACT:**

Your comments and questions are valued and encouraged. Contact the authors

at:

Name: Srihari Nagarajan
Enterprise: SAS Institute
Address: 100 SAS Campus Drive, Building T, Cary, NC
Work Phone: (405)-612-9811
E-mail: srihari.nagarajan@sas.com

Srihari Nagarajan is a Senior Associate Analytical Consultant at SAS Institute. He holds an MS degree in MIS from Oklahoma State University. He has three and half years of professional experience as Assistant Systems Engineer at Tata Consultancy Services Ltd. He is SAS® Certified Base Programmer for SAS® 9 and Certified Predictive Modeler Using SAS® Enterprise Miner TM 6.1.

Name: Hari Harasudhan
Enterprise: Kavi Associates
Address: 1250 S. Grove Ave, Suite 300,
City, State ZIP: Barrington, IL 60010
Work Phone: (720)–351-2518
E-mail:hari.duraidhayalu@okstate.edu

Hari Harasudhan is a Business Consultant at Kavi Associates and holds a MS degree in MIS from Oklahoma State University with specialization in Data Mining and Business Intelligence Tools. Has three years of professional experience as System Engineer at Infosys Technologies Ltd. He is SAS® Certified Base Programmer for SAS® 9 and Certified Predictive Modeler Using SAS® Enterprise Miner TM 6.1.

Name: Dr. Goutam Chakraborty
Enterprise: Oklahoma State University
Address: Department of Marketing, Oklahoma State University
City, State ZIP: Stillwater, OK - 74074
Work Phone: (405)744-7644
E-mail: goutam.chakraborty@okstate.edu

Dr. Goutam Chakraborty is a professor of marketing and founder of SAS® and OSU data mining certificate program at Oklahoma State University. He has published in many journals such as *Journal of Interactive Marketing*, *Journal of Advertising Research*, *Journal of Advertising*, *Journal of Business Research*, etc. He chaired the national conference for direct marketing educators in 2004 and 2005 and co-chaired the M2007 data mining conference. He is also a Business Knowledge Series instructor for SAS®.

## TRADEMARKS