ITCS 6156: Machine Learning
Assignment 1: Decision Tree Classification

Submitted by: Ashwin Venkatesh Prabhu
UNCC ID: 800960400
Email: avenka11@uncc.edu

Collaborated with: Febin Zachariah
UNCC ID: 800961027
Email: fzhacari@uncc.edu

Optical Recognition of handwritten digits dataset

Overview:

1) The dataset for this assignment is available at http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits
2) Number of attributes: Optical recognition of handwritten digits' data set has 65 attributes. 32x32 bitmaps are divided into non-overlapping blocks of 4x4 and the number of on pixels are counted in each block. This generates an input matrix of 8x8 where each element is an integer in the range of 0..16. All the attributes are integers in the range of 0..16. The last attribute is a class code 0..9.
3) It has handwritten digits from 43 people. 30 of them contributed towards training dataset and 13 contributed towards test dataset
4) The data available is already preprocessed
5) Number of observations: Number of observations in the training data set is 3823. Number of observations in the testing data set is 1797.
6) Mean and standard deviation of each attribute (Note: since the datasets do not have a header, the attributes are named from V1 to V65. V1 to V64 are feature attributes and V65 is the class attribute)

Attributes	Mean for train data	Std Deviation for train data	Mean for test data	Std deviation for test data
V1	0	0	0	0
V2	0.3013	0.8669861	0.303839733	0.907192095
V3	5.482	4.631601	5.204785754	4.75482634
V4	11.81	4.259811	11.83583751	4.248841848
V5	11.45	4.537556	11.84808013	4.287388007
V6	5.505	5.61306	5.781858653	5.666417727
V7	1.387	3.371444	1.362270451	3.325775186
V8	0.1423	1.051598	0.129660545	1.037382857
V9	0.002093	0.08857152	0.00556483	0.094221555
V10	1.961	3.052353	1.993878687	3.196160408
V11	10.58	5.435481	10.38230384	5.421455626
V12	11.72	4.01216	11.97941013	3.977542622
V13	10.62	4.788136	10.27935448	4.78268057
V14	8.296	5.935551	8.175848637	6.052960026
V15	2.2	4.062178	1.846410684	3.586320936
V16	0.152	0.9887783	0.107957707	0.827915045
V17	0.00497	0.1198569	0.002782415	0.062368293
V18	2.596	3.454065	2.601558152	3.576301267
V19	9.581	5.886126	9.903171953	5.690766949
V20	6.735	5.918303	6.992765721	5.802661721
V21	7.187	6.142687	7.097941013	6.175728516
V22	8.048	6.291498	7.806343907	6.197321772
V23	2.046	3.58174	1.78853645	3.259869702

V24	0.04918	0.435462	0.050083472	0.438597486
V25	0.001046	0.03233384	0.001112966	0.033351857
V26	2.336	3.085915	2.469671675	3.146532463
V27	9.239	6.128091	9.091263216	6.192037816
V28	9.134	5.902591	8.821368948	5.882936493
V29	9.673	6.282903	9.927100723	6.152092832
V30	7.868	6.002377	7.55147468	5.872555578
V31	2.34	3.62474	2.317751809	3.686455957
V32	0.003139	0.06462534	0.002225932	0.047140364
V33	0.001308	0.0361456	0	0
V34	2.043	3.211658	2.339454647	3.480372317
V35	7.659	6.259573	7.66722315	6.324687349
V36	9.238	6.190196	9.071786311	6.268391185
V37	10.35	5.920125	10.3016138	5.933490192
V38	9.2	5.879345	8.744017807	5.870647617
V39	2.913	3.486267	2.909293267	3.53728272
V40	0	0	0	0
V41	0.02746	0.3161931	0.008903728	0.145185429
V42	1.406	2.93420595	1.583750696	2.981816228
V43	6.457	6.50537325	6.881469115	6.537954672
V44	7.187	6.46906057	7.228158041	6.441377552
V45	7.922	6.31636836	7.672231497	6.259511442
V46	8.675	5.80592397	8.236505287	5.695526575
V47	3.51	4.36913146	3.456316082	4.330951228
V48	0.01988	0.2136677	0.027267668	0.307355887
V49	0.01779	0.26911025	0.007234279	0.204223166
V50	0.82	2.00901847	0.704507513	1.746152865
V51	7.869	5.66663614	7.506956038	5.64449606
V52	9.886	5.14156087	9.539232053	5.2269477
V53	9.765	5.31497679	9.416249304	5.302048472
V54	9.283	5.94088711	8.758486366	6.031154412
V55	3.744	4.90165704	3.725097385	4.919406035
V56	0.1483	0.76776137	0.206455203	0.984400918
V57	0.0002616	0.01617327	0.000556483	0.023589892
V58	0.283	0.92804553	0.27935448	0.934301798
V59	5.856	4.980012	5.557595993	5.10301937
V60	11.94	4.33450762	12.08903728	4.37469401
V61	11.46	4.99193441	11.80912632	4.933947353
V62	6.7	5.77581501	6.764051196	5.900622712

V63	2.106	4.02826574	2.067890929	4.090547887
V64	0.2022	1.15069446	0.364496383	1.860121722
V65	4.497	2.86983086	4.49081803	2.865303781

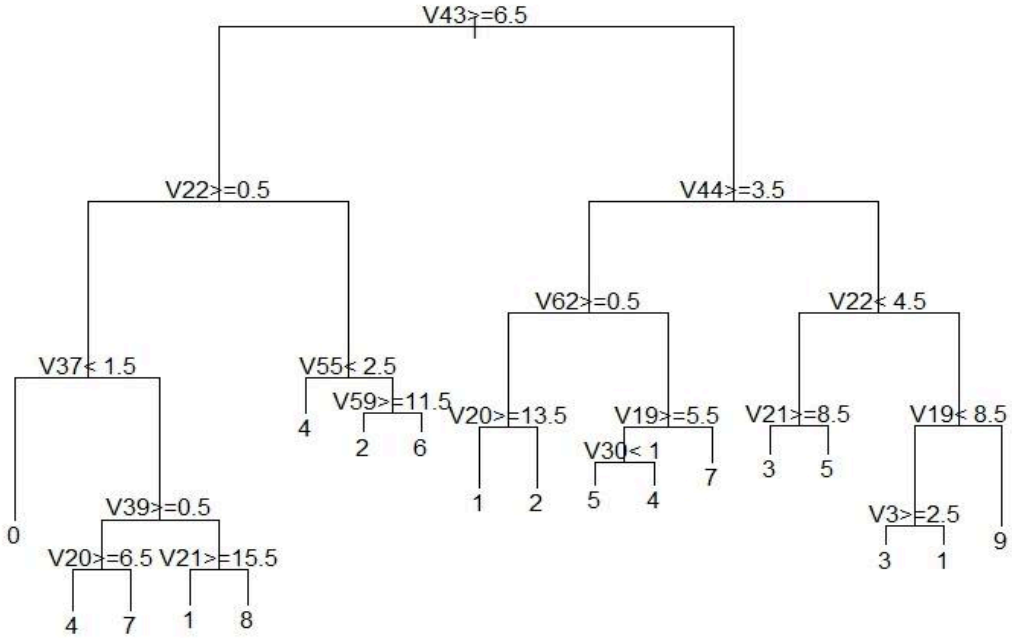
7) Distribution of different classes in each of the dataset:

Value of the Class attribute	Distribution in training set	Distribution in test set
0	376	178
1	389	182
2	380	177
3	389	183
4	387	181
5	376	182
6	377	181
7	387	179
8	380	174
9	382	180

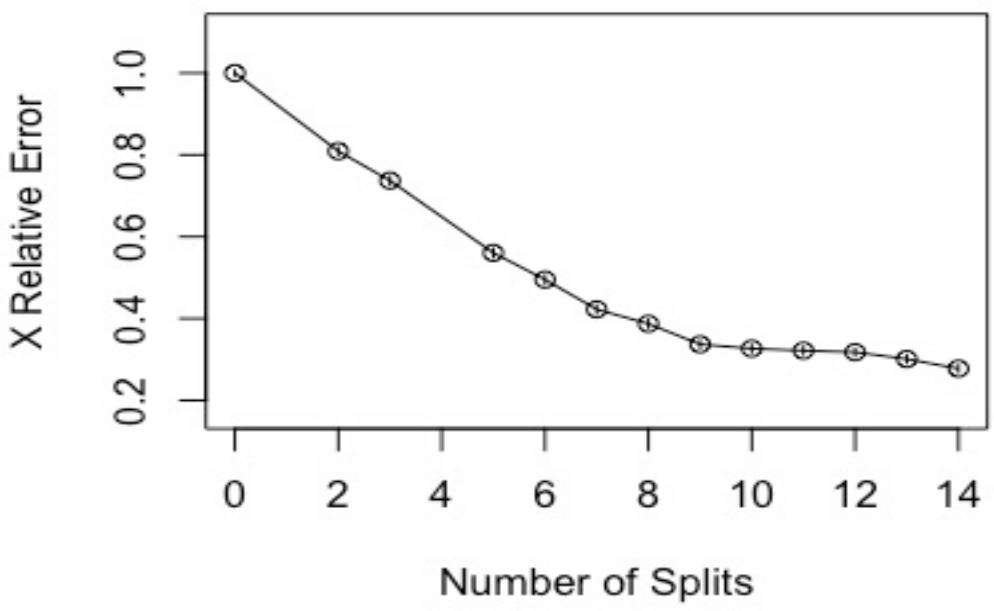
Implementation Details:

1) The program is written in R programming language.
2) Program uses "rpart" library to create a decision tree. [The *rpart* algorithm works by splitting the dataset recursively, which means that the subsets that arise from a split are further split until a predetermined termination criterion is reached. At each step, the split is made based on the independent variable that results in the *largest possible reduction in heterogeneity of the dependent (predicted) variable*.]

3) The datasets for training as well as test do not have a header, so the attributes are name V1 to V65. V1 to V64 represent feature attributes and V65 is for class attribute.
4) Main steps explaining the program:
    a. Train data available in “optdigits_raining.csv” is loaded into “train_data” variable
    b. Train data is sample for cross validation. 70% of the data is used for training and 30% of the data is used for cross validation
    c. Decision tree is created using rpart method. Formula used for creating the decision tree is “V65 ~ V1+V2+V3+…..+V64”. Data used will be 70% of the train data. Information gain is used for splitting
    d. Once the decision tree is generated, prune method is used to prune the data
    e. After the decision tree is pruned, the pruned tree is used on the 30% of the train data kept aside for cross validation, for prediction. predict method is used for this purpose.
    f. predict method will predict the class attribute for the 30% of the data. On checking for error in prediction, it is noticed that approximately 75% of the prediction are correct.
    g. Now the test data is loaded. predict method is used on the test data using the pruned decision tree. On checking for error in prediction in this test data, it is noticed that approximately 72% of the prediction are correct.
5) Decision tree drawn using rpart method is illustrated as below:

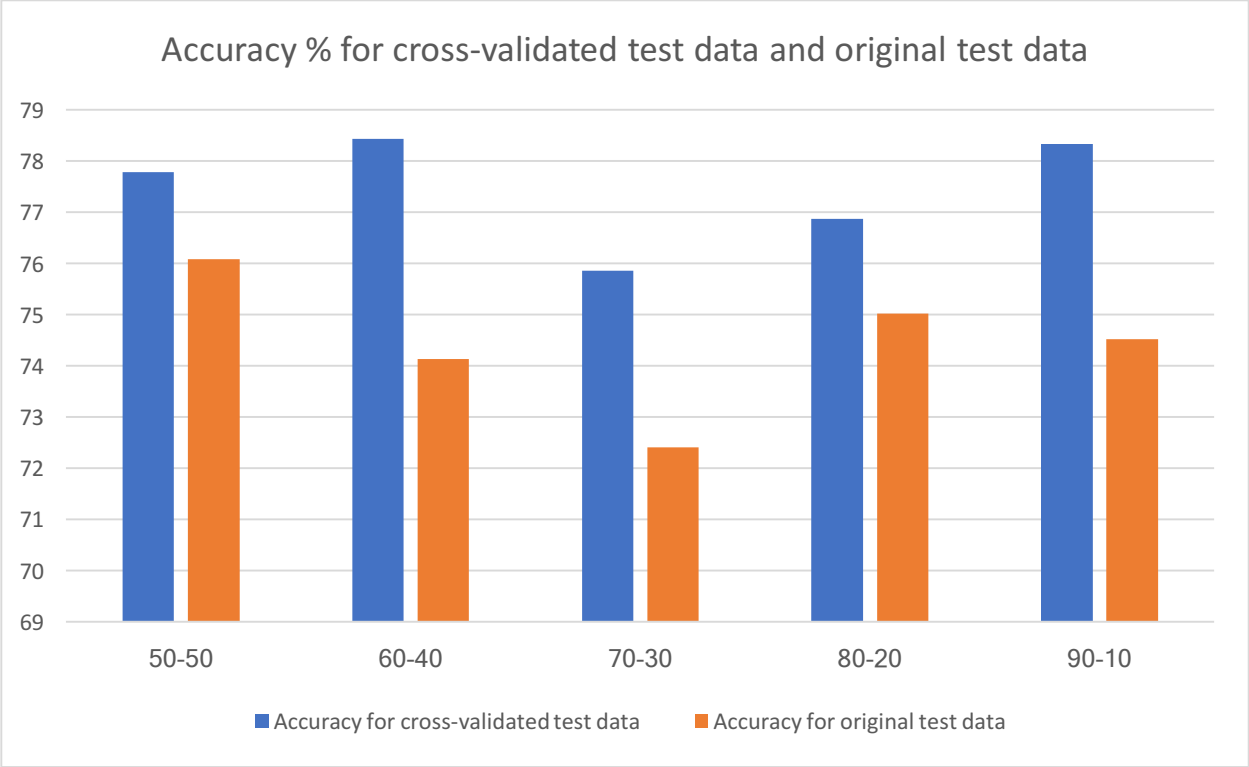


6) The below is plotted for the decision tree with “number of splits” against “relative error”.



7) The graph shows that the first 2 splits offers the most information, and not much information is gained after 3 splits. This graph suggests that the tree should be pruned to include only 2 to 3 splits.
8) Conclusion: After training the data with 5 different splits for cross validation, the accuracy calculated is as follows:

Split	Accuracy for cross-validated test data	Accuracy for original test data
50-50	77.77196653	76.07122983
60-40	78.43137255	74.12353923
70-30	75.85004359	72.39844185
80-20	76.8627451	75.01391208
90-10	78.3289817	74.51307735



50-50 split is not considered a good split since only 50% of the data is available for training. Out of the other splits, we can see that 80-20 split has the highest accuracy on the original data with over 75% of the prediction being accurate.

Amazon Reviews sentiment analysis

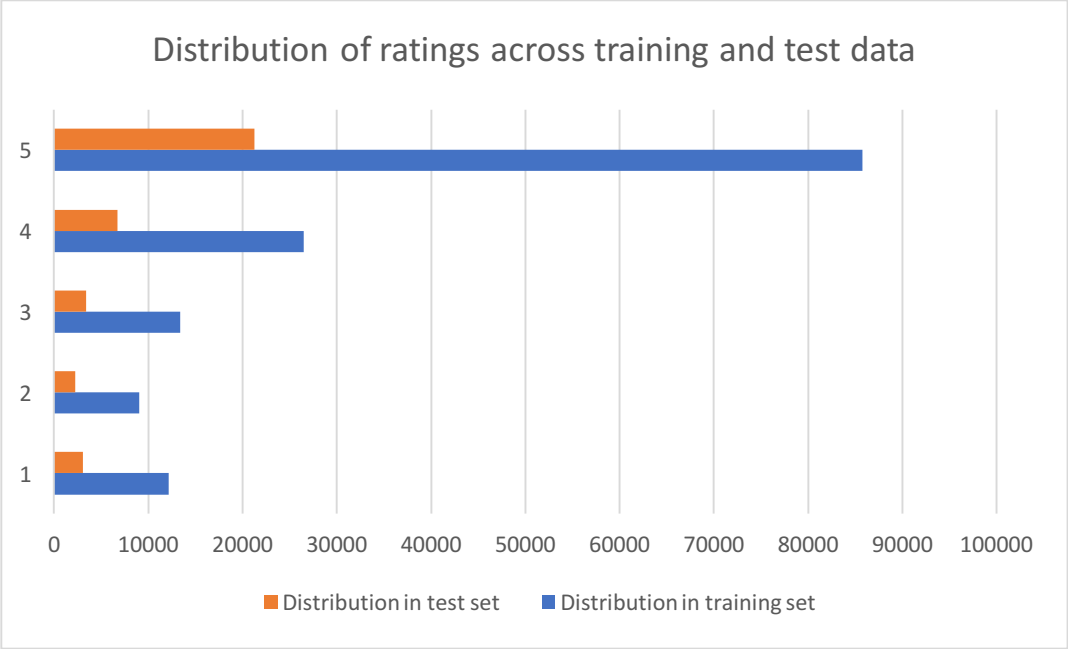
Overview:

1) The dataset for this assignment is available at https://snap.stanford.edu/data/web-Amazon.html
2) Number of attributes: Amazon Reviews data set has 3 attributes. Namely, name (specifying the name of the product bought), review (specifying the review written by the user for the product), and rating (specifying the rating given by the user for the product)
3) Number of observations: 146825 for training data. 36708 for test data.
4) Mean and standard deviation of each attribute:

	Mean for training data	Standard deviation for training data	Mean for test data	Standard deviation for test data
rating	4.121799	1.284966	4.115046	1.285226

5) Distribution of different classes in each of the dataset:

Rating	Distribution across training data	Distribution across original test data
1	12146	3037
2	9040	2270
3	13364	3415
4	26509	6696
5	85765	21289

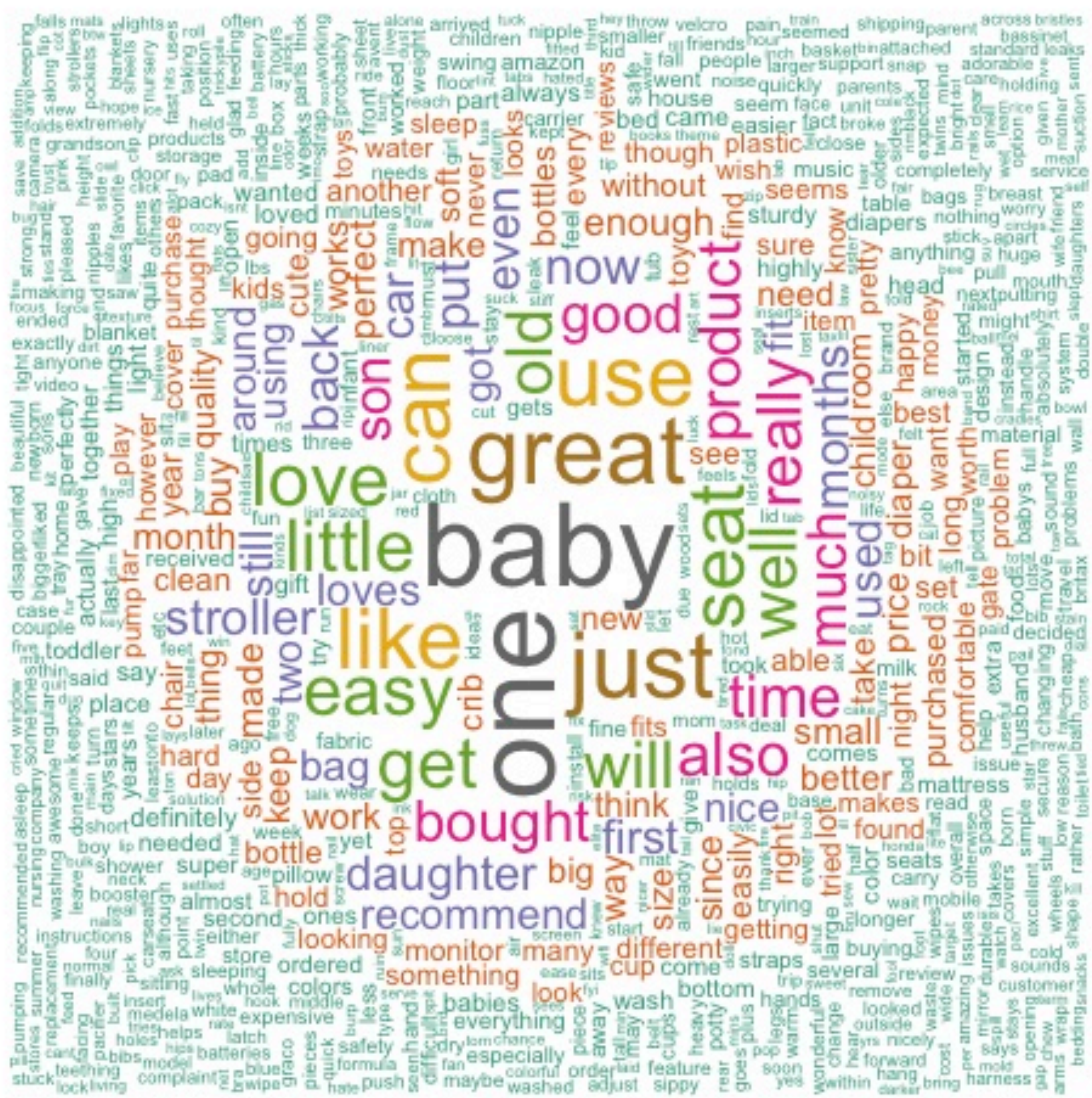


Implementation Details:

1) The program is written in R programming language.
2) Program uses “rpart”, “plyr”, “stringr” library to create a decision tree.
3) The datasets have 3 headers, namely, “name”, “review”, “rating”. Class attribute for this dataset is “rating”.
4) A set of positive words in “positive_words.txt” file is loaded into positive_words variable while running the program.
5) A set of negative words in “negative_words.txt” file is loaded into negative_words variable while running the program.
6) Main steps explaining the program:
    a. Once the positive and negative words are loaded into their respective variables, “score.sentiment” method is called to evaluate the sentiment score for each review in the training dataset.
    b. Score.sentiment method assigns a sentiment score to each review. A negative sentiment score means there are more number of negative words in the review than the positive words. A positive sentiment score means there are more number of positive words in the review than the negative words. If the sentiment score value is 0, then the review is neutral. An important point to note is that, a considerably good review with a rating 3 stars can have a negative sentiment score, if the review contains more number of negative words. Hence, the correlation between rating and sentiment score is loosely defined.
    c. Train data is sampled for cross validation. 70% of the data is used for training and 30% of the data is used for cross validation
    d. Decision tree is created using rpart method. Formula used for creating the decision tree is “rating ~ score + pos_count + neg_count”. Data used will be 70% of the train data. Information gain is used for splitting.
    e. Once the decision tree is generated, prune method is used to prune the data

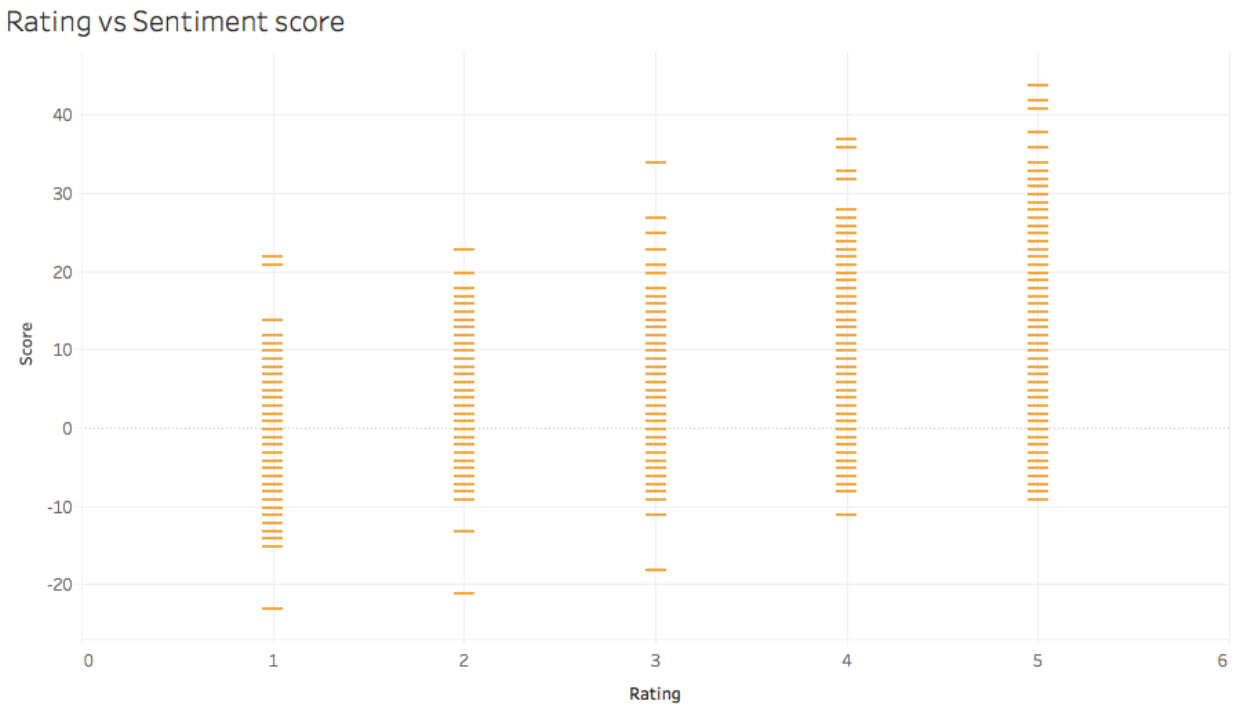
f. After the decision tree is pruned, the pruned tree is used on the 30% of the train data kept aside for cross validation, for prediction. predict method is used for this purpose.
g. predict method will predict the rating for the 30% of the data. On checking for error in prediction, it is noticed that approximately 60% of the prediction are correct.
h. Now the test data is loaded. predict method is used on the test data using the pruned decision tree. On checking for error in prediction in this test data, it is noticed that approximately 59% of the prediction are correct.

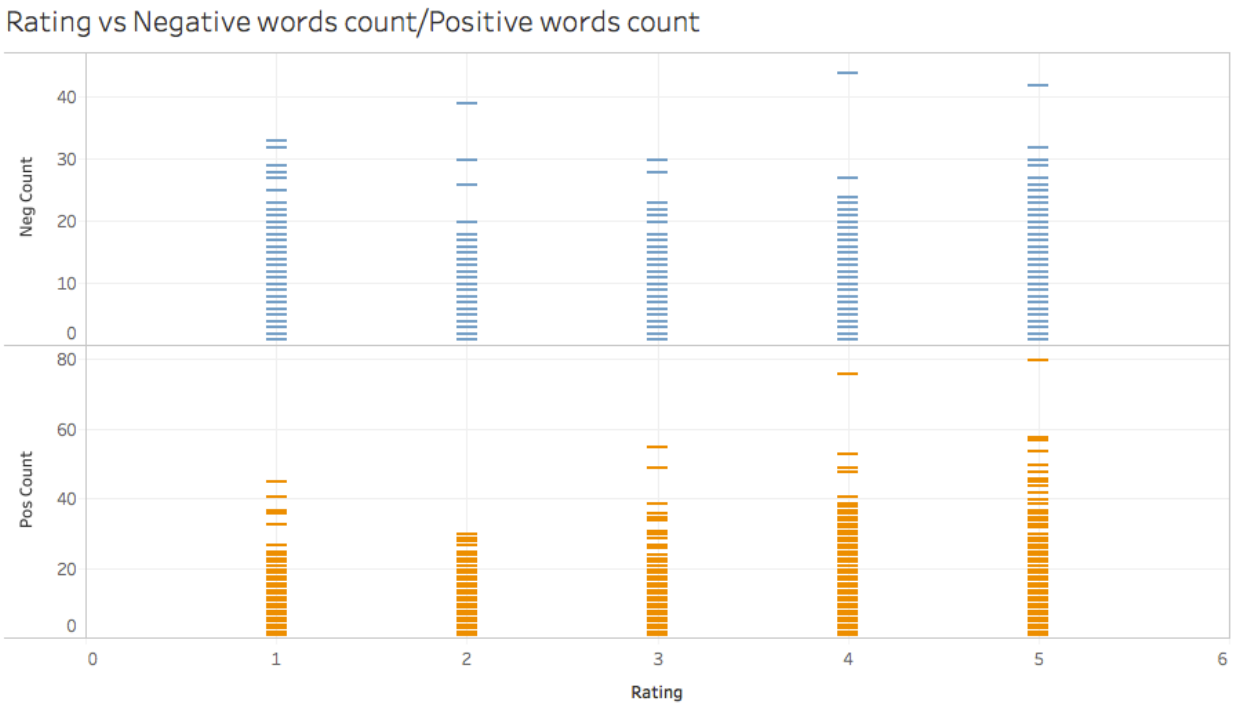
7) Wordcloud displaying the frequently occurring words in reviews in the current dataset



8) Conclusion: Sentiment analysis is a field of study that analyses people's sentiments, attitudes and emotions towards certain objects or entities. Online reviews from amazon's baby product dataset is used for this study. The idea here is to assign a sentiment score to each review and conduct an analysis based on the score.

The below graphs show the relationship between 1) rating vs sentiment score 2) rating vs positive word count/negative word count:





From the above graphs, we can see that, even for rating 5, there is a negative word count of 10 or more leading to a negative sentiment score. The same holds for rating 1 with a positive word count of above 10. The lowest sentiment score for a rating 5 is -9, and the highest sentiment

score for rating 1 is 23. So, the conclusion from this analysis is that rating does not give an accurate idea about the sentiment and the tone of the review.

The algorithm used to conduct the sentiment analysis for the amazon review dataset is the decision tree algorithm. The accuracy achieved for this algorithm is approximately 62% with the test dataset. The accuracy needs to be improved to uncover better results and anomalies.

References

1) https://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf
2) http://analyzecore.com/2014/04/28/twitter-sentiment-analysis/
3) https://github.com/iHub/sentiment-analysis-using-R
4) https://cran.r-project.org/web/packages/wordcloud/wordcloud.pdf
5) http://www.statmethods.net/advstats/cart.html
6) https://cran.r-project.org/web/packages/wordcloud/wordcloud.pdf