MACHINE LEARNING

Febin Zachariah – 800961027

SUPERVISED LEARNING-DECISION TREE

Optical Recognition of Handwritten Digits dataset

1) INTRODUCTION

- The dataset for this assignment is taken from **UCI Machine Learning repository**.
- It has handwritten digits from 43 people. 30 contributed for training set and 13 for test data set.
- The data taken for this assignment is already preprocessed.
- 32x32 bitmaps are divided into non-overlapping blocks of 4x4 and the number of on pixels are counted in each block
- It has 64 features each correspond to each pixel of the image of handwritten digit. Each feature is of range **0 to 16 to reduce dimensionality.**
- The values of the output attribute ranges between **0 to 9.**
- Training data consists of **3823 examples** and test data has **1797 examples.**

Columns	Mean for Train Data	Standard Deviation for Train Data	Mean for Test Data	Standard 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

Class Variable Values	Training Data Frequency	Test Data Frequency
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

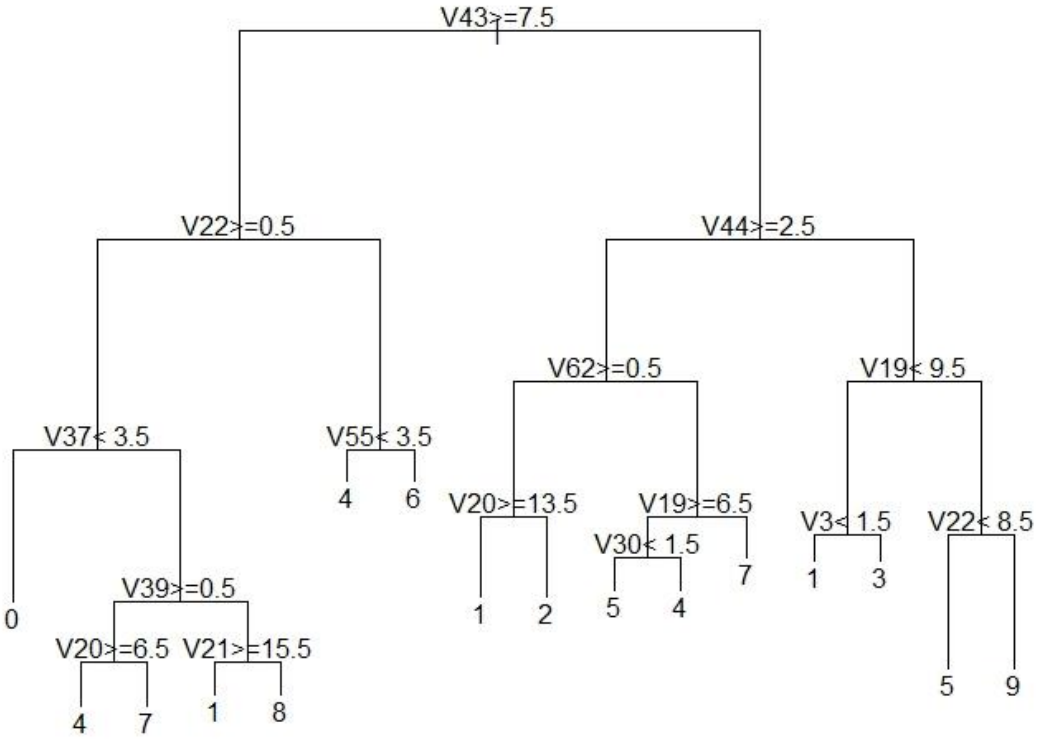
2) Implementation

- For the implementation of decision tree: **R language is used**
- Packages used: **rpart, rattle, rpart.plot, party, RColorBrewer**
- Load the packages by using **library** command in R if it is already installed, otherwise install the libraries by using **install.packages** command.
- After that load the training data by using **read.csv** command.
- Create the formula which shows the dependency of input attributes and output attributes.

  V65 ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 + V11 +
  V12 + V13 + V14 + V15 + V16 + V17 + V18 + V19 + V20 + V21 +
  V22 + V23 + V24 + V25 + V26 + V27 + V28 + V29 + V30 + V31 +
  V32 + V33 + V34 + V35 + V36 + V37 + V38 + V39 + V40 + V41 +
  V42 + V43 + V44 + V45 + V46 + V47 + V48 + V49 + V50 + V51 +
  V52 + V53 + V54 + V55 + V56 + V57 + V58 + V59 + V60 + V61 +
  V62 + V63 + V64
- Divide the training into two sets in the ratio 80:20 for creating a cross validation data sample.
- Apply the formula we created on the bigger sized dataset to create the decision tree by using rpart command.
- Prune the created decision tree by using prune command.
- Now, use the above decision tree to predict the values on cross validation set of data and verify our decision tree.
- Use this decision tree to predict the output for the test data by using the predict command.
- Final output is written into the file named “final_test_result.csv”.

3) Results and Observations

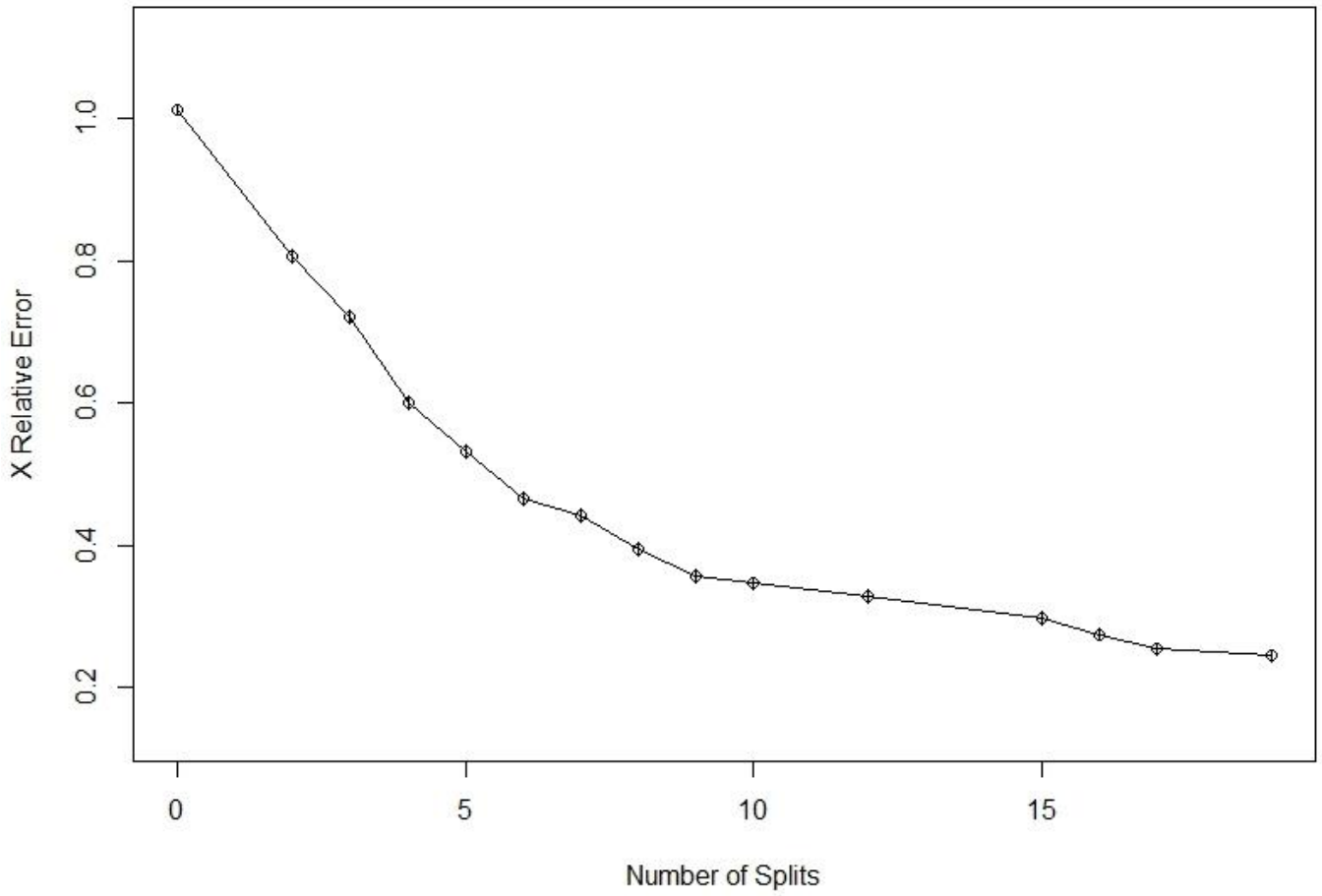
Decision Tree



We have divided the data into 80% for training and 20% for cross-validation from the actual training data. The following table shows the accuracy of the obtained decision tree on cross validation data set and test data.

Dataset	Accuracy
Cross Validation Data	78%
Test Data	76%

- We have also tried with different splitting of the training data and observations and the best result were obtained in the 80:20 split.



The above figure represents the relative error in each split. We can observe in the figure that information gain is higher in the first 3 splits and its getting reduced in the coming splits.

4)Conclusions

- Successfully implemented Decision tree algorithm on R language by using rpart library
- Obtained an accuracy of 75% on the test data.
- Performed analysis on the data by splitting the data in different ratios.

Amazon Reviews Sentiment Analysis Dataset

1) INTRODUCTION

- This dataset consists of Amazon baby product reviews a subset of a larger amazon review collection.
- The dataset is split into training and testing subsets.
- Reviews include product information, ratings, and a plaintext review.
- The values of the output attribute (Ratings) ranges between **1to 5.**
- Training data consists of **146824 observations** and test data has **36707 examples.**

	Mean for Training Data	**Standard Deviation for Training data**	**Mean for Test Data**	**Standard Deviation for Test data**
Rating	**4.12222**	**1.284**	**4.115**	**1.285**

Class Variable Values	Training Data Frequency	Test Data Frequency
1	12146	3037
2	9040	2270
3	13394	3415
4	26509	6696
5	85765	21289

2) Implementation

- For the implementation of decision tree: **R language is used**
- Packages used: **rpart, plyr, stringr**
- Load the packages by using **library** command in R if it is already installed, otherwise install the libraries by using **install.packages** command.
- Load the training data and split into two subsets in the ratio 80%:20% for creating the
- Since **reviews are categorical data**, we have implemented an algorithm to convert reviews into numerical score based on the following calculation.
  Score= Sum (Positive words)-Sum (Negative words)
- After getting the score for each review, we are creating additional columns to store score, number of positive words, Number of negative words.

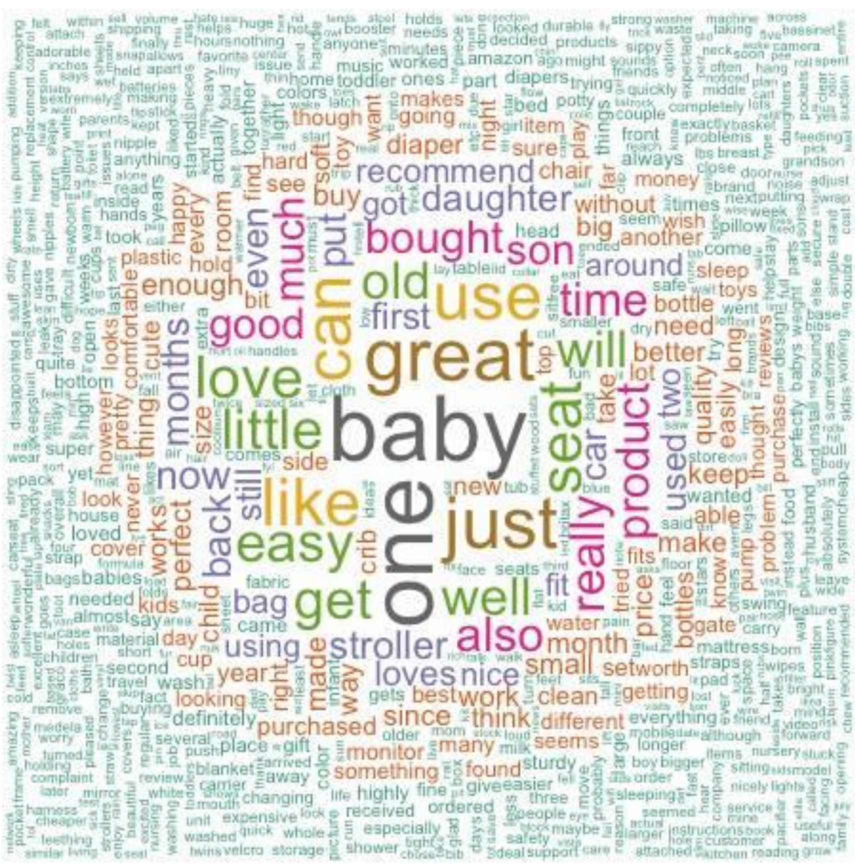
- Create the formula which shows the dependency of input attributes and output attributes. **rating ~ score + pos_count + neg_count.**
- Prune the created decision tree by using prune command.
- Now, use the above decision tree to predict the values on cross validation set of data and verify our decision tree.
- Use this decision tree to predict the output for the test data by using the predict command.
- Final output is written into the file named "amazon_testdata_result.csv".

3) Results and Observations

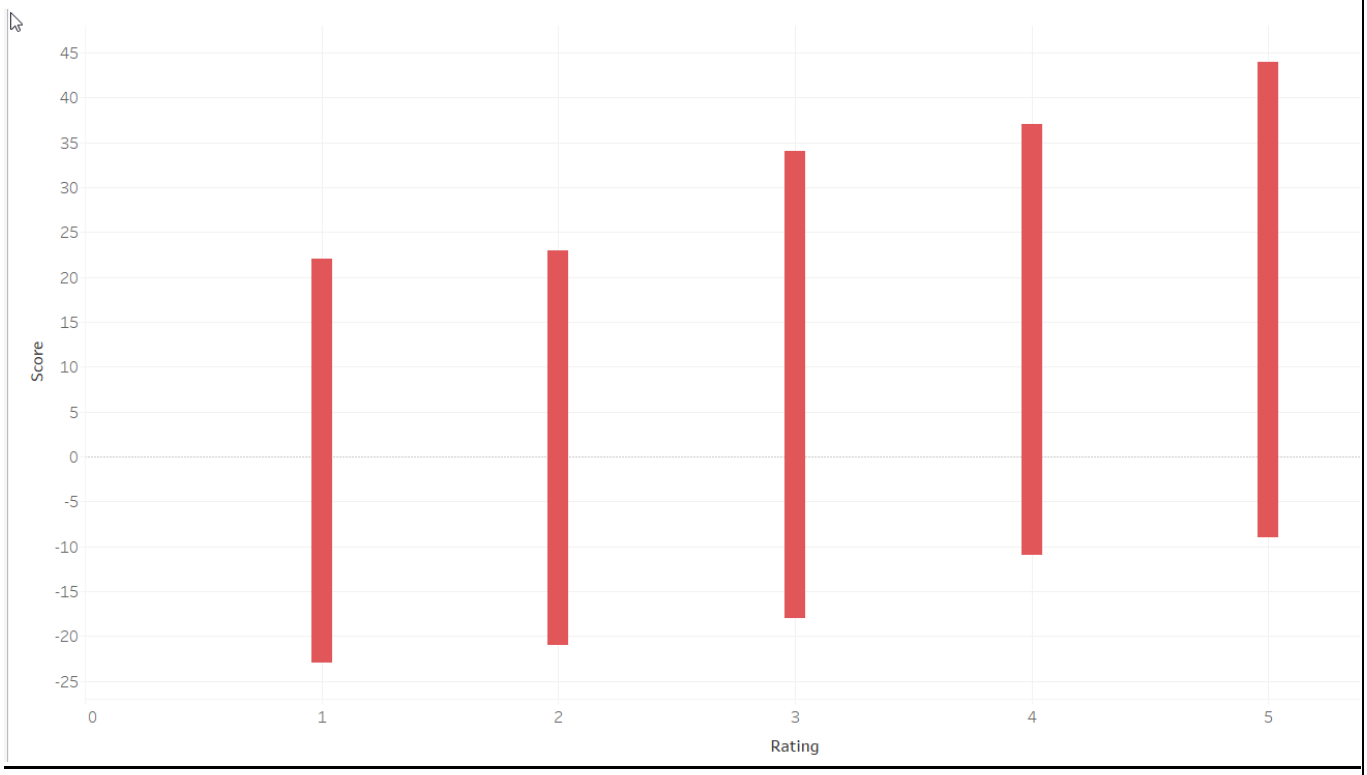
- We have divided the data into 80% for training and 20% for cross-validation from the actual training data. The following table shows the accuracy of the obtained decision tree on cross validation data set and test data for amazon reviews.

Dataset	Accuracy
Cross Validation Data	64%
Test Data	61%

- Word Cloud Obtained:

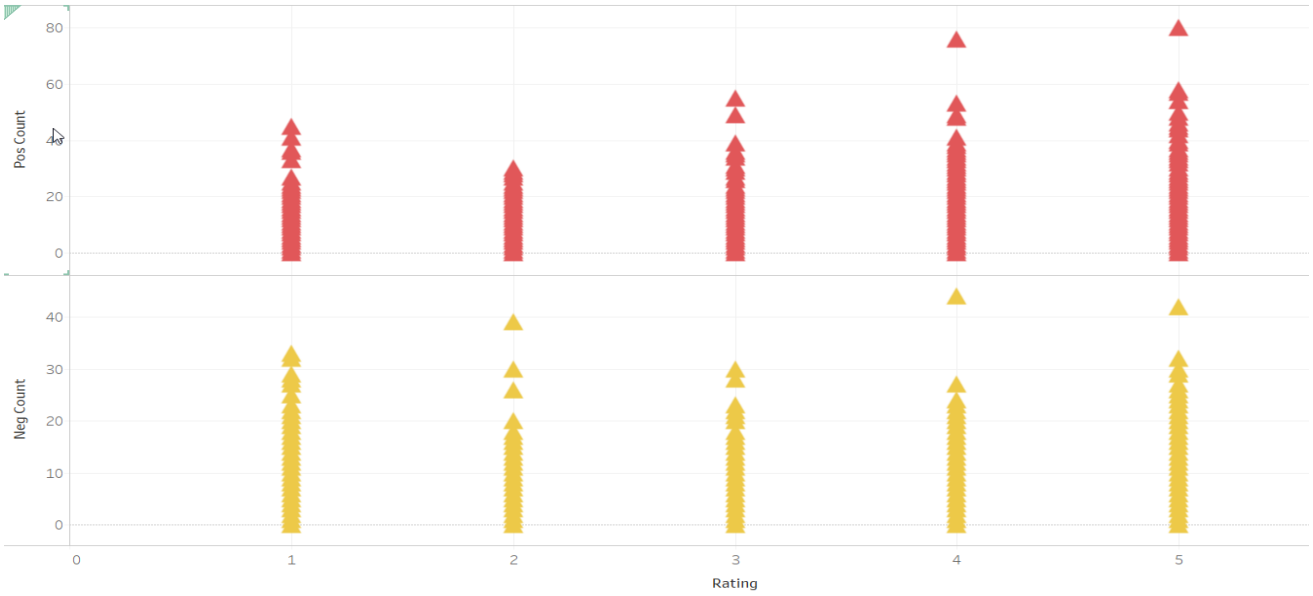


- Rating vs Semantic Score Analysis Graph



In this graph, we can see that even for high negative score the rating is coming as 5 and high positive value for rating 1. These kinds on anomalies are reducing the accuracy of the decision tree algorithm.

- Rating vs Positive Count and Rating vs Negative Count



4) Conclusions:

- From the above graphs, we can see that some five star ratings are having higher negative word count and hence total sentiment score is coming as negative. We can see similar anomalies in the dataset.
- Also, we can see low ratings having higher number of positive words count and resulting in higher sentiment score.
- We are now investigating on it to improve the efficiency of the sentimental analysis of the reviews further.

Collaborated with: Ashwin Venkatesh Prabhu (800960400)

References:

https://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf

http://analyzecore.com/2014/04/28/twitter-sentiment-analysis/

http://onepager.togaware.com/TextMiningO.pdf

http://rstatistics.net/decision-trees-with-r/

https://github.com/iHub/sentiment-analysis-using-R