**5th December 2018** **Course: CS 5593 - DATA MINING**

**PROJECT FINAL REPORT**

**FALL 2018**

**Sentiment Analysis using NLP to Classify Amazon Product Reviews using Supervised Classification Algorithms**

**INSTRUCTOR: Dr. Le Gruenwald**

**Professor, Department of Computer Science , University of Oklahoma**

**Link to Project:**

**Link to YouTube Demo:**

**Team:**

Gopal Krishna, Sudhindra [sudhi@ou.edu](mailto:sudhi@ou.edu)

Penki, Naveen Kumar [naveen.penki@ou.edu](mailto:naveen.penki@ou.edu)

Byakod, Pramod Arvind [byakodpramod@ou.edu](mailto:byakodpramod@ou.edu)

Mudduluru, Sanjana [sanjana@ou.edu](mailto:sanjana@ou.edu%09)

**ABSTRACT**

Classification has gained more attention these days. Given one or more inputs, a classification model will try to predict the class label of one or more outcomes. It has various applications like filtering emails (spam or not spam), fraudulent transaction detection, predicting the Cancer prediction and so on. Our main objective will classify a review submitted by a user and predicts whether it belongs to ‘positive’, ‘negative’, or ‘neutral’. We have used the Amazon’s Electronic Product Data taken from [here](http://jmcauley.ucsd.edu/data/amazon/) and trained a set of classification algorithms. The training data will have the class labels ranging from 1 to 5. On scale of 1 to 5, ‘positive’ lies in the scale of 4 to 5, ‘negative’ lies in the scale of 1 to 2 and ‘neutral’ falls into a scale of 3. Some of them include K-Nearest Neighbors, Random Forest Classifier, SVM (Support Vector Machine), Decision Trees. From our work, we have analyzed the performance of the classification algorithms in terms of accuracy, precision, recall, f-Measure.

**SECTION – I**

**INTRODUCTION**

**a. Objectives:**

* The idea of the project is to classify amazon product reviews by analyzing the sentiment [Liu, 2012].
* Electronic gadgets being one of the sold products on Amazon, the data chosen to analyze in this project are reviews related to the same.
* The application built on this project will help the user to determine whether the product is worth a buy.
* The review ranges from 1 through 5, 1 signifying the lowest rating, 5 signifying the highest rating and 3 signifying neutral rating, rather than a positive or negative binary rating.
* To achieve the level of abstraction four types of classification are carried out on the review data.
* To choose the best classification accuracy, precision, recall, and F-1 scores will be calculated [Amancio, 2014].

# b. Significance of the project:

The significance of this project is to determine the true rating of an electronic product when compared to the overall rating mentioned on the website. For an instance, the rating mentioned in the website need not be a legit or in other words, the rating always need not mean it is all about the product, sometimes the product could be exceptional but the service that the customer received would not be up to the mark and sometimes for a sake of refund, some customers would write a bad review so that, the developer or a seller would contact him/her to offer some sort of compensation. To avoid this havoc our application provides an actual review of the product by semantically analyzing the statements of each rating and determine whether or not the product is worth a money or are all the reviews legit by comparing our prediction with actual prediction. The existing system [Fang, 2015], semantically analyzes the data but it only predicts whether the rating is positive or negative in terms of product. The future application would also concentrate on different products and ideas that the user may obtain about similar products, if the product which the user is reviewing has lower review than that of other similar product [Hayashi, 2018] [Zhou, 2015].

**SECTION – II**

**LITERATURE REVIEW**

**RELATED WORK & GAPS**

**SECTION - III**

**DATASET DESCRIPTION**

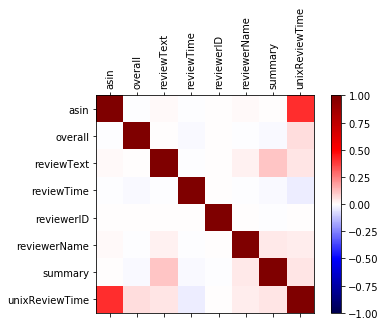
We have taken the data set from <http://jmcauley.ucsd.edu/data/amazon/>   
The dataset consists of nine attributes and the sample record of the user review looks like below:

|  |  |  |
| --- | --- | --- |
| Serial Number | Attribute Name | Sample Record Text |
| 1 | **reviewerId** | A2JXAZZI9PHK9Z |
| 2 | asin | 0594451647 |
| 3 | reviewerName | Billy G. Noland \"Bill Noland\" |
| 4 | helpful | [3, 3] |
| 5 | reviewText | I am using this with a Nook HD+. It works as described. The HD picture on my Samsung 52&#34; TV is excellent |
| 6 | overall | 5.0 |
| 7 | summary | HDMI Nook adapter cable |
| 8 | unixReviewTime | 1388707200 |
| 9 | reviewTime | 01 3, 2014 |

**DATA PROCESSING**

For the above dataset, we have computed the co-relation between each and other attributes to remove the attributed which are not co-related any way to other. The other preprocessing activity consists of cleaning the ‘reviewText’ field. We have removed the stop words from each ‘reviewtext’ and did the parts of speech tagging using NLTK for doing semantic analysis. We also removed the records from our dataset if the ‘reviewText’ has no enough information (based on the POS tagged chunks) to analyze. We have stored each of the chunks in a list to analyze the sentiment (Positive, Negative, or Neutral) of the review. ‘Overall’ is the prediction label for the review which we will be predicting using the data mining supervised classification algorithms.

**Co-relation Matrix**



**ALGORITHMS**

1. **KNN (K-Nearest Neighbor)**

**K-Nearest Neighbor** is a prominent supervised classification algorithm. When the K value is 1, it is called the Nearest Neighbor algorithm. It is a nonparametric and instance-based algorithm. It will not make any assumptions underlying in the distribution of the data. Given a vector with features, KNN algorithm makes the predictions based on the outcome of the K-Neighbors closest to that point. The closeness here can be measured in terms of distance. There are lot of popular distance metrics available and some of them are Euclidean, Cosine, Manhattan, Mahalanbois and Each has its own importance. The most challenging thing of the KNN algorithm is to find the suitable value of k [Moldagulova, 2018]. When we give a new record to predict the label, the classifier performs the following steps.

1. It runs through the whole dataset points computing the distance measure between the new point and each observation in the training set. We’ll call the K points in the training data that are closest to training observations.  Usually K is taken an odd to prevent tie situations between the observations.
2. Returns the label of the nearest observation as the predicted label

In our application, we are going to implement the K-NN algorithm to predict the ‘overall’ label with Euclidean as a distance metrics between the training vectors. The model will give the closest label of train vector when we give any new review data.

1. **SVM**

**SVM (Support Vector Machine)** is a supervised data mining algorithm which can be used mainly for classification. Here we plot every instance of the data in an n-dimensional space, n being a number of features, with value of feature being a value of each coordinate. Classification will be performed by first forming a optimum separation hyperplane. Hyperplane is a vector which divides the data instances in n-dimensional space into different classes. To differentiate the complex data, we make use of kernel technique of SVM algorithm, which will help convert the non-separable problem into separable problem. [Elmurngi,2018] [Algotar, 2017]. Kernel technique can be linear, quadratic or Gaussian. SVM needs the data to be divided into 2 distinct categories to achieve higher accuracy and it also makes it life easier for finding a hyperplane. So here we divide the data into, deception and useful [Algotar, 2017]. “Deception” predicts if the data is fake or not. “Useful” predicts if the data is useful or not. Once we have the data divided into two categories, kernel equation helps to decide on hyperplane that maximize the distance support vectors. This is the basic idea behind the SVM implementation on our dataset.

1. **Decision Tree**

Decision trees build classification models by breaking the dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node (e.g., helpful attribute) has 4 branches (e.g., 0, 1, 2 and 3). Leaf node (e.g., Overall attribute) represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node**.** ID3 is the algorithm chosen to construct the decision tree. It is a top-down, greedy search method through the space of possible branches with no backtracking. ID3 uses *Entropy* and *Information Gain* to construct a decision tree.

A set of decision rules are generated after the tree is constructed mapping all the possible routes from root node to the leaf nodes. [Song, 2015] [Elmurngi, 2018]

1. **Random Forest**

The algorithm for the random forest classifier is an existing system. Random forest is an estimator that fits a number of tree classifiers on various sub-samples. These sub-samples are the samples drawn from the training dataset with replacement meaning, a data row in a sub-sample could be repeated more than once and a data row from training dataset may not be drawn to any of those sub-samples. The sub-sampling technique in the random forest is referred to bootstrapping

Given a training dataset Td, classifier creates k bootstrap sample of Td, each denoted by Td(i), “i” ranging from 1 to k. The classifier constructs k decision trees which are known as “forest”. Then to classify, each tree returns its class prediction and prediction with the highest vote is considered as final decision [Fang, 2015] [scikit-learn]. Gini index is used to measure the tree induction for the CART algorithm.

This algorithm works fairly similar to that of decision tree algorithm mentioned before for this project, but in a broader exploration of different trees all at once.

**IMPLEMENTATION PROCESS**

**RESULTS & ANALYSIS**

**Performance evaluation plan:**

To evaluate the performance of each model, the following metrics are calculated using confusion matrix [Hossin, 2015].

1. **Accuracy:** The accuracy is the ratio between the number of correct predictions to the total predictions.
2. **Precision:** The precision of a multi class model can be calculated using either the ‘micro’ average or the ‘macro’ average. The micro precision is calculated from the individual true positives, true negatives, false positives and false negatives of the 5-class model. The macro precision is the average performance of each individual class.
3. **Recall:** Recall is defined as the number of true positives over the number of true positives and the number of false negatives of class.
4. **F-score:** F-measure is computed over each class first (micro F-measure) and then the average over all classes (macro F-measure) is taken. The macro-averaged F-score gives equal weight to each category regardless of its frequency. The score is influenced more by the classifier’s performance on rare classes.
5. **Receiver Operating Characteristics (ROC):** The key feature of ROC is the distinction between true positive rate and false positive rate as two performance measures. Pairwise comparison of classes and (Area Under the Curve) AUC is used to evaluate the performance of the multi-class classification model.

Each model is evaluated using the above measures to give a good comparison among the algorithms

**SOFTWARE APPLICATION OVERVIEW**

**Software Implementation Details**

|  |  |  |
| --- | --- | --- |
| Project Component | Software’s/ IDE | Functionalities |
| Programming Languages | Python 3.6  HTML/ CSS/ Java Script | 1. Data cleaning  2. Algorithm implementations  3. Web application development |
| Database | SQLite3 | 1. Model Data storage and accessing |
| IDE | Anaconda  PyCharm | 1. Project Packaging  2. Running test cases |
| Code Editors | Atom  Visual Studio Code | 1. File editing’s/ modifications |
| Version Control | GitHub | 1. Code Management  2. Version control |
| Visualizations | Python Libraries | 1. Statistics visualizations |

This web application can be used by the retailers to robustly classify the feedbacks from the consumer about the product. In our web application, the homepage provides a text area for the retailers to enter the feedbacks received from the consumers and a submit button to run the analysis. The feedback will be saved in a database at the backend. When the retailer wants to know the statistics of the feedback provided by the consumers, he can run the models which we developed in this project to know how many of them were positive, or negative or neutral. This application also provides the visualizations based on timeline in which most reviews have been submitted.

**Functionalities of GUI:**

* + User can submit the review in the front-end
  + The review will be stored in the back-end database
  + Retailer can run the data mining models to classify the user reviews
  + Retailer can visualize the statistics of the classified reviews

**Applications:**

* + With the statistics, retailer always have the chance to look at where the business is lagging and in which aspects, he needs to improve

**SECTION - IV**

**FUTURE WORK**

**CONCLUSION**

**SECTION - V**

**REFERENCES**

1. X. Fang and J. Zhan, "Sentiment analysis using product review data," *Journal of Big Data,* pp. 1-14, June 2015.
2. E. I. Elmurngi and A. Gherbi, "Unfair Reviews Detection on Amazon Reviews using Sentiment Analysis with Supervised Learning Techniques," *Journal of Computer Science,* pp. 714-726, May 2018.
3. D. R. Amancio, C. H. Comin, D. Casanova, G. Travieso, O. M. Bruno, F. A. Rodrigues and L. d. F. Costa, "A Systematic Comparison of Supervised Classifiers," *Pone Journal,* pp. 1-14, Mar 2014.
4. M. b. Hossin and M. N. Sulaiman, "A Review on Evaluation Metrics For Data Classification Evaluations," *International Journal of Data Mining & Knowledge Management Process (IJDKP),* pp. 1-11, March 2015.
5. A. Moldagulova and R. B. Sulaiman, "Using KNN Algorithm for Classification of Textual Documents," *International Conference on Information Technology (ICIT),* pp. 665-671, October 2017.
6. T. Hayashi, Y. Wang, Y. Kawai and K. Sumiya, "An E-Commerce Recommender System using Complaint Data and Review Data," WII, pp. 1-5, March 2018.
7. B. Liu and L. Zhang, "A Survey of Opinion Mining and Sentiment Analysis," in Mining Text Data, Springer US, 2012, pp. 415-463.
8. F. Zhou, R. Jianxin Jiao and J. S. Linsey, "Latent Customer Needs Elicitation by Use Case Analogical Reasoning From Sentiment Analysis of Online Product Reviews," Journal of Mechanical Design, pp. 1-12, July 2015.
9. Y. Song and Y. Lu, "Decision tree methods: applications for classification and prediction," Shanghai Arch Psychiatry, p. Shanghai Arch Psychiatry, April 2015.
10. K. Algotar and A. Bansal, "Detecting Truthful and Useful Consumer Reviews for Products using Opinion Mining," EMSASW, pp. 63-72, Jan 2018.
11. "scikit-learn," [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. [Accessed September, October 2018].