An Sentimental Analysis of Product Success Rate Prediction using Text Mining

1N.Ashwin Kumar, 2B.Vijay Kumar, 3S.Vishnu Prabhu, 4S.Naga Sasikanth Narayanan, 5S.Chidambaram

[1,2,3,4] Student, IT department, National Engineering College, Kovilpatti, Tamilnadu.
Assistant Professor(Senior), IT department, National Engineering College, Kovilpatti, Tamilnadu.

Abstract**—** In recent years, all the peoples like to do any kind of purchase through E-commerce. There is a necessity to analyze the customer feedback in order to improve their business. Here we proposed an advanced Sentiment Analysis for Product rating system that detects hidden sentiments in comments and rates the product accordingly. The system uses sentiment analysis methodology in order to achieve desired functionality. This project is an E-Commerce web application where the registered user will view the product and product features and will comment about the product. System will analyze the comments of various users and will rank product. We use a database of sentiment based keywords along with positivity or negativity weight in database and then based on these sentiment keywords mined in user comment is ranked. Comment will be analyzed by comparing the comment with the keywords stored in database. The System takes comments of various users, based on the comment, system will specify whether the product is good, bad, or worst. The role of the admin is to add product to the system and to add keywords in database. User can easily find out correct product for his usage. This application also works as an advertisement which makes many people aware about the product. This system is also useful for the users who need review about a product.

I. Introduction

Now a day's Information technology is used in the field of digital footprints over a long time period and it has been easily collected by various online service providers, such as blogs, forums, and social-networking services. Such contents have accumulated an increasing interest in data-driven business intelligence research, whose goal is to collect and analyze user's behavior. Particularly, the problem of predicting user's product adoption probability has been one of the emerging fields in this area.

Accurately predicting user's product adoption tendency is beneficial for a broad range for applications, such as targeted marketing and marketing strategy development for product providers, as well as personalized services for customers. In the literature, much of the active research has been devoted to the product adoption prediction problem. Specifically, these works usually classified users into two categories: the adopters that already consumed this product and the non-adopters that have not consumed it till now. In other words, these methods described users product adoption states with a binary buy or not representation. Then some learning algorithms are proposed to model the future adoption possibilities of those non-adopters. E.g., the popular recommender systems deal with the task of predicting users preferences to the products that they have not consumed before. In contrast to these products that are usually consumed only once (e.g., books and movies), there are plenty of products users may use frequently after buying them, such as smart devices. Actually, in a specific competitive market (*e.g., mobile devices*), it is nature for a user to switch among different products over time after they consume these products (*e.g., iPhone, Samsung, and Windows*). Compared to the traditional static buy-or-not adoption representation, the merchants care more about user's loyalty and commitment to the products over time after users consume the products. To better capture users loyalty to the frequently used products after purchase over time, we argue, the measure of *adoption rate*, i.e., the usage rate and regularity that consumers use a product at a particular time, is more appropriate to describe users preference changes to different products.

II. Existing System

There are so many alogrithms which support classification and prediction. In our work, we have considered K-means algorithm, Naïve bayes classifier and logistic regression..

A. K-means algorithm

Clustering is a process of partitioning a data set into clusters which contains set of meaningful sub-classes. Clustering helps users to understand the natural grouping or structure in a data set. Clustering is used either as a stand-alone tool to get insight into data distribution or as a pre-processing step for other algorithms. It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in pattern, machine learning, image analysis, information retrieval, and bioinformatics. Clustering used to identify similar dicom objects. This is used for unknown datasets in DICOM. This provides a base to find a distributed pattern and correlation among DICOM attributes. Thus it helps to create groups automatically based on the patient data. Researches shall use this group to identify hidden information. A. Mahendiran, N. Saravanan, N. Venkata Subramanian and N. Sairam, implements K-Means Clustering in Cloud environment. They deployed K-Means algorithm in Google Cloud using Google App Engine and Cloud SQL.They used the segmentation of brain images using K-means and FCM.

This study compared the efficiency of K-means and FCM for clustering MRI images. The segmentation of images using K-means is better than FCM for this dataset. Their work is the initial step for developing a system for information retrieval using data mining techniques. Clustering is pre-treatment part of other algorithms or kind of independent tool used to achieve data distribution, and can be used to determine isolated points . CURE, KMEANS, DBSCAN, and BIRCH are the commonly used Clustering Algorithms. Every clustering method has respective advantages like: KMEANS is simple and easy to understand, DBSCAN is capable of filtering noises magnificently, and CURE method lacks input. Thus Priti and team highlight the importance to improve the new techniques in clustering.

This method uses vector quantization from signal processing. K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. This cluster is used as the prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. This method produces exactly k different clusters of greatest possible distinction. The objective of K-Means clustering is to minimize total intra-cluster variance, or, the squared error function.

For small datasets, K-Means Algorithm provides good platform for mining. K-Means partitions n observations into k-clusters. Within the cluster, the nearest mean is inspected. In K-Means k = number of clusters needed, A case is allocated to the cluster whose distance to the cluster mean is the insignificant. The algorithm is based on finding the k-means . Using simple iterative method, K-means partitions the given dataset into number of clusters k(specified by user). The K-means algorithm operates on a set of D-dimensional vectors, $D = \{x_i \mid i = 1,..., N\}$, where $x_i \in d = i^{th}$ data point. The k points are picked in D is called centroids. The demerit of K-means is how to resolve quantify “closest” in the assignment.

B. Naive Bayes classifier

Naïve Bayes is a subset of Bayesian decision theory. It’s called naive because the formulation makes some naïve assumptions. Python’s text-processing abilities which split up a document into a vector are used. This can be used to classify text. Classifies may put into human-readable form. It is a popular classification method in addition to conditional independence, overfitting, and Bayesian methods. In the face of the simplicity of Naive Bayes, it can classify documents surprisingly well. Instinctively a potential justification for the conditional independence assumption is that if the document is about politics, this is a good evidence of the kinds of other words found in the document.

Naive Bayes is a reasonable classifier in this sense and has minimal storage and fast training, it is applied to time-storage critical applications, such as automatically classifying web pages into types and spam filtering. Considering a set of objects, each of which belongs to a known class, and each of which has a known vector of variables, the aim is to create a rule which enables to allocate future objects to a class, given just the vectors of variables marking out the future objects. These problems are known as supervised classification problem, are worldwide, and most of the methods for constructing such rules have been developed. It is very easy to establish, and no need any complicated repetitive parameter estimation schemes. This means it should be applied to huge data sets. It is easy to interpret, so unskilled users in classifier technology can make out the reason for it is making the classification it makes. Finally, it often does surprisingly well: it should not be the best possible classifier in any particular application, but it can usually be relied on to be robust and to do well.

Naive Bayes is a kind of classifier which uses the Bayes Theorem. It predicts membership probabilities for each class such as the probability that given record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class. This is also known as Maximum A Posteriori (MAP).

The MAP for a hypothesis is:

$$\begin{aligned}\mathbf{MAP(H)} &= \max(\ P(H|E)\ ) \\ &= \max(\ (P(E|H)*P(H))/P(E)) \\ &= \max(P(E|H)*P(H))\end{aligned}$$

P(E) is evidence probability, and it is used to normalize the result. It remains same so, removing it won’t affect.

Naive Bayes classifier assumes that all the features are **unrelated** to each other. Presence or absence of a feature does not influence the presence or absence of any other feature.

A fruit may be considered to be an apple if it is red, round, and about 4″ in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple. In real datasets, we test a hypothesis given multiple evidence (feature). So, calculations become complicated. To simplify the work, the feature independence approach is used to ‘uncouple’ multiple evidence and treat each as an independent one.

$$\mathbf{P(H|Multiple\ Evidences)} = P(E1|\ H)*\ P(E2|H) \ldots\ldots *P(En|H) * P(H) / P(Multiple\ Evidences)$$

C. Logisitic Regression

Logistic regression is a statistical analysis method used to predict a data value based on prior observations of a data set. A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables. For example, a logistic regression could be used to predict whether a political candidate will win or lose an election or whether a high school student will be admitted to a particular college. The resulting analytical model can take into consideration multiple input criteria -- in the case of college acceptance, things such as the student's grade point average, SAT score and number of extracurricular activities. Based on historical data about earlier outcomes involving the same input criteria, it then scores new cases on their probability of falling into a particular outcome category.

Logistic regression has become an important tool in the discipline of machine learning. The approach allows an algorithm being used in a machine learning application to classify incoming data based on historical data. As more relevant data comes in, the algorithm should get better at predicting classifications within data sets. For that reason, logistic regression has become particularly popular in online advertising, enabling marketers to predict as a yes or no percentage the likelihood of specific website users who will click on particular advertisements.

Types of Logistic Regression:

i. Binary Logistic Regression

The categorical response has only two 2 possible outcomes. Example: Spam or Not

ii. Multinomial Logistic Regression

Three or more categories without ordering. Example: Predicting which food is preferred more (Veg, Non-Veg, Vegan)

iii. Ordinal Logistic Regression

Three or more categories with ordering. Example: Movie rating from 1 to 5

III. PROPOSED SYSTEM

Support Vector Machine

Support vector machine is supervised learning models with associated learning algorithms that analyze data used for regression analysis and classification. A Support Vector Machine (SVM) performs classification by finding the hyper plane which maximizes the margin between the two classes. The vectors (cases) that define the hyper plane are the support vectors.

A support vector machine applies classification and regression techniques to implement associated learning algorithms which analyze DICOM and recognize patterns . The SVM works on hyper plane or set of hyper planes. SVMs are helpful in solving problems in bio-informatics, particularly useful in analyzing microarray expression data and detecting remote protein homologies. The SVM modeling error can be by limiting the model complexity. The model complexity shall be limited by applying the Structural risk minimization principle and VC theory (The Vapnik-Chervonenkis). This is useful in training the dataset. The Standard quadratic programming tools shall be used to solve optimization problems. The SVM Decision function is written as,

$$f(x) = sign \left(\sum_{i=1}^{N} a_i y_i \mathrm{k}(\mathrm{x}, x_i) + \mathrm{b} \right)$$

Basically SVM is a binary classifier. The objective of SVM is to classify two classes of instances by finding the maximum separating hyper plane between two. In order to allow more classes multiple methods are used. One Vs one method creates one binary class for each pair of classes. If there are three classes, then three binary classifiers will be created. In the simple form, Support vector machines are called as linear classifiers. Kernel trick shall be used to create non linear SVM by increasing the dimensionality of feature space. SVM uses many kernels. Most important kernels are Linear kernel, Polynomial kernel, Gaussian kernel. Gaussian kernel is special case for Radial Base Function. In the standard case, the distance used is the Euclidean distance. In the RBF kernel, the parameters determine, the width of the kernel, and d(x, y) is the distance metric. In Machine learning Applications, SVM offers more robust and accurate methods among all well known algorithms. Since SVM has strong theoretical foundation, it requires only a dozen examples for training. Strong community drives SVM development at good pace. Thus many efficient methods for training are evolving regularly. For binary classes, SVM finds the best classification function to distinguish between members of two classes in training data. For a linearly separable dataset, the two classes are separated by a linear classification function which passes through the middle of the two classes, separating the two in hyper plane f(x). After this, by testing the sign of the function f (x_n), The new data instance xn can be classified, where x_n belongs to the positive class for f (x_n) > 0. By maximizing the margin between the two classes, the best function is found. The margin defined as the amount of space, or separation between the two classes. In Geometry, the shortest distance between the closest data points to a point on the hyper plane is defined as margin. This helps to define which hyper planes are qualified and which are not qualified. There are an infinite number of hyper planes, so only a few will qualify for SVM.

The maximum margin in hyper plane offers the best generalization ability. This also provides best classification performance on the training data. A **Neural network** is a parallel distributed processor that has a propensity for storing experiential knowledge and making it available for users

(Rumelhart a.o). Neural computing is the study of networks of adaptable nodes which store experiential knowledge using learning process. A neural network is a finite-state machine made up of elementary units called neurons (Minsky). Darsana and team discusses about applying **Neural network** for image retrieval, fast computations. By using fuzzy c-means algorithm the Image retrieval problem is solved. Darsana.B., Dr.G.Jagajothi, proposes neural network classifications for image retrieval . They have used feature extraction method, Gabor filters and training neural network, precision and recall methods. They implemented the proposed system using java based platform. The query image must be pre-processed and the output objects extracted from the input query image in multi level database. The images are coached in the neural network. This helps in effective querying of images. They achieved good F-measure values using this system. It has proposed the new system with neural network for DICOM. They are using this technique to overcome the slow rate in data analysis. They are also suggesting new ways to improve the system.

Table 1: Performance comparison of various classifiers

Data Set	Accuracy Rate				Error Rate			
	K-Means Algorithm	Naive Bayes	Logistic Regression	SVM	K-Means Algorithm	Naive Bayes	Logistic Regression	SVM
amazon_alexa.tsv	88.20	82.65	92.93	93.6	11.80	17.35	7.07	6.4
flavours_of_cacao .csv	82.84	81.73	83.21	88	13.16	18.27	23.79	12
amazon_unblocked_mobile.csv	87.96	84.12	90.32	91.78	12.04	15.88	9.68	8.22
netflix_show.csv	75.15	68.54	81.15	85.29	24.85	31.46	18.85	14.71

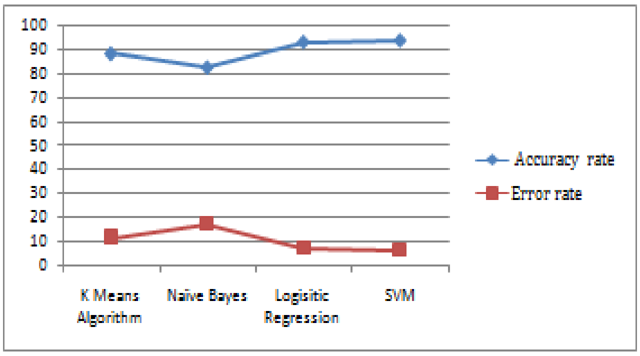

Figure 1: Comparison of various classification approaches.

CONCULSION

Thus by analyzing and executing the above mentioned data sets, it has been found that the production success rate prediction can be efficiently tracked with the help of support vector machine which is a proposed model in this paper. The error rates predicted from the data sets for support vector Machine is less when compared to the other algorithms such as naïve bayes algorithm, k-means clustering algorithm and logistic regression. support vector machine (SVM) offers higher accuracy of results and better product adoption rate for the benefit of the owners and the users. So this SVM model can be used by the owners and users for future use and adoption of product based on their success ratio.

References

[1]. Prasanna Desikan, Kuo-Wei Hsu, Jaideep Srivastava,"Data mining for healthcare management," International Conference on data mining, April 2011.

[2]. G. Adomavicius and A. Tuzhilin. Toward the next generationof recommender systems: A survey of the state-of-the-art and possible extensions. *TKDE*, 17(6):734–749, 2005.

[3]. H. Akaike. Fitting autoregressive models for prediction. *AISM*,21(1):243–247, 1969.

[4]. S. Bao, R. Li, Y. Yu, and Y. Cao. Competitor mining with the web.*TKDE*, 20(10):1297–1310, 2008.

[5]. F. M. Bass. Comments on a new product growth for model consumer durables the bass model. *Management science*, 50:1833– 1840, 2004.

[6]. Dwight A. Simon,"DICOM basics reference Basic DICOM Concepts with Healthcare Workflow," DICOM 2005 International Conference Budapest, Hungary ,September 2005.

[7]. J. Umamaheswari, Dr. G. Radhamani, "A Hybrid Approach for Classification of DICOM Image," World of Computer Science and Information Technology Journal , 2011

[8]. Krzysztof J. Ciosa, G. William Moore "Uniqueness of medical data mining," Artificial Intelligence in Medicine, March 2002

[9]. J. Ren, S. D. Lee, X. Chen, B. Kao, R. Cheng, and D. Cheung, ―Naive Bayes Classification of Uncertain Data,‖ no. 60703110.

[10]. Toon Calders, SiccoVerwer, ―Three naive Bayes approaches for discrimination-free classification‖, Data Min Knowl Disk, 2010.

[11]. Divdeep Singh Sukhpreet Kaur, "Scope of Data Mining in Medicine," International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 6, June 2014

[12]. Richard Chen MD, Pattanasak Mongkolwat and David Channin,"Radiology Data Mining Applications using Imaging Informatics," InTech 2008

[13]. L. Rabiner and B.-H. Juang. An introduction to hidden markov models. *ASSP Magazine*, 3(1):4–16, 1986.

[14]. J. Surowiecki. The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business. *Economies, Societies and Nations*, 2004.

[15]. J. Surowiecki, M. P. Silverman, et al. The wisdom of crowds. *AJP*, 75(2):190–192, 2007.

[16]. G. C. Wei and M. A. Tanner. A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association*, 85(411):699–704, 1990.

[17]. Q. Wei, D. Qiao, J. Zhang, G. Chen, and X. Guo. A novel bipartite graph based competitiveness degree analysis from query logs.*TKDD*, 11(2):21, 2016.

[18]. L. Wu, Q. Liu, E. Chen, X. Xie, and C. Tan. Product adoption rate prediction: A multifactor view. In *SDM*. SIAM, 2015.

[19]. K. Xu, S. S. Liao, J. Li, and Y. Song. Mining comparative opinions from customer reviews for competitive intelligence. *Decision support systems*, 50(4):743–754, 2011.

[20]. H. Zhang, G. Kim, and E. P. Xing. Dynamic topic modeling for monitoring market competition from online text and image data In *SIGKDD*, pages 1425–1434. ACM, 2015.

[21]. Y. Zhang and M. Pennacchiotti. Predicting purchase behaviors from social media. In *WWW*, pages 1521–1532, 2013.

[22]. Y. Zhang and M. Pennacchiotti. Recommending branded products from social media. In *RecSys*, pages 77–84. ACM, 2013.