Consumer Purchase Intention Prediction System

Ajinkya Hazare[1], Abhishek Kalshetti[2], Pratik Barai[3], Premsing Rathod[4], Prof. Pramila M. Chawan[5]

[1,2,3,4]B. Tech Student, Dept. of Computer Engineering and IT, VJTI College, Mumbai, Maharashtra, India
[5]Associate Professor, Dept. of Computer Engineering and IT, VJTI College, Mumbai, Maharashtra, India

---

Abstract - *Lately, the world has witnessed a symbolic rise in the e-commerce market, particularly in people buying products online. There has been a copious amount of research being done on discovering the purchasing patterns of a user, and above all on the factors determining whether a product will be bought or not by the user. Here, we will be finding out whether it is possible to identify and foretell the intention of a user towards purchasing a product. We will pick out that user and target him/her for the product using personalized commercials or agreements. Further, we intend to develop software for identifying potential customers by evaluating their purchase intention from their user profile data and tweets from Twitter for businesses. After using several text analytical models on the tweets' data, we have discovered that it is certainly possible to determine whether a user is displaying an intention towards purchasing a product or not. And some of our analysis shows that customers who initially showed interest in a product have mostly purchased it.*

***Key Words*: purchasing patterns, purchase intention, potential customers, profile data and tweets, e-commerce market**

1. INTRODUCTION

A number of research studies have been made to get insights into the purchasing behavior of online customers. Still, only a few have taken into consideration the users' purchasing intention for products. Here, we intend to develop a machine learning model that will estimate the customers' purchase intention for products and identify potential customers from tweets on Twitter. We have focused on a machine learning approach using text analytics because manual text analytics is inefficient. Natural language processing algorithms and text mining provide a much faster and efficient way of finding patterns and trends. To a certain extent, it can be said that identifying a customer's purchase intention is in some way or the other related to detecting customer wishes in product reviews.

1.1 Problem

Marketing managers use the customers' purchase intention as input for taking decisions on developing strategies towards new and existing services and products quite frequently. Even today many companies rely on the age-old method of collecting customer survey forms to get answers on the questions of their likeliness on buying a product to estimate the purchase intention. Here, we want to find the purchase intention using a model working on data from Twitter tweets.

1.2 Complexity

To calculate and measure customers' purchase intention from tweets is where lies the complexity of our approach. It will be a challenging task to select the best from the various text analytical methods for our task. This will involve the calculation of a lot of factors to decide the best one after measuring the results of our machine learning model.

1.3 Motivation

We intend to develop a machine learning model to predict a consumer purchase intention tweet's numerical value. This would justify that the big e-commerce companies can even use social media platforms like Twitter as a tool to decide strategies before targeting any customer. We expect that our work will stand valuable to those applications that depend upon social media for exploiting purchase intentions.

1.4 Challenges

Public datasets on purchase intention are not available, so this was our first challenge. We used a web scraper to scrape data from Twitter. The second challenge that we faced was the annotation of tweets manually as we gathered them ourselves. This was a hectic and time-consuming process as we ourselves decided the purchase intention of every tweet.

2. LITERATURE SURVEY

Online consumers' buying behavior has been studied through several research studies. Be that as it may, as it were, many have tended to the clients buying deliberately for products. A desire to buy a product or suggestions of a product were included in these wishes. Linguistic rules were used to detect these two types of wishes. Identifying the wishes using the rule-based approaches is effective, but they do not provide satisfactory coverage and their extension is not that easy. The task of wish identification to review products is close to purchasing intention detection. A machine learning approach is used rather than the rule-based approach, with features that are generic that are taken from the tweets' data.

Linear Regression, Random Forest, Naive Bayes, and Support Vector Machine are some of the frequent machine learning algorithms that can be used for text analysis. Sentiment analysis tells the consumer if they should buy a product or

not by giving them the information about that product. This analysis is primarily used by marketers and firms to understand the user's requirements so that they could offer the best possible products and services to the consumer.

Analyzing or searching the factual data present and processing that data is the main focus of Textual Information retrieval techniques. Although facts have an unbiased constituent, there are a few textual scripts that express subjective characteristics. The core of Sentiment Analysis is formed by content which is mainly appraisals, opinions, sentiments, attitudes, and emotions. Availability of enormous growth of information like blogs and social networks on online sources offers a huge amount of challenging opportunities for new application development. For example, recommendations of products given by a recommendation engine can be speculated by taking into consideration positive or negative opinions of the accounts about those products by applying the rules of Sentiment Analysis.

Pak and Paroubek (2010) [1] have proposed a model which classifies the tweets data into categories as objective, positive and negative. Annotation of tweets using emoticons automatically and collection of tweets using Twitter API was carried out to create a twitter corpus. A sentiment classifier was developed using that corpus which was based on the multinomial Naive Bayes method that proposes to use features like POS-tags and Ngram. The training set that they have taken into consideration was less effective as it consisted of tweets only having emoticons.

Barbosa et al. (2010) [2] in their model designed a two-phase sentiment analysis method which was automatic for classification of tweets data. In the first part classification of tweets as objective or subjective was completed and then in the second part, the classification of subjective tweets as positive or negative was carried out. Prior polarity of words, retweets done by the user, any link that existed in the data, punctuation marks, hashtags used by the consumer, exclamation marks in the texts and POS consisted of feature space which was taken into consideration.

Bifet and Frank (2010) [3] used hoeffding trees, multinomial naive Bayes and stochastic gradient descent in their approach. The data that was used by them was Twitter streaming data that was provided by Firehouse API. Publicly available messages in real-time of every use was given by this data. After considering all the three models in depth they came to the conclusion that when applied with proper learning, the stochastic gradient descent-based model performed better than the rest of the models under consideration.

Agarwal et al. (2011) [4] in this study took a unigram model, a tree kernel based model and a feature based model under his consideration. Classification of sentiments was performed through a 3-way model into negative, positive and neutral classes. Tweets were represented as trees in a tree kernel-based model. 10,000 features were used in the unigram model and 100 features were used in the feature based model. The features that played the main role in the classification task were the ones that combined prior polarity of words combined with their parts-of-speech(pos) tags. The conclusion of their study was that the other two models were outperformed by the tree kernel based model.

Davidov et al., (2010) [5] suggested a method that takes advantage of user-defined hashtags in tweets in Twitter for the classification of sentiment type for data analysis using single words, punctuation, patterns, and n-grams as non-identical feature types. Sentiment classification is performed on these features which are then combined into a single feature vector. The K-Nearest Neighbour algorithm was used by them for sentiment assignment alongside the construction of feature vectors in the test and training set that was implemented on each example.

Po-Wei Liang et.al. (2014) [6] proposed an approach using Twitter API for the collection of Twitter data. Camera, movie, and mobile are the three categories in which their training data is grouped. Non-opinions, positive and negative are the labels used to label the data. Filtration of tweets containing opinions was carried out. Implementation of the Unigram Naive Bayes model was successfully completed by employing the assumption of Naive Bayes simplifying independence. Method of Chi-square feature extraction and the Mutual Information was implemented to eliminate useless features. Eventually, a positive or negative tweet, that is the orientation of a tweet is predicted.

Pablo et. al. [7] in their paper extended discrepancies for polarity detection of English tweets from Twitter data of Naive Bayes classifiers. Naive Bayes classifiers with two different variations were built. One was Baseline which classifies tweets as neutral, positive, and negative. The other was Binary which neglects neutral tweets, applies polarity lexicon, and classifies the tweets as positive or negative. Lemmas (nouns, verbs, adjectives, and adverbs), Multiword from different sources, Polarity Lexicons, and Valence Shifters were the features that were considered by the classifiers.

Turney et al [8] implemented a method based on bag-of-words for sentiment analysis. In this method, the relationships between words were not taken into consideration. Also, a document is represented as just a collection of words. Aggregation functions are used to unite the values after determining the sentiments of each and every word so as to determine the sentiment of the whole document.

Xia et al. [9] implemented a model which utilizes an ensemble framework for the Classification of Sentiments. This was obtained by a combination of different feature sets and techniques of classification. Two types of features and three base classifiers were used in their work. Word Relations and Part-of-speech information were the features used. Maximum Entropy, Support Vector Machines, and Naive Bayes were the base classifiers used. Sentiment

classification was carried out with a Meta-classifier combination. Fixed combination, weighted combination, and Meta-classifier combination were ensemble approaches applied to obtain better accuracy.

3. PROPOSED SYSTEM

3.1 Problem Statement

To implement a web application that predicts the likelihood/certainty that a customer will buy a product that he's curious about, and is supported on his social media posts like Twitter tweets. This may help the company/business to target a specific customer more efficiently and boost their sales. First, we look for Twitter tweets of potential customers eager to buy a product. And who supported those tweets, we estimate/predict the likelihood that the customer will buy the merchandise.

3.2 Problem Elaboration

Currently, we have many recommendation systems available which recommend different products to the user, most of which are not efficient. No such effective model for businesses to spot potential customers. We would like to develop software that will help businesses identify potential customers for his or her products by estimating their purchase intention in measurable terms from their tweets and user profile data on Twitter.

We aim to research the tweets associated with a product and identify the acquisition intention in it. During this way we will rank the tweets which have high purchase intention and report the name of the one that tweeted as a possible customer of the merchandise.

We will make a model by gathering tweets from users who have already expressed intention to shop for the merchandise and see their tweet history and if possible, their web search history also. Using this model, we'll input potential customers who have tweeted about the merchandise but haven't bought it. And supporting the training data the model will estimate a prediction/likelihood of whether the customer will buy out or not.

We require data from twitter to research purchase intentions. For now, we are gathering the info by scraping the tweets of a product through a scraper. we'll need a mechanism to save lots of the scrapped tweets in storage for further processing and performing analysis and for that we've decided to use mongo db. We'll have to annotate data retrieved from scrapper. Once the model is trained, we'll have to develop an internet site in order that users can easily access our application through the graphical interface of the web site. we'll show on the web site the acquisition intention rank of tweets on level of 1 to five for the specified product of the user.

3.3 Proposed Methodology

In order to perform sentiment analysis, we are required to collect data from the required source (here Twitter). Tweet Collection Tweet collection involves gathering relevant tweets about the actual area of interest. The tweets are collected using Twitter's streaming API, or the other mining tool (for example WEKA), for the specified period of time of the study. The format of the retrieved text is converted as per convenience (for example JSON just in case of). The division of dataset into training and testing sets is additionally a deciding factor for the efficiency of the model. The training set is the main aspect upon which the results depend. The steps involved should aim for creating the info more computer readable so as to scale back ambiguity in feature extraction.

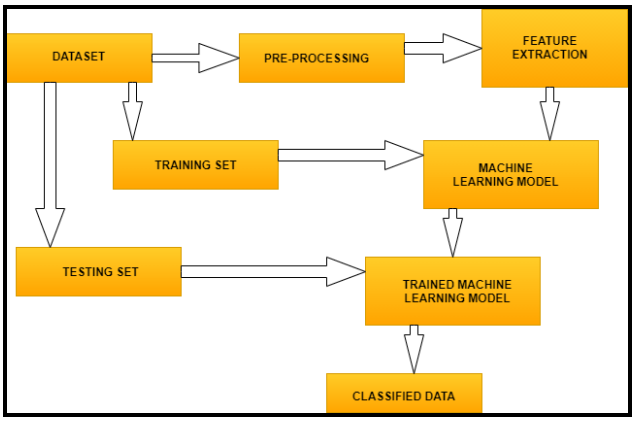

Fig-1: General Methodology for sentiment analysis

Below are a couple of steps used for preprocessing of tweets - Removal of retweets. Converting capital to lower case: just in case we are using case sensitive analysis, we'd take two occurrences of the same words as different thanks to their sentence case. It is important for an efficient analysis to not provide such misgivings to the model. Stop word removal: Stop words that don't affect the meaning of the tweet are removed (for example and, or, still etc.). Twitter feature removal: User names and URLs aren't important from the attitude of future processing, hence their presence is futile.

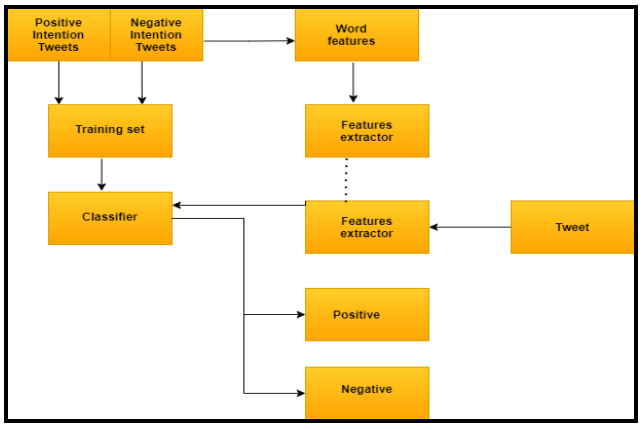

Fig-2: Sentiment Analysis Architecture

Stemming: Replacing words with their roots, reducing differing types of words with similar meanings. Sometimes they're mixed with words, hence their removal can help in associating two words that were otherwise considered different. Creating a dictionary to get rid of unwanted words and punctuation marks from the text. Expansion of slang and abbreviations. Spelling correction. Generating a dictionary for words that are important or for emoticons. a part of speech (POS) tagging: It assigns tag to every word in text and classifies a word to a selected category like noun, verb, adjective etc. POS taggers are efficient for explicit feature extraction.

As there are not any annotated Twitter tweets corpora available publicly for detection of purchase intent, we had to make our own. This was done employing a web crawler developed by JohnBakerFish which crawled the website to gather the info. We had collected over 100,000 tweets but since they weren't annotated, we had to chop right down to just 3200 tweets which were randomly selected out of the dataset and that we manually annotated them employing a basic criterion we had defined:

Table-1: Criteria for Labelling of tweets

	Tweet	Class
1	Comparing iPhone x with other phones and telling other phones are better?	No PI
2	Talking about good features of iPhone x?	PI
3	Talking about negative features of iPhone x?	No PI
4	Liked a video on YouTube about iPhone x?	PI

We used just 3200 tweets out of such an outsized dataset as we were limited by time. We defined Purchase Intention as an object that's having action words like (buy, want, desire) related to it. Each tweet was read by 3 people and the final class was decided by maximum voting.

Data preprocessing techniques:

We processed the tweets by implying these techniques in chronological order. First, we started our groundwork by converting our text into a small letter, to urge case uniformity. Then we passed that small letter text to punctuations and a special character's removal function. After the special character removal, we also applied the negation handling technique described by Dan Jurafsky in his book Tongue Processing. The technique is essential to feature NOT_ to every word between negation and the following punctuation. The next step was to stop word removal since the tweets also contain useless words which are a part of the sentence and grammar but don't contribute to the meaning of the sentence. Like "of", "the"', "a", "an", "in" and etcetera are the words mentioned above. So, we don't need these words, and it's better to get rid of them. Further, we also removed the highest 2 commonest words because their recurrence doesn't contribute to the meaning within the sentence. This will even be the result of a mistake because the data we are analyzing is off-the-cuff data where formal sentence norms aren't taken into consideration. We also removed some rare words like names, brand words (not iPhone x), overlooked HTML tags, etc. These are unique words that don't contribute much to interpretation within the model. Finally, we stemmed the words to their root. Stemming works like cutting the highest or beginning of the word, considering the common prefixes or suffixes which can be found therein the word. For our purpose, we used Porter's Stemmer, which is out there with NLTK. We also experimented with lemmatization. The analysis is performed in morphological order. A word is traced back to its lemma, and the lemma is returned because of the output. But it didn't yield a substantial change within the corpus. After preprocessing the tweets, we were left with about 1300 tweets for training data and remaining for testing.

Formation of Document Vector:

We made 3 sorts of document vectors for the aim of experimentation. First, is the term frequency document vector. We have stored text and its labeled class in the data frame. And we have constructed a new data frame with columns as the words and documents count as the rows. So, the individual frequency of words in a document count is recorded. Second, is that the inverse document frequency vector which may be a weighting method to retrieve information from the document. Term frequency and inverse document frequency scores are calculated and then the product of TF*IDF is called TF-IDF. IDF is vital to find how relevant a word is. Normally words like 'is', 'the', 'and' etc. have greater TF. So the IDF calculated a weight to tell how important the least occurring words are. Lastly, we also used the text blob library to help create the document vector. With the help of the text blob library, we calculated the sentiments of the individual words and then multiplied the sentiment score with TF and TF-IDF of that word.

Modelling:

At this stage, the info preparation was complete, and that we were able to build our model. As discussed above we chose these 5 text analytical algorithms; Support Vector Machine, Naïve Bayes, Logistic Regression, Decision Tree and Artificial Neural Network, because they're the foremost employed by researchers in this field.

To split our dataset for training and testing we first used the straightforward split of 70-30. However, since our dataset was limited, and that we also had an imbalance class problem we also used the k-fold technique with k=5.

1. For the first algorithm, the multinomial Naive Bayes classifier, we configured it as follows:

- Used Laplace smoothing for features not present in the learning samples to prevent zero probabilities in testing data.
- Also considered the prior probability of the features rather than using a uniform prior probability.

2. For the next algorithm, the Support Vector Machine classifier, we configured it as follows:
   - The algorithm we used was the linear SVM.
   - The penalty of a mistake was set to 1.
   - Considered probability estimates.
3. The next algorithm we used was Logistic Regression with the subsequent configuration:
   - The inverse of the regularization strength coefficient was set to 1 for stronger regularization.
   - A maximum number of iterations to converge was set to 100.
   - For optimization, we used the lab linear algorithm as it is best suited for small datasets.
4. We also tested the Decision Tree classifier with the following configuration:
   - The function to live the standard of a split was 'Gini'
   - At least 7 samples were required to separate an indoor node as this was giving the very best accuracy.
5. Lastly, we used the Artificial Neural Network (ANN) algorithm having the following configurations:
   - 'Relu' was the activation function used for implementing hidden layers.
   - 'lbfgs', a quasi-Newton methods optimizer, was used for weight optimization because it can perform better and converge faster for small datasets.
   - The weight updates' learning rate schedule was fixed as constant.
   - 50, 20, 10, 5 respectively were the configurations of the hidden layer.
   - The number of features constituted the input layer.
   - The 2 classes formed the output layer.

3.4 Algorithms

3.4.1 SVM

Support Vector Machine Algorithm: Support vector machines are supervised models with associated learning algorithms that analyze data used for classification and multivariate analysis. It makes use of the concept of decision planes that outline decision boundaries.

$g(x) = w^T Ø(X) + b$

X is a feature vector, 'w' is the weight of a vector and 'b' is a bias vector. Ø () is the non-linear mapping from input space to high dimensional feature space. So, we can say that SVMs machine learning model could be used for recognition of patterns.

3.4.2 Naive Bayes

it's a probabilistic classifier with a robust conditional independence assumption that's optimal for classifying classes with highly dependent features.

$P(X|y_i) = \pi_{i=1}^{m} P(x_i|y_i)$

X may be a feature vector defined as X = {x1, x2 ....} and yj may be a class label. Naïve Bayes may be a very simple classifier with acceptable results but not nearly as good as other classifiers.

3.4.3 Logistic Regression

Logistic regression predicts a binary outcome, i.e., (Y/N) or (1/0) or (True/False). It also works as a special case of linear regression. It produces an S-shaped curve better known as a sigmoid. It takes real values between 0 and 1. The model of logistic regression is given by:

Output: 0 or l

Hypothesis: Z = WX+B

$h_ø$ (x) = sigmoid (Z)

Basically, logistic regression has a binary target variable. There can be categories of target variables that can be predicted by it. The logistic classifier uses a cross- validation estimator.

3.4.4 Decision Tree

Decision Tree could even be a Supervised learning technique which may be used for both classification and Regression problems, but mostly it's preferred for solving Classification problems. It's a tree-structured classifier, where internal nodes represent the features of a knowledge set, branches represent the choice rules and every leaf node represents the result.

Attribute Selection Measures:

1. Information Gain: Information Gain= Entropy(S)-[(Weighted Avg) *Entropy (each feature)]

2. Gini Index: Gini Index= 1- $\sum_j P_j^2$

3.4.5 Neural Networks

It is a deep learning machine algorithm, which is arranged in a layer of neurons. It consists of input layer, output layer and hidden layers of neurons. Neuron network is adaptive as neurons in these layers learn from their initial input and subsequent runs.

4. CONCLUSIONS

In this paper, after carefully considering all the relevant research that had been carried out in the similar field, we provide a review and relative study of different machine learning approaches for analyzing the purchasing behavior of online customers. Twitter sentiment analysis fits in the profile of mining of text and opinions. The primary focus of sentiment analysis is to check the accuracy of a machine learning model by feeding data into it, therefore we can use this machine learning model for future purposes. It analyzes the sentiments of tweets so that we could analyze the data. Steps like collection of data, pre-processing of text, detection and classification of sentiments and model testing and training are part of sentiment analysis. Last decade witnessed a huge development in this research topic with the machine learning models outstretching the efficiency upto 85%-90%. Nevertheless, it remains to be deficient in the dimensions of distinctiveness in the data. Therefore, we can deduce that sentiment analysis has a sparkling opportunity of development in future.

5. REFERENCES

[1] A. Pak and P. Paroubek. „Twitter as a Corpus for Sentiment Analysis and Opinion Mining". In Proceedings of the Seventh Conference on International Language Resources and Evaluation, 2010, pp.1320-1326

[2] L. Barbosa, J. Feng. "Robust Sentiment Detection on Twitter from Biased and Noisy Data". COLING 2010: Poster Volume, pp. 36-44.

[3] Bifet and E. Frank, "Sentiment Knowledge Discovery in Twitter Streaming Data", In Proceedings of the 13th International Conference on Discovery Science, Berlin, Germany: Springer,2010, pp. 1-15.

[4] Agarwal, B. Xie, I. Vovsha, O. Rambow, R. Passonneau, "Sentiment Analysis of Twitter Data", In Proceedings of the ACL 2011Workshop on Languages in Social Media, 2011, pp. 30-38

[5] Dmitry Davidov, Ari Rappoport." Enhanced Sentiment Learning Using Twitter Hashtags and Smileys". Coling 2010: Poster Volumepages 241{249, Beijing, August 2010

[6] Po-Wei Liang, Bi-Ru Dai, "Opinion Mining on Social Media Data", IEEE 14th International Conference on Mobile Data Management, Milan, Italy, June 3 - 6, 2013, pp 91-96, ISBN: 978-1-494673-6068-5, http://doi.ieeecomputersociety.org/10.1109/MDM.2013.

[7] Pablo Gamallo, Marcos Garcia, "Citius: A Naive-Bayes Strategy for Sentiment Analysis on English Tweets", 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland, Aug 23-24 2014, pp 171-175.

[8] P. D. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in Proceedings of the 40th annual meeting on association for computational linguistics, pp. 417–424, Association for Computational Linguistics, 2002.

[9] R. Xia, C. Zong, and S. Li, "Ensemble of feature sets and classification algorithms for sentiment classification," Information Sciences: an International Journal, vol. 181, no. 6, pp. 1138–1152, 2011.

6. BIOGRAPHIES

Ajinkya Hazare, B. Tech Student, Dept. of Computer Engineering and IT, VJTI College, Mumbai, Maharashtra, India

Abhishek Kalshetti, B. Tech Student, Dept. of Computer Engineering and IT, VJTI College, Mumbai, Maharashtra, India

Pratik Barai, B. Tech Student, Dept. of Computer Engineering and IT, VJTI College, Mumbai, Maharashtra, India

Premsing Rathod, B. Tech Student, Dept. of Computer Engineering and IT, VJTI College, Mumbai, Maharashtra, India

Prof. Pramila M. Chawan, is working as an Associate Professor in the Computer Engineering Department of VJTI, Mumbai. She has done her B.E. (Computer Engineering) and M.E. (Computer Engineering) from VJTI College of Engineering, Mumbai University. She has 28 years of teaching experience and has guided 80+ M. Tech. projects and 100+ B. Tech. projects. She has published 134 papers in the International Journals, 20 papers in the National/International Conferences/ Symposiums. She has worked as an Organizing Committee member for 21 International Conferences and 5 AICTE/MHRD sponsored Workshops/STTPs/FDPs. She has participated in 14 National/International Conferences. She has worked as NBA Coordinator of the Computer Engineering Department of VJTI for 5 years. She had written a proposal under TEQIP-I in June 2004 for 'Creating Central Computing Facility at VJTI'. Rs. Eight Crore were sanctioned by the World Bank under TEQIP-I on this proposal. Central Computing Facility was set up at VJTI through this fund which has played a key role in improving the teaching learning process at VJTI.