International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 02 | Feb 2022 www.irjet.net p-ISSN: 2395-0072

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 - *The world is witnessing noteworthy changes in the way how the markets are functioning. Now a lot of things are visible on social media about how the people are thinking and what their thoughts are when they go for buying a product from their profile data and tweets. We here developed a model that will predict the intention of a customer of whether he or she is interested or not in buying a particular product using various text analytical models. Thus, customizing ads for different customers according to their taste and need at a particular point in time. We compared the results of our model with that of the original dataset and found out how accurate our model is. This is important because trends show that most of the customers who had shown interest in buying a product have mostly bought it.*

***Key Words*: social media, profile data and tweets, customer, text analytical models, customizing ads**

1. INTRODUCTION

Here, in this project, we identified potential customers based on their tweets on Twitter. Thus, targeting the correct audience instead of trying to sell the product to everyone present out there. We have used machine learning models here because manual testing is obsolete nowadays. The problem was to find the correct set of audiences for a particular product. Also, the reviews posted by the users on social media helped us analyze and test our model. Even today many companies use the age-old method of surveys for getting customer feedback. We come here to help them with our prediction.

The motivation behind this project was that our model can then be used by the big e-commerce industries to target customers. Thus, exploiting the available data on the internet to the fullest. Some of the challenges faced were getting the data in the desired format as it is not easily available and then annotating it according to the model.

2. LITERATURE REVIEW

Online consumers' buying behavior has been studied through several research studies. Several researches have been conducted to study the consumer's online buying behavior. Mostly these were focused on suggesting the client products so that they would buy the advertised products. The primary intention of these studies was to know the clients desire to buy a particular product. Rules based on linguistics were used to detect these types of wishes. Satisfactory coverage is not provided by these rule-based approaches and their extension is hard but identifying the wishes is effective. Purchase intention detection task is very close to task of finding and identification of review products. In these tasks, a machine learning approach is preferable than a rule-based approach because the features taken from tweet data are mostly generic.

Text analysis can be performed by using machine learning algorithms such as Linear Regression, Naive Bayes, Support Vector Machine and Random Forest. The process of telling a customer if they should purchase a product or just let it go by providing them the data about the product is carried out by sentiment analysis. Marketers and firms use this analysis to study the user's requirements so that the best possible matches for the product and customers are made by the system.

A tree kernel based model, a unigram model and a feature based model was taken into consideration by Agarwal et al. (2011) [1] in his study. Neutral, positive and negative classes were the elements of the 3-way model to perform the classification of sentiments. In a tree kernel based model tweets were represented as trees. The unigram model consisted of 10,000 features and in the feature based model 100 features were used. Prior polarity of words that was combined with tags that consisted of their parts-of-speech(pos) were the features that played a significant part in the task of classification. In conclusion, the tree kernel based model outperformed the other two models.

Pak and Paroubek (2010) [2] 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.

The Classification of Sentiments was done through a model which uses an ensemble framework was implemented by Xia et al. [3]. Different feature sets and classification techniques

were combined to create this model. Three base classifiers and two types of features were applied in their work. The base classifiers consisted of Support Vector Machines, Naive Bayes and Maximum Entropy. Meta-classifier combination was used to perform sentiment classification. Ensemble approaches such as weighted combination, meta-classifier combination and fixed combination were applied to achieve better accuracy.

Classification of tweets data was performed in a two phase sentiment analysis method by Barbosa et al. (2010) [4] in their model. Tweets were classified as objective or subjective in the first part of classification and as positive or negative in the second part of classification of tweets. Hashtags used by the consumer, retweets done by the user, any link that existed in the data, exclamation marks in the texts, prior polarity of words and POS were taken into consideration while assembling the feature space.

Multinomial naive Bayes, stochastic gradient descent and hoeffding trees were used by Bifet and Frank (2010) [5] in their approach. Twitter streaming data from Firehouse API was used by them as the primary dataset. The data provided the messages which were available publicly in real-time of everyday use. The conclusion to their work after in depth consideration of all the three models was that the stochastic gradient descent based model, when applied with proper learning, performed quite better than the rest of the models which were taken into consideration.

Sentiment analysis in which bag-of-words was used was implemented by Turney et al [6]. The mutual relationship between the words was not taken into account in this method. Also, in this model the document was represented as a collection of words. Uniting the value of each and every word after evaluating the sentiments was performed with the help of aggregation functions so that the sentiment of the entire document could be determined.

Pablo et. al. [7] paper was based on applying Naive Bayes classifiers on Twitter data for polarity detection of English tweets. Two different variations of Naive Bayes classifiers were constructed. Tweets were classified as positive, neutral, and negative in one of the Naive Bayes classifiers. Classification of the tweets as positive or negative, neglecting neutral tweets and applying polarity lexicon was carried out by another classifier which was binary. The classifier considered features such as Polarity Lexicons, Multiword extracted from different sources, Valence Shifters and Lemmas (verbs, nouns, adverbs and adjectives).

Punctuation, single words, n-grams, and patterns as non-identical feature types were used for data analysis for the sentiment type classification in Twitter of user-defined hashtags in tweets in the method proposed by Davidov et al., (2010) [8]. A single feature vector is made by combining these features after performing sentiment classification. Sentiment assignment was implemented on each example using the K-Nearest Neighbor algorithm in the test and training set alongside the feature vectors construction.

Collection of Twitter data using Twitter API was proposed by Po-Wei Liang et.al. (2014) [9] in their approach. The three categories in which the data is grouped are movie, mobile and camera. The data was labeled as positive, negative and non-opinions. Tweets containing opinions were filtered. Naive Bayes simplifying independence provided the assumption for the implementation of the Unigram Naive Bayes model whose implementation was successfully completed. To eliminate the useless features the method of Mutual Information and Chi-square feature extraction was carried out. Finally, whether the tweet is positive or negative is predicted.

3. PROPOSED SYSTEM

3.1 Problem Statement

Implement a web application that forecasts a customer's likelihood/certainty of purchasing a product that he's interested in, based on his social media posts such as Twitter tweets. This may enable the company/business to more effectively target a certain customer and increase revenues. First, we hunt for tweets from potential buyers who are eager to purchase a product on Twitter. Based on who backed those tweets, we estimate/predict the likelihood that the buyer will purchase the goods.

3.2 Dataset Description

We had to make our own because there are no publicly available annotated Twitter tweets corpora for detecting buy intent. This was accomplished by crawling the webpage with a web crawler built by JohnBakerFish. We had collected over 100,000 tweets, but because they were not tagged, we had to narrow it down to just 3200 tweets, which were chosen at random from the dataset and manually annotated using a criterion we defined:

Table -1: Criteria for Labelling of tweets

	Tweet	Class
1	Comparing iPhone x with another phone and telling other phone 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 video on YouTube about iPhone x?	PI

Due to time constraints, we only used 3200 tweets from such a vast dataset. We defined Purchase Intention as an item that is related with action words such as (purchase, want, desire). Three people read each tweet, and the final class was determined by the most votes.

3.3 Data Preprocessing

3.3.1 Data preprocessing techniques

We preprocessed the tweets using these techniques:

1. **Lowercase:** So, we started our groundwork by converting our text into lower case, to get case uniformity.
2. **Remove Punctuations:** Then we passed that lower case text to punctuations and special characters removal function. Text may contain unwanted special characters, spaces, tabs and etcetera which has no significant use in text classification.
3. **Stopwords Removal:** Text also contains useless words which are routine part of the sentence and grammar but do not contribute to the meaning of the sentence. Likes of "the", "a", "an", "in" and etcetera are the words mentioned above. So, we do not need these words, and it is better to remove these.
4. **Common Words Removal:** Then there also lots of repetitive words which from their recurrence do not contribute to the meaning in the sentence. This can also be the result of mistake as the data we are analyzing is an informal data where formal sentence norms are not taken into consideration.
5. **Rare Words Removal:** We also removed some rare words like names, brand words (not iPhone x), left out html tags etc. These are unique words which do not contribute much to interpretation in the model.
6. **Spell fix:** Social media data is full of misspellings. And it is our job to fix these errors and give the model the correct words as input.
7. **Stemming:** Then we stemmed the words to their root. Stemming works like by cutting the end or beginning of the word, considering the common prefixes or suffixes that can be found in that word. For our purpose, we used Porters Stemmer, which is available with NLTK.
8. **Lemmatization:** Next, we also performed text lemmatization. This analysis is performed in morphological order. The word goes back to the lemma and the lemma is returned as output.

3.3.2 Formation of Document Vector

We created three types of document vectors:

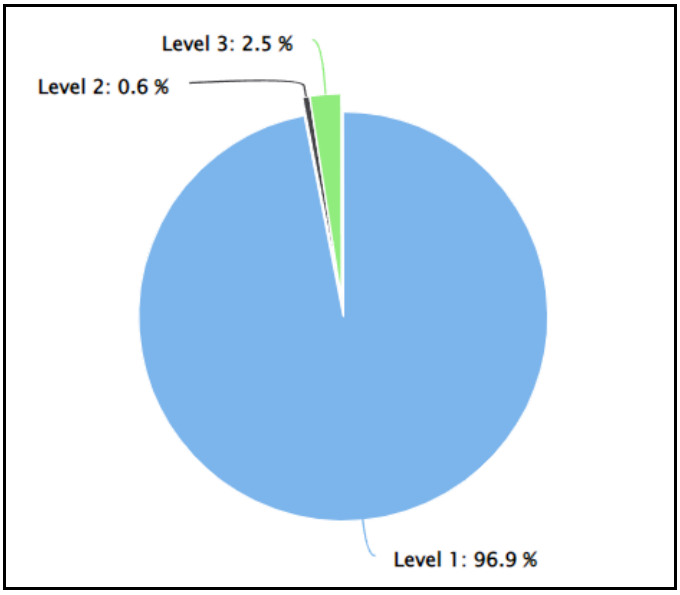
1. **TF:** The first is the term frequency document vector. I saved the text and its labeled class in a data frame. Then I built a new data frame with columns as words and number of documents as rows. Therefore, the individual frequency of words in the document count is recorded.
2. **IDF:** A weighting method for getting information from a document. The term frequency and inverse document frequency values are calculated and the product of TF * IDF is expressed in TFIDF. IDF is important for finding word relevance. Words such as `is`, ` the`, and `and` usually have a large TF. Therefore, the IDF calculated weights to indicate how important the least frequently occurring words are.
3. **TFIDF with Textblob Library:** We used the Textblob library to calculate the emotions of an individual word and multiply the emotion score by the word's TF and TFIDF.

4. IMPLEMENTATION AND RESULTS

We used the training dataset to build our models, and then tested them with the testing dataset. The following techniques based on the Confusion Matrix (A confusion matrix is a table that is typically used to assess the performance of a classification model on a set of test data for which the true values are known) were used to evaluate our models:

1. Accuracy: (TP + TN) / (TP + TN + FP + FN)
2. Precision: TP / (TP + FP)
3. Recall: TP / (TP + FN)
4. F-Measure: (2 * Precision * Recall) / (Precision + Recall)
5. True Negative Rate: TN / (TN + FN) (for imbalance class analysis) In addition, we looked at the True Positive Rate and the shape of the ROC curve for extra information.

Fig-1: Predicted Accuracy of Purchase Intention



Here, we predicted the accuracy for the model Decision Tree and Document Vector TF-IDF. The figure above shows the results in percentage. Level 1 stands for 80-100%, Level 2 stands for 60-80%, Level 3 stands for 50-60%. So, 96.9% of the data that was predicted by the model had an accuracy of 80-100%.

This is what we obtained using the simple split technique and adding all of the feature processing techniques:

Table -2: Accuracy table

	Naive Bayes	Logistic Regression	Support Vector Machine	Decision Tree	Artificial Neural Network
TF	78.2	80.2	80.5	69.3	76
TF-IDF	65.6	78.2	78.2	72.3	77.6
binary doc	77.5	**80.8**	80.2	72.6	78.9
text-blob + TF	-	79.5	78.5	66	75.2
text-blob + TF-IDF	-	78.9	76.9	69.6	75.6
text-blob + binary doc	-	79.5	78.5	72.3	79.2

The logistic regression approach employing the binary document vector provided the maximum accuracy, as seen in the accuracy table. With the TF document vector, SVM provided about the same accuracy.

Table -3: Precision table

	Naive Bayes	Logistic Regression	Support Vector Machine	Decision Tree	Artificial Neural Network
TF	83.4	83.2	85.4	83.8	84.9
TF-IDF	83.5	84.2	**86.2**	84.7	85.8
binary doc	82.5	83.8	85.9	85.1	86
text-blob + TF	-	83.4	83.9	85	84.2
text-blob + TF-IDF	-	84.8	85	85.2	86
text-blob + binary doc	-	83.4	84.5	85	83.6

Table -4: Recall table

	Naive Bayes	Logistic Regression	Support Vector Machine	Decision Tree	Artificial Neural Network
TF	90.3	**93.7**	90.8	75.7	84.5
TF-IDF	70.3	89.1	86.2	79.1	85.8
binary doc	90.7	**93.7**	89.5	79.1	87.5
text-blob + TF	-	92.5	89.9	69	84.5
text-blob + TF-IDF	-	89.1	85.8	74.5	82.4
text-blob + binary doc	-	92.4	89.1	78.6	91.6

Table -5: True Negative rate table

	Naive Bayes	Logistic Regression	Support Vector Machine	Decision Tree	Artificial Neural Network
TF	32.8	29.7	42.2	45.3	43.8
TF-IDF	48.4	37.5	48.4	46.9	46.9
binary doc	28.1	32.8	45.3	48.4	46.9
text-blob + TF	-	31.2	39.5	**54.7**	40.6
text-blob + TF-IDF	-	40.6	43.7	51.6	50
text-blob + binary doc	-	31.2	39	48.4	32.8

We also utilized the real negative rate to see if our model was skewed towards only one class because we had an imbalance class. Using the genuine negative rate measure, we can see that the model accurately predicted the negative class more than half of the time.

Then we utilized the k-fold method, which yielded the following table of results:

Table -6: Accuracy table

	Naive Bayes	Logistic Regression	Support Vector Machine	Decision Tree	Artificial Neural Network
TF + neg handling	75.2	76.9	74	69	74.2
TF-IDF + neg handling	70.2	74.4	77.7	70.4	67.8
TF + neg handling + lemmatization	75.4	77.4	74.4	70.9	72.7
TF-IDF + neg handling + lemmatization	69.6	72.8	75.9	70.4	73.7
TF + lemmatization	75.6	76.9	73.6	73.6	71.3
TF-IDF + lemmatization	73.9	74.2	**79.2**	69.3	73.6

Using the accuracy table, we can see that the highest accuracy was given by the support vector machine algorithm using lemmatization in the data and using TF-IDF as the document vector.

Table -7: True Negative rate table

	Naive Bayes	Logistic Regression	Support Vector Machine	Decision Tree	Artificial Neural Network
TF + neg handling	45.6	47	48.6	48.6	51
TF-IDF + neg handling	11.4	26.9	49.1	46.2	0
TF + neg handling + lemmatization	43.3	47.6	48.3	51.3	51
TF-IDF + neg handling + lemmatization	11.4	24.9	46	52.7	49.3
TF + lemmatization	49.4	46	47.1	**57.5**	51.7
TF-IDF + lemmatization	13.8	24.1	46	47.1	52.9

Using the true negative rate table, we can observe that of the 5 algorithms, the decision tree approach handled the imbalance class problem the best, although the SVM and ANN algorithms did as well.

5. CONCLUSIONS AND FUTURE SCOPE

We witnessed promising results from our model because both the dataset and the testing model were built from scratch. As there is no such dataset available for analyzing consumer purchase intention based on tweets from Twitter.

Five different models have been implemented by us here, therefore our project stands apart from the other researches that have been done in this field. From these five models we choose the best one suited for the product data. Accuracy of 80% or more with an imbalance class dataset would be a victory.

The Future Scope of our project would include implementing our model using deep learning algorithms. These would include algorithms like Convolutional Neural Networks (CNN), Recurrent Neural Network (RNN) and deep belief networks. Also, we would like to focus on some particular details of a product, instead of finding out the purchase intention of the product as a whole. This would help our model learn better and would provide more insights about consumer thinking and intentions.

6. REFERENCES

[1] 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

[2] 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

[3] 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.

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

[5] 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.

[6] 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.

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

[9] 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.

[10] Ajinkya Hazare, Abhishek Kalshetti, Pratik Barai, Premsing Rathod, Prof. Pramila M. Chawan, "Consumer Purchase Intention Prediction System", IRJET- Volume: 08,Issue: 11,Nov 2021, https://www.irjet.net/archives/V8/i11/IRJET-V8I1110.pdf

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.