Finding Purchase Intention Using Twitter Data

Supervisor: Miss Nida Sadaf Khan

Author: Hasan Imran

Co-Authors: M Ammar Saleem | M Saad Salman

Table of Contents

[Abstract 3](#_Toc14280099)

[1 Introduction 3](#_Toc14280100)

[1.1 Problem 3](#_Toc14280101)

[1.2 Complexity 3](#_Toc14280102)

[1.3 Motivation 4](#_Toc14280103)

[1.4 Challenges 4](#_Toc14280104)

[1.5 Organization of the paper 4](#_Toc14280105)

[2 Literary Review 4](#_Toc14280106)

[2.1 Model description 5](#_Toc14280107)

[3 Proposed Approach 6](#_Toc14280108)

[3.1 Data collection and annotation 6](#_Toc14280109)

[3.2 Data preparation 7](#_Toc14280110)

[3.2.1 Data preprocessing techniques: 7](#_Toc14280111)

[3.2.2 Formation of Document Vector 8](#_Toc14280112)

[3.3 Modelling 8](#_Toc14280113)

[4 Experimentation and Results: 9](#_Toc14280114)

[4.1 Limitations 12](#_Toc14280115)

[5 Conclusion: 12](#_Toc14280116)

[5.1 Future Work 12](#_Toc14280117)

[6 References: 12](#_Toc14280118)

# Abstract

Recently, there has been a significant rise in the ecommerce industry and more specifically in people buying products online. More and more people have started posting online about whether they want to buy the product or asking whether they should buy the product or not. There has been a lot of research being done on figuring out the buying patterns of a user and more importantly the factors which determine whether the user will buy the product or not. One such platform is Twitter which has become quite popular in recent years. In this study, we will be exploring the problem of identifying and predicting the purchase intention of a user for a product. After applying various text analytical models to tweets data, we have found that it is indeed possible to predict if a user have shown purchase intention towards a product or not, and after doing some analysis we have found that people who had initially shown purchase intention towards the product have in most cases also bought the product.

# 1 Introduction

We want to develop a machine learning approach that will identify potential customers for a product by estimating the purchase intention in measurable terms from tweets on twitter. We have used a text analytical machine learning approach because although text analytics can be performed manually, it is inefficient. By using text mining and natural language processing algorithms it will be much faster and efficient to find patterns and trends. In a way we can say that Purchase Intention detection task is close to the task of identifying wishes in product reviews.

## 1.1 Problem

Purchase intentions are frequently measured and used by marketing managers as an input for decisions about new and existing products and services. Up till now many companies still use customer survey forms in which they ask questions like how likely you are to buy a product in a given time frame and using that information they calculate the purchase intention. We want to see if we can use Twitter tweets to train a model to identify tweets which show purchase intention for a product.

## 1.2 Complexity

The complexity of our approach is that we have to calculate how to measure the purchase intention from a tweet. Exploring the different type of text analytical methods and choosing the best one for our task will be quite challenging. Measuring the results of our machine learning model and then deciding the best one will involve a lot of factors which we will have to calculate.

## 1.3 Motivation

We want to develop a machine learning model which can predict the numerical value for the consumer intention for a tweet. By doing this we can prove that we social media such as Twitter is also an important tool which marketers can use when deciding to target a customer. We believe that our work can be valuable to applications focusing on exploiting purchase intentions from social media.

## 1.4 Challenges

The first challenge we faced was that we were not able to find any public dataset regarding purchase intention. We had to scrap the data from Twitter using a web scraper. Secondly, since we ourselves gathered the data we had to manually annotate the tweets. Again, this process was extremely time consuming as we had to go through each tweet and decide the purchase intention. Thirdly, we had limited annotated data because of the lengthy process of manual annotation and time constraint.

## 1.5 Organization of the paper

The rest of this paper is organized as follows:

We review related work on purchase intention and online buying behavior in Section 2. In Section 3, we explain our data collection and annotation process, followed by model creation. In Section 4, we present the experiments and their results. Finally, Section 5 concludes the paper and provides the scope of future work.

# 2 Literary Review

There have been several research studies for analyzing the insights of online consumers buying behavior. However, only a few have addressed the customers buying intention for products. Studies on identification of wishes from texts, specifically Ramanand et al. (Ramanand, Bhavsar, and Pedanekar 2010) consider the task of identifying ‘buy’ wishes from product reviews. These wishes include suggestions for a product or a desire to buy a product. They used linguistic rules to detect these two kinds of wishes. Although rule-based approaches for identifying the wishes are effective, but their coverage is not satisfactory, and they can’t be extended easily. Purchase Intention detection task is close to the task of identifying wishes in product reviews. Here we don’t use the rule-based approach, but we present a machine learning approach with generic features extracted from the tweets.

Past studies have shown that it is possible to apply Natural Language Processing (NLP) and Named Entity Recognition (NER) to tweets (Li et al., 2012) (Liu et al., 2011). However, applying NER to tweets is very difficult because people often use abbreviations or (deliberate) misspelled words and grammatical errors in tweets. Nonetheless, Finin et al. (2010) tried to annotate named entities in tweets using crowdsourcing. Other studies used these techniques to apply sentiment analysis to tweets. The first studies used product or movie reviews because these reviews are either positive or negative. Wang et al. (2011) and Anta et al. (2013) analyzed the sentiment of tweets filtered on a certain hashtag (keywords or phrases starting with the symbol that denote the main topic of a tweet). These studies merely analyze the sentiment of a tweet about a product after the author has bought it. We will however be extracting features from tweets to find whether the user has shown purchase intention towards the product or not.

More recently, research articles like *Identifying Purchase Intentions by Extracting Information from Tweets* ( February 8, 2017, RADBOUD U NIVERSITY NIJMEGEN) and *Tweetalyst: Using Twitter Data to Analyze Consumer Decision Process* (The Berkeley Institute of Design) investigate if an artificial intelligence approach can predict (from existing user created content on twitter) if someone is a potential customer for a specific company or product and identify users at different stages of the decision process of buying a given product. Further looking at research reports like *The Impact of Social Network Marketing on Consumer Purchase Intention in Pakistan: Consumer Engagement as a Mediator* (Asian Journal of Business and Accounting 10(1), 2017) give us an insight of the impact of social network marketing on consumer purchase intention and how it is affected by the mediating role of consumer engagement. Based on UGT theory (Uses and Gratification Theory).

Some preprocessing techniques commanly used for twitter data are the sentiment140 API (Sentiment140 allows you to discover the sentiment of a brand, product, or topic on Twitter), the TweetNLP library (a tokenizer, a part-of-speech tagger, hierarchical word clusters, and a dependency parser for tweets), unigrams, bigrams and stemming. There are also some dictionary-based approaches such as using the textBlob library (TextBlob is a Python (2 and 3) library for processing textual data. It provides a consistent API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, and more).

## 2.1 Model description

After extensive research, we found that these 5 models were the most used text analytical models’ researchers have used to experiment with. We used the Scikit-learn library in python and configured our models according to the dataset.

1. Support Vector Machine (SVM): Simply put, SVM is a supervised machine learning algorithm which does complex transformation on the data. And then it tries to separate data on classes we have defined on our data.
2. Naive Bayes: Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes’ theorem with the “naive” assumption of conditional independence between every pair of features given the value of the class variable.
3. Logistic Regression: Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.
4. Decision Tree: Decision Trees are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.
5. Neural Network: It is deep learning machine algorithm, which is arranged in a layer of neuron. There is an 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.

# 3 Proposed Approach

In this section, we describe the details of our approach to tackle the problem of purchase intention detection. We will begin by describing our data collection and annotation process. Then we will describe our approach for data preprocessing and transforming the data to train text analytical models.

## 3.1 Data collection and annotation

As there are no annotated Twitter tweets corpora available publicly for detection of purchase intent, we had to create our own. This was done using a web crawler developed by JohnBakerFish which crawled the website to collect the data. We had collected over 100,000 tweets but since they were not annotated, we had to cut down to just 3200 tweets which were randomly selected out of the dataset and we manually annotated them using a basic criterion we had defined:

Criteria for Labelling of tweets

|  |  |  |
| --- | --- | --- |
|  | Tweet | Class |
| 1 | Comparing iphone x with other 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 |

We used just 3200 tweets out of such a large dataset as we were limited by time. We defined definition of Purchase Intention as object that is having action word like (buy, want, desire) associated with it. Each tweet was read by 3 people and final class was decided by maximum voting.

## 3.2 Data preparation

### Data preprocessing techniques:

We processed the tweets using these techniques in chronological order. First, we started our groundwork by converting our text into lower case, to get case uniformity. 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. After the special character removal, we also applied the negation handling technique described by Dan Jurafsky in his book Natural Language Processing. The technique is basically to add NOT\_ to every word between a negation and following punctuation. Next step was stop words removal since the tweets 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. Further we also removed the top 2 most common words because their recurrence does 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. 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. Finally, 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. We also experimented with lemmatization. The analysis is performed in morphological order. A word is traced back to its lemma, and lemma is returned as the output. But it did not yield a considerable change in 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 types of document vectors for the purpose of experimentation. First, is the term frequency document vector. We have stored text and its labeled class in data frame. And we have constructed a new data frame with columns as the words and document count as the rows. So, individual frequency of words in a document count is recorded. Second, is the inverse document frequency vector which is a weighting method to retrieve information from the document. Term frequency and inverse document frequency scores calculated and then product of TF\*IDF is called TF-IDF. IDF is important in finding how relevant a word is. Normally words like ‘is’, ‘the’, ‘and’ etc. have greater TF. So IDF calculated a weight to tell how important least occurring words are. Lastly, we also used the textblob library to help create the document vector. With the help of textblob library we calculated sentiments of individual word and then multiplied the sentiment score with TF and TF-IDF of that word.

## 3.3 Modelling

At this stage, the data preparation was complete, and we were ready 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 are the most used by researchers in this field.

To split our dataset for training and testing we first used the simple split of 70-30. However, since our dataset was limited, and 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:
   1. Used Laplace smoothing for features not present in the learning samples to prevent zero probabilities in testing data.
   2. Also considered 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:
   1. The algorithm we used was the linear SVM.
   2. The penalty of an error was set to 1.
   3. Considered probability estimates.
3. The next algorithm we used was Logistic Regression with the following configuration:
   1. The inverse of regularization strength coefficient was set to 1 for stronger regularization.
   2. Maximum number of iterations to converge was set to 100.
   3. For optimization we used the liblinear algorithm as it is best suited for small datasets.
4. We also tested the Decision Tree classifier with the following configuration:
   1. The function to measure the quality of a split was ‘gini’
   2. At least 7 samples were required to split an internal node as this was giving the highest accuracy.
5. Finally, we also used the Artificial Neural Network algorithm with the following configurations:
   1. ‘Relu’, the rectified linear unit function was used as the activation function for the hidden layer.
   2. ‘lbfgs’, an optimizer in the family of quasi-Newton methods, was the method used as the solver for weight optimization because for small datasets, ‘lbfgs’ can converge faster and perform better.
   3. The learning rate schedule for weight updates was kept to constant.
   4. The hidden layers were kept as follows 50, 20, 10, 5.
   5. The input layer was the number of features.
   6. The output layer were the 2 classes.

Once the models were configured, we used the training data to train our models and then test our data. The results are discussed in the next section.

# Experimentation and Results:

We built our models based on the training dataset and then experimented with the testing dataset on the models. To evaluate our models, we used the following techniques based on the Confusion Matrix (A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known):

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)

Further, we have also considered The True Positive Rate and the shape of the ROC curve for more insights.

Using the simple split technique and incorporating all the feature processing techniques, this is the results that we got:

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 |

Using the accuracy table, we can see that the highest accuracy was given by the logistic regression algorithm using the binary document vector. SVM also gave almost the same accuracy with the TF document vector.

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 |

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 |

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 used the true negative rate because we had an imbalance class and we had to check if our model was biased towards only one class. Using the true negative rate measure, we can see that more than half the time the model predicted the negative class correctly.

Next, we used the k-fold technique, and below are the table of results:

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.

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 see that the decision tree algorithm handled the imbalance class problem the most effectively amongst the 5 algorithms, however, SVM and ANN algorithms also handled it quite well.

## 4.1 Limitations

The 2 major problems that we faced were:

1. The imbalance class problem: Since our dataset was manually annotated by us, we had about 2000 positive tweets and 1200 negative tweets. Due to this we were getting a very low True Negative Rate and our model was not accurately predicting the negative class.
2. Limited annotated data: Since we had to manual annotate each tweet in the dataset and this process takes a lot of time, we were only able to annotate about 3200 tweets.

# 5 Conclusion:

Our results were quite promising since we had created our own dataset and were building the model from scratch. We had to create our own dataset because there does not exist a publicly available dataset for purchase intention based on twitter tweets.

Looking at the other researches that are done in the similar field, our project also stands apart since we have implemented 5 different models and after evaluating them, we choose the best one customized to the product data.

We were not able to get more than 80% accuracy because of the two problems highlighted above. To achieve even 80% accuracy with an imbalance class data and such a small dataset is a victory.

# 5.1 Future Work

To continue our work forward, it is worth trying out the dataset on deep learning models such as RNNs (recurrent neural networks), convolutional NN, and deep belief networks. Further, we can also use the dataset to find the intention shown towards specific features of the product rather than the product as a whole and target the user towards the specific feature of the product to increase the likeliness to purchase the product.

# 6 References:

1. Books:
   1. Speech and Language Processing (3rd ed. draft), Dan Jurafsky and James H. Martin.
2. Inspirations for code and designs:
   1. Building a prediction model, https://www.kaggle.com/gpayen/building-a-prediction-model
   2. Sentiment analysis, https://www.kaggle.com/laowingkin/amazon-fine-food-review-sentiment-analysis.
   3. TEXT PREPROCESSING USING PYTHON, https://www.kaggle.com/shashanksai/text-preprocessing-using-python.
3. Relevant Papers:
   1. Identifying Purchase Intentions by Extracting Information from Tweets, February 8, 2017, RADBOUD U NIVERSITY NIJMEGEN, BACHELOR ’S THESIS IN ARTIFICIAL INTELLIGENCE.
   2. Tweetalyst: Using Twitter Data to Analyze Consumer Decision Process, The Berkeley Institute of Design.
   3. The Impact of Social Network Marketing on Consumer Purchase Intention in Pakistan: Consumer Engagement as a Mediator, Asian Journal of Business and Accounting 10(1), 2017.
   4. Using Twitter Data to Infer Personal Values of Japanese Consumers, 29th Pacific Asia Conference on Language, Information and Computation pages 480 – 487 Shanghai, China, October 30 - November 1, 2015, Copyright 2015 by Yinjun Hu and Yasuo Tanida.
4. Websites:
   1. https://www.kaggle.com/snap/amazon-fine-food-reviews
   2. https://scikit-learn.org/stable/