Predicting Purchase Intention Using Twitter Data

Hasan Imran
RWTH Aachen University
Syed Zain Zaidi
Delta Galil Industries
Nida Sadaf Khan
Medical College, Agha Khan University
Nasir Uddin (nuddin@uit.edu)
UIT University

Research Article

Keywords: Text Analytics, Sentiment Analysis, Twitter Tweets, Deep Learning, Machine Learning, Purchase Intention

Posted Date: December 14th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2370113/v1

Abstract

Recently, there has been a significant rise in the ecommerce industry with the pandemic entailing lockdowns. More and more people have started posting online about the products they want to buy or asking whether they should buy the product or not. There has been a great deal of research going on in trying to figure 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. This study explores the problem of identifying and predicting the purchase intention of a user for a product. Employment of various text analysis models to tweets data reveals we can predict if a user has shown purchase intention towards a product or not. While there are lexicon and Machine learning based approaches to handle this task, our deep learning approach is a new take on purchase intention prediction from social media data. An accuracy of 85.7% was achieved by our model, applying the LSTM Neural Net with data in the Global Vectors (GloVE) embeddings format.

1 Introduction

Due to the rapid growth of the Internet and its use as a channel for shopping, today's consumers are able to shop from anywhere at any time with just a few clicks of their fingers. Despite the tremendous growth in online sales, there is significant evidence revealing purchase abandonment (i.e., consumers' search on retailer websites not leading to actual purchase behaviour) of a vast number of consumers. A Boston Consulting Group study found that 28 percent of consumers' purchase efforts failed when they could not find the products they wanted, could not finish their transactions, or did not complete their purchases to their satisfaction [1]. These statistics hint the importance of consumers' online search experiences prior to their online purchasing behaviours [2].

Businesses today have started marketing products online and this is becoming a growing trend. Due to this, many businesses want to know the prospective audience for their product. It's imperative for them to measure the interest of a customer in their product. Purchase intentions are frequently measured and used by marketing managers as an input for decisions about new and existing products and services. Conventionally, companies use customer survey forms in which they ask questions regarding the purchase of product in a stipulated time frame and using that information the purchase intention is measured. The surveys are effective, but they are limited in many ways such as lacking diversity, small sample size and high resource requirement. This approach can be modernized by using the data from social media to mitigate the need to conduct the surveys for information retrieval. This can prove the importance of social media such as Twitter as a powerful tool for marketers in making the decision to target a customer. The motivation behind this study is to find the purchase intention of a customer towards a product and quantify it. This will help the marketers to find out the potential customers base so that a tailored marketing campaign could be designed accordingly to increase their sales. One way to achieve this is by developing a data-based approach that identifies potential customers for a product using tweets from Twitter. Using the field of text analytical, one can employ a machine learning approach

using the tweets data. This can be done by using text mining and natural language processing algorithms which help in identifying patterns and trends in data.

The problem of purchase intention prediction is closely linked to sentiment analysis. Both are textual classification problems of a binary nature. However, it is worth noting that sentiments are opinions that usually contain words with a polarity, negative or positive. Texts indicating purchase intent may not have a polarity at all. Instead, they usually contain action phrases (verbs) indicating a desire to obtain something. Therefore, we have borrowed from the literature on sentiment analysis to tackle our particular problem.

2 Related Work

Multiple research studies have tried to tackle the problem of Purchase Intention prediction using distinct approaches. We believe the first known attempt in literature comes from Ramanand et al. publication in which a corpus from popular consumer review sites was prepared [3]. They used a lexicon rule-based approach to predict purchase intentions from product reviews. Hamroun et al. deployed a semantic pattern-based approach for customer intention [4]. Tools like OpenNLP and WordNet for Part-of-Speech (PoS) tagging for tweets and building ontological representations of words. These ontological representations along with PoS tags were used to match patterns and make predictions for intent.

Oele attempted to identify purchase intentions using knowledge-rich, knowledge-poort and their combinations to train different machine learning models. The models were tested using 10-fold cross validation. The best model was capable of predicting the potential customer with 90% of accuracy [5]. Korpusik et al. investigated deep learning methods for predicting purchase intentions from twitter data [6]. However, their problem set was dissimilar to this study because their data was limited to twitter users who eventually tweeted after purchasing the initial product. They also added constraints for collecting data. Their paper makes a “will buy” or “will not buy” prediction instead of predicting an inclination to buy.

As discussed earlier, our problem can be considered a derivative of the task of sentiment analysis. Therefore, we have also reviewed studies pertaining to sentiment analysis on twitter. Go et al. first approached the problem of sentiment analysis from twitter data. They converted the problem to a self-supervised task with noisy labels. SVM, Naïve Bayes and Max Entropy algorithms were used along with n-gram features [7]. Gamallo et al. also used a Naïve Bayes classifier but performed additional feature engineering steps such as PoS tagging and Negation handling. They also removed neutral tweets using a polarity lexicon [8]. Eshak et al. used the term s-commerce to define commerce platforms utilizing online social media platforms. They compared lexicon and ML based approaches, proving the superiority of ML methods [9]. Recently, twitter data was analysed for negative purchase intention using binary logistic regression. The model developed in this study had better F1 score as compared to a number of state of the art machine learning approaches [10]. Sharma and Shafiq [11] developed an artificial intelligence

based user intention assessment model. The model displayed high precision, accuracy and F1 score in determining different intents of online users based on reviews.

This research work proposes a deep leaning model to predict the purchase intention of an online user. The proposed model displays high accuracy when applied to twitter data.

2.1 Related Models Description

After extensive research, five most frequently used textual analytics models were selected for this study. The Scikit-learn library in python was used and the models were configured according to the dataset.

1. Support Vector Machine (SVM): A linear classification algorithm. It works by creating a decision boundary between the target classes from the feature set. This is a hyperplane that maximizes the margin between the target classes. The kernel function transforms features to make the classes linearly separable.

2. Naive Bayes: A classifier algorithm that uses Bayes Theorem to predict target classes from features. Conditional independence is assumed between all features. This algorithm performs very fast. Given our priors, the likelihood is maximized for posteriors.

3. Logistic Regression: A modified version of linear regression to solve for classification problems. It uses the Logistic (Sigmoid) function to form a decision boundary between classes. A cost function is set up which can be optimized using MLE by minimizing the Cross Entropy Loss.

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. Multi-Layer Perceptron: It is a deep learning machine algorithm, which is arranged in a layer of neurons. 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.

6. Long Short-Term Memory (LSTM): A type of neural network that is able to take sequential data as input and maintain contextual (temporal) information from features. It is an improvement from the Recurrent Neural Network (RNN) architecture which had problems with long sequence inputs (deeper layers) [12, 13].
In the following sections, the approach used in the paper is described and explained in detail. As shown in Figure 1 we start with the section of data collection, then explain the data pre-processing steps. The next section talks about document vector creation and the advantages and disadvantages of the approaches.

3 Methodology

3.1 Proposed Approach

In the following sections, the approach used in the paper is described and explained in detail. As shown in Fig. 1 we start with the section of data collection, then explain the data pre-processing steps. The next section talks about document vector creation and the advantages and disadvantages of the approaches.

3.2 Data Preparation

This section describes in detail the proposed approach to tackle the purchase intention detection issue. The data collection and annotation processes are described before explaining the data pre-processing and transformation as per the requirements of analytical models.

3.2.1 Data Collection and Annotation

Due to the unavailability of annotated Twitter corpora regarding the detection of purchase intention, manually annotated data set was created for this study. The product picked for our task was the iPhone X. The tweets were downloaded using a web crawler developed by JohnBakerFish [14] which crawled the website to collect the data [15]. Subsequently 3,200 randomly picked tweets were manually annotated using the basic criterion defined in Table 1. Each tweet was assigned a label from two people. In case of differing judgement, a third person reviewed the tweet before assigning a final label.

Table 1: Criteria for Labelling of tweets

Tweet	Class
Comparing iphone x with another phone and telling other phone are better	No PI
Talking about good features of iphone x	PI
Talking about negative features of iphone x	No PI
liked video on Youtube about iphone x	PI

Purchase intention (PI) can be defined as a situation where a consumer is willing and intends to make online transactions [16]. For the purpose of this research paper a purchase intention shown by a user in a Twitter tweet is represented by the tweet having action words like buy, want, and desire associated with the product the user intends to buy. After the annotation process, we had 661 tweets with Purchase Intention and 457 tweets with No Purchase Intention. The remaining tweets were too ambiguous or neutral.

3.2.2 Data Pre-processing

Figure 2 shows how the tweets were processed using the following techniques in sequential order. At first, the text of the tweets was converted into lower case. Word correction was applied to all tweets to remove spelling errors. URLs were also removed from all tweets as they would not be contributing towards the model's learning. Contractions like can't and you're were converted to their full form. The text was then passed through punctuations and special characters filters, as the text may contain unwanted special characters, spaces, tabs, and etcetera which have no significance in text classification. In some cases, the

tweets had negation modifiers, like the word “not” which modifies the meaning of the associated word. Handling Negation Modifiers is important as they may modify the sentiment conveyed by the associated word [17]. The negation modifiers were handled by appending a custom defined keyword “NOT_” to each word appearing between a negation and following punctuation [18]. Negation modifiers were only used for bag-of-word approaches.

Next step was stop-words removal since the tweets also contain frequent words which are part of the regular sentence and grammar but do not contribute to the meaning of the sentence. The removal of stop words like “the”, “a” and “an” also reduces the dimensions of the vocabulary. The words “phone” and “x” had the most occurrence and were independent of purchase intention, so they were removed from the corpora. Finally, the tweets were subjected to stemming. Stemming reduces a word to a root form by removing inflection through dropping unnecessary characters, usually a suffix. There are several stemming models such as Porter and Snowball [19]. For this study, Porter's Stemmer is used which is available with NLTK, a library of Python [20]. Some of the experiments were conducted with lemmatization. Lemmatization, unlike Stemming, also ensures that the root word belongs to the language. In Lemmatization, the root word is called lemma. A lemma is the canonical, dictionary, or citation form of a set of words [21]. Lastly, the tweets were broken down into word tokens using NLTK’s tokenizer which utilizers the TreeBank tokenizer algorithm.

3.3 Formation of Document Vectors

In order to pass the tweets to the model, they had to be converted to a numerical representation. A classical technique used was BOW (Bag of Words) representation. Bag of Words takes all unique words from the corpus to form a vocabulary. Words are assigned a sequence number and used as the index of the vector. In binary BOW, a one-hot encoding approach is taken and if the word occurs in the document, a 1 is marked. Otherwise, 0 stays. We used Count BOW instead of binary BOW which just marks the words’ index position with its number of occurrences in a document, instead of a 1. TF-IDF (Term Frequency – Inverse Document Frequency), the second BOW representation used, works similarly. Instead of assigning a 1 or count at the index position of the word, a ratio is assigned. Term Frequency is the number of occurrences of a word in a document, divided by the total number of words in said tweet. Inverse Document Frequency is the number of documents where a word occurs, divided by total number of documents. This ratio has an additional benefit of assigning smaller values to words that are common among all documents. These words usually do not provide any distinction capability [22].

The drawback of using classic techniques mentioned above is that they do not capture any contextual information in their word embeddings. No information regarding a word’s position is used and the words are clumped together in no specific order, hence the name Bag of Words. Using n-grams some-what mitigates this problem, but it has tremendous computational and practical limitations. Another approach to generate word representations is to use a co-occurrence matrix of word pairs (or windows) in the corpus [23]. Dimensionality reduction (Principal Component Analysis, Singular Value Decomposition) can be used to reduce the size of the matrix. While this approach can work, we decided to opt for a more complex approach (GloVe) that enhances this method.

A more modern approach is using distributed representation models like Word2Vec and GloVe to generate word embeddings in high dimensions. Both models work by taking a sequence vector (one-hot encoding of document) as input. The input vector undergoes multiplication with an embedding matrix of the form m x n where m is the vocabulary size and n are the number of dimensions specified to encode the word, a parameter that we can set. The output is a vector for each input word in the document. This is a tremendous improvement from the heavily sparse matrix that would have been created had we used a bag of words approach. Additionally, these embeddings maintain contextual information that should help the model capture more complex patterns and perform better.

Word2Vec embeddings come from a Neural Net that is trained to predict a word from a context window (Continuous Bag of Words) or predict context from a word (skip-gram).

GloVe embeddings come from a log bilinear regression model that is trained to generate embeddings such that they are associated with the ratio of co-occurrence matrix for word and context.

Since these models are trained to predict words from context (skip-gram predicts context words), the resulting vector representations hold semantic as well as syntactic information [24].

We used both Word2Vec and GloVe based word embeddings to generate word vector representations. Since these techniques generate vectors on a word level and not a document level, we were left with few options. The first option was to perform an average of all word vectors within a tweet and use that as an input feature to our Machine Learning Model. This was not promising. The second option was to flatten all word vectors in each tweet to become one long vector. This would lead to a curse of dimensionality and degraded performance of models. Positional information would be lost as well. The last option was to deploy a deep learning model of Recurrent (RNN) Architecture. These models are designed to take input sequences concurrently and are heavily utilised for NLP tasks. This allowed us to feed the embeddings of each tweet directly to the model for classification. We used the LSTM (Long Short Term Memory) network instead of the standard RNN as it overcomes the problems of vanishing and exploding gradients [13]. LSTM are also more capable of learning longer term dependencies.

The inputs for LSTM were generated sequence tokens of a fixed size of 55. Right padding was done so that all tweets had the same size.

3.4 Modelling

The problem of identifying purchase intent is one of classification. Therefore, we utilized supervised classification models to predict intent. Five most frequently used models namely, Support Vector Machine, Naïve Bayes, Logistic Regression, Decision Tree and Artificial Neural Network, were trained and tested using the cleaned data [25]. LSTM network was used for creating models from word embeddings of Word2Vec and GloVe.

3.5 Machine Learning Approaches

The first model used was the multinomial Naive Bayes classifier. The classifier was configured with Laplace smoothing to prevent zero probabilities for features missing from the training samples. The Prior probabilities of the features were used rather than using a uniform prior probability.

The second model used was the Support Vector Machine classifier. A Linear SVM was implemented with the penalty of an error set to 1 and trained using cross-validation.

The third model was the Logistic Regression having the inverse of regularization strength coefficient set to 1 for stronger regularization and the maximum number of iterations to convergence set to 100. For optimization of the model the liblinear algorithm was used, as it is best suited for small datasets.

The Decision Tree classifier was also tested. Gini Index was used to determine the quality of each split. The highest accuracy was obtained by splitting the internal node using at least seven samples.

The Multi-Layer Perceptron model was the last ML model, and it was optimized using stochastic gradient descent (SGD) with Cross Entropy loss. Rectified linear unit (ReLU) was used as the activation function for the hidden layer, as it is faster to compute [26]. It also converges faster and is not prone to vanishing gradients unlike sigmoid and tanh activations. The learning rate schedule for weight updates was kept constant. Four hidden layers with the first hidden layer having 50 neurons, the second one having 20 neurons, the third one having 10 neurons and the last hidden layer having 5 neurons. The output layer had only one node for binary classification. Training was done for 50 epochs.

3.6 Deep Learning Approaches

For the Word2Vec model, an embedding layer was used with 300 embedding dimensions. For the GloVe model, an embedding layer with 200 dimensions was used [27]. Both embedding layers are from the standard pre-trained models that were trained on corpus. Pre-trained models are especially helpful as they are supposed to contain more accurate representations of words and sentences.

An LSTM layer was added on top with 32 units and a dropout of 0.2 to prevent overfitting. Sigmoid activation was used for the output layer as the model was binary classification. Model was trained using a cross entropy loss function and the optimizer used was stochastic gradient descent.

4 Experimentation And Results

Two validation techniques were used. Using hold-out validation is the go-to method of choice but k-fold validation minimizes the chance of variance in the model. Five-fold cross validation was used on ML Models and average of evaluation metrics were reported. Since distribution of classes was slightly skewed, stratified cross validation was done to maintain class ratio.

Hold-out validation was done on all models with a split of 30% validation and 70% train dataset. Results were summarized using ROC Curve as well as other evaluation metrics.

TP: True Positive *TN*: True Negative *FP*: False Positive *FN*: False Negative

1. Accuracy

$$\frac{TP + TN}{TP + TN + FP + FN}$$

2. F-Measure

$$\frac{2 \times Precision \times Recall}{Precision + Recall}$$

3. Area Under ROC Curve (AUC)

(Receiver Operator Characteristic)

Table 2: Experiment Results

K-fold Validation

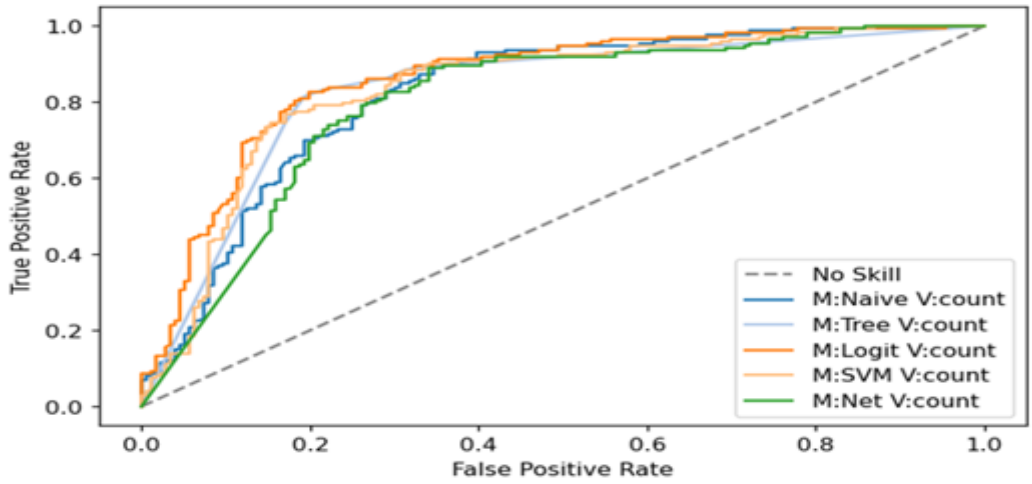
Vectorizer	Model	Accuracy	AUC	F-Score
Count BoW	**Naïve Bayes**	76	78	81.5
	Neural Net	74.1	76.2	79.2
	Support Vector Machine	74.6	76.6	79.9
	Decision Tree	73.9	75.3	79.1
	Logistic Regression	74.7	75.75	79.88
TF-IDF BoW	Naïve Bayes	73.7	76.9	79.4
	Neural Net	74.3	77	79.8
	Support Vector Machine	74.5	77.5	80
	Decision Tree	73.7	76.3	79.3
	Logistic Regression	74.2	77	79.8
Word2Vec	LSTM	84.8	90	87.3
GloVE	**LSTM**	**86.4**	**91**	**88.7**

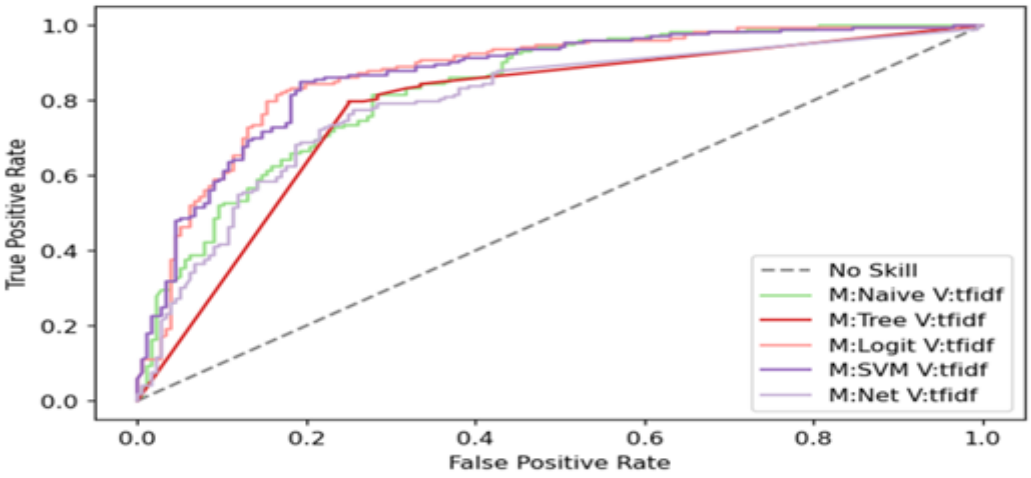
Hold-out Validation

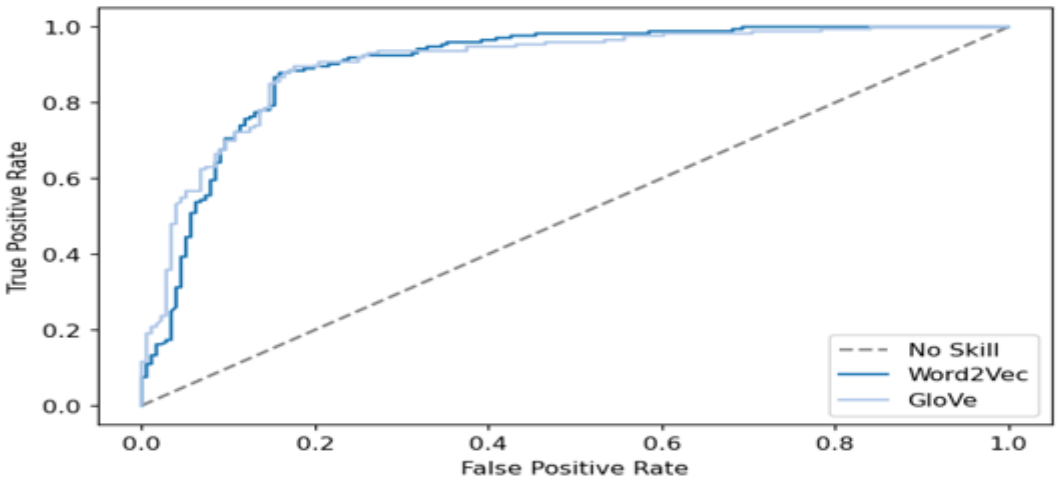
Vectorizer	Model	Accuracy	AUC	F-Score
Count BoW	Naïve Bayes	75.9	82.2	79.3
	Neural Net	77.1	83	78.7
	Support Vector Machine	77.4	83.7	79.3
	Decision Tree	77.2	82.7	79.6
	Logistic Regression	77.5	83.8	79.6
TF-IDF BoW	Naïve Bayes	74.6	83	77.6
	Neural Net	77.1	83	77.4
	Support Vector Machine	75	83.4	77.8
	Decision Tree	74.8	82.3	77.5
	Logistic Regression	74.8	82.9	77.6
Word2Vec	LSTM	84.8	90.1	85.2
GloVE	**LSTM**	**85.7**	**90.3**	**86.1**

When using the bag of words technique, our models did not show much deviation around 75% accuracy. Using word embeddings significantly improved the accuracy to around 85%. The best performing model was the LSTM recurrent network with Global Vector embeddings.

Table 3: Experiment Results ROC Curve







The ROC Curve is a visual representation of our models' performances. We can observe that the embedding classifiers have the greater area under the curve. They are also most distant to the 45-degree diagonal (No Skill line) [28].

5 Limitations

The data had to be scrapped from twitter using a web scraper due to the unavailability of a public repository regarding twitter purchase intention data. Besides that, the manual annotation of the tweets was very time consuming and consequently limited the final data size.

Limited annotated data: Since we had to manually annotate each tweet in the dataset and this process takes a lot of time, we were only able to annotate about 3,200 tweets.

Narrow Product Category: Our dataset only considered purchase intent for a specific product, the iPhone X. Broadening the product category would allow the model to generalize better and be used for purchase intention prediction on a wider set of products. It can even be extended to non-tangibles like services.

6 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 any publicly available dataset for purchase intention based on twitter corpora.

Looking at the other research that is done in the similar field, our project also stands apart since we have implemented 6 different models and after evaluating them, we choose the best one customized to the product data.

We were not able to get more than 86% accuracy because of the two problems highlighted above. To achieve even 86% accuracy with a small dataset is a victory.

7 Future Work

To continue our work forward, it is worth trying out the dataset on transformer architecture models such as BERT. This architecture introduces an attention mechanism that produces more meaningful word embeddings. BERT also deploys a sub-word tokenization technique called Byte Pair encoding which is superior to the word-based tokenization technique called TreeBank which we used.

It is also worth considering to use of sentence vectors instead of word vectors. Algorithms like Doc2Vec could potentially provide better accuracy.

Explainable AI is gaining a lot of traction these days. It is possible to run model interpretability techniques to determine the keywords and phrases that are driving our model's prediction.

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 likelihood to purchase the product.

Declarations

Declarations of interest: None

Funding:

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

1. Shop.org & Boston Consulting Group, “State of Online Retailing 3.0.,” *Washington, D.C.: National Retail Federation.*, 2000. http://www.shop.org
2. J. Kim, H. Lee, and H. Kim, “Factors affecting online search intention and online purchase intention,” *Seoul J. Bus.*, vol. 10, 2004.
3. J. Ramanand, K. Bhavsar, and N. Pedanekar, “Wishful thinking-finding suggestions and’buy’wishes from product reviews,” in *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, 2010, pp. 54–61.
4. M. Hamroun, M. S. Gouider, and L. B. Said, “Customer intentions analysis of twitter based on semantic patterns,” in *The 11th international conference on semantics, knowledge and grids*, 2015, pp. 2–6.
5. M. J. A. Oele, “Identifying Purchase Intentions by Extracting Information from Tweets,” 2017.
6. M. Korpusik, S. Sakaki, F. Chen, and Y.-Y. Chen, “Recurrent Neural Networks for Customer Purchase Prediction on Twitter.,” *CBREcsys Recsys*, vol. 1673, pp. 47–50, 2016.
7. A. Go, L. Huang, and R. Bhayani, “Twitter sentiment analysis,” *Entropy*, vol. 17, p. 252, 2009.
8. P. Gamallo and M. Garcia, “Citius: A Naive-Bayes Strategy for Sentiment Analysis on English Tweets.,” in *Semeval@ coling*, 2014, pp. 171–175.
9. M. I. Eshak, R. Ahmad, and A. Sarlan, “A preliminary study on hybrid sentiment model for customer purchase intention analysis in socialcommerce,” in *2017 IEEE conference on big data and analytics (ICBDA)*, 2017, pp. 61–66.
10. S. Atouati, X. Lu, and M. Sozio, “Negative purchase intent identification in Twitter,” in *Proceedings of The Web Conference 2020*, 2020, pp. 2796–2802.
11. A. Sharma and M. O. Shafiq, “A Comprehensive Artificial Intelligence Based User Intention Assessment Model from Online Reviews and Social Media,” *Appl. Artif. Intell.*, pp. 1–26, 2022.
12. D. Kumar, H. D. Mathur, S. Bhanot, and R. C. Bansal, “Forecasting of solar and wind power using LSTM RNN for load frequency control in isolated microgrid,” *Int. J. Model. Simul.*, vol. 41, no. 4, pp. 311–323, 2021.
13. C. Olah, “Understanding LSTM Networks–colah’s blog,” *Colah Github Io*, 2015.
14. Jon, “TweetScraper.” Nov. 18, 2022. Accessed: Sep. 17, 2019. [Online]. Available: https://github.com/jonbakerfish/TweetScraper

15. K. Crystal, “Scraping Twitter with TweetScraper and Python,” Jun. 11, 2019. https://medium.com/@kevin.a.crystal/scraping-twitter-with-tweetscraper-and-python-ea783b40443b (accessed Jun. 08, 2021).
16. P. A. Pavlou, “Consumer acceptance of electronic commerce: Integrating trust and risk with the technology acceptance model,” *Int. J. Electron. Commer.*, vol. 7, no. 3, pp. 101–134, 2003.
17. K. V. Ghag and K. Shah, “Negation handling for sentiment classification,” in *2016 International Conference on Computing Communication Control and automation (ICCUBEA)*, 2016, pp. 1–6.
18. D. Jurafsky and J. H. Martin, “Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.”
19. K. Fortney, “Pre-Processing in Natural Language Machine Learning,” *Medium*, Nov. 29, 2017. https://towardsdatascience.com/pre-processing-in-natural-language-machine-learning-898a84b8bd47 (accessed Sep. 17, 2019).
20. “NLTK :: Natural Language Toolkit.” https://www.nltk.org/ (accessed Jun. 08, 2021).
21. H. Jabeen, “Stemming and Lemmatization in Python.” https://www.datacamp.com/tutorial/stemming-lemmatization-python (accessed Sep. 17, 2019).
22. D. Munteanu, “Vector space model for document representation in information retrieval,” *Ann. Dunarea Jos*, pp. 43–44, 2007.
23. D. Jurafsky, “Vector Semantics. 2019,” *URl Httpsweb Stanf. Edu~ Jurafskyli15lec3 Vector Pdf*.
24. T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *ArXiv Prepr. ArXiv13013781*, 2013.
25. A. Singh, N. Thakur, and A. Sharma, “A review of supervised machine learning algorithms,” in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2016, pp. 1310–1315.
26. “WordNet.” https://wordnet.princeton.edu/ (accessed Nov. 18, 2022).
27. T. Shi and Z. Liu, “Linking GloVe with word2vec,” *ArXiv Prepr. ArXiv14115595*, 2014.
28. T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006, doi: 10.1016/j.patrec.2005.10.010.

Figures

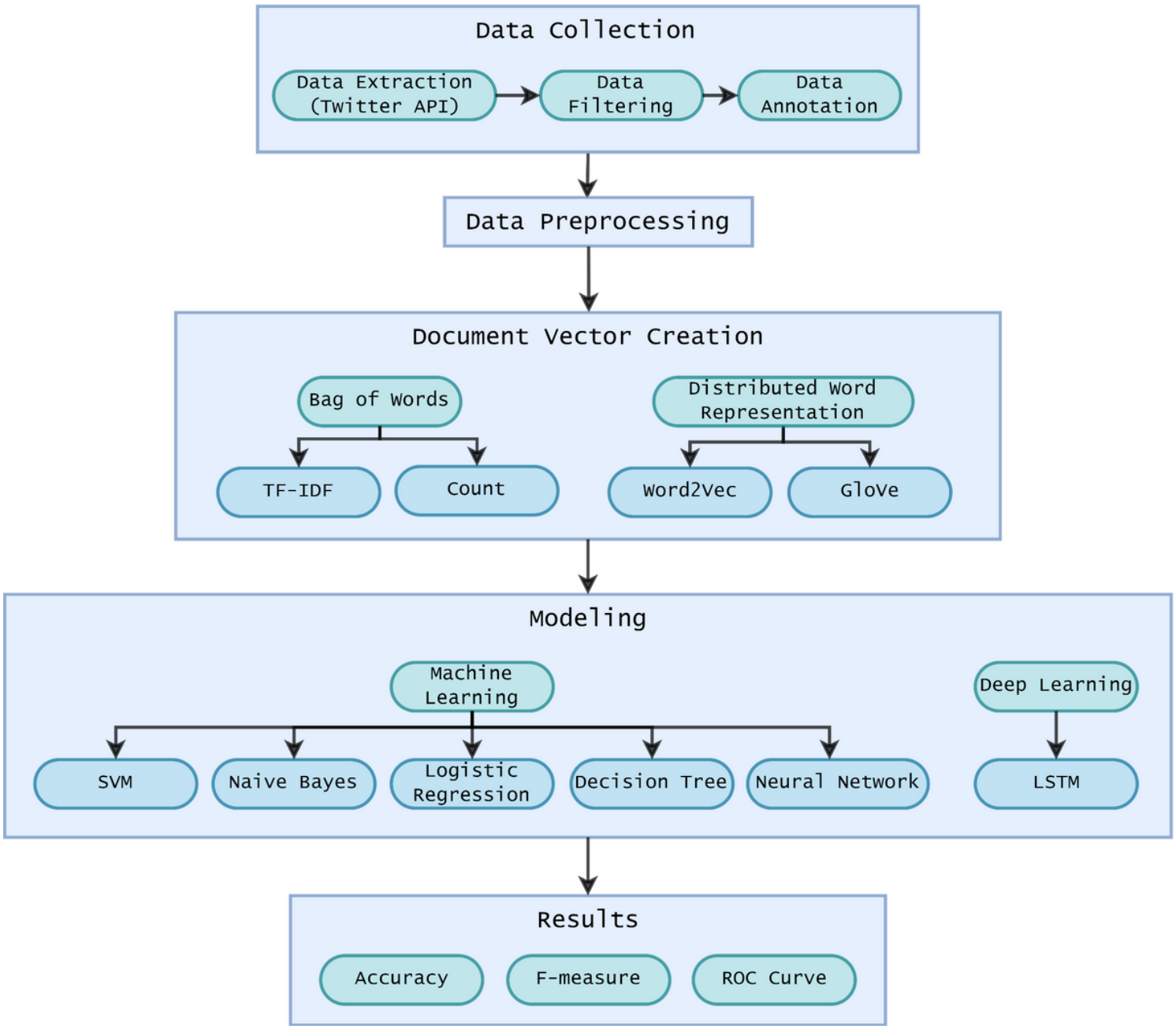


Figure 1

Methodology

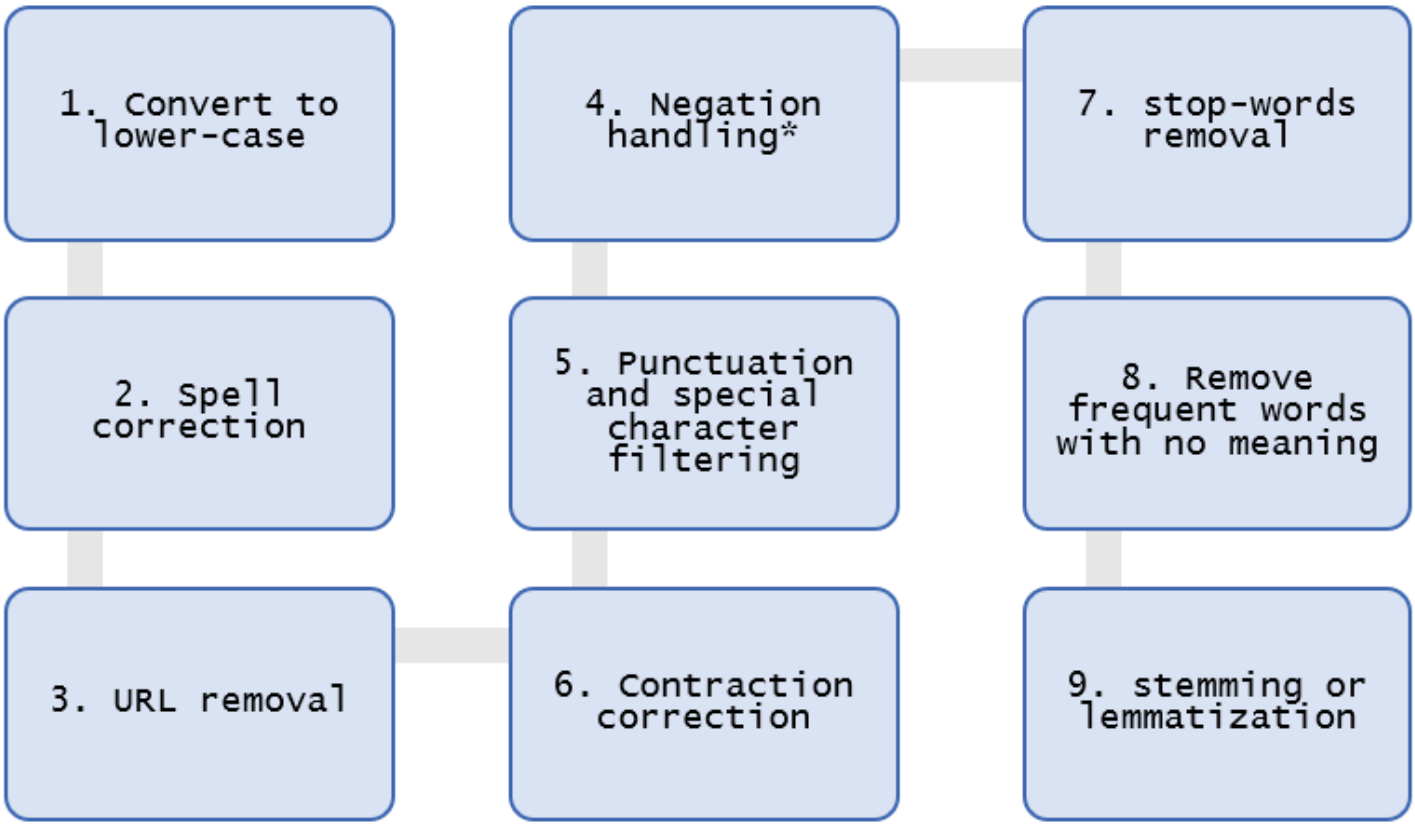


Figure 2

Data Pre-processing Steps