Fake Reviews Detection using Supervised Machine Learning Algorithms

Prof. Aditee Salokhe, Sanket Kharat, Rutvik Galande, Shripad Jadhav, Pratik Dhole, Pranav Dahatonde

Abstract

In recent years, the prevalence of fake reviews on online platforms has become a significant concern, as these deceptive reviews can mislead consumers and impact purchasing decisions. This research paper explores various methods for detecting fake reviews using machine learning techniques. We utilized the deceptive opinion spam dataset, which includes both truthful and deceptive hotel reviews for 20 Chicago hotels. The dataset comprises 1600 reviews, evenly split between truthful positive, deceptive positive, truthful negative, and deceptive negative reviews. Our primary objective was to classify these reviews as either truthful or deceptive using several machine learning algorithms.

We constructed a data frame with columns for the review text, polarity class, and spamity class. Polarity indicates whether a review is positive or negative, while spamity distinguishes between truthful and deceptive reviews. Stopwords were removed from the reviews using the nltk package from sklearn, and text mining techniques were applied to convert text strings into numerical data. We also extracted parts of speech from the reviews to use as features in our models.

We experimented with four classification techniques: Naïve-Bayes, Support Vector Machine (SVM), Decision Tree, and Random Forest classifiers. The Naïve-Bayes classifier, specifically the Multinomial NB algorithm, achieved an accuracy of 89.13%. The SVM yielded an accuracy of 82.155%, while the Decision Tree algorithm resulted in an accuracy of 65.55%. The Random Forest classifier demonstrated the highest accuracy, reaching 91.72%. Confusion matrices, generated using the sklearn metric module, validated the accuracy of each algorithm.

Given the superior performance of the Random Forest classifier and Naïve-Bayes, these models were selected for further analysis. Our findings indicate that these machine learning techniques can effectively identify fake reviews, thereby helping to mitigate their misleading impact on consumers.

For future work, we aim to expand our study to include datasets from other platforms such as Amazon and flipakrt, and to explore different feature selection methods. We also plan to apply sentiment classification algorithms using various tools like Python, R, Statistical Analysis System (SAS), and Stata, to detect fake reviews and evaluate the performance of these tools.

This research was supported by the Technical University of Kerala. We extend our gratitude to our colleagues for their expertise, which significantly aided the research, although they may not concur with all the interpretations presented in this paper. Through this study, we contribute to the ongoing efforts to enhance the reliability of online reviews and protect consumers from deceptive practices.

Introduction-

A fake review involves the exploitation of the user review system by false personas. Additionally, fake reviews can be generated by automated bots. These misleading reviews deceive customers into making uninformed decisions about products, leading to unnecessary expenditures. The reviews can be either positive or negative, aiming to either boost sales and promotion or undermine the products of competitors. Many individuals rely on online reviews to decide whether to purchase a product. Consequently, numerous companies utilize various applications that leverage machine learning to identify fake reviews.

In this research, we employ Sentiment Analysis to process the data. Sentiment Analysis typically involves classifying sentiments as either positive or negative. The core of Sentiment Analysis lies in determining the polarity of a given text or document. In this study, we categorize sentiments as either negative or positive.

Advancements in fields such as Natural Language Processing (NLP) have significantly enhanced the accuracy of understanding people's sentiments, emotions, and behaviors. Emotions like joy, anger, surprise, and disgust can be extracted from reviews. For instance, when booking a hotel, potential customers often check online reviews to gauge past customer experiences. These reviews greatly influence customer decisions. This application aims to detect potential fake reviews to minimize misleading information.

Various machine learning algorithms can be employed for classification and prediction in fake review detection. In our study, we use the Naive-Bayes classifier, Support Vector Machine (SVM), Random Forest Classifier, and Decision Tree to predict reviews. We aim to identify fake positive, fake negative, true positive, and true negative reviews and compare the accuracy of each algorithm. The primary goal of this paper is to classify reviews into true and fake categories using machine learning techniques.

Related Work-

Several studies and experiments have been conducted using various sample data sets. Numerous product reviews have been scraped from product webpages and analyzed in these studies. For our research, we have chosen to use the deceptive opinion spam dataset.

We extracted data from this dataset and stored it in a list before creating a data frame with corresponding labels. Sentiment analysis was then applied to all the reviews to determine their polarity, classifying them as either Positive or Negative. Additionally, we categorized the reviews as True or Deceptive. The polarity and Spamity classes were converted into binary values (0s and 1s) for further processing.

In our approach, we primarily utilized three methods: Naive Bayes classification, Support Vector Machine (SVM), and Decision Tree. Naive Bayes is a classification algorithm suitable for both binary (two-class) and multi-class classification problems.

Proposed work

We are utilizing the deceptive opinion spam dataset, which comprises both truthful and deceptive hotel reviews from 20 Chicago hotels.

This corpus includes 400 truthful positive reviews sourced from Trip Advisor, 400 deceptive positive reviews generated via Mechanical Turk, 400 truthful negative reviews from platforms such as Expedia, Hotels.com, Orbitz, Priceline, Trip Advisor, and Yelp, and 400 deceptive negative reviews also from Mechanical Turk.

In total, the dataset consists of 1600 reviews. Our objective is to employ a machine learning algorithm to classify these hotel reviews as either truthful or deceptive.

A.Preprocessing-

We constructed a data frame with three columns: reviews, polarity class, and spamity class. The reviews column contains the text of customer reviews, while the spamity class indicates whether a review is deceptive or true. The polarity class signifies whether the sentiment of the review is positive or negative.

First, we removed stopwords from the reviews using the nltk package from sklearn. We applied text mining techniques to convert the text strings into numerical representations. Additionally, we extracted parts of speech from the reviews to serve as feature inputs for the model. The reviews were stored in an array format.

The spamity class and polarity class columns were converted from True/False to binary values (0s and 1s) to facilitate model training. Given the dataset comprises only 1600 rows, we split the data into an 80:20 ratio for training and testing, and stored it as an array.

B.Model selection-

To fit our model, we utilized the sklearn library in Python, which offers essential tools for implementing classifiers. In machine learning, various classification techniques are available, such as Naïve-Bayes, Support Vector Machine (SVM), Decision Tree, and Random Forest classifiers. We experimented with these different methods to achieve the most accurate model.

The Random Forest algorithm is an ensemble model that constructs decision trees on data samples and aggregates predictions from each tree through a voting mechanism to determine the best solution. This approach often yields high accuracy and is applicable for both classification and regression tasks.

Naïve-Bayes is commonly used for text categorization, leveraging word frequencies as features to predict text categories. It typically employs a bag-of-words representation from Natural Language Processing (NLP) to identify patterns in text data.

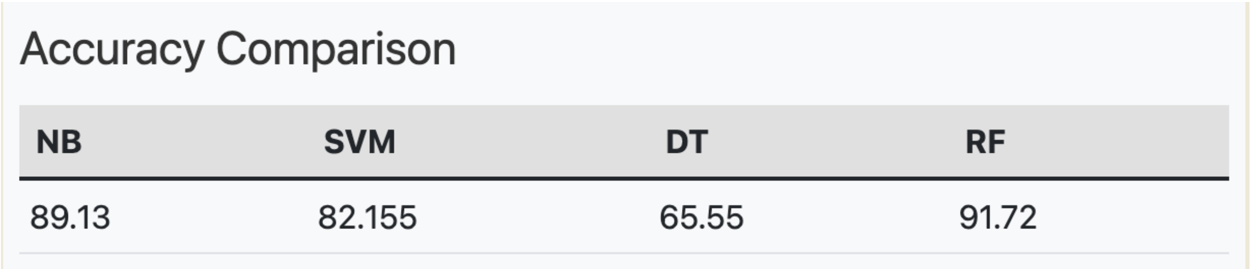
SVMs excel in performing non-linear classification by utilizing the kernel trick, which implicitly maps input data into high-dimensional feature spaces. For our SVM classifier, we kept the gamma parameter constant to achieve a well-fitted model.

Results

A.Machine used-
Our experiments were conducted on a machine with the following specifications: an Intel Core i5 11400H processor with a 3GHz CPU, 8GB of RAM, and a 64-bit Windows operating system. Python was the programming language used, along with the sklearn, numpy, and pandas packages. The development environment for the project was Spyder 4.1.1.

B.Results-
We utilized the Naïve Bayes classifier, Support Vector Machine (SVM), Decision Tree, and Random Forest classifiers to categorize the reviews dataset. This dataset, consisting of 1600 rows and three columns—reviews, polarity, and spamity—was split into training and testing sets in an 80:20 ratio for each classification process.

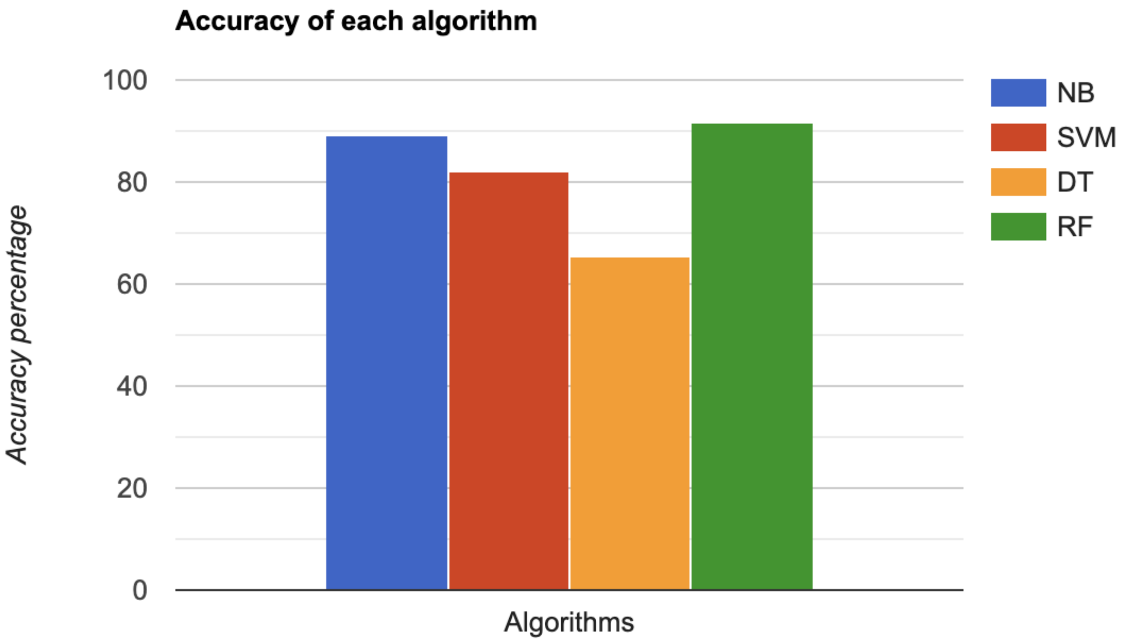
For the Naïve-Bayes classification, we applied the Multinomial NB algorithm. Upon fitting the model, the Naïve-Bayes classifier achieved an accuracy of 89.13%. The SVM achieved an accuracy of 82.155%, while the Decision Tree algorithm resulted in an accuracy of 65.55%. Comparing the performance of each algorithm, we found that the Random Forest classifier provided the highest accuracy, followed by the Naïve-Bayes classifier, with the Decision Tree classifier yielding the lowest accuracy.



Accuracy Comparison

NB	SVM	DT	RF
89.13	82.155	65.55	91.72

C.Analysis-
We opted to use both the Random Forest Classifier and Naïve-Bayes as our models since they provided the highest accuracy. The Random Forest Classifier, in particular, improved performance accuracy to 91.72 percent, the highest among all the techniques tested. By importing the metric module from the sklearn package, we were able to generate confusion matrices for the predictions, which confirmed the high accuracy of each algorithm.



Conclusion and future work-

In this paper, we proposed several methods to analyze a dataset of hotel reviews and introduced sentiment classification algorithms to apply supervised learning to this dataset.
For future work, we plan to extend our study to include other datasets, such as those from Flipkart or Amazon, and explore different feature selection methods. Additionally, we aim to apply sentiment classification algorithms to detect fake reviews using various tools such as Python, R or R Studio, Statistical Analysis System (SAS), and Stata. We will then evaluate the performance of our work using these tools.

Acknowledgement

This research was supported by the DY Patil College of Engineering and Technology,Kasba Bawda. We extend our gratitude to our colleagues for their expertise, which significantly aided the research, although they may not concur with all the interpretations presented in this paper.

References-

[1] J. W. Pennebaker, M. E. Francis, and R. J. Booth, “*Linguistic Inquiryand Word Count: Liwc,*” vol. 71, 2001.

[2] S. Feng, R. Banerjee, and Y. Choi, “*Syntactic stylometry for deceptiondetection*,” in Proceedings of the 50th Annual Meeting of the Associationfor Computational Linguistics: Short Papers, Vol. 2, 2012.

[3] J. Li, M. Ott, C. Cardie, and E. Hovy, “*Towards a general rule foridentifying deceptive opinion spam*,” in Proceedings of the 52nd AnnualMeeting of the Association for Computational Linguistics (ACL), 2014.

[4] E. P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, “*Detectingproduct review spammers using rating behaviors,*” in Proceedings ofthe 19th ACM International Conference on Information and KnowledgeManagement (CIKM), 2010.

[5] J. K. Rout, A. Dalmia, and K.-K. R. Choo, “*Revisiting semi-supervisedlearning for online deceptive review detection,*” IEEE Access, Vol. 5,pp. 1319–1327, 2017