



Article

Leveraging Stacking Framework for Fake Review Detection in the Hospitality Sector

Syed Abdullah Ashraf ^{1,2,*} , Aariz Faizan Javed ¹, Sreevatsa Bellary ¹, Pradip Kumar Bala ¹
and Prabin Kumar Panigrahi ³

¹ Department of Information Systems & Business Analytics, Indian Institute of Management Ranchi, Prabandhan Nagar, Nayasarai Road, Ranchi 835303, Jharkhand, India; aarizfaizan.javed18fpm@iimranchi.ac.in (A.F.J.); sreevatsa.b22ph@iimranchi.ac.in (S.B.); pkbala@iimranchi.ac.in (P.K.B.)

² Department of Analytics & Operations, Delhi School of Business, Outer Ring Rd, AU Block, Jal Board Colony, Pitampura, New Delhi 110034, Delhi, India

³ Department of Information Systems, Indian Institute of Management Indore, Prabandh Shikhar, Rau-Pithampur Road, Indore 453556, Madhya Pradesh, India; prabin@iimind.ac.in

* Correspondence: syedabdullah.ashraf18fpm@iimranchi.ac.in

Abstract: Driven by motives of profit and competition, fake reviews are increasingly used to manipulate product ratings. This trend has caught the attention of academic researchers and international regulatory bodies. Current methods for spotting fake reviews suffer from scalability and interpretability issues. This study focuses on identifying suspected fake reviews in the hospitality sector using a review aggregator platform. By combining features and leveraging various classifiers through a stacking architecture, we improve training outcomes. User-centric traits emerge as crucial in spotting fake reviews. Incorporating SHAP (Shapley Additive Explanations) enhances model interpretability. Our model consistently outperforms existing methods across diverse dataset sizes, proving its adaptable, explainable, and scalable nature. These findings hold implications for review platforms, decision-makers, and users, promoting transparency and reliability in reviews and decisions.

Keywords: fake review detection; machine learning; SHAP; fake review identification; review spam identification; hospitality; ecommerce; Yelp



Citation: Ashraf, S.A.; Javed, A.F.; Bellary, S.; Bala, P.K.; Panigrahi, P.K. Leveraging Stacking Framework for Fake Review Detection in the Hospitality Sector. *J. Theor. Appl. Electron. Commer. Res.* **2024**, *19*, 1517–1558. <https://doi.org/10.3390/jtaer19020075>

Academic Editor: Ercan Oztemel

Received: 14 September 2023

Revised: 11 November 2023

Accepted: 19 November 2023

Published: 15 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Effective decision-making heavily hinges on information search processes. Within the realm of e-commerce, consumer choices are significantly influenced not only by the information disseminated by companies but also by the evaluations of fellow product purchasers [1]. Gradually, these product reviews have transitioned into an integral facet of consumers' purchasing decisions, with a staggering 90% of customers consulting online reviews prior to making purchase-related choices. Remarkably, 88% of consumers place a level of trust in online reviews akin to personal recommendations [2]. This escalating reliance on such reviews, however, unravels a concern of manipulation in the decision-making process through the injection of fabricated reviews [3].

Managers have realized the potential of reviews on consumer engagement intention, leading some of them to engage in review manipulation [4]. Fabricated reviews encompass two distinct categories: destructive and deceptive. Destructive reviews often serve as mere promotional content that bears no relation to the actual product experience. On the other hand, deceptive reviews are particularly harmful as they spread false information that can seriously harm businesses and result in significant financial loss [5]. Notably, even renowned platforms like Tripadvisor have struggled to grapple with the pervasive issue of counterfeit reviews. This is demonstrated by their multiple shifts in slogans over the years, underscoring the complexity of distinguishing genuine reviews from fraudulent

ones. This issue has inspired our research into the detection of counterfeit reviews within the hospitality domain [6–8].

The magnitude of this predicament is starkly quantified, with the World Economic Forum estimating annual losses of an astonishing USD 152 billion due to the proliferation of counterfeit reviews, an economic loss that merits serious consideration [9]. The gravity of the problem has attracted the attention of international regulatory bodies, spanning from the European Union and the United States to Australia and India [10]. These authorities have enacted stringent measures to combat the endorsement and dissemination of counterfeit reviews. Regulations now mandate that platforms verify the authenticity of consumer reviews or face prosecution and penalties. Despite these efforts, the ground reality remains different [11–13].

The menace of counterfeit reviews has particularly targeted review aggregator platforms like Yelp, Tripadvisor, and more, compelling scholars' engagement for well over a decade [14,15]. These inauthentic reviews extend beyond external sites to influential platforms like Amazon, Walmart, and Flipkart. Notably, establishments are found to be more prone to receiving fake positive reviews on external sites like Tripadvisor, Yelp, MouthShut, etc. [16]. However, in cases where establishments feature on both internal and external platforms, the ratings on external platforms tend to be lower, owing to the deliberate injection of negative counterfeit reviews by competitors [17]. The lack of a robust purchase validation mechanism renders external platforms more susceptible to the infiltration of fake reviews.

While fake reviews span various sectors, they are notably concentrated in entertainment, hospitality, and e-commerce [18]. The initial response involved manual identification, but this approach was proven to be sluggish, imprecise, and resource-intensive. This paved the way for the pioneering work of Jindal and Liu [19], who introduced the concept of automated fake review detection. Subsequently, machine learning techniques encompassing Support Vector Machines (SVM), Random Forest (RF), Naïve Bayes, and neural networks (NN) have gained significant prominence for detecting counterfeit reviews [20]. Notably, the scope of fake review detection extends beyond supervised learning methods, encompassing semi-supervised and unsupervised approaches [14].

The prevailing research landscape predominantly revolves around feature engineering and training classifiers to effectively distinguish between genuine and fraudulent reviews. Comparative analyses of classifiers, such as Naïve Bayes Classifier and Support Vector Machine (SVM), have garnered attention [18,21,22]. Paradoxically, despite the capabilities of machine and deep learning to handle large datasets, research often focuses on comparatively small datasets (as mentioned in Section 2) when reporting superior performance. This raises valid concerns about overfitting and lack of real-world adaptability in scenarios where millions of reviews are generated daily. These observations culminate in the formulation of the ensuing research questions.

RQ1. In the context of fake review detection in the hospitality sector, does combining the power of several base classifiers with a meta-classifier lead to a consistently better result?

RQ2. Will the performance of such a classifier vary from the scaling of input data size?

To answer RQ1, we have built upon the methods of [23,24] by creating an ensemble model by using several well-reported classifiers (XGBoost, Random Forest, artificial neural network, etc.) in the fake review detection domain [15,25]. In answering RQ2, we have used the Yelp dataset by [26]. The dataset consists of reviews from review aggregator site Yelp.com and corresponds to hotels and restaurants. They have created three databases with varying sizes ideal to test our classifier's performance. The work offers some key theoretical contributions, including the proposition of a state-of-the-art framework using the staking ensemble technique for fake review detection, the demonstration of a comparable framework performance with the existing benchmark models irrespective of the size of the input, and the introduction of several new features (average rating provided to a product, subjectivity, average rating received by a product, etc.) with more distinguishing power. The study findings unequivocally demonstrate the enhancement in classification

performance through the application of classifier ensembling. This observed outcome remains invariant across varying dataset magnitudes. The empirical verification encompasses diverse evaluation metrics, including, but not limited to, average precision, recall, area under the curve, F1 score, and accuracy. Through comprehensive benchmarking, it is established that our approach distinctly excels in cumulative performance, thereby substantiating its superior efficacy in the domain of counterfeit review identification.

The remainder of this paper is organized as follows: Section 2 reviews the existing literature work. Section 3 highlights the proposed model along with its features, learning techniques, and evaluation. Section 4 delves into findings, implications, and future research, while Section 5 throws light on the conclusions.

2. Literature Review

2.1. Firms and Fake Reviews

It has been established that users base their opinions on reviews of a product or a service, a fact that manufacturers and service providers are well aware of [27,28]. Research indicates that service providers are more likely to engage in fake review activities when their reputation is at stake due to a limited number of reviews or negative feedback. Moreover, restaurants experiencing heightened competition are found to be more vulnerable to receiving negative reviews [29]. A study by [16] revealed that independent or single-unit hotels and restaurants are the primary beneficiaries of review manipulation and are, therefore, more susceptible to it. Conversely, [30] found that even a small number of fake reviews (50) can be sufficient to surpass the competitors in certain markets. Refs. [31] and [1] have reported that consumers associate themselves with the review website rather than other participants, and this relationship is moderated by homophily and tie strength, which foster source credibility in the context of electronic word-of-mouth (eWOM) on review websites.

There have been cases where prominent brands were prosecuted for availing services of a third party to promote or defend them online [32]. The pervasiveness of this issue has garnered attention not only from academic circles but also from major news outlets such as the BBC and the New York Times, which reported on a photography company for defaming its competitors by posting fake negative reviews [33]. Trends have shown that fake reviews are increasing day by day across platforms, thus causing a problem for online information accuracy and potential market regulation [34].

2.2. Fake Review Detection

The prevalence of fake reviews has garnered significant attention in academic circles. A range of studies have aimed to detect fake reviews. Some of them have applied supervised learning techniques, while others have utilized alternative methods (semi-supervised learning, unsupervised learning, probabilistic models, graph-based models, and deep learning) [35]. Supervised learning is a machine learning technique where the model is trained on a labeled dataset, learning to map input data to corresponding output labels through iterative adjustments, enabling it to make predictions on new, unseen data, whereas semi-supervised learning is a type of machine learning where the model is trained on a combination of labeled and unlabeled data, leveraging both the provided answers and the patterns it discovers independently to make predictions. Unsupervised learning is a machine learning approach where the model explores patterns and structures in unlabeled data without explicit guidance, uncovering hidden relationships or grouping similar items based on inherent similarities. Ref. [19] tackled this issue by developing a rule based on similarity: reviews with more than a 90% similarity were deemed as spam. They extracted 24 review-centric features and trained the model using logistic regression. Ref. [36] have reported that deceptive reviews have greater lexical complexity, contain more frequent mentions of brands and first-person pronouns, and display a sentiment tone that is more positive towards a product or service.

Ref. [37] have approached the problem of fake review detection by proposing a probability-based framework. Ref. [38] identified suspected spammers who consistently wrote either positive or negative reviews for a specific brand using a rule-based approach. In their seminal work, Ref. [39] subdivided the fake review identification problem into three parts viz, text categorization, psycholinguistic features extraction, and genre identification. They claim that deceptive reviews are part of imaginative writing, and truthful reviews are part of informative writing.

Ref. [40] explored the problem of identifying fake reviewer groups. They postulated that fake reviewer groups are more damaging than individual fake reviewers. They further used several behavioral models to detect the fake reviewers' group. They also concluded that although it is extremely difficult to label each review or reviewer as spam or truthful, it is relatively easy when it comes to reviewing groups. Ref. [41] found that parts of the speech feature strengthened with unigram features and context-free grammar is the most effective combination in enhancing the algorithm's performance.

Ref. [42] reported that behavioral cues are more distinctive than linguistic features in detecting fraudulence. They further mentioned that suspected fake reviewers could be identified from the psycholinguistic cues that they leave behind. Ref. [43] proposed a novel graph-based model optimized with joint probability via belief propagation to detect fake reviews and reviewers simultaneously.

Ref. [3] proposed a latent spam model, which is a generative model for clustering. Their findings suggest that, unlike traditional falsehoods, web-based lies tend to include more first-person pronouns. In a comprehensive study, Ref. [44] revealed that fake reviews are comparatively more negative than genuine ones, lack objectivity, are less related to the purchased items, and exhibit linguistic cues of deception. They showed that even without any apparent ulterior motivation, fake reviewers indulge in such activities. Ref. [45] showed that linguistic features based on review readability, review genre, and review writing style effectively differentiate between genuine and fake reviews. They established that in terms of readability, deceptive reviews are more readable. Furthermore, to make the reviews more believable, fake reviews often use more verbs and function words. Ref. [16] have studied the establishment involved in review manipulation. They found that small businesses and individual independent hotel owners are more prone to review-related fraud.

Ref. [46] extracted 83 linguistic features from the review dataset. Unlike others, they have made use of reviews as well as review titles for feature engineering. They have subdivided those features based on understandability, level of detail, writing style, and cognition indicators. Ref. [26] introduced an unsupervised and semi-supervised model for a fake review, reviewer, and targeted product identification. The model was based on relational network architecture that can exploit prior knowledge of class distribution.

Ref. [47] provided evidence that using semantics along with emotional features can improve classification accuracy. Ref. [48] provided the first deep learning-based framework for the problem. Deep learning is a subset of machine learning that employs artificial neural networks with multiple layers (deep neural networks) to automatically learn and represent complex patterns in data, enabling the model to make sophisticated decisions or predictions. It is like teaching a computer to think by simulating the intricate connections of the human brain through layers of virtual neurons. They emphasized that product-related features, as opposed to review-related features, are more effective for this purpose. Ref. [35] have shown that rather than verbal features, non-verbal features such as membership length, helpful votes, friend count, etc., are more effective in detecting fake online reviews that are present on Yelp's social network. They maintained that manipulating verbal features is easy, whereas copying and manipulating non-verbal features is time-consuming and challenging for the review spammers.

Ref. [49] provided a graph-based, unsupervised learning solution to the problem. The main advantage of their method is the ability to achieve commendable results even without a training dataset. Ref. [21] categorized the features extracted from the review dataset based on variance, temporal aspect, rating, textual, and burst dimensions. They found that

burst-related features are more relevant in identifying fake reviews. Ref. [50] proposed a bidirectional gated recurrent neural network, which helps in capturing global semantic information that standard discrete features fail to grasp. A bidirectional gated recurrent neural network is a type of artificial neural network designed for processing sequential data capable of analyzing information both forward and backward through time. It uses special gates to selectively remember and forget past information, enhancing its ability to understand context and relationships in the data bidirectionally.

In the experiment conducted by [51], they revealed that a weak brand suffers more when it excessively adds fake positive reviews, and this raises suspicion among the users, leading to a loss of credibility [52]. They further reported that deleting negative reviews is more subtle and leaves fewer manipulation cues. Ref. [22] illustrated the use of univariate and multivariate distributions in improving classifier accuracy.

Ref. [53] developed multiple deep learning-based solutions to address the issue of variable-length review texts. They proposed two approaches: one using multi-instance learning and the other employing a hierarchical architecture, both aimed at effectively handling reviews of varying lengths. Ref. [17] developed a supplementary method called trust measure that determines the genuineness of a review based on strongly positive and negative terms. They reported that fake reviews are more prevalent on open reviewing platforms than on closed platforms.

Ref. [54] derived several micro-linguistic cues using Linguistic Inquiry Word Count (LIWC) and Coh-Metrix to study their impact on positive and negative reviews being either fake or genuine. Their findings revealed that single posts, reiterating posts, and generic feedback are useful clues in identifying spam. Ref. [55] leveraged the capability of deep learning architectures and proposed a high-dimensional model conflating n-gram, skip-gram, and emotion-based features. Refs. [28,56] have postulated that fake reviews have more social and affective cues as compared to genuine reviews.

Ref. [57] examined the temporal impact on classifier performance. They proposed that algorithms dealing with text should have the capability to periodically update their vocabulary with words used in general parlance. Ref. [58] have explored an interesting concept of inconsistency and its potential to enhance classifier performance, defining it as a disparity between review content and star ratings, differing sentiments for the same rating, or a change in a reviewer's writing style for the same rating. Ref. [59] have stressed the presence of emotional cues in fake reviews. Refs. [60,61] have relied on feature engineering along with word embedding to identify fake and genuine reviewers.

A quick look at Table 1 solidifies our argument that the majority of the high-performing work has been reported on a relatively smaller dataset and often reported on a selective set of matrices, which can lead to a false sense of performance. Furthermore, combining the power of multiple classifiers has received less attention than warranted in academia, in contrast to machine learning contests where such architecture often features as best-performing models [62]. Additionally, training deep learning models is resource-intensive and time-consuming. To this end, we propose a stacking ensemble-based classifier that is faster and easier to train and performs well, irrespective of the size of the input dataset.

3. Methodology

For this study, the Yelp datasets curated by [26] are selected. The reviews were collected over four years, between 2010 and 2014, from Yelp.com. The dataset comprises reviews given to restaurants and hotels in the US. The investigation encompasses three datasets, namely YelpChi, YelpNYC, and YelpZip. The YelpChi dataset has reviews of hotels and restaurants situated in Chicago, whereas YelpNYC comprises reviews of restaurants in New York City. YelpZip specifically gathers reviews from restaurants in New York City based on their zip codes. Yelp employs a filtering algorithm designed to identify and segregate potentially fake or suspicious reviews into a distinct filtered list. These filtered reviews are publicly accessible, with a business' Yelp page showcasing recommended reviews, while a link at the page's bottom allows users to peruse the filtered or unrecommended

reviews. Although the Yelp anti-fraud filter is not infallible, it approximates a “near” ground truth and has demonstrated a propensity for accurate outcomes [63]. The datasets under consideration include both recommended and filtered reviews, denoted as genuine and potentially fraudulent, respectively. Table 2 presents the details of the datasets. The metadata includes features such as ‘user_id’, ‘prod_id’, ‘rating’, ‘label’, and ‘date’. These represent the encoded identifier of the user who submitted the review, the encoded identifier of the product being reviewed, the rating given by the user on a scale from 1 to 5, whether the review has been filtered out by the system, and the date on which the review was submitted. The ‘label’ feature has two values: ‘-1’ signifies that the review has been filtered, indicating that Yelp.com’s algorithm has marked it as fake or spam, while ‘1’ indicates that the review has not been filtered.

A series of preprocessing steps performed over the data for classification purposes are discussed in detail in Section 3.1. The proposed framework is shown in Figure 1. After the preprocessing was completed, feature engineering was performed, resulting in several features based on the previous literature. A total of six user-centric, six product-centric, and twenty-five review-centric features were derived. Details of these features are provided in Section 3.2.

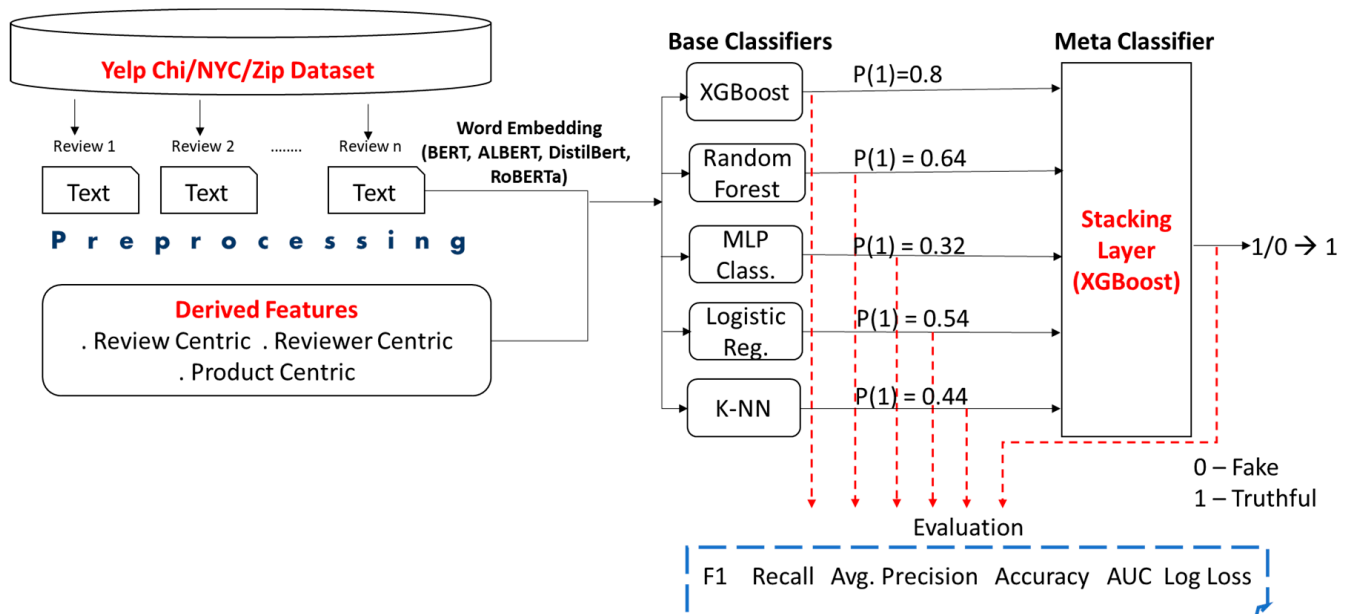


Figure 1. Proposed framework for fake review detection.

As machine learning models do not accept textual data as they are, review text was embedded. The most popular types of embeddings are BERT and its variants DistilBERT, RoBERTa, and ALBERTA. These embeddings, along with the user/product/review-centric features, were used in the classification task. The evaluation was performed in terms of classification accuracy, AUC, F1 score, average precision, and recall.

Table 1. Review of the selected literature on fake review detection in hospitality.

Authors	Dataset Used	Distribution	Methodology	Features	Description	Performance Reported
[39]	Self ⁶	400 positive, 400 negative deceptive reviews from Tripadvisor.com	Supervised Machine Learning	Psychologicistic and n-gram	Parts-of-speech tagging, Linguistic Inquiry and Word Count (LIWC) 2007, uni-gram, bi-gram, tri-gram	Acc ¹ = 89.8%, P ² = 89.8%, R ³ = 89.8%, F1 ⁴ = 89.8%
[41]	[39], Self	[39]; 400 positive, 400 negative deceptive reviews from Tripadvisor.com; 400 positive, 400 negative deceptive reviews from Yelp.com Yelp Hotel: 4876 genuine, 802 (14.1%) fake reviews Yelp Restaurant: 50,149 genuine, 8368 (14.4%) fake reviews	Supervised Machine Learning	Context-free grammar and linguistic features	Bag-of-words, parts-of-speech, CFG-based production rules	Acc = 91.2% Ott Acc = 76.6% TA Acc = 64.3% Yelp
[42]	Self	802 (14.1%) fake reviews Yelp Restaurant: 50,149 genuine, 8368 (14.4%) fake reviews	Supervised Machine Learning	Behavioral and n-gram	Review length, review deviation, content similarity, activity window, etc.	Yelp Hotel: Acc = 85.1%, P = 86.9%, R = 82.2%, F1 = 84.8%, Yelp Restaurant: Acc = 86.5%, P = 84.5%, R = 87.8%, F1 = 86.1%
[64]	Self	3523 fake and 6242 genuine reviews from Dianping.com	Supervised Machine Learning, collective classification models	Text features of reviews and behavioral features of users and IP addresses	Uni-gram, bi-gram	F1 = 74.5%
[35]	[42]	Yelp Restaurant: 50,149 genuine, 8368 (14.4%) fake reviews	Supervised Machine Learning	Verbal and non-verbal features	Review length, redundancy, capitalization, length of membership, average posting rate, etc. Number of words. Ratio of capital letters, rating, rating deviation, density, rating entropy, max rating per day, etc.	Acc = 87.81%, P = 87.12%, R = 89.63%, F1 = 88.31%
[21]	[26]	YelpZip: 512,905 genuine, 78,937 (13.3%) fake reviews	Supervised Machine Learning	Textual, meta-features, reviewer-centric, temporal features	FakeGAN uses two discriminator models D, D' and one generative model G.	YelpZip: Acc = 80.6%, P = 77.6%, R = 86.1%, F1 = 81.6%
[65]	[39]	400 positive, 400 negative deceptive reviews from Tripadvisor.com YelpChi: 67,392 genuine, 8919 (13.23%) fake reviews YelpNYC: 359,052 genuine, 36,885 (10.27%) fake reviews YelpZip: 512,905 genuine, 78,937 (13.3%) fake reviews	Semi-Supervised Deep Learning	FakeGAN, Fake Generative Adversarial Network	Multi-attention unit to extract user (product)-related review information	Acc = 89.1%, P = 98%, R = 81%
[66]	[26]	Hotel: 400 positive, 400 negative deceptive reviews from Tripadvisor.com Restaurant: 200 positive genuine, 200 positive deceptive reviews from	Deep Learning	HFAN: Hierarchical Fusion Attention Network Deep feed-forward neural network & Convolutional neural network is proposed	Model captures the complex features hidden in high-dimensional word, sentence, and emotion representations. Conducted by integrating word, sentence, and emotion representation	YelpChi: AP = 48.87%, AUC ⁵ = 83.24% YelpNYC: AP = 53.82%, AUC = 84.78% YelpZip: AP = 83.35%, AUC = 87.28%
[55]	[64,67]	Hotel: 400 positive, 400 negative deceptive reviews from Tripadvisor.com Restaurant: 200 positive genuine, 200 positive deceptive reviews from	Deep Learning	Deep feed-forward neural network & Convolutional neural network is proposed	Model captures the complex features hidden in high-dimensional word, sentence, and emotion representations. Conducted by integrating word, sentence, and emotion representation	Hotel: Acc = 89.56%, AUC = 95.1%, F1 = 89.6% DFFNN Restaurant: Acc = 89.80%, AUC = 96.5%, F1 = 90.1% CNN

Table 1. *Cont.*

Authors	Dataset Used	Distribution	Methodology	Features	Description	Performance Reported
[58]	Self	Yelp: 11,641 genuine and 12,898 fake reviews	Machine Learning	Content-based and language style	Noun count, verb count, review length, subjectivity, lexical diversity, sentiment, etc. Review representation model based on behavioral and sentiment-dependent	Acc = 92.1%, P = 93.1%, R = 90.9%, F1 = 92.0%
[60]	[26]	YelpZip: 512,905 genuine, 78,937 (13.3%) fake reviews	Deep Learning	Deep learning model on behavioral and sentiment features	linguistic features like, average rating, rating deviation, early time frame, etc.	AUC = 91.6% F1 = 83.0%

where 1-Acc represents accuracy, 2-P represents precision, 3-R represents recall, 4-F1 represents F measure and 5-AUC represents area under the curve, 6-Self in Dataset Used column means that data have been curated by the author.

Table 2. Details of the Yelp datasets.

	YelpChi	YelpNYC	YelpZip
Number of users	38,050	160,225	255,903
Number of products (Hotels and Restaurants)	200	923	5002
Number of reviews	67,392	359,052	591,842
Number of fake reviews	8919 (13.23%)	36,885 (10.27%)	78,937 (13.33%)
Number of spammers	7737 (20.33%)	28,496 (17.78%)	61,210 (23.91%)

3.1. Pre-Processing

3.1.1. Data Balancing

Table 2 shows that the dataset is highly imbalanced, which would cause a model trained on it to be biased towards the majority class. To balance the dataset, the ‘imbalanced-learn’ python package by [68] is used. The ‘RandomUnderSampler’ technique brings down the majority class instances to that of the minority class by randomly removing instances from the majority class. This technique is employed for two reasons: Firstly, oversampling the dataset would have augmented the data point, making it difficult for the classifier to converge, especially when word embeddings were also used in the model. Secondly, synthetically created embeddings do not represent the actual reviews and would not have made any sense.

3.1.2. Text Pre-Processing

The standard pre-processing steps are followed: removal of stop words, removal of punctuations, removal of URLs, lowercasing the text, and lemmatization. The label of filtered reviews is changed from -1 to 0 as this is a binary classification problem, and some of the classifiers used expect classes to be labeled as 0 and 1 only. In this study, the NLTK package is used for preliminary processing. Lemmatization was performed using the spaCy package to give the best result in real-world settings [69].

3.2. Feature Engineering

Apart from the existing meta-features, a total of thirty-seven psycholinguistic features have been derived that can be classified into user-centric, product-centric, and review-centric features. Besides the features that have already been reported in the literature, we

have engineered some new features intuitively and adopted some features from other fields of the literature. User-centric features are those that are concerned with user behavior. These features include the average rating provided by the user against all products, the total number of reviews written by a user, and their deviation. Product-centric features are those that describe the characteristics of the product from reviewers' point of view. These include the number of reviews received by the product and the average rating given by the user to the product. Lastly, review-centric features are mainly concerned with the linguistic aspect of the reviews and are not limited to the presence of exclamation marks in sentences or the count of uppercase and lowercase letters. They also include various emotional variables, such as sentiment scores and variables highlighting anger, joy, and trust in the review text. The complete list of features, along with their description and categorization, is presented in Table 3.

Table 3. Description of the features extracted from the dataset.

Type	Features	Taken From	Description
User-Centric Features	avg_Urating	[70]	average rating provided by a user
	UCcounts	[18]	number of comments provided by each user
	day_Urating	[42]	number of ratings provided by the user on a single day
	Uavg#word		average number of words in a comment of a user
	Var_Urating	[71]	variance of ratings
Product-Centric Features	Uvar#word	new	variance of number of words in a review
	avg_Prating	new	average rating received by a product
	PCcounts	new	number of comments received by the product
	day_Prating	new	number of comments received by a product on a single day
	Pavg#word	new	the average number of words in a review received by the product
	Var_Prating	new	variance of rating received by a product
Review-Centric Features	Pvar#word	new	variance of number of words in a review on a product
	#ofCharacter	[71]	number of characters in a review
	#ofword	[70]	number of words in a review
	count_punct	new	number of punctuations in a review
	Uppercase	new	number of uppercase characters in a review
	Lowercase	new	number of lowercase characters in a review
	subjectivity	[71]	a number representing the proportion of subjective words as opposed to objective words
	Exclaim	new	presence of exclamation mark (!) in the review
	positiveSent	[72] new	positive sentiment score between 0 to 10
	negativeSent		negative sentiment score between 0 to 10
	ld	[73] new	lexical diversity: ratio of unique words to all words in a review
	le_d	[74]	lexical density: ratio of opinion words to all words in a review
	anger	[75] new	anger emotion on a scale of 0 to 1
	anticipation		anticipation emotion on a scale of 0 to 1
	trust		trust emotion on a scale of 0 to 1
	fear		fear emotion on a scale of 0 to 1
	sadness		sadness emotion on a scale of 0 to 1
	joy		joy emotion on a scale of 0 to 1
	surprise		surprise emotion on a scale of 0 to 1
	disgust		disgust emotion on a scale of 0 to 1
	Entropy	[26]	rating entropy
	singleton		whether a review is the only review posted by the user
	date_entropy		time gap between each consecutive review
	similarity	[42]	maximum content similarity
	Ext		rating extremity
	ratio_LCAPS	[76]	ratio of uppercase character to lowercase character

Figure 2 shows the correlation matrix among the features of the YelpZip dataset. Most features show either a negative or negligible correlation (values below or closer to 0), suggesting that there is no multicollinearity issue with the acquired features. Figure 3 illustrates the cumulative distribution function for the engineered features from the YelpZip dataset. In this context, ‘Ham’ refers to genuine reviews and ‘Spam’ to fake reviews. Features like avg_Urating, day_Urating, Entropy, similarity, and day_entropy exhibit the greatest discriminatory power. For the correlation plot and CDF of the YelpChi and YelpNYC datasets, refer to Appendices B and C, respectively.

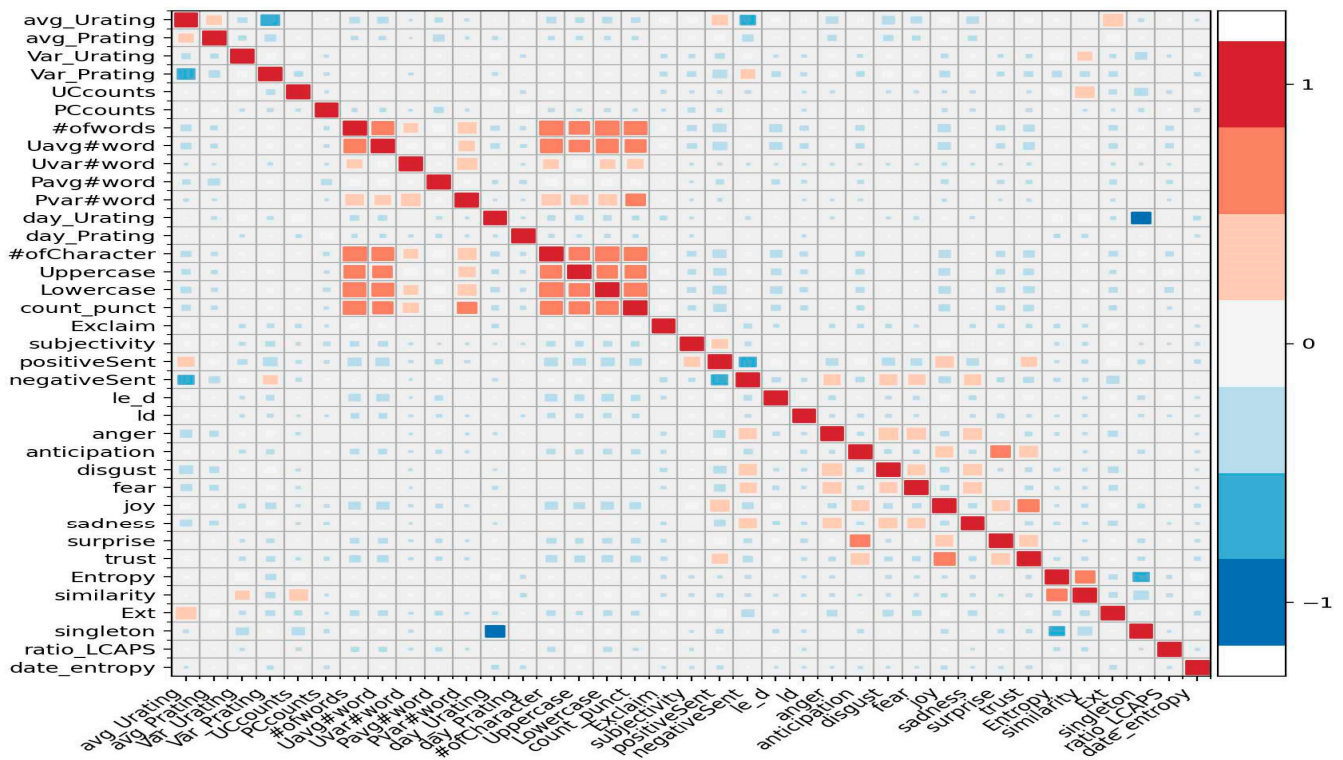


Figure 2. Correlation among the features of the YelpZip dataset.

3.3. Text Embedding

We have considered transformer-based architecture, BERT (Bidirectional Encoder Representations from Transformers) embedding, and its popular variants, ALBERT, RoBERTa, and DistilBERT, to convert textual input into machine-understandable numerical form. The transformer technique utilizes attention mechanisms to process input data in parallel, capturing contextual information efficiently. BERT can produce meaningful embeddings because it is trained on large-scale real-world datasets. It leverages an attention mechanism that dynamically calculates the relationships between input words based on their context within a sentence. BERT comprehends context in language by considering both preceding and succeeding words, capturing intricate relationships for better understanding. ALBERT optimizes BERT’s efficiency by sharing parameters among layers, offering similar performance with fewer parameters, making it computationally more efficient. RoBERTa refines BERT by modifying training methods, removing the next sentence prediction task, and using larger mini-batches for improved natural language understanding. DistilBERT retains essential language representations with reduced complexity, enabling faster training and lower resource requirements while maintaining performance.

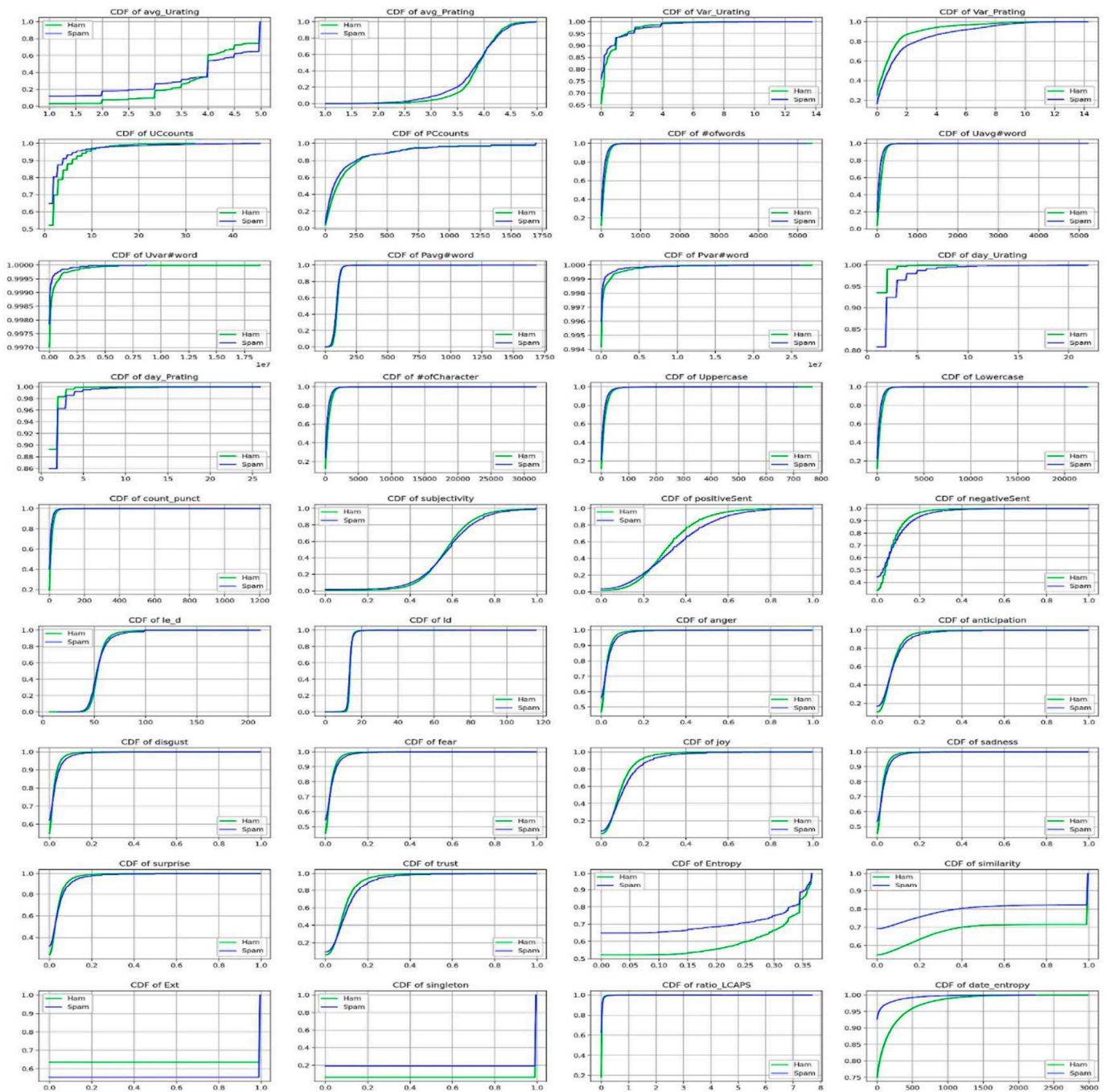


Figure 3. Cumulative distribution plot of YelpZip dataset features.

3.4. Fake Review Detection Model

Our approach utilizes a supervised learning framework known as the stacking-based ensemble technique. Table 1 lists various classifiers sourced from the literature, including the multi-layer perceptron classifier, Random Forest classifier (Bagging), logistic regression, k-nearest neighbor classifier ($K = 3$), and XGBoost classifier (Boosting). Multi-layer perceptron classifier is a neural network with multiple layers of interconnected nodes that learns complex patterns, making it effective for tasks like image recognition and classification. Random Forest classifier is an ensemble method that builds multiple decision trees during training and combines their predictions in classification tasks. Logistic regression is a statistical model used for binary classification, predicting the probability of an event occurring; it employs the logistic function to map input features into a probability range

between 0 and 1. K-nearest neighbor classifier classifies data points based on the majority class among their k ($=3$)-nearest neighbors, making it straightforward and adaptable for various datasets. An optimized gradient boosting algorithm that combines weak learners (usually decision trees) to create a strong predictive model. Boosting technique sequentially builds a series of models, with each new model focusing on correcting errors made by the previous ones, enhancing overall predictive performance.

Classifiers can be combined via two approaches: voting or stacking. As the name suggests, voting adds an extra layer that decides the final outcome based on the majority rule from the base classifiers' predictions. Stacking, however, is a more complex approach where the base classifiers' predictions are used as input for another classifier, creating a layered approach. Through experimentation, we chose the XGBoost classifier as the meta-classifier because it delivered the best results among the prospective classifiers. We have used the stacking method with probability over voting as it provides a better performance [77]. Furthermore, we have reported five-fold stratified cross-validation results across all the prominent metrics.

3.5. Performance Evaluation

In order to evaluate the performance of the classifier, several metrics are used. Table 4 shows a binary confusion matrix.

Table 4. A binary confusion matrix.

		Predicted Class	
		Truthful 1	Fake 0
Actual Class	Truthful 1	True Positive (TP)	False Negative (FN)
	Fake 0	False Positive (FP)	True Negative (TN)

Then,
Accuracy is given by

$$\text{Accuracy} = ((\text{TP} + \text{TN})) / ((\text{TP} + \text{FN} + \text{FP} + \text{TN}))$$

Precision is given by

$$\text{Precision} = \text{TP} / ((\text{TP} + \text{FP}))$$

Recall (Sensitivity) is given by

$$\text{Recall} = \text{TP} / ((\text{TP} + \text{FN}))$$

F1 score is given by

$$\text{F1} = (2(\text{Precision} \times \text{Recall})) / ((\text{Precision} + \text{Recall}))$$

Specificity is given by

$$\text{Specificity} = \text{TN} / ((\text{TN} + \text{FN}))$$

The area under the curve is a graph plotted between sensitivity and specificity at different threshold values. The closer this value is to 1, the better the classification.

4. Findings and Discussion

4.1. Model Evaluation

To address Research Questions 1 (RQ1) and 2 (RQ2), a series of empirical experiments were conducted, involving the systematic manipulation of feature sets and diverse embedding styles and dataset sizes. A comprehensive total of five distinct experiments were executed, delineated by variations in both embedding and feature configurations.

Specifically, Experiment 1 was designed to investigate classification performance solely utilizing engineered features. Subsequent experiments, namely Experiments 2 through 5, incorporated the fusion of features alongside word embeddings. Experiment 2 explored the amalgamation of features with BERT embeddings, while Experiments 3 and 4 sequentially incorporated ALBERTA and DistilBERT embeddings. Experiment 5 culminated with the utilization of RoBERTa embeddings. Within Experiment 1, a progressive strategy was adopted, commencing with the standalone utilization of the derived feature set, subsequently evaluating their combined effect. Detailed graphical representations of our model's performance across varied evaluation metrics were generated for the YelpZip, YelpChi, and YelpNYC datasets (Figures 4–6, respectively). Notably, the delineated feature sets encompass user-centric (FU), product-centric (FP), review-centric (FR), and the composite of all features (FA).

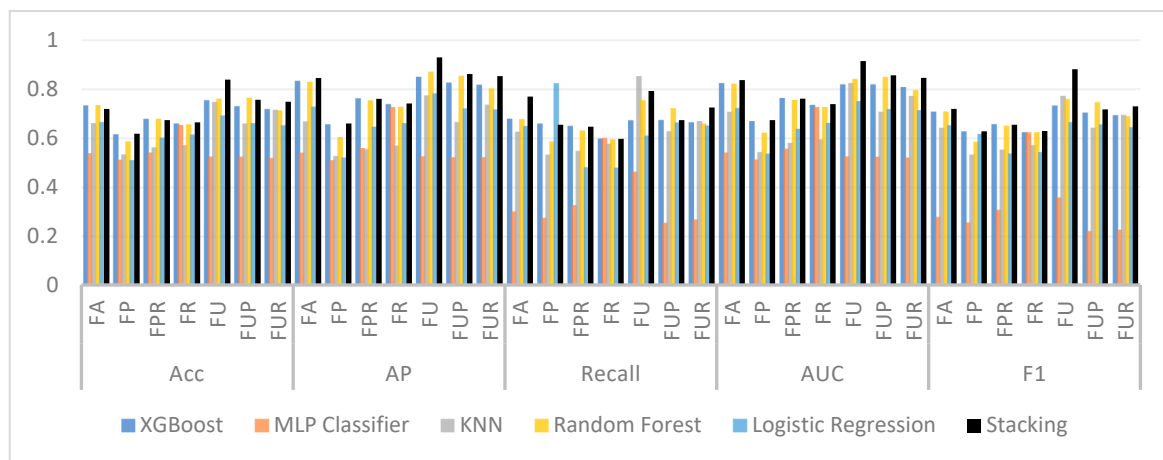


Figure 4. Graph showing the model's performance on the YelpZip dataset.

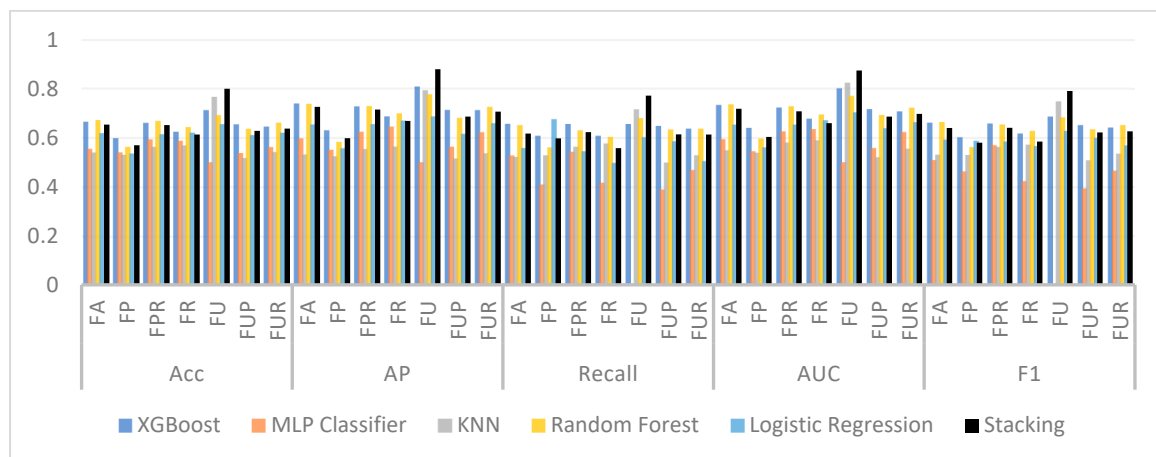


Figure 5. Graph showing the model's performance on the YelpChi dataset.

Figure 4 elucidates that the stacking technique consistently yielded optimal outcomes across diverse scenarios. Particularly, for the expansive YelpZip dataset, the model achieved notable performance metrics, with an accuracy of 83.89%, an average precision of 92.93%, a recall of 79.22%, an AUC of 91.46%, and an F1 score of 88.12%. Analogous trends were echoed in the context of YelpChi (Figure 5) and YelpNYC (Figure 6) datasets.

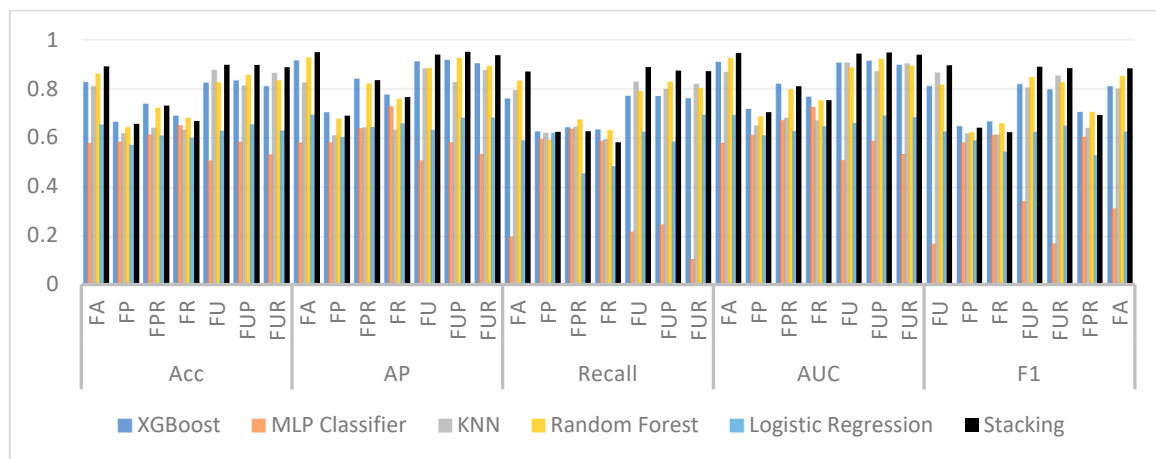


Figure 6. Graph showing the model's performance on the YelpNYC dataset.

For a comprehensive assessment of the model's performance, readers are encouraged to review Appendix A. Detailed examination of these sections indicates that feature engineering alone, as opposed to its combination with various large language models, is exceptionally effective. This underscores the idea that in the context of detecting counterfeit reviews, the use of feature engineering and the extraction of relevant features are fundamentally more appropriate than employing resource-heavy neural network-based large language models.

To explore the explainability aspect of our model, we plotted the feature importance using the full set of features. Since our model is based on stacking architecture, it is not advisable to depend upon the importance plot of individual classifiers such as XGBoost or Random Forest. Researchers propose various approaches, such as LIME [78] or DeepLIFT [79], to improve the interpretability of the model. For our study, we employed SHAP (Shapley Additive exPlanations), as recommended by [80]. SHAP is a comprehensive, model-agnostic method that amalgamates different feature importance techniques previously developed by researchers. In a nutshell, it performs sensitivity analysis based on accuracy. The SHAP values allow us to understand any prediction or classification as the sum effect of the features. Figure 7 shows the beeswarm plot of feature importance based on SHAP values.

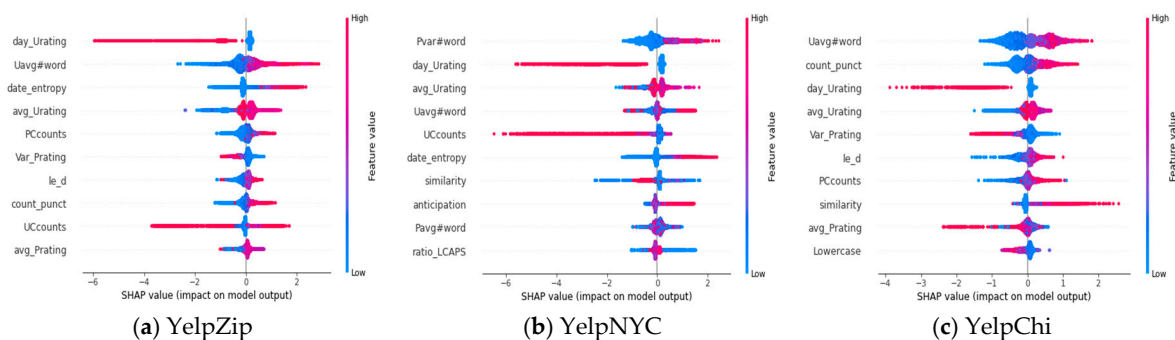


Figure 7. Feature importance plot for all three datasets.

As shown in Figure 7, user-centric features dominate the top five features in all three datasets. For YelpZip, the dataset plot suggests that an increased number of reviews written by a user in a single day can decrease the model's accuracy by as much as 6%. This observation is consistent with the model performance, which indicates that user-centric features generally lead to improved performance of the model. Number of ratings provided

by the user on a day, average rating provided by the user, and the average number of words written by the user in a review are among the top five features consistently.

The feature importance plot (Figure 7) also shows that behavioral features are more helpful in identifying fake and genuine reviews than linguistic features. The input variables are arranged in top-to-bottom order as per their mean absolute SHAP values for the first 1000 reviews in the test dataset. For Figure 7a, values for 'day_Urating' can be interpreted as if the number of ratings provided by a user in a day is higher, there is a lower chance of being a truthful user. Figures 8–10 show the local explanation of a fake review record using the SHAP force plot in each of the datasets. The plot can be interpreted as the binary target was fake (Label = 0) or truthful (Label = 1). In the plot above, the bold 0.00 is the model's score for this observation. Higher scores lead the model to predict 1, and lower scores lead the model to predict 0. The features that were important in making the prediction for this observation are shown in red and blue, with red representing features that pushed the model score higher and blue representing features that pushed the score lower. Features that had more of an impact on the score are located closer to the dividing boundary between red and blue, and the size of that impact is represented by the size of the bar. Again, behavioral features are significant in deciding to classify a record as fake or truthful.

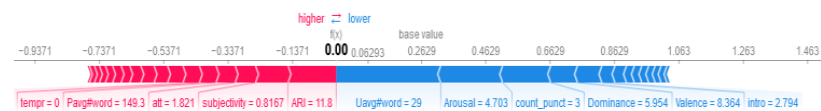


Figure 8. Force plot of a fake record in the YelpChi dataset.

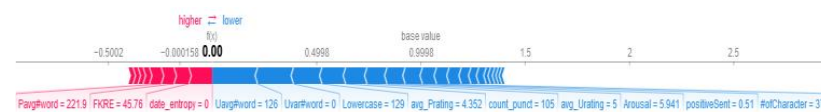


Figure 9. Force plot of a fake record in the YelpNYC dataset.

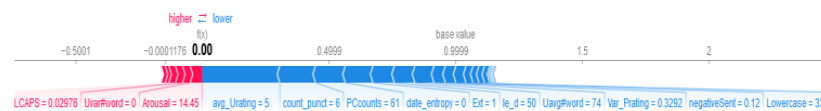


Figure 10. Force plot of a fake record in the YelpZip dataset.

Additionally, it can be noticed from Figures 4–6 that the results of the stacking classifier were closest to the best-performing classifier. It was noted that user-related features contributed significantly to the classification performance, both when used alone and when combined with other features or embeddings.

4.2. Benchmarking

The proposed model is benchmarked with that of existing models and frameworks. The following popular works based on the Yelp dataset are considered:

1. SpEagle [26]: An unsupervised learning approach (Spam Eagle) capable of integrating and scaling with labeled data using metadata along with relational data. The authors were the ones who curated the datasets. The performance of their model was tested on all three datasets.
2. Ref. [21]: Proposed an effective multi-feature-based model. They suggested some new features and conducted a performance evaluation based on burst features. The authors have considered only the YelpZip dataset.
3. Ref. [22]: They have developed a novel hierarchical supervised learning approach that analyzes user features and their interactions in univariate and multivariate distributions. They have also used the YelpZip dataset for modeling purposes.

4. SPR2EP [81]: They proposed a semi-supervised framework (SPam Review REPresentation) for fake review detection, which uses the feature vectors extracted from reviews, reviewers, and products. After combining these vectors for detection purposes, they demonstrated the performance of their model on all three datasets.
5. HFAN [66]: Hierarchical Fusion Attention Network (HFAN) is a deep learning-based technique that automatically learns reviews' semantics from the user and product information using a multi-attention unit.
6. Ref. [60]: Proposed a convolution neural network-based architecture connecting sentiment-dependent linguistic features and behavioral features via a fully connected layer to determine fake and genuine reviewers.
7. Ref. [82]: Proposed an integrated multi-view feature strategy, blending implicit and explicit features from review content, reviewer data, and product descriptions. They introduced a hybrid extraction method, combining word- and sentence-level techniques with attention. This extends to a classification framework with an ensemble classifier leveraging a convolutional neural network (CNN) for reviewer information, a deep neural network (DNN) for product-level analysis, and a Bidirectional-Long Short-Term Memory (Bi-LSTM) for review-level features. This comprehensive methodology aims to enhance analysis effectiveness across diverse dimensions.
8. Ref. [83]: Used the fine-tuned version of BERT to identify fake reviews just from review text.

As can be seen from Table 5, most of the work has been done on the YelpZip dataset—one reason for this could be the availability of a large amount of labeled data as compared to the other two datasets. Furthermore, the authors have not measured the performance of their model on other well-defined and accepted metrics such as accuracy, recall, and F1. Most of the authors have evaluated the performance of their model against the AUC metric.

Table 5. Comparative performance evaluation.

Metric Approaches		Accuracy	Average Precision	Recall	AUC	F1
SpEagle [26]	YelpChi	-	-	-	0.7887	-
	YelpNYC	-	-	-	0.7695	-
	YelpZip	-	-	-	0.7942	-
[21]	YelpChi	-	-	-	-	-
	YelpNYC	-	-	-	-	-
	YelpZip	0.806	-	0.861	-	0.816
[22]	YelpChi	-	-	-	-	-
	YelpNYC	-	-	-	-	-
	YelpZip	-	-	-	0.7686	0.7901
SPR2EP [81]	YelpChi	-	0.3351	-	0.8071	-
	YelpNYC	-	0.3202	-	0.8129	-
	YelpZip	-	0.422	-	0.8318	-
HFAN [66]	YelpChi	-	0.4887	-	0.8324	-
	YelpNYC	-	0.5382	-	0.8478	-
	YelpZip	-	0.6235	-	0.8728	-
[60]	YelpChi	-	-	-	-	-
	YelpNYC	-	-	-	-	-
	YelpZip	-	-	-	0.916	0.830
[82]	YelpChi	-	-	-	-	-
	YelpNYC	-	0.9693	0.8416	-	0.9009
	YelpZip	-	0.8318	0.7786	-	0.8043
[83]	YelpChi	-	-	-	-	-
	YelpNYC	-	-	-	-	-
	YelpZip	0.731	-	-	-	-
Our Model	YelpChi	0.8006	0.8804	0.773	0.8756	0.7916
	YelpNYC	0.8973	0.9393	0.8882	0.9433	0.8952
	YelpZip	0.8389 ¹	0.9293	0.7922	0.9246	0.8812

¹ Figures in bold are the cases where our model outperformed the existing state of the art

As apparent from Table 5, our model has delivered better performance as compared to the other models on almost all the performance metrics across the three datasets. It is

crucial to employ a model that exhibits robustness across various metrics, and our model successfully bridges this gap. The results also demonstrate the effectiveness of our approach of stacking the output of heterogeneous classifiers to reliably detect spam or fake reviews in both small and large datasets. Furthermore, our model is resource-efficient, unlike deep learning models that are complex to implement and scale, are resource-hungry, and require a heavy infrastructural investment. Our model's performance is comparable to the advanced deep learning model proposed by [66]. These obvious advantages of our model over other models make our work more practical and applicable for real-world deployment.

5. Conclusions

This study is dedicated to addressing the pervasive challenge of detecting counterfeit reviews within the hospitality and restaurant domain. Our approach centers around the development of a stacking-based model, which ingeniously amalgamates the outputs of divergent base classifiers and harnesses a meta-classifier to discern the veracity of a given review—whether it is genuine or fabricated. The framework leverages a comprehensive suite of user-centric, product-centric, and review-centric features, collectively providing a multifaceted perspective on review authenticity.

The potency of our strategy resides in the stacking technique's ability to consolidate the predictive capabilities of individual classifiers, culminating in an elevated overall efficiency. Evidently, the endeavor culminates in the establishment of a model proficient in discerning between fake and truthful reviews with commendable accuracy levels. Intriguingly, our model exhibits superior performance in comparison to well-established works across a spectrum of relevant performance metrics. Furthermore, the framework's adaptability manifests in its ability to robustly operate across varying dataset sizes, rendering its utility independent of the dataset's magnitude.

The ethical landscape pertaining to review authenticity is intricate, especially concerning the deployment of filtering algorithms designed to distinguish fraudulent or suspicious reviews. While these algorithms aim to uphold the quality and reliability of reviews, the potential for false positives—incorrectly classifying authentic reviews as fraudulent—poses a risk, potentially tarnishing the reputations of businesses or individuals. This gives rise to concerns about algorithmic bias and its implications for stakeholders. To mitigate this, we advocate for the labeling of reviews identified by the model as 'not suggested reviews' rather than outright deletion or labeling them fake/manipulated. Additionally, triangulating model findings with supplementary data sources, such as IP addresses, reviewer location, reviewer history, and businesses' history, can further enhance accuracy and reduce false positives. Routine model retraining involving deliberate introduction of fake reviews contributes to increased accuracy and resilience against false detection.

Furthermore, ethical challenges extend to the transparency of the filtering process and the disclosure of filtered reviews. Users may lack full awareness of the criteria guiding review identification and filtering, hindering their ability to evaluate information reliability. Balancing platform integrity and user transparency emerges as a nuanced ethical consideration. Addressing this challenge, our model prioritizes interpretability, facilitating a clear understanding of features significantly contributing to fake review identification.

The implications of this investigation transcend the confines of the hospitality and restaurant sector, extending to domains like movie reviews and e-commerce. This affords a broader utility for our model's insightful mechanisms. The hospitality and tourism review aggregators can embrace our model as an instrument to identify potentially inauthentic reviews, thus embarking on a path of enhanced credibility. Acknowledging the inherent uncertainty surrounding the authenticity of reviews, we advocate for a 'not-suggested' annotation for reviews flagged by our algorithm. This approach empowers website visitors to make more informed choices safeguarded against the ambiguity of potentially deceptive reviews.

As the reliance on reviews surges among consumers making purchasing decisions, it is imperative for stakeholders to proactively establish regulations and mechanisms to

thwart the influence of counterfeit reviewers. Our recommendations extend beyond mere detection mechanisms. We propose a preemptive approach, suggesting the integration of a pop-up prompt before a review is submitted. This prompt would serve as a reminder to users, encouraging them to ensure their reviews adhere to the platform's ethical guidelines. The wisdom advocated by [32] underscores the importance of consumer awareness and regulatory vigilance, effectively shielding prospective consumers from the deleterious impact of counterfeit reviews.

5.1. Theoretical Contributions

The proliferation of counterfeit reviews presents a dual menace, impacting both consumers and businesses. The increasing digital footprint of those who have grown up in the digital era heightens their vulnerability to misleading and deceptive information. This represents a significant concern as it has the capacity to profoundly skew decision-making processes by introducing cognitive biases. In light of this challenge, the imperative for a robust and scalable mechanism to counteract this influx of misinformation becomes strikingly evident.

Our contribution to the scholarly landscape is multifaceted. Firstly, our work extends the existing corpus of the deception detection literature through the development of an efficient system adept at identifying suspicious reviews. This accomplishment is underpinned by the formulation of a stacking-based framework adept at harnessing the capabilities of diverse underlying classifiers. The precedent established by [84], who validated the superior performance of this architecture through simulations using real-world data, highlights its effectiveness. Unlike deep learning-based architectures, these models demonstrate accelerated convergence and adaptability to varying data sizes. A notable shift from previous research emerges, affirming the pre-eminence of user-centric behavioral features over their linguistic counterparts. This underscores that the inherent characteristics of users exert a more profound influence on the classification process.

Secondly, the enhancement of model interpretability constitutes a pivotal facet of our approach. Rather than relying on conventional feature importance plots prevalent in the literature, our strategy leverages SHAP values. This choice enables a deeper level of insight, as SHAP-based importance plots not only unveil feature significance but also elucidate how variations in feature values impact the model's outcomes. Furthermore, as corroborated by [85], the conventional feature importance plots are inherently sensitive to the chosen methodology, posing a concern over robustness.

Our endeavor encompasses the introduction of novel features, as underscored by the insights gleaned from Table 5. Importantly, the architectural framework we have devised operates as a modular entity, thereby accommodating the integration of more efficient classifiers as they emerge in the future research landscape. Such adaptability can be achieved through minimal code adjustments, rendering our model remarkably customizable at a negligible expense.

In summary, empirical tests of our model against recognized benchmarks confirm its effectiveness. It consistently surpasses standard models in various evaluation metrics, regardless of the size of the dataset. This empirical evidence highlights the strength, robustness, and dependability of our approach.

5.2. Managerial Implications

In terms of managerial implications, the pervasive issue of counterfeit reviews poses a significant threat to trust and credibility within the review ecosystem. Platforms like Yelp, reporting the filtration of nearly 25% of their reviews, underscore the gravity of the challenge [34]. Notable instances such as Oobah Butler's experiment in 2017, where a fictitious profile became a top-rated restaurant on Tripadvisor, further accentuate the vulnerability of such platforms.

Our work provides a practical solution to the problem of counterfeit review identification. The proposed method distinguishes itself by its ease of implementation, circumventing

the complexities often associated with academic solutions. Its lightweight nature, swift convergence, and ability to flag fraudulent reviews at an acceptable threshold align seamlessly with existing operational standards. Moreover, the ordered importance of features offers review aggregator platforms actionable insights for identifying cues in reviewer comments, augmenting their mechanisms to combat fraudulent activities.

Beyond its immediate application, the generalizability and scalability of our approach extend its potential to diverse domains grappling with the scourge of fake reviews. The far-reaching implications of our study empower review platforms to mitigate the incursion of counterfeit reviews. Such liberation from the influence of fraudulent reviewers engenders a greater degree of trust in the disseminated information, culminating in heightened user traffic and ultimately translating into enhanced revenue for businesses.

From the consumer perspective, our study holds the promise of furnishing them with credible and dependable information for decision-making purposes. This will help the consumer make a more informed purchase, leading to lesser post-purchase dissonance and satisfactory experience.

Our model has the potential to significantly enhance the trustworthiness of the internet, particularly in the context of user reviews. By effectively identifying and flagging suspicious reviews, it acts as a robust deterrent against the proliferation of counterfeit feedback, thus mitigating the distortion of online information. This not only empowers users to engage with more reliable content but also instills confidence in the integrity of the digital space. An additional layer of transparency allows users to discern between reviews that may lack authenticity. In turn, businesses that heavily depend on user reviews stand to benefit immensely. The model's capability to distinguish between genuine and fraudulent reviews shields businesses from potential reputational harm caused by deceptive feedback. This safeguarding of reputations contributes to building and maintaining the credibility of businesses in the online sphere. Moreover, the adaptability of our approach across diverse dataset sizes ensures that businesses of varying scales can leverage the model to enhance their online presence.

5.3. Societal Implications

Our efforts to address counterfeit reviews wield profound societal impact by fostering a culture of authenticity and trust in the digital realm. In a broader context, our model not only benefits consumers and businesses but contributes to shaping responsible online behavior. By empowering users to make informed choices and safeguarding against misinformation, we align with societal goals of promoting consumer rights and digital literacy. The adaptability of our approach ensures that even smaller businesses integral to local economies can enjoy enhanced credibility. Ethical considerations embedded in our model, such as the 'not-suggested' annotation and transparency in the filtering process, reflect a commitment to responsible technology use. Our proactive approach, exemplified by the integration of a pop-up prompt, sets a precedent for ethical platform management, influencing the broader landscape. Moreover, our emphasis on interpretability contributes to increased transparency and accountability, enabling users to critically evaluate online information. In essence, our work goes beyond the technical realm, actively participating in the ongoing discourse on responsible technology development with the overarching aim of creating a digital landscape that positively impacts society at large.

5.4. Limitations

Our study encounters significant limitations originating from the choice of the foundational classifier. The configuration encompassing the count and nature of both the base classifiers and a meta-classifier wields substantial influence over the model's performance. A subsequent limitation pertains to the temporal relevance of our database, which potentially renders it an imperfect representation of contemporary lexical usage. Notably, features such as punctuation count, lexical diversity, and lexical density are susceptible to this dynamic linguistic landscape. This phenomenon has been examined by [57], who

unveiled its repercussions on classifier efficacy within a comparable context. An important limitation arises from the dataset’s balance or lack thereof. It is imperative to examine how our model will perform when confronted with an imbalanced dataset where the distribution of instances across different classes is uneven. Understanding the model’s behavior under such conditions is critical as it may impact its accuracy in real-world scenarios.

5.5. Future Work

For forthcoming endeavors, we are inclined to subject our model to assessment across alternate datasets. This proactive measure seeks to bolster the model’s adaptability and reinforce its empirical validity. Furthermore, our model awaits validation against expansive language models, such as GPT-4 and ChatGPT, which are capable of generating synthetic text dynamically. The cautionary standpoint underscored by [86] regarding the uncritical adoption of artificial intelligence without vigilant scrutiny for inherent biases is salient. To build a more resilient model, we intend to scrutinize our data for potential biases, a step aimed at enhancing its robustness in the face of inherent complexities. Yet another future direction could be to enhance the model to detect fake reviews posted by new users against whom the model lacks behavioral data.

An integral aspect of our future research trajectory involves the construction of a tangible regulatory framework firmly grounded in practical principles. This framework aspires to empower review aggregators in their efforts to combat the disruptive and malicious conduct of fake reviewers. Moreover, our model primarily integrates features derived from the textual content of reviews. Incorporating aspects of social networks, like reviewer interactions, could add a new layer of complexity to the model. This addition has the potential to enhance its performance even further.

Author Contributions: Conceptualization, S.A.A. and A.F.J.; methodology, S.A.A.; software, S.A.A.; validation, A.F.J., P.K.B., P.K.P. and S.B.; formal analysis, S.A.A.; investigation, S.A.A.; writing—original draft preparation, S.A.A.; writing—review and editing, S.A.A. and P.K.B.; visualization, S.A.A.; supervision, P.K.B. and P.K.P.; project administration, P.K.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The authors have not curated/collected any data. The dataset is available at <https://odds.cs.stonybrook.edu/> (accessed on 2 June 2020).

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. Classifier performance on the YelpZip dataset.

Features	Metric	XGBoost	MLP Classifier	KNN	Random Forest	Logistic Regression	Stacking
Experiment 1—Features Only							
FA	Acc	0.734	0.539	0.660	0.735	0.665	0.719
	AP	0.833	0.540	0.668	0.830	0.728	0.845
	Recall	0.679	0.301	0.626	0.677	0.649	0.769
	AUC	0.824	0.541	0.707	0.822	0.722	0.836
	F1	0.708	0.279	0.642	0.709	0.652	0.719
	Log loss	−0.523	−0.677	−4.157	−0.529	−0.620	−0.539

Table A1. Cont.

Features	Metric	XGBoost	MLP Classifier	KNN	Random Forest	Logistic Regression	Stacking
FP	Acc	0.615	0.512	0.533	0.586	0.510	0.618
	AP	0.656	0.509	0.527	0.604	0.521	0.660
	Recall	0.659	0.275	0.533	0.586	0.825	0.654
	AUC	0.670	0.513	0.543	0.622	0.537	0.673
	F1	0.628	0.256	0.533	0.585	0.617	0.627
	Log loss	−0.649	−0.692	−4.796	−0.870	−0.694	−0.647
FPR	Acc	0.679	0.541	0.562	0.678	0.602	0.673
	AP	0.762	0.559	0.555	0.754	0.646	0.760
	Recall	0.650	0.327	0.548	0.631	0.481	0.646
	AUC	0.763	0.557	0.580	0.756	0.638	0.761
	F1	0.657	0.308	0.553	0.650	0.537	0.654
	Log loss	−0.589	−0.679	−4.707	−0.594	−0.665	−0.594
FR	Acc	0.659	0.654	0.571	0.656	0.615	0.664
	AP	0.738	0.728	0.569	0.727	0.661	0.741
	Recall	0.598	0.601	0.577	0.595	0.480	0.597
	AUC	0.735	0.727	0.595	0.727	0.662	0.738
	F1	0.624	0.624	0.571	0.624	0.543	0.628
	Log loss	−0.605	−0.611	−4.582	−0.612	−0.657	−0.602
FU	Acc	0.754	0.525	0.747	0.761	0.692	0.839
	AP	0.851	0.526	0.774	0.871	0.782	0.929
	Recall	0.672	0.463	0.854	0.754	0.611	0.792
	AUC	0.819	0.525	0.824	0.841	0.750	0.915
	F1	0.732	0.358	0.772	0.759	0.665	0.831
	Log loss	−0.501	−0.684	−3.976	−0.575	−0.590	−0.359
FUP	Acc	0.730	0.524	0.659	0.764	0.661	0.756
	AP	0.827	0.522	0.665	0.854	0.721	0.862
	Recall	0.674	0.255	0.628	0.721	0.663	0.673
	AUC	0.819	0.524	0.707	0.849	0.718	0.856
	F1	0.704	0.221	0.642	0.746	0.655	0.717
	Log loss	−0.528	−0.683	−4.185	−0.499	−0.625	−0.519
FUR	Acc	0.718	0.519	0.715	0.713	0.652	0.749
	AP	0.817	0.522	0.736	0.803	0.717	0.853
	Recall	0.665	0.268	0.670	0.659	0.651	0.725
	AUC	0.808	0.521	0.772	0.795	0.714	0.846
	F1	0.693	0.226	0.694	0.689	0.644	0.729
	Log loss	−0.536	−0.689	−3.785	−0.553	−0.629	−0.514
Experiment 2—BERT embedding along with features							
B	Acc	0.680	0.532	0.611	0.695	0.538	0.778
	AP	0.659	0.629	0.611	0.624	0.614	0.659
	Recall	0.737	0.642	0.707	0.568	0.591	0.810
	AUC	0.623	0.658	0.589	0.506	0.713	0.717
	F1	0.732	0.690	0.704	0.516	0.698	0.776
	Log loss	−0.424	−0.401	−0.524	−0.453	−0.442	−0.620
BU	Acc	0.537	0.617	0.569	0.705	0.514	0.709
	AP	0.608	0.541	0.647	0.723	0.714	0.788
	Recall	0.677	0.523	0.631	0.583	0.743	0.667
	AUC	0.620	0.642	0.720	0.655	0.612	0.802
	F1	0.568	0.714	0.597	0.533	0.575	0.652
	Log loss	−0.425	−0.579	−0.306	−0.492	−0.543	−0.496

Table A1. Cont.

Features	Metric	XGBoost	MLP Classifier	KNN	Random Forest	Logistic Regression	Stacking
BP	Acc	0.518	0.606	0.593	0.512	0.587	0.776
	AP	0.614	0.672	0.650	0.609	0.730	0.765
	Recall	0.735	0.675	0.561	0.635	0.664	0.769
	AUC	0.660	0.536	0.700	0.729	0.740	0.724
	F1	0.687	0.707	0.549	0.541	0.574	0.718
	Log loss	−0.334	−0.547	−0.301	−0.325	−0.509	−0.439
BR	Acc	0.566	0.606	0.638	0.598	0.554	0.761
	AP	0.575	0.527	0.737	0.578	0.603	0.857
	Recall	0.577	0.549	0.696	0.539	0.672	0.679
	AUC	0.517	0.734	0.667	0.659	0.622	0.847
	F1	0.635	0.632	0.571	0.551	0.663	0.733
	Log loss	−0.383	−0.327	−0.585	−0.493	−0.353	−0.439
BUP	Acc	0.705	0.574	0.672	0.547	0.575	0.680
	AP	0.652	0.611	0.612	0.600	0.614	0.684
	Recall	0.711	0.744	0.550	0.692	0.689	0.724
	AUC	0.665	0.577	0.585	0.728	0.591	0.742
	F1	0.645	0.667	0.688	0.625	0.741	0.657
	Log loss	−0.495	−0.458	−0.472	−0.510	−0.441	−0.352
BUR	Acc	0.689	0.716	0.614	0.565	0.547	0.650
	AP	0.627	0.611	0.617	0.620	0.668	0.680
	Recall	0.554	0.622	0.697	0.528	0.658	0.719
	AUC	0.698	0.583	0.747	0.604	0.581	0.740
	F1	0.633	0.706	0.538	0.612	0.632	0.681
	Log loss	−0.367	−0.591	−0.394	−0.608	−0.528	−0.277
BPR	Acc	0.644	0.719	0.654	0.534	0.636	0.791
	AP	0.713	0.579	0.524	0.530	0.645	0.790
	Recall	0.524	0.635	0.598	0.676	0.681	0.671
	AUC	0.749	0.734	0.644	0.676	0.563	0.774
	F1	0.706	0.697	0.662	0.502	0.682	0.716
	Log loss	−0.460	−0.605	−0.303	−0.459	−0.403	−0.283
BA	Acc	0.581	0.712	0.536	0.640	0.709	0.741
	AP	0.744	0.724	0.588	0.560	0.674	0.690
	Recall	0.735	0.675	0.741	0.547	0.538	0.762
	AUC	0.659	0.544	0.555	0.716	0.602	0.775
	F1	0.601	0.742	0.510	0.742	0.733	0.795
	Log loss	−0.433	−0.411	−0.544	−0.562	−0.305	−0.318
Experiment 3—ALBERT embedding along with features							
A	Acc	0.490	0.486	0.490	0.489	0.488	0.505
	AP	0.489	0.483	0.492	0.486	0.486	0.509
	Recall	0.482	0.442	0.493	0.452	0.439	0.448
	AUC	0.485	0.478	0.486	0.483	0.481	0.509
	F1	0.485	0.453	0.491	0.469	0.458	0.472
	Log loss	−0.724	−0.697	−5.255	−0.708	−0.696	−0.696
AU	Acc	0.727	0.645	0.765	0.679	0.655	0.796
	AP	0.830	0.693	0.792	0.757	0.686	0.882
	Recall	0.680	0.510	0.714	0.611	0.602	0.747
	AUC	0.816	0.701	0.823	0.751	0.703	0.875
	F1	0.707	0.577	0.747	0.646	0.627	0.779
	Log loss	−0.522	−1.908	−3.628	−0.599	−0.643	−0.456

Table A1. Cont.

Features	Metric	XGBoost	MLP Classifier	KNN	Random Forest	Logistic Regression	Stacking
AP	Acc	0.599	0.542	0.533	0.567	0.510	0.585
	AP	0.627	0.556	0.527	0.579	0.521	0.605
	Recall	0.609	0.527	0.533	0.560	0.825	0.635
	AUC	0.642	0.565	0.543	0.599	0.537	0.622
	F1	0.599	0.514	0.533	0.562	0.617	0.602
	Log loss	−0.670	−0.903	−4.791	−0.678	−0.694	−0.677
AR	Acc	0.644	0.636	0.571	0.632	0.617	0.645
	AP	0.719	0.705	0.569	0.699	0.664	0.722
	Recall	0.599	0.518	0.577	0.545	0.481	0.538
	AUC	0.714	0.701	0.594	0.694	0.666	0.716
	F1	0.618	0.572	0.571	0.585	0.543	0.587
	Log loss	−0.626	−0.630	−4.585	−0.632	−0.655	−0.620
AUP	Acc	0.755	0.644	0.659	0.695	0.661	0.761
	AP	0.857	0.655	0.666	0.777	0.721	0.857
	Recall	0.709	0.580	0.628	0.630	0.664	0.679
	AUC	0.847	0.679	0.707	0.771	0.718	0.847
	F1	0.737	0.608	0.642	0.663	0.655	0.733
	Log loss	−0.491	−2.421	−4.184	−0.588	−0.625	−0.489
AUR	Acc	0.733	0.653	0.715	0.688	0.651	0.762
	AP	0.837	0.718	0.736	0.773	0.718	0.861
	Recall	0.688	0.459	0.669	0.620	0.655	0.691
	AUC	0.824	0.711	0.772	0.765	0.714	0.851
	F1	0.714	0.559	0.694	0.655	0.645	0.734
	Log loss	−0.515	−2.235	−3.782	−0.584	−0.629	−0.496
APR	Acc	0.671	0.634	0.562	0.655	0.608	0.686
	AP	0.749	0.690	0.555	0.724	0.651	0.767
	Recall	0.647	0.560	0.547	0.583	0.487	0.694
	AUC	0.749	0.683	0.580	0.724	0.646	0.762
	F1	0.654	0.595	0.553	0.616	0.543	0.684
	Log loss	−0.605	−0.639	−4.708	−0.618	−0.664	−0.585
AA	Acc	0.759	0.649	0.660	0.699	0.665	0.747
	AP	0.864	0.711	0.668	0.786	0.727	0.845
	Recall	0.714	0.488	0.626	0.633	0.647	0.665
	AUC	0.853	0.704	0.707	0.780	0.721	0.833
	F1	0.740	0.569	0.642	0.666	0.651	0.720
	Log loss	−0.484	−3.609	−4.157	−0.575	−0.621	−0.499
Experiment 4—DistilBERT embedding along with features							
D	Acc	0.644	0.636	0.571	0.632	0.617	0.645
	AP	0.719	0.705	0.569	0.699	0.664	0.722
	Recall	0.599	0.518	0.577	0.545	0.481	0.538
	AUC	0.714	0.701	0.594	0.694	0.666	0.716
	F1	0.618	0.572	0.571	0.585	0.543	0.587
	Log loss	−0.626	−0.630	−4.585	−0.632	−0.655	−0.620
DU	Acc	0.733	0.653	0.715	0.688	0.651	0.762
	AP	0.837	0.718	0.736	0.773	0.718	0.861
	Recall	0.688	0.459	0.669	0.620	0.655	0.691
	AUC	0.824	0.711	0.772	0.765	0.714	0.851
	F1	0.714	0.559	0.694	0.655	0.645	0.734
	Log loss	−0.515	−2.235	−3.782	−0.584	−0.629	−0.496

Table A1. Cont.

Features	Metric	XGBoost	MLP Classifier	KNN	Random Forest	Logistic Regression	Stacking
DP	Acc	0.490	0.486	0.490	0.489	0.488	0.505
	AP	0.489	0.483	0.492	0.486	0.486	0.509
	Recall	0.482	0.442	0.493	0.452	0.439	0.448
	AUC	0.485	0.478	0.486	0.483	0.481	0.509
	F1	0.485	0.453	0.491	0.469	0.458	0.472
	Log loss	−0.724	−0.697	−5.255	−0.708	−0.696	−0.696
DR	Acc	0.755	0.644	0.659	0.695	0.661	0.761
	AP	0.857	0.655	0.666	0.777	0.721	0.857
	Recall	0.709	0.580	0.628	0.630	0.664	0.679
	AUC	0.847	0.679	0.707	0.771	0.718	0.847
	F1	0.737	0.608	0.642	0.663	0.655	0.733
	Log loss	−0.491	−2.421	−4.184	−0.588	−0.625	−0.489
DUP	Acc	0.727	0.645	0.765	0.679	0.655	0.796
	AP	0.830	0.693	0.792	0.757	0.686	0.882
	Recall	0.680	0.510	0.714	0.611	0.602	0.747
	AUC	0.816	0.701	0.823	0.751	0.703	0.875
	F1	0.707	0.577	0.747	0.646	0.627	0.779
	Log loss	−0.522	−1.908	−3.628	−0.599	−0.643	−0.456
DUR	Acc	0.599	0.542	0.533	0.567	0.510	0.585
	AP	0.627	0.556	0.527	0.579	0.521	0.605
	Recall	0.609	0.527	0.533	0.560	0.825	0.635
	AUC	0.642	0.565	0.543	0.599	0.537	0.622
	F1	0.599	0.514	0.533	0.562	0.617	0.602
	Log loss	−0.670	−0.903	−4.791	−0.678	−0.694	−0.677
DPR	Acc	0.671	0.634	0.562	0.655	0.608	0.686
	AP	0.749	0.690	0.555	0.724	0.651	0.767
	Recall	0.647	0.560	0.547	0.583	0.487	0.694
	AUC	0.749	0.683	0.580	0.724	0.646	0.762
	F1	0.654	0.595	0.553	0.616	0.543	0.684
	Log loss	−0.605	−0.639	−4.708	−0.618	−0.664	−0.585
DA	Acc	0.759	0.649	0.660	0.699	0.665	0.747
	AP	0.864	0.711	0.668	0.786	0.727	0.845
	Recall	0.714	0.488	0.626	0.633	0.647	0.665
	AUC	0.853	0.704	0.707	0.780	0.721	0.833
	F1	0.740	0.569	0.642	0.666	0.651	0.720
	Log loss	−0.484	−3.609	−4.157	−0.575	−0.621	−0.499
Experiment 5—RoBERTa embedding along with features							
R	Acc	0.506	0.622	0.563	0.711	0.648	0.784
	AP	0.714	0.504	0.723	0.647	0.694	0.726
	Recall	0.619	0.525	0.552	0.532	0.640	0.745
	AUC	0.700	0.669	0.562	0.571	0.695	0.732
	F1	0.626	0.506	0.617	0.656	0.506	0.714
	Log loss	−0.448	−0.417	−0.367	−0.437	−0.376	−0.357
RU	Acc	0.745	0.705	0.641	0.616	0.681	0.720
	AP	0.517	0.695	0.666	0.541	0.512	0.738
	Recall	0.576	0.645	0.744	0.697	0.680	0.788
	AUC	0.500	0.608	0.691	0.678	0.616	0.717
	F1	0.664	0.508	0.607	0.574	0.539	0.692
	Log loss	−0.458	−0.392	−0.433	−0.401	−0.434	−0.288

Table A1. Cont.

Features	Metric	XGBoost	MLP Classifier	KNN	Random Forest	Logistic Regression	Stacking
RP	Acc	0.622	0.553	0.505	0.735	0.545	0.735
	AP	0.594	0.518	0.520	0.702	0.736	0.770
	Recall	0.523	0.514	0.714	0.676	0.505	0.712
	AUC	0.652	0.567	0.741	0.653	0.603	0.689
	F1	0.703	0.518	0.583	0.715	0.567	0.788
	Log loss	−0.376	−0.395	−0.411	−0.366	−0.452	−0.353
RR	Acc	0.641	0.511	0.532	0.537	0.742	0.761
	AP	0.571	0.729	0.516	0.560	0.609	0.857
	Recall	0.521	0.699	0.725	0.739	0.545	0.679
	AUC	0.607	0.747	0.714	0.642	0.634	0.847
	F1	0.537	0.694	0.740	0.724	0.517	0.733
	Log loss	−0.416	−0.427	−0.376	−0.417	−0.364	−0.408
RUP	Acc	0.520	0.543	0.711	0.545	0.713	0.765
	AP	0.708	0.616	0.572	0.520	0.506	0.654
	Recall	0.642	0.536	0.723	0.560	0.515	0.769
	AUC	0.662	0.734	0.742	0.571	0.658	0.659
	F1	0.516	0.526	0.523	0.534	0.619	0.783
	Log loss	−0.409	−0.410	−0.439	−0.443	−0.425	−0.415
RUR	Acc	0.559	0.720	0.525	0.523	0.639	0.704
	AP	0.619	0.528	0.739	0.654	0.742	0.761
	Recall	0.566	0.670	0.748	0.606	0.667	0.694
	AUC	0.574	0.736	0.722	0.632	0.516	0.806
	F1	0.591	0.647	0.738	0.563	0.630	0.726
	Log loss	−0.380	−0.446	−0.370	−0.450	−0.452	−0.347
RPR	Acc	0.681	0.613	0.652	0.736	0.660	0.690
	AP	0.703	0.700	0.662	0.677	0.688	0.737
	Recall	0.548	0.509	0.568	0.623	0.655	0.742
	AUC	0.699	0.661	0.559	0.567	0.646	0.690
	F1	0.625	0.529	0.679	0.648	0.676	0.754
	Log loss	−0.394	−0.444	−0.369	−0.393	−0.358	−0.413
RA	Acc	0.614	0.681	0.636	0.506	0.716	0.717
	AP	0.632	0.647	0.533	0.580	0.671	0.704
	Recall	0.547	0.557	0.723	0.638	0.556	0.798
	AUC	0.571	0.549	0.638	0.610	0.595	0.682
	F1	0.667	0.717	0.592	0.510	0.647	0.677
	Log loss	−0.380	−0.417	−0.425	−0.458	−0.420	−0.309

where

FA	All features
FP	Product-based features only
FPR	Product- and review-based features only
FR	Review-based features only
FU	User-based features only
FUP	User- and product-based features only
FUR	User- and review-based features only
B/D/A/R	BERT/DistilBERT/ALBERT/RoBERTa embeddings only
B/D/A/R U	Embedding type with user-based features only
B/D/A/R P	Embedding type with product-based features only
B/D/A/R R	Embedding type with review-based features only
B/D/A/R UP	Embedding type with user- and product-based features only
B/D/A/R UR	Embedding type with user- and review-based features only
B/D/A/R PR	Embedding type with product- and review-based features only
B/D/A/R A	Embedding type along with all features

Appendix B

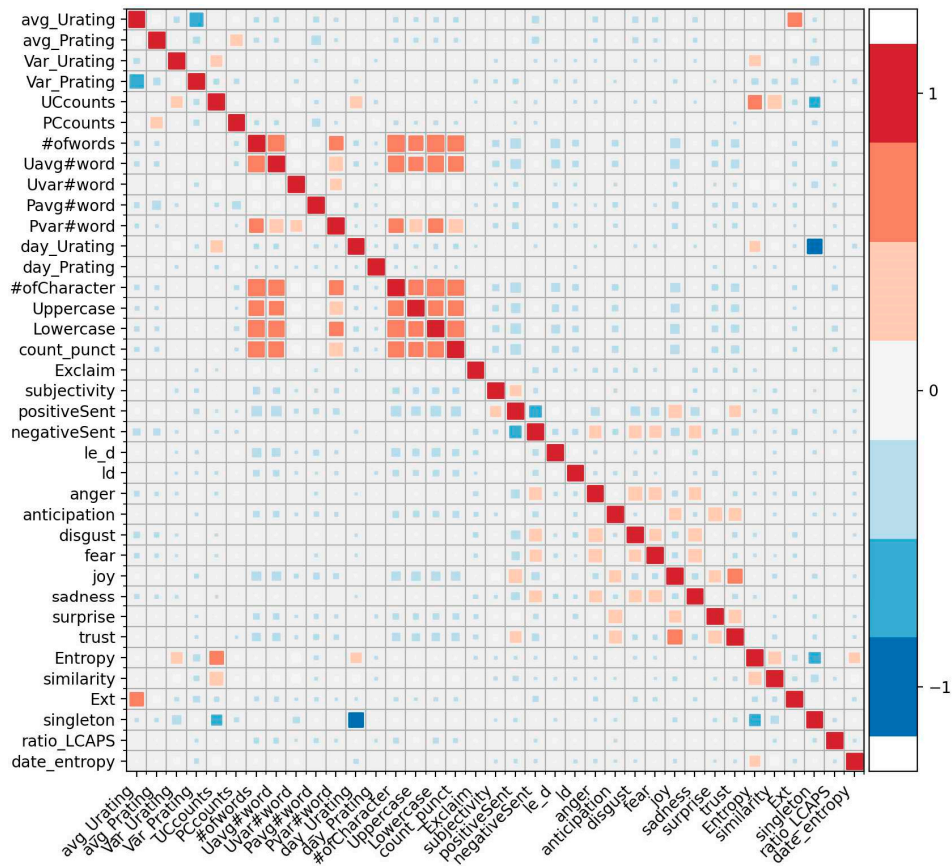


Figure A1. YelpChi correlation plot.

Table A2. Classifier performance on the YelpChi dataset.

Features	Metric	XGBoost	MLP Classifier	KNN	Random Forest	Logistic Regression	Stacking
Experiment 1—Features Only							
FU	Acc	0.71113	0.5	0.765185	0.691177	0.654522	0.800676
	AP	0.808927	0.499997	0.79186	0.77533	0.685625	0.880453
	Recall	0.655139	0.54387	0.714429	0.678298	0.601502	0.773096
	AUC	0.799302	0.499994	0.823125	0.768883	0.70267	0.875608
	F1	0.685183	0.56743	0.746677	0.681941	0.627078	0.791612
	Log loss	−0.54263	−0.69337	−3.63036	−0.66948	−0.64286	−0.44127
FP	Acc	0.59743	0.53969	0.53038	0.56228	0.53549	0.57013
	AP	0.62926	0.55021	0.52407	0.58218	0.55678	0.59872
	Recall	0.60758	0.40958	0.52831	0.56049	0.67472	0.59815
	AUC	0.63950	0.54472	0.53849	0.59616	0.56050	0.60374
	F1	0.60143	0.46270	0.52941	0.56143	0.58725	0.58063
	Log loss	−0.67369	−0.75793	−4.86395	−0.97317	−0.68334	−0.69055
FR	Acc	0.62361	0.58694	0.56834	0.64284	0.61941	0.61363
	AP	0.68549	0.64647	0.56344	0.69827	0.66864	0.66947
	Recall	0.60701	0.41682	0.57618	0.60275	0.49758	0.55846
	AUC	0.67667	0.63427	0.58870	0.69343	0.66978	0.66006
	F1	0.61619	0.42368	0.57109	0.62659	0.56546	0.58490
	Log loss	−0.66959	−0.72660	−4.59863	−0.63181	−0.65315	−0.66721

Table A2. Cont.

Features	Metric	XGBoost	MLP Classifier	KNN	Random Forest	Logistic Regression	Stacking
FUP	Acc	0.65383	0.53739	0.51670	0.63561	0.60965	0.62888
	AP	0.71174	0.56260	0.51535	0.68021	0.61513	0.68704
	Recall	0.64692	0.38959	0.49860	0.63302	0.58537	0.61430
	AUC	0.71513	0.55732	0.52048	0.69200	0.63831	0.68712
	F1	0.65033	0.39399	0.50756	0.63335	0.59907	0.62252
	Log loss	−0.63640	−1.26942	−4.92633	−0.66012	−0.66680	−0.64466
FUR	Acc	0.64430	0.56155	0.54104	0.66016	0.61985	0.63791
	AP	0.71145	0.62165	0.53580	0.72440	0.65851	0.70726
	Recall	0.63616	0.46873	0.52797	0.63594	0.50442	0.61363
	AUC	0.70588	0.62214	0.55463	0.72132	0.66251	0.69799
	F1	0.64042	0.46627	0.53448	0.65020	0.56828	0.62698
	Log loss	−0.65150	−0.90026	−4.95076	−0.61356	−0.65748	−0.64313
FPR	Acc	0.65955	0.59306	0.56239	0.66751	0.61358	0.65271
	AP	0.72651	0.62374	0.55360	0.72748	0.65449	0.71574
	Recall	0.65522	0.54254	0.56250	0.62932	0.54478	0.62372
	AUC	0.72157	0.62499	0.58005	0.72724	0.65237	0.70867
	F1	0.65714	0.56958	0.56202	0.65309	0.58351	0.64123
	Log loss	−0.63693	−0.71511	−4.58457	−0.61016	−0.65758	−0.63310
FA	Acc	0.66431	0.55432	0.53918	0.67081	0.61767	0.65428
	AP	0.73832	0.59675	0.53151	0.73710	0.65282	0.72696
	Recall	0.65601	0.52737	0.52147	0.65063	0.55779	0.61800
	AUC	0.73224	0.59332	0.54845	0.73481	0.65212	0.71915
	F1	0.66059	0.50820	0.53048	0.66307	0.59135	0.64049
	Log loss	−0.62650	−1.14665	−4.78246	−0.60312	−0.65784	−0.62072
Experiment 2—BERT embedding along with features							
B	Acc	0.49383	0.48464	0.49367	0.49552	0.48217	0.49540
	AP	0.49623	0.48270	0.49750	0.49975	0.48081	0.49824
	Recall	0.53910	0.23981	0.65569	0.43108	0.39295	0.55054
	AUC	0.49519	0.47583	0.49494	0.49881	0.47500	0.49809
	F1	0.47191	0.23196	0.53841	0.41614	0.36251	0.47902
	Log loss	−0.71741	−0.70104	−6.37769	−0.99291	−0.69750	−0.72700
BU	Acc	0.62961	0.51996	0.52422	0.61173	0.58521	0.61066
	AP	0.67436	0.56676	0.52222	0.64483	0.59790	0.65503
	Recall	0.61441	0.65214	0.51250	0.59894	0.59445	0.54927
	AUC	0.68177	0.56623	0.53327	0.65514	0.61816	0.66025
	F1	0.62260	0.52312	0.51845	0.60508	0.58748	0.58398
	Log loss	−0.66111	−1.34999	−5.60317	−0.71427	−0.67780	−0.66742
BP	Acc	0.59278	0.54322	0.53111	0.56425	0.53829	0.56514
	AP	0.62380	0.55748	0.52457	0.58081	0.55666	0.59568
	Recall	0.60298	0.35261	0.52898	0.55836	0.65925	0.58201
	AUC	0.63258	0.55808	0.53947	0.59547	0.56279	0.60011
	F1	0.59681	0.43108	0.53011	0.56161	0.58381	0.57150
	Log loss	−0.67896	−0.73740	−4.83348	−0.79076	−0.68289	−0.68931
BR	Acc	0.62456	0.60007	0.56828	0.63785	0.62232	0.62182
	AP	0.68158	0.67272	0.56340	0.69477	0.67003	0.67664
	Recall	0.61004	0.74841	0.57640	0.59322	0.50117	0.60432
	AUC	0.67462	0.67338	0.58861	0.68964	0.67044	0.66903
	F1	0.61810	0.64291	0.57116	0.61950	0.56904	0.61400
	Log loss	−0.66977	−0.69504	−4.60242	−0.63447	−0.65263	−0.67041
BUP	Acc	0.64699	0.54294	0.51670	0.64009	0.60820	0.63224
	AP	0.70374	0.57535	0.51535	0.68245	0.61290	0.68689
	Recall	0.63952	0.43347	0.49860	0.63235	0.57506	0.61071
	AUC	0.70625	0.56989	0.52048	0.69582	0.63471	0.68880
	F1	0.64345	0.46305	0.50756	0.63602	0.59404	0.62277
	Log loss	−0.64318	−1.11987	−4.92633	−0.64184	−0.66831	−0.64627

Table A2. Cont.

Features	Metric	XGBoost	MLP Classifier	KNN	Random Forest	Logistic Regression	Stacking
BUR	Acc	0.64256	0.54519	0.54104	0.65854	0.61800	0.64183
	AP	0.70852	0.60486	0.53580	0.72422	0.65622	0.70850
	Recall	0.63470	0.64353	0.52797	0.63235	0.49489	0.61699
	AUC	0.70344	0.59867	0.55463	0.72017	0.66007	0.70066
	F1	0.63834	0.55723	0.53448	0.64777	0.56188	0.63025
	Log loss	−0.65133	−1.03051	−4.95076	−0.61468	−0.65853	−0.63997
BPR	Acc	0.65674	0.60001	0.56239	0.66599	0.61206	0.64699
	AP	0.72367	0.63345	0.55360	0.72513	0.65288	0.71094
	Recall	0.65051	0.53887	0.56250	0.62966	0.55241	0.63325
	AUC	0.71919	0.63085	0.58005	0.72677	0.65048	0.70725
	F1	0.65376	0.56436	0.56202	0.65237	0.58555	0.64143
	Log loss	−0.63726	−0.77768	−4.58457	−0.61157	−0.65776	−0.63340
BA	Acc	0.65826	0.54446	0.53918	0.66924	0.61728	0.65086
	AP	0.73331	0.58611	0.53151	0.73341	0.65409	0.72270
	Recall	0.64850	0.63670	0.52147	0.64491	0.55241	0.62899
	AUC	0.72661	0.58803	0.54845	0.73304	0.65253	0.71329
	F1	0.65400	0.57299	0.53048	0.65985	0.58877	0.64209
	Log loss	−0.63099	−1.05104	−4.78246	−0.60592	−0.65750	−0.62920
Experiment 3—ALBERTA embedding along with features							
A	Acc	0.48369	0.47371	0.48206	0.47315	0.47276	0.50987
	AP	0.48533	0.46752	0.48729	0.47585	0.47534	0.51442
	Recall	0.48145	0.38010	0.48111	0.43660	0.41631	0.52001
	AUC	0.47515	0.45449	0.47645	0.46189	0.46219	0.51184
	F1	0.48191	0.41320	0.48132	0.45200	0.43559	0.51475
	Log loss	−0.83122	−0.79217	−5.47128	−0.71731	−0.70086	−0.71996
AU	Acc	0.60483	0.56407	0.52433	0.59681	0.58757	0.59513
	AP	0.64308	0.59195	0.52232	0.62177	0.60094	0.63293
	Recall	0.59737	0.35124	0.51261	0.57046	0.59816	0.54770
	AUC	0.65114	0.59229	0.53337	0.63769	0.62144	0.63945
	F1	0.60022	0.40920	0.51857	0.58351	0.59031	0.57287
	Log loss	−0.73088	−0.84285	−5.60492	−0.66523	−0.67647	−0.67737
AP	Acc	0.54956	0.53829	0.53117	0.51480	0.54429	0.55169
	AP	0.56568	0.56374	0.52509	0.51600	0.56644	0.56396
	Recall	0.55847	0.53604	0.52820	0.49030	0.61701	0.57092
	AUC	0.57134	0.56890	0.53984	0.52605	0.57734	0.57387
	F1	0.55345	0.51101	0.52977	0.50209	0.55244	0.55695
	Log loss	−0.76990	−0.76927	−4.81465	−0.69993	−0.68179	−0.70383
AR	Acc	0.59328	0.62641	0.56834	0.60657	0.61963	0.60903
	AP	0.64217	0.68479	0.56318	0.64152	0.66859	0.65755
	Recall	0.57484	0.56464	0.57652	0.56934	0.49007	0.56452
	AUC	0.63498	0.68310	0.58843	0.64417	0.66980	0.65538
	F1	0.58422	0.59410	0.57122	0.59014	0.56109	0.58881
	Log loss	−0.74441	−0.64562	−4.60443	−0.66054	−0.65389	−0.66765
AUP	Acc	0.61582	0.55561	0.51665	0.60629	0.60892	0.61520
	AP	0.66030	0.59065	0.51524	0.62884	0.61239	0.65382
	Recall	0.60914	0.57091	0.49871	0.58122	0.57674	0.58033
	AUC	0.66661	0.58943	0.52033	0.64852	0.63453	0.65897
	F1	0.61215	0.54188	0.50759	0.59433	0.59520	0.60009
	Log loss	−0.71584	−0.81014	−4.92646	−0.66074	−0.66843	−0.66356
AUR	Acc	0.61997	0.59620	0.54092	0.63466	0.61649	0.62193
	AP	0.68166	0.64351	0.53583	0.67895	0.65662	0.68058
	Recall	0.61049	0.51687	0.52797	0.60017	0.49074	0.60398
	AUC	0.67482	0.64514	0.55458	0.68045	0.66048	0.67007
	F1	0.61506	0.55206	0.53443	0.62002	0.55882	0.61382
	Log loss	−0.70762	−0.69041	−4.94896	−0.64238	−0.65808	−0.65507

Table A2. Cont.

Features	Metric	XGBoost	MLP Classifier	KNN	Random Forest	Logistic Regression	Stacking
APR	Acc	0.62737	0.61330	0.56239	0.62120	0.61537	0.62199
	AP	0.68230	0.64491	0.55352	0.66532	0.65502	0.67302
	Recall	0.61755	0.51854	0.56239	0.57988	0.54052	0.59546
	AUC	0.68166	0.65355	0.58000	0.66635	0.65319	0.66887
	F1	0.62260	0.56521	0.56198	0.60352	0.58274	0.60987
	Log loss	−0.70174	−0.70164	−4.58461	−0.64976	−0.65773	−0.66110
AA	Acc	0.63208	0.55057	0.53913	0.63785	0.61694	0.62417
	AP	0.69316	0.60826	0.53140	0.68747	0.65390	0.68721
	Recall	0.61968	0.74716	0.52158	0.60634	0.55476	0.60073
	AUC	0.68832	0.59833	0.54830	0.68888	0.65266	0.67797
	F1	0.62665	0.62410	0.53050	0.62428	0.58940	0.61232
	Log loss	−0.69416	−0.89231	−4.78630	−0.63770	−0.65728	−0.65726
Experiment 4—DistilBERT embedding along with features							
D	Acc	0.47029	0.47197	0.47236	0.47477	0.46524	0.51486
	AP	0.47016	0.46487	0.47814	0.46857	0.46419	0.52854
	Recall	0.46855	0.34713	0.45992	0.43178	0.43424	0.53582
	AUC	0.45918	0.45048	0.46053	0.45627	0.44587	0.52520
	F1	0.46909	0.39436	0.46528	0.44951	0.44479	0.52411
	Log loss	−0.86615	−0.83000	−5.87407	−0.71713	−0.70460	−0.71702
DU	Acc	0.59676	0.53520	0.52433	0.59127	0.58785	0.59093
	AP	0.62522	0.59079	0.52238	0.61305	0.60100	0.62071
	Recall	0.59131	0.62365	0.51261	0.55959	0.59805	0.62787
	AUC	0.63656	0.58655	0.53342	0.62867	0.62152	0.62852
	F1	0.59312	0.52110	0.51857	0.57587	0.59043	0.60520
	Log loss	−0.76028	−0.86846	−5.60488	−0.66880	−0.67646	−0.70422
DP	Acc	0.53694	0.53560	0.53139	0.51261	0.55029	0.54188
	AP	0.54707	0.56092	0.52510	0.50675	0.56409	0.55104
	Recall	0.53268	0.59539	0.52842	0.47652	0.71981	0.57631
	AUC	0.55237	0.56547	0.53982	0.51780	0.57730	0.56027
	F1	0.53477	0.53733	0.53000	0.49362	0.61397	0.55495
	Log loss	−0.80466	−0.78940	−4.82944	−0.70176	−0.68105	−0.71054
DR	Acc	0.58779	0.61817	0.56828	0.60276	0.62092	0.60382
	AP	0.63748	0.68121	0.56319	0.63801	0.66923	0.65901
	Recall	0.57338	0.53323	0.57652	0.57035	0.49410	0.52582
	AUC	0.62848	0.68085	0.58841	0.63977	0.67060	0.65374
	F1	0.58037	0.56076	0.57121	0.58804	0.56402	0.56009
	Log loss	−0.75852	−0.65251	−4.60445	−0.66259	−0.65396	−0.67340
DUP	Acc	0.61363	0.55825	0.51659	0.60920	0.60853	0.60932
	AP	0.65307	0.60168	0.51522	0.63211	0.61257	0.64361
	Recall	0.61352	0.59802	0.49871	0.58582	0.57562	0.59075
	AUC	0.66078	0.60567	0.52029	0.64986	0.63452	0.65319
	F1	0.61242	0.55591	0.50756	0.59836	0.59450	0.59990
	Log loss	−0.72987	−0.82009	−4.92650	−0.66033	−0.66840	−0.67235
DUR	Acc	0.60836	0.58908	0.54092	0.62877	0.61834	0.61083
	AP	0.66859	0.63012	0.53583	0.67427	0.65825	0.66723
	Recall	0.59558	0.39077	0.52797	0.59277	0.49915	0.55723
	AUC	0.66102	0.62858	0.55458	0.67663	0.66219	0.65571
	F1	0.60203	0.46161	0.53443	0.61345	0.56528	0.58372
	Log loss	−0.73206	−0.80277	−4.94896	−0.64445	−0.65737	−0.67706
DPR	Acc	0.62081	0.61094	0.56239	0.61974	0.61419	0.62204
	AP	0.67491	0.65303	0.55352	0.66246	0.65485	0.67071
	Recall	0.60847	0.54793	0.56239	0.58694	0.54142	0.58773
	AUC	0.67233	0.65925	0.58000	0.66443	0.65333	0.66624
	F1	0.61511	0.58154	0.56198	0.60549	0.58194	0.60737
	Log loss	−0.71715	−0.67030	−4.58461	−0.65107	−0.65742	−0.66416
DA	Acc	0.62569	0.56783	0.53913	0.63505	0.61700	0.62798
	AP	0.68772	0.61811	0.53140	0.68236	0.65242	0.68189
	Recall	0.61834	0.62072	0.52158	0.60589	0.55566	0.62417
	AUC	0.68275	0.60863	0.54830	0.68467	0.65092	0.67610
	F1	0.62187	0.57635	0.53050	0.62261	0.58987	0.62618
	Log loss	−0.70495	−0.82448	−4.78630	−0.63988	−0.65817	−0.65485

Table A2. Cont.

Features	Metric	XGBoost	MLP Classifier	KNN	Random Forest	Logistic Regression	Stacking
Experiment 5—RoBERTa embedding along with features							
R	Acc	0.47931	0.47797	0.48128	0.47920	0.46379	0.51200
	AP	0.48200	0.48120	0.48612	0.47755	0.46472	0.51173
	Recall	0.47629	0.47102	0.45992	0.42853	0.43582	0.52080
	AUC	0.47277	0.46942	0.47581	0.46814	0.44923	0.51279
	F1	0.47746	0.47355	0.46960	0.45051	0.44618	0.51605
	Log loss	−0.83244	−2.58975	−5.51829	−0.70854	−0.70808	−0.72229
RU	Acc	0.60304	0.57366	0.52439	0.60629	0.58549	0.60382
	AP	0.64069	0.60649	0.52231	0.63731	0.59801	0.63886
	Recall	0.59748	0.58854	0.51261	0.58021	0.59367	0.55634
	AUC	0.64844	0.61228	0.53341	0.65130	0.61837	0.64450
	F1	0.59917	0.53846	0.51860	0.59386	0.58728	0.57900
	Log loss	−0.73830	−0.77159	−5.60303	−0.65943	−0.67778	−0.67932
RP	Acc	0.55062	0.55455	0.53223	0.52730	0.53560	0.54614
	AP	0.56343	0.56863	0.52542	0.52880	0.55689	0.56287
	Recall	0.55051	0.47585	0.52977	0.49860	0.67483	0.56384
	AUC	0.57284	0.57372	0.54043	0.54304	0.56067	0.56754
	F1	0.55046	0.50042	0.53108	0.51304	0.58741	0.54896
	Log loss	−0.77311	−0.75215	−4.82701	−0.69216	−0.68331	−0.70985
RR	Acc	0.59861	0.62798	0.56845	0.60988	0.61694	0.60982
	AP	0.64606	0.67880	0.56372	0.64904	0.66605	0.66122
	Recall	0.58347	0.64312	0.57674	0.57932	0.49545	0.58291
	AUC	0.63954	0.67881	0.58896	0.65159	0.66718	0.65405
	F1	0.59106	0.63331	0.57139	0.59648	0.56300	0.59829
	Log loss	−0.73955	−0.64498	−4.59471	−0.65686	−0.65495	−0.66775
RUP	Acc	0.62524	0.56806	0.51670	0.61660	0.60864	0.62008
	AP	0.66729	0.59355	0.51535	0.64873	0.61245	0.65451
	Recall	0.61834	0.50724	0.49860	0.59546	0.57618	0.59804
	AUC	0.67467	0.60613	0.52048	0.66379	0.63468	0.66343
	F1	0.62158	0.50549	0.50756	0.60712	0.59479	0.61010
	Log loss	−0.70950	−0.85802	−4.92633	−0.65407	−0.66836	−0.66392
RUR	Acc	0.62042	0.57988	0.54104	0.63796	0.61851	0.62552
	AP	0.67961	0.63897	0.53580	0.68624	0.65603	0.68307
	Recall	0.61105	0.51134	0.52797	0.60084	0.50408	0.58885
	AUC	0.67550	0.63763	0.55463	0.68751	0.66042	0.67341
	F1	0.61536	0.51683	0.53448	0.62255	0.56713	0.61014
	Log loss	−0.71209	−0.70740	−4.95076	−0.63859	−0.65845	−0.65642
RPR	Acc	0.62574	0.60337	0.56239	0.62894	0.61408	0.62182
	AP	0.68523	0.63826	0.55360	0.67466	0.65332	0.67625
	Recall	0.61688	0.51104	0.56250	0.59019	0.54826	0.60376
	AUC	0.68058	0.64459	0.58005	0.67650	0.65021	0.67162
	F1	0.62098	0.54871	0.56202	0.61310	0.58567	0.61299
	Log loss	−0.70495	−0.68527	−4.58457	−0.64550	−0.65827	−0.65922
RA	Acc	0.63735	0.57574	0.53918	0.64222	0.61834	0.63068
	AP	0.69897	0.61613	0.53151	0.69679	0.65314	0.69176
	Recall	0.62630	0.64952	0.52147	0.60847	0.55857	0.63907
	AUC	0.69565	0.61403	0.54845	0.69870	0.65240	0.68689
	F1	0.63208	0.59398	0.53048	0.62843	0.59211	0.63230
	Log loss	−0.68758	−0.80225	−4.78246	−0.63235	−0.65753	−0.64827

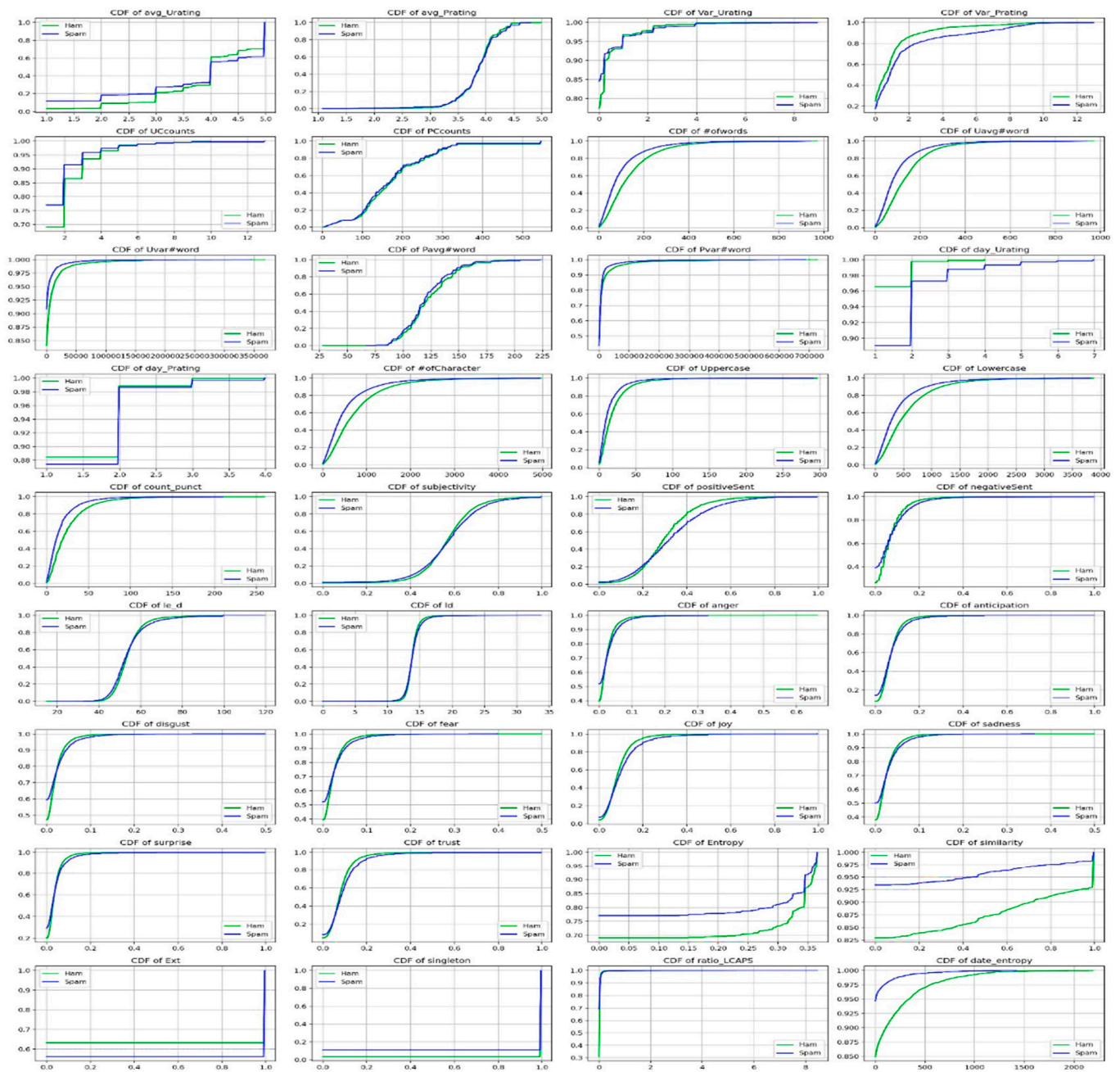


Figure A2. Cumulative distribution function of YelpChi dataset features.

Appendix C

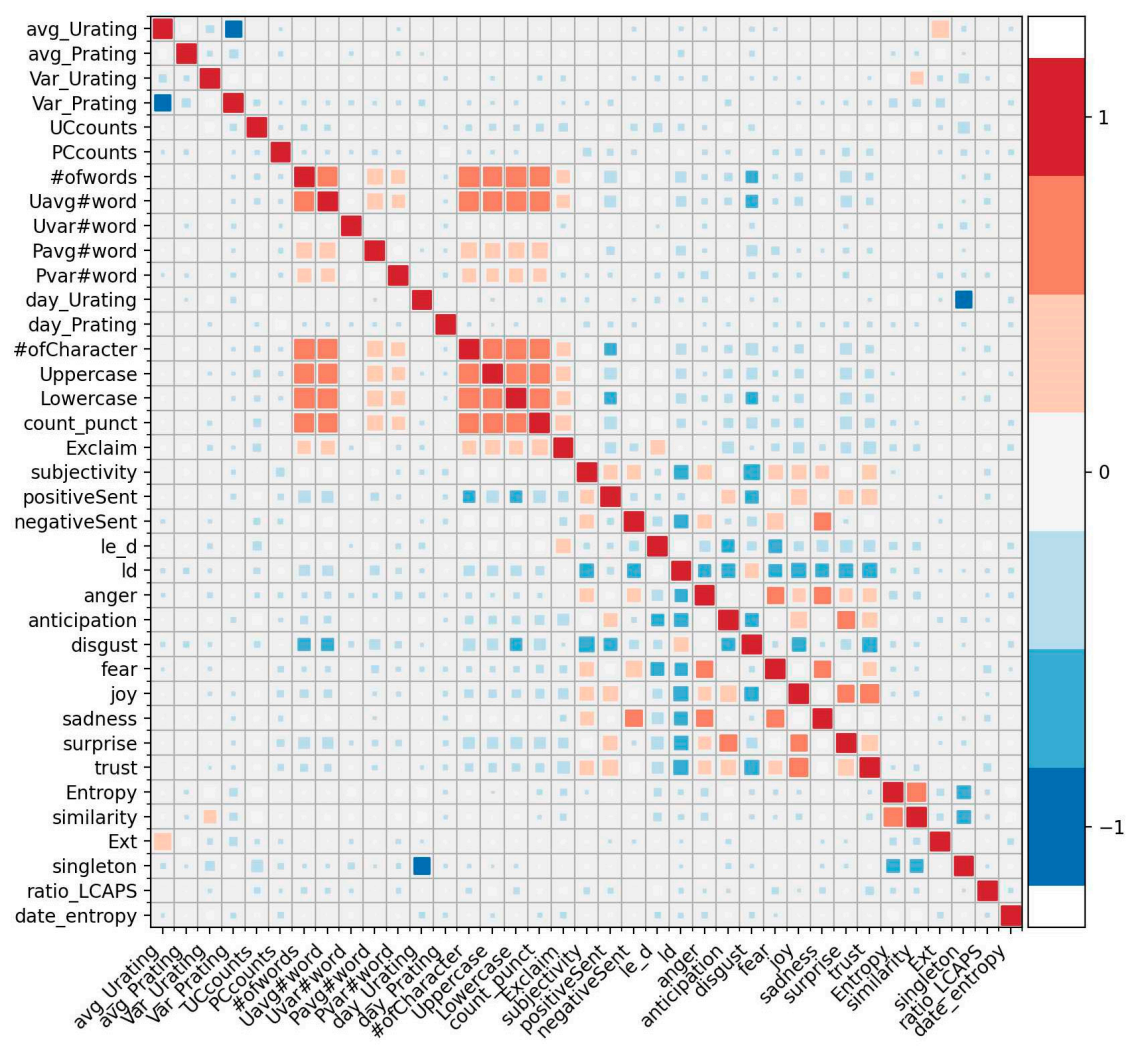


Figure A3. Correlation plot of YelpNYC features.

Table A3. Classifier performance on YelpNYC dataset.

Feature	Metric	XGBoost	MLP Classifier	KNN	Random Forest	Logistic Regression	Stacking
Experiment 1—Features Only							
FU	Acc	0.823343	0.507903	0.874814	0.825105	0.628494	0.897397
	AP	0.909526	0.507689	0.880684	0.882260	0.632096	0.939389
	Recall	0.770096	0.218544	0.827654	0.789264	0.623777	0.888220
	AUC	0.904265	0.508622	0.904942	0.885531	0.660067	0.943354
	F1	0.809802	0.169293	0.864504	0.815849	0.624963	0.895255
	Log loss	−0.400635	−0.690340	−2.846283	−1.449639	−0.659209	−0.294573
FP	Acc	0.664308	0.584316	0.618287	0.643215	0.570964	0.655741
	AP	0.703191	0.581352	0.609889	0.676881	0.602739	0.689264
	Recall	0.625403	0.595228	0.619845	0.592246	0.620306	0.623424
	AUC	0.716910	0.610641	0.649470	0.686536	0.609436	0.703967
	F1	0.646650	0.581673	0.617478	0.621878	0.588933	0.640241
	Log loss	−0.623862	−0.677843	−4.608506	−0.975397	−0.680482	−0.633884

Table A3. Cont.

Feature	Metric	XGBoost	MLP Classifier	KNN	Random Forest	Logistic Regression	Stacking
FR	Acc	0.689671	0.651037	0.632561	0.680832	0.599851	0.667819
	AP	0.775114	0.728550	0.633415	0.757692	0.658044	0.765502
	Recall	0.632886	0.588152	0.594008	0.630365	0.484397	0.580588
	AUC	0.766779	0.724823	0.669960	0.751487	0.646293	0.752572
	F1	0.665541	0.611735	0.612640	0.658793	0.543367	0.621959
	Log loss	−0.571589	−0.624506	−4.471583	−0.726444	−0.657270	−0.591367
FUP	Acc	0.832859	0.583747	0.811807	0.855239	0.654561	0.896408
	AP	0.917304	0.580612	0.825567	0.923462	0.681059	0.950505
	Recall	0.769337	0.248204	0.798292	0.827274	0.584492	0.873797
	AUC	0.912180	0.587628	0.870004	0.919285	0.690079	0.947489
	F1	0.817698	0.342292	0.802861	0.846219	0.624243	0.889673
	Log loss	−0.383452	−0.668992	−3.036401	−0.716218	−0.644859	−0.321913
FUR	Acc	0.808933	0.532723	0.862952	0.833537	0.629050	0.887339
	AP	0.902208	0.533478	0.874722	0.891534	0.681331	0.936354
	Recall	0.759930	0.107252	0.818083	0.802088	0.693182	0.871221
	AUC	0.895932	0.533617	0.901224	0.892144	0.683290	0.938758
	F1	0.795739	0.170673	0.852024	0.825147	0.649214	0.883618
	Log loss	−0.413841	−0.684622	−2.822766	−1.353577	−0.645459	−0.328789
FPR	Acc	0.737902	0.613190	0.640287	0.721947	0.609028	0.730568
	AP	0.839716	0.638369	0.641530	0.819747	0.643872	0.834517
	Recall	0.642592	0.636682	0.643975	0.673743	0.455605	0.625946
	AUC	0.818748	0.671390	0.681163	0.795934	0.627850	0.809419
	F1	0.704492	0.603431	0.639521	0.704214	0.529420	0.692354
	Log loss	−0.512079	−0.656739	−4.299946	−0.728527	−0.668499	−0.520907
FA	Acc	0.826162	0.579111	0.809137	0.859713	0.653138	0.891053
	AP	0.913784	0.580144	0.824329	0.928175	0.693019	0.949228
	Recall	0.758710	0.198563	0.792517	0.832208	0.589047	0.869730
	AUC	0.907599	0.579813	0.867165	0.923234	0.692810	0.945513
	F1	0.808846	0.312972	0.799218	0.850779	0.624842	0.883170
	Log loss	−0.392645	−0.666210	−3.106835	−0.716756	−0.639341	−0.348125
Experiment 2—BERT embedding along with features							
B	Acc	0.422990	0.500136	0.423058	0.424481	0.500014	0.526108
	AP	0.459620	0.506326	0.473826	0.467110	0.689688	0.535742
	Recall	0.407808	0.799214	0.407862	0.389698	0.000027	0.586065
	AUC	0.423943	0.504183	0.423112	0.417757	0.673964	0.536971
	F1	0.347004	0.533219	0.347164	0.335637	0.000054	0.492660
	Log loss	−3.008050	−0.708783	−19.921263	−14.641646	−0.693594	−1.748505
BU	Acc	0.482595	0.504528	0.446591	0.558506	0.629484	0.662342
	AP	0.710685	0.505975	0.483985	0.613428	0.625751	0.773045
	Recall	0.475776	0.620794	0.458398	0.604799	0.633672	0.809923
	AUC	0.678794	0.507355	0.447370	0.611861	0.655308	0.766311
	F1	0.433148	0.435770	0.425374	0.568692	0.631713	0.710851
	Log loss	−1.866017	−1.027121	−16.278267	−1.770812	−0.662008	−0.799907
BP	Acc	0.423397	0.493941	0.430704	0.450264	0.566951	0.569351
	AP	0.537193	0.499341	0.476872	0.499320	0.612592	0.615051
	Recall	0.407781	0.574895	0.427355	0.440152	0.462248	0.758303
	AUC	0.519306	0.499440	0.434066	0.493694	0.619957	0.625872
	F1	0.347158	0.391167	0.381895	0.414642	0.469761	0.631950
	Log loss	−2.630268	−0.792793	−18.583398	−2.389219	−0.690698	−0.762318

Table A3. Cont.

Feature	Metric	XGBoost	MLP Classifier	KNN	Random Forest	Logistic Regression	Stacking
BR	Acc	0.435407	0.511414	0.423844	0.555822	0.581266	0.586634
	AP	0.593029	0.507893	0.473875	0.552442	0.642640	0.674425
	Recall	0.410790	0.035489	0.409028	0.615725	0.467724	0.720320
	AUC	0.554367	0.506110	0.424093	0.565215	0.623383	0.647706
	F1	0.355190	0.058110	0.348722	0.570581	0.487494	0.633000
	Log loss	−2.519455	−0.726955	−19.818545	−3.837709	−0.675526	−0.749183
BUP	Acc	0.520361	0.509706	0.468307	0.631896	0.650332	0.633144
	AP	0.735965	0.510956	0.489910	0.755888	0.675151	0.812676
	Recall	0.532086	0.182486	0.489386	0.689711	0.567439	0.881090
	AUC	0.708055	0.514431	0.470803	0.743923	0.685030	0.800779
	F1	0.495957	0.148084	0.455417	0.650990	0.610003	0.709953
	Log loss	−1.670196	−0.735608	−14.617474	−0.800951	−0.650039	−0.965161
BUR	Acc	0.495405	0.530500	0.448678	0.616945	0.627870	0.665433
	AP	0.718306	0.527272	0.484911	0.714805	0.684597	0.794144
	Recall	0.496598	0.082798	0.461570	0.679626	0.648855	0.755131
	AUC	0.685033	0.528867	0.450079	0.709891	0.682087	0.775330
	F1	0.452778	0.113117	0.428404	0.635959	0.623350	0.692075
	Log loss	−1.777515	−0.732621	−16.141086	−1.082843	−0.651799	−0.774105
BPR	Acc	0.457761	0.525756	0.432818	0.624590	0.600854	0.660946
	AP	0.667747	0.525011	0.478011	0.725989	0.635883	0.780919
	Recall	0.445222	0.121377	0.430283	0.694212	0.348136	0.730216
	AUC	0.620116	0.527512	0.436706	0.709531	0.619900	0.763433
	F1	0.400006	0.127259	0.387251	0.645666	0.444738	0.682300
	Log loss	−2.138377	−0.716985	−18.263558	−0.961516	−0.676973	−0.677129
BA	Acc	0.540111	0.563129	0.469351	0.676142	0.651633	0.677240
	AP	0.743679	0.561517	0.490644	0.826319	0.690723	0.811853
	Recall	0.567033	0.183571	0.491338	0.738132	0.564322	0.812119
	AUC	0.713272	0.564857	0.471991	0.806044	0.690689	0.802636
	F1	0.524759	0.263176	0.457173	0.697021	0.607261	0.725856
	Log loss	−1.582503	−0.714157	−14.567983	−0.685565	−0.644227	−0.840616
Experiment 3—ALBERTA embedding along with features							
A	Acc	0.464403	0.473404	0.491514	0.466653	0.468917	0.500136
	AP	0.470288	0.470329	0.496642	0.469312	0.472001	0.504103
	Recall	0.414423	0.327586	0.380859	0.435543	0.446577	0.381591
	AUC	0.453517	0.453733	0.491426	0.452760	0.457566	0.501986
	F1	0.434175	0.381154	0.424206	0.448006	0.456089	0.415525
	Log loss	−0.726548	−0.733304	−6.115318	−0.731689	−0.702506	−0.710033
AU	Acc	0.793832	0.500122	0.874637	0.674285	0.628684	0.882351
	AP	0.892065	0.507346	0.880517	0.714243	0.632190	0.932566
	Recall	0.748190	0.797560	0.827654	0.661678	0.624075	0.870462
	AUC	0.882817	0.508350	0.904842	0.722544	0.660185	0.933315
	F1	0.780933	0.534169	0.864364	0.667723	0.625193	0.878646
	Log loss	−0.432160	−0.691376	−2.848825	−1.235472	−0.659607	−0.345269
AP	Acc	0.633672	0.598617	0.616131	0.577213	0.574814	0.637359
	AP	0.654097	0.602528	0.607836	0.585658	0.604687	0.663407
	Recall	0.604229	0.635733	0.615806	0.564674	0.594063	0.606588
	AUC	0.677220	0.634402	0.646702	0.605874	0.611449	0.683147
	F1	0.618957	0.605271	0.614553	0.570589	0.581739	0.620935
	Log loss	−0.656376	−0.669623	−4.658350	−0.844455	−0.681444	−0.647074

Table A3. Cont.

Feature	Metric	XGBoost	MLP Classifier	KNN	Random Forest	Logistic Regression	Stacking
AR	Acc	0.675790	0.652853	0.624305	0.650183	0.605002	0.661028
	AP	0.756145	0.734260	0.625129	0.707619	0.660615	0.746476
	Recall	0.629551	0.522326	0.615941	0.610275	0.467724	0.551444
	AUC	0.745555	0.723804	0.661227	0.705823	0.651008	0.731871
	F1	0.655765	0.587959	0.618985	0.632157	0.535148	0.612574
	Log loss	−0.595190	−0.614549	−4.915387	−0.847924	−0.656562	−0.602799
AUP	Acc	0.810668	0.616402	0.811807	0.713718	0.652881	0.889576
	AP	0.901556	0.621972	0.825485	0.781806	0.678018	0.939777
	Recall	0.747079	0.319805	0.798373	0.689006	0.584113	0.869974
	AUC	0.893803	0.621298	0.869939	0.775727	0.687660	0.940048
	F1	0.793766	0.448791	0.802893	0.702428	0.622878	0.883347
	Log loss	−0.413048	−0.650256	−3.037827	−0.614518	−0.645798	−0.340803
AUR	Acc	0.789196	0.591907	0.862830	0.697682	0.631137	0.883937
	AP	0.888069	0.603804	0.874594	0.748923	0.687633	0.931048
	Recall	0.737834	0.267046	0.818219	0.684153	0.678352	0.856852
	AUC	0.878995	0.602938	0.901130	0.752502	0.684898	0.933928
	F1	0.773984	0.365496	0.851965	0.690371	0.643397	0.878501
	Log loss	−0.438109	−0.672227	−2.824785	−1.100490	−0.646827	−0.330495
APR	Acc	0.721391	0.634770	0.637861	0.684818	0.606263	0.719954
	AP	0.823354	0.678143	0.639962	0.756622	0.644614	0.817200
	Recall	0.640640	0.619764	0.645710	0.673824	0.467453	0.605341
	AUC	0.801161	0.698737	0.679219	0.743785	0.629761	0.791903
	F1	0.691520	0.618031	0.638533	0.678450	0.532895	0.677537
	Log loss	−0.535193	−0.641814	−4.296107	−0.696352	−0.667802	−0.540149
AA	Acc	0.809218	0.636939	0.809123	0.738634	0.653003	0.875803
	AP	0.901654	0.654658	0.824281	0.826716	0.691365	0.939394
	Recall	0.740545	0.396421	0.792599	0.711292	0.582269	0.863278
	AUC	0.892682	0.652279	0.867124	0.811294	0.691627	0.939105
	F1	0.790648	0.517479	0.799240	0.727168	0.620230	0.871564
	Log loss	−0.415887	−0.639657	−3.107363	−0.578119	−0.640208	−0.345767
Experiment 4—DistilBERT embedding along with features							
D	Acc	0.463996	0.472455	0.480615	0.466992	0.468930	0.515969
	AP	0.469205	0.469084	0.489727	0.469094	0.470085	0.522134
	Recall	0.421174	0.325146	0.479761	0.436194	0.389806	0.459347
	AUC	0.452612	0.453532	0.479049	0.452949	0.454740	0.522263
	F1	0.437970	0.375672	0.478031	0.448541	0.420465	0.464800
	Log loss	−0.726284	−0.716010	−6.663249	−0.730338	−0.702663	−0.703843
DU	Acc	0.796299	0.538715	0.874570	0.676345	0.628806	0.889250
	AP	0.893267	0.555163	0.880386	0.714290	0.632332	0.929447
	Recall	0.752230	0.392707	0.827626	0.662844	0.624644	0.885021
	AUC	0.884858	0.558087	0.904701	0.723111	0.660353	0.934898
	F1	0.783573	0.348711	0.864298	0.669479	0.625503	0.887415
	Log loss	−0.430092	−0.702534	−2.852570	−1.250693	−0.659422	−0.316440
DP	Acc	0.634906	0.591528	0.616890	0.579002	0.573919	0.631720
	AP	0.657181	0.607242	0.606429	0.585461	0.603819	0.657528
	Recall	0.603985	0.580561	0.618680	0.567765	0.604853	0.598699
	AUC	0.678673	0.628607	0.645358	0.605122	0.610571	0.674496
	F1	0.619277	0.581567	0.616154	0.573046	0.584384	0.615222
	Log loss	−0.654246	−0.670598	−4.704216	−0.843267	−0.681525	−0.653589

Table A3. Cont.

Feature	Metric	XGBoost	MLP Classifier	KNN	Random Forest	Logistic Regression	Stacking
DR	Acc	0.675735	0.645208	0.625173	0.650400	0.594117	0.661543
	AP	0.757306	0.730009	0.623776	0.707217	0.647653	0.748202
	Recall	0.624780	0.530595	0.618680	0.610845	0.463847	0.574678
	AUC	0.745982	0.718661	0.660244	0.706204	0.631652	0.732337
	F1	0.654033	0.581408	0.620421	0.632626	0.520709	0.622468
	Log loss	−0.593284	−0.623136	−5.079042	−0.850721	−0.662627	−0.600561
DUP	Acc	0.812105	0.618802	0.811766	0.712675	0.653138	0.886553
	AP	0.902561	0.622518	0.825480	0.781093	0.678766	0.937333
	Recall	0.747702	0.320618	0.798373	0.686919	0.585495	0.867751
	AUC	0.895166	0.624513	0.869917	0.775566	0.687759	0.939136
	F1	0.795127	0.455017	0.802860	0.701179	0.623620	0.880667
	Log loss	−0.410696	−0.647028	−3.038303	−0.608179	−0.645595	−0.338815
DUR	Acc	0.791379	0.593331	0.862776	0.698306	0.630690	0.880019
	AP	0.888875	0.595571	0.874543	0.749142	0.688391	0.932360
	Recall	0.747675	0.267643	0.818165	0.685292	0.685970	0.869974
	AUC	0.879769	0.595452	0.901062	0.752670	0.684473	0.934043
	F1	0.778586	0.349084	0.851900	0.691091	0.646794	0.876226
	Log loss	−0.437108	−0.663626	−2.826221	−1.090813	−0.646802	−0.355753
DPR	Acc	0.722882	0.636180	0.637969	0.684818	0.606859	0.718666
	AP	0.825183	0.687403	0.638021	0.757165	0.643064	0.817608
	Recall	0.639718	0.598780	0.646469	0.672875	0.468971	0.610411
	AUC	0.801725	0.702461	0.677668	0.743784	0.626595	0.791503
	F1	0.692402	0.602246	0.638982	0.678206	0.532455	0.678204
	Log loss	−0.533660	−0.647868	−4.344742	−0.697351	−0.667907	−0.539344
DA	Acc	0.810641	0.580168	0.809082	0.737685	0.653585	0.880060
	AP	0.902527	0.589733	0.824251	0.826251	0.692341	0.937490
	Recall	0.742714	0.380371	0.792572	0.711726	0.591026	0.855849
	AUC	0.894422	0.589894	0.867089	0.810500	0.692090	0.937472
	F1	0.791963	0.400849	0.799196	0.726631	0.626052	0.872295
	Log loss	−0.413589	−0.746263	−3.108298	−0.575551	−0.639337	−0.327304
Experiment 5—RoBERTa embedding along with features							
R	Acc	0.463942	0.465962	0.499634	0.465826	0.462885	0.500732
	AP	0.469524	0.470331	0.494831	0.468111	0.467235	0.508341
	Recall	0.421282	0.416023	0.596909	0.435055	0.415209	0.428955
	AUC	0.452613	0.452955	0.491182	0.451633	0.449527	0.501905
	F1	0.438078	0.431994	0.537976	0.447377	0.434427	0.440534
	Log loss	−0.724444	−0.760509	−4.928406	−0.730004	−0.704515	−0.706719
RU	Acc	0.788410	0.503850	0.874664	0.675844	0.628806	0.888193
	AP	0.887736	0.505097	0.880470	0.716313	0.632248	0.936153
	Recall	0.741223	0.209326	0.827654	0.661326	0.624888	0.873200
	AUC	0.878049	0.506039	0.904803	0.724353	0.660303	0.935274
	F1	0.774601	0.151750	0.864378	0.668624	0.625587	0.884040
	Log loss	−0.439150	−0.692630	−2.848513	−1.254115	−0.659066	−0.344099
RP	Acc	0.632344	0.589305	0.616375	0.578799	0.573946	0.638593
	AP	0.648665	0.606412	0.606449	0.586913	0.603876	0.659591
	Recall	0.598943	0.602359	0.618165	0.567060	0.597289	0.594442
	AUC	0.674403	0.629384	0.645574	0.606889	0.610451	0.679281
	F1	0.615721	0.589222	0.615671	0.572548	0.582932	0.618436
	Log loss	−0.657602	−0.670727	−4.674726	−0.839443	−0.680725	−0.649592

Table A3. Cont.

Feature	Metric	XGBoost	MLP Classifier	KNN	Random Forest	Logistic Regression	Stacking
RR	Acc	0.674448	0.652935	0.624115	0.650495	0.594659	0.663278
	AP	0.756893	0.733484	0.625443	0.708923	0.650911	0.750643
	Recall	0.627762	0.577254	0.615969	0.609733	0.432073	0.587583
	AUC	0.745782	0.725941	0.660809	0.706902	0.637155	0.735540
	F1	0.654273	0.609037	0.618741	0.632197	0.508534	0.628815
	Log loss	−0.593553	−0.612419	−5.006094	−0.842396	−0.664129	−0.598193
RUP	Acc	0.807252	0.594591	0.811793	0.715575	0.652338	0.885916
	AP	0.898493	0.593898	0.825490	0.785778	0.677732	0.936537
	Recall	0.740220	0.245981	0.798428	0.689901	0.583787	0.866260
	AUC	0.890683	0.597386	0.869931	0.778646	0.687341	0.938387
	F1	0.788954	0.364508	0.802904	0.704177	0.622428	0.879668
	Log loss	−0.419379	−0.663754	−3.037858	−0.612832	−0.646393	−0.358529
RUR	Acc	0.783733	0.631869	0.862871	0.699756	0.630514	0.879314
	AP	0.883710	0.634917	0.874515	0.750584	0.682035	0.933317
	Recall	0.739000	0.423993	0.818273	0.686648	0.685292	0.876318
	AUC	0.873931	0.642400	0.901048	0.754322	0.681900	0.932056
	F1	0.770288	0.524623	0.852011	0.692507	0.647381	0.876698
	Log loss	−0.445633	−0.643519	−2.827084	−1.082337	−0.645318	−0.352196
RPR	Acc	0.722326	0.635638	0.638308	0.684791	0.612851	0.718354
	AP	0.822230	0.668318	0.638486	0.757739	0.642765	0.815394
	Recall	0.638471	0.611495	0.647119	0.673987	0.438959	0.603497
	AUC	0.800322	0.687140	0.678240	0.744549	0.627778	0.790138
	F1	0.691485	0.618353	0.639377	0.678381	0.520305	0.675061
	Log loss	−0.53561	−0.648846	−4.320839	−0.696305	−0.668128	−0.543369
RA	Acc	0.804189	0.626366	0.809096	0.741521	0.654019	0.869093
	AP	0.898631	0.634693	0.824245	0.828305	0.692656	0.932614
	Recall	0.733550	0.356812	0.792626	0.711699	0.587502	0.856988
	AUC	0.889893	0.634662	0.867077	0.812656	0.692537	0.931487
	F1	0.784698	0.485224	0.799229	0.729379	0.625015	0.862228
	Log loss	−0.420916	−0.652366	−3.108753	−0.573428	−0.639423	−0.373861

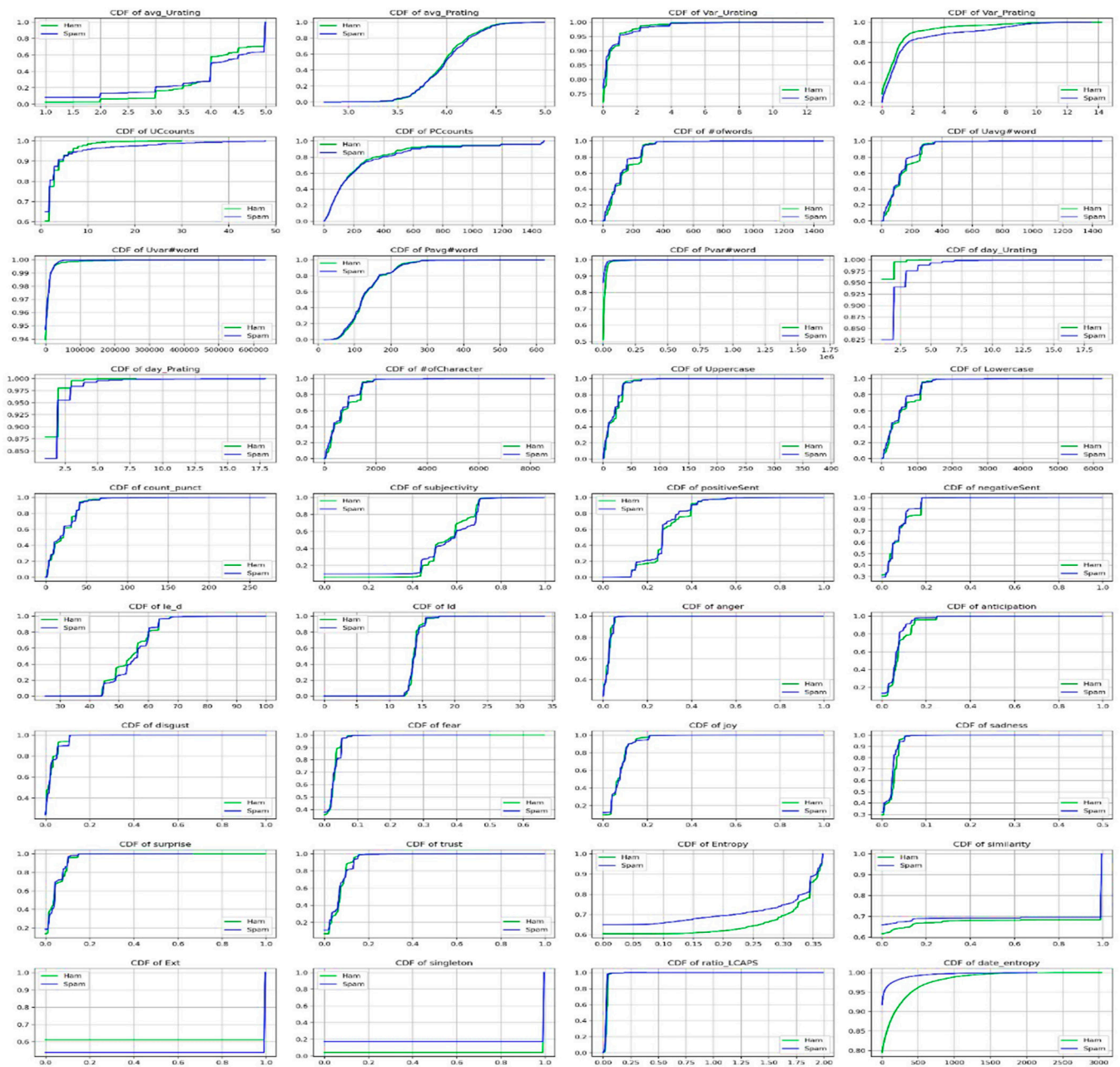


Figure A4. Cumulative distribution function of YelpNYC dataset features.

References

- Kim, S.; Kandampully, J.; Bilgihan, A. The Influence of EWOM Communications: An Application of Online Social Network Framework. *Comput. Human Behav.* **2018**, *80*, 243–254. [CrossRef]
- Rudolph, S. The Impact of Online Reviews on Customers' Buying Decisions [Infographic]. Available online: <http://www.business2community.com/infographics/impact-online-reviews-customers-buying-decisions-infographic-01280945#oaFtOjCMhi5CD7de.97> (accessed on 27 March 2020).
- Mukherjee, A.; Venkataraman, V. Opinion Spam Detection: An Unsupervised Approach Using Generative Models. 2014. Available online: https://www2.cs.uh.edu/~arjun/tr/UH_TR_2014_07.pdf (accessed on 20 June 2020).
- He, S.; Hollenbeck, B.; Proserpio, D. The Market for Fake Reviews. *Mark. Sci.* **2022**, *41*, 896–921. [CrossRef]
- Christopher, S.L.; Rahulnath, H.A. Review Authenticity Verification Using Supervised Learning and Reviewer Personality Traits. In Proceedings of the 2016 International Conference on Emerging Technological Trends (ICETT), Kollam, India, 21–22 October 2016. [CrossRef]
- Phil Trip Advisor Changes Its Slogan | TripAdvisorWatch: Hotel Reviews in Focus. Available online: <https://tripadvisorwatch.wordpress.com/2010/01/19/trip-advisor-changes-its-slogan/> (accessed on 6 December 2021).

7. Witts, S. TripAdvisor Blocked More than One Million Fake Reviews in 2022—The Caterer. Available online: <https://www.thecaterer.com/news/tripadvisor-block-fake-reviews-2022-hospitality> (accessed on 16 May 2023).
8. Butler, O. I Made My Shed the Top-Rated Restaurant on TripAdvisor. Available online: <https://www.vice.com/en/article/434gqw/i-made-my-shed-the-top-rated-restaurant-on-tripadvisor> (accessed on 12 December 2021).
9. Marciano, J. Fake Online Reviews Cost \$152 Billion a Year. Here's How e-Commerce Sites Can Stop Them | World Economic Forum. Available online: <https://www.weforum.org/agenda/2021/08/fake-online-reviews-are-a-152-billion-problem-heres-how-to-silence-them/> (accessed on 24 March 2023).
10. Govindankutty, S.; Gopalan, S.P. From Fake Reviews to Fake News: A Novel Pandemic Model of Misinformation in Digital Networks. *J. Theor. Appl. Electron. Commer. Res.* **2023**, *18*, 1069–1085. [CrossRef]
11. Online Product and Service Reviews | ACCC. Available online: <https://www.accc.gov.au/business/advertising-and-promotions/online-product-and-service-reviews> (accessed on 24 March 2023).
12. Press Information Bureau (PIB). Available online: <https://pib.gov.in/PressReleasePage.aspx?PRID=1877733> (accessed on 24 March 2023).
13. EUR-Lex-32019L2161-EN-EUR-Lex. Available online: <https://eur-lex.europa.eu/eli/dir/2019/2161/oj> (accessed on 24 March 2023).
14. Crawford, M.; Khoshgoftaar, T.M.; Prusa, J.D.; Richter, A.N.; Al Najada, H. Survey of Review Spam Detection Using Machine Learning Techniques. *J. Big Data* **2015**, *2*, 23. [CrossRef]
15. Vidanagama, D.U.; Silva, T.P.; Karunananda, A.S. Deceptive Consumer Review Detection: A Survey. *Artif. Intell. Rev.* **2020**, *53*, 1323–1352. [CrossRef]
16. Mayzlin, D.; Dover, Y.; Chevalier, J. Promotional Reviews: An Empirical Investigation of Online Review Manipulation. *Am. Econ. Rev.* **2014**, *104*, 2421–2455. [CrossRef]
17. Moon, S.; Kim, M.Y.; Bergey, P.K. Estimating Deception in Consumer Reviews Based on Extreme Terms: Comparison Analysis of Open vs. Closed Hotel Reservation Platforms. *J. Bus. Res.* **2019**, *102*, 83–96. [CrossRef]
18. Barbado, R.; Araque, O.; Iglesias, C.A. A Framework for Fake Review Detection in Online Consumer Electronics Retailers. *Inf. Process. Manag.* **2019**, *56*, 1234–1244. [CrossRef]
19. Jindal, N.; Liu, B. Review Spam Detection. In Proceedings of the 16th International World Wide Web Conference, WWW2007, Banff, AB, Canada, 8–12 May 2007; ACM Press: New York, NY, USA, 2007; pp. 1189–1190.
20. Ziora, L. Machine Learning Solutions in the Management of a Contemporary Business Organisation. *J. Decis. Syst.* **2020**, *29*, 344–351. [CrossRef]
21. Fontanarava, J.; Pasi, G.; Viviani, M. Feature Analysis for Fake Review Detection through Supervised Classification. In Proceedings of the 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Tokyo, Japan, 19–21 October 2017; pp. 658–666. [CrossRef]
22. Kumar, N.; Venugopal, D.; Qiu, L.; Kumar, S. Detecting Review Manipulation on Online Platforms with Hierarchical Supervised Learning. *J. Manag. Inf. Syst.* **2018**, *35*, 350–380. [CrossRef]
23. Wolpert, D.H. Stacked Generalization. *Neural Netw.* **1992**, *5*, 241–259. [CrossRef]
24. Van der Laan, M.J.; Polley, E.C.; Hubbard, A.E. Super Learner. *Stat. Appl. Genet. Mol. Biol.* **2007**, *6*, 25. [CrossRef] [PubMed]
25. Patel, N.A.; Patel, R. A Survey on Fake Review Detection Using Machine Learning Techniques. In Proceedings of the 2018 4th International Conference on Computing Communication and Automation (ICCCA), Greater Noida, India, 14–15 December 2018; pp. 1–6. [CrossRef]
26. Rayana, S.; Akoglu, L. Collective Opinion Spam Detection: Bridging Review Networks and Metadata. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015; pp. 985–994. [CrossRef]
27. Malbon, J. Taking Fake Online Consumer Reviews Seriously. *J. Consum. Policy* **2013**, *36*, 139–157. [CrossRef]
28. Zinko, R.; Patrick, A.; Furner, C.P.; Gaines, S.; Kim, M.D.; Negri, M.; Orellana, E.; Torres, S.; Villarreal, C. Responding to Negative Electronic Word of Mouth to Improve Purchase Intention. *J. Theor. Appl. Electron. Commer. Res.* **2021**, *16*, 109. [CrossRef]
29. Luca, M.; Zervas, G. Fake It till You Make It: Reputation, Competition, and Yelp Review Fraud. *Manage. Sci.* **2016**, *62*, 3412–3427. [CrossRef]
30. Lappas, T.; Sabnis, G.; Valkanas, G. The Impact of Fake Reviews on Online Visibility: A Vulnerability Assessment of the Hotel Industry. *Inf. Syst. Res.* **2016**, *27*, 940–961. [CrossRef]
31. Ismagilova, E.; Slade, E.; Rana, N.P.; Dwivedi, Y.K. The Effect of Characteristics of Source Credibility on Consumer Behaviour: A Meta-Analysis. *J. Retail. Consum. Serv.* **2020**, *53*, 101736. [CrossRef]
32. Hunt, K.M. Gaming the System: Fake Online Reviews v. Consumer Law. *Comput. Law Secur. Rev.* **2015**, *31*, 3–25. [CrossRef]
33. Lau, R.Y.K.; Liao, S.Y.; Chi-Wai Kwok, R.; Xu, K.; Xia, Y.; Li, Y. Text Mining and Probabilistic Language Modeling for Online Review Spam Detection. *ACM Trans. Manag. Inf. Syst.* **2011**, *2*, 1–30. [CrossRef]
34. Yelp Yelp Trust & Safety Report. 2021. Available online: <https://trust.yelp.com/wp-content/uploads/2022/02/Yelp-Trust-and-Safety-Report-2021.pdf> (accessed on 3 January 2023).
35. Zhang, D.; Zhou, L.; Kehoe, J.L.; Kilic, I.Y. What Online Reviewer Behaviors Really Matter? Effects of Verbal and Nonverbal Behaviors on Detection of Fake Online Reviews. *J. Manag. Inf. Syst.* **2016**, *33*, 456–481. [CrossRef]

36. Yoo, K.-H.; Gretzel, U. Comparison of Deceptive and Truthful Travel Reviews. In *Information and Communication Technologies in Tourism 2009*; Springer: Vienna, Austria, 2009; pp. 37–47.
37. Lai, C.L.; Xu, K.Q.; Lau, R.Y.K.; Li, Y.; Song, D. High-Order Concept Associations Mining and Inferential Language Modeling for Online Review Spam Detection. In *Proceedings of the 2010 IEEE International Conference on Data Mining Workshops*, Sydney, NSW, Australia, 13–13 December 2010; IEEE: Piscataway, NJ, USA; pp. 1120–1127.
38. Jindal, N.; Liu, B.; Lim, E.-P. Finding Unusual Review Patterns Using Unexpected Rules. In *Proceedings of the Proceedings of the 19th ACM international conference on Information and knowledge management-CIKM '10*, Toronto ON Canada, 26–30 October 2010; ACM Press: New York, NY, USA, 2010; p. 1549.
39. Ott, M.; Choi, Y.; Cardie, C.; Hancock, J.T. Finding Deceptive Opinion Spam by Any Stretch of the Imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, OR, USA, 19–24 June 2011; Volume 1, pp. 309–319.
40. Mukherjee, A.; Liu, B.; Glance, N. Spotting Fake Reviewer Groups in Consumer Reviews. In *Proceedings of the WWW '12—21st Annual Conference on World Wide Web Companion*, Lyon, France, 16–20 April 2012; pp. 191–200. [\[CrossRef\]](#)
41. Feng, S.; Banerjee, R.; Choi, Y. Syntactic Stylometry for Deception Detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, Jeju Island, Republic of Korea, 8–14 July 2012; pp. 171–175.
42. Mukherjee, A.; Kumar, A.; Liu, B.; Wang, J.; Hsu, M.; Castellanos, M.; Ghosh, R. Spotting Opinion Spammers Using Behavioral Footprints. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Chicago, IL, USA, 11–14 August 2013; Part F1288. pp. 632–640. [\[CrossRef\]](#)
43. Lu, Y.; Zhang, L.; Xiao, Y.; Li, Y. Simultaneously Detecting Fake Reviews and Review Spammers Using Factor Graph Model. In *Proceedings of the 5th Annual ACM Web Science Conference, WebSci '13*, Paris, France, 2–4 May 2013; pp. 225–233. [\[CrossRef\]](#)
44. Anderson, E.T.; Simester, D.I. Reviews without a Purchase: Low Ratings, Loyal Customers, and Deception. *J. Mark. Res.* **2014**, *51*, 249–269. [\[CrossRef\]](#)
45. Banerjee, S.; Chua, A.Y.K. A Linguistic Framework to Distinguish between Genuine and Deceptive Online Reviews. *Lect. Notes Eng. Comput. Sci.* **2014**, *2209*, 501–506.
46. Banerjee, S.; Chua, A.Y.K.; Kim, J.J. Using Supervised Learning to Classify Authentic and Fake Online Reviews. In *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication, IMCOM '15*, Bali, Indonesia, 8–10 January 2015. [\[CrossRef\]](#)
47. Li, Y.; Feng, X.; Zhang, S. Detecting Fake Reviews Utilizing Semantic and Emotion Model. In *Proceedings of the 2016 3rd International Conference on Information Science and Control Engineering, ICISCE 2016*, Beijing, China, 8–10 July 2016; pp. 317–320. [\[CrossRef\]](#)
48. Sun, C.; Du, Q.; Tian, G. Exploiting Product Related Review Features for Fake Review Detection. *Math. Probl. Eng.* **2016**, *2016*, 4935792. [\[CrossRef\]](#)
49. Shehnepoor, S.; Salehi, M.; Farahbakhsh, R.; Crespi, N. NetSpam: A Network-Based Spam Detection Framework for Reviews in Online Social Media. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 1585–1595. [\[CrossRef\]](#)
50. Ren, Y.; Ji, D. Neural Networks for Deceptive Opinion Spam Detection: An Empirical Study. *Inf. Sci.* **2017**, *385–386*, 213–224. [\[CrossRef\]](#)
51. Zhuang, M.; Cui, G.; Peng, L. Manufactured Opinions: The Effect of Manipulating Online Product Reviews. *J. Bus. Res.* **2018**, *87*, 24–35. [\[CrossRef\]](#)
52. Nakayama, M.; Wan, Y. Exploratory Study on Anchoring: Fake Vote Counts in Consumer Reviews Affect Judgments of Information Quality. *J. Theor. Appl. Electron. Commer. Res.* **2017**, *12*, 1–20. [\[CrossRef\]](#)
53. Jain, N.; Kumar, A.; Singh, S.; Singh, C.; Tripathi, S. Deceptive Reviews Detection Using Deep Learning Techniques. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; 11608 LNCS; Springer: Heidelberg, Germany, 2019; pp. 79–91. [\[CrossRef\]](#)
54. Plotkina, D.; Munzel, A.; Pallud, J. Illusions of Truth—Experimental Insights into Human and Algorithmic Detections of Fake Online Reviews. *J. Bus. Res.* **2020**, *109*, 511–523. [\[CrossRef\]](#)
55. Hajek, P.; Barushka, A.; Munk, M. Fake Consumer Review Detection Using Deep Neural Networks Integrating Word Embeddings and Emotion Mining. *Neural Comput. Appl.* **2020**, *32*, 17259–17274. [\[CrossRef\]](#)
56. Li, L.; Lee, K.Y.; Lee, M.; Yang, S.B. Unveiling the Cloak of Deviance: Linguistic Cues for Psychological Processes in Fake Online Reviews. *Int. J. Hosp. Manag.* **2020**, *87*, 102468. [\[CrossRef\]](#)
57. Mohawesh, R.; Tran, S.; Ollington, R.; Xu, S. Analysis of Concept Drift in Fake Reviews Detection. *Expert Syst. Appl.* **2021**, *169*, 114318. [\[CrossRef\]](#)
58. Shan, G.; Zhou, L.; Zhang, D. From Conflicts and Confusion to Doubts: Examining Review Inconsistency for Fake Review Detection. *Decis. Support Syst.* **2021**, *144*, 113513. [\[CrossRef\]](#)
59. Wang, E.Y.; Fong, L.H.N.; Law, R. Detecting Fake Hospitality Reviews through the Interplay of Emotional Cues, Cognitive Cues and Review Valence. *Int. J. Contemp. Hosp. Manag.* **2022**, *34*, 184–200.
60. Hajek, P.; Sahut, J.M. Mining Behavioural and Sentiment-Dependent Linguistic Patterns from Restaurant Reviews for Fake Review Detection. *Technol. Forecast. Soc. Chang.* **2022**, *177*, 121532. [\[CrossRef\]](#)

61. Kumar, A.; Gopal, R.D.; Shankar, R.; Tan, K.H. Fraudulent Review Detection Model Focusing on Emotional Expressions and Explicit Aspects: Investigating the Potential of Feature Engineering. *Decis. Support Syst.* **2022**, *155*, 113728. [CrossRef]
62. Carlens, H. State of Competitive Machine Learning in 2022. 2023. Available online: <https://mlcontests.com/state-of-competitive-machine-learning-2022/> (accessed on 5 January 2023).
63. Weise Karen A Lie Detector Test for Online Reviewers-Bloomberg. Available online: <https://www.bloomberg.com/news/articles/2011-09-29/a-lie-detector-test-for-online-reviewers?leadSource=uverifywall#xj4y7vzkg> (accessed on 7 November 2023).
64. Li, J.; Ott, M.; Cardie, C.; Hovy, E. Towards a General Rule for Identifying Deceptive Opinion Spam. In Proceedings of the Detecting Deceptive Reviews Using Generative Adversarial Networks, Baltimore, MD, USA, 23–25 June 2014; Volume 1, pp. 1566–1576. [CrossRef]
65. Aghakhani, H.; Machiry, A.; Nilizadeh, S.; Kruegel, C.; Vigna, G. Detecting Deceptive Reviews Using Generative Adversarial Networks. In Proceedings of the 2018 IEEE Symposium on Security and Privacy Workshops, San Francisco, CA, USA, 24 May 2018; pp. 89–95. [CrossRef]
66. Yuan, C.; Zhou, W.; Ma, Q.; Lv, S.; Han, J.; Hu, S. Learning Review Representations from User and Product Level Information for Spam Detection. In Proceedings of the 2019 IEEE International Conference on Data Mining (ICDM), Beijing, China, 8–11 November 2019; pp. 1444–1449.
67. Ott, M.; Cardie, C.; Hancock, J. Estimating the Prevalence of Deception in Online Review Communities. In Proceedings of the 21st International Conference on World Wide Web, WWW '12, Lyon France, 16–20 April 2012; pp. 201–210. [CrossRef]
68. Lemaître, G.; Nogueira, F.; Aridas, C.K. Imbalanced-Learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *J. Mach. Learn. Res.* **2017**, *18*, 1–5.
69. Desikan, B.S. *Natural Language Processing and Computational Linguistics: A Practical Guide to Text Analysis with Python, Gensim, SpaCy, and Keras*; Packt Publishing Ltd.: Birmingham, UK, 2018; ISBN 9781788837033.
70. Jindal, N.; Liu, B. Opinion Spam and Analysis. In Proceedings of the International Conference on Web Search and Web Data Mining—WSDM '08, Palo Alto, CA, USA, 11–12 February 2008; ACM Press: New York, NY, USA, 2008; p. 219.
71. Li, F.; Huang, M.; Yang, Y.; Zhu, X. Learning to Identify Review Spam. In Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Spain, 16–22 July 2011; pp. 2488–2493. [CrossRef]
72. Hutto, C.J.; Gilbert, E. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. In Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, MI, USA, 1–4 June 2014.
73. McCarthy, P.M. An Assessment of the Range and Usefulness of Lexical Diversity Measures and the Potential of the Measure of Textual, Lexical Diversity (MTLD). Ph.D. Thesis, The University of Memphis, Memphis, TN, USA, 2005.
74. Dewang, R.K.; Singh, A.K. Identification of Fake Reviews Using New Set of Lexical and Syntactic Features. In Proceedings of the Sixth International Conference on Computer and Communication Technology 2015—ICCCT '15, Allahabad India, 25–27 September 2015; pp. 115–119. [CrossRef]
75. Mohammad, S.M.; Turney, P.D. Crowdsourcing a Word-Emotion Association Lexicon. *Comput. Intell.* **2013**, *29*, 436–465. [CrossRef]
76. Li, J.; Cardie, C.; Li, S. TopicSpam: A Topic-Model-Based Approach for Spam Detection. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 4–9 August 2013; Volume 2, pp. 217–221.
77. Džeroski, S.; Ženko, B. Is Combining Classifiers with Stacking Better than Selecting the Best One? *Mach. Learn.* **2004**, *54*, 255–273. [CrossRef]
78. Ribeiro, M.T.; Singh, S.; Guestrin, C. “Why Should I Trust You?”: Explaining the Predictions of Any Classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 1135–1144.
79. Shrikumar, A.; Greenside, P.; Kundaje, A. Learning Important Features through Propagating Activation Differences. In Proceedings of the 34th International Conference on Machine Learning—Volume 70, Sydney, Australia, 6–11 August 2017; pp. 3145–3153.
80. Lundberg, S.M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 4768–4777.
81. Yilmaz, C.M.; Durahim, A.O. SPR2EP: A Semi-Supervised Spam Review Detection Framework. In Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Barcelona, Spain, 28–31 August 2018; IEEE: Piscataway, NJ, USA; pp. 306–313.
82. Mohawesh, R.; Xu, S.; Springer, M.; Jararweh, Y.; Al-Hawawreh, M.; Maqsood, S. An Explainable Ensemble of Multi-View Deep Learning Model for Fake Review Detection. *J. King Saud Univ.-Comput. Inf. Sci.* **2023**, *35*, 101644. [CrossRef]
83. Kennedy, S.; Walsh, N.; Sloka, K.; McCarren, A.; Foster, J. Fact or Factitious? Contextualized Opinion Spam Detection. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, Florence, Italy, 28 July–2 August 2019; Alva-Manchego, F., Choi, E., Khashabi, D., Eds.; Association for Computational Linguistics: Florence, Italy, 2019; pp. 344–350.
84. Farrelly, C.M. Deep vs. Diverse Architectures for Classification Problems. *arXiv* **2017**, arXiv:1708.06347.

85. Lundberg, S. Interpretable Machine Learning with XGBoost | by Scott Lundberg | Towards Data Science. Available online: <https://towardsdatascience.com/interpretable-machine-learning-with-xgboost-9ec80d148d27> (accessed on 12 December 2021).
86. Chowdhury, T.; Oredo, J. AI Ethical Biases: Normative and Information Systems Development Conceptual Framework. *J. Decis. Syst.* **2022**, *32*, 617–633. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.