Fake Reviews Filter based on Yelp Dataset

Cai Ying, Hua Xiuping, Huang Zhao, Shi Haohui, Yu Yue

Abstract

Customers' reviews are very important indicators of the quality and popularity of business. In many cases, consumers tend to refer to reviews and ratings when choosing a business. But in real life, many reviews are not genuine. Some business owners may use unethical ways to influence their online reputation by writing fake reviews of their businesses or competitors. In our project, we choose to use Yelp reviews as the object of the study, aimed at simulating the algorithm of Yelp to detect fake reviews and find a model that could well predict the authenticity of reviews as per Yelp's definition. We worked with 7 different models - Naive Bayes, Logistic Regression, KNN, SVM, Random Forests, XGBoost and LightGBM, of which LightGBM produced the most consistent results overall.

1. Introduction

Yelp is one of the most widely used platforms which allows users to find good local businesses. Consumers can write reviews and give a star rating based on product or service quality. Both the star rating and text reviews are very important indicators of the quality and popularity of each business. In many cases, consumers tend to refer to reviews and ratings when choosing a business, not just business owners' descriptions. But in real life, not all the reviews are genuine. Some business owners use unethical ways to influence their online reputation by writing fake reviews of their own businesses or competitors', thus manipulating consumers' decision making. This problem may mislead consumers to go to a business that is not good or miss a business that is actually worth-visiting. This inconsistency between reviews and the actual experiences will reduce consumers' trust in the platform.

In this project, we attempt to apply various machine learning models that are appropriate for textual and numeric data to detect fake reviews. Since Yelp has a relatively complete dataset and is also a widely used platform, we choose Yelp reviews as the object of the study. We aim to simulate the algorithm of Yelp to detect fake reviews and find a model that could well predict the authenticity of reviews as per Yelp's definition.

Applications of fake review detection are diverse. For business owners, they always want to know consumer's genuine feedbacks to improve their service and products and increase revenue. For prospective patrons, they also want to know the real opinions from customers who have been to this store before. For developers of similar platforms, our research method can also provide some references for them to detect fake reviews.

2. Literature Review

Research on spam detection has been explored continually. In the past few years, machine learning focusing on spam detection has been widely studied and explored to combat spam in media like email and web pages (Cardoso, Silva & Almeida, 2018). Many traditional machine learning methods have been applied, such as Support Vector Machine, Naïve Bayes (Almeida, Silva & Yamakami, 2013), Decision Trees and K-nearest neighbors (Alberto, Lochter & Almeida, 2015). As a whole, spam detection was based on textual content and was a binary text categorization problem where the categories were spam or non-spam (Chan et al, 2015).

Fake Review Filter is a specific kind of spam detection. While spam on email and webpages can be easily identified by users, detection of fake reviews imposes more challenges as even an experienced user might fail to detect them.

As a result, fake reviews detection has recently been studied extensively and existing works have made important progress. Harris (2012) and Ott, Choi, Cardie & Hancock (2011) used content-based filtering methods. Some researchers studied typical behaviors of reviewers. For instance, Li et al (2015) used temporal and spatial features and Mukherjee et al (2013) exploited various observed behavioral footprints of reviewers to create a distributional divergence between spammers and non-spammers. Other studies focused on detecting groups of spammers to filter fake reviews (Mukherjee et al, 2012).
Since many fake reviews are carefully written to look authentic and spammers use multiple IDs to avoid detection, more efforts are needed to invest in to build a generic model to filter fake reviews.

3. Dataset Description

In our project, we used the consumer review data of restaurants from different cities in the United States. The data source is shared by the author of "*Collective Opinion Spam Detection: Bridging Review Networks and Metadata*" (Rayana & Akoglu, 2015).

We verified the distribution of fake and genuine

reviews in the dataset and noticed that close to 86.8% of reviews in the dataset are genuine and only 13.2% are fake reviews, pointing to a huge imbalance in the dataset which has to be dealt at the later stages of our analysis. The column description of the dataset is listed in *Table 2*.

To better understand the data distribution, we tried to obtain insights through data visualization. *Figure 2* shows the review distribution in each day of a week, we observed that genuine users tend to give more reviews on Sundays and Mondays, while fake reviews are more equally distributed in one week. From *Figure 3* below, it can be concluded that ratings of fake reviews are extremer. Fake users tend to give ranting 1 or 5 while genuine users usually give rating 4 or 5 and rarely give 1. Through the preliminary exploration of data, we found that there are behavioral gaps between fake reviewers and genuine reviewers, which should be taken into consideration in the later phase.

Table 1:Overview of the Dataset.

	Total Reviews	Fake Reviews	Genuine Reviews	Unique Restaurants	Unique Users
YelpZip	608,598	80,466	528,132	5,044	260,277

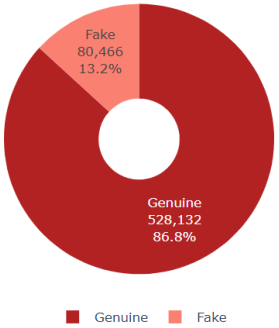

Figure 1: Review Distribution

Table 2: Column Description

Column	Data Type	Description
user_id	Integer	Yelp user ID.
prod_id	Integer	Yelp product ID, which represents a certain restaurant.
prod_name	String	Yelp product name, which represents the name of the restaurant.
date	String	The date when the user posted the review (format: YYYY-MM-DD).
rating	Integer	The star rating ranging from 1-5 given by a user to a certain restaurant.
review_text	String	The text content of the review
label	Boolean	0 or 1, "0" means "Genuine review" while "1" means "Fake review". The label is generated from Yelp`s filtering algorithm.

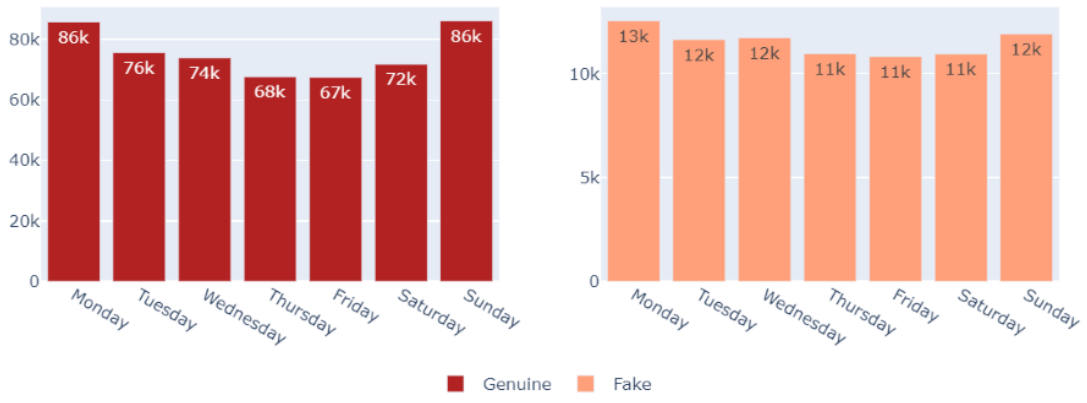

Figure 2: Number of Reviews vs. Day of Week

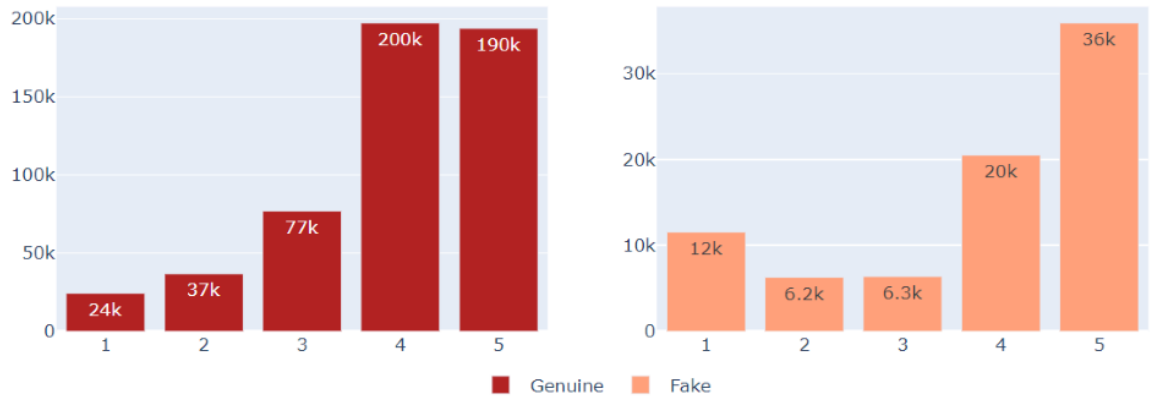

Figure 3: Rating Distribution

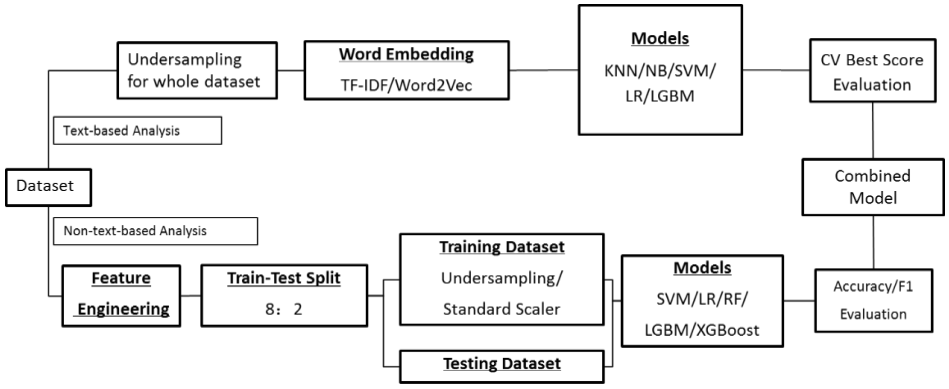

Figure 4: Workflow

4. Workflow

Due to the disparate nature of the quantitative features engineered on users, etc. and the text embedding features, the decision was made to segregate the two feature types initially and select separate models for them. Just as feature selection is a contributory step in conventional statistical modelling, it stands to reason that providing the full complement of available features at once to the same machine learning model does not necessarily optimize predictive performance, especially considering that there is no one algorithm that best handles all scenarios (Wolpert & Macready, 1996). Based on the findings from these preliminary tracks of investigation, a combined model would then be fitted, the performance of which would provide further insight into the workings of Yelp's model.

5. Text based Classification

5.1 Data Pre-processing for Modeling

In this section, since our dataset is unbalanced, we do under sampling for the whole dataset to make sure that our models are reliable and unbiased. The sample size is based on the number of fake reviews.

5.2 Feature Engineering

5.2.1 TF-IDF

TF-IDF is the basic tokenization, based on character and word frequency. The input data is feed into the TF-IDF vectorizer without preprocessing such as stop-words removal and punctuation removal.

Different level of grams as well as on both word and character levels are experimented for feature vector generation. On the word level, uni-gram level feature takes the occurrence of words in sentences into account, while the bi-gram, tri-gram, and uni-bi-gram consider the order of sentences. On the character level, each character is considered an individual unit, and tokens are made up of character n-grams. Grams level ranging from 3 to 6 is implemented. It is noteworthy that the increase in the number of n-grams will increase the size of the vocabulary significantly. Therefore, the maximum vocabulary is set to 50000 to avoid such issues.

5.2.2 WORD2VEC

Word2Vec is one of the most widely adopted methods to represent text as vectors. In this study, both the Gensim model (CBOW) trained using the Yelp training dataset and the pretrained Google Word2Vec are implemented to perform fake reviews detection.

The former is trained to predict the word by first applying the Yelp dataset. The size of the dense vector is set to 150, which is generally considered enough for similarity lookups. The smaller the window the more related words are detected. The value of window and worker are specified as 5 and 4 respectively. For the latter, the input dataset for training is the Google News dataset containing about 100 billion words. The resulting word

vectors comprise 300 features and a vast vocabulary of 3 million texts. After training and loading the word embedding models, feature vectors are generated and the corresponding output is averaged to represent the texts.

5.3 Models

5.3.1 MODEL SELECTION AND TUNING

In this section, we apply five models for text classification, including SVM, Naïve Bayes, Logistic Regression, KNN and Light GBM. All the above mentioned feature vectors are applied on these five algorithms. Model parameters are tuned using GridSearchCV and F1 score is chosen as evaluation metric since the class distribution of the cross-validation datasets is uneven. *Figure 5* gives a glance of overall model performance from our experiments.

Table 3: Model Hyper-parameter Tuning and Performance

Model	Best vector	CV F1
SVM	tfidf_char_tri_gram	0.6507
Naïve Bayes	tfidf_char_five_gram	0.6714
Logistic Regression	tfidf_char_five_gram	0.6749
KNN	tfidf_char_tri_gram	0.6697
LightGBM	tfidf_word_tri_gram	0.6580

Table 3 above displays the combination of model and text-based features vectors that yield the best results. Noticeably, the best model is found to be the logistic regression on character five-grams. TF-IDF on character levels in general outperformed other feature vectors.

Surprisingly, features created using the Word2Vec models does not give good performances in terms of F1-score compared with others. For one thing, the capability of the self-trained model to represent the data depends largely on the input data and the parameter settings. However, parameter tuning for the Word2Vec is not taken into consideration due to the computational power and time constraints. For another, using the pre-trained Google model does not necessarily give a proper representation of the texts especially when both datasets are differently distributed.

6. Non-text-based Classification

6.1 Feature Engineering

In addition to text-based method discussed in the previous section, the non-text-based method is also explored. We construct 23 new features based on gaps between fake reviews and genuine reviews in description, writing habits, sentiment of the review, behavioral features of the reviewer, and features of the restaurant. This method allows us to incorporate more information in the original data instead of just focusing on the text. Non-text-based features can be divided to 3 groups and are illustrated in *Table 5* and *Table 6*.

After finishing feature engineering, we visualize user and restaurant-based features to further observe the gaps between users who have given fake reviews and those whose reviews are all genuine, and the gaps between restaurants which have received fake reviews and those whose reviews are all genuine.

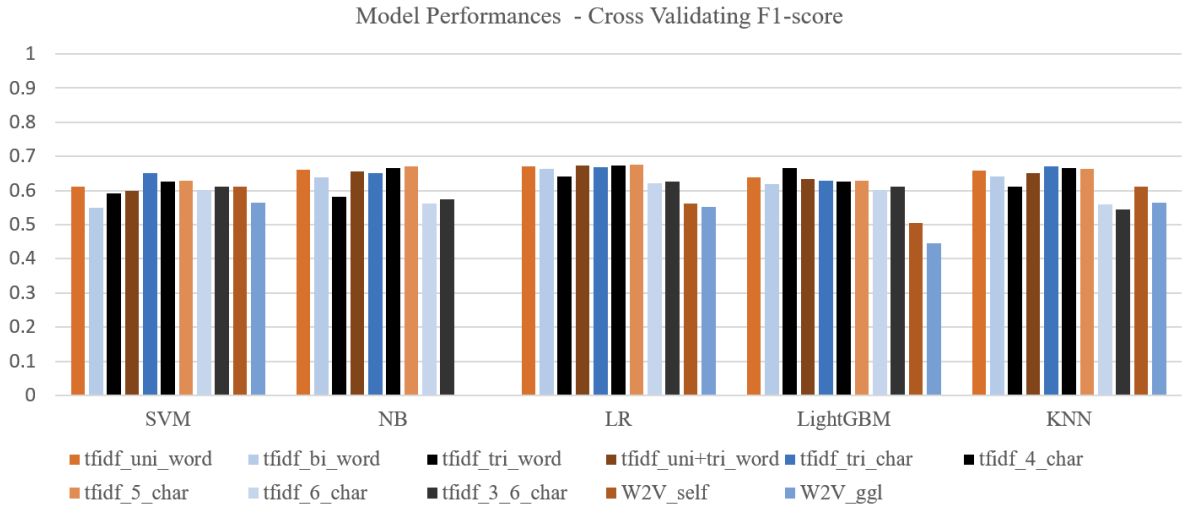

Figure 5: Model Performance Results

Table 5: Review Based Features

Feature	Description
day_of_week	Day of the week when the review was posted.
rating	The star rating ranging from 1-5 given by a user to a certain restaurant.
char_count	Number of characters in the review.
word_count	Number of words in the review.
word_density	Word density of the review: number of chars/(number of words + 1).
exclamation_percent	Percentage of exclamation in the review.
punctuation_percent	Percentage of punctuation in the review.
title_word_percent	Percentage of title words in the review.
upper_case_word_percent	Percentage of uppercase words in the review.
first_person_percent	Percentage of first person (I or i) in the review.
text_polarity	Polarity score calculated using Textblob after removing stop words and punctuation. The score is a float within the range [-1.0, 1.0]. 1.0 is very positive. -1 is very negative. 0 is neutral.

Table 6: User/Restaurant Based Features

Feature	Description
user_no_of_review/ prod_no_of_review	Total number of reviews the user has given/ the restaurant has received.
user_avg_rating/ prod_avg_rating	Average rating the user has given/ the restaurant has received.
user_max_no_reviews/ prod_max_no_reviews	Max number of reviews per day the user has given/the restaurant has received.
user_rating_std/ prod_rating_std	Standard deviation of ratings given by the user/ received by the restaurant.
user_avg_no_words/ prod_avg_no_words	Average number of words in the reviews the user has given/ the restaurant has received.
user_has_dup_text/ prod_has_dup_text	Whether the user has given exact same review text twice/ the restaurant has received exact same review text twice. 1 stands for yes and 0 is no.

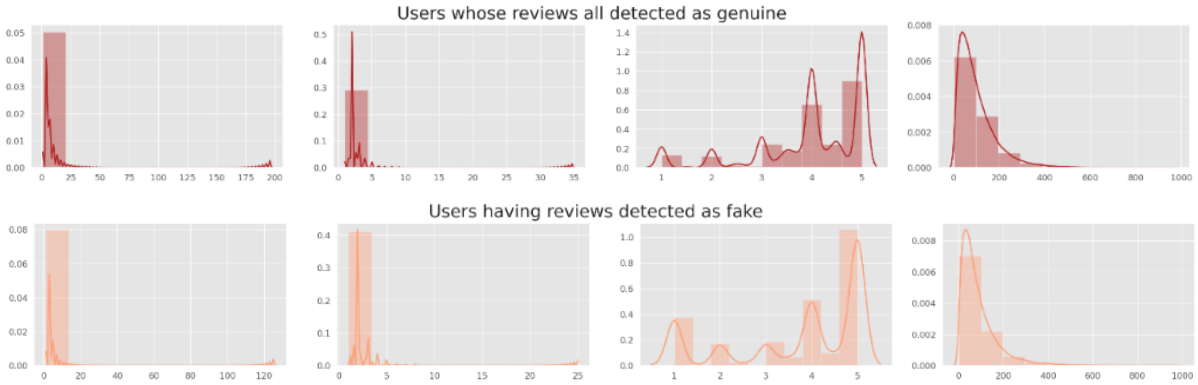

Figure 6: User-based Features Visualization

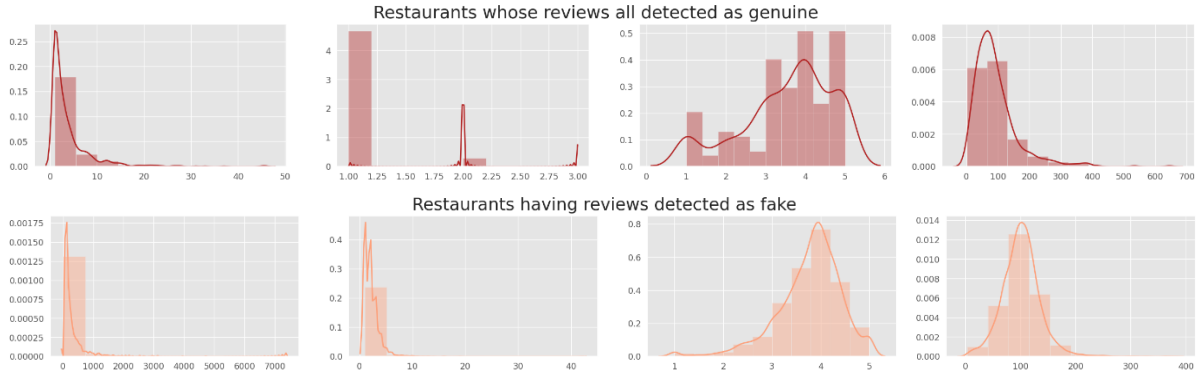

Figure 7: Restaurant-based Features Visualization

It can be found from *Figure 6* that fake users' total number of reviews is less than genuine users', but fake users tend to give more reviews per day and more extreme ratings such as 1 and 5 than genuine users.

Figure 6 illustrates that restaurants with larger number of reviews are more likely to receive fake reviews. On the one hand, these restaurants have longer history and accumulate more reviews, increasing the probability of receiving fake reviews from competitors. On the other hand, these restaurants may have more motivation to write fake reviews to improve their online reputation since they are familiar with the rules of Yelp platform. Meanwhile, new restaurants tend to have all reviews detected as genuine.

6.2 Data Pre-processing for Modeling

To prepare for modeling, we split the dataset into train and test sets in 8:2 ratio. In view of the unbalance of the dataset, under-sampling is implemented on the train set, then we get 64,256 genuine reviews and 64,256 fake reviews in train set. StandardScaler is applied to make all features have common scale.

6.3 Models

6.3.1 MODLE SELECTION AND TUNING

The process is similar to that for the text-based analysis, but without the choice of feature set as a search dimension. The candidates evaluated here are general high-performance algorithms whose ultimate test-set performance would be the sole focus. Once again five model classes are tested, with Logistic Regression, SVM and LightGBM being repeated from the text-based track for comparability and XGBoost and Random Forest being added to increase coverage of bagging and boosting techniques.

Table 7: Model Chosen and Hyper-parameter Tuning

Model	Parameter	Best
SVM	C	1
	gamma	scale
	kernel	rbf
Logistic Regression	C	10
Random Forest	n_estimators	96
	max_depth	7
	min_samples_leaf	3
LightGBM	boosting_type	gbdt
	learning_rate	0.1
	n_estimators	117
	num_leaves	31
XGBoost	n_estimators	986
	learning_rate	0.1
	max_depth	4

6.3.2 EVALUATION

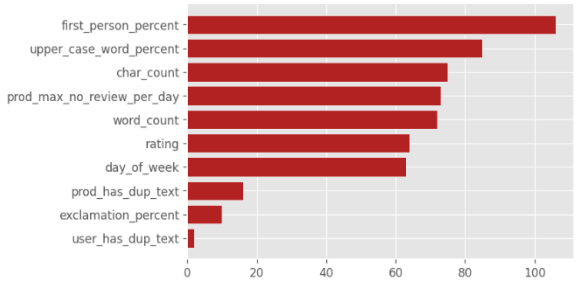

Figure 8: LightGBM Feature Importance

Table 8: Model and Evaluation Metrics

Model	Accuracy	F1	AUC
SVM	0.57	0.25	0.59
Logistic Regression	0.65	0.38	0.79
Random Forest	0.70	0.42	0.82
LightGBM	0.71	0.43	0.84
XGBoost	0.70	0.43	0.84

6.3.3 OUTPUT ANALYSIS

From the best-performed LightGBM, feature importances are extracted as a form of intuitiveness check and a source of model interpretability. The top 10 features comprised a mix of stylistic, behavioral and aggregate features (*Figure 8*). This suggests that a holistic approach considering both review-level characteristics and entity-level (user or restaurant) profiles is likely to have been in place at Yelp, or at least such features are able to proxy the signals that Yelp's model was looking for.

7. Combined Classification

A model incorporating both text-based and non-text-based features is created, its design informed by prior text-based and non-text-based analyses. All non-text-based features are included, and the text vectorization of choice is word unigrams based on its prior consistency in performance and computational efficiency. Similarly, LightGBM is chosen as the model base for its prior classification performance and compatibility with the sparse CSR matrix arising from the feature combination. The tuning conclusions are shown in *Table 9*.

Table 9: Combined LightGBM Parameterization

Parameter	Best
boosting_type	dart
learning_rate	0.01
num_leaves	80
feature_fraction	0.7
bagging_fraction	0.8
num_interations	4880

After the best configuration is found, the feature importances are extracted as before, and the top features are all found to be from the non-text-based side. This is to be expected, as the importance are calculated on the basis of splits and the text-based portion of the feature matrix is composed of a large number of columns each only applicable to a small subset of rows. Combined with the presence of feature and bagging fractions less than 1 in this model, it is highly unlikely that any particular text column would feature more prominently than the non-text columns which are dense to begin with.

Additionally, further downstream experimentation with the same model but tuned for ROC AUC and with the classification threshold being shifted around showes that positive-case F1 peaked at a threshold higher than 0.5 (*Table 10*), with the increase in precision more than compensating for the loss in recall. Naturally accuracy increases as well, but as the dataset is imbalanced, we do not consider that to be informative. That said, there is a business case for tuning up the threshold in this kind of classification, as it stands to reason that each positive label costs more to assign than a negative one, in terms of the work that needs to be done when accusing fakery.

Table 10: Combined LightGBM Results

Output Mode	Precision	Recall	F1	Accuracy
Classification	0.31	0.86	0.45	0.72
Probability + 0.65 Threshold	0.42	0.55	0.48	0.81

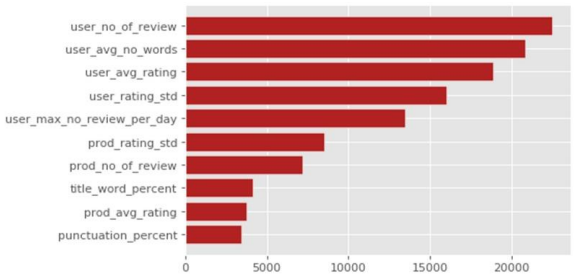

Figure 9 Combined Model: LightGBM Feature Importance

8. Conclusion & Insights

Overall, non-text based features outperform text-based features in fake review detection tasks, giving better accuracy. It should be noted that the nature of original datasets may have introduced bias in model prediction. Since the labels are assigned by the Yelp algorithm, the reviews may already have been misclassified in the first place. For example, we spot that many fake reviewers who give same reviews more than one time are not detected and assigned with the correct label accordingly.

Future work can be done to improve model performance by employing more advanced models such as Bert, CNN, and Bi-directional LSTM. Additionally, models should be accompanied by more thorough level of hyper-parameter tuning, which at the same time requires more computational power and time. Another affordance of increased computational power is the ability to calculate cosine similarities within user and restaurant aggregations. This had been an initial plan but is forgone as testing proved the computation prohibitive, instead being substituted by the similar but much less precise duplication search. Moreover, model overfitting which we observed in the large discrepancy between the training and testing accuracy can be avoided with the use of larger datasets, as well as more meaningful features such as user level characteristics and behavioral attributes.

References

[01] Cardoso E F, Silva R M, Almeida T A. *Towards automatic filtering of fake reviews*[J]. Neurocomputing, 2018, 309: 106-116.

[02] Almeida T, Silva R M, Yamakami A. *Machine learning methods for spamdexing detection*[J]. International Journal of Information. Security Science, 2(3): 86-107, 2013.

[03] Alberto T C, Lochter J V, Almeida T A. *Post or block? advances in automatically filtering undesired comments*[J]. Journal of Intelligent & Robotic Systems,80(1): 245-259, 2015.

[04] Chan P P K, Yang C, Yeung D S, et al. *Spam filtering for short messages in adversarial environment*[J]. Neurocomputing, 155: 167-176, 2015.

[05] Harris C G. *Detecting deceptive opinion spam using human computation*[C]//Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence. 2012.

[06] Ott M, Choi Y, Cardie C, et al. *Finding deceptive opinion spam by any stretch of the imagination*[C]//Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1. Association for Computational Linguistics, 309-319, 2011.

[07] Li H, Chen Z, Mukherjee A, et al. *Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns*[C]//ninth international AAAI conference on web and social Media. 2015.

[08] Mukherjee A, Kumar A, Liu B, et al. *Spotting opinion spammers using behavioral footprints*[C]//Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, 632-640, 2013.

[09] Mukherjee A, Liu B, Glance N. *Spotting fake reviewer groups in consumer reviews*[C]//Proceedings of the 21st international conference on World Wide Web, 191-200, 2012.

[10] Shebuti Rayana, and Leman Akoglu. *Collective opinion spam detection: Bridging review networks and metadata*. Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining. ACM, 2015.

[11] Wolpert, D. H., and W. G. Macready. *No Free Lunch Theorems for Search*. Santa Fe Institute repot. SFI-TR-95-02-010, 1996.