Online Fake Review Detection on Yelp Data

Yichen Wang
(CSCI 7000-003)
Student ID: 108588424
University of Colorado Boulder
yichen.wang@colorado.edu

Tao Ruan
(CSCI 7000-003)
Student ID: 108609439
University of Colorado Boulder
tao.ruan@colorado.edu

ABSTRACT

This report provides a summary of our project, which aims to implement an automatic fake review detection system on Yelp's dataset. We introduce the motivation of our project and list the related work to solve the problem. Then we describe the dataset we plan to use for the project and propose a workflow to solve our problem. Also we discuss the evaluation method for our problem. Our project finally achieved the AUC as high as 0.964 with ensemble methods. Besides, we extracted many self-designed features in feature engineering part and the feature ranking indicates that our self-designed features are among the most important ones to achieve the high AUC.

Keywords

user behavior, feature engineering, classification, fake reviews

1. INTRODUCTION

With the increasing development of online shopping sites and review sites, people's purchase behaviors are being more affected by the online product reviews. Statistics show that by the end of 2016, there have already been over 0.1 billion reviews on Yelp. Every year more than 18 million new reviews born on Yelp.[9] Researches show more than 80% of online users in US admitted that product reviews influence their purchase behaviors. However, not all the online reviews are reliable. Fake reviews are broadly spread and hard to catch. Among all the product reviews on Yelp, 14-20% of them are fake reviews.[12] These reviews use unrealistic words to either promote or defame certain products or service, which are also called deceptive review or untruthful opinions. Therefore, accurately detecting these fake reviews has become an important online security issue.

As a matter of fact, for ordinary people, detecting fake reviews is a hard task. According to the test by Ott et al., three human only got the average accuracy of 57.33% in a fake review detection.[13] So it is a significant and meaningful problem to use Neural Language Processing and other advanced machine learning technologies to help catch these reviews, due to the low accuracy obtained in human. To address this problem, our project aims to implement a system to automatically identify fake reviews in Yelp dataset. Our work will mainly focus on three parts: Data exploration, classification and evaluation. Obviously, our goal is to tell one comment is true or fake, which is a two-category classification problem.

Typical fake review detection has three layers: fake review text detection, fake reviewer detection and fake review and reviewer group detection. Fake reviews always have different syntactic features or styles from true reviews and similarly fake reviewers also have different profile features or behavior features. These features provide useful information for us when distinguishing them. We can see feature engineering is important for such tasks, so we aim to contribute by trying to extract more domain-specific features and applying machine learning methods to solving this. As for group detection, nowadays to have more effects on customers fraudulent reviewers always have organizations to release fake reviews. This type of fraudulent organizations has even worse outcomes and also requires even more behavior features and profile features to explore the intrinsic structure among the reviewers, which is an advanced topic in this area.

2. RELATED WORK

As we briefly mentioned in the first section, there are three layers in fake review's detection related work. This section will be focusing on the first two layers.

2.1 Fake review text detection

Neural language processing techniques obviously are a natural thought when analyzing review text. Through literature survey, three types of detection methods are always used here, which are syntax parsing, semantic analysis and stylometric analysis.

2.1.1 Syntax parsing

The syntax parsing based method is the most common way of extracting features from text, and one of the best-known is bag-of-words(BoW) features, or we can call as n-gram features. Bag-of-words take out the single word or multiple continuous words from the text and use them as features, which are quite useful in opinion mining, emotion analysis and other areas. Especially in fake review detection problem, bag-of-words features have significantly better performance than other text features. Even though merely using these features cannot obtain sufficiently good results, combining n-gram with other user behavior related features can have pretty good results.[10]

Besides n-gram, Part-of-Speech tagging (POS tagging) is another way to create features from text by marking up every word in sentences as noun, verb, adverb and so on. Research shows that fake reviews have different distributions of words in part of speech from true reviews.[7]

Using syntax parsing based methods to extract important syntactic features and applying classification methods like SVM or neural networks have pretty good detection results.[7] A lot of early studies were focusing on the syntax parsing methods and among the typical machine learning algorithms SVM has relatively better performance.

2.1.2 *Semantic analysis*

Semantic analysis is another common way to detect fake reviews, in which emotion analysis is especially useful. Emotion analysis focuses on the emotional words in reviews and tries to get these words' distribution. Related research [7] shows that fake reviews always has much stronger emotions than true reviews, either positive or negative. The reasons are straightforward, because fake reviews are always used to impact people, so it is more important to convey opinions than to describe facts.

Sparse additive generative model(SAGE) is used along with the features deriving from semantic analysis.[7] This is a generative Bayesian method and can be regarded as a combination of topic models and generalized additive models. According to the research by Li et al., 2014, the reason why SAGE is tailored for the task is that SAGE constructs multifaceted latent variable models by simply adding together the component vectors rather than incorporating multiple switching latent variables in multiple facets.[7]

2.1.3 *Stylometric analysis*

Besides the two methods introduced above, stylometric analysis is also applied in this problem. Stylometric analysis help us explore the different style of writing in different reviewers, which can generate informative features including lexical features and syntactic features. Pretty much work in similar areas can help this[14]. Ott el al. uses four categories as stylometric features, for example, which includes linguistic processes (e.g., the average number of words per sentence), psychological processes (all social, emotional perceptual processes), personal concerns (anything referred to work, leisure, money, etc.), and spoken categories (primarily filler and agreement words).[13] These prior works provided clear orientation towards how to extract stylometric features within review text.

Related research shows that indeed fake reviews have different styles from true ones. For example, Li et al. points out that fake reviews are more likely to start their text with the first person,[7] which are intentionally designed to make the review more convincing.

2.2 Fake reviewer detection

Fake reviewer detection is the second layer. The fake reviewer detection typically involves misbehaviors analysis for accounts. The behavior features of the accounts can greatly benefit our fake review detection.

Context similarity: It could take too long time for fake reviewers to write new context for every review. So directly copying former fake reviews will be easier. Research shows that high context similarity among reviews indicate higher probability of being fake reviews.[10]

Repeating rating behaviors: Writing multiple reviews for single product is also abnormal since in usual case people do not buy one product for multiple times.[10]

Reviewing burstiness: Research shows that fake reviews are always released in a short period and the accounts are also registered very recently. For example, Mukherjee points out that about 75% of fake reviewers submitted more than 6 reviews in one day, while in normal users, 90% of them will not submit more than 3 reviews.[5]

Early Time Frame: Many fake reviews are released in the early time of certain products, which makes sense since the fake reviewers want to maximize the effects on customers' purchase willing and earlier reviews will influence more people.[8]

Besides, there are a lot of other very interesting domain-specific features waiting to be explored and it proves these behavior facts benefit our tasks more than the linguistic features.

The reviewer behavior characteristics can be combined with linguistic features to feed into all the classification models feasible in the former part. Also, there are some other models that can be specifically designed for the behavior features. For instance, some researchers use graph algorithms to model the user behaviors. They use these features to create graph among users and regard the class they belong to as hidden nodes. Then they applied MRF model to explore the hidden nodes.[5]

3. DATA DESCRIPTION

Our data, Yelp CHI was crawled from Yelp. It was used in[10] and [15]. Yelp CHI contains the reviews of restaurants and hotels in Chicago. We only used restaurant reviews. There are 788471 reviews and 16941 reviewers. The reviews are from Jan. 1, 2006 to Sep. 9, 2012. In our data, there is a label that shows whether this review is filtered by Yelp. There are 402774 filtered reviews and 31878 normal reviews. Yelp has its own software to filter fake/unrelated reviews, as shown in Figure.1 and Figure.2. There is a label in the data that shows if the review was filtered by Yelp, which is the ground truth of our data. The content of the dataset is as follows:

Review profile: date, review ID, reviewer ID, content, rating, useful & cool & funny count for the review, restaurant/hotel ID, label(if filtered);

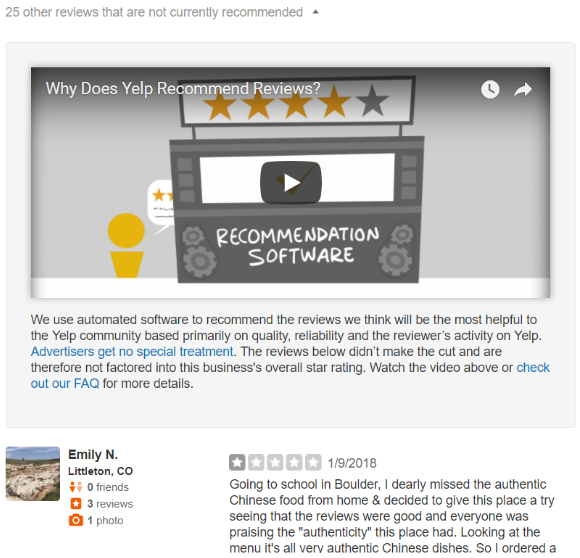
Reviewer profile: reviewer ID, location, name, join date, review count, friend/fan count, useful & cool & funny & compliment count for the reviews;

Restaurant profile: restaurant ID, name, location, reviewCount, overall rating, categories, varchar2(30), Hours, GoodforKids, credit card acceptance, Parking, Attire, GoodforGroups, PriceRange, TakesReservations, Delivery, Takeout, WaiterService, OutdoorSeating, WiFi, GoodFor, Alcohol, NoiseLevel, Ambience, HasTV, Caters, WheelchairAccessible, webSite, phoneNumber, filReviewCount.

4. FRAMEWORK

Fake review detection can be regarded as a typical binary classification problem. Based on this, our work is divided into three parts: data preprocessing, feature engineering and classification. Fig.3 shows the structure of our work.

We tried to drop the unralated information in data preprocessing step and also figured out ways to handle the unbal-



25 other reviews that are not currently recommended

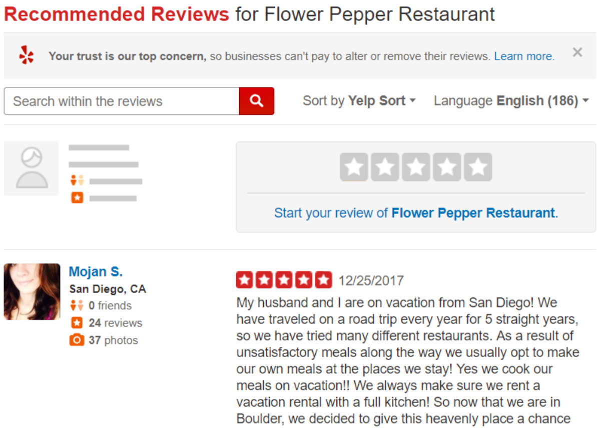
We use automated software to recommend the reviews we think will be the most helpful to the Yelp community based primarily on quality, reliability and the reviewer's activity on Yelp. Advertisers get no special treatment. The reviews below didn't make the cut and are therefore not factored into this business's overall star rating. Watch the video above or check out our FAQ for more details.

Emily N.
Littleton, CO
0 friends
3 reviews
1 photo

1/9/2018

Going to school in Boulder, I dearly missed the authentic Chinese food from home & decided to give this place a try seeing that the reviews were good and everyone was praising the "authenticity" this place had. Looking at the menu it's all very authentic Chinese dishes. So I ordered a

Figure 1: Yelp's filtered review



Recommended Reviews for Flower Pepper Restaurant

Your trust is our top concern, so businesses can't pay to alter or remove their reviews. Learn more.

Start your review of Flower Pepper Restaurant.

Mojan S.
San Diego, CA
0 friends
24 reviews
37 photos

12/25/2017

My husband and I are on vacation from San Diego! We have traveled on a road trip every year for 5 straight years, so we have tried many different restaurants. As a result of unsatisfactory meals along the way we usually opt to make our own meals at the places we stay! Yes we cook our meals on vacation!! We always make sure we rent a vacation rental with a full kitchen! So now that we are in Boulder, we decided to give this heavenly place a chance

Figure 2: Yelp's recommended(non-filtered) review

anced data problem.

In the feature engineering step, we tried to use uniform format for all our date information and not only extract the useful information directly given in the raw data, but explore more advanced complex features through the statistics and observation of our data.

For the classification part, we used many classical machine learning methods plus other useful ensemble methods like XGBoost, Random Forest and Blending method. We also applied neural networks here.

Some of these methods can provide us with scores to rank all our features so that we can know which features among them are relatively more important than others.

5. DATA OVERVIEW AND PREPROSSING

5.1 Extract Related Information

The dataset contains not only reviews for restaurant, but the reviews on reviewers' pages which may be for other businesses. First we need to filter them out. After the filtration, there are 67019 reviews, 19641 reviewers and 129 restaurants in the dataset.

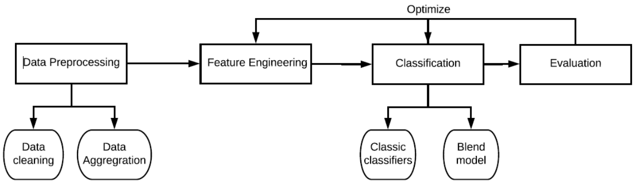


Figure 3: The framework of our work

5.2 Unbalanced Data

According to the statistics, only 11% of the users are repeat buyers, which indicate the ratio of #non-filtered reviews and #filtered reviews is approximately 8 : 1 which is very unbalanced for the classifcation. Therefore, we apply undersampling method here to deal with the unbalancing problem:

Undersampling, the basic idea is to remove some non-filtered reviews from the training data to make the labeled data more balanced. Therefore, we conduct random sampling for non-filtered reviews and filtered reviews to make their ratio decreasing to 4 : 1, our results shows this ratio works good for most classifers.

6. FEATURE ENGINEERING

For general data mining task, feature engineering is an essential, even the most essential part.

Before doing feature engineering, we performed some statistics analysis on our basic information which is directed given in raw data. The analysis results provide very interesting ideas about what kind of complex features we can further create.

Firstly, the average rating of all the reviews is 3.98.The average raing for the two groups:

- avg rating for filtered reviews: 3.86
- avg rating for non-filtered reviews: 3.99

However, when we look at the distributions of the rating for the two groups (Figure. 4, Figure. 5 and Figure. 6). We can obiously find that the filtered tend to give more extreme ratings (1 or 5).

We also calculated the average number of friends of the reviewers. The average friends of all users is 19.82.

- avg friends of users who have filtered reviews: 2.32
- avg friends of users who don't have filtered reviews: 28.91

Another interesting phenomenon we observed is that the average number of reviews in the two groups are also different. Average number of reviews for each user is 19.93.

- avg reviews of users who have filtered reviews: 6.19
- avg reviews of users who don't have filtered reviews: 58.43

These significant divergence of statistics between filter reviews and non-filters offers a lot of useful information and therefore can inspire us to create related advanced features. We will explore and utilize more of such kind of features in our complex feature part.

For this task, it is also an important part. We plan to generate different types features, which are described as follows:

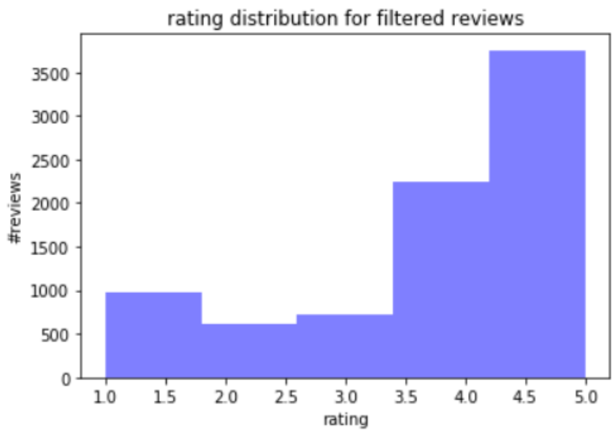


Figure 4: Rating distribution for filtered reviews

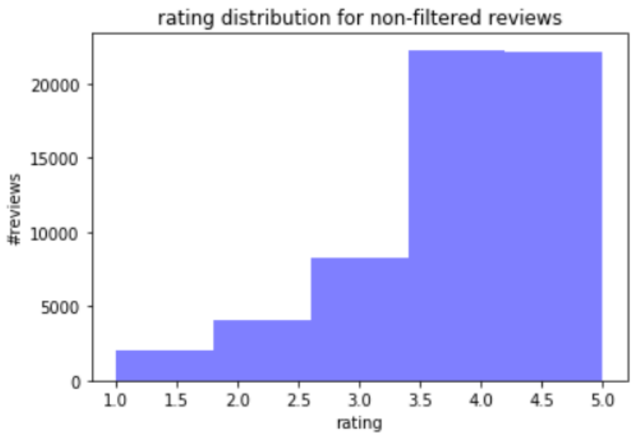


Figure 5: Rating distribution for filtered reviews

6.1 Basic Features

Basic features can give us first-hand information about the data. For example, the number of friends and average rating of each reviewer we talked about are two preliminary discriminative features. Number of reviews a reviewer wrote is also a useful feature, because fake reviewers tend to write some fake reviews and earn some quick money and quit. More reviews written usually means a benign reviewer. In addition, the "useful", "funny", "cool" remarks a review received is an indication whether this review is reliable. We extracted some of these basic features to help differentiate different reviews.

6.2 Language Features

Although language plays not as important as behavior features[10], it still can provide us with some useful hints whether the review is fake. From the plain review text, we extracted the text length and the content similarity(Cosine Distance) as our language features.

6.3 Entities related features

There are three types of entities in this task: review, reviewer and merchant(restaurant and hotel). Features we will extract are mainly based on their profile. For reviewers, there are filtered review ratio, number of restaurants reviewed, number of cities the reviewed restaurants locate in, average rating, etc. For example, the ratio of fake(filtered) reviews of a reviewer, which comes from the intuition that reviews from a reviewer with a high ratio of filtered reviews have more possibility to be fake reviews. For reviews, there are date, rating given, useful count from other customers, etc. If a review has more useful count, it's less likely to be fake. For merchants, there are average rating, number of reviews, days since on Yelp, etc. Usually fake reviewers are hired by the newly opened restaurants/hotels to write some good reviews to boost their business, which means merchants with a small "days since on Yelp" are more likely to have fake reviews.

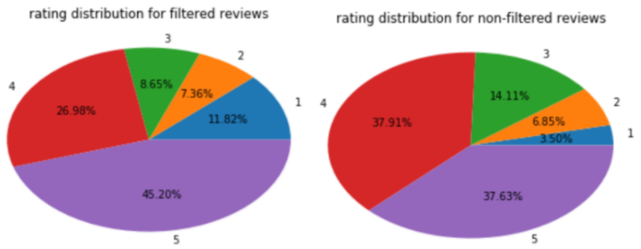


Figure 6: Rating percentage for the 2 types of reviews

6.4 Complex Features

We also generated complex features from the relationship network. This part is the most important one in feature engineering. We analyzed the business background behind the fake reviews and extracted the most possible features which might indicate the signs of suspicious or malicious reviews.Based on the literature survey in former part, we can also get pretty many inspirations.

6.4.1 Average Days between Reviews

The fake reviewers tend to release comments in a short time, which is pretty obvious due to the prohibitive cost of raising an account for a long time before using it for fake reviews. So the fake reviews tend to be written by some new accounts in a quite short time. Therefore the average time between reviews, in other words the frequency of releasing reviews, is an important sign of fake reviews.

6.4.2 Review Length of Reviewer

Based on the comparison of average length of filtered reviews and non-filtered reviews, we find there is a big difference. People tend to choose not spend much time in non realistic things like making up reviews. Thus the filtered reviews are generally shorter than the non-filtered ones. Figure 7 and figure 8 shows the difference.

6.4.3 Review Similarity of Reviewer

Since fake reviewers sometimes have to make up several different reviews in a short time, it is hard to create new reviews every time. It is the commom case when they choose to copy certain sentences from previous fake reviews.This truth also provides us with a new idea to identify them. We compared every pair of all combinations of reviews from one reviewer and calculated the cosine similarity of these pairs

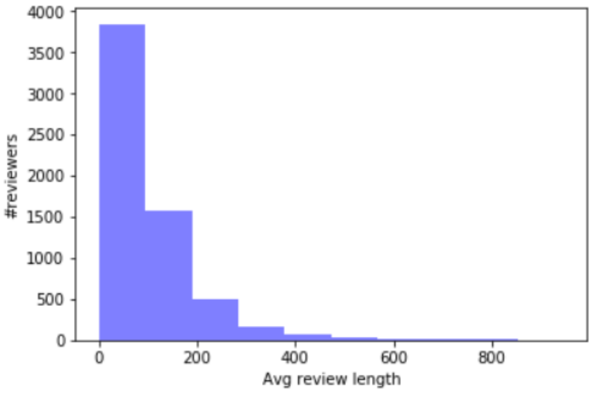


Figure 7: Average review length of reviewer in all reviewers distribution for filtered reviews

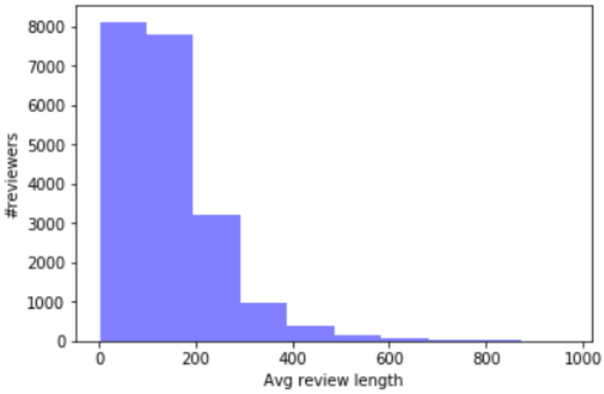


Figure 8: Average review length of reviewer in all reviewers distribution for non-filtered reviews

with using the maximum similarity as new feature. The reviewers with high value of max-similarity will have reasons to be believed to write fake reviews.

6.4.4 Extreme Rating Ratio of Reviewer

The motivation behind fake reviews, of course, is to convince people of some certain opinions. Therefore, fake reviews always come with extreme rating like one star or five star in order to convey strong emotions to the readers. According to this, we calculated the extreme rate (1 star or 5 star) ratio for every reviewer and used the ratio as one feature of every review.

6.4.5 Rating Deviation from the Mean Rating

Reviews with obviously divergence from everyone else's opinion are suspicious. Even though some people might have their own opinions, most reviews which are obviously different from others could be misleading and thus we calculated the raing deviation from average rating as one feature. As shown in Figure 9 and Figure 10 below. ——————————-

6.4.6 Time Ratio of Review in All Reviews

Fake reviews are always created in the earlier time because the earlier they write the more people they can effect. Based on this assumption, we sorted all the reviews belonging to the same restaurant by the date and then set the first review as 1 and the last one as 0. All other reviews between these two are assigned weight according to the ratio of the index to the total number of reviews. The time of review being created can be represented by the weight and it can be a good indicate of fake reviews.

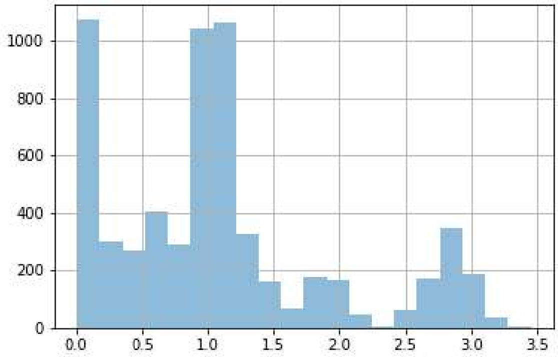


Figure 9: Rating Deviation from the Mean Rating for Filtered Reviews

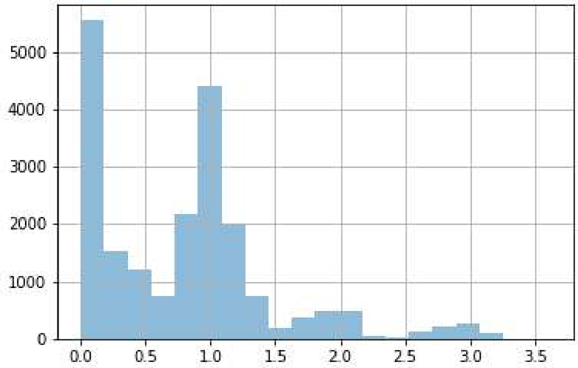


Figure 10: Rating Deviation from the Mean Rating for Non Filtered Reviews

6.4.7 Max Review Numbers per day of Reviewer

Accounts with filtered reviews are always new accounts and they generally write a lot of reviews in a short time. On the other hand, normal accounts do not buy a lot of things and write reviews on them in one day. So the max review numbers of fake reviewer are higher than normal ones. We calculated the max number of reviews in one day for every reviewer ID and used the value as another feature to identify fake reviews.

7. CLASSIFICATION AND EVALUATION

After exploration, we finally used 16 features to train our models: review's rating, review's useful count("useful" remarks received), funny count, cool count, useful count for each reviewer, compliment count (compliments received by a reviewer), fan count(number of a reviewer's followers), tip count(times a reviewer gave tips), friend count, review's length, average days between reviews for each reviewer, max review similarity of reviewer, extreme rating ratio of reviewer, max review numbers per day of reviewer, rating deviation from restaurants mean rating, time ratio of review in all reviews.

Based on all the extracted features we have trained multiple classification models, Logistic Regression(LR)[11], Support Vector Machine(SVM)[3], Random Forest(RF)[2], Adaboost[6], Xgboost[4] and neural networks(NNs)[1]. In order to achieve better results, we also blend those aforementioned models together.

In this section, the classification models we deployed will be described as well as their evaluation results.

7.1 Logistic Regression

Logistic Regression(LR) is a widely used classification which is based on Logistic Regression model to analyzing a dataset in which there are one or more independent variables that determine an outcome. It is a binary classification to classify two possible outcomes. In training the LR model, we scaled the features first and did the classification. We select 20% of the data randomly as testing data while using the remaining to train this model. The best AUC we can achieve with LR is 0.932.

7.2 Support Vector Machine

Support Vector Machine(SVM) is a very widely used regression and classification model. It could perform both linear and non-linear classification very well. The key strengths of SVM can be concluded as follows: first, SVM is different from traditional statistical methods since it generally does not involve the probability measure and the law of large numbers it is more simplified; second, the decision function of SVM is determined only by a few support vectors, in this case, the complexity of computation depends on the number of support vectors rather than the dimensions; third, SVM is robust and it currently could support the incremental data. However, there is an obvious limitation of this model, which is that when the size of training data is quite large(generally more than 10000 items), the training process will become very slow and less efficient.

In our experiment, we scale the data in advance as well. While we were training, we deployed C-SVC as our SVM type and radial basis function(RBF) as our kernel function. After grid search method, we find our optimized parameters C and gamma(C is the parameter for C-SVC and gamma is the parameter for kernel function).

With SVM, the best AUC we can achieve is 0.888.

7.3 Random Forest

Random Forest(RF) is an ensemble learning model for classification rather than a single decision tree. It grows many classification trees. In order to classify an object from an input vector, the classifier puts the input down through each of the trees in the forest. Each tree gives a classification, and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest). In training the model, we scaled the features first and employed the grid search to choose the best parameters. We focus on the parameters of "n_estimators", "min_samples_split", "min_samples_leaf" and "max_depth". The features we use are the same as other classifiers.

RF as a practical bagging method can boost our performance to AUC value of 0.954.

7.4 AdaBoost

Adaboost algorithm combines the output of other weak learning algorithms into a weighted sum. At each iteration of the training process, the weight is assigned to each sample based on the current error on that sample. These weights will be used to inform the training of the weak learning algorithms. In our experiment, we use decision tree as our weak learners. After scaling the features, we used grid search method to find the best parameters, including "n_estimators" and "learning_rate".

Adaboosting is such kind of algorithm that focuses on reducing training error. We achieved the best performance using Adaboost with AUC value as high as 0.964.

7.5 Xgboost

XGBoost is an open-source software library which provides the gradient boosting framework. Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function[16]. We also used this model to rank the features, which will be described later.

The best AUC performance of Xgboost is also 0.964.

7.6 Blend Model

To further improve the performance of prediction, we deploy a blending model to combine all the single classifiers together. The blending model is basically a weighted sum following the formula:

$$p = \sum_{i=1}^{k} w_i * p_i, s.t. \sum_{i=1}^{k} w_i = 1$$

where p is the probability that a specific user will become a repeated buyer of a specific merchant after blending, pi is the probability predicted by the ith classifier, w_i is the weight assigned to the ith classifier and k is the number of our models. Here we manually assign weights to our single models, where single models with higher AUC score receive bigger weights. All the results of the classification are in Table.1 and Figure 11.

7.7 Feature Ranking

In this section, we use the feature ranking function of XGBoost to analyze the importance of each feature and rank the features based on F score which indicates which indicates how many times the feature split on when constructing the boosted decision trees. The higher the f score, the more important the feature.

The results are shown in Figure.12. The rating deviation from mean rating of the restaurant, time ratio of review in all reviews, average review length, review count, and max review similarity of reviewer are the top 5 features that dominant the classification, which proves that our complex feature extraction is effective.

8. CONCLUSION AND DISCUSSION

Table 1: Classification results

	LR	SVM	RF	AdaBoost	Xgboost	NN	Blend Model
Accuracy	0.869	0.833	0.795	0.913	0.912	0.873	0.874
AUC	0.932	0.888	0.954	0.964	0.964	0.934	0.966
Recall	0.768	0.599	0.765	0.824	0.807	0.734	0.829
F1 score	0.734	0.628	0.78	0.817	0.809	0.732	0.819

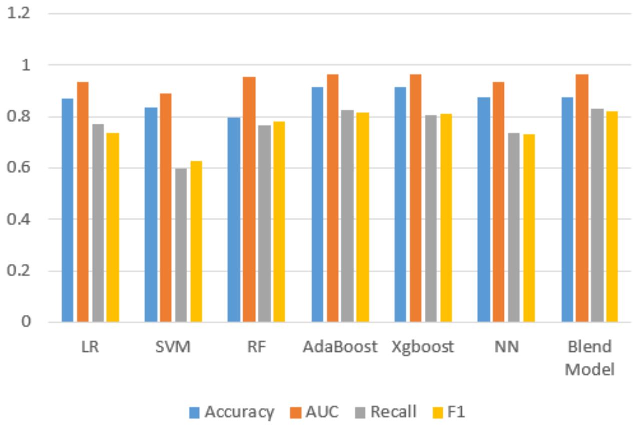


Figure 11: Classification results

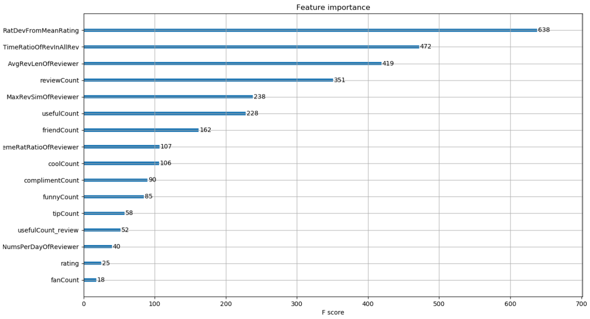


Figure 12: Feature ranking results

We proposed a new solution to detect fake/unrelated reviews in Yelp dataset during our project. In order to achieve our goal, we conducted data preprocessing, feature engineering, classification and evaluation. The complex features are the key to success. 7 complex features were extracted and we finally used 16 features to train the models.

From the figures of our evaluation results, we can find that when we put all of our generated features together, classifiers could achieve the best performance. Whats more, complex features contribute more in the classification model training process. Our framework finally got the highest classification accuracy 0.874 and AUC 0.966.

Our framework has the potential to be moved to cloud platform, so an interesting thing to consider is the speed of our models. During the training process, we found SVM and AdaBoost are slow, even on our relatively small dataset, so SVM and AdaBoost are not suitable for online training. It is the inherent shortcoming of these classifiers. Other models' speed is better. Figure.13 shows the training time of the classifiers.

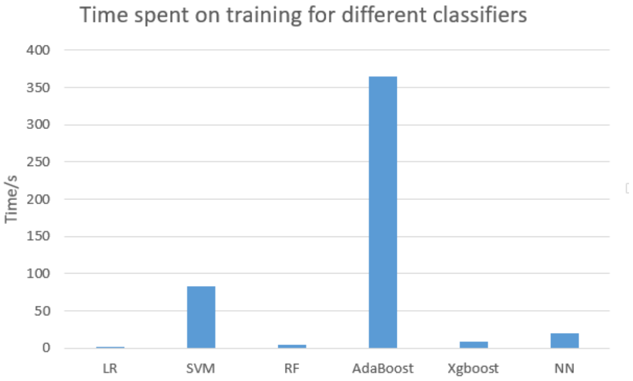


Figure 13: Training time of different classifiers

9. FUTURE WORK

We found that the blending technologies have only slight improvement on the performance, while blending works best when the base models have complementary functions. We might take the advantages of different models into considerations then choose those with complementary characteristics to blend.

The social connections among reviewers are also interesting areas to explore. Fake reviewers are always working together due to the low efficiency of the individual combat. Analyzing the social connections shall help identify such kind of fake reviewer groups.

Besides, one important issue is that we do not obtain enough labeled data for modeling. Crawling more labeled data should help. On the other hand, generative models like GAN is a potential way to solve this problem. Through generating artificial fake reviews and training adversarial models, we can expect to obtain models with better performance.

Finally, the NLP technologies and deep learning are not applied a lot in our project, while in other projects, deep learning can be used to detect toxical comments in Wikipedia. Given the similarities between these tasks, deep learning can be a potentially powerful tool for fake review detection in E-commerce area.

REFERENCES

[1] Y. Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[2] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[3] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27, 2011.

[4] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

[5] G. Fei, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh. Exploiting burstiness in reviews for review spammer detection. *Icwsm*, 13:175–184, 2013.

[6] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[7] J. Li, M. Ott, C. Cardie, and E. Hovy. Towards a general rule for identifying deceptive opinion spam. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1566–1576, 2014.

[8] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 939–948. ACM, 2010.

[9] V. López, S. del Río, J. M. Benítez, and F. Herrera. Cost-sensitive linguistic fuzzy rule based classification systems under the mapreduce framework for imbalanced big data. *Fuzzy Sets and Systems*, 258:5–38, 2015.

[10] A. Mukherjee, V. Venkataraman, B. Liu, and N. S. Glance. What yelp fake review filter might be doing? In *ICWSM*, 2013.

[11] J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Wasserman. *Applied linear statistical models*, volume 4. Irwin Chicago, 1996.

[12] M. Ott, C. Cardie, and J. Hancock. Estimating the prevalence of deception in online review communities. In *Proceedings of the 21st international conference on World Wide Web*, pages 201–210. ACM, 2012.

[13] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock. 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-Volume 1*, pages 309–319. Association for Computational Linguistics, 2011.

[14] J. W. Pennebaker, R. L. Boyd, K. Jordan, and K. Blackburn. The development and psychometric properties of liwc2015. Technical report, 2015.

[15] S. Rayana and L. Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*, pages 985–994. ACM, 2015.

[16] Wikipedia contributors. Xgboost — Wikipedia, the free encyclopedia, 2018.

APPENDIX

A. HONOR CODE PLEDGE

On my honor as a University of Colorado at Boulder student I have neither given nor received unauthorized assistance on this work.

B. INDIVIDUAL CONTRIBUTION

Tao Ruan: Literature Survey. Two models for Mid-checkpoint. Feature Engineering(Complex Feature Extraction).
Yichen Wang: Basic features extraction, preprocessing, classification and evaluation, feature ranking