Detecting Truthful or Deceptive Hotel Reviews through Multinomial Naive Bayes, Logistic Regression, Classification Tree, and Random Forest Methods

YANGFAN CHEN (5194652), YINGLUN PU (8023042), and REEM NAJJAR (9652011), Utrecht University, NL

To help address the issue of pervasive deceptive or fake reviews in online environments, we implement and evaluate four classifiers—Multinomial Naive Bayes, Logistic Regression, Classification Tree, and Random Forest—for detecting deceptive hotel reviews. Testing on a dataset with 1,600 reviews, we explore the effect of adding bigram features to these models, compare performance differences between models, and extract important terms (features) indicative of truthful or deceptive reviews. Our findings contribute to decision-making regarding which methods are most effective for detecting deceptive reviews and provide insights into the features most indicative of review authenticity.

1 Introduction

With the growth of the internet, online booking and shopping have become mainstream. Reviews of an item have a significant impact on people's perceptions of that item. For example, on platforms like Amazon, one of the largest e-commerce sites, reviews are essential to its business model. These reviews help build user trust, enabling customers to quickly place an order for an item. However, this also makes it easier for malicious actors to post fake negative reviews, which can be difficult to distinguish and may disrupt the user's choice.

To identify fake reviews, many researchers are currently trying to address this challenge using various approaches. For example, [Hernández Fusilier 2015] uses PU-learning in their report, building a binary classifier based on positive, to discriminate negative deceptive opinion spams. Additionally, [Crawford 2015] discusses the use of natural language processing to extract meaningful features from the text, as well as the use of machine learning techniques to devise methods for further investigation. In [Saumya 2018], the authors design a system with the innovative use of (i) sentiments of review and its comments, (ii) content-based factor, and (iii) rating deviation, to implement a stand-alone system that can be used to filter product review datasets.

In this assignment We will use a total of 800 fake and genuine hotel reviews, presented in [Ott et al. 2013, 2011]. Based on the theoretical basis shown in [Ott et al. 2013, 2011]. We will use both linear (Multinomial Naive Bayes and Logistic Regression) and non-linear (Classification Tree and Random forest) models for this task.

2 Data Description

For this assignment, the data was obtained from myleott.com, [Ott et al. 2013, 2011]. The corpus used in this research consists of two parts: one focusing on positive sentiment reviews, and the other on negative sentiment reviews [Ott et al. 2013].

The first part of the corpus consists of both truthful and deceptive 5-star reviews. The truthful positive reviews were sourced from TripAdvisor for 20 of the most popular hotels in the Chicago area. These reviews were selected from popular hotels to minimize the risk of extracting opinion spam and incorrectly labeling it as truthful. The hypothesis is that deceptive positive reviews are less likely to be found for popular establishments, as they would have minimal impact.

The deceptive positive reviews were sourced from Amazon Mechanical Turk. Turkers were tasked with writing a positive review while pretending to be part of a hotel's marketing team. Each Turker was assigned a specific hotel and was instructed to complete their review within 30 minutes. To ensure diversity, each Turker could only submit one review. Submissions of insufficient quality (e.g., plagiarized, unreasonably short, or written for the wrong hotel) were rejected. Ultimately, 400 deceptive reviews written by Turkers were combined with 400 truthful reviews from TripAdvisor to form the positive portion of the corpus.

The second part of the corpus is composed of truthful and deceptive 1- and 2-star reviews from popular online review websites, including Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor, and Yelp. While there is no guarantee that these reviews do not contain any deceptive ones, previous research suggests that the proportion of deceptive reviews on travel review portals is relatively small. The deceptive negative reviews were obtained using the same process as the positive reviews, again leveraging Mechanical Turk.

Although the dataset was cleaned and standardized as much as possible, slight inconsistencies may still exist. In total, the corpus contains 1,600 reviews. Each dataset consists of 20 reviews for each of the 20 hotels included in the study. This research will focus on the negative reviews and aims to classify them as either deceptive or truthful.To obtain truly deceptive reviews, [Ott et al. 2013, 2011] collected data from Mechanical Turk. Workers wrote these deceptive negative reviews from the point of view of a person at one of Chicago's most famous hotels, who would imagine and write a false negative review about a rival hotel. Each review was high quality, had the correct hotel information, was readable and not plagiarised, and conveyed a negative sentiment. The average length of these negative fake reviews was also higher than that of the positive reviews.

3 Methods and Results

We employed two linear and two non-linear models to classify truthful and deceptive hotel reviews. For the linear models, we used Multinomial Naive Bayes (Section 3.2) method for a generative linear classifier and Logistic Regression (Section 3.3) for a discriminative linear classifier. For non-linear classifiers, we implemented two models based on Classification Trees (Section 3.4) and Random Forests (Section 3.5) methods. A performance matrix of accuracy, precision, recall and F1-score is used to evaluate the models produced by these

https://www.tripadvisor.com

https://www.mturk.com

methods. Additionally, we compared the performance of only using unigram features versus using both unigram and bigram features.

3.1 Data Pre-processing

In this study we focused solely on classifying truthful and deceptive reviews. Therefore we first combined positive truthful data with negative truthful data and postive deceptive data with negative deceptive data with label 1 and 0 that represent truthful and deceptive respectively. We then split the dataset by using fold1 to fold4 (a total of 640 reviews) as the training set and fold5 (160 reviews) as the test set. We then converted text of both sets to lowercase, removed punctuation, and filtered out stopwords using the English stopwords list from the NLTK library.

3.2 Multinomial Naive Bayes Method

We applied the Multinomial Naive Bayes method to obtain a generative linear classifier. The preprocessed dataset was transformed into a matrix of token counts. We did not use TF-IDF for feature extraction, as Naive Bayes, being a generative model, works better with raw token counts. Observed word frequencies are more informative for our Multinomial Naive Bayes model.

We conducted feature extraction in two ways: using only unigram and using both unigram and bigram. Next, we used the chi-square test to select the top 1,000 features, aiming to retain the most informative features, reduce noise, and potentially enhance the model's performance. After selecting these features, we performed hyperparameter tuning through a grid search, focusing on two parameters for the Multinomial Naive Bayes model: the smoothing parameter `alpha` and the `fit_prior` setting, which determines whether to learn class priors from the data.

To ensure robustness in our model selection, we employed 10-fold cross-validation in the hyperparameter tuning process. This process selected two best models for the unigram and unigram + bigram methods. For both models, the optimal parameters are `alpha = 0.1` and `fit_prior = true`. The result for the performance matrix is shown at Table 1.

3.3 Logistic Regression Method

We employed Logistic Regression as a discriminative linear classifier. Initially, we preprocessed the text reviews by transforming them into a TF-IDF feature matrix, capturing both unigram and unigram + bigram options for feature extraction. To optimize model performance, we conducted hyperparameter tuning with 10-fold cross-validation across a range of regularization strengths (C values: 0.01, 0.1, 1, 10, 100). The model used an L1 (Lasso) penalty to enforce sparsity in the coefficients, allowing us to retain only the most impactful features. Based on cross-validation results, we selected two optimal models that used unigram and unigram + bigram feature extraction, and then trained them on the entire training dataset. For both the unigram and unigram + bigram models, the optimal regularization parameter was found to be `C = 100`. The performance of these models, measured by accuracy, precision, recall, and F1 score, is presented in Table 1.

https://www.nltk.org/

3.4 Classification Tree Method

We trained a Classification Tree classifier as a non-linear model to classify truthful and deceptive reviews. After loading and preprocessing the data, we represented the text as TF-IDF feature matrices using unigram and unigram + bigram features. We applied Chi-square feature selection, selecting the top 1000 features to reduce the dimensionality and focus on the most relevant terms.

For hyperparameter tuning, we employed grid search method with a 10-fold cross-validation to identify the optimal parameters for criterion, max depth and min sample split. Specifically, criterion defines the function to measure the quality of a split, with options including "gini" for the Gini impurity and "entropy" for information gain. Max depth limits the maximum depth of the tree, with values set to None (allowing the tree to expand until all leaves are pure) and fixed depths of 10, 20, and 30. Limiting the depth helps prevent overfitting by constraining the tree's complexity. Min samples split specifies the minimum number of samples required to split an internal node, with option values of 2, 5, and 10. Higher values create broader nodes with more samples, controlling tree growth and reducing the likelihood of overfitting.

Based on cross-validated accuracy, the two best models for (i). unigram and (ii). unigram + bigram were selected for use in predicting labels on the test data. For the unigram model, the optimal parameters are `criterion = `gini'`, `max_depth = 20`, and `min_samples_split = 2`. For the unigram + bigram model, the optimal parameters are `criterion = `gini'`, `max_depth = 20`, and `min_samples_split = 2`. The performance of these two models is shown at Table 1.

3.5 Random Forest Method

We applied Random Forest method, another non-linear model to classify truthful and deceptive reviews. First, we loaded and preprocessed the dataset, transforming the text reviews into a TF-IDF feature matrices, one with unigram features and the other with unigram + bigram features. To reduce the dimensionality and focus on the most relevant terms, we applied Chi-square feature selection, selecting the top 1000 features. For hyperparameter tuning, we employed a grid search with 10-fold cross-validation . Specifically, we focused on the parameters of number of estimators, maximum features and minimum samples split. Number of Estimators determines the number of trees in the forest. We opt for options of 100, 200, and 300 to evaluate the impact of adding more trees on model stability and accuracy. Maximum Features specifies the number of features to consider at each split, with options of "sqrt" (square root of the total features) and "log2" (logarithm base 2 of the total features). This parameter helps to diversify the trees in the forest to mitigate overfitting. We tested values of 2, 5, and 10 for minimum samples split. This parameter functions the same as in Classification Tree method.

Additionally, out-of-bag (OOB) scoring is used to help estimate accuracy, which avoids relying solely on the cross-validation results. Based on cross-validated accuracy, we selected the best model for each feature extraction method—one using only unigram and the other using both unigram and bigram—for final predictions on the test data. For the unigram model, the optimal parameters are

`max_features = 'log2'`, `min_samples_split = 2`, and `n_estimators = 300`. For the unigram + bigram model, the optimal parameters are `max_features = 'log2'`, `min_samples_split = 5`, and `n_estimators = 300`. The performance of these two models is shown in Table 1.

4 Discussion

In this section, we compare the performance of different models and identify important features that indicate truthful and deceptive reviews. We evaluate performance across four metrics: accuracy, precision, recall, and F1-score. We use statistical test for accuracy comparison. Although McNemar's test is a more comprehensive approach that accounts for paired comparisons, we applied two-tailed binomial test since it aligns with our goal of comparing accuracy across models. As for comparisons of precision, recall, and F1-score, we compare the values directly without a statistical test.

4.1 Performance Difference between models with and without bigram feature extraction

Our first point of interest is whether adding bigram feature can impact models' performance. For each pair of models with and without bigram features, we propose null hypothesis as: there is no difference between models with and without bigram features. To test these hypotheses, we applied four two-tailed binomial test for four pairs of models. To mitigate the increased risk of Type I errors arising from multiple comparisons, the Bonferroni correction was applied. The p-values and Bonferroni adjusted p-values for each comparison are presented in Table 2. The adjusted p-values suggest that, at the significant level of $\alpha = 0.05$, extracting feature using (i). unigram, and (ii). unigram + bigram do not significant influence model's accuracy. When considering precision, recall, and F1-score, we see worse performance on Logistic Regression and Random Forest models and better performance on Classification Tree and Multinomial Naive Bayes model. However, this conclusion regarding precision, recall and F1-score is not supported by a statistic test and therefore cannot exclude the possibility that the difference is caused mere by chance.

Table 2. Two-tailed binomial test results of adding bigram features

Model	P-value	Adjusted P-Value
Multinomial Naive Bayes	0.7905	3.1620
Logistic Regression	1.0000	4.0000
Classification Tree	0.7428	2.9712
Random Forest	0.1686	0.6744

Since the difference of adding bigram feature is not significant, for simplification and better comparison, we decided to represent each model with only one feature extraction approach. In the following sections, each model are represented by its unigram feature extraction version.

4.2 Performance Difference between Generative Linear Model and Discriminative Model

We investigated the performance difference between generative (i.e., Multinomial Naive Bayes in our study) and discriminative (i.e., Logistic Regression in our study) models. For the performance metric of accuracy, since our Multinomial Naive Bayes model has higher accuracy value, we propose our null hypothesis as:

NH: There is no difference between Multinomial Naive Bayes and Logistic Regression Models.

To test this null hypothesis, we applied one-tailed binomial test and came out with the result p-value of 0.0925. At significant level of $\alpha = 0.05$, we failed to reject the null hypothesis and conclude that there is not significant accuracy difference between Generative Linear Model (Multinomial Naive Bayes) and Discriminative Model (Logistic Regression). For other performance matrices, precision, recall and F1-score, our Multinomial Naive Bayes model performs better than Logistic Regression model. However, this conclusion is not supported by a statistic test.

4.3 Performance Difference between Random Forest Model and Linear Classifiers

We are also interested in the performance difference of Random Forest Model and Linear Classifiers (Logistic regression and Multinomial Naive Bayes models).

According to Table 1, for models with unigram feature extraction, Random Forest model have highest accuracy. Therefore for accuracy comparison, we propose two null hypotheses:

NH1: The accuracy of Random Forest model is not higher than the accuracy of Logistic Regression model.

NH2: The accuracy of Random Forest model is not higher than the accuracy of Multinomial Multinomial Naive Bayes model.

For testing each null hypothesis, we ran an one-tailed binomial test with the Bonferroni correction. The results are shown in Table 3. With significant level of $\alpha = 0.05$, we failed to reject both null hypotheses and cannot conclude a significant advancement of accuracy between Random Forest Model and Linear Classifiers. For other performance matrices, precision, recall and F1-score, our Random Forest model performs better than Logistic Regression model and similar to Multinomial Naive Bayes model. However, this conclusion is not supported by a statistic test.

Table 3. One-tailed binomial test results between Random Forest model and Linear Classifiers

Model Pair	P-value	Adjusted P-value
RF - LR	0.0610	0.1221
RF - MNB	0.5000	1.0000

Table 1. Performance comparison matrix across different models.

Model	Accuracy	Precision	Recall	F1-score
Multinomial Naive Bayes (Unigram)	0.8500	0.8182	0.9000	0.8571
Multinomial Naive Bayes (Unigram + Bigram)	0.8625	0.8372	0.9000	0.8675
Logistic Regression (Unigram)	0.8000	0.7927	0.8125	0.8025
Logistic Regression(Unigram + Bigram)	0.8000	0.8243	0.7625	0.7922
Classification Tree (Unigram)	0.6250	0.6163	0.6625	0.6386
Classification Tree (Unigram + Bigram)	0.6875	0.6923	0.6750	0.6835
Random Forest (Unigram)	0.8562	0.8276	0.9000	0.8623
Random Forest (Unigram + Bigram)	0.8063	0.8101	0.8000	0.8050

4.4 Important features pointing towards truthful or deceptive review

To extract important features from our classifiers, we first need to ensure our models' accuracy performs better than by chance. We propose null hypothesis for each model that the accuracy of the model is not greater than by chance, and run one-tailed binomial test for each model with the Bonferroni correction. The results are shown in Table 4.

Table 4. One-tailed binomial test results comparing model performance to chance

Model	P-value	Adjusted P-value
Multinomial Naive Bayes	1.72×10^{-20}	6.88×10^{-20}
Logistic Regression	4.21×10^{-15}	1.68×10^{-14}
Classification Tree	0.0010	0.0040
Random Forest	2.98×10^{-21}	1.19×10^{-20}

At significant level of $\alpha = 0.05$, we can reject all null hypotheses and conclude that all models perform better than by chance. Therefore it is meaningful to extract important features from these models.

We extract important features from all four classifier models individually using different methods:

- For Multinomial Naive Bayes model, we extract the feature importance using the log-odds of each feature. Specifically, we compute the log probability of features for each class, and determine their contribution based on the difference in log-odds between truthful and deceptive reviews.
- For Logistic Regression model, we use the coefficients of the features as a measure of their importance. The magnitude of the coefficient indicates how strongly the feature contributes to the prediction of a truthful or deceptive review.
- For Classification Tree model, we use SHAP (SHapley Additive exPlanations) [Lundberg 2017] values to interpret feature importance. We calculate the average SHAP values for each feature, highlighting the features that contribute the most towards predicting a review as either truthful or deceptive.
- For Random Forest model, we also use SHAP values for feature importance analysis. By averaging the SHAP values for each feature across all trees, we identify the most important features that help the model distinguish between truthful and deceptive reviews.

Based on these methods, we extracted the five most important features for truthful or deceptive reviews for each model, along with an additional five important features to cross-compare different models for a comprehensive view. The results are shown in Table 5.

The distinction between truthful and deceptive reviews can be largely attributed to linguistic style and writing strategies. Truthful reviews are characterized by specific, concrete details, often featuring nouns like "fridge," "elevator," and "requests." These verifiable elements reflect genuine personal experiences. In contrast, deceptive reviews often lack these specific details, opting instead for generalized language, such as "decided" or "settled," which can apply to any situation. Additionally, deceptive reviews frequently use emotionally charged adjectives like "hate" and "loved", aiming to evoke strong feelings and make the review seem more compelling. This exaggerated emotional content contrasts with the more balanced tone found in truthful reviews.

Moreover, deceptive reviews tend to use a higher frequency of adverbs such as "recently" and "finally" to emphasize sequence or events, which may make the narrative appear more structured or rehearsed. This use of adverbs may create an impression of overemphasis, which may be less common in genuine accounts.

4.5 Assumptions of binomial test

In applying the binomial test within this study, it is essential to ensure that the key assumptions [Freund and Wilson 2003] are met to validate the results:

- The test assumes that each classification outcome is independent. In this study, the models were evaluated on the same dataset, where each review classification is treated independently. Given that there is no dependency among the individual reviews, this assumption holds.
- The binomial test requires binary outcomes. Here, each classification is either correct or incorrect, aligning with the binary nature required by the test. Thus, this assumption is met.
- The probability of correct classification should remain constant across trials. In this study, the same dataset and consistent model conditions were maintained, supporting this assumption by ensuring no variation in success probability.

Table 5. The five most important terms (features) pointing towards a truthful or deceptive review

	Nr.	Multinomial Naive Bayes	Logistic Regression	Classification Tree	Random Forest
Truthful					
	1	priceline	star	chicago	downtown
	2	sofa	world	great	certainly
	3	fridge	cant	michigan	maid
	4	stated	clothing	hard	sensors
	5	honor	construction	found	spring
	6	seasons	returned	elevator	finally
	7	spoon	requests	concierge	luxurious
	8	removed	location	food	needless
	9	thru	line	sheets	settling
	10	blocks	outdated	weekend	currently
Deceptive					
	1	relax	chicago	decided	storm
	2	originally	finally	loved	hate
	3	settled	prices	conference	450
	4	eggs	recent	street	claimed
	5	steak	recently	room	advertised
	6	claims	smelled	construction	13
	7	hate	turned	hotel	pictures
	8	yellow	smell	pm	vacuum
	9	shown	decided	returned	nicer
	10	demanded	make	seasons	concierge

- Sufficient sample size is required for reliable results. The test dataset contains 160 reviews, which provides an adequate sample for the test.

Overall, the assumptions required for the binomial test are satisfactorily met in this study, validating its use for comparing model accuracy.

5 Conclusion and Future Work

In this study, we investigated the performance of four different classifiers for detecting opinion spam in hotel reviews. By applying Multinomial Naive Bayes, Logistic Regression, Classification Tree, and Random Forest methods, we evaluated their performance using metrics such as accuracy, precision, recall, and F1-score. Our findings indicate that, in the context of this study, adding bigram features does not significantly influence accuracy. It improves precision, recall, and F1-score for the Multinomial Naive Bayes and Classification Tree models, while decreasing these metrics for the Random Forest model. There is no significant difference in accuracy between the generative and discriminative models in our study. However, for precision, recall, and F1-score, the generative model performs better than the discriminative model. We also found no significant difference in accuracy between the Random Forest model and the two linear models used. For precision, recall, and F1-score, the Random Forest model performs better than the Logistic Regression model and similarly to the Multinomial Naive Bayes model.

As for the most important terms or features indicating whether a review is truthful or deceptive, we applied a feature extraction method for each model. We found that different models interpret features in diverse ways. According to our results, truthful reviews tend to contain specific and concrete nouns, while deceptive reviews use more emotional and generalized language. This may indicate differences in linguistic style and writing strategies between truthful and deceptive reviews.

Future work could involve further refinement of the parameters used in the models and the establishment of a more comprehensive data processing pipeline to enhance the performance of these methods. Additionally, exploring neural network-based approaches to train classifiers could provide improved results and offer more robust insights into the detection of opinion spam. Leveraging deep learning techniques could help uncover additional nuanced patterns that may not be fully captured by traditional machine learning models, thereby complementing the existing methods.

References

T. M.; Prusa J. D.; Richter A. N.; Al Najada H Crawford, M.; Khoshgoftaar. 2015. Survey of Review Spam Detection Using Machine Learning Techniques. *Journal of Big Data* 2 (1) (2015), 1–24.

Rudolf J Freund and William J Wilson. 2003. *Statistical methods.* Elsevier.

M.; Rosso P.; Guzmán Cabrera R Hernández Fusilier, D.; Montes-y-Gómez. 2015. Detecting Positive and Negative Deceptive Opinions Using Pu-Learning. *Information Processing and Management* 51 (4) (2015), 433–443.

Scott Lundberg. 2017. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874* (2017).

Myle Ott, Claire Cardie, and Jeffrey T Hancock. 2013. Negative deceptive opinion spam. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: human language technologies*. 497–501.

Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. *arXiv preprint arXiv:1107.4557* (2011).

J. P. Detection of Spam Reviews Saumya, S.; Singh. 2018. Detection of Spam Reviews: A Sentiment Analysis Approach. *Csi Transactions on Ict* 6 (2) (2018), 137–148.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009