Yelp Fake Review Detection

Dublin City University
School of Computing
CA683: Data Analytics and Data Mining
Assignment 2 Final Report
Date of submission - 22nd April 2019
Course – MCM

Student Number	**Name**	**Mail ID**
18210298	Aishwarya Gupta	aishwarya.gupta3@mail.dcu.ie
18210295	Apurva Gawad	apurva.gawad2@mail.dcu.ie
18210395	Archana Hule	archana.hule2@mail.dcu.ie
18210455	Gautam Shanbhag	gautam.shanbhag2@mail.dcu.ie
18210026	Vishvesh Kadam	vishvesh.kadam2@mail.dcu.ie
18210686	Paritosh Gupta	paritosh.gupta3@mail.dcu.ie

Declaration

We declare that this material, which we now submit for assessment, is entirely our own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. We understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should we engage in plagiarism, collusion or copying. We have read and understood the Assignment Regulations set out in the module documentation. We have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged, and the source cited are identified in the assignment references. This assignment, or any part of it, has not been previously submitted by us or any other person for assessment on this or any other course of study.

We have read and understood the referencing guidelines found recommended in the assignment guidelines.

Name: **Date: 22th April 2019**

Aishwarya Gupta (18210298)
Apurva Gawad (18210295)
Archana Hule (18210395)
Gautam Shanbhag (18210455)
Vishvesh Kadam (18210026)
Paritosh Gupta (18210686)

Abstract

Online reviews play a pivotal role in helping people purchase products eventually influencing the business verdicts. Online reviews have led to fake review writing, which can either be paid human writers or machine generated deceptive reviews with the aim to influence future customers opinion. In this project we have tried to tackle this problem with the help of a classifier that processes the review text and user's behavioral pattern as input and predicts whether the review is genuine or not. The learning algorithms we experimented include logistic regression, multinomial Naive Bayes, K-Nearest Neighbour (KNN), Random Forest classifier, Convolutional Neural Network (CNN) and CNN with Long short term memory (LSTM). From the results of our experiment we can see that Logistic Regression and KNN performed better with approximately 60-64% accuracy and Naive Bayes and Random Forest Classifier (RFC) does not work well with our dataset with only 50% accuracy and CNN-LSTM does not give a very high accuracy (21%) i.e. but has a good recall of 0.82.

I. Introduction

Online Reviews have become a predominant source of decision making for customers. However, with the advent of online reviewing forums, there has been a rise in the occurrence of Deceptive Opinion Spamming (Mukherjee,Venkataraman et al, 2013). For example, a significant percentage of reviews on Yelp are estimated to be fake by paid human writers . Even though many of the online websites do not attempt to identify and filter fake reviews, Yelp is an exception and has been filtering reviews since past few years (Mukherjee,Venkataraman et al, 2013). The menace has soared to such serious levels that Yelp.com has launched a sting operation to publicly shame businesses who buy fake reviews(Streitfeld, 2012b). In order to provide users with more reliable review information, we aim to build a classification system to detect fake reviews. For achieving this task we have acquired a labelled dataset to train our model, more information regarding the dataset can be found in section III, post cleaning the dataset, paying close attention to cleaning the review text, we have extracted linguistic as well as behavioral features from the dataset to enable our model to detect deceptive reviews. Consequently, we have applied three classifying algorithms onto our dataset, evaluated the performance of these algorithms and compared our result.

II. Literature review

(Mukherjee,Venkataraman et al, 2013) in their paper have attempted to uncover the trade secret of how Yelp.com is filtering its reviews. They have analyzed the techniques that are incorporated for detecting crowdsourced fake reviews generated using Amazon Mechanical Turk (AMT) and put these methods to test on the yelp dataset, from their research they found that behavioral features play an important role in detecting yelp fake reviews and linguistic features are less effective, learning from this we have included both behavioral as well as linguistic features in our dataset.

(Wang, Z et al) in their research have extracted various user centric features from the dataset for their analysis, they have used various methods for classifying their data and found that Neural network classifier provides the best performance in term sof accuracy, however other methods like Logistic regression also provide an acceptable performance.

(Rout et al, 2017) in their research have worked on many ways in which deceptive reviews can be detected using Supervised Learning as well as Unsupervised learning, since our dataset was labeled we concentrated mainly on the Supervised learning aspect of their research.They have used various linguistic features for fake review detection out of which we found text categorisation i.e. N-gram as a feature which enables us to model parts of text as a feature. They have used Unigrams and Bigrams for their dataset, however we found that Trigrams give us a better accuracy in our dataset. They have

also used Sentiment as a feature, this is due to the fact that many fake reviews tend to have a high negative or positive sentiment as they are written to influence the reader.

(Chowdhary et al, 2018) have also performed research in deceptive opinion spamming and worked on Amazon's fake reviews. They have used behavioral as well linguistic features for their analysis and found good results using Naive bayes and Random Forest Classification. We also implemented the same methods on our dataset but found the results to be inconsistent with them, this shows that the people writing fake reviews for Amazon are using different techniques than the ones writing for Yelp.

III. Dataset

The data that we have acquired originally belonged to Ryana and Akoglu who are the original researchers in the area of deceptive opinion spamming. They had collected this data from Yelp.com. The data included of five TSV files having information about Users, Review Content, products which consisted of restaurants and hotels and a mapping of review with label of it being true or fake. After merging the files together we were able to get our final dataset as per snapshot below:

User_Id	Prod_Id	Date	Review_Text	Rating	Label	Product Name
923	0	12/8/2014	The food at snack is a selection of popular Gr...	3	-1	Snack
924	0	5/16/2013	This little place in Soho is wonderful. I had ...	3	-1	Snack
925	0	7/1/2013	ordered lunch for 15 from Snack last Friday. ...	4	-1	Snack

Fig. 3.1: Actual Dataset

The Label field indicates whether the review is true i.e. 1 or fake i.e. -1. The dataset consists of 923 products and 160201 users, leading to a total of 358,957 records. The data set is skewed having a higher number of true records i.e 322097 compared to fake records i.e. 36860. From these records we can see that we have an unbalanced dataset with approximately 10% of the records as fake. Since, an unbalanced dataset can lead to false results we have used Oversampling techniques to overcome this issue., more information about this can be found in section --nu--. We divided the dataset into 75% training set and 25% testing set for our evaluation.

IV. Data Preprocessing

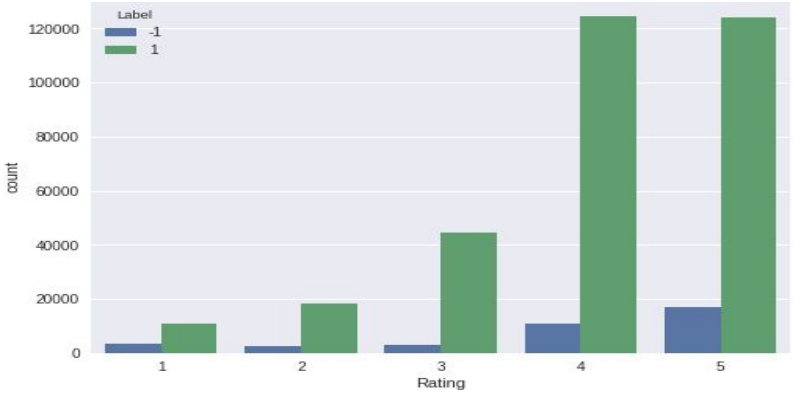
For pre-processing our text data, we first checked for null and missing values. For our data, there were no missing and null values. Then, we did some text pre-processing which includes- removing stop words (like: is, the, that, and, etc.), making entire text to lowercase, removing punctuations and removing numbers. Although numbers and capital words suggest important information, we removed them for simplicity of training models.

The next main thing was to apply stemming. Stemming is a process of checking for words with similar context but different spellings (like amaze, amazing, and amazed) and reducing all into one word (like amaz) usually by removing suffix. This help in reducing unnecessary features for training NLP. There are different stemming models in NLTK like porter and snowball. We have considered porter model to perform stemming. Finally, after applying all cleaning steps, we created a new corpus and used it for further analysis and machine learning.

V. Exploratory Data Analysis

Following visualizations were made to understand more about the nature of fake and true reviews. Our focus is to understand the pattern followed in the fake reviews

A) We can see below that most of the fake reviews that are labelled -1 and highlighted blue in colour are mostly rated as 4 and 5. The hypothesis deduced by the former statement is that these deceptive reviews might be a marketing strategy to promote their products and deceive the customers.



B) In order to strengthen the hypothesis made above, world cloud of fake and genuine reviews were created for comparison. Basically, word clouds is a display of a set of words in the form of a cloud. The frequency of a word in review text decides the size of that particular word, bigger the size of the word means that word occurred more frequently in the review text. Thus, it helps us to understand the pattern of most commonly used words.

Following is the comparison of fake word cloud with true word cloud:

Fake word cloud

Genuine word cloud

We can see that the degree of positive adjectives have been used in the fake review's word cloud more than in genuine word cloud which might be a business strategy to seek popularity amongst the competitors.

VI. Feature Engineering

As per our understanding of the Literature review, the accuracy of the model can be improved if new features extracted contain behavioural features as well as text features.

Behavioural features are those features which describe the dataset such as aggregated features. As per our dataset the features which we derived from the base table are average user reviews, average restaurant rating and count of individual user reviews.
Coming to text feature extraction, these features were derived purely from the raw review column present in the dataset.
Features like Number of Nouns present in each review, Number of capital Words, Number of Digits in the review, Review length were few of the features extracted.

Our primary assumptions based on which these features were extracted were that people who add genuine reviews normally add proper nouns to their reviews. People emphasis on words of text by adding it in Capital or Bold, use numbers to rate or provide information such as cost of a dish. Fake Reviews usually follow templates or short word counts. Thus review length was extracted too.

Next, we extracted Sentiment Score using VADER lexicon and rule based sentiment analysis using pre built python library.

Sentiment Analysis, or Opinion Mining, is a sub-field of Natural Language Processing (NLP) that tries to identify and extract opinions within a given text. The aim of sentiment analysis is to gauge the attitude, sentiments, evaluations, attitudes and emotions of a speaker/writer based on the computational treatment of subjectivity in a text.

VADER (Valence Aware Dictionary and Sentiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. VADER uses a combination of A sentiment lexicon is a list of lexical features (e.g., words) which are generally labelled according to their semantic orientation as either positive or negative.

VADER has been found to be quite successful when dealing with social media texts, NY Times editorials, movie reviews, and product reviews. This is because VADER not only tells about the Positivity and Negativity score but also tells us about how positive or negative a sentiment is.

It is fully open-sourced under the MIT License. The developers of VADER have used Amazon's Mechanical Turk to get most of their ratings [7].

Sentiment Analysis extracts 4 features ie Positive, Negative, Neutral and Compound Score. Positive, Negative and Neutral scores represent the proportion of text that falls in these categories. These fall between 0 to 1 and are normally proportional values. These 3 should add up to 1.

The Compound score is a metric that calculates the sum of all the lexicon ratings which have been normalized between -1(most extreme negative) and +1 (most extreme positive).

Machine learning algorithms operate on a numeric feature space, expecting input as a two-dimensional array where rows are instances and columns are features [8]. Thus we had to vectorise our review column such that it could be understood by the ML model.

Before directly passing it for vectorisation, we cleaned the review column to extract a new corpus column. We stripped out special characters, numerical values, stop words, from the review column. Converted every word in review to lower case. We then created a new bag of words which were basically the most common words in both fake reviews and true reviews, took the top 20 common words and added that too stop words. Thus our final corpus column was ready for vectorisation.

We decided to try out Count Vectorisation, TF-IDF Vectorisation and Word2vec for our problem statement.

Word2Vec is a neural network structure to generate word embedding by training the model on a supervised classification problem. CountVectorizer is the most straightforward one, it counts the number of times a token shows up in the document and uses this value as its weight. TF-IDF stands for "term frequency-inverse document frequency", meaning the weight assigned to each token not only depends on its frequency in a document but also how recurrent that term is in the entire corpora [9].

N-gram models are widely used in statistical natural language processing.For parsing, words are modeled such that each n-gram is composed of n words. In practice, n-gram models have been shown

to be extremely effective in modeling language data, which is a core component in modern statistical language applications [10].

As per our analysis TFIDF Vectoriser provided better accuracy than Count Vectoriser and Word2Vec. We fine tuned the vectoriser by parameterizing our python function to extract trigrams before vectorising and also provided max column size to 15000 to extract only the top 15000 important trigrams from the entire corpus.

Our final dataset thus had features extracted both containing behavioral & text engineering. Following snippet is an example of our final dataset used for data modelling.

User_Id	Prod_Id	Date_x	Review	Rating	Label	Product_Name	Avg_Prod_Rating	Avg_user_rating	Review_Len	user_total_reviews	corpus	compound	neg	neu	pos	number_Cap_Words	number_digit_Words	noun_count
923	0	08-12-14	The food at snack is a selection of popular Gr...	3	-1	Snack	4.009524	4.435897	215	39	food snack select popular greek dish appet tra...	0.6486	0.06	0.69	0.25	0	0	3
923	19	14-01-14	The restaurant is on the ground floor of a typ...	5	-1	Palo Santo	4.037152	4.435897	513	39	littl place soho wonder lamb sandwich glass wi...	-0.128	0.19	0.67	0.14	0	0	9
923	40	30-05-14	Really nice mousaka and lovely décor inside. A...	4	-1	Pylos	4.312869	4.435897	231	39	order lunch snack last friday time noth miss f...	0.7717	0.07	0.63	0.31	0	1	4

Fig. 6.1: Dataset After Feature Extraction

Apart from the above feature engineering, we had extracted another behavioral feature ie. Count of user Fake reviews. This specific feature added a huge bias to the model taking our accuracy to above 95%. Thus we learned that feature extraction should not be done considering the target column. Hence this feature was dropped.

We tested our model in 3 ways, solely based on behavioural features, solely based on text features & combination of both. As expected the combination gave us better results than the prior and thus was in sync with our understanding of text engineering and literature reviews.

VII. Handling Unbalanced data

We tried to apply the models to the preprocessed dataset and it was able to predict the dominating class only. As the data was unbalanced (10% of fake reviews), the model was not able to detect the fake review. In the ideal scenario, both the classes should have an equal number of datasets. There is a various method to overcome the situation of unbalanced data such as Random Under-sampling, Random Over-sampling, Synthetic Minority Over-sampling Technique (SMOTE), etc. The under-sampling technique may have reduced the dataset to 20% which would cause a huge loss in the dataset. So we have opted the below two techniques to overcome the situation.

1. Random Over-sampling:

It involves supplementing the training data with the multiple copies of minority class data until the dataset gets balanced. Below picture illustrate both, undersampling and oversampling.

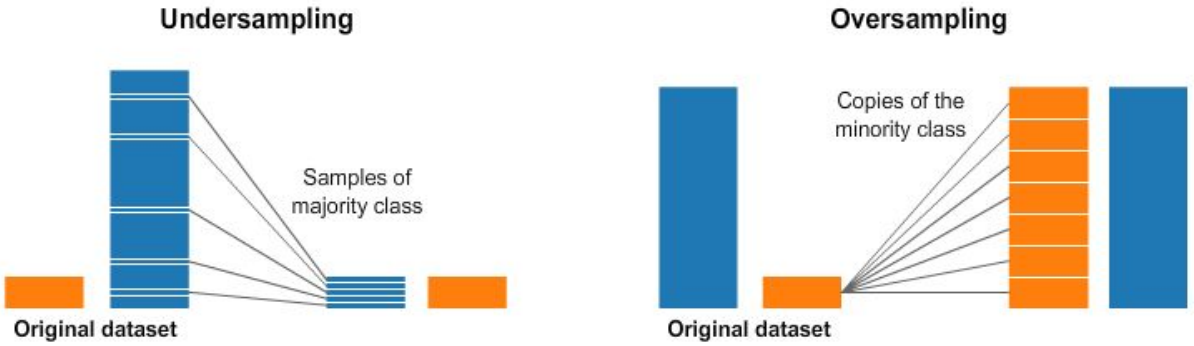


Fig. 7.1: Sampling Techniques

2. Synthetic Minority Over-sampling Technique (SMOTE):

Instead of just creating the copy of the minority class dataset, this technique synthesizes the minority class data and creates the feature nearby to it (like K nearest neighbor approach).

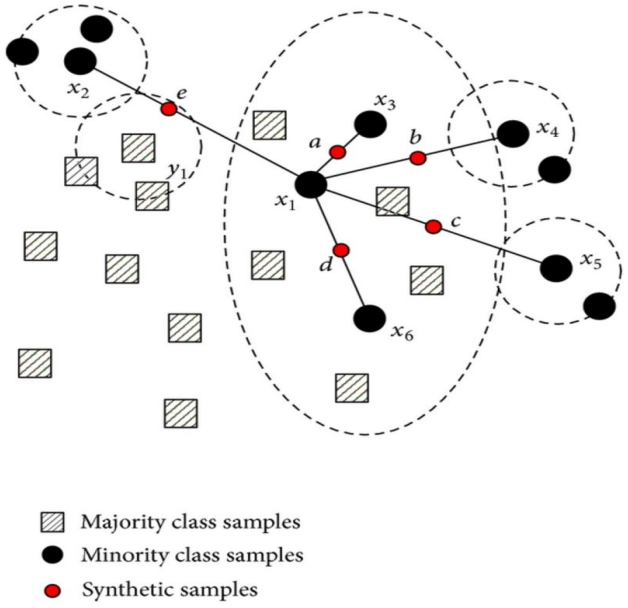


Fig. 7.2: SMOTE Illustration

VIII. Classification Methods & Comparison

Based on the Literature review and our understanding of classification techniques, we shortlisted the below models for our analysis.

1. Logistic Regression

This model apprehends a vector of variables and evaluates coefficients or weights for each input variable and then predicts the class of stated review. Looking mathematically logistics regression function estimates a multiple linear function which is defined as:

$$logit(S)=b0+b1M1+b2M2+b3M3\ldots bkMk$$

where S is the probability of presence of feature of interest.
M1,M2...Mk - predictor value & b0,b1….bk - intercept of the model[11]

2. **Naïve Bayes classifier**

   As discussed in [11], Naïve Bayes classifier is a probabilistic classifier which refers to Bayes Theorem. It is based on conditional probabilities with independent assumptions between its features. Mathematically defined as:

$$\mathrm{P}(\mathrm{X}|E_1,\ldots,E_n) = \frac{\mathrm{P}(E_1,\cdots,E_n|X)P(X)}{\mathrm{P}(E_1,\ldots,E_n)} \ldots\ \ldots\ \ldots$$

   Where X is the probability of an event, E is the given evidence

$P(E_1 \cdots En|X)$ Likelihood

$P(X)$ Prior

$P(E_1 \cdots E_n)$ Normalization constant

3. **K-Nearest Neighbors**

   The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. KNN captures the idea of similarity (sometimes called distance, proximity, or closeness). It is considered as a general rule of thumb to consider the number of neighbors as the square root of total rows in the data set for better results.[12]

4. **Random Forest Classifier**

   The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees. Higher number of estimator value gives a better result in the model.

5. **Convolutional Neural Network (CNN) — Deep Learning**

   CNN is a class of deep, feed-forward artificial neural networks (where connections between nodes do not form a cycle) & use a variation of multilayer perceptrons designed to require minimal preprocessing. CNNs are generally used in computer vision, however they've recently been applied to various NLP tasks and the results were promising. Hence, we tried applying CNN as one of the complex models along with the basic models implemented initially and it performed better than Naive Bayes.

Implementation

1. Implementation done using python packages on Google Collaboratory and Kaggle kernels.
2. For training our models, we have used Scikit-learn which is a free software machine learning library for the Python programming language.

Evaluation

Below are the results for performance of the models with different oversampling techniques.
Formulae for evaluation:

$$\text{Precision} = \frac{\text{True Positive}}{\text{Total Predicted Positive}} \qquad \text{Recall} = \frac{\text{True Positive}}{\text{Total Actual Positive}}$$

As the main aim for our project is to identify the false reviews from the data, we have focussed more on getting high value for recall parameter while evaluation of models.

Model	Oversampling Technique	Accuracy	Precision	Recall
Logistic Regression	Random Oversampling	60.56%	0.18	**0.76**
	SMOTE	60.32%	0.17	0.73
Naïve Bayes classifier	Random Oversampling	51.62%	0.85	0.16
	SMOTE	49.77%	0.88	0.16
K-Nearest Neighbors Algorithm	Random Oversampling	63.86%	0.19	**0.75**
	SMOTE	60.35%	0.18	**0.80**
Random Forest Regressor	Random Oversampling	89.46%	0.50	0.20
	SMOTE	88.76%	0.47	0.21
CNN	Random Oversampling	79.81%	0.20	0.31
CNN - LSTM	Random Oversampling	21%	0.14	**0.82**

Inferences

1. We found that the performance of Naive Bayes classifier was poor than the other two models. This is due to the excessive feature components which were added to the data as a part of Vectorization. We learned that Naive Bayes does not work well for huge number of features in data.

   Also, naive bayes predicted very less reviews as false. Out of these, most were correctly predicted. This explains why the precision value is high. However, as very few were predicted from the actual false reviews, the recall value is quite low.

2. Logistic Regression and KNN models performed well as per our requirement as they gave good results for Recall. This means both these models were correctly able to identify most of the false reviews.

References

1. Streitfeld, D. 2012b. Buy Reviews on Yelp, Get Black Mark. New York Times. http://www.nytimes.com/2012/10/18/technology/yelp-tries-to-halt-deceptive-reviews.html.
2. Mukherjee, A., Venkataraman, V., Liu, B. and Glance, N., 2013, June. What yelp fake review filter might be doing?. In *Seventh international AAAI conference on weblogs and social media*.
3. Shebuti Rayana and Leman Akoglu. Collective opinion spam detection:bridging review networks and metadata. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 985–994, 2015.
4. Wang, Z., Zhang, Y. and Qian, T., Fake Review Detection on Yelp.
5. Rout, J.K., Singh, S., Jena, S.K. and B akshi, S., 2017. Deceptive review detection using labeled and unlabeled data. *Multimedia Tools and Applications*, *76*(3), pp.3187-3211.
6. Chowdhary, N.S. and Pandit, A.A., 2018. Fake Review Detection using Classification. *International Journal of Computer Applications*, *180*(50), pp.16-21.
7. https://towardsdatascience.com/sentiment-analysis-with-python-part-1-5ce197074184
8. https://www.oreilly.com/library/view/applied-text-analysis/9781491963036/ch04.html
9. https://towardsdatascience.com/hacking-scikit-learns-vectorizers-9ef26a7170af
10. https://en.wikipedia.org/wiki/N-gram
11. Prabhat, A. and Khullar, V., 2017, January. Sentiment classification on big data using Naïve bayes and logistic regression. In *2017 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-5). IEEE.
12. https://towardsdatascience.com/knn-k-nearest-neighbors-1-a4707b24bd1d