SJSU SAN JOSÉ STATE UNIVERSITY

Uncovering Fake spam review on Yelp using Data Mining Techniques

Data Mining - CMPE 255 - Spring 2020

Submitted to: Prof. Gheorghi Guzun

Submitted By -

Hansraj Pabbati (SJSU ID : 012540541)

Rishika Machina (SJSU ID: 012525227)

Abstract

Online reviews play a very important role for decision-making in today's e-commerce. Large parts of the population, i.e. customers read product or store reviews before deciding what to buy or where to buy and whether to buy or not. Because writing fake/ fraudulent reviews comes with monetary gain, online review websites there has been a huge increase in tricky opinion spam. Basically, an untruthful review is a fake review or fraudulent review or opinion spam. Positive reviews of a target object can attract more customers and increase sales; negative reviews of a target object can result in lower demand and lower sales. Fake review detection has attracted considerable attention in recent years. Most review sites, however, still do not filter fake reviews publicly. Yelp is an exception that over the past few years has filtered reviews. Yelp's algorithm, however, is a business secret. In this work, by analyzing their filtered reviews, we try to find out what Yelp could do. The results will be useful in their filtering effort for other review hosting sites. Filtering has two main approaches: supervised and unmonitored learning. There are also about two types in terms of the characteristics used: linguistic characteristics and behavioral characteristics. Through a supervised learning approach we have tried to make a model which can identify the fake review with almost 70 percent accuracy.

Introduction

As the web continues to expand in size and value, the quantity and effect of online feedback is increasingly growing. Reviews can influence consumers across a wide range of industries but they are particularly relevant in e-commerce, where feedback and reviews on goods and services are often the most convenient, if not the only, way for a customer to decide whether to purchase them,it can produce online feedback for a variety of reasons.In order to enhance and develop their companies, online retailers and service suppliers also ask consumers to provide input about their experience with goods or services they have purchased. Whether you have particularly good or bad experience, your clients might often feel compelled to evaluate a product or service. While online reviews might be helpful, blind trust in these reviews is risky to both the seller and the

purchaser. Some of those take a gander at online reviews before placing an online order, however, reviews might have been poisoned or faked for profit or gain, so any decision based on random reviews must be made with prudence.

In addition, corporate owners may reward someone who reviews their goods well or may pay anyone who publishes misconstrued reviews of their competitor's products or services. Such fake reviews are seen as a spam review and can have a significant impact on the online market because of the value of feedback. Someone publishes unfavorable reviews of the products or services of their rivals. Since spam review is an all-embracing and harmful problem, designing methods that help companies and consumers discern true reviews from fake ones is significant but challenging.

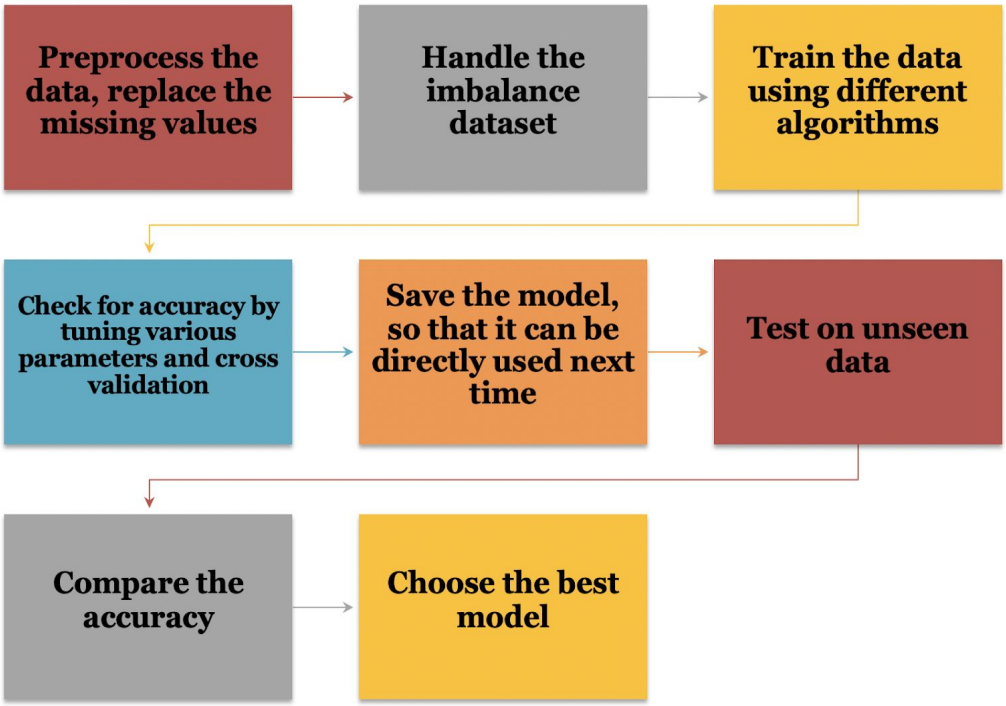
Dataset

We requested the Yelp Dataset from Dr. Bing Yu from UIC, for the Chicago area which has the records for 85 hotels and 130 restaurants. This is a labeled dataset with 688328 entities. This is the imbalanced dataset. Earlier studies used unlabeled data sets and found duplicate reviews or near-duplicate reviews to be spam. This is too narrow an interpretation of spam and does not identify opinion spam or fake reviews. In some previous work, manual labeling has also been done, but this methodology is misreckoning, so we aim to identify and flag the fake/spam review by using proper Text mining and Data mining techniques.

	date	reviewID	reviewerID	reviewContent	rating	usefulCount	coolCount	funnyCount	flagged	hotelID
0	6/8/2011	MyNjnxzZVTPq	IFTr6_6NI4CgCVavIL9k5g	Let me begin by saying that there are two kind...	5	18	11	28	N	tQfLGoolUMu2J0igcWcoZg
1	8/30/2011	BdD7fsPqHQL73hwENEDT-Q	c_-hF15XgNhlyy_TqzmdaA	The only place inside the Loop that you can st...	3	0	3	4	N	tQfLGoolUMu2J0igcWcoZg
2	6/26/2009	BfhqiyfC	CiwZ6S5ZizAFL5gypf8tLA	I have walked by the Tokyo Hotel countless tim...	5	12	14	23	N	tQfLGoolUMu2J0igcWcoZg
3	9/16/2010	OI	nf3q2h-kSQoZK2jBY92FOg	If you are considering staying here, watch thi...	1	8	2	6	N	tQfLGoolUMu2J0igcWcoZg
4	2/5/2010	i4HIAcNTjabdpG1K4F5Q2g	Sb3DJGdZ4Rg__CqxPbae-g	This place is disgusting, absolutely horrible,...	3	11	4	9	N	tQfLGoolUMu2J0igcWcoZg

Snapshot of the actual Dataset

System Design



Implementation

The most demanding part of this project is that predictive features and details are derived from feedback. We focused mainly on the following core features for this specific task -

1) **Semantic Features** - Review polarity plays an instrumental role in determining the actual feeling of the customer. *Textblob* has been used for the same to find out the polarity and sentiment for the review content.
2) **Structural Features** - For our project it includes review length, average word length, and standard deviation of the review length.
3) **n-gram features**: We have extracted unigram, Bigram, Trigram features from review content.

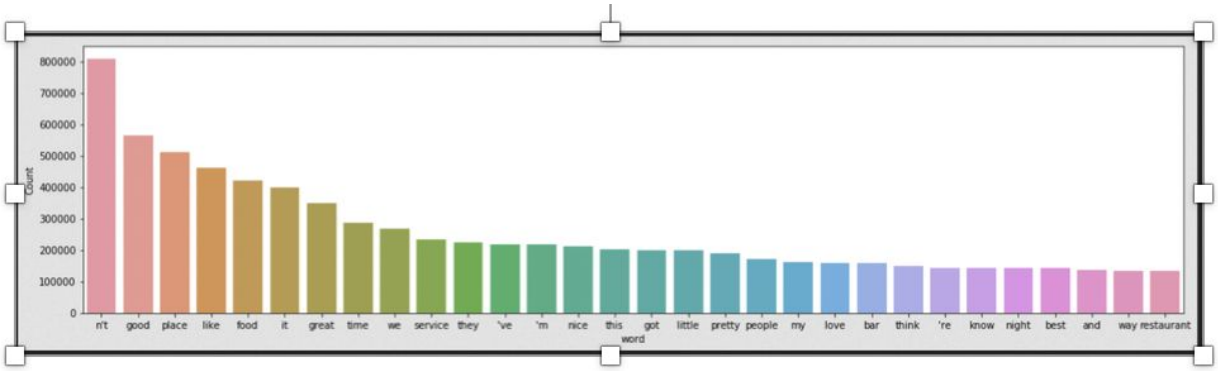
Data Preprocessing

Following were the steps taken for Data Preprocessing

1. Dividing the data into Train and Test data in 80:20 ratio before applying any preprocessing steps.
2. Removing Regex, URL's, Punctuations and special characters.
3. Converting all the text into lower case
4. Removal of all the Stopwords in order to reduce the space and time complexity.
5. Tokenization of Data, which has been done using Spacy in case of machine learning algorithms and Keras in Deep Learning Approaches.
6. Used TF-IDF vectorizer from sklearn directly to convert text data into CSR Matrix, by normalizing the input.
7. Removal of noise and most frequent words which were interfering in the prediction.

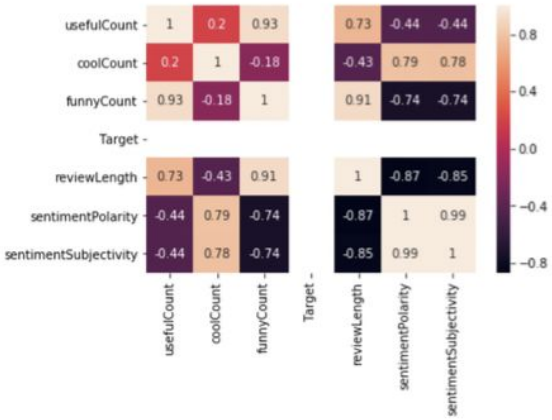
Feature Engineering

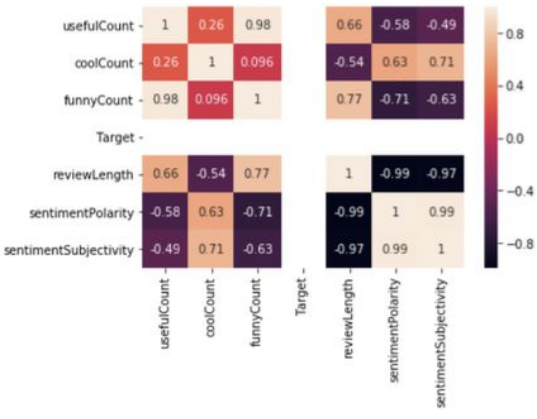
1) We added the review length feature using the feedback. In order to detect feelings of [-1,+1] and emotional subjectivity[0,1] for each review, we have used a pretrained textblob model. In addition, we used other characteristics, such as the ranking, which is useful for training models on other features than the analysis.
2) We also plotted word cloud for Spam and Non Spam reviews individually to analyse which terms occurs frequently in both type of reviews
3) We also generated a Heat Map to see if some of the attributes are highly correlated, in case that happens then we would have to remove one of the attributes from the actual prediction model.
4) Also we used grid search to find the best Hyperparameters for the Machine learning models used.
5) We used 5 fold cross validation in case of Deep Learning approaches.



Normal reviews

Fake reviews





Algorithms(Models) Used -

We have explored several machine learning and deep learning algorithms combined with Hyperparameter tuning in order to find the best model with maximum precision.

Below mentioned are the details for the same - :

1) **Multinomial Naive Bayes**: This is a generative model, in which a Bernoulli distribution produces the Label. We initially trained the model on 'CleanedToken' which was the attribute generated after text preprocessing of reviewContent.It gave nearly 67 per cent accuracy, when applied normally without any major hyperparameter tuning.

   Then we used Grid to find the best parameters and after using the same accuracy improved substantially upto 72 percent. We also used 3 fold cross validation along with other things.

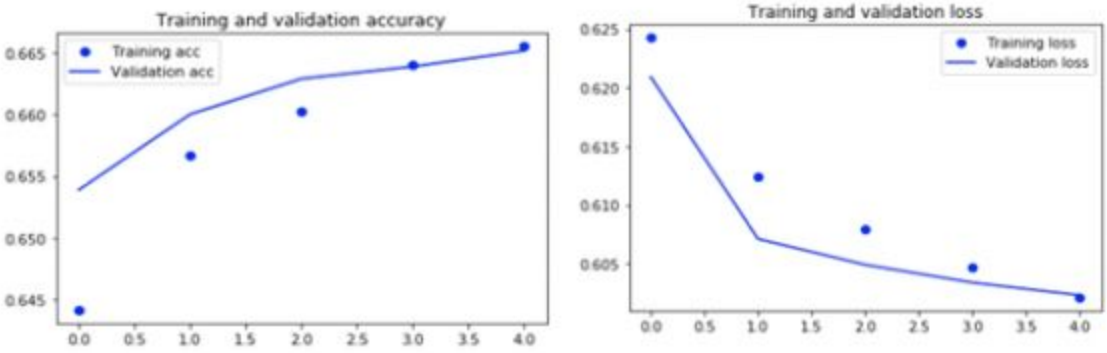
```
parameters = {'vect__ngram_range': [(1, 1), (1, 2)], 'tfidf__use_idf': (True, False), 'clf__alpha': (1e-2, 1e-3)}
gs_clf = GridSearchCV(text_clf, parameters, n_jobs=-1)
gs_clf = gs_clf.fit(df_X, df_y)
```

2) **XGBoost :** XGBoost is an implementation with a focus on speed and efficiency of gradient-boosted decision trees with L2 regularization. XGBoost used in 2 different ways where we got 57% precision once on non-text features like polarity , ranking, reviewlength, etc. Then XGBoost was implemented on CSR IDF matrix using text info, and we achieved better results with 64% accuracy.

3) **Neural Network with LSTM**: This model had an embedding layer, an LSTM layer, and the output of the LSTM served as input into a fully connected layer that was hidden. This layer's output size is 1, meaning it will either generate 1 or 0, 1 for Spam reviews, and 0 for legitimate reviews. 10,000 most common words were forwarded on to the embedding layer as attributes, 64 as the second parameter, and 200 as the input length of each sequence. First argument comes from the Embedding layer's second argument. We held dropout chances of 20 per cent. Training model with a batch size of 128, 5 epochs and split=0.2 validation, meaning the neural network will learn from 80% of the data and check itself on

the remaining 20% of the data.

```
Train on 440530 samples, validate on 110133 samples
Epoch 1/5
440530/440530 [==============================] - 681s 2ms/step - loss: 0.6242 - acc: 0.6442 - val_loss: 0.6209 - val_acc: 0.6539
Epoch 2/5
440530/440530 [==============================] - 669s 2ms/step - loss: 0.6124 - acc: 0.6566 - val_loss: 0.6071 - val_acc: 0.6600
Epoch 3/5
440530/440530 [==============================] - 665s 2ms/step - loss: 0.6080 - acc: 0.6603 - val_loss: 0.6049 - val_acc: 0.6629
Epoch 4/5
440530/440530 [==============================] - 662s 2ms/step - loss: 0.6047 - acc: 0.6640 - val_loss: 0.6034 - val_acc: 0.6639
Epoch 5/5
440530/440530 [==============================] - 660s 1ms/step - loss: 0.6021 - acc: 0.6655 - val_loss: 0.6023 - val_acc: 0.6651
CPU times: user 1h 16min 23s, sys: 5min 51s, total: 1h 22min 15s
Wall time: 55min 39s
```



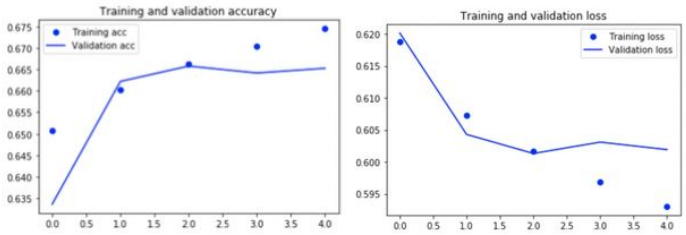
In the early three epochs, the training precision was lower than the validation accuracy, which indicates that the model was underfitting ,but the model generalized well and provided the same training and validation precision in the last 2 iterations. The accuracy on the test set was 66.26 percent.

4) **Neural network with an extra 1D CNN on top of LSTM layer:** The recurrent neural network concocting LSTM and convolutional neural networks is used for this model. For layer tuning an additional dropout layer was added after the embedding layer and max pooling layer.

The similar accuracy was seen in this model at 66.45 percent and overfitting after one iteration. Due to the additional convolutional layer, however it performed better in terms of speed.

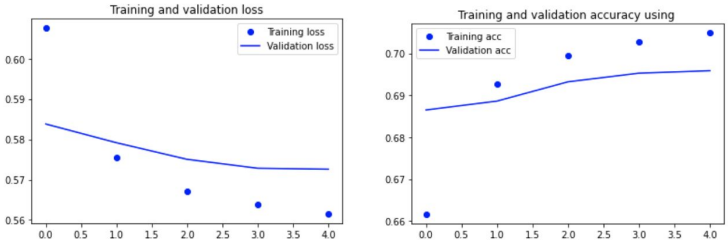
5) **Neural network with using bidirectional recurrent layers:** For this model we have used a bidirectional recurrent layer, as also used in the natural language processing approaches. The recurring two-way layer can provide more accurate data representations and patterns. Immediately after embedding layer, bidirectional recurrent layer was mounted, creating a second separate layer instance and using one for chronological and another for reversed processing.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 100, 128)          1280000
_________________________________________________________________
bidirectional_1 (Bidirection (None, 256)               263168
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 257
=================================================================
Total params: 1,543,425
Trainable params: 1,543,425
Non-trainable params: 0
_________________________________________________________________
```



F-Score of 66.45 percent was achieved with this approach.

6)**Neural network using Recurrent Neural Network:** This model has been trained by using a padded sequence and 5% of training data as a validation sample. We also had to adjust the weights for the recurrent system to provide the optimal output for an input series. It is handled in Tensorflow implicitly.



Overview of Results

	Classifier	F1 Score
1	Naive Bayes Classifier	67.00 %
2	Naive Bayes Classifier (With Grid search and ngram)	68.00 %
3	Neural Network with LSTM	66.26 %
4	Neural Network with 1D CNN on top of LSTM layer	66.39 %
5	Neural Network using Bidirectional recurrent layer	66.45 %
6	Recurrent Neural Network with GRU	70.00 %
7	XGBoost	67 % (Text), 61 %(Other)

Difficulties Faced

1) Due to the extremely large dataset , I came across memory leak issues several times.
2) Some models required a lot of dependencies and libraries which sometimes became difficult to figure out, and took a lot of time.
3) Tried Support Vector Machine on Text data but after running for 48 hrs also, it didn't converge.
4) When tried aggregating Textual and Non textual attributes using Stochastic gradient descent by assigning weight to features, it gave really poor results.

Conclusion

Contrary to our belief, it does not considerably improve model efficiency by incorporating reviewer-centered functionality. We assume that this is because most

reviewers only have a summary record that is not completely described (i.e. maximum number of reviews per day, standard rating deviation). At times the simple approach works best, and this project gave us nearly 77% of the accuracy, which was the highest among the other classifications, through Grid Search Cross validation with the Multinomial Naive Bayes Classifier and the best parameter.

Task Assigned

Hansraj Pabbati - Features addition, Data Preprocessing, Model Selection and Implementation and Report.

Rishika Machina - Features addition, Data Preprocessing, Model Selection and Implementation and Presentation.

References

- Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie Glance. What Yelp Fake Review Filter Might Be Doing. Proceedings of The International AAAI Conference on Weblogs and Social Media (ICWSM-2013), July 8-10, 2013, Boston, USA.
- M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, H. A. Najada, "Survey of review spam detection using machine learning techniques", Journal of Big Data, vol. 2, no. 23, Dec 2015.
- https://scikit-learn.org/stable/supervised_learning.html#supervised-learning