Online Review Spam Detection - Final Report

Group-2 : Tirth Patel, Harsh Kachhadia, Piyush Biraje
Department of Computer Science, North Carolina State University, Raleigh, NC
tdpatel2@ncsu.edu, hmkachha@ncsu.edu, pbiraje@ncsu.edu

1 Background

1.1 Problem

Today, majority of the companies have become consumer focused. Customer experience is directly related to company's success. Customers are directed towards those products where they have good experience. Take the example of Amazon. People prefer to shop via Amazon because Amazon prioritizes customer experience. They do so by providing other customer reviews on products. This help people make good decisions. But lately we have seen an increase in fake reviews on many websites like Amazon, TripAdvisor, Yelp, etc.

It is this problem that many companies have tried to solve in order to give their customers correct information about the product. But its not easy to say which review is fake and which isn't. Machine Learning has helped to detect fake reviews with a certain degree of certainty.

In this project, we plan to come up with a model which will be able to predict if a given review is fake or not.

1.2 Literature Survey

Mukherjee et al. used supervised learning for fake review detection using real as well as Amazon Mechanical Turk (AMT) generated fake reviews from TripAdvisor and Yelp. The results showed that the classification of filtered data is much easier much to classifying real-life data, with about 67% accuracy. The paper concludes that the models trained using AMT fake reviews are weak in detecting real fake reviews, which indicates that the AMT fake reviews are probably not representative of the real life fake reviews.

"Opinion Spam and Analysis" by Nitin Jindal and Bing Liu first identifies 3 types of spams in reviews, using review data collected from Amazon.com. The results show that logistic regression is very effective in detecting type 2 and type 3 as defined in the paper.
Link for the paper- https://www.cs.uic.edu/~liub/FBS/opinion-spam-WSDM-08.pdf

Olmsted et al. tokenized the datasets, extracted adjectives, adverbs, and verbs and tagged words as real or fake. Multinomial Naïve Bayes, Bernoulli Naïve Bayes, and logistic regression models were used to classify reviews. This paper extracted part-of-speed features and applied the above classification models and concludes that the Multinomial Naïve Bayes classification model achieves the highest accuracy.

2 Method

For the first part of this project, our team plan to use a hotel review dataset obtained from Kaggle and create different Machine Learning models and then come up with a best model. We plan to create models based on Multinomial Naïve Bayes, Logistic Regression, SVM, SGR, Random Forest, etc.

For the second part of the project, we plan to create a web-scrapper and make our dataset of online reviews. After creating the dataset, we plan to apply the model selected from part one of the project on this dataset.

3 Experimetal Setup

In order to fulfill the first phase of our project which is, selecting the best supervised machine learning algorithm for review/opinion classification into deceptive or truthful, we utilized the following data, algorithms, and model evaluation methods:

3.1 Data:

`https://www.kaggle.com/rtatman/deceptive-opinion-spam-corpus?select=deceptive-opinion.csv`

Data Exploration: This dataset contains hotel reviews from 20 chicago hotels.

Our dataset has 1600 records - 800 labelled as truthful and 800 as deceptive reviews.
Data that we will be using - "Text" and "deceptive" columns will be used for our project.

Dataset has the following columns:
1. Deceptive - class label - "deceptive" or "truthful"
2. Hotel - describes hotel for which review is given
3. Polarity - positive or negative
4. Source - the website on which the hotel review was posted
5. Text - the review text

- 400 Truthful positive polarity reviews from TripAdvisor.
- 400 Truthful negative polarity reviews from other websites like expedia, yelp, etc.
- 800 Deceptive (both positive and negative polarity) reviews from Mechanical Turk

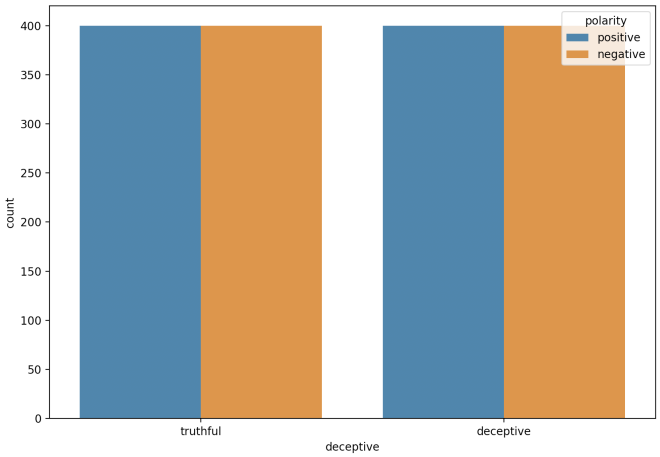

Figure 1: Data Distribution Table

Input Data Preprocessing: The "Text" column has been preprocessed with NLTK Library and Regular expressions in python. Following text-preprocessing has been done.

1. Convert text to lowercase
2. Remove - punctuation, text in square brackets, non-alphanumeric characters, hyperlinks, special

characters, new line characters, digits, extra space between words, front or trailing spaces in text
3. Removal of stopwords from the text.

Sample text with Stop Words	Without Stop Words
GeeksforGeeks – A Computer Science Portal for Geeks	GeeksforGeeks , Computer Science, Portal ,Geeks
Can listening be exhausting?	Listening, Exhausting
I like reading, so I read	Like, Reading, read

Figure 2: Stopwords Removal

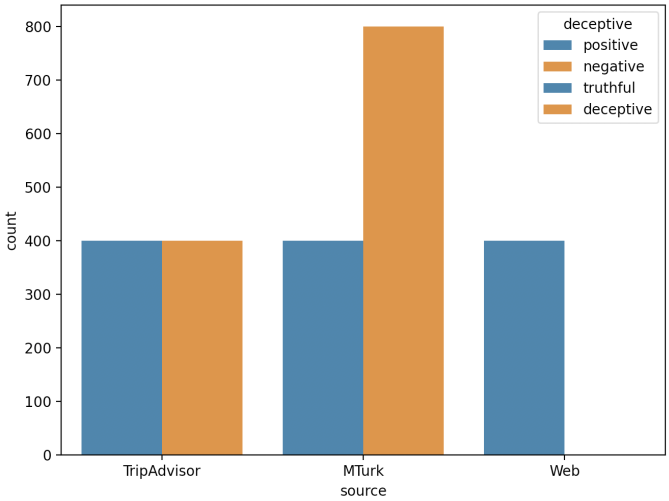

Figure 3: Data Distribution Table

Output Label Preprocessing: Before proceeding with implementing our algorithms, we have encoded our output class labels (deceptive or truthful) to numeric values with the help of sklearn Label Encoder.

Model Evaluation Methods implemented:

1. Accuracy
2. K-Fold CV - 5-CV, 8-CV, 10-CV
3. Holdout Method (train,test, split) (0.7-train, 0.3 test)

3.2 Hypothesis

- Main questions : How well can we identify whether a comment is fake or not. And to do so, which classifier works best. - We are expecting to find some patterns in the dataset using which we can predict whether a comment is fake or not.

3.3 Algorithms Implemented for experimentation:

1. Multinomial Naive Bayes Classifier

After performing data preprocessing on the "Text" column of the data, which contains reviews about hotels, we implemented a pipeline which performs 2 tasks: 1) Convert the pre-processed text in

the "Text" column to TF-IDF feature vectors. 2) Train the sklearn provided Multinomial Naive Bayes Classifier(default hyperparameters) on the TF-IDF feature vectors. This pipeline was also subsequently used for prediction on the test data.

This pipeline was utilized to try out different models of Multinomial Naive Bayes Classifier trained on different datasets obtained by performing the following model evaluation techniques on our preprocessed data.

K-Fold Cross Validation (K=5,8,10), Holdout Method (TTS) (0.7-train, 0.3 test).....(Eq. 1)

The accuracy score of each of these models has been recorded in the results table in the results section.

2. Stochastic Gradient Descent

After experimenting with different models of Multinomial Naive Bayes Classifier, we implemented another pipeline which performs 2 tasks: 1) Convert the pre-processed text in the "Text" column to TF-IDF feature vectors. 2) Train the sklearn provided SGD(stochastic gradient descent) Classifier with the following hyperparameters on the TF-IDF feature vectors.

```
SGDClassifier(loss='hinge', penalty='l2',alpha=1e-3, random_state=42, max_iter=5
,tol=None)
```

This pipeline was also subsequently used for prediction on the test data.

Similar to Multinomial Naive Bayes, this pipeline was also utilized to try out different models of SGD(stochastic gradient descent) Classifier trained on different datasets obtained by performing the model evaluation techniques in (Eq.1) on our preprocessed data. The accuracy score of each of these models has been recorded in the results table in the results section.

3. Logistic regression

After experimenting with different models of SGD(stochastic gradient descent) Classifier, we implemented another pipeline which performs 2 tasks: 1) Convert the pre-processed text in the "Text" column to TF-IDF feature vectors. 2) Train the sklearn provided Logistic Regression Classifier with the following hyperparameters on the TF-IDF feature vectors.

```
LogisticRegression(n_jobs=1, C=1e5)
```

This pipeline was also subsequently used for prediction on the test data.

Similar to Multinomial Naive Bayes, this pipeline was also utilized to try out different models of Logistic Regression Classifier trained on different datasets obtained by performing the model evaluation techniques in (Eq.1) on our preprocessed data. The accuracy score of each of these models has been recorded in the results table in the results section.

4. SVC (Linear SVM) Classifier

After experimenting with different models of Logistic Regression Classifier, we implemented another pipeline which performs 2 tasks: 1) Convert the pre-processed text in the "Text" column to TF-IDF feature vectors. 2) Train the sklearn provided SVC (Linear SVM) Classifier with the following hyperparameters on the TF-IDF feature vectors.

```
LinearSVC(loss='hinge', C=5, random_state=42)
```

This pipeline was also subsequently used for prediction on the test data.

Similar to Multinomial Naive Bayes, this pipeline was also utilized to try out different models of SVC (Linear SVM) Classifier trained on different datasets obtained by performing the model evaluation

techniques in (Eq.1) on our preprocessed data. The accuracy score of each of these models has been recorded in the results table in the results section.

5. Random Forest Classifier

After experimenting with different models of SVC (Linear SVM) Classifier, we implemented another pipeline which performs 2 tasks: 1) Convert the pre-processed text in the "Text" column to TF-IDF feature vectors. 2) Train the sklearn provided Random Forest Classifier with default hyperparameters on the TF-IDF feature vectors. This pipeline was also subsequently used for prediction on the test data.

Similar to Multinomial Naive Bayes, this pipeline was also utilized to try out different models of SVC (Linear SVM) Classifier trained on different datasets obtained by performing the model evaluation techniques in (Eq.1) on our preprocessed data. The accuracy score of each of this models has been recorded in the results table in the results section.

3.4 New Dataset Generation:

For the task of generating a new hotel reviews dataset, we scraped hotel reviews from popular hotel booking websites and review websites using web scraper built in Python language.

Targeted City: Las Vegas

We chose the city of Las Vegas as our targeted city whose hotels will be chosen for this dataset, as it has some of the most popular tourist attractions in the whole US. This leads to an active hotel industry in the city which makes this project more relevant for hotels in Las Vegas.

The hotels we targeted for this dataset are the top 20 hotels as per USNews 2020. The hotels name are mentioned below:

1. ARIA Resort Casino
2. Four Seasons Hotel Last Vegas
3. SKYLOFTS at MGM Grand
4. Encore at Wynn Las Vegas
5. Wynn Las Vegas
6. The Venetian Las Vegas
7. The Cosmopolitan of Las Vegas
8. The Palazzo Las Vegas
9. Bellagio Resort Casino
10. ARIA Sky Suites
11. Caesars Palace
12. NoMad Las Vegas
13. Red Rock Casino, Resort Spa
14. Vdara Hotel Spa at ARIA Las Vegas
15. Trump International Hotel Las Vegas
16. M Resort Spa Casino Las Vegas
17. Mandalay Bay Resort and Casino
18. Waldorf Astoria Las Vegas
19. Delano Las Vegas
20. Nobu Hotel at Caesars Palace

The reason for choosing the top 20 hotels and not other ones is, that, as per statistics, most popular hotels are the ones most targeted by such spam/deceptive reviews for either ameliorating or deteriorating the name of hotel.

The websites from which the reviews of Las Vegas hotels were scraped are as follows:

- **Trip Advisor**
- **Expedia**
- **Yelp**

Reason for selecting them:

Trip Advisor and **Expedia** are some of the most popular trusted sites for hotel reservation, which basically means that, these are the places where people will first look for hotel reviews.

Yelp is one of the most popular business review site where people write reviews about various places including hotels. A 2017 study shows that Yelp brings higher visitor conversions than Google and Facebook. Roughly 92% of visitors said they made a transaction after visiting the site. Thus, it proves meaningful to add this site to our sources.

Now, for scraping reviews for targeted hotels from the above mentioned sources, we built 3 web scrapers (each targeting one source website) in Python with the help of below mentioned Python libraries.

1. **BeautifulSoup** - It was used to parse HTML documents and makes it easier to scrap websites.
2. **Pandas** - The Dataframe feature of Pandas was used for managing large amounts of data in a tabular format.
3. **Sklearn** - This python library was used for text feature extraction.

With the help of these 3 web scrapers, we collected 20 reviews for each hotel from each source, totalling to 1200 reviews in all. Along with reviews, hotel name and the source website was also recorded along with reviews.

The text in this collected data was then pre-processed with the help of NLTK Library. After getting preprocessed, the data was converted to a large dataframe in pandas, which will make future handling and processing of data easier. The resultant dataframe had 3 columns - Hotel, Source, Text (Review).

This data was now tagged with the help of Stochastic Gradient Descent(SGD) Classifier, one which was trained on the previously mentioned Opinion Review Spam Detection from Kaggle and was selected for best performance in classifying deceptive/truthful hotel reviews. This added a new column in our existing dataframe by name "Label".

We also added the column called "Polarity" to the dataframe, which indicated the nature of each review as Positive, Neutral or Negative. This was done with the help of TextBlob python library's sentiment analysis property.

The resultant dataframe had 5 columns - Hotel, Source, Text, Label, Polarity. This dataframe was later converted to a .csv file to save this dataset.

The final dataset had the following statistics:
Link to the dataset: `https://github.com/tirthpatel7498/Online-Review-Spam-Detection/blob/main/hotel_review_spam_detection.csv`

The dataset generated has 1200 rows comprising of 20 hotels and each hotel has 60 reviews.

After passing the data through our selected classfier, i.e Stochastic Gradient Descent classfier and labelling the polarity of reviews using TextBlob Sentiment Analyser, the following statistics were obtained:

1. Label count: Truthful- 815, Deceptive- 316
2. Polarity count: Negative- 1072, Positive- 116, Neutral- 12

4 Results

We have achieved our primary objective of creating another dataset and using the best classifier to tag this dataset and open the discussion for others to try and compare the results. And if possible, verify it(which is difficult).

The results from the implementation of various models of algorithms in the experimental setup section have been tabulated here:

	Model	TTS	CV-5	CV-8	CV-10
0	Multinomial Naive Bayes	0.854167	0.85875	0.864375	0.860625
1	SGD	0.885417	0.87625	0.883750	0.882500
2	Logistic Regression	0.885417	0.87250	0.874375	0.876875
3	SVC (Linear SVM)	0.877083	0.86750	0.874375	0.872500
4	Random Forest	0.829167	0.85125	0.841875	0.855625

Figure 4: Accuracy Table

5 Conclusion and Learnings

• Out of the 5 algorithms implemented so far, the stochastic gradient descent (SGD) algorithm has returned the best results with consistently highest accuracy(88.5%) across all model evaluation techniques.

• One of the major problems that we faced in this project is the lack of large enough data for spam detection.

• We observed that there are very few openly available corpus for hotel review that are labelled with truthful and deceptive.

• In this project, we have created deceptive opinion spam dataset, which will be helpful for future projects, as this will be openly available.

References

1. https://www.researchgate.net/publication/325075174_Detection_of_fake_online_hotel_reviews
2. https://arxiv.org/pdf/1903.12452.pdf
3. https://www.cs.uic.edu/~liub/FBS/opinion-spam-WSDM-08.pdf
4. A. Mukherjee, V. Venkataraman, B. Liu and N. Glance, "Fake Review Detection: Classification and Analysis of Real and Pseudo Reviews," Department of Computer Science (UIC-CS-2013-03), Chicago, 2013.
5. Sandifer, Anna Wilson, Casey Olmsted, Aspen. (2017). Detection of fake online hotel reviews. 501-502. 10.23919/ICITST.2017.8356460.
6. Ott, Myle, et al. Negative Deceptive Opinion Spam. Association for Computational Linguistics, 2013.

Link to GitHub Repository

https://github.com/tirthpatel7498/Online-Review-Spam-Detection