OPINION SPAM DETECTION

(Final Project Report)

Team 9
Siddharth Sheth (UW ID:shethsid, Student ID: 2076180)
Patrick Moy(UW ID:moyp, Student ID: 1841305)
Srivatsav Gopalakrishnan (UW ID: sgopal9, Student ID:2075414)

TCSS 555(Spring '21)
Professor: Dr. Athirai A. Irissappane

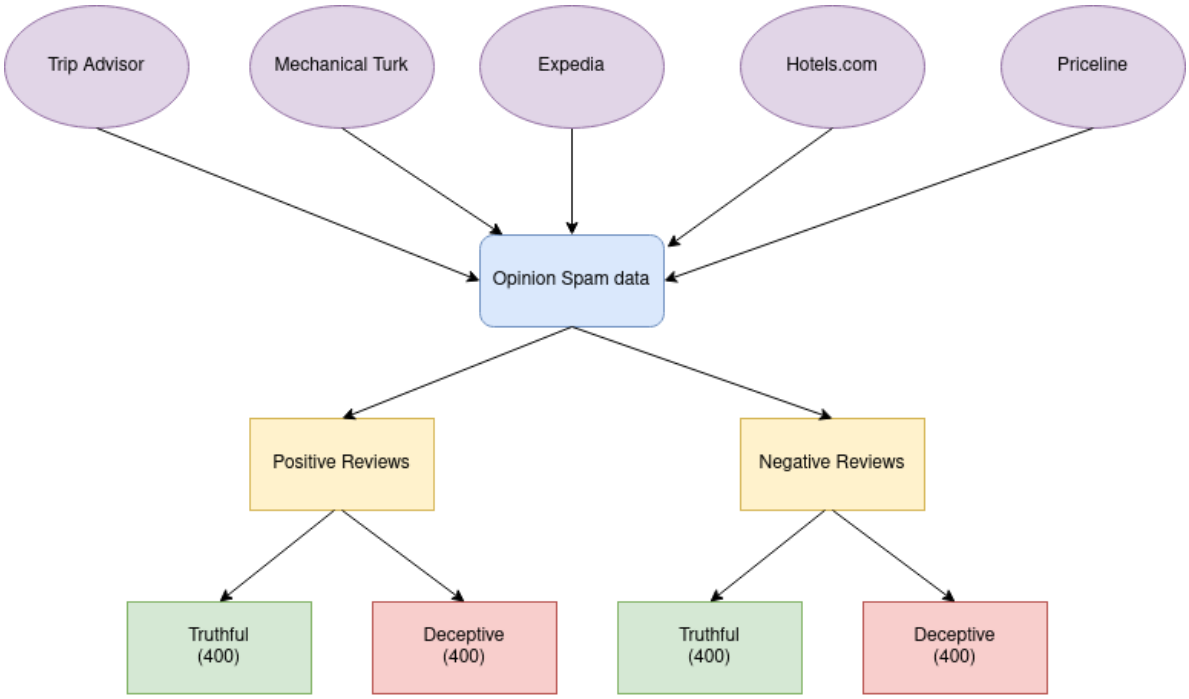
Outline

PART A	PART B
● Data Analysis ○ Overview ○ Preliminary Analysis ○ Review Length ○ Word Count ● Data Pre-Processing ○ Spell Checker ○ Lowercase ○ Stop words ○ Part of Speech Tagging ○ Polarity ○ Subjectivity ○ Feature importance ● Survey of Potential models ○ Survey of research papers ○ Support Vector Machines ○ Naive Bayes Model ○ Random Forest	● Train-Test Split ○ Train data and Test data distribution ○ Shuffle (to randomize data) ● Experimental Settings ○ Metrics (Accuracy Score, F-1 Score, Recall, Precision) ○ Train Time ○ Hardware used (with commands to find necessary settings) ● Training and Fine Tuning ● Evaluation of results ○ Train Results ○ Test Results ● Comparison of machine learning models ● Comparison with related research [*Extra Credit*] ○ Related works vs Team 9 Implementation (F-1 score) ● Findings and Conclusion
● Individual contributions (Effort Report)	
● References	

Data Analysis

Overview

In this dataset, we have a review of 20 hotels in chicago. The corpus has a total of 1600 reviews from various hotel booking sites. There are 800 positive and 800 negative reviews. Both the subsets have 400 truthful and deceptive reviews. Therefore the data is said to be balanced.



Preliminary Analysis

We can see that all the values are non-null (Figure 1). Therefore we don't need to manually remove reviews from this data set. We can also deduct that there are 1596 unique reviews from Figure 2. This means that there are 2 reviews that have the same content. One such duplicate review can be seen in Figure 3. We have not removed the duplicate reviews during preprocessing as it would disturb the balanced data we have currently.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1600 entries, 0 to 1599
Data columns (total 1 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   HotelReviews  1600 non-null   object
dtypes: object(1)
memory usage: 12.6+ KB
```

Figure 1: Details of Raw Data (Non-null values)

	HotelReviews
count	1600
unique	1596
top	my daughter and i woke in the morning wanting ...
freq	2

Figure 2: Details of Raw Data(Duplicate values)

```
negative_polarity/truthful_from_Web/fold3/t_omni_5.txt:1:My daughter and I woke in the morning wanting to go swimming. When we arrived at the pool t
he water was covered by a white scum. I then attempted to use both of the phones at the pool, one white phone and one emergency red phone, to call t
he desk. Both were out of service!!!! I am glad there wasn't an emergency. As we were exited the pool area I ran into a hotel employee and told her
about the problems and then asked her to call us when the pool was clean.... never heard back.
negative_polarity/truthful_from_Web/fold3/t_omni_7.txt:1:My daughter and I woke in the morning wanting to go swimming. When we arrived at the pool t
he water was covered by a white scum. I then attempted to use both of the phones at the pool, one white phone and one emergency red phone, to call t
he desk. Both were out of service!!!! I am glad there wasn't an emergency. As we were exited the pool area I ran into a hotel employee and told her
about the problems and then asked her to call us when the pool was clean.... never heard back. One more thing, in our room the furniture was mostly
child friendly except for a large metal and glass outdated coffee table. Within 30 minutes my 5 year old fell and hit her mouth and began bleeding.
They need to update this piece of furniture.
negative_polarity/truthful_from_Web/fold3/t_omni_9.txt:1:My daughter and I woke in the morning wanting to go swimming. When we arrived at the pool t
he water was covered by a white scum. I then attempted to use both of the phones at the pool, one white phone and one emergency red phone, to call t
he desk. Both were out of service!!!! I am glad there wasn't an emergency. As we were exited the pool area I ran into a hotel employee and told her
about the problems and then asked her to call us when the pool was clean.... never heard back.
```

Figure 3: Duplicate review sample

Review Length

We counted the number of characters in each review. We saw that most reviews have 300 to 1000 characters. Refer **Figure 4.** We then plotted similar histograms for deceptive and truthful reviews individually. Comparing **Figure 5** and **Figure 6,** we can see that there are more truthful reviews between the ranges of 0-200 words and greater than 500 words.

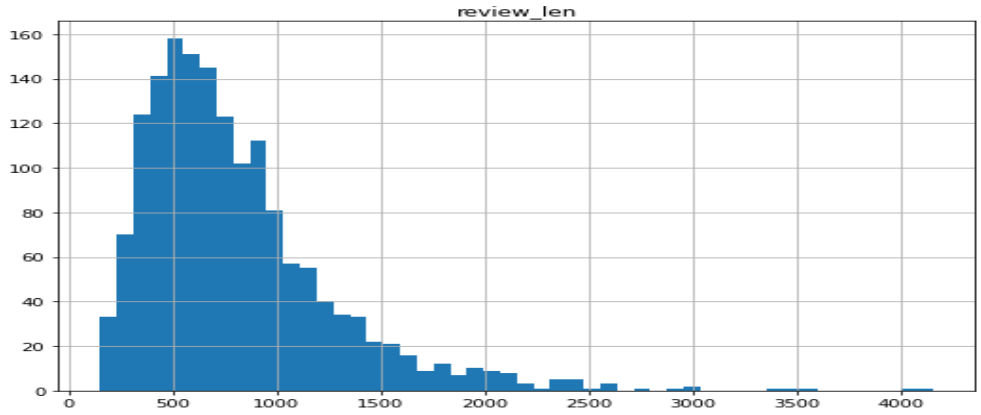


Figure 4: Reviews Length

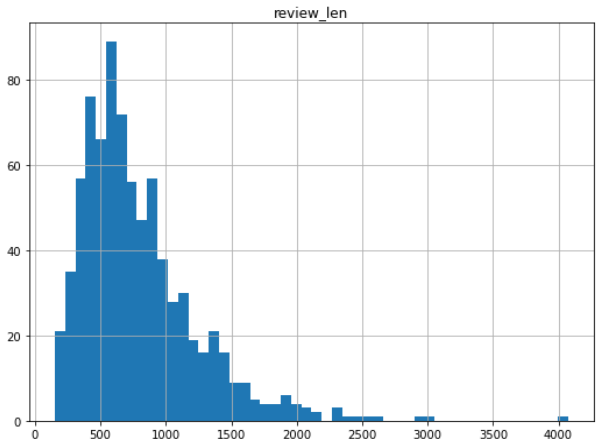


Figure 5: Deceptive Reviews Length

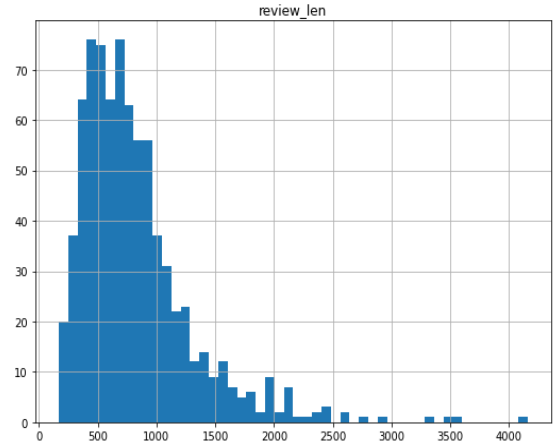


Figure 6: Truthful Reviews length

Word Count

Next, we counted the number of words in each review. We can see that most reviews have between 80 to 150 words(**Figure 7**). There are very few reviews that go beyond 400 words. We have plotted similar histograms for deceptive and truthful reviews individually, **Figure 8** and **Figure 9**. Analyzing them, we can see that there are more truthful reviews between the ranges of 0-200 words and greater than 500 words.

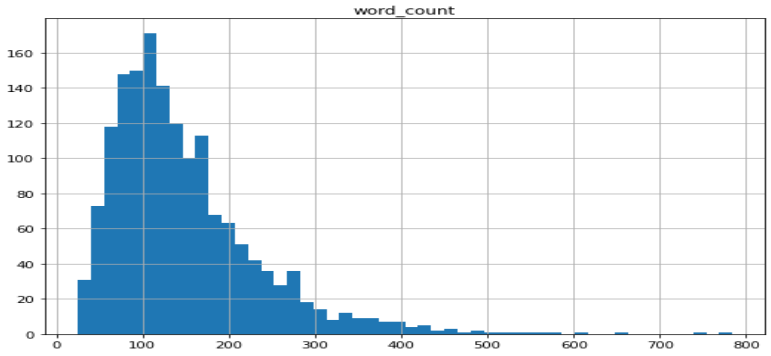


Figure 7: Word Count

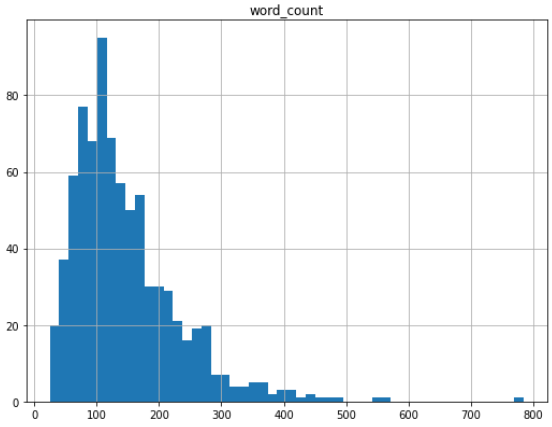


Figure 8: Deceptive Reviews Word Count

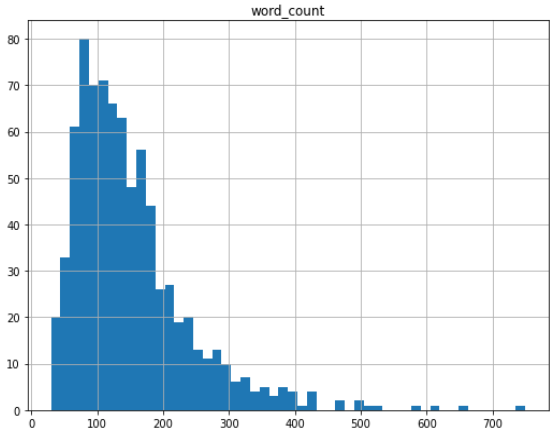


Figure 9: Truthful Reviews Word Count

Data Pre-Processing

Spell check: Sometimes reviews, or any blog data may contain typo errors, hence first we need to correct that data to reduce multiple copies of the same words, which represent the same meaning. We are using SpellChecker to remove misspelled words.

Lower case: The given corpus is in english. If we do not change the casing of the words, the word at the beginning of the sentence and the same word in the middle of the sentence will be treated differently. This might lead to decline in accuracy. So, lowercasing is important here. (But, if the corpus is not in English, then maybe this step won't be required. E.g. any corpus in German, the capitalization is an important part. Similarly, for programming languages, we need to be mindful while lowercasing the letters, as “System” in Java and “System” in python are quite different. And, lowercasing may cause classifiers to lose important predictive features.

As discussed in the Part -A presentation, there are various methods available for feature extraction and selection. We have discussed first all available methods briefly below, then we have discussed in depth about the methods we have implemented.

- **Bag of Words (N-gram words):** In a bag of words approach, individual or small groups of words from the text are used as features. These feature is also called n-grams and are made by selecting n contiguous words from a given sequence, i.e., selecting one, two or three contiguous words from a text. These are denoted as a unigram, bigram, and trigram (n = 1, 2 and 3) respectively.
- **Term Frequency:** Term frequency is similar to that of the bag of words dataset; however, instead of simply being concerned with the presence or absence of a term, we are concerned with the frequency with which a term occurs in each review, so we include the count of occurrences of a term in the review.
- **Linguistic Inquiry & Word Count(LIWC):** LIWC is a text analysis software(paid) tool in which users can build [their] own dictionaries to analyze dimensions of language specifically relevant to [their] interests.
- **POS tagging:** Part of Speech (POS) tagging involves tagging word features with a part of speech based on the definition and its context within the sentence in which it is found .
- **Stylometric:** In Stylometric features analyze the writing style of the reviewer. Lexical features give an indication of the types of words and characters that the writer likes to use and includes features such as number of uppercase characters or average word length. Syntactic features try to “represent the writing style of the reviewer” and include features like the amount of punctuation or number of function words such as “a”, “the”, and “of”.
- **Semantic:** Semantic features deal with the underlying meaning or concepts of the words to create semantic language models for detecting untruthful reviews. The rationale is that changing a word like “love” to “like” in a review should not affect the similarity of the reviews since they have similar meanings

- **<u>Review characteristics:</u>** In Review characteristics, metadata of the given corpus is analyzed. In review characteristics we count frequency of words appearing, what is the word length - like a truthful review has a high number of characters or words insight it or deceptive, and what is the polarity and subjectivity of each review.

Now, we will see details of Bag of Words (N-gram words), POS tagging, and Review Characteristics feature selection and extraction method. Also, code level implementation of these features is provided in the later part of this document in our codebase.

Stop words removal

Stop words are a set of commonly used words in any language. For example, in English, “the”, “is” and “and”, would easily qualify as stop words. In NLP and text mining applications, stop words are used to eliminate unimportant words, allowing applications to focus on the important words instead. Figure 10 and Figure 11 display the 20 most frequent words and bigrams before removing stop words.

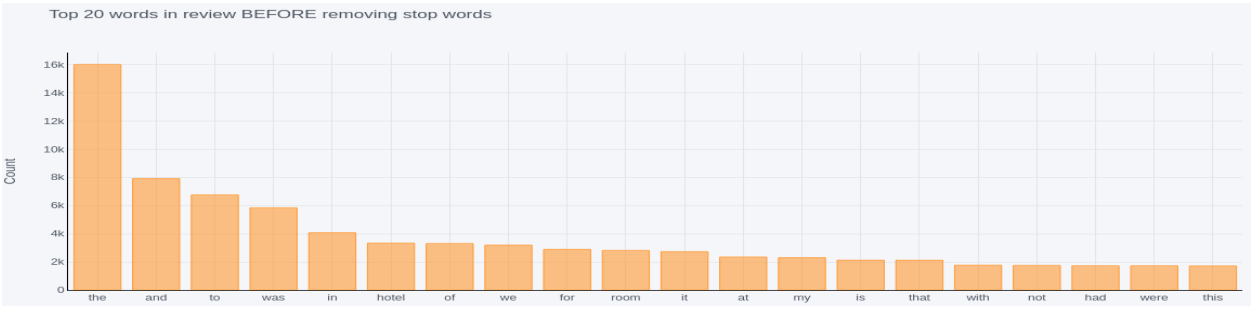


Figure 10: Top 20 unigrams before removing stop words

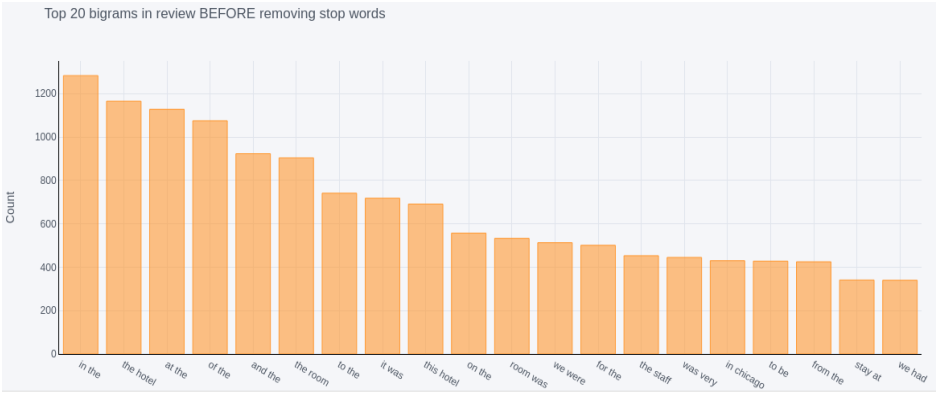


Figure 11: Top 20 bigrams before removing stop words

Figure 12 and Figure 13 display the 20 most frequent words and bigrams before removing stop words. Comparing the four figures helps us realize how unimportant words make up most of the data set and the importance of removing them.

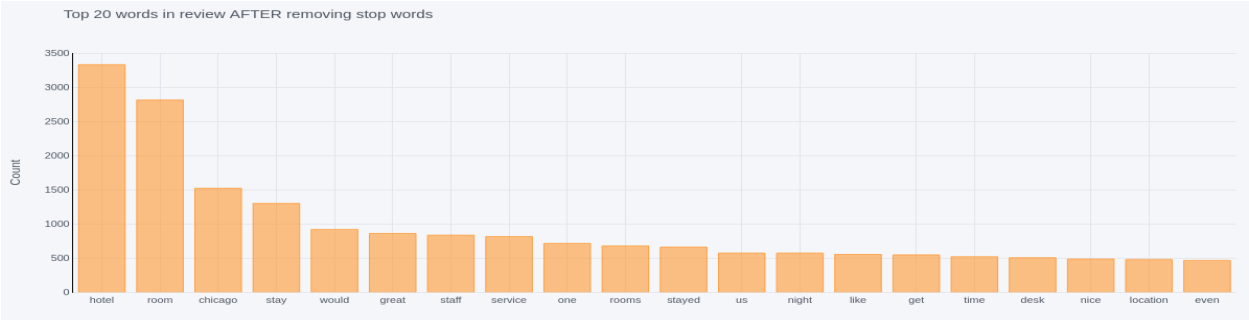


Figure 12 - Top 20 unigrams after removing stop words

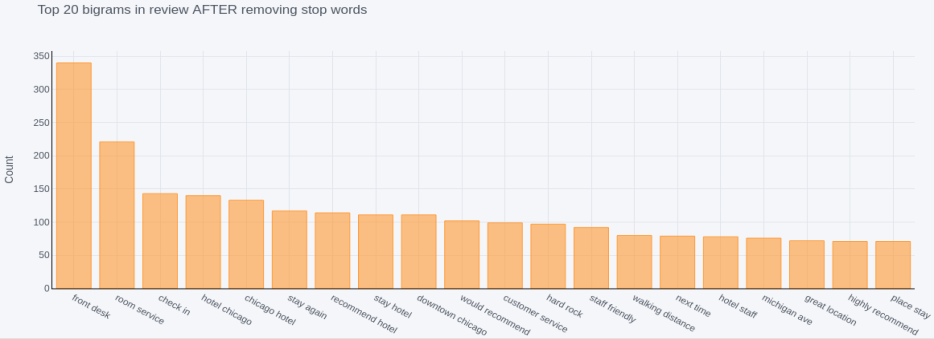


Figure 13 - Top 20 bigrams before removing stop words

But stop words (which needs to be removed from text) must be chosen carefully to avoid accidentally changing the meaning of sentences. For example "Hotel is not very good'" is a negative review. But if we consider words like "not" and "very" as stopwords and remove them, then the altered review would be "Hotel is good", (which is a positive review); and the whole meaning of the sentence got changed.

Thus, unmindful usage of stop words can impact the truthfulness and accuracy of the model. Therefore, instead of using stopwords provided by NLTK(Natural Language ToolKit), we have created our own bag of stop words and used that for preprocessing.

Part of Speech tagging

Part-of-speech (POS) tagging is a popular Natural Language Processing technique which refers to categorizing words depending on the definition of the word and its context. It uses a short representation referred to as **"tags"** to represent the categories (**Figure 14**). It has been generally observed that Informative content has more nouns, adjectives, prepositions and determiners. On the other hand, Imaginative content has verbs, adverbs and pronouns.

Lexical Term	Tag	Example
Noun	NN	Paris, France, Someone, Kurtis
Verb	VB	work, train, learn, run, skip
Determiner	DT	the, a
...	...	

Why	not	tell	someone	?
WRB	RB	VB	NN	.

Tag	Description
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun

Tag	Description
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Whdeterminer
WP	Whpronoun
WP$	Possessive whpronoun
WRB	Whadverb

Figure 14: POS Tagging

We split the reviews into truthful and deceptive and used a python library called TextBlob to implement part of speech tagging on each half. We found that nouns, adjectives and cardinal numbers form the majority of the tags in both deceptive and truthful reviews.

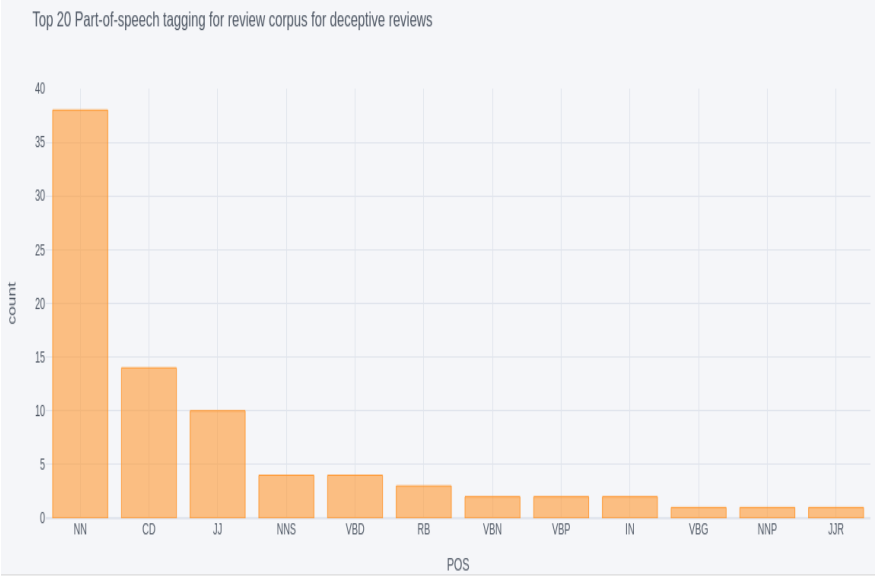


Figure 15 - Top POS tags for deceptive reviews

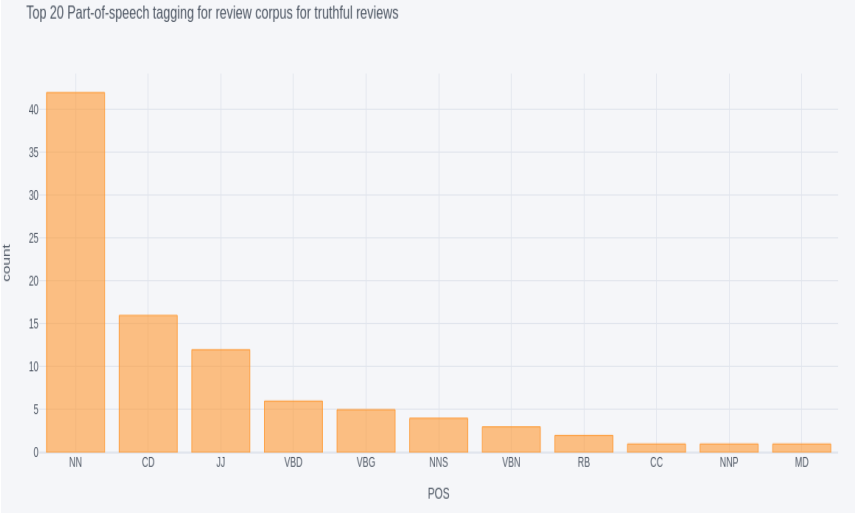


Figure 16 - Top POS tags for truthful reviews

Polarity

Polarity in sentiment analysis refers to positive or negative sentiment in text. We used the TextBlob package for this purpose. Here, "-1" represents a very negative sentiment and "+1" represents a completely positive sentiment. In our data set we have an equal number of positive and negative reviews. But from **Figure 17**, we can see that most of the reviews have a positive sentiment. Therefore, we can infer that even the negative reviews are not very harsh.

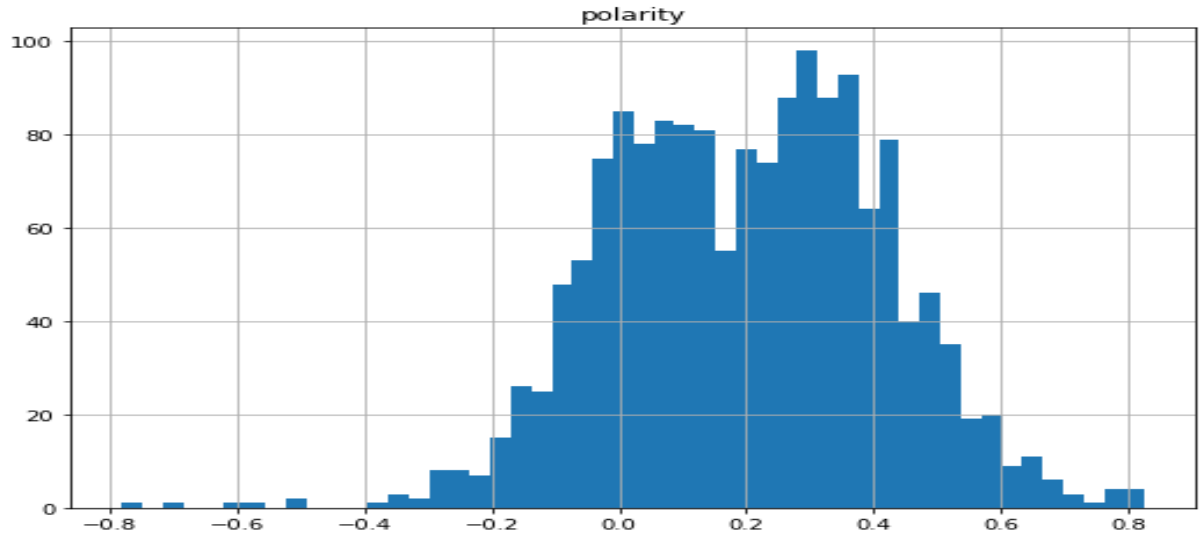


Figure 17 - Polarity of all reviews

After splitting the data set into truthful and deceptive, we calculated the polarity again. In **Figure 18**, we can see that deceptive reviews have more negative sentiment as there are more reviews in the range of "-0.4 to 0". But in **Figure 19**, for truthful reviews we see that the number of reviews with a negative sentiment is much lesser. This indicates that legitimate reviewers are not likely to be very harsh even in their bad reviews but the illegitimate ones tend to be more critical. One of the possible reasons for this could be the illegitimate reviewers' intention to harm the reputation of an establishment.

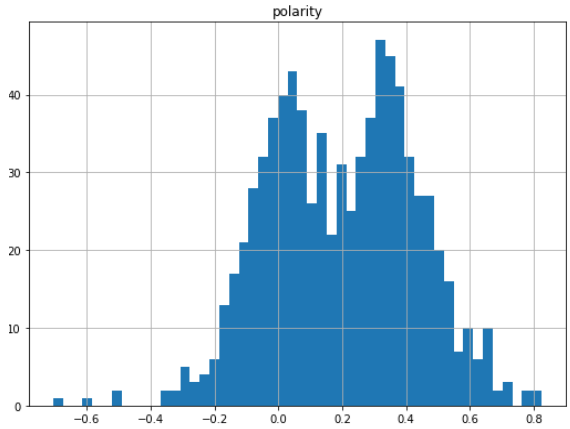


Figure 18 - Polarity of deceptive reviews

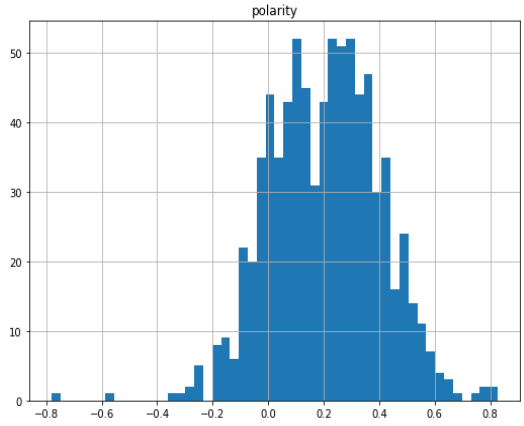


Figure 19 - Polarity of truthful reviews

Subjectivity

Subjectivity pertains to if the language is *objective* or *subjective*. Objective expressions are facts. Subjective expressions are opinions that describe people's feelings towards a specific subject or topic. For example, "This apple is red" is an objective statement. Whereas, "This apple tastes good." is a *subjective* statement as it expresses an opinion towards the taste of the apple. We have used the Textblob library to measure subjectivity. We get a float value within the range 0 to 1, where 0 is very objective and 1 is very subjective. In **Figure 20**, you can see that the reviews are more subjective. This is expected as people tend to share their opinions on reviews.

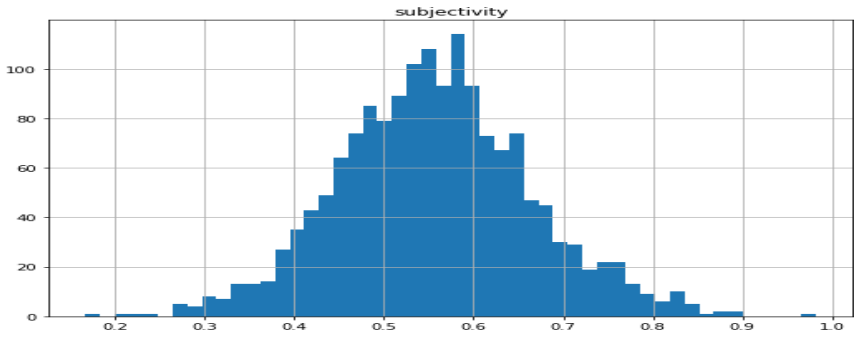


Figure 20 - Subjectivity of all reviews

Then we measured the subjectivity of deceptive and truthful reviews individually. In **Figure 21**, we can see that the deceptive reviews have fewer numbers of values that are closer to 0, hence are more subjectivity. On the other hand, the subjectivity of truthful reviews are more evenly split (**Figure 22**). So we can conclude that truthful reviews are more objective than deceptive reviews.

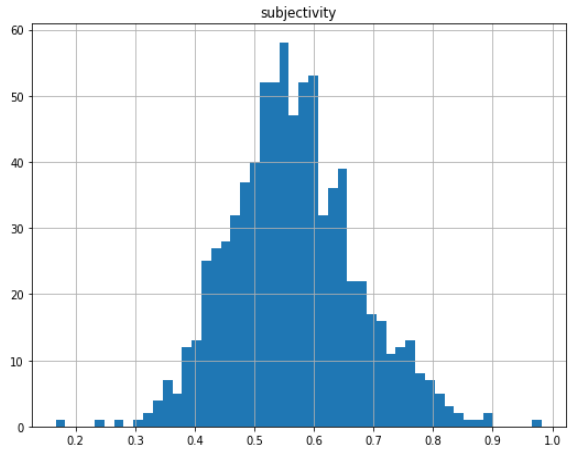


Figure 21 - Subjectivity of deceptive reviews

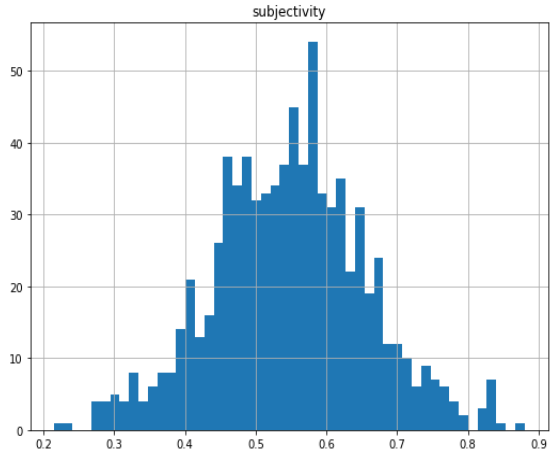


Figure 22 - Subjectivity of truthful reviews

After performing the above mentioned pre-processing steps, we end up with a dataframe that has the following attributes.

	HotelReviews	review_without_stopwords_x	pos	polarity	subjectivity	review_len	word_count	Labels
0	-review of the hard rock hotel- i find that th...	-review hard rock hotel- find hard rock hotel,...	-review/RB hard/JJ rock/NN hotel-/JJ find/RB h...	0.153030	0.516667	523	92	deceptive
1	the ambassador east hotel, located in downtown...	ambassador east hotel, located downtown chicag...	ambassador/NN east/JJ hotel/NN located/VBN dow...	0.403462	0.583590	778	146	deceptive

Feature Importance

We measured and compared the relative importance of word count, review length, subjectivity and polarity by running the model on extraTreesClassifier. Amongst these 4 features, you can see that polarity has the highest value, indicating that it plays an important role in determining the legitimacy of a review.

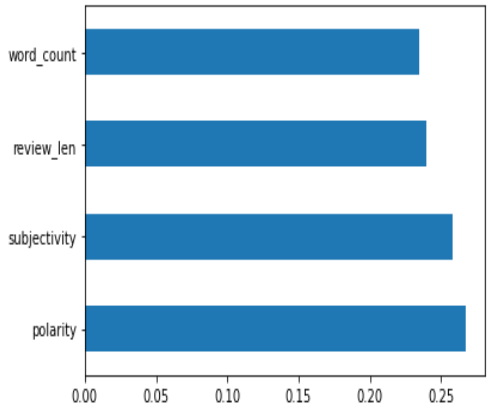


Figure 23 - Feature importance

Survey of Potential Models

To select a model, we decided to utilize existing research to inform our position. We've outlined some selected published papers and their results in **Figure 24 and Figure 25**. The accuracy scores used are F-measure and AUC. F-measure is the harmonic mean of precision and recall (balances both), where precision is true positives/all positives and recall is true positives/all relevant. (validity and completeness, respectively). AUC score on the other hand is the area under curve, after plotting sensitivity vs specificity, (based on false positive rate, true positive rate).

The first of these is a paper by Myle Ott from 2011. Ott and his research group used an SVM (Support Vector Machine) classifier using bigram and LIWC features on hotel review data, and found a 89.8% accuracy rating. Ott's group published again in 2013, using SVM with unigram and bigram features, and were able to get an 86% accuracy. Somahyeh Show-ji, in 2013, published using SVM and Naïve Bayes classifiers on stylometric features, and got 84% accuracy, compared to a 74% for Naïve Bayes. A 2013 dissertation (eventually published) by Hammad et.al got a whopping 99% accuracy rating on hotel data, but the reviews were in Arabic (limited applicability). Another 2013 study by Mishra et. al. evaluated Naïve Bayes, Random Forest, Random Tree. But they weren't very clear on what their Random Tree implementation was, which made our team reluctant to give too much weight to this study. The obscurity of the journal was also a point of concern.

M. Ott et. al, 2011. *Finding Deceptive Opinion Spam by Any Stretch of the Imagination*, (ACL)	• SVM classifier using bigram/LIWC • 89.8% F-measure on hotel review data
M. Ott et. al, 2013. *Negative Deceptive Opinion Spam*, (ACL)	• SVM classifier using unigram/bigram • 86% F-measure on hotel review data
S. Shojaee et. al, 2013. *Detecting deceptive reviews using lexical and syntactic features* (ISDA)	• SVM classifier using various stylometric features • 84% F-measure on hotel review data • Outperformed Naive Bayes (NB) classifier (74%)
A.A. Hammad et. al, 2013. *An Approach for Detecting Spam in Arabic Opinion Reviews (*Dissertation)	• NB classifier • 99.59% (!) F-measure on hotel data • Arabic language, not English
R. Mishra et a., 2013. *Analysis of Random Forest and Naive Bayes for Spam Mail using Feature Selection Categorization* (IJCA)	• NB, Random Forest classifiers • 60-70% F-measure on product reviews

Figure 24 - Survey of potential models - 1

In 2017, Etaiwi et. al. found SVM and NB to again be top performers relative to other classifiers when using word count features. In 2018, a paper by Algotar et. al. found all four of these methods, SVM, KNN, NB, Random Forest, to have a 90% accuracy evaluating spam emails, but the top performer was Random Forest. In 2018, Naveen Kumar found a moderate range. Many different classifiers, generally around 70-75%, plus or minus 10. These are restaurant reviews, and logistic regression scored highest.
In 2020, K.Sabira et. al. found that SVM and NB did very well on hotel reviews. But we found this paper and journal a bit questionable.

Based on these results, we see that SVM and NB do well across the board, and especially in the hotel review domain, both in prestigious and less prestigious journals. So, since they seemed to work particularly well with consistency, we went with those two. For our 3rd classifier, there's less of a clear winner in data, but random forest did perform very well on a different domain – spam emails, and it does decently across the board, and would be an improvement over decision trees, which perform acceptably.

W. Etaiwi et. al, 2017. *The Effects of Feature Selection Methods on Spam Review Detection Performance* (ICTCS)	• NB, Decision Tree, SVM, and Random Forest classifiers • Using bag of words and word count features to evaluate hotel reviews • SVM and NB top performers using word count (~87%) • SVM + bag of words did poorly, others comparable
K. Algotar et. al, 2018. *Detecting truthful and useful consumer reviews for products using opinion mining* (CEUR WP)	• SVM, KNN, NB, Random Forest classifiers • ~90% F-measure when evaluating spam emails • Random Forest top performer
N. Kumar et. al, 2018. *Detecting Review Manipulation on Online Platforms with Hierarchical Supervised Learning* (JMIS)	• Logistic Regression, KNN, NB, AdaBoost + Decision Stump, CART, Random Forest, SVM, Radial SVM • Restaurant reviews sourced from Yelp • AUC scores varying from 70.1% to 81.7% • F-measures varying from 63.0% to 76.3% • Logistic Regression scoring very high
K. Sabira et. al, 2020. *Predicting Fake Online Reviews using Machine Learning* (IJSRED)	• SVM, NB, and Decision Tree classifiers • 84% to 92% on hotel reviews for SVM, NB (DT poor)

Figure 25 - Survey of potential models - 2

Support Vector Machines

The concept is to find a distinctly classifying hyperplane in N-space, where N is the number of features. As shown in Figure 26, a line on a 2d space would be a plane in 3d space. Support vectors are the critical data points closest to the plane, and are the namesake of the classifier. The main concept is that when no plane exists that sufficiently divides data, we use the kernel method, and go to a higher dimension. In our implementation, we'll be doing a weight calculation using TF-IDF (term frequency inverse document frequency). TF-IDF is the number of times a word appears in a document, times the log of the proportion of total documents to documents containing the word.

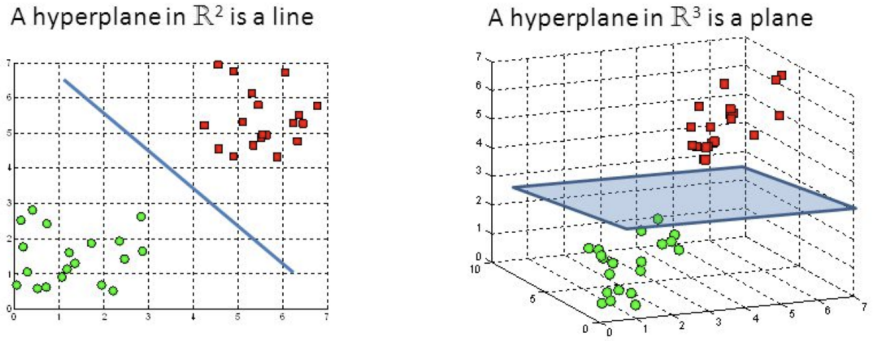


Figure 26 - Support Vector machines

Naive Bayes

The Naive Bayes model is based on Bayes Theorem, which, simplified, is just the probability of event A given that B has happened. The naive term refers to treating all the predictor features as independent - they don't affect each other. As an example, in a spam email, if we look at the independent probability of certain terms, like "Nigerian prince" -- that has a high probability of classifying as spam.

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

Figure 27 - Naive Bayes Theorem

Random Forest

Random Forests are based on Decision Trees, which are prone to overfitting. To address this shortcoming, Random Forest constructs a cluster of Decision Trees. It works on the principle that a large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models. These decision trees are randomized over the dataset, through bagging, each DT gets a random subset of the data. And then, each Decision Tree also selects randomly from the features. Decrease correlation between trees, which improves the model. The low correlation between models is the key. Just like how investments with low correlations (like stocks and bonds) come together to form a portfolio that is greater than the sum of its parts, uncorrelated models can produce ensemble predictions that are more accurate than any of the individual predictions. The reason for this wonderful effect is that the trees protect each other from their individual errors (as long as they don't constantly all err in the same direction).

PART B

Train-test split

For this project, we have split 80% data as training data and 20% data as test data. For the given dataset, we are reading either all truthful or deceptive first, so while training the data we have enabled shuffle to give randomness to the data.

Experiment settings

	SVM	**Random Forest**	**NB**	**NB(Count Vectorize, Alternate method)**
Train Data, Test Data	80%, 20%	80%, 20%	80%, 20%	80%, 20%
Metrics	Accuracy Score, F1 Score, Recall, Precision	Accuracy Score, F1 Score, Recall, Precision	Accuracy Score, F1 Score, Recall, Precision	Accuracy Score, F1 Score, Recall, Precision
Train Time(s)	3.839124822616577	6.32356448173523	0.007526922225952149	0.000457572937011171876
Hardware used for Training	Intel(R) Xeon(R) CPU @ 2.20GHz RAM:13305332 kB (colab CPU)	Intel(R) Xeon(R) CPU @ 2.20GHz RAM:13305332 kB (colab CPU)	Intel(R) Xeon(R) CPU @ 2.20GHz RAM:13305332 kB (colab CPU)	Intel(R) Xeon(R) CPU @ 2.20GHz RAM:13305332 kB (colab CPU)

Commands used for finding hardware settings

1. !lscpu
2. !grep MemTotal /proc/meminfo
3. !df -h

Google Colab gives us three types of runtime for our notebooks: CPUs, GPUs, and TPUs. Colab gives us 12 hours of continuous execution time. After that, the whole virtual machine is cleared and we have to start again. We can run multiple CPU, GPU, and TPU instances simultaneously, but our resources are shared between these instances. We have used CPU instances here for our project code execution.

Training and Fine Tuning

Find the number of trees for Random Forest model

Identifying the ideal number of forests (n_estimators), a hyperparameter is important to obtain the highest accuracy and f1 score while using the Random Forest model. Therefore, we plotted graphs to understand how the number of trees impacted Accuracy and F1 score(**Figure 28** and **Figure 29**). We identified 500 as the ideal number of trees for the given data set and used it to train and test.

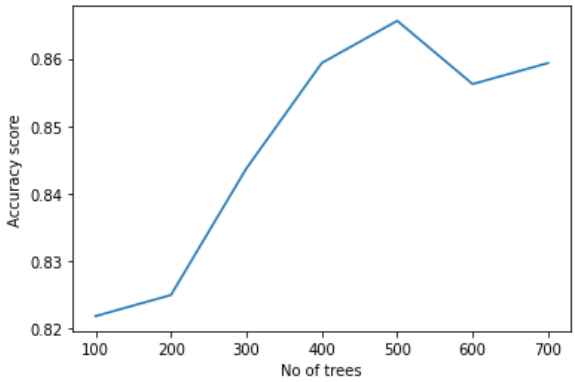


Figure 28 - Accuracy vs No of trees

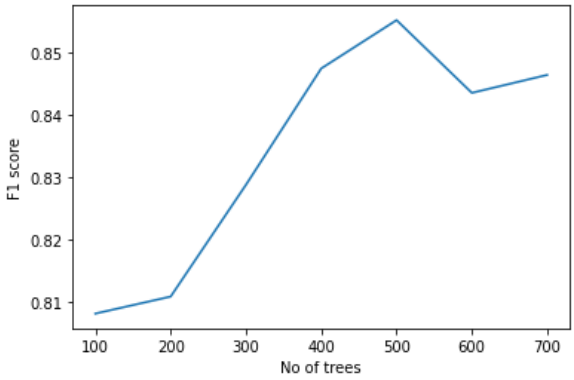


Figure 29 - F1 score vs No of trees

Find the alpha value for Naive Bayes model

Identifying the ideal alpha, a hyperparameter is important to obtain the highest accuracy and f1 score while using the Naive Bayes model. Therefore, we plotted graphs to understand how alpha impacted Accuracy and F1 score(**Figure 30** and **Figure 31**). We identified 0.1 as the ideal alpha for the given data set and used it to train and test.

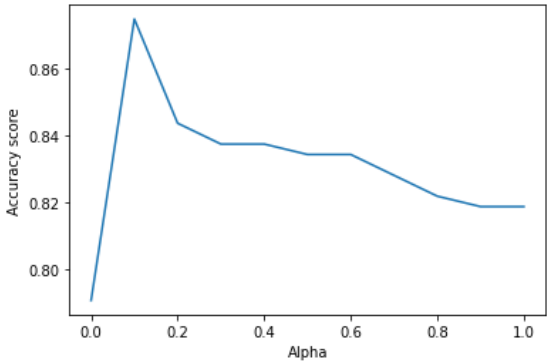


Figure 30 - Accuracy vs Alpha

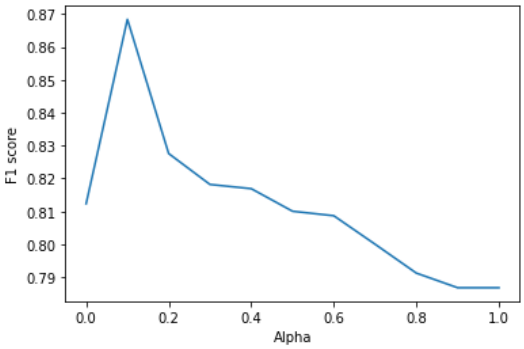


Figure 31 - F1 score vs Alpha

Evaluation results on train and test set

Note: Average of 5 runs is taken into consideration below.

	SVM, N-Gram Vectors	**Random Forest, N-Gram Vectors**	**Naive Bayes, N-Gram Vectors - TFIDF**	**Naive Bayes, CountVectorize (Alternate method)**
Train Result	Average accuracy score 0.99296875 Average f1 score 0.9929632525410476 Average Training time 3.864674425125122 Average Recall 0.9968602825745683 Average Precision 0.9890965732087228	Average accuracy score 1.0 Average f1 score 1.0 Average Training time 6.1178144931793215 Recall 1.0 Precision 1.0	Average accuracy score 0.99765625 Average f1 score 0.9976433621366849 Average Training time 0.009721755981445312 Average Recall 0.9968602825745683 Average Precision 0.9984276729559747	Average accuracy score 0.97 Average f1 score 0.9695945945945945 Average Training time 0.0010997295379638673 Average Recall 0.9630872483221478 Average Precision 0.976190476190476
Test Result	Average accuracy score: 0.878125 Average f1 score: 0.8761904761904763 (87.61%) Average Training time: 3.839124822616577 Average Recall: 0.8466257668711655 Average Precision: 0.9078947368421053	Average accuracy score: 0.865625 Average f1 score: 0.8552188552188552 (85.52%) Average Training time: 6.32356448173523 Recall: 0.7791411042944786 Precision: 0.9477611940298507	Average accuracy score: 0.875 Average f1 score: 0.868421052631579 (86.84%) Average Training time: 0.007526922225952149 Average Recall: 0.8098159509202454 Average Precision: 0.9361702127659575	Average accuracy score: 0.865 Average f1 score: 0.85 Average Training time: 0.00045757293701171876 Average Recall: 0.8138297872340425 Average Precision: 0.8895348837209303

Comparison of machine learning models

Train Results Comparison

	SVM	Random Forest	NB (TF/IDF)	NB(Count Vectorize, Alternate method)
Accuracy Score	0.99296875	1.0	0.99765625	0.97
F-1 Score	99.29%	100%	99.76%	96.95%
Recall	99.68%	100%	99.68%	96.30%
Precision	98.90%	100%	99.84%	97.61%
Training time	3.864674425125122	6.1178144931793215	0.009721755981445312	0.0010997295379638673

Test Results Comparison

	SVM	Random Forest	NB (TF/IDF)	NB(Count Vectorize, Alternate method)
Accuracy Score	0.878125	0.865625	0.875	0.865
F-1 Score	87.61%	85.52%	86.84%	85%
Recall	84.66%	77.91%	80.98%	81.38%
Precision	90.78%	94.77%	93.61%	88.95%
Training time	3.839124822616577	6.32356448173523	0.007526922225952149	0.00045757293701171876

Comparison with related research [optional]

SVM Comparison (F1 Score)

Related Works		Team 9 Implementation
Research Paper	F-1 score	F-1 score
M. Ott et. al, 2011. *Finding Deceptive Opinion Spam by Any Stretch of the Imagination,* (ACL)	89.8%	87.61%
M. Ott et. al, 2013. *Negative Deceptive Opinion Spam,* (ACL)	86%	
S. Shojaee et. al, 2013. *Detecting deceptive reviews using lexical and syntactic features* (ISDA)	84%	
W. Etaiwi et. al, 2017. *The Effects of Feature Selection Methods on Spam Review Detection Performance* (ICTCS)	87%	
K. Algotar et. al, 2018. *Detecting truthful and useful consumer reviews for products using opinion mining* (CEUR WP)	90%	
N. Kumar et. al, 2018. *Detecting Review Manipulation on Online Platforms with Hierarchical Supervised Learning* (JMIS)	70%	
K. Sabira et. al, 2020. *Predicting Fake Online Reviews using Machine Learning* (IJSRED)	84%	

Random Forest (F-1 Score)

Related Works		Team 9 Implementation
Research Paper	F-1 score	F-1 score
R. Mishra et a., 2013. *Analysis of Random Forest and Naive Bayes for Spam Mail using Feature Selection Categorization* (IJCA)	70%	85.52%
W. Etaiwi et. al, 2017. *The Effects of Feature Selection Methods on Spam Review Detection Performance* (ICTCS)	87%	
N. Kumar et. al, 2018. *Detecting Review Manipulation on Online Platforms with Hierarchical Supervised Learning* (JMIS)	76.3%	

NB (F-1 Score)

Related Works		Team 9 Implementation
Research Paper	F-1 score	F-1 score
S. Shojaee et. al, 2013. *Detecting deceptive reviews using lexical and syntactic features* (ISDA)	74%	86.84%
A.A. Hammad et. al, 2013. *An Approach for Detecting Spam in Arabic Opinion Reviews (*Dissertation)	99.59%	
R. Mishra et a., 2013. *Analysis of Random Forest and Naive Bayes for Spam Mail using Feature Selection Categorization* (IJCA)	65%	
W. Etaiwi et. al, 2017. *The Effects of Feature Selection Methods on Spam Review Detection Performance* (ICTCS)	87%	
K. Algotar et. al, 2018. *Detecting truthful and useful consumer reviews for products using opinion mining* (CEUR WP)	90%	
N. Kumar et. al, 2018. *Detecting Review Manipulation on Online Platforms with Hierarchical Supervised Learning* (JMIS)	73%	
K. Sabira et. al, 2020. *Predicting Fake Online Reviews using Machine Learning* (IJSRED)	92%	

Findings and conclusion

We see that SVM has the highest accuracy, F1-score and Recall amongst all models. Naive Bayes performs second, followed by Random Forest. Random Forest has the highest precision compared to all the models. One of the drawbacks of Random Forest is the need to use 500 trees to obtain an accuracy that is close to the other models. This results in it having the longest waiting time. However, the results for Random Forest on the train data is extremely high.

A comparison with similar works is not advised due to the difference in the input data set. But since the models compared also work on opinions, a crude comparison is possible. We see that the team's Support Vector Machine model outperforms four out of the seven works we encountered. Similarly, our Naive Bayes model performs better than three out of seven models and the Random forest model outperforms two out of three models.

Individual contributions (Effort Report)

Major contributions

Siddharth Sheth (UW ID:shethsid, Student ID: 2076180)

- Reading files and creating dataframes.
- Visualization for the unigrams and bigrams
- Read research papers for model selection
- Support Vector Machine model

Patrick Moy (UW ID:moyp, Student ID: 1841305)

- Sentiment analysis
- Part of Speech tagging
- Compiled the team's notes on research papers and made the final recommendation
- Random forest model

Srivatsav Gopalakrishnan (UW ID: sgopal9, Student ID:2075414)

- Stop words
- Histograms for the dataset.
- Read research papers for model selection
- Naive Bayes Model

Effort Report

Tasks (each task considered as 100% for effort distribution)	Siddharth	Srivatsav	Patrick
Reading file and creating dataframe	60	20	20
Stop words	20	50	30
Visualization (Histogram and Analysis)	20	60	20
Sentiment Analysis	30	20	50
Unigrams and bigrams	50	30	20
Part of speech tagging	20	20	60
Feature selection and importance	40	30	30
Survey of machine learning models	30	30	40

References

1. Aslam, U., Jayabalan, M., Aziz, H., & Sohail, A. (2019). A Survey on Opinion Spam Detection Methods. International Journal of Scientific & Technology Research, 8, 1355-1363.
2. Hussain, N., Turab Mirza, H., Rasool, G., Hussain, I., & Kaleem, M. (2019). Spam Review Detection Techniques: A Systematic Literature Review. Applied Sciences, 9(5), 987. https://doi.org/10.3390/app9050987
3. Zhang, W., Du, Y., Yoshida, T., & Wang, Q. (2018). DRI-RCNN: An approach to deceptive review identification using recurrent convolutional neural network. Information Processing & Management, 54(4), 576-592. https://doi.org/10.1016/j.ipm.2018.03.007
4. Lai, S., Xu, L., Liu, K., & Zhao, J. (2015). Recurrent Convolutional Neural Networks for Text Classification. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2267–2273.
5. Stanton, G., & A. Irissappane, A. (2019). GANs for Semi-Supervised Opinion Spam Detection. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, 5204–5210.
6. Crawford, M., Khoshgoftaar, T., Prusa, J.D., Richter, A.N. & Al Najada, H.(2015). Survey of review spam detection using machine learning techniques. Journal of Big Data, 2, 1-24. https://doi.org/10.1186/s40537-015-0029-9
7. Ott, M., Choi, Y., Cardie, C., & Hancock, J.T. (2011). 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, 309–319
8. Ott, M., Cardie, C., & Hancock, J.T. (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.
9. freeCodeCamp.org. (2018, March 23). An introduction to part-of-speech tagging and the Hidden Markov Model. freeCodeCamp.org. https://www.freecodecamp.org/news/an-introduction-to-part-of-speech-tagging-and-the-hidden-markov-model-953d45338f24/.
10. GeeksforGeeks. (2020, November 24). Removing stop words with NLTK in Python. GeeksforGeeks. https://www.geeksforgeeks.org/removing-stop-words-nltk-python/.
11. (2021, April 9). Text Classification: All Tips and Tricks from 5 Kaggle Competitions: Neptune Blog. neptune.ai. https://neptune.ai/blog/text-classification-tips-and-tricks-kaggle-competitions.
12. Bhargava, R., Baoni, A. & Sharma, Y. (2019). Composite Sequential Modeling for Identifying Fake Reviews. Journal of Intelligent Systems, 28(3), 409-422. https://doi.org/10.1515/jisys-2017-0501
13. Shojaee, S., Murad, M., Azman, A., Sharef, N., & Nadali, S. (2013). Detecting deceptive reviews using lexical and syntactic features. In Proceedings of the 13th International Conference on Intellient Systems Design and Applications, 53-58. https://doi.org/10.1109/ISDA.2013.6920707
14. Hammad, A.A., & El-Halees, A. (2015). An Approach for Detecting Spam in Arabic Opinion Reviews. International Arab Journal of Information Technology, 12(1), 9-16. https://doi.org/20.500.12358/20125
15. Mishra, R., & Thakur, R.S. (2013). Analysis of Random Forest and Naive Bayes for Spam Mail using Feature Selection Categorization. International Journal of Computer Applications, 80, 42-47. https://doi.org/10.5120/13844-1670
16. Etaiwi, W., & Awajan, A. (2017). The Effects of Features Selection Methods on Spam Review Detection Performance. In Proceedings of the 2013 Research in Adaptive and Convergent Systems, 116-120. http://doi.org/10.1109/ICTCS.2017.50

17. Algotar, K., & Bansal, A. (2018). Detecting truthful and useful consumer reviews for products using opinion mining. CEUR Workshop Proceedings, 2111, 63-72.
18. Kumar, N., Venugopal, D., Qiu, L., & Kumar, S. (2018). Detecting Review Manipulation on Online Platforms with Hierarchical Supervised Learning. Journal of Management Information Systems, 35, 350-380. http://doi.org/10.1080/07421222.2018.1440758
19. Sabira, K., & Kiruthiga, G. (2020). Predicting Fake Online Reviews Using Machine Learning. International Journal of Scientific Research and Engineering Development, 3(2), 269-273.