NOVA IMS Information Management School

MAA

Mestrado em Métodos Analíticos Avançados
Master Program in Advanced Analytics

Identifying Deception in Online Reviews

Application of Machine Learning, Deep Learning & Natural Language Processing

Bhupendra Roy

Dissertation presented as partial requirement for obtaining the master's degree in Data Science and Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

IDENTIFYING DECEPTION IN ONLINE REVIEWS

Application of Machine Learning,
Deep Learning & Natural Language Processing

by

Bhupendra Roy

Dissertation presented as partial requirement for obtaining the Master's degree in Data Science and Advanced Analytics

Advisor: Fernando José Ferreira Lucas Bação

February 2020

DEDICATION

For my father, Raj Ballav Roy and my mother, Rama Bati Devi Roy, who constantly encouraged and supported me.

A special thanks to my fellow classmate and friend, Susan Wang.

ACKNOWLEDGEMENTS

This thesis is an extension and application of my learning especially in the below class modules among various other class modules in the course of Masters in Advanced Analytics in Nova Information Management School under Universidade Nova de Lisboa.

- [200181] Text Mining by Professor Zita Marinho
- [200180] Deep Learning by Professor Mauro Castelli

As a result, I want to extend my gratitude to both above professors for the lectures and for training me on those topics.

In the class of Text Mining, I along with my fellow classmates (Susan Wang, Carolina Bellani, Yu Chun Liu - April) created a class project on Deceptive Opinion Spam Corpus. In the project, we applied some Natural Language Processing techniques, and utilized Naïve Bayes and Support Vector machine to create classifiers. My current research thesis is result of enhancement and further work continued from the project. Hence, I also wish to express my thanks to my classmates – Susan, Carolina & April.

I want to express my heartfelt gratitude to my Professor Jorge Antunes for constantly encouraging me to work further on the thesis, for guiding me throughout the process and for constantly helping me in every stage of the thesis.

I want to express many thanks to my supervisor, Professor Fernando Baçao for his feedback, time, help and guidance.

Last but not the least I wish to acknowledge the help of all the faculty (specially of course – Masters in Advanced Analytics) and staff of Nova IMS.

ABSTRACT

Customers increasingly rate, review and research products online, (Jansen 2010). Consequently, websites containing consumer reviews are becoming targets of opinion spam. Now-a-days, people are paid money to write fake positive review online, to misguide customer and to augment sales revenue. Alternatively, people are also paid to pose as customers and to post negative fake reviews with the objective to slash competitors. These have caused menace in social media and often resulting in customer being baffled.

In this study, we have explored multiple aspects of deception classification. We have explored four kinds of treatments to input i.e., the reviews using Natural Language Processing – lemmatization, stemming, POS tagging and a mix of lemmatization and POS Tagging. Also, we have explored how each of these inputs responds to different machine learning models – Logistic Regression, Naïve Bayes, Support Vector Machine, Random Forest, Extreme Gradient Boosting and Deep Learning Neural Network.

We have utilized the gold standard hotel reviews dataset created by (Ott, Choi, et al. 2011) & (Ott, Cardie and Hancock, Negative Deceptive Opinion Spam 2013). Also, we used restaurant reviews dataset and doctors' reviews dataset used by (Li, et al. 2014). We explored the usability of these models in similar domain as well as across different domains. We trained our model with 75% of hotel reviews dataset and check the accuracy of classification on similar dataset like 25% of unseen hotel reviews and on different domain dataset like unseen restaurant reviews and unseen doctors' reviews. We perform this to create a robust model which can be applied on same domain and across different domains.

Best accuracy for testing dataset of hotels achieved by us was at 91% using Deep Learning Neural Network. Logistic regression, support vector machine and random forest had similar results like neural network. Naïve Bayes also had similar accuracy; however, it had more volatility in cross domain accuracy performance. Accuracy of extreme gradient boosting was weakest among all the models that we explored.

Our results are comparable and at times exceeding performance of other researchers' work. Additionally, we have explored various models (Logistic Regression, Naïve Bayes, Support Vector Machine, Random Forest, Extreme gradient boosting, Neural network) vis a vis various input transformation method using Natural Language Processing (lemmatized unigrams, stemmed, POS tagging and a mix of lemmatization and POS Tagging).

KEYWORDS

Online deception

Deep Learning

Natural Language Processing

Neural Network

Logistics Regression

Naïve Bayes

Support Vector Machine

Random Forest

Extreme Gradient Boosting

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

CL – Classifier; (Asiri 2018). Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y). Classifier utilizes some training data to understand how given input variables relate to the class. When the classifier is trained accurately, it is used to predict the class of the unseen data.

ML –Machine learning is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead.

DL / NN – Deep learning is an artificial intelligence function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence (AI) that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network.

NLP – (Garbade 2018). Natural Language Processing is a branch of artificial intelligence that deals with the interaction between computers and humans using the natural language. The ultimate objective of NLP is to read, decipher, understand, and make sense of the human languages in a manner that is valuable.

RF – Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees

SVM – (Patel 2017) A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two-dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

NB - (scikit-learn 2007-2019). Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable.

LR – (Chandrayan 2015). Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). It is used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables.

XGB – Extreme Gradient Boosting is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework in prediction problems.

1 INTRODUCTION

Websites and online platforms are bombarded with online reviews from customers. Often individuals read online reviews, prior to making a purchase. Online reviews mold individual's decision to purchase or not. Quite often the reviews are fraudulent to manipulate potential customers purchase decision by presenting a review which shamelessly scream about self-promotion. Also, sometimes companies resort to put maligning negative reviews about competitors to eliminate or reduce competition. Social media managers or public relations executives are hired by companies to present the company in a very positive view and generate more business. In such context, it is extremely important to study and to find an automated method to differentiate between online truthful reviews and deceptive reviews.

There have been various studies in these fields. The most remarkable ones are (Ott, Choi, et al. 2011), (Ott, Cardie and Hancock, Negative Deceptive Opinion Spam 2013), (Li, et al. 2014), etc. among various others. We have studied these and many other works like (Cagnina and Rosso 2015), (Fusilier, et al. 2013), (Ren and Zhang 2016), etc. In these works, we have studied PU learning methods, low dimensionality representation and neural network among other methods used by various researchers.

We have utilized the gold standard positive dataset consisting of 400 truthful and 400 deceptive reviews of 20 hotels of Chicago collected by (Ott, Choi, et al. 2011). Further we have also used negative reviews of same 20 hotels consisting of 400 deceptive and 400 truthful, collected by (Ott, Cardie and Hancock, Negative Deceptive Opinion Spam 2013). We have amalgamated both the datasets and utilized 75% of them (1200 reviews) to train various models and utilized the rest 25% of dataset (400 reviews) to test the model. We further collected restaurant dataset and doctor dataset used by (Li, et al. 2014). We tested our models on restaurant dataset and doctor dataset to check how they perform across other industries.

We train all the models using four kinds of input. First is normalized lemmatized unigrams since word contain the clues to deception. Second is normalized stemmed unigrams since we wanted to check how different stemming input performs against lemmatization. Third is POS tagging since it is structurally very different from unigrams. POS tagging contains the grammatical structure of the review. Fourth is the mix of normalized lemmatized unigrams along with POS tagging.

We explore various models in our study. We have explored Logistic regression, Naïve Bayes, Support Vector Machine, Random Forest, Extreme Gradient Boosting and Neural Network. We have tried to find the best performer, balanced performer to check cross domain usability, weak performers to avoid for these kinds of datasets. Also, we explored which input methodologies give better performance than others.

We have attempted to explore various input methodology by applying different Natural Language Processing techniques and how each of these inputs perform against various machine learning and deep learning models with the motive to find robust tool which provides good performance on similar datasets and good and balanced performance on cross domain datasets. Our conclusion and results are expected to be of potential interest to the readers.

2 RELATED WORK

We have reviewed several research papers to better understand what has been done on the stated problem of identification of deceptive reviews from truthful online reviews. I have explored areas like what kind of dataset has been used by other researchers, source of such dataset, how humans performed at identification of deceit in such reviews and what are the automated approaches tried by other prominent researchers in this field and what results they have got from such automated approaches. Indeed, these research papers of other researchers have been a great source of knowledge and direction for my work.

2.1 QUEST FOR LABELLED DATA

A quest for novel gold standard labelled dataset has been an age-old challenge for performing classification based on textual reviews. In this quest we have explored various datasets available through the length and breadth of various websites and research papers. The problem of lack of gold standard dataset has plagued research in this area. Due to such challenges one of the most prominent researcher (Ott, Choi, et al. 2011) conducted the famous study "Finding Deceptive Opinion Spam by Any Stretch of the Imagination", wherein the researcher created a novel Gold standard dataset which laid the strong foundation of research in this field.

2.1.1 Creation of positive Hotel Reviews

(Ott, Choi, et al. 2011) selected 20 most popular hotels on TripAdvisor from Chicago, United States for studies. Crowdsourcing services such Amazon Mechanical Turk was utilized to collect deceptive opinions. A pool of 400 Human Intelligence Task (HITs) were created in Amazon Mechanical Turk and allocated across 20 chosen hotels. To ensure that opinions are written by unique authors, single submission per online worker was allowed. All online workers were in United States with an approval rating of at least 90%. Workers were allowed a maximum of 30 minutes and were paid one US dollar for an accepted submission. Each HIT presented the worker with the name and website of a hotel. The HIT instructed the worker to assume that they worked for the hotel's marketing department, and to pretend that their boss wanted them to write a fake review (as if they were a customer) to be posted on a travel review website; additionally, the review needed to sound realistic and portray the hotel in a positive light. A disclaimer was provided that indicated that any submission found to be of insufficient quality (e.g., written for the wrong hotel, unintelligible, unreasonably short, plagiarized, etc.) would be rejected. It took approximately 14 days to collect 400 satisfactory deceptive opinions.

(Ott, Choi, et al. 2011) additionally, mined 6977 truthful reviews from TripAdvisor for the 20 hotels. 3130 non-5-start reviews and 41 non-English reviews were rejected. 75 reviews with fewer than 150 characters were eliminated since, by construction, deceptive opinions were at least 150 characters long. 1,607 reviews written by first-time authors were also eliminated since new users who had not previously posted an opinion on TripAdvisor were more likely to contain opinion spam, which would reduce the integrity of the truthful review data. Finally, 400 reviews out of

the remaining 2,124 truthful reviews, were selected such that the document lengths of the selected truthful reviews were similarly distributed to those of the deceptive reviews.

In this way, 40 positive reviews (20 truthful and 20 deceptive) were collected for each of the 20 hotels. It totaled to 800 reviews (400 truthful and 400 deceptive) collected by (Ott et al., 2011).

2.1.2 Creation of negative Hotel Reviews

(Ott, Cardie and Hancock, Negative Deceptive Opinion Spam 2013) collected 800 negative hotel reviews. The process of dataset creation was like the process of dataset creation in (Ott, Choi, et al. 2011). The only difference was that (Ott, Cardie and Hancock, Negative Deceptive Opinion Spam 2013) was creating dataset for negative sentiment instead of positive sentiment. Same 20 hotels like the previous research were chosen for study of negative deceptive opinion. Crowdsourcing services such Amazon Mechanical Turk was utilized to collect deceptive opinions. A pool of 400 Human Intelligence Task (HITs) were created in Amazon Mechanical Turk and allocated across 20 chosen hotels. To ensure that opinions were written by unique authors, single submission per online worker was allowed. All online workers were in United States with an approval rating of at least 90%. Workers were allowed a maximum of 30 minutes and were paid one US dollar for an accepted submission. Each HIT presented the worker with the name and website of a hotel. The HIT instructed the worker to assume that they work for the hotel's marketing department, and their manager had asked them to write a fake negative review of a competitor's hotel to be posted online. A disclaimer was provided that indicated that any submission found to be of insufficient quality (e.g., written for the wrong hotel, unintelligible, unreasonably short, plagiarized, etc.) would be rejected. Submission were manually inspected to ensure that they conveyed a general negative sentiment and they were written for correct hotel.

Negative (1- or 2- star) truthful reviews were mined from six popular online review communities: Expedia, Hotel.com, Orbitz, Priceline, Trip Advisor and Yelp. A sample of the available negative truthful reviews were selected to retain an equal number of truthful and deceptive reviews (20 each) for each hotel. The truthful reviews were slightly larger than the deceptive reviews.

2.1.3 Gold standard balanced hotel reviews

In the above-mentioned method, 40 negative reviews (20 truthful and 20 deceptive) were collected for each of the 20 hotels by (Ott, Cardie and Hancock, Negative Deceptive Opinion Spam 2013). When this was combined with positive reviews collected by (Ott, Choi, et al. 2011), the researchers had 80 reviews (20 truthful and positive, 20 truthful and negative, 20 deceptive and positive, 20 deceptive and negative) for each of 20 hotels. It totaled to 1600 gold standard balanced reviews.

2.1.4 Restaurant reviews

(Li, et al. 2014) selected 10 most popular restaurants of Chicago and the researchers collected a mix of deceptive and truthful reviews of the 10 restaurants from online Turkers using Mechanical Turk, employees (waiter/waitress/cook) and customers.

2.1.5 Doctor reviews

(Li, et al. 2014) also collected doctors reviews – a mix of deceitful and truthful about doctors from online Turkers using Mechanical Turk, employees (doctors) and patients.

2.2 HUMAN PERFORMANCE

In this section, we present how humans perform in identifying deception in online hotel reviews for both positive and negative sentiments.

2.2.1 Positive Hotel Reviews

The efforts of the researchers from various countries helped in population of qualitative labelled gold standard dataset consisting a mix of deceitful reviews and truthful reviews. Now it was noteworthy to evaluate how good are human in identifying the deceitfulness in reviews. First, there were few other baselines for classification task; indeed, related studies (Jindal and Liu 2008); (Mihalcea and Strapparava 2009) had only considered a random guess baseline. Second, assessing human performance was necessary to validate the deceptive opinions gathered. If human performance was low, then deceptive opinions were convincing, and therefore, deserving of further attention.

(Ott, Choi, et al. 2011)'s initial approach was to assess human performance on this task through Amazon Mechanical Turk. Unfortunately, they found that some workers selected among the choices seemingly at random, presumably to maximize their hourly earnings by obviating the need to read the review. So, three volunteer undergraduate university students were solicited to make judgments on a subset of the data. Unlike the online workers, student volunteers were not offered a monetary reward. Consequently, their judgements were considered more honest than those obtained via AMT (Amazon Mechanical Turk).

Additionally, to test the extent to which the individual human judges were biased, performance of two virtual meta-judges were also evaluated by (Ott, Choi, et al. 2011). Specifically, the MAJORITY meta-judge predicted "deceptive" when at least two out of three human judges believed the review to be deceptive, and the SKEPTIC meta-judge predicted "deceptive" when any human judge believed the review to be deceptive.

			TRUTHFUL			**DECEPTIVE**		
		Accuracy	Precision	Recall	F-Score	Precision	Recall	F-score
HUMAN	JUDGE 1	61.9%	57.9	87.5	69.7	74.4	36.3	48.7
	JUDGE 2	56.9%	53.9	95.0	68.8	78.9	18.8	30.3
	JUDGE 3	53.1%	52.3	70.0	59.9	54.7	36.3	43.6
META	MAJORITY	58.1%	54.8	92.5	68.8	76.0	23.8	36.2
	SKEPTIC	60.6%	60.8	60.0	60.4	60.5	61.3	60.9

Table 1 : Human performance of deception detection for reviews - (Ott, Choi, et al. 2011)

(Ott, Choi, et al. 2011) found out from the results that human judges were not particularly effective at this task. Indeed, a two-tailed binomial test failed to reject the null hypothesis that JUDGE 2 and JUDGE 3 perform at-chance (p = 0.003, 0.10, 0.48 for the three judges, respectively). Furthermore, all three judges suffered from truth-bias, a common finding in deception detection research in which human judges were more likely to classify an opinion as truthful than deceptive. In fact, JUDGE 2 classified fewer than 12% of the opinions as deceptive! Interestingly, this bias was effectively smoothed by the SKEPTIC meta-judge, which produced nearly perfectly class-balanced predictions. A subsequent re-evaluation of human performance on this task suggests that the truth-bias could be reduced if judges were given the class-proportions in advance, although such prior knowledge was unrealistic; and ultimately, performance remained like that of Table 1.

Inter-annotator agreement among the three judges, computed using Fleiss' kappa, was 0.11. While there was no precise rule for interpreting kappa scores, (Landis and Koch 1977) suggested traits (Mairesse, et al. 2007), to study tutoring dynamics (Cade, Lehman and Olney 2010), and, most relevantly, to analyze deception (Hancock, et al. 2007); (Mihalcea and Strapparava 2009); (Vrij, et al. 2007).

Kappa values	Interpretation
<0	Poor agreement
0.0-0.20	Slight agreement
0.21-0.40	Fair agreement
0.41-0.60	Moderate agreement
0.61-0.80	Substantial agreement
0.81-1.0	Almost perfect agreement

Table 2: Interpretation of Fleiss' kappa (κ) (Landis and Koch 1977)

2.2.2 Negative Hotel Reviews

Similar approach was carried out by (Ott, Cardie and Hancock, Negative Deceptive Opinion Spam 2013) in assessing the performance of humans in identifying deceptive negative reviews. (Ott, Cardie and Hancock, Negative Deceptive Opinion Spam 2013) solicited three volunteer undergraduate university students to make assessment on a subset of the negative review. Additionally, to test the extent to which the individual human judges were biased, performance of two virtual meta-judges were also evaluated. Specifically, the MAJORITY meta-judge predicted "deceptive" when at least two out of three human judges believed the review to be deceptive, and the SKEPTIC meta-judge predicted "deceptive" when any human judge believed the review to be deceptive.

			TRUTHFUL			**DECEPTIVE**		
		Accuracy	Precision	Recall	F-score	Precision	Recall	F-score
HUMAN	JUDGE 1	65.0%	65.0	65.0	65.0	65.0	65.0	65.0
	JUDGE 2	61.9%	63.0	57.5	60.1	60.9	66.3	63.5
	JUDGE 3	57.5%	57.3	58.8	58.0	57.7	56.3	57.0
META	MAJORITY	69.4%	70.1	67.5	68.8	68.7	71.3	69.9

	SKEPTIC	58.1%	78.3	22.5	35.0	54.7	93.8	69.1

Table 3 : Detection deception performance of human judges - (Ott, Cardie and Hancock, Negative Deceptive Opinion Spam 2013)

(Ott, Cardie and Hancock, Negative Deceptive Opinion Spam 2013) found out from the results that human judges were not particularly effective at this task. The best human judge was 65% accurate. However, while the best human judge was accurate 65% of the time, inter-annotator agreement computed using Fleiss' kappa was only slight at 0.07 (Landis and Koch 1977). Even the majority meta judge could achieve only 69.4% accuracy.

Thus, it was clear from the above-mentioned researches carried out by (Ott, Choi, et al. 2011) & (Ott, Cardie and Hancock, Negative Deceptive Opinion Spam 2013) that humans performed miserably in identifying deception in textual reviews be it positive or negative.

2.3 AUTOMATED APPROACH

In this section we present various automated approaches utilized by various researchers to identify deception in online reviews and the outcomes received by them.

2.3.1 Genre identification

In the genre identification approach to deceptive opinion spam detection, it will be fruitful to test if a relationship exists between truthful and deceptive reviews by constructing, for each review, features based on the frequencies of each POS (Parts of Speech) tag. Such approaches were carried out in various research as carried out by (Ott, Choi, et al. 2011) on positive reviews. Such features also provide a good baseline with which to compare our other automated approaches. (Bachenko, Fitzpatrick and Schonwetter 2008) utilized POS Tagging and achieved 74.9% of True / False propositions correctly in research "Verification and Implementation of Language-Based Deception Indicators in Civil and Criminal Narratives".

2.3.2 LIWC

The Linguistic Inquiry and Word Count (LIWC) software (Pennebaker, et al. 2007) is a popular automated text analysis tool. It has been used to detect personality itraits (Mairesse, et al. 2007), to study tutoring dynamics (Cade, Lehman and Olney 2010), and, most relevantly, to analyze deception (Hancock, et al. 2007); (Mihalcea and Strapparava 2009); (Vrij, et al. 2007).
While LIWC does not include a text classifier, features derived from the LIWC output can be used. LIWC counts and groups the number of instances of nearly 4,500 keywords into 80 psychologically meaningful dimensions. It was iutilized by (Ott, Choi, et al. 2011). One feature for each of the 80 LIWC dimensions were constructed by (Ott, Cardie and Hancock, Negative Deceptive Opinion Spam 2013), which were summarized broadly under the following four categories:
1. Linguistic processes: Functional aspects of text (e.g., the average number of words per sentence, the rate of misspelling, swearing, etc.)

2. Psychological processes: Included all social, emotional, cognitive, perceptual and biological processes, as well as anything related to time or space.
3. Personal concerns: Any references to work, leisure, money, religion, etc.
4. Spoken categories: Primarily filler and agreement words.

(Ott, Choi, et al. 2011)'s focus was on psycholinguistic approach to deception detection on LIWC-based features. LIWC was also utilized by (Cagnina and Rosso 2015) to perform deception identification for textual reviews. LIWC software was also utilized by (Williams, et al. 2014) to distinguish between truthful and deceptive statements.

2.3.3 Text Categorization

This approach to deception detection allows the model to learn from both content and context with n-gram features. Specifically, following three n-gram feature sets, were considered with the corresponding features lowercased and unstemmed by (Ott, Choi, et al. 2011): UNIGRAMS, BIGRAMS+, TRIGRAMS+, where the superscript + indicated that the feature set subsumes the preceding feature set.

(Ott, Choi, et al. 2011) established that automated classifiers outperform human judges for every metric. They used all the three automated approaches mentioned like genre identification, psycholinguistic deception detection and text categorization.

			TRUTHFUL			**DECEPTIVE**		
Approach	Features	Accuracy	Precision	Recall	F-score	Precision	Recall	F-score
GENRE IDENTIFICATION	POS_{SVM}	73.0%	75.3	68.5	71.7	71.1	77.5	74.2
PSYCHOLINGUISTIC DECEPTION DETECTION	$LIWC_{SVM}$	76.8%	77.2	76.0	76.6	76.4	77.5	76.9
TEXT CATEGORIZATION	$UNIGRAMS_{SVM}$	88.4%	89.9	86.5	88.2	87.0	90.3	88.6
	$BIGRAMS^{+}_{SVM}$	89.6%	90.1	89.0	89.6	89.1	90.3	89.7
	LIWC+ $BIGRAMS^{+}_{SVM}$	89.8%	89.8	89.8	89.8	89.8	89.8	89.8
	$TRIGRAMS^{+}_{SVM}$	89.0%	89.0	89.0	89.0	89.0	89.0	89.0
	$UNIGRAMS_{NB}$	88.4%	92.5	83.5	87.8	85.0	93.3	88.9
	$BIGRAMS^{+}_{NB}$	88.9%	89.8	87.8	88.7	88.0	90.0	89.0
	$TRIGRAMS^{+}_{NB}$	87.6%	87.7	87.5	87.6	87.5	87.8	87.6
HUMAN / META	JUDGE 1	61.9%	57.9	87.5	69.7	74.4	36.3	48.7
	JUDGE 2	56.9%	53.9	95.0	68.8	78.9	18.8	30.3
	SKEPTIC	60.6%	60.8	60.0	60.4	60.5	61.3	60.9

Table 4: Performance for three approaches against human judges - (Ott, Choi, et al. 2011)

Standard n-gram–based text categorization techniques have been shown to be effective at detecting deception in text (Jindal and Liu 2008); (Mihalcea and Strapparava 2009); (Ott, Choi, et al. 2011).
(Ott, Cardie and Hancock, Negative Deceptive Opinion Spam 2013) evaluated the performance of linear Support Vector Machine (SVM) classifiers trained with unigram and bigram term-frequency features on the novel negative deceptive opinion spam dataset. They employed the same 5-fold stratified cross-validation (CV) procedure as (Ott, Choi, et al. 2011), whereby for each cross-validation

iteration they trained their model on all reviews for 16 hotels and tested their model on all reviews for the remaining 4 hotels. The SVM cost parameter, C, were tuned by nested cross-validation on the training data. Results appear in Table 4. Each row lists the sentiment of the train and test reviews, where "Cross Val." corresponds to the cross-validation procedure described above, and "Held Out" corresponds to classifiers trained on reviews of one sentiment and tested on the other. The results suggested that n-gram– based SVM classifiers could detect negative deceptive opinion spam in a balanced dataset with performance far surpassing that of untrained human judges. Cues to deception differ depending on the sentiment of the text. Combined sentiment dataset results in performance that was comparable to that of the "same sentiment" classifiers.

			Truthful			**Deceptive**		
Train Sentiment	Test Sentiment	Accuracy	Precision	Recall	F-score	Precision	Recall	F-score
Positive (800 reviews)	Positive (800 reviews, Cross Val.)	89,3%	89,6	88,8	89,2	88,9	89,8	89,3
	Negative (800 reviews, Held Out)	75,1%	69,0	91,3	78,6	87,1	59,0	70,3
Negative (800 reviews)	Positive (800 reviews, Held Out)	81,4%	76,3	91,0	83,0	88,9	71,8	79,4
	Negative (800 reviews, Cross Val.)	86,0%	86,4	85,5	85,9	85,6	86,5	86,1
Combined (1600 reviews)	Positive (800 reviews, Cross Val.)	88,4%	87,7	89,3	88,5	89,1	87,5	88,3
	Negative (800 reviews, Cross Val.)	86,0%	85,3	87,0	86,1	86,7	85,0	85,9

Table 5: Performance for different train and test sets - (Ott, Cardie and Hancock, Negative Deceptive Opinion Spam 2013)

2.3.4 Character n-grams in token

The main difference of character n-grams in tokens which (Cagnina and Rosso 2015) used, with respect to the traditional NLP feature character n-grams, was the consideration of the tokens for the extraction of the feature. Tokens with less than n characters were not considered in the process of extraction neither blank spaces. Character n-grams in tokens preserved the main characteristics of the standard character n-grams: effectiveness for quantifying the writing style used in a text, the independence of language and domains, the robustness to noise present in the text, and, easiness of extraction in any text. But unlike the traditional character n-grams, the proposed feature obtained a smaller set of attributes, that is, character n-grams in tokens avoided the need of feature dimension reduction. The number of attributes obtained with the character n-grams in tokens feature was considerably less, although the effectiveness of this representation was still being good.

2.3.5 Emotions-based feature

(Cagnina and Rosso 2015) - There were some evidences that liars use more negative emotions that truth-tellers. Based on that, researchers obtained the percentages of positive, negative, and neutral emotions contained in the sentences of a review. Then, they used those values as features to

represent the polarity of the text. They obtained the polarities of each sentence and then they obtained the percentages of the polarities associated to the whole review. Finally, they used those values as features.

(Cagnina and Rosso 2015) used term frequency inverse document frequency (tf-idf) weighting scheme. The only text preprocessing made was to convert all words to lowercase characters. Naïve Bayes and SVM algorithms in Weka were used to perform the classification. Researchers only showed the results obtained with SVM because its performance was the best. For all experiments they performed a 10-fold cross-validation procedure in order to study the effectiveness of the SVM classifier with the different representation. F-measure of various features for different sets of reviews (positive, negative and all-combined) are enumerated in Table 6.

Features	**F-measure**		
	800 positive reviews	800 negative reviews	1600 all reviews
3-grams in tokens	0.821	0.826	0.766
4-grams in tokens	0.871	0.851	0.867
LIWC	0.697	0.69	0.676
3 + 4-grams in tokens	0.873	0.832	0.854
3-grams + POSNEG	0.871	0.827	0.858
4-grams + POSNEG	0.873	0.851	0.87
3 + 4-grams + POSNEG	0.877	0.827	0.851
3-grams + LIWC	0.883	0.85	0.866
4-grams + LIWC	**0.890**	**0.865**	**0.879**

Table 6 : F-measure of various applied features - (Cagnina and Rosso 2015)

Four single features (3-grams in tokens, 4-grams in tokens, LIWC & POSNEG) were tried as an input to the model. Single emotions-based feature (POSNEG in the table) did not obtain good results; for that reason, it was not included in the first part of the table. In the second part of the table, the combination of each single feature was used as representation of the reviews. The best result was obtained with the combination of 4-grams in tokens and the articles, pronouns and verbs (LIWC). With the combination of 3-grams and LIWC feature the F-measure is quite similar.

2.3.6 PU Method

(Fusilier, et al. 2013) utilized PU-learning which is a partially supervised classification technique. It is described as a two-step strategy which addresses the problem of building a two-class classifier with only positive and unlabeled examples. Broadly speaking this strategy consisted of two main steps:

- To identify a set of reliable negative instances from the unlabeled set
- To apply a learning algorithm on the refined training set to build a two-class classifier.

In the first step the whole unlabeled set was considered as the negative class. Then, classifier was trained using that set-in conjunction with the set of positive examples. In the second step, the classifier was used to classify (automatically label) the unlabeled set. The instances from the unlabeled set classified as positive were eliminated; the rest of them were considered as the reliable negative instances for the next iteration. This iterative process was repeated until a stop criterion was reached. Finally, the latest built classifier was returned as the final classifier.

Algorithm for PU Learning
i ←1
$\|W_0\| \leftarrow \|U_1\|$
$\|W_1\| \leftarrow \|U_1\|$
While $\|Wi\| <= \|Wi-1\|$ do
$C_i \leftarrow$ Generate Classifier (P, U_i)
$U^L_i \leftarrow C_i(U_i)$
$W_i \leftarrow$ Extract_Positives (U^L_i)
$U_i+1 \leftarrow U_i - W_i$
i ← i +1
Return Classifier C_i

Table 7: PU-Learning algorithm for opinion spam detection - (Fusilier, et al. 2013)

- P is the set of positive instances and
- U_i represents the unlabeled set at iteration i;
- U_1 is the original unlabeled set.
- C_i is used to represent the classifier that was built at iteration i, and
- W_i indicates the set of unlabeled instances classified as positive by the classifier Ci.

Table 7 (algorithm) presents the formal description of the method. In this algorithm, instances (W_1) had to be removed from the training set for the next iteration. Therefore, the negative class for next iteration was defined as $U_i - W_i$. Line 4 of the algorithm showed the stop criterion that was used in our experiments, $|Wi| <= |Wi-1|$. The idea of this criterion was to allow a continue but gradual reduction of the negative instances.

They used Naive Bayes and SVM classifiers as learning algorithms in PU-learning method. These learning algorithms as well as the one-class implementation of SVM were also used to generate baseline results. In all the experiments they used the default implementations of these algorithms. Results were divided into three groups: baseline results, one-class classification results, and PU learning results which are in Table 8 and Table 9:

- Baseline results: The baseline results were obtained by training the NB and SVM classifiers using the unlabeled dataset as the negative class. It also corresponded to the results of the first iteration of the proposed PU learning based method. The rows named as "BASE NB" and "BASESVM" showed these results. The results clearly indicated the complexity of the task and the inadequacy of the traditional classification approach. The best f-measure in the deceptive opinion class (0.68) was obtained by the NB classifier when using 120 positive opinions for training. For the cases considering a smaller number of training instances, this approach generated very poor results. NB outperformed SVM in all cases.

- One-class classification results: This algorithm only used the positive examples to build the classifier and did not take advantage of the available unlabeled instances. Its results were shown in the rows named as "ONE CLASS"; those results were very interesting since clearly show that this approach was very robust when there were only some examples of deceptive

opinions. On the contrary, it was also clear that this approach was outperformed by others, especially by our PU-learning based method, when more training data was available.

- PU-Learning results: Rows labeled as "PU-LEA NB" and "PU-LEA SVM" showed the results of the proposed method when the NB and SVM classifiers were used as base classifiers respectively.

Results indicated that:

- The application of PU learning improved baseline results in most of the cases, except when using 20 and 40 positive training instances;
- PU-Learning results clearly outperformed the results from the one-class classifier when there were used more than 60 deceptive opinions for training.
- Results from "PU-LEA NB" were usually better than results from "PU-LEA SVM". It was also important to notice that both methods quickly converged, requiring less than seven iterations for all cases. In particular, "PU-LEA NB" took more iterations than "PU-LEA SVM", leading to greater reductions of the unlabeled sets, and, consequently, to a better identification of the subsets of reliable negative instances.

Original Training Set	**Approach**	**Truthful**			**Deceptive**			**Iteration**	**Final Training Set**
		Precision	Recall	F-score	Precision	Recall	F-score		
20-D	ONE CLASS	0.500	0.688	0.579	0.500	0.313	0.385		
	BASE NB	0.506	1.000	0.672	1.000	0.025	0.049		
	PU-LEA NB	0.506	1.000	0.672	1.000	0.025	0.049	5	20-D/493-U
520-U	BASE SVM	0.500	1.000	0.667	0.000	0.000	0.000		
	PU-LEA SVM	0.500	1.000	0.667	0.000	0.000	0.000	4	20-D/518-U
40-D	ONE CLASS	0.520	0.650	0.578	0.533	0.400	0.457		
	BASE NB	0.517	0.975	0.675	0.778	0.088	0.157		
	PU-LEA NB	0.517	0.975	0.675	0.778	0.088	0.157	4	40-D/479-U
520-U	BASE SVM	0.519	1.000	0.684	1.000	0.075	0.140		
	PU-LEA SVM	0.516	0.988	0.678	0.857	0.075	0.138	3	40-D/483-U
60-D	ONE CLASS	0.500	0.500	0.500	0.500	0.500	0.500		
	BASE NB	0.569	0.975	0.719	0.913	0.263	0.408		
	PU-LEA NB	0.574	0.975	0.722	0.917	0.275	0.423	3	60-D/449-U
520-U	BASE SVM	0.510	0.938	0.661	0.615	0.100	0.172		
	PU-LEA SVM	0.517	0.950	0.670	0.692	0.113	0.194	3	60-D/450-U

Table 8 : Comparison of classifiers using 20, 40 & 60 D samples-(Fusilier, et al. 2013)

Original Training Set	**Approach**	**Truthful**			**Deceptive**			**Iteration**	**Final Training Set**
		Precision	Recall	F-score	Precision	Recall	F-score		
80-D	ONE CLASS	0.494	0.525	0.509	0.493	0.463	0.478		
	BASE NB	0.611	0.963	0.748	0.912	0.388	0.544		
	PU-LEA NB	0.615	0.938	0.743	0.868	0.413	0.559	6	80-D/267-U
520-U	BASE SVM	0.543	0.938	0.688	0.773	0.213	0.333		
	PU-LEA SVM	0.561	0.925	0.698	0.786	0.275	0.407	3	80-D/426-U

100-D 520-U	ONE CLASS	0.482	0.513	0.497	0.480	0.450	0.465		
	BASE NB	0.623	0.950	0.752	0.895	0.425	0.576		
	PU-LEA NB	0.882	0.750	0.811	0.783	0.900	0.837	7	100-D/140-U
	BASE SVM	0.540	0.938	0.685	0.762	0.200	0.317		
	PU-LEA SVM	0.608	0.913	0.730	0.825	0.413	0.550	4	100-D/325-U
120-D 520-U	ONE CLASS	0.494	0.525	0.509	0.493	0.463	0.478		
	BASE NB	0.679	0.950	0.792	0.917	0.550	0.687		
	PU-LEA NB	0.708	0.850	0.773	0.789	0.781	0.780	5	120-D/203-U
	BASE SVM	0.581	0.938	0.718	0.839	0.325	0.468		
	PU-LEA SVM	0.615	0.738	0.670	0.672	0.538	0.597	6	120-D/169-U

Table 9 : Comparison of classifiers using 80, 100 & 120 D Samples - (Fusilier, et al. 2013)

The best result obtained by the proposed method (F-measure of 0.837 in the deceptive opinion class) was a very important results since it was comparable to the best result (0.89) reported for this collection/task, but when using 400 positive and 400 negative instances for training. Moreover, this result was also far better than the best human result obtained in this dataset, which, according to (Ott, Choi, et al. 2011) was around 60% of accuracy.

This study was conducted by researchers on the same gold standard dataset of 800 reviews - (Ott, Choi, et al. 2011). Here researchers proposed a method based on the PU-learning approach which learns only from a few positive examples and a set of unlabeled data. Evaluation results in a corpus of hotel reviews demonstrated the appropriateness of the proposed method for real applications since it reached a f-measure of 0.84 in the detection of deceptive opinions using only 100 positive examples for training.

2.3.7 Bi-gated Neural Network

(Ren and Zhang 2016) conducted an empirical study where they explored a neural network model to learn document-level representation for detecting deceptive opinion spam. In this paper, the researchers emphasized that the seminal work of (Jindal and Liu 2008), employing classifiers with supervised learning, represented linguistic and psychological cues but failed to effectively represent a document from the viewpoint of global discourse structures. For example, (Ott, Choi, et al. 2011) and (Li, et al. 2014) represented documents with Unigram, POS and LIWC (Linguistic Inquiry and Word Count) features. Although such features performed quite strongly, their sparsity made it difficult to capture non-local semantic information over a sentence or discourse. Researchers highlighted the three-fold potential advantages of using neural networks for spam detection. First, neural models use dense hidden layers for automatic feature combinations, which can capture complex global semantic information that is difficult to express using traditional discrete manual features which is very useful in addressing the limitation of discrete models mentioned above. Second, neural networks take distributed word embeddings as inputs, which can be trained from a large-scale raw text, thus alleviating the sparsity of annotated data to some extent. Third, neural network models can be used to induce document representations from sentence representations, leveraging sentence and discourse information.

The researchers here proposed a three-stage neural network system. In the first stage, a convolutional neural network was used to produce sentence representations from word representations. Second, a bi-directional gated recurrent neural network with attention mechanism was used to construct a document representation from the sentence vectors. Finally, the document representation was used as features to identify deceptive opinion spam. Such automatically induced dense document representation was compared with traditional manually designed features for the task. They compared the proposed models on a standard benchmark (Li, et al. 2014), which consisted of data from three domains (Hotel, Restaurant, and Doctor). Results on in-domain and cross-domain experiments showed that the dense neural features significantly outperformed the previous state-of-the-art methods, demonstrating the advantage of neural models in capturing semantic characteristics. In addition, automatic neural features and manual discrete features were complementary sources of information, and a combination leads to further improvements.

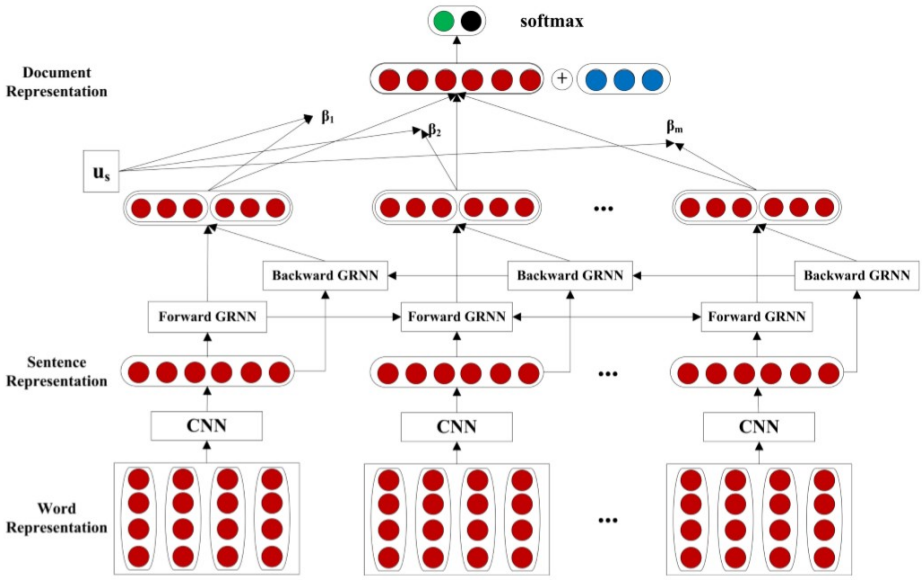


Figure 1 : Neural network model structure used - (Ren and Zhang, 2016)

2.4 Interaction of Sentiment and Deception

Common - (Ott, Choi, et al. 2011) observed that fake positive reviews included less spatial language (e.g., floor, small, location, etc.) because individuals who had not actually experienced the hotel simply had less spatial detail available for their review. This was also the case for our negative reviews, with less spatial language observed for fake negative reviews relative to truthful as per (Ott, Cardie and Hancock, Negative Deceptive Opinion Spam 2013). Likewise, fake negative reviews had more verbs relative to nouns than truthful, suggesting a more narrative style that is indicative of imaginative writing.

(Ott, Cardie and Hancock, Negative Deceptive Opinion Spam 2013) also found following differences. First, fake negative reviewers over-produced negative emotion terms (e.g., terrible, disappointed) relative to the truthful reviews in the same way that fake positive reviewers over-produced positive emotion terms (e.g., elegant, luxurious). Combined data suggested that the more frequent negative emotion terms in the dataset were not the result of “leakage cues” that reveal the emotional distress

of lying. Instead, the differences suggest that fake hotel reviewers exaggerated the sentiment they were trying to convey relative to similarly balanced truthful reviews.

Second, the effect of deception on the pattern of pronoun frequency was not the same across positive and negative reviews. While first person singular pronouns were produced more frequently in fake reviews than truthful, consistent with the case for positive reviews, the increase was diminished in the negative reviews examined here. These results suggested that the emphasis on the self, perhaps as a strategy of convincing the reader that the author had actually been to the hotel, was not as evident in the fake negative reviews, perhaps because the negative tone of the reviews caused the reviewers to psychologically distance themselves from their negative statements, a phenomenon observed in several other deception studies, e.g., (Hancock, et al. 2007).

2.5 CONCLUSION

In the above literature review we have analyzed various papers to explore and find balanced dataset for analysis, to check how human's perform at identification of deceit in online reviews and the evolution of various automated approaches. Also, we explored various classification models like Support vector machine and Naïve Bayes employed by other researchers. Additionally, we explored the interaction between sentiment vs deception. Each passing year add new methodologies or approach to the process makes it leaner or makes it score better accuracy or makes it faster and efficient.

3 METHODOLOGY

In this section, we present dataset, the source of the dataset, various transformations applied, various models used. We also lay down the output capture methodology. Also, we present how we split the data and what kind of experimentation we perform in this research.

3.1 DATASET

In this section, we introduce the data, present the metadata, and perform initial analysis.

3.1.1 Hotel Reviews

In this section, we present below the online hotel reviews:

- Source - We borrow the data from (Ott, Choi, et al. 2011) & (Ott, Cardie and Hancock, Negative Deceptive Opinion Spam 2013).
- Metadata - We begin our journey with our attempt to understand the dataset including its variables. Dataset consists of 1600 records and 5 fields. Description of 5 fields are:
    - Deceptive – It consists of two class of data with values ("truthful" and "deceptive") signifying if the data is a truthful review or fake/deceptive review.
    - Hotel – It consists of name of either of the 20 unique hotels.
    - Polarity – It consists of the sentiment of the review if it is positive or negative.
    - Source – It consists of three classes of data. It specifies the source of the review i.e., TripAdvisor, MTurk, Web.
    - Text – It consist of the actual review provided by the candidate/customer.
- Data Listing - Table 10 is the listing of unique data in each of the 5 fields:

Deceptive	Polarity	Source	Hotel	Text
Deceptive Truthful	negative positive	MTurk TripAdvisor Web	Affinia, allegro, Amalfi, ambassador Conrad, Fairmont, Hardrock, Hilton, Homewood, Hyatt, Intercontinental, James, knickerbocker, Monaco, omni, Palmer, Sheraton, Sofitel, Swissotel,	1600 reviews

			Talbott	

Table 10 : Unique list of data in each field

- Gold Standard dataset - Dataset is gold standard since it is perfectly balanced for each category as depicted in Table 11:

Particulars	Truthful or Deceptive	Source	Polarity	Hotels & reviews
Categories	Deceptive 800	Mturk 800	Positive 400	20 hotels ✕ 20 reviews = 400 reviews
Categories			Negative 400	20 hotels ✕ 20 reviews = 400 reviews
Categories	Truthful 800	TripAdvisor 400	Positive 400	20 hotels ✕ 20 reviews = 400 reviews
Categories		Web 400	Negative 400	20 hotels ✕ 20 reviews = 400 reviews
Total	1600 reviews	1600 reviews	1600 reviews	1600 reviews

Table 11 : Gold standard perfectly balanced dataset

- Text Length Analysis - We create a new variable to measure the length of the textual reviews. We perform in Table 12 exploration of the length. Although no robust conclusion can be drawn regarding classification of reviews into deceptive or truthful.

Measures	Text Length
Count	1600
Median	700
Mean	806.40
Std	467.27
Min	151.00
25%	487.00
50%	700.00
75%	987.50
Max	4159.00

Table 12 : Five descriptive statistics of the length of text

Values	Negative	Positive	All
Deceptive	947.52	636.0150	791.76750
Truthful	964.32	677.7100	821.01500
All	955.92	656.8625	806.39125

Table 13 : Deception vs sentiment mean text length

Deceptive	Deceptive		Truthful		All
Polarity	Negative	Positive	negative	positive	
Hotel					
Affinia	883.40	569.550	1048.00	563.40	766.08750
Allegro	887.35	721.100	1012.50	628.80	812.43750
Amalfi	889.15	810.100	1011.50	624.05	833.70000

Ambassador	937.85	651.400	991.75	765.85	836.71250
Conrad	1243.80	592.750	1050.10	668.40	888.76250
Fairmont	942.75	706.750	935.30	761.30	836.52500
Hardrock	925.45	689.200	1052.95	793.60	865.30000
Hilton	1109.90	494.100	926.25	693.65	805.97500
Homewood	1082.25	590.750	1039.10	635.15	836.81250
Hyatt	841.75	635.400	1082.65	745.70	826.37500
Intercontinental	921.20	717.700	830.05	600.85	767.45000
James	950.70	663.800	979.35	770.85	841.17500
Knickerbocker	1002.45	561.850	896.00	799.50	814.95000
Monaco	748.45	607.350	825.20	609.60	697.65000
Omni	831.85	751.600	767.25	793.90	786.15000
Palmer	901.45	650.300	854.30	652.15	764.55000
Sheraton	962.85	567.350	747.85	666.45	736.12500
Sofitel	864.85	659.650	833.45	585.25	735.80000
Swissotel	994.60	499.100	926.70	539.10	739.87500
Talbott	1028.35	580.500	1476.15	656.65	935.41250
All	947.52	636.015	964.32	677.71	806.39125

Table 14 : Mean text length for each hotel vs category

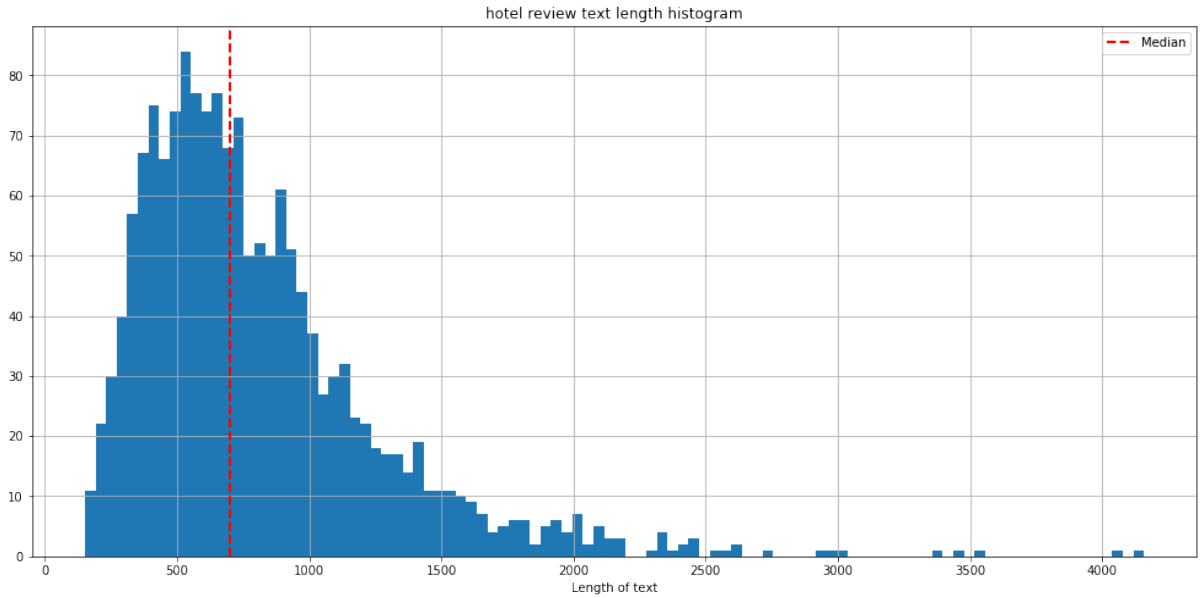


Figure 2 : Distribution of text length of hotel reviews

The distribution is unimodal since we have one peak and it is slightly positive skewed. The tail is pulled by some outliers consisting of long textual reviews. This is also explained by the fact that mean (806.40) is larger than the median (700).

3.1.2 Restaurant Reviews

In this section, we present below the restaurant reviews:

- Source - We borrow the data from (Li, et al. 2014).
- Metadata - We upload 400 files – reviews related with restaurants.

We create three fields in the data frame in python. Description of the fields are:

- Type – It consists of 0 or 1 where 0 is truthful and 1 is truthful reviews
- Number – This is the name of the file.
- Review – It contains the review.

➢ Data Listing - Table 15 is the listing of unique data in each of the 5 fields:

Type	Number	Review
0 1	1.txt 2.txt 400.txt	400 textual reviews

Table 15 : Unique list of data in each field

➢ Gold Standard dataset - Dataset is gold standard since it is perfectly balanced for each category as depicted in table 16:

Particulars	Truthful or Deceptive	Restaurant reviews
Categories	Deceptive 200	400 reviews
Categories	Truthful 200	400 reviews
Total	400 reviews	400 reviews

Table 16 : Gold standard perfectly balanced dataset

➢ Text Length Analysis - We performed analysis of length of the text. In table 17 are some key statistics around it.

Measures	Text length
Count	400.00000
Mean	762.91500
Std	584.24463
Min	174.00000
25%	422.50000
50%	636.00000
75%	940.25000
Max	6148.00000

Table 17 : Five descriptive statistics of the length of text (Restaurant)

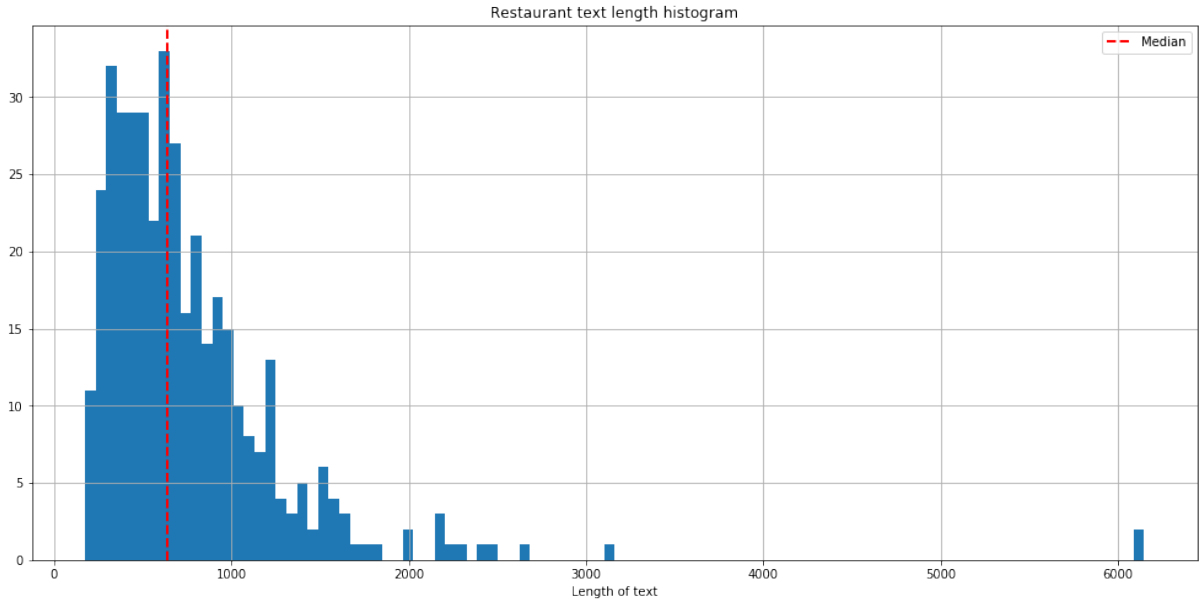


Figure 3 : Distribution of text length of restaurant reviews

The distribution is unimodal since we have one peak and it is slightly positive skewed. The tail is pulled by some outliers consisting of long textual reviews. This is also explained by the fact that mean is larger than the median.

3.1.3 Doctors Reviews

In this section, we present below the doctors' reviews:

- Source - We borrow the data from (Li, et al. 2014).
- Metadata

  We upload 400 files – reviews related with doctors.

  We create three fields in the data frame in python. Description of the fields are:
  - Type – It consists of 0 or 1 where 0 is truthful and 1 is truthful reviews
  - Number – This is the name of the file.
  - Review – It contains the review.
- Data Listing - Table 18 is the listing of unique data in each of the 5 fields:

Type	Number	Review
0 1	1.txt 2.txt 400.txt	400 textual reviews

Table 18 : Unique list of data in each field

- Gold standard dataset - Dataset is gold standard since it is perfectly balanced for each category as depicted in table 19:

Particulars	Truthful or Deceptive	Doctor reviews

Categories	Deceptive 200	400 reviews
Categories	Truthful 200	400 reviews
Total	400 reviews	400 reviews

Table 19 : Gold standard perfectly balanced dataset

- Text Length Analysis - We performed analysis of length of the text. In table 20 are some key statistics around it. The statistics and numbers are uncannily like restaurant dataset.

Measures	Text length
Count	400.00000
Mean	762.91500
Std	584.24463
Min	174.00000
25%	422.50000
50%	636.00000
75%	940.25000
Max	6148.00000

Table 20 : Five descriptive statistics of the length of text (Doctor)

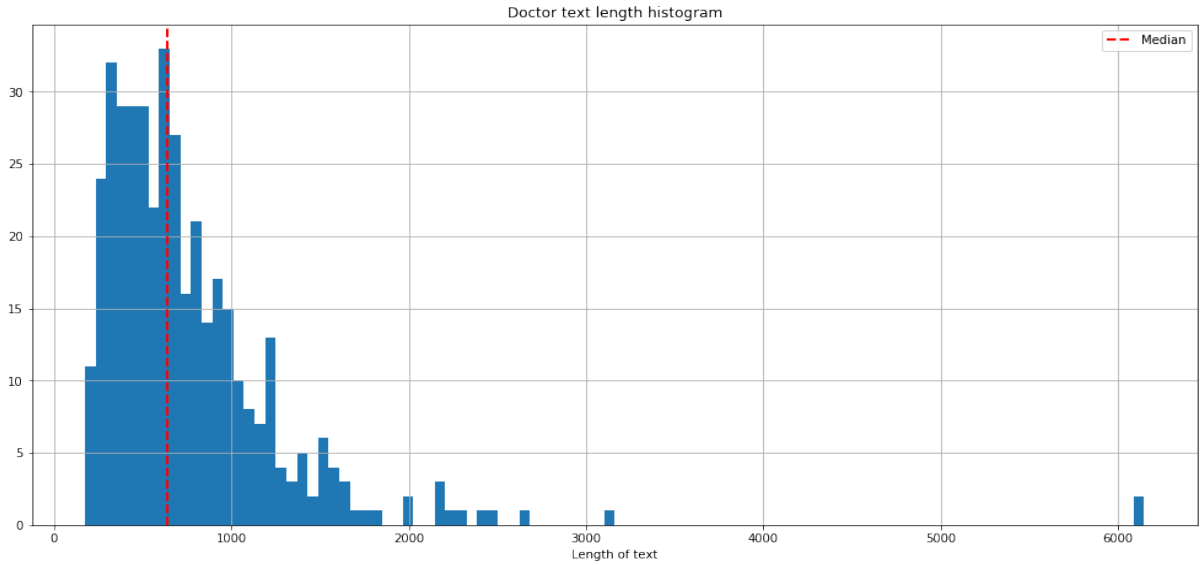


Figure 4 : Distribution of text length of Doctor reviews

The distribution is unimodal since we have one peak and it is slightly positive skewed. The tail is pulled by some outliers consisting of long textual reviews. This is also explained by the fact that mean is larger than the median. The distribution is same as distribution of restaurant dataset.

3.2 NATURAL LANGUAGE PROCESSING

Natural language processing (NLP) is the ability of a computer program to understand human language. NLP is a component of artificial intelligence (AI). Difficulty of NLP arise from the fact that human language is ambiguous whereas traditionally we communicate with computers based on

highly structured and precise programming language. Below are the key concepts of NLP which we utilize in our current project.

3.2.1 Stemming

Stemming is the process of cutting off the word or reducing the word. The stemmed word may or may not have meaning.

Example: "Happening" is converted to "Hap" by stemming. Now "Hap" does not have meaning.
"Happened" is converted to "Hap" by stemming. Now "Hap" does not have meaning.
"Seeing" is converted to "See" by stemming. But "See" have meaning.

3.2.2 Lemmatization

Lemmatization is the process of reducing the word to its base form. The lemmatized word always has meaning.

Example: "Happening" is converted to "Happen" by lemmatization.
"Happened" is converted to "Happen" by lemmatization.

3.2.3 Parts of Speech Tagging

Parts of Speech Tagging is the process of marking each word in the corpus to a part of speech.

Example: "My name is Bhupendra"

"My" is marked as Pronoun; "name" is marked as Noun; "is" is marked as Verb;"Bhupendra" is marked as Noun.

3.3 TEXT TRANSFORMATION METHOD

We perform text normalization before feeding the textual reviews to the models. The transformations are:

- Lower case conversion – To convert the entire textual review to lower text.
- Punctuation removal – To remove all punctuations
- Numbers removal – To get rid of all numeric characters
- Stop word removal – To get rid of all stop words like "a", "an"," and", etc. since they do not add much value.
- Extra white space removal – To get rid of extra white spaces.

After performing above text normalization, we perform one of the below transformations prior to feeding the input to the model.

3.3.1 Lemmatization

Post text normalization, we lemmatize the textual reviews to convert all words to base root and create an array like a binary matrix of all words in the review. This matrix is fed to various models utilized as an input.

3.3.2 Stemming

Post text normalization, we perform stemming on all the textual reviews to convert all words and brutally truncate the endings even if it doesn't make sense and create an array like a binary matrix of all words in the review. This matrix is fed to various models utilized as an input.

3.3.3 POS Tagging

Post text normalization, we create a matrix of continuous numbers where each number represent the percentage of each of the grammatical structures. This matrix is fed to various models as an input.

3.3.4 Lemmatization and POS Tagging

Post text normalization, we lemmatize all words in the reviews to convert them to the base root. Post lemmatization, we further create a matrix of continuous numbers where each number represent the percentage of each of the grammatical structures. This matrix is fed to various models as an input.

3.4 DATA SOURCE

We borrow the 1600 hotel reviews consisting of 800 deceptive and 800 truthful reviews from (Ott, Choi, et al. 2011) & (Ott, Cardie and Hancock, Negative Deceptive Opinion Spam 2013). Also, we utilize the restaurant reviews dataset and doctors' reviews dataset from (Li, et al. 2014).

3.5 DATA SPLIT

We split the hotel reviews dataset into 75% training data and 25% test data. Further we use the entire dataset of restaurant reviews and doctors for testing although we train our models with just hotel reviews. We do this to evaluate cross domain usability.
Additionally, we perform K4-fold validation on 75% training data.

3.6 MODELS

In this section, we present all the six models used in the research.

3.6.1 Logistic Regression

According to (Korkmaz, Güney and Yİğİter 2012), logistic regression analysis is one of the mostly preferred regression methods that can be implemented in modelling binary dependent variables. Logistic regression is a mathematical modelling approach used to define the relationship between such independent variables as X1, X2, …, Xn and Y binary dependent variable which is coded as 0 or 1 for two possible categories.

3.6.2 Naïve Bayes

As per (Kaviani and Dhotre 2017), Naive Bayes is a classification algorithm which is based on Bayes theorem with strong and naïve independence assumptions. It simplifies learning by assuming that features are independent of given class. Naïve Bayes is a subset of Bayesian decision theory. It's called naive because the formulation makes some naïve assumptions. Python's text-processing abilities which split up a document into a vector are used. This can be used to classify text. Classifies may put into human-readable form. It is a popular classification method in addition to conditional independence, overfitting, and Bayesian methods. In the face of the simplicity of Naive Bayes, it can classify documents surprisingly well. Instinctively a potential justification for the conditional independence assumption is that if the document is about politics, this is a good evidence of the kinds of other words found in the document. Naive Bayes is a reasonable classifier in this sense and has minimal storage and fast training, it is applied to time-storage critical applications, such as automatically classifying web pages into types and spam filtering.

3.6.3 Support Vector Machine

(Srivastava and Bhambhu 2010) mentioned that SVMs are set of related supervised learning methods used for classification and regression. They belong to a family of generalized linear classification. A special property of SVM is, SVM simultaneously minimize the empirical classification error and maximize the geometric margin. So SVM called Maximum Margin Classifiers. SVM is based on the Structural risk Minimization (SRM). SVM map input vector to a higher dimensional space where a maximal separating hyperplane is constructed. Two parallel hyperplanes are constructed on each side of the hyperplane that separate the data. The separating hyperplane is the hyperplane that maximize the distance between the two parallel hyperplanes. An assumption is made that the larger the margin or distance between these parallel hyperplanes the better the generalization error of the classifier will be.

3.6.4 Random Forest

(Breiman 2001) described that random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges a.s. to a limit as the number of trees in the forest becomes large. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them. A random forest is a classifier consisting of a collection of tree-structured classifiers {h(x,Θk), k=1, ...} where the {Θk} are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x .

3.6.5 Extreme Gradient Boosting

(Chen and Guestrin 2016), the pioneers of extreme gradient boosting, described Tree boosting as a highly effective and widely used machine learning method. It is a scalable end-to-end tree boosting system called Extreme Gradient Boosting, which is used widely by data scientists to achieve state-of-the-art results on many machine learning challenges. It is a novel sparsity-aware algorithm for sparse data and weighted quantile sketch for approximate tree learning. Extreme Gradient Boosting scales beyond billions of examples using far fewer resources.

3.6.6 Neural Network

A neural network is series of algorithms where inputs are passed through layers with certain weights. Then the predictions are compared with actual output using loss function. Loss score is calculated based on the comparison. An optimizer uses the loss score to optimize the weights and the process is repeated with new weights. Process terminates when a stop criterion is reached.

3.7 "360 DEGREE" PROCESS VIEW

Here, we present below in figure 5 entire 360-degree view of the process.

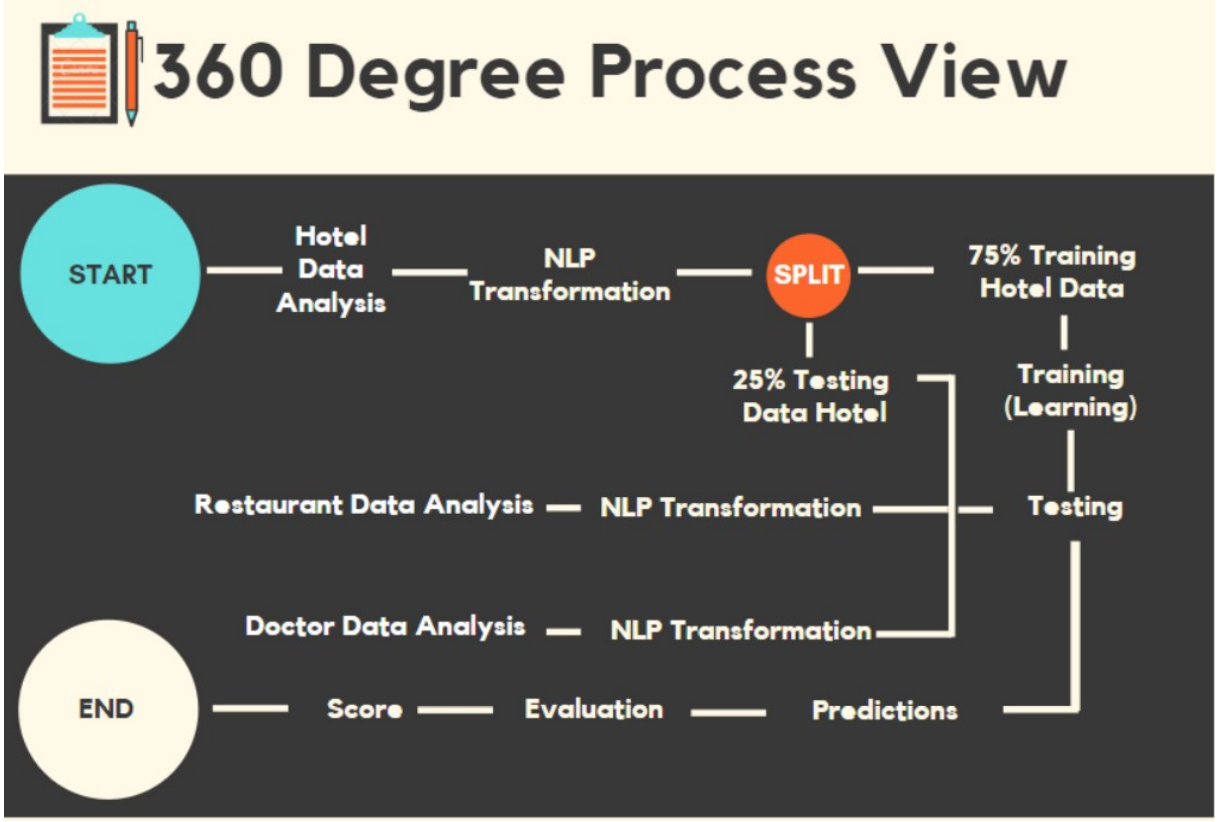


Figure 5 : 360-degree process view from start to end

We present below a summary of the action items performed in this research

- Data Analysis – We kick start our actions on the datasets by performing basic statistical analysis on the text length of hotel, restaurant and doctors' reviews. We check the count of deceptive and truthful reviews in all the three available datasets. We also check the count of positive and negative sentiment reviews in hotel reviews. We try to explore if the dataset is balanced or not.
- Transformations – We perform the transformation on the textual reviews as described in section "TEXT TRANSFORMATION METHOD". We transform the label (output variables) – deceptive and truthful to binary numbers.
- Split – We split the transformed hotel reviews in 75% for training and 25% for testing.
- Training – We train the model using one of the models discussed above and feeding it 75% of training of hotel reviews.
- Testing – We feed the trained model with 25% of testing data of hotel reviews, restaurant reviews and doctors' reviews
- Predictions – We perform the classification on the testing data of hotel reviews, restaurant reviews and doctors' reviews using the knowledge learned from training.
- Evaluation – We compare the classification output received from testing against the actual output.
- Score – We calculate various metrices like precision, recall, accuracy, f-score, etc. for the predicted classification output provided by the models.

3.8 Evaluation Measures

In this section, we present below output capturing methodologies used to evaluate the various transformations used on inputs and to evaluate various models used,

3.8.1 Confusion Matrix

We use confusion matrix to measure and evaluate the performance of our classification models.

		Prediction	
Actual	Values	0	1
	0	True Negative (TN)	False Positive (FP)
	1	False Negative (FN)	True Positive (TP)

Table 21 : Confusion matrix

True Positive – These are those observations where both and predicted and actual output is true (1).

True Negative - These are those observations where both and predicted and actual output is false (0).

False Positive - These are those observations where the predicted output is true (1) whereas the actual output is false (0).

False Negative – These are those observations where the predicted output is false (0) whereas the actual output is true (1).

$True\ Positive\ Rate\ (TPR) = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)}$
$False\ Positive\ Rate\ (FPR) = \frac{False\ Positive\ (FP)}{True\ Negative\ (TN) + False\ Positive\ (FP)}$
$Precision = \frac{True\ Positive\ (TP)}{True\ Positive(TP) + False\ Positive\ (FP)}$
$Recall = \frac{True\ Positive\ (TP)}{True\ Positive(TP) + False\ Negative\ (FN)}$
$F\ score = 2 * \frac{Precision * Recall}{Precision + Recall}$

Table 22 : TPR, FPR, Precision, Recall and F score

3.8.2 Accuracy

Accuracy score plays a vital role in our analysis, later in the thesis. Since we use this measure to compare various inputs and various models used.

$$Accuracy = \frac{True\ Positive\ (TP) + True\ Negative\ (TN)}{True\ Positive(TP) + True\ Negative(TN) + FalsePositive(FP) + False\ Negative(FN)}$$

Figure 6 : Accuracy formulae

3.8.3 Area under curve (AUC)

AUC represents degree or measure of separability. It tells how much model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. By analogy, Higher the AUC, better the model is at classification. ROC curve is plotted with TPR (True Positive Rate) against the FPR (False Positive Rate) where TPR is on y-axis and FPR is on the x-axis. Area under ROC curve is referred to as AUC.

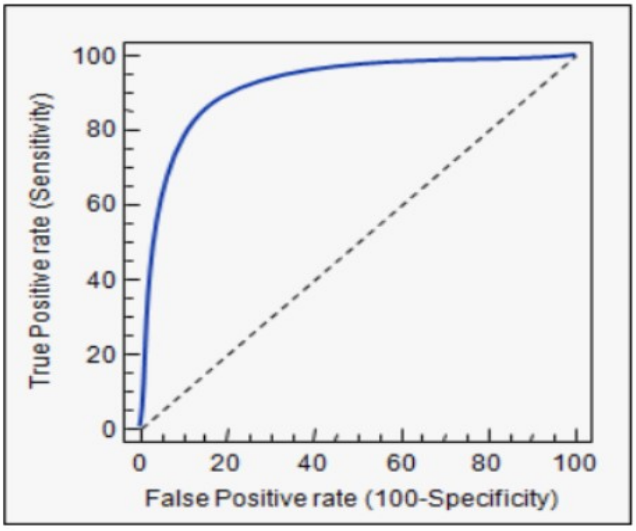


Figure 7 : Area under curve

3.8.4 Hypothesis Testing

It is used to check if different outputs(accuracies) are statistically similar or different at a desired confidence level.

Hypothesis Null Hypothesis - H0: Accuracy$_1$ (x_1) = Accuracy$_2$ (x_2) Alternative Hypothesis: Ha: Accuracy$_1$ (x_1) <> Accuracy$_2$ (x_2)
T test score $$t_{\frac{\propto}{2}},\ df = \frac{x_1 - x_2 - 0}{\sqrt{\left[\left\{\frac{(n_1-1)*(sd_1^2)+(n_2-1)*(sd_2^2)}{n_1+n_2-2}\right\}*\left\{\frac{1}{n_1}+\frac{1}{n_2}\right\}\right]}}$$ Where x_1 = average accuracy of 1st sample x_2 = average accuracy of 2nd sample sd_1 = standard deviation of 1st sample sd_2 = standard deviation of 2nd sample n_1 = sample size of 1st sample n_2 = sample size of 2nd sample t = t statistic of the model α = alpha or confidence level df = degrees of freedom

Table 23 : Hypothesis testing using t-test

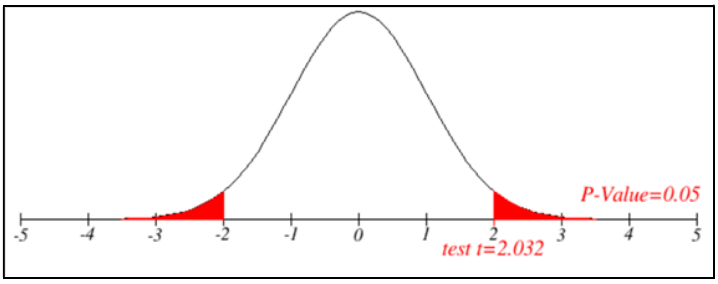


Figure 8 : Distribution and t-test at 95% confidence with (n_1 + n_2 -2) df (Lippman 2020)

Area marked by arrow is rejection are for two tailed z test at 95% confidence level.

3.8.5 Output Capture Methodology

Table 24 is the methodology to capture output

Sr. No.	Headers	Description	Sample Values
1	Model	Name of the model used	LR, NB, SVM, RF, XGB, DL
2	Measures	Name of the measure used to evaluate	Precision, Recall, F-Score, Support, Accuracy
3	Features	Name of the input variable used	norm_lemma_stopword, norm_stem_stopword, POS Tagging, Lemma_POS_Tagging
4	Train	Training score	1
5	Test	Testing score on hotel dataset.	0.867830
6	Restaurant	Testing score on restaurant dataset.	0.77
7	Doctor	Testing score on doctor dataset.	0.66
8	Parameters	Parameters used to achieve the above score	C=1000, kernel=rbf, gamma=0.01

Table 24: Output capture template design

Scope of the output = 6 Models X 5 Measures X 4 Features = 120 records of Output

4 RESULTS AND DISCUSSION

In this section, we present the output of our experiments of the various transformations applied to input vs each of the models utilized. We also present an analysis of the outcomes to derive best or balanced models and input transformations. Post which we present discussions which lays emphasis on the reasoning behind our derived outcomes.

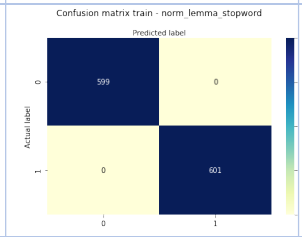
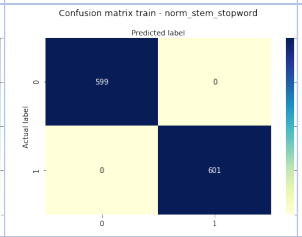
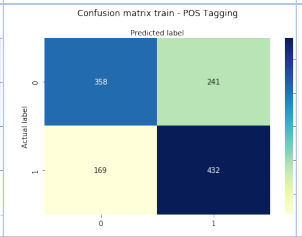
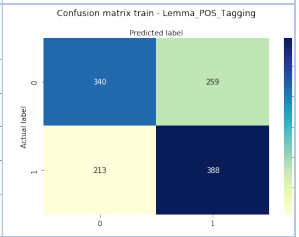
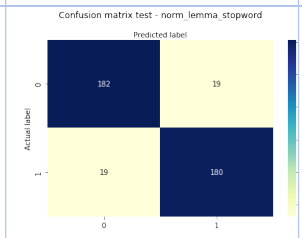
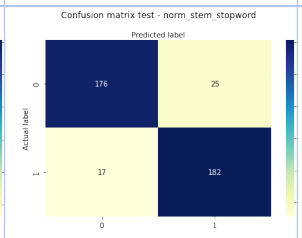
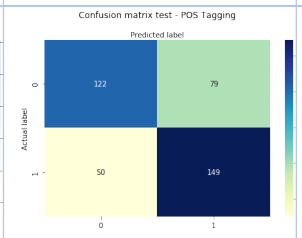
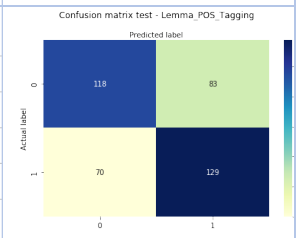
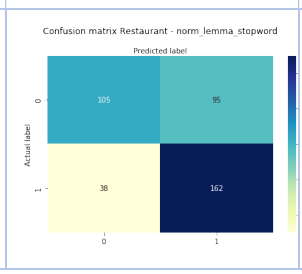
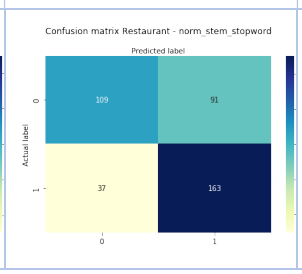
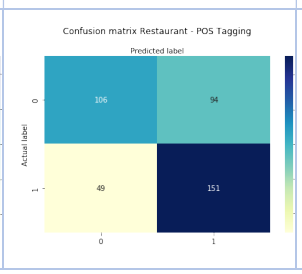
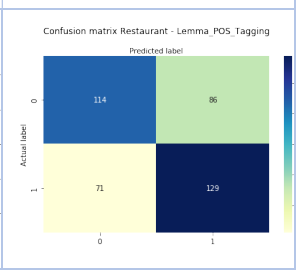
4.1 RESULTS

In this section, we present the results obtained from various models using the various transformed inputs. We use confusion matrix to represent the results from each of the models which depicts the true positives, true negatives, false positives, and false negatives. Also, we present the AUC (area under curve) to present the results.

4.1.1 Logistic regression

We present below in Table 25 the results of training and testing using Logistic regression on various transformations method applied on the input.

Logistic Regression

	Lemmatization	Stemmatization	POS Tagging	Lemmatization + POS Tagging
Hotel – Train				
Hotel – Test				
Restaurant – Test				

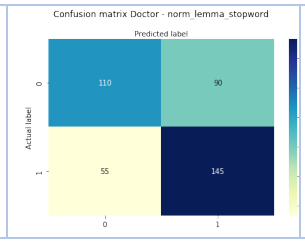
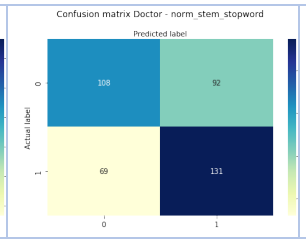
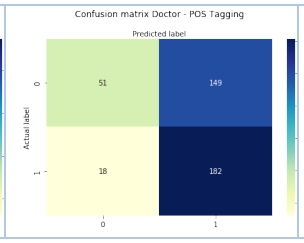
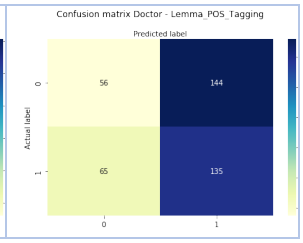
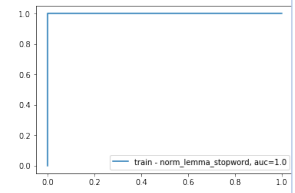
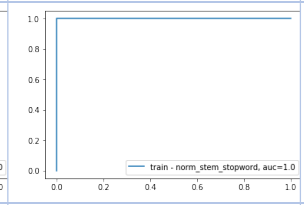
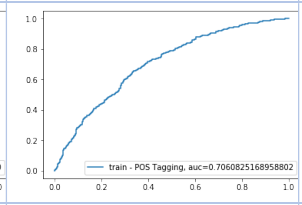
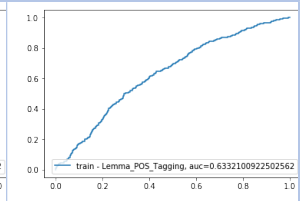
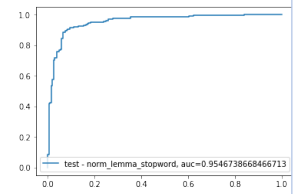
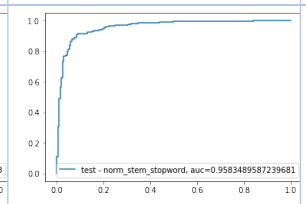
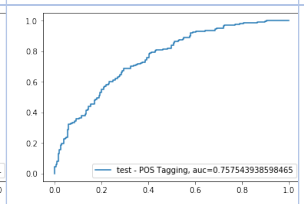
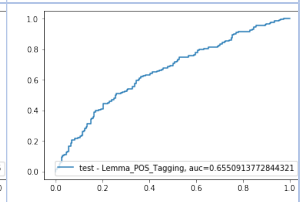
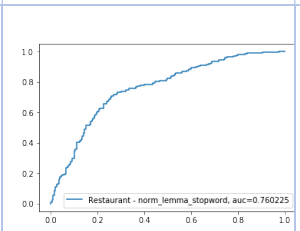
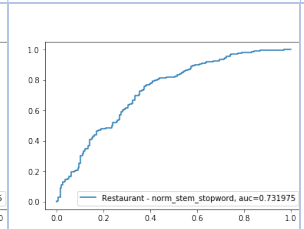
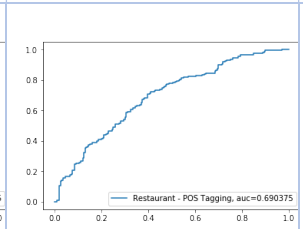
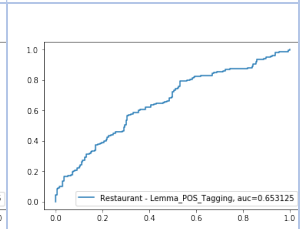
Doctor – Test				

Table 25 : Confusion matrix for all datasets/inputs used in Logistic Regression

Table 25 is a heatmap for the model logistic regression which is applied on all four inputs – normalized lemmatized words, normalized stemmed words, pos tagging of the reviews and a mix of lemmatization and pos reviews, across all kinds of data – training dataset of hotel reviews, test dataset of hotel reviews, restaurant reviews and doctors reviews. Logistic regression performed well and gave a strong performance of 91% and 90% for lemmatized input words and stemmed input words respectively on testing dataset of 25% of hotel reviews. It secured an accuracy of 68% and 67% for stemmed input words and lemmatized input words respectively on restaurant reviews whereas the performance declined to 64% and 60% for lemmatized and stemmed input on doctor dataset. The performance of logistic regression drops when the trained logistic regression is used on POS tagging input. It barely manages to secure an accuracy of 68% on testing dataset of hotel reviews, 64% on restaurant dataset and 58% on doctor dataset. Also, the input which is pos tagging on lemmatized textual reviews gave a weak performance for logistic regression. It's accuracy score are as follows – 62% on testing data of hotel reviews, 61% on restaurant reviews and 48% on doctors' reviews.

Logistic Regression				
	Lemmatization	Stemmatization	POS Tagging	Lemmatization + POS Tagging
Hotel – Train				
Hotel – Test				
Restaurant – Test				

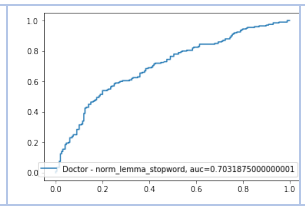
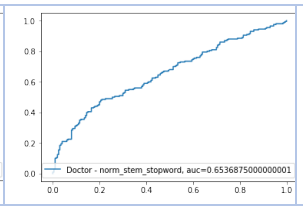
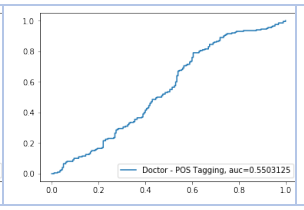
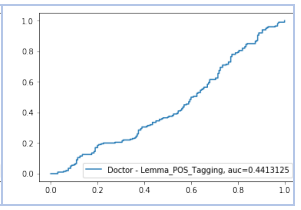
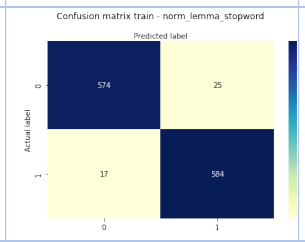
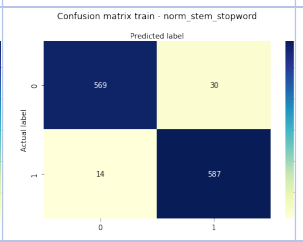
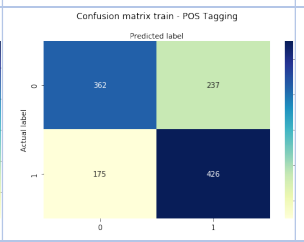
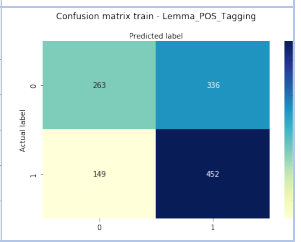
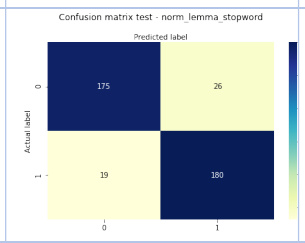
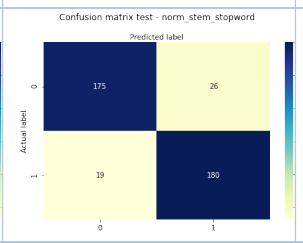
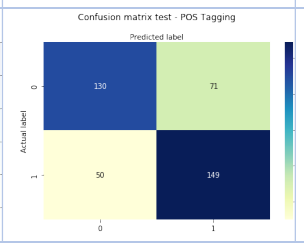
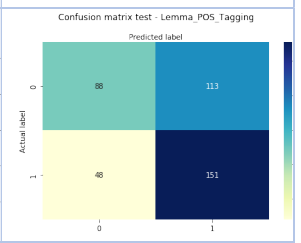
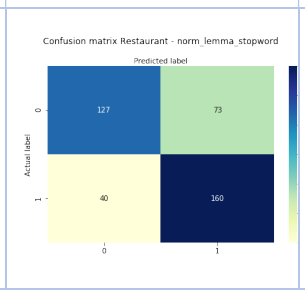
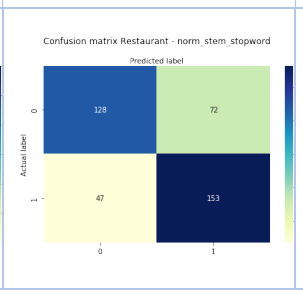
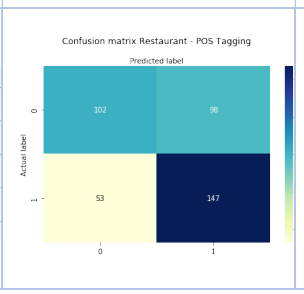
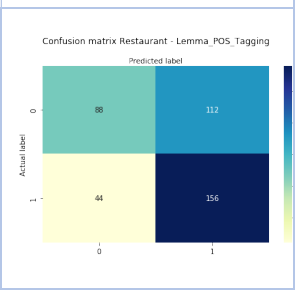
Doctor – Test				

Table 26 : AUC for all datasets/inputs used in Logistic regression

In Table 26 we can clearly see that area under curve declines as we move from testing dataset consisting of 25% of unseen data of hotel reviews to restaurant dataset and it reduces further on doctor dataset. This observation comes across in both lemmatized and stemmed inputs.
Clearly the observation is similar in POS tagging input but the decline from one dataset to another dataset is very minimum in POS tagging. In fact, the difference in AUC between testing dataset and restaurant dataset is negligible for POS tagging. However, the lemmatized with POS tagging gave the worst and high volatility performance since performance on doctors' dataset drastically drops to a very minimum.

4.1.2 Naïve Bayes

We present below in Table 27 the results of training and testing using Naïve Bayes on various transformations method applied on the input.

Naïve Bayes				
	Lemmatization	Stemmatization	POS Tagging	Lemmatization + POS Tagging
Hotel – Train				
Hotel – Test				
Restaurant – Test				

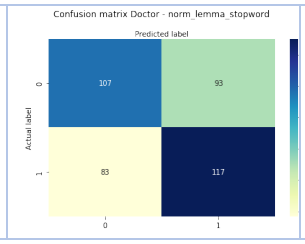
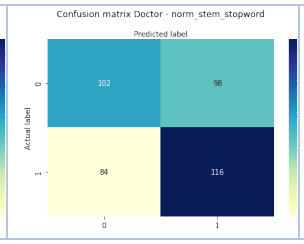
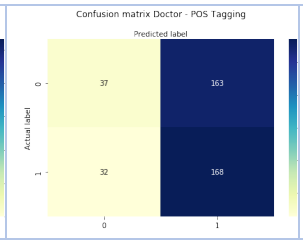
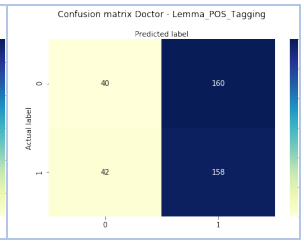
Doctor- Test				

Table 27 : Confusion matrix for all datasets/inputs used in Naive Bayes

Table 27 is heatmap for the model Naïve Bayes which is applied on all four inputs – normalized lemmatized words, normalized stemmed words, pos tagging of the reviews and lemmatized pos tagging, across all kinds of data – training dataset of hotel reviews, test dataset of hotel reviews, restaurant reviews and doctors reviews. Naïve Bayes performed remarkably well and gave a decently strong performance of 89% for both stemmed input words and lemmatized input words on testing dataset of 25% of hotel reviews. It secured an accuracy of 72% and 70% for lemmatized input words and stemmed input words respectively on restaurant reviews whereas the performance declined to 55% and 56% for stemmed and lemmatized input on doctor dataset. The performance of Naïve Bayes drops when the trained model is used on POS tagging input. It barely manages to secure an accuracy of 70% on testing dataset of hotel reviews, 62% on restaurant dataset and 51% on doctor dataset. Furthermore, the worst performance is from lemmatized pos tagging which just secured accuracy of 60% for 25% of testing hotel reviews, 61% for restaurant reviews and 50% on doctors' reviews.

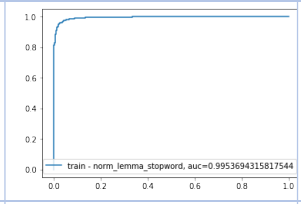
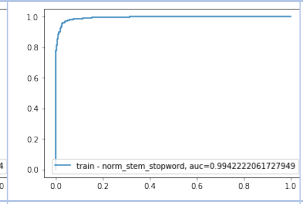
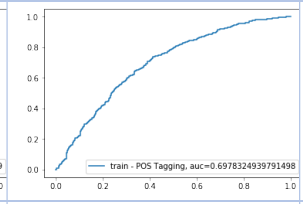
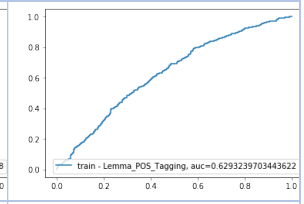
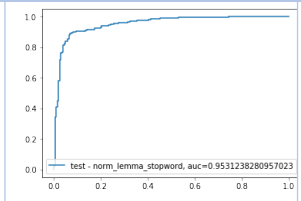
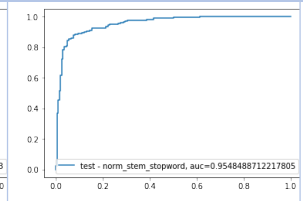
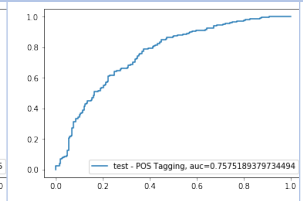
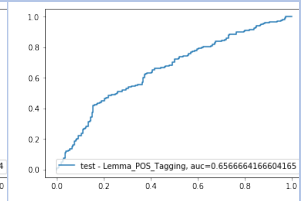
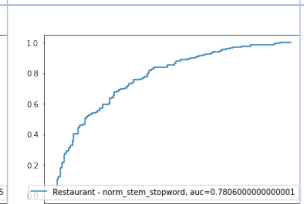
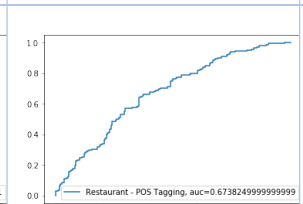
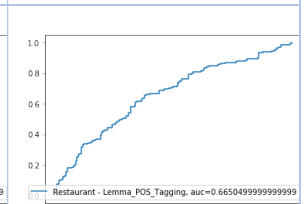
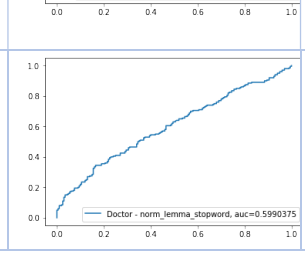
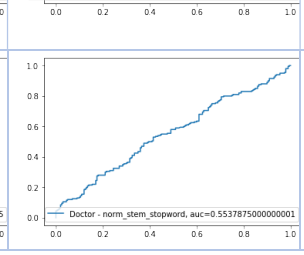
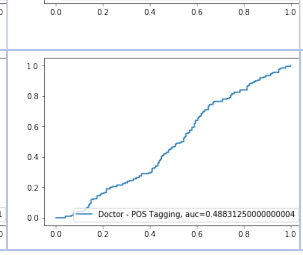
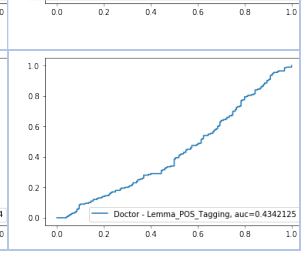
Naïve Bayes				
	Lemmatization	Stemmatization	POS Tagging	Lemmatization + POS Tagging
Hotel – Train				
Hotel – Test				
Restaurant – Test				
Doctor - Test				

Table 28 : AUC for all datasets/inputs used in Naive Bayes

In Table 28 we can clearly see that area under curve declines as we move from testing dataset consisting of 25% of unseen data of hotel reviews to restaurant dataset and if reduces further on doctor dataset. This observation clearly comes across in both lemmatized and stemmed inputs. Clearly the observation is similar in POS tagging input but the decline from one dataset to another dataset is less in POS tagging. The weakest performance is given by lemmatized pos tagging.

4.1.3 Support Vector Machine

We present below in Table 29 the results of training and testing using Support Vector Machine on various transformations method applied on the input.

Support Vector Machine

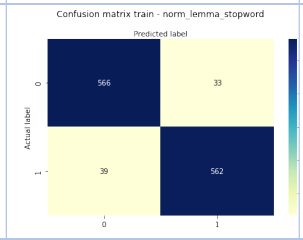
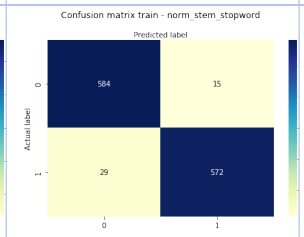
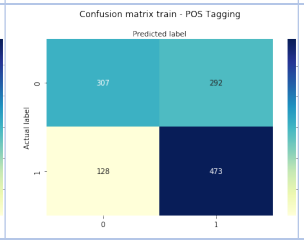
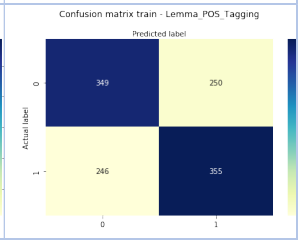
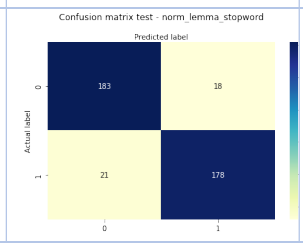
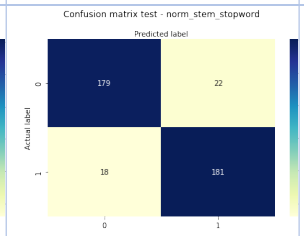
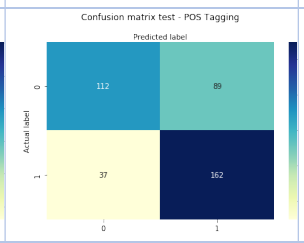
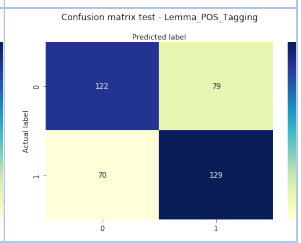
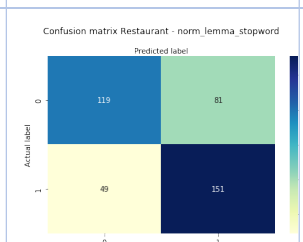
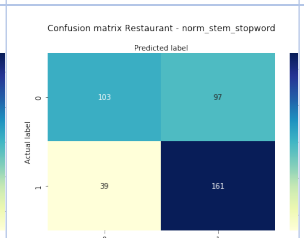
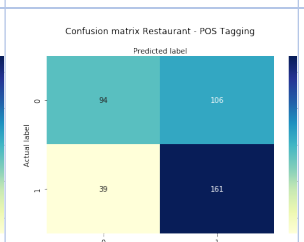
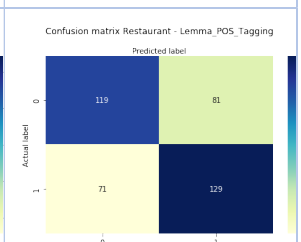
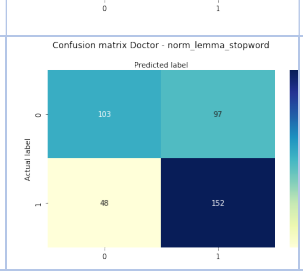
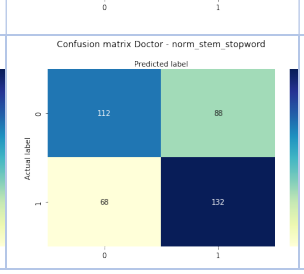
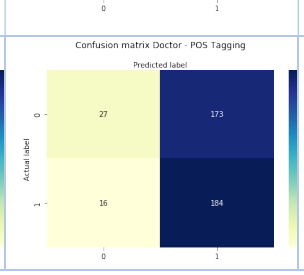
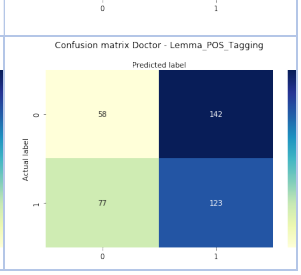
	Lemmatization	Stemmatization	POS Tagging	Lemmatization + POS Tagging
Hotel – Train				
Hotel – Test				
Restaurant – Test				
Doctor- Test				

Table 29 : Confusion matrix for all datasets/inputs used in Support Vector Machine

In table 29 is heatmap for the model support vector machine which is applied on all four inputs – normalized lemmatized words, normalized stemmed words, pos tagging and lemmatized POS tagging of the reviews, across all kinds of data – training dataset of hotel reviews, test dataset of hotel reviews, restaurant reviews and doctors reviews. Support vector machine performed well and gave a strong performance of 90% for both stemmed input words and lemmatized input words respectively

on testing dataset of 25% of hotel reviews. It secured an accuracy of 66% and 68% for stemmed input words and lemmatized input words respectively on restaurant reviews whereas the performance declined to 61% and 64% for stemmed and lemmatized input on doctor dataset. The performance of support vector machine drops when the trained model is used on POS tagging input. It barely manages to secure an accuracy of 69% on testing dataset of hotel reviews, 64% on restaurant dataset and 53% on doctor dataset. Also, lemmatized pos tagging just merely secures 63% on hotel test reviews, 62% on restaurant reviews and 45% on doctors' reviews.

Support Vector Machine

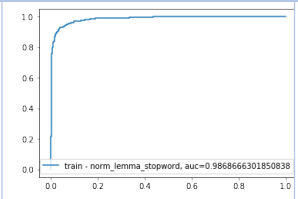
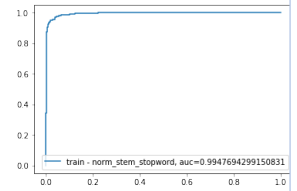
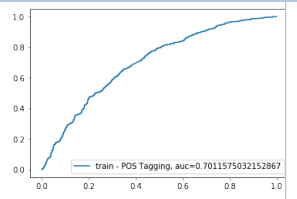
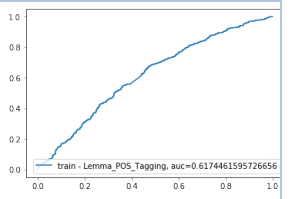
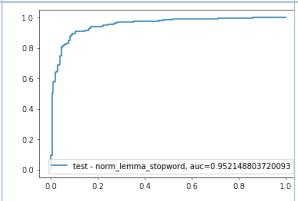
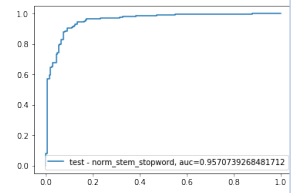
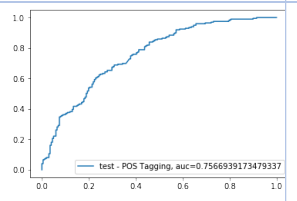
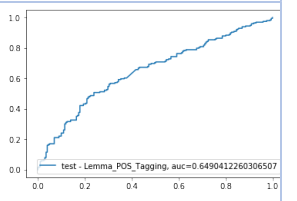
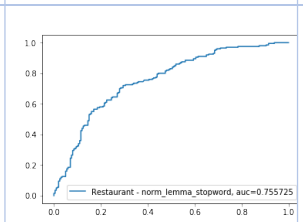
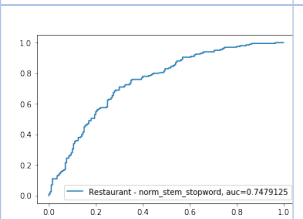
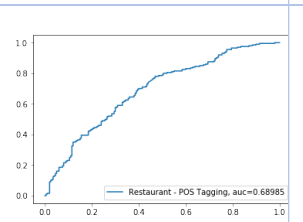
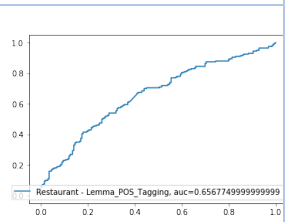
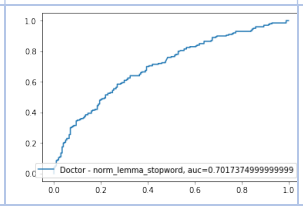
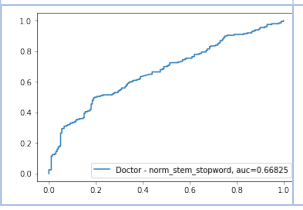
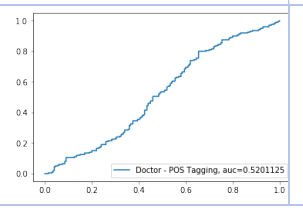
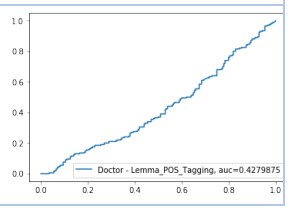
	Lemmatization	Stemmatization	POS Tagging	Lemmatization + POS Tagging
Hotel – Train				
Hotel – Test				
Restaurant – Test				
Doctor – Test				

Table 30 : AUC for all datasets/inputs used in Support Vector Machine

In table 30 we can clearly see that area under curve steeply declines as we move from testing dataset consisting of 25% of unseen data of hotel reviews to restaurant dataset and it reduces further on doctor dataset. This observation clearly comes across in both lemmatized and stemmed inputs. Clearly the observation is similar in POS tagging input but the decline from one dataset to another dataset is less in POS tagging. In fact, the difference in AUC between testing dataset and restaurant dataset is quite small for POS tagging. Here too, lemmatized pos tagging's performance is weakest and there is negligible difference in accuracy of testing data reviews of hotels and restaurant, however the accuracy of doctors' review takes a sharp dip to just an embarrassingly 45%.

4.1.4 Random Forest

We present below in Table 31 the results of training and testing using Random Forest on various transformations method applied on the input.

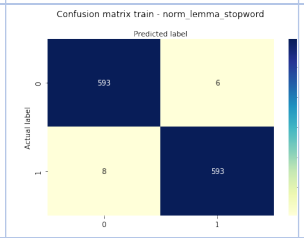
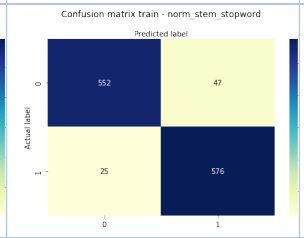
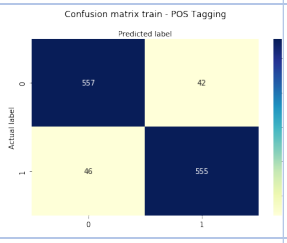
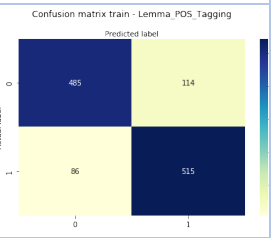
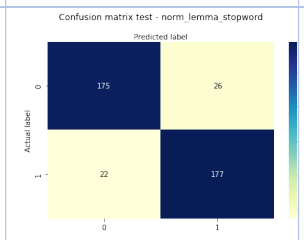
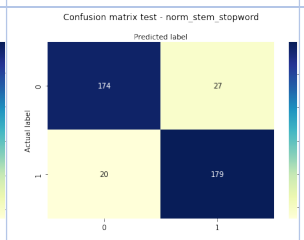
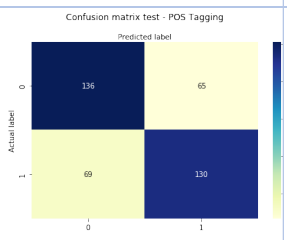
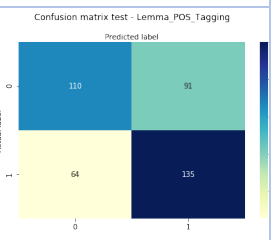
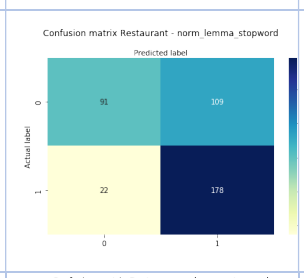
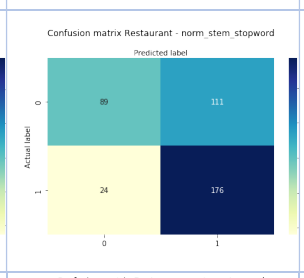
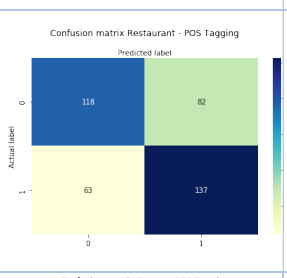
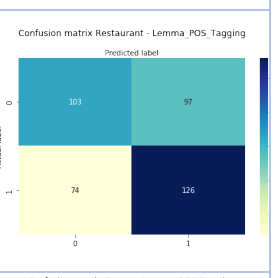
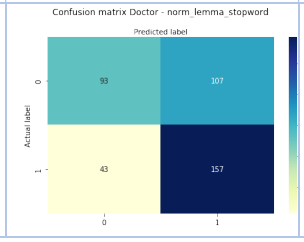
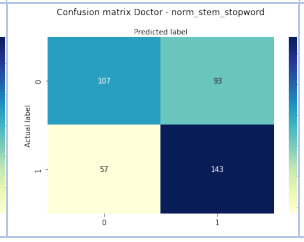
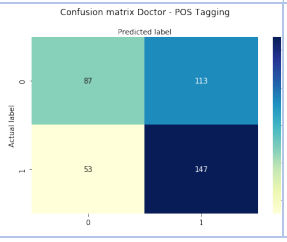
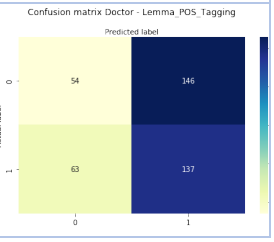
Random Forest				
	Lemmatization	Stemmatization	POS Tagging	Lemmatization + POS Tagging
Hotel – Train				
Hotel – Test				
Restaurant – Test				
Doctor- Test				

Table 31 : Confusion matrix for all datasets/inputs used in Random Forest

Table 31 is heatmap for the model random forest which is applied on all four inputs – normalized lemmatized words, normalized stemmed words, pos tagging and lemmatized pos tagging of the reviews, across all kinds of data – training dataset of hotel reviews, test dataset of hotel reviews, restaurant reviews and doctors reviews. Random forest performed well and gave a robust performance of 88% for both stemmed input words and lemmatized input words respectively on testing dataset of 25% of hotel reviews. It secured an accuracy of 66% and 67% for stemmed input words and lemmatized input words respectively on restaurant reviews whereas the performance declined to 63% for both stemmed and lemmatized input on doctor dataset. The performance of random forest drops when the model utilizes POS tagging input. It barely manages to secure an accuracy of 67% on testing dataset of hotel reviews, 64% on restaurant dataset and 59% on doctor dataset. Furthermore, the weakest performance is achieved by lemmatized pos tagging which barely secures 61% on testing data of hotel reviews, 57% on restaurant reviews and 48% for doctors' reviews.

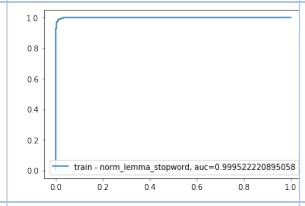
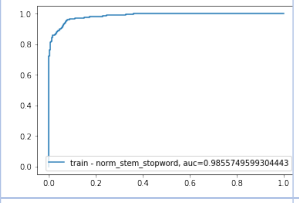
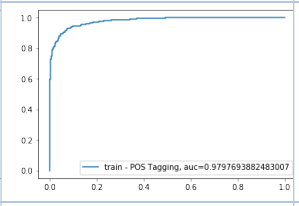
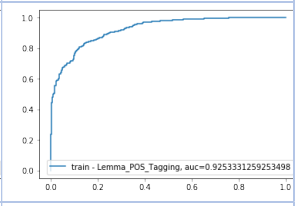
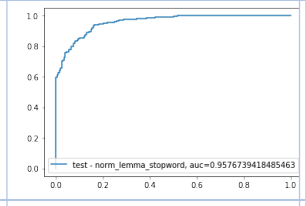
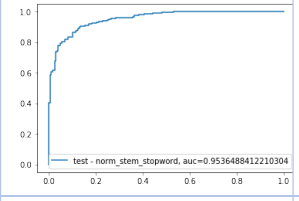
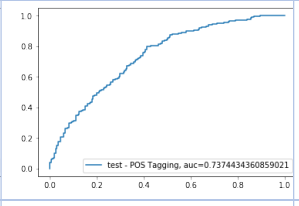
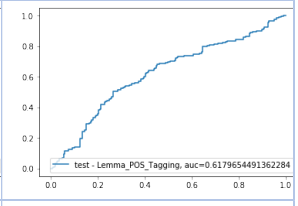
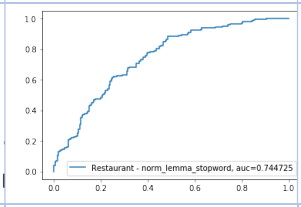
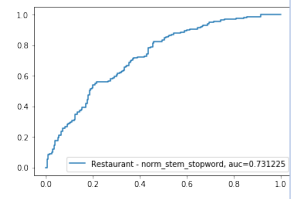
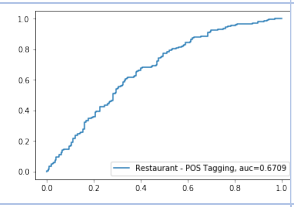
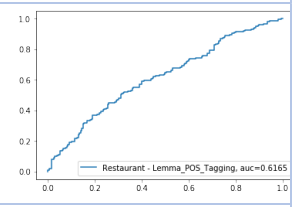
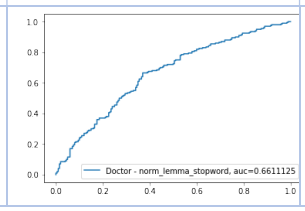
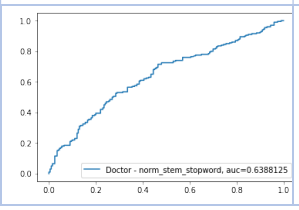
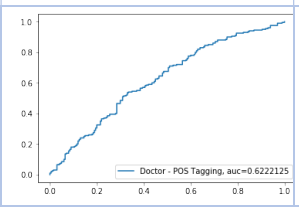
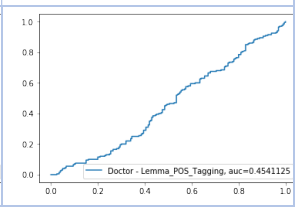
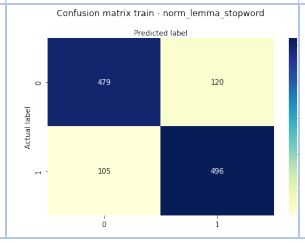
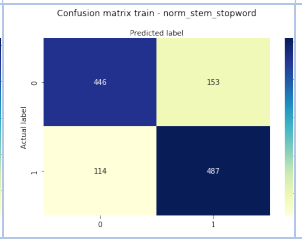
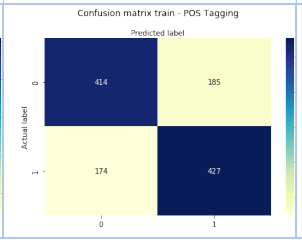
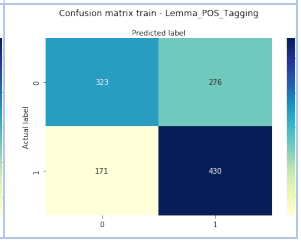
Random Forest				
	Lemmatization	Stemmatization	POS Tagging	Lemmatization + POS Tagging
Hotel – Train				
Hotel – Test				
Restaurant –				
Doctor – Test				

Table 32 : AUC for all datasets/inputs used in Random Forest

In table 32 we can clearly see that area under curve declines as we move from testing dataset consisting of 25% of unseen data of hotel reviews to restaurant dataset and if reduces further on doctor dataset. This observation comes across in both lemmatized and stemmed inputs. Clearly the observation is similar in POS tagging input but the decline from one dataset to another dataset is less in POS tagging. Performance of lemmatized pos tagging is again the weakest.

4.1.5 Extreme Gradient Boosting

We present below in Table 33 the results of training and testing using Extreme Gradient Boosting on various transformations method applied on the input.

Extreme Gradient Boosting				
	Lemmatization	Stemmatization	POS Tagging	Lemmatization + POS Tagging
Hotel – Train				

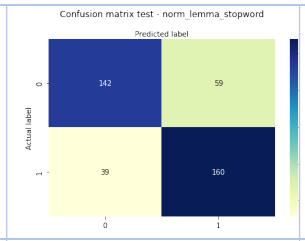
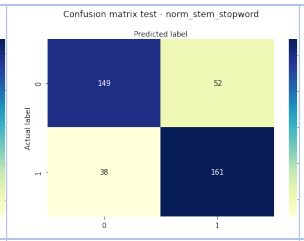
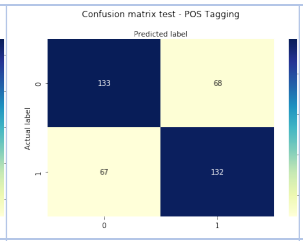
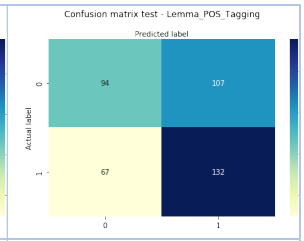
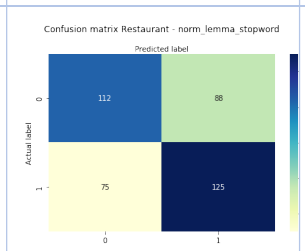
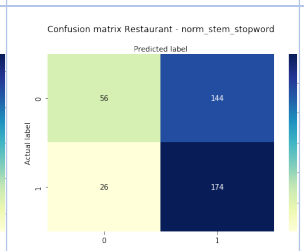
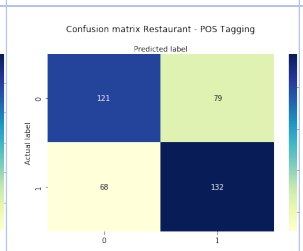
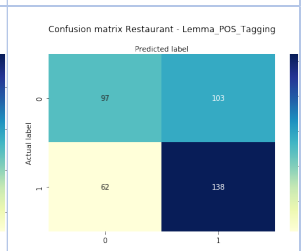
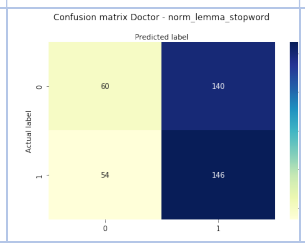
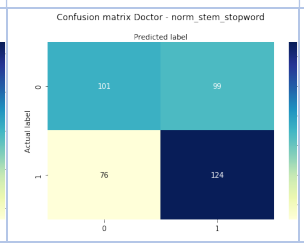
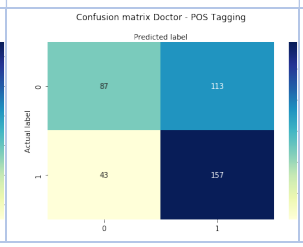
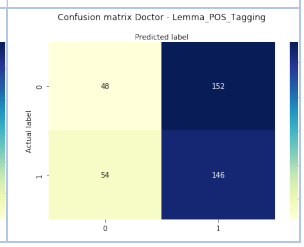
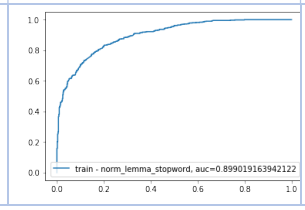
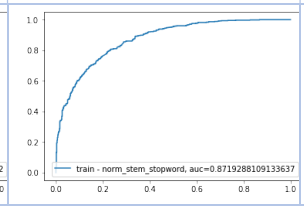
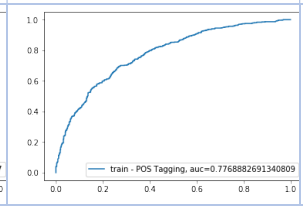
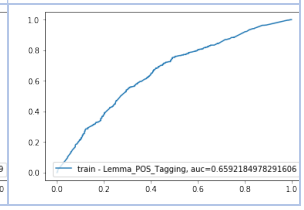
Hotel – Test				
Restaurant - Test				
Doctor- Test				

Table 33 : Confusion matrix for all datasets/inputs used in Extreme Gradient Boosting

Table 33 is heatmap for the model extreme gradient boosting which is applied on all four inputs – normalized lemmatized words, normalized stemmed words, pos tagging of the reviews and a mix of lemmatization and POS tagging, across all kinds of data – training dataset of hotel reviews, test dataset of hotel reviews, restaurant reviews and doctors reviews. Extreme gradient boosting performed moderately and gave a performance of 78% and 76% for stemmed input words and lemmatized input words respectively on testing dataset of 25% of hotel reviews. It secured an accuracy of 58% and 59% for stemmed input words and lemmatized input words respectively on restaurant reviews whereas the performance declined to 56% and 52% for stemmed and lemmatized input respectively on doctor dataset. The performance of extreme gradient boosting drops when the trained model is used on POS tagging input. It barely manages to secure an accuracy of 66% on testing dataset of hotel reviews, 63% on restaurant dataset and 61% on doctor dataset. The accuracy further drops to 57% on testing data of hotel reviews, 59% on restaurant reviews and 49% on doctors dataset for lemmatized POS Tagging.

Extreme Gradient Boosting				
	Lemmatization	Stemmatization	POS Tagging	Lemmatization + POS Tagging
Hotel - Train				

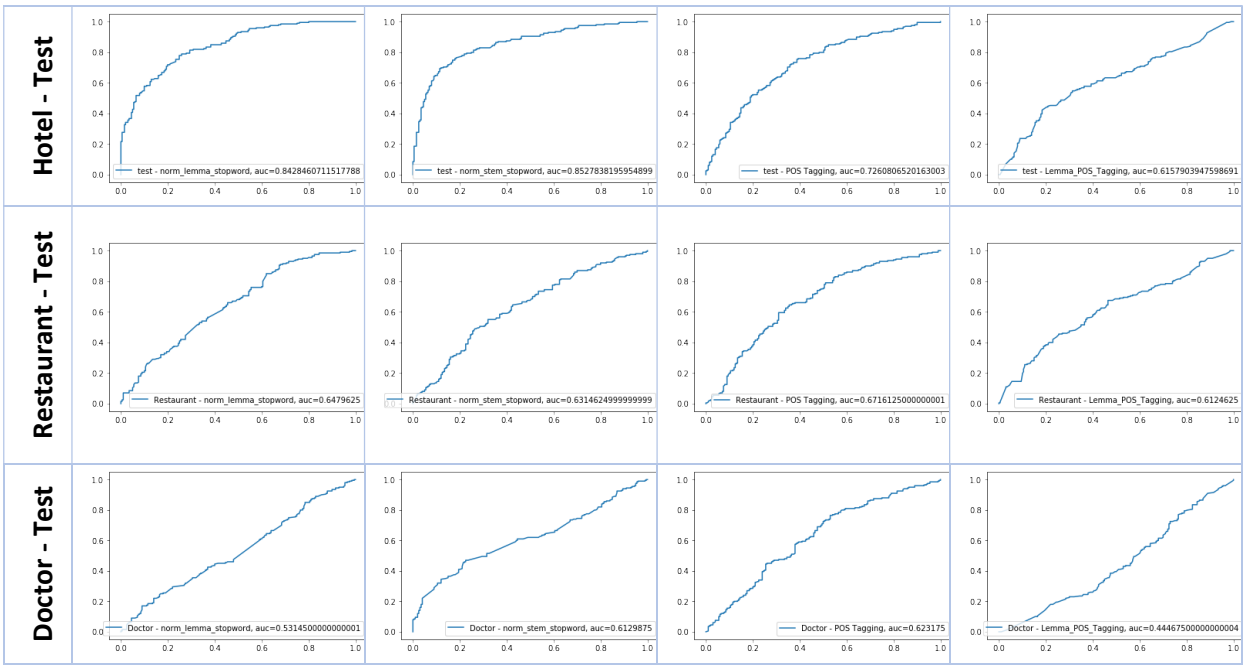
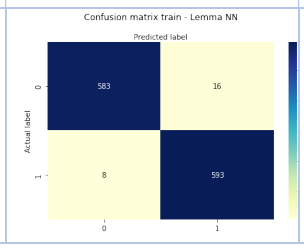
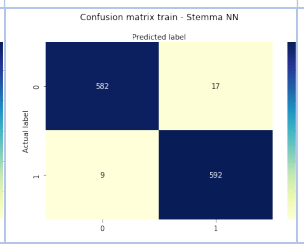
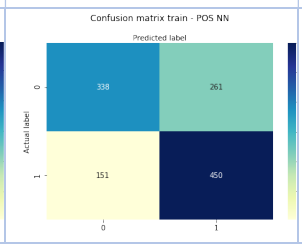
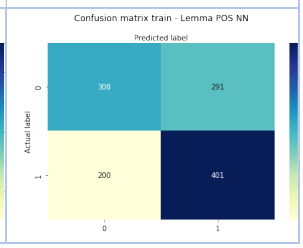


Table 34 : AUC for all datasets/inputs used in Extreme Gradient Boosting

In table 34 we can clearly see that area under curve declines as we move from testing dataset consisting of 25% of unseen data of hotel reviews to restaurant dataset. But strikingly, there is a negligible decline in performance from restaurant dataset to doctor dataset. This observation clearly comes across in both lemmatized and stemmed inputs. In POS tagging input the performance of the model trained with extreme gradient boosting is similar for all the datasets with very minimal drops at every stage – testing dataset of hotel reviews, restaurant dataset and doctor dataset. The overall accuracy of lemmatized POS Tagging is weakest compared to the rest.

4.1.6 Deep Learning - Neural Network

We present below in Table 35 the results of training and testing using Deep Learning Neural Network on various transformations method applied on the input.

Neural Network				
	Lemmatization	Stemmatization	POS Tagging	Lemmatization + POS Tagging
Hotel – Train				

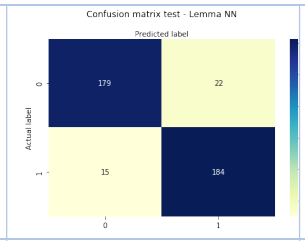
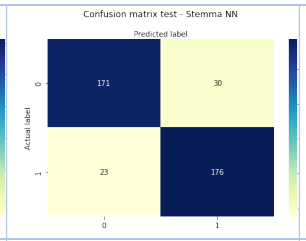
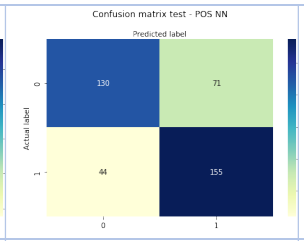
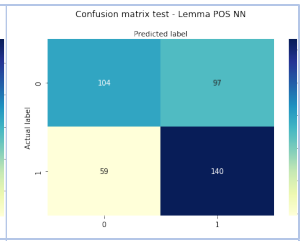
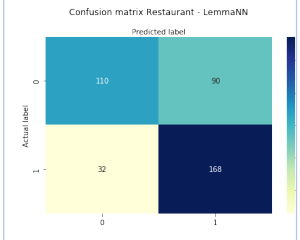
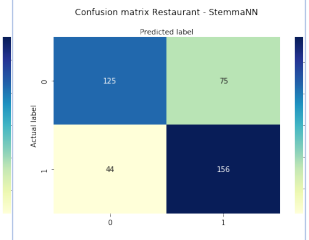
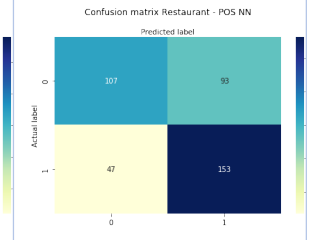
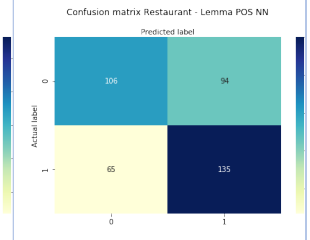
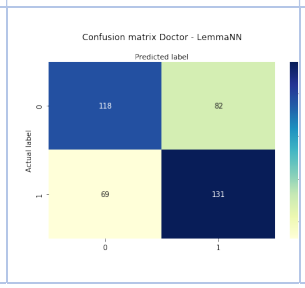
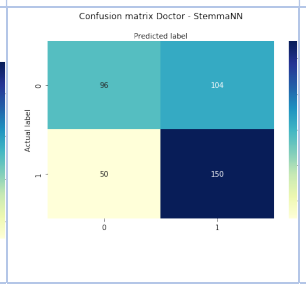
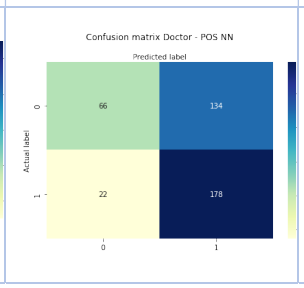
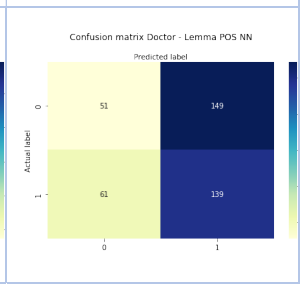
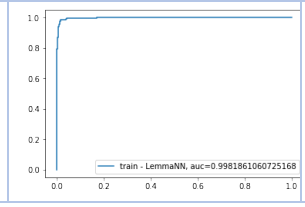
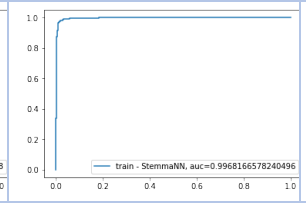
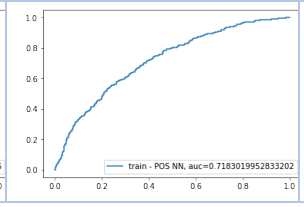
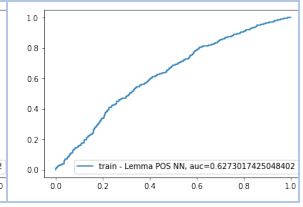
Hotel – Test				
Restaurant – Test				
Doctor- Test				

Table 35 : Confusion matrix for all datasets/inputs used in Neural Network

Table 35 is heatmap for the deep learning neural network model which is applied on all four inputs – normalized lemmatized words, normalized stemmed words, pos tagging of the reviews and a mix of lemmatization and POS Tagging, across all kinds of data – training dataset of hotel reviews, test dataset of hotel reviews, restaurant reviews and doctors reviews. Neural Network model performed well and gave the best performance of 91% for lemmatized input words and it secured an accuracy of 87% for stemmed input words on testing dataset of 25% of hotel reviews. It secured an accuracy of 70% for both stemmed input words and lemmatized input words on restaurant reviews whereas the performance declined to 62% for both stemmed and lemmatized input on doctor dataset. The performance of neural network drops when the trained model is used on POS tagging input. It barely manages to secure an accuracy of 59% on testing dataset of hotel reviews, 65% on restaurant dataset and 61% on doctor dataset. Unlike the other models, accuracy of lemmatized POS Tagging is better than accuracy of POS Tagging. Lemmatized POS Tagging secured an accuracy of 61% on testing data of hotel reviews. It secures an accuracy of 60% and 48% on restaurant and doctors' dataset.

Neural Network				
	Lemmatization	Stemmatization	POS Tagging	Lemmatization + POS Tagging
Hotel – Train				

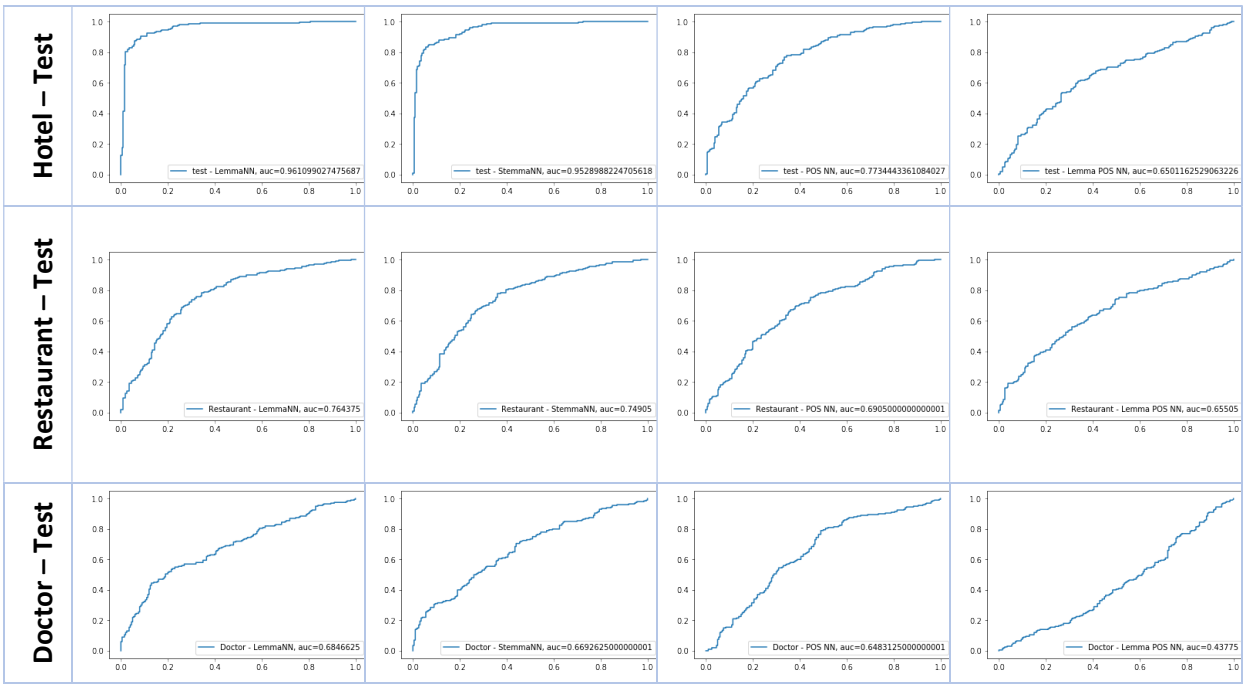


Table 36 : AUC for all datasets/inputs used in Neural Network

In table 36 we can clearly see that area under curve declines as we move from testing dataset consisting of 25% of unseen data of hotel reviews to restaurant dataset. But strikingly, there is a negligible decline in performance from restaurant dataset to doctor dataset. This observation clearly comes across in both lemmatized and stemmed inputs. In POS tagging input the performance of the model trained with extreme gradient boosting is similar for all the datasets with very minimal drops at every stage – testing dataset of hotel reviews, restaurant dataset and doctor dataset. The overall accuracy of lemmatized POS Tagging is weakest compared to the rest.

4.2 ANALYSIS OF RESULTS

In this section, we present analysis of the results received from various experimentation.

4.2.1 Lemmatized vs Stemmed

In this section, we analyze to find out how the change in input affects the output (i.e., accuracy of the prediction). For this, we perform hypothesis testing of output of 25% of unseen hotel reviews, restaurant reviews and doctors' reviews for a pair of input. Based on statistical evaluation using t test, we conclude which input parameters to focus on and group some of the similar inputs.

Null hypothesis:
Accuracy of Lemmatized input of all models for all test data = Accuracy of Stemmed Input of all models for all test data

Alternative hypothesis:
Accuracy of Lemmatized input of all models for all test data <> Accuracy of Stemmed Input of all models for all test data

25% of unseen hotel reviews + restaurant + doctor	Lemmatized	Stemmed	Degrees of freedom	T statistic	T score @95%, 34 *df*	Status
Mean	0,7142	0,7081	34	0,1437	2,0320	Accept Null
Standard Deviation	0,1285	0,1266				
Number	18	18				

Table 37 : Hypothesis testing of lemmatization and stemming

It is concluded that the null hypothesis is not rejected. Therefore, there is not enough evidence to claim that the output population proportion of lemmatized input is different than stemmed input, at the 0.05 significance level.

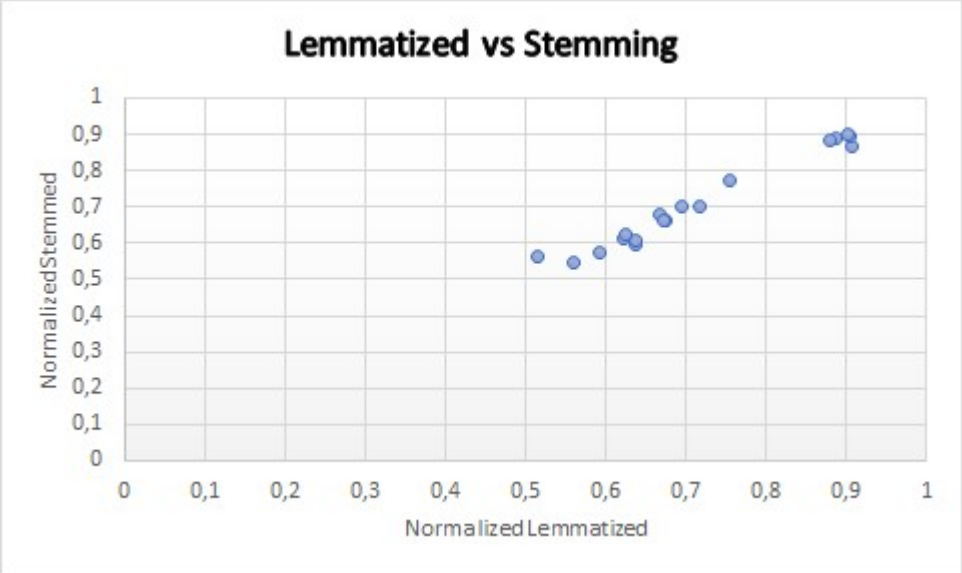


Figure 9: Plot of accuracy for lemmatization vs stemming

Correlation co-efficient of accuracies of lemmatized input against stemmed input is 0.9867. Hence, we ignore the stemmed input and only consider lemmatized input further in our analysis since they are statistically similar.

4.2.2 POS Tagging vs Lemmatized POS Tagging

Null hypothesis:
Accuracy of POS tagging input of all models for all test data = Accuracy of lemmatized POS tagging input of all models for all test data

Alternative hypothesis:
Accuracy of POS tagging input of all models for all test data <> Accuracy of lemmatized POS tagging Input of all models for all test data

25% of unseen hotel reviews + restaurant + doctor	POS Tagging	Lemmatized POS Tagging	Degrees of freedom	T statistic	T score @95%, 34 *df*	Status
Mean	0,6235	0,5607	34	3,2908	2,0320	Reject Null
Standard Deviation	0,0506	0,0632				
Numbers	18	18				

Table 38 : Hypothesis testing of POS Tagging and Lemmatization POS Tagging

It is concluded that the null hypothesis is rejected. Therefore, there is enough evidence to claim that the output population proportion of POS tagging is different than lemmatized POS tagging, at the 0.05 significance level. Hence, we treat POS tagging and lemmatized POS tagging separately.

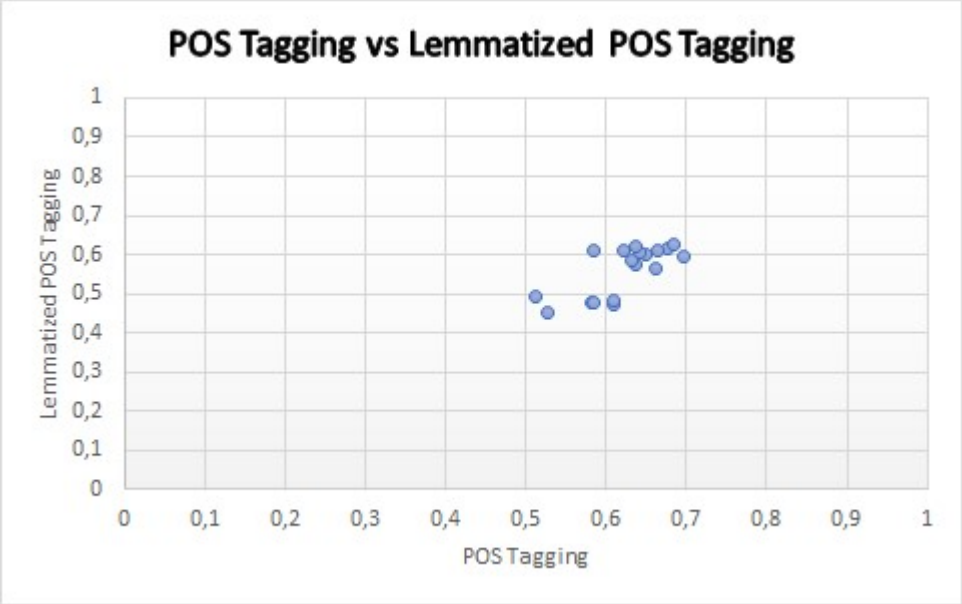


Figure 10: Plot of accuracy for POS Tagging and Lemmatization POS Tagging

Correlation co-efficient of POS Tagging against Lemmatized POS Tagging is 0.7358. We consider both POS Tagging and lemmatized POS Tagging in our analysis since their outcomes are statistically different at 95% confidence interval and the correlation coefficient of their accuracy outcomes is less than 75%.

4.2.3 Lemmatized vs POS Tagging

Null hypothesis:
Accuracy of Lemmatized input of all models for all test data = Accuracy of POS tagging input of all models for all test data

Alternative hypothesis:
Accuracy of Lemmatized input of all models for all test data <> Accuracy of POS tagging Input of all models for all test data

25% of unseen hotel reviews + restaurant + doctor	Lemmatized	POS Tagging	Degrees of freedom	T statistic	T score @95%, 34 *df*	Status
Mean	0,7142	0,6235	34	2,7866	2,0320	Reject Null
Standard Deviation	0,1285	0,0506				
Numbers	18	18				

Table 39 : Hypothesis testing of Lemmatization and POS Tagging

It is concluded that the null hypothesis is rejected. Therefore, there is enough evidence to claim that the output population proportion of lemmatized input is different than POS tagging, at the 0.05 significance level. Hence, we treat lemmatized input and POS tagging separately

.

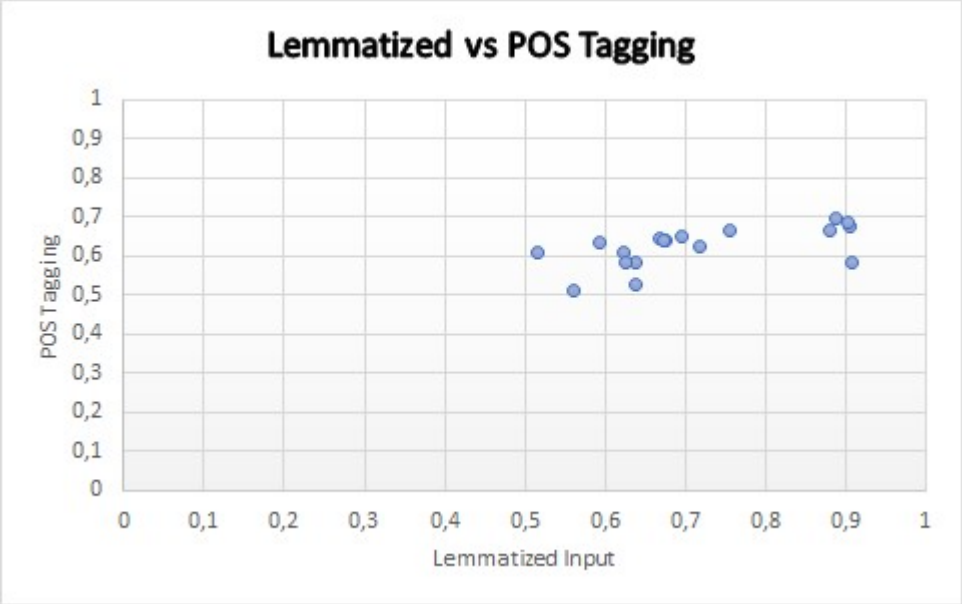


Figure 11: Plot of accuracy for lemmatization vs POS Tagging

Correlation co-efficient of accuracies of lemmatized input against POS Tagging is 0.6072. We consider both Lemmatized and POS Tagging in our analysis since their outcomes are statistically different in many cases at 95% confidence interval and the correlation coefficient of their accuracy outcomes is less than 75%.

4.2.4 Lemmatized vs Lemmatized POS Tagging

Null hypothesis:
Accuracy of Lemmatized input of all models for all test data = Accuracy of Lemmatized POS tagging input of all models for all test data

Alternative hypothesis:
Accuracy of Lemmatized input of all models for all test data <> Accuracy of Lemmatized POS tagging Input of all models for all test data

25% of unseen hotel reviews + restaurant + doctor	Lemmatized	Lemmatized POS Tagging	Degrees of freedom	T statistic	T score @95%, 34 *df*	Status
Mean	0,7142	0,5607	34	4,5476	2,0320	Reject Null
Standard Deviation	0,1285	0,0632				
Number	18	18				

Table 40: Hypothesis testing of Lemmatized and Lemmatized POS Tagging

It is concluded that the null hypothesis is rejected. Therefore, there is enough evidence to claim that the output population proportion of lemmatized input is different than lemmatized POS tagging, at the 0.05 significance level. Hence, we treat lemmatized input and lemmatized POS tagging separately.

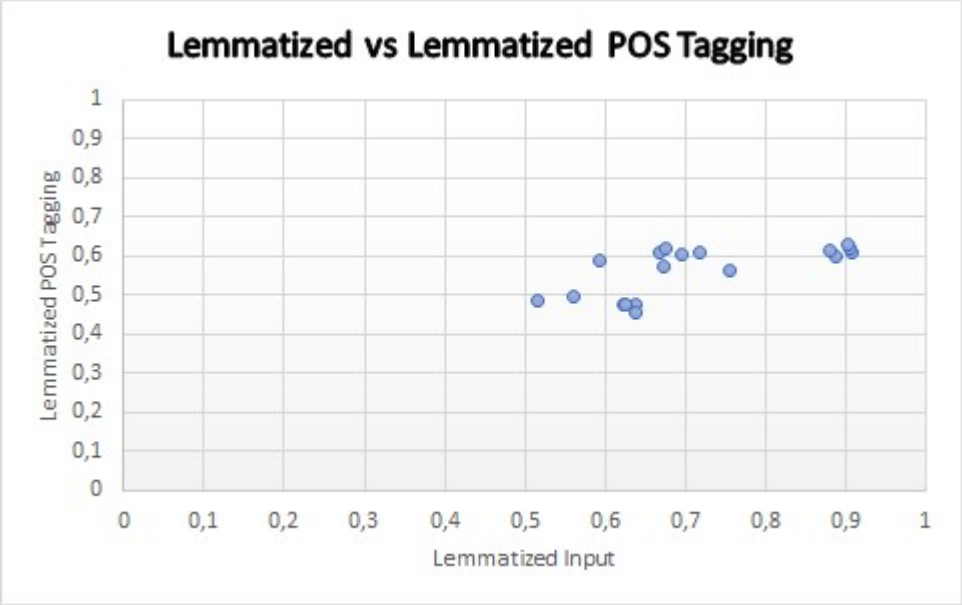


Figure 12: Plot of accuracies for Lemmatized vs Lemmatized POS Tagging

Correlation co-efficient of accuracies of lemmatized input against Lemmatized POS Tagging is 0.6837. We consider both Lemmatized and Lemmatized POS Tagging in our analysis since their outcomes are statistically different at 95% confidence interval and the correlation coefficient of their accuracy outcomes is less than 75%.

4.2.5 Comparison of Inputs and Models

Since lemmatized and stemmed inputs provides statistically similar output at 95% confidence interval, we drop stemmed input from further research and continue our research with lemmatized input and other inputs like POS tagging and lemmatized POS tagging.

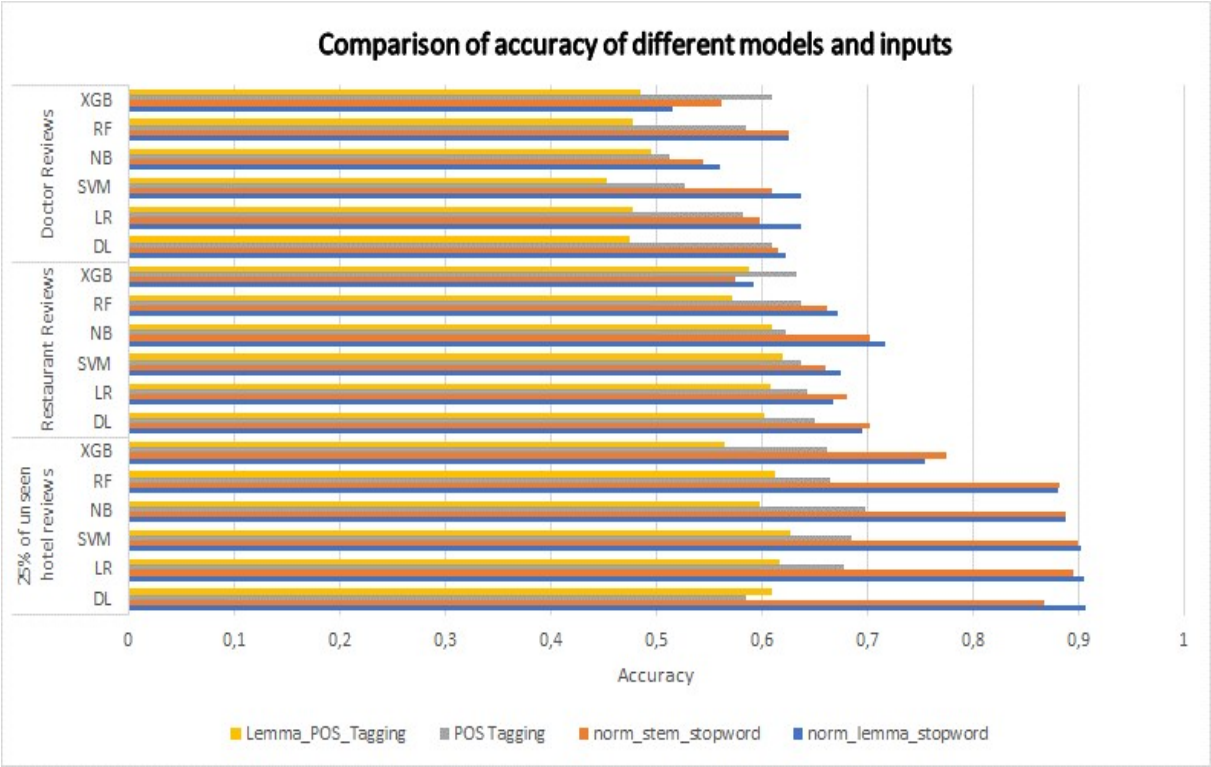


Figure 13: Comparison of accuracies of different models for different inputs for different datasets.

Now, we observe in the figure 13 that accuracy of prediction of lemmatized input is always better or at least the same as lemmatized POS tagging. Hence, we drop lemmatized POS tagging from further analysis since it doesn't improve the results (accuracy). So, we are left with just two input method – Lemmatized input and POS Tagging, utilizing 6 models – Logistic Regression, Naïve Bayes, Support vector, Random Forest, Extreme Gradient Boosting and Neural Network.

Values	Features\Model	DL	LR	SVM	NB	RF	XGB
25% of unseen hotel reviews	Lemmatized	0,9075	0,9050	0,9025	0,8875	0,8800	0,7550
	POS Tagging	0,5850	0,6775	0,6850	0,6975	0,6650	0,6625
Restaurant Reviews	Lemmatized	0,6950	0,6675	0,6750	0,7175	0,6725	0,5925
	POS Tagging	0,6500	0,6425	0,6375	0,6225	0,6375	0,6325
Doctor Reviews	Lemmatized	0,6225	0,6375	0,6375	0,5600	0,6250	0,5150
	POS Tagging	0,6100	0,5825	0,5275	0,5125	0,5850	0,6100

Table 41: Tabular representation of accuracies obtained using Lemmatization and POS Tagging.

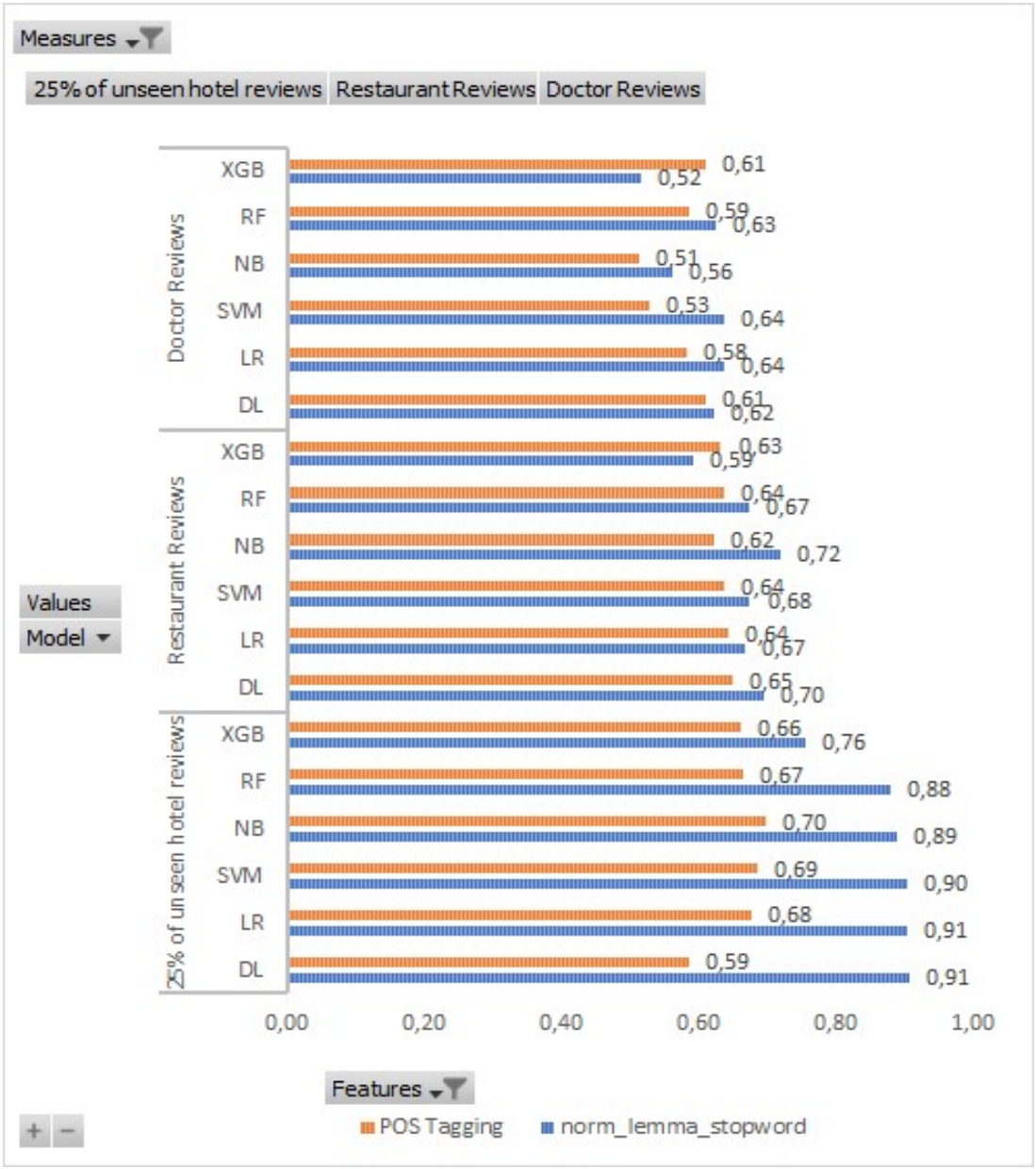


Figure 14: Comparison of accuracy of lemmatized input vs POS Tagging for all models

We observer that accuracy of Extreme Gradient Boosting is one of the lowest or lower than most in majority of cases. Hence, we decide not to continue with extreme gradient boosting to continue in our research. Similarly, we also observe that accuracy of POS Tagging is lower is most cases in comparison to accuracy of lemmatized texts. Hence, we also decide to drop POS tagging from further research. Thus, we are left with lemmatized texts as the only input since it provides the best accuracies for most models of the other option that we have explored. Also, we are left with outputs of five models.

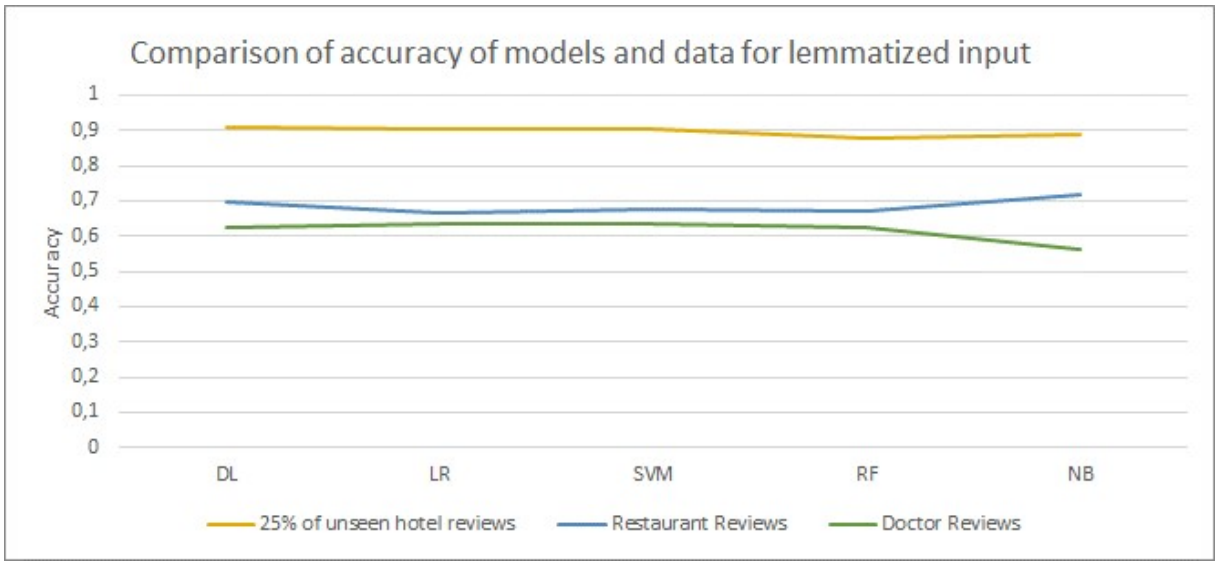


Figure 15: Comparison of accuracy of lemmatized input for top 5 models

We observe that output of all five models – Deep Learning Neural Network, Logistic regression, Naïve Bayes, Support Vector and Random Forest are similar for all three-testing data of hotel reviews, restaurant reviews and doctor reviews. If we further deep dive in the figure 15, we notice that Naïve Bayes is slightly more volatile than the rest of the four models. The accuracy of doctors' reviews for Naïve Bayes drops more in comparison of the rest.

Input	Model	Hotel Test	Restaurant	Doctor
Lemmatized	DL	0,9075	0,695	0,6225
	LR	0,905	0,6675	0,6375
	SVM	0,9025	0,675	0,6375
	RF	0,88	0,6725	0,625
	NB	0,8875	0,7175	0,56
	XGB	0,755	0,5925	0,515
Stemmed	DL	0,8675	0,7025	0,615
	LR	0,895	0,68	0,5975
	SVM	0,9	0,66	0,61
	RF	0,8825	0,6625	0,625
	NB	0,8875	0,7025	0,545
	XGB	0,775	0,575	0,5625
POS Tagging	DL	0,585	0,65	0,61
	LR	0,6775	0,6425	0,5825
	SVM	0,685	0,6375	0,5275
	RF	0,665	0,6375	0,585
	NB	0,6975	0,6225	0,5125
	XGB	0,6625	0,6325	0,61
Lemmatized POS Tagging	DL	0,61	0,6025	0,475
	LR	0,6175	0,6075	0,4775
	SVM	0,6275	0,62	0,4525
	RF	0,6125	0,5725	0,4775
	NB	0,5975	0,61	0,495

	XGB	0,565	0,5875	0,485

Table 42: Heat table of accuracies of all testing data

In table 42, we present heat map of the accuracies of all transformed inputs for each of the six models on the testing data. We present the heatmap to get an intuitive view of our results which clearly highlights that lemmatized input provides the best accuracies. Also, neural network, random forest, support vector machine and logistic regression has no output in red for lemmatized input with highest accuracies.

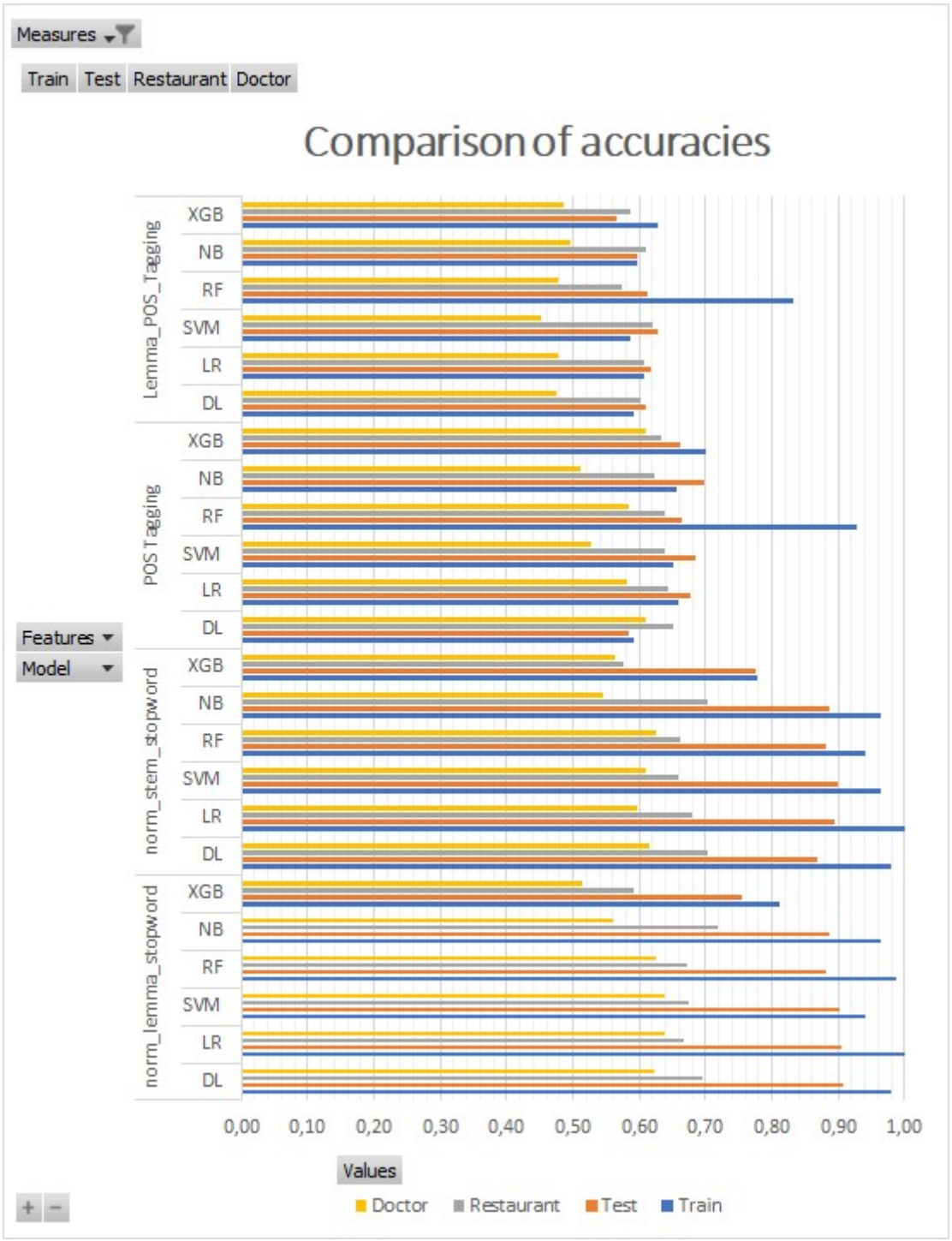


Figure 16: Comparison of accuracies of all models, datasets, input methods.

In figure 16, we present all the accuracies output ranging from training accuracies of hotel reviews, testing accuracies of hotel reviews, accuracies of restaurant reviews and accuracies of doctors' reviews. Noteworthy point here is that even training accuracy for POS tagging and lemmatized POS tagging is strikingly low.

4.3 DISCUSSION

In this section, we present the reasoning behind the results and analysis.

4.3.1 Input Transformation Perspective

Low accuracy of POS tagging signifies that the grammatical structures of truthful and deceptive reviews are not very different. Percentage of parts of speech in the text cannot help to obtain a good and balanced accuracy in classification of deceptive vs truthful reviews.

Normalized and lemmatized words hold the clues for deception and helps to achieve maximum of 91% accuracy for 25% of unseen hotel reviews. So, we can undoubtedly conclude that this is the best transformation method for the available data to perform classification between truthful and deceptive reviews. Also, the cross-domain results of restaurant reviews and doctors' reviews are very decent and better in comparison to others transformation method.

Stemming is slightly different from lemmatized words in the sense that stemming simply and brutally chops off the word ending whereas lemmatization convert the words to the meaningful base word. Stemming provides similar output to lemmatization. However, we prefer lemmatization since its performance is slightly better reasoning being sometimes stemmed words are meaningless and hence, often spoils the classification process by slight degree.

Lemmatized POS tagging performance was weakest. It's clear that in the process of lemmatization, we lost some vital parts of speech which couldn't be considered in POS tagging. We simply do not endorse this method based on our observation and outcome.

4.3.2 Models Perspective

Extreme gradient boosting gave weakest performance in comparison to other models.

Naïve Bayes' accuracy performance is comparable to other models like logistic, support vector, random forest and neural network, however it is slightly more volatile when we try to draw conclusion on different datasets like doctors' review. Hence, this is not the most preferred method for our analysis based on the outcomes.

The accuracies of Random forest, Logistic regression, Support vector and neural network are very similar and follow the same trend. Excellent accuracy for 25% of unseen hotel reviews, medium and decent accuracy for restaurant reviews. We do acknowledge that accuracy drops in restaurant

reviews, but it is expected since restaurants will have little different terminologies and we haven't trained the model using restaurant data. Finally, satisfactory accuracy for doctors' review. Again, the accuracy drops further since doctors' reviews will contain very different terminologies and words. No training of the models has been done using doctors' reviews dataset.

The star is Deep Learning – Neural Network for achieving 91% accuracy in classification on 25% of unseen testing data on hotel reviews. Features like early stopping when the performance starts dropping improves the performance of neural network.

4.3.3 Result Comparison

We compare the accuracy that we have received on 25% of unseen test data of hotel reviews against the accuracies obtained by other researchers and against humans. In certain cases, other researchers calculated f-measure or f-score, hence no accuracy was available. Hence, we have considered f-score in absence of accuracy.

We state with great happiness that our accuracy exceeded the accuracy obtained by other researchers. We achieved one of the highest accuracy score of 90.75% on testing data of hotel reviews (25% of unseen hotel reviews) using deep learning neural network and applying lemmatization on textual reviews.

Type	Particular	Comments	Positive	Negative	Combined
Human Performance	(Ott et al., 2011)	Judge 1	61,9		
	(Ott et al., 2013)	Majority		69,4	
Automated Performance	(Ott et al., 2011)	LIWC + Bigrams (SVM)	89,8		
	(Ott et al., 2013)	Cross Validation	88.4	86.0	
	(Cagnina and Rosso, 2015)	4-grams + LIWC (SVM) *F – Measure	89.0	86.5	87.9
	(Fusilier et al., 2013)	100-D 520-U *F – Measure			83,7
	(Ren and Zhang, 2016)	Integrated Hotel Customer/ Employee/ Turker			81,3
	(Li et al., 2014)	Hotel - Three - Class – Unigram			66,4
	Accuracy scored by us	Deep Learning Neural Network Accuracy F-score			 90,7 90,9

Table 43: Comparison of results against other researchers

5 CONCLUSIONS

Normalized lemmatization and normalized stemmed data provide robust and better performance in the accuracy of identification of deception in reviews in same sectors/industries. However, it lacks generation capability across different sector/industries. Its performance declines rapidly as we move across to more different sectors. POS tagging provides better generation capability across sector or industries, but its performance is not as strong as the lemmatized or stemmed inputs. Hence, we can conclude that n-grams provides robust performance for similar industries since it derives its strength from actual words which contains the clues of deception. Whereas POS tagging has a good generalization capability since it doesn't focus on words but the structural aspect of the grammar. So, POS is good for application across different industries which may contain different words or terminology. We do not endorse Lemmatized POS Tagging since it suffers from the disadvantage of both lemmatization and POS Tagging. Accuracy scores drops in Lemmatized POS Tagging since due to lemmatization we lose lots of grammatical structures which makes further POS Tagging less effective. Overall performance of lemmatization has far exceeded POS tagging. There is no difference in output of lemmatization and stemming at 95% significance level.

Logistic regression, support vector machines, random forest and deep learning neural network have shown very robust performance on various datasets. Accuracy expectedly drops when the vocabulary of the reviews changes due to change in industry from hotels to restaurant and finally accuracy drops even more for doctors' dataset. However, the decline is less steep than other models. Hence, it has better cross industries application. These are the recommended model for such research.
Deep Learning Neural Network achieved the highest accuracy of 91% in classification on 25% of unseen testing data on hotel reviews.

Naïve Bayes tends to perform similarly good like Logistic regression, support vector machine, random forest, and neural network on unseen data of hotel reviews but it also tends to perform comparatively worse in our analysis on unsimilar unseen data of doctor's dataset. Naïve Bayes has a volatile performance.

Extreme gradient boosting's accuracy was weakest among other models; hence we do not endorse it for such classification for these kinds of dataset.

6 LIMITATIONS AND RECOMMENDATIONS FOR FUTURE WORKS

There is a potential risk of overfitting while applying multiple models. Also, this is a very controlled experiment on a small dataset. Large volume of dataset may completely change the outcome of the study.

There can be endless, tedious work on this topic to increase accuracy of the models used. Also, it will be interesting and important to enhance the performance for cross domain usability.

Also, we can explore and do more on this topic by exploring below areas:

- more models
- other NLP techniques
- other features

7 BIBLIOGRAPHY

Asiri, Sidath. 2018. *https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623.* June 11. Accessed February 20, 2020. https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623.

Bachenko, Joan, Eileen Fitzpatrick, and Michael Schonwetter. 2008. "Verification and Implementation of Language-Based Deception Indicators in Civil and Criminal Narratives." *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)* 41–48. https://www.aclweb.org/anthology/C08-1006.pdf.

Breiman, Leo. 2001. "Random Forests." *Machine Learning, Volume 45, Issue 1* 5-32. https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf.

Cade, Whitney L., Blair A. Lehman, and Andrew Olney. 2010. "An Exploration of Off Topic Conversation." *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* 669-672. https://www.aclweb.org/anthology/N10-1096.pdf.

Cagnina, Leticia, and Paolo Rosso. 2015. "Classification of deceptive opinions using a low dimensionality representation." 58-66. https://www.aclweb.org/anthology/W15-2909.pdf.

Chandrayan, Pramod. 2015. *https://towardsdatascience.com/logistic-regression-for-dummies-a-detailed-explanation-9597f76edf46.* - -. Accessed February 20, 2020. https://towardsdatascience.com/logistic-regression-for-dummies-a-detailed-explanation-9597f76edf46.

Chen, Tianqi, and Carlos Guestrin. 2016. "XGBoost: A Scalable Tree Boosting System." *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 785-794. https://dl.acm.org/doi/10.1145/2939672.2939785.

Fusilier, Donato Hernández, Rafael Guzmán Cabrera, Manuel Montes-y-Gómez, and Paolo Rosso. 2013. "Using PU-Learning to Detect Deceptive Opinion Spam." *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis* 38-45. https://www.aclweb.org/anthology/W13-1606.pdf.

Garbade, Dr. Michael J. 2018. *https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b32.* October 15. Accessed February 20, 2020. https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b32.

Hancock, Jeffrey T., Lauren E. Curry, Saurabh Goorha, and Michael Woodworth. 2007. "On Lying and Being Lied To: A Linguistic Analysis of Deception in Computer-Mediated Communication." *Discourse Processes, Volume 45, 2007 - Issue 1* 1-23. https://www.tandfonline.com/doi/full/10.1080/01638530701739181.

Jansen, Jim. 2010. *pewresearch.org/internet/2010/09/29/online-product-research.* September 29. Accessed February 20, 2020. https://www.pewresearch.org/internet/2010/09/29/online-product-research/.

Jindal, Nitin, and Bing Liu. 2008. "Opinion Spam and Analysis." *WSDM '08: Proceedings of the 2008 International Conference on Web Search and Data Mining* 219-230. https://dl.acm.org/doi/10.1145/1341531.1341560.

Kaviani, Pouria, and Mrs. Sunita Dhotre. 2017. "Short Survey on Naive Bayes Algorithm." *International Journal of Advance Engineering and Research Development, Volume 4, Issue 11* 607-611. https://www.researchgate.net/publication/323946641_Short_Survey_on_Naive_Bayes_Algorithm.

Korkmaz, Murat, Selami Güney, and Şule Yüksel Yİğİter. 2012. "THE IMPORTANCE OF LOGISTIC REGRESSION IMPLEMENTATIONS IN THE TURKISH LIVESTOCK SECTOR AND LOGISTIC REGRESSION IMPLEMENTATIONS/FIELDS." *J.Agric. Fac. HR.U.* 25-36. http://static.dergipark.org.tr/article-download/imported/1101000025/1101000021.pdf?

Landis, J. Richard, and Gary G. Koch. 1977. "The Measurement of Observer Agreement for Categorical Data." *Biometrics, Vol. 33, No. 1* 159-174. https://www.jstor.org/stable/2529310.

Li, Jiwei, Myle Ott, Claire Cardie, and Eduard Hovy. 2014. "Towards a General Rule for Identifying Deceptive Opinion Spam." *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* 1566-1576. https://www.aclweb.org/anthology/P14-1147.pdf.

Lippman, David. 2020. *http://www.imathas.com/stattools/norm.html.* Accessed February 20, 2020. http://www.imathas.com/stattools/norm.html.

Mairesse, François, Marilyn A. Walker, Matthias R. Mehl, and Roger K. Moore. 2007. "Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text." *Journal of Artificial Intelligence Research 30* 457-500. https://www.aaai.org/Papers/JAIR/Vol30/JAIR-3012.pdf.

Mihalcea, Rada, and Carlo Strapparava. 2009. "The Lie Detector: Explorations in the Automatic Recognition of Deceptive Language." *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers* 309-312. https://www.aclweb.org/anthology/P09-2078.pdf.

Ott, Myle, Claire Cardie, and Jeffrey T. Hancock. 2013. "Negative Deceptive Opinion Spam." *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* 497-501.

Ott, Myle, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. 2011. "Finding Deceptive Opinion Spam by Any Stretch of the Imagination." *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* 309-319. https://www.aclweb.org/anthology/P11-1032.pdf.

Patel, Savan. 2017. *https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72.* May 3. Accessed February 20, 2020. https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72.

Pennebaker, James W., Cindy K. Chung, Molly Ireland, Amy Gonzales, and Roger J. Booth. 2007. "The Development and Psychometric Properties of LIWC2007." *LIWC2007 Manual* 1-22. https://www.liwc.net/LIWC2007LanguageManual.pdf.

Ren, Yafeng, and Yue Zhang. 2016. "Deceptive Opinion Spam Detection Using Neural Network." *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* 140-150. https://www.aclweb.org/anthology/C16-1014.pdf.

scikit-learn. 2007-2019. "https://scikit-learn.org/stable/modules/naive_bayes.html." *https://scikit-learn.org.* - -. Accessed February 20, 2020. https://scikit-learn.org/stable/modules/naive_bayes.html.

Srivastava, Durgesh K., and Lekha Bhambhu. 2010. "Data Classification using Support Vector Machine." *Journal of Theoretical and Applied Information Technology, Volume 12, No. 1* 1-7. http://www.jatit.org/volumes/research-papers/Vol12No1/1Vol12No1.pdf.

Vrij, Aldert, Samantha Mann, Susanne Kristen, and Ronald P. Fisher. 2007. "Cues to Deception and Ability to Detect Lies as a Function of Police Interview Styles." *Law and Human Behavior volume 31* 499-518. https://link.springer.com/article/10.1007/s10979-006-9066-4.

Williams, Shanna Mary, Victoria Talwar, R. C. L. Lindsay, Nicholas Bala, and Kang Lee. 2014. "Is the Truth in Your Words? Distinguishing Children's Deceptive and Truthful Statements." *Journal of Criminology* 1-9. http://downloads.hindawi.com/archive/2014/547519.pdf.

8 APPENDIX

Python packages used the thesis are below in italics:

- *import matplotlib.pyplot as plt*
- *import nltk*
- *import numpy as np*
- *import os*
- *import pandas as pd*
- *import re*
- *import string*
- *import xgboost, textblob, string*
- *nltk.download('universal_tagset')*
- *from itertools import product*
- *from keras import layers, models, optimizers*
- *from keras.callbacks import ModelCheckpoint, EarlyStopping*
- *from keras.models import Sequential, load_model*
- *from keras.preprocessing import text, sequence*
- *from keras.wrappers.scikit_learn import KerasClassifier*
- *from mpl_toolkits.mplot3d import Axes3D*
- *from nltk import word_tokenize, pos_tag*
- *from sklearn import datasets*
- *from sklearn import decomposition, ensemble*
- *from sklearn import model_selection, preprocessing, linear_model, naive_bayes, metrics, svm*
- *from sklearn.cluster import KMeans*
- *from sklearn.datasets import make_classification*
- *from sklearn.ensemble import GradientBoostingClassifier*
- *from sklearn.feature_extraction.text import CountVectorizer*
- *from sklearn.feature_extraction.text import TfidfTransformer*
- *from sklearn.linear_model import LogisticRegression*
- *from sklearn.metrics import accuracy_score, f1_score, classification_report, confusion_matrix, precision_recall_fscore_support*
- *from sklearn.metrics import make_scorer*
- *from sklearn.model_selection import cross_val_score*
- *from sklearn.model_selection import GridSearchCV*
- *from sklearn.model_selection import train_test_split*
- *from sklearn.pipeline import Pipeline*
- *from sklearn.preprocessing import scale*
- *from sklearn.svm import LinearSVC, SVC*
- *from sklearn.utils.multiclass import unique_labels*
- *from time import time*