Sentiment Analysis of Fake News

James Carr: X2H03,
Supervisor : Bappaditya Mandal
Keele University : MSc Advanced Computer Science
24/09/2020

Contents

System Information : Processor Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz, 3192 Mhz, 4 Core(s), 4 Logical Processor(s) GPU GeForce GTX 1060, 225 GB SSD.

Key Terms: Machine Learning, Sentiment Analysis, TF-IDF, BOW

I. Abstract

Fake News is a symptom of delirious socio-political attitudes where persons of frequent exposure demonstrate symptoms of unhealthy confusion and distorted understandings of reality. To combat misinformation the developments of fake news detection systems aim to identify illegitimate sources for users to be prominently informed about news content. Systems approaching this use a technology known as “Sentiment analysis”, a sub-field of machine learning. This paper aims to add to sentiment analysis research via experimenting with TF-IDF and Binary Bow feature engineering techniques, also investigating strengths of feature methods using Chi2 scores as a form experimental data. Classification of text uses: K-nn, Naïve Bayes, Decision Tree's and SVM algorithms. Each classifier is additionally configured using Boosting and Bagging for noise reduction. 100% accuracy using a dataset with 40,000 samples was found, methods where reproduced where a rate of 82 % accuracy using a secondary dataset with 1056 samples was found.

Honourable recognition and gratitude are given to my project supervisor Bappaditya Mandal for his insightful support, experienced knowledge and strong guidance throughout this project.

II. Introduction & Background

The definition of Fake News from the Oxford Dictionary is *“false information broadcast or published as news for fraudulent or politically motivated purposes”.* This term was popularised in 2016 during the U.S. presidential election, where ongoing discussion from academics, commentators, and politicians has resulted in greater pressure on social media giants to moderate misleading content and disinformation on their platforms. A prominent example thereof would be Facebook, who is considered by many to be a hotbed for disseminating false information. In response to the ongoing controversy and court hearings, as of this year 'Facebook's advertising policy was amended to state: "Adverts are reviewed to make sure that they meet our Advertising Policies". Failure to adhere to this policy would result in the removal of a source from Facebook's platform.

The results have been promising thus far. Policies have been enforced via experimental machine learning tools where access to privileged databases yielded improved accuracy via the "crowd-powered" feature, a system where users tag misinformation to reinforce machine learning classification. Stephen Barrett (2020) praised Facebook's efforts of removing 5G related conspiracies; however, concern is met over what Facebook considers as misinformation and not using an umbrella ban on all misinformation where some misinformation is hand-picked for removal. Such a case occurred during the Australian Elections in 2019, where it was stated from a Facebook spokeperson: *"it was not their responsibility to deal with fake news"* (Murph & Knaus 2019). There has been additional cases where employees themselves have raised concerns such as the whistleblower Sophie Zhang, who leaked a report to Buzzfeed in 2020 stating *"she had blood on her hands"* regarding her role of Facebooks handling of misinformation. Concerns likely to be shared by Zharaya's (ex) professional peers.

Due to controversy attached to social media companies calls for alternative systems which facilitate a greater degree of transparancy would assist in quenching problems associated with social media companies. Such efforts from academics and software engineers such as Gautam, et al., 2019 & Zhang, et al., 2020 yielded positive results, though research is still proceeding with continuous room for improvement. This research aims to add to the pool of research proposing a thorough analysis of typical sentiment analysis methods via extensive testing of algorithms and feature engineering method - tools further specified in the methodology.

III. Background Literature

Despite innovative developments in neural networks, there have been cases of inferior results in comparison to traditional classical machine learning techniques. Research from (Islam, et al., 2017), tested a model where Naïve Bayes and k-NN surpassed accuracy over Artificial N.N.'s. Furthermore, concluding that NN’s are superior for tasks such as image analysis although an unnecessary tool for sentiment analysis. Also, indicative of classic machine learning superiority is in a Facebook journal

"*Applying machine learning science to Facebook products*" (2020), stating use of machine learning methods although not of neural networks – hence insinuating classical classifiers are not arcane with relevant and viable application in sentiment analysis.

Thus, this research area reviews a collection of recent approaches using Classical ML tools which focus on sentiment analysis. Doing so shapes model methodology by best practices whilst also identifying holes in research for discovering the potential in model optimisation and error avoidance. N.N.'s are not used due to the limited time frame of the project also due to the author's (myself) limited experience with Data Science

A. **Table of Literature**

Author	*Methodology*	*Findings*
Samonte (2018)	News articles modelled based on the plurality of emotions. Tokenises 200 articles with stemming and count vectoriser. Classification used K-NN, Naïve Bayes and SVM.	Greater SVM accuracy classifying batch samples i.e. individual news sources and SVM was inferior to k-NN when classifying the complete dataset.
Bhutani, et al (2019)	Twitter data used for modelling plurality of emotions. Tokenises using TFIDF with text merged using cosine distancing. Classification used Naïve Bayes and random forest. Dataset size was of 74,000	Greater TFIDF accuracy over binary BOW using Naïve Bayes, accuracy of pre-processing further improved using cosine distancing. Classification of entire dataset gave superior results with random forest. The additional bigrams and trigrams testing caused no difference in accuracy rates.
Sawarna & Gupta (2020)	Uses amazon reviews for modelling with TFIDF and count vectorisation. Classification used random forest classifier and boosting as a base estimator.Dataset size was of 568,454	Greater accuracy of random forest used with and without boosting. TFIDF also improved accuracy than count vectorisation. Implemented use of noise reduction tools caused increased accuracy.
Reis & Correia (2020)	News articles use a News dataset with 2282 articles and 141 textual features tokenised using count vectorisation.	Greater accuracy of random forest and boosting which where statistically even over other classifiers. K-nn achieved highest accuracy rates and boosting with

	Classification used k-NN, Naïve Bayes, Random Forest, SVM and boosting, boosting was used individually and without a base estimator.	decision tree surpassed accuracy rates of K-nn.
Conroy, et al (2015)	Tokenisation used binary BOW, TFIDF with TFIDF additionally coupled with word negation. Used K-nn, SVM classifiers.	Greater accuracy using K-nn over other classifiers. TFIDF model more effective over binary BOW, TFIDF improved further coupled with word negation.
(Soucy & W. Mineau, 2018)	Improves weighting of Tfidf using a method named conf weight compared over. Data was classified using Naïve Bayes. Use noise reduction via boosting.	Greater model accuracy using modification of conf weight than using traditional vectorisation techniques. Superior results of classification came from using decision tree

B. **Literature Analysis**

Results from Sawarna & Guptac, Reis & Correia, Conroy, et al concluded that k-NN was best suited classifier. Research from Soucy & W. Mineau found Decision, Tee surpassed classifiers not using adaptive methods when benchmarked upon other methodologies which did not use adaptive methods. Research from Sawarna & Gupta discovered adaptive techniques also provided excellent results which where best-found using Decision Tree which exceeded K-nn.

Research from Conroy, et al found TFIDF accuracy was superior over Binary BoW. This is an appealing technique based on accuracy alone; however, there is still a risk of using TF-IDF as valuable tokens are not universally accepted in same fashion as Binary BoW. With a frequent amount of terms TF-IDF will rank a term to a lower condition of context which is of issue where within an article for instance an author's writing style may focus writing densely on a specific topic and TF-IDF will disregard these terms despite them being a key determination of context . This issue may be a problem approaching complex statements as much more precision is required for feature engineering though the risks involved may not be shared when modelling features of high density of data, as is the case in this paper's context.

Limitations in methodologies are evident where Samonte achieved low accuracy rates in proportion to the corpus size. Although superior classifier results were still present from Samonte when results where compared, the accuracy rates were inferior to the total proportion of data, where an impractical result of 63% was found. The works from Sawarna & Gupta prove that merely extending the corpus size capture more diverse language contexts where accuracy rates of a sample size of 99 to 100% where found. Due to a greater corpus size; feature engineering methods greater potential of capturing trends of text sequences and thus increase chances of capturing as many sentiment styles as possible. Such a technique is suitable for addressing fake news analysis, though potentially an unviable method for alternative research contexts.

IV. Additional Readings

The following literature is not used for this project and will follow those discussed in the literature analysis section. This section acts partially as a portion of literature with some grounding for future research. Discussion of the model's future research regarding the implemented methodology is within the conclusions section.

This section addresses the scope of problems impossible to be addressed by nominal word correspondence techniques, though alternatively requiring classification of text into multiple attributes, an area known commonly as complex sentiment analysis - referring to research of quantifying idea’s within sentiment. Emerging complex sentiment analysis research is limited and is widely considered a problem difficult to advance; regardless, this is an exciting field leading to significant advancements for computational linguistics with definite calls for more research. Research of complex analysis shares potential with fake news detection; where such systems would provide richer methods of classification such as isolating satirical, fake, and real types of news.

Researchers commonly use sarcasm as an example of a complex sentiment as it provides a grounding of transferable methods to classify other examples of complex terms. Dimensions of sarcasm, according to John D. & Katz (2012), are failed expectations, pragmatic insincerity, negative tesnsion and prescene of a victim.For valuable results, high-quality and dense data is necessary for capturing diverse examples of sarcasm and to satisfy dimensions mentioned by John D. & Katz. A Reddit dataset mined by (Khodak, et al., 2018) marks sarcastic text via comments associated with "/s", tags, a term users frequently mark posts with to state sarcasm. Despite no publications modelling the Reddit dataset, the extensive corpus size of 533 million is an improvement over the frequently used Twitter dataset mined by Emily Chen et al (2017) which possesses a corpus size of 780 thousand. Reddit's dataset is of high density though of low quality as sarcasm uses isolated conditions via the "/s" tag as a nominal value, a tag used to confirm sarcasm within text. Furthermore, relying on Redditors implying sarcasm text may not satisfy dimensions stated by John D. & Katz causing data to be inaccurate with an overlap of non-sarcastic and sarcastic content which may be challenging to model requiring extensive feature engineering techniques.

Despite the low density of Twitter data, high quality is present via emoji metrics i.e. frequency and type of emoji, offering accessible data for multi-dimensional text classification.

Wilson, et al., (2017) used this dataset by using modifications on word embedding techniques yielding 40% accuracy. An undesirable outcome though experiments paved the way for novel developments. Work from Chen & You (2019) incorporated emoji frequency modeling yielding 90% accuracy. An impressive result but a misclassification margin too wide for practical use. Research from Joshi, et al (2020) took a similair approach also using the Twitter set where wide margins in classifcation was further present; concluding that until updated data mining techniques are released then potential of research is limited.

Thus, developments in data mining techniques will expose the further potential for improving complex*, and all,* sentiment analysis techniques by opening new perspectives for identifying novel patterns in data. Research from Hans-Peter Kriegel (2007) details the future of data mining techniques where key developments are summarised into: "*Standardisation of data mining languages, Data Pre-Processing, Complex objects of data, Computing Resources, Scientific Web Mining, Computing Business Data".*

Specifically, the advancements in data pre-processing reduces common errors leading to voluminous data easier to maintain and manipulate. Research from García, et al., 2016 experimented with pre-processing to manage overflowing issues which are present when scaling and merging large datasets – a frequent problem disrupting data quality. Issues of overflowing was also shared in works from Lin & Guo, 2011& Yaqoob, et al., 2016, where it was collectively agreed overcoming these problems lie with advancements in data processing frameworks. It is demonstrated within Apache news forums that developments of the Spark library are continuously underway for optimising data engineering techniques and furthermore resolve big data processing errors.

When a satisfactory level of data mining advancements is achieved, possibilities of sourcing from widely diverse websites provide means of extensive amounts of training data to thus capture extensive language contexts not presently available. This is ultimately, the next step in resolving present limitations of sentiment analysis whilst sharing promising potential amongst other artificial intelligence fields for chat and speech recognition bots.

V. Methodology

The methodology proposed follows the same techniques in the literature analysis section. Firstly, data collection via collecting suitable dataset on Kaggle. Secondly, Feature engineering implementing Term Frequency -Inverse Document Frequencies and Binary Bow on separate models with stemming and stop word removal. Thirdly classification with Naïve Bayes, K-NN, SVM Logistic Regression and Decision Tree. Chi2 graphs where used as a form of experimental data to investigate additional insights of model performance.

A. **Used Datasets**

Data acquisition via Kaggle elected sets based on popularity via user ratings as this was an indicator of high quality and low risk of errors. Two datasets were used to expand the scope of the project, validating how transferable the methodology was working with data of varying sizes and rates of total articles possessing misinformation.

Each downloaded collection provided two different CSV files, these were assembled into a homogeneous dataset where they were classified via the addition of *fake, true* columns acting as positive or negative identifiers for classification algorithms to recognise where predictions are conducted upon features. Additionally, as feature engineering required slicing[1]. of the datasets, *original size* and *half-size* variables are assigned so vectorisation required no manual assignment of dataset sizes unless the reducing the sample size is necessary - as was in-fact the case with Dataset 2.

Dataset 1: Named, *Fake News Data* is collected by Antonis Maronikolakis[2] contains columns; *id, URL, title, tweet ids.* The original size of the dataset is 5280: column's *URL* and *I.D.* are dropped as these do not provide sentiment content resulting in data size of 2112. Due to the size of the dataset, the maximum sample size is possible for tokenisation, being 2122: 1056. The total sample size post tokenisation of 1056. Train set has total 844 entries with 39.93% fake news, 60.07% real news

Dataset 2: Named, *Fake and Real News* dataset is collected by Clément Bisaillon [3]contains columns; title, text, subject and date; with the date and subject columns dropped. The original size of the dataset is 224,490, removal of columns is 134,694 and the total tokens post stemming and stop word removal is 40,000. The sampling size has been reduced on Dataset 2 to 40,000 for the machine in use testing to reliably perform tests. Train set has total 32,000 entries with 53% fake news and 48% real news.

https://www.kaggle.com/antmarakis/fake-news-data/notebooks

https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset

[1] A split size is due to the lack of a-priori dictionary with no need for an analyser which does feature selection as the number of features will be equal to the vocabulary size found by analysing data (Sklearn, 2020).

[2] First Dataset:

[3] Second Dataset:

B. **Feature Engineering**

In sentiment analysis, Natural Language Processing (NLP) refers to deciphering the human language for analysis. Tokenisation refers to the principal technique of NLP by converting text sequences into matrices possessing the samples of tokens. Two tokenisation methods are employed for this model; Binary Bag of Words and TFIDF.

The results of each Feature Eng. method construct a vectorised n-series as {n1,n2….n}.

1. Binary Bag of words :

Binary Bag of words tokenises features based on the preceding words into different values. Within n-space where each token of word with a similar value associates to a given vector; bag of words obtains overall similarity based on the quantity of shared features

Sequence	1 Donald)	2 Trump	2 Just	3 Couldn't	4 Wish	5 All	6 Of	7 Americans	8 Happy	9 New	10 Year	Similarity
Donald Trump said this about all Americans	1	1	0	0	0	1	0	1	0	0	0	30%
All Republicans are in shock for new year	0	0	0	0	0	1	0	0	0	1	1	30%
Donald Trump just couldn't help himself	1	1	0	0	0	0	0	0	0	0	0	20%

Figure 1 – Count Matrix Term frequency – Inverse Document Frequency (TF-IDF):

Considering the example given in Fig.1 30% the similarity is limited with individual resemblance across set of features. However, with overwhelming lenience towards a given range features of a specific criterion, comparatively to features in a separate criterion, predicts what type of classification is given to a testing sample.

2. TF-IDF :

TF-IDF is used as an alternative method over BoW. TFIDF uses a similarity rate to determine how closely resembled classes are another across n-series. Tokens with rarer tokens are weighted over the less influential common terms. This is calculated in two portions.

Term Frequency for measuring word counts on single features:

Tf(t,dx) = t/f. T represents the token, where dx, d represents the document and n represents the document count. T/F refers to f as the total features over the number of tokens within a document. For instance, given tf("Trump",d1) = 60/1056. Thus, amount of (t) tokens titled "Trump", within d1, given as (d1) over 1056(dataset 1 size), features (f).

Following Term frequency sum, inverse Document Frequency is employed for measuring universal tokens with a given token. Using the input used for term frequency, IDF = (t,(*trump*), D(*total shared documents*) = $log = \frac{1056}{|\{1 \in : trump : \in 1|}$ which compares the term *“trump”* across total number of shared terms across a corpus.

Fig.3 demonstrates TFIDF ranges given from a single article:

Token	*TF-IDF score*
Trump	85%
Republican	60%
USA	45%
After	33%
Debate	31%
Senator	29%
Killed	15%

Figure 2 - TF-IDF Scores – An example extracted from an individually selected feature within dataset 1.

C. Stop word Removal and Stemming:

To improve feature selection, stop word removal and word stemming techniques have been employed for more precise model predictions.

An article with stop words, e.g. "the" & "and", are not terms which influence sentiment of text content. Other features within n-space sharing these stop word attributes will result in unrelated content sharing resemblance, hindering accuracy as binding sentiment of classification is shadowed by the abundance of these terms. Secondly, the use of stemming reverts synonyms to root meanings, E.g. "attain" & "accomplish" stemmed to "achieve*”*. Such as with stop words a variety of these terms provides an overabundance of word variety potentially causing misclassifications.

Given the text from Fig.2; "*The*", (removed), "*Republicans"*, "Shock" (changed to fear synonym) "*For" & "the"* (removed stop words)", *"New Year"*. Thus, resulting a vector with key terms of *"Republicans" &" Shock, & "New Year*.

VI. N-Gram's

For both, Binary BoW and TFIDF vectorisation, it is possible to adjust the N-Gram range within vectorisation parameters. For this model, unigrams, bigrams and trigrams have been tested using a Naïve Bayes Classifier for standard vectorisation and TF-IDF vectorisation.

Unigrams are formulated via : $(x_i \,|(n-1).\, x_i - 1)$ where n-1 predict words based on the occurrence of -1 word, and n-2 for bigrams, two preceding words and n-3 is used for trigrams, three preceding words. By adjusting the size of n-grams, there is potential to capture more complicated expressions for improved predictions. These experiments where decided upon as alternating n-gram sizes provide specific benefits results depending on prediction context; for instance, when working with larger or smaller corpora or different forms of writing styles.

n-gram	*Dataset 1 Accuracy (BoW)*	*Dataset 2 Accuracy (BoW)*	*Dataset 1 Accuracy (TFIDF)*	*Dataset 2 Accuracy (TFIDF)*
n-1	78.77%	99.57%	74.87%	85%
n-2	75.94%	98%	72.25%	88%
n-3	76.88%	95%	71.73%	70%

Table 1 - N-Gram scores - Based on the superior accuracy results from using unigrams for both BoW and TFIDF methods it decided that this is the most appropriate method for the model.

A. Splitting:

To finalise feature engineering; data is split into subsets of training and testing data. Training data referring to pre-categorised data and test data referring to uncategorised data. Testing data is used to makes predictions upon training data via producing results of model performance based on quantity of classifications.

Splitting parameters are configured with 80 per-cent of training data and 20 per-cent reserved for testing data to classify into training data. This has been decided upon as the "80/20 ratio" is typical for statistical practices for classifying relative importance; as discovered via Pareto Analysis who noted most phenomena have 80 per cent of consequences stemming from 20 per cent of causes (Blanc, 2009) *analogous to the ratios e.g. human settlement sizes* (Reed, 2002), *file distribution on internet traffic* (Tudajarov, et al., 2004), *hard drive error rates* (Abramovitch, et al., 1988). Causation, in this case being the un-classified text "consequentially" influenced by dichotomies of fake or real (true or false). Randomisation has not been used due to the context of testing focusing on observing classification performance thus use of randomisation would produce inconsistent results.

B. **Supervised Classification**

Supervised machine learning algorithms tested in the model are: Naïve Bayes, Logistic Regression, Decision Tree, Support Vector Clustering, Random Forest and K-Nearest Neighbour. Each classifier additionally is used as a base estimator with boosting and bootstrap aggregation for reducing misclassifications. The following section covers basic concepts behind each classifier with inputs of dummy data for demonstration. When classifier used in practice given algorithm repeats until maximum features within an n-space are computed.

1. Gaussian Naïve Bayes**:**

Sing (2019) describes the Naïve Bayesian theorem as; "finding conditional properties through assuming the presence of features in a class that is unrelated or related to other features given a class variable, basically, "naive", by making predictions that may or may not turn out to be correct." The objective of this model is measuring rates of true or false likelihoods, using Gaussian Naïve Bayes, as its name suggests, uses Gaussian (or normal) distribution, for measuring nominal probabilities, i.e. measuring accuracy rates of true or false and has been used for this model. This is compared to Classical Naïve Bayes which classifies multiple criteria using multinomial distribution

Fig.3 uses mean and variance from data points where x_i represents a range of features within a token. Given range of n-space within a vector as $(x) = -10, 10, 0$. Standard deviation is σ = 4.7140, with variance $\sigma^2 = 2.22$, with y as total sum of x = 20. Given testing data where $x = 0$ input into $P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{1}{2}} \left(\frac{x-\mu}{\sigma}\right)^2$, results in μ in mean of 0 thus standard deviation of 1 indicating maximum shared features within the distribution.

Other variant deviation from peak either positively or negatively indicates lower resemblance of features within n-space(stated as x samples).

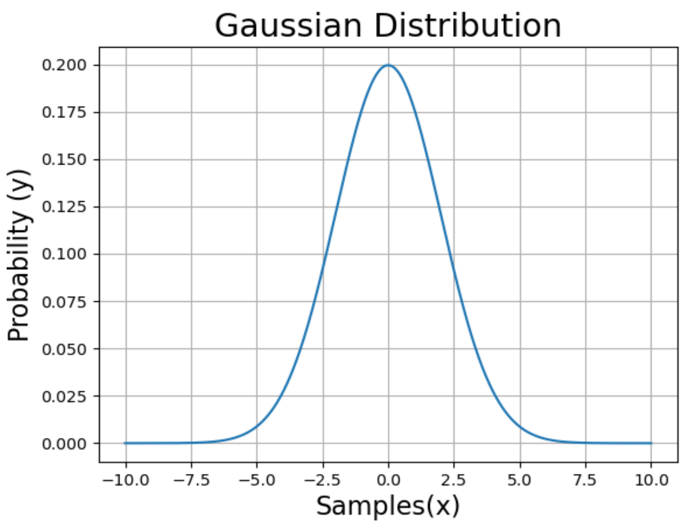


Figure 3 - Gaussian Distribution

2. Logistic Regression:

Logistic Regression is used for modelling dichotomies of nominal values. Bewick, (2005) describes Logistic Regression within two portions, where logistic refer to probabilities of mapping constant values from features and regression refers is relationship outcomes of independent predictions.

For each value of x, probabilities, i.e. p are used to denote the threshold within each prediction axis. For the prediction axis, positive values, set with $p > 0.5, fakedata = 1$, where fake news is positioned and $p < 0.5, truedata = 0,$ where the threshold of real news is positioned. Given x as a testing sample, probability is calculated via $p(x) = \frac{1}{1+e^{-x}}$. e.g. if x resembled an n-space features related to article count of a vector of 400 then there is a 60% likelihood of fake news classification *(In reference to fig.4).* .

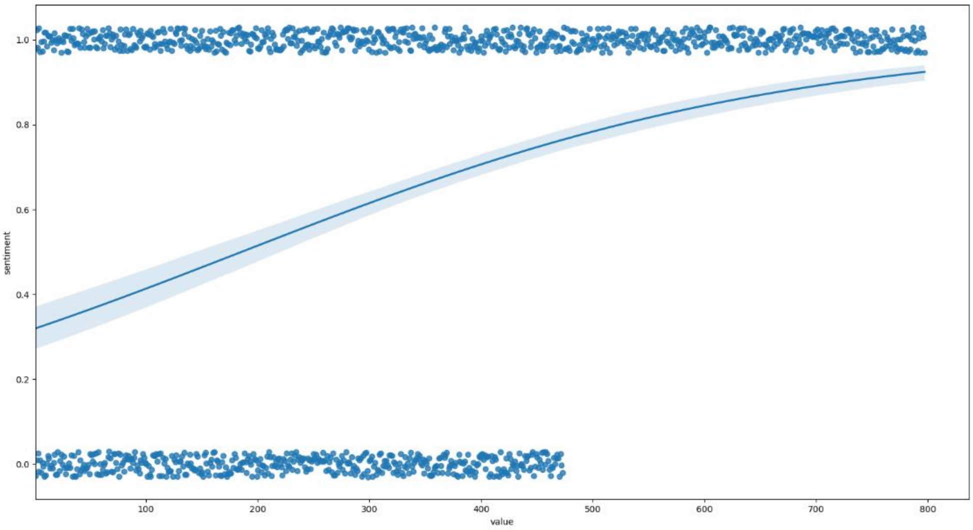


Figure 4 - Logistic Regression – (Using data from Datset1)

3. Decision Tree Classifier:

Safavian (1991) describes hierarchical classifiers, as a "*method of breaking up complex decisions into a union of simpler decisions to obtain a result in a way which would resemble the desired solution"*. Decision Trees are one of the most commonly used hierarchical classifiers, as Song & Lu (2015) describes as a *"formation of a hierarchy of branches, stemming from a root node further to internal nodes with each path determining a classification based on "if true" rules".*

The root of the tree is where the training data is configured, and decisions are distributed based on amount of similarity within different child nodes. Decision Trees determine pathing uses Gini Indexing where greater Gini weight resemblance indicates characteristics of information associated with stronger or weaker features. (Jia & Sun, 2012).

Given the root node p_o with o with values of $\{7.5\}$ creating two children based on features present in data. The first two children of values of $\{1,2\}$ and, for a true branch and $\{2,0\}$ index based for a false branch. Given test data of values $\{4, 2\}$would be classified for a true branch where all child nodes are scanned to measure resemblance of features. This is computed via $Gini = 1 - \sum_{i=1}^{C}(p_i)^2$, resulting in 0.444 index values acting as a positive classification (of fake news) due to due to greater resemblance than to the alternative child node of $\{2,0\}$.

4. Random Forest Classifier:

Random Forest Classification is a modification upon traditional decision tree classifiers, this is described *"as using a set of multiple decision trees which grow in selected subspaces of data (or forest). Within each forest, via random selection, data is tested to estimate how well testing data would fit into each tree with voting methods" (*Xu, 2012 & Liaw & Wiener, 2002). As Random Forest uses a collection of Hierarchical Classifiers the Gini Indexing is used to incorporate purity weight, as previously described for Decision Tree classification

Random Forest implements an additional function to this known as entropy to measure the amount of information transported via randomised trials within a range of 0- and 1-bits. Increased entropy, scores indicate vaster distribution and more disorderly predictions which not ideal; low entropy values are of greater value to indicate ensure more concise predictions.

Entropy is determined given a C, with (i) at 1 (or 100%), where C is calculated from the amount of repeated attempt to satisfy the probability p, exponentially as a product of the maximum of 1 i.e. $Entropy = \sum_{i=1}^{C} - p_i \cdot \log_2(p_i)$ Given the probability of misinformation with Gini indexing as $p_i = 20/80$ likelihood, represented via entropy as $\log = \frac{1}{0.80}$ resulting in $\log = 0.096$, where compared to the ulterior probability not being fake news is $\log = \frac{1}{0.20}$ resulting in $\log = 0.698$. These are positive entropy values indicating low cases of disorder and randomisation whereas greater entropy values would indicate uncertainty of whether a fake news classification was correct or not.

5. Support Vector Clustering (SVC):

The objective of clustering is to partition a data set into groups of criteria to organise data into a meaningful form, clustering of datasets may proceed parametrically(assumptions across data points), or by grouping . (Ben-Hur, et al., 2001). SVC's incorporate clustering tools via kernels, linear kernel is a popular technique used for binominal problems though there are other suitable options such as, Gaussian Sigmoid and Polynomial.

The equation of a linear kernel is given via $K(x_i, y_i) = x_i \cdot y_i$, where the product within a Kernel, K with x_i as an independent variable and y as secondary variable. Features are multiplied, dispersing based on y value and isolating clusters where greater dispersion indicates less similarity.

Let = 1 and y_i = 0, using $w^t x_i$, where w is an empty parameter, and t represents the difference of x_i to y_i Giving $x_i \cdot y_i$ as $\{1,0\} \cdot \{5,0\}$ cluster are positioned based on {5,0} for x_i and y_i as {5,0}. Whereby percentile of resemblance of each feature indicates maximum accuracy and thus similarity assumes resemblance and greater dispersion based on the samples in each class. In this case indicating that 0 (negative class cation) is present within a real news cluster.

6. K-Nearest Neighbour(k-nn):

k-NN incorporates the use of distance measures for measuring the distance between two points of data in order to find the "the nearest neighbour". To expand the scope of this model, results from Minkowski Manhattan, Euclidean, Cosine distancing measures have been implemented to note any influence on model performance. The greater space between two vectors in proportion to max corpus feature size determines accuracy

For all measures demonstrated q_1 is given as { 1.1 , 2.1} and p_2 {1.0 , 1.1} as coordinates.

Minkowski Generalisation: The Minkowski equation is distance measure for generalising n-space. Minkowski is formulated as $d(q_i, p_i) = (\sum_{i=1}^{n} |q_i - y_I|^{1/p})$ using the coordinates result in distance of 0.100 with total accuracy of $100 - 0.100 = 99.99\%$.

Euclidean Distancing: Measure n-dimensional space eq.2, between pair of samples,q_i, p_i via Pythagoreans Theorem. Using previously mentioned coordinates within $d(q_i, p_i)^2 = \sqrt{\sum_{i=0}^{m} (q_1 - q_2)^2}$ results in distance between points as 1.005 with accuracy of $100 - 1.005 = 98.995\%$.

Manhattan Distancing(City block): Assumes distance by calculation of distance of n space using axis Given the points of data mentioned previously The sum is a result of each coefficient of each portion of an n-space as points $d(q_i, p_i) = (\sum_{i=1}^{m} |q_{i-} p_i|^r + (y - x2)$ results in distance between two points as 1.100 with an accuracy of $100 - 1.110 = 98.89\%$

Cosine similarity: Functions independently without n-space generalisation like Euclidean and Manhattan and Minkowski. Cosine measures n-dimensions via calculating the dot product between two vectors divided by product of vector lengths. Using the same $q_i p_i$ figures mentioned for Minkowski, $cosine \cos(0) = \frac{q \cdot p}{||q|| B||} = \frac{\sum A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$ results in a cosine distance of 1.15 where 100 -1.15 = 99.95 accuracy.

Results from testing dummy data indicate Minkowski as superior. Figures in appendix validated this assumption where Minkowski achieved higher accuracy than competitors.

7. Adaptive Boosting (Boosting):

The premise of boosting is improving weak learning classifiers via a combination of enhancing relatively weak and inaccurate rules into a superior isolated learner as means of reducing error rates (Schapire, 2010). Boosting has been tested as a base estimator for all mentioned classifiers.

The process of adaboost is as follows:

Let $x = 1$ and $y = 5$ and $x_i \in x, y_i \in \{-1, +1)$. Where all x inputs are the element of the total set x and y outputs are the element of a set comprising of only two values, -1, negative class, and 1 as a positive class. Weights are initialised with value of 1 divided by the amount training samples as $x = 1\ l^M$. Hypothesises of weight, h are given within the error range of {-1, +1} with estimated weights and thus, error weights, ε are selected based on low weightings where less than 0 via $P\ \ \sim Dt\ [h(xi)\ not\ equal\ to\ y_i$ and i is updated with appropriate weighting as $\alpha\_t\ =\ 1/2\ *\ ln(1 - \varepsilon\ /\ \varepsilon),$

For instance, given 4 samples with weights 0.5, 0.2, 0.1 and 0.04 with predictions of 1,1,-1 and -1 though with incorrect weighting predictions of. -1, 1, -1, 1. Misclassification Rate is calculated via (0.5*1 + 0.2*0 + 0.1*0 + 0.04*1) / (0.5 + 0.2 + 0.1 + 0.04) resulting in an error rate 0.642, This error rate exceeds the error weight threshold thus boosting will now initialise an error reduction hypothesis. Weights are calculated via ε = 0.3, α = 1/2 * ln(1- 0.3 / 0.3) = 0.42365 resulting in ε = 0.7 α = 1/2 * ln(1- 0.7 / 0.7) = -0.42365. Where α > 0 if ε <= 0.5), 2 does not contribute to final prediction where weights are updated.

The weight is then determined based on m's positivity where the normalisation factor ensures that the total instances of weights are equal to 1, identifying a successful classification. Misclassifications which are correctly fixed would therefore be updated with greater weights, using the +1 condition. After each iteration the boosting will continue until attempted misclassifications are resolved. Classification differences potentially occur however when original classifications become misclassifications when giving inappropriate weightings.

Described following methods from (Cat, 2017)

8. Bootstrap Aggregation (Bagging):

Bootstrap aggregation also known as bagging is an ensemble estimator with similar benefits to boosting; this method improves performance of algorithms via reducing overfitting and error rates by repeating calculations of weak learners to achieve more informed predictions.

Given a simple example where attempting to calculate the $Mean(x_i) = 1/m \cdot \sum x$, where in this example x represents 5 as the total vector space. A single calculation results in inconsistent results thus employment of aggregation repeats the calculation based on size of vector space. Given mean values as 0.4, 0.4, 1.4, 1.2 aggregation calculates the average from this cluster, resulting in X = 0.88.

Calculations such as this example are normative to math used of base estimator employing bagging. Decision Tree for instance would form a collection of hypothesises based on the present samples possibly shared with corresponding Gini Indexes to validated for predictions.

VII. Chi2 - Experimental Data

The experimental data used for this paper uses Chi2 on Dataset 1 to visualise success of TFIDF and Binary Bag of Words methods discussed in the Feature Engineering section. The Chi-square (χ2) provides insights of model performance by measuring how expectations of classification are predicted in theory, over predictions made in practice. This is calculated via $(r, c) = \frac{n(r) \cdot c(r)}{n}$ measuring how much independence between r (row of feature) and c (related feature) is present compared within total distribution n.

Each degree of freedom indicates total independence where independence indicates a methods ability to determine relationships between two variables; i.e. text features that recognise familiar characteristics of other features.

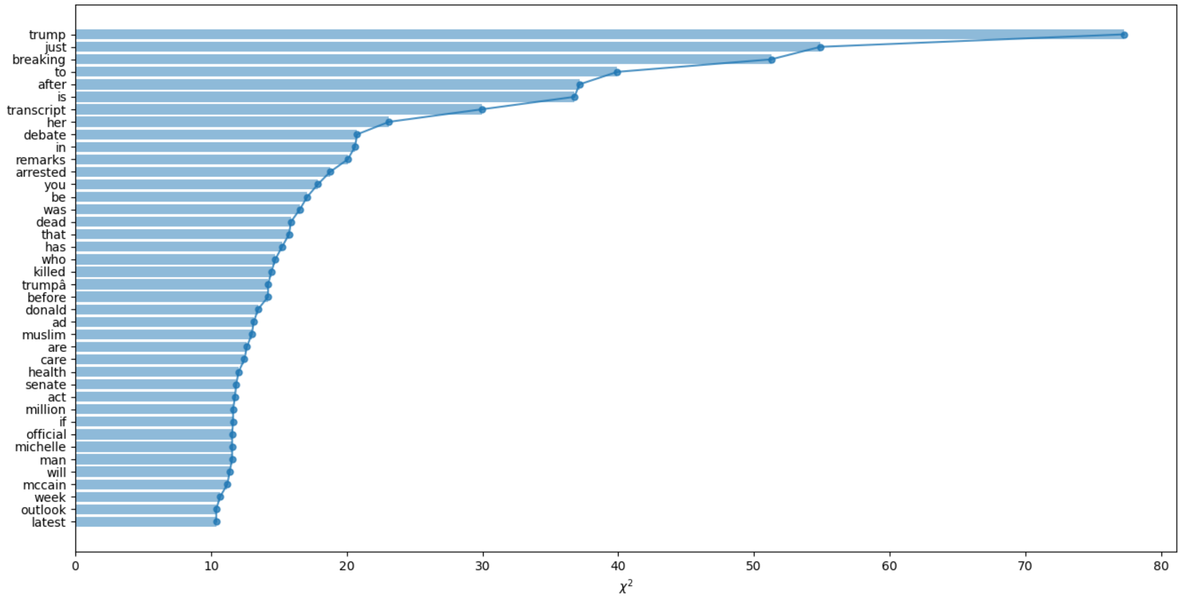


Figure 5- Chi2 test using Binary BoW

Fig.5. results show poor correlation between terms as Y axis terms with not share similar resemblances. X2 score at maximum of 80 indicates high independence degrees and inaccurate contingencies.

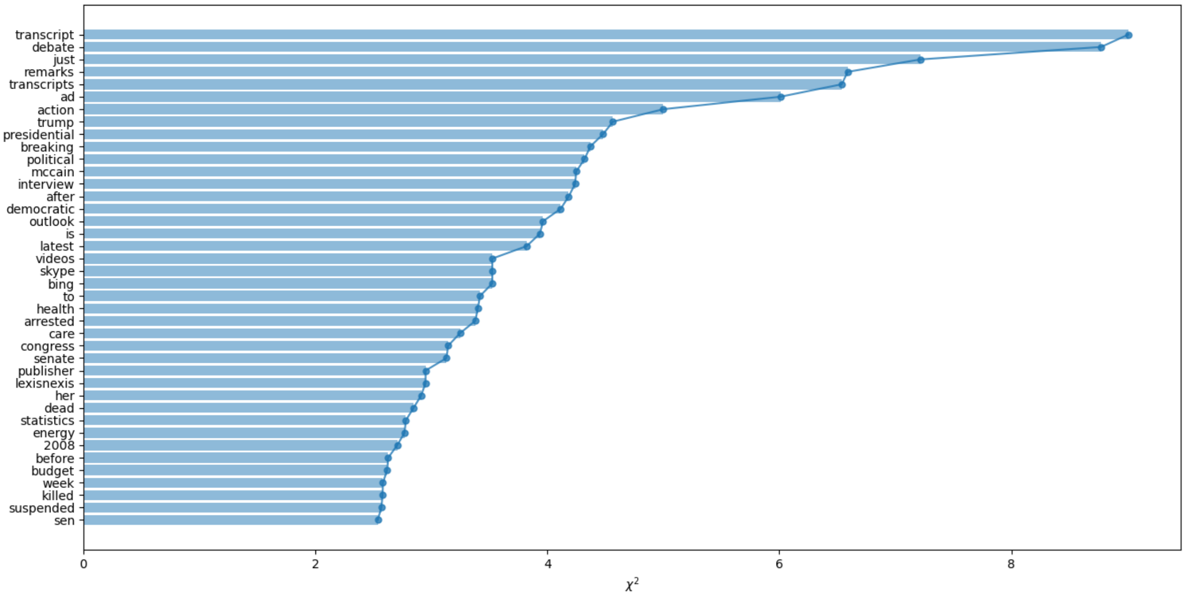


Figure 6 - Chi2 test using TF-IDF

Fig.6 demonstrates improved relationships over Binary BoW. The X2 score is at 8 maxima in comparison to dispersion of 80 with BoW Chi2. Features of the y axis demonstrate coherent sequences indicating denser predictions and relationships.

VIII. Analysis Methods

Classification results are observed through results of 2x2 confusion matrices. Following prediction of testing data, matrices are plotted by predictions * unclassified predictions. Results are positioned based on, false: 0, or true: 1 associating confusion matrix plots to rates of true positives, (1,1) true negatives, (1,0) false positives (0,1) false negatives (0,0). representing true positives – (fake news predicting fake news) or false negatives- (fake news incorrectly predicting real news) and vice versa, thus, providing results in different metrics.

1. Recall = TP / TP + FN = Fake News Identified / Fake News Identified + Fake News Incorrectly Labelling Real News
2. Precision = TP / TP + FP = Fake News Identified / Fake News Identified + Real news Incorrectly Labelling fake news
3. F-Measure = 2 * (precision * recall / precision + recall)

The F-measure score is the most valued metric as it combines the macro average of negative and positive loss rate from precision and recall providing an accurate performance review of classification. Additionally, to measure run-time of predictions timer functions from the stock Python library are employed which trigger when a classifier loads training data and cease timing once testing data is predicted.

The Collection of Matrices in tabulated data section use accuracy as a term for f1 score.

IX. Tabulated Data

1. Unigram Results Dataset 1

Classifier	*Base*	*Boosted*	*Bagged*
Naïve Bayes	79%	70%	81%
Logistic Regression	79%	69%	76%
Decision Tree	%	82%	82%
Random Forest	76%	72%	75%
Support Vector Machines	75%	N/A	N/A
K-N-Neighbour	77%	N/A	N/A

2. Bigram Results Dataset 1

Classifier	*Base*	*Boosted*	*Bagged*
Naïve Bayes	76%	68%	76%
Logistic Regression	76%	70%	75%
Decision Tree	75%	77%	77%
Random Forest	75%	76%	76%
Support Vector Machines	74%	53%	74%
K-N-Neighbour	75%	N/A	N/A

3. Trigram Results Dataset 1

Classifier	*Base*	*Boosted*	*Bagged*
Naïve Bayes	77%	60%	77%
Logistic Regression	77%	69%	76%
Decision Tree	71%	77%	77%
Random Forest	73%	76%	74%
Support Vector Machines	77%	55%	74%
K-N-Neighbour	75%	N/A	N/A

4. Unigram Results Dataset 2

Classifier	*Base*	*Boosted*	*Bagged*
Naïve Bayes	91%	89%	91%
Logistic Regression	91%	99%	99%
Decision Tree	99%	100%	100%
Random Forest	99%	100%	100%
Support Vector Machines	99%	N/A	N/A
K-N-Neighbour	90%	N/A	N/A

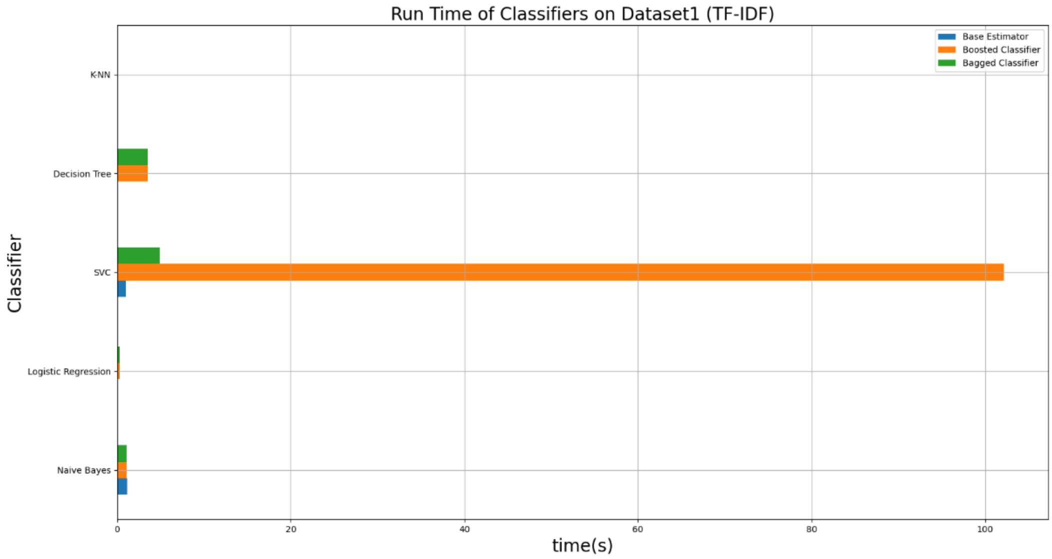
5. Bigram Results Dataset 2

Classifier	*Base*	*Boosted*	*Bagged*
Naïve Bayes	96%	90%	96%
Logistic Regression	96%	99%	100%
Decision Tree	99%	99%	100%
Random Forest	99%	100%	100%
Support Vector Machines	99%	N/A	N/A
K-N-Neighbour	91%	N/A	N/A

6. Trigram Results Dataset 3

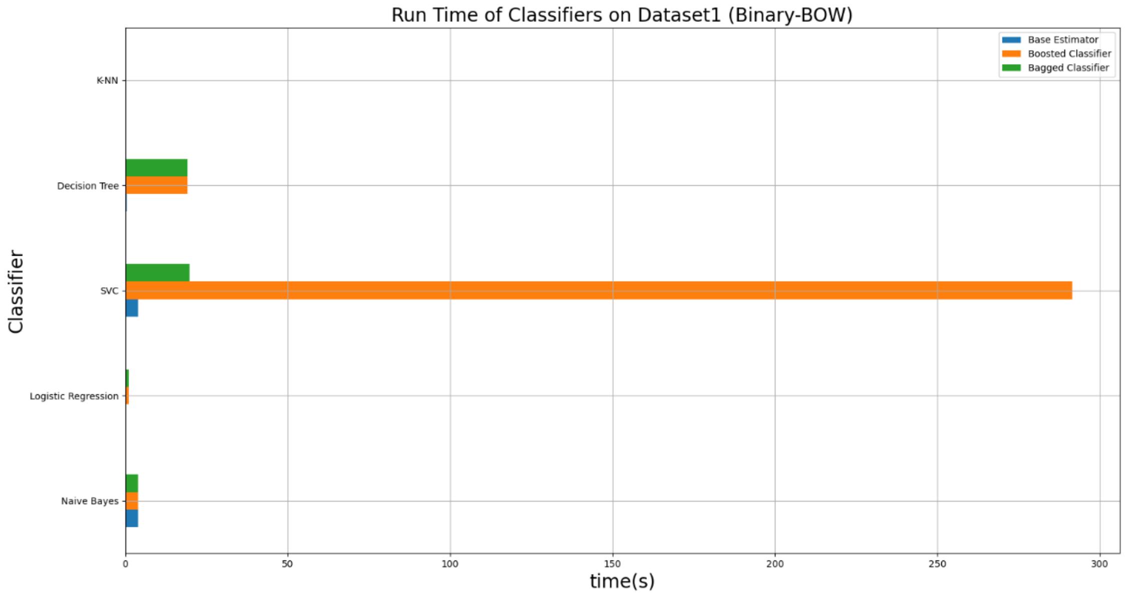
Classifier	*Base*	*Boosted*	*Bagged*
Naïve Bayes	96%	90%	96%
Logistic Regression	96%	99%	100%
Decision Tree	99%	99%	100%
Random Forest	99%	100%	100%
Support Vector Machines	99%	N/A	N/A
K-N-Neighbour	91%	N/A	N/A

X. Runtime of classifiers



Classifier	*Base Estimator*	*Boosted*	*Bagged*
Naïve Bayes	1.13	1.12	1.12
Logistic Regression	0.01	0.29	0.29
SVC	1	99.84	4.91
Decision Tree	0.09	3.52	3.52
K-NN	3.52	N/A	N/A

Figure 7 - TF-IDF Runtime



Classifier	*Base Estimator*	*Boosted*	*Bagged*
Naïve Bayes	4.02	3.99	3.99
Logistic Regression	0.01	1.14	1.14
SVC	5.00	353.05	37.92
Decision Tree	0.29	37.9	37.95
K-NN	0.04	N/A	N/A

Figure 8- Binary Bow Runtime

XI. Findings

Findings are summarised as:

- Naïve Bayes gives optimal error rates over all classifiers.
- N-gram size of 1 provided marginal yet beneficial results of classification
- TFIDF run time was faster than using Binary Bow though no difference on accuracy.
- The use of boosting improved the predictive performance of unstable learners such as decision trees, but not of stable learners for Support Vector Machines and Naïve Bayes.
- SVM caused run time of model to be very slow for both datasets taking 5 minutes to model 1056 samples and up to 3 hours modelling 40,000 samples .
- Poor accuracy rates on dataset 1 at a peak of 82%.
- K-NN gave optimal run time speeds and high classification rates.
- Maximum Classification accuracy using Dataset 2 using boosted and bagged decision trees.

XII. Conclusion & Future Work

This paper has experimented with sentiment analysis tools for classify fake news articles. Maximum classification accuracy was achieved on a Dataset using 40,000 samples. Typically, max accuracy indicates overfitting; though employment of noise reduction tools, used as a means of reducing overfitting, preserved maximum accuracy rates and thus validated results. Evaluation of feature engineering methods discovered TF-IDF is recommended over Binary Bow providing superior classifications on smaller datasets while as an extra bonus improving run-time of classifiers.

There is plenty of room for more experiments in the future. Merging the two different datasets into a complete set would develop a larger corpus and capture a wider degree of training data. It is also recommended to perform tests by feeding data into the model to measure practical use. A significant error in this model was SVC run time; possibly a fault of not using scaling mechanisms. Fixing errors with SVC would provide more meaningful results as the extensive run time deems SVC presently unviable with no valuable findings, this is contrary to findings in literature where SVC was considered as a strong classifier. There are also more algorithms to test; notably “Infinite boost”, and ‘SLIDE’ with apparent potential of improved run time and high accuracy. Comparisons of traditional machine learning tools benchmarked against emerging developments in neural networks would be of high value in future research.

XIII. Works Cited

Abramovitch, D., Hurst, T. & Henze, D., 1988. An overview of the PES Pareto Method for decomposing baseline noise sources in hard disk position error signals. *IEEE Transactions on Magnetics,* 34(1), pp. 17-23.

Ben-Hur, A., Horn, D., Siegelman, H. T. & Vapnik, V., 2001. Support Vector Clustering. *Journal of Machine Learning Research,* 1(1), p. 13.

Bewick, V., 2005. Statistics review 14: Logistic regression. *Critical Care,* 9(1), p. 7.

Bhutani, B., Rastogi, N., Sehgal, P. & Purwar, A., 2019. Fake News Detection Using Sentiment Analysis. *International Conference on Contemporary Computing* , 1(1), p. 5.

Blanc, L. A. L., 2009. Data mining of university philanthropic giving: Cluster-discriminant analysis and Pareto effects. *Berry College Campbell School of Business,* p. 19.

Cat, S., 2017. *Boosting algorithm: AdaBoost.* [Online]
[Accessed 2020 09 21].

Chen, Y. & You, Q., 2018. Twitter Sentiment Analysis via Bi-sense Emoji Embedding and. *Microsoft Research AI,* 1(1), p. 9.

Chetashri, B., Hardi, D. & Heenal, D., 2015. Sentiment analysis: Measuring opinions. *International Conference on Advanced Computing Technologies and Applications (ICACTA2015),* 1(1), p. 7.

Conroy, N., Chen, Y., Cornwell, S. & Rubin, V., 2015. Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News. *Language and Information Technology Research Lab (LIT.RL),* 1(1), p. 10.

D.Lavanya, 2012. Ensemble Decision Tree Classifiers For Breast Cancer Data. *International Journal of Information Technology Convergence and Services,* 2(1), p. 8.

Eibe, F. & Hall, M., 2002. Locally weighted naive bayes. *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence,* 1(1), pp. 249-256.

Emily Chen, 2020. Tracking Social Media Discourse About the COVID-19 Pandemic: Development of a Public Coronavirus Twitter Data Set. *University of Southern California, Information Sciences Institute,* 1(1), p. 10.

Esteban, M. D., 1995. GNU Grep 3.4. *Kybernetika,* 31(4), p. 11.

Facebook, 2020. *Advertising Policies.* [Online]
Available at: https://www.facebook.com/policies/ads/
[Accessed 15 09 2020].

Facebook, 2020. Applying machine learning science to Facebook products. *Research at Facebook,* 1(1), p. 1.

Ferrara , E., 2016. The Rise of. *Communications of the ACM,* 59(7), p. 59.

García, S., Gallego, S. & Luengo, J., 2016. Big data preprocessing: methods and prospects. *Big Data Analytics* , 1(9), p. 40.

Gautam, A., Yadav2, S., Kataria, R. & Desai4, M., 2019. Fake News Detection. *International Research Journal of Engineering and Technology (IRJET)* , 06(04), p. 1.

Guynn, J., 2020. *Facebook says it's cracking down on climate change misinformation. Scientists say it's not doing enough.*. [Online]
Available at: https://eu.usatoday.com/story/tech/2020/09/15/facebook-climate-change-misinformation-disinformation-conspiracy-theories-wildfires/5799418002/
[Accessed 2020 09 15].

Hur, A. B., 2008. Support vector clustering. *Computer Science Department Coloardo State University,* 1(1), p. 2.

Islam, S. et al., 2009. Modeling Spammer Behavior: Naïve Bayes vs. Artificial Neural Networks. *International Conference on Information and Multimedia Technology,* 1(1), p. 4.

Jahromi, A. H., 2017. A non-parametric mixture of Gaussian naive Bayes classifiers based on local independent features. *IEEE,* p. 10.

Jia , X. & Sun, J., 2012. An Improved text Classification Method Based on Gini Index. *Journal of Theoretical and Applied Information Technology* , 43(2), p. 7.

Joshi, A., Ohattacharay, P. & J.Carman, M., 2017. Automatic Sarcasm Detection: A Survey. *Monash Univeristy & Indian Insitution of Technology Bombay,* 22(1), p. 22.

Khodak, M., Saunshi, N. & Vodrahalli, K., 2018. A Large Self-Annotated Corpus for Sarcasm. *Computer Science Department, Princeton University,* 1(1), p. 6.

Liaw, A. & Wiener, M., 2002. Classification and Regression by Random Forest. 2/3(1), p. 5.

Lin, F. & Guo, J., 2011. Improving support vector machine by preprocessing data with decision tree. *International Conference on Computer Science and Service System (CSSS),* 5(14), p. 30.

Luoman, 2018. *Sampling Life.* [Online]
Available at: https://jkmsmkj.blogspot.com/2018/10/confusion-matrix.html
[Accessed Monday August 20200].

Murph, K. & Knaus, C., 2019. *Facebook says it was 'not our role' to remove fake news during Australian election.* [Online]

Available at: https://www.theguardian.com/technology/2019/jul/31/facebook-says-it-was-not-our-role-to-remove-fake-news-during-australian-election
[Accessed 2020 09 15].

Nembrini, S., 2020. The revival of the Gini importance?. *Bioinformatics,* 01(21), pp. 3711-3718.

Oracle, 2020. *Naive Bayes.* [Online]
Available at: https://docs.oracle.com/cd/B28359_01/datamine.111/b28129/algo_nb.htm#DMCON018
[Accessed 25 08 2020].

Peterson, L. E., 2009. K-nearest neighbor. *Center for Biostatistics, The Methodist Hospital Research Institute,* 4(2), p. 5.

Rapp, D. N. & Salovich, N. A., 2018. Can't We Just Disregard Fake News? The Consequences of Exposure to Inaccurate Information. *Sage Journals.*

Reed, W. J., 2002. On the Rank-Size Distribution for Human Settlements. *Journal of Regional Science,* 1(1), p. 17.

Reis, J. & Correia, A., 2020. Supervised Learning for Fake News Detection. *Affective Computing and Sentiment Analysis*, Volume 1, p. 6.

Safavian, S. R., 1991. A Survey of Decision Wee Classifier Methodology. *IEEE Transactions on Systems Mans and Cybernetics,* 21(3), p. 15.

Samonte, M. J. C., 2018. Polarity Analysis of Editorial Articles towards Fake News. *International Conference Proceeding Series (ICPS),* 112(1), p. 108.

Sawarna, A., 2020. Comparative Analysis of Bagging and Boosting Algorithms for Sentiment. *International Conference on Smart Sustainable Intelligent Computing and Applications* , 1(1), p. 10.

Sawarna, A. & Guptac, M., 2020. Comparative Analysis of Bagging and Boosting Algorithms for Sentiment Analysis. *International Conference on Smart Sustainable Intelligent Computing and Applications under ICITETM2020,* Volume 172, pp. 210-215.

Schapire, R. E., 2010. Explaining AdaBoost. p. 16.

Schölkopf, B., Tabibian, B., Oh, A. & Kim, J., 2018. Information, Leveraging the Crowd to Detect and Reduce the Spread of Fake News. *MPI for Intelligent Systems,* 18(1), p. 9.

Silverman, C., Mac, R. & Dixity, P., 2020. *Facebook-ignore-political-manipulation-whistleblower-memo.* [Online]
Available at: https://www.buzzfeednews.com/article/craigsilverman/facebook-ignore-political-manipulation-whistleblower-memo
[Accessed 2020 09 15].

Sing, S., 14/04/2019. *Ques10.* [Online]
Available at: https://www.ques10.com/p/43611/why-naive-bayesian-classification-is-called-naive-/
[Accessed 2020 08 14].

Sklearn, 2020. *Scikit.* [Online]
Available at: https://scikit-learn.org/
[Accessed 22 08 2020].

Song, Y.-y. & Lu, Y., 2015. Decision tree methods: applications for classification and prediction. *Shangai Archives of Psychiatry* , 1(1), pp. 130-15.

Soucy, P. & W. Mineau, G., 2018. Beyond TFIDF Weighting for Text Categorization in the Vector Space Model. *International Joint Conferences on Artificial Intelligence Organization,* 1(1), p. 6.

Stephen Barrett, 2020. Facebook Should Do More to Combat Vaccine Misinformation. *Quackwatch,* 1(1), p. 1.

Tudajarov, A., Temokv, D., Janevski, T. & Firfov, O., 2004. Empirical modeling of Internet traffic at middle-level burstiness. *Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference,* 1(1), p. 20.

Wakefield, J., 2020. *Facebook staffer sends 'blood on my hands' memo.* [Online]
Available at: https://www.bbc.co.uk/news/technology-54161344
[Accessed 2020 09 15].

Wallach, H. M., 2015. Topic Modeling: Beyond Bag-of-Words. *Cavendish Laboratory, University of Cambridge,* p. 8.

Westfall, P., 2020. *Standard Deviation vs. Variance: What's the Difference?.* [Online]
Available at: https://www.investopedia.com/ask/answers/021215/what-difference-between-standard-deviation-and-variance.asp#:~:text=Standard%20deviation%20is%20a%20statistic,square%20root%20of%20the%20variance.&text=Standard%20deviation%20is%20calculated%20as,point%20rel
[Accessed 2020 09 10].

Wilson, T., Kouloumpis, E. & Moore, J., 2020. Twitter Sentiment Analysis:. *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media,* 1(1), p. 4.

Xu, B., 2012. An Improved Random Forest Classifier for Image Classification. *International Conference on Information and Automation,* 1(1), p. 6.

Yang, Y. & Pederson, J., 2018. A Comparitive Study on Feature Selection in Text Catagorisation. *Carnegie Mellon Univeristy, School of Computer Science* , 1(1), p. 9.

Yaqoob, I., Targio, I., Gani, A. & Mokhtar, S., 2016. Big data: From beginning to future. *International Journal of Information Management*, pp. 1231-1247.

Zhang, J., Dong, B. & Yu, P. S., 2020. FAKEDETECTOR: Effective Fake News Detection with Deep Diffusive Neural Network. *Department of Computer Science, Florida State University*, 1(1), p. 13.

Zhang, W., 2011. A comparative study of TF*IDF, LSI and multi-words for text classification. *Expert Systems with Applications*, 38(3), pp. 2783-2796.

A.