**Alternus Vera (draft)**

Asha Aher     Monica Dommaraju     Swati Narkhede                             Sruthi Chilukuri

San Jose State University     San Jose State University     San Jose State University               San Jose State University

ashabalshiram.aher@sjsu.edu    monica.dommaraju@sjsu.edu    swatiganesh.narkhede@sjsu.edu      srisruthi.chilukuri@sjsu.edu

***Abstract- The sources for news have become so many from online news and social media to news articles and newspapers. In this paper, we as a team would like to put forward a machine learning model to make a prediction on the fake news and differentiate it from the real news based on various aspects. The data has been collected from Kaggle and other sources which contained texts along with the speakers and authors; giving us good scope to build a separate model that can classify the truthful news content from the fake news.***

**Keywords**-Data enrichment, Tf-idf, lemmatization, stemming, stop words, tokenization, LDA, Topic Modelling, Cosine Similarity, spell check.

1. INTRODUCTION

The importance of detecting fake news has become prominent in today's world where the outgrowth and abundance of news content is growing day by day. With all the online news, social media, newspapers and articles, it's important to differentiate between the fake news from real news.

II. ABOUT DATASETS

1. Liar-Liar

Liar-Liar Pants on Fire is a multi-class classification dataset having 3 files trained, tested and valid; has 14 columns: ID, label, statement, subjects, speaker’s job title, state information, etc.

1. Fake News

Fake News dataset is selected from Kaggle containing Id, author, text and label.

1. ISOT Fake News

This dataset consists of both Fake news and Real news. Real news was obtained by crawling articles from Reuters. Fake news was collected from unreliable websites flagged by Politifact. These datasets mostly cover topics on World news and political news.

1. Stance Dataset From Fake News Challenge

This is a multi-class labeled dataset and it consists of 4 files which are train body, train stance, test body and test stance. Altogether combined they have 4 columns like headline, body, stance(which is a label) and body id(which is used to combine all the datasets.

III. DATA CLEANING

Data cleaning and preprocessing of raw contents, containing the following steps:

* Tokenization
* Stemming
* Stop words removal
* Lemmatization
* Spell check

IV. FEATURE ANALYSIS:

***FACTOR1: Content Statistics***

 For any text based applications; it is important to understand the statistics or metrics of any content/data that is undergoing processing.

Applications like textual analysis, information retrieval or text summarization use these metrics to better understand the term's frequency, their importance in the given document.

When it comes to exploring data using statistical methods, we use ***Exploratory graphics.***

These are the preliminary steps we apply to the data in hand to:

* feel the data- see their distributions and shape
* check assumptions-and dependencies; correlations
* check if the data assumptions match the methods/models we got to use
* check for anomalies-outliers and unusual distributions and errors
* get suggestions-help us pursue a different angle or different method of analysis

But why do we have to use Exploratory graphics? It is because the graphical representations are information dense and are often the best way to check for shape, gaps or outliers.

Depending on the data, we use univariate distributions like bar charts when there is one variable or swarm plots and scatterplots for multi-variable data.

Similarly, we use box plots for qualitative variables and histograms to detect outliers. scatterplot

Overall, content statistics help us in answering the following questions:

* Do we have what you need?
* Are there clumps or gaps?
* Are there any exceptional cases?
* Are there any errors in data?

When it comes to exploring data using numerical methods, we use ***Exploratory statistics:***

What do we do when we say we are dealing with numeric values in the data? We explore it to get what are called the empirical estimates.

In simple words, we manipulate data in many ways like the following:

* Transform variables
* Check sensitivity of results
* Use robust statistics like resampling technique or  cross validation
* Transforming operations like smoothing functions fix skewed dataset

Then comes the ***descriptive statistics.*** To tell the data story; we use little data to stand in for a lot of data. These include mean, mode, & median to represent the center of the data and range, percentiles, interquartile range, variance and standard deviation to represent the spread of the data.

**Application of the factor on the amalgamated dataset:** After performing data amalgamation of the three datasets: Kaggle Fake news, Politifact News Dataset and the Liar-Liar dataset; I have applied various cleaning techniques like Stemming, Lemmatization and tokenization.

This was followed by feature engineering based on the content statistics factor.

 The lengths of the textual columns like “Content Length” and “Title Length” have been taken to create new features and normalize them.

Later, the median, mean and variance of these textual columns have been calculated to generate new features called “applicability\_text” and “applicability\_title” which is formulated as the following expression which closely represents the standard deviation:

Applicability

 = normalized\_length\*mean(actual\_text\_length)

**Model Evaluation:**

The features that we so generated have been fed to various machine learning algorithms to check how impactful these new features have been on the performance of the model. I have applied Gaussian Naive Bayes, K-NN, Support Vector machines, Decision Tree classifier, XGBoost Classifier & MLP classifier.

As far as evaluation metrics are concerned; Have used cross validation, precision score, recall score, ROC-AUC, F1 score, accuracy and confusion matrix.

**Interpretation of Results:**

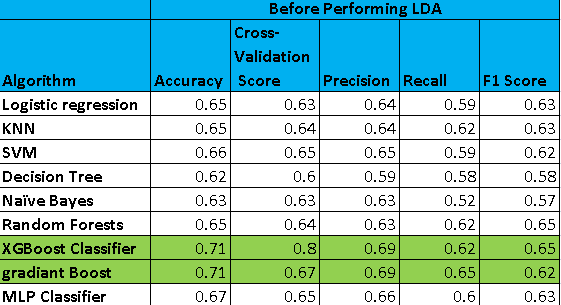
XGBoost classifier has given the best results for the given data with an accuracy score of 73%. The Type I and Type II errors generated from the confusion matrix have been the least. Looking at the F1 score; which is a combined representation of both precision and recall is 73%; showcasing close consistency.

Another approach to distill the dataset is running LDA. LDA ensures that the most frequent words have been to be taken; as they too qualify for content statistics.

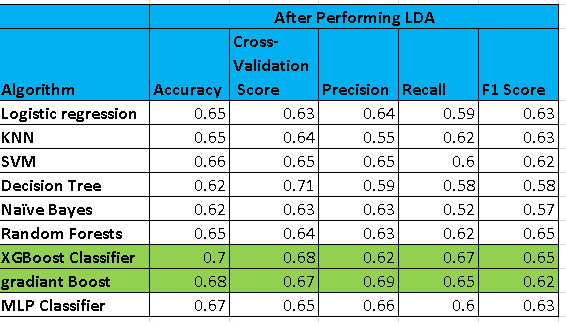
However, the accuracies have been merely the same before and after performing the LDA on the features and hence, it can be stated that there has not been much improvement.

**Results Interpretation:**

Before Performing LDA

****

After Performing LDA:



***FACTOR 2: Title vs Body***

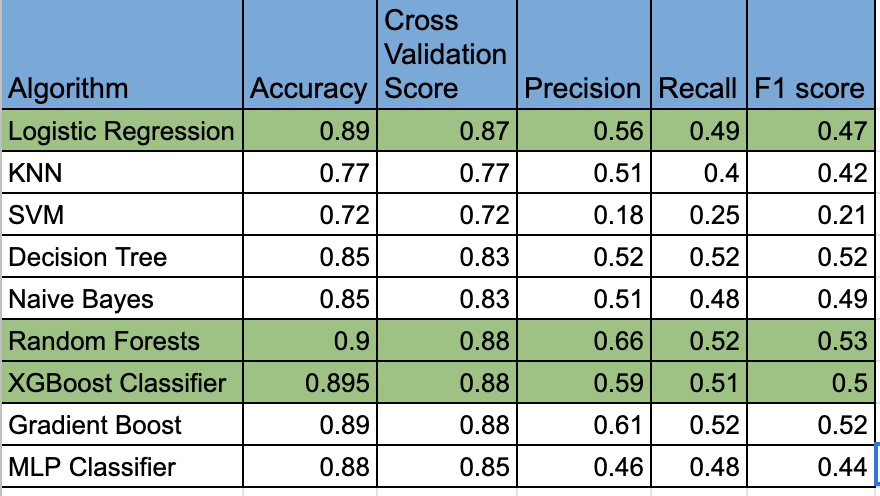
Title vs Body is another major factor which is really useful in identifying if the news is fake or real. For this factor I have used three datasets. I have amalgamated Kaggle fake news dataset and Politifact news dataset and performed all the steps mentioned below. I have also taken the Stance dataset and performed the steps mentioned below.

* Performed data cleaning on the title and text columns that include Removing symbols, punctuation, stop word removal, Lemmatization, Stemming, etc.. Generated n-grams (unigram, bigrams and trigrams).
* Content statistics that include calculating the length of sentences, count of n-grams and count of overlapping n-grams between title and text.
* I then simplified and enhanced my TF-IDF algorithm and calculated cosine similarities between the title and text.
* Used SVD to perform dimensionality reduction on TF-IDF vectors and to perform Latent Semantic Analysis (LSA) as part of topic modelling.
* Performed cosine similarities between the title and text topics generated from above step.
* Used Word2Vec using Google News corpus to find synonyms and next probable words on both title and text and calculated the similarities from the outcome.
* Performed Sentiment analysis on both title and text was extracted the sentiment scores for (pos, neg, neutral).
* Performed normlization on all the distilled features using Standard Scaler.

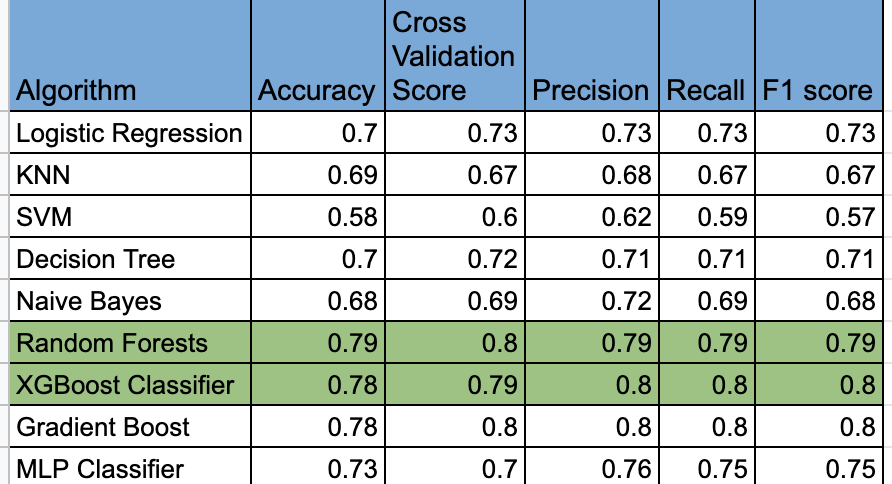
Ran all the above steps on two datasets. One with a multiclass label and the other on the binary classified amalgamated data set.

**Interpretation of Results:**

Ran all the models on Stance Dataset which is a multi class label. Random Forest and XGBoost gave best results with an accuracy of 90%



Ran the models on the amalgamated dataset of kaggle fake news and politifact dataset which is a binary class label. Compared to other models Random Forest and XGBoost performed better with an accuracy of 79%.



Comparing models on binary and multiclass labels, multiclass labels gives good accuracy.

Analyzed the results and derived a polynomial equation to generate the fakeness score by taking the predictions from both multi classification and binary classification models. Randomly picked a few samples and tested them by inputting them to the prediction function which then generated the overall fakeness score.

**FACTOR3: Misleading Intentions**

Mis- leading intentions is described as the diverted or false sentence/text i,e misleading someone when you point them in the wrong direction, metaphorically or literally. Below are some examples:

Example: If you give a stranger/passenger direction away from the place he’s looking for, then you are misleading him.

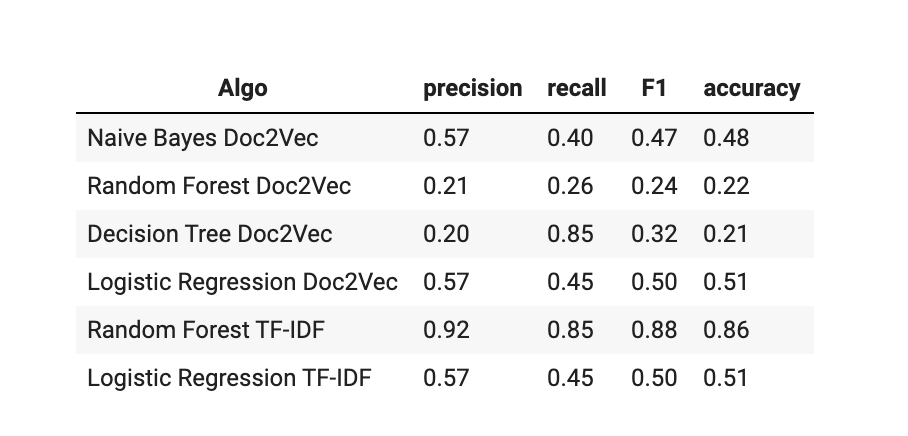
Predicting whether the sentence/test is misleading or not. This could help to save from the false intentions. Using this feature, we can predict if the given content is honest or dishonest. Below are steps followed.

* Adding labels to the dataset.
* Removing Special Characters and Punctuations.
* Converting characters to Lowercase.
* Removing Stop Words
* Checking for null values and dropping the non-required columns.
* One hot encoding for converting categorical values to numeric values.
* Checking for null values and dropping the non-required columns.
* Lemmatization for grouping together the different inflected forms of a word so they can be analyzed as a single item.
* Stemming for producing morphological variants of a root/base word.
* Created the class of the whole feature so that the class can be used on any dataset easily.
* The newly created class has all the preprocessing in place.
* later created an object of that class and tested against a text/test data.
* Analyzed the results and derived a polynomial equation to generate the misleading accuracy by taking the predictions from multi classification models. Randomly picked a few samples and tested them by inputting them to the prediction function which then generated the overall misleading accuracy.

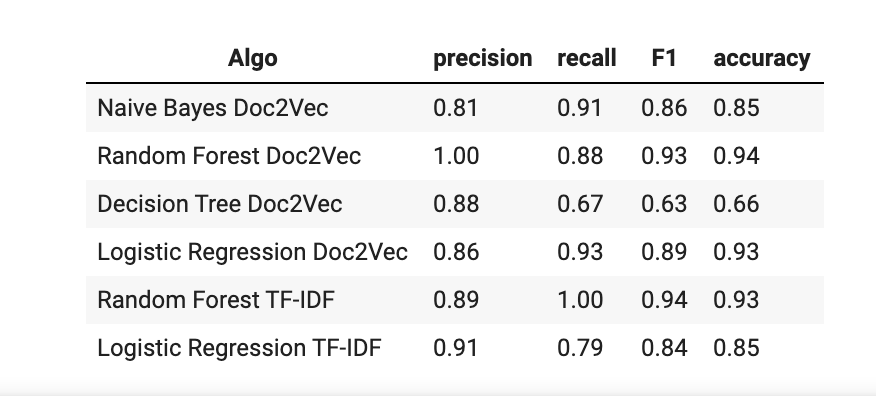
**Interpretation of Results:**

Ran all the models on Dataset which is a multi class label. and Random Forest gave best results with an accuracy of 94%

Ran all the model on original dataset and below is the comparison accuracy table

****

and below table represent After dataset amalgamation result



So it has been realised that after amalgamation The accuracy has increased.

**FACTOR4: Writing Style**

Writing style of any article depicts many things about the content. By examining the writing style of the content, one can determine whether the content is True or False.

There are some features that we can use to examine the writing style of a news content.

When a person tries to write a fake message in a news content, the sentiment polarity of the content is usually high. True articles have a neutral sentiment. To identify the fakeness of any article we can examine the grammatical mistakes, Writing quality and professionalism of the article.

To implement the writing style factor, we first cleaned the data by applying lemmatization, stemming, removing stop words, etc. Followed by which we checked the writing quality of a content by checking the number of contracted words used in the content. Further we performed Sentiment Analysis to determine the sentiments generated by the content. We considered the compound score in sentiment analysis. A compound score is a metric that calculates the sum of all lexicon ratings which have been normalized in range -1 to +1. Here, -1 depicts the most extreme negative and +1 depicts most extreme positive. A positive sentiment has compound score greater than or equal to 0.05 whereas the neutral sentiment has compound score between -0.05 to 0.05. Negative sentiments have compound score less than or equals -0.05. Topic modelling using LDA is also performed. We created Bag of words corpus of the text feature in the dataset and applied the LDA model to it. We considered the LDA score generated by the model to analyze the writing style of a content.

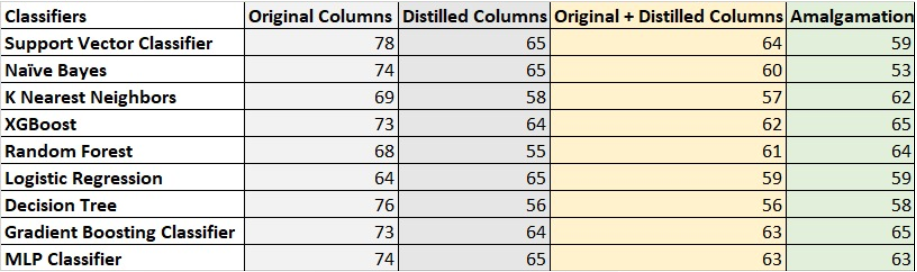
We trained the model using the dataset updated with the scores generated by those features. We used classification algorithms such as the Support vector machine, Naïve Bayes, and K- nearest neighbors.

The accuracy of this factor is within the range of 49% to 55%.

**Model Interference and Results:**

Accuracies have been measured on original features of the dataset as well as the distilled columns. It is observed the original features with cleaned data have performed better compared to other distilled and amalgamated columns.

However, I could achieve an accuracy of 65% using MLP classifier after performing distillation.



VI. Conclusion:

Generated a polynomial equation which takes the prediction of our model as input along with its weighted average. Whenever we input a text to that model; it will predict whether the given text is fake or not by considering all our features.

VII. References:

1. <https://arxiv.org/abs/1705.00648>
2. <https://www.smartinsights.com/content-management/content-marketing-strategy/essential-content-marketing-statistics/>
3. <https://en.wikipedia.org/wiki/Wikipedia:Statistics>
4. <https://www.theguardian.com/books/2008/sep/25/writing.journalism.news>
5. <https://github.com/aarsanjani/AlternusVera>