# CS 579: Online Social Network Analysis

## Project II- FAKE NEWS CLASSIFICATION

Prof. Kai Shu

YASH PATEL(A20451170) NIRAV SONI(A20435538)

### 1. Introduction:

As an expanding measure of our lives is spent connecting on the web through online media stages, an ever-increasing number of individuals will in general search out and devour news from web-based media as opposed to customary news associations. The broad spread of phony news can have a genuine negative effect on people and society. To help mitigate the negative effects caused by fake news we develop methods to automatically detect fake news.

### 2. Methods and Design:

Here, we are using two methods naïve bayes and deep neural network for better efficiency and outcomes.

Data Splitting into train and validation:

- We are training model on 90% of training data 10% of trained data to validate.

Data Preprocessing:
To train and test our model we first apply some data processing on input data which is explain as below.

- Firstly, we convert articles in to list of sentences by splitting and create corpuses of articles.
- Convert all sentences of corpuses to lower case.
- Remove all non-word characters and useless all punctations in the corpuses.
- Create tokens of articles using these corpuses.

Models:

1. Naïve Bayes

- To create Naïve Bayes model, we are using train data and validation data to create a model where training data used to train model and validation data used to validate model.
- Using train data, we calculate label wise (agreed/disagreed/unrelated) frequency of each token/word in processed data for both fake and upcoming article.
- Using this calculated frequency, we are predicting probability of labels (agreed/disagreed/unrelated) and classify instance (fake article and upcoming article) of test data.
- We are getting accuracy about 74.38% on training data.
- During validation of this model we got 71.75% accuracy on test data.

2. Multilayer Perception

Additionally, for preprocessing, we are using One-hot encoder for categorical variables where no such ordinal relationship exists, the integer encoding is not enough.

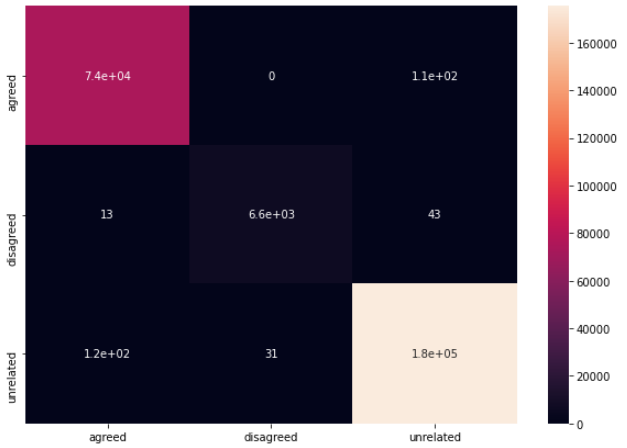
A multilayer perceptron (MLP) is an artificial neural network, with an input layer, one or more hidden layers, and an output layer. MLP can be as simple as having each of the three layers; however, in our experiments we have fine-tuned the model with various parameters and number of layers to generate an optimum predicting model. We used classifier with 1 hidden layer with 100 neurons.

- We calculate frequency of each token/word in processed data.
- We take 40,618 most frequent word to create feature vectors.
- To train this model we take feature vector size of 81,237,
  - First and last 40,618 features are tf-idf (bag of words) of fake and upcoming articles.
  - One 1 feature is for cosine similarity between tf-idf vector of fake and upcoming article.
- We are getting accuracy about 99.83% on training data.
- We achieve accuracy for this model 86.218% while testing on Kaggle.

## 3. Results:

- The F1 score is the harmonic mean of the precision and recall. The highest possible value of an F-score is 1, indicating perfect precision and recall, and the lowest possible value is 0, if either the precision or the recall is zero.
- Accuracy is the most intuitive performance measure, and it is simply a ratio of correctly predicted observation to the total observations.
- Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.
- Recall is the ratio of correctly predicted positive observations to all observations in actual class.

For Multilayer Perception:




| | agreed | disagreed | unrelated |
|---|---|---|---|
| agreed | 74128 | 0 | 110 |
| disagreed | 13 | 6550 | 43 |
| unrelated | 122 | 31 | 175445 |

A chart and data represent about numbers of counts of agree, disagree and unrelated label data.

| | precision | recall | fscore | support |
|---|---|---|---|---|
| **agreed** | 0.998182 | 0.998518 | 0.998350 | 74238.0 |
| **disagreed** | 0.995289 | 0.991523 | 0.993403 | 6606.0 |
| **unrelated** | 0.999129 | 0.999129 | 0.999129 | 175598.0 |

Label-wise Precision, recall, F-score for Multilayer Perception Classifier.

For Naïve Bayes:


| | agreed | disagreed | unrelated |
|---|---|---|---|
| **agreed** | 58499 | 856 | 14883 |
| **disagreed** | 108 | 3941 | 2557 |
| **unrelated** | 41776 | 6139 | 127683 |

A chart and data represent about numbers of counts of agree, disagree and unrelated label data.

| | precision | recall | fscore | support |
|---|---|---|---|---|
| **agreed** | 0.582758 | 0.787993 | 0.670011 | 74238.0 |
| **disagreed** | 0.360369 | 0.596579 | 0.449322 | 6606.0 |
| **unrelated** | 0.879826 | 0.727132 | 0.796225 | 175598.0 |

Label-wise Precision, recall, F-score for Naïve Base Classifier.

| Model | Training Accuracy | Testing Accuracy |
|---|---|---|
| **Multilayer Perception** | 99.83% | 86.218% |
| **Naïve Bayes** | 74.38% | 71.75% |

Accuracies on models

With the clean and preprocessed dataset, the testing phase of the algorithms begins, where Naive Bayes, and MLPClassifier (Multilayer Perceptron) are tested. Among the tested algorithms, the one that has presented the most satisfactory was the **MLPClassifier**.

## 4. Conclusion:

The need to battle Fake News and its results in the public eye, affecting individuals' reasoning and dynamic, were the underlying reason for the improvement of this work so it could distinguish and battle these sorts of information. A few examinations are being led and various advancements are being acquainted with manage the phony news. Be that as it may, per users need to search

for dependable sources and recognize genuine and bogus news. In this investigation, the proposed arrangement was to utilize the Multi-layer Perceptron (MLP) neural organization calculation that got palatable outcomes as found in the outcomes area, to make a model that had the option to get familiar with the examples of preprocessed messages and for distinguish bogus news.

## 5. References:


[1] https://scholarspace.manoa.hawaii.edu/bitstream/10125/59664/0224.pdf

[2] https://stackabuse.com/python-for-nlp-creating-bag-of-words-model-from-scratch/

[3] https://janav.wordpress.com/2013/10/27/tf-idf-and-cosine-similarity/

[4] https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2761239.pdf

[5] https://www.hindawi.com/journals/complexity/2020/8885861/