COSC 4931/5931: Topics in Computer Science: Social and Collaborative Computing

Project 1: Fake news detection

Due: Friday, 4th March 2022 at 11:59pm.

Overview and Logistics

In this project, you will apply machine learning algorithm(s) on news articles to detect fake news articles. You will be evaluated on the following criteria:
1. Strategic preprocessing of the data
2. Logical explanation of your choice of features and models.
3. Evaluation of your classifier
4. Visual representation of the performance of your classifier.

Background

Social media platforms such as Facebook and Twitter are extremely powerful and useful for their ability to allow users to discuss and share ideas and debate over thousands of issues on various domains. However, these platforms can also be abused for monetary gain, creating biased opinions, manipulating mindsets, and spreading satire or absurdity. The phenomenon is commonly known as fake news. There has been a rapid increase in the spread of fake news in the last decade, most prominently observed in the 2016 US elections. Such proliferation of sharing articles online that do not conform to facts has led to many problems not just limited to politics but covering various other domains such as sports, health, and also science. A recent study shows that social media platforms are likely to influence the spreading of fake news and the formation of echo chambers. In this project, you are tasked to understand the underlying patterns that exist in the content of fake news articles and apply machine learning algorithm to identify them. The dataset you are provided with contains articles from various news websites and blogs. There are 20,387 articles as training data and 5127 articles as test data. To get an idea of what algorithms have been studied to detect fake news, check out this link.

This project is to be completed in a team of two members.

News article analysis

In this assignment you are supposed to build a classifier that can distinguish between legitimate and fake news. Download the training and test csv files from D2L. Your submission would include a three-page report that discusses how you built the classifier, and then presents the performance of your classifier. The classifier would be based on text features extracted from the articles.

Note: you do not have to write a classifier from scratch. You are free to use one or more of the many open-source (or other) tools and packages that allow you to use a variety of different classifiers. You can pick any package or programming language you like or are most comfortable with.

The actual deliverables

Every team will hand-in a report summarizing their findings. The expected length is about 3 pages, composed as described below. When writing up your findings, look to the research papers we have read in class. For each one, please *have a short Method section and a longer Results section*. The Method section describes what data you used and how you got it. The Results section describes your analysis and contains your graphics.

Contents of the report

1. **Feature construction:** Constructing a classifier involves extracting relevant and meaningful features from the data under consideration. In your report, you will need to first present the various features you derived

from the textual content of the original and fake news items. Features can include (but not limited to) unigrams, bigrams, TF-IDF, Part-of-Speech tags, length of words etc. of the messages. Also you will need to discuss why you chose the particular set of features.

2. Description of the classifier: Discuss what is the particular classifier you chose (e.g., Support Vector Machine, Naive Bayes, Random Forest, or some other), and a justification or rationale behind its choice and applicability to the dataset in question. That is, if you picked classifier X, why is it a good fit for this problem? Why is it a better choice compared to another classifier Y?

3. Evaluation technique: Present how you evaluated how well your classifier of choice performed in distinguishing between original and fake news. You will also need to present what metrics you used to evaluate performance of the classifier. For instance, typical metrics would include percentage accuracy, precision, recall, and F-score.

4. Implementation:

- **a.** Discuss how you preprocessed your data. If you used stopword removal, stemming, or tokenization over the content of the messages, you need to report it here. Point to the particular libraries or functions you needed for this.
- **b.** Discuss how you extracted the features you presented above from the news dataset. This needs to include what libraries and which particular functions you used for extracting each feature. If you did not use an existing library, you need to write about the method you used to compute the features from the data. Report if you did some filtering or feature selection to disregard not-so- common features (e.g., if you ignored all unigrams which occurred less than five times). Also report if you did any kind of normalization or standardization of each feature, and your justification behind doing or not doing so.
- **c.** Next, discuss how you implemented/used a library for your chosen classifier. Report what were the inputs and outputs to the particular library function you used and if/how you tuned parameters of the classifier (e.g., if you chose SVM, report the particular kernel you used).
- **d.** Discuss how you partitioned the dataset for *k-fold* cross validation, along with what was your chosen *k* here. Here you will also discuss based on your chosen *k-fold* cross validation setup, what were your training and test sets in each of the *k*-iterations.
- **e.** Discuss how you calculated the metrics of performance evaluation, e.g., accuracy, precision, recall etc. It is again okay to use an existing library that gives precision and recall values, in which case you need to present in your report which libraries/functions you used for the purpose, and what was your input and output to those functions.

5. Analysis of results: Report the performance of your classifier based on the above discussion. You will need to use charts, graphs, or tables to report actual numbers--i.e., the values of the 2 performance metrics you chose above (accuracy, precision, recall etc.). These numbers should be reported for each of the *k* iterations of the *k-fold* cross validation setup. You should also report the average performance over all *k* cross validation folds, corresponding to each evaluation metric.

6. Extra credit (up to 2%):

- **a.** There will be extra credit for extracting novel text-based features from the news articles (don't be afraid to be creative here.
- **b.** There will be additional extra credit for comparing (per the above evaluation technique) two or more classifiers in their ability to distinguish the two sets, original and fake, when applied to the same data. For instance, you can compare your chosen classifier SVM's performance over another classifier Naive Bayes, and in the analysis section discuss which classifier performs better based on your chosen evaluation metrics like accuracy, precision, recall.

Grading

10%	Preprocessing
30%	Feature construction
30%	Implementation
15%	Evaluation
15%	Analysis of results