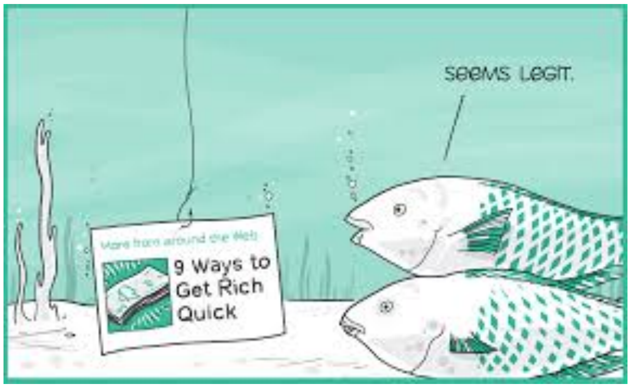
DS 310 Machine Learning

Fall 2017 / Dongwon Lee

Project #1: Clickbait Tweet Classification (200 points)

DUE: **October 8 SUNDAY 11:59PM** (Canvas Project #1 dropbox)



NOTE: This is a group project. All group members receive the same points.

Task: The aim of this project is to build an accurate binary classification model that can discriminate clickbait tweets from legitimate ones with high F-measure and AUC scores.

Here is a Wikipedia definition for clickbait: https://en.wikipedia.org/wiki/Clickbait. Each data instance consists of two parts: (1) tweet itself, and (2) actual news article (linked from the tweet). You are free to use ANY features that you can think of, and any tools/libraries/software in any programming languages.

Dataset: You can download datasets from here: **https://goo.gl/YA96nt**:

`dataset_no_figure.zip`	: main dataset without photos (37MB)
`dataset.zip`	: main dataset + photos (1GB)
`sample_training.arff`	: sample Weka ARFF file with 5 example features

If you will extract features from photos, use `dataset.zip`. Else, use the smaller `dataset_no_figure.zip`. If you unzip either dataset file, you get 3 files as follows:

`instances_train.jsonl`	: training data
`truth_train.jsonl`	: labels for training data
`instances_test.jsonl`	: test data (unlabeled)

These are all line delimited JSON file (http://jsonlines.org/). Each line is a JSON-Object containing the information that we have extracted for a tweet post and its target news article. The JSON schema for both training and test data are shown below:

```
### Fields in instances.jsonl:
{
  "id"                 : "<instance id>",
  "postTimestamp"      : "<weekday> <month> <day> <hour>:<minute>:<second>
                          <time_offset> <year>",
  "postText"           : ["<text of the tweet post with links removed>"],
  "postMedia"         : ["<path to a file in the media archive>"],
  "targetTitle"       : "<title of target news article>",
  "targetDescription": "<description tag of target news article>",
  "targetKeywords"    : "<keywords tag of target news article>",
  "targetParagraphs"  : ["<text of the i-th paragraph in the target
                        article>"],
  "targetCaptions"    : ["<caption of the i-th image in the target article>"]
}

### Fields in truth.jsonl:
{
  "id"                : "<instance id>",
  "truthJudgments"    : [<number in [0,1]>],
  "truthMean"         : <number in [0,1]>,
  "truthMedian"       : <number in [0,1]>,
  "truthMode"         : <number in [0,1]>,
  "truthClass"        : "clickbait | no-clickbait"
}
```

Your task is to train your model using the training samples from `instances_train.jsonl` along with their labels from `truth_train.jsonl`. For instance, if you use Weka, you have to first generate an input training data file in ARFF format using both `instances_train.jsonl` and `truth_train.jsonl` files. As an example, we extracted 5 features and saved them in `sample_training.arff` file, where features are:

- "word count" NUMERIC
- "average word length" NUMERIC
- "length of the longest word" NUMERIC
- "whether start with number" {True, False}
- "whether start with who/what/why/where/when/how" {True, False}

These features may or may not be useful for actual modeling. I expect some teams to have hundreds of features together to get a good result. You may borrow some ideas about effective features and models from these clickbait research articles:

- Potthast, M., Köpsel, S., Stein, B., & Hagen, M. (2016, March). Clickbait detection. In European Conference on Information Retrieval (pp. 810-817). Springer International Publishing.
- BIYANI, P.; TSIOUTSIOULIKLIS, K.; BLACKMER, J. (2016, Feb) "8 Amazing Secrets for Getting More Clicks": Detecting Clickbaits in News Streams Using Article Informality. AAAI Conference on Artificial Intelligence, North America.
- Chakraborty, A., Paranjape, B., Kakarla, S., & Ganguly, N. (2016, August). Stop clickbait: Detecting and preventing clickbaits in online news media. In Advances in Social Networks Analysis and Mining (ASONAM), 2016, IEEE/ACM International Conference on (pp. 9-16). IEEE.
- Wei, W., & Wan, X. (2017). Learning to Identify Ambiguous and Misleading News Headlines. arXiv preprint, arXiv:1705.06031.

You need to first contemplate over many possible features, extract them from the given dataset, and play with different models and their (hyper)parameters. This step is called "Feature Engineering/Optimization," arguably the MOST critical step of your project.

Once your model training is over, for each test instance from `instances_test.jsonl` file, run your model, figure out (ie, "predict") the corresponding label ("clickbait" or "no-clickbait"), and submit your final prediction result.

Deliverables: Each team needs to upload the following THREE to Canvas:

1. Project report **in PDF** that contains all the details of the major steps of the project such as cover page (with team name and member names), data pre-processing, feature engineering, model building and comparison, performance evaluation, and analysis of your choices (of features and models) and results. Make your report self-contained such that after reading your report, a student in the data science degree program should be able to replicate your results. Page is limited to 10 pages using reasonable formatting (including cover and all appendix). As such, within the limit, provide all important details of your project as clear/concise as possible. Do not make your report unnecessarily long with figures or codes (ie, no need to have 10-page report).

2. Prediction result saved as `prediction.csv` file that has the following contents:

```
id,label                    ← this "id" value must match the "id" value
                              from instances_test.jsonl
609300383358418945,0
609095030926635011,1        ← 0=no-clickbait, 1=clickbait
…
```

   In this example, your prediction result says the test instance with the id 609300383358418945 in the test file `instances_test.jsonl` is *not* a clickbait, while another one with the id 609095030926635011 is a clickbait. The order of the rows is not important.

3. Prediction results and extracted features saved as `prediction.arff` file that has the following contents:

```
% comment here              ← indicate which machine learning
                              model and parameters that you used
                              to get your predicted labels in prediction.csv

@RELATION team-foo          ← team name here

@ATTRIBUTE "id" numeric     ← this "id" value must match the "id" value
                              from instances_test.jsonl
@ATTRIBUTE "feature 1" …
@ATTRIBUTE "feature 2" …
…
@ATTRIBUTE "feature k" …
@ATTRIBUTE "label" {0, 1}   ← 0=no-clickbait, 1=clickbait

@DATA
```

```
609300383358418945, feature 1 value, …, feature k value, 0
609095030926635011, feature 1 value, …, feature k value, 1
…
```

In the above example, your prediction result says the tweet instance with the id 609300383358418945 in the test file `instances_test.jsonl` is *not* a clickbait, while another one with the id 609095030926635011 is a clickbait.

While the above `prediction.csv` file has your predicted labels (ie, your answers), using this ARFF file with both features and predicted labels, we will test if we are able to get similar results in Weka as you have reported. If you used different software/tools to work on your models (other than Weka), we understand that your ARFF file cannot truly capture what you did in other software/tools, so approximation is ok here. For instance, when we grade your prediction using `prediction.csv`, we get F-measure=0.85. Then, when we run this ARFF file using the extracted features and model/parameters (as you indicate), if we get 0.83, that is acceptable. However, if the difference of performance is substantial, say, we get 0.77, then we may suspect that you got your prediction labels incorrectly. In those cases, we will ask for a live demo to verify the results.

Grading Rubric (200 points):

- 100 points: performance measured by both **F-measure** and **ROC Area (= AUC)** using your `prediction.csv` file
- 20 points: if we are able to load your `prediction.arff` file into Weka, compute evaluation measures using the model/parameters that you indicate, and verify your performance approximately
- 80 points: quality of report

Grading Exceptions

The goal of this project is to encourage you to emerge yourself in a real-life data science process to learn the materials better in a hands-on fashion. **DO NOT CHEAT IN ANY WAYS**. If you do not attempt to get your hands dirty in this project, you will not learn much.

All team members need to participate in the project and contribute as much as he/she could. Each is expected to bring different skills on the table—ie, some are good in programming, while others in analytics, yet others are more creative, etc. Find out what you can contribute within the team, and play the team work. While all members get the same points by default, I will intervene in an exceptional situation (eg, one member does not participate at all).