Assignment 3

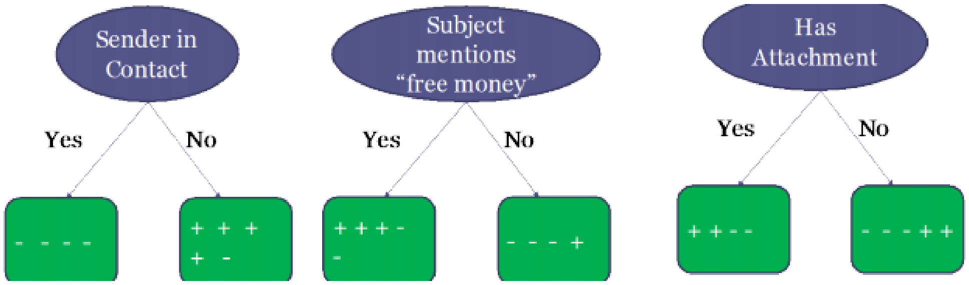
Jiarong Ye

October 16, 2018

Q1

(1) (20 points) The figure below shows three possible feature test for the root node of a decision tree to predict spam e-mail messages (based on the example discussed in class).

- (a) Calculate the expected information gain for each feature test (please show your formula).
- (b) Which feature test will be selected by the Decision Tree Learning algorithm?



(a)

- entropy before split: $-\frac{4}{9}\log\frac{4}{9} - \frac{5}{9}\log\frac{5}{9} = 0.991$
- entropy after split:
  - feature set 1: `Sender in Contact`
    - $-\frac{4}{9}\left(\frac{4}{4}\log\frac{4}{4} + 0\right) - \frac{5}{9}\left(\frac{4}{5}\log\frac{4}{5} + \frac{1}{5}\log\frac{1}{5}\right) = 0.401$
    - info gain (entropy reduction) = $0.991 - 0.401 = 0.590$
  - feature set 2: `Subject mentions 'free money'`
    - entropy after split: $-\frac{5}{9}\left(\frac{3}{5}\log\frac{3}{5} + \frac{2}{5}\log\frac{2}{5}\right) - \frac{4}{9}\left(\frac{3}{4}\log\frac{3}{4} + \frac{1}{4}\log\frac{1}{4}\right) = 0.900$
    - info gain (entropy reduction) = $0.991 - 0.900 = 0.091$
  - feature set 3: `Has Attachment`
    - entropy after split: $-\frac{4}{9}\left(\frac{2}{4}\log\frac{2}{4} + \frac{2}{4}\log\frac{2}{4}\right) - \frac{5}{9}\left(\frac{3}{5}\log\frac{3}{5} + \frac{2}{5}\log\frac{2}{5}\right) = 0.984$
    - info gain (entropy reduction) = $0.991 - 0.984 = 0.007$

(b)

Therefore, the info gain:

$$\text{Sender in Contact} > \text{Subject mentions 'free money'} > \text{Has Attachment}$$

so the feature with largest info gain **Sender in Contact** should be chosen

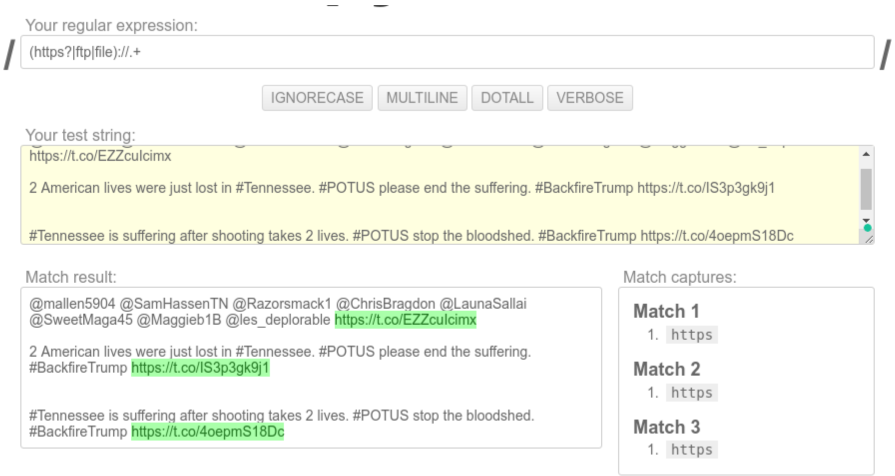
Q2

(2) (20 points) Construct a regular expression as the value for the parameter token_pattern so that CountVectorizer can extract hashtags, twitter user names (e.g., @realDonalTrump), and words from tweets as tokens. Explain how you construct the regular expression.

regular expression:

- first clean the urls using:

(https?|ftp|file)://.+



Your regular expression:

```
/ (https?|ftp|file)://.+ /
```

Your test string:

```
https://t.co/EZZculcimx

2 American lives were just lost in #Tennessee. #POTUS please end the suffering. #BackfireTrump https://t.co/IS3p3gk9j1


#Tennessee is suffering after shooting takes 2 lives. #POTUS stop the bloodshed. #BackfireTrump https://t.co/4oepmS18Dc
```

Match result:

```
@mallen5904 @SamHassenTN @Razorsmack1 @ChrisBragdon @LaunaSallai
@SweetMaga45 @Maggieb1B @les_deplorable https://t.co/EZZculcimx

2 American lives were just lost in #Tennessee. #POTUS please end the suffering.
#BackfireTrump https://t.co/IS3p3gk9j1


#Tennessee is suffering after shooting takes 2 lives. #POTUS stop the bloodshed.
#BackfireTrump https://t.co/4oepmS18Dc
```

Match captures:

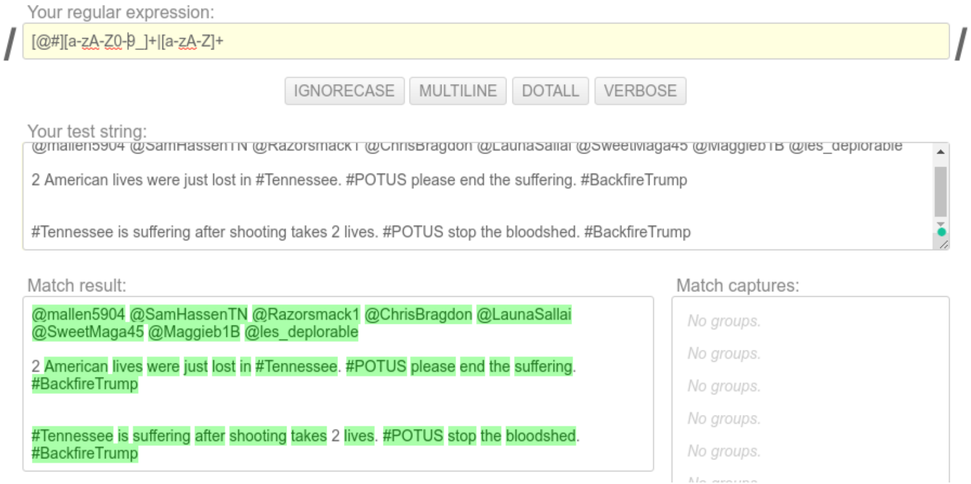
Match 1
1. `https`

Match 2
1. `https`

Match 3
1. `https`

- then extract the hashtags, usernames and words in tweets using:

[@#][a-zA-Z0-9]+ | [a-zA-Z]+



```
/ [@#][a-zA-Z0-9_]+|[a-zA-Z]+ /
```

```
@mallen5904 @SamHassenTN @Razorsmack1 @ChrisBragdon @LaunaSallai @SweetMaga45 @Maggieb1B @les_deplorable

2 American lives were just lost in #Tennessee. #POTUS please end the suffering. #BackfireTrump


#Tennessee is suffering after shooting takes 2 lives. #POTUS stop the bloodshed. #BackfireTrump
```

```
@mallen5904 @SamHassenTN @Razorsmack1 @ChrisBragdon @LaunaSallai
@SweetMaga45 @Maggieb1B @les_deplorable

2 American lives were just lost in #Tennessee. #POTUS please end the suffering.
#BackfireTrump


#Tennessee is suffering after shooting takes 2 lives. #POTUS stop the bloodshed.
#BackfireTrump
```

Note: I could not figure out a tidy way to extract hashtags, usernames and words in tweets all together while excluding the urls because if the urls do not get cleaned up first, it would break up as individual words that do not make sense at all, such as, 'https://t.co/' → 'https' as a word, 't' as a word, 'co' as word. So I clean up the urls first.*