Identifying an authors writing DNA?

Derek Pyne

The Problem

- Given a text sample, can we identify which of these three prolific horror authors wrote it?

HP Lovecraft

Mary Wollstonecraft

Edgar Allan Poe

Obtain and Scrub

- Dataset part of Kaggle competition
- Needed to first load in Pandas, remove extra quotations, and write back to csv before Databricks upload tool would parse it correctly
- With data in Databricks database, can directly load into Spark Context

```
“id","text","author"
"id21164","Mein Gott do you take me vor a shicken?""
""No oh no"" I replied, much alarmed, ""you are no
chicken certainly not.""","EAP"
"id08820","Perpetual fear had jaundiced his
complexion, and shrivelled his whole person.","MWS"
"id17407","The sun set; the atmosphere grew dim and
the evening star no longer shone
companionless.","MWS"
"id05166","The rain ceased; the clouds sunk behind
the horizon; it was now evening, and the sun
descended swiftly the western sky.","MWS"
"id07779","Nothing, however, occurred except some
hill noises; and when the day came there were many
who hoped that the new horror had gone as swiftly as
it had come.","HPL"
"id14749","The state rooms were sufficiently roomy,
and each had two berths, one above the other.","EAP"
"id10634","It was getting dark, and the ancient
roofs and chimney pots outside looked very queer
through the bull's eye window panes.","HPL"
```

Extract

"How can I thank you, my best and only benefactor?"

Input

["'how","can","i","thank","you,","my","best","and","only","benefactor?"]

Tokenization

Normalize and split on whitespace

word	count
how	1
can	1
I	1
thanks	1
...	...

Count Vector

Count term frequency and build sparse vector

word	IDF
how	1.9
can	2.1
I	0.5
thanks	4.2
...	...

Inverse Document Frequency

Give more weight to terms that are rarer in our corpus

Model

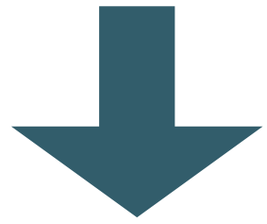
- Assemble previous steps into Spark pipeline
- Use a Naive Bayes Classifier as our learner

★ Uses the conditional probability of each term, given an author, to calculate the probability of an author given a sample of text

★ Assumes independence between terms but still works for comparing likelihood of authors given a sample of text

$$P(A \mid B) = \frac{P(B \mid A)}{P(B)} P(A)$$

$$P(author \mid sentence) = \frac{P(sentence \mid author)}{P(sentence)} P(author)$$



$$P(author \mid \mathbf{t}) = \frac{P(\mathbf{t} \mid author)}{P(\mathbf{t})} P(author)$$

Interpret

- **Hyperparameter tuning**
  - Used a grid search for classifier smoothing factor and count vectorizer dictionary size
  - Cross validation with 80/20 train/test split and 4 folds
  - Achieved a test set accuracy of 76.1 %
- **Next Steps**
  - N-gram features
  - Sentence structure: Are grammatical devices characteristic of an author?
  - Part of Speech Tagging: Can proportions of nouns, adjectives, verbs add value?