Spooky Authors Identification

Edgar Allan Poe

HP Lovecraft

Mary Shelley

Introduction

- Kaggle Data Science Competition
- Small extracts from horror stories written by three authors: Edgar Allan Poe (EAP), Mary Shelley (MWS) , and HP Lovecraft (HPL)
- Data:
    - One Training set (TR0): 19,579 extracts whose authors are known
    - One Test set (TS0): 8,392 extracts whose author must be identified

Goal: **For each extract, give probability to the potential authors (among the three mentioned above) to determine which one is the most likely to be its author**
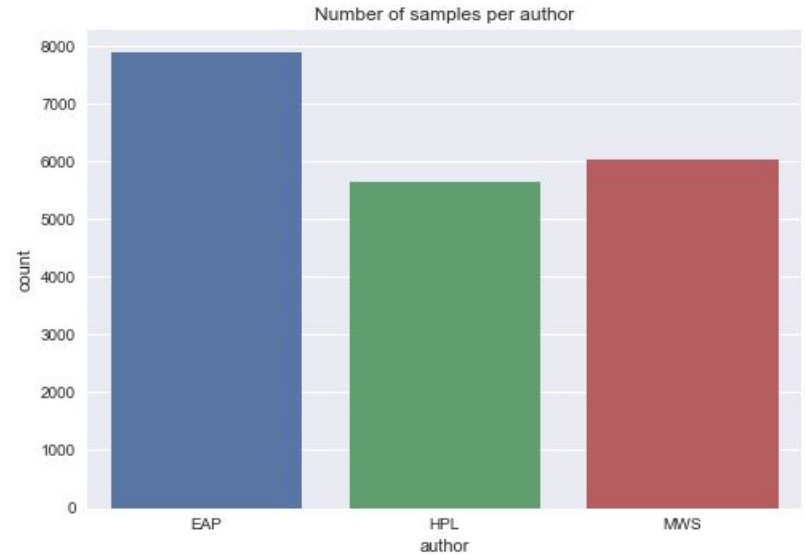
Outline

I.   About the dataset
II.   Our strategy
III. Features and highlights
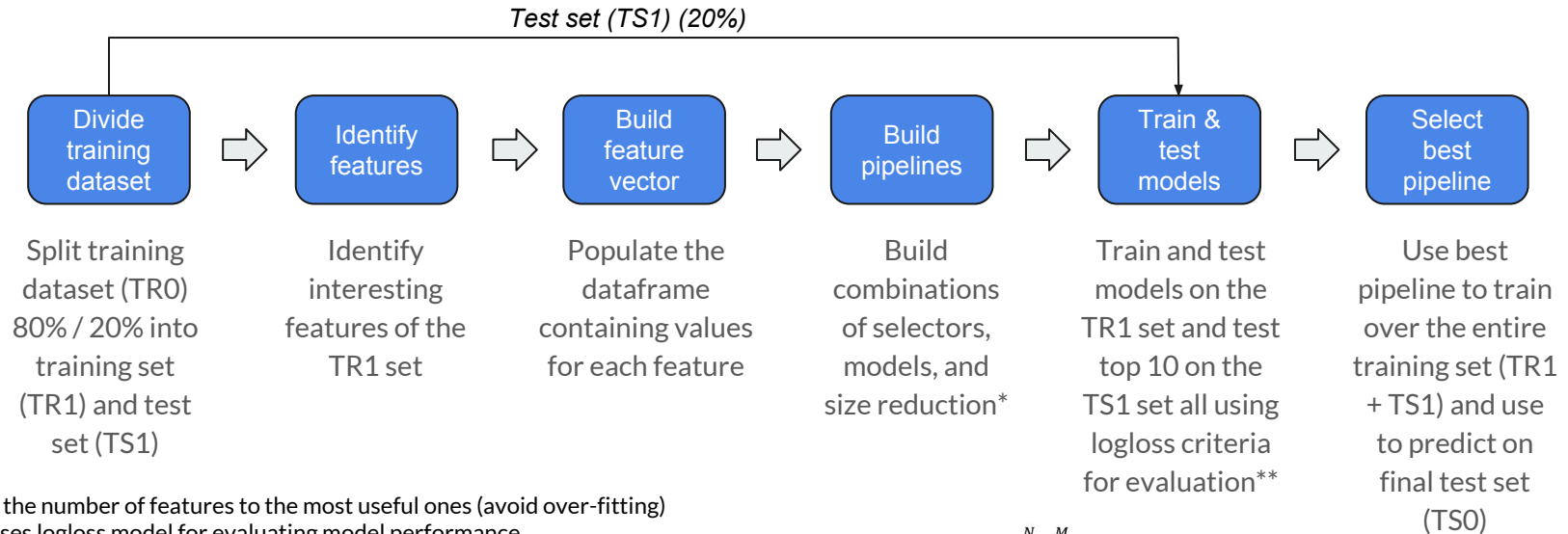IV. Results

About the dataset

- ID - a unique identifier for each sentence
- Text - some text written by one of the authors
- Author - author of the sentence (EAP/HPL/MWS)

Sample extract:

ID	Text	Author
id02499	"Verney," said he, "my first act when I become King of England, will be to unite with the Greeks, take Constantinople, and subdue all Asia.	MWS
id04092	"Upon honor," said I. "Nose and all?" she asked.	EAP

Number of samples per author

Our Strategy - *Structure*

Test set (TS1) (20%)

Divide training dataset	Identify features	Build feature vector	Build pipelines	Train & test models	Select best pipeline
Split training dataset (TR0) 80% / 20% into training set (TR1) and test set (TS1)	Identify interesting features of the TR1 set	Populate the dataframe containing values for each feature	Build combinations of selectors, models, and size reduction*	Train and test models on the TR1 set and test top 10 on the TS1 set all using logloss criteria for evaluation**	Use best pipeline to train over the entire training set (TR1 + TS1) and use to predict on final test set (TS0)

* Reducing the number of features to the most useful ones (avoid over-fitting)
** Kaggle uses logloss model for evaluating model performance

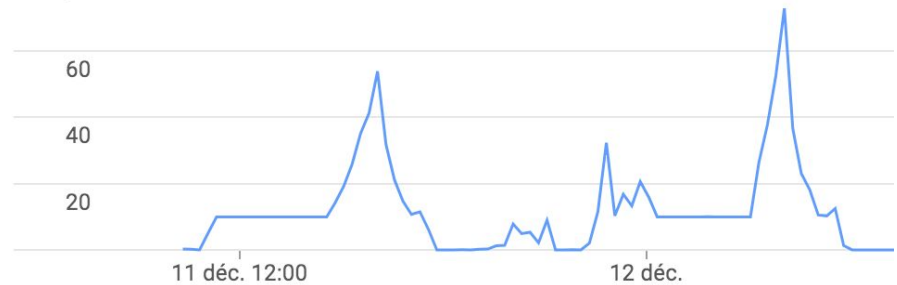$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij})$$

Our Strategy - *Computation on the Cloud*

Use of a Google Cloud instance (with 10 vCPU)

- Quicker: Parallelisation using up to 7 processors
- Less risk of system error: Use of GNU Screen (safely kill the SSH connection)
- Doesn't burn down our laptops

Processeur

% du processeur

Features and Highlights - *(1/4)*

Meta Features	Text Features
1. Sentence length (characters & words) 2. Word length 3. Punctuation density 4. Percentage of unique words 5. Stopword count 6. Noun/adjective/verb density 7. Adjective to noun ratio 8. Emphases on words or phrases 9. Dialogue density 10. Feminine to masculine words ratio 11. Use of foreign languages	1. POS tag of first/last word of a sentence 2. Emotions (NRC data), positive/negative 3. TF-IDF (words n-grams): degree to which an author uses a word more than the two other authors 4. TF-IDF (characters n-grams) 5. TF-IDF (POS tags n-grams)

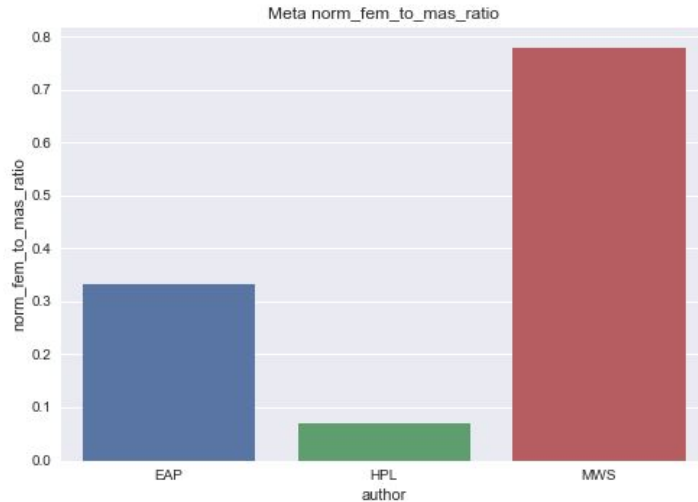# Features and Highlights - *(2/4)*
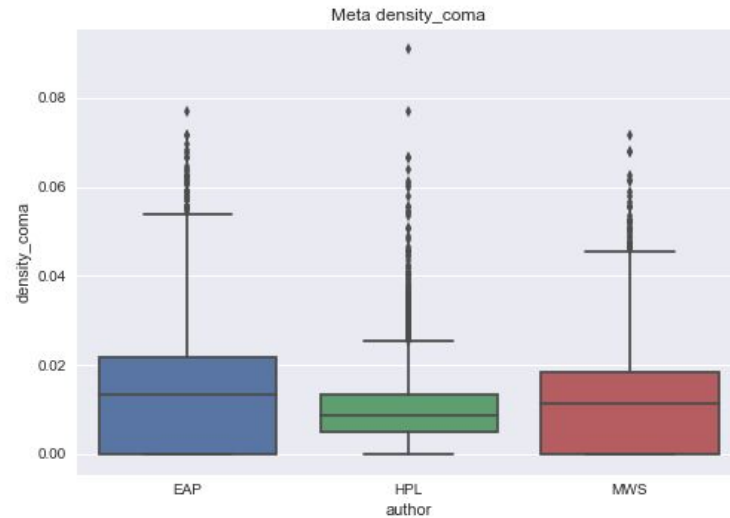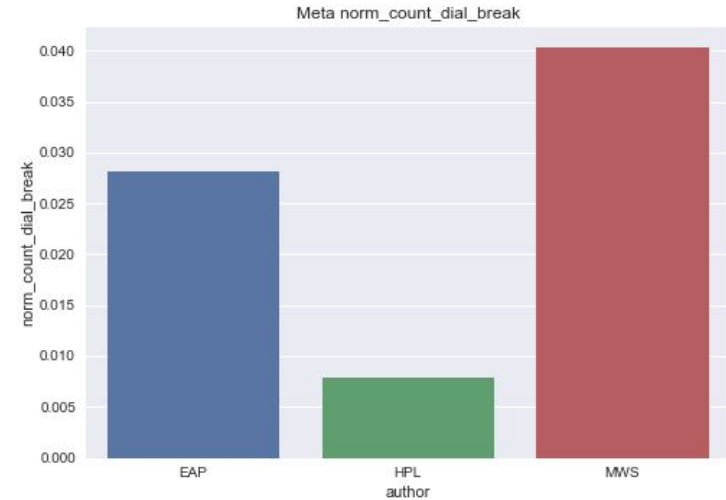

Comma density
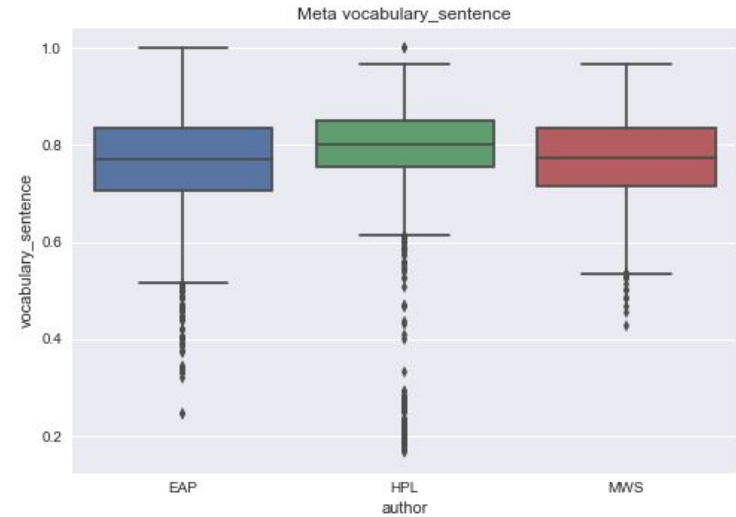
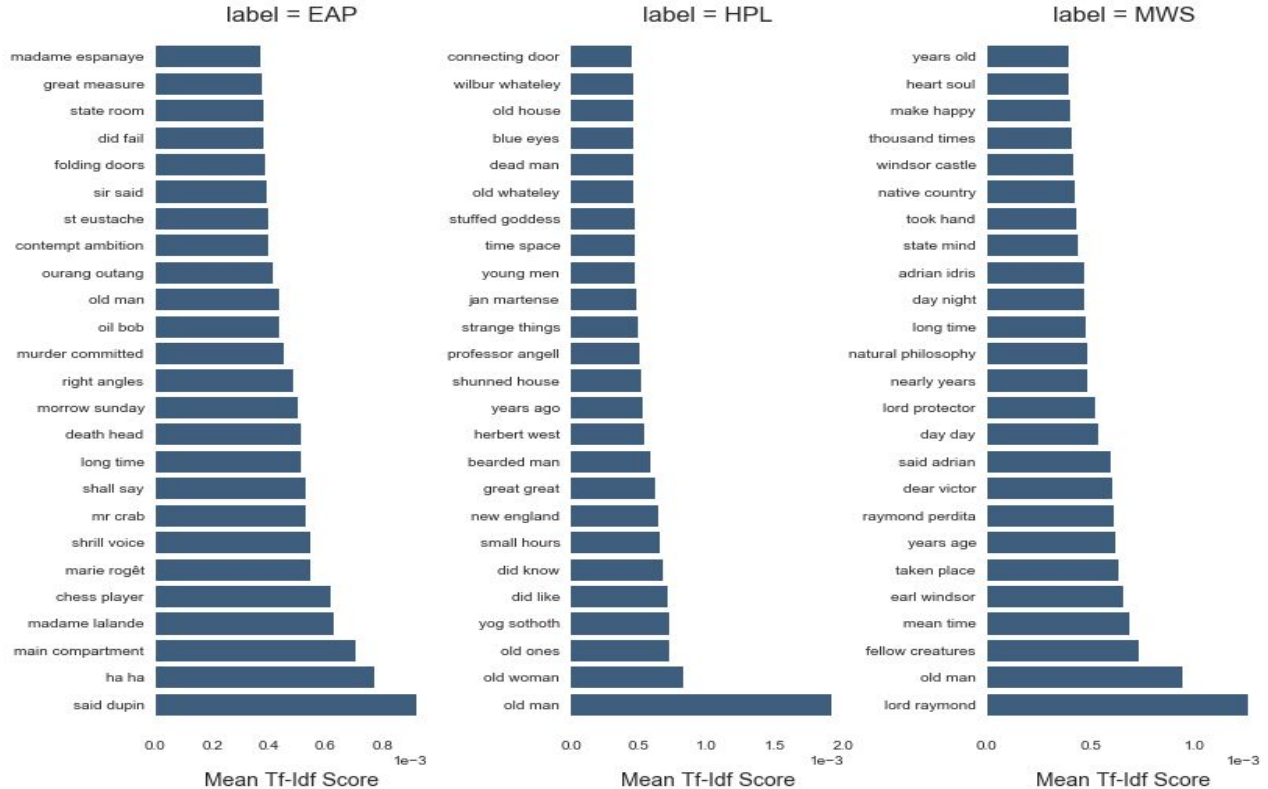Vocabulary Variation

Features and Highlights - *(3/4)*

Feminine to Masculine Word Ratio

Use of Dialogue Breaks

Features and Highlights - *(4/4)*

TF-IDF Bigrams

Pipeline Combinations

Feature Count	X	Feature Selector	X	Predictive Model	
- Exactly 10 - Exactly 20 - One-fourth - Half		- Univariate Feature Selection - Recursive Feature Elimination - Principal Components Analysis		- Logistic Regression - K-Neighbors Classifier - Decision Tree Classifier - Gaussian NB - Gradient Boosting Classifier	- Ada Boost Classifier - Extra Trees Classifier - Random Forest Classifier - Calibrated Bernoulli NB - Calibrated Huber

Results: Pipe Selection
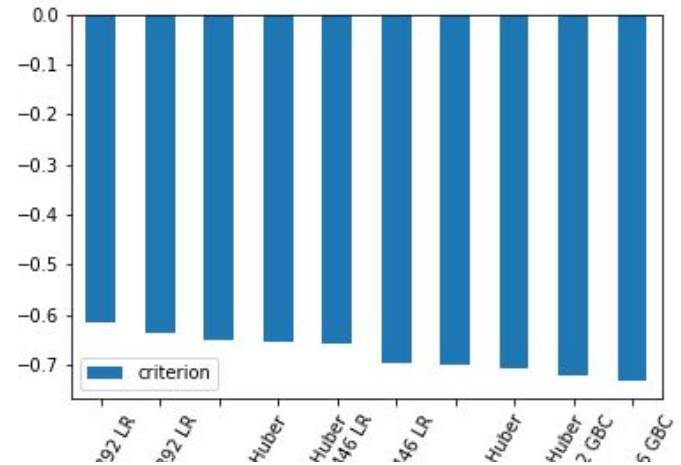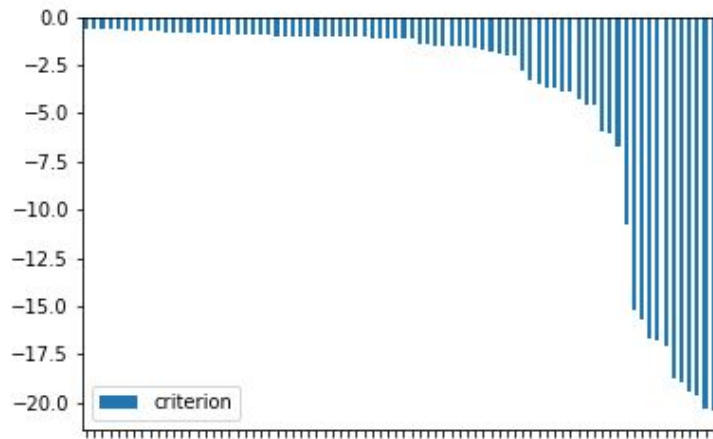
Results: On TS1
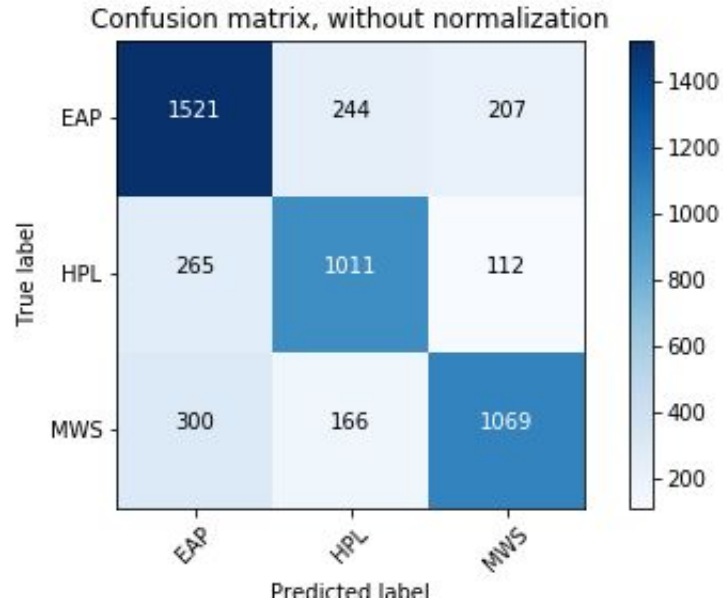
- Logloss = 0.64
  for Half (892) - PCA - Logistic Regression

Confusion matrix, without normalization

	EAP	HPL	MWS	Formula
Sensitivity	0.77	0.73	0.70	tp/(tp +fn)
Specificity	0.81	0.88	0.91	tn/(tn+fp)
Precision	0.73	0.71	0.77	tp/(tp+fp)
f-score	0.75	0.72	0.73	2*prec*sens/ (prec+sens)
Accuracy	0.79	0.84	0.84	(tp+tn)/total

Conclusion / Improvements

Gradient descent for coefficients of the selected pipeline

Adding features

Ensembling

Voting system

Deep Learning / Neural Networks

Questions ?