



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



Advanced Statistics for Physics analysis

Lorenzo Valentini

Naive Bayes Classifier for Fake News recognition

Introduction | Fake News

According to the New York Times, fake news is described as "a made-up story with an intention to deceive," intended to confuse or mislead people.

The concept of fake news is not new and has been present long before the internet became prevalent. Publishers have historically utilized false and misleading information to serve their own interests. However, with the rise of the internet, an increasing number of people have shifted from traditional media to online platforms, including social media, where fake news is abundant.

These fabricated stories are omnipresent in our daily lives, particularly on social media platforms and online applications. The proliferation of fake news presents a significant challenge for the news industry, making it crucial for individuals to be able to discern real news from false content.

The goal of the project is to implement a Multinomial Naive Bayes classifier in R and test its performances in the classification of social media posts.

Introduction | Bayes Classifier

Naive Bayes serves as a straightforward technique for constructing classifiers.

These classifiers are responsible for assigning class labels to instances of a problem, represented as vectors of feature values, where the class labels are selected from a finite set.

All the Naive Bayes Classifiers assume that the value of a specific feature is independent of any other feature's value, given the class variable: a Naive Bayes Classifier treats each feature as contributing independently to the probability of an instance belonging to a certain class, regardless of potential correlations between features.

Introduction | Project Outline

1. Introduction
2. Dataset
3. Algorithm
 1. Theory
 2. Preprocessing
 3. Feature Generation
 4. Classifier
 5. Predictions
4. Results
 1. Single Words
 2. Higher Order N-Grams
 3. Sentiment Analysis
 4. Technique Combination
 5. Binomial Classifier

Dataset | the Twitter dataset



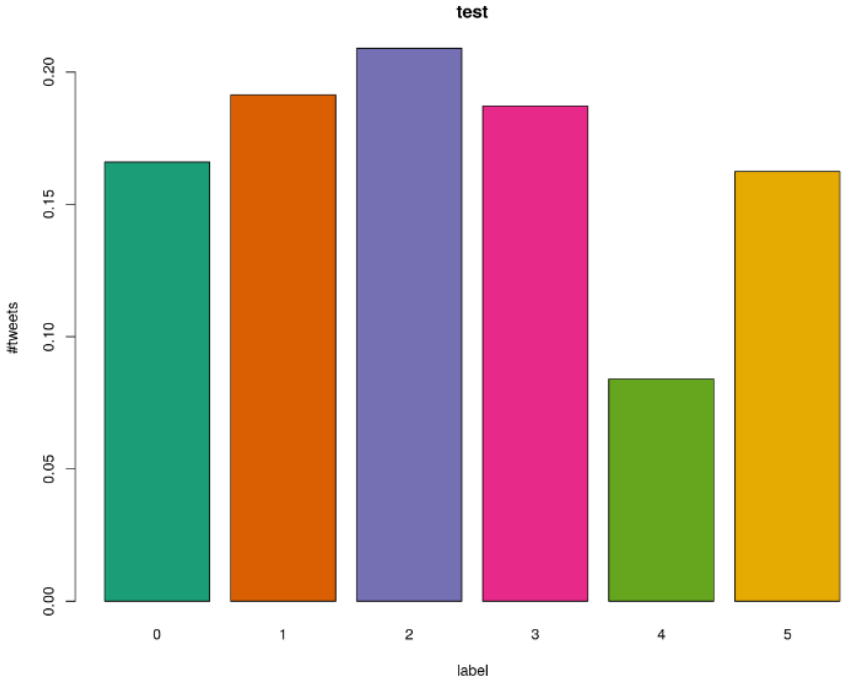
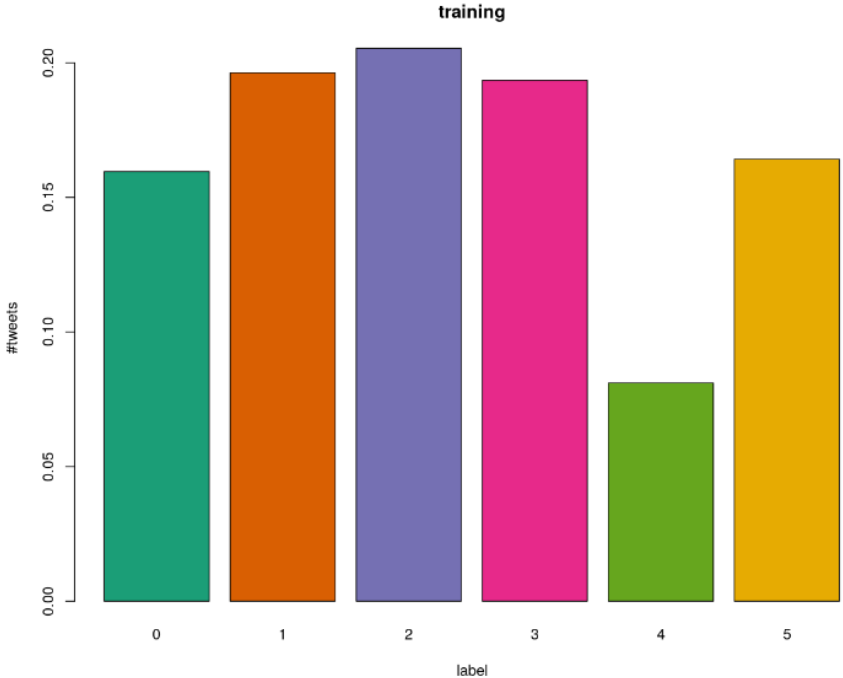
Kaggle Twitter dataset:
10k tweets to be divided in training and test dataset,
Topic, content, level of truthfulness.

Labels		Text	Text_Tag
<int>	<int>		
1	1	Says the Annies List political group supports third-trimester abortions on demand.	abortion
2	2	When did the decline of coal start? It started when natural gas took off that started to begin in (President George W.) Bushs administration.	energy,history,job-accomplishments
3	3	Hillary Clinton agrees with John McCain "by voting to give George Bush the benefit of the doubt on Iran."	foreign-policy
4	1	Health care reform legislation is likely to mandate free sex change surgeries.	health-care
5	2	The economic turnaround started at the end of my term.	economy,jobs
6	5	The Chicago Bears have had more starting quarterbacks in the last 10 years than the total number of tenured (UW) faculty fired during the last two decades.	education

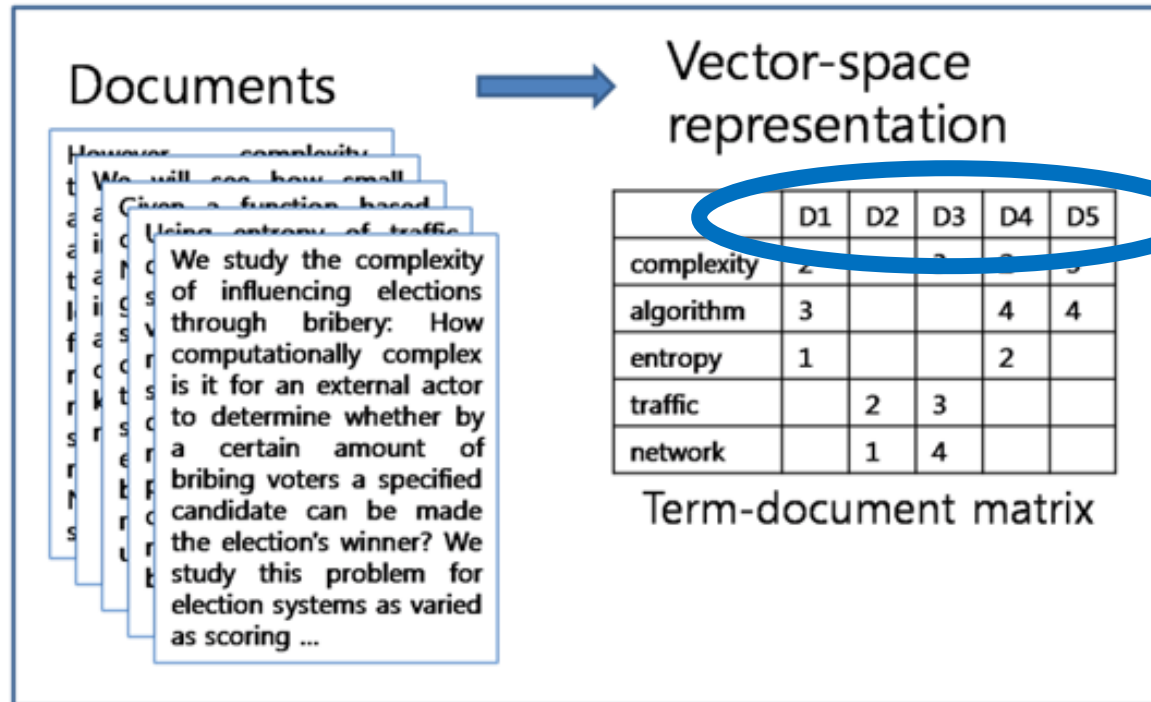
Dataset | Distribution



Labels	Values
True	5
Mostly True	3
Half True	2
Barely True	0
False	1
Not Known	4



Algorithm | Idea: DTM, Bayes



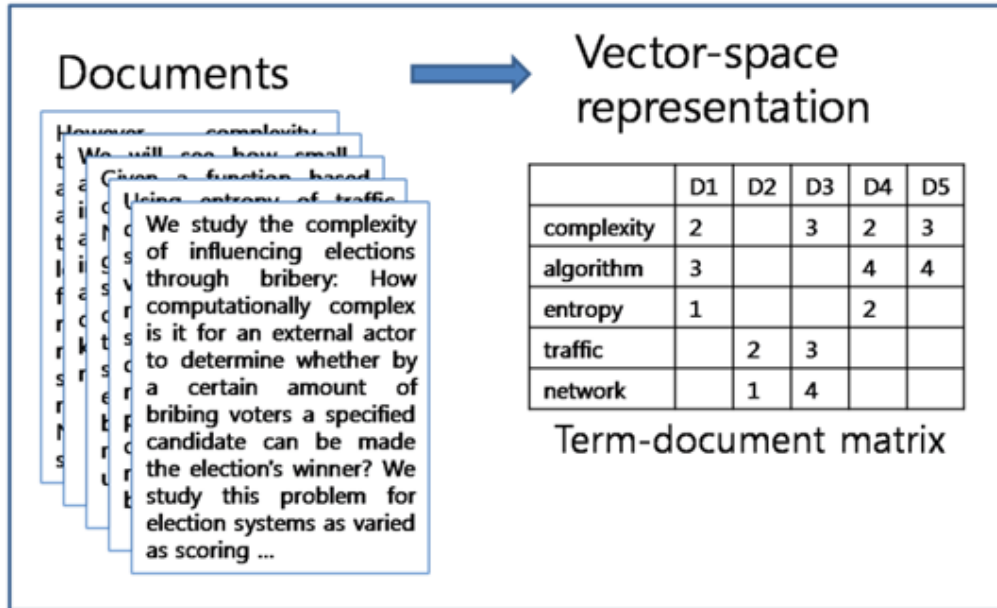
$$B = (b_1, b_2, \dots, b_n)$$

Creating a Document-Term-Matrix.

Applying the Bayes theorem in order to obtain the label of a document from the words contained in it

$$P(A|B, I) = \frac{P(B|A, I)P(A|I)}{P(B|I)} \quad \Rightarrow \quad P(A|b_1, b_2, \dots, b_n, I) = \frac{P(b_1|A, I)P(b_2|A, I) \dots P(b_n|A, I)P(A|I)}{P(b_1|I)P(b_2|I) \dots P(b_n|I)}$$

Algorithm | Idea: DTM, Bayes



$$P(A|b_1, b_2, \dots, b_n, I) = \frac{P(b_1|A, I)P(b_2|A, I) \dots P(b_n|A, I)P(A|I)}{P(b_1|I)P(b_2|I) \dots P(b_n|I)}$$

$$P(A|b_1, b_2, \dots, b_n, I) \propto P(A|I) \prod_{i=1}^n P(b_i|A)$$

$$A = \operatorname{argmax} P(A|I) \prod_{i=1}^n P(b_i|A)$$

Algorithm | Idea: Priors, Add One Smoothing

$$A = \operatorname{argmax} P(A|I) \prod_{i=1}^n P(b_i|A)$$

$$\left\{ \begin{array}{l} P(A|I) = \frac{N_A}{\sum_{\alpha \in V} N_{\alpha}} \\ P(b|A, I) = \frac{N_{bA}}{\sum_{\tau \in V} N_{\tau A}} \end{array} \right.$$

The prior for the label is the frequency of that label across all the documents.

The probability of a word to belong to a specific label class is the frequency of that label across all the words in all the documents

When making predictions, the probability of a specific label may collapse to 0 if a single word of the document wasn't encountered in that class during the training: need of a better formula.



$$P(b|A, I) = \frac{N_{bA} + 1}{\sum_{\tau \in V} (N_{\tau A} + 1)}$$

Algorithm | Preprocessing

Enforcing lowercase characters.

Word Stemming.

	Text <chr>	Labels <int>	Text_Tag <chr>	doc_id <chr>
1	... study after study has shown that the death penalty deters murders.	2	crime	studi after studi has shown that the death penalti deter murder
2	...Over 30 years, federal spending on education has grown by 375 percent, but test scores remain flat.	3	education,federal- budget	over 30 year feder spend on educ has grown by 375 percent but test score remain flat
3	...some of those (tax increases) were either court-ordered, or they were voted on by the people and approved by the people for (such) things as roads.	2	taxes	some of those tax increas were either court order or they were vote on by the peopl and approv by the peopl for such thing as road
4	"I have received more contributions than any other candidate in the race Republican or Democrat."	1	elections	i have receiv more contribut than ani other candid in the race republican or democrat
5	(After the auto bailout) General Motors is back on top as the worlds No. 1 automaker.	2	economy,jobs	after the auto bailout general motor is back on top as the world no 1 automak

Algorithm | Feature Generation

- Few steps necessary to obtain the Document Term Matrix:
 - Tokenization
 - Excluding words that are too common or too rare
 - Casting to dataframe

doc_id	Text	Labels	Text_Tag	1	3	cannot	career	colleg	drop	...	the highest corpor tax rate in the	almost as much wealth as the bottom 90 percent	as much wealth as the bottom 90 percent	much wealth as the bottom 90 percent	own almost as much wealth as the	are open serv in the u congress
<chr>	<chr>	<int>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	...	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	1 3 of our kid drop out of high school cannot go to colleg or start a career	0	1/3 of our kids drop out (of high school), cannot go to college or start a career.	corrections-and-updates,education	1	1	1	1	1	...	0	0	0	0	0	0

Algorithm | Classifier

- The steps to obtain the classifier are:
 - Grouping by the labels, counting the documents and summing over the word absolute frequencies,
 - Adding 1 to the words frequencies,
 - Normalizing the document count wrt the total number of documents,
 - Normalizing the words frequencies wrt the total number of words.


Labels	1	3	cannot	career	colleg	drop	who are open serv in the u	under the new health care law the	label_densities
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
0	0.0005659576	0.0001650710	1.414894e-04	1.886525e-04	0.0004008867	2.358157e-04	2.358157e-05	2.358157e-05	0.16442543
1	0.0005910683	0.0002626970	4.378284e-05	8.756567e-05	0.0002626970	8.756567e-05	2.189142e-05	2.189142e-05	0.19767726
2	0.0008535734	0.0002909909	7.759758e-05	1.551952e-04	0.0004849849	1.745946e-04	1.939939e-05	1.939939e-05	0.20770171
3	0.0009079008	0.0002533677	8.445589e-05	1.055699e-04	0.0006545332	1.900258e-04	2.111397e-05	2.111397e-05	0.18704156
4	0.0004170774	0.0001516645	1.137484e-04	3.791613e-05	0.0001895806	1.516645e-04	1.516645e-04	7.583226e-05	0.08325183
5	0.0007258123	0.0001935499	7.258123e-05	4.838749e-05	0.0004596811	1.935499e-04	2.419374e-05	2.419374e-05	0.15990220

Algorithm | Classifier

$$A = \operatorname{argmax} P(A|I) \prod_{i=1}^n P(b_i|A)$$

$$P(b|A, I) = \frac{N_{bA}}{\sum_{\tau \in V} N_{\tau A}}$$

$$P(A|I) = \frac{N_A}{\sum_{\alpha \in V} N_{\alpha}}$$



Labels	1	3	cannot	career	colleg	drop	...	who are open serv in the u	under the new health care law the	label_densities
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>		<dbl>	<dbl>	<dbl>
0	0.0005659576	0.0001650710	1.414894e-04	1.886525e-04	0.0004008867	2.358157e-04	...	2.358157e-05	2.358157e-05	0.16442543
1	0.0005910683	0.0002626970	4.378284e-05	8.756567e-05	0.0002626970	8.756567e-05		2.189142e-05	2.189142e-05	0.19767726
2	0.0008535734	0.0002909909	7.759758e-05	1.551952e-04	0.0004849849	1.745946e-04		1.939939e-05	1.939939e-05	0.20770171
3	0.0009079008	0.0002533677	8.445589e-05	1.055699e-04	0.0006545332	1.900258e-04		2.111397e-05	2.111397e-05	0.18704156
4	0.0004170774	0.0001516645	1.137484e-04	3.791613e-05	0.0001895806	1.516645e-04		1.516645e-04	7.583226e-05	0.08325183
5	0.0007258123	0.0001935499	7.258123e-05	4.838749e-05	0.0004596811	1.935499e-04		2.419374e-05	2.419374e-05	0.15990220

Results | Single Words

- Standard version of the Naive Bayes
- The too common words are handpicked
- Scores
 - Training score: 48.7%
 - Test score: 24.5 %
- No particular insights from the confusion matrix

Results | Higher Order N-Grams

- Attempt of finding a workaround to the assumption of independent features.
- No necessity of excluding words that are too common: assumption that they will be often close to a less common word.
- Scores ($N = 2$)
 - Training score: 63.9 %
 - Test score: 25.5 %
- No particular insights from the confusion matrix

Results | Sentiment Analysis

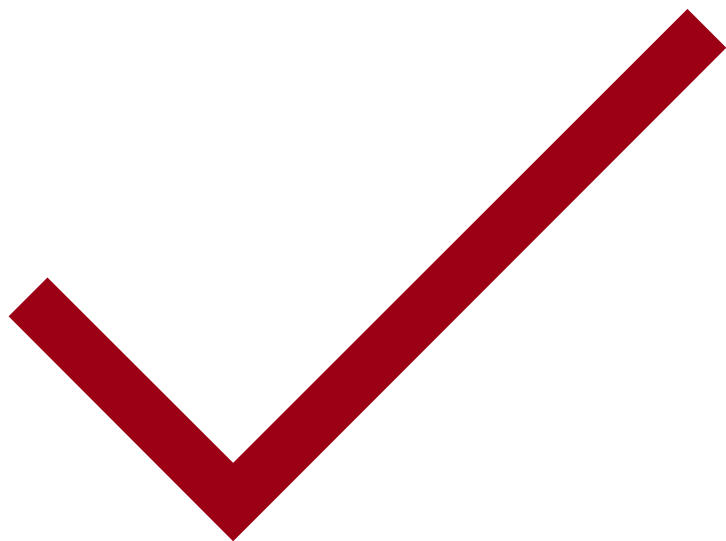
- Trying to bypass the problem of the small dataset vs high number of features: extracting more general information like the sentiment and the use of the punctuation.
- Scores
 - Training score: 20.5 %
 - Test score: 19.0 %

Results | Combined Techniques

- Using the document to term matrix to which are added the sentiment features.
- Scores
 - Training score: 52.5 %
 - Test score: 24.3 %

Results | Binomial Classifier

- Testing the classifier on binary classification, grouping the tweets in only two categories (False, True).
- Scores
 - Training score: 70.1 %
 - Test score: 62.1 %



**THANK YOU FOR
THE ATTENTION**