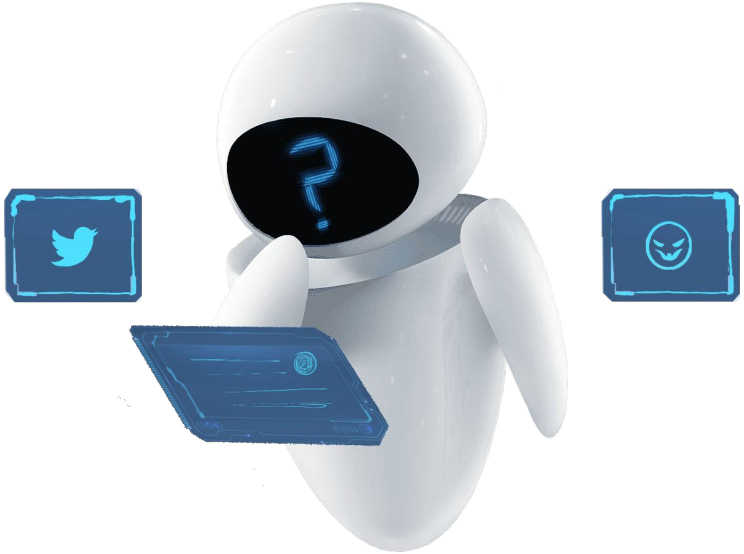
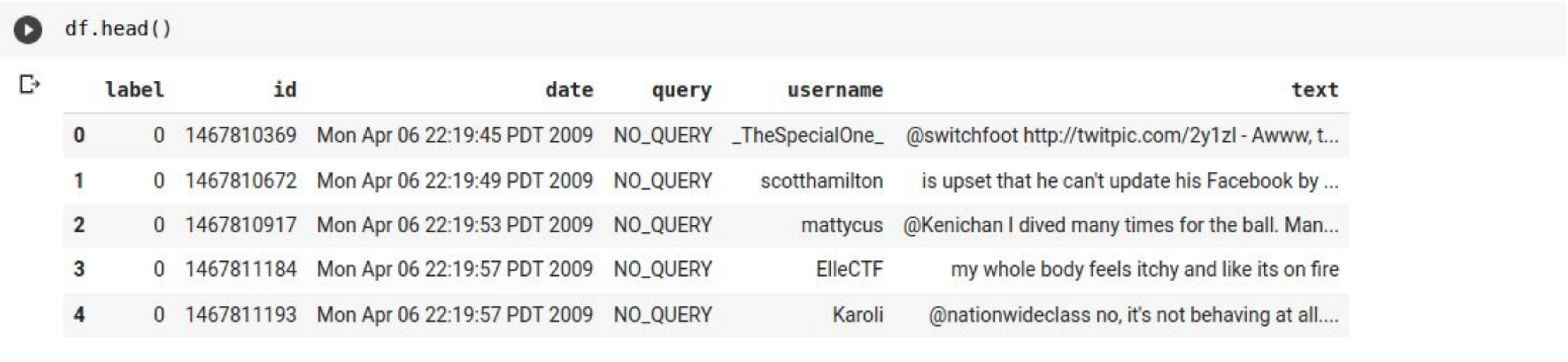
Plaint or Tweet



Tommaso Battistini
Edoardo De Matteis
Leonardo Emili
Mirko Giacchini
Alessio Luciani

The Problem

- Sentiment analysis on tweets.
- A good dataset with annotated tweets: Sentiment140 (about 1.6M of tweets).
- Motivation: interesting in research, and used in industry.

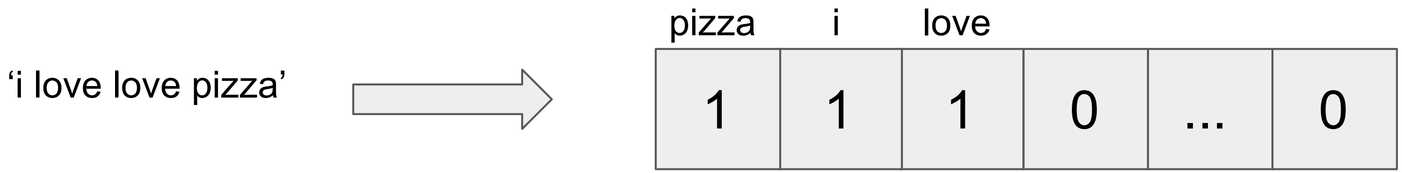


```
df.head()
```

	label	id	date	query	username	text
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
2	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
3	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
4	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....

The Models - Naïve Bayes

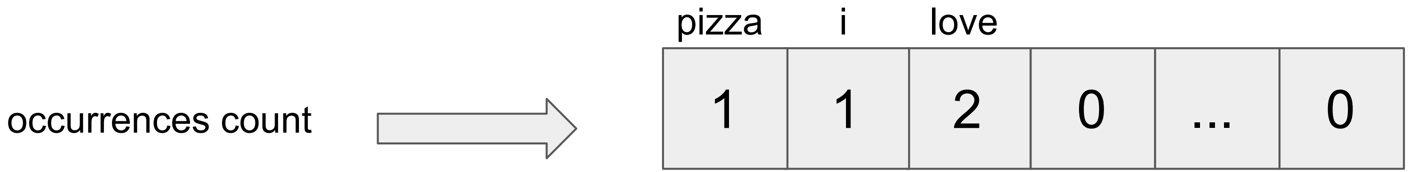
Bernoulli Event Model:



‘i love love pizza’ →

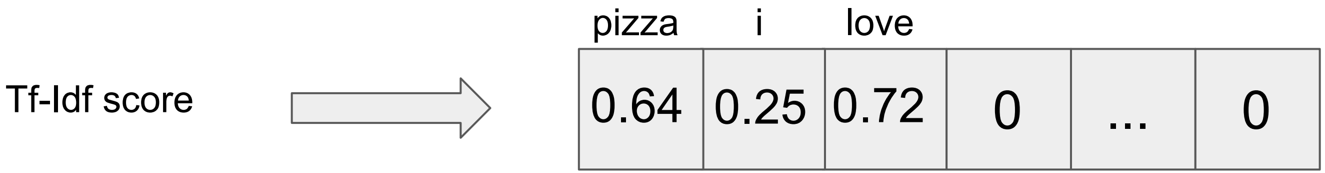
pizza	i	love			
1	1	1	0	...	0

Multinomial Event Model:



occurrences count →

pizza	i	love			
1	1	2	0	...	0



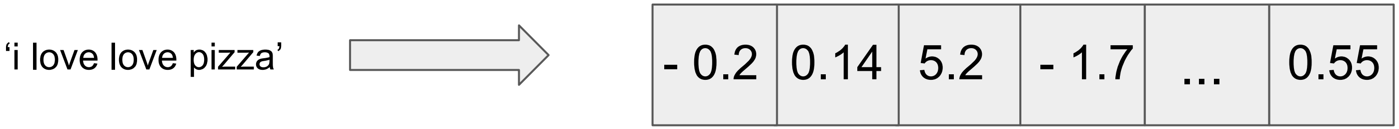
Tf-Idf score →

pizza	i	love			
0.64	0.25	0.72	0	...	0

The Models - Naïve Bayes on embeddings

Word embeddings: using Word2Vec/FastText

Tweet embeddings: average of words embeddings



‘i love love pizza’

- 0.2	0.14	5.2	- 1.7	...	0.55

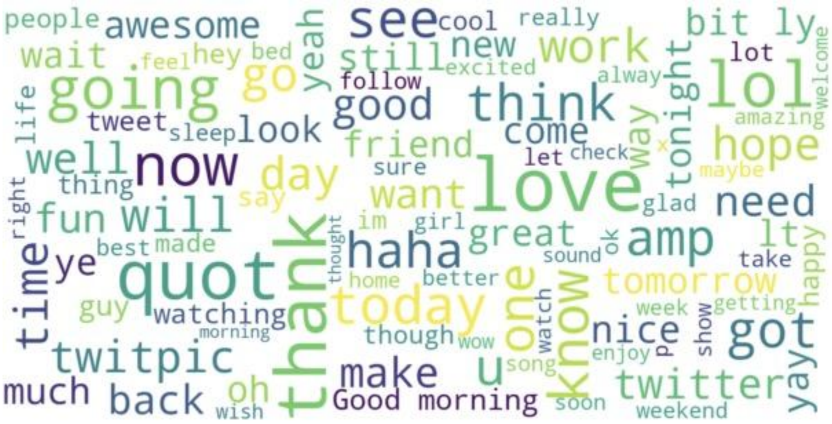
We tried:

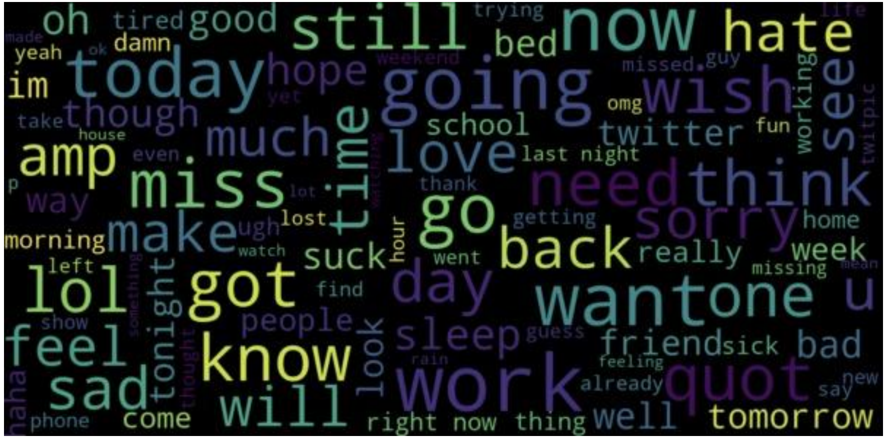
- Multinomial NB (discretizing and not)
- Gaussian NB

Preprocessing… or not?

- Common user typo correction.

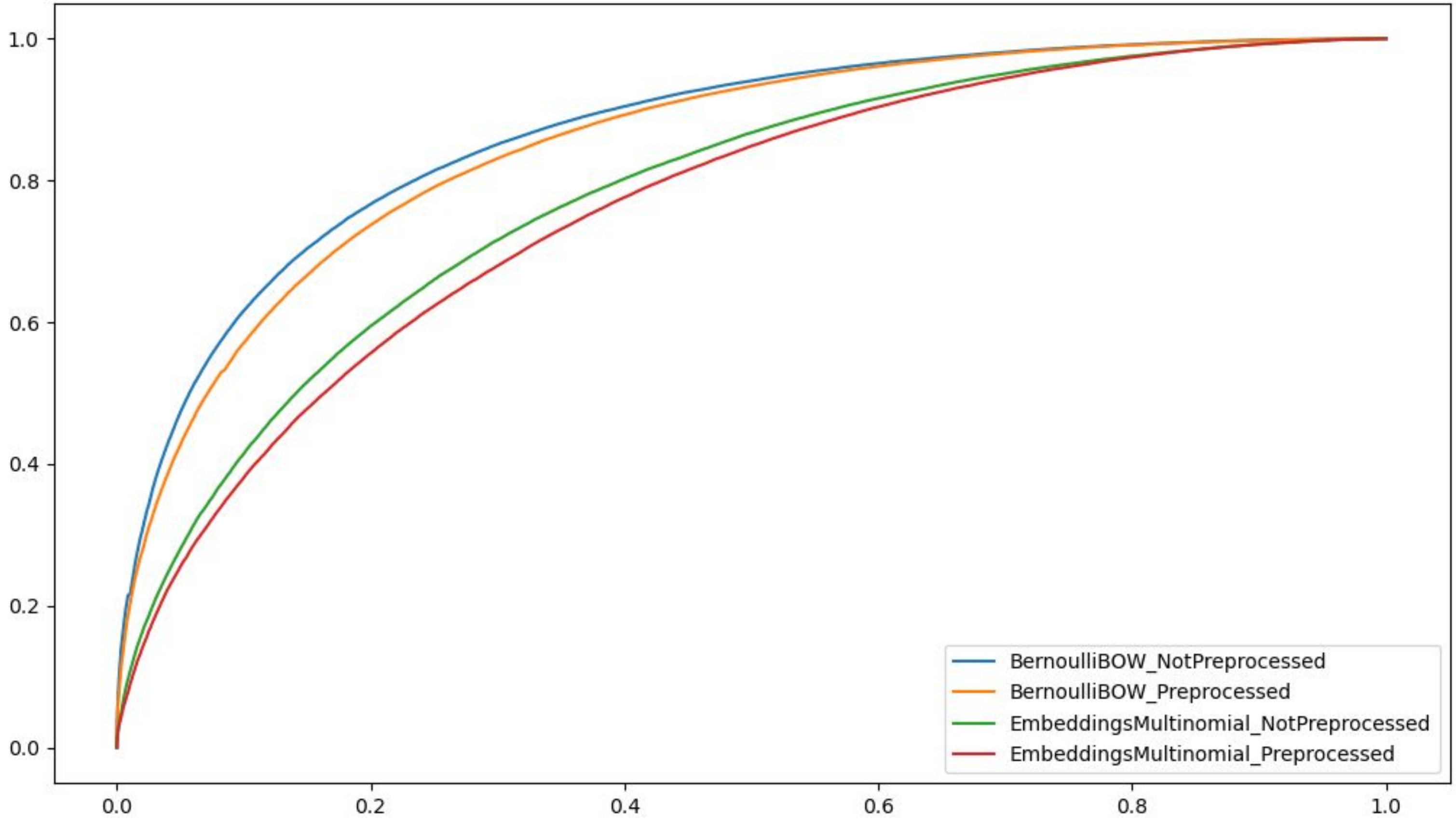
- We used lemmatization, stopwords removal (from spaCy).





Preprocessing… or not?

ROC curve with
80% training
20% testing



Preprocessing… or not?

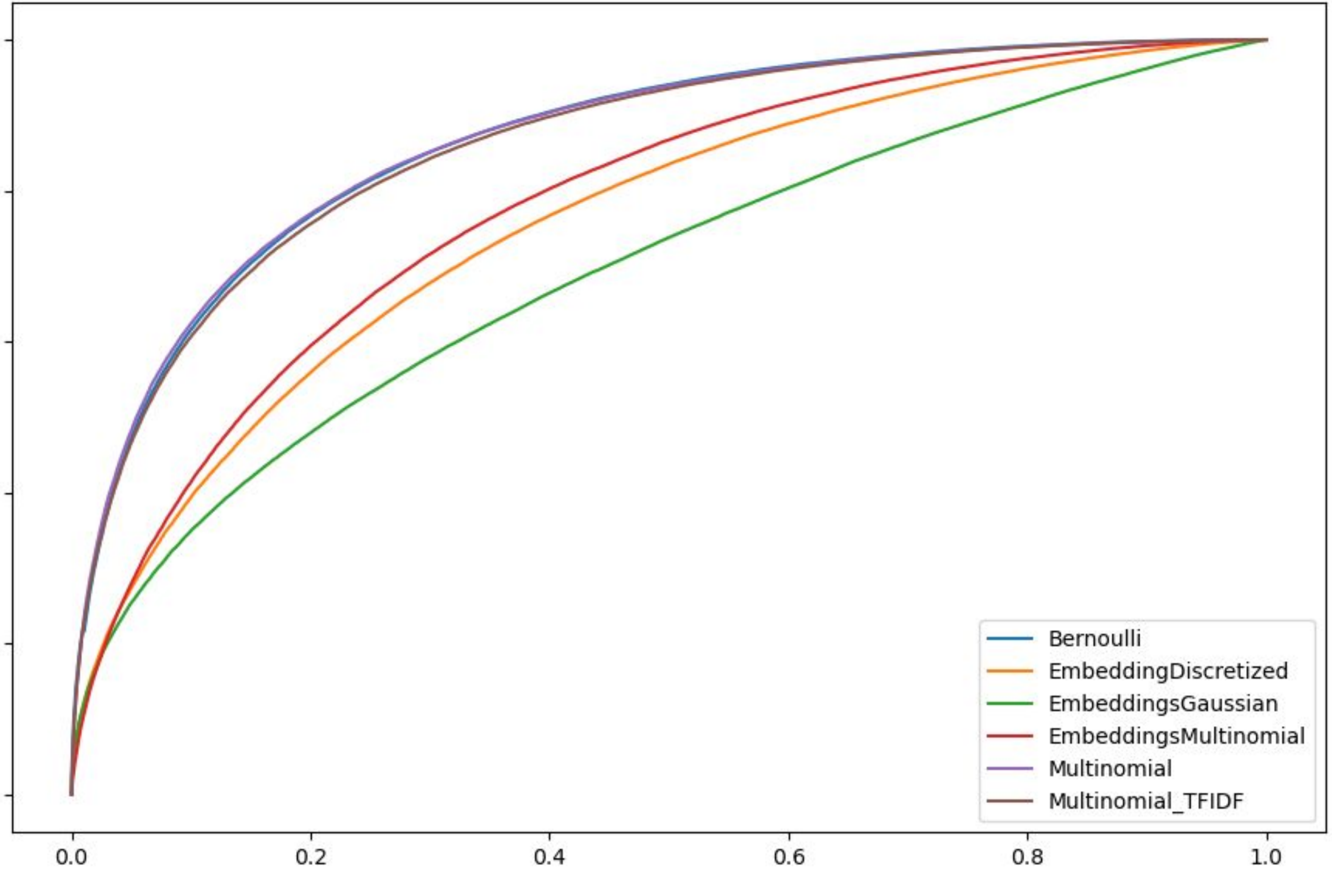
- Predefined stopwords lists include useful words

Our Bernoulli model:

preprocessing	accuracy
complete	0.7690
none	0.7828
ad-hoc stopwords removal	0.7834

Classic NB vs Embeddings

80% training
20% testing
without preprocessing



Classic NB vs Embeddings

- Classical approaches work better
- Tuning hyperparameters (i.e. ngram), 60% training, 20% validation, 20% test:

model	accuracy	f1-score	auroc
Bernoulli	0.797	0.799	0.880
Multinomial	0.802	0.802	0.884
Multinomial TF-IDF	0.805	0.804	0.886

Comparison with Kaggle notebooks

Twitter, Modelling with Naive Bayes + Streamlit

uses Multinomial Naive Bayes with Tf-Idf

We replicated their dataset split:

model	AUROC
From Notebook	0.839
Our Tf-Idf	0.841
From Notebook, removing some preprocessing	0.849

Comparison with Kaggle notebooks

Sentiment Analyzer

Uses LSTM network on word embeddings

They run only one test, we averaged over different seeds:

model	Accuracy
From Notebook	0.779
Our Tf-Idf	0.799
From Notebook, leaving stopwords	0.816

A curious test...

- We trained our model on Twitter’s dataset and tested it against the IMDB one.
- The distributions of IMDB and Twitter are very different.

Test	Accuracy
Train: Twitter Test: IMDB	0.732
Train: IMDB Test: IMDB	0.891
Train: IMDB Test: Twitter	0.549
Train: Twitter Test: Twitter	0.797

Conclusions

- Preprocessing might decrease performance, a careful ad-hoc preprocessing can boost performances.
- Naive Bayes works better with simpler representations.
- Multinomial Naive Bayes with Tf-Idf gives competitive results.

Thank you.