Depression detection in social networks

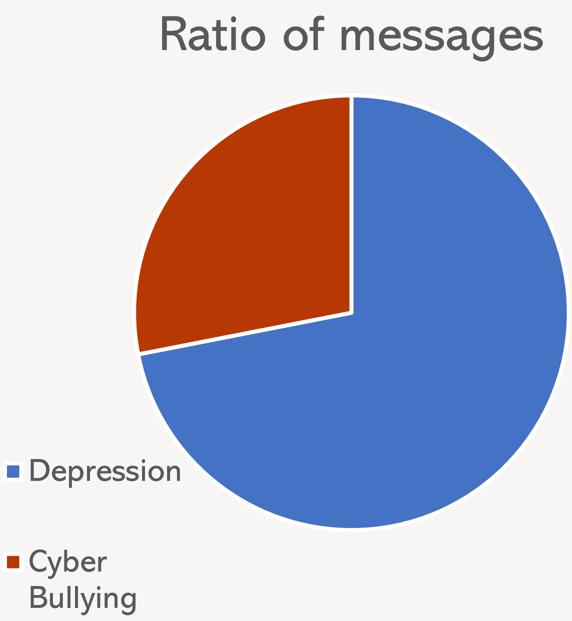
Martí Caixal i Joaniquet

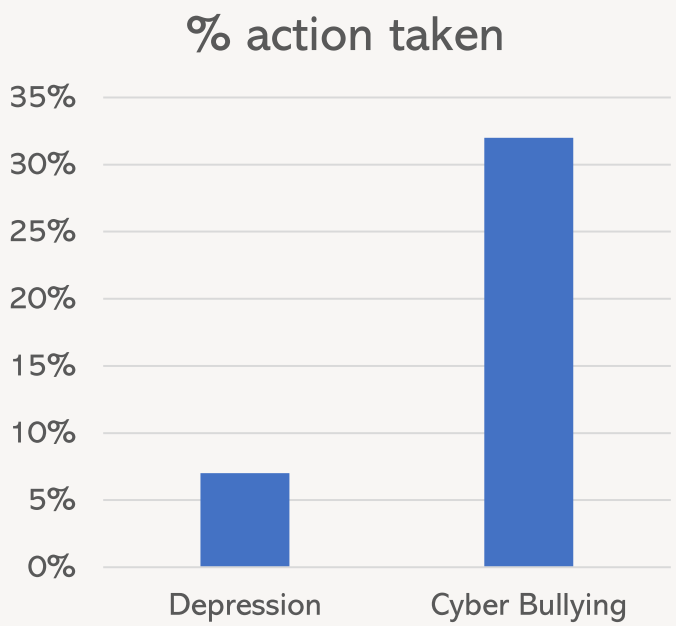
Agenda

- Introduction
- Objectives
- Methodology & Development
- Results
- Conclusions

What problem are we facing?

- More depression messages than other types
- Yet, they receive less attention





Find out which method gives the best results and how they differ

Shallow Learning:

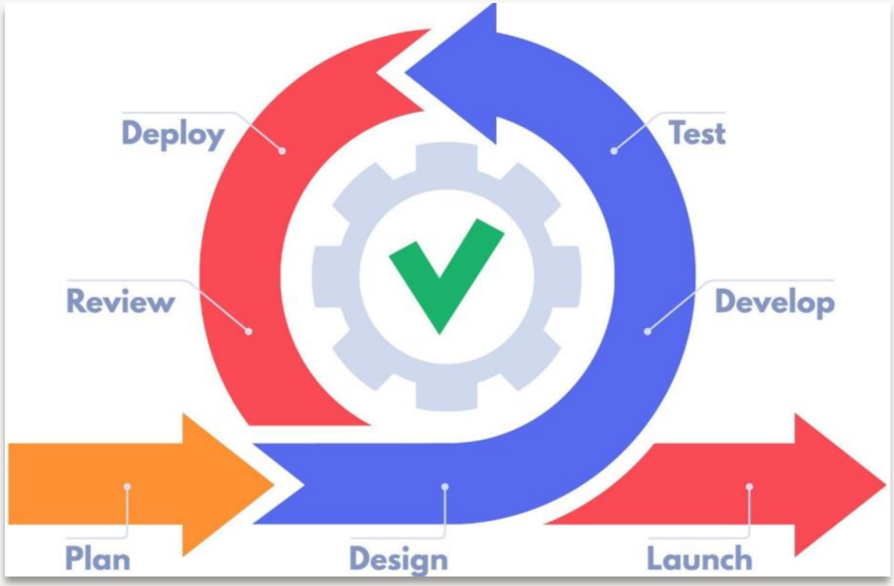
- Naïve Bayes
- Decision Tree
- Random Forest
- SVM
- KNN
- Hyper Parameter Search

Deep Leaning

- RNN
  - RNN LSTM
  - RNN GRU
- Transformers (BERT)

Planification

- Short iterations
- Good control of flow
- Friendly to changes
- Independent subobjectives
- Easy to detect errors



Agile Methodology

Data used

Mental Health Twitter (Twitter 3)

- Only the message and label

Depression Twitter (Twitter Scale)

- Labeled in a scale from 1 to 5

Depression Reddit (Reddit)

- Already cleaned

- 10000 messages
- 2 classes
- Unbalanced (80/20)

“ @cosmicgirlie Thinking of you. Everything crossed Turn baby turn! “

Data used

Mental Health Twitter (Twitter 3)

- Only the message and label

Depression Twitter (Twitter Scale)

- Labeled in a scale from 1 to 5

Depression Reddit (Reddit)

- Already cleaned

- 45000 messages
- 4 classes (Scale from 0 to 3
- Unbalanced (40/20/30/10)

“ humm dodgers scored a hr stupid dodgers i hate them“

Data used

Mental Health Twitter (Twitter 3)

- Only the message and label

Depression Twitter (Twitter Scale)

- Labeled in a scale from 1 to 5

Depression Reddit (Reddit)

- Already cleaned

- 40000 messages
- 2 classes
- Unbalanced (60/40)
- Already cleaned

“ i used to be highly functional before but it now i can barely function at all i take everything just… “

Data used

Unbalanced,
target class being minority:

× Undersampling

× Oversampling

- Recall instead of accuracy
- Macro average

Initial preprocessing

- Delete usernames
- Delete Stop Words
- Delete numbers
- Lemmanization
- Delete punctuation

NLTK

Specific approaches

Shallow Learning

- Bag of Words
- TF-IDF

Deep Learning

- Word Embedding (GloVe)

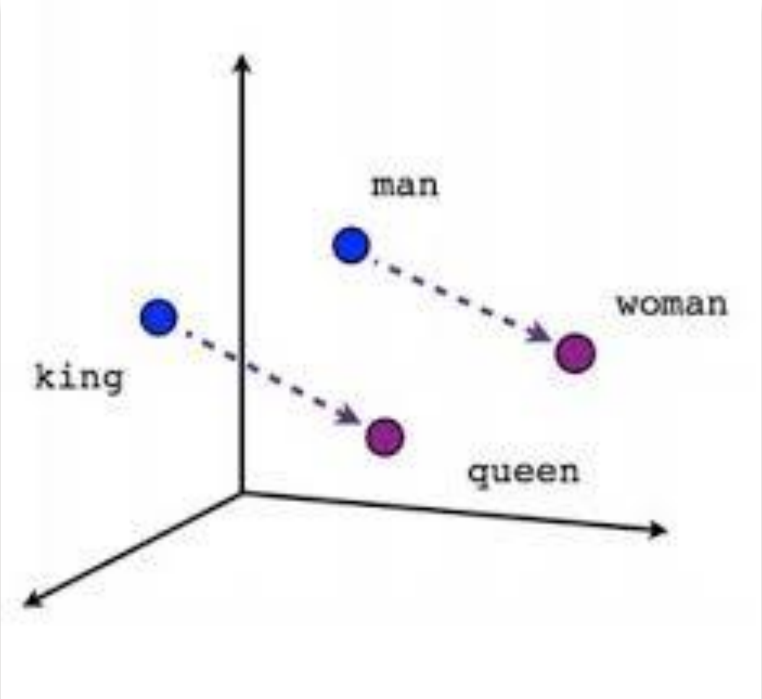
Bag of Words

Document	the	cat	sat	in	hat	with
the cat sat	1	1	1	0	0	0
the cat sat in the hat	2	1	1	1	1	0
the cat with the hat	2	1	0	0	1	1

TF-IDF

Word	TF		IDF	TF*IDF	
	A	B		A	B
The	1/7	1/7	log(2/2) = 0	0	0
Car	1/7	0	log(2/1) = 0.3	0.043	0
Truck	0	1/7	log(2/1) = 0.3	0	0.043
Is	1/7	1/7	log(2/2) = 0	0	0
Driven	1/7	1/7	log(2/2) = 0	0	0
On	1/7	1/7	log(2/2) = 0	0	0
The	1/7	1/7	log(2/2) = 0	0	0
Road	1/7	0	log(2/1) = 0.3	0.043	0
Highway	0	1/7	log(2/1) = 0.3	0	0.043

Bag of Words

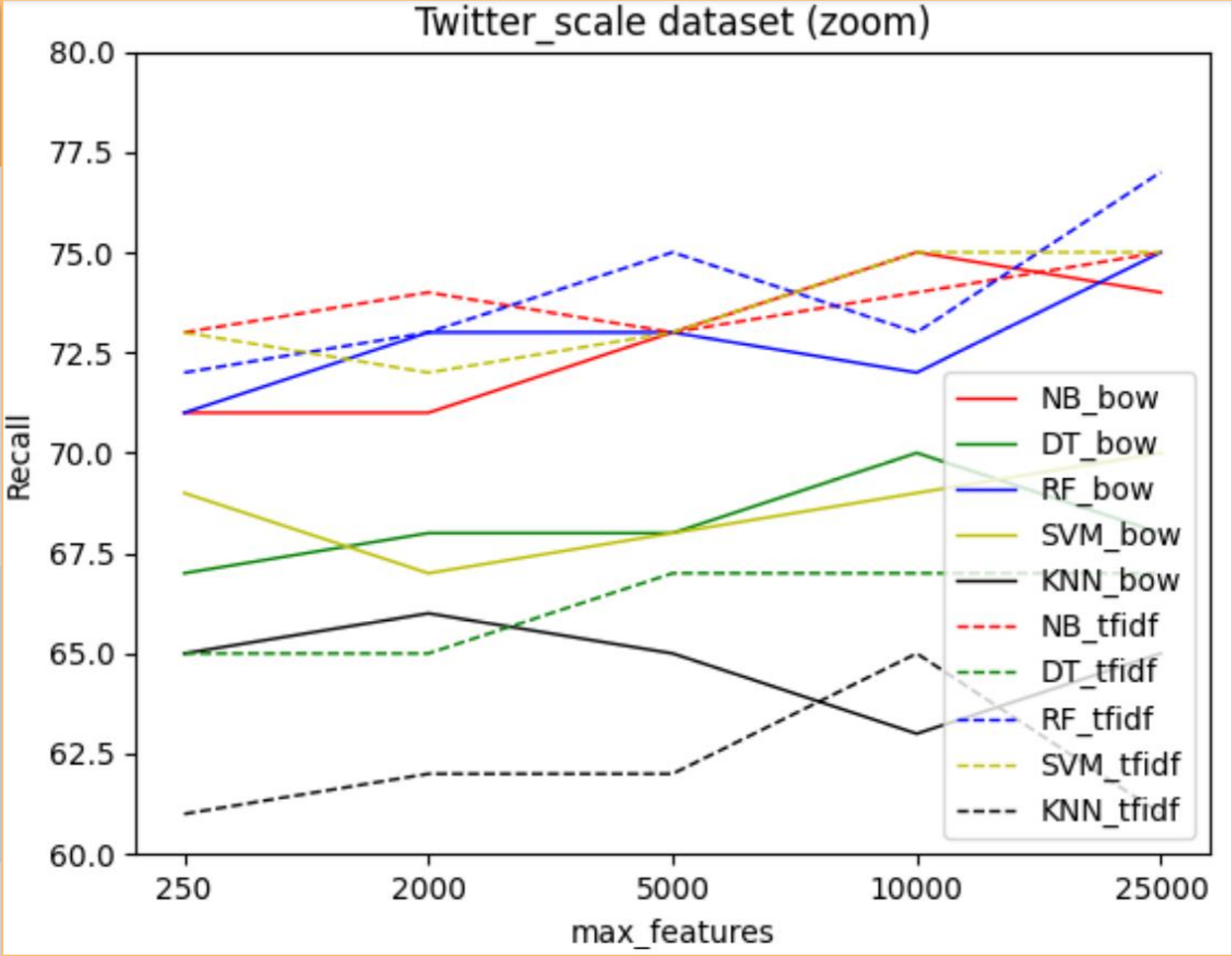


Shallow learning results

- Scikit Learn library
- Default parameters

- Naïve Bayes
- Decision Tree
- Random Forest
- SVM
- KNN
- Hyper Parameter Search

TF-IDF vs BoW & feature size

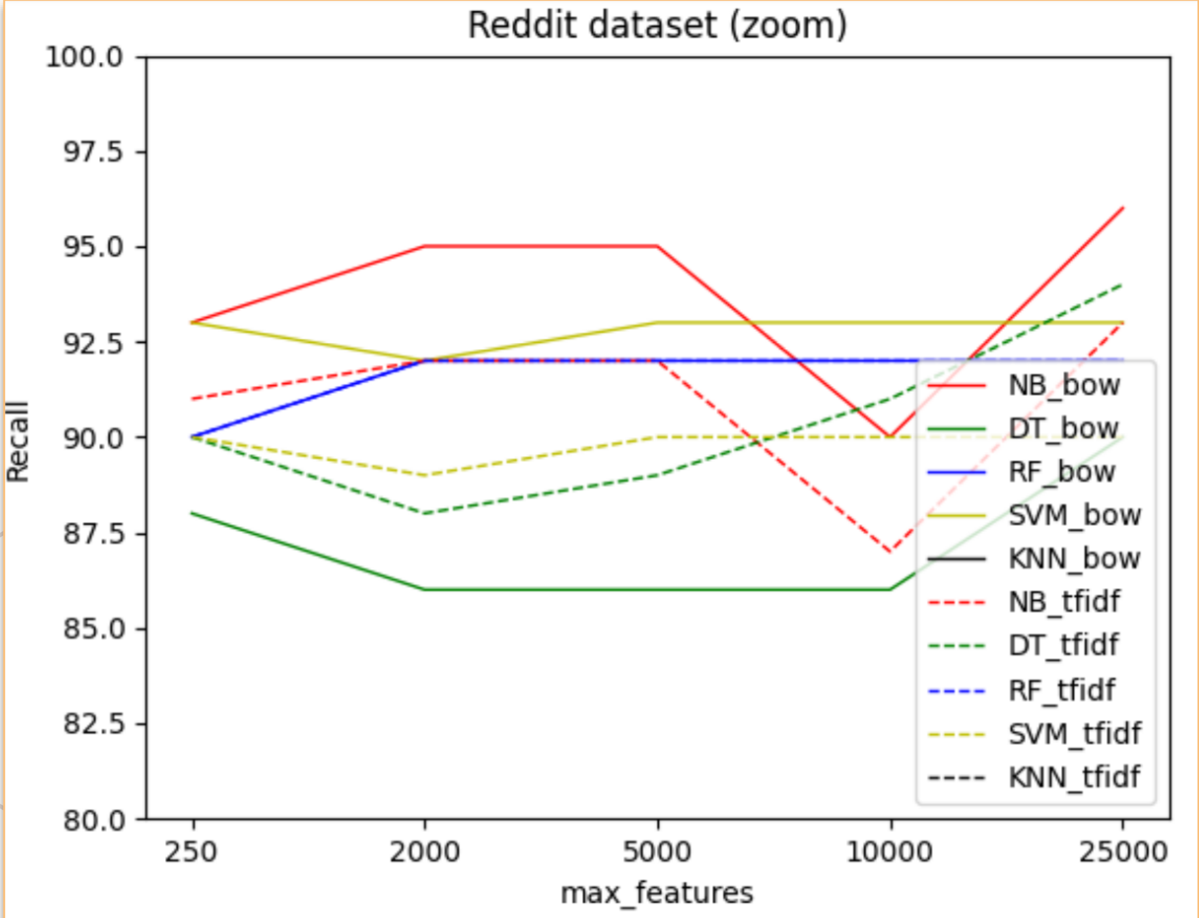


TF-IDF - - - -

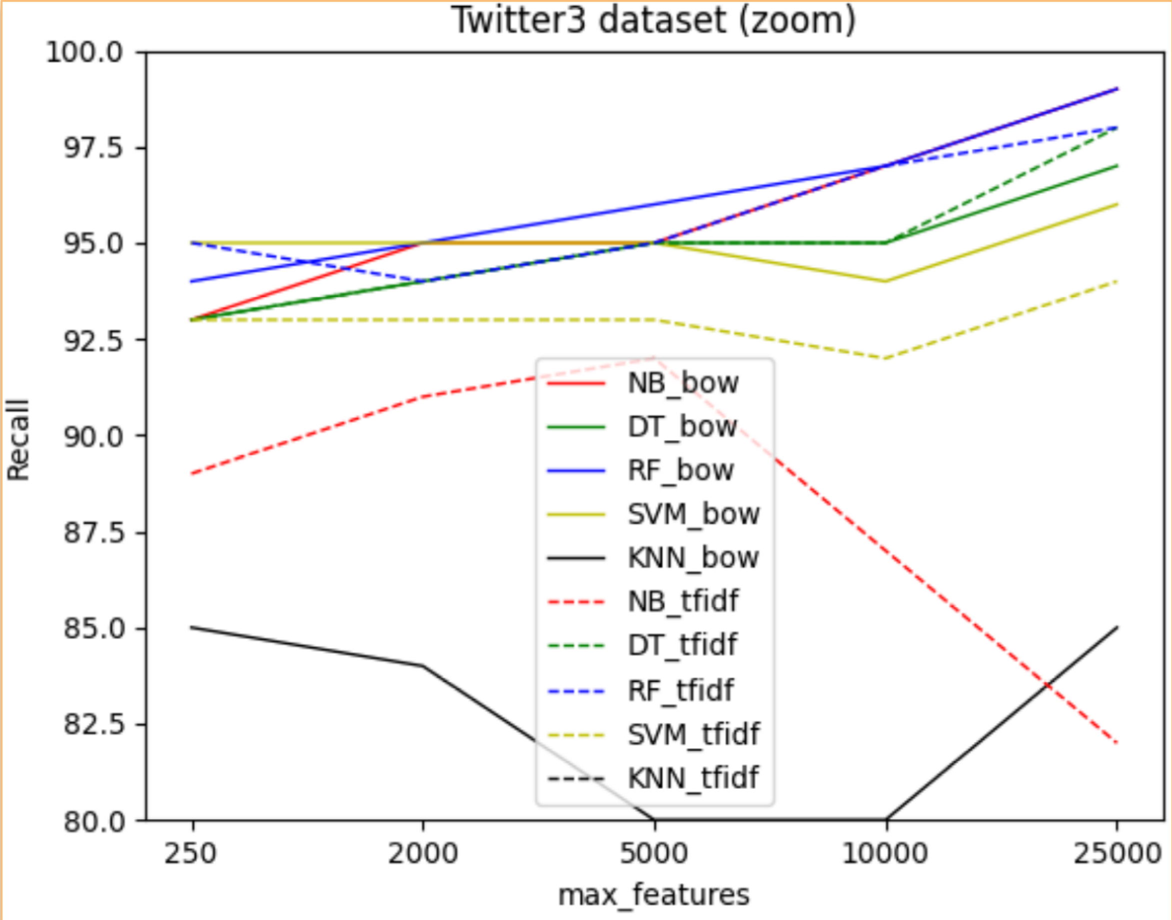
BoW ———

- ✓ TF-IDF slightly better
- ✓ nº max_features improves results

Results



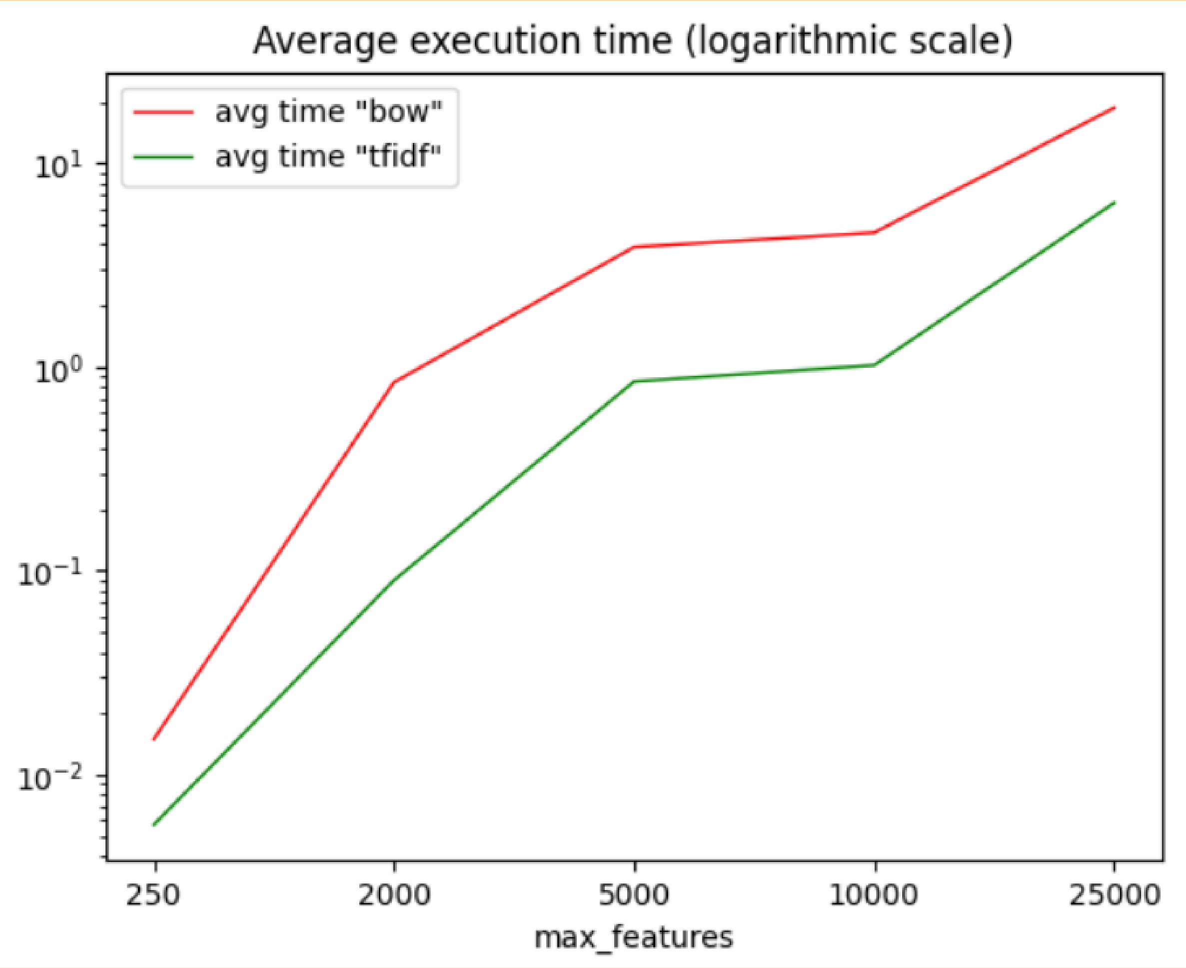
Reddit dataset (zoom)
Recall
100.0
97.5
95.0
92.5
90.0
87.5
85.0
82.5
80.0
250
2000
5000
10000
25000
max_features
NB_bow
DT_bow
RF_bow
SVM_bow
KNN_bow
NB_tfidf
DT_tfidf
RF_tfidf
SVM_tfidf
KNN_tfidf



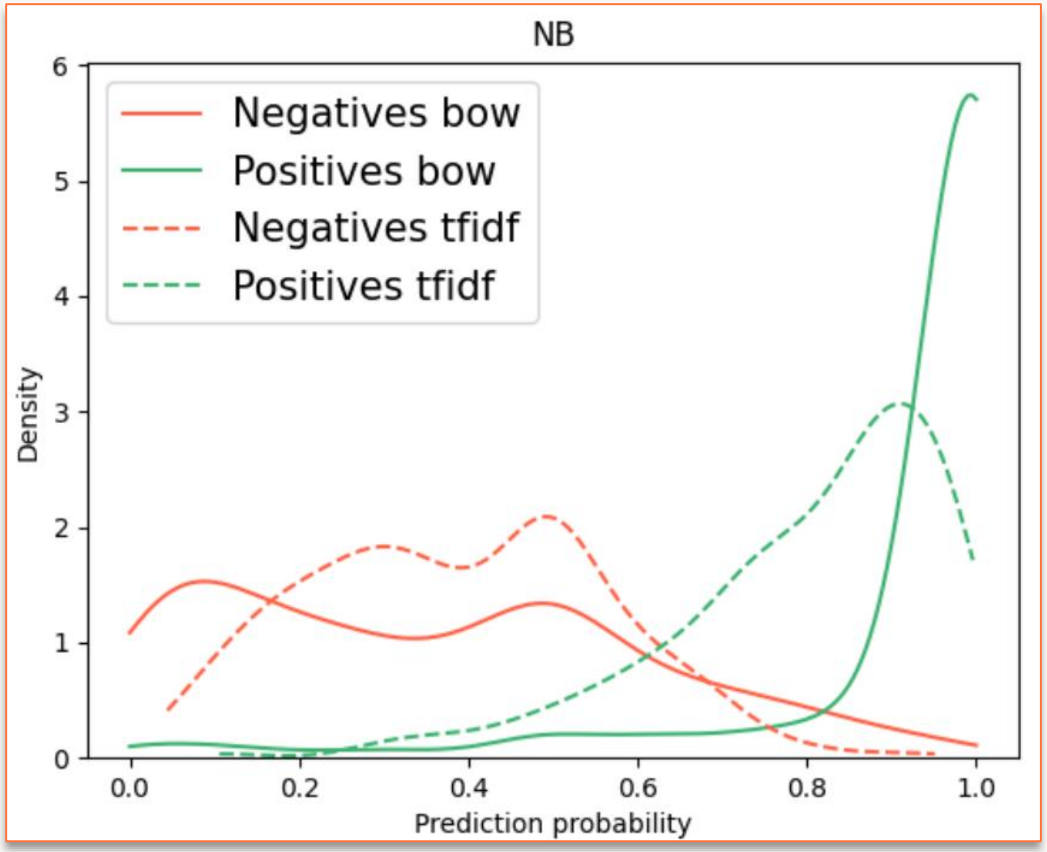
Twitter3 dataset (zoom)
Recall
100.0
97.5
95.0
92.5
90.0
87.5
85.0
82.5
80.0
250
2000
5000
10000
25000
max_features
NB_bow
DT_bow
RF_bow
SVM_bow
KNN_bow
NB_tfidf
DT_tfidf
RF_tfidf
SVM_tfidf
KNN_tfidf

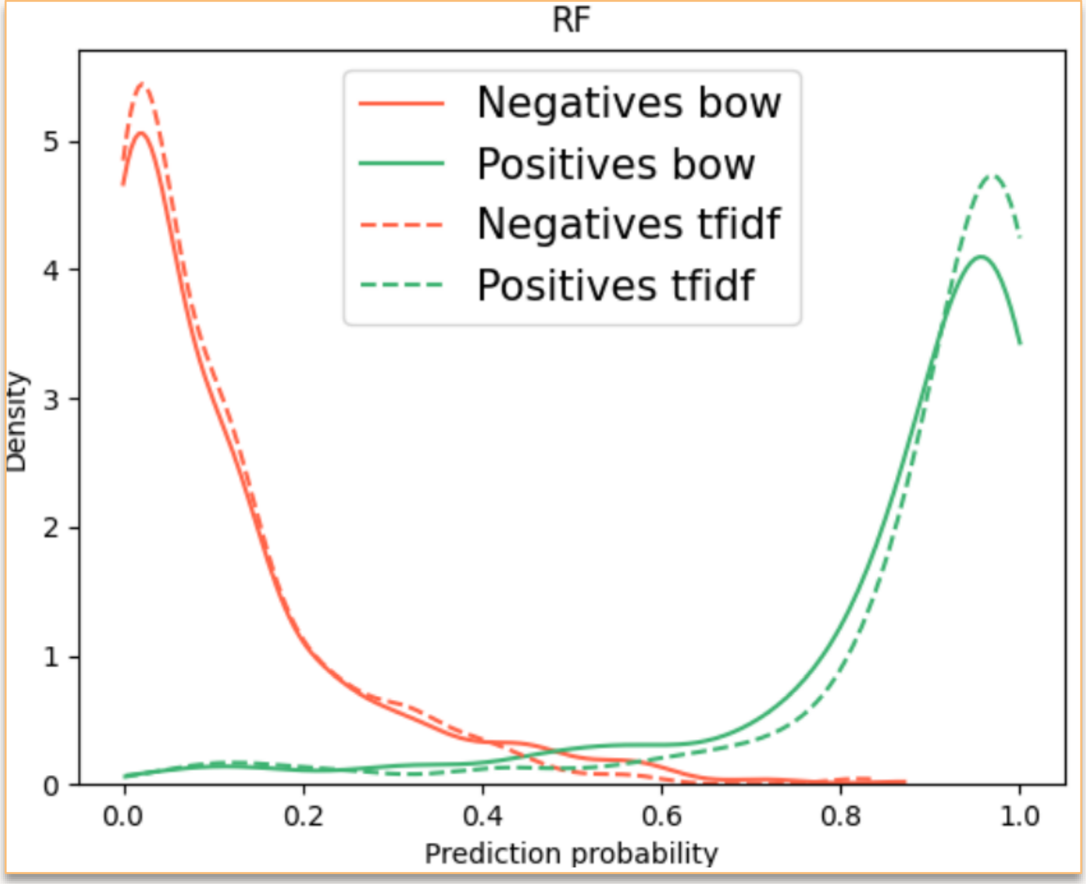
Execution time

- ✓ TF-IDF slightly better
- ✓ nº max_features improves results
- ✓ Better execution time



Confidence in predictions





Hyperparameter serach

asdfadf

- N]
- S
- S
- s

{”knn_n_neighbours”: 3, ”knn_p”: 3, ”knn_weight”: üniform”}
{”knn_n_neighbours”: 3, ”knn_p”: 3, ”knn_weight”: üniform”}
{”knn_n_neighbours”: 4, ”knn_p”: 3, ”knn_weight”: ”distance”}
{”knn_n_neighbours”: 6, ”knn_p”: 3, ”knn_weight”: ”distance”}
{”knn_n_neighbours”: 8, ”knn_p”: 3, ”knn_weight”: ”distance”}
{”knn_n_neighbours”: 5, ”knn_p”: 3, ”knn_weight”: ”distance”}
{”knn_n_neighbours”: 5, ”knn_p”: 3, ”knn_weight”: üniform”}

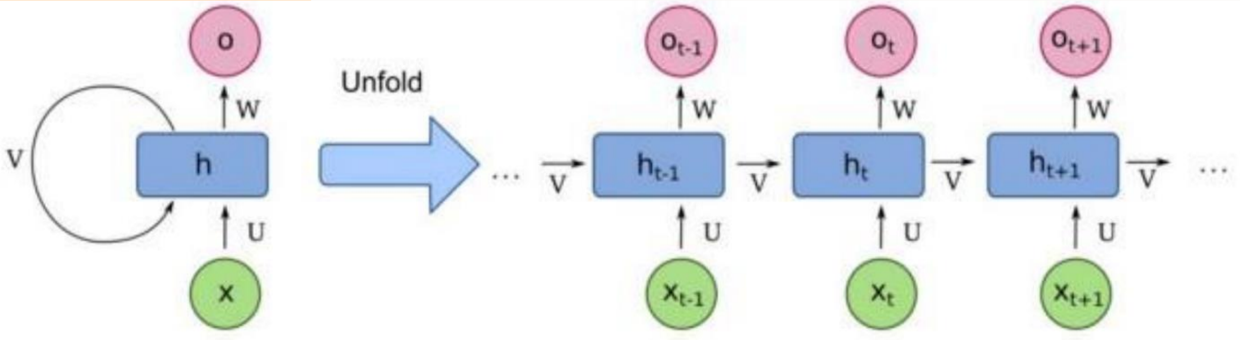
TAULA 3: MILLORS HYPERPARAMETERS KNN

Deep learning results

- Keras library
- RTX 3070 Ti

- RNN
- RNN GRU
- RNN LSTM
- BERT

RNN
- Sequence of layers
- Input, activation function, output

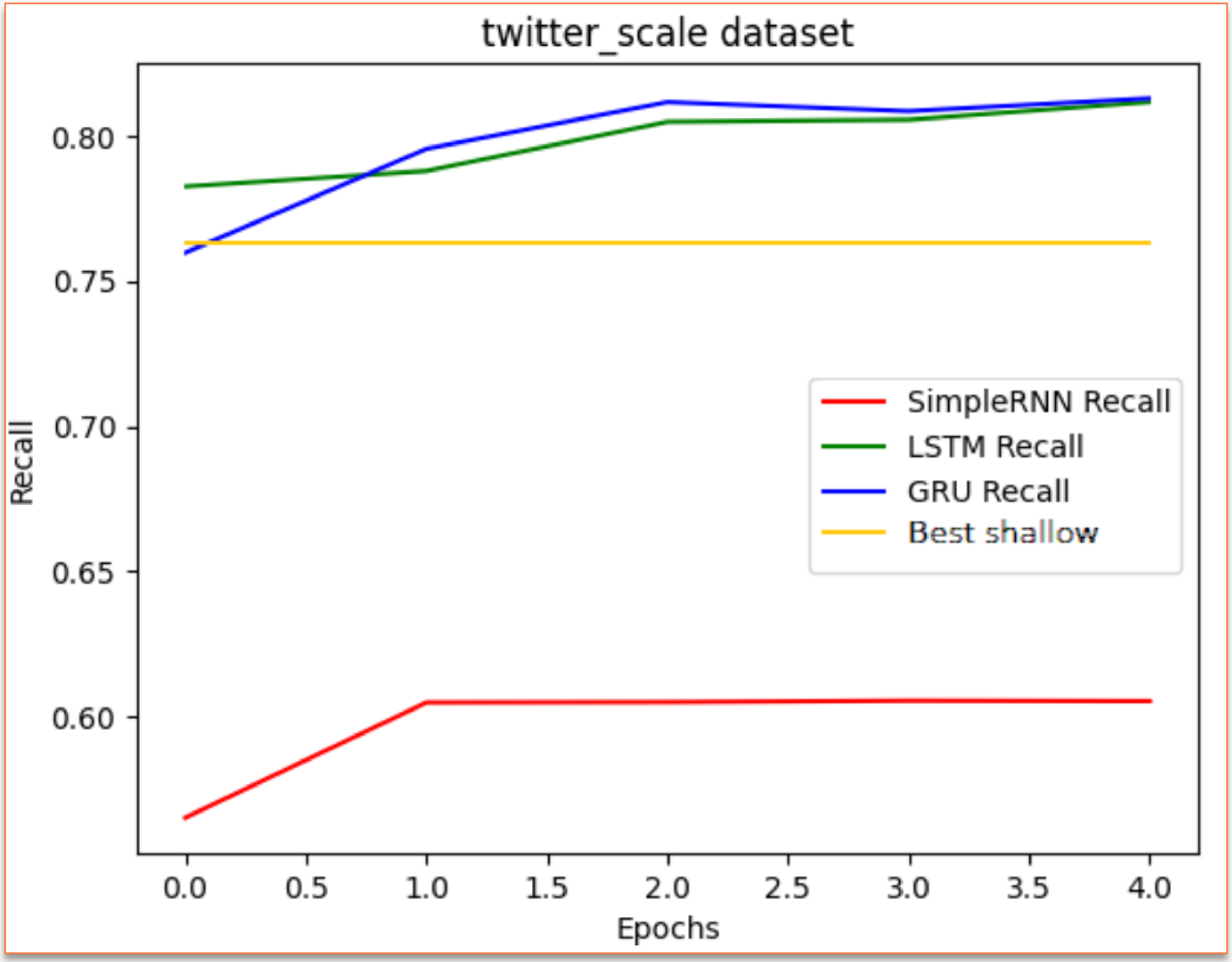


RNN LSTM
- 3 gates
- Memory

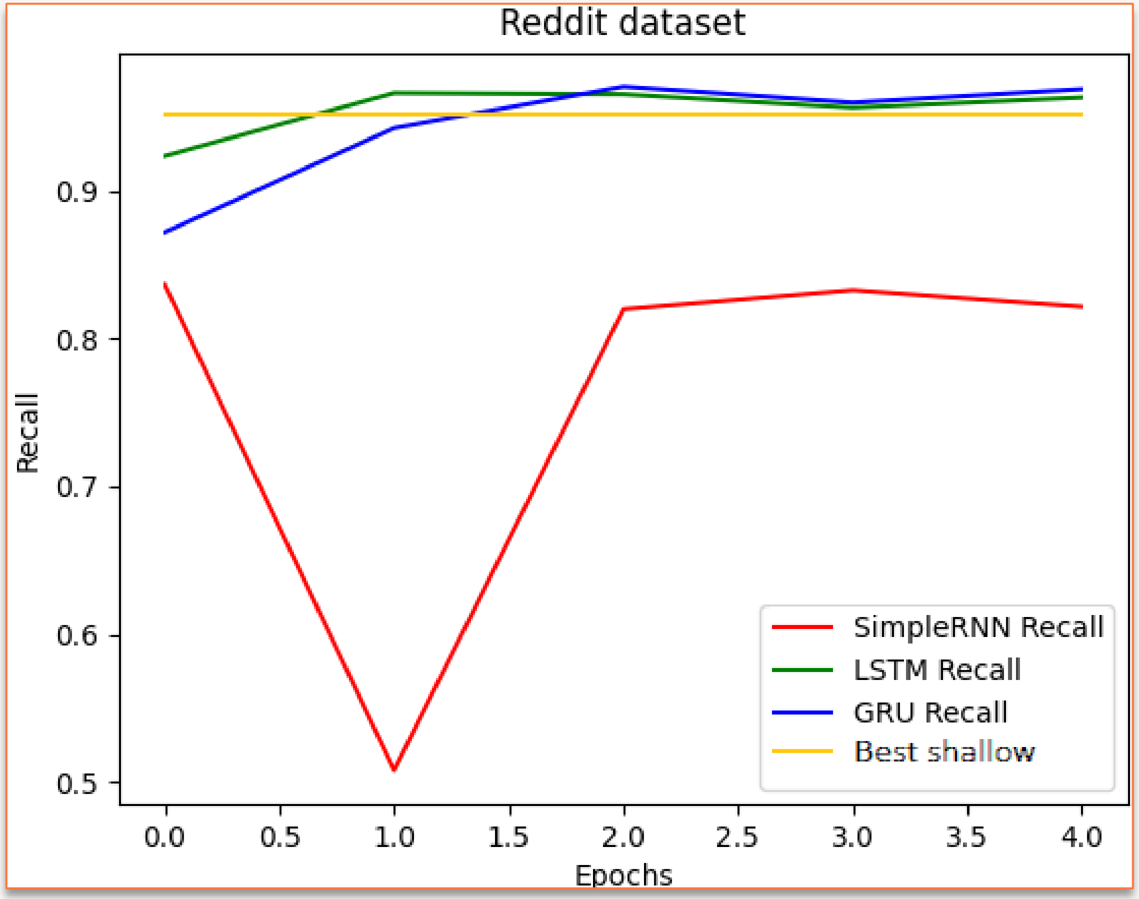
RNN GRU
- 2 gates
- No memory

Simplified LSTM

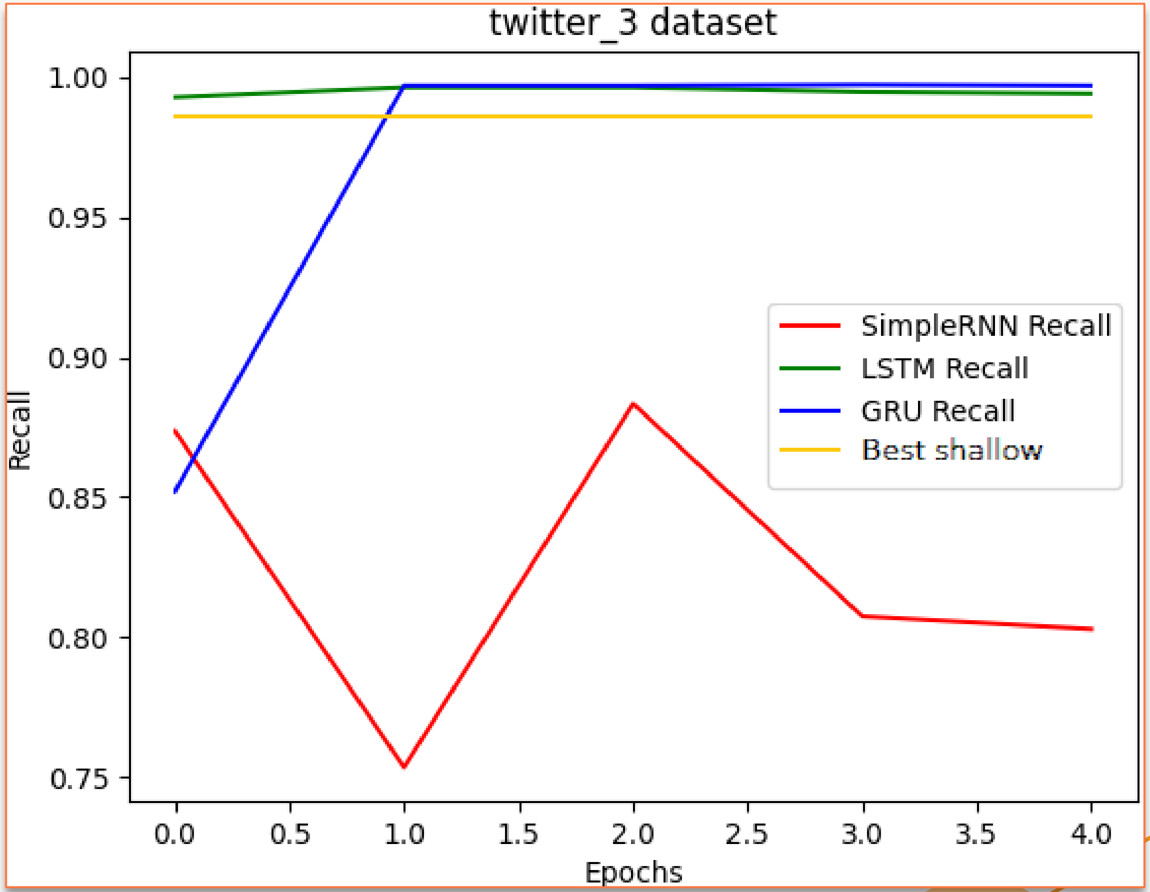
RNN


twitter_scale dataset
0.80
0.75
0.70
0.65
0.60
Recall
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Epochs
SimpleRNN Recall
LSTM Recall
GRU Recall
Best shallow

Results



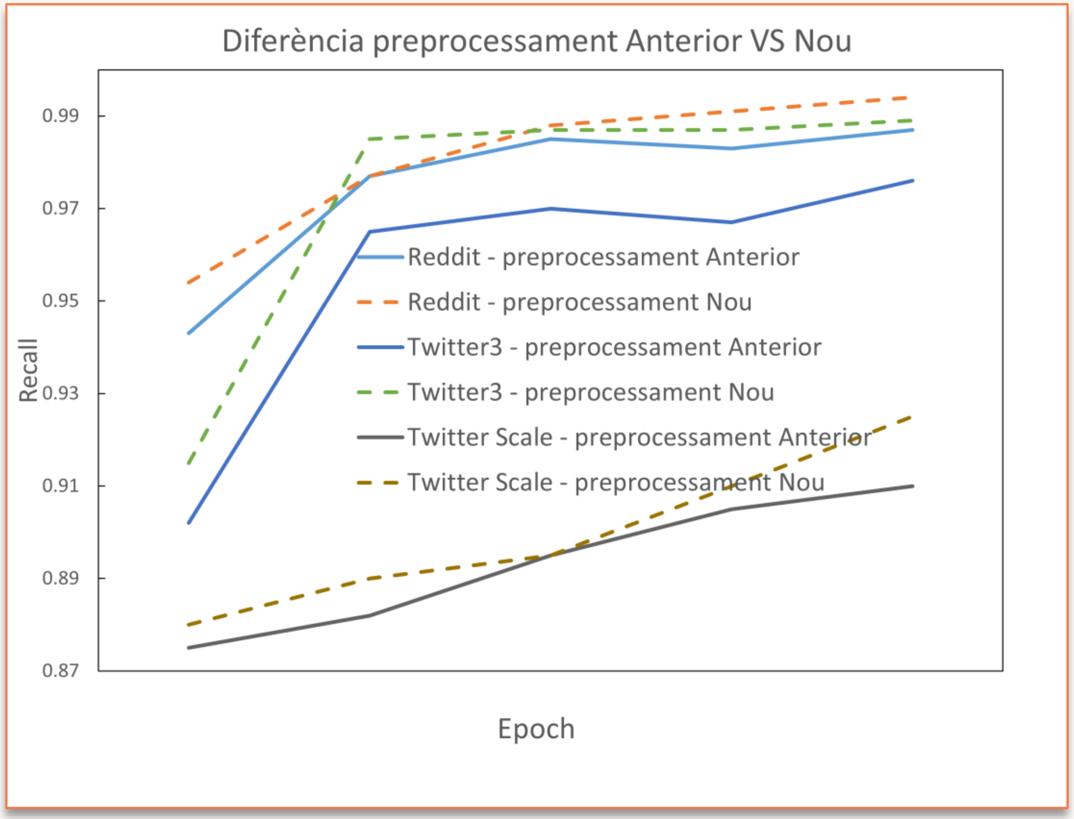
Reddit dataset
Recall
0.9
0.8
0.7
0.6
0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Epochs
SimpleRNN Recall
LSTM Recall
GRU Recall
Best shallow



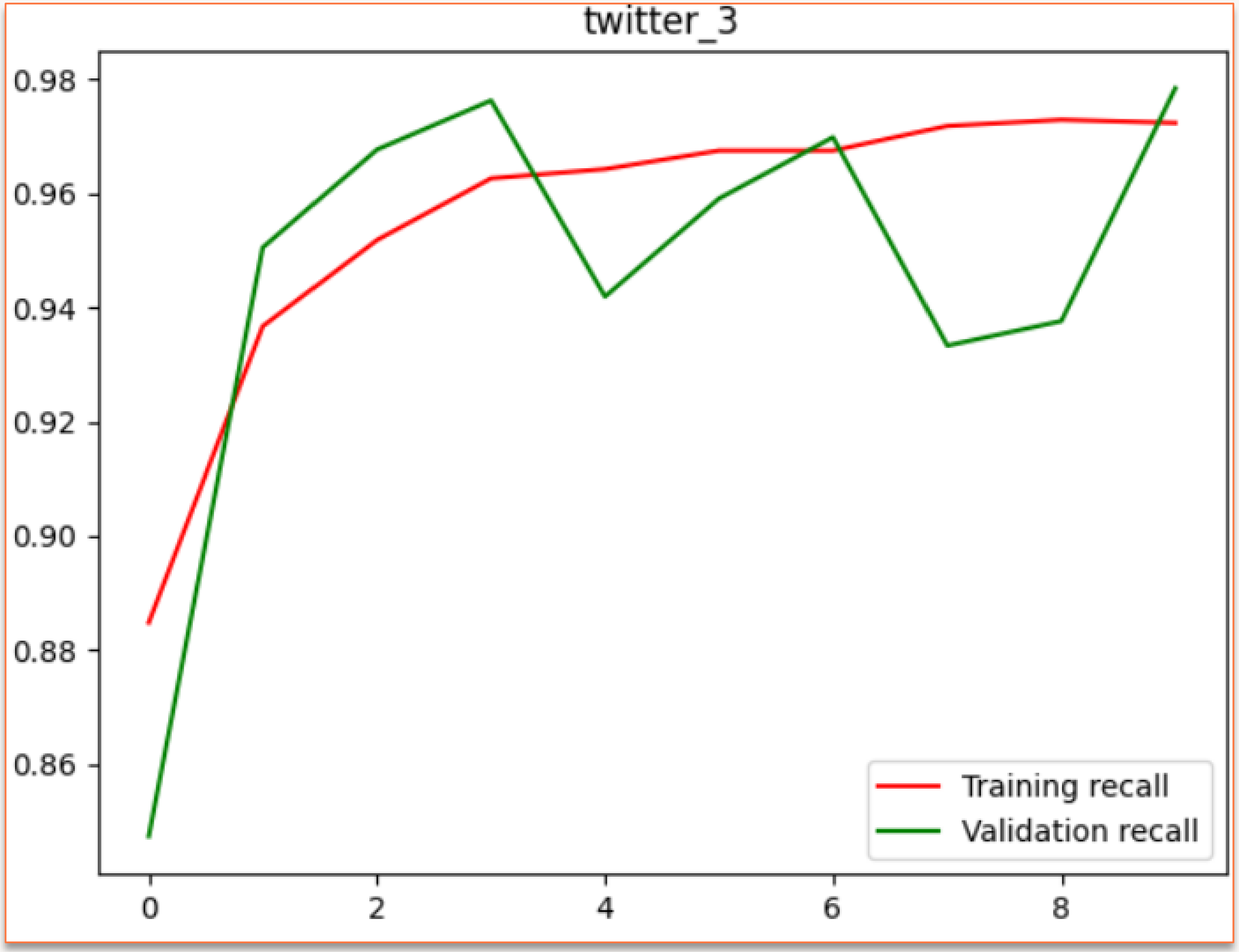
twitter_3 dataset
Recall
1.00
0.95
0.90
0.85
0.80
0.75
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Epochs
SimpleRNN Recall
LSTM Recall
GRU Recall
Best shallow

New preprocessing

- ✓ No lemmanization
- ✓ Keeping stopwords



BERT (transformers)



Differences in predictions

“study finds no casual relationship between cannabis and depression”

“dailytonic exposure to the bacteria in soil can be good for mental hearlth and could treat depression and prevent ptsd”

“don’t be sad, armys are here for you we will always suport you btstwt be strong”

Conclusions

Shallow learning

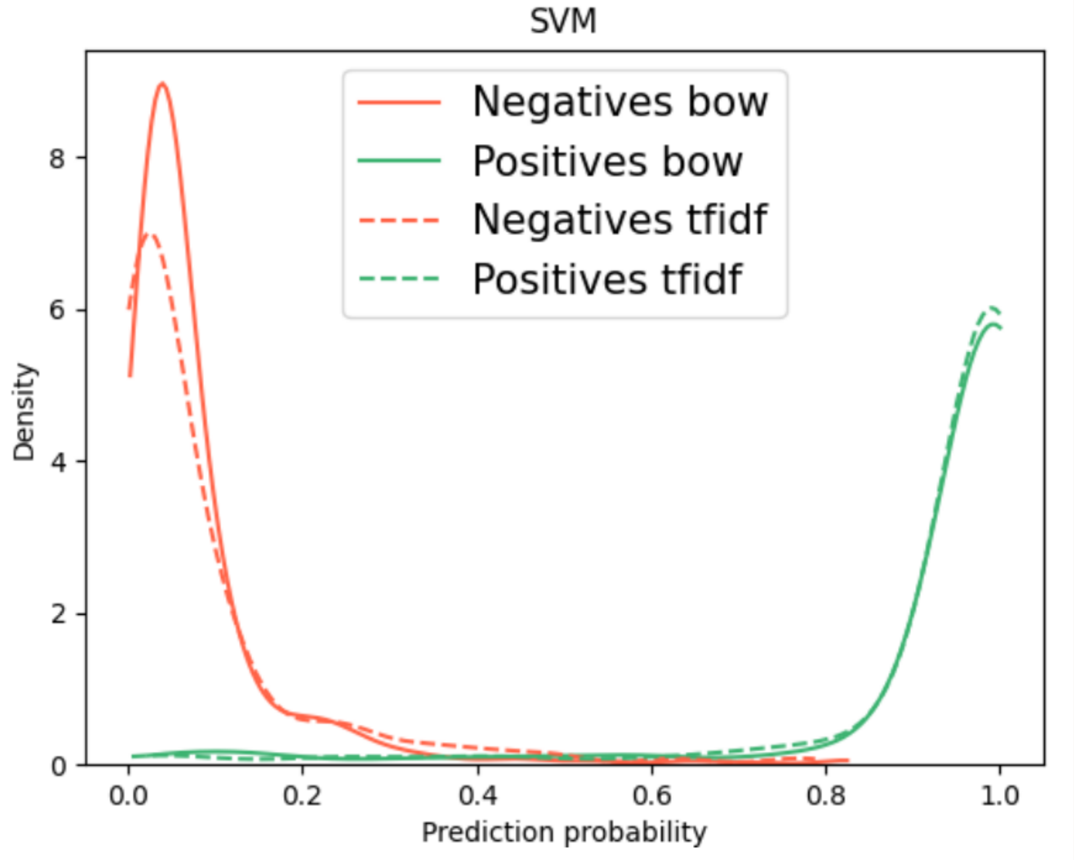
- Best: SVM and RF (relative to confidence)
- Preprocessing highly affects on metrics
- Feature extraction highly affects on execution time
- Parameters are not decisive

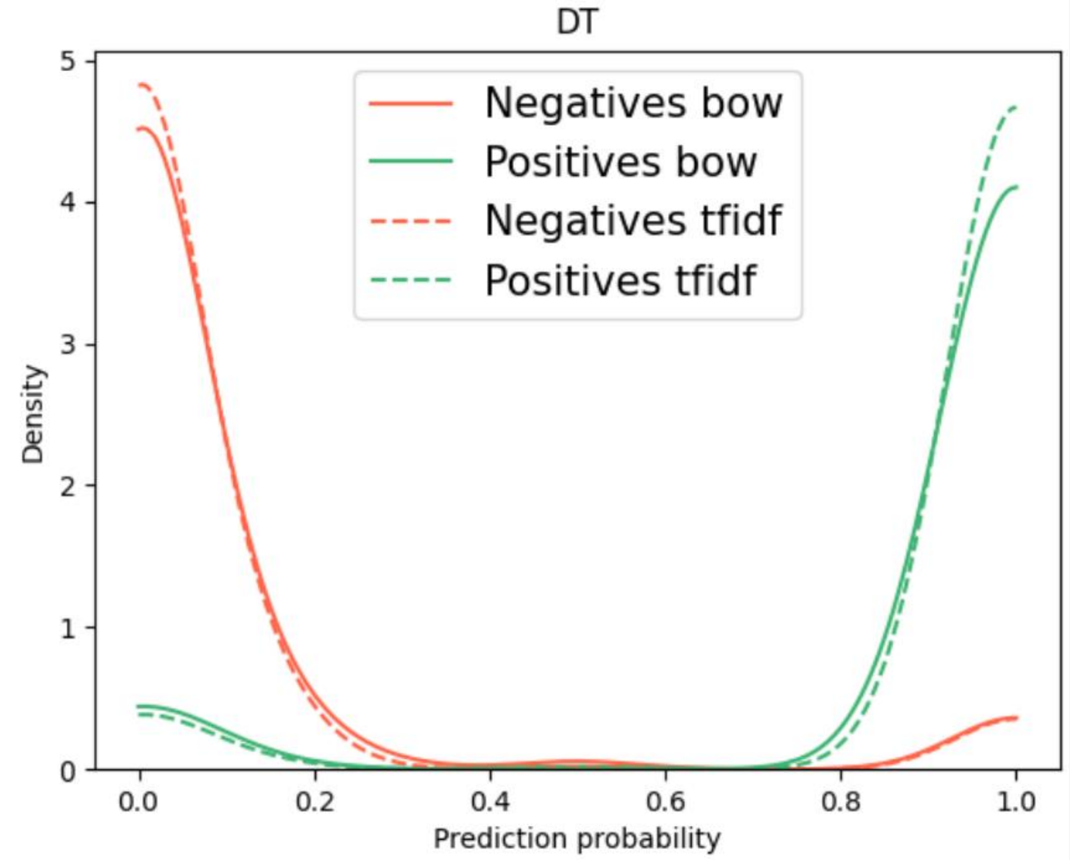
Deep learning

- Results about 10% better
- Simple RNN not good at all, GRU and LSTM are needed
- LSTM better than GRU with long messages
- Gets the semantics instead of the relations
- BERT needs more data and computing resources

Thank you

Martí Caixal i Joaniquet


SVM
Negatives bow
Positives bow
Negatives tfidf
Positives tfidf
Density
0
2
4
6
8
0.0
0.2
0.4
0.6
0.8
1.0
Prediction probability


DT
Negatives bow
Positives bow
Negatives tfidf
Positives tfidf
Density
0
1
2
3
4
5
0.0
0.2
0.4
0.6
0.8
1.0
Prediction probability

Results

Execution time