




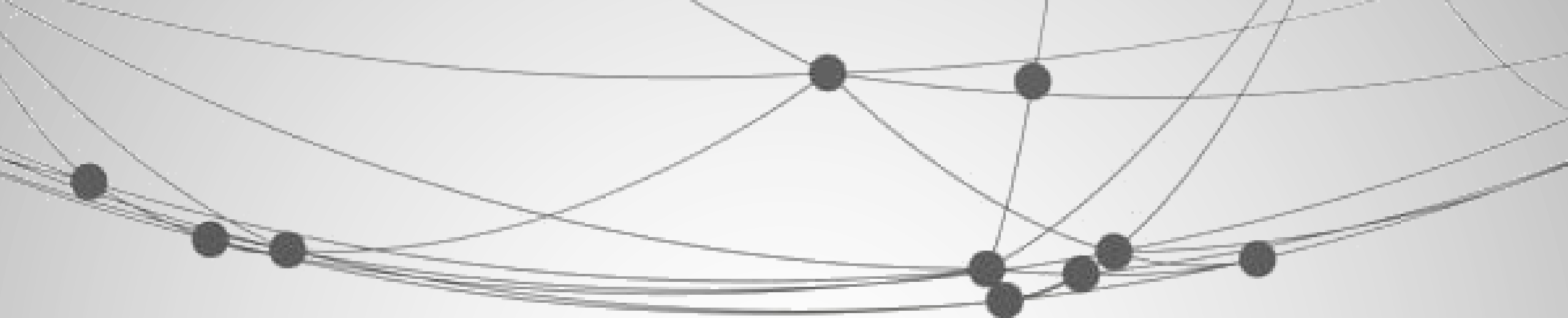
AI Engineer PATH **Project 7**

Predict sentiment associated with a tweet

A x e l F a v r e u l

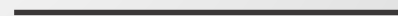
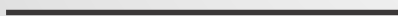
Contents

- 
- 1. Dataset overview / Methods
 - 2. API and basic customized model
 - 3. Advanced model : preprocessing
 - 4. Advances model : the models



01

Dataset overview



Dataset overview



Azure cognitive services API

Number:

1 600 000 tweet

Balanced dataset :

800 000 : positive sentiment (label 1)

800 000 : negative sentiment (label 0)

Miscellaneous :

Dataset already cleaned : no Null value, no specific data type.

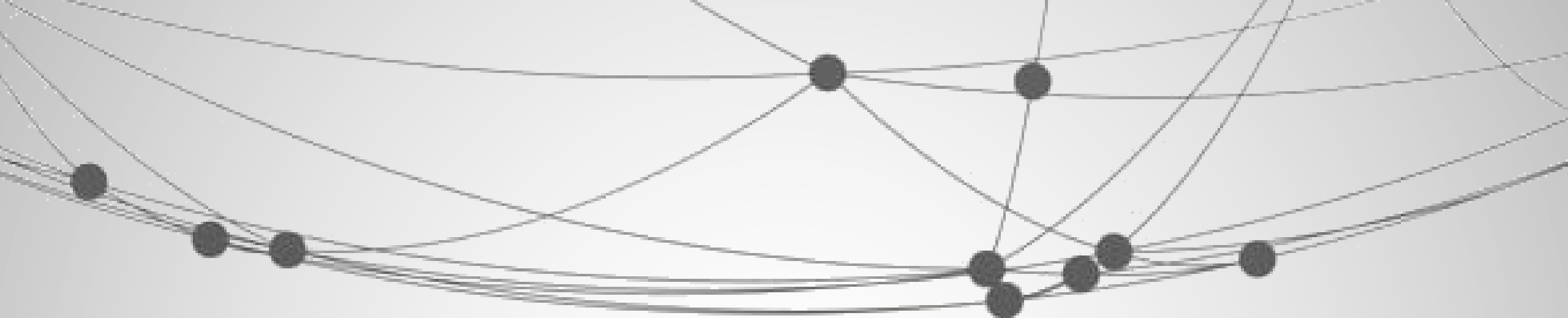
Text : tweet downloaded, no further cleaning related to the text.

Split :

80% training.

15% validation.

15% test.



02

**API and basic customized
model**



API

...

Azure cognitive services API

Methods : `textAnalyticsClient`

Sent a text to the Client.

4 Methods : `detect_language`, `entities`, `key_phrases` and `sentiment`

Response:

Contain different information including confidence score and three different scores for sentiment → Negative, neutral and positive. (Between 0 and 1).

Limitation:

Batch of 10 texts per request (for sentiment analysis)

Maximum size per text: 5120 characters

Rate, request per second low tier: 100 (300 per minute)

Performance on sample 1000:

Accuracy = 0,710

API



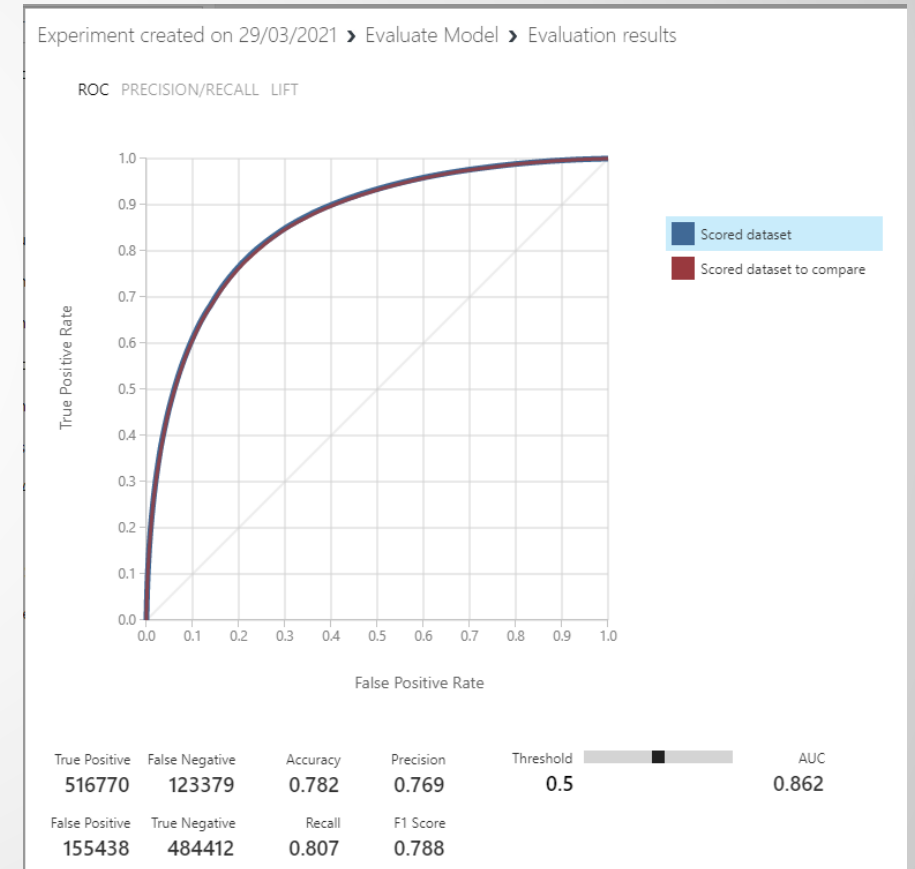
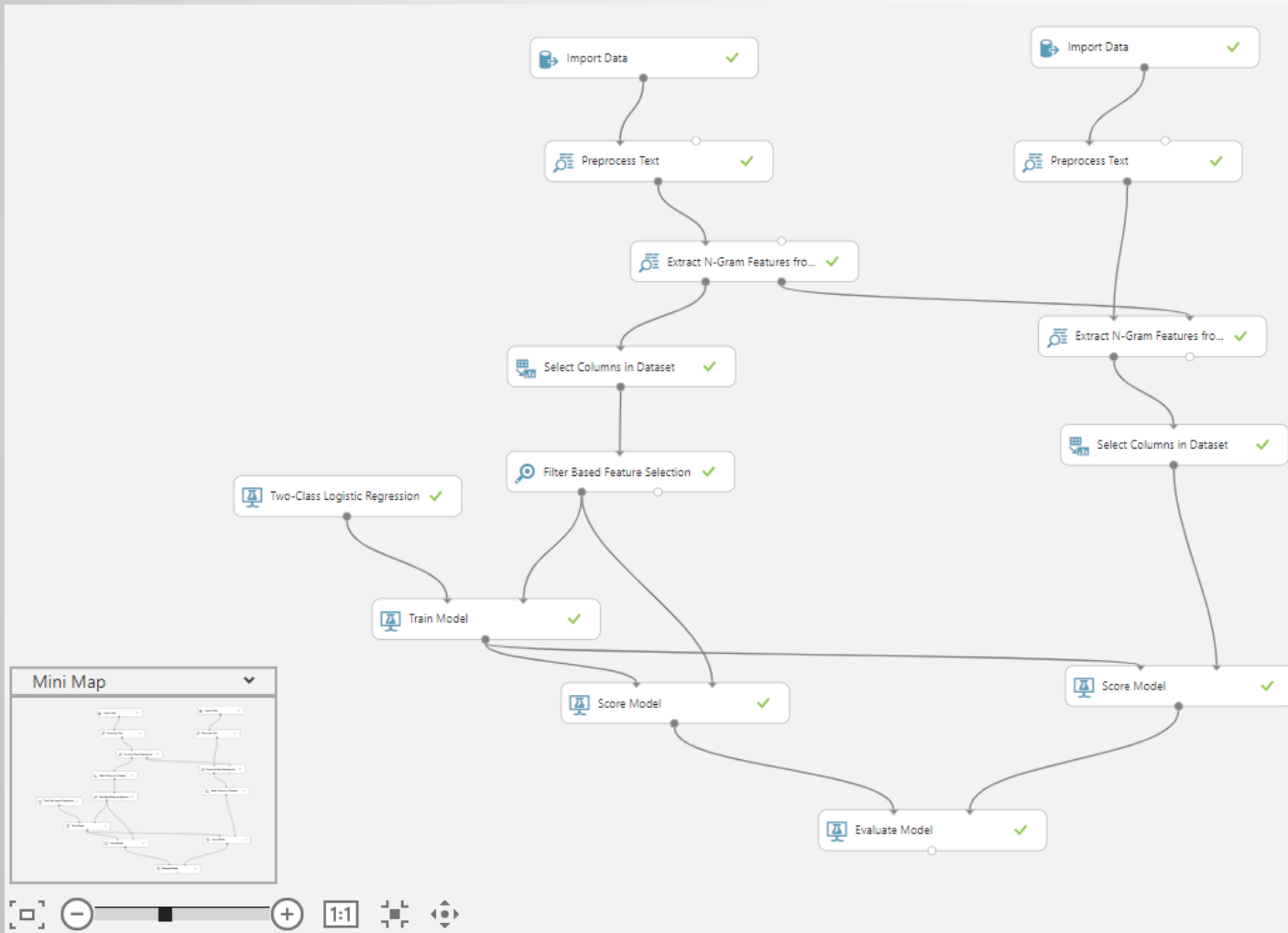
Azure cognitive services pricing

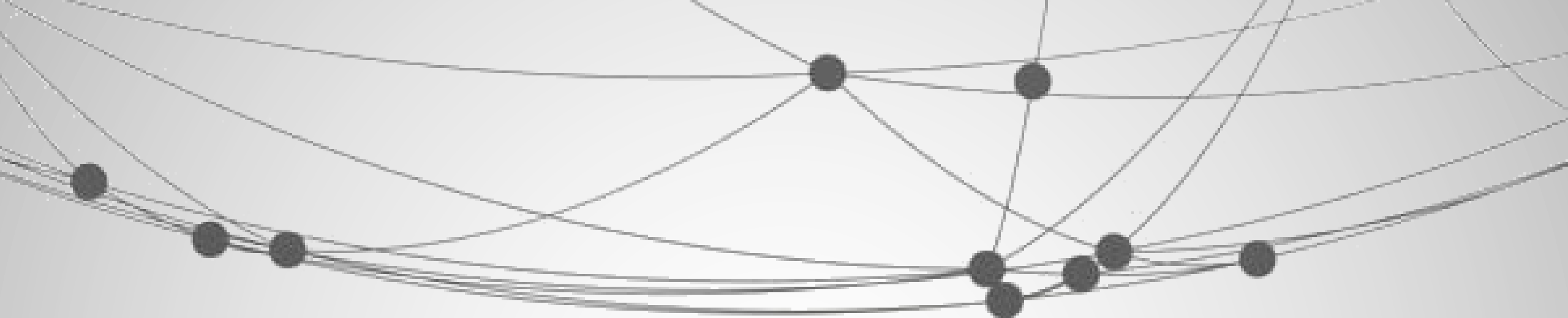
Instance	Features	Price
Free - Web/Container	Sentiment Analysis (and Opinion Mining) Key Phrase Extraction Language Detection Named Entity Recognition (not available in Container)	5,000 transactions free per month
Standard - Web/Container	Sentiment Analysis (and Opinion Mining) Key Phrase Extraction Language Detection Named Entity Recognition (not available in Container)	0-500,000 text records — \$1 per 1,000 text records 0.5M-2.5M text records — \$0.75 per 1,000 text records 2.5M-10.0M text records — \$0.30 per 1,000 text records 10M+ text records — \$0.25 per 1,000 text records
	Text Analytics for health (Preview)*	Free

Customized model



Customized model using Azure ML Studio





03

**Advanced model :
preprocessing**

Preprocessing



Two different models : Part 1 Common functionalities.

Stopwords:

NLTK English words

Lower case:

Convert all character to lower case

URL/Mail:

Delete all.

Contractions:

Python library to address a part from slang language :

“yallre beautiful” → “you all are beautiful”

ASCII:

Remove non ASCII

Repeating characters:

Remove repeating characters
Whaaaaaat => whaat

Number:

Remove all.

Stemming.

Remove occurrence less than 2 in the corpus.

Preprocessing

...

Two different models : Part 2 differences.

Model 1:

Remove all punctuation

Vocabulary size: ~ 56 000

Model 2:

Use the tweet tokenize function from NLTK to identify and keep emoji.

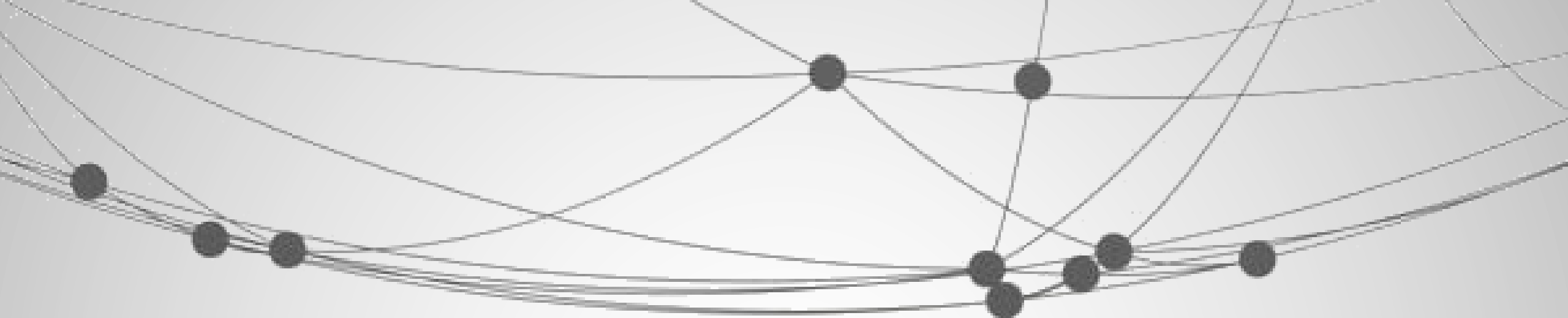
“@remy this is coooooool :-)” →
[this, is, cool, :-)]

Vocabulary study

Most used emoji hit the top 100 most used words

Vocabulary size: ~57 000

Remarks : 60 000 usually top bracket for NLP



04

Advances model : the models

Model

...

3 different models :

Model 1:

Logistic regression

Tokenizer:

CountVectorize

Parameters:

Solver : Saga (fast for big dataset) uses random sample of previous gradient values.

Max iteration could be optimized with gridsearch

C parameters : see after

Model 2:

Simple RNN

Tokenizer:

Keras

Layer:

Embedding,
SimpleRNN,
Dense,
Dropout,
Dense.

Model 3:

LSTM

Tokenizer:

Keras

Layer:

Embedding,
LSTM,
Dense,
Dropout,
Dense.

Model

...

3 different models :

Model 1:

Logistic regression

Performance:

```
Accuracy for C=0.01: 0.74508984375
Accuracy for C=0.05: 0.76069140625
Accuracy for C=0.25: 0.76687890625
Accuracy for C=0.5: 0.7676171875
Accuracy for C=1: 0.76751171875
```

Model 2:

Simple RNN

Training parameters:

Epoch: 50
Batch_size = 128

Accuracy:

0,729

Model 3:

LSTM

Tokenizer:

Epoch: 20
Batch_size = 128

Accuracy:

0,785

Model pre-trained embedding

...

Glove special tweeter matric:

Contain flag for the following :

- Emoji
- Repeating letter
- Repeating punctuation
- Hashtag
- Out of Vocabulary

Pre-processing:

No stemming

Keep punctuation

Contraction applied (slang)

Pre-tuning :

49,87% of the vocabulary from the corpus

95,12% of all text

Post tuning :

92,84 % of the vocabulary from the corpus

99,59% of all text

OOV example :

Lvatt, twitterfon, mcflyforgermany,
haveyouever, mileymonday.

Model 3:

LSTM

Tokenizer:

Epoch: 20

Batch_size = 128

Accuracy:

0,83

Conclusion

...

API

- Limitation quite important
- Cannot be fit to tweet text
- Quick use

Azure ML Studio

- Fast design
- Fast training
- Fast deployment



Customized model:

- Easy follow of model improvement
- Can be fit to the model
- 24h training
- Deployment TBD
- Good performance from Glove Tweeter embedding.

Way ahead:

- Facebook just release a new pre-trained embedding including misspelling and slang. (MOE) Only dataset available not the matrix.



QUESTION ?

OpenClassRooms: Project 3

A x e l F a v r e u l