The pandemic behind

The Pandemic

Social media response to Covid-19 breakthrough

Our Goals

A

Covid-19 and media influence

Finding relations between day by day Covid-19 evolution and information flow

1

Social Network response - Reddit

Targeted analysis on how Covid-19 discussion evolved in a social network: Reddit

2

Journalism response

Regional analysis on how Covid-19 discussion evolved in journalism

Work progress

DATA COLLECTION

We collected data from Reddit, various news outlets and LaProtecioneCivile

DATA ANALYSIS

We developed for polarity detection models, topic models and predictive models

CONCLUSIONS

We tried to gain insight on how people generally reacted to the breakthrough

Our group

Negin Amininodoushan
Marco Muscas
Leonardo Placidi
Stefano Rando
Davide Zingaro

Let’s get started!

Enjoy your ride!

Reddit Section 1:
Sentiment Analysis on Reddit

Detecting polarity of comments on Reddit

Contents

SENTIMENT ANALYSIS

Classifying documents as either positive, negative or neutral

MODELS

Naive Bayes, SVMs and BERT pre-trained models

APPLICATION

Classifying Reddit italian comments concerning Covid-19

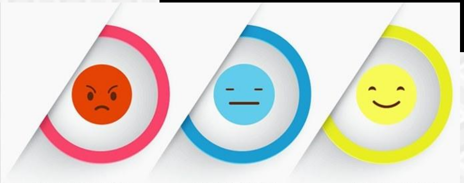
01.

Sentiment Analysis

Classifying documents on the basis of polarity

What is Sentiment Analysis?

In sentiment analysis, or opinion mining, (SAOM), the goal is to discover people’s opinions expressed in written language (text). Sentiment in term means “what one feels about something”, “personal experience, one’s own feeling”, “an attitude toward something” or “an opinion”[4].



02.

Models

A brief journey in SVM and Bayes classifiers.

Recap of SVM

SVM

Support Vector Machine

TASK

Build an hyperplane or set of hyperplanes in high-dim space

GOAL

Predict the sentiment of comments (classification)

REDUCTION

Dimensionality reduction methods

Bayes Classifiers

We tried many methods based on Bayes theory, more on this later!

Gaussian
Bernoulli
Multinomial

BERT

A.

NEURAL NETWORK

Multilayers and flexible

B.

PRE-TRAINED

The first layers are trained on huge data!

C.

RE-TRAINED

Train set with already classified texts

D.

ATTENTION LAYERS

Layers to avoid RNNs (usually too slow)

03.

Application

Scoring methods and model selection

Our Train Data comes from 3 sources:

01

Reddit comments classified **BY HAND BY US** ~#600

02

Amazon reviews that we scraped and classified ~#500

03

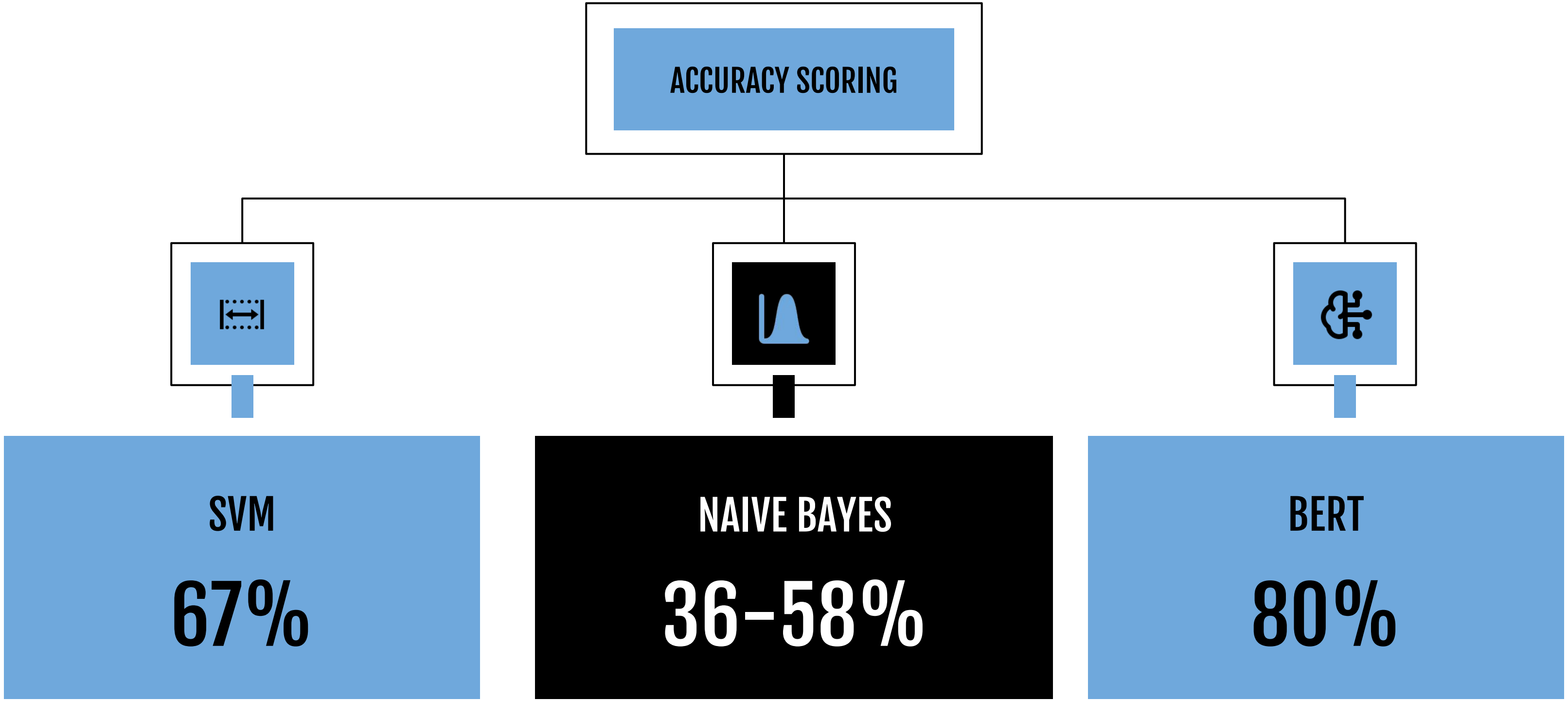
Wikipedia articles as example of neutrality ~#140

Validation set ~200.

OUR MONSTER TEST SET!

#154656 of comments

AND THE BERT METHOD IS.... BEST



ACCURACY SCORING

SVM

67%

NAIVE BAYES

36–58%

BERT

80%

Project data

Reddit italian comments on Covid-19 specific threads

Data structure

FEBRUARY

Wednesday

26

“Eccetto grandi carenze di amuchina non mi pare tutto sto casino, la gente esce, lavora normalmente.”

MARCH

Saturday

14

“Domani si festeggia ufficialmente il funerale di tutte le partite Iva d’Italia.”

APRIL

TUESDAY

21

“Dite quello che volete, ma il silenzio della sera è una cosa bellissima e mi mancherà.”

Reddit Section 2: Topic modeling of Reddit comments

Latent Dirichlet Allocation and evaluation methods

Contents

TOPIC MODELING

What is a Topic model and problem formalization

LDA

Method presentation: Latent Dirichlet Allocation

EVALUATION

How to measure efficiency of a Topic model

01.

Topic Modeling

How to classify documents in an unsupervised setting

The main importance of topic modeling is to discover patterns of word-use and how to connect documents that share similar patterns.

—[1]

Main aspects of Topic Modeling

A.

Unsupervised setting

The model has to find patterns by itself

B.

Text mining

Use of text processing techniques

C.

Clustering

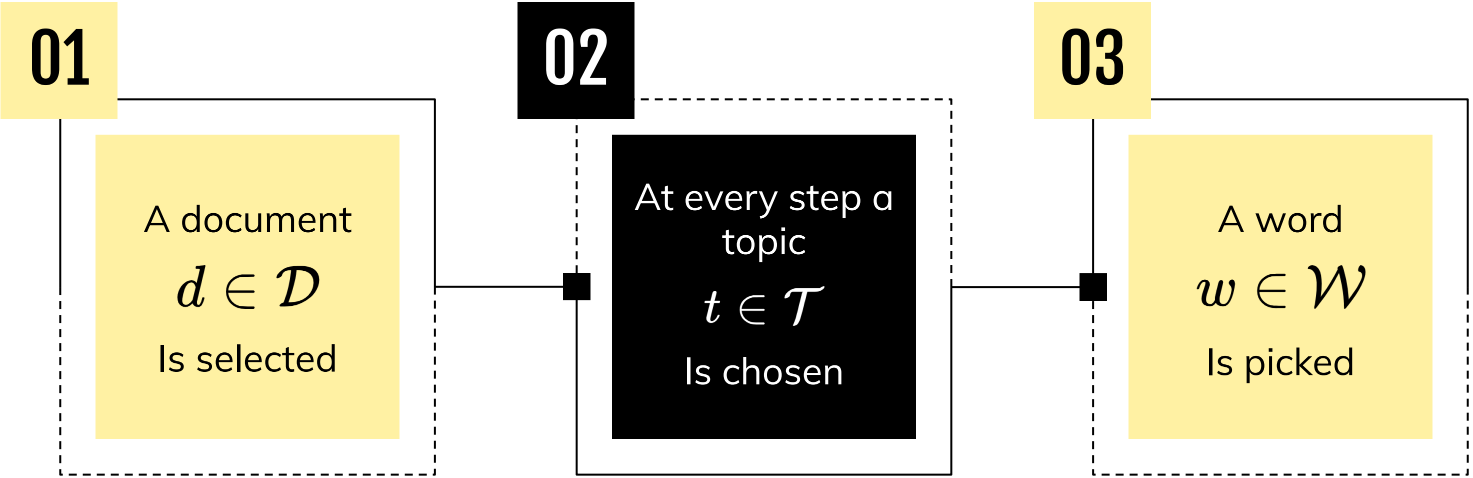
Associations of documents to topics

D.

Generative approach

Leverage on Bayesian models

Formalization and assumptions



01

A document

$d \in \mathcal{D}$

Is selected

02

At every step a topic

$t \in \mathcal{T}$

Is chosen

03

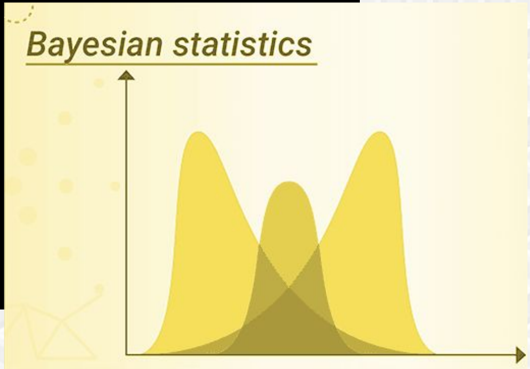
A word

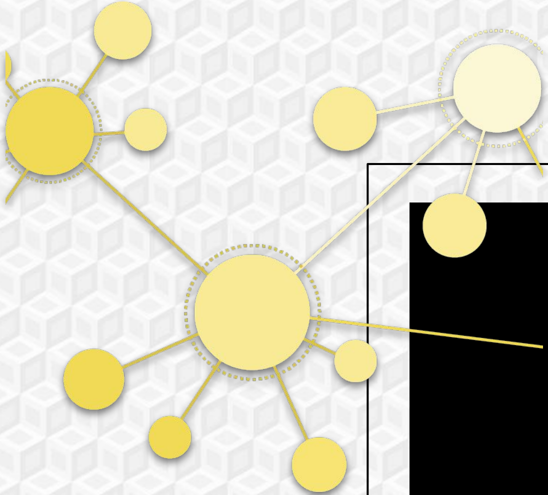
$w \in \mathcal{W}$

Is picked

Our Problem – Find

$$P(d, t)$$





Research

Trying to identify around which topics Covid-19 discussions revolved in social networks like Reddit.

02.

Latent Dirichlet Allocation

A Bayesian hierarchical model for topic modeling

Recap of Bayesian statistics

MODEL

A model, depending on some parameters, is assumed

PRIOR PROBABILITY

Parameters of the model are treated as random themselves

HIERARCHICAL

More models can sequentially depend from one another

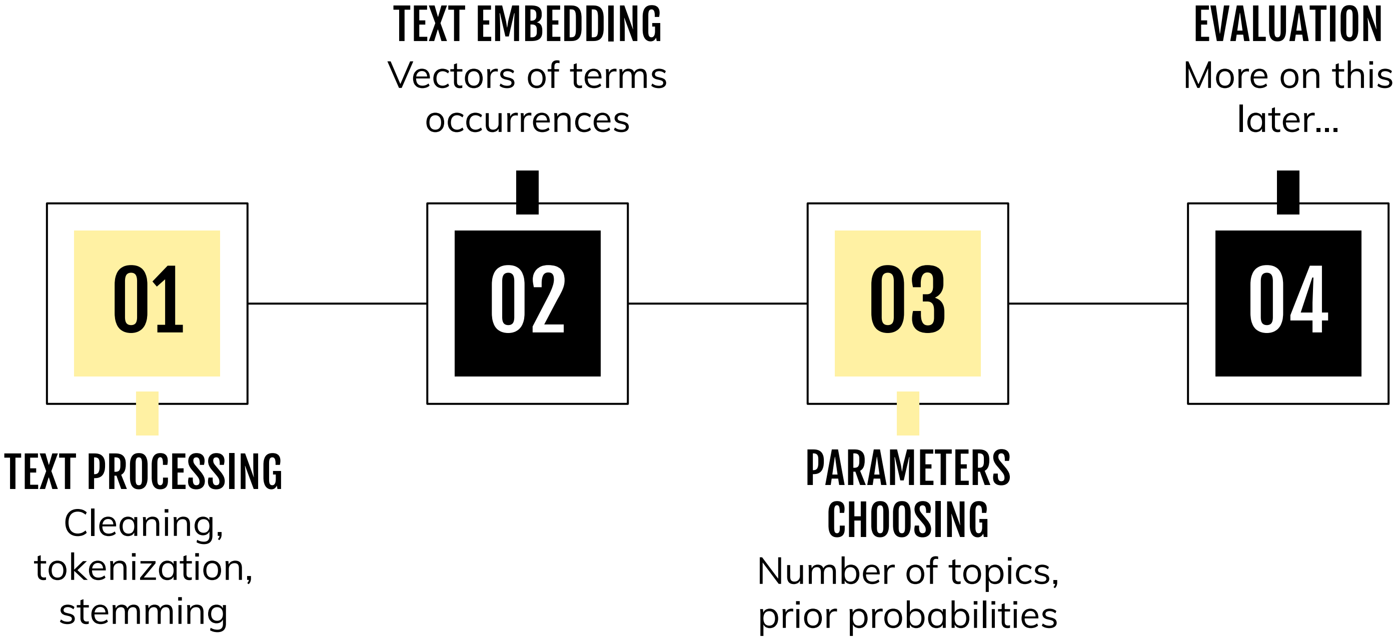
POSTERIOR PROBABILITY

Distribution of parameters is updated based on data

Latent Dirichlet Allocation

	DISTRIBUTION	PARAMETER	KNOWING
Choose number of words N	Poisson	η	-
Choose topics distribution θ	Dirichlet	α	-
Then for every word...			
Choose a topic z_n	Multinomial	θ	-
Choose a word w_n	Multinomial	β	z_n

Steps



01

TEXT PROCESSING

Cleaning, tokenization, stemming

02

TEXT EMBEDDING

Vectors of terms occurrences

03

PARAMETERS CHOOSING

Number of topics, prior probabilities

04

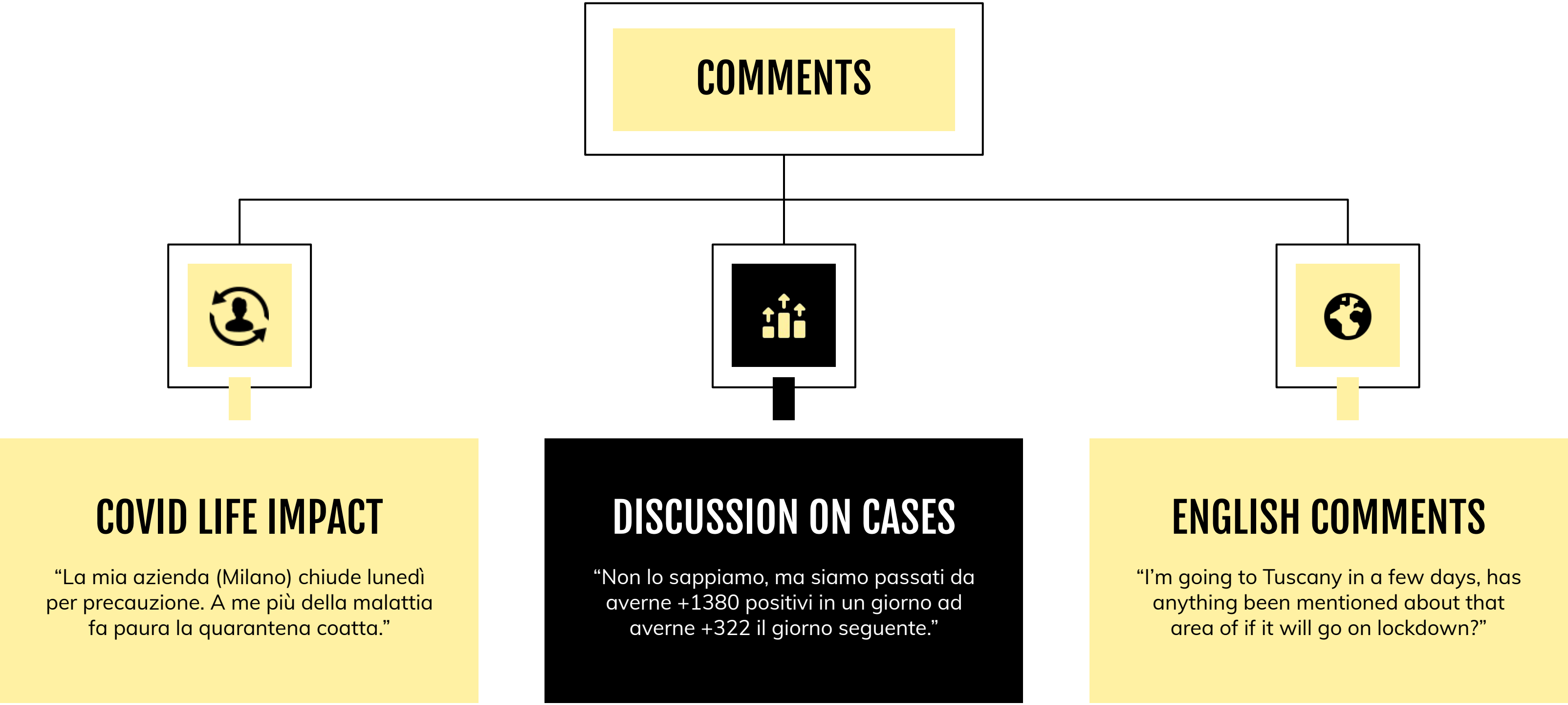
EVALUATION

More on this later...

Application on Reddit comments

LDA applied to project dataset identified three trends

Categories found with LDA



COMMENTS

COVID LIFE IMPACT

“La mia azienda (Milano) chiude lunedì per precauzione. A me più della malattia fa paura la quarantena coatta.”

DISCUSSION ON CASES

“Non lo sappiamo, ma siamo passati da averne +1380 positivi in un giorno ad averne +322 il giorno seguente.”

ENGLISH COMMENTS

“I’m going to Tuscany in a few days, has anything been mentioned about that area of if it will go on lockdown?”

casa
solo
molto
fare
stato
persone
lavora
capisco
fatto
essere

68%

22%

italia
ieri
numeri
giornata
tamponi
positivo
nuovi
oggi
dati

coronavirus
reddit
italywiki
youtube
italyhtml
deleted
news
wiki
html

10%

03.

Evaluation

Measures of efficiency for Topic models

Problems

Can we trust LDA topics recognition?
How many topics should we try to detect?

Approaches

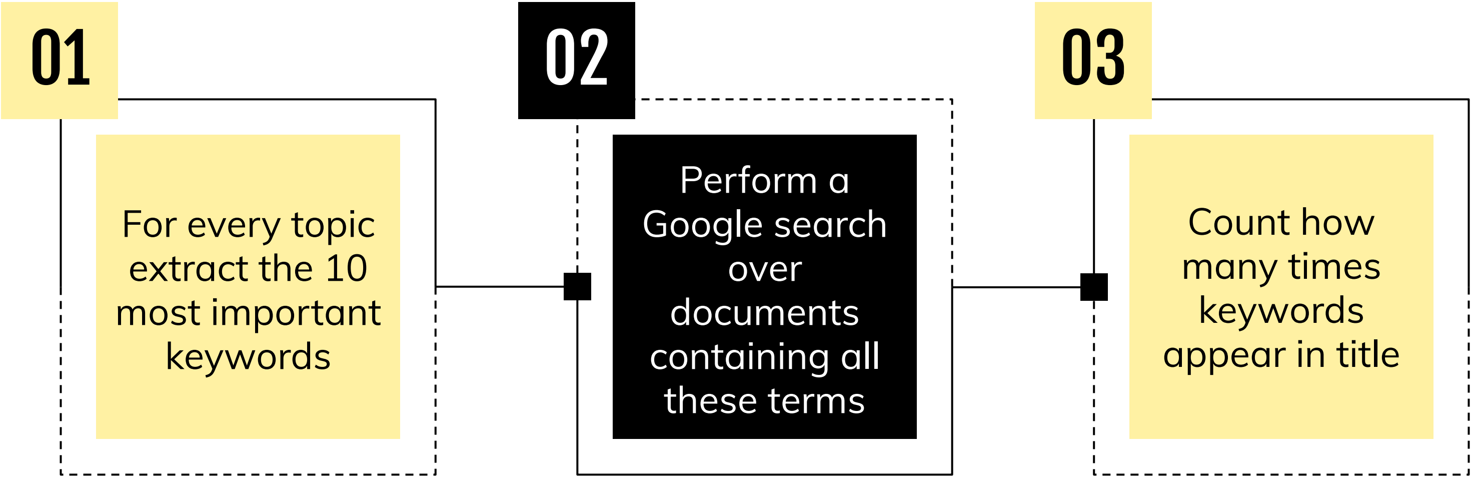
AUTOMATIC

Unsupervised evaluation. No human intervention or classification.

HUMAN-BASED

Supervised evaluation. Different people subjectively evaluate effectiveness.

Automatic: Google titles matches [2]



01

For every topic extract the 10 most important keywords

02

Perform a Google search over documents containing all these terms

03

Count how many times keywords appear in title

Human-Based: Word intrusion [3]

TOTALE	□ □ □ □	Related - No votes
MORTI	□ □ □ □	Related - No votes
GIORNATA	■ ■ ■ ■	Related - 4 votes
REGIONE	■ ■ □ □	Related - 2 votes
CORONAVIRUS	■ □ □ □	Related - 1 vote
DEFIBRILLATORE	■ ■ ■ ■	Unrelated - 4 votes

Evaluation of the comments model

Number of topics selection and effectiveness evaluation

TITLES for choosing number of topics and model

	2	3	4	5	6	7
LDA	77.5	77.3	25.3		0	
Median	77.5	99	14		0	
R-LDA	77.5	77.3	25.3		0	
Median	77.5	99	14		0	

Word intrusion for assessing effectiveness

60%

Average rate of correct answers

83%

Questions which obtained most votes for correct answers

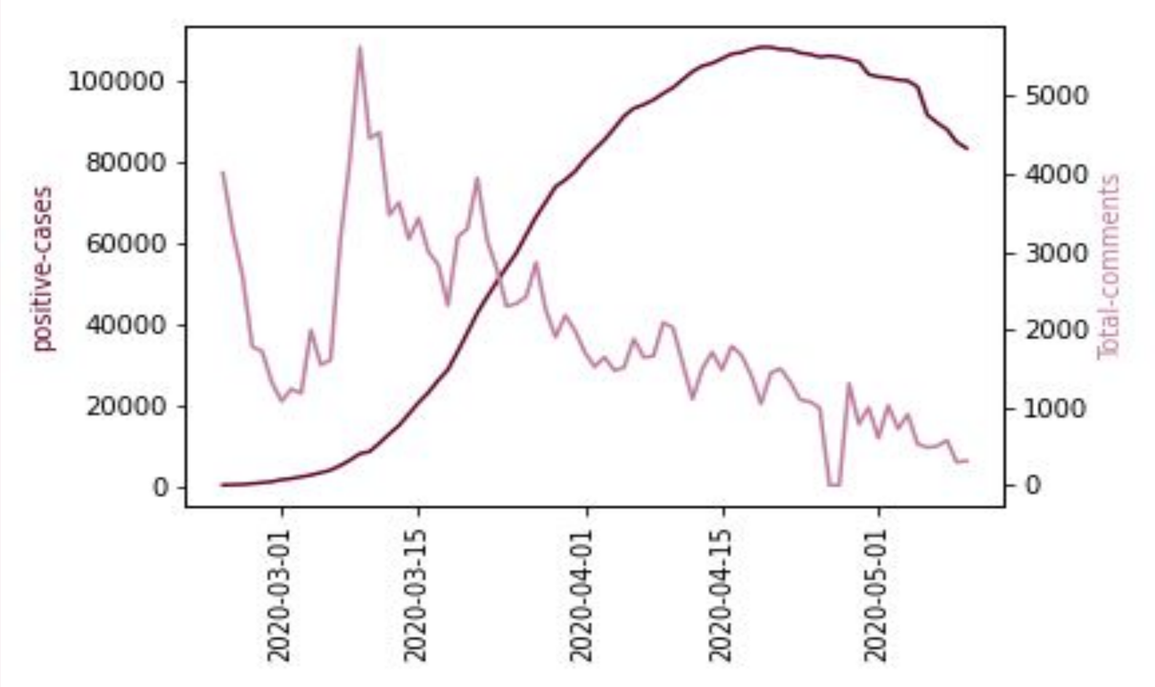
11

Participants

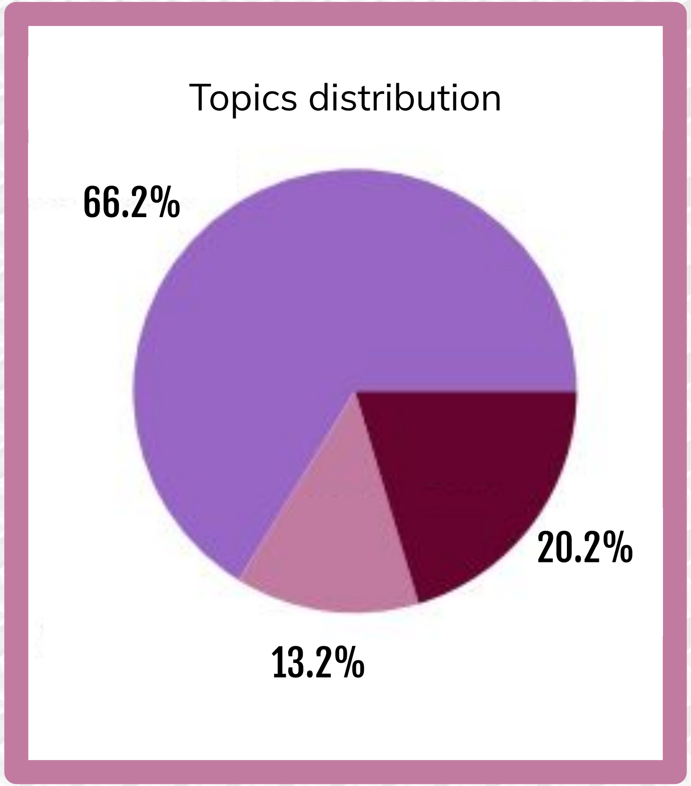
Reddit Section 3: Time analysis of Reddit comments trends

Evolution of topic and sentiment trends on comments over time

Reddit comments and total positive



Topic modeling results

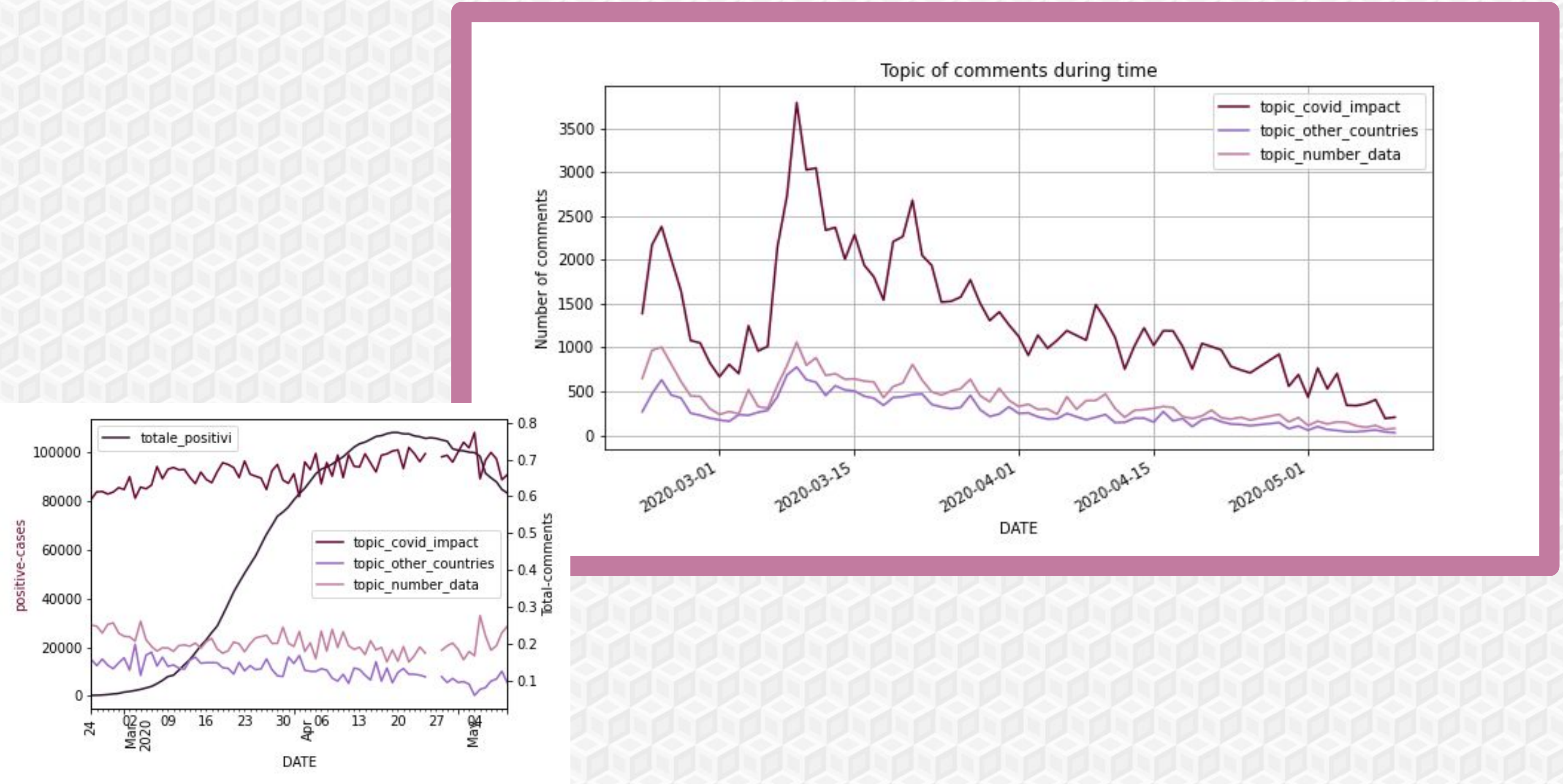


Covid impact on life

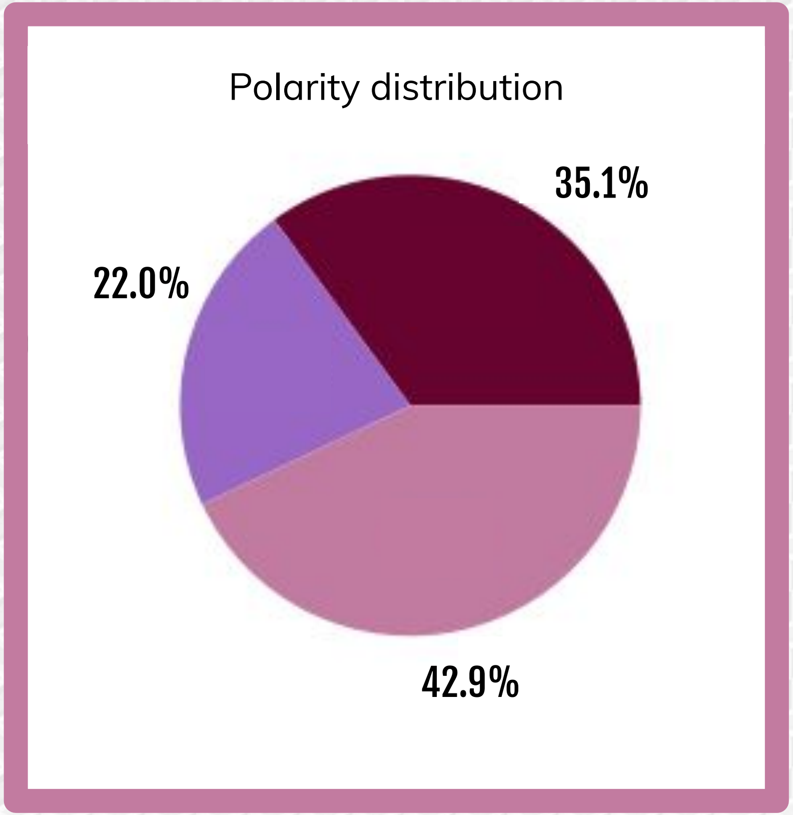
English comments

Discussion on cases and numbers

Topic distribution over time



Sentiment analysis results

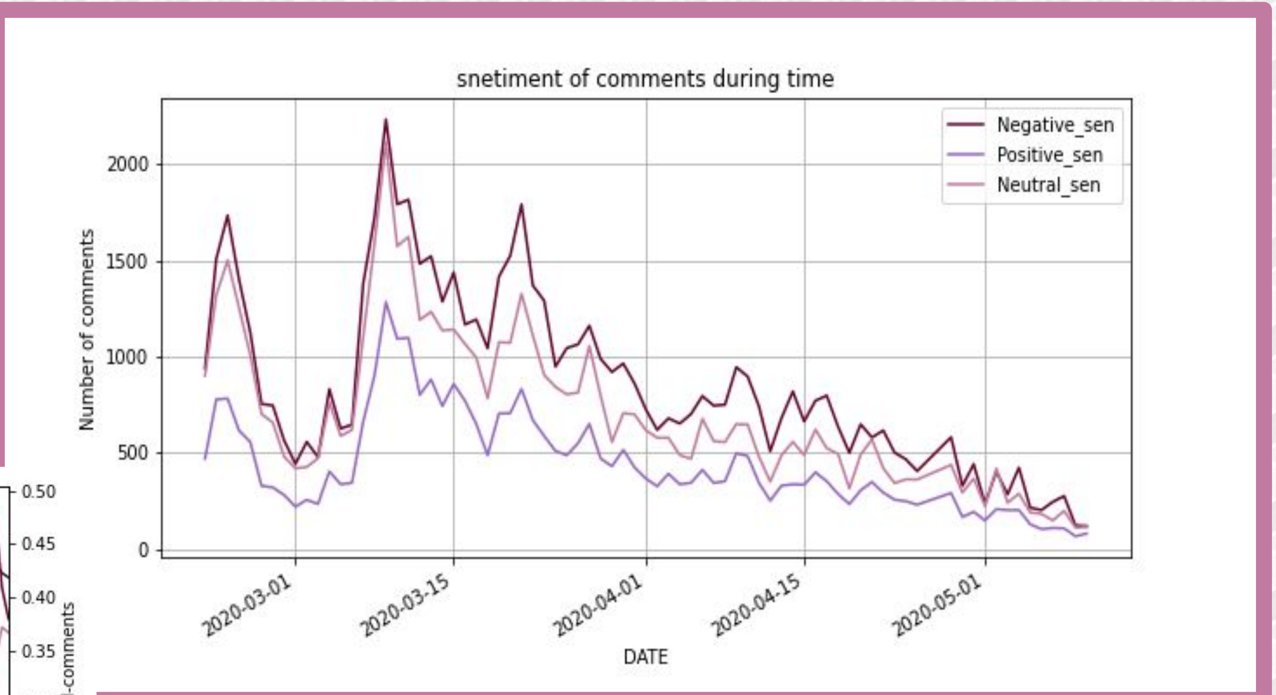


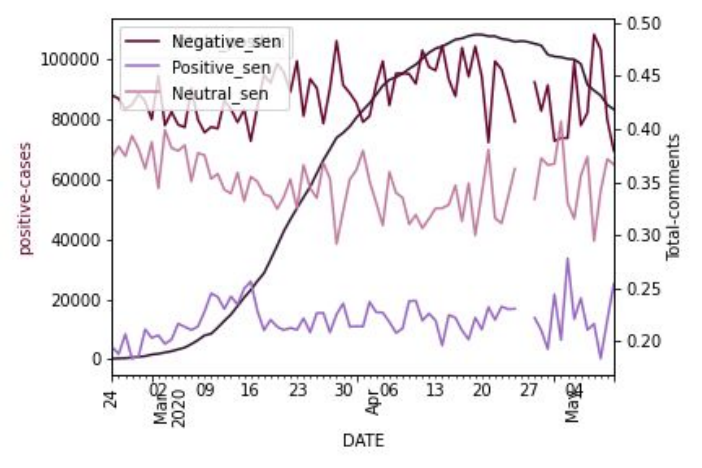
Positive comments

Negative comments

Neutral comments

Sentiments over time





References

[1] - Alghamdi, Alfaqi “A Survey of Topic Modeling in Text Mining” *International Journal of Advanced Computer Science and Application* 2015

[2] - Newman, Han Lau, Grieser, Baldwin “Automatic Evaluation of Topic Coherence” *Association for Computer Linguistics* 2009

[3] - Wang “Topic Modeling: A Complete Introductory Guide” *ResearchGate* 2017

[4] - Fundamentals of Sentiment Analysis and Its Applications
Mohsen Farhadloo and Erik Rolland, *ResearchGate* 2016

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.

Topic Modeling for articles & predictive model

Contents:

1. Data collection:
    a. Download of html pages
    b. Creating the parsers for each news outlet website
    c. Filling of missing data
2. Preprocessing of data:
    a. Stemming, Removal of special characters from text
    b. Removal of stop-words
3. Classification:
    a. Arbitrary keywords classifier
    b. Latent Dirichlet Allocation
    c. Non-negative matrix factorization
4. Plotting of classified data
5. Adding new features to the data
6. Creation of predictive model
7. Test and performance evaluation

1. Data Collection

a. Download of html pages

A first try....

Downloading from an **news indexing** site:

Pro:

- Ease of access and download

Con:

- Missing articles
- Imbalance of available data
- Anti crawling

a. Download of html pages

The final solution...

Target the <u>individual</u> news outlets websites

b. Creating the parsers for each news outlet website

We had an URL for each article:

1. Individual parsers for each news outlet

2. Parallelizing the requests and parsing for each news outlet

Note: We were careful not to overload the websites with requests

C. Filling of missing data

Each **Parser** created a CSV.

The CSVs were **merged**…

…**Incomplete data** was dropped

Result:

24584 articles

2. Preprocessing of Data

a. Stemming, Removal of special characters from text

Conversion to lower-case

Both title and content

~~?!#~~

Special character removal

Library defined punctuation and RegEx, removal of stop-words

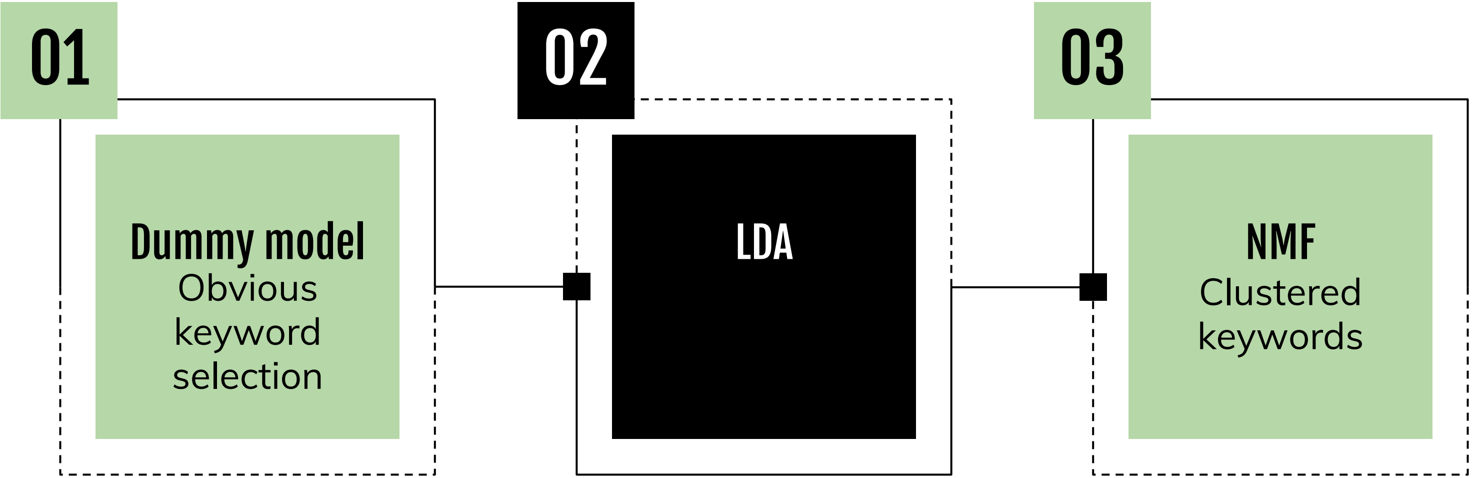
Notizie → notiz

Stemming

Snowball Stemmer method for italian language

3. Classification

We chose between 3 possible models...



01

Dummy model
Obvious keyword selection

02

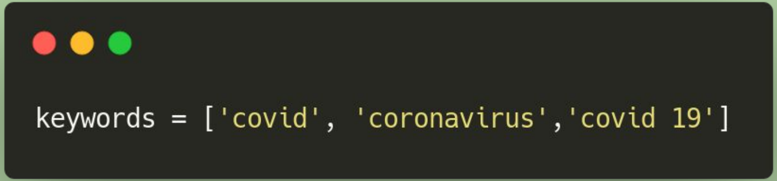
LDA

03

NMF
Clustered keywords

a. Arbitrary keywords classifier

- Used as **Reference Model…**
- Matching arbitrary keywords

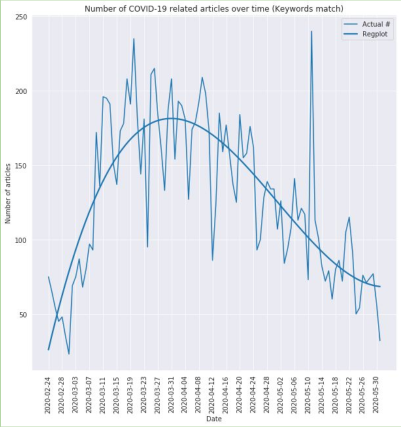


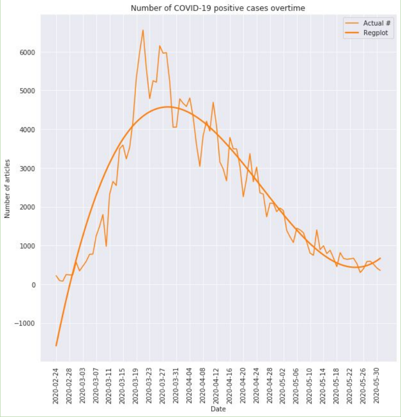
```
keywords = ['covid', 'coronavirus','covid 19']
```

a. Arbitrary keywords classifier

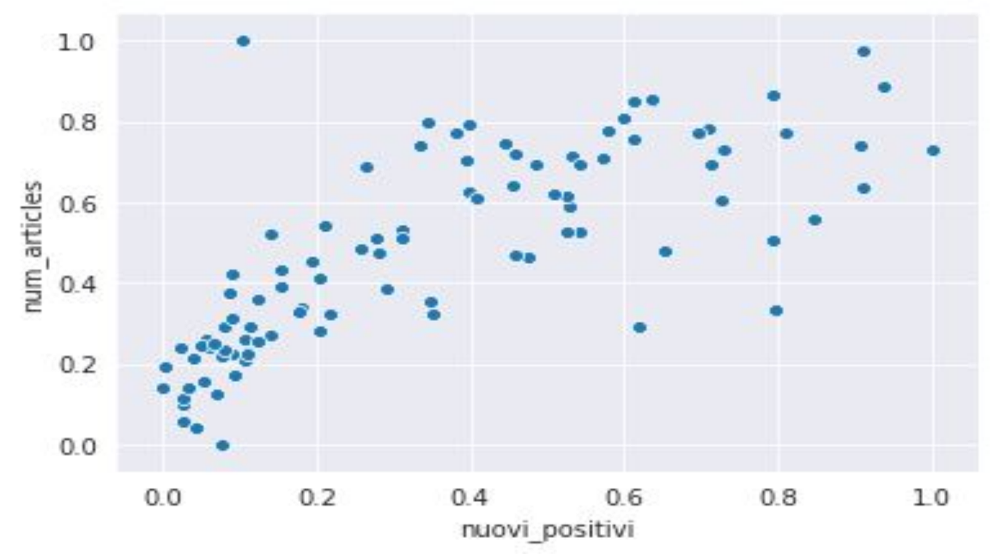
It was **not super great**…

…Too many **false positive** as more keywords were added





a. Arbitrary keywords classifier



The correlation value is **0.81,** p-value: **6,99*E-24**

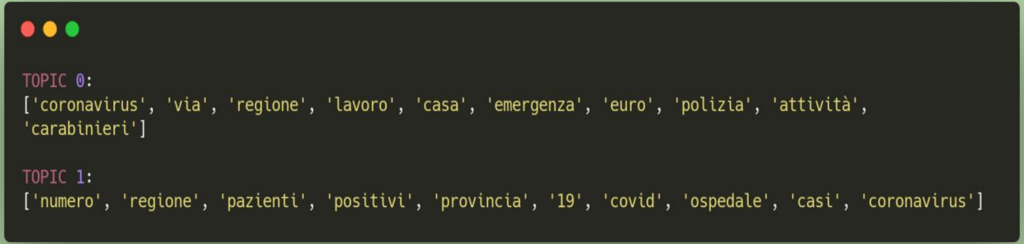
b. Latent Dirichlet Allocation

1. We created a bag of words matrix using all content from each article
2. We then used the matrix to train the LDA classifier

The results were not useful...

b. Latent Dirichlet Allocation

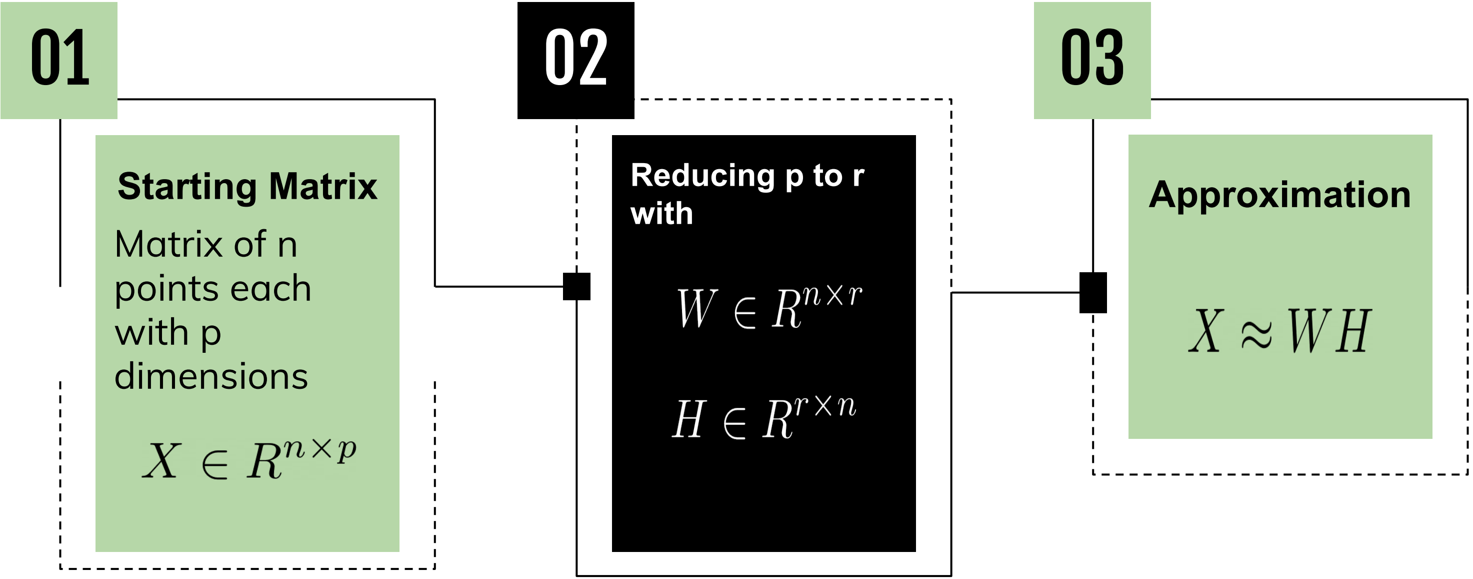
Even for two clusters...



```
TOPIC 0:
['coronavirus', 'via', 'regione', 'lavoro', 'casa', 'emergenza', 'euro', 'polizia', 'attività',
'carabinieri']

TOPIC 1:
['numero', 'regione', 'pazienti', 'positivi', 'provincia', '19', 'covid', 'ospedale', 'casi', 'coronavirus']
```

We couldn’t use these keywords, since some appeared in both. Classification was ambiguous.

Non-negative Matrix Factorization



01

Starting Matrix

Matrix of n points each with p dimensions

$$X \in R^{n \times p}$$

02

Reducing p to r with

$$W \in R^{n \times r}$$

$$H \in R^{r \times n}$$

03

Approximation

$$X \approx WH$$

c. Non-negative matrix factorization

For **Text Mining:**

Consider the **bag-of-words matrix representation…**

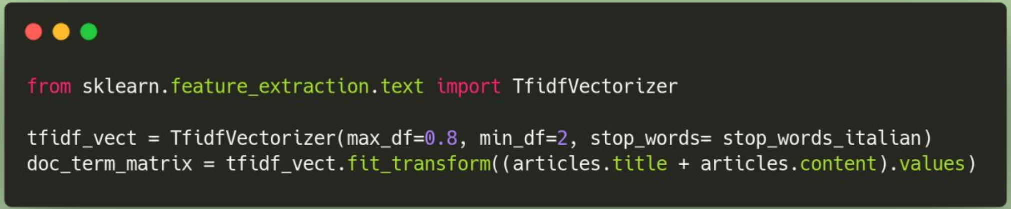
1. **Row** corresponds to a word
2. **Column** to a document

C. Non-negative matrix factorization

To perform...

1. **Clustering**
2. **Dimensionality reduction**

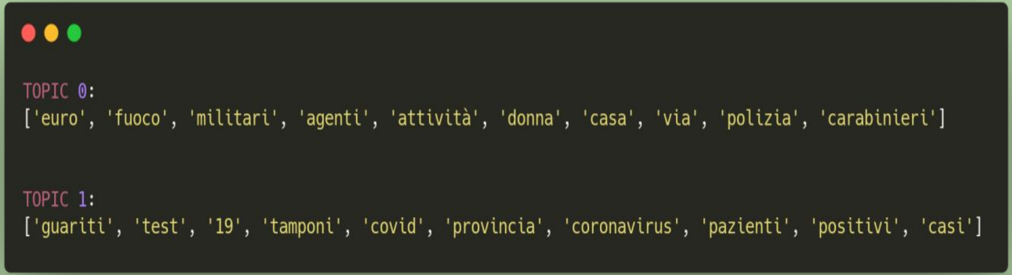
In combination con **TF-IDF scheme** on content data.



```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf_vect = TfidfVectorizer(max_df=0.8, min_df=2, stop_words= stop_words_italian)
doc_term_matrix = tfidf_vect.fit_transform((articles.title + articles.content).values)
```

C. Non-negative matrix factorization

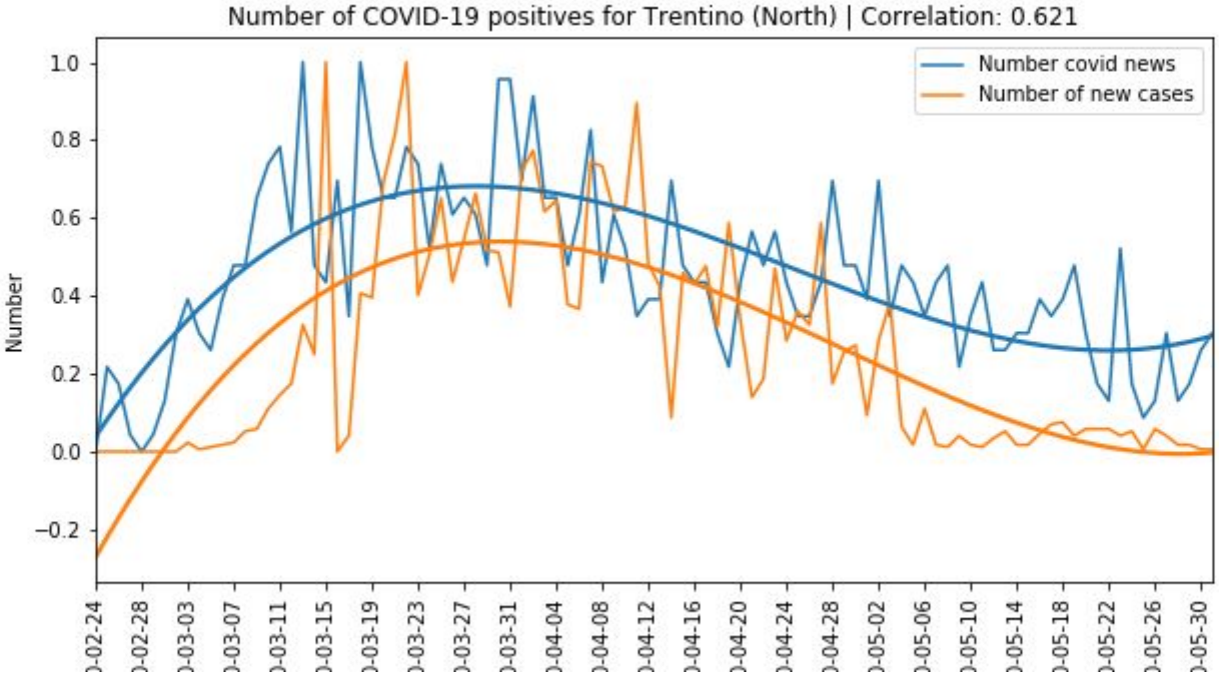
We got very distinct cluster



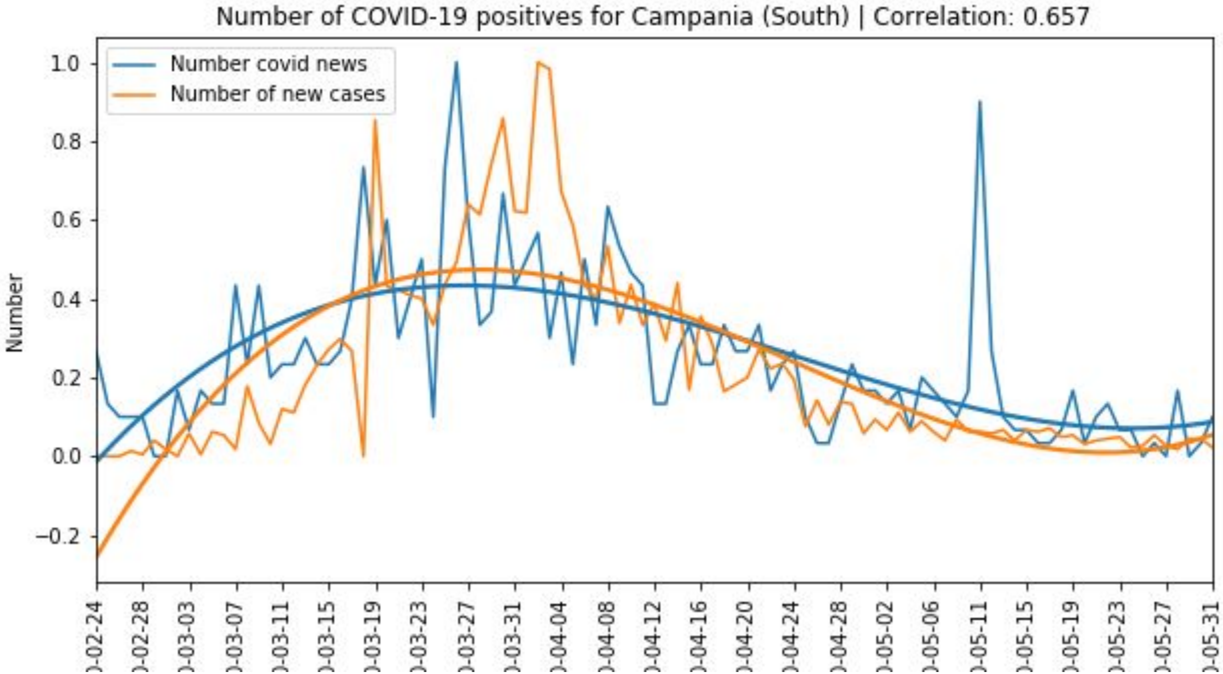
```
TOPIC 0:
['euro', 'fuoco', 'militari', 'agenti', 'attività', 'donna', 'casa', 'via', 'polizia', 'carabinieri']


TOPIC 1:
['guariti', 'test', '19', 'tamponi', 'covid', 'provincia', 'coronavirus', 'pazienti', 'positivi', 'casi']
```

We got two clusters of keywords that were deeply different in respect to each other.

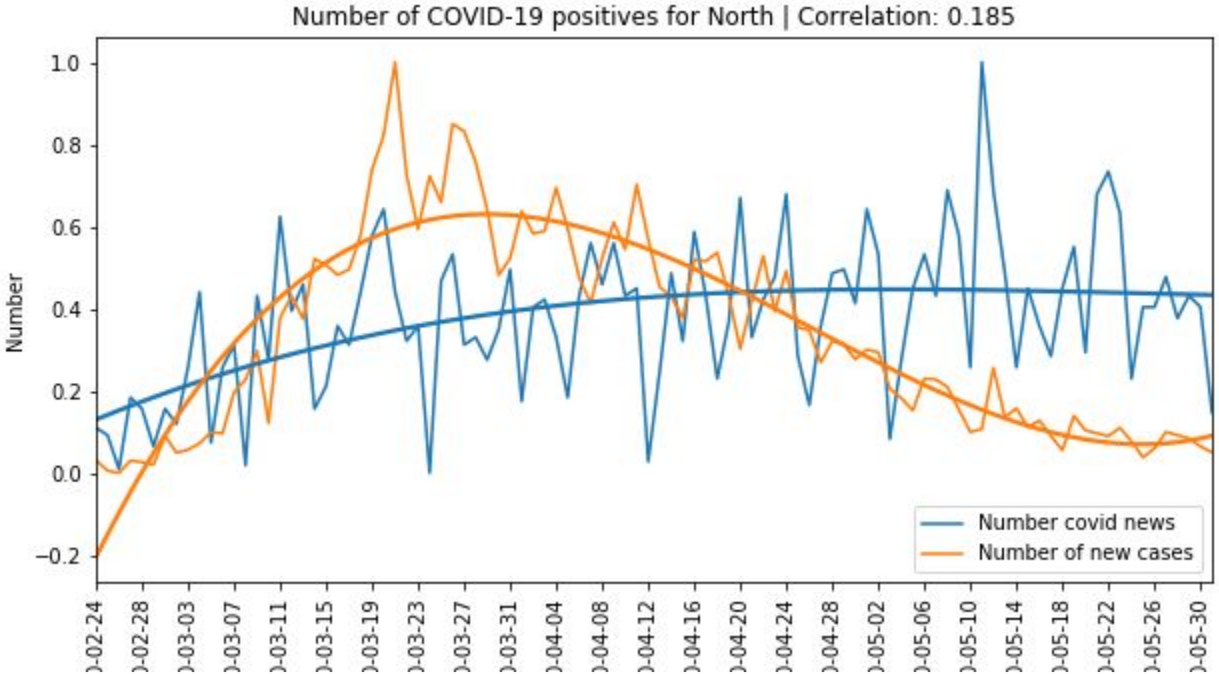
Region and Zones plots



Region and Zones plots



North of Italy...



C. Adding new features to the data, and creation of predictive model

Objective:

Predicting national data from articles

Raw text (‘title’, ‘content’ columns) by itself was not suitable[3]...

C. Adding new features to the data, and creation of predictive model

We selected the most important terms that we obtained from the **NMF model**…

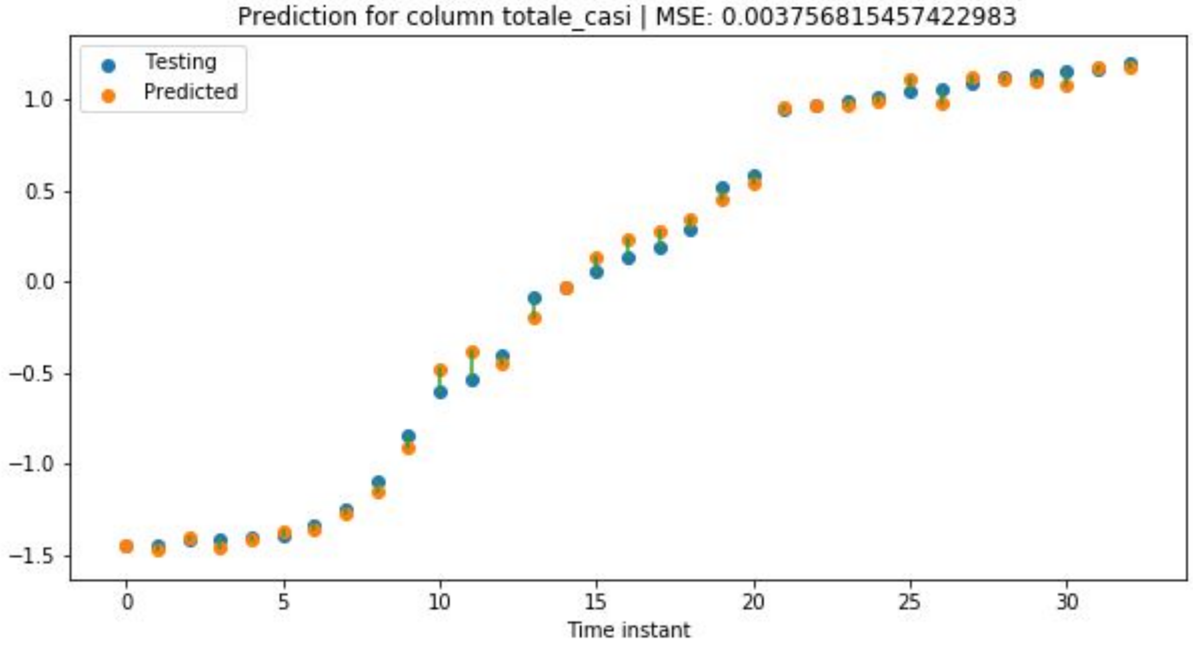
Using those **keywords** we did One hot encoding on the content of each article…

Grouping up by date, we got the total number of daily news where each term appeared.

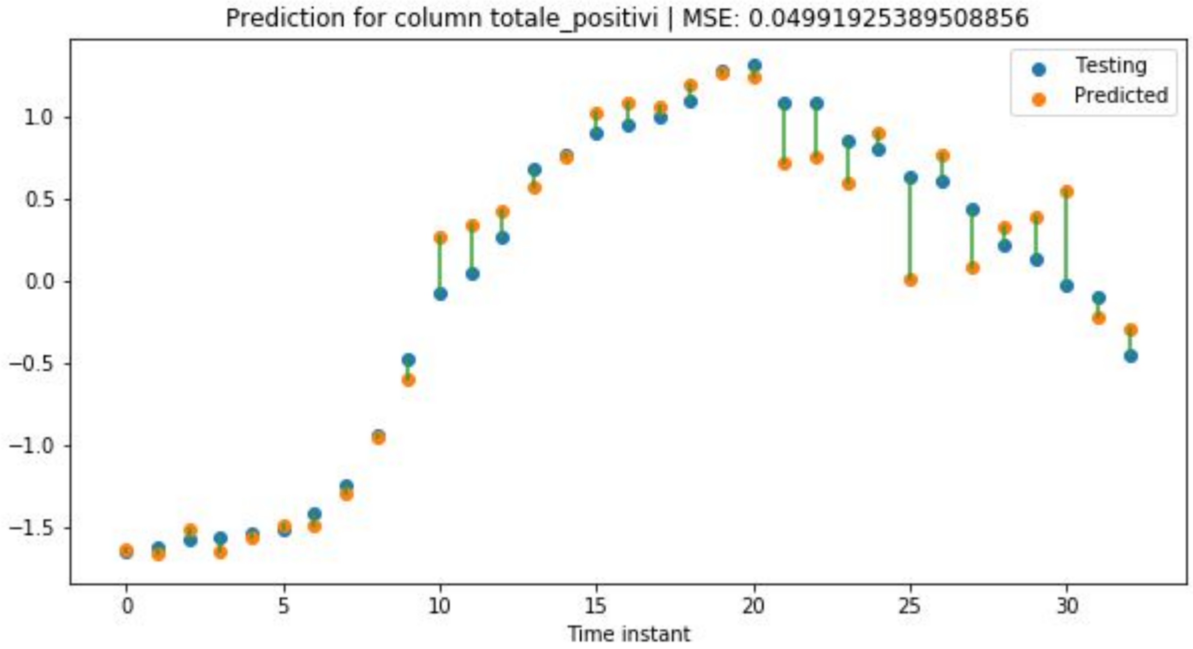
C. Adding new features to the data, and creation of predictive model

We selected a **Random Forest model** to try to predict the ‘nuovi_positivi’ column from the **Protezione Civile data**…

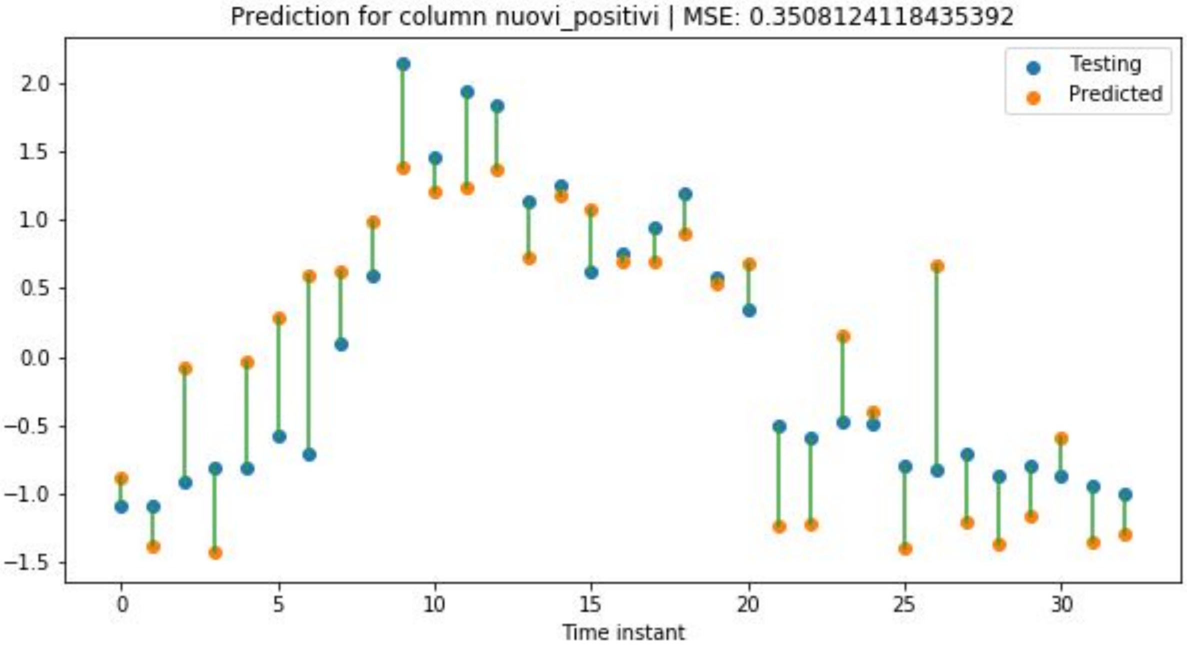
‘Totale_casi’


Prediction for column totale_casi | MSE: 0.003756815457422983
● Testing
● Predicted
1.0
0.5
0.0
−0.5
−1.0
−1.5
0
5
10
15
20
25
30
Time instant

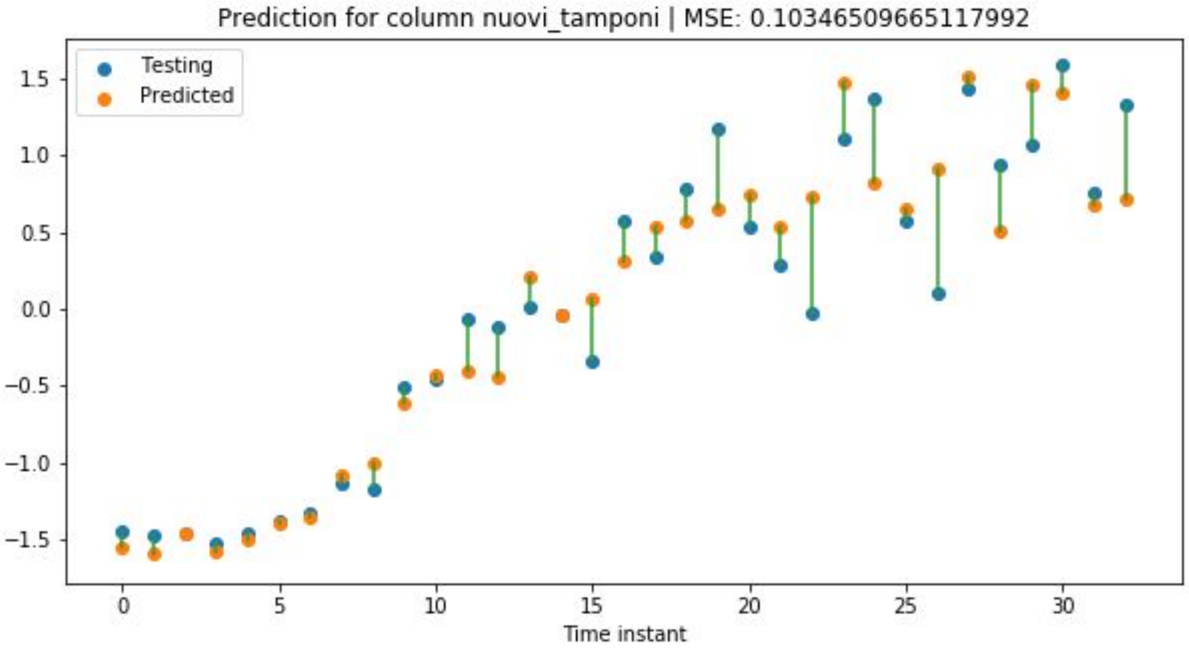
‘Totale_positivi’



‘Nuovi_positivi’



‘Nuovi_tamponi’



References

[1] - Berry, Gillis and Glineur “Document Classification Using Nonnegative Matrix Factorization and Underapproximation” 2009

[2] - Okun “Non-negative matrix factorization and classifiers: experimental study” 2008

[3] - Caragea, Cornelia, Jian Wu, Kyle Williams, Das, Khabsa, Teregowda and Giles. “Automatic Identification of Research Articles from Crawled Documents.” (2014).

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.