

POLITECNICO DI TORINO

Master's Degree in Data Science and Engineering



Master's Degree Thesis

Data-driven Analysis of Interactions and Popularity Increase in Online Social Networks

Supervisor

Prof. Luca VASSIO

Candidate

Matteo VILLOSIO

Co-Supervisors

Prof. Martino TREVISAN

Prof. Francesco VACCARINO

December 2021

Abstract

The rise of Online Social Networks in the last decade has changed society by irreversibly mingling the bounds between tangible reality and the online and, consequently, has shifted the paradigms of popularity relationships between famous entities and the public.

It has been observed that Online Social Network (OSN) content popularity can be forecasted thanks to prediction algorithms applied to early metrics and that, more in general, predicting metrics can be both endogenous and exogenous to the OSN.

This thesis introduces a novel approach to popularity forecasting based on historical information as well as attributes in control of the content creator instead of early popularity metrics and content quality attributes.

We utilise data gathered between 2015 and 2021 about 1611 Instagram Italian influencer profiles from the Crowdtangle database, a public insights tool from Facebook that allows to follow, analyze, social media public content; such dataset comprises 2 036 966 posts, each characterised by the attributes generated by the users and metrics regarding its popularity.

This dissertation proposes two algorithms, a Random Forest Regressor and a Recurrent Neural Network, implemented in several variations and some evaluation metrics with the goal of generating meaningful predictions about the number of reactions to a post without being subject to the extreme variance of metrics and the high number of outliers.

The findings appear to indicate that a limit in the information contained in the data does not permit us to perform exact forecasts. Nonetheless, we are able to reach satisfactory results that usually predict future trends in the popularity of the influencer.

*Come 'l viso mi scese in lor più basso,
mirabilmente apparve esser travolto
ciascun tra 'l mento e 'l principio del casso;*

*ché da le reni era tornato 'l volto,
e in dietro venir li convenia,
perché 'l veder dinanzi era lor tolto.*

DANTE ALIGHIERI
DIVINA COMMEDIA - INFERNO
CANTO XX

Table of Contents

List of Tables	VIII
List of Figures	IX
Acronyms	XIII
1 Introduction	1
1.1 Motivation	1
1.2 What are Online Social Networks	2
1.2.1 Brief History of Online Social Networks	2
1.2.2 The OSN Communication Paradigm	3
1.2.3 Instagram Social Network	4
1.2.4 Influencers: the new VIPs	4
1.3 The Online Social Network Forecasting Problem	6
1.4 Existing literature	8
1.4.1 Social media Forecasting	8
OSN-F Endogenous-Input Endogenous-Output Predictions .	8
OSN-F Endogenous-Input Exogenous-Output Predictions . .	11
1.4.2 Previous works from the research group	14
1.4.3 Our Contribution	18
2 The Instagram Posts Dataset	19
2.1 Platform choice	19
2.2 Characterisation of the dataset	20
2.3 Statistical Analysis and Cleaning	20
2.3.1 Follower characterisation	21
2.3.2 Interactions and Engagement Rate characterisation	21
2.3.3 Textual Characterisation	24
2.3.4 Time and Frequency Characterization	25
2.3.5 Other metrics	26

3	Relevant Theory	28
3.1	Classical Regression Methods	28
3.1.1	Linear Regressor	28
3.1.2	Random Forest Regressor	29
3.1.3	Gradient Boosting Regressor	31
3.2	Neural Network	32
3.2.1	Dense Layer	33
3.2.2	LSTM Layer	34
3.2.3	GRU Layer	35
3.2.4	Bidirectional Layer	36
3.3	Outlier Detection Methods	37
3.3.1	Isolation Forest	37
3.3.2	Z-score Outlier detection	38
4	Data Mining, Transformation and Loading	39
4.1	Insights	39
4.2	Data Mining	39
4.3	The Instruments: Pyspark, HDFS and the Cluster	40
4.4	Data Transformation	43
4.4.1	Identification Characteristics	44
4.4.2	Temporal Characteristics	46
	Absolute dimensions	46
	Periodic dimensions	46
4.4.3	Volumetric Characteristics	47
	Average Reduction	48
	Series Reduction	48
4.4.4	Point-wise Characteristics	50
	Textual Information	50
	Media Information	50
	Popularity Information	51
4.4.5	Graph Topological Characteristics	52
4.5	Data Loading	52
5	Training and Results	55
5.1	Evaluation Metrics	55
5.2	The Baseline	56
5.2.1	Hyperparameters, Architecture and Training	56
5.2.2	Results	56
5.3	Classical Regression Methods	56
5.3.1	Hyperparameters, Architecture and Training	57
5.3.2	Results	58

5.4	Neural Network Regression	61
5.4.1	Hyperparameters, Architecture and training	61
5.4.2	Results	66
6	Conclusions	70
6.1	Future Work	71
A	Complete List of Hyperparamters	72
A.1	Random Forest Regressor	72
A.2	Initial Dual-Legged Neural Network	72
A.3	Bidirectional Variants	73
A.4	Histograms of Text Metrics	73
	Bibliography	78

List of Tables

2.1	Division in quartiles of the followers	21
4.1	Attributes of the datapoints contained in the dataset	44
4.2	Example of account entries, url not visible for pagination constraints	45
5.1	Performances of on the typed dataset	66
5.2	Performances of on the Complete dataset	67
5.3	Performances of on the quantiled dataset	68

List of Figures

1.1	Number of Users per OSN - 2004 to 2019 [3]	2
1.2	Instagram profile of Bill Gates, in the upper part, the number of followers and followees and a short description, below, the content created and shared by Bill Gates Account	5
1.3	stories on Instagram	6
1.4	Instagram profile of Chiara Ferragni, famous Italian influencer	6
1.5	Plots of correctness score, Taken from [11]	9
1.6	Volume of hostile comments over time observed, Taken from [14]	11
1.7	Conceptual pipeline, Taken from [16]	11
1.8	Model Agreement on M5S supporters according, Taken from [19]	13
1.9	Regions generated by hierarchical GeoSOM clusters onto Boston, Taken from [20]	14
1.10	COVID-19 outbreak in Italy in the first six months timeline, Taken from [28]	17
1.11	Fractional average post evolution with respect to interactions over 3 days, Taken from [29]	18
2.1	ECDF of the subscribers	22
2.2	Histograms of reactions and comments	22
2.3	ECDF of reactions and comments with respect to the posts	23
2.4	Histogram of the engagement rate of the posts	23
2.5	histograms of the description metrics	24
2.6	Histograms of the frequencies	25
2.7	Distribution of the post throughout the time period	26
3.1	Example of Linear Regression. Taken from [34]	29
3.2	Example of Random Forest. Taken from [36]	30
3.3	Example of Bagging. Taken from [37]	30
3.4	Example of the gradient descent procedure in a 2D space. Taken from [39]	31
3.5	schematisation of GBR. Taken from [40]	32

3.6	Artificial Neuron and Natural Neuron. Courtesy of Giulia Marchisio	32
3.7	Some of the most common activation functions. Taken from [43]	33
3.8	Comparison between the two classes of NN. Taken from [44]	33
3.9	Dense layer connections. Taken from [45]	34
3.10	Structure of an LSTM cell. Taken from [47]	35
3.11	Architecture comparison between LSTM and GRU. Taken from [48]	36
3.12	Differences in the general architecture of a unidirectional and bidirectional NN. Taken from [51]	36
3.13	Comparison of the number of cuts to isolate an inlier (<i>a</i>) and an outlier (<i>b</i>)	37
3.14	outlier detection given a normal distribution. Taken from [55]	38
4.1	Sources of Data	39
4.2	Organisation of a Spark Cluster, Taken from spark.apache.org	40
4.3	Components of Spark, Spark R not included. Taken from [57]	41
4.4	Libraries used	42
4.5	View of a spawned Jupyter server with an opened notebook	43
4.6	post identifier, composed by long int and account identifier	45
4.7	Example of how the time interval was divided	47
4.8	Averaging method, red points are posts on the timeline, intervals are weeks	48
4.9	Average correlation (and σ) between posting frequency change and increase in followers	49
4.10	Series method, red points are posts on the timeline, intervals are weeks, the length of the window creating the series varies depending on the number of posts to include	49
4.11	regexes used to extract respectively hashtags, mentions and words	50
4.12	The classical method to split in train and test set the data vs the method used for this thesis	53
4.13	datapoints of the train set could contain information about the regression variable of the test set	54
5.1	The model was trained on the three datasets	57
5.2	Histogram of the prediction absolute error	59
5.3	Prediction line (orange) vs actual values (blue). The orange shadow is the expected range of error	60
5.4	Consecutive steps in the generation of the general architecture of the neural network	62
5.5	The final general architecture of the Neural Network	62
5.6	Structure of the first implementation of the actual neural network regressor	64

5.7	Pure LSTM Neural Network	65
5.8	Pure GRU Neural Network	65
5.9	Mixed LSTM GRU Neural Network	65
5.10	Prediction line (orange) vs actual values (blue). The orange shadow is the expected range of error	69
A.1	Bidirectional LSTM	74
A.2	Bidirectional LSTM	75
A.3	Bidirectional LSTM	76
A.4	histograms of the description metrics regarding posts	77

Acronyms

AI

artificial intelligence

OSN

Online Social Network

NLP

Natural Language Processing

NN

Neural Network

RNN

Recurrent Neural Network

MdAPE

Median Absolute Percentage Error

MAPE

Mean Absolute Percentage Error

R²

Coefficient of determination

Chapter 1

Introduction

1.1 Motivation

The tension to knowledge, to understand reality is what characterize humans and make them so different from anything else on earth. In particular, predicting the future, being able to anticipate fate has been considered of paramount importance from the most ancient times. A scientific approach to such a field began rising during the renaissance: Nicolaus Copernicus, with its *On the Revolutions of the Heavenly Spheres*, proposed a first model that, based on knowledge and observations, tried to predict the movement of planets. With the rise of nation-states, this approach to reality started being applied not only to nature but also to the economy and, as a consequence, society.

Online Social Networks have revolutionized the second decade of the XXI century and changed, probably irreversibly, our society: Online Social Networks and the "online" are now profoundly intermingled with the "real life", the "offline world". The mix of life and cyberspace mix creates a chimaera, the "onlife"[1], that cannot be considered orthogonal to any of the two dimensions. It is then manifest how important and impacting the ability to predict the future can be.

A data-driven approach to popularity prediction opens space to a multitude of practical applications. If the first thoughts fall on the usefulness that such a tool could have in the field of communication strategies and advertising, more various utilizations could be benefitted considerably as well. If it is considered natural, currently, to address popularity prediction with a trial-and-error methodology or utilizing polls, an approach based on data would permit to produce prediction with a higher granularity, precision and more fine-tuned for the particular need of the final user.

1.2 What are Online Social Networks

According to the Encyclopedia Britannica [2], a (online) social network is:

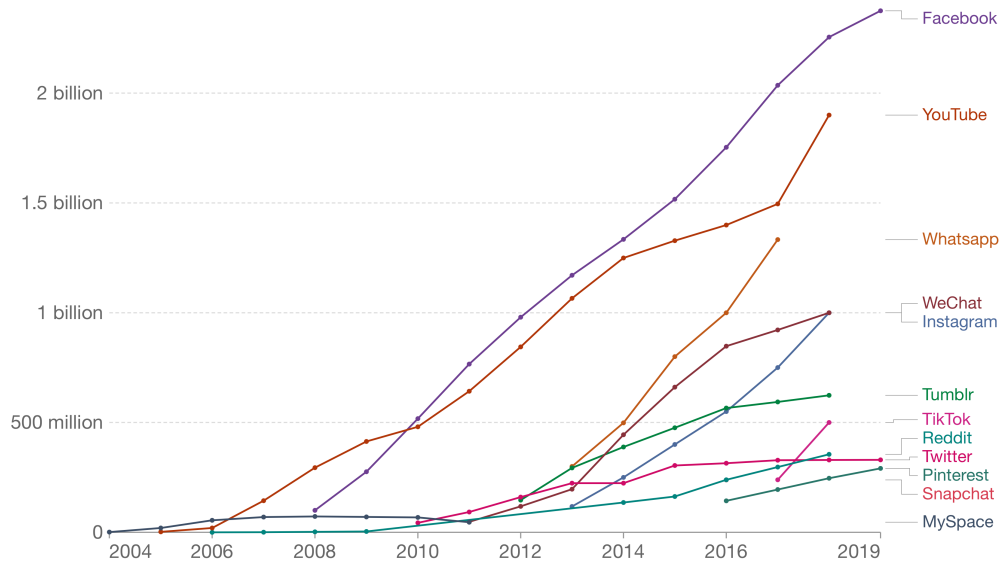
[...] an online community of individuals who exchange messages, share information, and, in some cases, cooperate on joint activities.

1.2.1 Brief History of Online Social Networks

The internet, the network that changed everything, first appeared with the name of ARPANET in the 60s for internal usage of the US military apparatus. If in the following decades it began to spread in various fields, first and foremost the academic one, the 90s can be considered the turning point for its rise to instrument of the general public. This change of paradigm was caused by the invention, by Sir Timothy John Berners-Lee, a CERN researcher, of the World Wide Web: a network of resources, usually websites, identified by URLs, connected to each other by means of hyperlinks, that could be accessed by anyone with a phone connection and a web browser. With the transfer to the WWW of many activities that were at first performed only in person, a new word, previously used only in the technical field, began being used to indicate everything that happens on "the internet": online.

Number of people using social media platforms, 2004 to 2019

Estimates correspond to monthly active users (MAUs). Facebook, for example, measures MAUs as users that have logged in during the past 30 days. See source for more details.



Source: Statista and TNW (2019)

CC BY

Figure 1.1: Number of Users per OSN - 2004 to 2019 [3]

The continuous increase in importance of the "online" brought to the creation of countless websites with the most various contents: from newspaper to normal people began creating their "webspace". Websites, even more now than in the past, can be subdivided in categories depending on their content, if some of them are static, that means, they show the same content to anyone connecting to them, many others can be considered dynamic: their content, in fact, changes dynamically depending on who is connecting, an example of them could be an internet banking website. Users, on the web, are not only consumers but also producers: websites with user generated content, such as wikipedia, are now among the most connected ones: there, users not only consume information but produce it as well.

If the forefathers of Online Social Networks, the email system, IRC and later the forums, rose almost with the world wide web, what can be considered their fathers began to appear in the late 90s and early 2000s with SixDegrees.com and Classmates.com [4], but only with Facebook and MySpace this new social paradigm became mainstream and known to the general population. In the following years, many OSN with different focuses and communication models began to rise on the internet; nonetheless, on the throne of western social networks, Facebook and Instagram remain kings[5].

Lately, new services, leveraging new communication paradigms, are beginning to penetrate this very monopolistic market, especially with particular demographics: the two most famous examples are the Chinese Tik-Tok¹ and the American Clubhouse².

1.2.2 The OSN Communication Paradigm

The great variety in social networks brought a relative diversity in their shape and praxes: nonetheless, it is possible to find some common characteristics to almost all of them[6].

They are:

- A public or partially public profile containing information about the user and eventually the created content.
- One or more lists of contacts sharing a connection with; the user's contacts are possibly organised in a single list where the relation between them and the user is bijective or in more than one list having injective relations. In the first situation, we can call those contacts friends, while in the second, they are followers or followees.

¹www.tiktok.com

²www.clubhouse.com

- A dashboard, a wall, containing all the content a user has access to, for example, friends' posts and shared publications.
- Content, shared with the friends, in the form of posts: we can define a post as a container for some information, which can be textual, media or a mix of both, to be shared, commented and liked by other users.
- The possibility to react to the content created by someone using one (for example, on Instagram) or more predefined reactions (like on LinkedIn) used to communicate the user's mood.
- The possibility to share other's posts, possibly with some additional personal content, on the user's profile.
- The possibility to comment is to add some text or media below the content created by someone.

1.2.3 Instagram Social Network

The Online Social Network of choice for our research was Instagram. Instagram was born in 2010 in the California as a transformation of a previous Check-in Social Network similar to FourSquare, becoming quickly one of the most downloaded mobile apps[7]. The said service, now part of the Facebook Group, is focused on sharing media such as photos and videos with an audience that can be private or public.

The published content appears in the feed of the subscribers, called followers, of an influencer or can be reached by exploring the social network using hashtags, words introduced by a hash that categorize the content they have coupled. Instagram introduced, moreover, one feature now familiar to many social networks: the "stories"; stories are short videos that can be shared for a limited timeframe by users and are shown on the followers' dashboard, sequentially.

Unfortunately, even if widely adopted by many ordinary users and influencers, it was not possible to study such communication method in this research due to the lack of data by the available dataset.

1.2.4 Influencers: the new VIPs

The word influencer is a neologism used to designate an internet celebrity, that is, someone that became famous due to their activity on the internet or that bases their influence on their internet presence.

The progenitors of modern days influencers' rise are identifiable in the user-base of the first online forums and chatrooms, but only with the emergence of

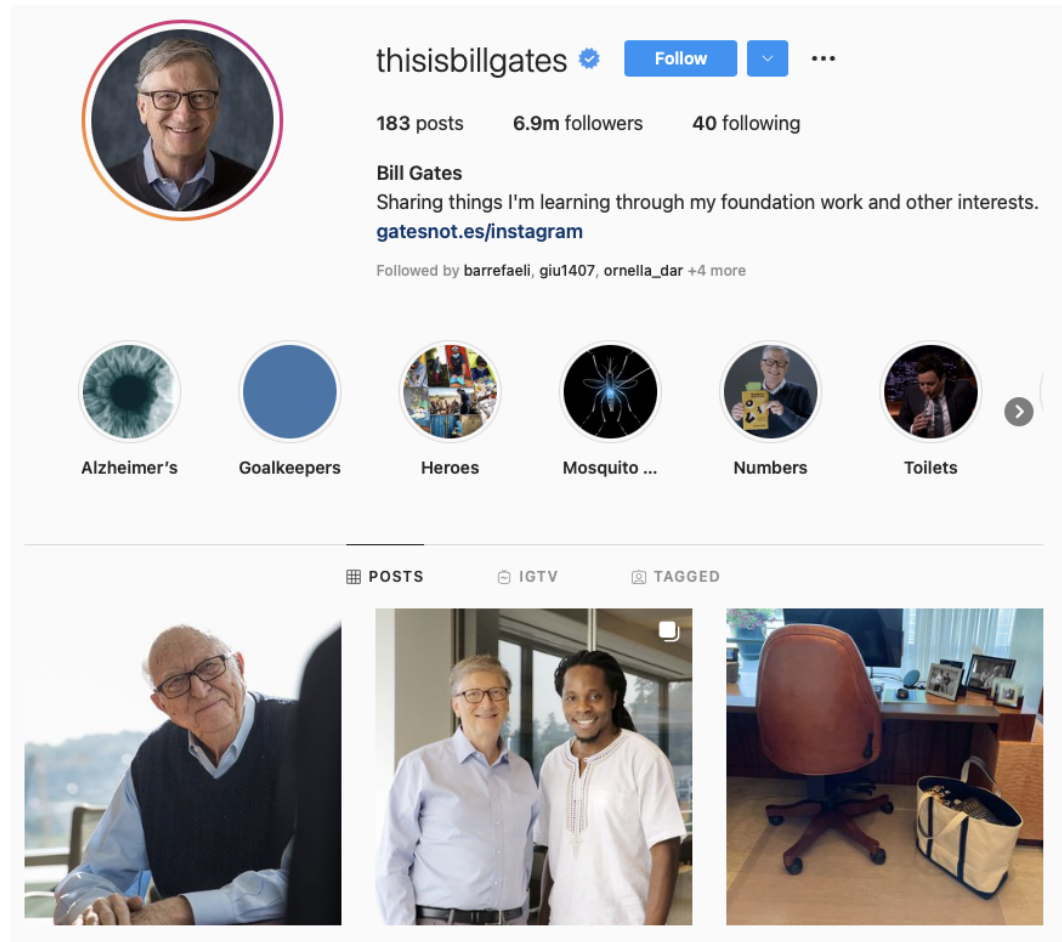


Figure 1.2: Instagram profile of Bill Gates, in the upper part, the number of followers and followees and a short description, below, the content created and shared by Bill Gates Account

contemporary OSN, such figures started to become more defined and "influencing": in the past, for example in chatrooms and forums, the concept of "following" was not present and, as a consequence, the effective "influencing power" of an internet personality was highly reduced.

There are many ways to categorize them, such as their rise to fame, their field of entertainment or their audience size; the latter is usually the most used[8][9] thanks to the easiness of application, and it can be organized as follows[8]:

- Nano influencers: less than 10 000 followers.
- Micro influencers: between 10 000 and 100 000 followers.
- Macro influencers: between 100 000 and 1 000 000 followers.

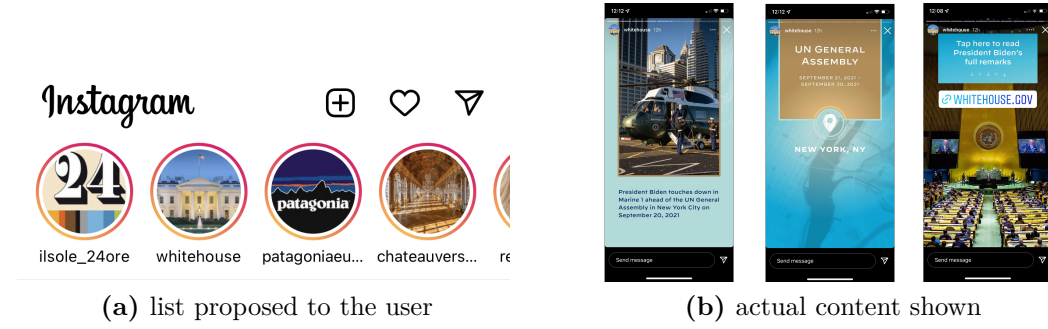


Figure 1.3: stories on Instagram

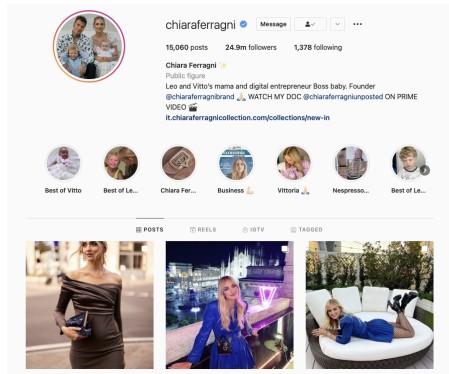


Figure 1.4: Instagram profile of Chiara Ferragni, famous Italian influencer

- Mega influencers: over 1 000 000 followers.

Some examples of Influencers are **Chiara Ferragni**³ or **followtiffsjourney**⁴.

1.3 The Online Social Network Forecasting Problem

Seeing the future has always been of great interest for humans; from the dawn of time, we tried forecasting the outcome of the most various occurrences. It is then natural that such a deep interest has been applied to the OSN.

Social networks are now ubiquitous in our personal, professional and economic lives. Consequently, the possibility of gathering enormous amounts of information

³<https://www.instagram.com/chiara ferragni/>

⁴<https://www.instagram.com/followtiffsjourney/>

on virtually all planes of life and using them to perform predictions has become a task of extreme interest for various players: researchers, governments, and companies are interested in finding the best way to leverage OSN information for their benefit.

Online Social Networks and, more in general, the internet, the "online" are now profoundly intermingled with the "real life", the "offline world": this mix creates a chimaera, the "onlife"[1], that cannot be considered orthogonal to any of the two dimensions and, therefore, any analysis performed on OSN is dependent on both endogenous and exogenous characteristics.

Online Social Network forecasting, from now on OSN-F, then, is the procedure of performing a prediction concerning OSN in any of its steps; we can hence classify OSN-Forecasting using four, not mutually exclusive, categories concerning its relation with the online and the offline:

- OSN-F Endogenous-Input Prediction: in this case, the forecast utilises information that is generated by the socials and is not, at least directly, caused by external events; an example of a pure endogenous prediction is one that uses only the number of words in a post to predict a particular attribute.
- OSN-F Exogenous-Input Prediction: here, the prediction is based on information external to the OSN; an example could be predicting an influencer's popularity based on its participation in TV events.
- OSN-F Endogenous-Output Prediction: This prediction aims to forecast something that concerns the social network, such as the number of likes a post will get.
- OSN-F Exogenous-Output Prediction: This prediction aims to forecast something external to the social network; an example could be to prognosticate the outcome of a rally based on the reception on Twitter.

It is clear that for the problem to be an OSN-F, at least one between the input and the output must be endogenous.

Online Social Network forecasting, which has been widely researched and we will mention in section 1.4.1, has rarely seen Instagram influencers' content popularity as the focus of the analysis.

In this thesis work, we focused on an OSN-F Endogenous-Input Endogenous-Output Prediction problem; in particular, we developed a new approach to the popularity forecasting problem: we decided to base our forecasting on the lack of knowledge about early performances and the quality of the content while focusing on the historical information and characteristics in control of the user at the moment of the posting. To do so, a regression predictor is implemented utilising two methods: a classical regression method, Random Forest Regressor, and a

more modern one, a Recurrent Neural Network. Such a strategy intends to offer a strategy to be utilised before publishing the content to predict its performances and make informed decisions about its production.

1.4 Existing literature

1.4.1 Social media Forecasting

Research in the field of Social media Forecasting can be subdivided in three macrogroups according to the definitions of section 1.3:

- OSN-F Endogenous-Input Endogenous-Output Predictions
- OSN-F Endogenous-Input Exogenous-Output Predictions
- OSN-F Exogenous-Input Endogenous-Output Predictions

OSN-F Endogenous-Input Endogenous-Output Predictions

Popularity is one of the core metrics of success on OSN, and, as a consequence, it is the focus of many pieces of research.

The problem of predicting its evolution is interestingly investigated by Ahmed et al.[10]: the authors identify temporal general evolution patterns and use those to reach two goals:

- classify content using its own popularity evolution
- predict future popularity content

To do so, data from Youtube, Digg and Vimeo are used. The problem is defined as follows: given some content N , O observations made on it and a time period T the aim is to extract features that identify the growth in popularity relative to other observed content in T and to use the said extracted features to generate a dictionary of behaviours to group N in subsets. In the paper, after defining the feature space, a clustering method and a prediction algorithm are run on the data giving some promising results: concerning the clustering, the authors were able to decrease of 75% the MSE error in the subdivisions in behaviour classes with respect to previous works; Concerning the regression, using the knowledge generated by clustering, the developed algorithm reached a significantly lower error than the baseline.

Another study that addresses the problem of popularity analysis is the one by Hu et al.[11]; in their work the authors, instead of trying to predict the popularity value, investigate popularity by subdividing its evolution in three key moments: "burst",

"peak", and "fade". Then, they attempt to predict when popularity experiences those pivotal events. After finding a framework to recognise such events, classifying (six) different popularity evolutions and identifying the importance of promptness with respect to accuracy a Support Vector Regression was employed to predict the occurrence of the "popularity evolution key moments". To better address the problem of having a prompt and accurate prediction, a new metric, called CPScore, is defined. The found solution not only seems to outperform others but also recognises that more than half of the hashtags used have a sudden burst followed by a quick peak and a fast fade.

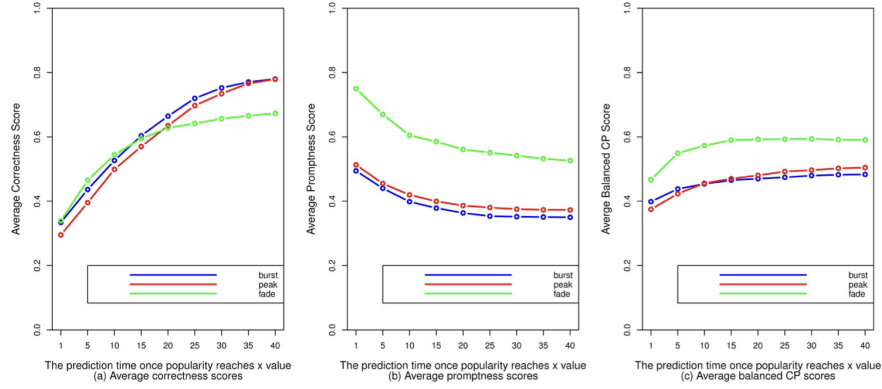


Figure 1.5: Plots of correctness score, Taken from [11]

In their research, a slightly different approach is taken by Yu et al.[12] The authors try to predict when popularity reaches its peak without focusing on other events. In the paper, the writers focus on the popularity of Twitter hashtags from a Social Network application point of view to introduce three research aspects: first, when popularity reaches its peak, then, the paper discusses how to identify when to trigger a popularity prediction, finally, a Deep Learning model is designed to perform the needed predictions. To achieve the forecasting, three different typologies of data are used: topological network information, social information, and Hashtag strings. It is discovered that the vast majority of the hashtags reaches their popularity peak in more or less two days and many in only 10 hours; this translates into the fact that only a few have their peak during a later stage of their evolution, numerous before the half. These pieces of information and the architecture chosen for the Deep Learning model bring results that outperform the baseline.

A somewhat different popularity typology is the one related to news: in this domain, the main concern is to reach the highest number of readers and trigger a viral expansion. The work from Bandari et Al[13]. distinguishes itself from the others on this domain due to the fact that it proposes to avoid basing its prediction

on early popularity to develop an adequate algorithm to be used for decision-making strategies. A feature space is generated from the characteristics of the article, of the source and from historical data; then both regression, such as linear and SVR, and classification algorithms, SVM, Naive Bayes, DT and Bagging, are applied to perform a prediction. The outcomes show that it is not possible to produce exact predictions of the number of tweets an article will collect: the regression results were, in fact, quite lacking; nonetheless, it is possible to provide effective ranges of popularity, with a precision of 84%, by using classification methods. Furthermore, some insights were gathered: only a fraction of articles are able to reach a wide public, the majority reaches a medium audience that should be targeted to be of highly interested readers; it can also be surprising to discover that top news sources on Twitter are not inevitably the traditional popular news agencies. Finally, another interesting result of the paper was the finding that one of the most relevant predictors of popularity was the publisher of the article.

The work from Guberman et Al[14]. addresses, on the other hand, a different type of problem: OSN are blighted by antisocial behaviours such as cyberbullying, harassment and trolling; those feuds damage free discussion and produce countless difficulties to netizens. While numerous solutions have been studied to recognise those behaviours after they happen, almost no research has been completed regarding methods to anticipate the insurgence and the force of hostility. The paper considers the problem of forecasting future hatred by subdividing it into two steps: first, given a sequence of non-hostile comments, a method to prognosticate if hostility will arise in the future is developed; then, if a first hostile comment is recognised, another algorithm is used to predict if such this will cause a growth in hostility. The two tasks are performed using linguistic content of the comments; the features are extracted using Unigram and n-grams, Word2vec, ProfaneLexicon, a list of profane words, the content of the most recent comment, the comment history of users, the post history of the poster and user activity. The developed model reaches an AUC of 0.82 to forecast the insurgence of hostility in the next 10 hours and 0.92 into identifying the magnitude (number of hateful comments) of such hostility.

The 2016 elections in the US and, more in general, almost all western political events in the last years were deeply influenced by online activities [15]: a multitude of unfortunate events have made clear that organised malicious behaviour has severe real-world effects. The paper by Weber et al.[16] observes coordination tactics such as pollution, boosting, bullying, and metadata shuffling and attempts to find a method to analyse and detect those tactics and infer the hidden communities that use them. The researchers proceeded with an approach based on temporal windows of varying length, user interactions and metadata to detect accounts engaging in operations that, in coordination, execute goal-based strategies. The analysis was performed on two datasets crawled from Twitter: one generated by the IRA and

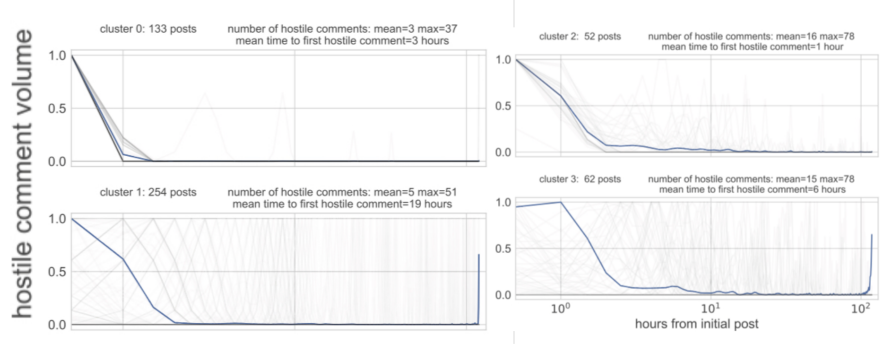


Figure 1.6: Volume of hostile comments over time observed, Taken from [14]

based on October 2018 general activities, while the other sampled during the 2018 Regional Australian Elections. The paper seems to confirm the employment of the strategies mentioned above in organised operations and opens space to explore real-time applications to recognise, and eventually intervene, those schemes.

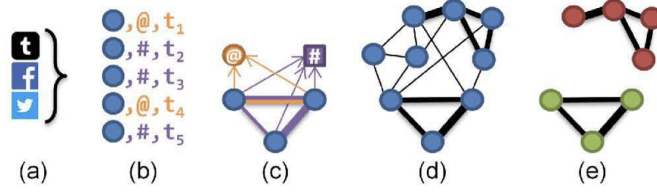


Figure 1.7: Conceptual pipeline, Taken from [16]

OSN-F Endogenous-Input Exogenous-Output Predictions

The usage of Online Social Networks to model and study real-world events raises the problem of the reliability of data collection: not only data must be correct, but they should also be complete enough to construct meaningful and unbiased networks. The work from Weber et Al.[17] addresses this problem by applying a systematic comparison approach: two parallel datasets were concurrently collected from Twitter concerning some particular events using different methodologies and tools with the aim of identifying how alterations in data acquisitions influence the outcomes of social network analyses. To do so, the authors examined the two datasets, one generated with the Twarc Library while the other using the RAPID platform, analysing Dataset statistics such as count of tweets, Network statistics like component diameter, centrality values and cluster comparison. The outcomes of the investigation are not surprising; while the most important content values

remain comparable, other metrics may vary broadly: different numbers of sampled accounts caused a different number of nodes, extra tweets created extra edges and, more in general, network structure appeared to be different.

In the paper by Walt et Al.[18], researchers used data from Facebook to predict users personality traits, from the Big Five Personality Model (Agreeableness, Conscientiousness, Extroversion, Neuroticism, and Openness), using demographic and text-based attributes generated using public information from their profile. To proceed with the classification 111 dimensions were produced: 31 from demographic information harvested from the user-profile and 80 text-based features mined from posts and photo descriptions using a crawler and NLP techniques. The datapoints described in such a way were then fed to three different predictors: Linear Regression, REPTree and Decision Tables; the predictions were then used to rank people according to their traits: while accuracy varied depending on the personality trait used to rank the individuals, for some characteristics, such as openness, the researchers managed to achieve 75% of accuracy in finding the top 10% most characterised personalities from that trait. These results show that automatic analysis can identify people with specific characteristics and, eventually, use them for the most varied goals: from targetted advertising to social engineering attacks.

Given the possibility of performing analyses such as the ones of the previous article, it is of great interest to try leveraging new machine learning methods and OSN data to perform predictions about users' political orientation. In this direction goes the work from Cardaioli et Al.[19]: using Twitter, a dataset of more than six thousand users and almost ten million tweets were generated and labelled manually by a pool of humans as part of six categories ranging from extreme right to extreme left; then, the authors trained the classification algorithm on the profiles that were identified as supporters of ideologically well-defined parties. Finally, the algorithm was used to predict the orientation of people labelled as "Movimento 5 Stelle" fans, an Italian party that can hardly be classified with traditional political categories. When predicting left-right membership, the authors were able to reach an accuracy of 93%. The outcome of the "Movimento 5 Stelle" supporters labelling was even more interesting: using an ensemble of classical machine learning methods (SVM, Linear Regression, SGD, Random Forest, XGB) those voters were subdivided in political leaning as in figure 1.8; the subdivision was highly adherent to the political analyses made public at that time.

On a different subject, The paper from Psyllidis et Al.[20] takes a geosocial approach to Social Network predictions: in their work, in fact, the researchers proceeded with proposing a framework to identify homogenous regions of social interaction and, eventually, predict appropriate locations of new POIs. The developed method considers nine dimensions generated from Twitter and Foursquare; those attributes are subdivided into four categories: spatial (longitude and latitude),

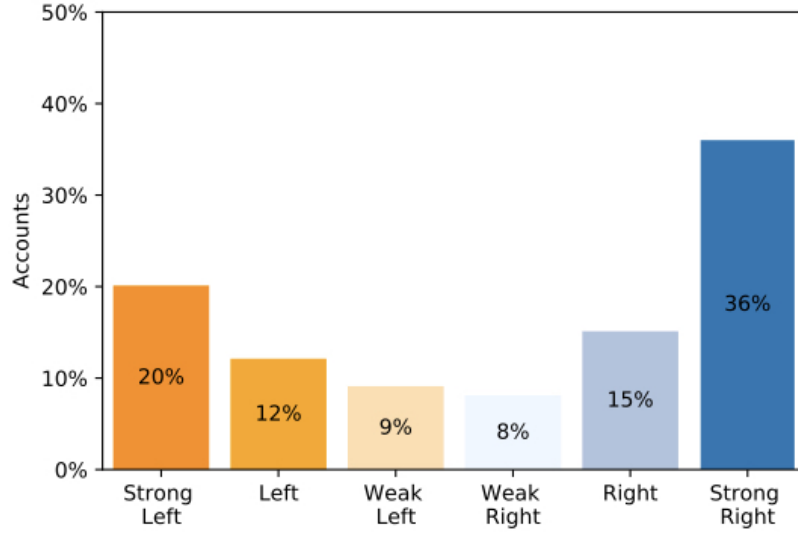


Figure 1.8: Model Agreement on M5S supporters according, Taken from [19]

temporal, semantic and sociodemographic. The framework uses a combination of Geo-Self-Organising Maps, clustering and classical machine learning supervised methods attempting to overcome the difficulty in elaborating high-dimensional noisy user-generated unstructured data. At first sight, the results seem to be quite unpromising: with F-measures fluctuating in the range of 0.01-0.1, one could think that there is no improvement with respect to the baseline; this is not the case, it must be kept in mind, actually, that those performances are expected in recommendation systems.

A research on a similar topic is the one by Zhao et Al[21]. regarding the forecasting of spatiotemporal events in Social media. They worked to produce a model able to predict events in space and time using intelligence produced by Twitter. The study states that it is possible to forecast the insurgence of events such as flu or unrests with an acceptable spatial and temporal accuracy using dynamic programming and the information contained in Twitter posts. This is possible because the messages have the following characteristics:

- they are posted instantly from users
- the OSN is ubiquitous and widely used
- the messages posted have geoinformation embedded

The generative model developed from these assumptions not only outperforms the baseline, but it also reaches more than acceptable results.

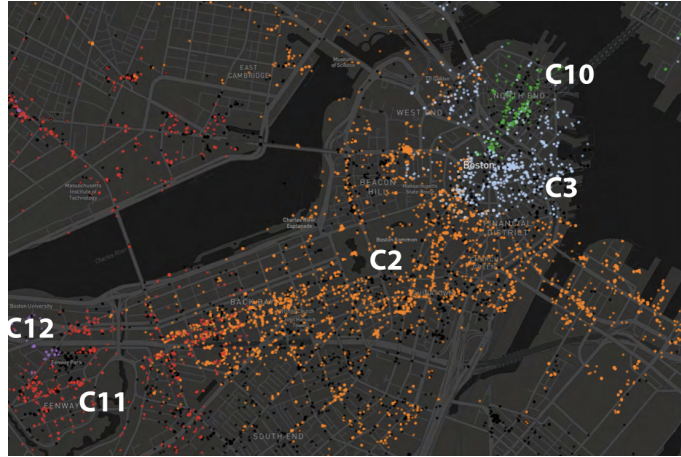


Figure 1.9: Regions generated by hierarchical GeoSOM clusters onto Boston, Taken from [20]

A paper that, on the other hand, tries to use popularity to predict external events is the one of Krauss et Al.[22]: in their investigation, using a mining approach to extract information from movie related forums, the researchers were able to find a correlation between social network structure and sentiment with box office revenue and Academy Awards Nominations. Three metrics are computed: Intensity Index: it is the frequency at which the subject, that is, the movie, is brought up in the discussion Positivity Index: it is the level of positivity shown about the movie displayed by users Trendsetter Index: it is a metric that weights the favour of users by their centrality in the network, in short, what the influencers think about the movie The first two metrics were used for the "Oscar Model", that is, the model used to predict Academy Awards nominations, while all three were used to predict the box office success. Both models seem to show promising results by correctly sorting the films both by sales and Oscars.

1.4.2 Previous works from the research group

The research for this thesis stems from the broader work on Online Social Networks done by the SmartData@Polito research group; this research group focuses on Big Data technologies, Data Science, and Machine Learning applied to the most varied topics and domains.

One initial work of Data Analytics applied to OSN is by Trevisan et Al.[23]: it is a preliminary study of interactions on Instagram. A custom crawler was used to mine Data to perform the analysis; the script downloaded and stored data and metadata regarding profiles and related activities. The focus of the study was on the differences in the way of interacting between political communities and general

ones: one of the most visible discrepancies is the one in the number of comments; when analysing political posts, a higher number of comments is seen, more than three times larger than for other topics. Comments are not only more numerous, but they are also lengthier and posted for longer times. The total number of comments is not the only distinction; politics shows a more significant amount of unsolicited replies: it seems, in fact, that users are not dragged into the discussion but reply autonomously after reading the previous interactions. It is then clear that political posting has very distinct characteristics from other categories and those distinctions are well defined qualitatively and quantitatively.

Another paper related to politics and OSN is the one by Ferreira et Al.[24]: in this paper, the researchers tried revealing the fundamental characteristics and dynamics of OSN interactions: to reach this goal, a probabilistic model was established to extract the backbones of the interaction networks, structures able to grasp interactions with evidence of coordination, among Instagram commenters and, using that, identify communities. The analysis was performed on a dataset crawled from Instagram and focused on a ten-week interval centred around political elections in Italy and Brazil; politicians and general influencers were both analysed. Such research was able to uncover some interesting observations: Commenter networks are split into few communities. The structure of those communities is weaker if related to politics, apparently due to the variety of positions on the political spectrum, with some users connecting conflicting political influencers. Despite their weaker and blurrier structure, political communities have more participants and are more active. Unsurprisingly, political communities have a boost in their characteristics during electoral periods. Moreover, a methodology to extract salient interactions, interactions of co-commenters that happen more frequently than expected, based on a comparison with a null model, was designed and used to generate the backbone networks mentioned earlier.

The natural continuation of previous research is the paper by Ferreira et Al.[25] Similarly to the other work, the analysis was performed on the dataset crawled from Instagram. The research group studied the evolution of communities of users who frequently interact by commenting on the same post and, as a result, could drive the online discussion. To do so, *salient-interactions*, defined in the previous paragraph, are taken into account because they are essential elements used to drive online discussion and information dissemination. This analysis confirms some insights from the previous papers[23][24], e.g. political communities are more engaged and write more, and discover new important information: while positive sentiment is usually prevalent, it tends to be more negative in political communities than in general users that seem to come from communities built around a particular politician will leave negative comments onto profiles associated with the opposite political side Stronger communities are built by commenters showing salient interactions; those communities seem not to have as their focus one influencer but a subset of posts

from one or more influencers.

The research group did not only focus on the mix between politics and OSN but also on more general topics such as advertising: in the paper by Vassio et Al.[26] the attention is directed to the commercial realm and on the result of user interaction and response to targeted advertising operations. After analysing the dynamics of such a world, a new metric is defined for specific campaigns: the new KPI is called Click-through-intensity and it is proportional to the profits of both the advertiser and publisher. It is proposed as an instrument to be used side by side with the click-through-rate, another metric usually used for measuring the performances of advertisement campaigns: while significant, the CTR does not explain the influence of impression frequency in the case of repeated actions on the same campaign. As a consequence, CTI is a better metric to be employed in the case of the advertising of services that aim to be accessed several times. Some further research on the topic is possible, for example, about using different methodologies to describe the activity pattern of users.

On the more general topic of popularity prediction, it is fascinating the work of Bertone et Al.[27]. This paper draws an interesting parallel between the OSN world and the stock market: Influencers are considered stocks while users are private investors; following an influencer is compared to buying a stock, a decision based on information from external resources and individual preferences. For this study, 60 Italian public figures were chosen and their data from Instagram and google trends were mined: the first source was considered the "stock market", while the second was used to mimic the "external resources" used by investors to estimate the value of a stock. A widely used finance tool, Bollinger Bands, used to generate a value interval in which a stock is considered to be "fairly priced", was applied to external resources trends to generate a price/follower estimate for the influencers/stocks and eventually recognise when an influencer was overpriced (and as a consequence would see in the future a decrease) or underpriced (and as a consequence a future increase). The study shows how this market-like approach successfully estimates short-term trends in OSN personalities' success and provides a strong correlation between different evolutions.

COVID19 profoundly changed society, and, as a consequence, it was of great importance to study its effect on Online Social Networks. In the study by Trevisan et Al.[28] the research was focused on how the total lockdown at the beginning of the year 2020 affected social life and, as a consequence, online social life. The analysis was performed on a dataset generated from the Instagram and Facebook posts of 639 influencers published during the first six months of 2020. Some interesting differences between the periods before, during and after the lockdown and between the two social networks are noticed: Facebook shows an increment in the number of posts, comments and likes during the lockdown weeks while, on the other hand, Instagram seem to show a flat trend (or a step decrease regarding the

reactions). Analysing the level of debate, it was observed that during the first weeks of lockdown a drastic decrease of replies in discussion for politicians profiles; at the same time, isolated comments increased. The prohibition on social activities during that period had effects on the hourly patterns as well: during Friday and Saturday afternoons, online activity increased up to 50% with respect to the situation before the restrictions. Finally, topics and psycholinguistic properties presented some changes: negative sentiment had a spike during the most challenging times and returned (almost) to normal levels after, topics on the other hand, viewed a shift to those more related to personal life during COVID19 emergency.

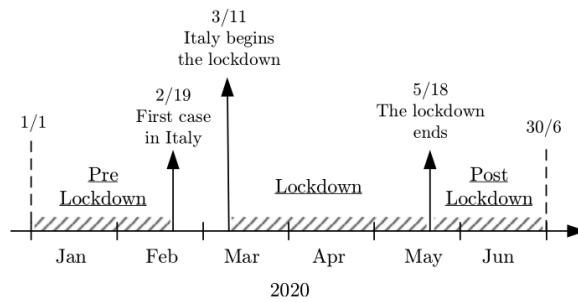


Figure 1.10: COVID-19 outbreak in Italy in the first six months timeline, Taken from[28]

On a track more similar to [27], the research group, in the paper from Vassio et Al.[29], focused on studying how the freshness of content plays a role with respect to content popularity evolution. The dataset generated using Crowdtangle API on a five-year interval contains 4 million posts and 13 billion interactions from Instagram and Facebook Italian influencers. Some interesting insights are generated: the PDF of influencer hourly activity shows that it is common for online activity to have two peaks during the day and a decrease during the night, with followers being more active in the late evening with respect to content creators; more in general, it is seen that followers activity can be considered as a log-normal distribution. Despite those similarities, the research shows that individual posts can have highly diverse patterns of interaction accumulation over time. Some differences are also seen between the two social networks: by considering the time at which a post has reached the 95th percentile of the total interactions, the median "lifetime" seems to be somewhat shorter on Facebook. Finally, the researchers observed that the creation of new content by the same influencer progressively fades away the attractiveness of the original post, presumably because users concentrate their attention at the top of the timeline.

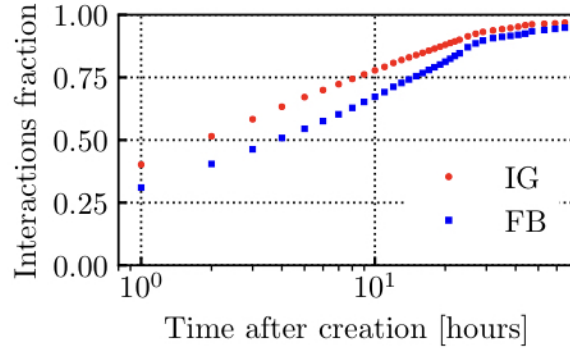


Figure 1.11: Fractional average post evolution with respect to interactions over 3 days, Taken from [29]

1.4.3 Our Contribution

The present thesis project places itself in the current literature as an Endogenous-Input Endogenous-Output Prediction problem: while such class of forecasting problem has been thoroughly explored, the field has been rarely approached with the intent of basing the prediction not on early performances or content quality but solely on historical information and the characteristics of the post that are controllable by the influencer. To do so, in chapter 1 we define the problem and create a classification framework for Online Social Network Forecasting Problems; in chapter 2 a description of the dataset and its characteristics provides an overview of the data used to train the model; in chapters 4 and 3 we illustrate both the transformation pipeline and the theory used to produce, from the raw datapoints, the information used for the prediction. Finally, the performances of the model are presented in chapter 5.

Chapter 2

The Instagram Posts Dataset

The need for a vast, various dataset for the research pushed us to look for a source able to respect three main properties:

- provide a considerable number of posts
- the sourcing of the post should come from a variegated amount of influencers
- the time range of such posts should be wide enough to contain a significant part of the history, and consequently the evolution, of the influencer

Despite the possibility to use previously crawled datasets from Facebook and Instagram, the choice fell on using Crowdtangle, a Facebook-owned tool that tracks interactions on public content from Facebook pages and groups, verified profiles, Instagram accounts, and subreddits. It does not include paid ads unless those ads began as organic, non-paid posts that were subsequently “boosted” using Facebook’s advertising tools. It also does not include activity on private accounts, or posts made visible only to specific groups of followers.

2.1 Platform choice

The decision to perform our search on Instagram was taken because of the characteristics of the said OSN: it is, in fact, a social network that is highly addressed to the creation of public content accessible by anyone, and as a consequence, easily accessible, the content creators are often real people and the “success metrics” are few and readily identifiable.

The most used ones are in fact, the number of "love reaction", the number of comments, the number of followers and, eventually, a mix of them, the engagement.

On facebook, on the other hand, the different weight given to the reaction, especially due to the fact that there are many of them that can veiculate different messages, and the lack of widely used unilateral interest action, such as the follow make more difficult to study the popularity.

2.2 Characterisation of the dataset

The dataset was fetched using the public Crowdtangle API and selecting the activities of the top 1611 Italian influencers, selected by the portal influenceritalia.it¹, in six years, from Jan 1, 2015, to Dec 31, 2020: the result of this selection was the generation of a dataset of 2 036 966 posts. The datapoints, that is, the posts, were characterized by a considerable number of parameters that could be divided into five categories:

- Account features: those characteristics are the ones that described the influencer account at the sampling time, of this type are: the account handle, the name, the URL, the IDs used by the platform and account metrics such as the number of followers.
- Advertising features: each datapoint was defined also concerning the eventual advertising and sponsorships; in particular, the Account features of the sponsor (if present) were reported into the record.
- User-generated post features: those were traits that the user defined at the moment of the content creation: those features were, for example, the creation time, the metadata of the media, the media type, the text and the description.
- Auto-generated post features: those characteristics are created automatically by the platform at the post creation, for example, the URLs and what the platform expects to be the outcome, in terms of popularity, of the post.
- Outcome features: those are the attributes that define the post’s success and that the user or the platform cannot control: the number of comments and reactions and the historical trend in those metrics for the current post.

2.3 Statistical Analysis and Cleaning

Given the subdivision, as mentioned earlier, it is possible to characterize the dataset from different points of view.

¹containing politicians, VIPs, institutions and football teams

2.3.1 Follower characterisation

One of the first analyses was to check the distribution of the followers at the moment of the sampling by Crowdtangle. A simple division in quartiles showed that a considerable part of the dataset was composed of posts whose creator followers count was equal to zero: sampling and checking their actual count, we noticed that this occurrence was due to an error, probably caused by the fact that the said influencers started being tracked after the "corrupted" activity and, as a consequence, some data were not recorded correctly. Due to the dimension of the dataset and the difficulties in restoring the correct metrics, we decided to purge out all those posts whose subscribersCount dimension was equal to zero. The purged dataset has been used for the following analysis.

Table 2.1: Division in quartiles of the followers

[Quartile]	[Dataset]	Purged	Complete
1st		143 - 255629	0 - 0
2nd		255629 - 523255.0	0 - 276415
3d		523255 - 1085122	276415 - 76788
4th		1085122 - 45926689	76788 - 45926689

The effect of such an operation is visible when performing a quartile analysis of the dataset.

The distribution, that due to the logarithmic axis appears to be Gaussian, can be approximated to a log-normal distribution with a peak of followers of half a million and the vast majority under one: a foreseeable occurrence since we can imagine that only a handful of influencers can reach an enormous audience.

2.3.2 Interactions and Engagement Rate characterisation

The following essential metrics to be analyzed were the distribution of the number of reactions, comments and post engagement rate.

To have a more precise visualization of those metrics, the posts having no comments or reactions had them mapped to a synthetic value of 0.1. Each datapoint was colour-coded based on the number of subscribers transformed in their quartile subdivision since comparing the metrics mentioned above for such a diverse range of influencers was pointless.

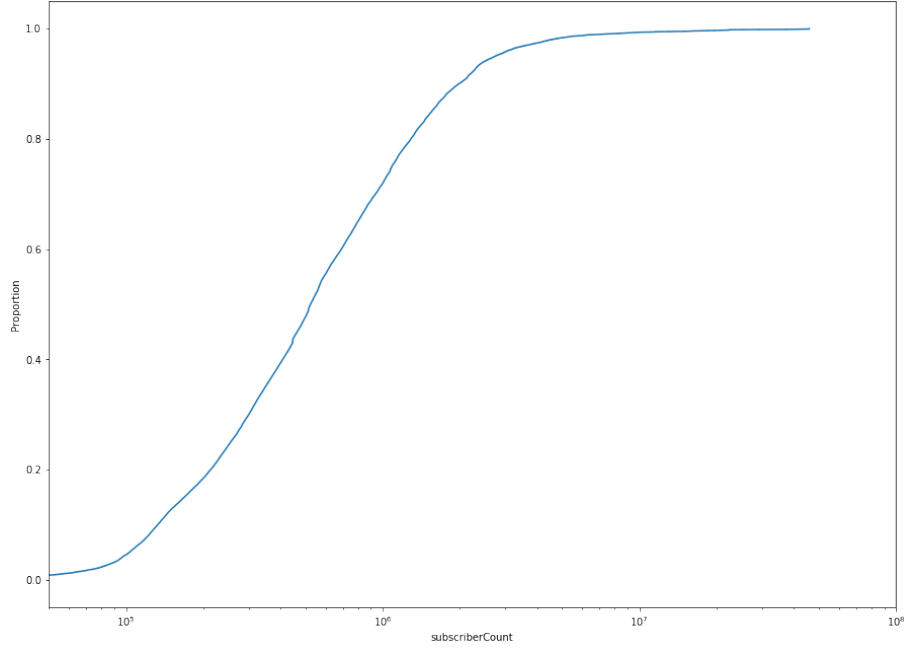


Figure 2.1: ECDF of the subscribers

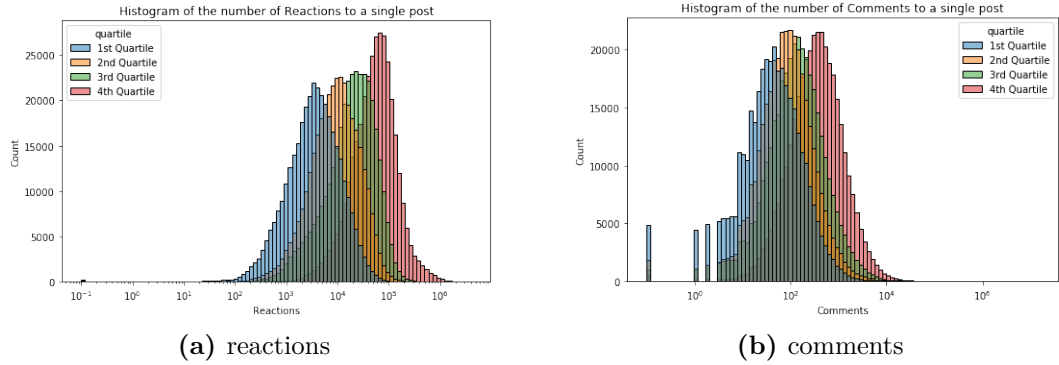


Figure 2.2: Histograms of reactions and comments

From the histograms, we notice a logarithmic Gaussian that has its mean moved more to greater magnitude the higher is the quartile: this does not come as a surprise since we can imagine that influencers with a higher audience will have a higher amount of interactions. The said differences are minor if we consider the comments since they tend to be less used than the simple reaction.

We see in fact, from the ECDF plots 2.3, that almost all posts have a small amount of comments compared to the number of reactions.

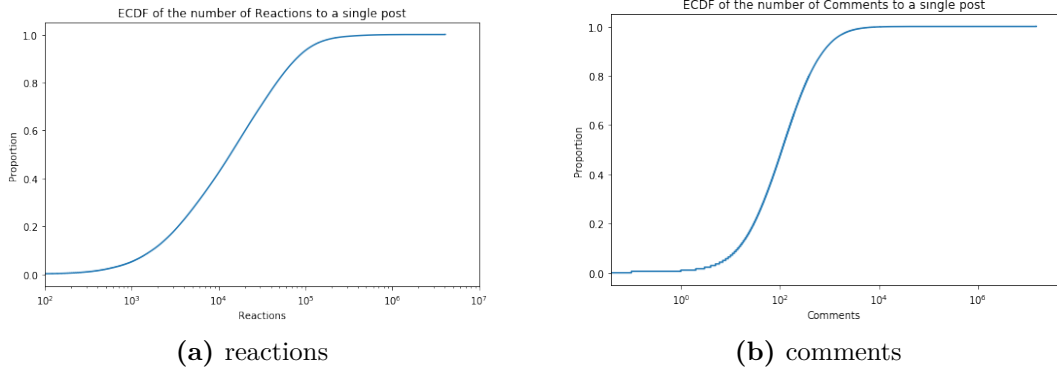


Figure 2.3: ECDF of reactions and comments with respect to the posts

Due to the need of taking into account the said occurrences, the engagement considered: while Instagram defines the first two metrics, the engagement rate has been a research topic already discussed in multiple papers computing it with different formulas[30][31]. In this exploratory analysis, we used a simplified version of commonly used metrics computed as follows:

$$Engagement = \frac{Comments + Reactions}{followers} \quad (2.1)$$

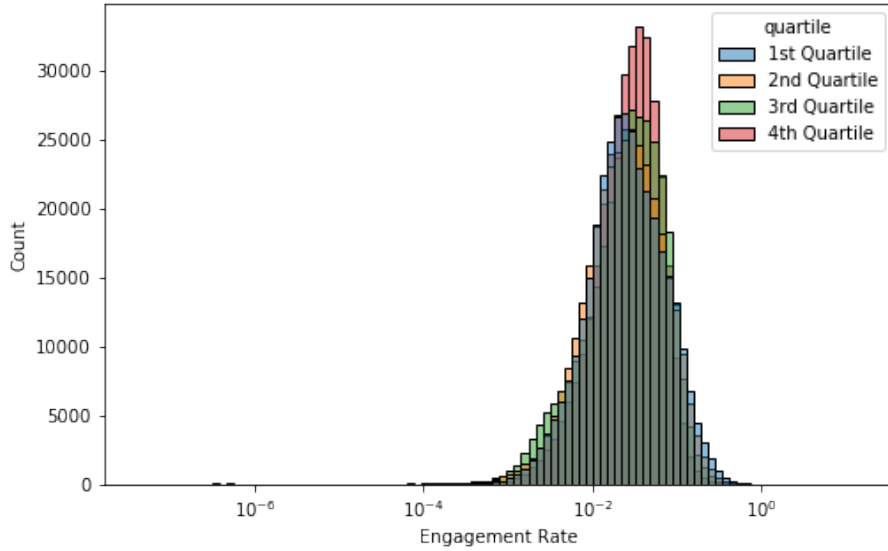


Figure 2.4: Histogram of the engagement rate of the posts

In this case, given what seems to be a higher tendency for more popular accounts

to produce more content, we see that for the forth quartile the distribution is more concentrated and, as a consequence, less variable, other than that, there are no further significant difference between the categories.

2.3.3 Textual Characterisation

As previously specified, each datapoint has some characteristics that directly controlled by the creator and measurable. The description, a short text of 2200 characters, is used to describe the content, mention someone, or write some hashtags. A quantitative analysis of those metrics , visible in appendix A.4, confirms that there are minor behavioural differences between the categories of influencers. In particular, it is easily noticeable how the first quartile deviates from the others regarding the number of mentions and hashtags used and not with respect to the number of words used. If, in fact, there is no reason for which a "smaller" influencer should write less or more than a larger one, it can be understandable that creators with a smaller userbase would try to use more mentions (to create network effects with other similar influencers) and hashtags (to be more easily reachable by new followers) than people that are already famous.

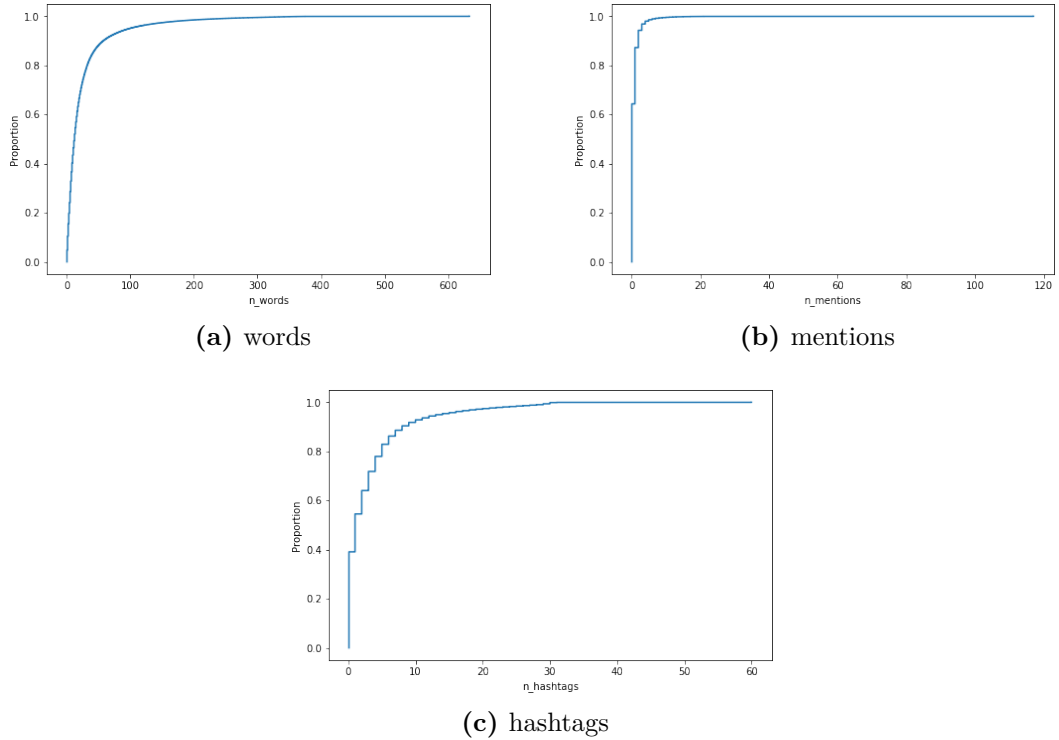
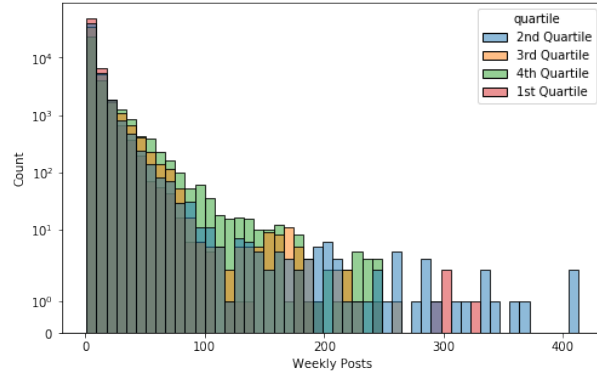


Figure 2.5: histograms of the description metrics

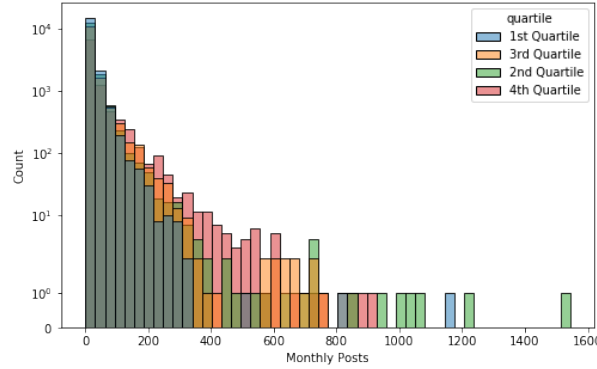
Clearly, the vast majority of the posts contain very few mentions and hashtags (that have been lately limited by Instagram) due to their nature, while the length of the text, in general, tends to be more logarithmic-like.

2.3.4 Time and Frequency Characterization

Finally, the distribution of the posts in time, their frequency and the the different performances related to the posting time can give us some insights. For intervals of one week and one month (28 days), the distribution in time shows that the great majority of users produces a smaller volume of posts, far less than one per day while only some outliers tend to produce multiples per day. By checking those exceptions, we notice that they are newspapers or institutions that need to communicate multiple times per day.



(a) Monthly



(b) Weekly

Figure 2.6: Histograms of the frequencies

Regarding the periodicity of the posts, while no particular effect is given on larger time intervals (for example, throughout the year) we see that the posts are

have visible peaks in the daily distributions 2.7:

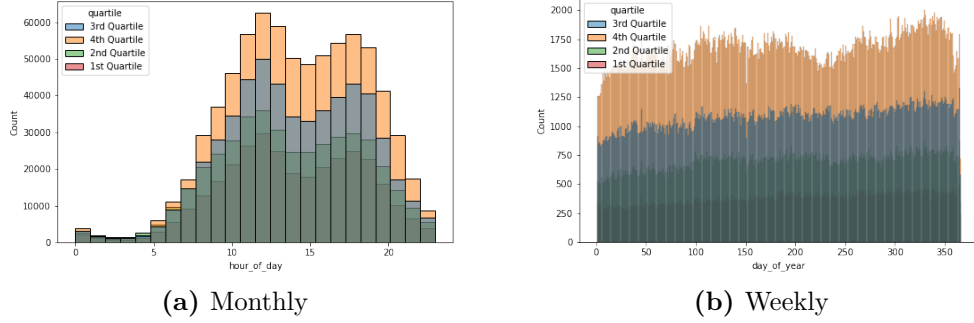


Figure 2.7: Distribution of the post throughout the time period

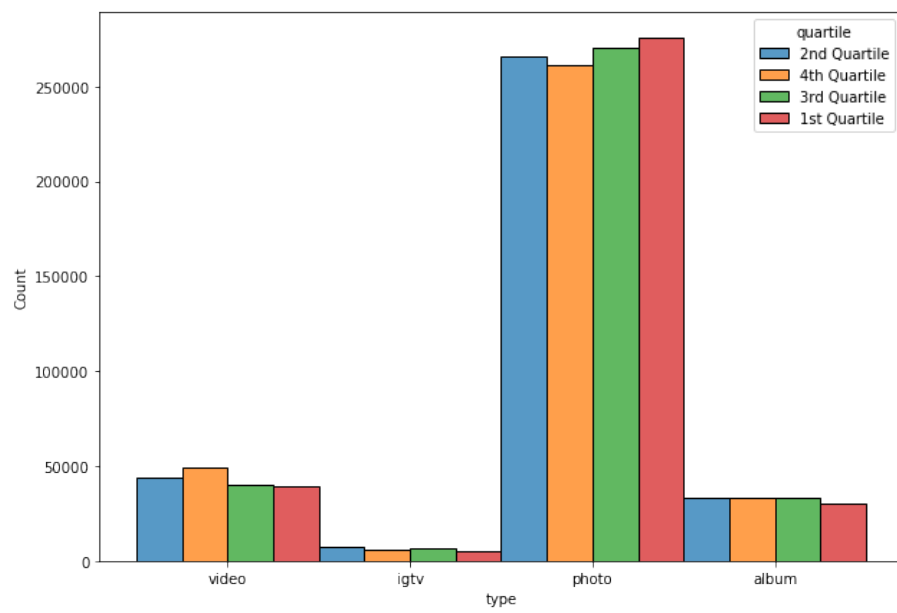
2.3.5 Other metrics

At publishing time the influencer can decide what type of post to create, 4 types of post can be created:

- Photo: the first type of post that instagram implemented, it contains one single picture.
- Video: this type of post contain one single video of limited lenght
- Album: it can contain both picture(s) and video(s)
- IGTV: the last addition to the typologies that can be published, they are longer videos, in higher quality, and can be seen with the IGTV app.

Unsurprisingly the most published posts are the ones containing one single photos followed by videos, albums and then IGTV with no particular difference between different classes of influencers.

Regarding sponsorthips, only 0.66% of the posts is sponsored but more than 40.97% of the accounts published at least one sponsored post.



Chapter 3

Relevant Theory

In the following section, we will explore the theory relevant to this thesis work. First, the regression problem will be defined both in general and relatively to the various methods employed; then, other theoretical tools used will be presented.

3.1 Classical Regression Methods

The concept of regression, in its modern form, and the term itself were born in the XIX Century with Sir Francis Galton and Karl Pearson[32]. Regression analysis is, in general, a method aimed at calculating the association between the outcome variable and one or more features. While classification aims at predicting a discrete variable, that is, a class, regression has as a goal the forecast of a continuous quantity. Mathematically, the regression can be defined as in 3.1:

$$\begin{aligned} Y &= g(X, \beta) + e \\ \beta &: \text{unknown parameters} \\ X &: \text{independent variables or features} \\ Y &: \text{dependent variables or outcome variables} \\ e &: \text{error terms} \\ f() &: \text{function fitting the data} \end{aligned} \tag{3.1}$$

3.1.1 Linear Regressor

Linear regression is commonly used to map a linear relationship between some descriptive variables and a real-valued outcome: depending on the number of

explanatory variables, the linear regression is named simple or multiple linear regression[33].

$$y_i = \beta_0 + \sum_{j=1}^{j < N} \beta_j x_{ij} \quad (3.2)$$

The equation in 3.2 is composed by the predicted value y , the independent variable vector x and the coefficients vector β . Linear regression then finds the β coefficients that produce the best-fitting line for the input data, the line that produces the lowest loss. Loss, or cost, is the measure of how far is the model from the actual training data; the higher the loss, the lower the prediction accuracy.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_{pred} - y_{true})^2 \quad (3.3)$$

$$\text{minimize}(MSE) \quad (3.4)$$

One of the most commonly used cost functions is the Mean Square Error, which is defined as in 3.3: the problem of linear regression can then be reduced to the 3.4 minimization problem.

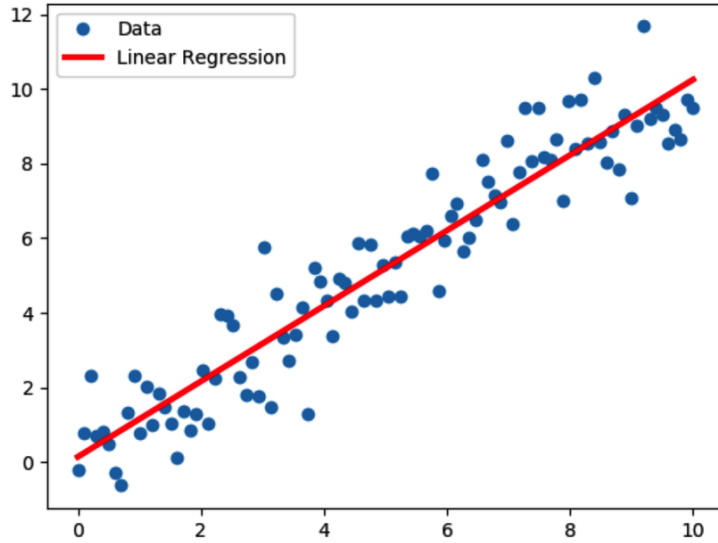


Figure 3.1: Example of Linear Regression. Taken from [34]

3.1.2 Random Forest Regressor

Random Forest Regressor[35] is an ensemble method used for regression: it fits an aggregation of decision trees on subsets of the dataset.

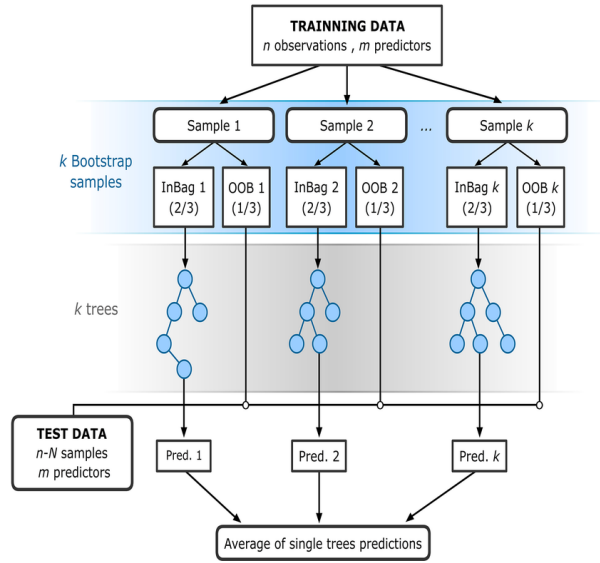


Figure 3.2: Example of Random Forest. Taken from [36]

A Decision tree is composed of a set of nodes and leaves; nodes are features of the dataset while branches are decisions regarding the said feature; leaves, finally, are the outcome of a chain of decisions. Those trees are organized in an ensemble of predictors: said simply, many regressors are run "in parallel" and their output combined to reach a more accurate result. For the algorithm to perform as wanted, the trees must have some slight differences; for this reason, a method called bagging is used to train them: with bagging, the training set, before being fed to a tree for fitting, is randomly sampled with replacement.

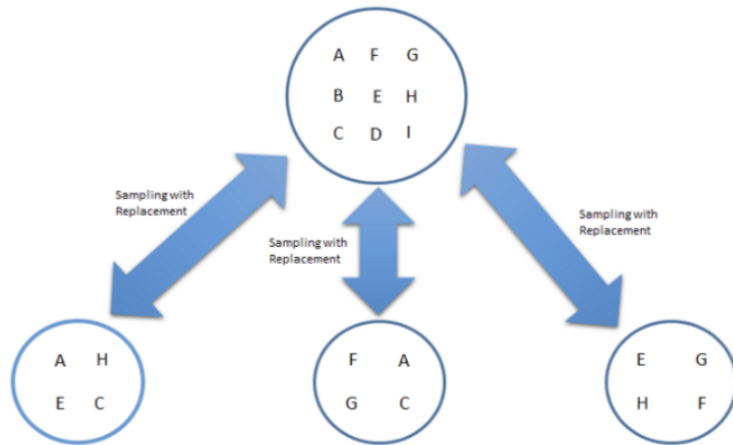


Figure 3.3: Example of Bagging. Taken from [37]

To lower the correlation between trees, which features with a high predicting weight could cause, only a random sample of the features is considered for the split at every training step of each tree. Similar to what happens in classical decision trees, the splits are computed in the optimal cut point using metrics such as entropy or Gini impurity as a measure.

3.1.3 Gradient Boosting Regressor

Gradient Boosting regression[38], similarly to the random forest regressor, of whom is commonly seen as the evolution, is a regression algorithm that uses an ensemble of decision trees to generate a prediction. Unlike its forefather, in GBR, the forest of predictors is built one tree at a time in an additive model to improve the deficiencies of the existing weak learners. At a given step with N trees, the loss of the prediction is computed, and the gradient descent procedure is performed. To do so, a new learner is added so to reduce the error: the $N + 1^{th}$ tree is joined with the others with its parameters selected to reduce the residual loss.

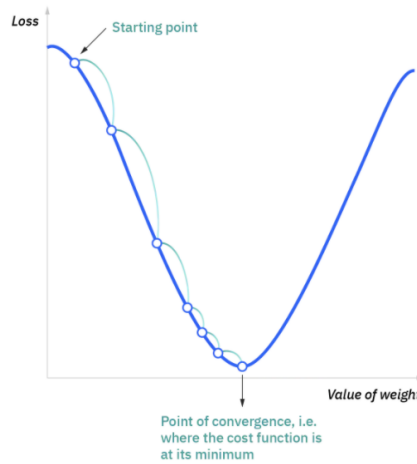


Figure 3.4: Example of the gradient descent procedure in a 2D space. Taken from [39]

If with the Random Forest Regressor the final prediction of the model was computed as the average of the learners, with the Gradient Boosting Regression Algorithm the outputs of the new trees are combined sequentially and weighted with the depending on the learning rate. While usually better at making predictions, GBR can be more prone to overfitting and slower in both the training and testing phase.[41]

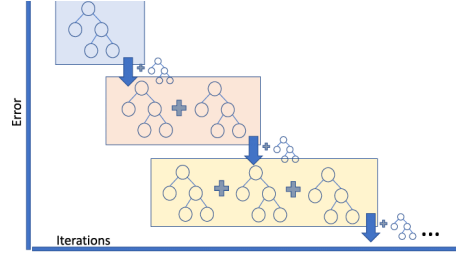


Figure 3.5: schematisation of GBR. Taken from [40]

3.2 Neural Network

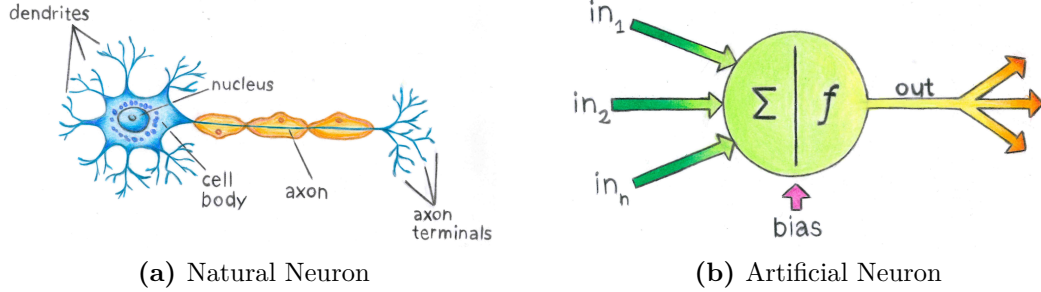


Figure 3.6: Artificial Neuron and Natural Neuron. Courtesy of Giulia Marchisio

Neural Networks (NNs), also identified as Artificial Neural Networks (ANNs)[42], are complex computing models that are inspired by the organic neural circuits that constitute the brain of sentient beings. Those complex structures are composed of multiple units organised into layers, called neurons, and connected to each other through directional bonds called links. The link between two neurons, x and y , is needed to propagate an activation function α from one to another, and it is characterised by the weight w , giving the power of the connection.

The activation function can be of the most various forms, from a dummy linear one to softer thresholds.

Depending on the manner different neurons are organised, NNs can be subdivided into two classes:

- **Feed-Forward Network:** connections between neurons are organised as a directed acyclic graph; being directed in one direction, the outputs are the product only of their current inputs and there is no internal state or memory.
- **Recurrent Neural Network:** those particular Neural Networks are characterised by the fact that their neurons feed the downstream one and can be potentially

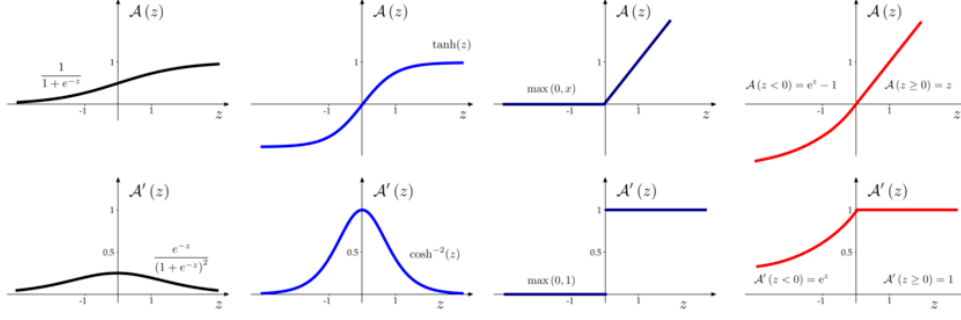


Figure 3.7: Some of the most common activation functions. Taken from [43]

connected to other layers that are not next in the sequence. Consequently, the current output is the effect of both the inputs and all previous states (the initial one as well).

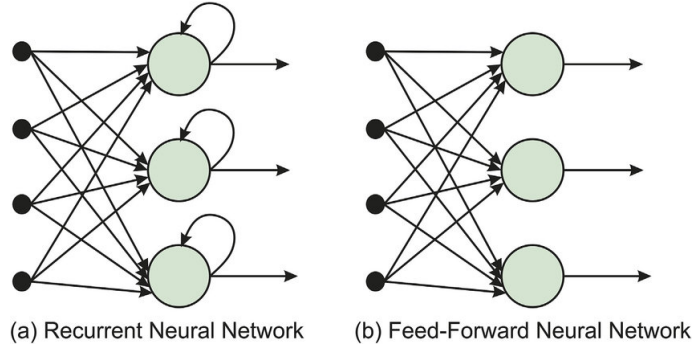


Figure 3.8: Comparison between the two classes of NN. Taken from [44]

The learning process consists of the network adjusting the weights and biases of its connections to improve accuracy in the given tasks; gradient descent is used to find the direction in the parameter space that minimises the loss and consequently use it to adapt the hyperparameters. Given the shortcomings of classical gradient descent, other methods such as SGD or Adam are often used. The taxonomy of the layers used in neural networks is exceptionally vast; the following section gives a short description of the ones used in this thesis.

3.2.1 Dense Layer

The Dense Layer is one of the simplest and most commonly used layers used in NN. It is called in such way because every unit of one layer is connected to any other unit of the following layer; connection density is high. This layer operation is

substantially performed applying an activation function α to the dot product of the input i and the kernel (or weight) w plus the bias b .

$$output = \alpha(i \cdot w + b) \quad (3.5)$$

It is clear that this layer, if a linear activation function is chosen, performs a simple product between matrices: it is usually used with the appropriate activation functions so that multiple stacked layers can be used to describe a highly non-linear polynomial.

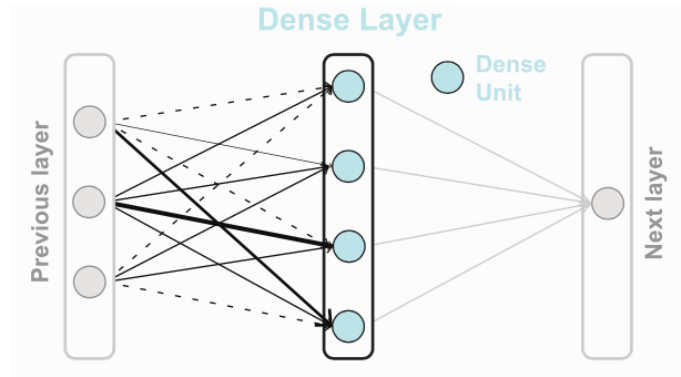


Figure 3.9: Dense layer connections. Taken from [45]

3.2.2 LSTM Layer

The main idea of LSTM is that, to analyse sequential data, the network would benefit from the capacity of learning what should be remembered for a certain amount of time, what should be forgotten and what should be used right away[46]. This functionality is permitted by the presence, as components of the LSTM unit, of four parts:

- Cell
- Input Gate
- Output Gate
- Forget Gate

While the so called cell is used to store information through the learning process, other components are used to govern the flow of information: the forget gate is used to regulate what knowledge is retained and what is thrown away; this is done by feeding data from the current input and previous hidden state to a sigmoid:

depending on how near the output will be to zero or one the more information will be forgotten or remembered. As the name can hint, the input gate selects what information should pass to the cell, working as a filter; finally, the output gate determines the value of the following hidden state or, put simply, selects what is important from previous steps and what is not.

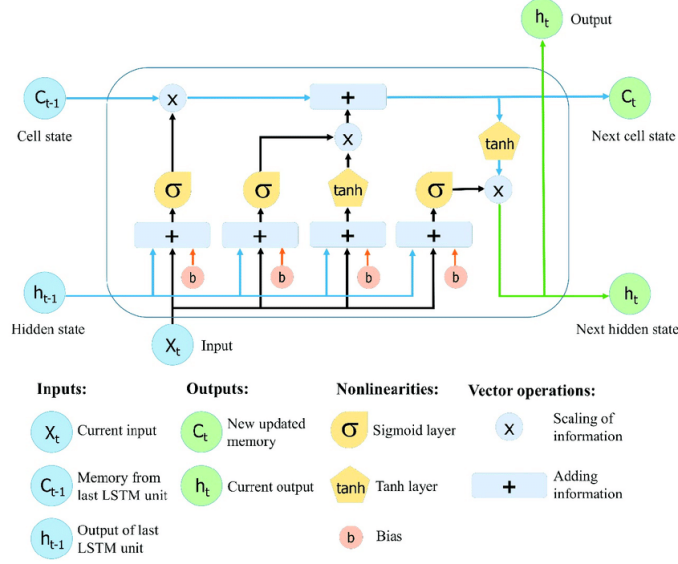


Figure 3.10: Structure of an LSTM cell. Taken from [47]

3.2.3 GRU Layer

Gated Recurrent Units[48], or GRUs, are an evolution of LSTM that implement a gating mechanism: the main differences between them is a lower number of parameters and the lack of the output gate. GRUs, instead of having the Input, Output and the Forget gate, are equipped with two gates called reset and update gate. The first gate decides the amount of information to be passed, the second one how much should be forgotten.

Thanks to the lack of one gate and the cell, the number of operations to be performed is lower, and, as a consequence, training is speedier. While there is no prominent winner between classical LSTMs and GRUs, it seems from empirical evidence that the latter performs better for shorter sequences while LSTMs are better layers for longer sequences: this is probably motivated by the fact that LSTM is more memory oriented thanks to the presence of the cell component.

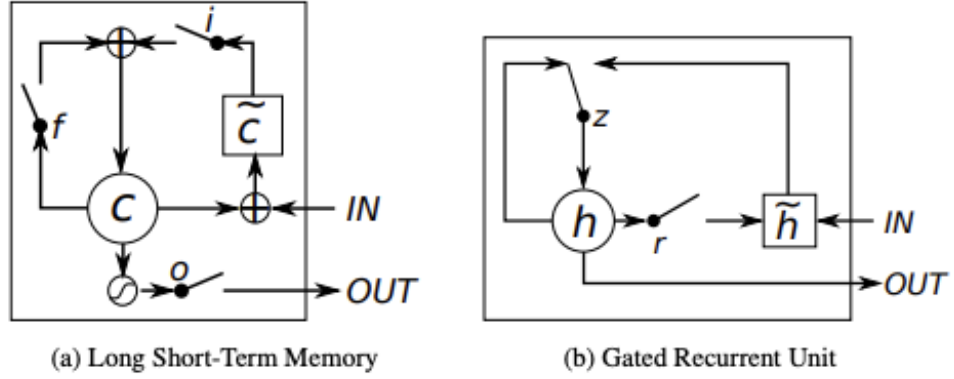


Figure 3.11: Architecture comparison between LSTM and GRU. Taken from [48]

3.2.4 Bidirectional Layer

The bidirectional layer is a wrapper layer[49][50] offered by the Keras framework to implement the bidirectional version of layers such as RNN, LSTM and GRU ones. From a theoretical point of view, the main idea of the bidirectional architecture is to divide the neurons into two groups, one communicating in the positive while the others in the negative direction. This strategy permits the neural network to interpret each point not only with respect to its past but to a more general neighbourhood.

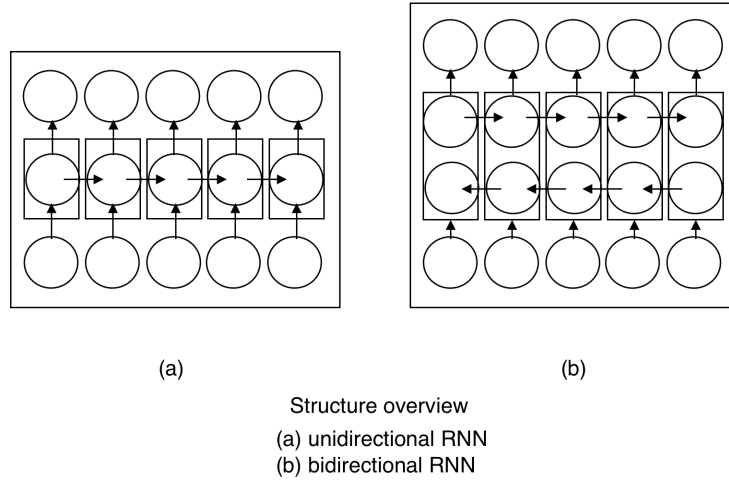


Figure 3.12: Differences in the general architecture of a unidirectional and bidirectional NN. Taken from [51]

This particular layer can, as a consequence, have better performances at the cost of being slightly slower during the training phase.

3.3 Outlier Detection Methods

Outlier detection is the process of discovering anomalous, singular or wrong observations from data[52]. Outlier detection can be performed in a supervised or unsupervised manner: in the first case, training examples labelled as "inliers" and "outliers" are given to the system to learn recognising the, in the second, on the other hand, no label is given, and the algorithm will assume that those datapoints that "lie out" of the population distribution should be flagged.

3.3.1 Isolation Forest

The isolation forest outlier detection algorithm is an ensemble method based on partitioning[53][54]; for each tree, the algorithm proceeds by randomly selecting a feature and then generating a split in the interval of the observed values; such partitioning is performed recursively until a point is isolated. The partitioning path, the number of splits needed to isolate a sample from the dataset, is way shorter for anomalies since their abnormality makes them "stand out" from the normal distribution, as in 3.13; the average path length is calculated on the whole forest of random trees.

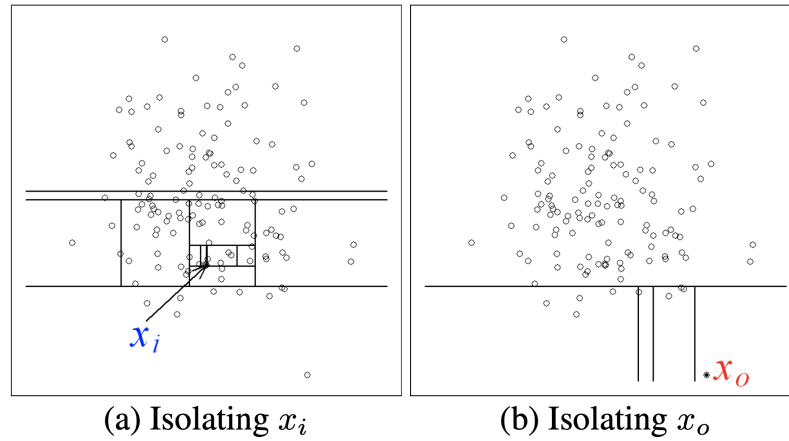


Figure 3.13: Comparison of the number of cuts to isolate an inlier (a) and an outlier (b)

Despite suffering in its more straightforward implementations of the same problems of distance-based methods, this method can reach optimal performances

in high dimensional feature spaces.

3.3.2 Z-score Outlier detection

Z score is a statistical tool, also called standard score, used to measure the distance of a point with respect to the mean: the score of a point, which can fall in the interval $[0, +\infty)$, is equal to the number of standard deviations a point is from the mean of the distribution. The first three integer values of the Z-score, 1, 2 and 3, are commonly used because they comprehend, respectively, the 68%, 95% and 99.7% of the distribution.

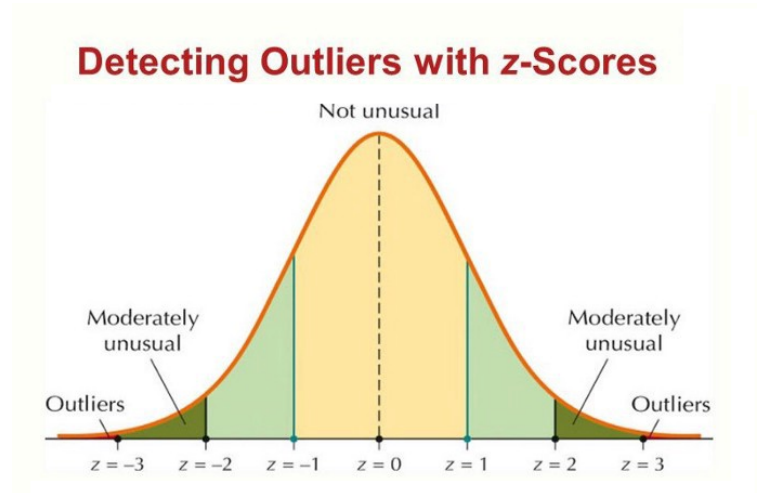


Figure 3.14: outlier detection given a normal distribution. Taken from [55]

Then, given a particular attribute, a cutoff value, such as $z = 3$, can be chosen to select which points are possible outliers. In the case of more than one dimension, a "combined score might be used.

Chapter 4

Data Mining, Transformation and Loading

4.1 Insights

As described in section 2.2, the dataset was generated using Crowdtangle API and selecting the activities of top Italian influencers in six years, from Jan 1, 2015, to Dec 31, 2020: the result of this selection was the generation of a dataset of 2 036 966 posts.

4.2 Data Mining

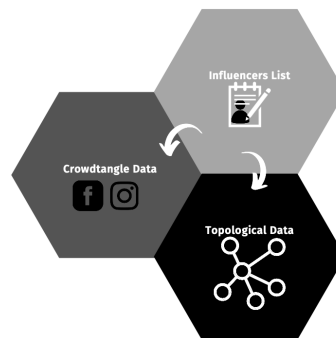


Figure 4.1: Sources of Data

Three datasources where mined to produce the data used in the research:

- list of influencers to be analysed using crowdtangle: a list of the top 1611 italian influencers was downloaded from www.influenceritalia.it/ .
- Crowdtangle Data for the list of influencers: the huge dataset of 2 036 966 posts and their characteristics relative to the influencers of the list.
- Graph Topological Data: used to relate the influencers with each others on the basis of follower/following relation, extracted as in 4.4.5.

The script used to produce download the data needed four days of run-time keeping a precautionary query limit of one each ten second so to stay below the suggested limit and avoid stressing both our hardware and crowdtangle servers.

4.3 The Instruments: Pyspark, HDFS and the Cluster

Apache Spark, developed in 2009 at Berkley, is a unified analytics engine used for big data and machine learning written in scala[56]. Thanks to its capability to perform processing tasks over enormous datasets by distributing them among multiple working nodes, it has become a widely used solution by countless companies and research institutions. In its most common usage, Apache Spark is said to be deployed in "Cluster Mode", which means it is deployed on multiple computers or servers. Its architecture, at the conceptual level, consists of three elements:

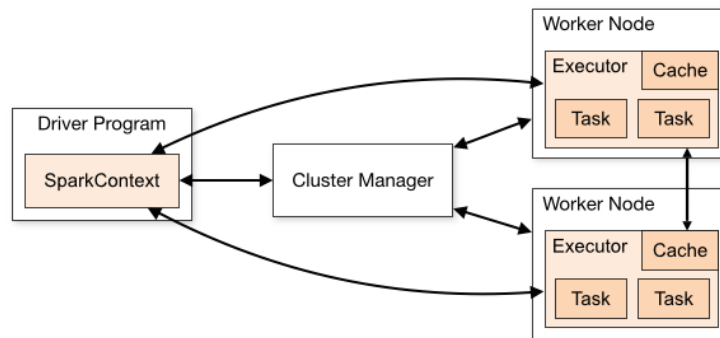


Figure 4.2: Organisation of a Spark Cluster, Taken from spark.apache.org

- The Driver: an interface that converts user code to multiple tasks and distributes them across workers
- The Executor: the program that runs on the said nodes to execute the assigned tasks

- Cluster Manager: some form of management is necessary to distribute the tasks, manage the workers and perform control routines. Numerous managers are available; some of them are YARN and Kubernetes.

From a logical point of view, the Spark framework is based on five components:

- Spark Core: the nucleus of the Apache Spark framework, it provides the execution engine to the platform, defines and manages the framework's native data structure, and is used by other components to perform a multitude of operations.
- Spark SQL: a component to work on data in an SQL way. It defines one of the most valuable abstractions used in Spark, that is, the DataFrame.
- Spark Streaming: Library used to work on streaming data. Despite its name, it does not work in an authentic streaming way but divides the streaming into chunks to be processed.
- MLlib: a low-level machine learning library that supports a few widespread statistical, analytical and machine learning methods.
- GraphX: it is a tool used to organize, elaborate and transform graph data in a distributed manner.
- Spark R: package of Apache Spark able to process large amount of data with R.

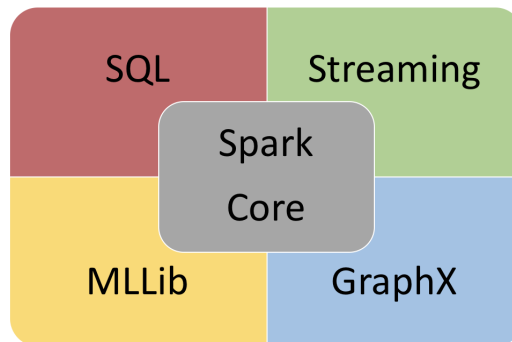


Figure 4.3: Components of Spark, Spark R not included. Taken from [57]

Finally, Spark offers essentially three types of data structures:

- RDD: Resilient Distributed Dataset, Spark's fundamental data structure that implements immutable partitions of data. While fast and fault-tolerant, they are more challenging to use and cannot be used with SparkSQL.

- DataFrame: defined by Spark SQL, offers the possibility to organize data into table-like structures with named columns.
- Dataset: while it does not offer named columns, it is a data structure that offers the benefits of RDD and the SQL optimized execution engine.

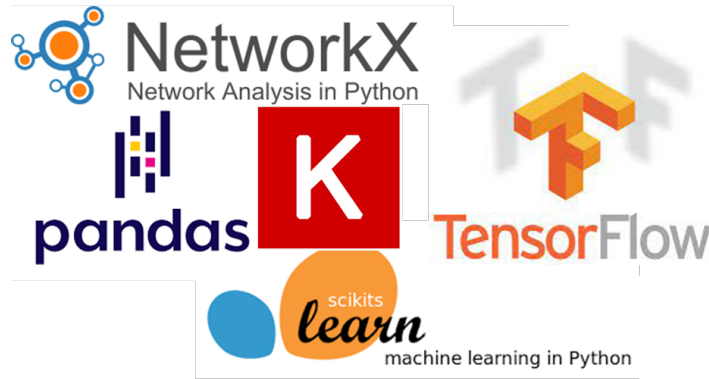


Figure 4.4: Libraries used

In the last part of the pipeline, the Data was elaborated essentially with four tools: Pandas, SciKit-Learn, TensorFlow (and, consequently, Keras) and NetworkX.

- Pandas[58] is a Data Analysis Library for Python and is extensively employed for data science and machine learning tasks; it provides data structures and operations for managing tables and time series in an Excel-like manner.
- SciKit-Learn[53] is a Python library that implements classification, regression, clustering and, more in general, supervised and unsupervised learning algorithms.
- TensorFlow[59] is a library for machine learning and used principally for deep neural networks and artificial intelligence development; its frontend is offered in python while the backend utilizes C++ to provide high performances. Keras[49] is a high-level API for the TensorFlow library, mainly used to simplify some operations.
- NetworkX is a python package for generating, transforming, and studying networks' topology, dynamics, and functions.

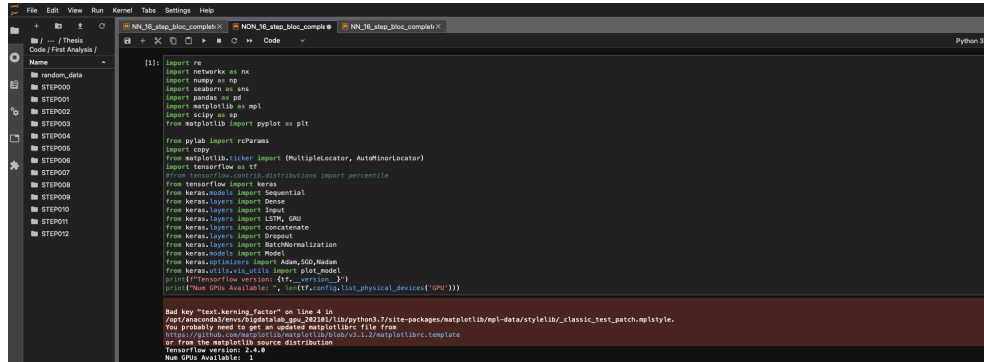
The Data, to be elaborated by Spark, was saved on the computing cluster using the HDFS filesystem: the Hadoop Distributed File System is the principal data storage used by Hadoop applications (and, consequently, by pyspark). In short, on

HDFS, files stored are divided into blocks saved onto DataNodes, nodes that manage the storage of the nodes they run on; those nodes have their mapping managed by one (or more) NameNode that additionally performs filesystem operations and mapping.

This complex software infrastructure was part of Politecnico's BigData@Polito Cluster. The cluster, after 2020, reached a capacity of:

- 33 storage workers equipped each with:
 - 216 TB of disk storage
 - 384 GB of RAM
 - Two CPUs with 18 cores/36 threads each
- Two nodes equipped with 4 GPUs for experimentation

In particular, for the present thesis work, two instances were used: one with 24 reserved CPU Threads/120 GB memory, max 70 CPU Threads/320 GB memory, while the other, used for Deep Learning training, with 1 reserved GPU. Reserved 32 CPU/64 GB memory, max 64 CPU/256 GB memory. Jupyter Lab provided access to the cluster and its software by allowing the spawn of a virtual server to work with Jupyter Notebooks, interactive computational environments accessed from web browsers.



```
[1]: import re
import networkx as nx
import numpy as np
import os
import pandas as pd
import matplotlib as mpl
import scipy as sp
from matplotlib import pyplot as plt

from pylab import rcParams
import copy
from matplotlib.ticker import MultipleLocator, AutoMinorLocator
import tensorflow as tf
from tensorflow.keras.layers import Dense, LSTM, GRU, Concatenate, Dropout, BatchNormalization, Model, Adam, SGD, Nadam
from keras.optimizers import Adam, SGD, Nadam
from keras.callbacks import ModelCheckpoint

print('TensorFlow version: (tf.__version__)')
print('Num GPUs Available: ', len(tf.config.list_physical_devices('GPU')))
```

Bad key "text_kerning_factor" on line 4 in
 /opt/conda/lib/python3.7/site-packages/matplotlib/mpl-data/typelib/_classic_test_patch.mplstyle.
 You probably need to get an updated matplotlib file from
 https://github.com/matplotlib/matplotlib/blob/master/mplstyle/mplstyle.template
 or from the matplotlib source distribution
 TensorFlow version: 2.4.0
 Num GPUs Available: 1

Figure 4.5: View of a spawned Jupyter server with an opened notebook

4.4 Data Transformation

The data, after being downloaded locally, was first stored onto the HDFS filesystem of the cluster to be elaborated by the spark tools listed at 4.3. Two original datasets were used, the one crawled by CrowdTangle API was organised as follows:

Attribute	Description
account	Contains user information
brandedContentSponsor	Contains sponsor information
date	Date of posting of the content
description	Text description of the content
expandedLinks	Links contained in post
history	list of elements, composed by: actual comments and reactions date predicted comments and reactions
id	Crowdtangle content ID
imageText	Text contained in the image
legacyId	ID of post now not used anymore by the system
media	List of urls and dimensions of the medias
platform	Platform the content was posted, here instagram
platformId	Instagram content ID
postURL	URL of the post
score	Score given to the post by Crowdtangle
statistics	Similar to history
subscriberCount	Number of followers at the time of the posting
type	Content type
updated	Update time of the post

Table 4.1: Attributes of the datapoints contained in the dataset

From this raw form, the data was transformed in essentially four types of characteristics:

- Temporal Characteristics
- Volumetric Characteristics
- Pointwise Characteristics
- Identification Characteristics

This division was decided empirically on the basis of the type of elaboration it needed to reach its final form and its format.

4.4.1 Identification Characteristics

I identified each datapoint by attributes that were needed to distinguish it univocally from the other during the analysis phase. Instagram attributes three unique

identifiers to each content: an alphanumeric string, called *id* composed by the numerical id of the account prepended to a LongInt is used as the unambiguous identifier of the post(as in 4.2). Slightly less than half of the posts have a numerical identifier, the *legacyId* probably used earlier on the platform; a *URL* is provided as well to point to the webpage containing it.

$$\overbrace{\underbrace{183367}_{\text{account identifier}} \mid \underbrace{23494958853485762}_{\text{LongInt}}}^{\text{post identifier}} \quad (4.1)$$

Figure 4.6: post identifier, composed by long int and account identifier

Accounts characterisation is slightly more generous than for posts: other than the expected *id* and *URL*, they are also designated by means of a variable *Name* that can be chosen and changed by the user and can contain a variety of characters, a *handle* (commonly known as username) that can contain only roman letters, numbers, underscores and full stops and a *platformId*, an identifier that is different from the one employed to create the post id and that it refers to Instagram, not crowdtangle.

handle	id	name	platformId
la_setta_dei_poeti_estinti	7573068	La Setta dei Poeti estinti	4126248485
vale.exposito	4830238	Valentina Esposito	1562286000
dio	8280535	Il Dio di Instagram	8356455280
corriere	1588220	Corriere della Sera	481777763
skysport	1411771	Sky Sport (Italia)	2105417318

Table 4.2: Example of account entries, url not visible for pagination constraints

The same information used to characterise the accounts was present in the *brandedContentSponsor* for those post that were sponsored by an external account.

The relative redundancy of those characteristics made possible to select only some of them, in particular:

- *id*, as *post_id* to avoid confusion
- *platformId* as *AccountID*
- *account.handle* as *username*

4.4.2 Temporal Characteristics

As stated previously, the dataset was mined during a time interval of six years. Crowdtangle provided substantially three typologies of temporal data:

- the date, saved as a string, of the time of publishing of the post
- the date, saved in the same format, of the (eventual) updating of the post
- the data of each sampling, saved in an array parallel to the one of the referred metric

The information contained, while exhaustive, necessitated some transformations to be used in the regression. In chapter 2, it was shown that some periodic dynamics in the number of reactions of a post were present; moreover, it was self-evident that the past development of an influencer would have some type of effect on present and future evolution. As a consequence, two groups of temporal dimensions were generated:

- absolute dimensions: they were used to place the datapoints in the time-space
- periodic dimensions: employed to locate them in some period intervals

Absolute dimensions

Due to the fact that a timestamp would have been too accurate and would have identified the datapoints too precisely, probably bringing to overfitting, the time granularity was discretely decreased: in particular; it was decided to identify the first day, week and month of the dataset as 0 and then assign to each point the number of the time interval from the zeroth one. Since there was no interest in having a correspondence of those intervals with actual weeks (intended as from Monday to Sunday) and months (from January 1 to January 31), months were considered consecutive intervals of 28 days and weeks intervals of 7.

Periodic dimensions

As seen in section 2.3.4 and in previous literature, some periodic dynamics are observed on different temporal levels: those dynamics are usually caused by real-life occurrences such as day-night shift, working hours, holidays and workdays. In this situation, there was no interest in the absolute placement in time of the action but on the relative placement with respect to the interval; moreover, due to the circular essence of time, the creation of a circular transformation of the time-line was needed to model the cyclicity of the day, week and year. In practice, some intervals of interest were chosen:

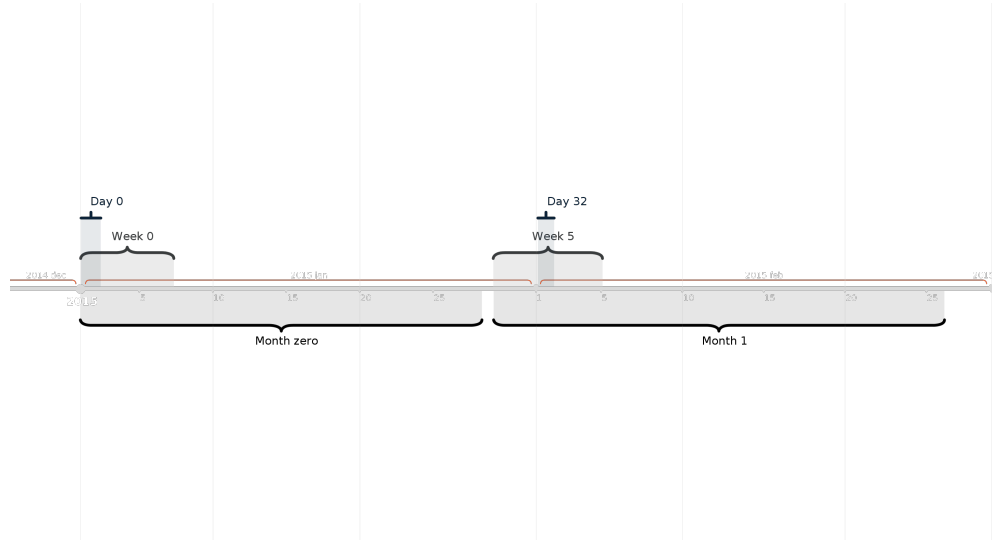


Figure 4.7: Example of how the time interval was divided

- the hour of the day
- the day of the week
- the week of the year

They were then placed on a circle with their beginning and end around the 0 degree and projected on the x-axis and y-axis using sine and cosine to have a bidimensional representation of time that respected the relative distance between timepoints.

$$\begin{cases} x_t = \sin 2\pi \frac{t}{T} \\ y_t = \cos 2\pi \frac{t}{T} \end{cases} \quad (4.2)$$

t : time unit of between{hour_of_week,day_of_week,week_of_year}

T : respectively, 24,7,52

4.4.3 Volumetric Characteristics

The second typology of characteristic, and probably the most important one, referred to the historical evolution of the influencer popularity. Three typologies of popularity metric were chosen:

- number of favourite reactions of the post at a specific time
- number of comments of the post after a particular time

- number of followers at the time of the creation post

This choice was made because those metrics, and their elaborations, such as the engagement, are present for all the influencers, easily accessible and commonly considered the most tangible effect of notoriety. Depending on the regression method implemented, namely, Random Forest Regressor or Neural Network, the volumetric characteristics were transformed to series or point attributes.

Average Reduction

Due to the limits of the regression method chosen, all metrics attributes had to be numerical attributes. Analysing the dataset, it appeared that the effect of an influencer's actions did not correlate to its popularity after one month, as seen in image 4.9. To perform this analysis the correlation was computed between the change in the posting frequency and the change in the number of followers in the T next months (T on the x axis of image 4.9)..

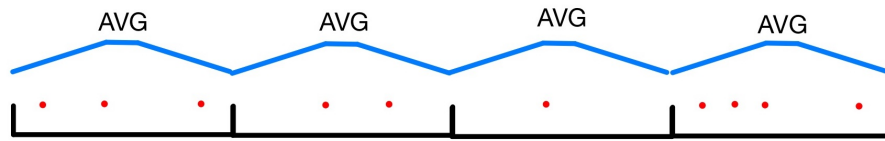


Figure 4.8: Averaging method, red points are posts on the timeline, intervals are weeks

The decision, confirmed by a simple Sequential Backward Search, was then to take into consideration only posts done in the last month, decreasing their granularity to weekly averages.

Series Reduction

The usage of more complex and modern regression methods, such as a dual legged neural network with an RNN branch, permitted to avoid decreasing the granularity of the historical data. With a different approach from the previous section, the time-series were not chosen as a predefined time period but as the list of the last N posts: this permitted to implement an event-based neural network able to adapt smoothly to very different posting behaviours. The value N was significantly varied to maximise the performances of the neural network while minimising the running time. After numerous runs, the decision fell on $N = 16$.

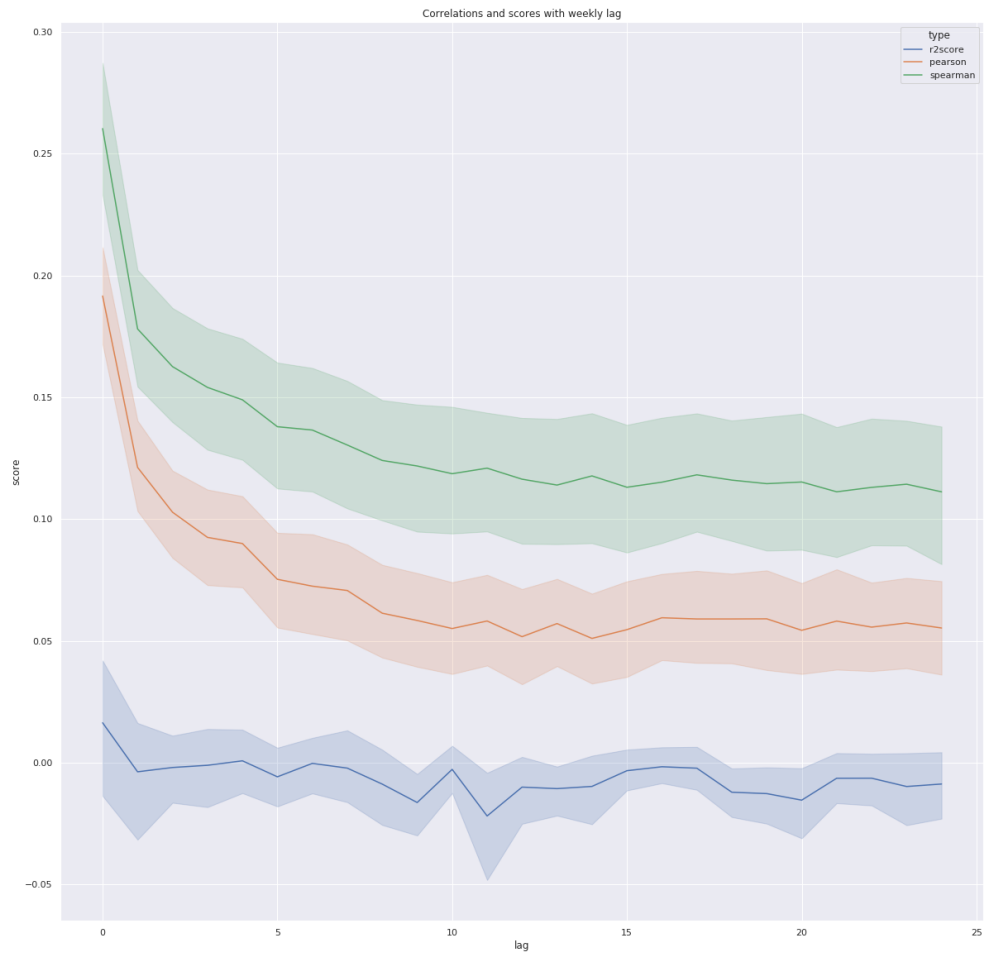


Figure 4.9: Average correlation (and σ) between posting frequency change and increase in followers

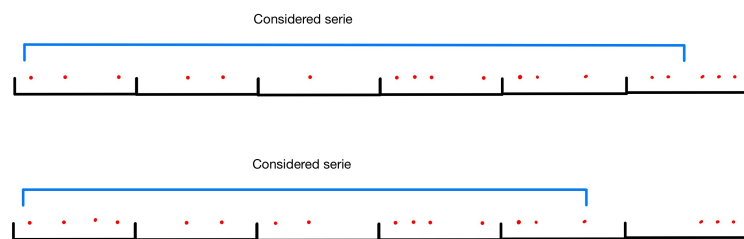


Figure 4.10: Series method, red points are posts on the timeline, intervals are weeks, the length of the window creating the series varies depending on the number of posts to include

4.4.4 Point-wise Characteristics

In this category falls the majority of the attributes extracted; Due to the nature of this thesis, it was decided to avoid using as attributes qualitative information from the post's content: no NLP algorithm was applied to the description nor Convolutional Neural Network on the images and videos. The lack of qualitative data motivated us to extract even more knowledge from the post features.

The features can be divided into X categories:

- Textual Information
- Media Information
- Popularity Information

Textual Information

Despite not being the central focus of the social network, it is not uncommon for Instagram posts to have a text description: this text can contain up to 2200 characters of any type; other than a message, it is the praxis of this social to add some hashtags here (or eventually in the first comment, but this occurrence could not be tracked due to the lack of information in the database) or mentions. While this limit was higher in the past, the maximum number of mentions is now 20, 30 for hashtags. The textual dimensions extracted where:

- number of hashtags
- number of mentions
- number of words

Those attributes were generated by extracting a list using regular expressions and then calculating the list's length.

`#(\w+)` `\B@\w+` `[\w-]+`

Figure 4.11: regexes used to extract respectively hashtags, mentions and words

Media Information

Due to the fact that there was no interest in extracting the qualitative content of the images nor their meanings, the data extracted was not particularly complex or diverse. Crowdtangle saves the images and video contained by the post, the

size of the image (height and width), and the text it contains. Since it seemed unnecessary to have two dimensions for the size of an image, they were combined in the area of the media embedded in the post; the image text, on the other hand, was transformed as in the previous section by counting the number of words contained.

Instagram allows its user not only to share single images, but also a videos of various length or a mix media types; as a consequence, four one-hot encoded labels were added:

- `type_album`: a post containing a list of photos or videos of up to 10 elements
- `type_IGTV`: a post containing one single video of length longer than 30s
- `type_photo`: a post containing one single image
- `type_video`: a post containing one single video of length shorter than 30s

Popularity Information

Both as final dimensions and as intermediate steps to generate more complex attributes such as the volumetric ones, some pieces of information about popularity were extracted from each post: comments, favourites and followers, already prepared by crowdtangle, were used as previously said or as the objective variable for the regression. The historical evolution of the reactions to a post permitted us to define a new metric called *die_time*: *die_time*, not a new concept in literature, is a metric that aims at showing the amount of time passes before a post can be considered dead, that is, it does not show any more growth. The *die_time* was computed as follow: given an array of timestamps and the array of the reactions at those timestamps, the percentage increment of the reactions was computed between time t and $t + 1$, then, the first occurrence of a percentage below a certain threshold was taken to select the corresponding date; the difference between the selected date and the posting date, converted in hours, was identified as the amount of time for a post to die.

$$die_time_{hour} = Time[argmin(reaction_increment > threshold)] \quad (4.3)$$

The chosen threshold was 0.5% Another metric used to estimate popularity is the engagement: as mentioned in section 2.3.2, it can be computed as the ratio between the number of reactions of the post and the number of followers of the influencer at the time of the post's publishing.

$$engagement = \frac{reactions}{followers} \quad (4.4)$$

At last, the number of posts in the previous week and a binary attribute indicating the eventual sponsorship of an influencer were computed and extracted

4.4.5 Graph Topological Characteristics

The lack of data about the existence of categories of influencers, other than the ones depending on their popularity, in the Crowdtangle dataset appeared a lack of information that could be tackled smoothly and that, when fixed, could provide interesting insights and improve the performances of the regression.

It appeared a natural solution to this problem to create graphs of influencers and then use community detection algorithms to select categories of interest. Three graphs were generated:

- Hashtags graph: a graph with influencers as nodes and the use of common hashtags as edges between the nodes
- Mentions graph: a graph constructed with the influencers as nodes and the edges as the mention between them
- Followee graph: a graph constructed with the influencers as nodes and the edges as followee relation between them

On those graphs, 4 community detection algorithms were run: unfortunately, the results were highly inadequate: the only stable solutions were limit cases (e.g. one enormous category containing almost all the influencers while some other with one influencer) or, on the other hand, they were extremely unstable and depending on the run. As a consequence of those results, it was decided to avoid implementing them in the data pipeline and use the generated information.

4.5 Data Loading

After transforming the data, it was necessary to organise them for the regression methods. Due to the differences in input between the prediction models employed and to the interest in testing different use cases, four datasets were prepared:

- Multi-type single-model dataset: the dataset contained all media typologies and was not divided into quartiles depending on the number of followers in the previous week to train different models
- Single-type single-model dataset: the post content of this dataset was homogeneous, and one single model was trained indifferently from the number of followers in the previous week.
- Single-type multi-model dataset: the post content of this dataset was homogeneous, and different models were trained on four intervals of followers; a post was assigned to a different interval and, consequently, to a different model depending on its historical data from the previous week.

- Multi-type Multi-model dataset: the post content of this dataset was not homogeneous, and, as in the previous case, different models were trained on four intervals of followers.

The final part, except for saving the format into an easily readable file format that the following steps of the pipeline could interpret, was to prepare split them into train and test sets. It is common in the literature to perform this split randomly by sampling a fraction of the dataset to be used as the train set and the rest as the test set. While this procedure is an excellent solution to avoid splitting the dataset into two very pure subsets with significant differences from each other, in such a case where the temporal relationship between points has an extremely strong weight, this solution could be considerably dangerous.

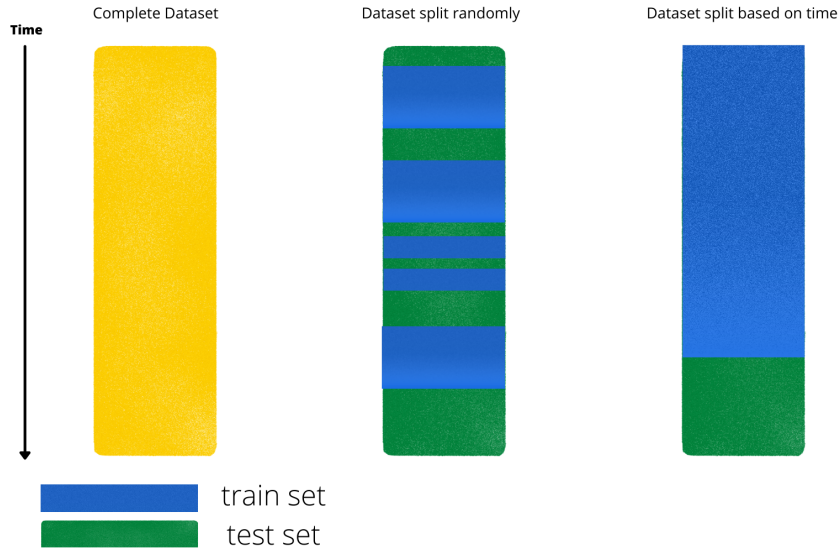


Figure 4.12: The classical method to split in train and test set the data vs the method used for this thesis

With this approach, in fact, given a random split, a point in the test set could be used as a feature in the train set and, more in general, the future, of the popularity of an influencer, could be used to train the learner to predict its past: it is then evident that a different approach had to be found. Given the fact that the data was nearly evenly distributed across the years, the first 5 years were taken as the train set, the remaining time (1 year) as the test set.

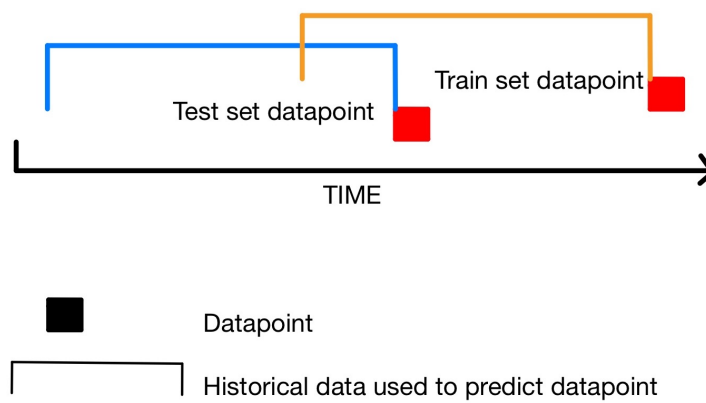


Figure 4.13: datapoints of the train set could contain information about the regression variable of the test set

Chapter 5

Training and Results

In this chapter, we will define the architecture of the regressors, their hyperparameters, the training step, and the results reached.

5.1 Evaluation Metrics

A set of evaluation metrics was chosen to assess the models and compare them with each other. Three, in particular, were chosen so that, used together, they could better describe the actual performances of the regressor:

- R2 score: also called coefficient of determination, is a measure of how well the model captures the variance of the data. The equation 5.1 can have a value that falls in the interval $(-\infty, 1]$, with one as the best result, 0 the outcome of a straight line equal to the mean of the samples.
- Mean Absolute Percentage Error: or MAPE, is a measure of how much percentage error is present between the predicted values and the actual values; computed as in 5.2, while quite valuable when dealing with regular distributions, in this case, where there are numerous outliers regarding the number of reactions it could be skewed.
- Median Absolute Percentage Error: also called MdAPE, tries to solve the problem of the low robustness to outliers of the MAPE thanks to the fact that it uses the median in place of the mean, as seen in 5.3; a significant drawback of this method is that median is not easily differentiated and, consequently, it cannot be used as loss for neural networks.

$$R2 = 1 - \frac{RSS}{TSS} \tag{5.1}$$

$$MAPE = \frac{1}{n} \sum \left| \frac{y_{true} - y_{predicted}}{y_{true}} \right| \quad (5.2)$$

$$MdAPE = median \left(\left| \frac{y_{true} - y_{predicted}}{y_{true}} \right| \right) \quad (5.3)$$

5.2 The Baseline

To better understand the regression task, a baseline was developed. An extremely simple linear regression was chosen as the so-called "stupid model".

5.2.1 Hyperparameters, Architecture and Training

The model learned on a simplified version of the dataframe having, as attributes, the average number of favourite reactions in the past week *first_week_favs* and the number of subscribers at the time of posting *subscriberCount* and as objective variable the likes to the given content. Given the lack of precise topological knowledge of the feature space a toy gridsearch was used to find the best combination of the basic hyperparamters, that is, the eventual normalisation and the fitting of the intercept.

The training was then run on the following hyperparameters:

- *fit_intercept* = *True*
- *normalize* = *False*
- *n_jobs* = *-1*
- *positive* = *False*

5.2.2 Results

Given the complexity of the problem the performances were unsurprisingly low:

- R2 Score: 0.21
- MAPE: 872%
- MdAPE: 74%

5.3 Classical Regression Methods

After assessing the baseline, the analysis proceeded by implementing classical regressor methods.

5.3.1 Hyperparameters, Architecture and Training

At first, a Gradient boosting regressor (described in 3.1.3) was used but then, given the lack of actual improvements with respect to predictions, a Random Forest Regressor, as in 3.1.2. The regressor was trained and tested on three datasets:

- Typed dataset: containing only one type of content, the post
- Complete dataset: containing the totality of the datapoints
- Quantiled dataset: five datasets, split according to which quantile a datapoint is in with respect to the past week average number of reactions

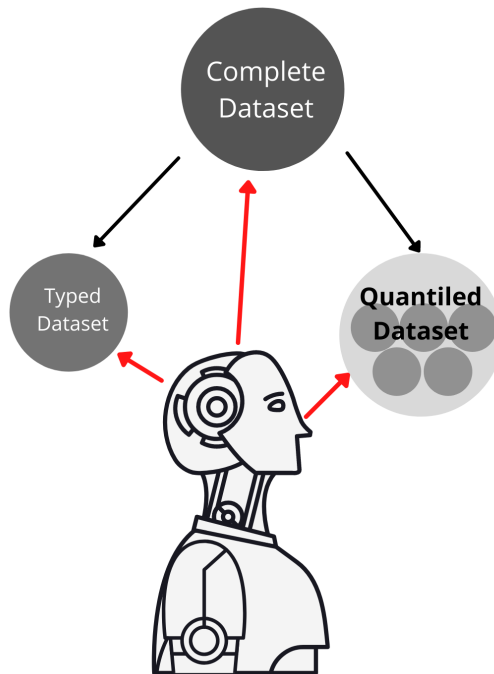


Figure 5.1: The model was trained on the three datasets

Given the discretely vast number of hyperparameters of the Random Forest Regressor algorithm, some tuning was necessary to perform the best possible training. Despite the powerful tools available, the largeness of the dataset made almost impossible to perform a suitable hyperparameter search on the totality of the data. The dataset was sampled to a fifth of its original size (both in the test and train set) and fed to a gridsearch method. GridsearchCV is a sklearn function used to perform an exhaustive search of the hyperparameters: a table, called parameters grid, of all the parameters, is given to the method that then

tries all the combinations on multiple folds, the best performing combination is then produced. After performing the search on the sampled dataset, the best hyperparameters were used to train the complete dataset model.

At first, a larger pool of hyperparameters was fed to the search, but after some trials the following set was used:

- *max_depth*: the maximum depth a tree can reach
- *max_leaf_nodes*: the maximum number of final nodes that a tree can have
- *max_features*: the maximum number of features that can be in the pool of the possible features used for the best split

The outcomes of the search, with respect to all three models, was:

	Max Depth	Max Leaf Nodes	Max Features
Model trained on Complete Dataset	5	None	Auto
Model trained on Typed Dataset	5	None	Auto
Model trained on Quantiled Dataset Q1	5	None	Auto
Model trained on Quantiled Dataset Q2	5	None	Auto
Model trained on Quantiled Dataset Q3	5	None	Auto
Model trained on Quantiled Dataset Q4	5	None	Auto
Model trained on Quantiled Dataset Q5	5	None	Auto

From the table 5.3.1 it is visible that, despite the difference in the dataset composition, the best hyperparameters are stable, the complete list in A.1.

5.3.2 Results

The regressor models were finally trained and tested on the datasets containing the data points of interest totality.

The results are positively interesting: for all the models, as guessed previously, the Mean Absolute Percentage Error appears to be the highest and least promising one due to the presence of outliers in the number of reactions to a post; if, for example, a post had a low number of reactions concerning the usual trends, e.g. for a disruption in the Instagram service a post by Fedez was seen by very few users and got only 100 reactions, then the forecasting would cause an error skewing the stats. A more interesting measure, for this reason, is then the MdAPE: in this case, we see how the error appears to be far more acceptable; it is, in fact, low enough to give us information on the trend and to proceed with meaningful insights about the user. At last, the R2 scores seem to show that the variance of the trends is acceptably explained by our models, with no remarkable differences between them, while still having room for improvements.

	MAPE	MdAPE	R2 Score
Model trained on Complete Dataset	258%	36%	0.75
Model trained on Typed Dataset	70%	36%	0.76
Model trained on Quantiled Dataset Q1	140%	35%	0.73
Model trained on Quantiled Dataset Q2	173%	36%	0.64
Model trained on Quantiled Dataset Q3	269%	37%	0.68
Model trained on Quantiled Dataset Q4	290%	37%	0.67
Model trained on Quantiled Dataset Q5	354%	32%	0.70

It is moreover of interest to try explaining the slight discrepancies between the models: not surprisingly, the quantiled datasets, being smaller, seem to show the least robustness to outliers, shown in both the MAPE and the R2 scores. finally, the typed datasets show a MAPE of half the magnitude: this is probably caused by the fact that videos seem to be the most difficult to forecast

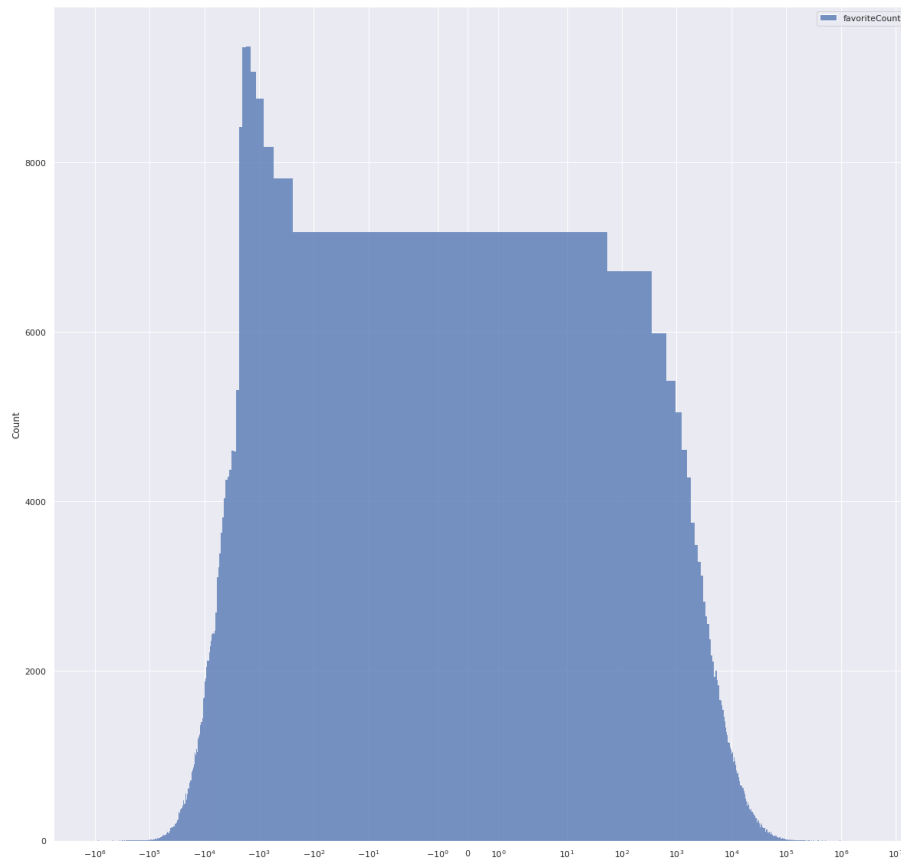


Figure 5.2: Histogram of the prediction absolute error

Furthermore, it is pretty interesting to view the distribution of the signed error as in 5.2: it is, in fact, visible a sudden peak in the negative part in the histogram of error. Such occurrence is probably caused by the fact that the regressor can hold the error as low as possible by keeping its prediction in the interval that goes between zero (the minimum possible) and what is considered the maximum: an average error of 100% is, as we have seen from previous experiments, a local minimum of discrete goodness and that can be reached easily by underestimating the prediction to zero (or, at least, to a very low value).

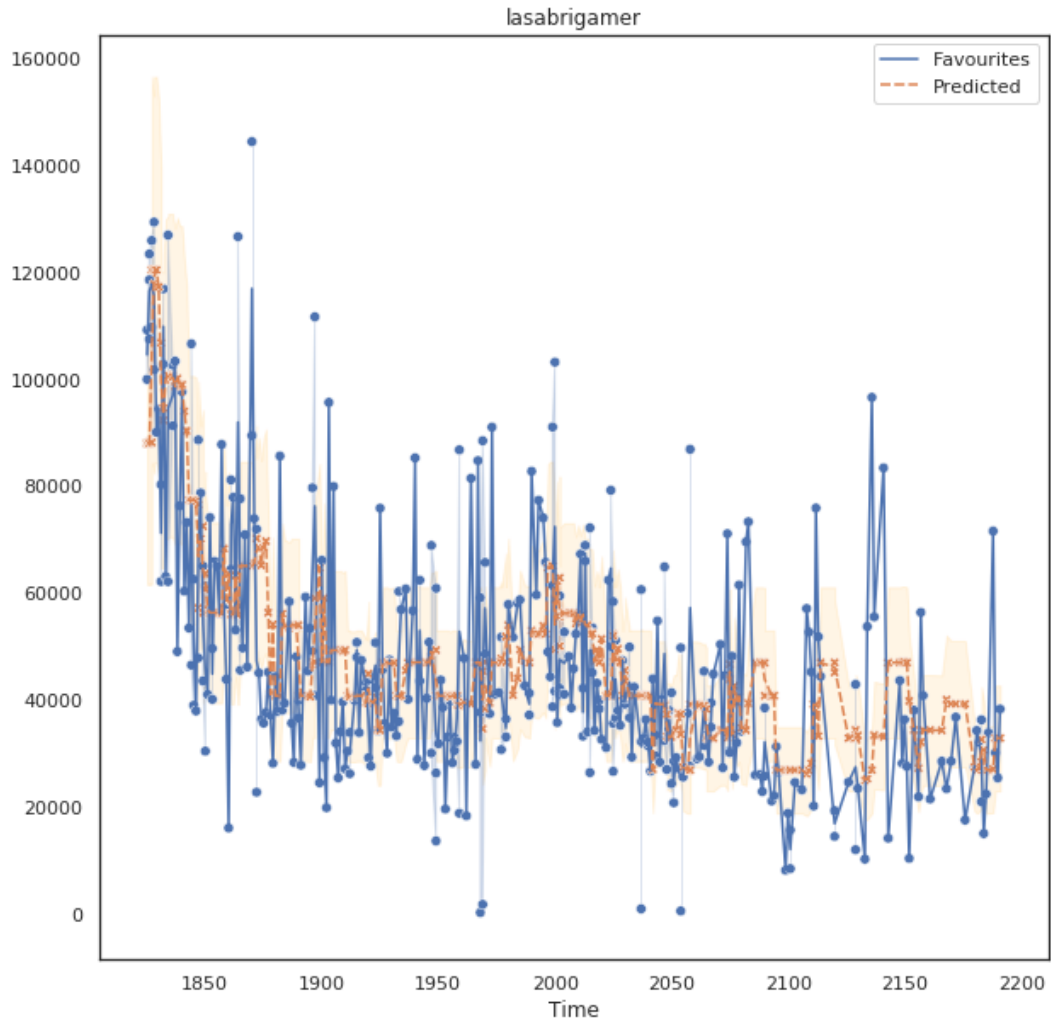


Figure 5.3: Prediction line (orange) vs actual values (blue). The orange shadow is the expected range of error

If we observed the predictions performed on the year 2020 for the account

lasabrigamer¹, a prediction that can be considered "average" in terms of accuracy², it is easily noticeable how the regression line appears to be the smoothed version of the actual line: peaks, both low and high, seem dampened. Such behaviour could be interpreted as a deficiency of the data to describe those characteristics of the post that make content exceptionally well (or bad) performing.

5.4 Neural Network Regression

Given the previous results, the doubt that the scores and the error were caused by a limit in the information contained in the data began to emerge. To confirm, or hopefully refute, such a hypothesis, the usage of a new, more complex regression method was needed: to do so, various Neural Network architectures were implemented and tested.

5.4.1 Hyperparameters, Architecture and training

Given the almost infinite possible architecture for a neural network and the actual complexity of tuning it, the process of creating it was done step by step:

- At first, a toy neural network of one single LSTM layer ingesting only the number of reactions of the previous post was implemented [a].
- Then, the width of the input was incremented to allow the network to use not only the number of reactions but also the followers and the comments of the previous post[b].
- In parallel, the length of the number of reactions, the temporal horizon, was increased up to ten steps in the past[c].
- After verifying that the previous implementations reached the baseline, the last two solutions were mixed to create a neural network that inputted the whole volumetric dimensions with a "memory" of 10 steps in the past[d].
- Finally, the remaining point attributes were used by means of a more complex architecture described in the following paragraph[e].

After identifying the general structure of the neural network, three possible architectures were thought of. The general structure was based on the idea that we had two very different types of data: timeseries, which we called volumetric

¹www.instagram.com/lasabrigamer/

² $R^2 : 0.28, MdAPE : 29\%, MAPE : 127\%$

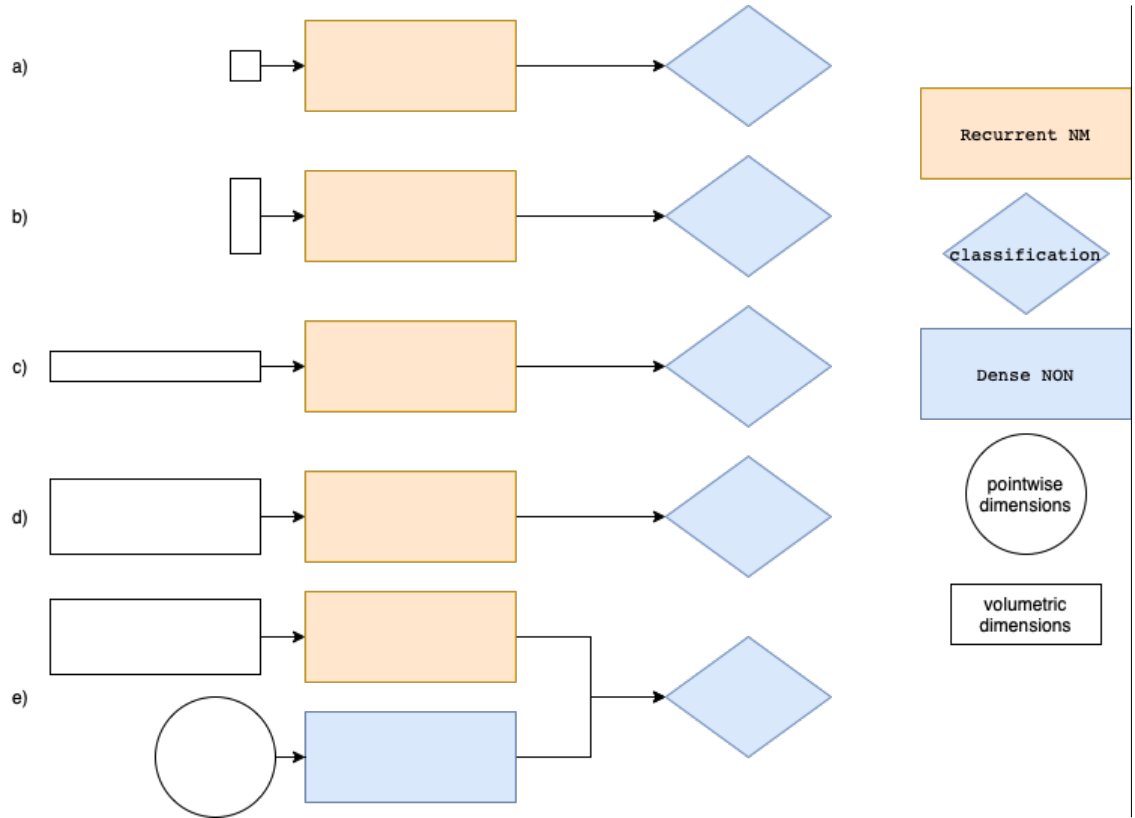


Figure 5.4: Consecutive steps in the generation of the general architecture of the neural network

attributes, describing the past performances of an influencer and scalar attributes, which described the characteristics of the post object of the forecast.

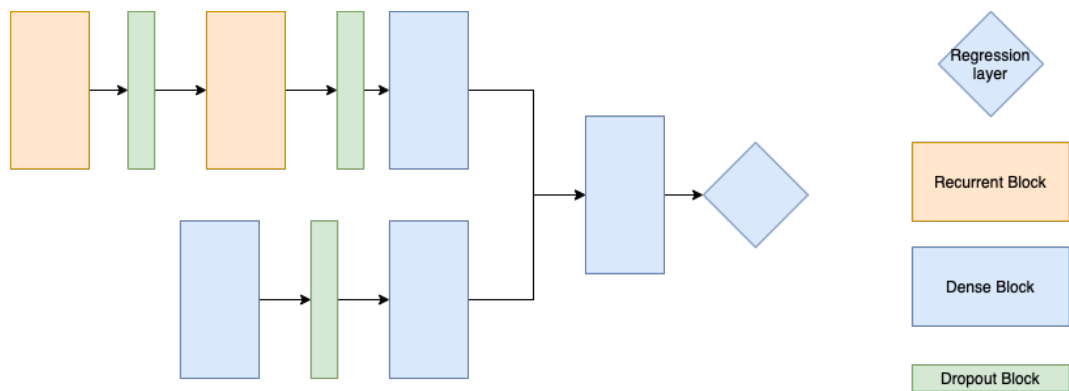


Figure 5.5: The final general architecture of the Neural Network

Due to the differences that were not only logical but also related to their representation (the first were arrays, the second numerical attributes), a dual legged architecture was conceived. The neural network was organised with two legs of similar length and with a similar design:

- The first leg was constructed with two recurrent blocks interrupted by dropout layers and having, at the end, one dense layer so to have a familiar interface with the following part.
- On the other hand, the second leg was more standard, being built with two blocks of dense layers interrupted by dropout.
- Finally, the outputs of the two legs were concatenated and combined by three plain dense layers.

Despite the variety of the architectures that were then actually used for the regression, the schema 5.5 was utilised as a guideline. The first Neural Network employed single LSTM layers, visible in fig 5.6, as recurrent blocks with a limited number of units per layer.

For the training phase, the Adam Optimizer with Exponential Learning rate decay was chosen with parameters as in A.2.

Given the non-satisfying performances of the previous solution, a strategy to perform the tuning of the hyperparameters, similar to the gridsearch for the random forest regressor, was needed. The choice fell on the Hyperband tuner offered by TensorFlow [60]. Hyperband is an optimization approach stemming from Bayesian optimization: differently from this method, which only uses probability to find the best performing model, Hyperband tries to early stop the least promising methods while only allowing the most promising configurations to "survive" until the best one is discovered.

Unfortunately, the computing power and time were not sufficient to complete an exhaustive exploration, and, as a consequence, the optimal results regarded only a modest part of the hyperparameter space.

Hyperparameter	Best Value So Far
units_lstm1	168
units_lstm2	168
units_lstm3 (dense)	28
units_dense_w1	40
units_dense_w2	24
units_dense_w3	26
units_dense_final1	10
units_dense_final2	6
learning_rate	0.0001

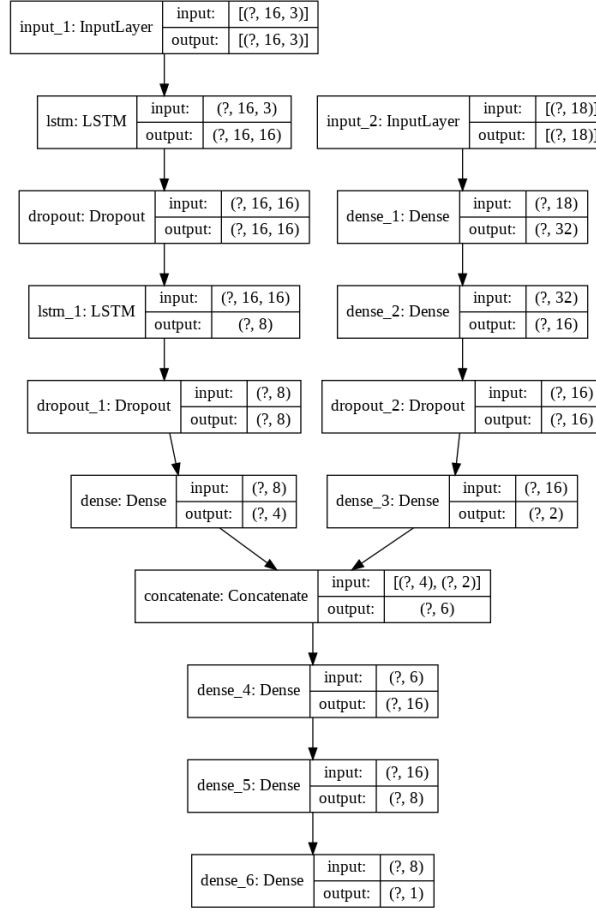


Figure 5.6: Structure of the first implementation of the actual neural network regressor

After assessing the relative goodness of the following architecture, the final three models, each with one bidirectional variant visible in A.3, were generated:

- pure LSTM model: the simplest version, similar to the version described in 5.6.
- pure GRU model: quite similar to the previous architecture, the LSTM layers were changed with GRU layers
- mixed LSTM-GRU model: the most complex one, in this case, after every LSTM layer, one GRU layer was appended.

After defining all these steps, the actual training was performed.

All the neural networks were trained for 200 hundred epochs with no early stopping since it appeared, from many runs, that the longest the training, the

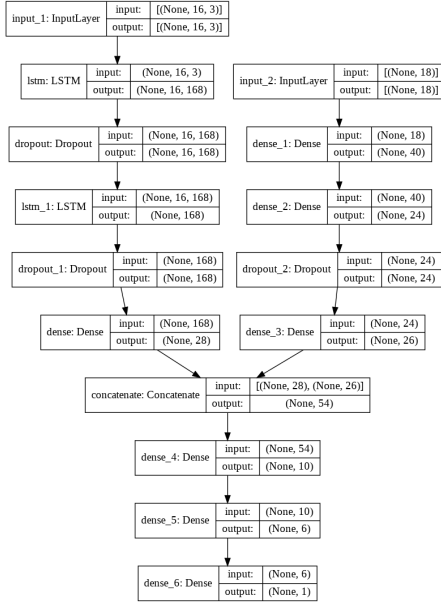


Figure 5.7: Pure LSTM Neural Network

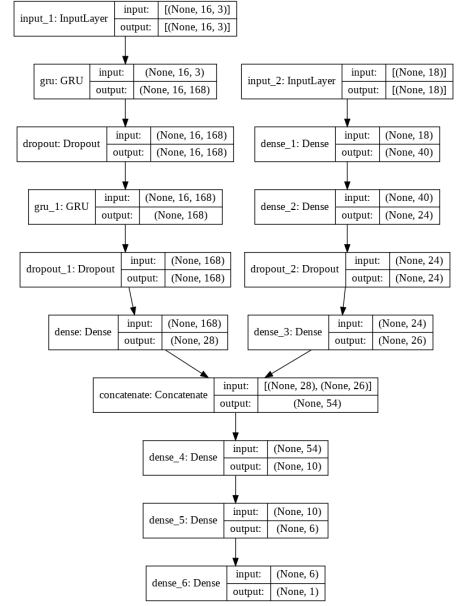


Figure 5.8: Pure GRU Neural Network

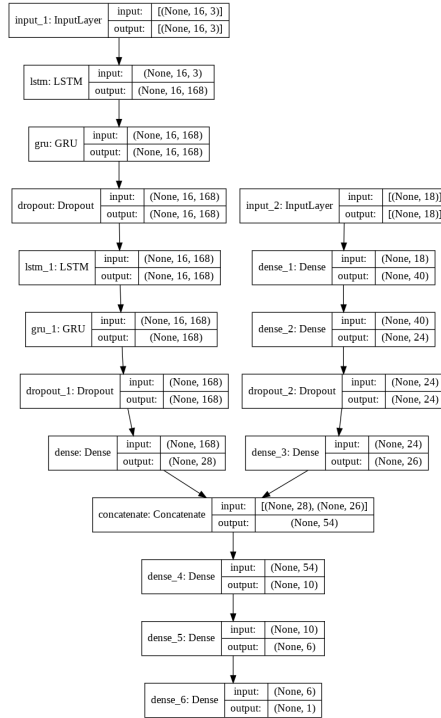


Figure 5.9: Mixed LSTM GRU Neural Network

Training Time	Quantiled Dataset	Typed Dataset	Complete Dataset
Unidirectional LSTM	3h 29min 19s	2h 46min 38s	3h 28min 9s
Bidirectional LSTM	6h 36min 36s	3h 54min 33s	4h 47min 45s
Unidirectional GRU	3h 12min 34s	2h 35min 22s	3h 26min 19s
Bidirectional GRU	6h 11min 45s	4h 1min 8s	4h 32min 16s
Unidirectional LSTM + GRU	4h 53min 13s	4h 0min 25s	5h 7min 24s
Bidirectional LSTM + GRU	7h 39min 53s	6h 22min 6s	4h 47min 45s

stabler the learning with no signs of overfitting. It is not excludable that a longer training time could have brought better results, but unfortunately this was not possible due to resource time constraints.

Unsurprisingly, the training times appear to vary widely depending on both the type of architecture and the size of the dataset.

5.4.2 Results

After a longer training phase with respect to classical machine learning methods, the results were gathered.

	R2	MdAPE	MAPE
Unidirectional LSTM	0.78	30.55%	61%
Bidirectional LSTM	0.78	31.67%	66.79%
Unidirectional GRU	0.78	30.60%	61.72%
Bidirectional GRU	0.78	31.63%	66.48%
Unidirectional GRU-LSTM	0.78	31.56%	62.22%
Bidirectional GRU-LSTM	0.78	30.62	62.22%

Table 5.1: Performances of on the typed dataset

At first, we observe the results with respect to the different datasets: not dissimilarly from the Classical Machine Learning Regressor, we notice that the dataset containing all types of content shows a slightly higher MdAPE and lower R2 score; the difference between the two datasets gets more visible when analyzing the MAPE: here, similarly to the Random Forest Regressor, the error rise up to more than 200%. As with such a prediction model, the reason is probably caused by the higher variance in the number of reactions for Video and IGTV content.

Comparing, on the other hand, the different regressors across the datasets, we notice that while R2 is almost indifferent, the most simple model, the unidirectional LSTM, is the best performing model; when compared with the Random Forest Regressor, we see even more difference: if the R2 score has an increment of only a couple of percentage points, the MdAPE seems to improve visibly, it goes, in fact,

	R2	MdAPE	MAPE
Unidirectional LSTM	0.77	33.80%	257.26%
Bidirectional LSTM	0.77	33.96%	260.06%
Unidirectional GRU	0.77	33.72%	256.18%
Bidirectional GRU	0.77	33.96%	260.18%
Unidirectional GRU-LSTM	0.77	33.96%	260.8%
Bidirectional GRU-LSTM	0.77	34.21%	262.86%

Table 5.2: Performances of on the Complete dataset

from 36% to 30%, an improvement that, while not shocking, seems to gets nearer to the limits of the data.

Finally, we move our attention to the regression on the quantiled dataset: as seen in 5.3, the results are pretty dissimilar to the ones given by the random forest regressor. In fact, in 5.3.2, the model seemed to perform far better with the smaller influencers (Q1 to Q4) concerning the R2 score; on the other hand, the MdAPE is visibly smaller for the most prominent influencers (Q4 and Q5). The largest quantile is the only one where the performances can be considered on par, or, for some usages, better than the ones of the classical regression methods: if, in fact, the R2 score appears to be slightly worse, the MdAPE is smaller of almost 10%.

Comparing the Neural Network Regression on the aforementioned dataset with the complete and typed dataset, we notice that there seem to be a worsening in almost all the metrics, except for the median error of the influencers comprised in the fifth quantile.

The predictions for a single influencer in 2020, in this case, the account **sbriser__**³ give us a view⁴ 5.10 similar to the one of the classical regression method: while the model forecasts the general trend, the peaks are challenging to foretell. That said, even for the single prediction, an increment in the R2 score is visible.

³www.instagram.com/sbriser__/

⁴0.40, *MdAPE* : 100%, *MAPE* : 108%

		R2	MdAPE	MAPE
Unidirectional LSTM	Q1	-	-	-
	Q2	0.08	42.63%	282.08%
	Q3	0.10	36.86%	224.87%
	Q4	0.12	31.12%	325.00%
	Q5	0.63	25.58%	201.07%
Bidirectional LSTM	Q1	-	-	-
	Q2	0.08	43.01%	287.88%
	Q3	0.10	36.58%	221.45%
	Q4	0.12	31.06%	328.40%
	Q5	0.64	25.97%	207.13%
Bidirectional LSTM	Q1	-	-	-
	Q2	0.08	42.17%	274.83%
	Q3	0.10	37.08%	227.57%
	Q4	0.12	31.04%	321.90%
	Q5	0.65	25.74%	201.47%
Bidirectional GRU	Q1	-	-	-
	Q2	0.08	42.47%	278.82%
	Q3	0.10	36.51%	220.97%
	Q4	0.12	31.53%	328.85%
	Q5	0.65	26.25%	209.42%
Unidirectional GRU-LSTM	Q1	-	-	-
	Q2	0.09	43.08%	290.01%
	Q3	0.10	37.19%	227.26%
	Q4	0.12	31.09%	320.97%
	Q5	0.67	25.70%	205.70%
Bidirectional GRU-LSTM	Q1	-	-	-
	Q2	0.08	42.34%	277.78%
	Q3	0.10	36.79%	221.94%
	Q4	0.12	31.06%	321.66%
	Q5	0.66	25.79%	205.05%

Table 5.3: Performances of on the quantiled dataset

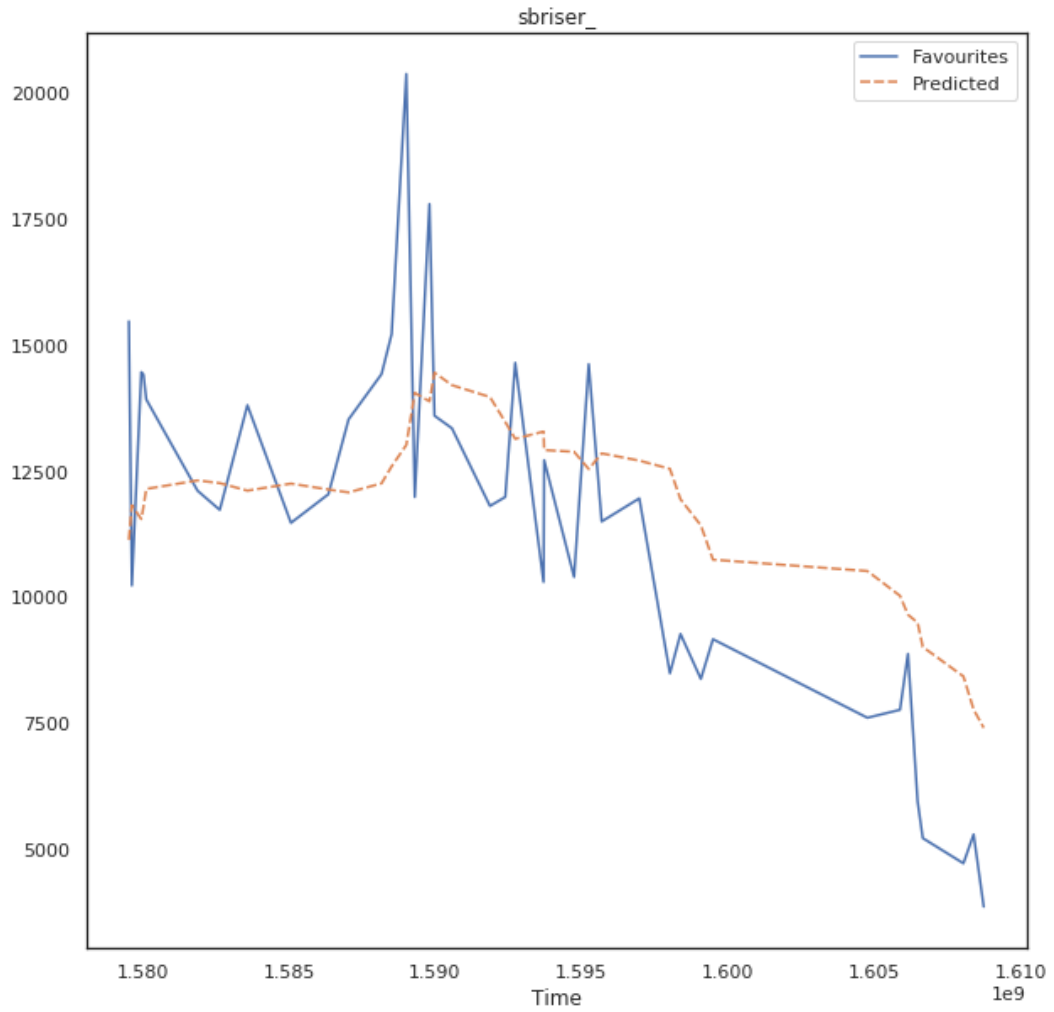


Figure 5.10: Prediction line (orange) vs actual values (blue). The orange shadow is the expected range of error

Chapter 6

Conclusions

In this thesis work, we presented the problem of forecasting the popularity trend on Online Social Networks. At first, we defined the praxises and characteristics of online presence, OSN and, more generally, online life with a short historical description of the past and present players in the market.

Then, we created a taxonomy able to better describe the variety of problems that fall in the category of OSN prediction: we established this classification on the basis of the endogenous or exogenous origin attributes and objective variables.

Later, we applied the proposed classification to subdivide the existing literature into categories and identify the shortcomings of other proposed solutions.

The work introduced in this thesis project proposes a new approach to the popularity forecasting problem based on the lack of knowledge about early performances and the quality of the content; the method employs the historical information and characteristics in control of the user at the moment of the posting with the intent to offer a method to be utilised before the posting of the content to forecast its performances and make informed decisions about its production.

To perform such research, the Instagram Crowdtangle Database by Facebook was used to produce a dataset of all the posts between 2015 and 2021 with the most descriptive non-qualitative attributes that could be extracted.

The transformed data were then fed to two regression methods, a Random Forest Regressor and a Neural Network, that produced promising results: both of them did not reach an accuracy such as to allow an exact prediction of the number of reactions of a post but, nonetheless, did produce a trend corresponding to the actual metrics. Moreover, such occurrence showed that, presumably, a limit in the information contained in the data is present and that knowledge on exogenous trends and events is probably needed.

6.1 Future Work

While done with the best of efforts, this thesis work could be extended and improved in several directions.

Different Evaluation Metrics - As repeated several times, the evaluation metrics used to train and tune the regression algorithms can only give a partial view of the actual performances of the regressor. A study on developing more relevant and robust metrics could improve the work proposed.

Addition of Exogenous Data - During the analysis of the database, we noticed that it is not uncommon for some influencers to have their online popularity influenced by "offline events", such as the participation in a tv program or contest; possessing a dataset containing the before-mentioned knowledge could bring to a better understanding of both the popularity trends and their burst.

Deeper Hyperparameter Tuning - While some exploration of the hyperparameters was performed, both time and computing power constraints did not allow us to perform an exhaustive search. A more profound research in these terms could indeed improve the learning capabilities of the solutions proposed.

Quality Information - It would be of interest, moreover, to enrich the current dataset with information regarding the content of the posts: knowledge about what is present in the images and videos, on the visual composition and, more in general, on the content quality could improve even more the prediction powers of the algorithms.

Appendix A

Complete List of Hyperparameters

A.1 Random Forest Regressor

```
1 bootstrap: True
2 ccp_alpha: 0.0
3 criterion: mse
4 max_depth: 5
5 max_features: auto
6 max_leaf_nodes: None
7 max_samples: None
8 min_impurity_decrease: 0.0
9 min_impurity_split: None
10 min_samples_leaf: 1
11 min_samples_split: 2
12 min_weight_fraction_leaf: 0.0
13 n_estimators: 100
14 n_jobs: -1
15 oob_score: False
16 random_state: None
17 verbose: 0
18 warm_start: False
```

A.2 Initial Dual-Legged Neural Network

```
1 z = LSTM(16, return_sequences=True, input_shape=ts_shape)(inputTS)
2 z = Dropout(0.5)(z)
3 z = LSTM(8, return_sequences=False)(z)
4 z = Dropout(0.5)(z)
5 z = Dense(4)(z)
6 z = Model(inputs=inputTS, outputs=z)
7 w = Dense(32, activation="relu")(inputP)
8 w = Dense(16, activation="relu")(w)
9 w = Dropout(0.5)(w)
10 w = Dense(2, activation="relu")(w)
11 w = Model(inputs=inputP, outputs=w)
12 combined = concatenate([z.output, w.output])
13 final = Dense(16)(combined)
14 final = Dense(8)(final)
15 out = Dense(1, activation="linear")(final)
16
17 initial_learning_rate = 1e-3
18 lr_schedule = tf.keras.optimizers.schedules.ExponentialDecay(
19     initial_learning_rate,
20     decay_steps=10000,
21     decay_rate=0.9,
22     staircase=True)
23
24 opt = Adam(learning_rate=lr_schedule)
```

A.3 Bidirectional Variants

A.4 Histograms of Text Metrics

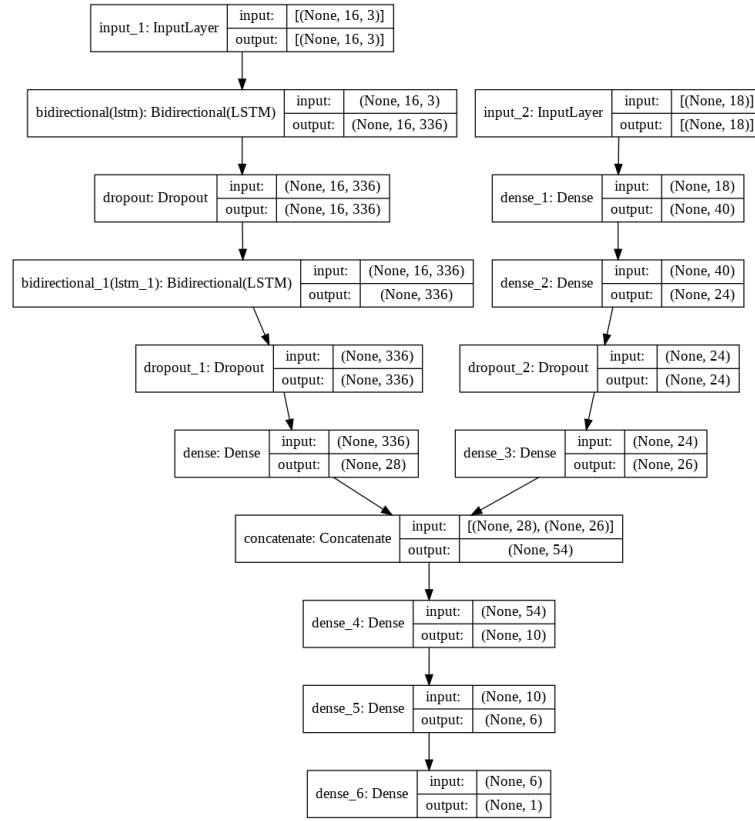


Figure A.1: Bidirectional LSTM

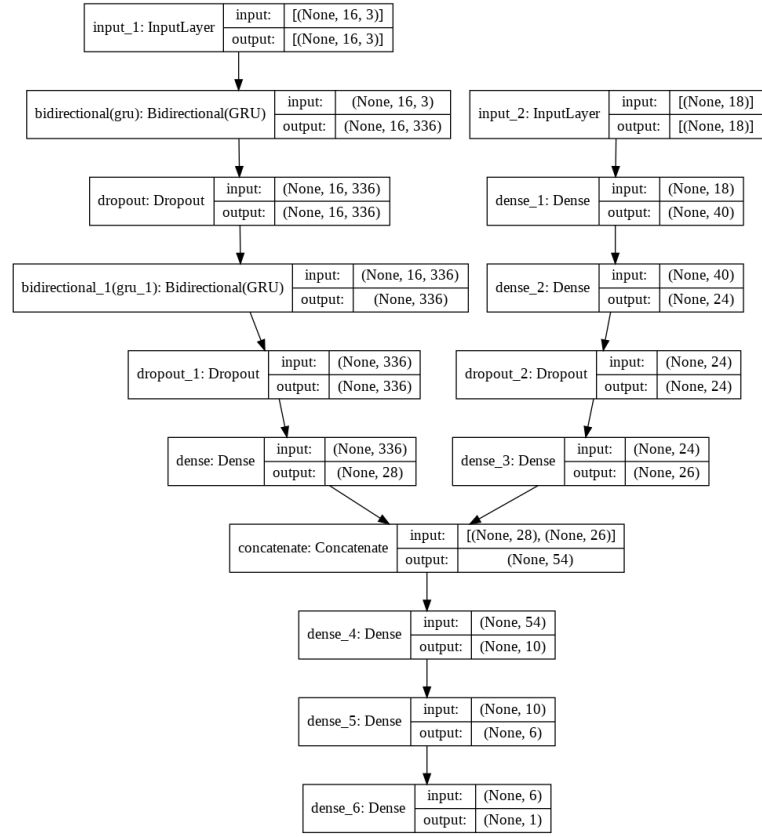


Figure A.2: Bidirectional LSTM

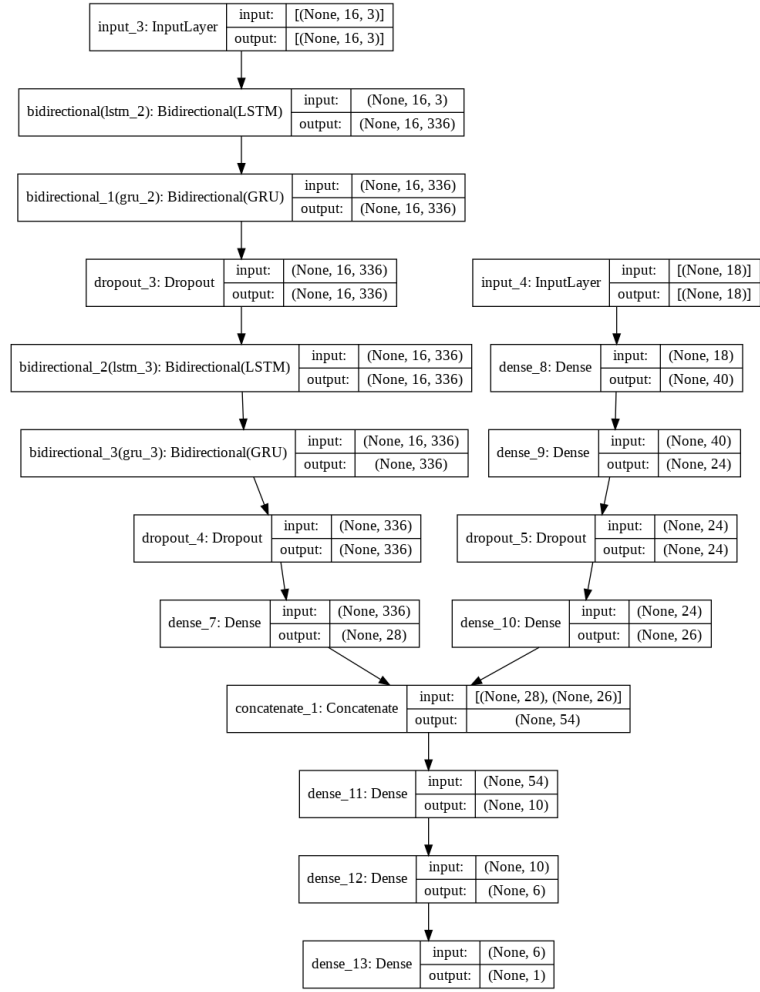
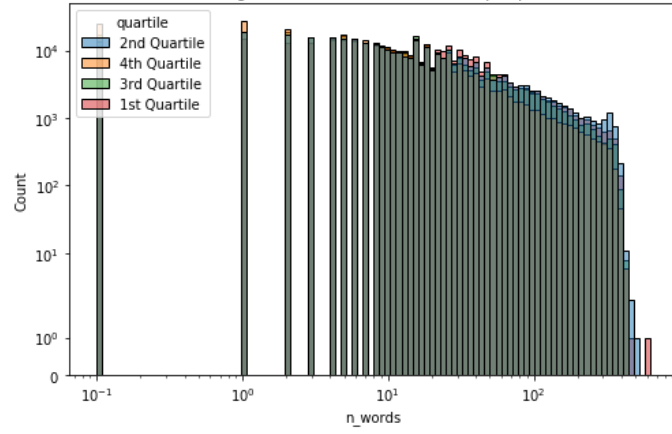
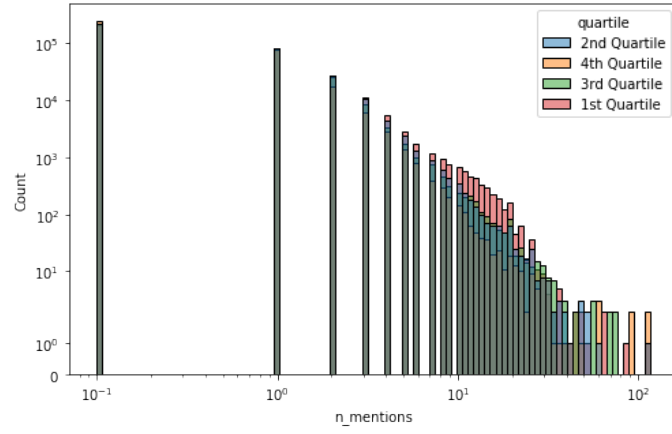


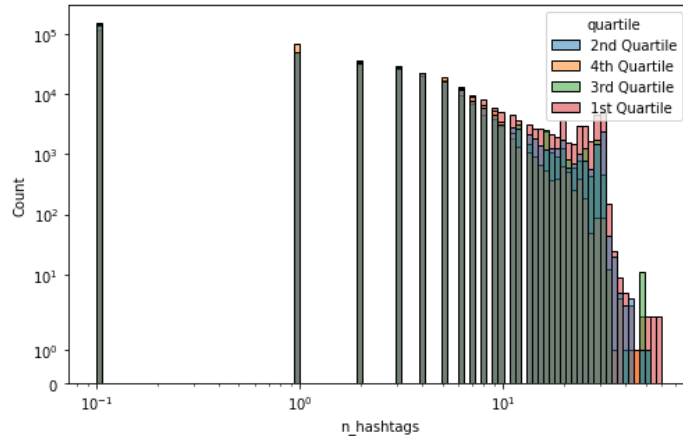
Figure A.3: Bidirectional LSTM



(a) words



(b) mentions



(c) hashtags

Figure A.4: histograms of the description metrics regarding posts

Bibliography

- [1] Luciano Floridi, ed. *The Onlife Manifesto*. Springer International Publishing, 2015. DOI: 10.1007/978-3-319-04093-6. URL: <https://doi.org/10.1007/978-3-319-04093-6> (cit. on pp. 1, 7).
- [2] Michael Ray. *Social network*. June 2019. URL: <https://www.britannica.com/technology/social-network> (cit. on p. 2).
- [3] *Number of people using social media platforms*. URL: <https://ourworldindata.org/grapher/users-by-social-media-platform?time=2018> (cit. on p. 2).
- [4] Danah M. Boyd and Nicole B. Ellison. «Social Network Sites: Definition, History, and Scholarship». In: *Journal of Computer-Mediated Communication* 13.1 (Oct. 2007), pp. 210–230. ISSN: 1083-6101. DOI: 10.1111/j.1083-6101.2007.00393.x. eprint: <https://academic.oup.com/jcmc/article-pdf/13/1/210/22316979/jjcmcom0210.pdf>. URL: <https://doi.org/10.1111/j.1083-6101.2007.00393.x> (cit. on p. 3).
- [5] Published by Statista Research Department. *Most used social media 2021*. Sept. 2021. URL: <https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/> (cit. on p. 3).
- [6] Danah Boyd. «Why Youth Heart Social Network Sites: The Role of Networked Publics in Teenage Social Life». In: (2017). DOI: 10.31219/osf.io/22hq2 (cit. on p. 3).
- [7] Chance Miller. *These were the most-downloaded apps and games of the decade*. Dec. 2019. URL: <https://9to5mac.com/2019/12/16/apps-and-games-of-the-decade/> (cit. on p. 4).
- [8] Koosha Zarei, Damilola Ibosiola, Reza Farahbakhsh, Zafar Gilani, Kiran Garimella, Noël Crespi, and Gareth Tyson. «Characterising and Detecting Sponsored Influencer Posts on Instagram». In: *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. 2020, pp. 327–331. DOI: 10.1109/ASONAM49781.2020.9381309 (cit. on p. 5).

- [9] Kaya Ismail. *Social Media Influencers: Mega, Macro, Micro or Nano*. Dec. 2018. URL: <https://www.cmswire.com/digital-marketing/social-media-influencers-mega-macro-micro-or-nano/> (cit. on p. 5).
- [10] Mohamed Ahmed, Stella Spagna, Felipe Huici, and Saverio Niccolini. «A Peek into the Future: Predicting the Evolution of Popularity in User Generated Content». In: *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*. WSDM '13. Rome, Italy: Association for Computing Machinery, 2013, pp. 607–616. ISBN: 9781450318693. DOI: 10.1145/2433396.2433473. URL: <https://doi.org/10.1145/2433396.2433473> (cit. on p. 8).
- [11] Ying Hu, Changjun Hu, Shushen Fu, Mingzhe Fang, and Wenwen Xu. «Predicting Key Events in the Popularity Evolution of Online Information». In: *PLOS ONE* 12.1 (Jan. 2017), pp. 1–21. DOI: 10.1371/journal.pone.0168749. URL: <https://doi.org/10.1371/journal.pone.0168749> (cit. on pp. 8, 9).
- [12] Hai Yu, Ying Hu, and Peng Shi. «A Prediction Method of Peak Time Popularity Based on Twitter Hashtags». In: *IEEE Access* 8 (2020), pp. 61453–61461. DOI: 10.1109/ACCESS.2020.2983583 (cit. on p. 9).
- [13] Roja Bandari, Sitaram Asur, and Bernardo Huberman. «The Pulse of News in Social Media: Forecasting Popularity». In: *Proceedings of the International AAAI Conference on Web and Social Media* 6.1 (Aug. 2021), pp. 26–33. URL: <https://ojs.aaai.org/index.php/ICWSM/article/view/14261> (cit. on p. 9).
- [14] Ping Liu, Joshua Guberman, Libby Hemphill, and Aron Culotta. «Forecasting the presence and intensity of hostility on Instagram using linguistic and social features». In: *CoRR* abs/1804.06759 (2018). arXiv: 1804.06759. URL: <http://arxiv.org/abs/1804.06759> (cit. on pp. 10, 11).
- [15] Yevgeniy Golovchenko, Cody Buntain, Gregory Eady, Megan A. Brown, and Joshua A. Tucker. «Cross-Platform State Propaganda: Russian Trolls on Twitter and YouTube during the 2016 U.S. Presidential Election». In: *The International Journal of Press/Politics* 25.3 (Apr. 2020), pp. 357–389. DOI: 10.1177/1940161220912682. URL: <https://doi.org/10.1177/1940161220912682> (cit. on p. 10).
- [16] Derek Weber and Frank Neumann. «Who’s in the Gang? Revealing Coordinating Communities in Social Media». In: *CoRR* abs/2010.08180 (2020). arXiv: 2010.08180. URL: <https://arxiv.org/abs/2010.08180> (cit. on pp. 10, 11).
- [17] Derek Weber, Mehwish Nasim, Lewis Mitchell, and Lucia Falzon. «A method to evaluate the reliability of social media data for social network analysis». In: *CoRR* abs/2010.08717 (2020). arXiv: 2010.08717. URL: <https://arxiv.org/abs/2010.08717> (cit. on p. 11).

- [18] Randall Wald, Taghi Khoshgoftaar, and Chris Sumner. «Machine prediction of personality from Facebook profiles». In: *2012 IEEE 13th International Conference on Information Reuse Integration (IRI)*. 2012, pp. 109–115. DOI: 10.1109/IRI.2012.6302998 (cit. on p. 12).
- [19] Matteo Cardaioli, Pallavi Kaliyar, Pasquale Capuozzo, Mauro Conti, Giuseppe Sartori, and Merylin Monaro. «Predicting Twitter Users’ Political Orientation: An Application to the Italian Political Scenario». In: *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. 2020, pp. 159–165. DOI: 10.1109/ASONAM49781.2020.9381470 (cit. on pp. 12, 13).
- [20] Achilleas Psyllidis, Jie Yang, and Alessandro Bozzon. «Regionalization of Social Interactions and Points-of-Interest Location Prediction With Geosocial Data». In: *IEEE Access* 6 (2018), pp. 34334–34353. DOI: 10.1109/ACCESS.2018.2850062 (cit. on pp. 12, 14).
- [21] Liang Zhao, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. «Spatiotemporal Event Forecasting in Social Media». In: *Proceedings of the 2015 SIAM International Conference on Data Mining (SDM)*, pp. 963–971. DOI: 10.1137/1.9781611974010.108. eprint: <https://epubs.siam.org/doi/pdf/10.1137/1.9781611974010.108>. URL: <https://epubs.siam.org/doi/abs/10.1137/1.9781611974010.108> (cit. on p. 13).
- [22] Jonas Krauss, S. Nann, Daniel Simon, Kai Fischbach, and Peter Gloor. «Predicting Movie Success and Academy Awards through Sentiment and Social Network Analysis». In: June 2008 (cit. on p. 14).
- [23] Martino Trevisan, Luca Vassio, Idilio Drago, Marco Mellia, Fabricio Murai, Flavio Figueiredo, Ana Paula Couto da Silva, and Jussara M Almeida. «Towards Understanding Political Interactions on Instagram». In: *Proceedings of the 30th ACM Conference on Hypertext and Social Media*. 2019 (cit. on pp. 14, 15).
- [24] Carlos Henrique Gomes Ferreira, Fabricio Murai, Ana Paula Couto da Silva, Jussara Marques de Almeida, Martino Trevisan, Luca Vassio, Idilio Drago, and Marco Mellia. «Unveiling Community Dynamics on Instagram Political Network». In: *ACM Conference on Web Science*. 2020 (cit. on p. 15).
- [25] Carlos H.G. Ferreira, Fabricio Murai, Ana P.C. Silva, Jussara M. Almeida, Martino Trevisan, Luca Vassio, Marco Mellia, and Idilio Drago. «On the dynamics of political discussions on Instagram: A network perspective». In: *Online Social Networks and Media* 25 (2021), p. 100155. ISSN: 2468-6964. DOI: <https://doi.org/10.1016/j.osnem.2021.100155> (cit. on p. 15).

- [26] Luca Vassio, Michele Garetto, Carla Chiasserini, and Emilio Leonardi. «User Interaction with Online Advertisements: Temporal Modeling and Optimization of Ads Placement». In: *Association for Computing Machinery Journal.2* (2020). ISSN: 2376-3639. DOI: 10.1145/3377144 (cit. on p. 16).
- [27] Fabio Bertone, Luca Vassio, and Martino Trevisan. «The Stock Exchange of Influencers: A Financial Approach for Studying Fanbase Variation Trends». In: *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. 2021 (cit. on pp. 16, 17).
- [28] Martino Trevisan, Luca Vassio, and Danilo Giordano. «Debate on online social networks at the time of COVID-19: An Italian case study». In: *Online Social Networks and Media* 23 (2021), p. 100136. ISSN: 2468-6964. DOI: <https://doi.org/10.1016/j.osnem.2021.100136> (cit. on pp. 16, 17).
- [29] Luca Vassio, Michele Garetto, Carla Chiasserini, and Emilio Leonardi. «Temporal Dynamics of Posts and User Engagement of Influencers on Facebook and Instagram». In: *2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. 2021. DOI: 10.1145/3487351.3488340 (cit. on pp. 17, 18).
- [30] Roy Ling Hang Yew, Syamimi Binti Suhaidi, Prishtee Seewoochurn, and Venantius Kumar Sevamalai. «Social Network Influencers' Engagement Rate Algorithm Using Instagram Data». In: *2018 Fourth International Conference on Advances in Computing, Communication Automation (ICACCA)*. 2018, pp. 1–8. DOI: 10.1109/ICACCAF.2018.8776755 (cit. on p. 23).
- [31] Arry Akhmad Arman and Agus Pahrul Sidik. «Measurement of Engagement Rate in Instagram (Case Study: Instagram Indonesian Government Ministry and Institutions)». In: *2019 International Conference on ICT for Smart Society (ICISS)*. Vol. 7. 2019, pp. 1–6. DOI: 10.1109/ICISS48059.2019.8969826 (cit. on p. 23).
- [32] Jeffrey M. Stanton. «Galton, Pearson, and the Peas: A Brief History of Linear Regression for Statistics Instructors». In: *Journal of Statistics Education* 9.3 (2001), null. DOI: 10.1080/10691898.2001.11910537. eprint: <https://doi.org/10.1080/10691898.2001.11910537>. URL: <https://doi.org/10.1080/10691898.2001.11910537> (cit. on p. 28).
- [33] David Freedman. *Statistical models theory and practice*. Cambridge University Press, 2009 (cit. on p. 29).
- [34] Hieu Tran. *Survey of Machine Learning and Data Mining Techniques used in Multimedia System*. Sept. 2019. DOI: 10.13140/RG.2.2.20395.49446/1 (cit. on p. 29).

- [35] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning with applications in R*. Springer, 2021 (cit. on p. 29).
- [36] Victor Rodriguez-Galiano, Manuel Sánchez Castillo, Jadunandan Dash, Peter Atkinson, and Jose Ojeda-Zujar. «Modelling interannual variation in the spring and autumn land surface phenology of the European forest». In: *Biogeosciences* 13 (June 2016), pp. 3305–3317. DOI: 10.5194/bg-13-3305-2016 (cit. on p. 30).
- [37] Ankit Chauhan. *ENSEMBLE METHODS - Bagging, Boosting, and Stacking*. Feb. 2021. URL: <https://medium.com/analytics-vidhya/ensemble-methods-bagging-boosting-and-stacking-28d006708731> (cit. on p. 30).
- [38] Jason Brownlee. *A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning*. Aug. 2020. URL: <https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/> (cit. on p. 31).
- [39] By: IBM Cloud Education. *What is Gradient Descent?* URL: <https://www.ibm.com/cloud/learn/gradient-descent> (cit. on p. 31).
- [40] Ivanna Baturynska and Kristian Martinsen. «Prediction of geometry deviations in additive manufactured parts: comparison of linear regression with machine learning algorithms». In: *Journal of Intelligent Manufacturing* 32 (Jan. 2021). DOI: 10.1007/s10845-020-01567-0 (cit. on p. 32).
- [41] *Random Forests and Boosting in MLlib*. Aug. 2020. URL: <https://databricks.com/blog/2015/01/21/random-forests-and-boosting-in-mllib.html> (cit. on p. 31).
- [42] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 3rd ed. Prentice Hall, 2010 (cit. on p. 32).
- [43] Filippo Masi, Ioannis Stefanou, Paolo Vannucci, and Victor Maffi-Berthier. «Thermodynamics-based Artificial Neural Networks for constitutive modeling». In: *Journal of the Mechanics and Physics of Solids* 147 (Feb. 2021). DOI: 10.1016/j.jmps.2020.104277 (cit. on p. 33).
- [44] Ashkan Eliasy and Justyna Przychodzen. «The role of AI in capital structure to enhance corporate funding strategies». In: *Array* 6 (July 2020), p. 100017. DOI: 10.1016/j.array.2020.100017 (cit. on p. 33).
- [45] Stichelen Malard. *Fully Connected (Dense)*¶. URL: <https://epynn.net/Dense.html> (cit. on p. 34).
- [46] Aurelien Geron. *Hands-on machine learning with scikit-learn, keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. 2nd ed. O'Reilly Media, 2019. ISBN: 9781492032649 (cit. on p. 34).

- [47] Xuan Hien Le, Hung Ho, Giha Lee, and Sungho Jung. «Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting». In: *Water* 11 (July 2019), p. 1387. DOI: 10.3390/w11071387 (cit. on p. 35).
- [48] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. 2014. arXiv: 1412.3555 [cs.NE] (cit. on pp. 35, 36).
- [49] François Chollet et al. *Keras*. <https://keras.io>. 2015 (cit. on pp. 36, 42).
- [50] Mike Schuster and Kuldeep Paliwal. «Bidirectional recurrent neural networks». In: *Signal Processing, IEEE Transactions on* 45 (Dec. 1997), pp. 2673–2681. DOI: 10.1109/78.650093 (cit. on p. 36).
- [51] Wikipedia contributors. *Bidirectional recurrent neural networks* — *Wikipedia, The Free Encyclopedia*. [Online; accessed 15-October-2021]. 2021. URL: https://en.wikipedia.org/w/index.php?title=Bidirectional_recurrent_neural_networks&oldid=1024058341 (cit. on p. 36).
- [52] Victoria Hodge and Jim Austin. «A Survey of Outlier Detection Methodologies». In: 22.2 (Oct. 2004), pp. 85–126. DOI: 10.1023/b:aire.00000445502.10941.a9. URL: <https://doi.org/10.1023/b:aire.00000445502.10941.a9> (cit. on p. 37).
- [53] F. Pedregosa et al. «Scikit-learn: Machine Learning in Python». In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (cit. on pp. 37, 42).
- [54] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. «Isolation Forest». In: *2008 Eighth IEEE International Conference on Data Mining*. 2008, pp. 413–422. DOI: 10.1109/ICDM.2008.17 (cit. on p. 37).
- [55] *Detecting and Treating Outliers: How to Handle Outliers*. May 2021. URL: <https://www.analyticsvidhya.com/blog/2021/05/detecting-and-treating-outliers-treating-the-odd-one-out/> (cit. on p. 38).
- [56] *Spark Overview*. URL: <https://spark.apache.org/docs/latest/> (cit. on p. 40).
- [57] Shrey Grade Akash. Mehrotra. *APACHE SPARK QUICK START GUIDE: quickly learn the art of writing efficient big data applications ... with apache spark*. PACKT Publishing Limited, 2019 (cit. on p. 41).
- [58] The Pandas Development Team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. DOI: 10.5281/zenodo.3509134. URL: <https://doi.org/10.5281/zenodo.3509134> (cit. on p. 42).
- [59] The TensorFlow Team. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: <https://www.tensorflow.org/> (cit. on p. 42).

- [60] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. *Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization*. 2018. arXiv: 1603.06560 [cs.LG] (cit. on p. 63).