





Article

# Popularity Prediction of Instagram Posts

Salvatore Carta<sup>1</sup>, Alessandro Sebastian Podda\*,<sup>1</sup>, Diego Reforgiato Recupero<sup>1</sup>, Roberto Saia<sup>1</sup>, and Giovanni Usai<sup>1</sup>

<sup>1</sup> Department of Mathematics and Computer Science, University of Cagliari, 09124 Cagliari, Italy.

\* Correspondence: Alessandro Sebastian Podda (sebastianpodda@unica.it).

‡ Authors contributed equally to this work.

**Abstract:** Predicting the popularity of posts on social networks has taken on significant importance in recent years, and several social media management tools now offer solutions to improve and optimize the quality of published content and to enhance the attractiveness of companies and organizations. Scientific research has recently moved in this direction, with the aim of exploiting advanced techniques such as machine learning, deep learning, natural language processing, etc., to support such tools. In light of the above, in this work we aim to address the challenge of predicting the popularity of a future post on Instagram, by defining the problem as a classification task and by proposing an original approach based on Gradient Boosting and feature engineering, which led us to promising experimental results. The proposed approach exploits big data technologies for scalability and efficiency and it is general enough to be applied to other social media as well.

**Keywords:** Popularity Prediction; Classification; Social Network; Machine Learning; Instagram.

## 1. Introduction

Nowadays, the social network market grows both in number of operators and in number of posts, exponentially. With millions of monthly active users, both on mobile devices and web browsers, Instagram represents a leading platform in this market. Launched in 2010, it has gradually gained a leading role among photo-sharing platforms, introducing several innovative features over times, including – not exhaustively – *filters*, *stories*, and an internal messaging system. Recently, these features have attracted not only ordinary users and photography enthusiasts, but also companies, organizations and global brands, thanks to the possibility that Instagram has offered to explore new business models and marketing strategies.

In such a context, approaches aimed to predict the popularity of social content are gaining increasing attention, not only from a commercial and industrial standpoint, but also from a scientific perspective. Indeed, thanks to the advancement of knowledge in the fields of data analysis and artificial intelligence, together with the development of new techniques based on Machine/Deep learning, Natural Language Processing (NLP), Data mining, Big Data, etc. [1–3], it is now possible to provide advanced tools for companies and individuals, and promote the consolidation of these businesses in the social market. To this purpose, such tools and techniques usually aim to extract hidden information that may be exploited in several directions, spanning from targeted advertisements to political strategies.

Within this context, the present work proposes a novel approach for predicting the future popularity of Instagram posts. In particular, the existing literature often focuses on the prediction of the so-called *engagement factor* (i.e., the ratio between expected likes and number of followers of the account), thus addressing a regression problem. Conversely, this paper aims to determine whether, for the post to be published, the deviation from the average like of recent posts will be positive

35 or negative, therefore solving a binary classification task. More precisely, our method predicts the  
36 expected popularity class regardless of the selected media (image or video), which we assume as  
37 fixed by the user: vice versa, we analyze the metadata associated with the post (e.g., the caption, the  
38 chosen hashtags, the expected time and date of publication, the used emojis), as well as the additional  
39 information related to the account and the popularity of recent posts.

40 Therefore, the main contributions of this work are:

- 41 1. an original formulation of the problem to address, by defining it as a binary classification task,  
42 with the aim of determining whether the popularity of a future post will increase or decrease  
43 compared to the recent account average;
- 44 2. a novel approach based on feature engineering and machine learning techniques for the  
45 prediction of the expected popularity class of posts on Instagram (although the approach is  
46 general and applicable to other social networks);
- 47 3. an experimental evaluation of the approach, comparing the results against a set of strong  
48 baselines, in different scenarios (obtained through an extensive exploration of the problem  
49 parameters);
- 50 4. a big data infrastructure that we have leveraged to run the proposed approach and that makes it  
51 scalable and flexible;
- 52 5. an analysis of the execution performance of the proposed algorithm, by considering a distributed  
53 cloud deployment, and by exploiting the aforementioned big data infrastructure.

54 The remainder of this paper is then organized as it follows. In Section 2, we present some  
55 background notions related to the adopted machine learning techniques, and a wide overview of the  
56 related work. Then, in Section 3 we outline the problem to be addressed and describe it in a formal way.  
57 Section 4 contains the procedure adopted for collecting the data and building the dataset, whereas in  
58 Section 5 we describe in detail the proposed approach. Finally, in Section 6 we illustrate the results of  
59 the experimental evaluation, and in Section 7 we conclude the work, indicating some possible future  
60 developments.

## 61 2. Background and Related Work

62 This section provides a background on the techniques and methods exploited in this work, along  
63 with an overview and some representative examples of state-of-the-art works focused on this research  
64 field.

### 65 2.1. Machine Learning Algorithms

66 On the basis of the target strategy, the type of involved input/output data, and type of problem  
67 to face, the existing literature proposes several types of machine learning algorithms (e.g., supervised  
68 learning, unsupervised learning, semi-supervised learning, reinforcement learning, self learning,  
69 feature learning, etc.). However, the most adopted classification essentially comprises three main  
70 groups: *supervised*, *unsupervised*, and *semi-supervised* [4,5].

71 Intuitively, in the supervised case, the learning is achieved by leveraging on a previous knowledge  
72 of the possible output value for each sample. The objective is hence to learn a function able to  
73 approximate the relationship between the input and the output. More formally, given an instance  
74 space  $X$  and a label space  $Y$ , let us suppose there exists a target function  $y = f(x)$ , with  $x \in X$  and  
75  $y \in Y$ , able to map each  $x$  to the correct  $y$ : the supervised algorithm tries to generalize the function  $f(x)$ ,  
76 by observing a (possibly large) set of  $(x, y)$  sample pairs. Supervised learning can thus address both  
77 regression [6] and classification [7] problems: the first case, when the output variable  $y$  is numerical (or  
78 continuous); the second case, when the output variable  $y$  is categorical (or discrete). For these reasons,  
79 supervised algorithms have been widely exploited in literature: some example are represented by  
80 Random Forests [8], Support Vector Machines [9], Gradient Boosting [10], Neural Networks [11].

81 On the other hand, in the unsupervised learning, labeled outputs are not necessary for the training  
82 phase. Indeed, unsupervised algorithms try to detect some hidden patterns on the input data, in order  
83 to find a structure in them. They are mainly grouped in clustering algorithms [12] (that aggregate data  
84 into similar sets, according to one or more defined metrics) and association algorithms [13] (that search  
85 for rules able to describe large portions of the given data). The two most common techniques, in this  
86 category, are the K-means [14] for clustering tasks, and the Apriori [15] for association tasks.

87 Finally, as previously mentioned, there exists also a third approach, defined semi-supervised [5]  
88 learning, which represents a combination of the supervised and unsupervised approaches: here, a  
89 small part of the labeled samples is used for the training stage, to facilitate the learning of the remaining  
90 large amount of unlabeled input data.

## 91 2.2. Ensemble learning

92 With the expression *ensemble learning*, literature usually refers to a machine learning paradigm  
93 that exploits multiple models, i.e. the *weak learners*, which are trained and combined together in order  
94 to solve specific problems. Indeed, this strategy relies on the idea that such a combination of several  
95 single (weak) models, in particular if associated with a proper feature selection step [16], can lead to  
96 an improvement of the final accuracy.

97 In this context, a machine learning technique able to face both regression and classification tasks,  
98 effectively, is the aforementioned Gradient Boosting [10]: here, a model is created in a gradual, additive,  
99 and sequential way, and it is generalized by allowing the optimization of an arbitrary differentiable  
100 loss function [17]. On the other side, a second well-known algorithm, based on the same ensemble  
101 learning paradigm, and that is widely used in literature thanks to its performance, is represented by  
102 the previously mentioned Random Forests [8]. It exploits the same Gradient Boosting mechanism to  
103 build the prediction model. Specifically, it uses a large number of single decision trees, which operate  
104 in an ensemble way. Each single tree outputs a class prediction, and the final prediction of the model is  
105 given by the class with the most votes.

106 For these reasons, in this work we exploit both Gradient Boosting and Random Forest based  
107 techniques in order to build our general algorithm for predicting the popularity class of Instagram  
108 posts, as better explained in Section 5. In particular, the former is used as the main method, whereas the  
109 latter represents a variant used for the comparison. The paradigm adopted is therefore a supervised  
110 one and, specifically, the ensemble learning described above.

## 111 2.3. Popularity in Social Media

112 Several literature approaches have recently emerged on this topic, proposing different strategies  
113 in terms of the process of data collection, the method of classification, and the measure of popularity  
114 used. For example, Gayberi et al. [18], in their proposal, exploit a tailor-made dataset of 210,630 posts  
115 extracted from common Instagram accounts, enriched with several features related to posts, user  
116 profiles, images and statistical features, addressing the problem as a regression task. Experiments have  
117 been performed using different machine learning algorithms, from Random Forest to MLP and Deep  
118 Learning, comparing the results in terms of MAE and RMSE. De et al. [19] analyze 1,280 Instagram  
119 posts from Indian users, trying to predict the popularity score as a range of achieved likes, grouped  
120 by 25 (e.g.: 0-25, 25-50, etc.). Their solution, based on a Deep Learning approach, leverages the type  
121 of filter applied to the image, the location, the day of the week and time of posting, the caption, the  
122 number of users tagged, and the hashtag list.

123 Other studies have been also conducted in different social contexts such as, for instance, Twitter,  
124 where in [20] the authors propose a study aimed at the prediction of the tweet popularity as a  
125 classification problem. Similarly, in [21], a sentiment analysis task has been performed on the Twitter  
126 posts, in order to measure the positive or negative influence of popular users. In this direction,  
127 the prediction of the retweet rate of a given tweet has been also investigated in [22], whereas a

128 micro-prediction model aimed at evaluating the Twitter message propagation for a user has been  
 129 proposed in [23].

130 A study focused on the analysis and prediction of the news popularity in Telegram have then been  
 131 performed in [24], where the authors underline the differences between the definition of a popularity  
 132 score in such a platform, with respect to other social media. In [25], the authors propose a regression  
 133 method to predict the online video popularity on the basis of the number of views, by taking into  
 134 account YouTube and Facebook platforms. Specifically, the literature shows how the exploitation of  
 135 information from social media becomes increasingly important. In [26], such information have been  
 136 exploited in order to predict e-commerce products prices, whereas, in [27], the authors performed an  
 137 evaluation of products and services through unsolicited social contents.

138 On the other side, many studies are instead oriented to a deeper analysis of the social media  
 139 posts, such as in [28,29], where the authors propose approaches for toxic comment classification.  
 140 Indeed, similarly to other contexts [30–35], a considerable importance is given to the transformation  
 141 of the original data domain, such as, for instance, in [36], where an approach aimed to predict the  
 142 popularity of online videos by exploiting the Fourier transform has been presented. Another example is  
 143 represented by the work in [37], where the authors propose a solution based on the wavelet transform  
 144 to detect human, legitimate bot, and malicious bot in online social networks. However, as it happens  
 145 in related fields [38], the preferences of the users over time could be biased by several factors, not  
 146 reflecting their real preferences [39]. A systematic overview has been provided in [40], where the  
 147 authors reviewed the recent literature, offering statistics and discussing about methods, algorithms,  
 148 techniques, and challenges.

### 149 3. Problem Formulation

150 Differently from other literature works, which commonly formulate the problem as a regression  
 151 task, with the goal of estimating the so-called *engagement factor* (i.e., the ratio between expected likes  
 152 over account followers) of a future post, we model the problem as a binary classification task.

153 More specifically, our goal is to determine if a future Instagram post will be popular or unpopular,  
 154 regardless of the type of visual content published (image or video), but mainly focusing on the post  
 155 metadata such as the caption, the time of publication, and the account typology. In particular, we label  
 156 as popular a post whose number of (expected) likes will exceed a specified threshold (roughly, the  
 157 moving average of likes of the account), or as unpopular otherwise.

158 To formalize this concept, we first need to introduce the preliminary definition of the *Likes Moving*  
 159 *Average* (LMA). Intuitively, given the  $i$ -th post of an Instagram account, the LMA represents the average  
 160 number of likes achieved by its previous  $K$  posts. In formulae, let  $\mathbf{P}_A$  be the ordered set of posts  
 161 published by an account  $A$ ; then, we have:

$$LMA_K(i, A) = \frac{\sum_{j=i-K}^{i-1} \text{like\_count}(\mathbf{P}_A[j])}{K} \quad (1)$$

162 where  $K$  is the number of previous posts considered (i.e., the size of the moving average window),  
 163 and  $\text{like\_count}(\mathbf{P}_A[j])$  is the number of likes obtained by the  $j$ -th post of  $A$ <sup>1</sup>.

164 Given  $LMA_K$ , we can now derive the *Popularity Class* (PC) associated with the  $i$ -th post of the  
 165 account  $A$ , as it follows:

$$PC_{K,\Delta}(i, A) = \begin{cases} 1 \text{ or } \textit{popular}, & \text{if } \text{like\_count}(\mathbf{P}_A[i]) > (1 + \Delta) \cdot LMA_K(i, A) \text{ or} \\ 0 \text{ or } \textit{unpopular}, & \text{otherwise.} \end{cases} \quad (2)$$

<sup>1</sup> In order to simplify our model, we assume to collect the number of likes of a given post, only after this value has stabilized over time and will not vary significantly in the future.

166 that, when the parameter  $\Delta = 0$ , essentially defines as popular a post  $i$  of  $A$ , if its expected  
 167 number of likes is greater than the average likes of the previous  $K$  posts (i.e.,  $LMA_K$ ). In this context,  $\Delta$   
 168 represents a tolerance threshold for the definition of popular posts: for instance, given  $\Delta = 0.2$  and  
 169  $K = 50$ , a post is labelled as popular if its achieved likes are *greater or equal* than 20% of the average  
 170 likes of its preceding 50 posts.

#### 171 4. Data Collection

172 In this section, we describe in detail the process adopted in order to build the dataset used for the  
 173 study and testing of our approach.

174 Although some Instagram datasets already exist in literature or are available on the web, we  
 175 opted to build a new one from scratch. This choice was essentially driven by two reasons: (i) the  
 176 social network sector is constantly evolving, and sees a steady emergence of new features, different  
 177 recommendation policies, and explosive growth in content, therefore databases generated a few years  
 178 earlier may not fully reflect the current situation; and, (ii) for the type of analysis to be performed, we  
 179 needed a large dataset, with raw and genuine contents, and the widest possible set of features.

180 In order to build the dataset, we first collected a preliminary list of Instagram account identifiers,  
 181 from which the full posts are extracted. The steps required for this process are depicted in Figure 1:  
 182 starting from the preliminary list of accounts, we exploited a browser extension<sup>2</sup> for Google Chrome to  
 183 export – for each of them – the full list of followers. Thereby, we iteratively extended the list of accounts,  
 184 until a sufficiently large number of accounts is reached. Then, we leveraged our software module,  
 185 developed in Python, to remove duplicates and filter accounts according to the following specifications:  
 186 first, we selected *ordinary* profiles only (i.e., those with less than 25,000 followers); second, we required  
 187 that each selected account has at least 100 published posts; and, third, we discarded accounts with  
 188 private visibility.

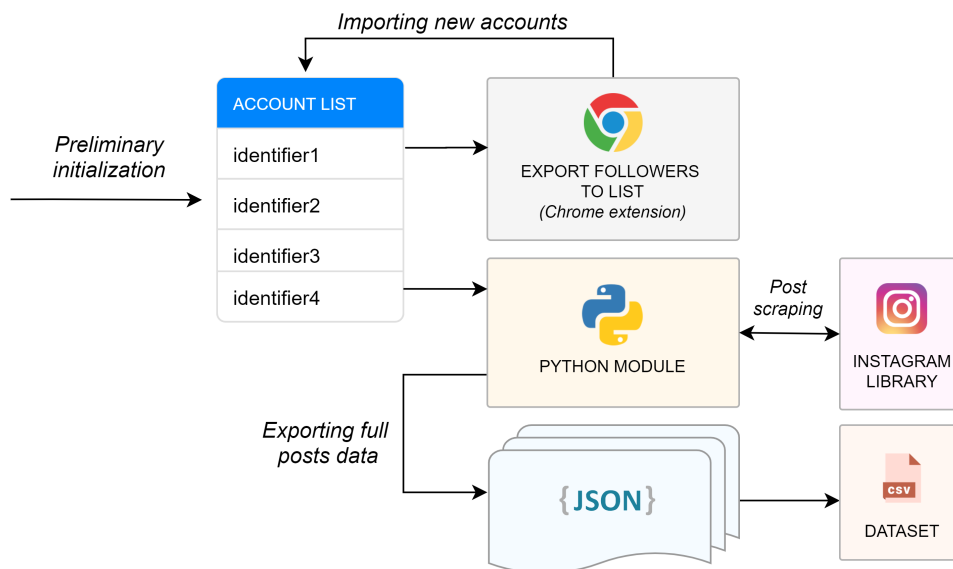


Figure 1. High-level schema of the data collection process.

189 Once we completed this step, and we have then obtained the final list of accounts on which to  
 190 build the dataset, we used our same Python module to interface with Instagram, through integration  
 191 with an open-source crawler<sup>3</sup>. In this way, we proceeded to the extraction, on average, of the last 100

<sup>2</sup> <https://instagramhelpertools.com/>

<sup>3</sup> <https://github.com/huaying/instagram-crawler>

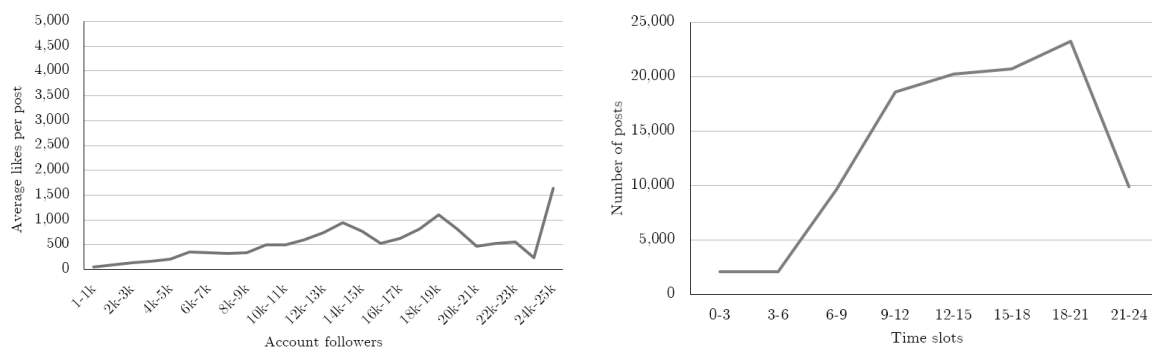
192 posts of each collected user; in particular, the list of complete posts of each account (along with some  
 193 general profile information) has been separately stored on disk, organized in an appropriate directory  
 194 as a json file.

195 As already mentioned, although we also stored the URL of the visual content related to the  
 196 post (i.e., the image or video) into the final json document, this content is not taken into account by  
 197 this work (it is stored for possible future purposes, see Section 7), which instead aims to determine  
 198 the popularity of a future post of Instagram according to the metadata chosen (caption, publication  
 199 time, hashtags, etc.), thus assuming that the visual content itself is predetermined and not amendable.  
 200 Hence, following this scheme, we collected the post and profile features described in Table 1.

FEATURE	DESCRIPTION
<i>is_video</i>	A binary feature that indicates if the post content is an image or a video.
<i>likes_num</i>	The number of likes received by the post.
<i>timestamp</i>	The publication date and time of the post.
<i>followers_num</i>	The number of followers of the post author.
<i>caption</i>	The full caption of the post, including emojis and hashtags.

Table 1. Dataset features collected from Instagram.

201 Overall, the final dataset we collected thus consists of 106,404 rows, for a total of 2,545 different  
 202 users. On average, each user has 2,071 followers. Finally, in Fig. 2, we show some statistics related  
 203 to the accounts considered: in particular, the sub-figure (a) shows that the average of achieved likes  
 204 for post is, in proportion to followers, higher for smaller accounts, whereas (b) outlines the typical  
 205 publication times chosen by these accounts, in which the 18:00-21:00 time range appears to be the most  
 206 popular one.



(a) Average likes for range of account followers.

(b) Most preferred time slots for posting.

Figure 2. Some information about the Instagram accounts included in our dataset: in (a), the average likes for different ranges of account followers is shown; in (b), the most preferred times for post publishing are provided.

## 207 5. Proposed Method

208 In this section, we now describe the proposed method to address the problem of predicting the  
 209 popularity class of a future Instagram post (as defined in Section 3), through a supervised learning  
 210 technique and by exploiting user-generated metadata (caption, hashtag, publication time, etc.) as  
 211 features. As previously mentioned, our method does not perform any kind of analysis on the type of  
 212 visual content that the user intends to publish (picture or video), since the goal is to verify whether the  
 213 background information generated by the user is useful to promote the aforementioned content, or,  
 214 vice versa, whether it may penalize the expected popularity. Moreover, such a consideration allows us  
 215 to generalize our approach and apply it to other social networks where only text is reported.

216 In particular, the proposed method, that we named *XGBoost Instagram Predictor* (XGB-IP), consists  
 217 of two main steps:



- 218 1. a *feature engineering* phase, in which, starting from the fetched data (according to the procedure  
 219 described in Section 4), we enriched the dataset with some derived information, as well as  
 220 removing data whose contribution was negligible or not interesting for the class prediction  
 221 purposes;  
 222 2. a *supervised learning* step, where the classification model is built on the features obtained from  
 223 the step (1), and then used for the popularity class prediction of new posts.

224 The details of these two steps are then described below.

### 225 5.1. Feature engineering

226 Starting from the data collected as described in Section 4, we performed a feature engineering  
 227 stage, where we enriched the dataset with several additional features, in order to boost the performance  
 228 of the classifiers and to improve the overall accuracy of the approach.

229 A first set of features provides some indicators related to the last posts published by the account  
 230 in question. Then, a second set of features is derived by processing the chosen caption of the future  
 231 post, as well as the expected timestamp of publication. A summary of these extra features is provided  
 232 in Table 2.

FEATURE TYPE	DESCRIPTION
<i>Average likes</i>	Average number of likes of the $K$ most recent posts of the account, for different values of $K$ .
<i>Recent likes</i>	The exact number of likes achieved by the most recent published posts.
<i>Time features</i>	The scheduled date and time of the post to be published.
<i>Text-related features</i>	The features derived from the caption (number of words, sentiment score, hashtags popularity, emoji, etc.).

**Table 2.** Advanced features generated by processing the collected input data.

233 First of all, we considered the average likes achieved by the  $K$  most recent posts published by the  
 234 account. This information in fact provides an effective indicator for estimating an account popularity  
 235 trend, and has also been exploited to define the baselines used as a comparison in Section 6. In this  
 236 regard, we extended the dataset with a column for each  $K \in \{5, 10, 15, 20, 30, 50\}$ . Then, we introduced  
 237 a few more targeted features, providing the exact number of likes of the latest published posts. To this  
 238 purpose, we only considered the last 5 previous posts because, from preliminary analysis, we observed  
 239 that data from older posts had a negligible impact on the accuracy results (since their information was  
 240 well absorbed by the averages).

241 We therefore transformed the data related to the scheduled time and date of publication into  
 242 separate features, independently specifying the time, day of the week, month and season of planned  
 243 publication.

244 Finally, as a last but not less important step, we analyzed the caption of the posts, extracting the  
 245 number of words, the number of users tagged, the number (and importance) of chosen hashtags, and,  
 246 notably, a sentiment score [41–43], calculated using the SentiStrength library<sup>4</sup>, after filtering hashtags  
 247 and mentions. We also paid particular attention to the presence of *emojis* in the caption. Hence, we  
 248 defined 10 macro-categories, corresponding to 10 different features/columns: *happiness, love, sadness,*  
 249 *travel, food, pet, angry, music, party* and *sport*. Each of these represents a binary feature, whose value  
 250 depends on the presence, in the caption, of at least one emoji which is relevant to the category (note  
 251 that the set of categories has been reduced as a result of preliminary empirical testing; similarly, the

<sup>4</sup> <http://sentistrength.wlv.ac.uk/>

252 use of binary features has shown to be more effective than the use of discrete features which also count  
 253 the number of emojis present in the text).

254 In parallel, we also collected all the hashtags referred in the posts of our dataset, and then we  
 255 associated a weight to each of them, intended as the number of posts in which they appear, relative  
 256 to the whole Instagram network. Then, similarly to what we did for the emojis, we created 10  
 257 macro-categories of hashtags, corresponding to 10 different levels of hashtag popularity (determined  
 258 by dividing into 10 parts the range of weights between the most used and the least used hashtag,  
 259 according to logarithmic scale). In this way, each macro-category corresponds to a binary feature that  
 260 is set to 1 when the caption includes at least one hashtag which belongs to that level.

261 Finally, we added the popularity class (PC) to the dataset (to be used for the training stage),  
 262 represented by a binary label that assumes the value 1 if the post considered is popular, and 0  
 263 otherwise (according to the definition mentioned in Section 3). Specifically, we added a separated  
 264 label for each different pair of parameters  $K$  and  $\Delta$  (with  $K$  indicating the number of previous posts  
 265 taken into account, and  $\Delta$  the tolerance threshold for the classification of the future post as popular).  
 266 In particular, for our experimental scenarios, described in Section 6, we consider  $K$  values equal to 10,  
 267 30 and 50, while  $\Delta$  values equal to 0, 0.05, 0.1 and 0.15.

## 268 5.2. Supervised learning

269 The key contribution of the proposed approach lies in the exploitation of supervised learning  
 270 techniques that, through the training of classification tools using the labeled input data from our  
 271 dataset, generate the appropriate models for the prediction of the expected popularity of future posts.  
 272 To this purpose, as a result of preliminary tests and also depending on the type of data processed, the  
 273 choice has fallen mainly on the *XGBoost* algorithm (which represents an efficient implementation of the  
 274 previously mentioned Gradient Boosting). We called the overall method *XGBoost - Instagram Predictor*  
 275 (hereafter, XGB-IP). However, for the classification phase, we also implemented a variant based on the  
 276 Random Forest algorithm, as a further point of comparison. Similarly, this variant has been named  
 277 *Random Forest - Instagram Predictor* (hereafter, RF-IP).

Table 3. Algorithm parameters.

ALGORITHM	PARAMETERS	
<i>XGBoost - Instagram Predictor (XGB-IP)</i>	<code>learning_rate</code>	0.1
	<code>n_estimators</code>	750
	<code>max_depth</code>	5
<i>Random Forest - Instagram Predictor (RF-IP)</i>	<code>n_estimators</code>	750
	<code>max_depth</code>	5

278 We tested the classification algorithms with different configurations, by performing a parameters  
 279 exploration, mainly on the following: `learning_rate`, which represents the shrinkage of each tree  
 280 contribution; `n_estimators`, which is the number of boosting stages to perform; and `max_depth`, which  
 281 represents a limitation of the number of nodes in each tree. Table 3 shows the final configurations  
 282 obtained, for both the classification algorithms considered. In addition to those mentioned above,  
 283 several tests were also performed on the other parameters. However, since they did not bring significant  
 284 improvements in accuracy or execution time, we opted to use the default values.

285 Once the parameters are defined, the supervised learning is then performed in two stages: (i) by  
 286 fetching the input data (from the enriched dataset), to build the appropriate training and test sets; (ii)  
 287 by training the classifier. Finally, the prediction module, which integrates the output model, is made  
 288 accessible either as a software API or through a facilitated user interface.



## 289 6. Experimental evaluation

290 Below, we describe the experiments conducted to validate the effectiveness of our method, carried  
291 out on the dataset described in Section 4.

### 292 6.1. Monte Carlo cross-validation

293 The experiments were carried out on the entire dataset, by exploiting the *Repeated hold-out*  
294 validation technique, also known as *Monte Carlo cross-validation* [44]. Compared to the K-fold approach,  
295 although both methods achieve the statistical significance of the result, Monte Carlo allows to explore  
296 a larger number of partitions of the dataset, providing a more accurate estimate of the real accuracy of  
297 the algorithm (as it reduces the variance), in order to decrease the risk of overfitting.

298 To this end, we performed multiple independent runs, each with random partitioning of the  
299 dataset in a training set with the 90% of the samples, and a test set with the remaining 10% of the  
300 samples. Moreover, the split between train and test is performed without replacement, and with  
301 *stratification*, i.e. ensuring that the distribution of the classes of the full dataset is preserved in the  
302 individual sets.

### 303 6.2. Baselines

304 In order to better evaluate the effectiveness of the proposed method, as well as to highlight the  
305 contribution of the used feature engineering and supervised learning techniques, we now provide a  
306 set of strong baselines for the comparison of the results.

307 There are three primary baselines and a fourth obtained as the average of the three. They are not  
308 based on machine learning techniques. Informally, for each of first three of them, the expected class of  
309 popularity of the future post is defined as the class (already known *a priori*) of - respectively - the most  
310 recent, second most recent and third most recent post already published by the same Instagram user.

311 In formulae (by recalling the definition of Popularity Class from Equation 2):

$$Baseline_{j-PC_{K,\Delta}}(i, A) = PC_{K,\Delta}(i - j, A), \quad \text{with } 0 < j \leq 3 \quad (3)$$

312 We also observe that, although these baselines globally obtain good performance (especially as  
313 the value of the considered  $K$  parameter increases), the best one is given by fixing  $j = 1$  (Baseline 1); in  
314 particular, for values of  $j > 3$ , we found a rapid decay of the baseline accuracy. Therefore, hereafter, we  
315 only consider  $j \in \{1, 2, 3\}$  for our comparison.

316 However, we introduce an additional baseline, which represents a special case, and we indicate it  
317 with index  $j = 4$  (Baseline 4). It is simply defined as the average of the first three baselines:

$$Baseline_{4-PC_{K,\Delta}}(i, A) = \frac{\sum_{j=1}^3 PC_{K,\Delta}(i - j, A)}{3} \quad (4)$$

318 The baselines defined above will then be used in the remainder of this section for the comparison  
319 with the proposed method.

### 320 6.3. Evaluation Metrics

321 Before proceeding with the presentation of the experimental results, let us illustrate the metrics  
322 we considered for the assessment of our approach.

#### 323 6.3.1. Accuracy

324 This metric gives us information about the number of instances correctly classified, compared to  
325 the total number of them. It provides an overview of the performances of the classification. Formally,  
326 given a set of  $X$  Instagram posts on which to make a popularity prediction, the accuracy is calculated

327 as shown in the following equation, where  $|X|$  stands for the number of posts and  $X^{(+)}$  stands for  
 328 those correctly classified.

$$Accuracy(X) = \frac{X^{(+)}}{|X|} = \frac{tp + tn}{tp + tn + fp + fn} \quad (5)$$

329 where  $tp$  represents the number of true positives,  $tn$  the number of true negatives,  $fp$  the number  
 330 of false positives and  $fn$  the number of false negatives.

### 331 6.3.2. Balanced accuracy

332 This is a suitable metric for evaluating how good a binary classifier is, especially when the classes  
 333 are unbalanced. It is defined as the average of *recall* obtained on each class. The recall metric  $R_c$  for a  
 334 generic class can be defined as follows:

$$R_c = \frac{tp}{(tp + fn)} \quad (6)$$

335 Using this definition we can obtain the balanced accuracy in the following way:

$$\frac{R_{c0} + R_{c1}}{2} \quad (7)$$

336 where  $R_{c0}$  and  $R_{c1}$  represent, respectively, the recall value for class 0 (*unpopular*) and class 1  
 337 (*popular*), respectively.

### 338 6.3.3. F1-score

339 The F1-Score is a weighted average of two metrics: *precision* and *recall*. Recall metric  $R_c$  is  
 340 calculated as defined in equation (5). Similarly we define the precision  $P_c$  for a generic class:

$$P_c = \frac{tp}{(tp + fp)} \quad (8)$$

341 with  $tp$  representing the number of true positives and  $fp$  the number of false positives. Based on  
 342 the equations (5) and (6) we can then define F1-Score  $F_c$ , for a generic class  $c$ , as:

$$F1_c = 2 * \frac{P_c * R_c}{P_c + R_c} \quad (9)$$

343 This metric is calculated independently for each of the two classes  $c0$  and  $c1$ , resulting in two  
 344 values  $F1_{c0}$  and  $F1_{c1}$ . At this point, the final result is obtained as follows:

$$weighted_{F1} = F1_{c0} * W_0 + F1_{c1} * W_1 \quad (10)$$

345 With  $W_0$  and  $W_1$  representing the weights associated with the two classes, whose value depends  
 346 on the number of true instances of each class.

## 347 6.4. Results

348 We now show the results of the experimental evaluation of our algorithm, in terms of the metrics  
 349 described in Section 6.3, and by comparing it with the baselines outlined above.

350 We performed a total of 12 experiments, by spanning different combinations of  $K$  (10, 30 and 50),  
 351 i.e. the number of previous posts used to calculate the *recent* average of likes, and different values of  $\Delta$   
 352 (0, 0.05, 0.1 and 0.15), i.e. the minimum positive deviation from the average to classify the future post  
 353 as popular. As an example, if we consider  $K = 30$  and  $\Delta = 0.05$ , a post will be classified as popular, if  
 354 its number of expected likes will be at least 5% higher than the average like of the last 30 posts of that  
 355 user.

Specifically, Table 4 shows the results for  $K = 10$  and all the considered thresholds  $\Delta$ . This is the worst case scenario, since the prediction of the post popularity takes into account the average likes of the 10 most recent posts only, thus examining a short-term trend. Focusing on the competitors, we observe that the best baseline is the #4, although for  $\Delta = 0$ , the baseline 1 reaches slightly better results. In this context, our method (XGB-IP) gets the best overall performance for all the  $\Delta$  thresholds (except for the F1-Score with  $\Delta = 0.15$ ). Here, the most significant value is a 57.22% of balanced accuracy for  $\Delta = 0$ : in this case, we get a relative improvement of +8.59% compared to the best baseline, as well as a +7.46% compared to the Random Forest variant of our method. This gap decreases as  $\Delta$  grows, as it becomes more difficult to predict a significant increase in likes than the recent average. For  $\Delta = 0.15$ , in fact, the performance is very similar to that of the competitors.

	$K = 10$					
	$\Delta = 0$			$\Delta = 0.05$		
	<i>Accuracy</i>	<i>Balanced accuracy</i>	<i>F1-Score</i>	<i>Accuracy</i>	<i>Balanced accuracy</i>	<i>F1-Score</i>
Baseline 1	52.86%	52.69%	52.86%	53.92%	52.53%	53.83%
Baseline 2	51.05%	50.83%	51.02%	52.06%	50.49%	51.89%
Baseline 3	49.92%	49.68%	49.89%	51.44%	49.77%	51.22%
Baseline 4	52.51%	52.56%	52.56%	54.15%	52.61%	53.98%
RF-IP	55.12%	53.23%	49.76%	58.60%	51.11%	45.82%
XGB-IP	<b>57.63%</b>	<b>57.22%</b>	<b>57.42%</b>	<b>59.59%</b>	<b>55.63%</b>	<b>56.86%</b>
	$\Delta = 0.1$			$\Delta = 0.15$		
Baseline 1	56.21%	52.65%	55.94%	58.91%	52.61%	58.43%
Baseline 2	54.28%	50.41%	53.90%	57.45%	50.69%	56.80%
Baseline 3	53.94%	49.85%	53.45%	57.19%	50.13%	56.39%
Baseline 4	57.56%	52.81%	56.53%	61.61%	52.84%	<b>59.48%</b>
RF-IP	62.86%	50.38%	49.18%	66.95%	50.34%	54.16%
XGB-IP	<b>63.10%</b>	<b>53.97%</b>	<b>57.17%</b>	<b>67.13%</b>	<b>53.13%</b>	59.30%

Table 4. Results of the experiments for  $K = 10$  (best values are highlighted in bold).

By moving to  $K = 30$  (Table 5), i.e. by increasing the number of recent posts considered, all methods considered get better results. However, our approach is confirmed as the best, with the exception - also in this case - of  $\Delta = 0.15$  (but with a result still comparable to that of baseline 4). The balanced accuracy for  $\Delta = 0$  is 61.19%, with a relative increase of +5.12% compared to baseline 4. We also note that, for this configuration, even the variant based on Random Forest obtains a good result, but it worsens dramatically as  $\Delta$  increases.

	$K = 30$					
	$\Delta = 0$			$\Delta = 0.05$		
	<i>Accuracy</i>	<i>Balanced accuracy</i>	<i>F1-Score</i>	<i>Accuracy</i>	<i>Balanced accuracy</i>	<i>F1-Score</i>
Baseline 1	56.78%	56.77%	56.79%	57.09%	56.51%	57.08%
Baseline 2	55.65%	55.64%	55.65%	55.95%	55.29%	55.91%
Baseline 3	54.29%	54.26%	54.29%	54.94%	54.21%	54.87%
Baseline 4	58.14%	58.21%	58.11%	58.37%	57.94%	58.41%
RF-IP	59.22%	59.20%	59.22%	59.97%	57.78%	58.33%
XGB-IP	<b>61.17%</b>	<b>61.19%</b>	<b>61.17%</b>	<b>61.92%</b>	<b>60.49%</b>	<b>61.30%</b>
	$\Delta = 0.1$			$\Delta = 0.15$		
Baseline 1	58.65%	56.72%	58.59%	60.34%	56.52%	60.19%
Baseline 2	57.28%	55.19%	57.17%	58.97%	54.84%	58.73%
Baseline 3	56.54%	54.27%	56.36%	58.88%	54.54%	58.55%
Baseline 4	60.21%	58.18%	60.08%	62.46%	<b>58.00%</b>	61.95%
RF-IP	62.52%	55.05%	56.10%	64.98%	51.77%	53.68%
XGB-IP	<b>64.09%</b>	<b>59.22%</b>	<b>61.77%</b>	<b>66.70%</b>	57.81%	<b>62.70%</b>

Table 5. Results of the experiments for  $K = 30$  (best values are highlighted in bold).

Finally, for  $K = 50$ , our method obtains the best overall result, with a balanced accuracy of 64.72% for  $\Delta = 0$  (in relative terms, it means a +5.03% if compared to the best baseline, and a +3.9% if compared to the Random Forest variant), hence achieving a good predictivity.

375 Moreover, in this last setup, besides clearly overcoming all the competitors, the result of our  
 376 method for  $\Delta = 0.15$  (i.e. the ability to predict whether the considered post will reach a number of likes  
 377 more than 15% higher than the average of the last 50 published ones) has shown to be particularly  
 378 significant. Indeed, as illustrated in Table 6, for this difficult scenario, our XGB-IP method achieves a  
 379 balanced accuracy of 62.68%, and an F1-score of 65.81%.

	K = 50					
	$\Delta = 0$			$\Delta = 0.05$		
	Accuracy	Balanced accuracy	F1-Score	Accuracy	Balanced accuracy	F1-Score
Baseline 1	59.57%	59.54%	59.57%	59.91%	59.73%	59.90%
Baseline 2	58.74%	58.72%	58.74%	58.82%	58.63%	58.81%
Baseline 3	57.51%	57.49%	57.51%	57.77%	57.53%	57.74%
Baseline 4	61.75%	61.62%	61.64%	61.80%	61.78%	61.84%
RF-IP	62.39%	62.29%	62.33%	62.64%	62.29%	62.53%
XGB-IP	<b>64.82%</b>	<b>64.72%</b>	<b>64.76%</b>	<b>64.95%</b>	<b>64.57%</b>	<b>64.83%</b>
	$\Delta = 0.1$			$\Delta = 0.15$		
Baseline 1	60.51%	59.59%	60.49%	61.71%	59.51%	61.66%
Baseline 2	59.50%	58.53%	59.46%	60.82%	58.47%	60.73%
Baseline 3	58.67%	57.56%	58.58%	60.02%	57.47%	59.85%
Baseline 4	62.44%	61.64%	62.44%	63.79%	61.47%	63.65%
RF-IP	63.57%	61.09%	62.41%	65.38%	59.21%	62.34%
XGB-IP	<b>65.93%</b>	<b>63.84%</b>	<b>65.15%</b>	<b>67.50%</b>	<b>62.68%</b>	<b>65.81%</b>

Table 6. Results of the experiments for  $K = 50$  (best values are highlighted in bold).

### 380 6.5. Implementation details

381 We conclude this section by describing the distributed architecture and the associated  
 382 implementation details, used for the construction of the dataset and for the execution of the experiments  
 383 described above.

384 With respect to the creation of the dataset (i.e. the collection of the list of Instagram accounts, the  
 385 download of all the posts data, etc.), we exploited a typical laptop, running Microsoft Windows 10.  
 386 Conversely, for the training phase, we opted for a distributed implementation, in order to evaluate  
 387 the scalability of the approach and to exploit big data technologies and tools. To this purpose, we  
 388 leveraged on the following software:

- 389 • the *Amazon AWS / EC2* cloud computing platform<sup>5</sup>, to build the cluster infrastructure used for  
 390 our experiments;
- 391 • the *HashiCorp Terraform* tool<sup>6</sup>, for the management and provisioning of the AWS instances;
- 392 • the *Apache Spark* framework<sup>7</sup>, for the distributed computing, in particular its Python wrapper  
 393 *XGBoost4J-Spark*<sup>8</sup>, which allowed us to integrate Spark and XGBoost;
- 394 • the *Apache Hadoop* framework<sup>9</sup>, to handle the dataset.

395 Through these tools, we set up different distributed clusters, consisting of 1, 2, 4 and 8 parallel  
 396 *workers* respectively, to compare their performance. In this context, each worker corresponds to a  
 397 *t2.medium* AWS instance, each one featuring the following technical specifications:

- 398 • Operating System: Ubuntu Server 18.04 LTS;
- 399 • Processor: Intel Xeon (up to 3.3 GHz);
- 400 • vCPU (#): 2;

<sup>5</sup> <https://aws.amazon.com/it/ec2/>

<sup>6</sup> <https://www.terraform.io/>

<sup>7</sup> <https://spark.apache.org/>

<sup>8</sup> [https://xgboost.readthedocs.io/en/latest/jvm/xgboost4j\\_spark\\_tutorial.html](https://xgboost.readthedocs.io/en/latest/jvm/xgboost4j_spark_tutorial.html)

<sup>9</sup> <https://hadoop.apache.org/>

- Memory (GB): 4.

Table 7 shows the execution times of the training phase, in seconds, as the cluster size increases. We observe how the exploitation of the big data management tools, as well as a parallel architecture for running the training phase, allowed us to obtain a significant time saving of  $\sim 38\%$  in terms of execution performance, when a cluster size consisting of 8 workers was used (compared to the use of a single machine).

Workers	Instance type	Execution time
1	t2.medium	55.59 s
2	t2.medium	51.75 s
4	t2.medium	48.15 s
8	t2.medium	40.25 s

Table 7. Execution time of a single iteration as cluster size increases.

This result makes it possible to affirm that the distributed implementation of the method is adequately scalable, and can be therefore extended in order to exploit a much larger dataset (with millions of posts) and a possibly larger number of features.

## 7. Conclusions and Future Work

In this work, we proposed an original definition of popularity for Instagram posts, by modelling it as a binary classification problem. Then, we designed a novel approach to predicting such a popularity, combining feature engineering and supervised learning techniques, and big data technologies.

Our method is developed to take advantage of the metadata associated with the post to be published (account data, scheduled date and time of publication, caption, hashtags, emojis, etc.), regardless of the visual content proposed (video or image), and thus making it easily extensible to different social networks too. The validation of the method was performed considering two different implementations (one based on Gradient Boosting, and one based on Random Forest), against several strong baselines not based on machine learning techniques. Notably, all the experiments were carried out on a newly built dataset of over 100,000 Instagram posts, adequately representative of common Italian and English accounts, as well as through the use of a distributed infrastructure based on AWS EC2 and Apache Spark.

The results showed that the implementation based on Gradient Boosting has a good effectiveness and is promising, reaching a balanced accuracy of 64.72%, in the real-world scenario, when considering  $K = 50$  and  $\Delta = 0$  (i.e. when predicting that the future post will be popular if the expected likes are higher than the average of the last 50 published posts). However, the method proved to be successful in almost all the parameter thresholds analyzed, and almost always exceeded both the Random Forest-based variant and the considered baselines, in terms of balanced accuracy and F1-score. Additionally, for several thresholds, the relative improvement against the best competitor is between 4-8%. In addition, the adoption of the distributed architecture for the training stage showed a reduction of up to 38% of the execution times, compared to a non-parallel approach, and thus making it possible to apply the method to much larger datasets, also related to different social networks and/or a greater number of features.

In light of these results, it is possible to outline some possible future research developments. First of all, the addition of new features that, for example, taking advantage of Natural Language Processing techniques already widely adopted in related research areas, allow to achieve even higher accuracy through a deeper analysis of the caption text. Secondly, the analysis of classification tools different from those already considered, for example based on convolutional neural networks (CNNs) and deep learning techniques [45–47], which - in the presence of a very large amount of data - could lead to better performance. And, finally, the development of new tools that, using as input the results of our approach, can potentially help users and social media managers to optimize their contents, in order to achieve a good level of expected popularity for the new posts to be published.

443 **Author Contributions:** Conceptualization, S.C., A.S.P., D.R.R., R.S., and G.U.; data curation, S.C., A.S.P., D.R.R.,  
444 R.S., and G.U.; formal analysis, S.C., A.S.P., D.R.R., R.S., and G.U.; methodology, S.C., A.S.P., D.R.R., R.S., and G.U.;  
445 resources, S.C., A.S.P., D.R.R., R.S., and G.U.; supervision, S.C.; validation, S.C., A.S.P., D.R.R., R.S., and G.U.;  
446 writing, original draft, A.S.P. and G.U.; writing, review and editing, S.C., A.S.P., D.R.R., R.S., and G.U. All authors  
447 have read and agreed to the published version of the manuscript.

448 **Acknowledgments:** The authors gratefully acknowledge Andrea Catania and Stefano R. Chessa for their useful  
449 suggestions and comments, which contribute to improve the final quality of this work.

450 **Conflicts of Interest:** The authors declare no conflict of interest.

## 451 References

- 452 1. Recupero, D.; Nuzzolese, A.; Consoli, S.; Presutti, V.; Peroni, S.; Mongiovi, M. Extracting knowledge  
453 from text using SHELDON, a semantic holistic framEwork for LinkeD ONtology data. 2015, pp. 235–238.  
454 doi:10.1145/2740908.2742842.
- 455 2. Consoli, S.; Recupero, D. Using FRED for named entity resolution, linking and typing for  
456 knowledge base population. *Communications in Computer and Information Science* **2015**, *548*, 40–50.  
457 doi:10.1007/978-3-319-25518-7\_4.
- 458 3. Dridi, A.; Reforgiato Recupero, D. Leveraging semantics for sentiment polarity detection in social media.  
459 *International Journal of Machine Learning and Cybernetics* **2019**, *10*, 2045–2055. doi:10.1007/s13042-017-0727-z.
- 460 4. Meena, K.S.; Suriya, S. A Survey on Supervised and Unsupervised Learning Techniques. *International*  
461 *Conference on Artificial Intelligence, Smart Grid and Smart City Applications*. Springer, 2019, pp. 627–644.
- 462 5. Van Engelen, J.E.; Hoos, H.H. A survey on semi-supervised learning. *Machine Learning* **2020**, *109*, 373–440.
- 463 6. Tehrani, A.F.; Ahrens, D. Supervised regression clustering: A case study for fashion products. *International*  
464 *Journal of Business Analytics (IJBAN)* **2016**, *3*, 21–40.
- 465 7. Sen, P.C.; Hajra, M.; Ghosh, M. Supervised Classification Algorithms in Machine Learning: A Survey and  
466 Review. In *Emerging Technology in Modelling and Graphics*; Springer, 2020; pp. 99–111.
- 467 8. Breiman, L. Random forests. *Machine learning* **2001**, *45*, 5–32.
- 468 9. Steinwart, I.; Christmann, A. *Support vector machines*; Springer Science & Business Media, 2008.
- 469 10. Natekin, A.; Knoll, A. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics* **2013**, *7*, 21.
- 470 11. HECHT-NIELSEN, R. III.3 - Theory of the Backpropagation Neural Network\*\*Based on “nonindent” by  
471 Robert Hecht-Nielsen, which appeared in Proceedings of the International Joint Conference on Neural  
472 Networks 1, 593–611, June 1989. © 1989 IEEE. In *Neural Networks for Perception*; Wechsler, H., Ed.; Academic  
473 Press, 1992; pp. 65 – 93. doi:https://doi.org/10.1016/B978-0-12-741252-8.50010-8.
- 474 12. Grira, N.; Crucianu, M.; Boujemaa, N. Unsupervised and semi-supervised clustering: a brief survey. *A*  
475 *review of machine learning techniques for processing multimedia content* **2004**, *1*, 9–16.
- 476 13. Cios, K.J.; Swiniarski, R.W.; Pedrycz, W.; Kurgan, L.A. Unsupervised learning: association rules. *Data*  
477 *Mining*. Springer, 2007, pp. 289–306.
- 478 14. Hartigan, J.A.; Wong, M.A. Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal*  
479 *Statistical Society. Series C (Applied Statistics)* **1979**, *28*, 100–108.
- 480 15. Hegland, M. The apriori algorithm—a tutorial. In *Mathematics and computation in imaging science and*  
481 *information processing*; World Scientific, 2007; pp. 209–262.
- 482 16. Pes, B. Ensemble feature selection for high-dimensional data: a stability analysis across multiple domains.  
483 *Neural Computing and Applications* **2019**, pp. 1–23.
- 484 17. Jena, P.C.; Mishra, D.; Pani, S.K.; others. A novel approach for regularization of ensemble learning  
485 in classification and regression analysis. *Indian Journal of Public Health Research & Development* **2018**,  
486 *9*, 1406–1411.
- 487 18. Gayberi, M.; Gunduz Oguducu, S. Popularity Prediction of Posts in Social Networks Based on User, Post  
488 and Image Features. 2019, pp. 9–15. doi:10.1145/3297662.3365812.
- 489 19. De, S.; Maity, A.; Goel, V.; Shitole, S.; Bhattacharya, A. Predicting the Popularity of Instagram Posts for a  
490 Lifestyle Magazine Using Deep Learning. 2017. doi:10.1109/CSCITA.2017.8066548.
- 491 20. Hong, L.; Dan, O.; Davison, B.D. Predicting popular messages in twitter. *Proceedings of the 20th*  
492 *international conference companion on World wide web*, 2011, pp. 57–58.
- 493 21. Bae, Y.; Lee, H. Sentiment analysis of twitter audiences: Measuring the positive or negative influence of  
494 popular twitterers. *Journal of the American Society for Information Science and Technology* **2012**, *63*, 2521–2535.



- 495 22. Hoang, T.B.N.; Mothe, J. Predicting information diffusion on Twitter—Analysis of predictive features.  
496 *Journal of computational science* **2018**, *28*, 257–264.
- 497 23. Rao, P.G.; Venkatesha, M.; Kanavalli, A.; Shenoy, P.D.; Venugopal, K. A micromodel to predict message  
498 propagation for twitter users. 2018 International Conference on Data Science and Engineering (ICDSE).  
499 IEEE, 2018, pp. 1–5.
- 500 24. Naseri, M.; Zamani, H. Analyzing and predicting news popularity in an instant messaging service.  
501 Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in  
502 Information Retrieval, 2019, pp. 1053–1056.
- 503 25. Trzciński, T.; Rokita, P. Predicting popularity of online videos using support vector regression. *IEEE*  
504 *Transactions on Multimedia* **2017**, *19*, 2561–2570.
- 505 26. Carta, S.; Medda, A.; Pili, A.; Reforgiato Recupero, D.; Saia, R. Forecasting E-Commerce Products Prices  
506 by Combining an Autoregressive Integrated Moving Average (ARIMA) Model and Google Trends Data.  
507 *Future Internet* **2019**, *11*, 5.
- 508 27. Peláez, J.I.; Martínez, E.A.; Vargas, L.G. Products and services valuation through unsolicited information  
509 from social media. *Soft Computing* **2020**, *24*, 1775–1788.
- 510 28. Carta, S.; Corrigan, A.; Mulas, R.; Recupero, D.R.; Saia, R. A Supervised Multi-class Multi-label Word  
511 Embeddings Approach for Toxic Comment Classification. KDIR, 2019, pp. 105–112.
- 512 29. Georgakopoulos, S.V.; Tasoulis, S.K.; Vrahatis, A.G.; Plagianakos, V.P. Convolutional neural networks for  
513 toxic comment classification. Proceedings of the 10th Hellenic Conference on Artificial Intelligence, 2018,  
514 pp. 1–6.
- 515 30. Saia, R.; Carta, S. Evaluating the benefits of using proactive transformed-domain-based techniques in  
516 fraud detection tasks. *Future Generation Computer Systems* **2019**, *93*, 18–32.
- 517 31. Saia, R.; Carta, S. Evaluating Credit Card Transactions in the Frequency Domain for a Proactive Fraud  
518 Detection Approach. SECRYPT, 2017, pp. 335–342.
- 519 32. Saia, R.; Carta, S.; others. A Frequency-domain-based Pattern Mining for Credit Card Fraud Detection.  
520 IoTBDS, 2017, pp. 386–391.
- 521 33. Saia, R.; Carta, S. A fourier spectral pattern analysis to design credit scoring models. Proceedings of the  
522 1st International Conference on Internet of Things and Machine Learning, 2017, pp. 1–10.
- 523 34. Saia, R. A discrete wavelet transform approach to fraud detection. International Conference on Network  
524 and System Security. Springer, 2017, pp. 464–474.
- 525 35. Saia, R.; Carta, S.; Fenu, G. A wavelet-based data analysis to credit scoring. Proceedings of the 2nd  
526 International Conference on Digital Signal Processing, 2018, pp. 176–180.
- 527 36. Zhou, Y.; Wu, Z.; Zhou, Y.; Hu, M.; Yang, C.; Qin, J. Exploring Popularity Predictability of Online Videos  
528 With Fourier Transform. *IEEE Access* **2019**, *7*, 41823–41834.
- 529 37. Jr, S.B.; Campos, G.F.; Tavares, G.M.; Igawa, R.A.; Jr, M.L.P.; Guido, R.C. Detection of human, legitimate bot,  
530 and malicious bot in online social networks based on wavelets. *ACM Transactions on Multimedia Computing,*  
531 *Communications, and Applications (TOMM)* **2018**, *14*, 1–17.
- 532 38. Boratto, L.; Carta, S.; Fenu, G.; Saia, R. Semantics-aware content-based recommender systems: Design and  
533 architecture guidelines. *Neurocomputing* **2017**, *254*, 79–85.
- 534 39. Wu, L.; Ge, Y.; Liu, Q.; Chen, E.; Hong, R.; Du, J.; Wang, M. Modeling the evolution of users' preferences  
535 and social links in social networking services. *IEEE Transactions on Knowledge and Data Engineering* **2017**,  
536 *29*, 1240–1253.
- 537 40. Rousidis, D.; Koukaras, P.; Tjortjis, C. Social media prediction: a literature review. *Multimedia Tools and*  
538 *Applications* **2020**, *79*, 6279–6311.
- 539 41. Reforgiato Recupero, D.; Cambria, E. ESWC 14 challenge on Concept-Level Sentiment Analysis.  
540 *Communications in Computer and Information Science* **2014**, *475*, 3–20. doi:10.1007/978-3-319-12024-9\_1.
- 541 42. Recupero, D.; Consoli, S.; Gangemi, A.; Nuzzolese, A.; Spampinato, D. A semantic web based  
542 core engine to efficiently perform sentiment analysis. *Lecture Notes in Computer Science (including*  
543 *subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2014**, *8798*, 245–248.  
544 doi:10.1007/978-3-319-11955-7\_28.
- 545 43. Recupero, D.; Dragoni, M.; Presutti, V. ESWC 15 challenge on concept-level sentiment analysis.  
546 *Communications in Computer and Information Science* **2015**, *548*, 211–222. doi:10.1007/978-3-319-25518-7\_18.

- 547 44. Xu, Q.S.; Liang, Y.Z. Monte Carlo cross validation. *Chemometrics and Intelligent Laboratory Systems* **2001**,  
548 56, 1–11.
- 549 45. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *nature* **2015**, 521, 436–444.
- 550 46. Barra, S.; Carta, S.M.; Corrigan, A.; Podda, A.S.; Recupero, D.R. Deep learning and time series-to-image  
551 encoding for financial forecasting. *IEEE/CAA Journal of Automatica Sinica* **2020**, 7, 683–692.
- 552 47. Deng, L.; Yu, D. Deep learning: methods and applications. *Foundations and trends in signal processing* **2014**,  
553 7, 197–387.