POLITECNICO DI MILANO
School of Industrial and Information Engineering
Department of Electronics, Information and
Bioengineering
Master of Science in Computer Engineering

Social Media Posts Popularity
Prediction During Long-Running Live
Events
A case study on Fashion Week

Supervisor: Prof. Marco Brambilla

Candidate:
Alireza Javadian Sabet
Matr. 841848

Academic Year 2018-2019

to my mother, Parvaneh

Acknowledgement

I would like to thank my advisor Prof. Marco Brambilla for giving me an opportunity to work on this project and for his support. I had a chance to improve my skills and gain a different point of view about research.

Thanks to ever since my biggest support, my lifetime teacher, my brother, Dr. Mohammadreza Valipour.

Thanks to my wife, Mahsa Shekari, for her contributions and endless supports.

Thanks to all my professors in Politecnico di Milano University, specially Prof. Matteo Matteucci, without whom I would never started this path.

Thanks to my professors Dr. Morteza Saberi and Dr. Peyman Shahverdi who encouraged me to pursue my study.

Thanks to all of my friends who supported me while doing this thesis or gave contribution to it, specially Dr. Marcello M. Bersani, Christian Tagliabue, Andrea Radaelli, Ali Adhami, Reza Mohebbali Panta, Sohrab Alipour, Sasan Golchin, Hamidreza Akbarirad, Hassan Nazeer Chaudhry, Amarildo Likmeta, Eldar Alasgarov, Ladan Hosseinzadeh, Dilara Çinar, Beyza Topçu, Ali Nadeb, David Guzzo and Adeleh Esmailzadeh.

My special thanks to Sergio Vicentini for his supports.

At the end, I would like to thank the person to whom words are not enough to express my gratitude, Marjan Hosseini.

Contents

Abstract	XIV
Sommario	XV
1 Introduction	1
1.1 Context and Motivations	1
1.2 Objectives	2
1.3 Proposed Solution	3
1.4 Structure of the Thesis	4
2 Background	5
2.1 Social Media Networks	5
2.2 Feature Selection	6
2.3 Regression Models	7
2.3.1 Ridge Regression	7
2.3.2 Gradient Tree Boosting	8
2.3.3 Support Vector Regression	9
2.3.4 Deep Neural Networks	10
2.4 Performance Evaluation Methods	12
2.4.1 Evaluation Metrics	12
2.4.2 Distance Correlation Index	13
2.4.3 Spearman's Rank Correlation	14
2.4.4 Validation Methods	14
3 Related Works	16
4 Long Running Live Events	23
4.1 High Level Overview	23

4.1.1	Potentials	24
4.1.2	Elements	24
4.1.3	Procedure	26
4.1.4	Challenges	27
4.2	Case Study	28
4.2.1	Fashion Week	28
4.2.2	Instagram	31
4.2.3	Big Four’s Fashion Week Fall/Winter 2018 on Instagram	32
4.2.4	Case Study Challenges	33
5	Posts Popularity Prediction	34
5.1	Popularity Definition	34
5.2	Sampling	36
5.3	Feature Extractions	37
5.4	Data Preprocessing	38
5.5	Feature Selection	39
5.6	Hyper-parameters Tuning	40
6	Implementation	42
6.1	Dataset	42
6.1.1	Data Collection	42
6.1.2	Data Cleaning	43
6.1.3	Data Preparation	44
6.2	Exploratory Data Analysis	45
6.2.1	Posts Related Analysis	45
6.2.2	Users’ Behavioral Analysis	47
6.2.3	Brands Related Analysis	50
6.3	Posts Popularity Prediction	51
6.3.1	Sampling	51
6.3.2	Feature Extraction	52
6.3.3	Image Related Features	52
6.3.4	Microsoft Azure’s Computer Vision Validation	55
6.3.5	Data Preprocessing	57
6.3.6	Base Model	58
6.3.7	Feature Selection	58
6.3.8	Hyper-parameter Tuning	62
7	Experimental Results	65

7.1	Exploratory Data Analysis Results	65
7.1.1	Posts Related Analysis Results	65
7.1.2	Users Related Analysis Results	77
7.1.3	Brands Related Analysis Results	85
7.2	Post Popularity Prediction Rresults	89
7.2.1	Base Model Results	89
7.2.2	Sampling Results	91
7.2.3	Feature Selection Results	91
7.2.4	Hyper-parameter Tuning Results	94
8	Conclusions	99
	Bibliography	101
A	Appendices	111

List of Abbreviations

WWW	<i>World Wide Web</i>
UGC	<i>User-Generated Contents</i>
OSN	<i>Online Social Networks</i>
OECD	<i>Organisation for Economic Co-operation and Development</i>
FS	<i>Feature Selection</i>
ML	<i>Machine Learning</i>
RSS	<i>Residual Sum of Squares</i>
CART	<i>Classification And Regression Trees</i>
XGBoost	<i>eXtreme Gradient Boosting</i>
SVR	<i>Support Vector Regression</i>
FFNN	<i>FeedForward Neural Networks</i>
DNN	<i>Deep Neural Networks</i>
MLP	<i>MultiLayer Perceptron</i>
MSE	<i>Mean Square Error</i>
RMSE	<i>Root Mean Square Error</i>
MAE	<i>Mean Absolute Error</i>
dCor	<i>distance Correlation index</i>
SRC	<i>Spearman's Rank Correlation</i>
HOCV	<i>Hold-Out Cross Validation</i>
<i>k</i> -FCV	<i>k-Fold Cross Validation</i>
SME	<i>Small and Medium sized Enterprises</i>
SN	<i>Social Network</i>
RUS	<i>Random Under Sampling</i>
CSV	<i>Comma Separated Values</i>
EDA	<i>Exploratory Data Analysis</i>
LBP	<i>Local Binary Patterns</i>
MAPE	<i>Mean Absolute Percentage Error</i>
FW	<i>Fashion Week</i>

List of Figures

2.1	Example of a feedforward neural networks with three hidden layers.	11
4.1	A simplified overview of the main elements of a long-running live event and the relationships among them.	25
4.2	Overview of the process to extract knowledge from long-running live events.	27
4.3	Percent of U.S. adults who use at least one social media site according to different age ranges, since 2005.	30
4.4	Daily U.S. online users' engagement with brands in 2016. . . .	31
4.5	Most popular social media platforms usages in percent among U.S. adults since 2012.	32
4.6	Big Four's Fall/Winter 2018 events calendar and the experiment period.	33
5.1	Hierarchical representation of the case study's features types. .	38
6.1	Distribution of likes count in the posts dataset resulting from: a) Cleaning phase, and b) Sampling phase of the proposed method.	52
6.2	Comparison of accuracy according to different thresholds for discretizing the a)Objects, b)Categories and c)Tags features scores provided by Microsoft Azure's Computer Vision service. .	57
6.3	Average of RMSE error <i>w.r.t.</i> different number of selected features for the Ridge, SVR, XGBoost and DNN methods for 10 independent runs.	61
6.4	Boxplot of RMSE error <i>w.r.t.</i> different number of selected features for the Ridge, SVR, XGBoost and DNN methods for 10 independent runs.	62

7.1	Usage frequency of the hashtags in Big Four's Fall/Winter 2018 fashion week. The <i>x-axis</i> lists the usage ranks of the hashtags, while <i>y-axis</i> reports the logarithm of the frequency. .	66
7.2	Top 15 most frequent hashtags in Big Four's Fall/Winter 2018 fashion week. The <i>x-axis</i> lists the hashtags ordered by their percentage of usage, while the <i>y-axis</i> reports the percentage of the posts contain those hashtags.	67
7.3	Word cloud representation for the of most frequent used hashtags in Big Four's Fall/Winter 2018 fashion week.	67
7.4	Instagram users' responses to Big Four's Fall/Winter 2018 fashion week for the entire experiment period. The granularity is 1 hour. The related signal to each city is defined by different color and the colored boxes in background, specify the official calendar for each sub-events.	69
7.5	Venn diagram representing the portion of Big Four's Fall/Winter 2018 fashion week posts contain hashtags of the different combination of cities.	71
7.6	Venn diagram categorizing Instagram users who posted about Big Four's Fall/Winter 2018 fashion week in: Users with pure posts group (<i>in green</i>) meaning whether each of their posts contain hashtags related to just one city, in users with impure posts group (<i>in red</i>) which each all of their posts contain hashtags related to more than one city and (<i>in orange</i>) which is the overlap of two categories meaning having both pure and impure posts.	73
7.7	Worldwide geographical dispersion of the Geo-located Instagram posts about Big Four's Fall/Winter 2018 fashion week differentiated by purple, green, blue, red and yellow dots representing posts related to Multiple, Milan, Paris, London and New York respectively.	75
7.8	Worldwide geographical dispersion of the Instagram users posted about Big Four's Fall/Winter 2018 fashion week and having Geo-location information in their profile. Each red dot represents a user residence location.	75

7.9	Heatmap representing the worldwide geographical density of the Geo-located Instagram posts about Big Four's Fall/Winter 2018 fashion week for Milan, Paris, London, and New York from top to bottom respectively.	76
7.10	Heatmap representing the regional geographical density of the Geo-located Instagram posts about Big Four's Fall/Winter 2018 fashion week for a)Paris in Europe, b)London in Europe, c)Milan in Europe, and d)New York in the U.S.	77
7.11	The number of followers (<i>y-axis</i>) vs. number of followings (<i>x-axis</i>) for each user Instagram user's profile who posted about Big Four's Fall/Winter 2018 fashion week, scattered in blue dots. Line $y = x$ is drawn in red and both the x and y axes are limited to 10,000.	78
7.12	The logarithm scaled number of followers (<i>y-axis</i>) vs. logarithm scaled number of followings (<i>x-axis</i>) for each user Instagram user's profile who posted about Big Four's Fall/Winter 2018 fashion week, scattered in blue dots. Line $y = x$ is drawn in red and both the x and y axes are limited to 10,000.	79
7.13	Histogram of the number of following (<i>blue</i>) and the number of followers (<i>orange</i>) on <i>x-axis</i> both limited to 10,000, and number of users with the correspondent numbers on <i>y-axis</i> for the Instagram user's profile who posted about Big Four's Fall/Winter 2018 fashion week. The peaks with high values are crossed in red.	80
7.14	Histogram of the number of posts per user for the Instagram user's profile who posted about Big Four's Fall/Winter 2018 fashion week. The <i>x-axis</i> is the number of posts, while <i>y-axis</i> reports the logarithm number of users having the particular number of posts.	81
7.15	Elbow effect showing the result of clustering in terms of WSS, each time with different number of clusters, to find the optimal users' clusters numbers based on the duration they wait until they post again (their posting waiting time behaviour). The red sign shows the optimal number of clusters (= 5) where increasing the numbers does not decrease WSS considerably.	82

7.16	Information about users' clusters with optimal number of clusters (= 5) based on their the duration they wait to post again (users' posting waiting time behaviour) - the clusters (groups) are defined based on users' number of posts in the intervals [1,9], [10,35], [36,92], [93,221] and [222,500] for groups 1 to 5 and above 500 are outliers. a) The number of users in clusters. b) The percentage of users in clusters.	83
7.17	Top brands having more than 1,500 related posts in Big Four's Fall/Winter 2018 fashion week. The <i>y-axis</i> lists the brands ordered by their frequency, while the <i>x-axis</i> reports the number of posts contain hashtags related to each brand.	85
7.18	Instagram users' responses to the brands presented in Big Four's Fall/Winter 2018 fashion week for the entire experiment period. The granularity is 1 hour. The related signal to each brand is defined by different color and just brands with more than 1,500 relevant posts are considered.	86
7.19	Instagram users' responses to Dior brand in red (<i>left</i>) vs. Chanel brand in blue (<i>right</i>) in Big Four's Fall/Winter 2018 fashion week for the entire experiment period. The granularity is 1 hour and the signal for each city is plotted separately from top to bottom.	87
7.20	Heatmap matrix of the Spearman's correlation analysis showing the correlation coefficients among the values obtained from Instagram users' responses to Chanel and Dior brands for each of the cities in Big Four's Fall/Winter 2018 fashion week for the entire experiment period.	88
7.21	Predicted likes count by the base model (ridge regressor $\alpha = 1$) vs. true likes count considering all the features. a) For the training, and b) for the test sets both resulted from the sampling phase of the proposed method to sample from the dataset of Instagram users' responses to Big Four's Fall/Winter 2018 fashion week for the entire experiment period.	89
7.22	Likes count distributions in the: a) Training, and b) Test datasets.	91

7.23	First 50 frequently selected features in 10 runs by FS phase of the proposed method using <i>dCor</i> index. The <i>y-axis</i> lists the top 50 features ordered by their frequency, while the <i>y-axis</i> reports the corresponding number of selection.	92
7.24	First 50 frequently selected features in 10 runs by FS phase of the proposed method using <i>SRC</i> index. The <i>y-axis</i> lists the top 50 features ordered by their frequency, while the <i>y-axis</i> reports the corresponding number of selection.	93
7.25	Predicted likes count vs. true likes count by the <i>ridge</i> model with parameter ($\alpha = 0.1$) considering top 50 features selected by the proposed FS method using dCor index. a) For the training (RMSE=41.08), and b) for the test (RMSE=31.03) sets both sampled by the proposed method from the dataset of Instagram users' responses to Big Four's Fall/Winter 2018 fashion week.	96
7.26	Predicted likes count vs. true likes count by the <i>SVR</i> model with parameters (<i>kernel</i> = linear, $\mathcal{C} = 2$, $\varepsilon = 0.9$) considering top 50 features selected by the proposed FS method using dCor index. a) For the training (RMSE=45.6), and b) for the test (RMSE=30.85) sets both sampled by the proposed method from the dataset of Instagram users' responses to Big Four's Fall/Winter 2018 fashion week.	96
7.27	Predicted likes count vs. true likes count by the <i>XGBosst</i> model with parameters (<i>learning rate</i> = 0.1, <i>reg lambda</i> = 1, <i>min child weight</i> = 1, <i>max depth</i> = 6) considering top 50 features selected by the proposed FS method using dCor index. a) For the training (RMSE=58.49), and b) for the test (RMSE=59.1) sets both sampled by the proposed method from the dataset of Instagram users' responses to Big Four's Fall/Winter 2018 fashion week.	97
7.28	Predicted likes count vs. true likes count by the <i>DNN</i> model with parameters (<i>learning rate</i> = 0.001, <i>batch size</i> = 512) considering top 50 features selected by the proposed FS method using dCor index. a) For the training (RMSE=30.62), and b) for the test (RMSE=33.48) sets both sampled by the proposed method from the dataset of Instagram users' responses to Big Four's Fall/Winter 2018 fashion week.	97

List of Tables

2.1	Support vector regressor's kernels employed in the thesis. . . .	10
6.1	Information regarding the resulting datasets from data collection phase implemented in our method.	43
6.2	Characteristics of the datasets created by the proposed method to validate Microsoft Azure's Computer Vision service. . . .	56
6.3	Correlation of each feature type to likes count obtained from dCor index along with the number of columns each provide. .	59
6.4	Summary of the DNN architectures for the proposed method.	64
6.5	Selected hyper-parameters for Ridge, SVR, XGBoost and DNN methods.	64
7.1	Percent of the posts targeting each of the sub-events regarding Big Four's Fall/Winter 2018 fashion week. Each outer rows of the table show which portion of the posts contained hash-tags related to one city, two cities, three cities and four cities respectively, while the inner rows shows the details related to each of the possible combinations from the outer rows. . . .	70
7.2	Details about users' clusters with optimal number of clusters (= 5), in terms of posts numbers, based on their the duration they wait to post again (users' posting waiting time behaviour. Outliers are the 57 users who published more than 500 posts.)	84
7.3	k_means clustering results in approach 1 (considering all the waiting times for all the users in a group at once for a single clustering). Optimal k denotes the the best number of clusters for each group, and boundaries are discretization points in groups (<i>in hours</i>). max_value is the longest waiting time for the users in that group.	84

7.4	k_means clustering results in approach 2 (considering the waiting time vector for each user's sample individually). Optimal k denotes the the best number of clusters for each group, and boundaries are discretization points in groups (<i>in hours</i>). max_value is the longest waiting time for the users in that group.	84
7.5	Detailed information about the base model (ridge regressor $\alpha = 1$) settings and results of its performance metrics on the training and test sets both resulted from the sampling phase of the proposed method to sample from the dataset of Instagram users' responses to Big Four's Fall/Winter 2018 fashion week for the entire experiment period.	90
7.6	Detailed information presenting the hyper-parameters achieved during training models for Ridge, SVR, XGBosst and DNN regressors using 50 first ranked features according to dCor index along with their corresponding performance metrics on the test dataset.	94
A.1	List of hashtags used for collecting the Instagram users' responses to Big Four's Fall/Winter 2018 fashion week for the entire experiment period.	112
A.2	Information about the <i>posts CSV file</i> resulted after cleaning phase by the proposed method from the dataset of Instagram users' responses to Big Four's Fall/Winter 2018 fashion week for the entire experiment period.	113
A.3	Information about the <i>users CSV file</i> resulted after cleaning phase by the proposed method from the dataset of Instagram users' responses to Big Four's Fall/Winter 2018 fashion week for the entire experiment period.	114
A.4	Detailed information presenting the hyper-parameters achieved during training models for Ridge, SVR, XGBosst and DNN regressors using 50 first ranked features according to dCor index along with their corresponding performance metrics on the training and test datasets.	115

Abstract

In the last few years, social media has dominated various aspects of people's life including social events. Users participate more and more in *long-running* periodical events in social media, by sharing their experiences and preferences. This information provides unprecedented opportunities allowing businesses to promote their brands coverage by using word-of-mouth (WOM), that is enabled by the user generated contents (UGCs). Studying social media content popularity by considering the societies' behavioral patterns is, therefore, paramount. In this thesis, we inspect users' engagement motives in long-running events by means of a comprehensive statistical analysis of fashion week events on Instagram. Additionally, we develop a multi-modal approach to solve the problem of post popularity prediction that exploits potentially influential factors and apply it on fashion week events. We employ two metrics for implementing a filter feature selection technique, together with an automated grid search for optimizing hyper-parameters in four regression methods: ridge, support vector regressor, gradient tree boosting and neural networks.

Sommario

Negli ultimi anni, i social media hanno dominato vari aspetti della vita delle persone, tra cui gli eventi sociali. Gli utenti, infatti, partecipano sempre di più a eventi periodici di lunga durata sui social media condividendo le loro esperienze e preferenze. Queste informazioni offrono preziose opportunità che consentono alle aziende di promuovere i loro marchi mediante i contenuti generati dall'utente (UGC) utilizzando il passaparola (WOM). Studiare la popolarità dei contenuti dei social media considerando gli schemi comportamentali delle società diventa, quindi, fondamentale. In questa tesi, vengono esaminate le motivazioni del coinvolgimento degli utenti in eventi di lunga durata attraverso un'analisi statistica degli eventi accaduti durante la settimana della moda e apparsi su Instagram. Inoltre, viene sviluppato un approccio multimodale per risolvere il problema della previsione della popolarità dei post che sfrutta i fattori influenti; l'approccio è applicato agli eventi della settimana della moda. L'implementazione della tecnica di selezione delle caratteristiche del filtro utilizza due metriche e sfrutta il grid search per l'ottimizzazione degli iper-parametri applicandolo a quattro metodi di regressione: ridge, support vector regressor, gradient tree boosting e neural networks.

Chapter 1

Introduction

This chapter provides an overview of the thesis. In section 1.1, we explain the context and motivations which led us to conduct this research. In sections 1.2 and 1.3 we specify the objectives and summarize the proposed solution to address them. Then, section 1.4 presents information about the structure of the thesis.

1.1 Context and Motivations

Over the last few decades, social media has dominated day to day people's life by providing platforms for the users to share content easily. From another point of view, similar to other types of media, it possesses a powerful potential in many events [31], such as Arab spring in 2010 [37] and the 2008 U.S. presidential elections [39]. Besides, social media is one of the valuable source of data in terms of reliability, since it encompasses the information originated from true feelings and experiences of large groups of people, due to its wide accessibility and user friendliness, which is now achieved thanks to the technological advents. Accordingly, studying social media is appealing for many communities. For researches, it is a new door for exploring several unexplained topics such as societies' behavioural patterns. For brands and businesses, social media is a profitable place for commerce, in which they could potentially attract extra public attention with lower cost compared to

other forms of advertising. These advantages are achieved by word-of-mouth (WOM), as social media is a modern type of media governed by individuals who actively produce user-generated contents (UGCs). Furthermore, the events that are covered by social media would potentially promote the users' engagement, due to the information cascade [27] [57], specifically if the events are well-established in the society. Enjoying rewarding properties like being held periodically and extensive coverage by social media, long-running live events are typically appreciated by the users, and this participation would pave the way for the above-mentioned communities to benefit the opportunity even more. With these motivations, one of the promising field of study for uncovering people's preferences and societies' today trends is analyzing the factors which most affect attracting public attention in a post that is published in the frame of an event in social media which we are going to accomplish in this thesis.

1.2 Objectives

As discussed in the previous section, we would like to study people's behaviour in a long-running live event covered by Instagram, by first introducing and providing a high-level overview of such events. After that we analyze a specific case study, namely fashion week events, and inspect the main aspect of it by performing an exploratory data analysis to address the following questions. How are the temporal dynamics of related posts during the events? *i.e.* understanding how the event actually changes the numbers of relevant posts, when it start or finish (how much is the participation). Do users follow patterns regarding for publishing post? What is the geographical distribution of the posts and users? Which brands were tagged the most during the event? And concerning the brands related posts' dynamics, is the event host city is more impactful or the brands? *i.e.* the dynamic of brands' posts signal is more affect by the location of the event or the brands are tagged independent from the location? After acquiring insight about the data through mentioned analysis, the main goal of the thesis would be designing a predictive model to estimate the popularity of a post. Additionally, considering the nature of the problem, interpretability of the model is a crucial demand, since not only a reasonable estimation should be provided, but also we are interested in the factors which contribute the popularity the

most. In this regard, relevant features should be identified through machine learning techniques.

1.3 Proposed Solution

With the mentioned motivations, to address the objectives of this thesis we conducted a comprehensive statistical analysis on dynamic patterns regarding an event, namely fashion week of fall 2018, held in four cities; Milan, New York, Paris and London as a long-running event in a sense that it happens every year and is of high importance for brands in fashion sector. We focused our analysis on two main entities in social media, contents (*post*) and participants who publish the contents (*user*), together with brands activities in the above mentioned event. Instagram was chosen as our target platform since compared to other social media, it attracts less attention in research community. One of the challenges was that, up to the time that we started this thesis, there was no ready and thorough benchmark dataset to be used, in order to accomplish our goal, we have to collect all the posts and users information regarding the fall 2018 fashion week from Instagram API, from a few days before starting the first sub-event up to some days after the last sub-event. After data collection and cleaning steps, we ended up with raw information from around one million posts and 172,000 users. The posts were collected according to their relevance to the fashion week event by mentioning the target *hashtag(s)* in their captions, and the users are the ones publishing those posts. As a result, the main connection among our users' network is through fashion week event. Then we analyzed available information statistically from several perspectives, namely, posts, users and brands in seek to answers for relevant questions.

After acquiring insight about the data and case study through the mentioned analysis, we investigated the problem of post popularity in the frame of fashion week event by following a prospective road map. In order to be able to evaluate our methodology, we divided our data into training and test sets, then pursued a multi-modal approach for predicting posts popularity through regression methods. To build an accurate predictive model, we needed to extract and include some potentially influential features that were not originally in the data, with the rationale that a complex and social related real-world

phenomena such as post popularity would depend on many variables. In this regard, we provide a hierarchy consists of feature types. The features are either related to posts entity like the hashtag information, to users profile such as followers count, to the post content which in our case study is the published image, or to the information about the event. In case of image-related features, we proceeded with considering both low-level visual attributes and high-level semantic properties. We extracted some extra features on top of high-level features such as the estimated average age of people in the images, if any, and showed that these *higher* level features are effective in popularity of the post. As far as we know, no research has utilized these semantically combined features, so it could be a starting point to extract more features of this kind in future researches, since it enhances the interpretability and effectiveness of the final model. In order to avoid unnecessary complexity of the predictive model and overfitting, we adopted a filter feature selection method applying two statistical indices that measure the correlation among the features, Spearman’s rank correlation which is a univariate metric and distance correlation which is multivariate. We found an optimal number of features to be included in the model using machine learning validation techniques, based on which, we selected the most correlated features. For predicting the popularity (likes count), four different types of regressors were chosen; ridge, support vector regressor, gradient tree boosting and neural networks. For all we tuned hyper-parameters using a grid search mechanism. At the end, we tested the performance of our methodology on the test set and reported the results in terms of relevant performance evaluation metrics.

1.4 Structure of the Thesis

The rest of the thesis is organized as follow. Chapter 2 explains some theoretical background regarding the applied methods in the thesis, chapter 3 provides state of the art methodologies concerning the problem of post popularity prediction on social media platforms. Then, in chapter 4, we discuss some aspects of long-running live events by providing a high-level view of the problem and a case study on the fashion week event. In chapter 5, we discuss the methodology. Chapters 6 and 7 explain our implementation experiences regarding the explanatory analysis and the methodology and report the obtained results.

Chapter 2

Background

In this chapter, we provide the reader with the basic theoretical frameworks and concepts used in this thesis. First, we discuss about some of social media concepts, then we explain feature selection approaches, the regressors we employed and the performance metrics and evaluation indices.

2.1 Social Media Networks

Term Web 2.0 has been introduced in 2004 which resulted in a way for utilizing the World Wide Web (WWW) [47]. Social media which is a Web 2.0, blends different platforms across the WWW to provide the ability of creating and/or exchanging user-generated contents (UGC) [69] [58]. There exists several types of online social networks (OSNs) such as virtual social worlds, blogs, collaborative projects, social networking sites, content communities and virtual game worlds [87].

Stated by the Organisation for Economic Co-operation and Development (OECD, 2007), UGC are required to accomplish three fundamentals to be considered as such: first, it needs to be published either on a publicly accessible website or on a social networking site accessible to a selected group of people; second, it needs to show a certain amount of creative effort; and finally, it needs to have been created outside of professional routines and practices [82].

Regardless of the design of the network and how it is managed, an OSN formally is described by a graph where users are the nodes and the relationships between them are the edges. In some OSNs like Twitter¹ and Instagram² the connections are *unilateral*. On the other hand, some OSNs like Facebook³ is connected *bilateral*. To categorize them into different models, the former ones are the model of *following* while the later considered as the *friendship* model. It worth mentioning that the structure of OSNs are extremely dynamic [31] which makes the analysis of the network more challenging.

2.2 Feature Selection

One of the most crucial steps in the analysis of data is *feature selection* (FS) (*a.k.a.* variable selection) especially in dealing with high-dimensional data (data with a high number of features) like social media data, which makes the machine learning methods more challenging. FS is considered as a combinatorial optimization problem aiming at selecting from a set of available features only the relevant and non-redundant ones to build a regressor or classifier with the required performance. FS reduces the computational cost of the model building and makes the regressor more simplified which increases both the model interpretability and its accuracy [18]. Due to the exponential growth in the complexity with an increase in the size of features, large feature vectors dramatically scale down the process of learning. On the other hand, excess number of features potentially cause the model to overfit the training data and consequently losing its generalization capability [51]. As a result, specialized FS techniques must be developed to overcome these issues.

Three main types of FS algorithms comprise *filter-based* (*a.k.a.* open-loop) methods which typically make use of independent metrics, *wrapper* methods and *embedded* methods which both employ dependence measures. In this thesis, we developed FS methods from filter-based category. When using filter-based methods, the importance of individual features (or subsets of features) are determined by evaluating their intrinsic characteristics in terms of some criteria such as dependency, information, distance, consistency and

¹<https://twitter.com>

²<https://www.instagram.com>

³<https://www.facebook.com>

statistical measures [2] [46]. Among their advantages, their ability to scaled to large-sized datasets such as social media datasets and computation effectiveness can be named as the main reasons for their wide use. Moreover, filter-based FS methods measure the chosen criteria independently from any learning task which helps providing a general solution for every regressor or classifier. The filter-based FS methods are divided into *univariate* or *multivariate* methods depending on whether the goodness of a single feature is evaluated or a subset of features [2]. In section 2.4, we will discuss the used metrics from each category.

2.3 Regression Models

Broadly speaking, in the field of machine learning (ML) *supervised learning* methods consist of methods which build models by learning from the samples which contain labels or output value. Regression is a subcategory of the supervised methods which outputs continuous values based on the amount learned from the training data. The following section is dedicated to explain the regressors which are used in this thesis for predicting the popularity of the posts in long-running live events.

2.3.1 Ridge Regression

Ridge regression is one of the most recognized shrinkage methods and it is very similar to *least squares*. Unlike the *subset selection methods* which employ least squares to fit a linear model that contains a subset of the predictors p , models built by shrinkage methods like ridge are fit in a way that all the predictors p are contained in the model, however thanks to the regularization (shrinkage) techniques the coefficient estimates tends towards zero. In the fitting procedure, least squares makes the estimation of $\beta_0, \beta_1, \dots, \beta_p$ which minimize the Residual Sum of Squares (RSS) as follow:

$$\text{RSS} = \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 \quad (2.1)$$

While in Ridge Regression, the coefficients are estimated according to equation 2.2.

$$\text{RSS} + \alpha \sum_{j=1}^p \beta_j^2 \quad (2.2)$$

The second term in equation 2.2 is called *shrinkage penalty*. The non-negative α serves as the *tuning parameter*. By setting the $\alpha = 0$, the penalty term will lose its effect and the ridge regression will produce similar estimates as least squares does. To elucidate, when $\alpha = 0$, it means that ridge regression coefficients estimates will grow and when $\alpha \rightarrow \infty$ they tend to zero.

The main advantage of ridge regression over least square is helping the bias-variance trade-off *i.e.* by increasing α , the variance decrease and bias increases [44].

2.3.2 Gradient Tree Boosting

The *tree ensemble model* is a set of classification and regression trees (CART). The strategy of ensemble models is to build several models to predict the same target and combine them which results in to possibly have a better estimation. *Boosting* algorithms such as *gradient boosting* are a category of ensemble models which by weak training of the learners in a sequential manner, subset of predictors learn from the mistakes of the previous ones. Gradient boosting is associated with three components including: a *loss function* which is needed to be optimized and vary according to the problem, *weak learner* is in charge of making predictions and an *additive model* which is added to minimize the loss function by the weak learners. The main issue of gradient boosting algorithms arises due to their type which are greedy algorithms so they potentially might overfit on the training data.

To have a better control on this issue they can be enriched by some strategies as follow: Defining *tree constraints* such as number of trees, its depth, number of nodes, number of leaves, number of observations per split and minimum improvement to the loss. Additionally, making use of *weighted updates* (Shrinkage) but it slows down the learning. Moreover, *stochastic gradient boosting* and making advantage of penalized gradient boosting are the other

possible enhancements to tackle the overfitting problem [11].

*XGBoost*⁴ is introduced as an implementation of a scalable machine learning system for tree boosting which runs more than ten times faster with respect to other existing solutions on a single machine and it scales well to billions of data [16].

2.3.3 Support Vector Regression

Support Vector Regression (SVR) is one of the popular regression algorithms. The intuition behind the SVR is minimizing the loss function that finds the best hyper-plane with the largest margin while tolerating some errors. The degree in which the error can be tolerated is set by *margin of tolerance* (ε). Accordingly, ε -insensitive error function [7] which computes the error equal to zero if the absolute difference between target and predicted value is lower than the *margin of tolerance* (ε) [81] which is given by equation 2.3.

$$E_\varepsilon(y(X) - t) = \begin{cases} 0, & \text{if } |y(X) - t| < \varepsilon; \\ |y(X) - t| - \varepsilon, & \text{otherwise} \end{cases} \quad (2.3)$$

The error function $y(X) = W^T \phi(X) + b$ computes the linear predictions' cost for the ones which are not inside the margin of tolerance region. The regularized error function is according to the equation 2.4.

$$\mathcal{C} \sum_{n=1}^N E_\varepsilon(y(X_n)) - t_n + \frac{1}{2} \|W\|_2^2 \quad (2.4)$$

Where $\mathcal{C}$ is the regularization parameter which is used to tune the error sensitivity. The equations 2.5 and 2.6 represent two slack variables introduced by SVR in order to relax the constraint on the maximum allowed error [73].

$$t_n \leq y(X_n) + \varepsilon + \xi_n \quad (2.5)$$

⁴<https://github.com/dmlc/xgboost>

$$t_n \geq y(X_n) - \varepsilon - \hat{\xi}_n \quad (2.6)$$

Both linear and non-linear mappings are possible in SVR, among the kernels functions, table 2.1 provides the ones which are employed in this thesis.

Kernel Name	Kernel Function
Linear	$k(x, y) = x^T y + \mathcal{C}$
Radial basis function	$k(x, y) = B_{2p+1}(x - y)$
Polynomial with degree d	$k(x, y) = (x^T y + \mathcal{C})^d$
Sigmoid	$k(x, y) = \tanh(\alpha x^T y + \mathcal{C})$

Table 2.1: Support vector regressor's kernels employed in the thesis.

2.3.4 Deep Neural Networks

Deep feedforward networks, (*a.k.a.* feedforward neural networks(FFNN) or multilayer perceptrons (MLPs)) approximate some function f^* . It defines a mapping $y = f(x; \vartheta)$ and to achieve the best function approximation, the value of the parameters ϑ are learned. The reason to call them *networks* is that they include several functions which make the model as a directed acyclic graph. Figure 2.1 depicts an example of a FFNN composed of *input layer*, three *hidden layers* and *output layer*. The connections between layers are through the *wights* $W^{(l)} = \{w_{ji}^{(l)}\}$ and the output of a *neuron* is dependent only on the previous layers $h^{(l)} = \{h_j^{(l)}(h^{(l-1)}, W^{(l)})\}$ ⁵.

Nonlinear function of x can be represented by a linear model applied on a transformed input $\phi(x)$ where ϕ is a nonlinear transformation describing x by providing set of features for this purpose. Deep learning model tries to learn this mapping and is defined in equation 2.7.

$$y = f(x; \theta, w) = \phi(x; \theta)^T w \quad (2.7)$$

Parameters θ is used to learn ϕ which is a hidden layer from a broad class of functions and parameter w maps $\phi(x)$ to the target output. Then we use

⁵Source: https://chrome.deib.polimi.it/images/b/be/CognitiveRobotics_04_Neural_Networks_2019_v2.pdf

Figure 2.1: Example of a feedforward neural networks with three hidden layers.

optimization algorithms to find ϕ which can be a good representation. *Adam* is one of the adaptive learning rate optimization algorithms introduced in 2014 [49]. Among the reasons which made Adam optimizer widely popular is its fair robustness in selecting hyper-parameters. In some cases, changing the learning rate is needed.

To compute the hidden layers weights values the choice of right *activation function* is crucial. *Rectified Linear Units* (ReLU) uses $g(z) = \max\{0, z\}$ as the activation function. Due to their similarity to linear units, their optimization is easy. They are commonly used on top of an affine transformation:

$$h = g(W^T x + b). \quad (2.8)$$

During the initialization of the parameters, it is recommended to set a small and positive values for all the elements of b like 0.1 which makes it possible to initially activate ReLU for most of the training inputs. The drawback of ReLU is in examples with the activation equal to zero in which learning cannot be done through the gradient-based methods.

During the training phase, the error which is computed as the difference between predicted value and true value will be propagated backward by apportioning them to each node's weights according to the amount of this error

the node is responsible for. This process is called as the *back-propagation* (*a.k.a.* backprob) [71]. While training the large models, there is a possibility to *overfit* the task which results in loosing the generalization capability of the model and increases in the validation set error while the training set error is still decreasing. To cope with the problem, every time that the error on the validation set rises, a copy of the model parameters should be stored to be used as the model parameters after the training termination, this technique known as *early stopping* [28].

2.4 Performance Evaluation Methods

In this section we provide the methods that are used in this thesis.

2.4.1 Evaluation Metrics

In the following three metrics to measure the performance of the method are discussed.

Mean Square Error (MSE) measures the average difference between values predicted by the model and true values. In other words, it is the standard deviation of the residuals and is calculated according to equation 2.9.

$$\text{MSE} = \frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2 \quad (2.9)$$

Where $(y_j - \hat{y}_j)^2$ is the squared difference between each observation and its predicted value.

Root Mean Square Error (RMSE) is the root of MSE and is calculated according to equation 2.10.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2} \quad (2.10)$$

Mean Absolute Error (MAE) is another error measurement metric which does not consider the direction of the errors and averages the magnitude of the errors in a set of predictions. In other words, it averages the absolute value of differences between prediction and actual observation where all individual differences have equal weight. MAE is computed according to equation 2.11.

$$\text{MAE} = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j| \quad (2.11)$$

2.4.2 Distance Correlation Index

Distance correlation (dCor) index proposed by Szekely *et al.* [77] is a statistical test that provides a reliable correlation-based dependence measure between the random vectors. dCor does not require any assumption on data distribution and unlike Pearson's correlation coefficient which is limited to continuous variables, it can be employed for both continuous and discrete random variables. It ranges from zero (no dependency) to 1 (a linear dependency) among vectors.

Let X and Y be two random vectors with finite first moments and if they take values in $\mathbb{R}^p$ and $\mathbb{P}^q$ respectively, the distance covariance between them $\text{dCov}(X, Y)$, or $\mathcal{V}(X, Y)$ is a non-negative number defined by equation 2.12.

$$\text{dCov}(X, Y) = \int_{\mathbb{R}^{p+q}} |\phi_{X,Y}(t, s) - \phi_X(t)\phi_Y(s)|^2 w(t, s) dt ds, \quad (2.12)$$

Where $w(t, s)$ is the euclidean norm in $\mathbb{R}^d$. The distance correlation $\text{dCor}(X, Y)$, or $R(X, Y)$, is calculated according to equation 2.13

$$\text{dCor}(X, Y) = \begin{cases} \frac{\mathcal{V}^2(X, Y)}{\mathcal{V}^2(X, X)\mathcal{V}^2(Y, Y)} & \text{if } \mathcal{V}^2(X, X)\mathcal{V}^2(Y, Y) > 0 \\ 0 & \text{if } \mathcal{V}^2(X, X)\mathcal{V}^2(Y, Y) = 0. \end{cases} \quad (2.13)$$

It is worth mentioning that dCor is sensitive to redundant terms as tested by Brankovic *et al.* [9].

2.4.3 Spearman's Rank Correlation

Spearman's Rank Correlation (SRC) also known as Spearman's rho denoted by ρ is a nonparametric method for measuring the statistical dependency between two variables. The relationship assessment between two variables is described through a monotonic function which determines whether this relationship is linear or non-linear. If all the samples ranks are distinct integers, it is computed according to equation 2.14.

$$r_s = 1 - \frac{6\sum_i d_i^2}{n(n^2-1)} \quad (2.14)$$

where d_i is the difference between two ranks of observations and n is the sample size. Its advantage over the other correlation metrics like Pearson is its robustness to the outliers placed in the tails of both samples which is because of limiting them to the values of their rank [20].

2.4.4 Validation Methods

To enhance the generalization capability of the method and its performance evaluation, some methods should be applied to test the model on *unseen* data. Among these methods Hold-Out Cross Validation (HOCV) and k -Fold Cross Validation (k -FCV) are the popular ones that are employed in this thesis.

Hold-Out Cross Validation (HOCV) is a commonly used validation approach that keeps some samples as test data and the rest will be used for the learning procedure. The error will be reported according to the performance metric applied on the test part. One of the drawback of this method is its high dependency on the data division procedure that is done for sampling. As an example, it may happen that the training or test samples will be the result of a split which has divided the features and/or values unevenly which will results error that is biased toward a specific part of the distribution. The other drawback is the reduction in training sample size. To limit the effects of the mentioned drawbacks, it is suggested to repeat this procedure for more times randomly and calculate the average error of each iteration [44].

k -Fold Cross Validation (k -FCV) splits the data into k mutually exclusive parts. It reserves one of the parts for the testing time of the regressor while the others are used for training. This procedure is repeated k times and each time the error of the test part will be calculated and at the end, the errors are averaged among all the k iterations. The benefit of this method is that it guarantees the use of all the samples in the training and once in the test [44].

Chapter 3

Related Works

In this chapter we provide some of the relevant state of the art methodologies for predicting post popularity.

One of the relevant study in this context is done by khosla *et. al* in 2014, entitled *What makes an Image popular?* [48]. They investigated key components related to an image for predicting its popularity in social network. One component is the image content that includes low-level and high-level visual features and the second factor is social context such as the number of friends and photos posted by the user while discarding the spatio-temporal features related to the image. In their study, they define popularity as log normalized view counts in Flickr¹. Their work can be considered as one of the first studies which combined both the social cues and image content features. They included subset of features according to categories to find the importance of each, by employing support vector regression (SVR). Their experiment resulted in a Spearman's rank correlation of up to 0.81 using both the mentioned features categories.

Another study is conducted in 2015, used Flickr for data collection and performed classification to find *positively* or *negatively* of an image in terms of the popularity. In their proposed method they have extracted features as follow: *Visual Sentiment Feature* extracted from Visual Sentiment Ontology (VSO) which is collection of 3,244 Adjective-Noun-Pairs (ANPs) defined by

¹<https://www.flickr.com>

[8] and DeepSentiBank [15] which classified images into one of the 2,096 ANPs. They designed two descriptors namely SentANPs and FeatANPs. *Object Features* to detect 1,000 objects from ILSVRC 2014 challenge. *Context Features* such as tags and descriptions, and *User Features*. They utilized support vector machine and included the mentioned features. The final model outperformed the baseline method. They concluded by identifying the ANPs which are positively or negatively correlated with the popularity [25].

Another recent study in 2018 explored information from videos and images collected from three Instagram² business accounts consist of 271 instances for predicting popularity. They have divided the features into five categories namely, time (*i.e.* the season, month, weekday, etc.), common features (*i.e.* media type, topic, width, height, orientation and etc.), text features (*i.e.* hashtags count and whether the post contains text or not), video exclusive features, visual features (*i.e.* RGB channels information) and number of account's follower as the predictive features to predict the popularity score. They have not considered high-level visual features such as objects and faces and other information related to the user profile. They considered popularity score as the number of likes divided by the number of followers. In their study, they regarded the prediction problem as both regression and classification problems. As regression methods, they applied linear regression, local polynomial regression and support vector regressor. They achieved the lowest RMSE of 0.002 using local polynomial regression. Prior the application of the classification, they categorizes popularity into three classes: low, medium, high, then utilized k-nearest neighbor, random forest, naive bayes, C4.5 and decision tree. They achieved the accuracy of 90.77% [90].

Another research in 2014, entitled *The Impact of Visual Attributes on Online Image Diffusion* collected pins from Pinterest³ and chose *number of reshare* as the popularity score. Due to the lack of pinning timestamp, they randomly collected 210,000 user identifiers and using a time span they provided the equal exposure time for the resharing of the images in a more fairly manner. They divided the features into three groups according to what they encode: visual aesthetics properties (*i.e.* RGB channel statistics, basic colors, dominant colors, colorfulness, contrast, etc.), semantic information (*i.e.* low-level features, natural elements, environments, people, etc.), and social-

²<https://www.instagram.com>

³<https://www.pinterest.com>

network properties (category, gender, day of the week, etc.). They reduced the problem into a binary classification whether an image will be highly popular or unpopular and by defining two thresholds for identifying each of the mentioned classes, they have excluded the posts popularity score in a certain margin from the thresholds. In other words, it can be said that they have studied the two extreme classes. To do so, they have employed a random forest ensemble of 200 tree estimators [80].

The other study entitled *Revealing some user practices in Instagram* has collected 1,265,080 posts of 256,398 Instagram users including both the popular and the ordinary ones. In their study, they have excluded the media content and focused on the user-related and post related features. They have concluded that users with more followers receive more attention and popularity, as the *the rich get richer* phenomenon. To add to the aforementioned point, they discovered that use of more hashtags can result in attracting more audiences. At the end, they realized that users tend to post during the weekend and in afternoon while their temporal investigation did not provided information whether such a behavior is the same for long-running live events or not [3].

Another relevant research is conducted in 2014 on Chictopia⁴ which is a fashion-focused online community. They modeled the popularity of outfits pictures by incorporating visual, social and textual factors. To do so, they studied more than 320,000 pictures, from more than 34,000 unique users who posted from March 2008 till Dec. 2012 and chose the number of votes, comments and bookmarks as the observable popularity metrics. By applying linear regression analysis on the log-votes, they confirmed that user’s social features such as their connections dominates the popularity of posts. In addition, they reformulated the problem to a binary classification and built a model for identifying whether the post will be among the top $k\%$ of the most popular ones or not. It resulted in knowing that recognizing most popular posts is easier than the least popular [86].

Multi-modal Learning for Image Popularity Prediction on Social Media is another compatible research in 2016 on 10,000 samples extracted from Yahoo Flickr Creative Commons 100M (YFCC100M) dataset [79]. The authors considered tags and visual features (employing Caffe [45] for visual features

⁴<http://www.chictopia.com>

extraction) in order to model the popularity which was defined as the number of views of the Flickr posts. Accordingly, they built SVR with linear and RBF kernels and Multiple Kernel Learning (MKL). They reported the results in terms of Spearman’s rank correlation among the true and predicted output. The best method was SVR with RBF kernel. Considering both tag and visual features resulted in correlation equal to 0.488, while by the same model and only tags features, correlation was 0.619. They did not apply any feature selection method and considered subset of features according to their types at a time [38].

In 2017, Jaakonmaki *et al.* studied the *The impact of content, context, and creator on user engagement in social media marketing* [43] by exploiting the contributing factors in social engagement regarding 13,000 posts of an Instagram account owned by a German marketing and advertising company. To measure engagement, they summed the number of likes as an indicator of the extent of interest and the number of comments which signals the degree of verbal interaction. They included creator-related, contextual and content features. The later were extracted through the Natural Language Tool Kit (NLTK)⁵ and Clarifai Image Recognition API⁶. They utilized LASSO as the model and found that 40% of the deviance in engagement can be explained by only 10 predictors while to reach 50%, 381 predictors were needed (half of the total number of predictors). The most impactful features were reported mainly the creator-related ones such as the number of followers, age and sex.

In Social Media Prediction Task-1 (SMP-T-1) of the ACM Multimedia 2017 Grand Challenge the second ranked team proposed an approach which was combining multiple features for the image popularity prediction in social media. Accordingly, they studied a dataset consists of 432,000 images from 135 users collected from Flickr and defined the popularity as log transformation of the view count received by the photo. They adopted both ridge regression and gradient boosting regression tree (GBRT). To build the models they included the following features: user, postdate, comment count, has people, title length, tag count, avg view, group count and avg member count, ignoring the visual features. They set up two settings namely, univariate and Ablation. In the former setting, the best result achieved by the GBRT model and the selected features was user, avg member count and group count as

⁵<https://www.nltk.org>

⁶<https://www.clarifai.com>

the most effective ones. The model performance was 2.146 in terms of MSE, however, after including posts date feature to the GBRT model the error decreased to 0.955 [83].

Multi-feature fusion for predicting social media popularity is another attempt in Social Media Prediction Challenge 2017. The authors combined both the low-level and high-level (deep features) visual features with social features including the context-related features such as time and applied linear regression (LR), Matrix Factorization Based on Time and feature Cluster (MFTC) and Support Vector Regression (SVR) to predict the popularity score. Among the models, SVR model using radial-based function (RBF) kernel outperformed by 25% in Spearman’s ranking correlation, and decreased MAE and MSE by 48.5% and 33.7% respectively. They reported that compared to the color names and local maximal occurrence features, deep features increase the Spearman’s rank correlation by 1.6 - 15.8% [59].

A study performed on an Instagram account owned by a popular Indian lifestyle magazine performed Deep Neural Networks (DNN) to predict the popularity of its future posts. The dataset contained 1,280 posts and the features were growth rate in subscriber base, tags associated with the post, time information such as week day and hour, image color descriptors, elapsed time between two consecutive posts and the number of likes of the first post. The metrics have been simplified by using word-tree integration technique. DNN architecture was a four layer Stacked Auto-Encoder network, including four hidden layers followed by a multi-layer perceptron (MLP) with a sigmoid layer as the output for computing the final score and ReLU was used as the activation function. At the end, the average accuracy of the network was 88% [19].

A recent study in 2017 employed the temporal evolution of data and proposed a *Deep Temporal Context Network* (DTCN) to achieve a better performance in the sequential data scenarios. The dataset consisted of 600,000 posts from Flickr. Log-normalized number of views has been defined as the popularity score. The structure of the DTCN consisted of a multi-modal joint embedding for converting user and visual features to an embedding space, while the temporal context learning constructs temporal contexts and learns contextual information thanks to long short-term memory (LSTM). The final prediction process is assisted by multiple time-scale temporal at-

tention mechanism based on temporal attention. They compared DTCN with six baselines CNN-AlexNet [52], CNN-VGG [72], SVR [48], MLP [89], LSTM [36] and CLSTM [26] and SRC was reported which show not only does the DTCN outperform the mentioned methods but also this improvement is significant in most of the cases [84].

Another study was on 65,000 posts crawled from Instagram in order to answer the question *What makes a post belonging to a specific category popular*. They partitioned the posts into four groups, namely, action, scene, people and animal and predicted the popularity defined as the number of likes each category achieved. For each post they extracted the following types of features. Concept features with the help of GoogleNet Inception V3 [75], low-level features of the images through max pooling of the 8×8 convolutional pool layer, visual sentiment features expressed through 1200-dimensional vectors obtained from SentiBank detector [15], word-to-vec features represented as a 300-dimensional vector through the (W2V) model [70], bag-of-word features with the size of 1000-dimension and textual sentiment features employing SentiStrength [78]. The experiment set up in two ways, namely category-mix and category-specific and they employed SVR, random forest regressor (RFR) and multi-layer perceptron (MLP). Spearman’s rank correlation coefficient was the evaluation metric. The results of their work states that prediction based on a specific category increases the accuracy of the prediction. Additionally, information related to the objects and scene have strong descriptive power which provides highest correlation with popularity among all the categories [62].

Multimodal context-aware recommender for post popularity prediction in Social Media, is a research conducted in order to predict the popularity of items (*i.e.* places) considering individuals’ preferences regarding the items in the model. In their study they used a dataset containing 600,000 posts collected from Instagram which are related to different touristic places in The Netherlands (as items). The predictor is designed based on *Factorization Machine* (FM) [68] which has been extended in their case employing visual and textual contents as information. Their results suggest that it is beneficial to apply multi-modal context-aware recommender for modeling the popularity of a post [61].

Another study regarding post popularity prediction proposed a ranking ap-

proach in which considers the visual cues which affect both the popularity and unpopularity of the images. They investigated image popularity prediction as a information retrieval problem with a latent-SVM objective and picked Spearman’s correlation coefficient for the evaluation. Their model was trained separately by both the popular and unpopular latent senses and finally two sets of learned weights are combined to shape it. To do the experiments they have explored VSO dataset [8] which consists of 930,000 posts from Flickr, and another collected dataset consists of 1,000,000 images from Twitter. They compared the performance of their method with Khosla *et al.* [48] and MacParlane *et al.* [21]. The latent model outperforms both in terms of Spearman’s rank correlation by 0.03 and 5% accuracy respectively [14].

Chapter 4

Long Running Live Events

In this chapter, we would like to provide a high-level view of conducting a possible machine learning approach for analyzing and studying long-running live events. This is as a result of the challenges, efforts, and experiences undertaken during this work and our main motivation for presenting this chapter is facilitating future studies in this context. The chapter is organized in two sections. In section 4.1 an abstract level overview of long-running live events is provided. Then, in section 4.2 we discuss an instance of the long-running live events, which is our case study in this thesis.

4.1 High Level Overview

In this section Long-running live events are discussed from high-level point of view. In section 4.1.1 we present the definition of long-running live events and the potential researches that can be performed in this regard, then section 4.1.2 explains the main required elements 4.1.2 of long-running events. Next, in section 4.1.3 the required steps of post popularity prediction will be presented. Finally, in section 4.1.4 we discuss some of the existing challenges.

Long-running live events are defined as the periodically repeated like festivals or cultural events, and the participants usually share their relevant experiences in social media.

4.1.1 Potentials

In the following some of the potential research topics in the context of long-running live events are provided.

- **Content popularity prediction**
- **Profile success classification**
- **Influencers detection**
- **Crowd preferences identification**
- **User profiles clustering**
- **Recommender systems applications**
- **Event detection**
- **Behavioural patterns recognition**

4.1.2 Elements

In this section, we briefly discuss the pillar elements of the long-running live events that should be considered and are depicted in figure 4.1.

Location of the events is one of the elements of long-running events and can be a single particular place or many places. If the event is held in multiple locations, one should investigate that are the events in those places affect each other or not? Furthermore, exploring the events' location will possibly be helpful to consider the cultural preferences and behavior of the participants which will be discussed in the following.

Time of the event can vary occasionally. Also the frequency of the events might change. Some of long-running events have multiple sub-events, and the sub-events can happen in different time of the year. The key point regarding the timing is that the events are holding periodically and it is the motivation for studying them so that one could learn from the past and apply the extracted knowledge on the future ones. The schedule of an event may overlap other events which may cause possible interactions among

Figure 4.1: A simplified overview of the main elements of a long-running live event and the relationships among them.

them which should be inspected too because it might uncover a chain of circumstances.

Context is the topic of the event, for example the context of fashion week events is *fashion* and determine who are the participants and what the types of the generated content(s) are. The context can be related to a single industry or multiple industries. By identifying the main industries engaged in the event we will have a clearer insight about the target group(s) who participate in that event and main type of content in which they will generate. It can be said that it is the context which defines the main goals of the study. As an example, the goal in an event might be spreading of the user-generated contents without considering their popularity, while in another context it could be participants' acceptance. Some examples of different events from different industries are EXPO, Comic-Con and Fashion Week.

Contents such as videos, audios, images and/or texts, are generated by live events’ participants and knowing the type of it in advanced would save money and time. The type of the content which users prefer to generate depends on the nature of the context. For instance, in fashion week events, generating images could be more favorable for the users, since the images are more expressive for sharing their experiences, while in political events, textual participation is required. So, in order to study specific events, one should consider the possible preferable type of the content to select the best tools for processing it. For example, in case of texts, natural language processing methods might be needed while image processing techniques are more useful for the images.

Platforms are the source of the data collection about long-running live events. Due to their abundance, the most informative one should be targeted which depends on the content of the event, as an example in fashion week event the most informative platforms are the ones which provide sharing visual contents. Data collection is a costly step which may poses challenges such as APIs’ limitations, legal issues and heterogeneity of the acquired data from different platforms which should be considered.

Participants of an event are categorized into two groups. The first group consists of the event organizers like brands and industries, and the second group are people who attend the event, for various purposes like shopping or entertainment. The first group of the participants motivation is maximizing the attendance of the second group in the event and the second group are the ones who generate and share the online contents, along with their profile information such as gender, number of followers, number of followings, location and preferences. Extensive part of the state of the art related to content popularity prediction, analyze users’ profiles among the most important factors (see section 3).

4.1.3 Procedure

In this section, we discuss the main steps for studying the long-running live events and predicting posts’ popularity. As shown in figure 4.2, the procedure

starts with choosing the case study event which it is followed by acquiring the domain knowledge, based on which the most suitable platform(s) should be selected to collect data. After that, cleaning of raw data should be performed such as removing noise, missing values and corrupted files. The process continues by applying exploratory data analysis which encompasses statistical analysis of main elements. In this step, it might be required to redo the data collection and cleaning steps as a result of newly acquired information. Then popularity should be defined and formulated considering the context of the event and type of social media. The procedure finishes by applying machine learning techniques to solve the problem of popularity prediction which uncovers the underlying patterns and impactful factors in the long-running live events. In chapter 5 we discuss our approach for post popularity prediction in long-running live events.

Figure 4.2: Overview of the process to extract knowledge from long-running live events.

4.1.4 Challenges

In this section, some of the issues and challenges during studying long-running live events are discussed.

- **API limitations** such as allowed number of requests for data collection
- **Platform privacy regulations modification**

- **Data heterogeneity** resulted from collecting data from different platforms which might hinder applying ML methods and validation phases.
- **High velocity of data** causes noise and missing values in collected data and is usually due to changes performed by users during data collection.

4.2 Case Study

This chapter is intended to provide the necessary information regarding the fashion week¹ event which is considered as one of the world-wide most popular long-running live events. At first, we briefly discuss its history which is a fundamental part to understand some of the underlying factors which have affected its evolution and acceptance. Then we focus on *Big Four's Fashion Week Fall/Winter 2018* as the case study of this thesis and Instagram as the target social media platform to explore.

4.2.1 Fashion Week

History

Before the invent of the smart phones, the fashion was defined simply as promoting the latest products of the couturiers to the wealthy clients thorough small-scale marketing vehicles.²

In late 1800s and early 1900s, it was the acuminous idea of avant-garde couturiers, to make use of around promenades surrounding racetracks which was attracting huge number of people in order to promote their latest products by models which enabled them to become more visible through the lenses of photographers and media in that time. Around the turn of the 20th century, many sophisticated fashion designers privately exhibited their latest designs to their exclusive audiences by hiring mannequins. Notwithstanding, by 1910, due to the popularity of the shows, they turned to scheduled salon-hosted *fashion parades* which lasted for several weeks. Some couturiers have

¹https://en.wikipedia.org/wiki/Fashion_week

²<https://fashionista.com/2016/09/fashion-week-history>

reshaped the events into remarkable social events by sending out invitations to their clients. Astonishing costume parties like *Thousand and Second Night* by Soirée can be considered as steps that revolutionized the interactive catwalks. By 1918, as a result of a booming in the overseas purchasers coming to Europe, the shows became organised with fixed dates, establishing fashion week [10] [23].

Participants

The most high-profile fashion week which attracts most of the press coverage and attentions are held in Milan, New York, Paris and London which are known as fashion capitals (the *Big Four*). During the past years, other cities in the world like São Paulo, Mumbai, Beirut, Berlin, Dubai, Los Angeles, Madrid, Monaco, Rome, Taipei, Shanghai, New Delhi, Vancouver, Copenhagen, Sibiu, Jakarta, Tokyo, Amman, Borneo have joined the ever growing list of cities who host fashion week events.

Hundreds of active brands in different product lines ranging from Apparel to watches and jewelry with different market size participate in Fashion Week events. Some of them are Nike, ZARA, H&M, Louis Vuitton, Fendi, Gucci, Dior, Balenciaga, Calvin Klein, Tommy Hilfiger.

Timing of the events

Traditionally, fashion week events of Big Four capitals are being organized twice a year namely Spring/Summer (*a.k.a.* semi-annual) and Fall/Winter fashion week which vary for each of Big Four from September to October and January to March respectively.³

Fashion and Social media

Figure 4.3, clearly depicts the increase in the people's tendency toward the use of social media platforms.⁴

³<https://fashionweekonline.com/fashion-week-dates>

⁴<https://www.pewresearch.org>

Source: Surveys conducted 2005–2018.
PEW RESEARCH CENTER

Figure 4.3: Percent of U.S. adults who use at least one social media site according to different age ranges, since 2005.

Social media can be considered as one of the most important communication channel which enables brands to participate on online discussions in order to promote their products and enhance their reputation.⁵

One of the main reasons for individuals for following brands' social pages is their passion towards that brand which can be considered as their behavioral loyalty. They are expected to buy the merchandise and services of brands that they are following on social media [64].

Social media content is employed by about 92% of the B2B marketers in North America as the mean of marketing tactics. As reported, 68% of the small and medium sized enterprises (SME) owners also have a profile on social networking sites.⁶ As presented in figure 4.4, the result of a survey conducted during August 2016 on the U.S. consumer social media brand engagement

⁵<https://contentmarketinginstitute.com/2018/10/research-b2b-audience>

⁶<https://contentmarketinginstitute.com/2018/10/research-b2b-audience>

rate provides clear evidence that about 80 percent of the respondents interacted daily with brand posts on social media.⁷

Figure 4.4: Daily U.S. online users' engagement with brands in 2016.

4.2.2 Instagram

Instagram is a video and photo-sharing social networking service that provides the users (business and non-business) with a platform to share their daily experiences and businesses advertisements. Although it started its work as an iOS application, now it is available and widely used in all the other platforms. Importantly, Instagram's social integration design enables users to share their content on a variety of Social Networks (SNs) such as Facebook, Flickr, Twitter and Tumblr. In addition, it actively updates its app compatible with different content editing tools which are among the reasons of high user engagement on Instagram.

⁷<https://www.statista.com>

4.2.3 Big Four's Fashion Week Fall/Winter 2018 on Instagram

We would like to study user generated content popularity on Instagram, related to a specific event of *Big Four's Fashion Week of Fall/Winter 2018*. The motivation for our choice is that compared to the abundant efforts in other social media platforms such as Tweeter and Flickr, Instagram has gained less attention and the number of available datasets gathered from this social media platform is much less. Moreover, the results of a survey conducted by Pew Research Center⁸ (as shown in figure 4.5), confirms that Instagram had the highest growth which accounts for around 29% since 2013 compared to other platforms, so we selected the Instagram as the objective platform for our case study.

Figure 4.5: Most popular social media platforms usages in percent among U.S. adults since 2012.

Among the events typically covered by Instagram, we selected Fall/Winter

⁸<https://www.pewinternet.org/fact-sheet/social-media>

Fashion Week 2018 for Big Four as our target event. To the best of our knowledge, there exists no benchmark dataset regarding the fashion week, so it was needed to collect dataset as part of the work. Figure 4.6 shows the timing of each of the events. London hosts two fashion week events namely *Mens London Fashion Week* (LFW(MENS)) and *London Fashion Week* (LFW) which started in January 6th, 2018 and February 16th, 2018, each lasts for three and five days respectively. However, *New York Fashion Week* (NYFW) doesn't split its events in time slots with a gap in between as the other cities do. Its event started in February 2nd, 2018 and ended in February 20th. Being 19 days long, makes it the longest one among the others. The fashion week event held in Milan consists of two separate events. The first one *Mens Milan Fashion Week* (MFW(MENS)) started in January 12th, 2018 and continued for 4 days while the second one which is *Milan Fashion Week* (MFW) started in February 20th, 2018 which is the last day of both LFW and NYFW, and finished in February 26th, 2018 which is the starting day of Paris fashion week. Paris hosts the second longest event, and follows the same strategy as London and Milan for splitting the events into two sub-events; *Haute Paris Fashion Week* (PFW(HAUTE)) and *Paris Fashion Week* (PFW) each of which lasts for 9 days, starting from January 17th, 2018 and February 26th, 2018 respectively.

Figure 4.6: Big Four's Fall/Winter 2018 events calendar and the experiment period.

4.2.4 Case Study Challenges

Due to the lack of some of the information accessible through the API, the collected dataset could potentially be noisy and less reliable. For example, unlike Flickr as the most similar platform, Instagram API does not provide information about how many times a post has been viewed so far. Limited information makes it difficult to establish a fair definition of the popularity, for example, Instagram API provides only the number of likes and comments at data collection time without keeping likes corresponding timestamps.

Chapter 5

Posts Popularity Prediction

In this chapter, the main idea to predict the post popularity along with the procedure to accomplish it are discussed. Section 5.1 explores different aspect of the definition of popularity in social media. Then in section 5.2, the need for applying sampling and our approach for doing it are discussed. Section 5.3 and 5.4 will explain potentially influential factors that we considered and preparation of the data for predicting popularity. Section 5.5 discuss the challenges and our approach for feature selection, and finally section 5.6 presents our strategy for building the model and evaluating the method.

5.1 Popularity Definition

Although there is no consensus about defining a single fair metric to measure the level of popularity of posts by users in social media, there are several ways to quantify it. The reason is that popularity is a social concept and difficult to judge. These types of phenomena are generally very difficult to model because they depend on many factors, some of them unknown to be measured. It might be convincing that the popularity refers to the extent at which the society positively react to the posts, however, people's reaction to a generated content cannot be considered as the sole way of measuring the real popularity, for example one post can be interesting to society at a specific moment regarding individuals' mood or public trends, while not interesting

in another time. However in most of the research so far, the popularity in social media is related to the amount of attention a post receives. Depending on the type of social media this attention is quantified in different manners. For example in Flickr, the popularity can be considered as the number of views a post acquires, in Pinterest the number of pins and retweeting in Tweeter. Similarly in multiple research regarding post popularity prediction in Instagram, the number of likes of a post (also referred to as *likes count*) has been considered as the unnormalized popularity. In some of the studies [48] [85] [84], a log-normalized form of popularity (equation 5.1) has been used which is a more justified metric where temporal aspects are important in the sense that they seek for the evolution of popularity in time, so, the elapsed time after publishing the post is considered:

$$\text{popularity score} = \log_2 \frac{r}{d} + 1 \quad (5.1)$$

where *popularity score* is normalized value, r is the likes count of a post, and d is the number of days since the post was published. In some other studies [90], the popularity score is obtained simply by computing the ratio between the number of likes and the number of followers (see equation 5.2).

$$\text{popularity score} = \frac{r}{\# \text{ followers}} \quad (5.2)$$

Still, we believe some aspects of a fair popularity score is missing in the state of the art in this context. To the best of our knowledge, none of the metrics considered the visibility of the post. A simple definition of the visibility is the total amount of users for whom a particular post pops up. The most close concept to the visibility might be the number of followers, while the exposure of the post also depends on how many users, the followers follow, because in some social media platform users cannot see the posts from all their following list, besides, even if all the followings' posts are shown, still the order of showing is important. Therefore, to capture the real visibility, the shape of the network should be investigated, as well as the social media's underlying policy regarding a post visibility such as advertising behavior. In general, a wide variety of factors might affect a post visibility, and indirectly the post popularity, since the popularity can never be completely fair if its visibility is unknown, and the visibility should be accessible through the

framework API.

In this thesis, the first metric is not relevant, since the time of data collection is far enough from the event to assume that the number of likes of the posts have been reached to a stable number and would not be changed anymore. The second equation would be also unnecessary, since we have considered the the number of followers as a potential factor to find popularity. As many other studies, we simply consider the number of likes a post has received (likes count) as its popularity which we set as the target and the main goal is providing a predictive model in order to estimate it. In the next sections, all the efforts to achieve this target are explained and the procedure’s steps are summarized in procedure 1.

5.2 Sampling

The original dataset contains about 1,000,000 samples (the posts) and is larger than minimum number of instances to be called *big data* according to its definition in the literature [74] [13] [55]. Unlike many research conducted for post popularity prediction so far such as [90], these samples have been generated from a wide variety of users in terms of profile types. Besides, many unexpected hinders imposed difficult challenges. The challenges include noise and dimension inconsistency which is due to divergent sources and imbalanced target distribution. Another challenge was Instagram API policy changes during data collection and the fact that data is not collected using initial seeds or a specific network of users, but crawled based on manually selected hashtags related to an event.

The imbalance distribution of the target adversely affects the accuracy and robustness of the built model by introducing bias towards the output values with more population. To alleviate this effect, sampling methods can be employed to produce more balanced dataset and particularly analyze the effect of skewness in the target variable distribution.

In the current study, we performed Random Under Sampling (RUS) [4], while preserving the target distribution using stratification technique on the output variable (line 1 in procedure 1). However, considering that not much work has been done on sampling in this context, this could be a worthwhile further

research topic to study the effect of modification in output distribution on the performance of the method including regressions and classifications. This should be noted that We applied a random stratification strategy to split the data into training and test part (line 4 in procedure 1).

5.3 Feature Extractions

Almost in all the studies done for predicting social media content popularity, researchers developed a multi-modal approach [63] [48] [50] [80] [29] [61] [88] [1] [85], *i.e.* incorporating different aspects of posts, particularly characteristics of the post generator and the content of the post which are the features encoded in the posted media. Moreover, some aspects of posts such as hashtags information and generation time are extracted and employed as potential influential factors for building a predictive model depending on the type of social media. In addition, since the case study is about a particular event, other type of features related to the event might have correlation to the popularity of the post. In general all the mentioned features fall into four main categories; *user-related*, *content-related*, *post-related* and *event-related*. User-related features group, also referred to as *social context* properties, gathers information concerning the user who posted, such as their followers and followings count, profile type and so on. Content-related information also depends on the type of social network. For example, in the case of YouTube the posted media is videos or in Tweeter, is text. In Instagram, the major content is images, so this category, should aggregate visual features in the images. Visual features can be low-level (basic) image-related features, mostly resulted from statistical data extracted from pixel-level operations, like image brightness or dominant color. Additionally, visual features include high-level properties which give information about the semantic of the post, for example the presence of particular objects or the images topic. Acquiring these kind of attributes needs more image processing efforts. On top of high-level image-related features one might extract some even *higher* properties obtained from statistical inferences from high level image features. As an example, if the image contain human faces, the estimated average age of the faces could be considered as these type. On the other hand, image is not the sole source of semantic information which might impact the popularity of the post, as there are several examples where the same image has gained differ-

ent level of attention, so we included also post-related category of features which incorporates some other properties such as hashtags, tags, number of comments, *etc.* All these information can be accessible through Instagram API. In this thesis, we also analyze factors correlated to the popularity of the posts in specific events, such as whether the post has been published before, during or after the event period. Figure 5.1 reports the features hierarchy and details about the groups and sub-groups that we extracted in this thesis. We considered the higher level image related attributes in the same category of the high-level features. Feature extraction part as it is shown in line 2 of the procedure 1 is done right after sampling.

Figure 5.1: Hierarchical representation of the case study's features types.

5.4 Data Preprocessing

The extracted raw features cannot be directly inserted into the dataset, and some modifications are necessary before adding them (line 3 in procedure 1). One case is the categorical features which should be quantified in some way, one of them is considering dummy variables for each and assigning numbers to instances accordingly. The problem is that in this case, ML methods including regressors usually consider the order by which categories

are organized. As an example, *dominant color* feature provides 12 colors as the categories, while there is no logical sequence by which the colors should be organized. As a result, we applied one-hot encoding method to the categorical features. Another modification has been applied on the high-level image features extracted from Microsoft Azure’s Computer Vision. The raw form of the features determine high-level concepts such as presence of objects, faces or brands in the images by providing a confidence level between 0 and 1, called score. One option is adding the score for all the detected properties of instances. The other option is descriptizing scores, to be either 0 or 1. We applied the second approach. In order to find the optimal point for imposing a threshold for the scores, we generated several datasets and validated the method for the features that are provided by the scores, the details of such are explained in section 6.3.4.

5.5 Feature Selection

The dataset characterised by large number of features. Depending on the parameters for acquiring low or high-level image features and the attributes provided by Instagram API, total number of features might vary, but still more than 1,000. The high number of features hinders applying ML techniques, both in terms of methods’ running speed and model interpretation. In many real-world problems in which the interpretability of the model is crucial, feature selection (FS) methods can be applied to reduce the computational cost and simplify model structure, which in turn increases the accuracy and robustness of the model and understanding of the dataset. As mentioned in section 2.2, among FS methods, filtering is the fastest, due to its independence to the regressor or classifier error and the most common used in high-dimensional datasets. Filter FS methods rank the features according to properties of data obtained using evaluation metrics. They are categorized into univariate and multivariate. Univariate methods first make a ranking list of the features according to their individual characteristics, then the top ones are selected [22]. However univariate methods ignore the inherent interactions among the features so in many cases redundant terms appear close in the ranking list and consequently will be selected [56]. Multivariate filter methods, however, evaluate the importance of subsets of features according to a metric which is capable of capturing potential redundancy among features.

On the other hand, applying wrapper methods is subject to inefficiency in big data analysis due to computational cost, but still other more complex FS methods such as distributed and randomized FS scheme proposed by [9] can be developed to overcome vertical and horizontal dimensionality curse in the data. Since to the best of our knowledge no effort in employing other dimensionality reduction algorithms except filtering (screening) has been done, it could be a desirable future work for improving the accuracy and robustness of the model.

In this thesis we have employed Spearman’s rank correlation metric [34] which is a univariate criteria, and distance correlation index (dCor) [77] [76] as a multivariate ranking function. As a result, two main modes of FS can be applied according to the type of the metric being used. Both metrics give a score to individual features based on the extent to which the output is dependent on the feature variable and provide a ranked list of features, among which the best ones located in the top of the list are selected. Selecting the suitable number of selected features in this stage would impact the model’s variance and bias trade off; if we select more features than necessary, the model would overfit the training data while, perform weak on the test part. In this regard, to find the optimal number to be screened from the top of the list, we conducted an analysis on the performance of the built model on the validation data with different number of features, which will be discussed in section 6.3.7. The function call `FEATURE SELECTION` in procedure 1 executes the process of selecting features while using the fixed number found in 6.3.7 and returns the best set of the features ($\mathcal{F}^\star$).

5.6 Hyper-parameters Tuning

Typically, learning algorithms aim at optimization problems include some parameters that should be defined before starting the algorithm. These parameters are called hyper-parameters and should be selected such that objective function would be maximized. This process is called hyper-parameter tuning or optimization which formally is a combinatorial problem for finding the best set of parameters in the graph of configuration space so that a loss function is minimized [5]. There are several techniques in the literature to solve this problem according to their approach for exploring the configura-

Procedure 1 Post Popularity Prediction

Input: *Original Data*, *FS param.*, *Regressors*, *Hyperparameters*, *k*.**Output:** $\mathcal{M}_r^*$, Accuracy_r .

```
1: Sampled Data = SAMPLING(Original Data)
2: Data = FEATURE EXTRACTION(Sampled Data)
3: Prepared Data = PREPARE(Data)
4: (TR, TE) = SPLIT(Prepared Data)
5:  $\mathcal{F}^*$  = FEATURE SELECTION(TR, FS param.)
6: for r in Regressors do
7:   for h in Hyperparameters do
8:      $\mathcal{H}^* = k\text{-FOLD-CV}(k, \mathcal{F}^*, h, r)$ 
9:   end for
10:   $\mathcal{M}_r^* = \text{BUILD MODEL}(\mathcal{H}^*, \mathcal{F}^*, r)$ 
11:   $\text{Accuracy}_r = \text{EVALUATE}(\mathcal{M}_r^*, \text{TE})$ 
12: end for
```

tion space, and some take into account the budget in terms of the hardware costs. Examples of hyper-parameters tuning algorithms, are stochastic such as iterated racing [6] method, or random algorithms. Moreover, some of the sequential and greedy algorithms can be applied if the evaluation of the objective function is expensive [40] [41].

However, in this thesis, the computational cost is not of high importance, and we use *grid search* strategy in the sense that given all the hyper-parameters in the configuration space as a tree structure, such that leaf nodes being a set of values for each hyper-parameter configuration, we search through all the nodes by training the model using variables in the nodes. We repeat this procedure for all the regressors as it is stated in line 6 through 12 (the outer *for* loop) in procedure 1. Once the features to be included in the model are selected, for each regressor, a combination of hyper-parameters, as it was discussed in section 2.3, should be tested in a grid search using only training data. For each hyper-parameter configuration (leaf node), we divide the training partition of the data into train and validation sets and evaluate the performance metric (RMSE) of that node using a *k*-fold cross validation on the validation part. Then we select the set of hyper-parameters that gives minimum error in validation set for that specific regressor, and build the final model using the optimal ones. After that, we evaluate the performance of the whole method using the test partition and the built model (line 11).

Chapter 6

Implementation

In this chapter, the implementation experiences and tools employed in order to study the case study in section 4.2 are provided. Section 6.1 gives information about the implementation procedure regarding data collection, cleaning and preparation. Then, sections 6.2 and 6.3 explain the steps for implementing exploratory data analysis and the methodology discussed in chapter 5 respectively.

6.1 Dataset

The following sections (6.1.1 - 6.1.3) explain the implementation stages regarding the collection of the data, cleaning and other necessary preparations to convert from raw JSON format to the final Comma Separated Values (CSV) datasets.

6.1.1 Data Collection

The whole experimental analysis presented in this thesis are based on a database of posts and media shared on Instagram during Big Four Fall/Winter Fashion Week 2018 (see chapter 4.2), collected through the API service [42]. The process began with finding the most used hashtags during the events by manually exploring the Instagram's search function and other

online resources. In September 2018, the final lists of the hashtags for each event (see appendix A.1) have been used to collect publicly available posts. Then, after applying data cleaning phase explained in 6.1.2, we aggregated publicly available posts’ images and all the users’ profiles who published the posts. Collected raw data consisted of two JSON sets namely, posts and users profiles and one image dataset which have been publicly accessible in the time of data collection and the details of them are reported in table 6.1.

Dataset Name	Size	Format
Posts	3,011,320	JSON
Users Profiles	192,205	JSON
Images	723,831	JPG

Table 6.1: Information regarding the resulting datasets from data collection phase implemented in our method.

6.1.2 Data Cleaning

Data collection for a specific topic in social media requires keyword-based search, which naturally ends in extremely noisy results [12]. To achieve a less noisy dataset, applicable data cleaning approaches should be exploited. Data cleaning (*a.k.a.* data cleansing) undertakes detecting and removing errors and inconsistencies from data [67]. In the following, the cleaning tasks that have been done on the posts dataset are discussed.

Duplication removal is the process of removing duplicated posts. These duplicates existed in the dataset due to collecting the posts having multiple hashtags presented in appendix A.1 *i.e.* collecting the same posts multiple times. Therefore, we detected the duplicates by posts’ primary keys (PK) and kept just one instance.

Field error removal is another practice in cleaning the data. Field errors are generally as a result of API or network related problems during data collection step. For example, it might happen that we collect posts with several empty fields which are necessary for our analysis. Although valid, they eradicate the existence of mentioned posts in the dataset, because it

increases the uncertainty due to several NaN values and in turn adversely affects the performance of the method.

Out of interest duration removal is necessary because the API had to inevitably crawl backward from the collection date, which accumulated many unwanted posts published in out of interest dates. As it is stated in figure 4.6, the selected duration for the study starts from Jan. 1, 2018 (five days before the first event which is London Fashion Week Men) till March 11, 2018 (five days after the last event which is Paris Fashion Week). After the field error removal, the data cleaning step have followed by removing the posts which where out of selected duration for the study.

Off-topic removal is another cleaning-related action that we took to eliminate the posts which do not contain any of the target hashtags used for data collection (see appendix A.1). These posts are collected because of Instagram’s API design which retrieves the posts, even if the target hashtags exist in the comment(s) of the posts, and not necessarily just in the caption. It should be noted that the captions are created by the author and only the hashtags in *this* part should be considered for collection criteria, but the API looks also into the comments even when generated by other users. Keeping these posts (*i.e.* the ones with target hashtags only in the comments instead of in the captions) would expose the dataset having irrelevant information. For this reason, they are removed from the dataset.

6.1.3 Data Preparation

Collected posts and user profiles datasets in JSON pose many difficulties for further statistical analysis due to the following reasons. Firstly, because the number of raw JSON files exceeds 20 GB of physical storage which dramatically increases the processing time. Secondly, they include many fields which were not useful. The last but not the least, most of the well-known packages and libraries provided by Python were not optimised for the current format. So we arranged the data in a more appropriate framework, namely CSV, which is more desirable for statistical analysis and it is possible to include only the required fields. The most prominent part of the data are either re-

lated to the posts or the users who have posted, so the mentioned entities are considered as the core body of the generated frameworks. We added some extra features that were not present in the raw datasets for these entities, such as the average number of likes each user obtained, the complete list of which are reported in the appendices A.2 and A.3 with information about the header of CSV files for posts and users respectively, along with their data types and descriptions. The fields already existed in the JSON files are in black and the added ones are highlighted. Primary key to join these two tables (CSV files) is User's PK attribute. The final posts CSV file contains 905,726 posts, each 24 features, and User CSV encompasses 171,078 users, each 24 features.

6.2 Exploratory Data Analysis

As suggested in chapter 4 as fundamental part of studying the long-running live events, to explore the underlying patterns and structure of the data and the case study, we have applied the Exploratory Data Analysis (EDA) techniques on the dataset resulted from previous section regarding three entities of the dataset, namely, *posts*, *users* and *brands*.

6.2.1 Posts Related Analysis

In this section the techniques used to perform the statistical investigation on the post CSV file (see section 6.1.3) are explained.

Hashtags Frequency Analysis

As noted in appendix A.2, posts CSV file contains *Hashtag list* field, on which we performed *Natural Language Processing* to explore statistical information about existing hashtags in the posts captions. We first extracted the hashtags from the caption text body, then obtained the set of unique hashtags and their usage percentage in the posts captions *i.e.* for each hashtag we calculated the ratio between the number of posts containing that particular hashtag to the total number of posts. Finally, by trying different number of k we found top- k which represent the main hashtags of each city.

To have a better illustration of used hashtags, we have employed *WordCloud*¹ library in Python. Word cloud (*a.k.a.* tag cloud) represents the most frequent tags with font size or color which is a suitable approach for perceiving the most prominent words quickly.

Temporal Analysis of the Posts

We investigated information obtained from the posts regarding the date and time they were published. To do so, we considered events (cities) as temporal signals in terms of dates and hours, all starting from the first day of the study until the last day. Then, we assigned all the posts in data to one or some of the four signal events according to the presence of representative hashtags of that particular event in the post. In case posts contain more than one representative hashtags, we assigned multiple labels to them. The magnitude of each event signal at a specific hour is equal to the total number of posts labeled to that event in that hour. It should be noted that the events' representative hashtags are the ones employed for collecting data (see appendix A.1).

Hashtag Relevancy Analysis

In this section, we would like to inspect the extent to which posts are truly related to the event represented by the hashtags in their caption. To achieve this goal, it is possible to add four extra boolean fields, entitled *Milan*, *Paris*, *London* and *New York* to each post (see appendix A.2). Their values represent whether the captions of a specific post contains at least one of the hashtags from the original hashtag list for the data collection (see appendix A.1) corresponds to each city or not. After which we calculated the the percentage of posts relevant to how many cities and making use of *Pyvenn*² package in Python to visualise the degree in which the posts of each city overlaps the others. At the end, we have made two sets containing the users who posted at least one time purely (posting just about one event) and the other containing the ones at least posted one time impurely (posting about more than one event).

¹https://github.com/amueller/word_cloud

²<https://pypi.org/project/venn>

Geographical Analysis

To study how the posts and users entities are geographically distributed, we analyzed the users and posts containing geographical metadata and employed *Basemap*³ and *Folium*⁴ Python packages to show their distribution with scatter and heatmap respectively.

6.2.2 Users' Behavioral Analysis

This section explains the details of the approaches taken to analyze users' behavioral patterns in terms of the duration of time they wait until their next post in the event. In other words, we seek for potential similarities among users in posting time distribution in the desired period. Intuitively, one might consider the whole time of the event as the *time* axis and a binary signal for each user i behaviour ($\mathcal{B}_i(\cdot)$), with 0 when the user did not post and 1 when they posted:

$$\mathcal{B}_i(t) = \begin{cases} 1 & t \in \text{posting time stamp for user } i, \forall i \in \mathcal{U} \\ 0 & \text{others} \end{cases} \quad (6.1)$$

in which $\mathcal{U}$ represents the set of all the users in dataset and i is the index for users. The problem in this setting is that the whole event is almost three months and the number of posts by each user is much less than the period, so we end up with extremely sparse vectors for users. Alternatively, we decided to study the vector containing the time differences between consequent posts, which we call the *waiting time* vector ($\mathbf{w}$), such that:

$$w_{ij} = \mathcal{T}_{j+1}(\mathbf{P}_i) - \mathcal{T}_j(\mathbf{P}_i) \quad (6.2)$$

The value for w_{ij} is the time difference between the post j and post $j + 1$ of i^{th} user in seconds. If $\mathcal{U}$ and $\mathcal{P}$ are the sets of users and posts respectively, $\mathbf{P}_i$ is the sequence of posts for i^{th} user and $\mathcal{T}_j(\cdot)$ collects the timestamp of j^{th} post for users. Since the event period is quite long, (even if we consider

³<https://matplotlib.org/basemap/users/index.html>

⁴<https://python-visualization.github.io/folium>

each event individually), the range of values in waiting time vectors for users are still too wide and comparing behaviour signals ($\mathcal{B}_i(.)$) is ineffective. To tackle the problem of measuring the similarity among users waiting time distributions ($\mathbf{w}_i$), one solution could be discretizing values of these vector into a user defined number of bins, and compare the histograms. However the number of bins that should be defined to incorporate the values of $\mathbf{w}_i$ cannot be the same for all the users. For the clarification, we consider a case in which user u_1 has posted 10 times during the event and user u_2 , 200 times. In this case, vector $\mathbf{w}_{u_2}$ is 20 times lengthier than $\mathbf{w}_{u_1}$ for discretization. As a reminder, the difference between minimum and maximum number of posts in the dataset is terms of several hundreds. This diversity might lead to obtaining more similarity among the users having closer number of posts. This might be intuitive that users with similar post counts have more similar behaviour in terms of posting. In this sense we considered four different groups of users according to the number of times they posted.

Another reason to categorize users is that the average waiting time for each user i is the event period divided by the number of times they posted:

$$\overline{\mathbf{w}}_i = \frac{t_{end} - t_{start}}{|\mathbf{w}_i| + 2} \quad (6.3)$$

It makes more sense to compare vectors having nearer mean. (note that in equation 6.3, $\overline{\mathbf{w}}_i$ is inversely proportional to the number of posts $|\mathbf{w}_i|$) Moreover, for each user, the waiting time vector length would be equal to the number of their posts minus one ($|\mathbf{w}_i| = |\mathcal{P}_i| - 1$), and from implementation point of view, for adding each vector as a sample in a CSV file, zero-padding is inevitable. However, should we place all the users in a single dataset, we obtain lots of zeros while padding for the users with less number of posts. Besides, we may also be able to detect behavioral changes among different groups. Due to these reasons, we decided to categorize users according to their number of posts and analyze each group individually. Clustering methods such as *k-means* can be applied to achieve categorization in this scenario. First, we collected the number of posts of all users in the dataset and obtained a histogram of number of users having posted a particular number of posts.

After removing outliers, we ordered the remaining number of posts as data

points such that each data point corresponds to a user's post number in the resulted histogram (see figure 7.14), then applied *k-means* clustering with different number of clusters. In fact our goal was to cluster the users according to their number of posts and it is natural to categorize users according to their activity level.

Waiting time datasets construction

We have divided the users into groups and created a separated dataset for each category. In each dataset, the time difference between each posts of user (waiting time) is computed. As a result, each row in the datasets corresponds to the waiting time of a specific user and column j of the dataset is the time difference between the post j and post $j + 1$ of users. If $\mathcal{U}$ and $\mathcal{P}$ are the sets of users and posts respectively, $\mathbf{P}_i$ is the sequence of posts for i^{th} user and $\mathcal{T}_j(.)$ collects the time stamp of post j for users:

$$w_{ij} = T_{j+1}(\mathbf{P}_i) - T_j(\mathbf{P}_i) \quad (6.4)$$

Waiting times discretization

Since the ranges of values in waiting time vectors in all groups are very wide, it might be a good practice to discretize them into different ranges and instead of keeping the actual waiting time between posts for each user, store the number of waiting times in discretized ranges. In this way, for each user, a histogram can be obtained and it is feasible to cluster them. However, this discretization should be as meaningful as possible. Accordingly, we tried to find the best ranges for discretization. To achieve this, we investigated two approaches:

Approach 1:

In the first approach, we considered the waiting time vectors values of all the users in each group and applied a one dimensional *k-means* clustering to all the points. Then we computed *WSS* to find the optimal number of clusters using elbow effect. (The optimal number of clusters is the point when

decrease in WSS is less than a threshold by increasing the number of clusters). The breaking points are set to the halfway of point with maximum value in each cluster and the point with minimum value in the very next cluster to obtain histogram bins. A different set of breaking points are computed for different groups and vectors of users' waiting times in distinct groups are discretized based on the defined bins.

Approach 2:

In the second approach, we applied *k-means* clustering in a different manner. Instead of considering all the waiting times for all users in a group at once for a single clustering, we clustered the waiting time vector values for each user's sample *individually*, then we found the optimal number of clusters using elbow effect in WSS for each user. After that, to obtain a unique set of bins for each group, we picked the *mode* of all optimals in the corresponding group, then we clustered each sample/user's waiting time vector values again with the best optimal number of clusters and found a set of bins for each sample. At the end, we picked the *median* of breaking points to align the bins, and made the histograms.

6.2.3 Brands Related Analysis

In this section we performed an analysis on some of the main brands in the frame of fashion week events. Considering the distribution of posts related to each brand in four cities, we would like to know whether the city would affect the coverage of a brand or not, *i. e.* whether the distribution of the number of posts related to a particular brand is dependent on the city hosting the fashion week or not.

We divided the whole time of the fashion week to hours and extracted a temporal signal for each main brand, such that the magnitude of the signal is equal to the number of posts containing the brands' hashtags which were published in that hour, exactly as temporal analysis in section 6.2.1. Then we analyzed temporal dynamics of some of the main brands in the frame of fashion week events. In this regard, we first extracted the total number of posts with the hashtags of the main brands, and reported the results

in section 7.1.3 (figures 7.17 and 7.18), then focused on two of the most frequently tagged ones, *Chanel* and *Dior* by further decomposing their signal to include just the posts tagged in one city at a time and obtained 8 signals presented in figure 7.19. Besides, we computed pair-wise Spearman’s rank correlation among the acquired signals and provided a heatmap (figure 7.20).

6.3 Posts Popularity Prediction

The CSV files containing some user-, post- and event-related attributes which are created as explained in section 6.1.3 can be directly employed to predict the popularity. Yet, the method can be more robust if we could extend the existing dataset to include visual content features, as the popularity might also be dependant to the image that is posted. In this regard, we decided to extract some image content related properties. These properties are categorized into low-level and high-level features. We have utilized Microsoft Azure’s Computer Vision API to extract high-level and some of the low-level features. However, due to the cost limitations, it is not possible to extract these features for all the samples. In this regard, we implemented a mechanism for sampling some instances as the representative for all the dataset. We kept the original dataset available for future analysis. Besides, to validate the accuracy of the Microsoft Azure’s extracted features, we set up an experiment which not only proves the accuracy of the mentioned platform, but also gives a threshold by which it is possible to further improve the accuracy of the method and will be discussed in section 6.3.4.

6.3.1 Sampling

In order to setup the experiment for post popularity prediction, we need to extract high-level image related features as explained in section 5.5. We decided to extract them using existing platforms like Microsoft Azure’s Computer Vision, which provides plenty of high-level image processing services. However, due to the budget limitations, it was impossible for us to process all the images of the original dataset through these services. As a result, we sampled the data and kept 5,583 posts. Nonetheless retained the output distribution using stratification as it is presented in figure 6.1.

Figure 6.1: Distribution of likes count in the posts dataset resulting from: a) Cleaning phase, and b) Sampling phase of the proposed method.

6.3.2 Feature Extraction

The existing attributes in the CSV files (see appendices A.2 - A.3) are not enough for post popularity estimation, due to the lack of information regarding the visual content of the posts (images) which affect post popularity as part of the post entity. For example, there are published posts by the same user or the same tags, while attracting different amount of attention. Consequently, as motivated in section 5.3, image-related information should be extracted and taken into account too. In section 6.3.3, we explain the details about image related features that we extracted, then section 6.3.4 illustrates how we used Microsoft Azure’s Computer Vision service for the extraction and finally section 6.3.5 explains the preparation of the data for further analysis.

6.3.3 Image Related Features

As described in section 5.3, apart from the social context (user-related characteristics), the existing correlation between the popularity of a post and its content suggests that for improving the predictive capability of the model, these kinds of features, namely image-related or visual, should be considered.

These attributes are categorized into high and low-level features, which are extracted as follow.

Low-level features are the features that can be acquired by pixel-level operations and present some statistical information about the image as a signal.

- **Colorfulness** score would be assigned to each image as the quantification of how wide is the range of colors in the image. The score is a real number between 0 and 1. The more an image has different colors, the more the score tends to 1. If the image is black and white it would be considered 0. The score has been implemented as suggested in [33].
- **Dominant color** feature is another type of low-level attribute that we added to the data using Microsoft Azure’s Computer Vision API. The documentation implies that the reference color list contains 12 colors and for each image the API would provide one of them as the dominant color. Then we applied one-hot encoding and added 12 extra columns to the data.
- **HSV** channel consists of *hue*, *saturation* and *value*, and for each image the average of the pixels for the image channels have been computed as different features, which ranges between 0 and 1. Totally it adds 3 columns to the dataset.
- **RGB** is another set of channels for describing colors in images, in which channel *red*, *green* and *blue* represent the intensity of presence of these colors in each pixel. Similar to HSV we added 3 columns to the data containing the average value of each channel throughout the image. The values are originally between 0 and 255, but we normalized them to be consistent to other added features.
- **Entropy** determines to what extent pixel values are similar. In another words, it quantifies the grayness of the image and provides a value between 0 and 1 for each sample.
- **Texture** is another characteristics of the image describing gradient, which could impact the popularity. We included it because human brain has shown different reactions to images with different textures and many researchers have considered it as a potential descriptive set

of features. [48]. Consequently, we implemented *Local Binary Patterns* (LBP) [65] which is a gray scale and rotation invariant texture descriptor and provides 1,024 columns within the range of 0 and 1 to the dataset. After the images are converted to grayscale, the angular space of each pixel in the obtained images is quantized into 1,022 bins, and the spatial resolution of the LBP operator equal to 8.

High-level features are human-perceivable properties of the images for extracting which we employed Microsoft Azure’s Computer Vision API. The list of them are detailed as follow.

- **Category** feature distinguishes and categorizes the semantic topic(s) of the image among 77 existing taxonomies with parent/child hereditary hierarchy, by giving a score which is the probability (between 0 and 1) of belonging an image to the potential detected categories.
- **Moderate content** features detect racy and adult content in the images and provide two columns *Adult* and *Racy* as a confidence score with the values between 0 and 1.
- **Faces** related attributes determine the presence of faces in images and return some properties about the detected faces’ genders and ages. In our approach, the provided faces features have been converted into four columns as follow. *Faces Count* presents the number of detected faces, *Age avg* calculates the average age of the detected faces, *Female Portion* and *Male Portion*, show the ratio of the female and male faces among all identified ones accordingly. It should be noted that these features were added on top of Microsoft Azure features.
- **Tag** features identify potential tags as subject of the image from a set of 1,620 tags including objects, actions and scenery. They provide a confidence score between 0 and 1 which is the probability of the image being related to the detected tags.
- **Brands** feature detects the presence of commercial global brands’ logos with a confidence score between 0 and 1. In our approach, we added one column which contains the count of presented brands in the image if any, zero otherwise.

- **Objects** features provide a confidence score between 0 and 1 if distinguish any objects from a set of 205 in the images.
- **Types** features provide two columns namely *line drawing type* and *clip art type* , the former is 1 if the image is line drawing, 0 otherwise. The later takes one of the integer values between 0 and 3 which are representatives of non-clip-art, ambiguous, normal-clip-art and good-clip-art respectively.

6.3.4 Microsoft Azure’s Computer Vision Validation

As discussed previously, Microsoft Azure’s Computer Vision API provides some low-level information such as dominant color of the image and also services for retrieving high-level information like object detection, presence of human face and image categories. For each detected category, tag or object in an image, it also reports a corresponding score between 0 and 1 which represents the confidence of existing such property. To add them to the dataset there are two strategies. One is employing the scores directly as features values, and the second one is discretizing the values to exactly 0 or 1, *i.e.* considering a threshold above which the feature value (score) is set to 1, and to 0 otherwise. In this thesis, we adopted the second approach. To do so, we designed a procedure by which the accuracy of Microsoft Azure’s Computer Vision services is validated and the best threshold for each type of features to be discretized are acquired. The procedure is the following.

The main feature types are tags with 1,620, categories with 77 and objects with 205 possible labels which can be detected assigned to images along with a score or confidence level. For example, in the *category* feature, Azure’s Vision service can assign *outdoor* label to an image or not. Among these labels we found a subset of the most important ones by looking for the most correlated according to dCor index and the most frequent appeared in the dataset. We extracted these sets for categories, tags and objects, then for each, created a dataset out of the existing samples, and labeled the images manually.

Concerning the objects, seven datasets are built and the presence of car, toy, luggage and bags, glasses, chair, footwear and person in the images are human-labeled in each dataset. Similarly, for categories three datasets

regarding outdoor street, text sign and people, and concerning the tags seven datasets for clothing, fabric, footwear, human face, outdoor, person and street are built. Table 6.2 illustrates the characteristics of the generated datasets. Consequently, we applied discretization according to different thresholds and obtained the accuracy of the Microsoft Azure’s Computer Vision assigned labels by comparing them with the labels committed by human. Figure 6.2 reports the result of the above-mentioned experiment.

Feature group	Dataset name	Size	P	N
Objects	Car	100	50	50
	Toy	100	20	80
	Luggage and bags	100	59	41
	Glasses	100	59	41
	Chair	100	46	54
	Footwear	100	74	26
	Person	100	59	41
Categories	Outdoor street	100	56	44
	Text sign	100	54	46
	People	100	81	19
Tags	Clothing	100	74	26
	Fabric	100	21	79
	Footwear	100	55	45
	Human face	100	71	29
	Outdoor	100	47	53
	Person	100	59	41
	Street	100	50	50

Table 6.2: Characteristics of the datasets created by the proposed method to validate Microsoft Azure’s Computer Vision service.

Figure 6.2: Comparison of accuracy according to different thresholds for discretizing the a)Objects, b)Categories and c)Tags features scores provided by Microsoft Azure's Computer Vision service.

Figure 6.2 suggests that the best thresholds for discretization in objects, categories and tags feature groups are 0.5, 0.1 and 0.5 respectively, since at these points in the corresponding diagrams, there is a clear drop in classification accuracy. So for these features, we set these thresholds as the discretization point.

6.3.5 Data Preprocessing

To prepare the dataset to be used for the post popularity prediction, the following preprocessing steps have been applied.

One-hot encoding for categorical variables has been employed to transform these kinds of features to numerical features for building the model. Using dummy variables are not preferable because they impose logical orders for different values of the feature as discussed in section 5.4. This process is applied to *edited caption*, *verified*, *isBusiness*, *clip art type*, *line drawing type* features.

Normalization step is another necessary action to take by which all the features in the dataset have been normalized using Python's *sklearn* standard scaler package to have zero mean and standard deviation equal to 1, in order to accelerate training of the regressors and hyper-parameters optimization.

6.3.6 Base Model

After preprocessing of the data, we applied a simple regressor to obtain the base error estimate, on top of which we can evaluate the improvement of the performance of proposed method. We applied the default implementation of the ridge offered in *sklearn* library in Python [66] and considered all the extracted features. In the default setting, ridge uses α hyper-parameter equal to 1 which imposes the strongest regularization. We partition the 5,583 samples to training and testing part using stratification described in sections 5.2 and 6.3.1 to keep the same distribution in both parts. About 90% of the data (5,024 samples) is used for training and the rest of the instances are kept for the test data. No feature selection method has been applied, so the number of features is 2,244. The base method is run for 10 times and the average of runs are depicted in figure 7.21 and table 7.5 in section 7.2.1.

6.3.7 Feature Selection

As discussed in 5.5, due to the abundant number of extracted features in the dataset, if the regressors are fit to all of them, it would be unnecessarily complex and subject to overfitting. As a result, it is a good practice to apply a Feature Selection (FS) method. In this thesis, we have implemented two modes of FS, both filter methods (*a.k.a.* ranking or screening) methods, one using Spearman's rank correlation and the other dCor metric for evaluating the dependence of the *likes count* variable to a single or a subset of features. Spearman's rank correlation is a univariate metric, so it only can measure the correlation of a single feature to the output, while dCor, being a multivariate index can also evaluate the dependence of a subsets of variable to the output. In this regard, table 6.3 presents the dCor values between feature categories and the output.

feature types hierarchy		dCor	columns
all features		0.355	2244
event related		0.057	8
	time	0.060	3
	location	0.036	5
post related		0.334	8
user related		0.355	14
	event activities	0.120	7
	activities	0.117	1
	followers	0.457	1
	followings	0.089	1
	business info	0.142	4
image content		0.142	2214
	low-level	0.141	229
	color	0.037	13
	entropy	0.045	1
	size	0.140	2
	hsv	0.045	3
	rgb	0.036	3
	texture	0.041	277
	high-level	0.083	1915
	brand	0.012	1
	adult	0.025	2
	faces	0.082	4
	art type	0.028	6
	obj	0.078	205
	category	0.042	77
	tag	0.103	1620

Table 6.3: Correlation of each feature type to likes count obtained from dCor index along with the number of columns each provide.

Higher dCor values in the table (dCor column) reflect higher correlation. An interesting point in this table is that the sensitivity of the dCor index to redundant terms is clear; Almost in all the categories, adding extra sub-features not only does not increase the correlation, but drops it. For example, consider the *user-related* features category. The dCor value for dependence of the output to the sub-feature *followers* alone is 0.457, but this value for the whole category is considerably lower. The table reveals the same situation for all the other sets and subsets. This is one of the rational behind applying

feature selection before building models. According to the table, the highest value for subcategory followers suggests that visibility of the post generator is more relevant to the popularity than many other attributes. Post and image content related features' dCor values are 0.334 and 0.142 respectively. Among image low-level features, the most correlated feature is image size and the presence of faces among the high-level ones.

As it was discussed, filter methods make a list, ranking the features based on their individual dependence to the output, by placing the most correlated ones on top of the list, makes it possible to select a portion of the list. On the other hand, the table 6.3 suggests that including more features does not necessarily increase the correlation of the input to the output, consequently does not always improve the model as it introduces unnecessary flexibility and overfitting to the model. In this regard, one should find an optimal number of features, above which the performance of the model on the validation data starts to decrease. This phenomena is called *elbow effect*, which elucidates the point where the model starts to learn not from the trustworthy information from the training set, but from the noise, which leads to deteriorating the performance in the test set. Due to the mentioned reason, we implemented an approach, very similar to wrapper FS methods, for finding the most appropriate number of features that should be selected according to our data. In this regard, we trained four types of regressors (Ridge, SVR, XGBoost and DNN), using different number of features. To have a fair comparison, we fixed the regressors' hyper-parameters and stopped any penalization in methods, if any. Then we ran the experiment for 10 times and evaluated the performance in terms of RMSE on the validation set. Figure 6.3 provides the average of the results of the runs.

Figure 6.3: Average of RMSE error *w.r.t.* different number of selected features for the Ridge, SVR, XGBoost and DNN methods for 10 independent runs.

The figure revealed that by increasing the complexity of the model beyond 40 features, the overall performance would drop. Note that, by selecting 20 features which seems be a local minima still RMSE is not as low as RMSE in point 40 for most of the regressors. The reason XGBoost acts very differently perhaps is due to its several hyper-parameters that might indirectly affect overfitting, so future experiments could be fixing each of them at a time. This is also the same for DNN. To have a clearer idea, we illustrate RMSE errors for all the methods with respect to different number of selected features in box plots (figure 6.4), to also illustrate the variance of the results.

Figure 6.4: Boxplot of RMSE error *w.r.t.* different number of selected features for the Ridge, SVR, XGBoost and DNN methods for 10 independent runs.

6.3.8 Hyper-parameter Tuning

For the further possible improvements on top of the base model, firstly, we have exploited other regression methods, namely, SVR, XGBoost, and DNN. The reason for selecting these regressors is analysing the data using different kinds of learning methods. For example, selecting ridge and linear regression at the same time would be unnecessary due to the fact that their learning processes are very similar. Secondly, as discussed in 5.6, complex learning

problems such as applied regression methods, require setting several parameters that should be defined prior to the training process, optimizing which would elevate the performance of the method. Consequently, the action of selecting the best set of hyper-parameters in the configuration space, *a.k.a.* hyper-parameter tuning task, is a crucial step in these methods. For tuning the hyper-parameters, we considered the configuration space as a tree structure in which the leaf nodes represent a unique combination of hyper-parameters for each method. Then, we set up a grid search mechanism by which all the leaf nodes for all the regressors are tested [35] [54] [53]. We also included ridge regression to the grid search and ran the algorithm for 10 independent runs. In each run, the shuffling of the samples are the same for all the regressors, *i.e.* the same samples are in the test set and training set for all the regressors in a single run, but we changed the seed for independent runs to increase the certainty of the results. For the evaluation of each combination, we applied a k -fold cross validation ($k = 5$) in which the initial training set is divided into train and validation parts, so that we ensure all the training set samples are included in the validation part once. The model with the combination of the hyper-parameters is fit to train part for each k , and the performance metrics are measured for the validation part. Then the final performance is the average of k measurements. The grid search output for each regressor would be the best subset of parameters giving the best performance (the least RMSE) on the validation set among other combinations. To implement SVR, XGBoost and DNN, we exploited *sklearn* [66], *xgboost*⁵ and *keras* [17] Python packages respectively. Section 2.3 provides some information about the learning process and the effect of hyper-parameters in these regressors.

In case of DNN, we implemented two sequential architectures [24] [32] [30] which are summarized in table 6.4 (also see section 2.3.4). The results suggests that architecture 2 performs much better than the first one, indicating that the excess layers in architecture 1 adds unnecessary complexity to the model which deteriorate the accuracy. In this regard, we utilized architecture 2 and did not report the results of architecture 1.

⁵<https://xgboost.readthedocs.io/en/latest/python/>

Architecture	Layer (type)	Output Shape	Param #
Arch. 1	input (Dense)	(None, 128)	5248
	hidden 1 (Dense)	(None, 256)	33024
	hidden 2 (Dense)	(None, 256)	65792
	hidden 3 (Dense)	(None, 256)	65792
	output (Dense)	(None, 1)	257
	Total params: 170,113		
	Trainable params: 170,113		
	Non-trainable params: 0		
Arch. 2	input (Dense)	(None, 128)	5248
	output (Dense)	(None, 1)	129
	Total params: 5,377		
	Trainable params: 5,377		
	Non-trainable params: 0		

Table 6.4: Summary of the DNN architectures for the proposed method.

Concerning the hyper-parameters tuning, the details of regression methods along with the candidate values for each hyper-parameter to be checked in the grid search, are reported in table 6.5. All the other parameters not mentioned in the table are fixed to their default values provided by the packages. Further improvement could be analysing the impact of tuning other hyper-parameters in the final performance, specifically, in DNN the effect of changing many hyper-parameters and other potential architectures are yet to be explored. Moreover, other tuning strategies mentioned in section 5.6, would increase the grid search speed.

Method	Hyper-parameter	Values
Ridge	α	0.01, 0.1, 0.5, 1
SVR	kernel	linear, poly, rbf, sigmoid
	C	1, 2, 4, 6
	ε	0.01, 0.1, 0.5, 0.9
XGBoost	learning rate	0.1, 0.2, 0.3, 0.4, 0.5
	reg lambda	1, 2, 3
	min child weight	1, 2, 3
	max depth	2, 4, 6
DNN	learning rate	0.001, 0.01, 0.1
	batch size	32, 64, 128, 256, 512

Table 6.5: Selected hyper-parameters for Ridge, SVR, XGBoost and DNN methods.

Chapter 7

Experimental Results

In this chapter, the results of the implementations explained in chapter 6 are presented and discussed. The chapter is organised in two main parts. In section 7.1, results conducted during exploratory data analysis phase will be presented and discussed. The rest of the chapter as presented in section 7.2 is intended to provide the results of the post popularity prediction method as explained in chapter 5.

7.1 Exploratory Data Analysis Results

In this section, we provide some information about the case study using statistical analysis on the main entities of the long-running live events including posts, users and brands.

7.1.1 Posts Related Analysis Results

In this section the results of statistical investigation performed on the post CSV file as implemented in 6.2.1 are presented and discussed.

Hashtags Frequency Analysis

Total number of hashtags used in all the posts are 13,880,586, which consisted of 476,907 unique hashtags and just 69,353 of the unique ones have been used with a frequency greater or equal than 10 times (14.54%). Considering total number of posts (905,726), the average number of hashtags per post is approximately 15.

Since the distribution of hashtags usage frequency is extremely *heavy-tailed*, it has been presented in the logarithmic scale in the figure 7.1.

Figure 7.1: Usage frequency of the hashtags in Big Four’s Fall/Winter 2018 fashion week. The *x-axis* lists the usage ranks of the hashtags, while *y-axis* reports the logarithm of the frequency.

Top 15 most used hashtags with the percentage of their usage are reported in the figure 7.2. It is worth mentioning that all the main hashtags representing fashion week events in four cities are included in the list.

Figure 7.3 is the *WordCloud* representation of the used hashtags. The words bigger in size and closer to the center are the most frequently used ones.

Temporal Analysis of the Posts

Figure 7.4 illustrates the acquired signals depicted in different colors for each events along with the actual time of events according to the fashion week calendar in the background. For the better overview of the relationship between the posts signals and calendar, the actual time of event color is the same as the signal related to that particular events.

The plot approves that the temporal dynamic of the users posting for the events is in direct relationship with the actual events, coinciding the peak of event signals to the middle of actual events. Besides, the signal value increases sharply just before the start of each corresponding events, continues its growth until the middle of the event, and then decreases moderately after the event conforming the temporal dynamics of popular topics suggested by [31]. This acute dynamic of the signals, motivates our approach for reflecting temporal aspects of posts in feature extraction stage of post popularity prediction as discussed in sections 5.3 and 6.3.2, by one-hot encoding of posts timestamps to be either *before*, *in* or *after* the event.

Concerning individual signals, as it is clear in the figure 7.4, New York event has attracted more attention than the other events, which might be as a result of its single duration. Interestingly, the first peaks in Milan and London being related to men fashion weeks in these cities submit that men fashion week events were less popular.

Figure 7.4: Instagram users' responses to Big Four's Fall/Winter 2018 fashion week for the entire experiment period. The granularity is 1 hour. The related signal to each city is defined by different color and the colored boxes in background, specify the official calendar for each sub-events.

Hashtag Relevancy Analysis

As implemented in section 6.2.1, table 7.1 reports all the possible logical state of a post belonging to the fashion week events and Venn diagram 7.5 presents a better visualization of the table.

Relevant cities	Cities	Percent	Overall Percent
One city	Milan	18.44%	92.866 %
	Paris	23.02%	
	London	20.373%	
	NYC	31.033%	
Two cities	Milan and Paris	1.002%	4.073 %
	Milan and London	0.245%	
	Milan and NYC	0.385%	
	Paris and London	0.508%	
	Paris and NYC	1.4%	
	London and NYC	0.533%	
Three cities	Milan and Paris and London	0.315%	1.819 %
	Milan and Paris and NYC	0.899%	
	Milan and London and NYC	0.132%	
	Paris and London and NYC	0.473%	
Four cities	Milan and Paris and London and NYC	1.242%	1.242 %

Table 7.1: Percent of the posts targeting each of the sub-events regarding Big Four's Fall/Winter 2018 fashion week. Each outer rows of the table show which portion of the posts contained hashtags related to one city, two cities, three cities and four cities respectively, while the inner rows shows the details related to each of the possible combinations from the outer rows.

Figure 7.5: Venn diagram representing the portion of Big Four’s Fall/Winter 2018 fashion week posts contain hashtags of the different combination of cities.

Table 7.1 suggests that majority of posts (92.866%) have hashtags which are all related to one city hashtag list. This means that with more certainty these posts are truly related to that specific event, while the other posts include hashtags related to more than a single city. The later could increase the uncertainty about real association of these posts and the corresponding events. Users who posted these contents might have used these bunch of hashtags just to increase the visibility of the their posts, since each post logically can occur for one of the events, unless the user might have intended to post more general than the scope of a single event (for examples, about *fashion week*

in general, or to compare events of multiple cities. The distinguishable characteristics of these users - who have posts with hashtags related to multiple events (cities) at the same time - can be further analyzed. In the first step, we can categorize users according to their manners in this regard. Diagram 7.6 divides users into two categories (sets):

- **Users with pure posts category:** the users who have posted at least once using hashtags related to just one city.
- **Users with impure posts category:** the users who have posted using multiple event related hashtags, at least once.

The users in the intersection of these two sets perhaps are the ones justifying posts' relevance to the event, because they are the ones having both kinds of posts. This might suggest they know the difference between these kind of captioning. If it was just for visibility, they could have posted with multiple related hashtags all the time. Further analysis could be studying the users' characteristics which represent their inclusion into these categories, *i.e.* understanding the dependence of belonging to these categories and users-related features.

Figure 7.6: Venn diagram categorizing Instagram users who posted about Big Four’s Fall/Winter 2018 fashion week in: Users with pure posts group (*in green*) meaning whether each of their posts contain hashtags related to just one city, in users with impure posts group (*in red*) which each all of their posts contain hashtags related to more than one city and (*in orange*) which is the overlap of two categories meaning having both pure and impure posts.

Geographical Analysis

In this section, we provide the results of investigation conducted in section 6.2.1 to show how the posts and users are geographically distributed, by providing several geographical plots (figures 7.7 - 7.10).

Among 905,726 posts remained after the cleaning phase (see section 6.1.2), 42.59% of them have been geo-tagged. Figure 7.7 depicts how the posts scattered geographically. Posts related to the cities have been shown by

different colors according to their hashtags, not their location provided as the metadata (geo-location obtained from the API). The same scattering plot illustrates dataset users' spatial distribution in figure 7.8. The location of the users are extracted directly from their profiles. Indeed, the red dots in the map account for 53.16% of the users for whom location metadata was available at the time of data collection.

From figures 7.7 and 7.8 it can be grasped that most of the posts have been published in Big Four's regions and mostly by the users who are living in the same regions, or in the other cities which host other fashion week events. In other words, these maps imply that the case study events on Instagram are mainly participated by the local people and attracts less tourists attentions as one might expect. The reason for this claim is that the distribution of the data in these figures are very similar, despite the location information in figure 7.7 is extracted from the post body, while geo information in the second plot (figure 7.8) is extracted directly from the users' profile, without considering the location of posts that specific user published.

In order to analyze and compare the geographical posts distribution, specific to a single event, sub-figures in 7.9 provide the heatmap of the posts related to that particular event individually. On the other hand, figure 7.10 shows the same heatmap distribution, but with a *close-up* of the corresponding continent in which the event was held. Some interesting patterns can be observed from these representations. For example, among European Big Four cities (Milan, Paris and London), Paris fashion week has been targeted by a wider geographical range of posts coming from outside of Paris than the others. In addition, Milan fashion week can be said to be the city in which national users (Italian residence) have made the biggest contribution in terms of publishing posts. Regarding the United States, except New York which is one of Big Four, Los angels and Miami are the cities almost actively engaged in the events of each of Big Four cities specially for the New York. It should be noted that these two cities host their own fashion week.

Figure 7.7: Worldwide geographical dispersion of the Geo-located Instagram posts about Big Four's Fall/Winter 2018 fashion week differentiated by purple, green, blue, red and yellow dots representing posts related to Multiple, Milan, Paris, London and New York respectively.

Figure 7.8: Worldwide geographical dispersion of the Instagram users posted about Big Four's Fall/Winter 2018 fashion week and having Geo-location information in their profile. Each red dot represents a user residence location.

Figure 7.9: Heatmap representing the worldwide geographical density of the Geo-located Instagram posts about Big Four's Fall/Winter 2018 fashion week for Milan, Paris, London, and New York from top to bottom respectively.

Figure 7.10: Heatmap representing the regional geographical density of the Geo-located Instagram posts about Big Four's Fall/Winter 2018 fashion week for a)Paris in Europe, b)London in Europe, c)Milan in Europe, and d)New York in the U.S.

7.1.2 Users Related Analysis Results

In this section some statistical analysis are provided based on information in the users' CSV file.

Followings and Followers

One of the interesting analysis on users could be investigating profiles' following and followers. Even though not completely, this analysis could give some idea about the shape of the network of users in the data.

Figure 7.11 scatters the correlation among the number of followers and fol-

lowings for the users in the dataset, each (dot) point correspond to a single user, with the number of their followings on the x-axis and their followers count on the y-axis. The x and y axes have been limited to show the users having less than 10,000 following and followers. The red line represents line $y = x$. The log-log plot of the same diagram is also provided in figure 7.12.

Figure 7.11: The number of followers (y -axis) vs. number of followings (x -axis) for each user Instagram user's profile who posted about Big Four's Fall/Winter 2018 fashion week, scattered in blue dots. Line $y = x$ is drawn in red and both the x and y axes are limited to 10,000.

In diagram 7.11, the most upper-left part and the bottom-right corner are related to the extreme type of users, probably celebrities and bots with a high unbalance between followers and followings counts. In addition, the majority of the data points fall above the red line which suggests that most of the users have more followers than followings. To be precise, out of 171,078 user profiles in the dataset, the number of users having more followers than followings is

Figure 7.12: The logarithm scaled number of followers (y -axis) vs. logarithm scaled number of followings (x -axis) for each user Instagram user's profile who posted about Big Four's Fall/Winter 2018 fashion week, scattered in blue dots. Line $y = x$ is drawn in red and both the x and y axes are limited to 10,000.

116,191 (67.92%), while the users who follow other accounts more than they are being followed, are 51,559 (30.14%). Only 3,328 accounts (1.94%) have exactly the same number of following as followers which shows our users are mostly influencers. This is reasonable considering the nature of the collected users in the dataset, who are the people who published for a long-running live international event.

From a different perspective, if we consider the community of the users in dataset as a subset of Instagram network as a directed graph, this diagram suggests that the number of incoming vertices (extrenal users who follow the subset of users in dataset) are more than the number of outgoing vertices (the users whom our dataset users follow).

Figure 7.13: Histogram of the number of following (*blue*) and the number of followers (*orange*) on *x-axis* both limited to 10,000, and number of users with the correspondent numbers on *y-axis* for the Instagram user's profile who posted about Big Four's Fall/Winter 2018 fashion week. The peaks with high values are crossed in red.

Moreover, the scatter 7.11 reveals some compelling patterns where the number of following at a few parts of the plot is clearly more than their neighboring areas, for example, around 1,000 and 7,500 on the *x-axis* and only a few users have number of following more than the peak in 7,500, even though the plot in the *x-axis* is limited to interval $[0, 10,000]$. We compared our log-log plot (7.12) to the same plot manikonda *et al.* [60] provided; the general distributions in both are the same, but these artifacts are specific to the users in our dataset. To investigate more, we plotted a histogram which has the followings and followers counts on the *x-axis* and the number of users that have such numbers as their followings and following counts in the *y-axis* (figure 7.13), which confirms that the population of users in these areas has a peak. In a closer look, three most prominent peaks in the following histogram (crossed in red), located in position 0, 999 and 7,500, correspond to the followings count of 840, 169 and 234 users in the dataset. The first peak is as a result of 840 users with exactly 0 followings, but different followers count. They are probably celebrities who decided not to follow anyone, or unused or fake accounts. There is also a possibility that Instagram has

banned some users from following other people. On the other hand, the third peak which collects all the dots in the very right part of the diagram 7.11 are probably the bots, who follow many accounts to attract followers or increase the visibility of their posts or their account. Their overall success in this regards have been reflected in the same figure. Their followers counts on average are less than their followings count. This peak corresponds to 234 users who follow exactly 7,500 accounts, but have different followers count. This specific number could be as a result of Instagram policy which limits users to follow more than 7,500 profiles.

Users' Behavioral Analysis

This section provides the results of studying the users' behaviour as explained in section 6.2.2. Figure 7.14 reports the histogram of the number of users having posted a particular number of posts in a logarithmic scale.

Figure 7.14: Histogram of the number of posts per user for the Instagram user's profile who posted about Big Four's Fall/Winter 2018 fashion week. The *x-axis* is the number of posts, while *y-axis* reports the logarithm number of users having the particular number of posts.

Only 57 users (about 0.033%) have more than 500 posts and can be safely discarded as outliers. Majority of the users have posted less than 10 times.

We decided to keep them in the analysis because given the duration of the event, they are probably real users.

Figure 7.15 presents the result of applying different number of k resulted from *k-means* clustering technique on the users having number of posts less or equal than 500. As the elbow effect in figure 7.15 suggests, $k = 5$ can be considered as the optimized number of clusters for the users. Figure 7.16 shows the number and percentages of the users in each group. The details about the resulting groups after clustering are as follow (see table 7.2).

Figure 7.15: Elbow effect showing the result of clustering in terms of WSS, each time with different number of clusters, to find the optimal users' clusters numbers based on the duration they wait until they post again (their posting waiting time behaviour). The red sign shows the optimal number of clusters ($= 5$) where increasing the numbers does not decrease WSS considerably.

- **Group 1:** users having 1 to 9 posts during the entire event.
- **Group 2:** users having 10 to 35 posts during the entire event.
- **Group 3:** users having 36 to 92 posts during the entire event.
- **Group 4:** users having 93 to 221 posts during the entire event.
- **Group 5:** users having 222 to 500 posts during the entire event.

Figure 7.16: Information about users' clusters with optimal number of clusters (= 5) based on their the duration they wait to post again (users' posting waiting time behaviour) - the clusters (groups) are defined based on users' number of posts in the intervals [1,9], [10,35], [36,92], [93,221] and [222,500] for groups 1 to 5 and above 500 are outliers. a) The number of users in clusters. b) The percentage of users in clusters.

Waiting times discretization

As explained in 6.2.2 we have suggested two ways to apply the waiting times discretization. The resulting bins for approach 1 which is considering all the waiting times for all the users in a group at once for a single clustering, are reported in table 7.3. While the second approach was to cluster the waiting time vector values for each user's sample *individually*, and finding the optimal number of clusters using elbow effect in *WSS* for each user. Final bins for the groups are reported in table 7.4. It should be noted that the obtained bins after clustering samples with this approach were extremely diverse.

	Post #	Centroid	Users #	Users %
Group 1	[1, 9]	2.177	153806	89.904 %
Group 2	(9, 35]	17.126	13665	7.987 %
Group 3	(35, 92]	53.686	2685	1.569 %
Group 4	(92, 221]	131.567	720	0.420 %
Group 5	(221, 500]	311.800	145	0.084 %
Outliers	(500, max]	-	57	0.033 %

Table 7.2: Details about users' clusters with optimal number of clusters (= 5), in terms of posts numbers, based on their the duration they wait to post again (users' posting waiting time behaviour. Outliers are the 57 users who published more than 500 posts.)

	Optimal k	Bins boundaries (hour)
Group 1	6	[0, 62.52, 180.13, 351.81, 588.01, 870.70, max_value]
Group 2	6	[0, 35.99, 108.88, 231.00, 423.83, 702.08 , max_value]
Group 3	6	[0, 14.88, 56.04, 135.20, 283.24, 575.72 , max_value]
Group 4	7	[0, 10.08, 38.57, 103.31, 230.64, 415.41, 685.30 , max_value]
Group 5	6	[0, 8.81, 36.34, 104.95, 246.02, 576.08, max_value]

Table 7.3: k-means clustering results in approach 1 (considering all the waiting times for all the users in a group at once for a single clustering). Optimal k denotes the the best number of clusters for each group, and boundaries are discretization points in groups (*in hours*). max_value is the longest waiting time for the users in that group.

	Optimal k	Bins boundaries (hours)
Group 1	-	-
Group 2	3	[0, 38.06, 157.63, max_value]
Group 3	6	[0, 8.47, 21.38, 39.87, 78.15, 161.60, max_value]
Group 4	6	[0, 5.85, 16.40, 32.54, 59.79, 134.32, max_value]
Group 5	6	[0, 4.07, 12.48, 25.74, 51.93, 119.30, max_value]

Table 7.4: k-means clustering results in approach 2 (considering the waiting time vector for each user's sample individually). Optimal k denotes the the best number of clusters for each group, and boundaries are discretization points in groups (*in hours*). max_value is the longest waiting time for the users in that group.

Further analysis could investigate more the user groups histograms, by finding their similarities in each group in two approaches and compare the results.

7.1.3 Brands Related Analysis Results

In the following we provide the results of the explorations regarding the brands as explained in section 6.2.3.

Brands' participation

Figure 7.17 shows the top targeted brands having more than 1,500 related posts duration the event. Among the depicted ones, *Chanel*, *Gucci* and *Dior* were the most active brands in the events in terms of the number of posts with a considerable difference to the other brands.

Figure 7.17: Top brands having more than 1,500 related posts in Big Four's Fall/Winter 2018 fashion week. The *y-axis* lists the brands ordered by their frequency, while the *x-axis* reports the number of posts contain hashtags related to each brand.

In figure 7.18, the temporal dynamics of the mentioned brands are depicted. As it is clearly obvious from the figure 7.18, the dynamic shapes and magnitude of the brands are different which will be investigated in the following.

Figure 7.18: Instagram users' responses to the brands presented in Big Four's Fall/Winter 2018 fashion week for the entire experiment period. The granularity is 1 hour. The related signal to each brand is defined by different color and just brands with more than 1,500 relevant posts are considered.

Predicting the dynamic of brands

Figure 7.19 compares the temporal dynamics of Dior (*left*) and Chanel (*right*) categorised according to the posts related to each Big Four cities (*top to bottom*).

Figure 7.19: Instagram users' responses to Dior brand in red (*left*) vs. Chanel brand in blue (*right*) in Big Four's Fall/Winter 2018 fashion week for the entire experiment period. The granularity is 1 hour and the signal for each city is plotted separately from top to bottom.

The shapes suggest that the dynamic shape of brands in the same city are

more similar to each other than the dynamic shape of the same brand in different cities. Heatmap presented in figure 7.20 shows the values obtained from the Spearman's correlation metric. Even though the signals of the events were not shifted to be aligned with each other, the results confirm that the city is more impactful than brands in terms of both dynamic shape and magnitude. Let's take Dior posts in Paris as an example, its correlation score with Chanel in Paris is 0.59 which is higher than Dior's scores in different other cities such as Milan which is 0.1. With these results, one may interpret given the temporal dynamic of a brand in a city, it is possible to predict the temporal dynamic of other brands in the same city.

Figure 7.20: Heatmap matrix of the Spearman's correlation analysis showing the correlation coefficients among the values obtained from Instagram users' responses to Chanel and Dior brands for each of the cities in Big Four's Fall/Winter 2018 fashion week for the entire experiment period.

7.2 Post Popularity Prediction Results

In this section we provide the results obtained by applying the method discussed in chapters 5 and 6 on the case study.

7.2.1 Base Model Results

As explained in section 6.3.6, we predicted the popularity using a base model, namely a ridge regressor ($\alpha = 1$), taking all the features into account. We ran the algorithm for 10 times within each, 5,024 samples are shuffled and separated with the ratio of 90:10 into the training and test part. Figure 7.21 and table 7.5 report the results of the prediction.

Figure 7.21: Predicted likes count by the base model (ridge regressor $\alpha = 1$) vs. true likes count considering all the features. a) For the training, and b) for the test sets both resulted from the sampling phase of the proposed method to sample from the dataset of Instagram users' responses to Big Four's Fall/Winter 2018 fashion week for the entire experiment period.

All the obtained metrics in training and test sets are averaged in 10 runs and the mean and std of the results are reported in table 7.5, as well as the best run among all the runs according to the least value for MSE in test data. Note that in the base model the highest degree of regularization is applied on the features due to the default value of α hyper-parameter.

Table 7.5 includes the mean value, standard deviation of the obtained performance metrics along with the best run of the method. These values uncover the sensitivity of the method to the data partitioning, *i.e.* how performance might change with having different samples in training and test parts, which is also the indicator of how reliable the method is. For example in the test set, the RMSE results standard deviation is 4.213, and the best run RMSE is about 7 units different from the mean value of the RMSE, which makes the model less reliable for unseen data since these values in the training set is much lower. On the other hand, comparing RMSE mean value in training and test set (57.503 and 33.161) suggests that the model is overfitted to the train part.

Model	Base model hyper-parameter feature selection number of features	Ridge regression default ($\alpha = 1$) no 2244		
		Mean	std	best run
Training set	MSE	1099.896	29.996	1130.917
	RMSE	33.161	0.452	33.629
	MAE	17.961	0.282	18.088
	Spearman	0.895	0.003	0.894
	dCor	0.933	0.001	0.933
		Mean	std	best run
Test set	MSE	3324.35	484.508	2531.067
	RMSE	57.503	4.213	50.310
	MAE	33.358	1.463	31.412
	Spearman	0.755	0.022	0.765
	dCor	0.810	0.017	0.830

Table 7.5: Detailed information about the base model (ridge regressor $\alpha = 1$) settings and results of its performance metrics on the training and test sets both resulted from the sampling phase of the proposed method to sample from the dataset of Instagram users' responses to Big Four's Fall/Winter 2018 fashion week for the entire experiment period.

7.2.2 Sampling Results

As mentioned in 5.2 and 6.3.1, in order to split the dataset into training and test parts, we have implemented a custom random under sampling by stratification on the likes count to have the distribution of the output the same in training and test set. Figure 7.22 depicts the distribution of the training set likes count vs. the test set.

Figure 7.22: Likes count distributions in the: a) Training, and b) Test datasets.

7.2.3 Feature Selection Results

As discussed in sections 5.5 and 6.3.7, dimensionality reduction (FS) techniques would improve the overall performance of the method, as it removes unnecessary complexity of the model. In this regard, we implemented two modes of FS, one using dCor index and the other employing Spearman's rank correlation index as the metrics for the evaluation of dependence of the output to the individual features. We applied both modes, each for 10 times and plotted the 50 most frequently selected features in both modes in figures 7.23 and 7.24. Note that the results provided in this section do not depend on the regression methods or hyper-parameter tuning outcome, but just depend on the data shuffling and sampling during independent runs. However,

the frequent presence of some features in above-mentioned figures represent the reliability of FS method, suggesting that in most of the sampling of data the same features are selected. On the other hand, the size of intersection of selected features in two modes demonstrates to what extent dCor and Spearman's indices measurements act similar in ranking the features.

Figure 7.23: First 50 frequently selected features in 10 runs by FS phase of the proposed method using *dCor* index. The *y*-axis lists the top 50 features ordered by their frequency, while the *x*-axis reports the corresponding number of selection.

As it was expected, the frequently selected features in figures 7.23 and 7.24 are mainly user-related features such as the ones related to business profile and the number of followers and followings, which emphasizes the importance of influencers' network of connections on the popularity of the posts they publish, and indirectly implies the visibility of the users can have a huge impact on the popularity. This could be one of the future research to study more thoroughly on the users' profile in general, and specifically on the shape of network of their activities on Instagram. Other features that have often been selected were among event-related and high-level image-related features,

such as the presence of certain objects or human faces in the images and the semantic of the images like outdoor scenes. Among post-related features, the number of used hashtags has been selected more often.

In both FS modes, the average age of faces in the images, if any, (*Age avg* in the figures) that was added on top of the high-level image features is being revealed in these plots that has been unanimously selected in all the runs. As far as we know, there has been no effort for retrieving and adding these kinds of features for studying post popularity in social media. These types of features are actually obtained from statistical analysis on other feature types and could have a worthwhile added value, since they might provide hidden social behavioural patterns and people preferences.

Figure 7.24: First 50 frequently selected features in 10 runs by FS phase of the proposed method using *SRC* index. The *y-axis* lists the top 50 features ordered by their frequency, while the *y-axis* reports the corresponding number of selection.

7.2.4 Hyper-parameter Tuning Results

As discussed in sections 5.6 and 6.3.8, the grid search of all the nodes in hyper-parameters configuration space has been selected as our approach. The algorithm is run for 10 times and in each, the samples are shuffled differently, but the samples for all the methods in training and test in a single run is the same for all the regressors so that the comparison among them would be fair. Table 7.6 is the result of tuning hyper-parameters that are involved in the applied regressors. Due to the lack of space, we report only the results on the test set, but the whole table is available in appendix A.4.

Ridge	$\alpha = 0.1$				
		MAE	RMSE	MSE	Spearman dCor
	mean	21.254	38.664	1522.367	0.906 0.915
	std	2.152	5.237	419.669	0.008 0.017
	best	18.893	31.030	962.883	0.914 0.939
SVR	$kernel = \text{linear}, \mathcal{C} = 2, \varepsilon = 0.9$				
		MAE	RMSE	MSE	Spearman dCor
	mean	18.200	40.265	1665.247	0.931 0.924
	std	2.173	6.632	542.022	0.006 0.016
	best	14.895	30.851	951.794	0.941 0.952
XGBoost	$learning\ rate = 0.1, reg\ lambda = 1,$ $min\ child\ weight = 1, max\ depth = 6$				
		MAE	RMSE	MSE	Spearman dCor
	mean	34.133	61.875	3830.605	0.881 0.884
	std	1.059	1.461	180.252	0.010 0.016
	best	32.543	59.102	3493.001	0.895 0.901
DNN	$learning\ rate = 0.001, batch\ size = 512$				
		MAE	RMSE	MSE	Spearman dCor
	mean	21.619	38.345	1482.473	0.893 0.906
	std	1.645	3.482	270.069	0.012 0.016
	best	19.112	33.480	1120.896	0.908 0.930

Table 7.6: Detailed information presenting the hyper-parameters achieved during training models for Ridge, SVR, XGBosst and DNN regressors using 50 first ranked features according to dCor index along with their corresponding performance metrics on the test dataset.

The best obtained result is obtained by training a SVR model with linear kernel, $\mathcal{C} = 2$ and $\varepsilon = 0.9$, as it is depicted bold in table 7.6. In this configuration MAE is equal to 14.895, and Spearman’s rank correlation between output and input equal to 0.952. Unfortunately, there are few studies done on Instagram data and as far as we know, none of them was done on a case study similar to ours. Most of the research on post popularity prediction were conducted either on Flickr or other social media platforms, or on incomparable smaller scales in terms of the number of instances or users’ profile types. However, in this thesis case study, our main criteria for collecting posts was their potential relations with four events held in different countries. Moreover, in most of the other works, the initial seeds were particular group(s) of users, based on which post data were collected, while regarding this case study, we first collected the posts, then we extracted the users who published them. Because of the mentioned reasons, unfortunately it was impossible to compare the obtained results with the others. The only metric which was provided in case of regression on popularity prediction was mostly the correlation among predicted value and true values of popularity, which we provided here. In all the regressors, this correlation is reported to have very small standard deviation among 10 runs of the methods, suggesting that independent from the partitioning of training and test data, the value of correlation is reliable.

For the sake of completeness we provided figures 7.26 - 7.28 which depict the true likes count versus the predicted values in applying the ridge, SVR, XGBoost and DNN regressors which are built using tuned hyper-parameters.

Figure 7.25: Predicted likes count vs. true likes count by the *ridge* model with parameter ($\alpha = 0.1$) considering top 50 features selected by the proposed FS method using dCor index. a) For the training (RMSE=41.08), and b) for the test (RMSE=31.03) sets both sampled by the proposed method from the dataset of Instagram users' responses to Big Four's Fall/Winter 2018 fashion week.

Figure 7.26: Predicted likes count vs. true likes count by the *SVR* model with parameters (*kernel* = linear, $C = 2$, $\varepsilon = 0.9$) considering top 50 features selected by the proposed FS method using dCor index. a) For the training (RMSE=45.6), and b) for the test (RMSE=30.85) sets both sampled by the proposed method from the dataset of Instagram users' responses to Big Four's Fall/Winter 2018 fashion week.

Figure 7.27: Predicted likes count vs. true likes count by the *XGBosst* model with parameters (*learning rate* = 0.1, *reg lambda* = 1, *min child weight* = 1, *max depth* = 6) considering top 50 features selected by the proposed FS method using dCor index. a) For the training (RMSE=58.49), and b) for the test (RMSE=59.1) sets both sampled by the proposed method from the dataset of Instagram users' responses to Big Four's Fall/Winter 2018 fashion week.

Figure 7.28: Predicted likes count vs. true likes count by the *DNN* model with parameters (*learning rate* = 0.001, *batch size* = 512) considering top 50 features selected by the proposed FS method using dCor index. a) For the training (RMSE=30.62), and b) for the test (RMSE=33.48) sets both sampled by the proposed method from the dataset of Instagram users' responses to Big Four's Fall/Winter 2018 fashion week.

One of the most important point uncovered by these figures is that in none of the methods, overfitting has happened during the learning process, as the error rate in training set is comparable to the test set, even in case of XGBoost regressor which is the worst in terms of predicting the output. As demonstrated in figure 7.27, the trained XGBoost model shows a tendency towards predicting popularity less than its true value and most of the data is below the line $y = x$; this could be an indication of insufficiency in grid search configuration space and to investigate more, other XGBoost hyper-parameters should be tuned.

Moreover, the absence of any similar benchmark prevents from judgment about the overall performance of the method. For example, we are not sure whether the obtained value for RMSE is sufficiently good or not, considering the fact that the range of likes count is between 0 and 500, and the regression errors in samples having higher values would extensively deteriorate the results, even if just one sample, as a result of the absence of fair performance metric capable of dismantling this effect. It is worth mentioning that *Mean Absolute Percentage Error* (MAPE) was also utilized in this thesis, but since its value is very similar to MAE, we did not report it. Besides, the imbalanced distribution of the output would further degrade the results by introducing bias in the regressors' learning phase, and should be tackled by applying other sampling methods.

Chapter 8

Conclusions

In this thesis, we performed a comprehensive statistical analysis on the dynamics of fashion week event in four cities on Instagram, as a case study of big data related to a long-running live event in Milan, New York, Paris and London. We collected about 1 million relevant posts, around 172,000 users who generated those posts and posts images from Instagram API. Then we conducted a statistical analysis on the main entities of the data, posts and users, along with an investigation about the active brands in the case study. We reported the most frequent hashtags and a temporal analysis on the posts signal in each city and on the geographical distribution of the posts and users. We also provided detailed information about the following and followers distribution of users, and tried to find users' behavioral patterns for publishing a post. Moreover, we found the most top active brands in the events according to the frequency of related hashtags and answered the question of whether the event city would affect the dynamic of activity of a brand or the brand is more impactful.

Also, we provided a high-level road map for big data analysis on a long-running live event as a case study and explored the problem of post popularity prediction by conducting a multi-modal approach. The data is divided into training and test parts, then a wide variety of feature types were extracted in a hierarchical fashion, including attributes related to posts, users, content and event plus some extra semantic features obtained from high-level image properties. Then a filter feature selection method was applied using

two correlation indices (Spearman’s rank and distance correlation) and the best ones were selected to be included in the model. We found out that the semantic (higher level) type of features could be potentially effective in predicting popularity which could be a possible future study in this context. We chose four types of regression methods, namely ridge, support vector regression, gradient tree boosting and neural networks and performed hyper-parameters tuning, utilizing a grid search mechanism. The final predictive models were evaluated using the test data by several performance metrics. Since there is neither other research similar to our work with the properties of our data and case study, nor a benchmark dataset in the literature, it was impossible to compare our results and we solely reported them.

Bibliography

- [1] Khaled Almgren, Jeongkyu Lee, et al. Predicting the future popularity of images on social networks. In *Proceedings of the The 3rd Multidisciplinary International Social Networks Conference on SocialInformatics 2016, Data Science 2016*, page 15. ACM, 2016.
- [2] Jun Chin Ang, Andri Mirzal, Habibollah Haron, and Haza Nuzly Abdull Hamed. Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection. *IEEE/ACM transactions on computational biology and bioinformatics*, 13(5):971–989, 2015.
- [3] Camila Souza Araújo, Luiz Paulo Damilton Corrêa, Ana Paula Couto da Silva, Raquel Oliveira Prates, and Wagner Meira. It is not just a picture: Revealing some user practices in instagram. In *2014 9th Latin American Web Congress*, pages 19–23. IEEE, 2014.
- [4] Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29, 2004.
- [5] James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems*, pages 2546–2554, 2011.
- [6] Mauro Birattari, Zhi Yuan, Prasanna Balaprakash, and Thomas Stützle. F-race and iterated f-race: An overview. In *Experimental methods for the analysis of optimization algorithms*, pages 311–336. Springer, 2010.
- [7] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

- [8] Damian Borth, Rongrong Ji, Tao Chen, Thomas Breuel, and Shih-Fu Chang. Large-scale visual sentiment ontology and detectors using adjective noun pairs. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 223–232. ACM, 2013.
- [9] Aida Brankovic, Marjan Hosseini, and Luigi Piroddi. A distributed feature selection algorithm based on distance correlation with an application to microarrays. *IEEE/ACM transactions on computational biology and bioinformatics*, 2018.
- [10] Christopher Breward. *Fashion*. Oxford University Press, 2003.
- [11] Jason Brownlee. A gentle introduction to the gradient boosting algorithm for machine learning. *Machine Learning Mastery*. Nov, 9, 2016.
- [12] Emre Calisir and Marco Brambilla. The problem of data cleaning for knowledge extraction from social media. In *International Conference on Web Engineering*, pages 115–125. Springer, 2018.
- [13] Min Cao, Roman Chychyla, and Trevor Stewart. Big data analytics in financial statement audits. *Accounting Horizons*, 29(2):423–429, 2015.
- [14] Spencer Cappallo, Thomas Mensink, and Cees GM Snoek. Latent factors of visual popularity prediction. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 195–202. ACM, 2015.
- [15] Tao Chen, Damian Borth, Trevor Darrell, and Shih-Fu Chang. Deepsentibank: Visual sentiment concept classification with deep convolutional neural networks. *arXiv preprint arXiv:1410.8586*, 2014.
- [16] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.
- [17] François Chollet et al. Keras. <https://keras.io>, 2015.
- [18] Manoranjan Dash and Huan Liu. Feature selection for classification. *Intelligent data analysis*, 1(1-4):131–156, 1997.
- [19] Shaunak De, Abhishek Maity, Vritti Goel, Sanjay Shitole, and Avik Bhattacharya. Predicting the popularity of instagram posts for a lifestyle magazine using deep learning. In *2017 2nd International Conference on*

- Communication Systems, Computing and IT Applications (CSCITA)*, pages 174–177. IEEE, 2017.
- [20] Yadolah Dodge. *The concise encyclopedia of statistics*. Springer Science & Business Media, 2008.
 - [21] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014.
 - [22] Włodzisław Duch, Tomasz Winiarski, Jacek Biesiada, and Adam Kachel. Feature selection and ranking filters. In *International conference on artificial neural networks (ICANN) and International conference on neural information processing (ICONIP)*, volume 251, page 254. Citeseer, 2003.
 - [23] Caroline Evans. The enchanted spectacle. *Fashion Theory*, 5(3):271–310, 2001.
 - [24] Laurene V Fausett et al. *Fundamentals of neural networks: architectures, algorithms, and applications*, volume 3. prentice-Hall Englewood Cliffs, 1994.
 - [25] Francesco Gelli, Tiberio Uricchio, Marco Bertini, Alberto Del Bimbo, and Shih-Fu Chang. Image popularity prediction in social media using sentiment and context features. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 907–910. ACM, 2015.
 - [26] Shalini Ghosh, Oriol Vinyals, Brian Strope, Scott Roy, Tom Dean, and Larry Heck. Contextual lstm (clstm) models for large scale nlp tasks. *arXiv preprint arXiv:1602.06291*, 2016.
 - [27] Manuel Gomez Rodriguez, Jure Leskovec, and Bernhard Schölkopf. Structure and dynamics of information pathways in online media. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 23–32. ACM, 2013.
 - [28] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
 - [29] Steve Göring, Konstantin Brand, and Alexander Raake. Extended features using machine learning techniques for photo liking prediction. In

- 2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6. IEEE, 2018.
- [30] Stephen Grossberg. Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural networks*, 1(1):17–61, 1988.
 - [31] Adrien Guille, Hakim Hacid, Cecile Favre, and Djamel A Zighed. Information diffusion in online social networks: A survey. *ACM Sigmod Record*, 42(2):17–28, 2013.
 - [32] Martin T Hagan, Howard B Demuth, Mark H Beale, and Orlando De Jesús. *Neural network design*, volume 20. Pws Pub. Boston, 1996.
 - [33] David Hasler and Sabine E Suesstrunk. Measuring colorfulness in natural images. In *Human vision and electronic imaging VIII*, volume 5007, pages 87–96. International Society for Optics and Photonics, 2003.
 - [34] Jan Hauke and Tomasz Kossowski. Comparison of values of pearson’s and spearman’s correlation coefficients on the same sets of data. *Quaestiones geographicae*, 30(2):87–93, 2011.
 - [35] Geoffrey Hinton. A practical guide to training restricted boltzmann machines. *Momentum*, 9:1, 2010.
 - [36] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
 - [37] Philip N Howard, Aiden Duffy, Deen Freelon, Muzammil M Hussain, Will Mari, and Marwa Maziad. Opening closed regimes: what was the role of social media during the arab spring? *Available at SSRN 2595096*, 2011.
 - [38] Jiani Hu, Toshihiko Yamasaki, and Kiyoharu Aizawa. Multimodal learning for image popularity prediction on social media. In *2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, pages 1–2. IEEE, 2016.
 - [39] Amanda Lee Hughes and Leysia Palen. Twitter adoption and use in mass convergence and emergency events. *International journal of emergency management*, 6(3-4):248–260, 2009.

- [40] Frank Hutter. *Automated configuration of algorithms for solving hard computational problems*. PhD thesis, University of British Columbia, 2009.
- [41] Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *International Conference on Learning and Intelligent Optimization*, pages 507–523. Springer, 2011.
- [42] Instagram. Instagram developer guide. <https://www.instagram.com/developer/>, May 2018.
- [43] Roope Jaakonmäki, Oliver Müller, and Jan vom Brocke. The impact of content, context, and creator on user engagement in social media marketing. In *Proceedings of the 50th Hawaii international conference on system sciences*, 2017.
- [44] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.
- [45] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.
- [46] Alan Jović, Karla Brkić, and Nikola Bogunović. A review of feature selection methods with applications. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1200–1205. IEEE, 2015.
- [47] Andreas M Kaplan and Michael Haenlein. Users of the world, unite! the challenges and opportunities of social media. *Business horizons*, 53(1):59–68, 2010.
- [48] Aditya Khosla, Atish Das Sarma, and Raffay Hamid. What makes an image popular? In *Proceedings of the 23rd international conference on World wide web*, pages 867–876. ACM, 2014.
- [49] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

- [50] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.
- [51] Sotiris Kotsiantis. Feature selection for machine learning classification problems: a recent overview. *Artificial Intelligence Review*, 42(1):157–176, 2011.
- [52] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [53] Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, pages 473–480. ACM, 2007.
- [54] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [55] Joffrey L Leevy, Taghi M Khoshgoftaar, Richard A Bauder, and Naeem Seliya. A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5(1):42, 2018.
- [56] Yukyee Leung and Yeungsam Hung. A multiple-filter-multiple-wrapper approach to gene selection and microarray data classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 7(1):108–117, 2010.
- [57] Chuang Liu, Xiu-Xiu Zhan, Zi-Ke Zhang, Gui-Quan Sun, and Pak Ming Hui. How events determine spreading patterns: information transmission via internal and external influences on social networks. *New Journal of Physics*, 17(11):113045, 2015.
- [58] Brian D Loader. Social movements and new media. *Sociology Compass*, 2(6):1920–1933, 2008.
- [59] Jinna Lv, Wu Liu, Meng Zhang, He Gong, Bin Wu, and Huadong Ma. Multi-feature fusion for predicting social media popularity. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1883–1888. ACM, 2017.

- [60] Lydia Manikonda, Yuheng Hu, and Subbarao Kambhampati. Analyzing user activities, demographics, social network structure and user-generated content on instagram. *arXiv preprint arXiv:1410.8099*, 2014.
- [61] Masoud Mazloom, Bouke Hendriks, and Marcel Worring. Multimodal context-aware recommender for post popularity prediction in social media. In *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, pages 236–244. ACM, 2017.
- [62] Masoud Mazloom, Iliana Pappi, and Marcel Worring. Category specific post popularity prediction. In *International Conference on Multimedia Modeling*, pages 594–607. Springer, 2018.
- [63] Mayank Meghawati, Satyendra Yadav, Debanjan Mahata, Yifang Yin, Rajiv Ratn Shah, and Roger Zimmermann. A multimodal approach to predict social media popularity. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 190–195. IEEE, 2018.
- [64] Tahir M Nisar and Caroline Whitehead. Brand interactions and social media: Enhancing user loyalty through social networking sites. *Computers in Human Behavior*, 62:743–753, 2016.
- [65] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7):971–987, 2002.
- [66] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [67] Erhard Rahm and Hong Hai Do. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, 23(4):3–13, 2000.
- [68] Steffen Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000. IEEE, 2010.
- [69] Daniel M Romero, Wojciech Galuba, Sitaram Asur, and Bernardo A Huberman. Influence and passivity in social media. In *Joint European Con-*

- ference on Machine Learning and Knowledge Discovery in Databases*, pages 18–33. Springer, 2011.
- [70] Xin Rong. word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*, 2014.
 - [71] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.
 - [72] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.
 - [73] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.
 - [74] Robert C Soltysik and Paul R Yarnold. Megaoda large sample and big data time trials: Separating the chaff. *Optimal Data Analysis*, 2:194–197, 2013.
 - [75] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
 - [76] Gábor J Székely and Maria L Rizzo. The distance correlation t-test of independence in high dimension. *Journal of Multivariate Analysis*, 117:193–213, 2013.
 - [77] Gábor J Székely, Maria L Rizzo, Nail K Bakirov, et al. Measuring and testing dependence by correlation of distances. *The annals of statistics*, 35(6):2769–2794, 2007.
 - [78] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558, 2010.
 - [79] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. The new

- data and new challenges in multimedia research. *arXiv preprint arXiv:1503.01817*, 1(8), 2015.
- [80] Luam Catao Totti, Felipe Almeida Costa, Sandra Avila, Eduardo Valle, Wagner Meira Jr, and Virgilio Almeida. The impact of visual attributes on online image diffusion. In *Proceedings of the 2014 ACM conference on Web science*, pages 42–51. ACM, 2014.
 - [81] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.
 - [82] Graham Vickery and Sacha Wunsch-Vincent. *Participative web and user-created content: Web 2.0 wikis and social networking*. Organization for Economic Cooperation and Development (OECD), 2007.
 - [83] Wen Wang and Wei Zhang. Combining multiple features for image popularity prediction in social media. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1901–1905. ACM, 2017.
 - [84] Bo Wu, Wen-Huang Cheng, Yongdong Zhang, Qiushi Huang, Jintao Li, and Tao Mei. Sequential prediction of social media popularity with deep temporal context networks. *arXiv preprint arXiv:1712.04443*, 2017.
 - [85] Bo Wu, Tao Mei, Wen-Huang Cheng, and Yongdong Zhang. Unfolding temporal dynamics: Predicting social media popularity using multi-scale temporal decomposition. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
 - [86] Kota Yamaguchi, Tamara L Berg, and Luis E Ortiz. Chic or social: Visual popularity analysis in online fashion networks. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 773–776. ACM, 2014.
 - [87] Sheng Yu and Subhash Kak. A survey of prediction using social media. *arXiv preprint arXiv:1203.1647*, 2012.
 - [88] Wei Zhang, Wen Wang, Jun Wang, and Hongyuan Zha. User-guided hierarchical attention network for multi-modal social image popularity prediction. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 1277–1286. International World Wide Web Conferences Steering Committee, 2018.

- [89] Zhengyou Zhang, Michael Lyons, Michael Schuster, and Shigeru Akamatsu. Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron. In *Proceedings Third IEEE International Conference on Automatic face and gesture recognition*, pages 454–459. IEEE, 1998.
- [90] Alireza Zohourian, Hedieh Sajedi, and Arefeh Yavary. Popularity prediction of images and videos on instagram. In *2018 4th International Conference on Web Research (ICWR)*, pages 111–117. IEEE, 2018.

Appendix A

Appendices

City	Hashtags
Milan	#mfw2018, #mfw, #mfwss18, #mfwf, #mfw18, #mfwlive, #mfwreporter, #mfwfw18, #mfwstreetstyle, #milanf18, #mfwplus, #mfwss2018, #cameramoda, #wmfw, #mfwf, #mfwaw18, #milanfashionweek, #milanfashionweek18, #milanfashionweekss18, #milanfashionweek2018, #milanf, #milanfashionweek, #milanfashionweek18, #milanf18, #milanf2018, #milanfashionweek2018, #mfwadventures, #milanf
Paris	#pfwmenswear, #pfw_post, #pfwstreetstyle, #pfwss18, #pfw2018, #pfwss2018, #pfwfw18, #pfwcouture, #parisfw, #parisfashionweek2018, #parisfashionweek, #parisfwss18, #pfwlive, #pfwaw18, #parisfashionweekscenes, #pfw18, #parisfashionweekmens, #par?sfashionweek, #parisfw18, #pfwfashionweek, #pfw
London	#lfw, #lfw18, #lfw2018, #lfw, #lfw2018, #lfwmen, #londonfashionweek18, #londonfw18, #lfashionweek, #londonfashionweek2018, #londonfashionblogger, #lfw, #londonfashion2018, #londonfashionweekmen, #lfwmen, #londonfw, #londonfashionweek, #londonfashion, #londonfashionweekmens
New York	#nyfashionweek, #newyorkfashionweek, #nyfwcastings, #newyorkcityfashionweek2018, #newyorkcityfashionweek, #nycfashionweek, #nycfashionweek2018, #nyfwkidsshows, #nyfw18, #nyfashionweek2018, #nyfw2018, #nyfw2018ss, #nyfwss18, #nyfwss, #nyfwstreetstyle, #nyfw, #nyfw, #newyorkfashionweek2018, #nyfwmen, #nyfw, #nyfw4all, #nyfwblogger, #nyfwbridal, #nyfwaw18, #nyfwmodel

Table A.1: List of hashtags used for collecting the Instagram users' responses to Big Four's Fall/Winter 2018 fashion week for the entire experiment period.

Field	Type	Description
User's PK	String	Unique identifier of the user who published the post
Post's PK	String	Unique identifier of the post
Timestamp	Integer	Unix timestamp of the post
Device Timestamp	Integer	Unix timestamp of the post according to the device
Location	String	The name of the post's location
Latitude	Float	Latitude of the post's location
Longitude	Float	Longitude of the post's location
Edited Caption	String	True if the caption of the post is edited after posting
Media Type	Integer	The media type of the post (Photo, Album, Video)
Likes Count	Integer	The number of likes
Cmnt. Count	Integer	The number of comments on the post
Cmnt. Likes Enabled	Boolean	True if others are allowed to like the post's comments
Tagged Users Count	Integer	The number of tagged users in the media
Hashtag list	String	List of unique hashtags extracted from the caption
Hashtags Count	Integer	The number of used hashtags in the caption of the post
Caption Length	Integer	The number of characters in the caption of the post
Milan	Boolean	True if the post is about Milan FW
Paris	Boolean	True if the post is about Paris FW
London	Boolean	True if the post is about London FW
NewYork	Boolean	True if the post is about NY FW
Multiple	Boolean	True if the post is about more than one event
Time In	Boolean	True if the post's ts is in the target event period
Time Other	Boolean	True if the post's ts is in other events periods
Time None	Boolean	True if the post's ts isn't in none of the events period

Table A.2: Information about the *posts CSV file* resulted after cleaning phase by the proposed method from the dataset of Instagram users' responses to Big Four's Fall/Winter 2018 fashion week for the entire experiment period.

Field	Type	Description
User's PK	String	Unique identifier of the user who owns the account
Media Count	Integer	No. of posts published by the account
Geo Media Cnt.	Integer	No. of Geo-tagged posts the account
Follower Count	Integer	No. of accounts who follow the account
Following Count	Integer	No. of accounts followed by the account
Users Tagged Count	Integer	No. of tagged accounts in the post
Following Tagged Cnt.	Integer	No. of tagged following accounts in post
City Name	String	City's name of the account
Longitude	Float	City's longitude of the account
Latitude	Float	City's latitude of the account
Verified	Boolean	True if the account's owner is verified by Instagram
Is Business	Boolean	True if the account's type is business
Category	String	If business account, shows the business sector
Anon. Profile Picture	Boolean	True if the user did not provide profile picture
Event Posts Count	Integer	No. of user's posts just in the collected dataset
Event Highest Like	Integer	Highest like among the user's posts in the event for the entire experiment period
Event Likes Sum	Integer	The total likes of user's posts in the event for the entire experiment period
Event Likes Avg	Float	The average likes of user's posts in the event for the entire experiment period
Event Likes Med	Float	The median of user's posts likes in the event for the entire experiment period
Event Highest cmnt.	Integer	Highest comment among the user's posts in the event for the entire experiment period
Event cmnt. Sum	Integer	The total comment of user's posts in the event for the entire experiment period
Event cmnt. Avg	Float	The average comment of user's posts in the event for the entire experiment period
Event cmnt. Med	Float	The median of user's posts comments in the event for the entire experiment period
Event Geo-tagged pct.	Float	The percent of the user Geo-tagged posts

Table A.3: Information about the *users CSV file* resulted after cleaning phase by the proposed method from the dataset of Instagram users' responses to Big Four's Fall/Winter 2018 fashion week for the entire experiment period.

Ridge		$\alpha = 0.1$					
			MAE	RMSE	MSE	Spearman	dCor
Training	mean	21.241	41.027	1683.603	0.910	0.915	
	std	0.344	0.617	50.277	0.001	0.002	
	best	21.653	41.802	1747.417	0.908	0.912	
			MAE	RMSE	MSE	Spearman	dCor
Test	mean	21.254	38.664	1522.367	0.906	0.915	
	std	2.152	5.237	419.669	0.008	0.017	
	best	18.893	31.030	962.883	0.914	0.939	
SVR		<i>kernel</i> = linear, $\mathcal{C} = 2$, $\varepsilon = 0.9$					
			MAE	RMSE	MSE	Spearman	dCorr
Training	mean	17.962	44.712	1999.857	0.465	0.919	
	std	0.198	0.849	75.274	0.000	0.002	
	best	18.255	45.596	2078.974	0.465	0.917	
			MAE	RMSE	MSE	Spearman	dCor
Test	mean	18.200	40.265	1665.247	0.931	0.924	
	std	2.173	6.632	542.022	0.006	0.016	
	best	14.895	30.851	951.794	0.941	0.952	
XGBoost		<i>learning rate</i> = 0.1, <i>reg lambda</i> = 1, <i>min child weight</i> = 1, <i>max depth</i> = 6					
			MAE	RMSE	MSE	Spearman	dCor
Training	mean	32.281	58.975	3478.770	0.445	0.910	
	std	0.358	0.880	104.544	0.001	0.009	
	best	32.072	58.485	3420.529	0.445	0.916	
			MAE	RMSE	MSE	Spearman	dCor
Test	mean	34.133	61.875	3830.605	0.881	0.884	
	std	1.059	1.461	180.252	0.010	0.016	
	best	32.543	59.102	3493.001	0.895	0.901	
DNN		<i>learning rate</i> = 0.001, <i>batch size</i> = 512					
			MAE	RMSE	MSE	Spearman	dCor
Training	mean	16.375	28.334	813.203	0.926	0.942	
	std	1.534	3.221	174.110	0.007	0.009	
	best	17.723	30.620	937.577	0.924	0.935	
			MAE	RMSE	MSE	Spearman	dCor
Test	mean	21.619	38.345	1482.473	0.893	0.906	
	std	1.645	3.482	270.069	0.012	0.016	
	best	19.112	33.480	1120.896	0.908	0.930	

Table A.4: Detailed information presenting the hyper-parameters achieved during training models for Ridge, SVR, XGBosst and DNN regressors using 50 first ranked features according to dCor index along with their corresponding performance metrics on the training and test datasets.