

MOSAIC: Modeling Online Sharing of Animal Image Collections

BY

Lorenzo Semeria

Laurea, Politecnico di Milano, Milan, Italy, 2016

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2019

Chicago, Illinois

Defense Committee:

Tanya Berger-Wolf, Chair and Advisor

Chris Kanich

Pier Luca Lanzi, Politecnico di Milano

ACKNOWLEDGMENTS

The work presented in this thesis would have never been possible without the support of many people.

First I would like to express my gratitude towards my academic advisors. Professor Berger-Wolf not only helped me navigate through the complexities and challenges of academic research, but also – and more importantly – shared invaluable advice that guided my life during the past year in the United States. Thanks to professor Lanzi I was introduced to all the tools and challenges of data science and machine learning.

I would not be the person I am today without the support and love of my friends, near and far. A special mention must go to my roommates who were closest to me during this part of my life: Elvio, Paolo, Simone, Claudia, Hélène and Jacopo.

Finally, I must thank my parents and my family, who supported me and enabled me to be the person I am today.

L. S.

TABLE OF CONTENTS

<u>CHAPTER</u>	<u>PAGE</u>
1 INTRODUCTION	1
1.1 Terminology	4
2 RELATED WORK	6
2.1 Online Surveys	6
2.2 Reliability of Mechanical Turk workers	7
2.3 Shareability	7
2.4 Social media as data source	8
2.5 Wildbook	8
3 PROBLEM STATEMENT	10
4 DATASET	12
4.1 Initial dataset	12
4.2 Characteristics of GGR1 and GGR2	13
4.3 Data handling	15
4.4 Result of IBEIS processing	16
4.5 Labeling on MTurk(TM)	17
4.6 Further processing and Feature Engineering	18
4.6.1 Image Level Features	18
4.6.2 Collection level features	19
4.6.3 Further Processing	21
4.7 Preliminary analysis of data	21
4.8 Feature correlation	23
4.9 Final dataset used	24
5 MODELS	36
5.1 Classification Models	36
5.1.1 Baseline classifiers	36
5.1.2 Logistic Regression	36
5.1.3 K-Nearest Neighbors	37
5.1.4 Decision Tree	37
5.1.5 Random Tree Forest	38
5.1.6 Boosting models	38
5.2 Performance Evaluation metrics	39
5.2.1 Evaluating Classification	39

TABLE OF CONTENTS (Continued)

<u>CHAPTER</u>		<u>PAGE</u>
6	EXPERIMENTAL SETUP	41
6.1	Data Collection	41
6.2	Model Creation: pipelines	42
6.3	Main components	42
6.4	Training and evaluating models	44
6.5	Comparing GGR1 and GGR2	47
6.6	Surveys	50
7	EXPERIMENTS	53
7.1	Overall Design	53
7.2	Features to be tested	54
7.3	Survey structure	54
7.3.1	Position in the SD Card	54
7.3.2	Number of animals in the SD Card	56
8	RESULTS	58
8.1	Performance on GGR1	58
8.1.1	Use of collection level features	62
8.1.2	Most important features	63
8.2	Performance on GGR2	66
8.3	GGR1 and GGR2 comparison	67
8.4	Results of our experiments	68
8.4.1	Position	69
8.4.2	Number of Animals in the SD Card	70
9	CONCLUSIONS	73
9.1	Future work	74
	APPENDICES	76
	Appendix A	77
	CITED LITERATURE	86

LIST OF TABLES

<u>TABLE</u>		<u>PAGE</u>
I	DATASET: SUMMARY OF THE COMPOSITION	13
II	IBEIS API: MAIN ENDPOINTS USED	15
III	WELCH'S T-TEST RESULTS FOR SOME FEATURES	23
IV	SELECTION OF PEARSON CORRELATION COEFFICIENTS	24
V	MODEL PARAMETER SETTING	49
VI	PICTURE GROUPS STRUCTURE FOR THE EXPERIMENTS	57
VII	RESULTS FOR BASELINE CLASSIFIER (GGR1)	60
VIII	RESULTS FOR LOGISTIC REGRESSION (GGR1)	60
IX	RESULTS FOR K-NEAREST NEIGHBORS (GGR1)	61
X	RESULTS FOR DECISION TREE (GGR1)	61
XI	RESULTS FOR ENSEMBLE MODELS (GGR1)	62
XII	GGR1 RESULTS, IMAGE-LEVEL FEATURES UNLESS NOTED	64
XIII	RESULTS FOR THE GGR2 DATASET (CROSS-VALIDATION)	70
XIV	RESULTS FOR GGR2, ONLY IMAGE-LEVEL FEATURES . . .	71
XV	RESULTS FOR DUMMY CLASSIFIERS (GGR2)	71
XVI	RESULTS FOR THE MERGED DATASET (GGR1+GGR2) . .	72

LIST OF FIGURES

<u>FIGURE</u>		<u>PAGE</u>
1	Biases that affect data collection	2
2	Steps in data collection and processing	26
3	Map of sightings for GGR1	27
4	Sightings and species for GGR2	28
5	Examples of images from GGR2	29
6	GGR2: a group of elephants	30
7	GGR2: a big group of zebras	30
8	GGR2: a giraffe and some zebras	31
9	Examples of images from GGR1	32
10	Distribution of the “normalized position” feature	33
11	Distribution of the “structural similarity with next” feature	34
12	Distribution of the “arousal” beauty feature	35
13	Main components of the data preprocessing and model training pipelines	44
14	Possible pipelines for logistic regression	45
15	Possible pipelines for k-nearest neighbors	46
16	Possible pipelines for decision tree	47
17	Possible pipelines for our ensemble models	48
18	Structure of a survey	51
19	Example of a single image in a survey	52
20	Examples of images used in our experiments	55
21	Performance of our models on GGR1	59
22	Comparison of the best models in GGR1	63
23	Feature importance (SelectKBest)	65
24	XGBoost feature importance (cover)	66
25	XGBoost feature importance (weight)	67
26	XGBoost feature importance (gain)	68
27	Results for GGR2 (various approaches)	69
28	Feature distributions	78
29	Feature distributions	79
30	Feature distributions	80
31	Feature distributions	81
32	Feature distributions	82
33	Feature distributions	83
34	Feature distributions	84
35	Feature distributions	85

LIST OF ABBREVIATIONS

AMS	American Mathematical Society
CTAN	Comprehensive T _E X Archive Network
MTurk	Amazon Mechanical Turk
TUG	T _E X Users Group
UIC	University of Illinois at Chicago
UICThESI	Thesis formatting system for use at UIC.

SUMMARY

Predicting how people share images on social media is crucial in understanding the bias that affects any image collection found online. For this reason, this work aims at providing a better understanding of how animal pictures are shared with the ultimate goal of improving future estimates based on images extracted from online sources, with a focus on social networks. The focus on images is driven by the availability of effective tools – namely WildbookTM [1] [2] – that allow the identification of individual animals in images. However, obtaining a rich dataset of pictures can be challenging. Using online sources of images, for example social networks, can make the data collection process both cheaper and more extensive. Unfortunately the fact that users arbitrarily choose what images they share online will inevitably bias the dataset – for example, younger individuals may be overrepresented. Understanding this bias is at the core of this work. In order to do it, we created a model to predict which images will be shared from collections.

Our models are designed to take into account both the image-specific features and the collection-specific ones – for example, the structure of the SD card from which the images are chosen (ordering, distribution of species, ...). The introduction of features able to account for the collection in addition to the single image is a novelty and improved the models' performance.

CHAPTER 1

INTRODUCTION

Our world is dominated by social media. Every month, more than two billion users are active on Facebook alone [3], resulting in more than one-quarter of the total world population [4] using this social network. Moreover – and more interestingly for this project – an impressive number of images is shared on social media. In 2013 an average of 350 million images per day was posted on Facebook [5] and this number cannot but have increased since, as the active user population of Facebook doubled since then [6].

It can be easily noticed that images are abundant and, being published online, are easily accessible and cheap to retrieve. Websites dedicated to sharing images (e.g. Flickr, Instagram) make this task even easier for researchers. The abundance of images, coupled with the availability of tools for the detection and identification of animals (such as Wildbook, [1] [2]), allows for the possibility of using social networks as sources of data. Specifically, Wildbook.org focuses its effort for selected species that risk extinction or are considered threatened [7] and allows to estimate the population size from images. The project is continuously evolving in order to improve the estimates and to include more animals in the species it can analyze. The effectiveness of this approach is already been demonstrated in [8], where images collected through the joint effort of many citizen scientists [9] [10] are successfully used to estimate animal populations.

Now it's time to use images from social media, too. For this reason, my research aims at understanding how users share images online. In a real scenario a tourist that goes to a safari

will likely shoot hundreds (if not thousands) of pictures, later deciding which are worthy of being uploaded to a social media platform. This selection will depend on the user’s personal likes as well as characteristics of both the image and the overall composition of the collection of all the pictures he or she shot. This work builds upon the “shareability” introduced by S. Menon in [8] as a measure based on the individual picture’s features only. We expanded in this direction adding features accounting for the structure of the SD card, therefore taking into account more than the individual image characteristics. This will allow for a better understanding of users’ behavior, in turn allowing for a better analysis of bias.

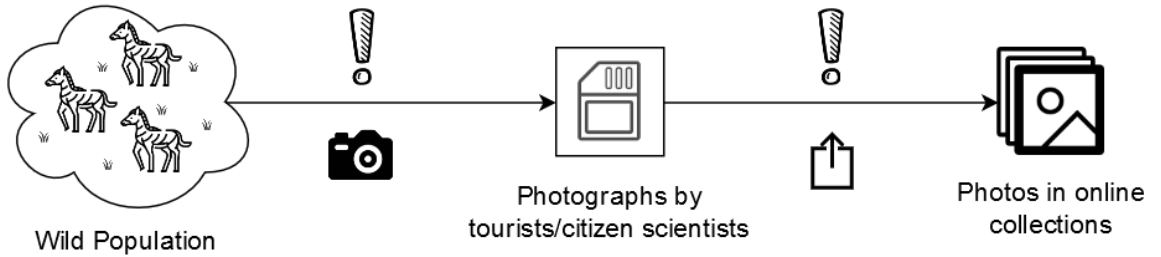


Figure 1. Biases that affect data collection

From the actual wildlife to online available images every step introduces a bias, as noted in [8] and summarized by Figure 1. In this figure, the exclamation marks denote where bias can be introduced. Starting from the left, the users will decide what pictures to take. This adds a first bias. Once the images are taken and stored in the memory, as shown in the figure,

a second bias is added when the user decides which images he wants to share. This second bias is the one we aim at understanding in this work.

We approach the problem of understanding the users’ behavior by analyzing the images that are shared by the users and the original collection from which they chose. The data, which we describe in detail in Chapter 4, was provided by Wildbook and is based off the data collected from the Great Grevy’s Rally events (GGR1 [9] and GGR2 [10]).

The Great Grevy’s rally were both held in Kenya, in 2016 and 2018 respectively. They attracted a high number of citizen scientists that contributed by taking pictures on the field. The goal of these events is to gather a high amount of good quality pictures. Chapter 4 provides more details on these events.

The images gathered through the GGR events were labeled by hand through surveys published on Amazon Mechanical Turk (or MTurk as shorthand) [11]. The images were simply labeled “shared” or “not shared” by the respondents.

The interviewees of MTurk (“workers” in the service terminology) are real people that answer to surveys and are compensated for their time. Workers have been shown to be in general reliable as shown in [12]. Maybe more importantly, the demographics are more diverse than other frequently used sample population (e.g. college students), allowing for the least bias on the population [12].

Using this dataset we trained a model that learns both the image specific characteristics and the characteristics of the collection from which the image is chosen. The chapters are divided as follows:

- **Chapter 2** analyzes the existing work related to some of the core aspects of this thesis: Online surveys and Mechanical Turk, Shareability, use of Social Media, Wildbook.
- **Chapter 3** states the specific problem in the scope of the thesis.
- **Chapter 4** provides more insight into the data we are using and introduces the features that will later be used.
- **Chapter 5** introduces the models that were used and the metrics they were evaluated with.
- **Chapter 6** presents the setup used for our models, including data preprocessing and model parameters, when relevant.
- **Chapter 7** introduces some experiments that are designed to test some of the features on real people.
- **Chapter 8** analyzes the results more thoroughly in the various setups.
- **Chapter 9** provides some conclusions and some ideas for future extensions of this work

1.1 Terminology

Throughout the thesis we will use the following terms, which must be clearly defined to avoid any ambiguity.

- **SD Card:** the collection of all pictures that were taken by an individual – including those that were not shared on any social media or otherwise. Intuitively, it is the “raw” content of a user’s memory card without any type of filtering or other interference. Although the

term “SD” refers specifically to the memory device developed by the SD Association [13] we use it more broadly to refer to any memory support used by digital photo cameras.

- **Online Album:** as opposed to an SD card, an online album is the collection of the pictures that were shared on a Social media. As will be detailed in later chapters, an Online Album will often be a subset of an SD Card.

CHAPTER 2

RELATED WORK

This chapter analyzes the current state of the art for the most relevant aspects of this thesis: surveys and the use of Mechanical Turk, past works on shareability of images and the use of social media as data source. We also give a brief description of Wildbook, the tool that provides us with many fundamental functions.

2.1 Online Surveys

While our dataset is provided by the Great Grevy's Rally events [9] [10], the labeling of the images was done through surveys administered using Amazon Mechanical Turk. The use of surveys is well documented and established in social sciences, where they have been used for a long time to collect information from users. Surveys have evolved in their form over time, as technology has allowed more ways of contacting respondents.

Thanks to the broad availability of internet, online surveys are being adopted more and more frequently, also thanks to the greater range of possible questions [14]. Additionally, online surveys are in general cheaper and have access to a more representative pool of respondents. The choice of using an online survey instead of other survey methods is easily motivated, as it allows for cheap and quick creation of numerous questionnaires, each containing many images. Using a printed format would be more demanding, as it would require printing and mailing the surveys, recruiting the participants directly and manually collecting their responses.

Relying on an online service designed for this purpose is not only cheaper but also allows to easily process a much higher number of surveys. Amazon Mechanical Turk is the platform of choice for its reliability, its wide pool of users, low cost and ease of use among other advantages. For further information on the data collection step refer to [15].

2.2 Reliability of Mechanical Turk workers

The responses from users are at the very core of the dataset used in this thesis. Therefore the reliability of the interviewees is pivotal to ensure that the results can be trusted. On this topic [12] notes that Amazon Mechanical Turk is at least as reliable as other survey methods while providing with a more diverse pool of respondents. Users online are more diverse, as shown in [16], and even if not representative of the entire population they are still a less biased sample of the target population. This increase in diversity reduces the biases related to the participants being pooled from a limited population (e.g. college students' ages being mainly in a small range).

2.3 Shareability

Shareability is generally defined as “The property of being able to be shared” [17]. In this context, we use this term to refer specifically to the prediction on the “shared” label of an image. A preliminary analysis of this topic is available in [8], which proposes a framework to estimate the probability for an image to be shared. This work takes into account the features of the individual image, for example its “Beauty” as defined in [18] and the EXIF data it contains.

Little research is available in this specific direction and this thesis aims at expanding the understanding of this aspect. Specifically, the framework proposed by [8] focuses on single

images without taking into account the other images in the collection. We aim at improving this approach, by expanding in this direction and adding information about the collections from which these images are chosen.

2.4 Social media as data source

As mentioned, the end goal is using social media as the sole source of data. Social media has been extensively used in diverse fields for years with positive results [19]. For example, the use of this source of data can improve in tracking outbreaks of flu and in giving real-time dynamics of disease [20].

It has been noted that the search data alone may not prove effective in predicting flu outbreaks [21]. However, this approach has shown good results in helping authorities in monitoring public health [20] and has also been proven to have good predictive results [22]. For this reason, we feel confident in considering Social Media as a good source of data, with the added benefit of being more abundant than ad hoc data collected for the purpose of population estimation.

2.5 Wildbook

WildbookTM is an open source software framework that makes it possible for researchers to use citizen scientists to scale their research, mainly for mark-recapture, molecular ecology, and social ecology studies [2]. This framework combines Artificial Intelligence and computer vision to provide wildlife researchers with massive scale computer vision capabilities for the first time. It allows advanced processing of images thanks to the use of Deep convolutional neural networks (DCNN) to analyze the image and detect the animals in it. A further step, powered by computer vision, matches the individuals and assigns them name labels. This step allows

to recognize the same individual across multiple pictures, therefore allowing the application of Mark-recapture based methods (Lincoln-Petersen [23] and Jolly-Sebers [24] being the most commonly used), as done in [8] and [15].

All the functions available on Wildbook are made available through the Image Based Ecological Information System (IBEIS) [25]. We will refer to Wildbook when discussing the functionalities that are made available, while IBEIS will be used when we make a more specific reference to the APIs.

CHAPTER 3

PROBLEM STATEMENT

In this chapter we describe the problem in more detail and also give a formal definition as a learning problem.

This work aims at understanding how people share animal images online. Understanding this is only part of a more complex problem: estimating animal populations using social media as source of data. To do so, it is vital to understand the bias that is added by the way people select their images.

We expect that people, when deciding which images to share, do not take into account only the individual image they are looking at, but also the other images that are in the same collection. To better understand this we create models able to predict what images will be shared from a collection, relying on labeling gathered from surveys. We expand existing models by introducing features that try in various ways to account for the "context" (the collection from which the image is chosen) along with features representing the individual image's characteristics (beauty, animals in it, ...). Analyzing these models provides some insight into what is important when images are selected to be shared online.

We can formalize our problem as follows. We start with a set of images, further divided in many disjoint collections which represent an individual user's collection of photos – we will refer to these collections as SD Cards, as defined in Chapter 1. Each image is described by a set of features, described in Chapter 4, including the label ("shared" or "not shared"). We define

our problem as a binary classification, with labels “shared” and “not shared”, and we aim at maximizing the Accuracy and the F-1 Score. For our classification we consider the “shared” labels to be the positive class.

We enrich our analysis further by trying to verify the relevance of the features by surveying people through Amazon Mechanical Turk. Interviewing real people with carefully designed surveys should expand our understanding of the underlying behavior related to the features our models have identified.

In the next chapters we describe the dataset, the features and the experimental setup. We finish presenting our results and our conclusion.

CHAPTER 4

DATASET

In this chapter we introduce the datasets that we used, the features that we extracted and how they were processed. We then explain how the data was stored and all the information merged in the final dataset. We present an initial analysis of our data showing the distribution of the features.

The steps are summarized in Figure 2 on page 26.

4.1 Initial dataset

The datasets we are using are based on the data collected during the Great Grevy’s Rally (GGR) events in Kenya in 2016 [9] and 2018 [10]. These two events, which we will refer to as “GGR1” and “GGR2”, allowed to collect a high amount of data thanks to the contribution of many citizen scientists. The participants were asked to take pictures of animals and then share the content of their SD Cards with the organizers. The data from the SD Cards was minimally processed to remove images that were guaranteed to be out of the timeframe relevant for the event to protect the participants’ privacy.

The images were analyzed using Wildbook to compute many features, based on the detection of animals and identification of individuals in each photo. The resulting data was further processed to include features not provided directly by Wildbook. Before introducing the pre-processing steps we can introduce some of the details of these datasets, summarized in 4.1.

TABLE I

DATASET: SUMMARY OF THE COMPOSITION

	GGR1	GGR2
Number of total pictures	40810	53194
Number of annotations	33150	54812
Predominant species	Grevy's Zebra	Grevy's Zebra
Number of photographers	162	212
Negative class instances (not shared)	0.79	0.62
Positive class instances (shared)	0.21	0.38

The two datasets have a comparable number of total pictures, while GGR2 has a much higher number of annotations. This is due to the presence of many images depicting multiple animals.

4.2 Characteristics of GGR1 and GGR2

In this section we will highlight both the similarities and the differences between the two datasets.

They both are composed of pictures of animals, taken in Kenya by citizen scientists. The events were held in 2016 (GGR1) and 2018 (GGR2) and both lasted for two days. In both cases, citizen scientists were given instructions by the event organizers. The areas covered are

shown in the images, Figure 3 for GGR1 and Figure 4 for GGR2. The latter also shows the different species present in the dataset.

For GGR1, participants were asked to take photos of Grevy’s zebras, preferably capturing the right side of the animal. They were also asked to take photos of single animals, having the individual take a big portion of the frame.

For GGR2 the rules were much less restricting. Pictures containing more than one animal were allowed along with having mixed species in the same picture. Citizen scientists were also instructed to take photos of reticulated giraffes, as they too are endangered and share the same habitats. Pictures of other animals are also present, although in lesser quantity.

Therefore, there are some big differences across the two datasets. First and foremost, GGR2 has more variety in its images: different species, viewpoints and number of animals. Another important rule in GGR1 was to center the animals and have them take up a significant part of the frame. This rule was also lifted, yielding more variety in the composition of images and in the relative size of the animals in the picture.

All these differences can be expected to make the dataset intrinsically harder to learn, as many more variables can change and should be modeled. On the other hand, GGR2 is undoubtedly more realistic when compared to GGR1. For this reason it is important to analyze both, as a real dataset is likely to be complex, as GGR2.

We show some images to highlight the differences between the two datasets.

Some examples of GGR2 images can be found in Figure 5, Figure 6, Figure 7, Figure 8.

For GGR1, some examples are shown in Figure 9

4.3 Data handling

The two datasets were stored on a remotely hosted Wildbook instance. We had full access to the data through the APIs provided through IBEIS [25]. The endpoints that we used are summarized in Table II and are also discussed in [15].

TABLE II

IBEIS API: MAIN ENDPOINTS USED	
Feature	Endpoint
AID	/api/annot/
GID	/api/image/
Species	/api/annot/species/
Bounding box	/api/annot/bbox/
Orientation	/api/image/orientation/str/
Image URL	/api/image/src/
Image size	/api/image/size/
Viewpoint	/api/image/size/

This API allows to retrieve all the information outlined in the next section, as well as retrieving the image file itself. Due to the specific processing needs for this project, reduced-quality images were downloaded locally to a lab server and further processed (as detailed later in Section 4.6).

Since the images were needed only for the initial processing while all the metadata had to be available (and retrievable) constantly, we decided to store the data as follows. The images were stored in a dedicated folder on the machine’s file system, with their names being the unique identifiers for images on the IBEIS instance in use. All the metadata, both created by IBEIS and by our processing, was stored in a local database to allow for more advanced querying. This division of data makes the database light, allowing for faster querying of the frequently-accessed information we stored (metadata), while still making it possible without much overhead to access the image files in case they need to be processed further. The Server machine runs on Linux CentOS version 6.7 and uses Ext4 as file system. The database technology of choice is MongoDB for its well written Python drivers and ease of deployment.

4.4 Result of IBEIS processing

We used IBEIS to analyze the SD Card images. To gather the required information we used the API endpoints outlined in Table II.

This tool can extract a number of image features, as we summarized in Table II. Those used in the analysis for this work are as follows:

- The number of **animals** it contains

- The rectangular box around each animal (referred to as **bounding box**), identified by its upper left and lower right pixel x and y coordinates, along with the rotation angle.
- The **species** of each animal
- The **individual** animal identity, for selected species
- The **viewpoint** – the side of the animal we are looking at

The information we gather at this step will also be used to engineer other features, as will be documented later.

4.5 Labeling on MTurk(TM)

The labeling of the images (“share” or “not share”) has been performed on Amazon Mechanical Turk [11], an online survey service, and was jointly performed with [15].

Every respondent had to go through a sequence of images, picking the ones they thought they would share on social media. This process allows collecting sharing preferences that account for both the image and the ordering of the SD card.

Specifically, every survey reflected the content of an entire SD Card. This allowed to have the most realistic responses, using the most realistic data: we expect an average tourist to go through their photos after a trip, the same way one of the MTurk workers has to make their choices looking at all the pictures in the SD Card.

The surveys presented a list of images, allowing to select “share” or “not share” for each individually. The choice had no default, to prevent the respondents from potentially cheating. The design choice also aims at mimicking the actual process of going through pictures taken in the past and selecting the best ones.

4.6 Further processing and Feature Engineering

Further processing includes mainly engineering new features and adjusting the existing ones (normalization, scaling, etc.). In addition to the image-related features extracted directly from IBEIS, we added the following features. Some are aimed at capturing more of the image’s characteristics while others at modeling the SD Card structure.

4.6.1 Image Level Features

The features in this category are all those that only depend on an image and not on its “context” – the SD Card, in our case.

- **Beauty Features:** these features are based on [8], which in turn uses the work found in [18]. Namely, these features are: Luminance Channel ([26]), Weber Contrast ([27]), Hue Saturation Value (HSV, [28]) channels and “affective” features (Pleasure, Dominance and Arousal) based on [29].
- **Species count:** the number of animals for each species, computed by a combination of IBEIS API calls (see Table II for details).
- **Size of the animal in the picture:** we approximated the area effectively covered by an animal with that of the bounding box around it. This feature is the ratio of the animal’s bounding box and the image total area.
- **Rule of thirds:** the “rule of thirds” is a well-known photographic composition rule [30]. To apply the rule, an image must be divided both horizontally and vertically in three equal parts for a total of nine equal rectangles and four lines. The rule states that the

main elements of a picture should be on one of the lines or their intersections. More information and some examples can be found in [31] and [30]. The rule of thirds is known for making images more appealing and balanced, therefore it can be an important factor in deciding what image to share. For this reason, we have computed the scaled distance from each animal to one of the thirds and tracked both the closest animal and the average distance from thirds. This is a simplification of the broader problem of detecting whether an image conforms to the rule of thirds [32]. Since we are focusing on the behavior related to animal images, and both the original photographers and the interviewees were instructed to focus on animals, we believe that this is a reasonable simplification of the problem.

4.6.2 Collection level features

The features that somehow depend on the collection of images rather than the single image fall in this category. These features have been introduced to account for the characteristics of the whole SD Card.

- **Number of pictures in the SD:** this feature is simply the number of images that were in the SD card.
- **Number of individuals in the SD:** this feature is the total number of individuals present in the SD Card, computed thanks to Wildbook.
- **Animals in the picture vs total animals in the SD:** this feature tracks how many animals are present in the picture compared to the whole SD card, as it is possible that

an SD card has few pictures containing animals and we expect that pictures accounting for a higher portion of animals will be more likely to be shared. It has been implemented as a ratio.

- **Average number of animals per picture across the SD:** this feature is designed to track how "packed" is an SD card, following the intuition that images with one or few animals will be less likely to be chosen if, on average, the images in the SD card have many animals in them.
- **Position:** the position in the SD can influence the behavior of the human looking at the photos. We incorporated this aspect both using the ordinal number of the image in the SD and its normalized value, to track the portion of the album in which the photo is.
- **Beauty feature difference** with the previous and the following image: this set of features is the difference in the value of the beauty features defined before.
- **Similarity** with the previous and the following image: the similarity of subsequent images is captured by two measures of similarity, Structural Similarity Index [33] and Mean Squared Error. For Structural Similarity we relied on scikit-image implementation, while MSE was implemented from scratch using standard NumPy functions. The well-known formula for MSE can be easily adapted to two images, A and B , provided that they are modified to be monochrome and with same dimension $n \times m$. Since a monochrome image has a single channel, its pixels can be interpreted as the elements of a matrix, and the

images can be considered matrices. Therefore the Mean Square Error of two images can be computed as

$$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (a_{ij} - b_{ij})^2$$

Where a_{ij} and b_{ij} are the elements of the two matrices A and B , representing the images.

4.6.3 Further Processing

The features were scaled or normalized, depending on the model the data was to be fed to. More details will be provided when discussing the models. In general, the features were normalized or scaled. Normalization (mean centering and variance scaled to one) was performed using the “StandardScaler” class provided by scikit-learn. Scaling of the features in a unit range was done using the “MinMaxScaler” provided by the same library. The categorical features were all encoded using one-hot encoding, either directly when applicable or by using the “MultiLabelBinarizer” class (from scikit-learn), depending on the format of data. Principal Component Analysis [34] and Feature Selection were also used to reduce the dimensions of the feature space, as detailed in later chapters.

4.7 Preliminary analysis of data

It is now interesting to have a look at our data, by checking the distributions of some of the features, both qualitatively (plots) and quantitatively (statistical test). As statistical tests, we opted to use Welch’s T-Test [35]. This statistical test is a modification of Student’s T-Test for samples with different variance and unequal sample size. Both tests aim at testing whether two populations have the same mean. The Null Hypotheses, H_0 , is that the two populations

have the same means. Being able to reject the null hypothesis with a good significance allows affirming that the two populations have a different mean. We generally compare the portion of the images that were shared against those that were not. Plots for all the features will be provided in the appendix. Here we show some of the features.

Figure 10 shows the distribution of the normalized position. The normalized position of an image is the normalized value of the ordinal position of an image in the SD Card. A lower value indicates an image placed at the beginning of the SD Card, a higher value means that the image was more towards the end.

From the plot of this feature we can notice two important details. The difference in the distribution of the values between the images that were shared and that were not is noticeable, although they mainly overlap. There are however more shared images among the pictures that were in the beginning compared to those towards the end of the SD Card.

We explain this by thinking of the process of going through an SD Card for the first time. At the beginning all images will look new and interesting, therefore will be more likely shared. After the user has seen many pictures he will likely find them less exciting and reduce the rate at which he shares them.

Even if this feature shows some differences between the shared and the not shared parts of the dataset, this feature is not strongly separated. By plotting other features, we notice that all of them do not show a strong separation between the two subsets of the data. This is most evident in Figure 12 or in Figure 11. None of the features shows a strong separation, therefore we expect this problem to be hard to learn. However, according to the statistical significance

tests we performed, the features have statistical differences between the two subsets of data. We expect the models to be able to learn this problem, at least to some extent, by combining the differences among all the features. Some of the statistical tests results are reported in Table III.

TABLE III

WELCH'S T-TEST RESULTS FOR SOME FEATURES

Feature	P-Value	Reject H_0	Conclusion
Normalized Position	1.93^{-17}	Yes	Different distribution
Number of images in the SD	1.74^{-154}	Yes	Different distribution
Structural Similarity difference with following image	2.62^{-31}	Yes	Different distribution

4.8 Feature correlation

We computed Pearson's Correlation Coefficient between our features and the label ("shared"). Overall, many features are not correlated with our output variable. We report the correlation coefficients in Table IV, for the possible datasets, when the coefficients suggest some correlation.

The values of the Pearson Correlation Coefficient further suggest that the problem can be learned. The correlation values are high enough, especially given the type of data we are using, to provide information to the learning models. It should be also noted that the values are in

TABLE IV

SELECTION OF PEARSON CORRELATION COEFFICIENTS			
Feature	GGR1	GGR2	GGR1+GGR2
Number of animals in SD	-0.176	-0.08	-0.12
Number of images in SD	-0.18	-0.08	-0.10
Position	-0.15	-0.08	-0.09
Beauty (pleasure)	0.08	0.08	0.12
Beauty (dominance)	0.08	0.08	0.12
Rule of thirds (avg distance)	0.01	0.12	0.18
Percent animal in image vs in SD	0.08	0.04	0.05

general lower for GGR2, when compared with GGR1. This indicates that it is going to be harder to learn this dataset, as we also noted when analyzing the general characteristics of GGR2.

4.9 Final dataset used

The final dataset contains all the features outlined in this chapter. Every image is represented by both its beauty and the characteristics of the SD card used for the survey used to label it. The scale of the features varies, sometimes by many orders of magnitude, between

different features. This can be a problem for some models, as will be described in later chapters.

However, since all features are numeric, scaling or normalizing them is trivial.

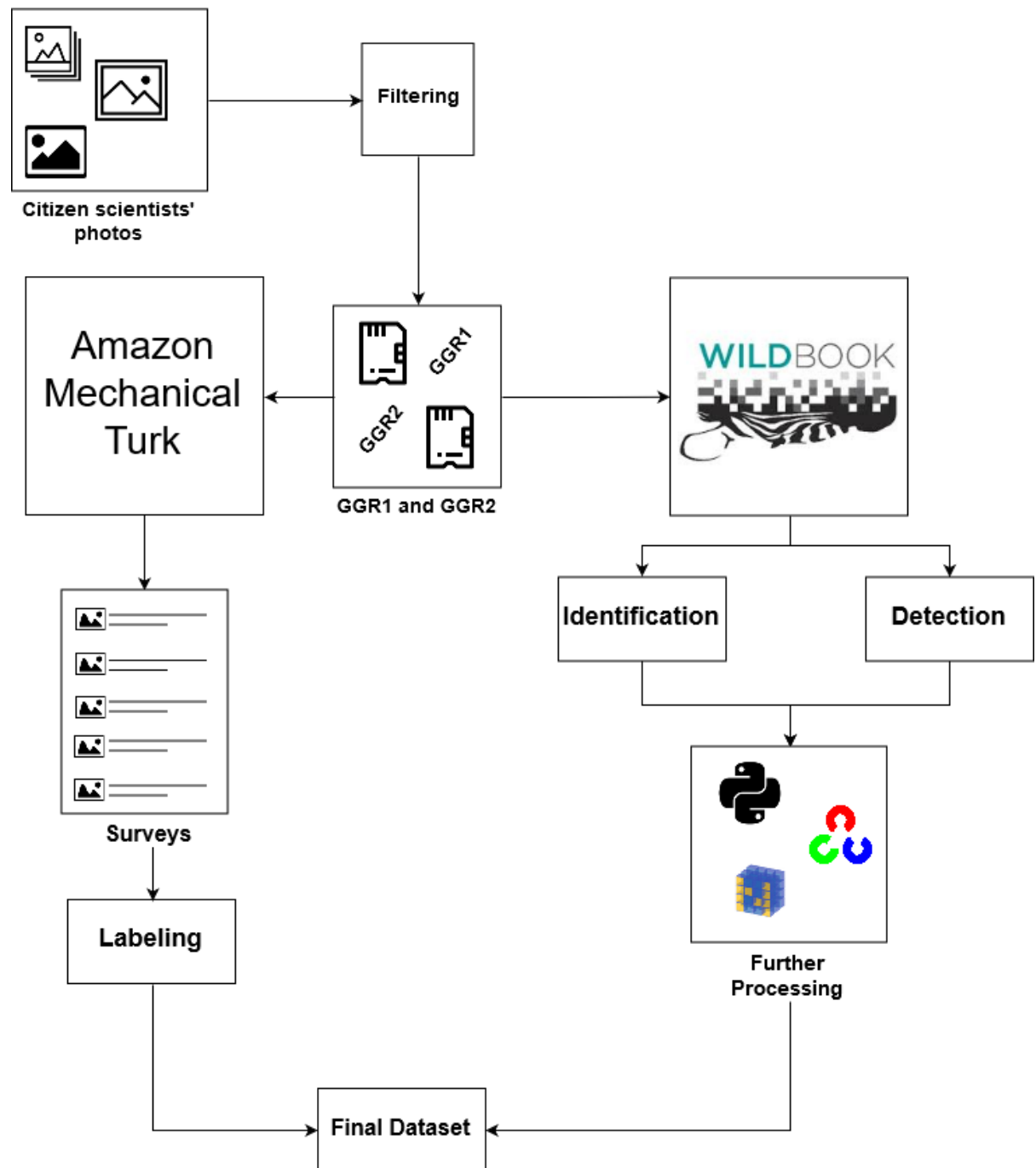


Figure 2. Steps in data collection and processing

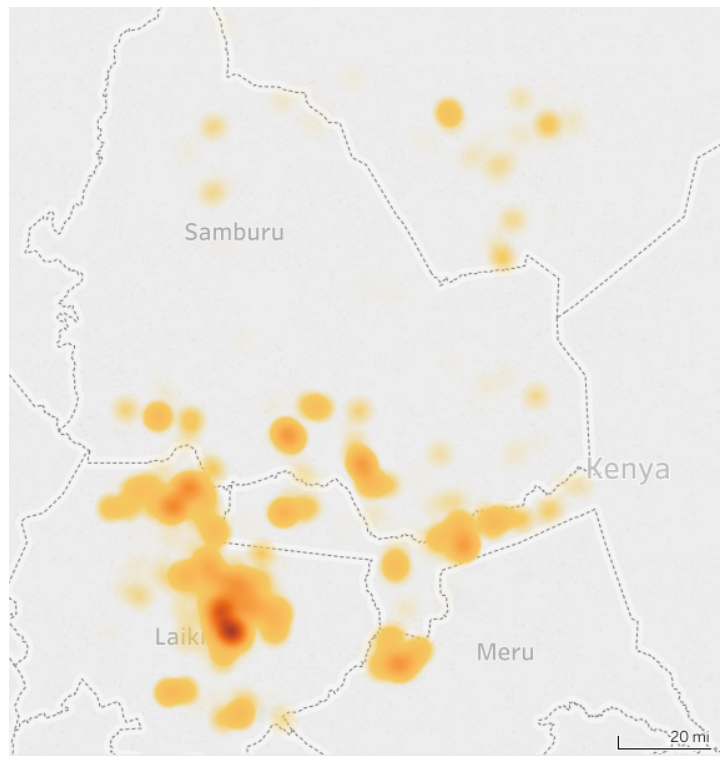


Figure 3. Map of sightings for GGR1

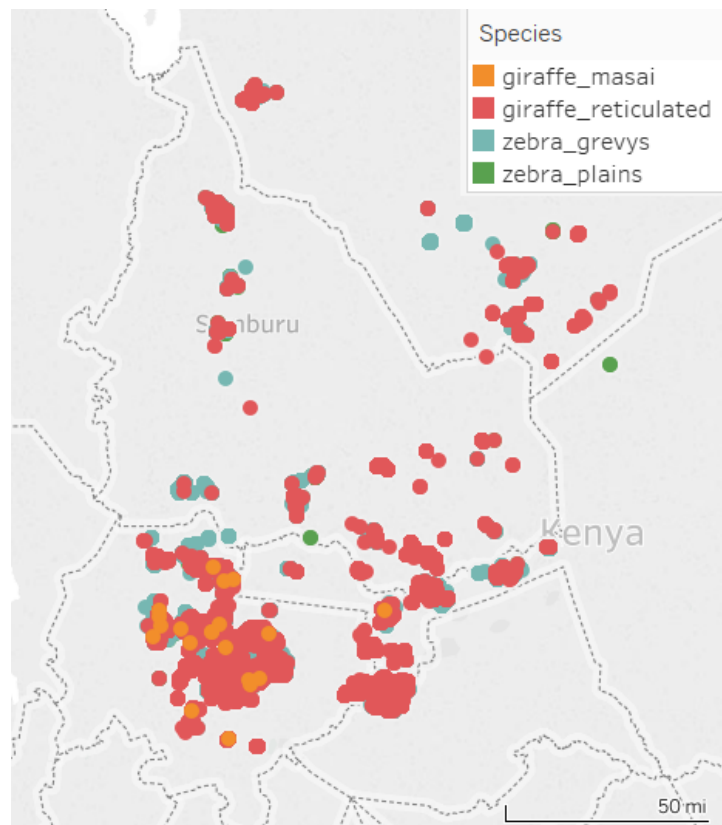


Figure 4. Sightings and species for GGR2

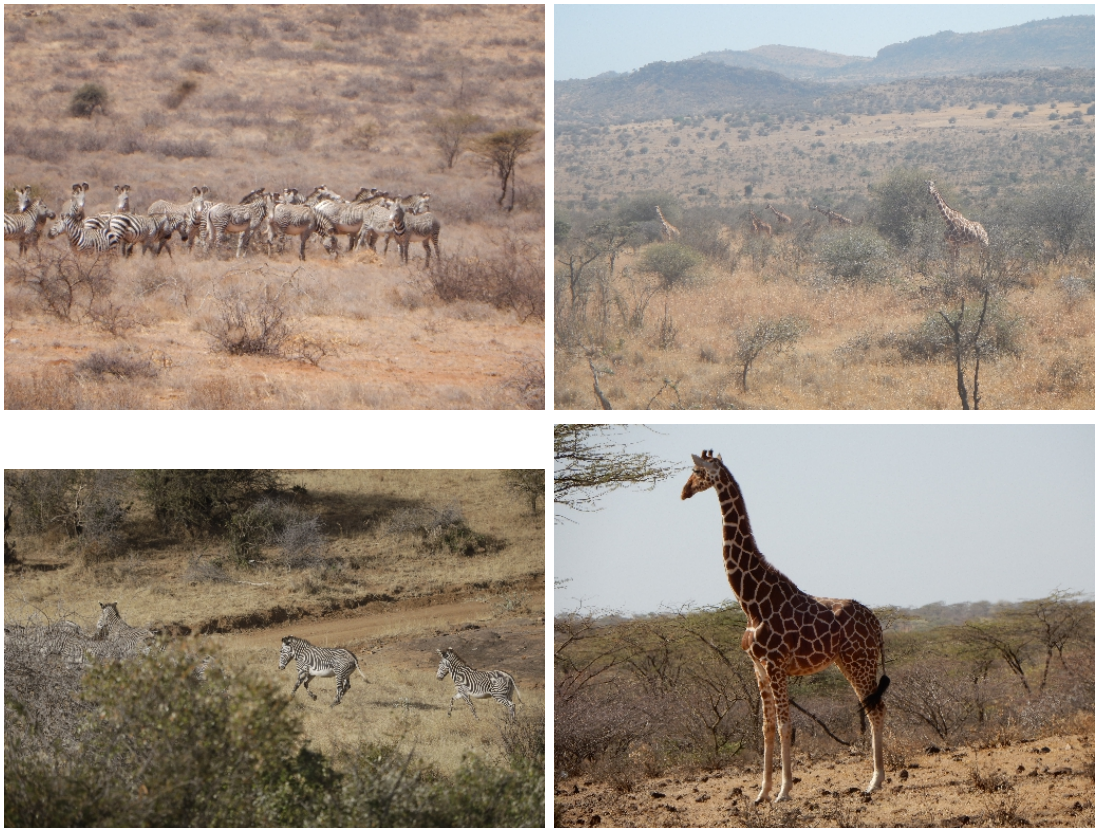


Figure 5. Examples of images from GGR2



Figure 6. GGR2: a group of elephants



Figure 7. GGR2: a big group of zebras



Figure 8. GGR2: a giraffe and some zebras

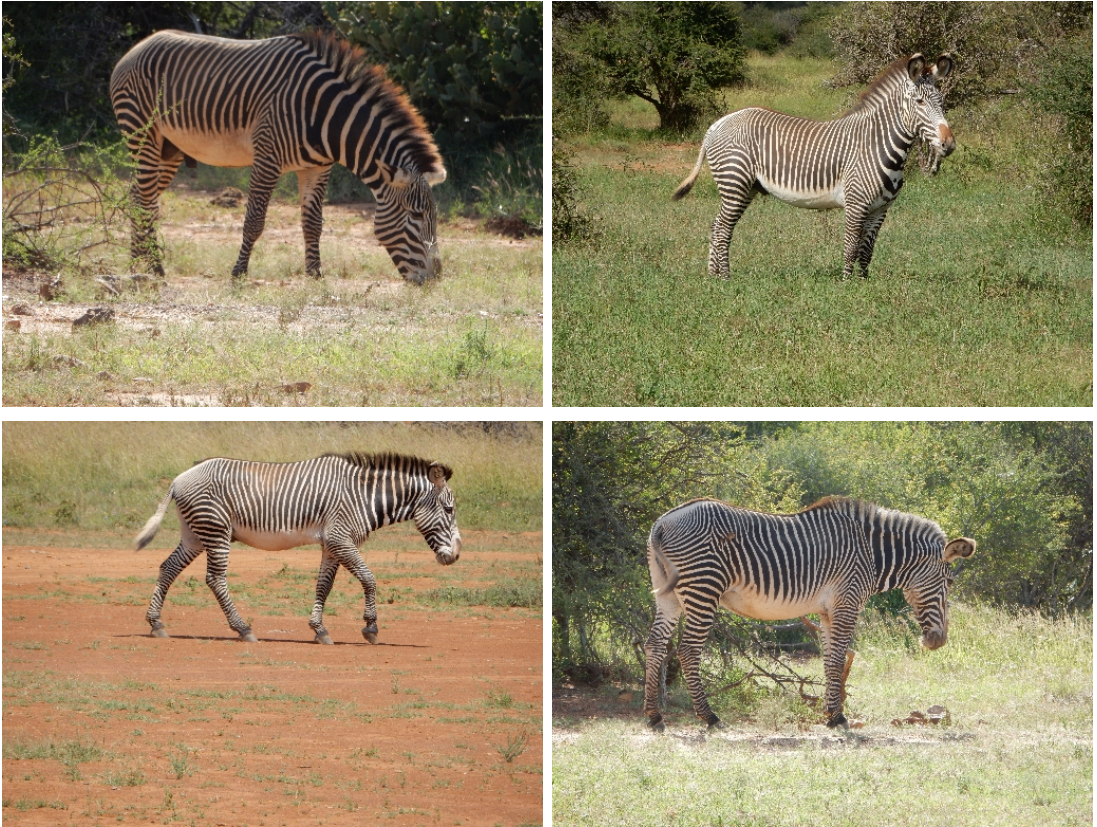


Figure 9. Examples of images from GGR1

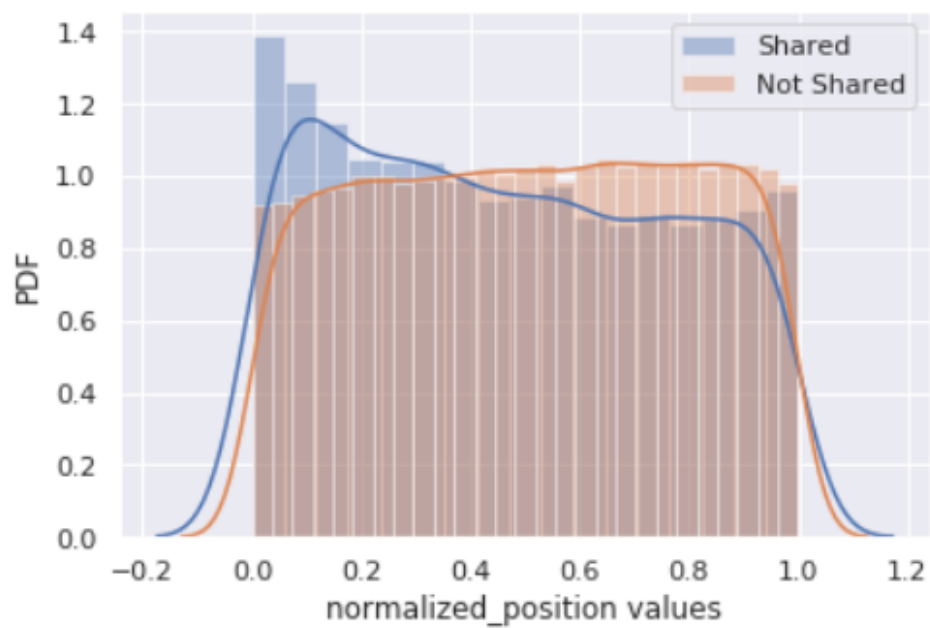


Figure 10. Distribution of the “normalized position” feature

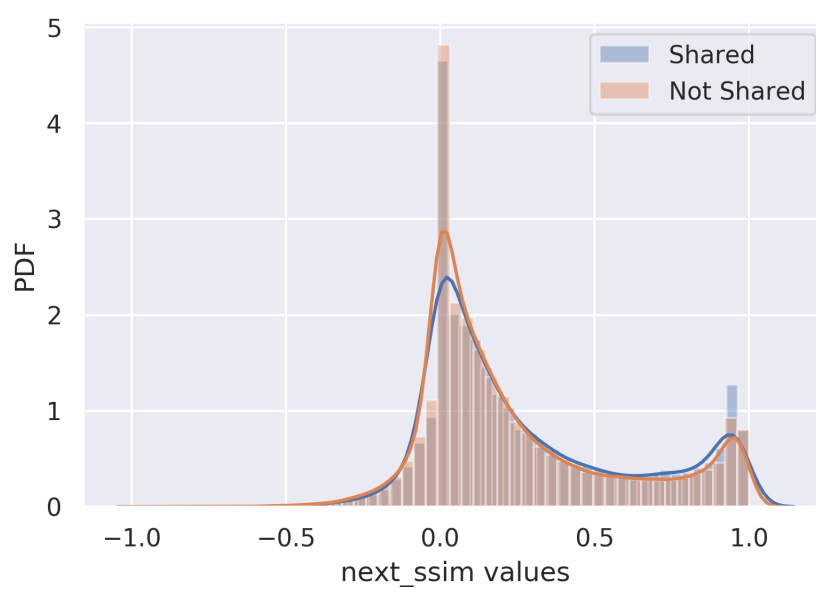


Figure 11. Distribution of the “structural similarity with next” feature

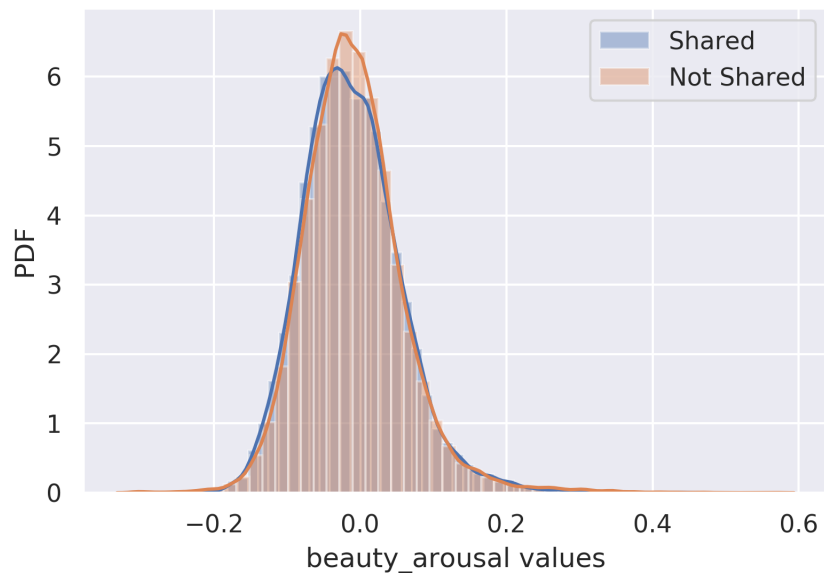


Figure 12. Distribution of the “arousal” beauty feature

CHAPTER 5

MODELS

Before presenting the whole setup it is worthwhile to introduce the main models we used and the main evaluation metrics used to compare them.

5.1 Classification Models

In this section we introduce the models that were tested for this project. The models are only described briefly, for further details we refer to more complete texts that can be easily found.

5.1.1 Baseline classifiers

Although we are mainly interested in improving the models provided by [8], comparing against baseline classifiers will provide a better understanding of our performance.

We choose two dummy classifiers as our baseline. The first is a model that always predicts the majority class, therefore always outputs the same label.

We will also use a random predictor that will make predictions based on the class balance. Since this model is random, it cannot be used for real applications regardless of its performance. Since both models are based only on the classes, any preprocessing of the features is useless.

5.1.2 Logistic Regression

Logistic regression is a method designed to work with both dichotomous and polychotomous data [36]. In the case of a binary target variable, it aims at finding a linear separator in the

feature space, trying to divide the hyperspace in two regions, each containing only examples with one of the two class labels. This approach can be adapted to work for the case of multi-labeled data too. It is a good starting point thanks to its simplicity and the quickness to train and test the models.

5.1.3 K-Nearest Neighbors

K-Nearest Neighbors is an example of instance-based learning. The algorithm, given an example, makes a decision based on the K nearest neighbors of the example in the dataset. The decision is usually the majority vote between the neighbors, while their weight can be either uniform or scaled on the distance.

Measuring the distance is at the core of this algorithm and it is important to note that usual distance metrics (e.g. Euclidean) are sensitive to feature scaling. It is, therefore, good practice to normalize the values in the dataset.

5.1.4 Decision Tree

Decision trees are unsupervised models that can learn to predict the label of the data by creating decision rules [37]. The tree is constructed starting from the "best" feature, which is used to make a decision at the root node. The decision will be either to split to a lower node or to output a decision. At the root node all training data is present, while at subsequent nodes only the relevant portion is. Defining what feature is the best one is not trivial and many solutions are available. The core idea is trying to use the feature that allows to better separate the classes. Among the possible ways of computing the value of a feature, the most commonly used are Information Gain (ID3), Information Gain Ratio (C4.5) and Gini index (CART). Care

must also be taking to avoid overfitting, since a tree can be grown to learn every detail of the training data, likely overfitting it. Some of the solutions to this problem are limiting the depth of the tree or allowing a node to be split only if it contains at least a given number of examples. Another common technique is pruning – removing some of the least promising branches, which are expected to be learning anomalies or noise. Pruning comes in two flavors: pre-pruning stops the creation of a subtree if it would worsen a goodness measure, while post-pruning removes the branches after a full-grown tree is created.

5.1.5 Random Tree Forest

Random Tree Forests are a well-known ensemble method. Ensemble methods, in general, construct a strong predictor by combining the output of many weaker predictors.

The final classifier is built by combining many decision trees trained using Bagging (Bootstrap Aggregating), therefore is trained on a subsample of the data, with repetition.

5.1.6 Boosting models

Boosting is a learning method based on ensembles (like Random Tree Forests). However, the way the individual learners are trained is different. The idea is to improve the base model by boosting its performance using many weak learners. Many boosting models have been proposed, among the most well-known we can briefly introduce Adaptive Boosting (AdaBoost) [38] and Extreme Gradient Boosting (XGB) [39], an efficient implementation of Gradient Boosting [40]. They both aim at improving on the current model’s errors, by trying to improve the predictions in the part of the dataset where the current model is weaker.

AdaBoost: at every iteration, the weights given to instances of the dataset are updated to give more importance to the instances that were wrongly predicted. This makes the weak learner that is trained at the specific iteration to focus more on the data that causes more errors.

XGB: at every iteration, a weak learner is trained on pseudo-residuals (the errors in the existing model), in order to reduce the errors. The error is computed with a loss function, which is minimized (using gradient descent) in order to select the weak model the is performing best.

5.2 Performance Evaluation metrics

In this section we introduce the main evaluation metrics used to compare the models. We used stratified cross-validation to have reliable results when evaluating models.

5.2.1 Evaluating Classification

Classification can be evaluated using many metrics. We focus on four common metrics that provide meaningful insight into the model's behavior. It may be useful to first show the general definition of a confusion matrix, which is a representation of a model's performance. A confusion matrix is a 2×2 matrix where the columns represent the instances' real labels and the rows the predicted labels.

		True label	
		Positive	Negative
Predicted Label	Positive	TP	FP
	Negative	FN	TN

Accuracy (ACC) is probably the most intuitive metric and is defined as the ratio between the correct predictions (positive or negative) over the total population, therefore it informs us on how many correct labels our model chose.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision or Positive Predicted Value (PPV) is defined as the fraction of instances correctly predicted as positive over all the instances predicted positive.

$$Precision = \frac{TP}{TP + FP}$$

Recall or True Positive Rate (TPR) is defined as the ratio between the instances correctly predicted as positive over all the positive instances,

$$Recall = \frac{TP}{TP + FN}$$

F1-Score is a metric designed to combine Precision and Recall and is the harmonic mean of these two metrics.

$$F1 - Score = 2 \cdot \frac{P \cdot R}{P + R}$$

Where P is Precision and R is Recall.

CHAPTER 6

EXPERIMENTAL SETUP

In this chapter we present the whole pipeline that we used to produce our results. The first section summarizes the data collection process. The next sections delve in more details on how each model was trained and the specific setups used. We show the results obtained on the GGR dataset. The last section introduces the issues encountered with the GGR2 dataset and some insight on the differences on the dataset. The code was implemented in Python3 within a Jupyter Notebook and relies on many open source libraries: Scikit-Learn [41], NumPy [42], pymongo (MongoDB [43] driver) , Seaborn (visualization) [44], OpenCV (image analysis) [45].

6.1 Data Collection

We used images gathered through two separate events, the Great Grevy's Rally of 2016 [9] and of 2018 [10]. Both consist of many images, divided into SD cards, reflecting the pictures taken during the trip by citizen scientists. Detection and identification of animals were deferred to Wildbook [1], which allowed us to retrieve all the necessary information through an API. These features were added in the dataset and enriched with more features, as we detailed in Chapter 4.

Labeling this dataset, one of the critical steps in the collection process, was done using Amazon Mechanical Turk [11], as [15] details. Every survey consisted of all the images contained in one of the SD cards and the interviewees were forced to decide whether they would share or

not each of the images. The images were displayed in the given order, to gather information related to the order in the most effective way. The labels created through the surveys were merged in the database to be used in our models.

6.2 Model Creation: pipelines

In this section we introduce the specific steps taken to create and train models – our pipeline, in the broad meaning of the term. It should be noted that we used the “pipeline” functionality of Scikit-Learn [41] extensively. The context should make it easy to understand what pipeline we are referring to.

We decided to use Scikit’s pipelines in order to have a clear definition of the steps taken to create a model also in the code. An added benefit is that the pipelines behave like “regular” models, therefore they can be trained, tested and evaluated using the library provided methods.

Some models require the features to be either scaled or normalized to work properly. It is important that this step is applied to the training set only, as using the whole dataset will scale the features using more information than what should be available. For this reason, we created simple blocks implementing these preprocessing steps to later use them in pipelines. This allows also to easily change a model, adding preprocessing steps (PCA, Feature selection) with minimum modification to the code.

6.3 Main components

For PCA we used the standard implementation provided by Scikit-Learn, which provides a standard implementation that relies on Singular Value Decomposition. Among the possible parameters, we focused only on selecting the unexplained variance. This parameter allows to

specify how much information we accept to lose when removing dimensions and is easier to control than the actual number of components.

For feature selection we chose a similar class, `VarianceThreshold`, as it allows to specify a simple criterion for feature selection: all features that have a variance lower than the given threshold should be removed. This allows removing only the features that will probably contribute the least to the models. This step is indicated in the table together with the threshold used. We also selected a fixed number of features using the `SelectKBest` class, which selects the best features according to the ANOVA F-score between the feature and the label. We report the results obtained with this method in the tables, indicating the value of K , the number of features to select.

To perform Feature Scaling and Feature Normalization we relied on two similar classes provided by Scikit-Learn: `MinMaxScaler` and `StandardScaler`, respectively. Both components can be seamlessly added to any of our pipelines as needed.

We opted to perform repeated, stratified cross-validation, provided by the class `RepeatedStratifiedKFold`. Using the folds provided by this class we computed the metrics we are focusing on (Accuracy, Precision, Recall and F-1 Score) using the functions provided by the `metrics` module of Scikit-Learn.

For the models we relied on the standard implementations for every model. All models with the exception of XGB are available by default in Scikit-Learn. For XGB we relied on the implementation made available through Python Package Index (PyPI).

A graphical summary of these three basic components – PCA, Feature selection and Model – can be found in Figure 13.

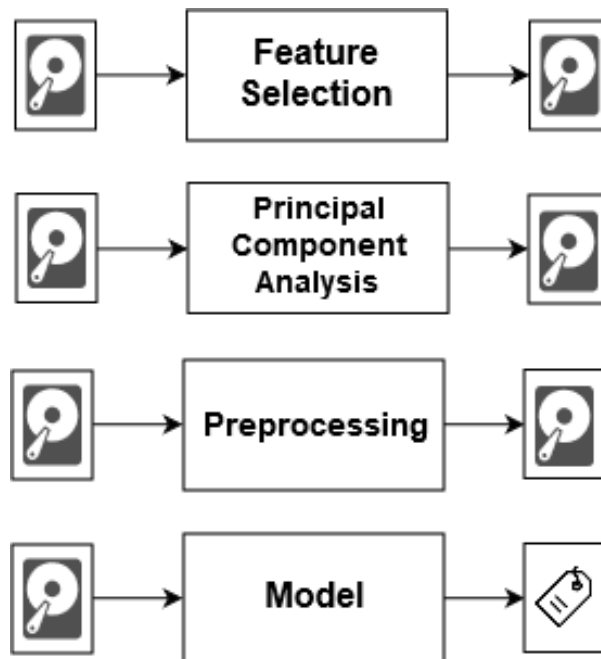


Figure 13. Main components of the data preprocessing and model training pipelines

6.4 Training and evaluating models

Having defined the basic components it is quite simple to decide exactly how to configure each. All setups described here were evaluated using cross-validation, as mentioned above, and we tried to find models that performed well across the board. For almost all proposed models we

tested the results on the full dataset, also applying feature selection (to remove lower variance features) and PCA. Results are presented in the different setups.

The setup used for each model is summarized in Table V, where we highlight the settings that were changed from the default. Tweaking some of the parameters did not yield statistically significant improvements, therefore those changes were reverted and not used for the final version.

Logistic Regression: for logistic regression, the inputs were first standardized since solutions are affected by the scaling of variables, as suggested in [46]. We then tested whether feature selection or PCA could improve the results, adding them to the Logistic Regression pipeline. The possible combinations are summarized in Figure 14

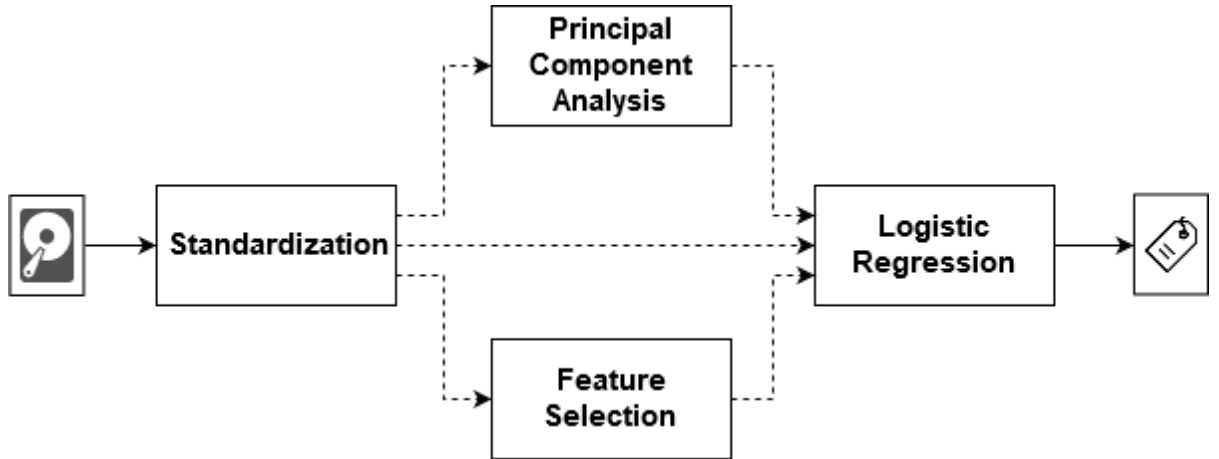


Figure 14. Possible pipelines for logistic regression

K-Nearest Neighbors: since this model relies on measuring distances it is important to scale (normalize) features. A feature that has a higher magnitude will be given more importance, therefore we always scaled the values. In addition to this, we reduce the dimensionality using either PCA or feature selection, as shown in Figure 15.

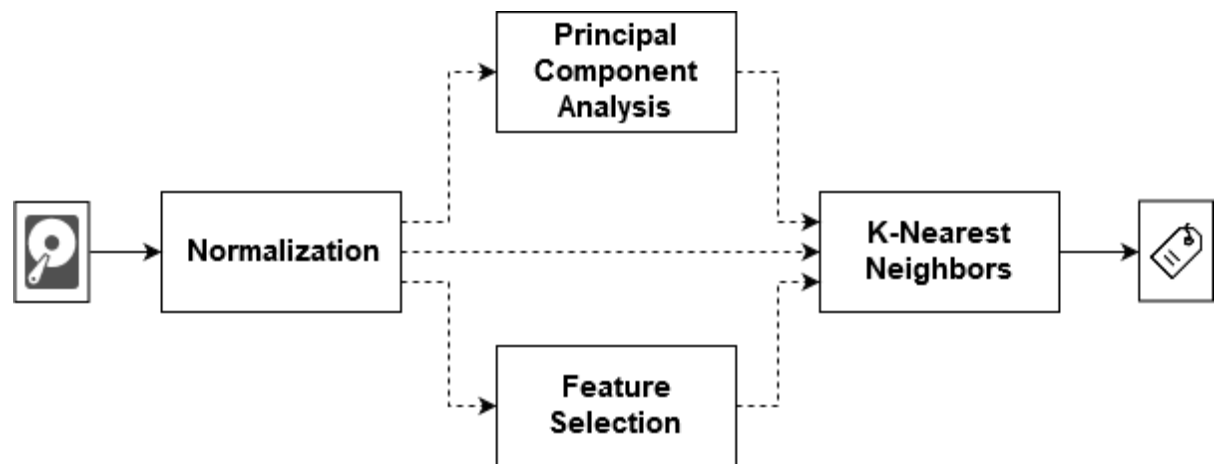


Figure 15. Possible pipelines for k-nearest neighbors

Decision Tree: decision trees can overfit if there are too many features. For this reason, we tested different thresholds for the `VarianceThreshold` feature selection class. However, since the training data does not have a gigantic number of features, this did not heavily impact the performance. We also tried to reduce the number of variables using Principal Component Analysis, see Figure 16 for reference.

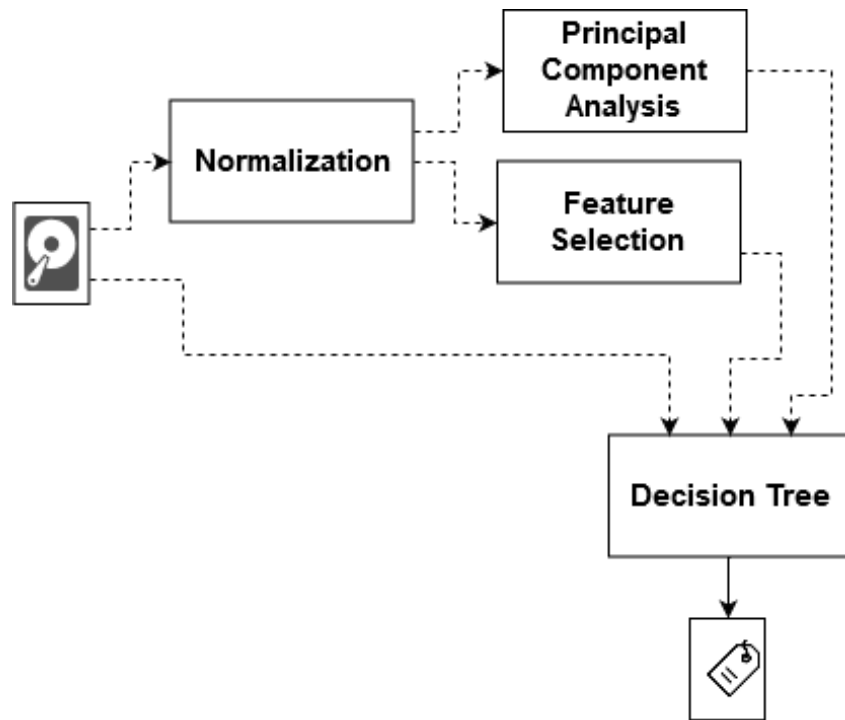


Figure 16. Possible pipelines for decision tree

Random Tree Forest, AdaBoost, XGBoost: these ensemble models should handle well our dataset as-is. We tested preprocessing steps to see if they affected the performance or the run time. The possible combinations are shown in Figure 17.

6.5 Comparing GGR1 and GGR2

To compare the performance across the two datasets we started by evaluating separately the two datasets, using stratified cross-validation.

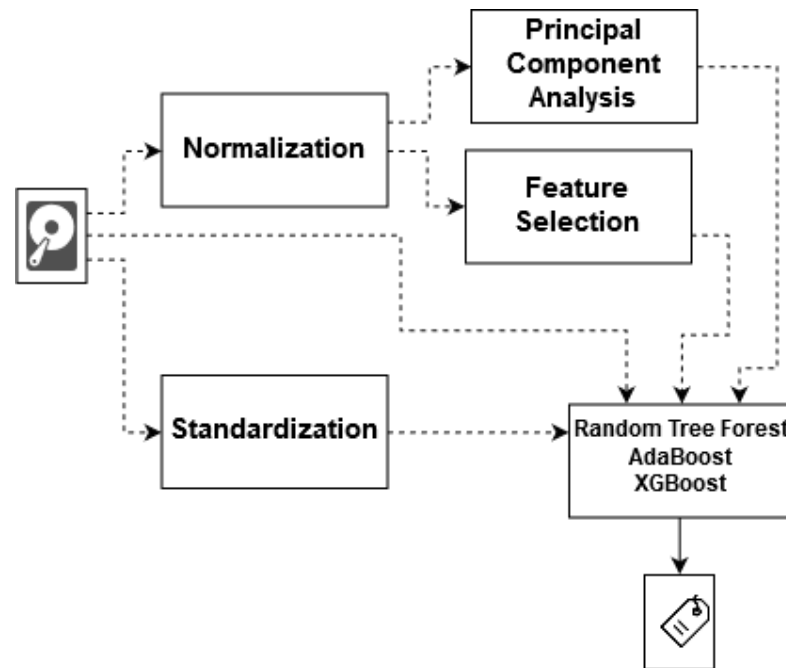


Figure 17. Possible pipelines for our ensemble models

In addition to that, we trained our models on the whole GGR1 dataset and used the trained model to predict the data from GGR2. For this second approach we evaluated the metrics on the predictions obtained.

We also combined the two datasets in one bigger dataset and performed cross-validation, as we did for the other setups. We also tried using this combined dataset as training data to predict either GGR1 or GGR2.

These different approaches aim at giving more insight into the behavior of the models. Specifically, training on GGR1 and predicting GGR2 will help us understand if the importance

of the features is constant across the different datasets. The second approach – merging the datasets – aims at understanding if the differences across the two datasets affect the performance.

TABLE V

MODEL PARAMETER SETTING	
Model	# Parameters (if not default)
Baseline (majority class)	<code>strategy=most_frequent</code>
Baseline (class balance)	Default settings (class balance)
Logistic Regression	<code>n_jobs=4</code> for efficiency, other parameters default
K-Nearest Neighbors	<code>metric="euclidean"</code>
Decision Tree	Default settings
Random Tree Forest	<code>n_estimators=100</code>
AdaBoost	<code>n_estimators=100</code>
XGBoost	Default settings

6.6 Surveys

The structure of the surveys is constant across all the experiments, as the fundamental question does not change: the respondents always have to choose which images they would share online.

Every image was shown along with two radio buttons – “share” and “not share” – and the images were displayed one per row. The radio button forces the workers to perform an action when making a selection, in order to prevent completely random behavior.

In addition to this, we asked the workers a few questions to understand their relationship with social networks. This information was not used in this work.

Every survey reflected an entire SD Card taken from either GGR1 or GGR2, for a total of 356 surveys. Every survey reflected the order in which the pictures were taken and was answered by one MTurk worker.

Figure 18 shows a portion of a survey with more than one image, while Figure 19 shows how a single image is presented to the respondents.

The next section details how some of the specific experiments for this work were designed.





	<p>Would you share this image on social media?</p> <p><input type="radio"/> Share</p> <p><input type="radio"/> Not Share</p>
	<p>Would you share this image on social media?</p> <p><input type="radio"/> Share</p> <p><input type="radio"/> Not Share</p>
	<p>Would you share this image on social media?</p> <p><input type="radio"/> Share</p> <p><input type="radio"/> Not Share</p>

Figure 18. Structure of a survey

A photograph of a zebra standing in a savanna landscape. The zebra is facing right, with its head slightly lowered. The ground is reddish-brown soil, and there are dry, leafless trees and some green shrubs in the background.

Would you share this image on social media?

☐ Share

☐ Not Share

Figure 19. Example of a single image in a survey

CHAPTER 7

EXPERIMENTS

One of the objectives of this work is understanding how people behave. The models we created can provide us with useful insight on this matter. Checking if the conclusions drawn from our models are reflected by the behavior of users is important. Testing all combination of features would require an enormous effort, therefore we focus on the features that were identified as most important by our models. Furthermore, we focus on collection-level features, as the image-level features have already been shown to be significant by [8].

7.1 Overall Design

We will select a limited number of pictures in order to ensure that they are as homogeneous as possible with regard to any feature but the one we are testing. Ideally, we would select a few SD Cards that share similar features across all images, with the exception of the feature we are testing.

No SD Card fits these stringent requirements, therefore we selected images that share similar properties and designed the surveys accordingly.

For our experiments we require that all features – except the one being tested – be similar, in order to observe the effect of a specific feature. We will then survey users on Amazon Mechanical Turk, similarly to what was done to label the data, in order to see how the feature we are targeting affects the responses.

7.2 Features to be tested

We are testing the features that our model considers most important. We have to further narrow the scope of the experiments to reduce the complexity of the surveys. Out of the features that were highly ranked by our model we have selected the number of animals in the whole SD and the position of the image.

These features were chosen because, first and foremost, they appear to be important across our evaluation. In addition, they were all introduced by this work and understanding the way they affect users' behavior will improve our understanding of their online behavior.

7.3 Survey structure

We created surveys based on the feature that we were going to test, choosing images that were as similar as possible among each other. The number of images for each survey was chosen based on [8] and was limited by the number of available images. In general, we could not create surveys with a number of images that reflected SD Cards' numbers as too few images fit our selection criteria.

Examples of images used for the surveys can be found in Figure 20. Given how we selected our images, we expect to see mainly images of good quality. Further inspection of the images confirms that the images used for the surveys reflect the characteristics of those in the example.

7.3.1 Position in the SD Card

We selected a small pool of images, all having the same number of animals and similar "beauty", computed as we introduced in previous chapters. These images were then used to create surveys. All these surveys presented the same images, ordered differently, in order to



Figure 20. Examples of images used in our experiments

isolate the effect of ordering from other features of the image. These images are summarized as Group **3** and **4** in Table VI. The surveys were generated using the same images (that shared similar characteristics), randomizing the order.

7.3.2 Number of animals in the SD Card

This feature is probably the hardest to effectively test. The number of animals in an SD Card can be modified by adding more images, adding images containing more animals, or both. We expect that both the number of images and the animals per picture could have an effect on the sharing behavior.

We can reasonably expect from previous testing that images with more animals will be shared more. Although this needs some care when analyzing the results it appears to be the better option as it can be more easily checked and accounted for. Changing the number of animals only by changing the length of the survey would affect more variables making it nearly impossible to control for all of them: the length of the SD card, the position of the image, and its relationship with other images in the survey may all affect our result.

For this reason we, structured our surveys using two groups of images, Groups **1** and **2** in the table. We created a total of 20 surveys, half for each group of images. The first group was composed of images with two or three animals, the second group was composed of images with mainly one animal, two in few cases. The overall structure of the two groups is summarized in Table VI.

TABLE VI

PICTURE GROUPS STRUCTURE FOR THE EXPERIMENTS

Group	# Animals	# Pictures
1	50	20
2	25	20
3	60	30
3	30	30

CHAPTER 8

RESULTS

In this chapter we summarize the results obtained using the models introduced in Chapters 5 and 6. We also introduce the results from the experiments of Chapter 7.

8.1 Performance on GGR1

Overall the models performed very well on the GGR1 dataset, achieving high accuracy and F1-Scores. The best performing model is Extreme Gradient Boosting, closely followed by the other ensemble methods we presented.

The results are summarized in the tables for the various algorithms: Logistic Regression in Table VIII, K-Nearest Neighbors in Table IX, Decision Tree in Table X, Ensemble Methods in Table XI. The performance of the dummy classifiers can be found in table Table VII. All the approaches presented here can outperform the dummy predictions. Figure 21

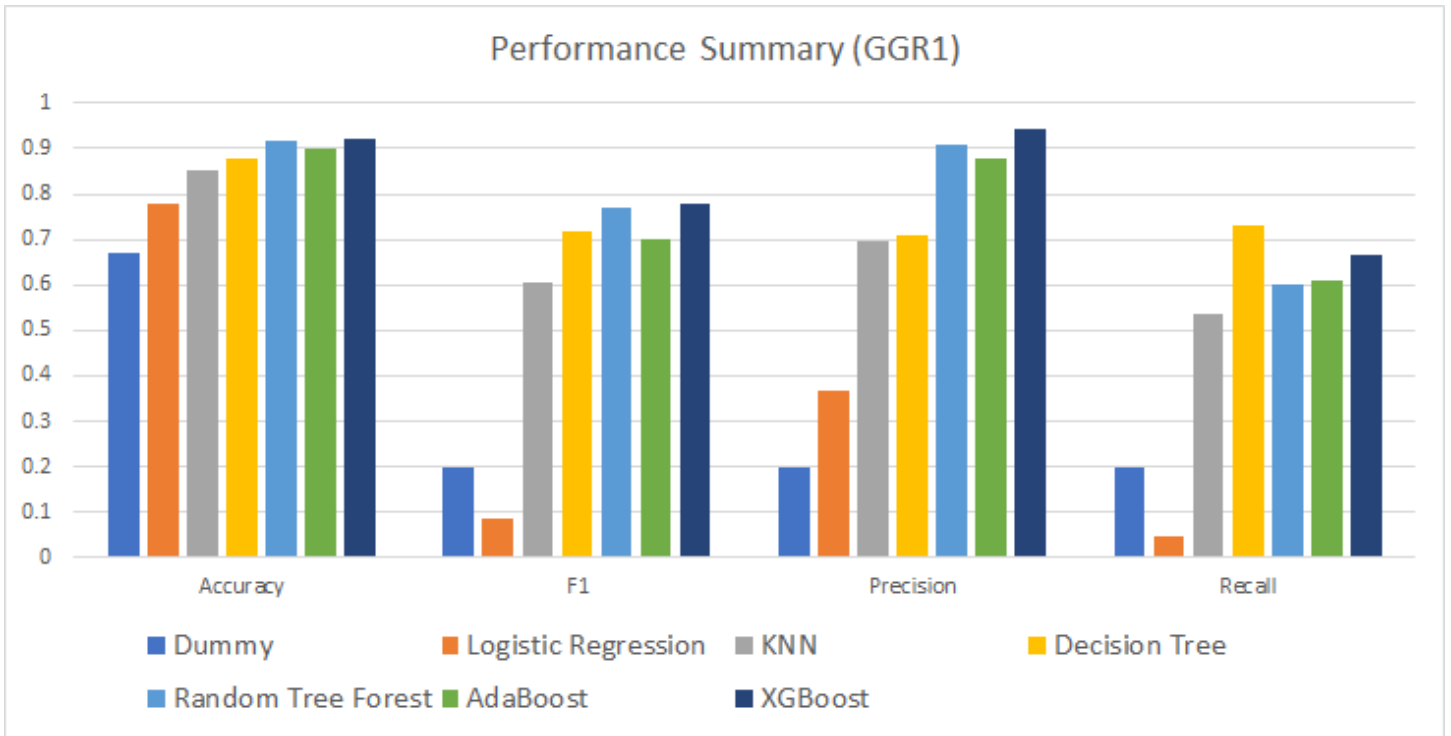


Figure 21. Performance of our models on GGR1

We recreated the models that were introduced in [8] to make a comparison, as those models only had access to the “image level features” while we enriched our analysis with “Collection level features”. As per the author suggestion, the models were recreated using the implementation provided by Scikit-Learn. The models were evaluated with the same metrics and cross-validation we used for all others, the results are shown in Table XII. The table also presents our best model performance and the results obtained with XGBoost using only the features available in [8], although this model was not part of the initial analysis, for a fair comparison.

TABLE VII

RESULTS FOR BASELINE CLASSIFIER (GGR1)

Model	Accuracy	F1-Score	Precision	Recall
Majority class	0.792 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Random, based on class distribution	0.67 ± 0.007	0.209 ± 0.014	0.208 ± 0.013	0.209 ± 0.014

TABLE VIII

RESULTS FOR LOGISTIC REGRESSION (GGR1)

Configuration	Accuracy	F1-Score	Precision	Recall
Standardization	0.784 ± 0.003	0.085 ± 0.012	0.365 ± 0.041	0.048 ± 0.007
Standardization + Ftr Selection (thr = 0.01)	0.787 ± 0.002	0.023 ± 0.007	0.276 ± 0.065	0.012 ± 0.004
Standardization + PCA	0.792 ± 0.0	$0.0 \pm$	$0.0 \pm$	$0.0 \pm$

Our models have an overall better performance, as we summarize in Figure 22, which shows the performance of the best models.

TABLE IX

RESULTS FOR K-NEAREST NEIGHBORS (GGR1)

Configuration	Accuracy	F1-Score	Precision	Recall
Feature Scaling	0.84 ± 0.005	0.556 ± 0.013	0.661 ± 0.016	0.48 ± 0.015
Ftr Selection (thr = 0.01) + Scaling	0.854 ± 0.004	0.605 ± 0.012	0.697 ± 0.014	0.534 ± 0.015
PCA + Standardization	0.778 ± 0.004	0.296 ± 0.015	0.436 ± 0.02	0.224 ± 0.013

TABLE X

RESULTS FOR DECISION TREE (GGR1)

Configuration	Accuracy	F1-Score	Precision	Recall
Ftr Selection (thr = 0.01) + Standardization	0.883 ± 0.005	0.723 ± 0.011	0.713 ± 0.013	0.734 ± 0.016
Model only	0.881 ± 0.004	0.72 ± 0.009	0.706 ± 0.012	0.734 ± 0.014
Scaling	0.881 ± 0.004	0.719 ± 0.009	0.706 ± 0.012	0.733 ± 0.012
PCA	0.703 ± 0.007	0.309 ± 0.016	0.3 ± 0.015	0.319 ± 0.019

TABLE XI

RESULTS FOR ENSEMBLE MODELS (GGR1)

Model	Configuration	Accuracy	F1-Score	Precision	Recall
RTree Forest	Model only	0.905 ± 0.004	0.725 ± 0.014	0.92 ± 0.011	0.599 ± 0.017
	Ftr Selection (thr = 0.01)	0.919 ± 0.004	0.777 ± 0.011	0.908 ± 0.012	0.679 ± 0.015
	PCA	0.786 ± 0.003	0.182 ± 0.014	0.451 ± 0.029	0.114 ± 0.01
AdaBoost	Model only	0.899 ± 0.004	0.714 ± 0.014	0.878 ± 0.019	0.602 ± 0.017
	Ftr Selection (thr = 0.01)	0.901 ± 0.005	0.719 ± 0.016	0.879 ± 0.021	0.609 ± 0.022
	PCA	0.788 ± 0.002	0.045 ± 0.011	0.378 ± 0.063	0.024 ± 0.006
XGBoost	Model only	0.92 ± 0.003	0.772 ± 0.011	0.956 ± 0.007	0.648 ± 0.015
	Ftr Selection (thr = 0.01)	0.921 ± 0.003	0.777 ± 0.012	0.951 ± 0.007	0.657 ± 0.017
	Ftr Selection (K=5)	0.922 ± 0.004	0.78 ± 0.015	0.941 ± 0.011	0.666 ± 0.024
	PCA	0.791 ± 0.001	0.006 ± 0.004	0.373 ± 0.21	0.003 ± 0.002

8.1.1 Use of collection level features

The improvement in performance obtained by the use of collection level features shows that these features are actually useful for the model. All the performance metrics have higher values.

In particular, the F1-Score is greatly improved, most prominently in the recall term.

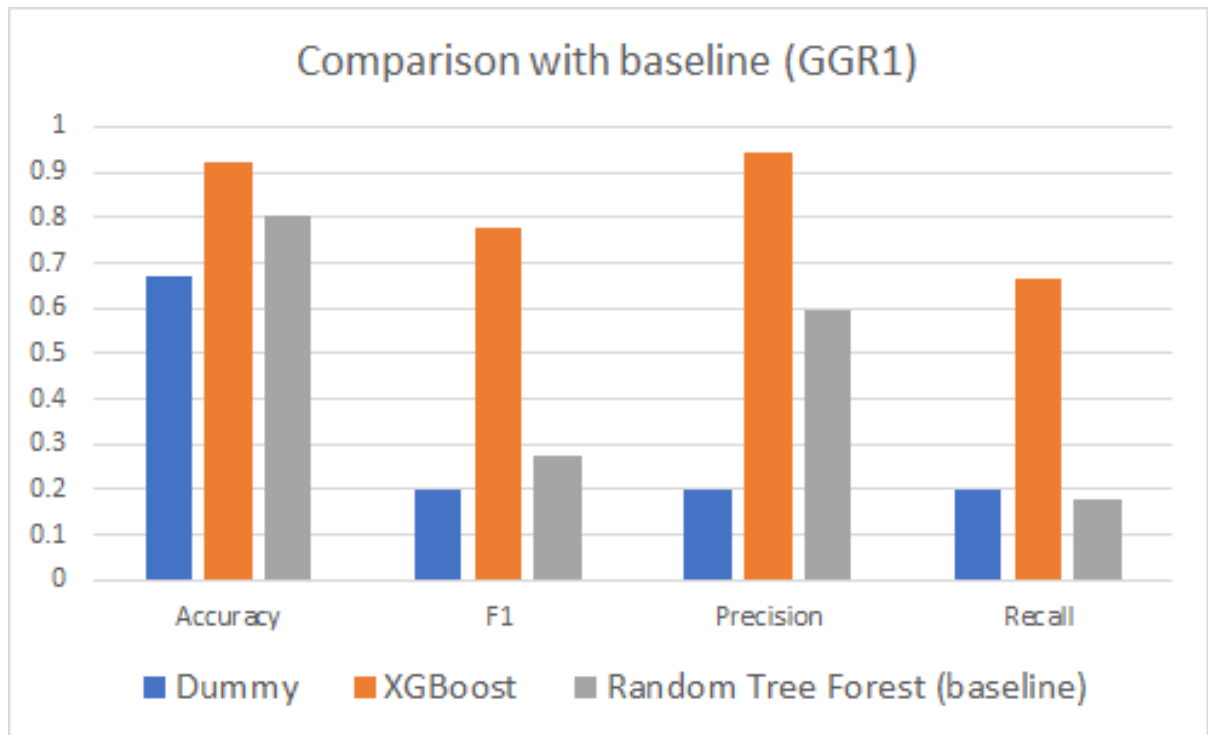


Figure 22. Comparison of the best models in GGR1

8.1.2 Most important features

The best performing model is XGBoost and adding a feature selection step – even selecting a small number of features – does not impact the performance. This is expected from XGBoost as the training process performs a selection of the most important features.

It is worth looking at the most important features selected both by `SelectKBest` and by looking at the plot of importance provided by XGBoost, trained without any preprocessing step. `SelectKBest` uses the ANOVA F-score to find the best features, the 10 best features

TABLE XII

GGR1 RESULTS, IMAGE-LEVEL FEATURES UNLESS NOTED

Model	Accuracy	F1-Score	Precision	Recall
Naive Bayes	0.792 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Logistic Regression	0.792 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Decision Tree	0.722 ± 0.006	0.351 ± 0.013	0.342 ± 0.012	0.362 ± 0.015
Random Tree Forest	0.803 ± 0.003	0.275 ± 0.015	0.593 ± 0.025	0.179 ± 0.011
AdaBoost	0.791 ± 0.001	0.015 ± 0.008	0.47 ± 0.142	0.008 ± 0.004
XGBoost	0.797 ± 0.001	0.074 ± 0.011	$0.766 \pm 0.0.65$	0.039 ± 0.006
XGBoost w/ <u>NEW features</u>	0.921 ± 0.003	0.777 ± 0.012	0.951 ± 0.007	0.657 ± 0.17

and their importance is plotted in Figure 23 For the importances provided by the XGB library there are three choices on how to calculate them: number of times a feature appears in a tree (`'weight'`, Figure 25), average gain of the splits with a feature (`'gain'`, Figure 26), average number of splits affected by the feature (`'cover'`, Figure 24).

It should be noted that there are some differences between the results, depending on how the importance is computed. However, some features are always ranked in the top 10 features regardless of the method used. Features related to the beauty of the images (name starts with "beauty_") – perhaps unsurprisingly – are always important in the prediction. In addition to

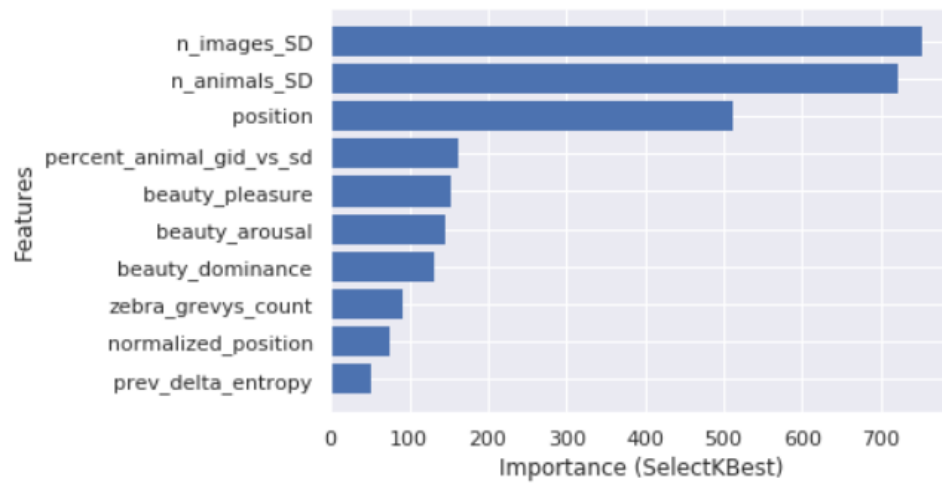


Figure 23. Feature importance (SelectKBest)

that, we can see that features related to the SD card structure are consistently in the top ranks. Among the most effective features are the number of animals and images in the SD, the average of animals per image in the SD (`sd_avg_animals_image`) and the position.

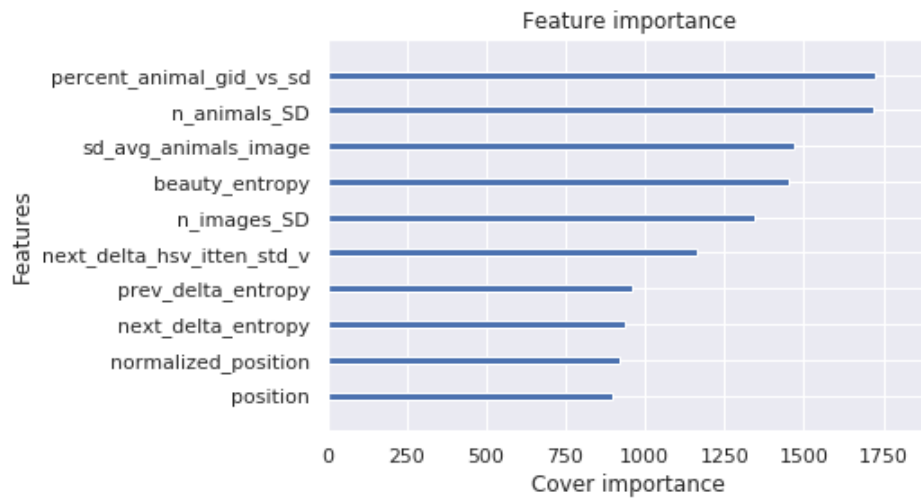


Figure 24. XGBoost feature importance (cover)

8.2 Performance on GGR2

We report the results of the model evaluation on GGR2 in Table XIII. The results are worse across all metrics, models and approaches when compared to the performance obtained on GGR1. We report the results of the best models using all the features introduced in this work (Table XIII), the models presented in [8] (Table XIV) and a dummy classifier (Table XV). The results are also summarized in Figure 27, along with some of the approaches described in the next section.

The degrade in performance with this dataset is further analyzed in the conclusions, where we also provide insight into the likely causes.

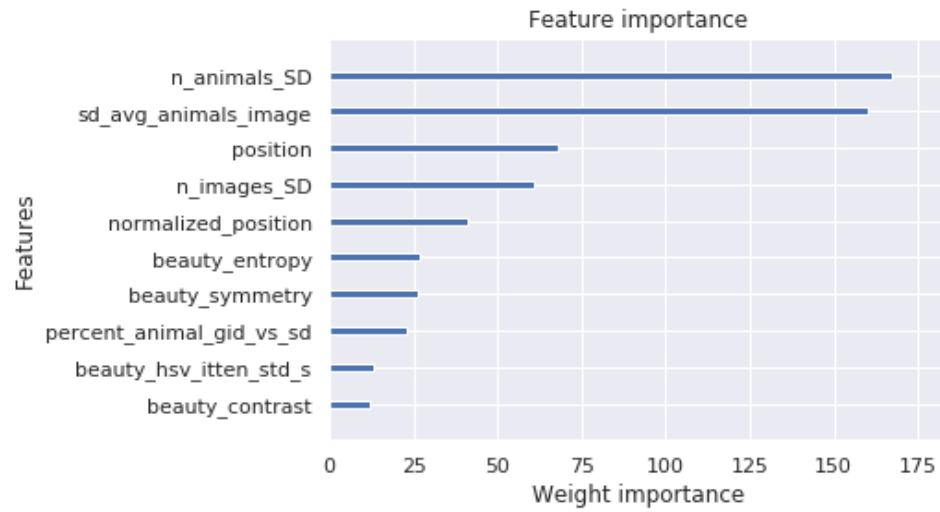


Figure 25. XGBoost feature importance (weight)

The next section provides further results we obtained by applying the methods we introduced to further compare the two datasets.

8.3 GGR1 and GGR2 comparison

As we showed, the performance varies noticeably between GGR1 and GGR2. For this reason, we tried different approaches as specified in Chapter 6 to have a better understanding of this result.

Using GGR1 as training data to predict the labels for GGR2 was a slight improvement in accuracy, at the cost of a lower F1-Score. We used K-Nearest Neighbors as model, achieving an accuracy of 0.59 with an F-1 of 0.202, which is an improvement over the results obtained

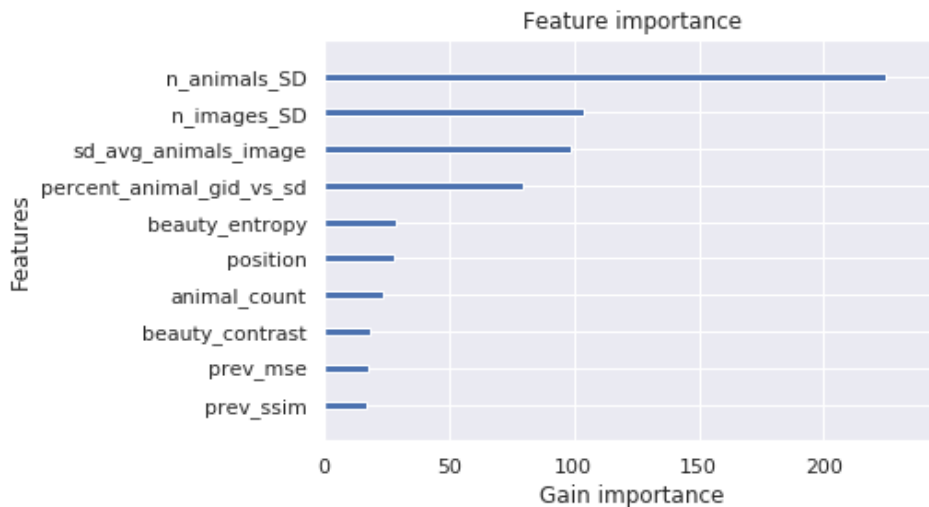


Figure 26. XGBoost feature importance (gain)

by using our approach on GGR2 only. The performance, in this case, is comparable to the performance obtained by the models introduced in [8], evaluated on GGR2 only.

Merging the two datasets into just one and evaluating the models yields much better results as we present in Table XVI.

8.4 Results of our experiments

Our experiments tested, as we outlined in the previous chapter, some of the features that our models indicated as important. One of the conclusions we can draw, already presented in [8], is that respondents overall agree on how they label the images. Our testing was limited but the participants show an overall good agreement in their decisions, consistently with what was already available in the literature.

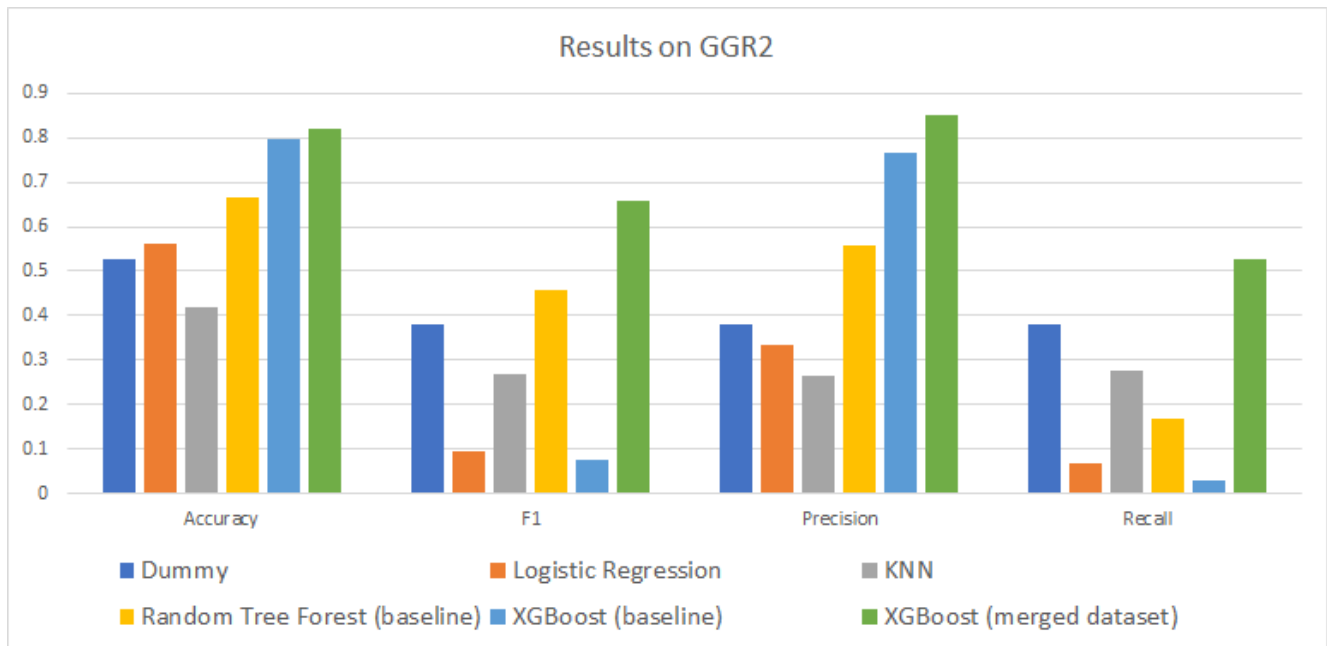


Figure 27. Results for GGR2 (various approaches)

8.4.1 Position

To test the effect of the position, as we mentioned, we randomized the order of images in our surveys. The expectation of this experiment was to highlight the effect of the image position on the sharing rate. We evaluate our results looking mainly at the least often shared images. For those images that were shared less than 4 times, 75% of the shared images were in the first 10 positions of the ordering. Conversely, being in the first ten images did not have a positive effect on only 11% of the images.

TABLE XIII

RESULTS FOR THE GGR2 DATASET (CROSS-VALIDATION)

Model	Accuracy	F1-Score	Precision	Recall
Logistic Regression	0.562 ± 0.057	0.094 ± 0.055	0.334 ± 0.193	0.067 ± 0.055
Decision Tree	0.26 ± 0.148	0.229 ± 0.078	0.204 ± 0.099	0.274 ± 0.054
K-Nearest Neighbors	0.419 ± 0.069	0.269 ± 0.032	0.266 ± 0.057	0.276 ± 0.017
Random Tree Forest	0.357 ± 0.125	0.171 ± 0.055	0.192 ± 0.117	0.17 ± 0.028
AdaBoost	0.394 ± 0.084	0.201 ± 0.075	0.211 ± 0.096	0.199 ± 0.072
XGBoost	0.291 ± 0.162	0.172 ± 0.112	0.18 ± 0.152	0.176 ± 0.091

8.4.2 Number of Animals in the SD Card

This experiment highlights the effect of the number of animals present across the survey, as we introduced in the previous chapter. Our experiment shows a wide gap in the number of images shared for the two types of surveys. Contrary to our expectation, the number of images shared in the surveys that contained more animals was much lower than for the other group: on average only 5 ± 1 images were shared, out of 20, for this group. By comparison, the surveys of the group with fewer images had on average 14.2 ± 5 images shared, out of 20. This shows a noticeable effect on the sharing behavior in response to the number of animals that are present in the images.

TABLE XIV

RESULTS FOR GGR2, ONLY IMAGE-LEVEL FEATURES

Model	Accuracy	F1-Score	Precision	Recall
Naive Bayes	0.618 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Logistic Regression	0.618 ± 0.001	0.061 ± 0.016	0.479 ± 0.124	0.033 ± 0.009
Decision Tree	0.608 ± 0.005	0.492 ± 0.006	0.488 ± 0.007	0.497 ± 0.008
Random Tree Forest	0.667 ± 0.004	0.459 ± 0.008	0.606 ± 0.009	0.369 ± 0.008
AdaBoost	0.631 ± 0.003	0.258 ± 0.012	0.558 ± 0.013	0.168 ± 0.01
XGBoost	0.797 ± 0.001	0.074 ± 0.011	$0.766 \pm 0.0.65$	0.039 ± 0.006

TABLE XV

RESULTS FOR DUMMY CLASSIFIERS (GGR2)

Model	Accuracy	F1-Score	Precision	Recall
Majority class	0.62 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Random, based on class distribution	0.526 ± 0.005	0.381 ± 0.008	0.381 ± 0.009	0.381 ± 0.009

TABLE XVI

RESULTS FOR THE MERGED DATASET (GGR1+GGR2)

Model	Accuracy	F1-Score	Precision	Recall
Majority Class	0.678 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Random, class distribution	0.564 ± 0.004	0.321 ± 0.006	0.322 ± 0.006	0.321 ± 0.007
K-Nearest Neighbors	0.636 ± 0.003	0.33 ± 0.007	0.403 ± 0.007	0.279 ± 0.003
XGBoost	0.821 ± 0.004	0.658 ± 0.009	0.851 ± 0.011	0.526 ± 0.012
Random Tree Forest <u>Image-level features only</u>	0.706 ± 0.003	0.388 ± 0.008	0.589 ± 0.01	0.289 ± 0.007

We will now present our conclusions given the results we obtained and suggest future works that we expect can improve the understanding of this problem.

CHAPTER 9

CONCLUSIONS

We have created a new model to predict whether an image will be shared on social media, introducing features that model the collections from which images are chosen. We tested our models on two datasets, both consisting of a collection of images divided in "SD Cards" – collections that reflect a single user's photos. We analyzed the main differences between these two datasets. In summary, GGR1 has similar images across the whole dataset: just Grevy's zebras, one animal per picture, carefully photographed animals that take up a big portion of the frame. GGR2, however, is more diverse: giraffes in addition to zebras, pictures with many animals and species, less careful framing of animals.

We first analyze the results obtained on GGR1. The models perform better than previous solutions across all our metrics. Through our analysis, we also showed that the most important features are both related to the whole collection and to the individual images. Among these features, the most important were the number of animals in the SD Card, the position of the image in the SD Card, the beauty (encoded with multiple, image-specific features).

We conclude that the improvement of the performance of our models is an effect of the use of SD Card related features that we introduced.

We draw the conclusion that for a "simpler" dataset like GGR1 – predominantly single species, with animals consistently photographed from the same side and taking a significant part of the frame – our model can be expected to be really effective.

The performance on the GGR2 dataset, however, is less consistent. Model evaluation on GGR2 data alone has low-performance metrics. To gain a better understanding we made further tests on this dataset.

In particular, using GGR1 data as training and predicting the labels on GGR2 data we noticed an increase in accuracy when compared to the model evaluation results on GGR2 alone. This shows that the features learned from GGR1 are still meaningful for GGR2 data, confirming that our approach is overall meaningful.

Using the two datasets combined into one yields good performance, comparable to the one obtained in the evaluation of the models on GGR1 data. This result shows that our solution is effective and can work even on complex datasets. The drastic improvement in performance may depend on the introduction in the dataset of some "easy" examples that allowed the models to better learn the problem. We conclude that our approach is valid and effective and a more complex dataset can be effectively learned given that other, simpler data is available.

9.1 Future work

As we have found from our analysis, a complex dataset like GGR2, containing multiple species and high diversity in the pictures is hard to learn. A less complex dataset (like GGR1) can be learned with better performance, while also helping the learning process: we showed that using both datasets yields a big improvement in performance.

Since GGR2 is a more realistic dataset, we think that creating a model that can learn this type of harder dataset is important. Therefore we propose a few extensions that should help in improving our framework. Using more realistic training data should help, in particular

in designing other features that can better capture the complexity of the data. Capturing, for example, the diversity of species across a single SD card will probably yield noticeable improvements: one of the main differences between the two datasets is species diversity. We expect that to capture these relationships, features accounting for the relationship among the images (not limiting to those adjacent) can be informative. Among these, a better estimation of the similarity of images is needed.

Using more realistic datasets will also provide with a larger amount of samples. This, in turn, can be exploited by the introduction of more advanced models that account for complex relationships.

APPENDICES

Appendix A

DISTRIBUTION OF FEATURES

Here we show the distributions of all features.

Appendix A (Continued)

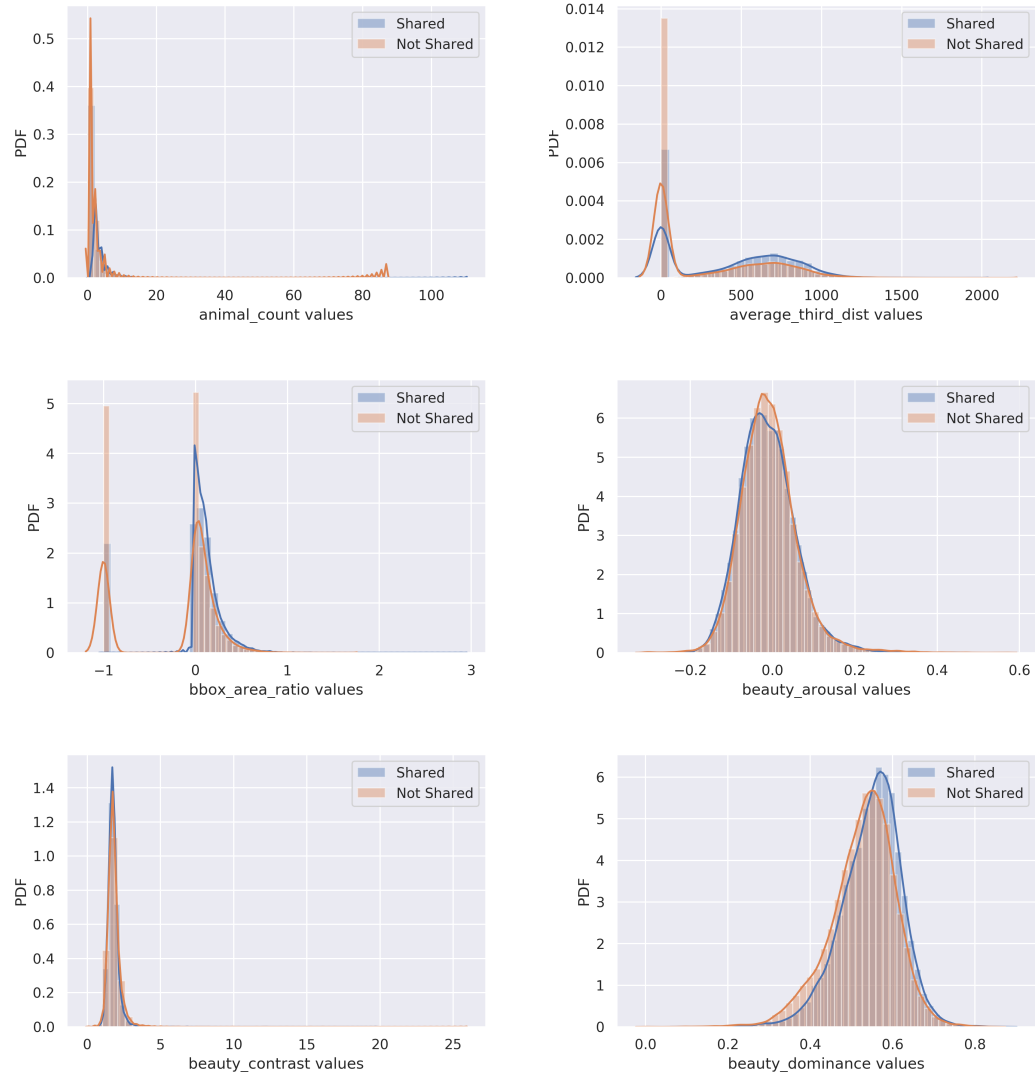


Figure 28. Feature distributions

Appendix A (Continued)

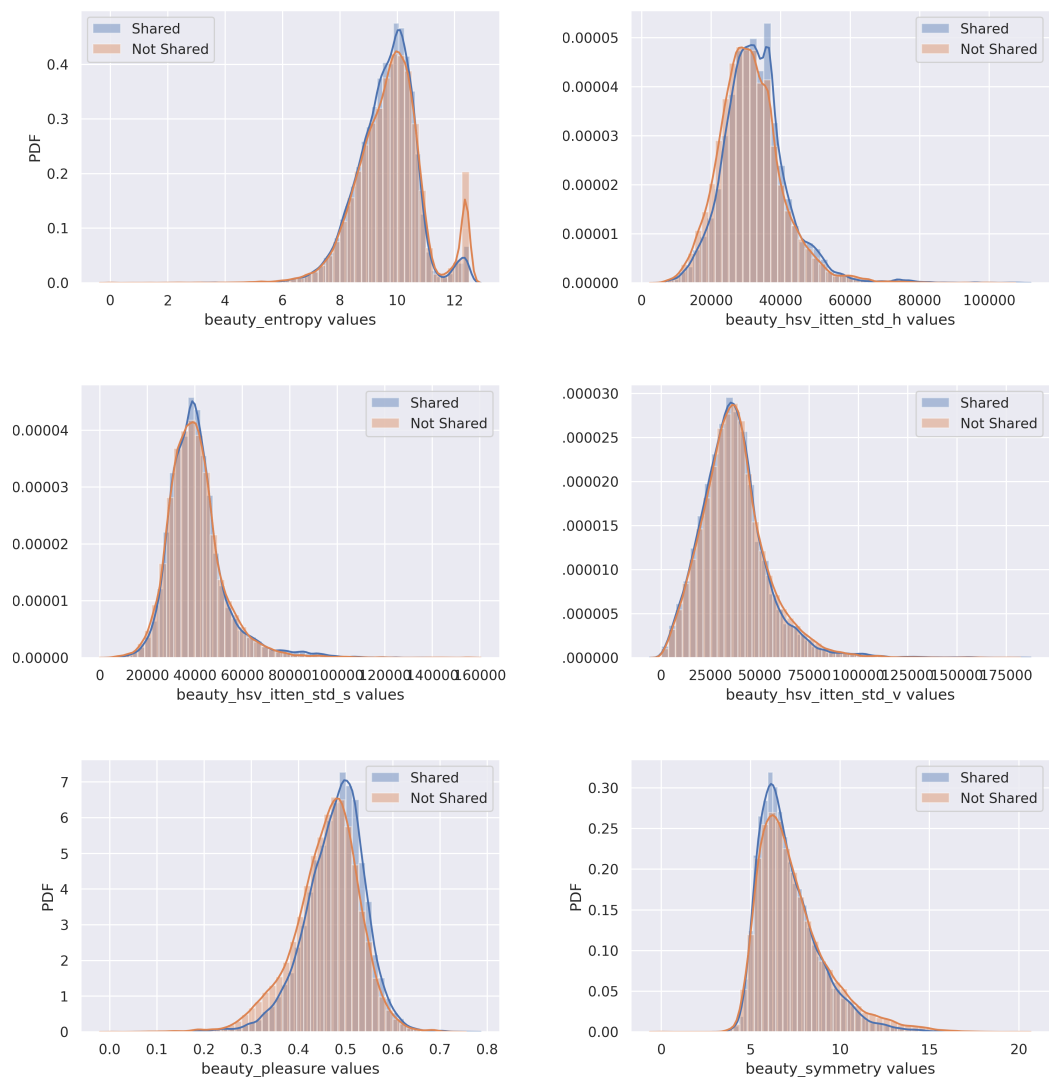


Figure 29. Feature distributions

Appendix A (Continued)

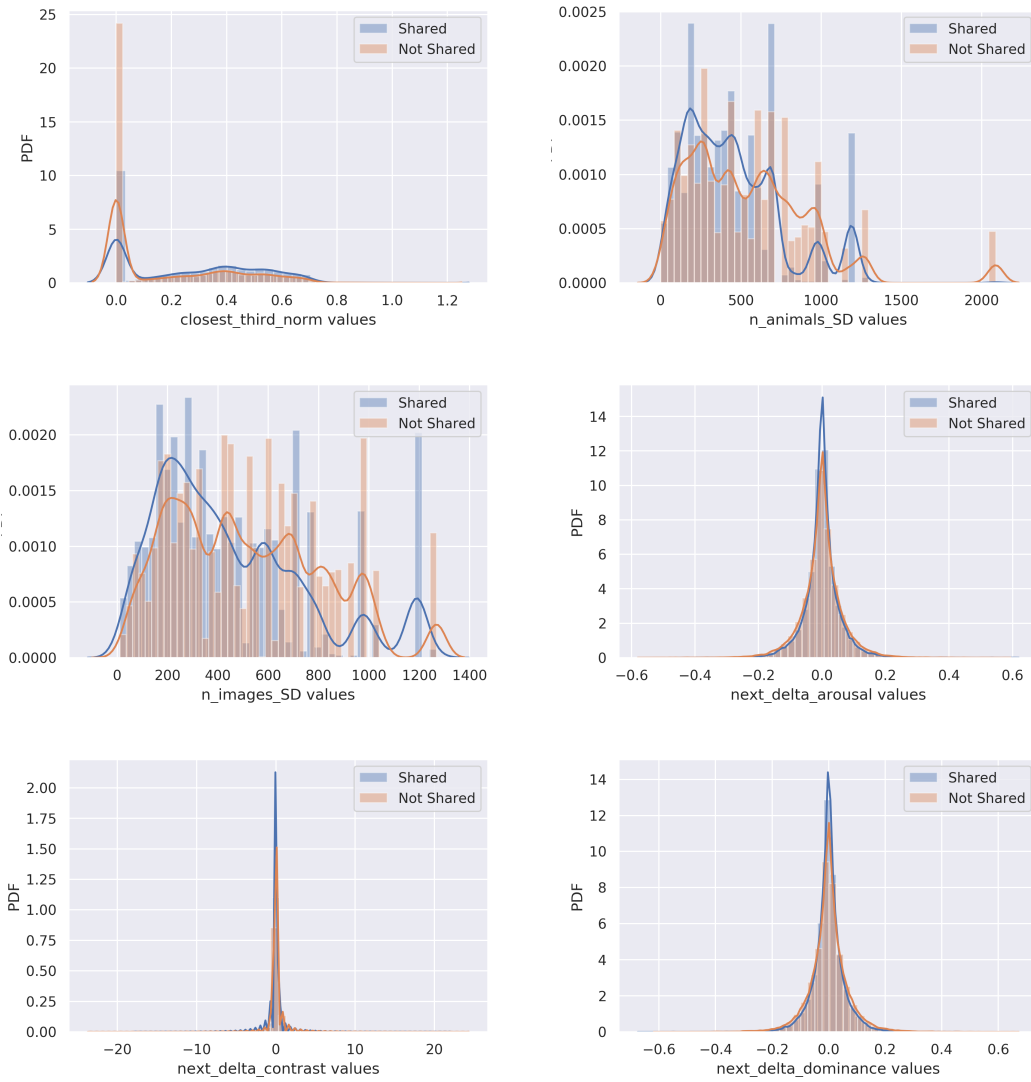


Figure 30. Feature distributions

Appendix A (Continued)

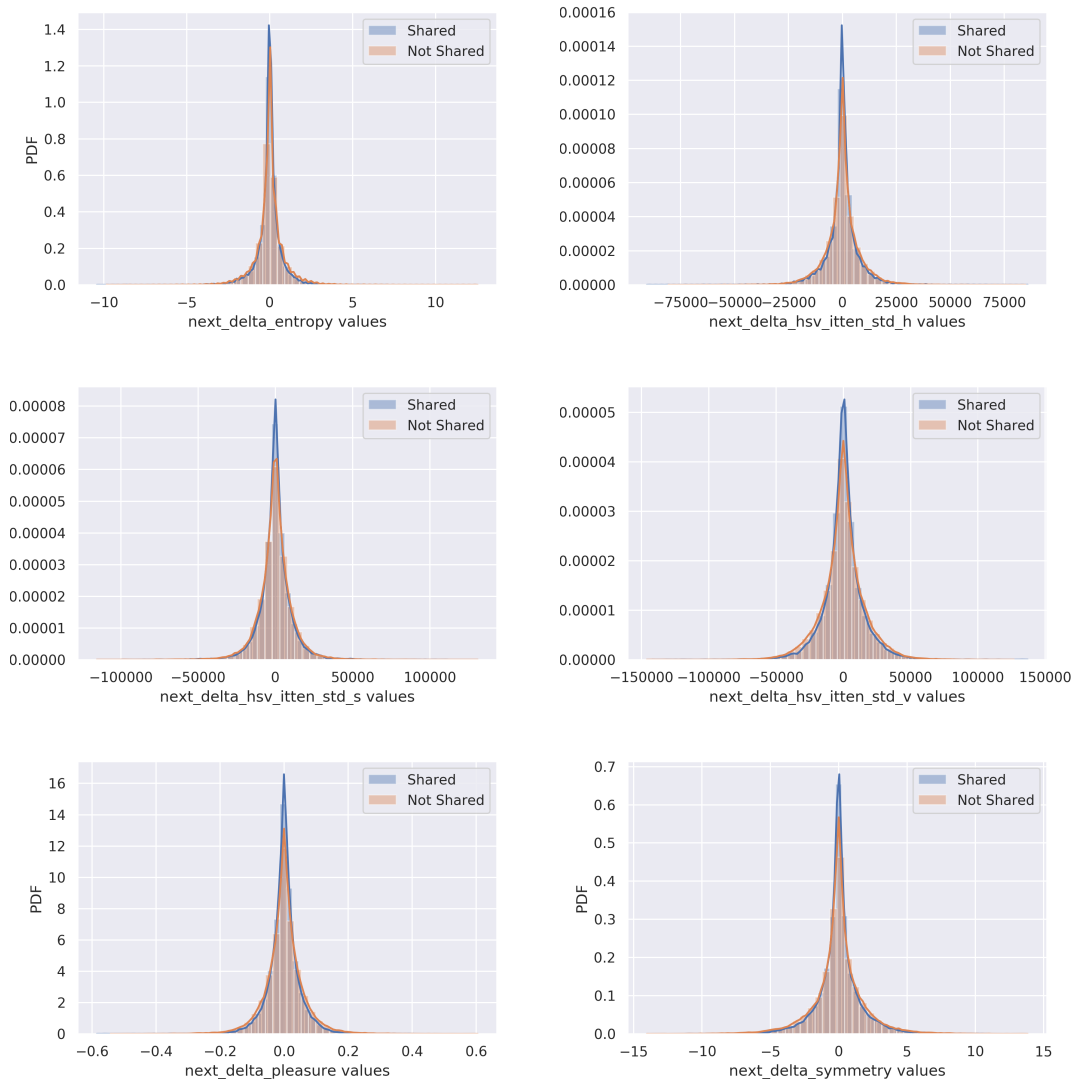


Figure 31. Feature distributions

Appendix A (Continued)

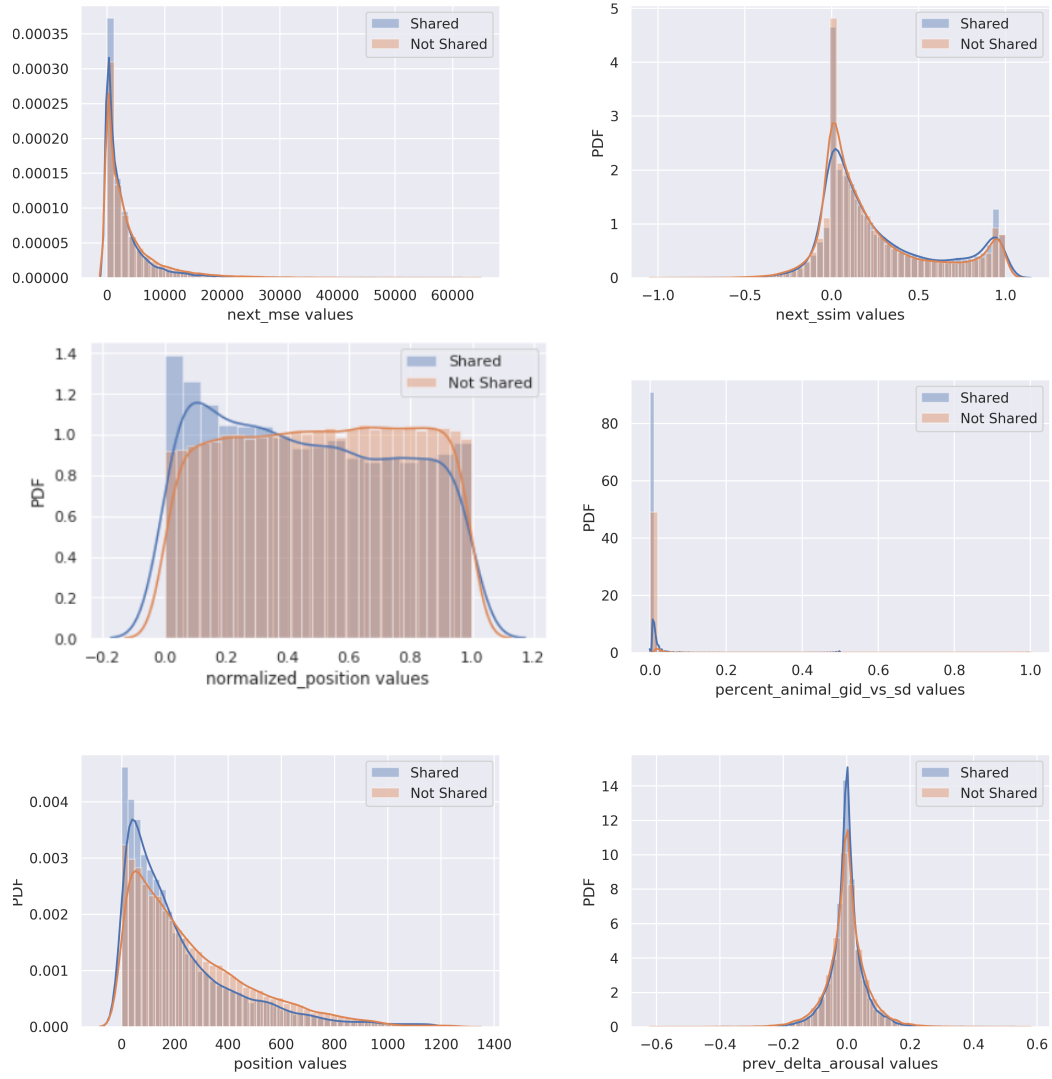


Figure 32. Feature distributions

Appendix A (Continued)

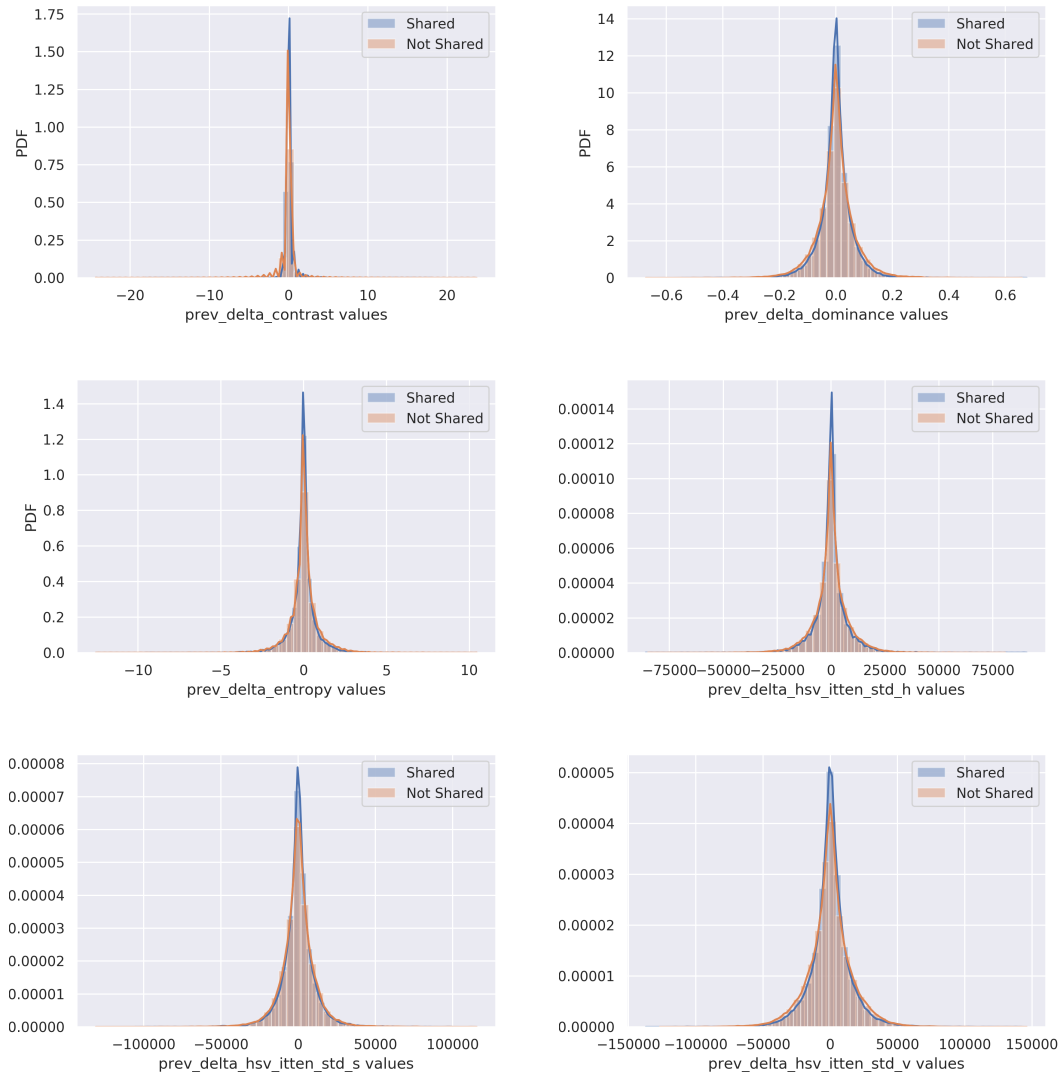


Figure 33. Feature distributions

Appendix A (Continued)

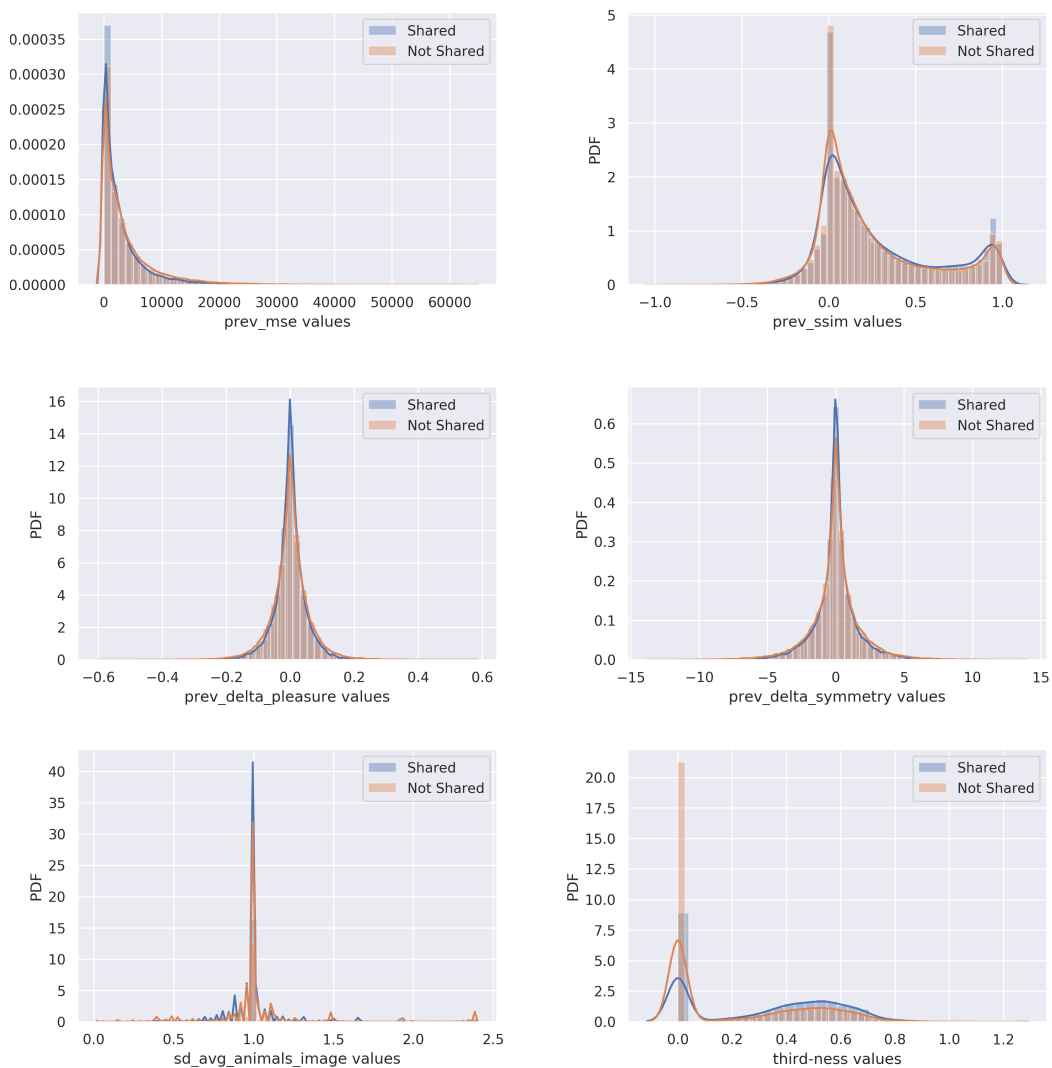


Figure 34. Feature distributions

Appendix A (Continued)

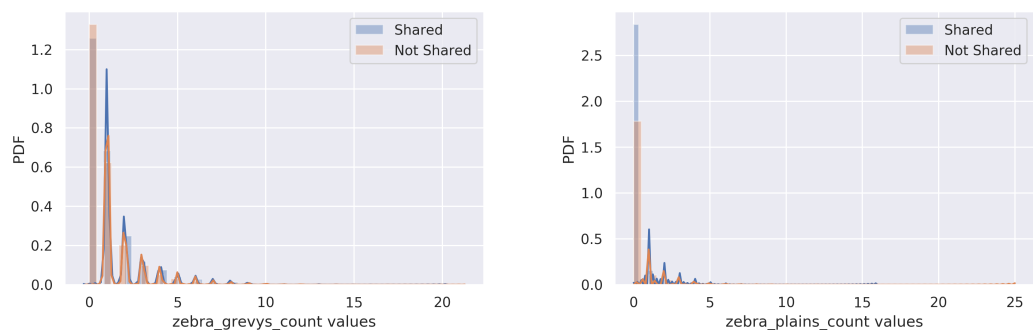


Figure 35. Feature distributions

CITED LITERATURE

1. Berger-Wolf, T. Y., Rubenstein, D. I., Stewart, C. V., Holmberg, J. A., Parham, J., Menon, S., Crall, J., Van Oast, J., Kiciman, E., and Joppa, L.: Wildbook: Crowdsourcing, computer vision, and data science for conservation. *The Data for Good Exchange* , 2017.
2. Wildbook.org: Wildbook: Software to combat extinction (official web page), 2018.
3. Tech Crunch: Facebook now has 2 billion monthly users... and responsibility. Accessed: 10-19-2018.
4. US Census Bureau official website: Population Clock: World, 2018.
5. Business Insider: Facebook users are uploading 350 million new photos each day, 2013.
6. The Associated Press: Number of active users at facebook over the years, 2013.
7. International Union for Conservation of Nature (IUCN): Iucn’s red list of threatened species, 2018.
8. Menon, S.: *Animal Wildlife Population Estimation Using Social Media Images* . 2017.
9. Berger-Wolf, T., Crall, J., Holberg, J., Parham, J., Stewart, C., Mackey, B. L., Kahumbu, P., and Rubenstein, D.: The great grevys rally: The need, methods, findings, implications and next steps. Technical report, Technical Report, Grevys Zebra Trust, Nairobi, Kenya, 2016.
10. Rubenstein, D., Parham, J., Stewart, C., Berger-Wolf, T., Holmberg, J., Crall, J., Low Mackey, B., Funnel, S., Cockerill, K., Davidson, Z., Mate, L., Nzomo, C., Warungu, R., Martins, D., Ontita, V., Omulupi, J., Jennifer, W., Anyona, G., Chege, G., David, K., Tombak, K., Gersick, A., and Rubenstein, N.: The state of kenyas grevys zebras and reticulated giraffes: Results of the great grevys rally 2018.
11. Amazon Mechanical Turk, Inc.: Amazon mechanical turk website homepage, 2018.

CITED LITERATURE (continued)

12. Buhrmester, M., Kwang, T., and Gosling, S. D.: Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data? *Perspectives on psychological science* , 6(1):3–5, 2011.
13. Specification, S. C.: Simplified version of secure digital input. *Output (SDIO) Card Specification, Version* , 1:1–20, 2001.
14. Evans, J. R. and Mathur, A.: The value of online surveys. *Internet research* , 15(2):195–219, 2005.
15. Foglio, M.: *Animal Population Estimation using Social Media Images Collection* . 2018.
16. Gosling, S. D., Vazire, S., Srivastava, S., and John, O. P.: Should we trust web-based studies? a comparative analysis of six preconceptions about internet questionnaires. *American psychologist* , 59(2):93, 2004.
17. Collins English Dictionary Online: Definition of the word "shareability", 2018.
18. Schifanella, R., Redi, M., and Aiello, L. M.: An image is worth more than a thousand favorites: Surfacing the hidden beauty of flickr pictures. In *ICWSM* , pages 397–406, 2015.
19. North Carolina State University Library: Social media data research and use, 2018.
20. Salathé, M., Freifeld, C. C., Mekaru, S. R., Tomasulo, A. F., and Brownstein, J. S.: Influenza a (h7n9) and the importance of digital epidemiology. *The New England journal of medicine* , 369(5):401, 2013.
21. Olson, D. R., Konty, K. J., Paladini, M., Viboud, C., and Simonsen, L.: Reassessing google flu trends data for detection of seasonal and pandemic influenza: a comparative epidemiological study at three geographic scales. *PLoS computational biology* , 9(10):e1003256, 2013.
22. Cook, S., Conrad, C., Fowlkes, A. L., and Mohebbi, M. H.: Assessing google flu trends performance in the united states during the 2009 influenza virus a (h1n1) pandemic. *PloS one* , 6(8):e23610, 2011.
23. Pradel, R.: Utilization of capture-mark-recapture for the study of recruitment and population growth rate. *Biometrics* , pages 703–709, 1996.

CITED LITERATURE (continued)

24. Hightower, J. E. and Gilbert, R. J.: Using the jolly-seber model to estimate population size, mortality, and recruitment for a reservoir fish population. *Transactions of the American Fisheries Society* , 113(5):633–641, 1984.
25. Wild Me: Ibeis web: web interface and api for ibeis. Accessed: 03-18-2019.
26. Lennie, P., Pokorny, J., and Smith, V. C.: Luminance. *JOSA A* , 10(6):1283–1293, 1993.
27. Fechner, G. T., Howes, D. H., and Boring, E. G.: *Elements of psychophysics* , volume 1. Holt, Rinehart and Winston New York, 1966.
28. Georges, V.: Color television system, December 27 1949. US Patent 2,492,926.
29. Machajdik, J. and Hanbury, A.: Affective image classification using features inspired by psychology and art theory. In *Proceedings of the 18th ACM international conference on Multimedia* , pages 83–92. ACM, 2010.
30. Grill, T. and Scanlon, M.: *Photographic composition* . Amphoto Books, 1990.
31. Rowse, D.: Rule of thirds. *Digital Photography School* , 2012.
32. Mai, L., Le, H., Niu, Y., and Liu, F.: Rule of thirds detection from photograph. In *Multimedia (ISM), 2011 IEEE International Symposium on* , pages 91–96. IEEE, 2011.
33. Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P.: Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* , 13(4):600–612, 2004.
34. Jolliffe, I.: Principal component analysis. In *International encyclopedia of statistical science* , pages 1094–1096. Springer, 2011.
35. Lu, Z. and Yuan, K.-H.: Welch’s t test. *NEIL J. SALKIND (HG.): Encyclopedia of research design. Thousand Oaks, Calif: Sage, S* , pages 1620–1623, 2010.
36. Freedman, D. A.: *Statistical models: theory and practice* . cambridge university press, 2009.
37. Leo, B., Friedman, J. H., Olshen, R. A., and Stone, C. J.: Classification and regression trees. *Wadsworth International Group* , 1984.

CITED LITERATURE (continued)

38. Freund, Y. and Schapire, R. E.: A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* , 55(1):119–139, 1997.
39. Chen, T., He, T., Benesty, M., et al.: Xgboost: extreme gradient boosting. *R package version 0.4-2* , pages 1–4, 2015.
40. Friedman, J. H.: Greedy function approximation: a gradient boosting machine. *Annals of statistics* , pages 1189–1232, 2001.
41. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* , 12:2825–2830, 2011.
42. Oliphant, T. E.: *A guide to NumPy* , volume 1. Trelgol Publishing USA, 2006.
43. Banker, K.: *MongoDB in action* . Manning Publications Co., 2011.
44. Waskom, M.: seaborn: statistical data visualization. Accessed: 11-19-2018.
45. Bradski, G. and Kaehler, A.: OpenCV. *Dr. Dobbs journal of software tools* , 3, 2000.
46. Friedman, J., Hastie, T., and Tibshirani, R.: *The elements of statistical learning* , volume 1. Springer series in statistics New York, NY, USA:, 2001.

VITA

NAME: LORENZO SEMERIA

EDUCATION: M.Sc., Computer Science, University of Illinois at Chicago,
Chicago, Illinois, 2018.

B.Sc., Computer Engineering, Politecnico di Milano, Milano,
Italy, 2016.

ACADEMIC Research Assistant, Computational Population Biology Lab,

EXPERIENCE: Department of Computer Science, University of Illinois at
Chicago, January - May 2018.