



D6.5

USEMP disclosure scoring framework and disclosure setting framework – v3

V1.1 2017-02-06

Georgios Petkos (CERTH), Symeon Papadopoulos (CERTH), Eleftherios Spiromitros-Xioufis (CERTH), Polychronis Charitidis (CERTH), Emmanouil Krasanakis (CERTH), Theodoros Michalareas (VELTI), Vasilios Barekas (VELTI), Yiannis Kompatsiaris (CERTH)

This deliverable provides an update on the activities carried out in relation to the development and integration of the disclosure scoring and control assistance framework (tasks T6.1 and T6.2) during the third year of the project. Significant progress is reported both in the direction of the disclosure scoring framework (T6.1) as well as in the direction of the control assistance framework (T6.2). More particularly, in the direction of the disclosure scoring framework, a refined and extended implementation of the disclosure scoring framework and its visualization have been delivered and integrated to DataBait. A number of additional inference mechanisms have been built in order to address the issue of limited training data for the collection-based inference module. Moreover, new collection-based models that leveraged MyPersonality data and new data collected from the pilots were built and integrated to the system. In addition, the problem of class imbalance in statistical learning, a problem particularly important for the USEMP scenario, has been examined. In terms of the control assistance framework, the interface has been appropriately enriched with hints and a tutorial was created to educate users and support them to more effectively control their disclosure behavior. Also, lists of pieces of content that have been ranked according to the associated disclosure score are now computed and presented to end users, suggesting them to reconsider sharing the related content. Finally, we have enriched our work on trackers by characterizing domains along the set of disclosure dimensions, thereby allowing users to obtain a better idea about what type of information each tracker has collected about them.



Project acronym	USEMP
Full title	User Empowerment for Enhanced Online Presence Management
Grant agreement number	611596
Funding scheme	Specific Targeted Research Project (STREP)
Work program topic	Objective ICT-2013.1.7 Future Internet Research Experimentation
Project start date	2013-10-01
Project Duration	36 months

Workpackage 1	User Assistance for Shared Personal Data Management
Deliverable lead org.	CERTH
Deliverable type	Prototype
Authors	Georgios Petkos (CERTH) Symeon Papadopoulos (CERTH) Eleftherios Spiromitros-Xioufis (CERTH) Polychronis Charitidis (CERTH) Emmanouil Krasanakis (CERTH) Theodoros Michalareas (VELTI) Vasilios Barekas (VELTI) Yiannis Kompatsiaris (CERTH)
Reviewers	Rob Heyman Katja DeVries Ali Padyab
Version	1.1
Status	Draft
Dissemination level	PU: Public
Due date	2016-08-31
Delivery date	2016-08-31 (revised 2017-02-06 after reviewers' comments)

Version Changes

0.1	First outline ToC by Georgios Petkos and Symeon Papadopoulos
0.2	First versions of Chapters 1-4 by Georgios Petkos, Symeon Papadopoulos and Eleftherios Spyromitros-Xioufis

-
- | | |
|-----|---|
| 0.3 | Refinements of Chapters 1-4 by Georgios Petkos, Symeon Papadopoulos and Eleftherios Spyromitros-Xioufis |
| 0.4 | Updates in all chapters by Georgios Petkos, Symeon Papadopoulos and Eleftherios Spyromitros-Xioufis |
| 0.5 | Review from Eleftherios Spyromitros-Xioufis |
| 0.6 | Updates on all chapters by Georgios Petkos |
| 0.7 | Updates in Chapter 4 by Eleftherios Spyromitros-Xioufis |
| 0.8 | Updates in all Chapters after internal reviews |
| 0.9 | Further updates after proof-reading and additional recommendations |
| 1.0 | Submitted version |
| 1.1 | Minor revisions after reviewers' comments |
-

Table of Contents

- 1 Introduction 3**
 - 1.1 Overview 3
 - 1.2 Main achievements 3
 - 1.3 Addressed reviewer recommendations..... 4
- 2 Disclosure Scoring Framework..... 8**
 - 2.1 Review of disclosure scoring framework..... 8
 - 2.2 Implementation updates and integration..... 9
 - 2.3 Visualization updates11
- 3 Inference modules.....15**
 - 3.1 Collection-based classification module16
 - 3.2 URL mapper.....17
 - 3.3 Likes mapper.....20
 - 3.4 Visual concepts mapper22
 - 3.5 Overall evaluation of integrated modules.....23
 - 3.6 Investigating the impact of MyPersonality.....24
 - 3.7 Improvements using data from the pilots27
 - 3.8 Improvements using class rebalancing.....28
 - 3.8.1 Previous work.....29
 - 3.8.2 Background and methodology.....30
 - 3.8.3 Generalizing the plug-in rule.....31
 - 3.8.4 Experiments33
- 4 User Perceptions on Predictability of Disclosed Personal Information37**
- 5 Disclosure control assistance.....40**
 - 5.1 Training and alerting DataBait users41
 - 5.2 Sharing suggestions based on the disclosure scoring framework.....43
 - 5.3 Control assistance based on image privacy.....48
- 6 Web Trackers and Do-Not-Track policies49**
 - 6.1 Enhancements on the DataBait plugin49
 - 6.2 Trackers and URL classification49
 - 6.3 Future of DataBait web tracking tool.....51
- 7 Conclusions and Next Steps52**
 - 7.1 Summary of this document.....52
 - 7.2 Overall WP6 outputs53

7.3 Directions for future work.....54

Annex 1 – Evaluation of the collection-based classifier.....56

Collection-based classifier56

Annex 2 – Mappings from likes categories to user attributes.....63

Annex 3 – OSN information disclosure tutorial.....66

Introduction on disclosure control in OSNs66

A taxonomy of personal information on social networks.....67

Some legal issues with respect to OSN providers.....68

Sharing settings basics69

Creating friends’ lists.....71

Managing the disclosure of your profile info73

Images privacy.....73

Examining the activity log.....74

View profile as seen by other users76

Blocking other users77

Applications78

Final guidelines.....80

Annex 4 – Web Plugin URL Mapped Domains.....81

References.....84

1 Introduction

1.1 Overview

This is the final deliverable produced within WP6, reporting on the achievements of work towards the development, integration and evaluation of the disclosure scoring and control assistance framework for Online Social Networks (OSNs). During the reporting period, the framework was significantly enriched and extended by: a) integrating the disclosure scoring framework and developing a user-friendly visualization, b) improving the quality of the inferences via better exploitation of existing and incorporation of additional data sources and c) integrating a new disclosure assistance module. These extensions were shown to have a very positive impact on user experience, based on the feedback received from the pilots.

The current document provides an update of D6.4, reporting on work carried out within both tasks 6.1 and 6.2 since its submission. The deliverable is structured as follows. Chapter 2 discusses the progress achieved in the context of the disclosure scoring framework. Chapter 3 presents the work carried out on developing inference mechanisms that are used in conjunction with the disclosure scoring framework. Chapter 4 presents an empirical study that was conducted about the relationship of the predictability of different types of personal information and the perceptions of users about them. Chapter 5 focusses on disclosure settings assistance. Chapter 6 discusses an extension of our work on trackers, in which domains were associated to user attributes, allowing the user to understand what type of information each tracker has accumulated about him/her, i.e. when possible visited domains are associated to some disclosure dimension and attribute. Finally, Chapter 7 concludes the document and provides a summary of the work done and a discussion on future research.

1.2 Main achievements

Significant progress has been achieved in all directions of work of WP6. In particular, the main achievements of WP6 work during the reporting period are the following:

- A refined and extended implementation of the disclosure scoring framework was delivered and integrated to DataBait (**CERTH**). This java-based implementation has been made publicly available in open source form¹.
- The visualization of the disclosure scoring framework was developed (**CERTH**) and integrated to DataBait (**Velti**). This implementation is aligned to the guidelines produced by task 6.3 (as reported in D6.3 and D6.6) and also its final version was heavily adjusted based on user feedback received internally from the consortium through the pilot studies.
- A number of new inference modules have been developed, evaluated and integrated to DataBait (**CERTH**):
 - An inference module that associates individual *likes* to personal attributes of the disclosure scoring framework.

¹<https://github.com/MKLab-ITI/usemp-pscore>

- A module that associates *URLs* to personal attributes (this module is used both for analyzing the URLs posted by the user and the domains that the user has visited, more on this will be presented in Chapters 3 and 6).
- A module that associates *visual concepts* detected in the images posted by a user to personal attributes of the disclosure scoring framework.

The implementation of these modules has also been made publicly available in open source form together with the implementation of the disclosure scoring framework. Moreover, this deliverable includes a thorough evaluation of the inference models that have been built. This evaluation helped appropriately select from the pool of developed models the ones that were integrated in the system.

- Work was carried out to investigate if the MyPersonality dataset could be used to improve the prediction accuracy of our models for specific personal attributes (**CERTH**).
- Different mechanisms for taking into account class imbalance in statistical learning have been developed and evaluated (**CERTH**). This piece of work is particularly important for the USEMP use cases, as it addresses an increasingly important problem of machine learning systems: for many of the inferred attributes, the most sensitive class is typically under-represented and, as a result, is predicted less accurately than the other classes.
- The collection-based classifiers that had been initially trained on data coming from the pre-pilots were retrained using additional data from the pilots resulting to models of higher accuracy (**CERTH**). These models have also been integrated to DataBait. Note that this and the previous three items address one of the comments that came up during the second year review, i.e. that the training data for the inference modules was rather limited and therefore prediction accuracy was not optimal.
- The interface has been enriched with hints about potential threats associated with the disclosure of different types of information and a tutorial on information disclosure in social networks (**CERTH**). This has the goal of assisting users in adjusting their disclosure settings and controlling their social network presence.
- A module that ranks content (likes, images or posts) according to its contribution to the disclosure score has been built (**CERTH**). The module also suggests to users to reconsider sharing specific pieces of content or to consider changing the relevant sharing settings. A relevant visualization has also been developed (**CERTH**) and integrated to DataBait (**Velti**). This piece of work is also part of the USEMP disclosure settings assistance tools.
- The work on trackers was extended by associating tracked domains with specific user attributes. This allows the users to better perceive what type of information (i.e. which disclosure dimensions) each tracker knows about them (**Velti and CERTH**).

1.3 Addressed reviewer recommendations

In the following, we discuss the recommendations that were made by reviewers during the second year review in relation to WP6, and describe the actions that were taken in order to address them.

Recommendation: Integration activities should be speeded up. The number of components integrated into the USEMP system is not fully satisfactory after two years of project work.

Actions: The following modules developed within WP6 have been integrated during the reporting period:

- Disclosure scoring framework (Chapter 2)
- Visualization of the disclosure scoring framework (Chapter 2)
- Collection-based classifiers (Chapter 3)
- URLs mapper (Chapter 3)
- Likes mapper (Chapter 3)
- Visual concepts mapper (Chapter 3)
- Control settings assistance/ranked list of content (Chapter 5)
- URL classification for web trackers (Chapter 6)

Recommendation: It is not clear, why nearly all of the deliverables of WP5 and WP6 are of dissemination level RE. Especially, for the work on the disclosure setting framework, impact could be increased by allowing the community to read and react on the work by making those deliverables public. Although this was defined in the DoW, it is recommended to revisit those decisions. For a publicly funded project it should be carefully checked, what really requires the status of restricted access.

Actions: Both the previous and this deliverable are now of dissemination level PU.

Recommendation: There has been very diverse development of concepts and technologies in WP5 and WP6. This was fine for year 2 to also foster advances in research in the respective areas. However, now it is time to focus on stabilizing the technologies, which are relevant for the implementation of some of the characteristic USEMP features such as the disclosure scoring framework or the support fine granularprivacy settings. Technical activities should focus on integrating and stabilizing those parts that are core to USEMP in order to ensure impact of the project, also focusing project resources on these tasks. A critical assessment should be performed there.

Actions: To address this recommendation, it was decided that focus should be put on the disclosure scoring framework and the associated inference modules. This decision was made because it was felt that the disclosure scoring framework has the potential to mostly affect the users' feeling about information disclosure in OSNs. In fact, as reported in D8.5, most users in both sites of the pilots found the disclosure scoring framework very useful for identifying potentially sensitive information and at the same time they found it very easy to use. It should also be noted that pilot users in the Swedish site found that the inference results produced were a good summary of their habits and personality, whereas Dutch users were rather neutral about it. In addition, further effort focused on the development of the disclosure settings assistance module and its integration with the disclosure scoring framework (please see Chapter 5).

Recommendation: As already raised in the previous review, the situation with the limited data set is still an issue. It seems that the system will at least in the beginning have a quality problem due to a lack of training data. This is critical for the project

impact, since it is expected that it will be exactly the initial performance of the system that will influence its take up. It is recommended that the consortium carefully revisits this issue and also reconsiders the legal viability of using anonymized training data from other sources at least in the start-up phase of the system.

Actions:

- Additional inference modules that were formulated using other sources of data were further developed and integrated: The likes mapper, the URLs mapper and the visual concepts mapper.
- The collection-based inference modules have been retrained with the additional data collected during the pilots.
- The technical feasibility of using data from the MyPersonality dataset for training improved classifiers was investigated. We experimentally demonstrated that leveraging external data such as MyPersonality could result in new, more accurate models for specific user attributes.
- The problem of class imbalance was examined with the goal of increasing accuracy on classes with a scarcity of training data.

Recommendation: The project has evolved considerably during the last years especially regarding the coordination efforts in a highly multidisciplinary environment. However, considering that the project is entering into its third and last year a deeper coordination and integration among all the individuals and professional involved in the project should be achieved. This integration, based on the standpoints and milestones already reached shall finally move forward the USEMP project chances to get a great outcome achieving its goals and objectives.

Actions: WP6 has worked in close collaboration with WP5 in various ways, the most prominent being the collaboration for the image privacy module that assigns a privacy label to the images posted by the user and subsequently makes appropriate suggestions to the user in order to change the images' sharing settings (please see D5.6 as well as Chapter 5 of this document). Moreover, feedback from WP3 regarding any legal aspects of the work carried out within WP6 (e.g. licenses for the datasets used) was often received. Additionally, WP6 has worked in close collaboration with WP8, especially during the pilot studies. In particular, valuable feedback about the integrated modules was fed into WP6, resulting also in improvement of these modules.

Recommendation: Pursuant the previous recommendation, it should be considered not to restrict the scope of the project to just one OSN. At this stage the previous number of OSNs has increased. However, considering that the project is entering into its last year it would be advisable to broaden the scope and to try to integrate as many OSNs as possible.

Actions: Almost all the developed inference modules presented here are either directly applicable or can be easily adapted to work with any OSN (and in most cases with non-OSN personal data). For instance, the URLs mapper is directly applicable to any OSN in which users may post text that contains URLs (practically all OSNs) and the visual concepts mapper is directly applicable to any OSN in which users may post images (practically all OSNs). The collection-based classifiers, on the other hand, currently take into account

Facebook likes, posts and images simultaneously but can be easily adapted to work with any subset of these data types. Thus, all outcomes of WP6 are possible to either directly apply on data coming from different OSNs, or are easy to adapt to be applicable to new OSNs.

2 Disclosure Scoring Framework

This chapter presents the progress made with respect to the disclosure scoring framework since the submission of D6.4. As a quick reminder, the disclosure scoring framework is a tool that aims at quantifying and succinctly representing the exposure of OSN users' personal information. It is associated with one of the main goals of the work carried out within the project: *raising the awareness* of users with respect to their presence at an OSN.

A first design of the disclosure scoring framework was presented in D6.1 and then an update was provided in D6.4. In the period since the submission of D6.4, the focus has been on the integration of the disclosure scoring framework to DataBait and the improvement of its visualization according to the latest results of task 6.3 (the final results of which were presented in D6.6). Importantly, this effort resulted in a refined and extended implementation of the framework, which has been successfully integrated to DataBait and has been made publicly available in open-source form² along with a number of inference modules. More details about the inference modules will be provided in the following chapters.

In the remainder of this chapter, we first provide a quick reminder of how the disclosure scoring framework is structured and then look into some details of the refined implementation and its integration to DataBait. Finally, we look into the development of the visualization.

2.1 Review of disclosure scoring framework

In this section, the structure of the disclosure scoring framework is briefly reviewed. The core of the framework comprise a set of personal attributes that have been identified as potentially sensitive under different scenarios (e.g. 'gender', 'location', 'sexual orientation', etc.). Each attribute can take a number of values; e.g. the attribute 'gender' can take the values 'male' or 'female'. To make the set of user attributes more structured and therefore easier to communicate, the attributes have been grouped in a number of *dimensions*. For instance, the attributes 'age', 'gender' and 'nationality' are grouped under the *demographics* dimension. Moreover, it is recognized that the exposure of the user can be of interest at different levels of granularity, e.g. at the dimension-, attribute- or value-level; therefore, scores are provided at all three levels. Effectively, each dimension, attribute or value may be characterized by diverse scores such as the visibility of the corresponding information, the level of control that the user has with respect to the disclosure of this information, the sensitivity of different types of information, etc. The hierarchical structure of the disclosure scoring framework (each dimension has a number of attributes and each attribute can take a number of values) along with the scores at each level is shown in Figure 1. The most important score is the *disclosure score* that attempts to quantify the overall risk associated with the disclosure of the information associated with the particular dimension, attribute or value. At the values level, this is computed as the product of visibility, confidence and sensitivity. All scores at the upper levels – including the overall disclosure score - are computed by aggregating the scores of the lower levels using appropriate operators.

²<https://github.com/MKLab-ITI/usemp-pscore>

It is important to note that all input data are fed into the framework by a number of inference modules that process the OSN data of users and attempt to detect the possible values of users' attributes. In the next chapter, the inference modules that have been developed and integrated into the system are presented in detail. In the next section— where some details about the implementation of the scoring framework are provided - we will also look at the way that the inference modules interact with the disclosure scoring framework.

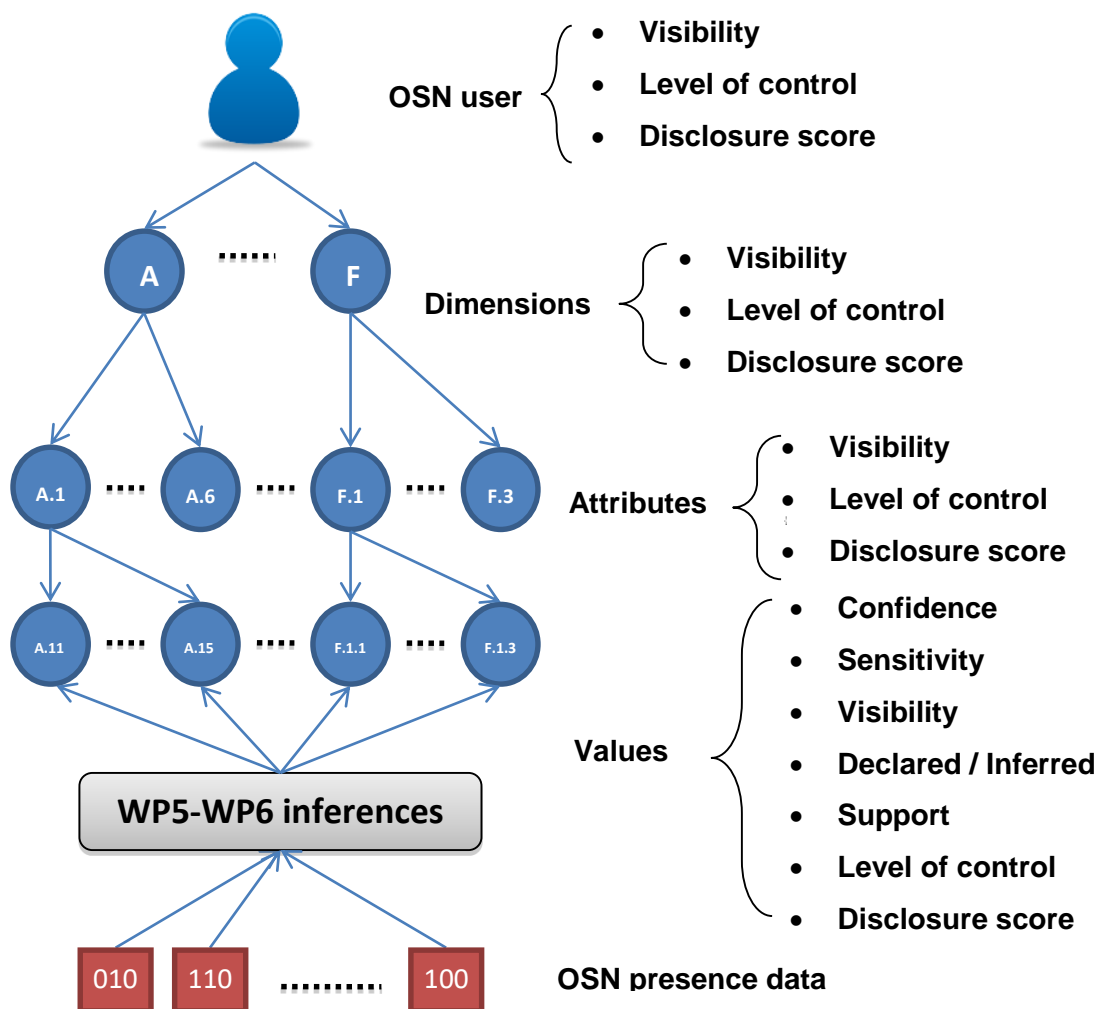


Figure 1: Overview of the disclosure scoring framework

For more details on the inner workings of the framework, the different scores as well as on the design rationale, please refer to D6.1, D6.4 and (Petkos et al., 2015).

2.2 Implementation updates and integration

One of the main activities on the reporting period has been the refinement of the framework implementation (in Java) and its integration to DataBait. In the following, we look into some details of the implementation, its integration to DataBait and how it interacts with the rest of the system components.

The first thing to note about the implementation of the disclosure scoring framework is that it stores its data in a mongo database instance and that a record is maintained for each user.

The serialization of the objects that represent the scores of the user, as well as the storage to and retrieval from the database take place via the services that have already been developed at the backend of the system. Therefore, integration with respect to storage is seamless.

Internally, the main Java class of the framework is the *DisclosureScoringFramework* class. This class interacts with the mongo database and can be used to retrieve or store the scores of a given user. The retrieval method of this class returns an object of type *ScoringUser*, which represents the scores of the given user.

The implementation of the disclosure scoring framework is also flexible with respect to the interaction with the inference modules. New inference modules can be added seamlessly to feed data into the scoring framework by just calling the appropriate data feeding method, named *ScoringUser.addSupport*, which requires the following arguments:

- String *dimension_name*, *attribute_name*, *value_name*
- List<String>*support_pointer_data_ids*
- Double *confidence*
- Constants.*InferenceMechanism inferenceMechanism*

The first three attributes are the dimension, attribute and value name for which the inference result applies. Subsequently, there is a list of references to the OSN data that were used for the inference, a numerical value that represents the confidence of the inference and finally the type of inference mechanism that has been used. In case an inference mechanism produces results for multiple values under the same attribute, then it should call this method multiple times, once per value. After a new inference result is added to the scoring structure, the scores are automatically recomputed. Moreover, the addition of new dimensions, attributes or values is very simple and takes place by calling the above method with a new dimension, attribute or value name. Internally, the above method also computes, a number of scores such as the visibility score and the visibility label.

Triggering of the computation and upwards aggregation of the scores take place whenever the above method is called by some inference module. In its turn, an inference module is called when there is a notification to DataBait that some data of the user have changed. In particular, a callback function is triggered when there is a notification from the OSN that there has been an update in the user's data. This callback function actually fetches the data from the OSN and according to the type of the updated data, it triggers the execution of the appropriate inference module. In the case that there is a notification about the deletion of some piece of data, then apart from the re-execution of the inference module, any parts of the scores of the user that depend on that piece of data are deleted and the scores are re-aggregated.

An important part of the scores handled by the framework is the sensitivity of the different dimensions, attributes or values. Initially, for a new attribute or value falling under a specific dimension, default sensitivity values are used. These default values have been determined based on feedback obtained from the users that took part in the early pilots. In particular, on a scale from 1 (less sensitive) to 7 (most sensitive), participants indicated how sensitive they considered each dimension to be. The values, obtained by averaging the responses of users and mapping on the interval from 0 to 1, can be seen in Table 1.

Table 1. Default sensitivity of different dimensions

Dimension	Default sensitivity
Demographics	0.469
Employment	0.831
Relationship	0.592
Psychology	0.771
Sexuality	0.544
Politics	0.718
Religion	0.531
Health	0.938
Location	0.587
Hobbies	0.613

Importantly though, it is recognized that the importance of the different types of information varies among different users. Therefore, although we provide default values (that express the *average* attitude of users towards dimensions), we allow each individual user to set the sensitivity of different dimensions, attributes or values according to their preferences. We provide methods that can be called via a REST call (*/user/scoring/update*) from the front-end. As will be shown in the next section, the user can select some dimension, attribute or value and adjust the value of its sensitivity using a slider. The new value of the sensitivity is then sent to the back-end, which performs the required computations and updates the scores. Finally, the front-end receives the data from the back-end and passes it to the visualization via another REST call (*/user/scoring*).

2.3 Visualization updates

In D6.4, a first version of the visualization of the disclosure scoring framework was presented. That version of the visualization was a first attempt at implementing the design that was delivered by task 6.3 and at that time described in D6.3. In the meantime, D6.6, which was an update of D6.3, became available.

Therefore, the focus in the reporting period was on updating the visualization of the scoring framework and on integrating it to the DataBait platform. First, its interaction with the DataBait back-end is based on two main REST methods:

- */user/scoring*: This returns the scoring record of the user in JSON format. The record is then parsed by the client JavaScript code which creates the visual elements.
- */user/scoring/update*: This sends sensitivity values (provided by the user) to the back-end. The back-end re-computes the scores and then the front-end receives again the updated scores and refreshes the visualization.

Importantly, after focus group discussions with end users and other consortium partners, it was made clear that the scoring framework included too much information that could potentially overwhelm or confuse users. For instance, explicit confidence values as well as level of control and visibility scores may obstruct some users from focusing on the main disclosure score. Therefore, it was decided to have two versions of the visualization: a simple one intended for use by less experienced users and a full version that includes all information and is intended for more advanced users. A snapshot of the simplified version of the visualization is shown in Figure 2, whereas the full version of the visualization is shown in

Figure 3. The main difference between the two versions is that, while the simplified version only shows the overall disclosure score and sensitivity (it also describes confidence in a verbal manner), the full version additionally provides explicit values for the level of control, the visibility score, the visibility label and confidence.

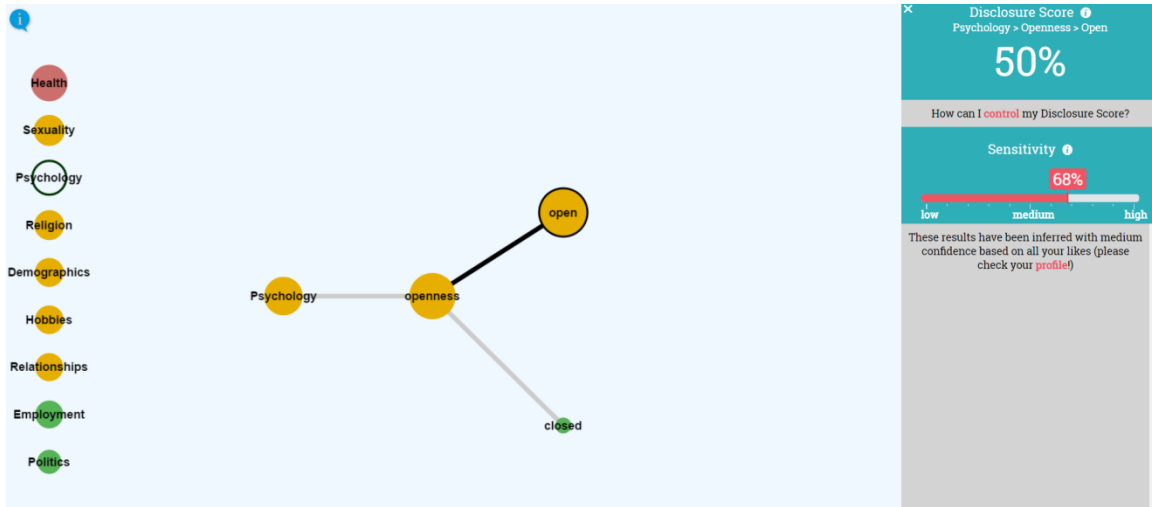


Figure 2. Snapshot of the simplified version of the visualization

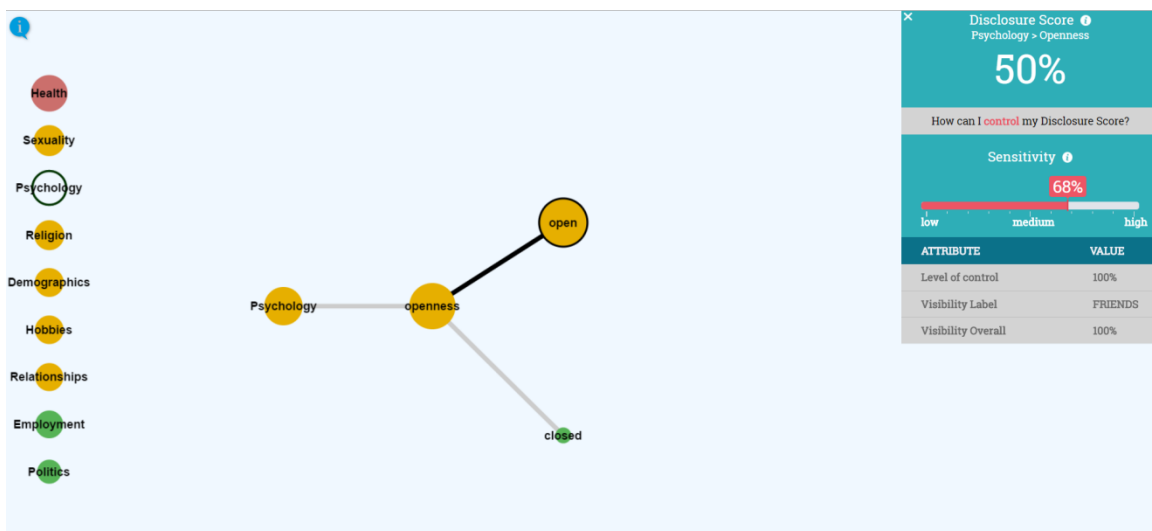


Figure 3. Snapshot of the full version of the visualization

The two versions of the visualization are the same in all other respects. For instance, the top level visualization that shows the dimensions upon start up (Figure 4) is the same in both versions.



Figure 4. Top level visualization showing a summary of the dimensions

Of particular note is the bar at the top right of the visualization (also shown separately in Figure 5) that allows users to adjust the sensitivity of some dimension, attribute or value, according to their preferences. Importantly, the system responds almost instantly after a change in sensitivity, as both the required computation at the backend and the update of the scores at the front-end are performed in almost real time.

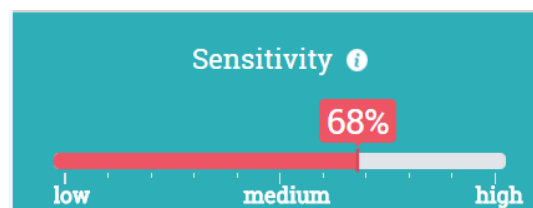


Figure 5. The slider used in the visualization to adjust the sensitivity.

There are a few additional design details worth discussing. The first is that the overall disclosure score is emphasized in both the simplified and full visualizations. This was a conscious decision and was made in order to guide the user to the concept of the overall disclosure score as an overall measure of risk associated with information disclosure in the social network.

In addition, the overall disclosure score is associated to both the size and the color of the bubbles. Moreover, information buttons were added to various points of the visualization to explain various aspects of the scoring framework, the individual scores and the internal computation. Also, an introductory information panel is now shown when the visualization loads and explains the main aspects of the disclosure scoring framework (Figure 6).

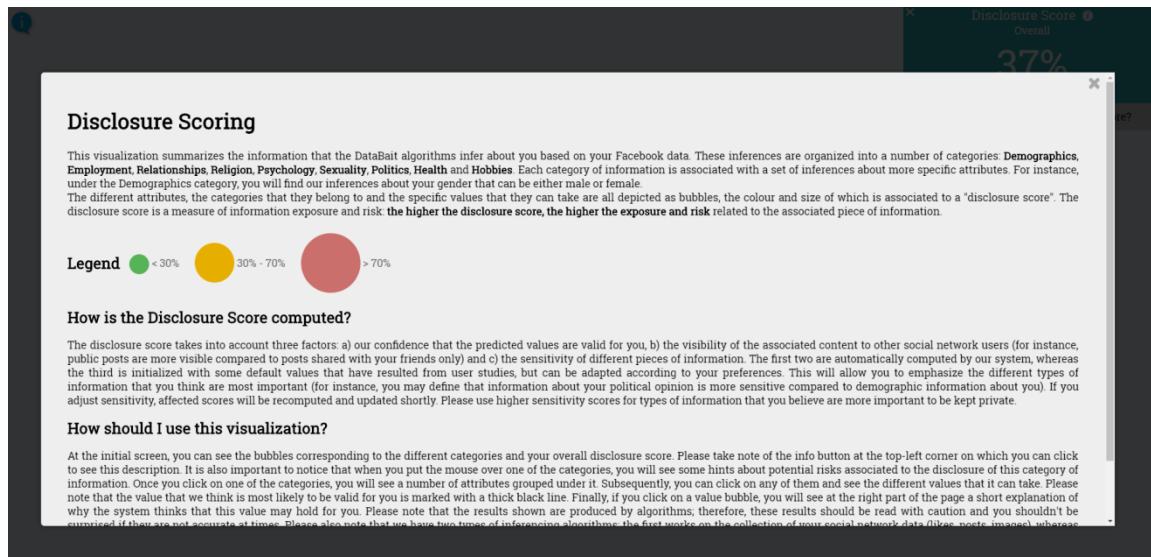


Figure 6. Introductory information panel about the disclosure scoring framework

A number of additional pop-ups and hints have also been added to the visualization with the goal of assisting the user to better control the disclosure of their information. These examine specific risks and scenarios and are going to be presented in Chapter 5, along with the disclosure settings assistance framework.

Finally, it should be stressed that the disclosure scoring framework is independent of any specific social network: as long as there are appropriate inference mechanisms that can analyze data from a social network (or even personal data generated by different systems and applications), the disclosure scoring framework will be able to produce an appropriate summary of the disclosed information. The applicability of the different inference modules to data coming from different social networks will be analyzed in the next chapter, where the developed inference modules are presented.

3 Inference modules

The disclosure scoring framework relies on the results of a number of inference modules to provide end users with valuable insights regarding their personal information disclosure. Therefore, the development of high-quality inference modules has been among the key goals of the work carried out within WP6.

In D6.4 a set of classifiers was developed. These classifiers are what we will refer to as *collection-based*, i.e. they take as input the complete set of OSN items associated with a user, and in particular they take into account the following:

- set of *likes* of the users;
- *text content* of users' posts;
- set of *visual concepts* detected in the images posted by the users.

It is important to emphasize that these classifiers take into account the *collection* of all these types of content and do not work on a per-item basis. These classifiers examined all the considered personal attributes and were initially trained with data that were obtained during the pre-pilots (OSN data were used as inputs, while responses to the pre-pilot survey were used as ground truth for the target values). Moreover, a wide variety of classification methods as well as learning mechanisms (feature selection, fusion, multi-label classification, etc.) were tested. Considering the fact that the pre-pilots provided data for 170 users as well as the difficulties posed to learning algorithms from this type of data (e.g. class imbalance, systematically misreporting the response to sensitive questions in survey data), the obtained accuracy of the classifiers (can be considered satisfactory (average AUC for all attributes was 0.63, which is much higher than random). Nevertheless, we found that there was space for improving the accuracy of our predictions. To this end, during the third year, we worked on a number of improvements:

- 1) Several additional inference modules were developed, each of which – contrary to the collection-based classifiers – works on individual pieces of content. All these inference modules have been fully integrated in DataBait.
- 2) We investigated the potential of using external datasets for training. In particular, we explored the possibility of leveraging the MyPersonality dataset, a dataset that is far more extensive than the one obtained through the USEMP pre-pilots. With its use, we demonstrate that it is possible to improve the accuracy of the predictions of the collection-based classifiers for particular attributes.
- 3) We also took advantage of the additional data that was made available from the USEMP pilots. Despite the limited number of additional training examples that became available, a noticeable increase in performance was obtained.
- 4) We examined a number of approaches for balancing the classification performance between different classes of a user attribute. We observed that due to class imbalance (large differences in the frequencies of different classes of a target attribute), the models tended to accumulate most of their errors on the minority classes (which are usually the most sensitive ones). While this behavior leads to a smaller overall number of errors, it is not well-fitting to the USEMP scenario because users that belong to minority classes (e.g. non-heterosexuals) are usually those most

vulnerable to the disclosure of their personal information. Applying the developed balancing methods, we managed to significantly reduce the number of errors related to the minority classes, while preserving high overall accuracy.

In the next sections of the chapter, we look at each of these directions of work in detail. Note that the following inference modules have been made publicly available as part of the open source package that also contains the code for the disclosure scoring framework.

3.1 Collection-based classification module

The collection-based classifiers are first examined. These classifiers take as input the complete collection of a user's OSN data and are based on work that has been presented in previous deliverables. More specifically, having tested different configurations, e.g. different classifiers, fusion techniques, input features, etc. (for more details please see D6.4), the best performing configuration was picked for each attribute in order to integrate the respective model to the system. The models were built using the Weka machine learning library³. Importantly, this inference module applies to almost all considered attributes and produces results for all users. The only attributes that it does not consider are a few attributes that were added after the pre-pilots and for which there were no training data. One such attribute is the 'medicines' attribute under the 'health' dimension, for which there was no relevant question in the pre-pilot questionnaire.

From an integration point of view, since this module takes as input the collection of all likes, posts and visual concepts detected in the images of the user, the execution of the module is triggered when there is a change in any of these types of content. For instance, when a like is added or removed by a user and DataBait is notified about it, then all inferences are re-computed for that user. Also, when feeding the results of an inference module, it is important to define a value expressing the confidence of the prediction. In the case of the collection-based classifiers, this is provided by the probability that is output by the classifier itself as provided by Weka.

We now proceed to the evaluation of this module. To this end, we utilized the ground truth details of the 170 users that took part in the pre-pilot to compute the following performance measures:

- % of users for which results are produced (coverage)
- Precision, Recall and F-score for each class value
- Accuracy for each attribute

The evaluation of the collection-based classifier differs a bit to that of the other inference modules, in that evaluation needs to be carried out on the same data as the data that was used for training. To obtain valid performance estimates, a 10-fold cross validation procedure was carried out (therefore for each user attribute, the respective user's data was not used for training). Moreover, due to the limited size of the dataset, the 10-fold cross validation procedure is repeated multiple times and the estimates are subsequently averaged. This is done in order to obtain reliable and stable estimates. This issue is discussed further in later

³ <http://www.cs.waikato.ac.nz/ml/weka/>

sections. The obtained results for the collection-based classifier can be seen in Table 14 in Annex 1.

Various remarks can be made about the performance of the collection-based classifier. The first is that the overall average accuracy (0.7331 over all attributes) is satisfactory given the size of the data and is much better than random guessing. Moreover, the performance of the classifiers varies for different attributes. For instance, the accuracy is rather low for the attribute 'reading' (0.5764) and for the attribute 'living situation' (0.4882), but it is high for 'agreeableness' (0.8764) and for the attribute 'nationality' (0.9224). Another important remark is that due to the fact that examples of some classes are very scarce (e.g. very few people in the dataset go to the theatre or have animals), in some cases the classifier always predicts the majority class. This is an issue that we attempt to deal with our work on rebalancing that is presented later in the chapter. Moreover, it should be kept in mind that there is an upper limit on the accuracy of the considered classifiers. The main reason is that some user attributes may not be reflected at all in the user's OSN data.

One important comment is that although the collection-based classifiers have been trained with data coming from Facebook, it is possible to also apply them on data from other social networks with only limited adaptations. For instance, in the case of Flickr, it would not be possible to use likes, but using the set of visual concepts detected in the images and the textual elements of the posted images, the collection-based classifier would still be able to make predictions for several of the attributes of the disclosure scoring framework.

From collection-based to item-based inferencing: Whereas the collection-based classifier works on the collection of all the OSN data of the user, the inference modules presented in the next sections work on individual pieces of content:

- URLs included in the posts of the user.
- Facebook pages liked by the user.
- Visual concepts detected in the images posted by the user.

An advantage of item-based inference modules is the following: whereas in the case of the collection-based classifier the association of the results that it produces to a single piece of content is not easy, in the case of these modules it is the direct result of their output. This is important for explaining to the user the rationale behind the generated inferences. Indeed, for inferences produced using any of these three modules, the visualization of the scoring framework associates the inference results to the corresponding liked pages, images or posts. On the other hand, for results produced using the collection-based classifier, the user is pointed to the set of all the content that has been used as input. This problem is relevant also for the work carried out in the context of disclosure settings assistance and will be revisited in Chapter 5.

3.2 URL mapper

This module detects the URLs that have been included in the posts of the user and maps them to some specific attribute's value. For instance, assuming that the user has posted a URL pointing to a sports website, the module could infer that the user is likely interested in sports. Only the URL itself is used for inference and not the content of the respective page.

To obtain the mapping from URLs to attribute values, the well-known Dmoz⁴ directory is used. Dmoz is an open web directory that associates URL domains to categories. It contains more than 5 million websites and more than 1 million categories. To obtain the required mapping from URL domains to the attribute values of interest, we first manually selected a number of keywords for each attribute (for instance, for the attribute 'alcohol', some of the keywords used include *beer, vodka, whiskey, alcohol, gin, wine, brandy*) and then retrieved and filtered the Dmoz categories that contain any of them. Subsequently, we identified the URL domains that belong to these categories and included them in a static file that directly associates them to attribute values. For instance, the URL domain <http://www.sports.com/> is associated to the value 'yes' of the attribute 'sports' under the 'hobbies' dimension.

It should be noted that it is not possible to assign all attributes to URL domains with this method. For instance, it is difficult to imagine a URL domain that is associated to the attribute 'BMI class' (that takes the value 'healthy' and 'non-healthy'). Additionally, not all URLs posted by the users can be associated with some of the considered attributes using this method. This becomes clear by examining the domains of the most frequent URL domains posted by the users of the pre-pilots:

- 1) <https://www.facebook.com/> (20,465)
- 2) <http://www.youtube.com/> (5,290)
- 3) <http://www.standaard.be/> (566)
- 4) <http://apps.facebook.com/> (520)
- 5) <http://www.hulstaertphoto.us/> (466)
- 6) <http://www.demorgen.be/> (378)
- 7) <http://www.nieuwsblad.be/> (365)
- 8) <http://youtu.be/> (275)
- 9) <https://foursquare.com/> (212)
- 10) <http://vimeo.com/> (151)

It is clear that just from these URL domains it is very difficult (if at all possible) to make any associations to personal attributes. To make use of such URLs, it would be necessary to take into account the content of the pages that they point to. To give an example, consider the case of a user that has posted a link to an article about beer in www.cnn.com. When only the URL domain is taken into account, it would be difficult to make an appropriate association to an appropriate attribute value. Instead, if the URL page content was taken into account, a sophisticated system would likely be able to make an association to the attribute 'alcohol' under the 'health' dimension. However, developing such a system was not possible in the context of this project because it would involve: a) performing a significant number of potentially costly HTTP requests to fetch the content from these URLs, and b) developing reliable automated ways of identifying the part of the web pages that are of interest (e.g. out of a complex web page that contains a lot of text snippets, one would need to select only the one that is of interest for the user visiting the page). Due to the large effort that would be necessary to address these complications, this option was not pursued.

There are some details that should be clarified about this module. Whereas the collection-based classifier would return an estimate about the likelihood of all possible values of an attribute, this is not always the case with the other inference modules. For instance, when a

⁴<http://www.dmoz.org/>

user posts the URL of a bookshop, then we can gain some confidence that the user is a reader and we can accordingly set the confidence of the value 'yes' for the attribute 'reading'. On the other hand, it is not likely to come across some URL that implies that a person is not a reader. Therefore, the URL mapper may detect users that are readers but it cannot detect users that are not readers. On the other hand, this would be possible for the collection-based classifier, as it has been trained with data for both readers and non-readers. In this perspective, this and the other item-based modules act as detectors of specific values of the users' attributes, rather than classifiers. Moreover, regarding the confidence value that is fed from the URLs mapper to the disclosure scoring framework, a fixed value (0.8) is used when there is a single URL domain posted that is related to some attribute. When the number of URL domains related to that attribute increase, then the confidence increases in a linear manner: the more pages relevant to some attribute a user has posted about, the more our confidence increases.

Moving on to examine the performance of the URL mapper, the relevant results are shown in Table 15 in Annex 1. Please keep in mind that contrary to the case of the collection-based classifier, the data on which the evaluation is carried out has not been used for training and therefore there is no need for resorting to cross-validation. The same also holds for the other two inference modules that will be examined in the next subsections.

Examining the results in Table 15 it is clear that the URL mapper performs particularly well in terms of accuracy for specific attributes. For instance, the accuracy for the attribute 'reading' is 0.8751, which is quite higher than that achieved by the collection-based classifiers. It is also interesting to note that the accuracy for the attribute 'sexual orientation' is perfect (1.0).

Another remark about the results is that clearly, it would not be possible to consider completely replacing the collection-based classifier with the URLs mapper. The reason is that only a fraction of the users have posted any URL that can be associated to some user attribute and therefore, recall is quite low for all attributes. In particular, 141 URLs out of the 7716 (1.82%) that were included in the users' posts were recognized and could be used by the URL mapper. This led to the possibility of classifying 106 of the 442 DataBait users (24%) to at least one of the disclosure scoring framework attributes solely based on the URL mapper. Given that there are some attributes for which the precision and accuracy of URL mapper is quite high, the goal is that for these attributes this inference module could be combined with the collection-based classifier and other item-based inference modules in order to catch some cases that the other inference modules cannot. In that case, the overall precision and accuracy should also increase.

In the results table, please also notice that attributes for which accuracy is above 0.5 are marked with bold and underlined. These are attributes that are considered for integration to the system (please see the overall evaluation later in the chapter). Eventually, there is a process for selecting the inference modules that will be used for each attribute.

It should also be noted that the URL mapper can handle some attributes for which it was not possible to produce evaluation data. There are two reasons for this. The first is that some attributes have not been considered in the pre-pilot survey and therefore there is no ground truth for them. The second is that for some attributes no URLs posted by the users that took part in the pre-pilot match those that the URL mapper considers for this attribute and therefore it is not possible to evaluate the precision and accuracy of these mappings. Despite the fact that it was not possible to evaluate the inferences made with respect to these attributes, it may be that these mappings are still useful and may produce useful results in a

more extensive dataset. This is an issue that applies to the other new inference modules as well and will be discussed later, when the issue of selecting among the available inference modules is discussed. Practically, the inference modules for these specific attributes are used as ‘reserve’ inference modules; that is, we have included them with caution in DataBait and are providing them with the open source package that we made available, allowing the user of the package to decide whether they will be used or not. The mappings for the following attributes could not be evaluated:

- Demographics / has child
- Politics / ideology
- Health / coffee
- Health / smoking
- Health / alcohol
- Health / energy drinks
- Health / cannabis
- Health / drugs
- Health / medicines
- Health / is pregnant
- Hobbies / cooking
- Hobbies / camping
- Hobbies / video games

It is also important to stress that the URL mapper can also be applied on data coming from other social networks, as in almost any social network the user can include URLs in their posts. For instance, on Flickr, a URL can be posted in the caption or comments of an image.

3.3 Likes mapper

The next inference module that is examined is similar to the URL mapper, but works by analyzing the Facebook pages that the user has liked, rather than the posted URLs. The association between liked Facebook pages and attribute values has been performed using two distinct mappings that were obtained using the procedures described below.

The first mapping was obtained as follows. Initially, we manually selected a set of attribute values for which we expected to find relevant pages. Then, for each of them, a number of relevant keywords were selected and used for querying the Facebook API for relevant pages. For instance, for the attribute “smoking”, the keywords “smoke”, “smoking”, “cigarette”, “cigar” were used. This resulted in an initial pool of pages that are likely related to each attribute. In order to discard irrelevant pages from this initial pool, we hand-labelled a subset of them (around 5-10% of the collected pages) as relevant or irrelevant and used these labelled examples in order to train text-based classifiers that classified the rest of the pages as relevant or irrelevant for each attribute. Different classifiers were tested, among which Naive Bayes and Logistic regression (the implementations offered by Weka) yielded the best results. In all cases, the input was a bag-of-words representation of the name, about section and description of each page as provided by the Graph API. In terms of pre-processing, we removed punctuation and numbers, applied stemming, transformed all words to lower-case and filtered out stop-words using a multi-language stop-word list. 10-fold cross validation was used in order to obtain an estimate of the accuracy of the models and the results can be seen in Table 2.

Table 2. Accuracy of liked pages classifiers

Attribute	Accuracy
Camping	81.4 %
Medicines	72.1 %
Gardening	84.0 %
Is pregnant	79.8 %
Exercising	86.9 %
Smoking	77.8 %
Belief	72.3 %
Alcohol	80.7 %
Orientation	83.5 %
Energy drinks	82.7 %
Drugs	80.5 %
Cannabis	88.5 %
Reading	76.4 %
Health status	72.2 %
BMI class	76.9 %
Supplements	79.3 %
Coffee	88.6 %
Relationship status	84.3 %

Eventually, a set is obtained that comprises pages that have been classified as relevant to some value/attribute, and the ids of these pages are used by the integrated module.

The second way by which pages are mapped to attribute values is by taking into account the categories of pages as provided by the Graph API. For instance, the following is a small example set of mappings from Facebook page categories to attributes, which have been identified and used in the system:

- Library → Hobbies / reading
- Movie Theater → Hobbies / series movies
- Concert Venue → Hobbies / music
- Sport → Hobbies / sports
- Author → Hobbies / reading

In total, 81 such associations have been identified by manually examining the list of Facebook page categories. The complete set of mappings can be found in Annex 2.

Regarding integration, the module is executed every time the set of likes of a user changes. It should also be noted that for this module, just like in the case of the URL mapper, a fixed 0.8 confidence value is used, which grows linearly as the number of pages relevant to some attribute increases. The likes-based mapper was evaluated using the data from the pre-pilot in a manner similar to the previous modules. The results are shown in Table 16 in Annex 1.

Similarly to the URL mapper, the likes mapper performs better than the collection-based classifier for specific attributes in terms of accuracy. For instance, for the attribute 'alcohol', accuracy is 0.8939, whereas for 'political ideology' and 'smoking', accuracy is perfect (1.0).

Another first comment about the results is that the percentage of users for which results are produced is much higher compared to that of the URL mapper. In particular, 14,031 likes out of the 92,128 likes (15.2%) of pilot users were leveraged by the module for classification. This led to the possibility of classifying 378 of the 442 DataBait users (85.5%) to at least one

of the disclosure scoring framework attributes solely based on the likes mapper. Moreover, for quite a few of the attributes, the accuracy is quite high (again, those for which accuracy is higher than 0.5 are marked with bold and were considered for integration).

It should also be noted that the evaluation of some mappings related to specific attributes was not possible. The reasons are the same for which the evaluation of the mappings for specific attributes was not possible for the URL mapper. In the case of the likes mapper, these attributes are the following:

- Relationship / status
- Health / energy drinks
- Health / cannabis
- Health / drugs
- Health / supplements
- Health / medicines
- Health / is pregnant
- Hobbies / video games

Again, the lack of evaluation results does not mean that the relevant mappings cannot be used. This will be discussed later.

3.4 Visual concepts mapper

This module utilizes a number of associations between the visual concepts detected in the images posted by the users and their personal attributes. Some examples of such associations are listed below:

- drunkard→Health / alcohol
- beer hall→Health / alcohol
- disco→Hobbies / music
- keyboardist→Hobbies / music
- temple→Religion / practice
- sport→Hobbies / sports
- basketball→Hobbies / sports
- domestic cat→Hobbies / animals
- dancer→Hobbies / dancing

In total, around 220 such associations were manually identified by searching for ImageNet concepts that are relevant to the considered user attributes. Execution of the module is triggered when there is a change in the images of a user. The confidence of the association is provided by the confidence by which the relevant concepts have been detected (as provided by the visual concept detection module). If a concept has been detected in multiple images with different confidences, the maximum confidence is used.

This module has also been evaluated in a manner similar to the previous modules. The evaluation results are shown in Table 17 in Annex 1. Similarly to the previous modules, the visual concepts mapper can predict specific attributes more accurately – in terms of accuracy - than the collection based classifier. For instance, the accuracy for the attribute ‘alcohol’ is 0.8650 and for the attribute ‘coffee’ it is 0.8095.

It is also clear that the percentage of users for which results are produced is quite higher compared to that of the URL mapper. The accuracy of classification is high only for some attributes. Those are again marked with bold and considered for integration.

Note that with image-based inferences, it is often unclear whether a detected concept concerns the current user or not. For instance, an image posted by a user may show a friend of the user smoking, rather than the user him/herself. This is a clear source of errors; yet it is evident from the results that for specific attributes, accuracy can be sufficient.

Finally, it is worth noting that the visual concepts mapper can also be utilized on data coming from any social network, as in almost any social network the user can upload images.

3.5 Overall evaluation of integrated modules

In the previous subsections, a number of different inference modules were presented and evaluated. For the pilots, all of these inference modules were used; that is, for each attribute, any developed inference module that can handle it accurately enough was integrated. In the case that multiple inference modules produce results for the same attribute, the relevant inference results are fed into the disclosure scoring framework as different 'supports' and are appropriately aggregated by the framework. Here, we examine different options for selecting specific models from the available pool of models, and in particular:

- For each attribute select only the inference module that gives the highest accuracy.
- Select all collection-based classifiers and from the rest, select only those that achieve accuracy higher than 0.5 (those that have been marked in the tables in Annex 1).

These two scenarios are compared to the case that only the collection-based classifiers are used. The results are shown in Table 3.

Table 3. Comparison of different options for selecting inference modules.

	Av. Precision	Av. Recall	Av. Accuracy
Collection-based	0.6964	0,4688	0.7331
Only best acc.	0.7345	0.4034	0,7681
Prec. > 0.5	0.7165	0.4708	0,7383

The two scenarios that are considered as alternative to using only the collection-based classifiers result in significant increase in the performance measures. More particularly, using only the classifier with the highest accuracy for each attribute results in an increase in both precision and accuracy, with a drop in average recall. Using all collection-based classifiers and those other classifiers that have accuracy higher than 0.5 results in a measureable increase in all performance measures. These results, particularly in the case that only the classifier with the best accuracy is used, attest to the value of the newly developed inference modules. In the final version of the system, the decision is to use only the best-performing classifier for each attribute, as this results in the highest increase in accuracy and it is preferred to show more accurate predictions, at the cost of showing fewer inference results.

There are some further remarks that can be made about the developed inference modules and their evaluation:

- Different modules examine different types of data: posts, images, likes. The attributes of different users may be manifested through different data, e.g. some users may post more photos than like pages or vice versa. Some attributes may be expressed by a single like and may not be manifested at all by any image or post. This justifies the decision to examine different classifiers that consider different types of data.
- It is important to take into account both information that is almost explicitly declared (as in the case of URLs, likes, visual concepts) as well as information that can be inferred in a more indirect way (as in the case of the collection-based classifiers).
- As mentioned briefly before, it should be kept in mind that there is an upper limit on the accuracy of the considered classifiers. The main reason is that some user attributes may not be reflected in any of the user's OSN data, either in a direct or indirect manner.

3.6 Investigating the impact of MyPersonality

In this set of experiments, we investigate whether the accuracy of the collection-based classifiers can be further improved by leveraging external data (besides those contained in the pre-pilot dataset, hereby referred to as the USEMP dataset) for training. To this end, the MyPersonality⁵ dataset (Kosinski et al., 2015) is employed. MyPersonality contains a wide variety of data about Facebook users, such as records of users' likes and the values of several personal user attributes, including four attributes that we make inferences for in DataBait using training data from the USEMP pilots: a) gender, b) political ideology, c) sexuality, d) relationship status. Given that in previous experiments (as described in D6.4), we found that likes compare favorably to other features (e.g. like categories, visual concepts, etc.) in terms of predictive accuracy, MyPersonality could potentially make an effective alternative source of training data for the collection-based classifiers.

To evaluate the merits of pooling data from MyPersonality (along with data from the USEMP dataset) to make inferences for the users of DataBait, we conduct the following experiment. On each attribute, we compare the cross-validation performance in terms of Area under ROC (AUC) of using only examples from USEMP for training (results comparable to those obtained in D6.4) to the performance of adding a number of examples from MyPersonality to the USEMP examples comprising the training set of each iteration in a k -fold cross-validation. Concretely, if we denote as $D_U^1, D_U^2, \dots, D_U^k$ the partitioning of D_U (the USEMP dataset) into k folds, in each iteration i of a typical k -fold cross-validation D_U^i is used as the test set and $D_U \setminus D_U^i$ as the training set. In our modified k -fold cross-validation procedure, D_U^i is again used as the test in each iteration but instead of $D_U \setminus D_U^i$ we use $D_U \setminus D_U^i \cup D_M^S$ as the training set, where D_M^S is a subset of the MyPersonality dataset.

Before applying the above evaluation procedure, two issues had to be addressed. First, as shown in Table 4, there is a mismatch between the classes of some attributes in USEMP and MyPersonality. Thus, in order to be able to combine examples from the two datasets into a single training set we aligned their output spaces by making the following transformations:

⁵ <http://mypersonality.org>

- Political ideology: right→conservative, left→liberal
- Sexuality: {straight man, straight woman}→heterosexual, {gay, lesbian}→homo/bi
- Relationship status: in_relationship→ removed

These transformations resulted in the final classes shown in the rightmost column of Table 4.

Table 4. Correspondence between USEMP and MyPersonality attribute classes.

	USEMP classes	MyPersonality classes	Final classes
gender	male, female	male, female	male, female
political stance	right, left	conservative, liberal	right, left
sexuality	heterosexual, homo/bi	straight man, straight woman, gay, lesbian	heterosexual, homo/bi
relationship status	single, married, in_relationship	married, single	married, single

Second, likes are anonymized in MyPersonality, preventing the creation of a common feature representation, i.e. a common like dictionary, between USEMP and MyPersonality. To overcome this problem, we used a subset of the dataset that contained de-anonymized like information. Fortunately, despite containing a small fraction (2.5%) of the total number of user-like dyads (about 1.8 billion) of the full version of the dataset, we still found several thousands of users in MyPersonality, which had at least one like in common with a user of USEMP (this is a required condition for cross-dataset learning to be possible). Table 5 shows the numbers of examples for each attribute (and class) in MyPersonality and USEMP datasets, after removing MyPersonality users with no common likes with a USEMP user and after applying the output space transformations mentioned above.

Table 5. Number of examples for different attributes and classes in MyPersonality and USEMP.

		# examples MyPersonality	# examples USEMP
Gender	Male	69,926	100
	female	106,537	62
political stance	Right	1,570	8
	Left	1,944	21
sexuality	heterosexual	46,081	141
	homo/bi	1,895	21
relationship status	married	69,075	63
	Single	20,866	31

Due to the small size of the USEMP dataset, before applying the modified k -fold cross-validation procedure described above we first analyzed the stability of the typical k -fold cross-validation estimates (i.e. when only examples from USEMP are used).

Figure 7 shows the evolution of the average AUC performance after adding the result of each iteration in a repeated ($n = 100$) 10-fold cross-validation (1000 iterations in total). In this, as well as subsequent experiments, we use LibLinear’s (Fan, 2008) L2-regularized logistic regression as the classification algorithm as it was found comparable or better than other state-of-the-art classifiers in D6.4. We see that for relationship status, sexuality and gender the estimates stabilize after about 100 iterations (i.e. 10 repetitions of 10-fold cross-validation) while for political stance the estimates become relatively stable after about 500 iterations (i.e. 50 repetitions of 10-fold cross-validation)⁶. Based on these results, we opted for applying 50 repetitions of 10-fold cross-validation to ensure reliable performance estimates for all attributes.

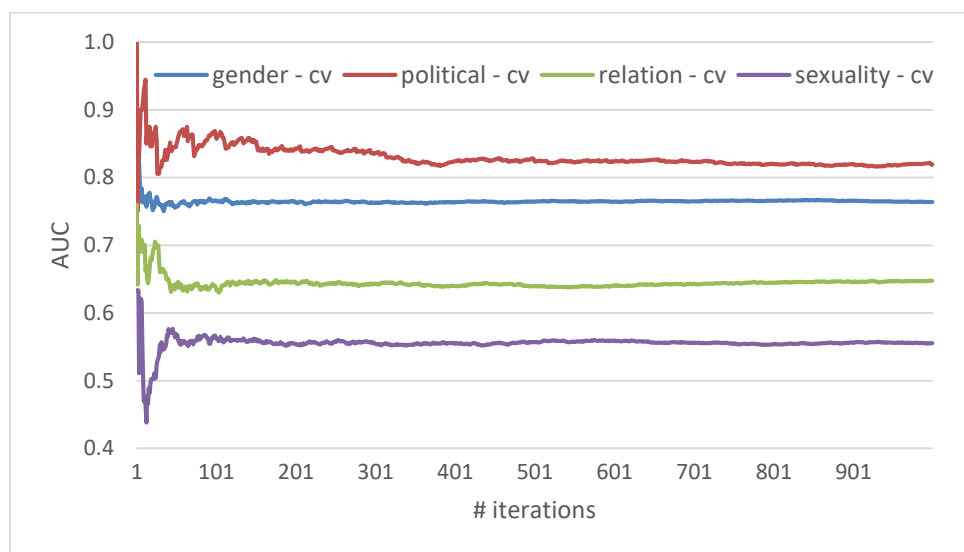


Figure 7. Stability of k -fold cross-validation performance estimates

Figure 8 plots the performance on USEMP as a function of $|D_M^S|$, i.e. the number of examples from MyPersonality used in the modified k -fold cross-validation procedure ($|D_M^S| \in \{0, 50, 100, 200, 500, 1000, 2000\}$). Each point represents the average of 5 runs of 50x10-fold cross-validation where in each run a different random sample (without replacement) was taken from MyPersonality. This was done to remove any performance artifacts resulting from the randomness of the sampling procedure (especially for small sample sizes).

We observe that the addition of training examples from MyPersonality leads to a steady performance increase for gender and sexuality, while the opposite is observed for political stance and relationship status. In the case of political stance, the decrease in performance is attributed to the inappropriateness of mapping MyPersonality’s conservative class to USEMP’s right class and MyPersonality’s liberal class to USEMP’s left class. In the case of relationship status, on the other hand, the decrease is probably related to the differences in behavioral patterns (Facebook like activity) between USEMP’s users and MyPersonality’s

⁶This larger instability of political stance is attributed to the very small number of examples used for evaluation in each fold (~3 examples).

users, perhaps due to the specificity of USEMP’s user population (mostly Belgian and Swedish) compared to the more varied Facebook user population that MyPersonality represents. Such differences seem to have a much smaller impact on the prediction of gender and sexuality. Hence, our study demonstrated that using external data for training our models would result in noticeable improvements in the accuracy of our classifiers.

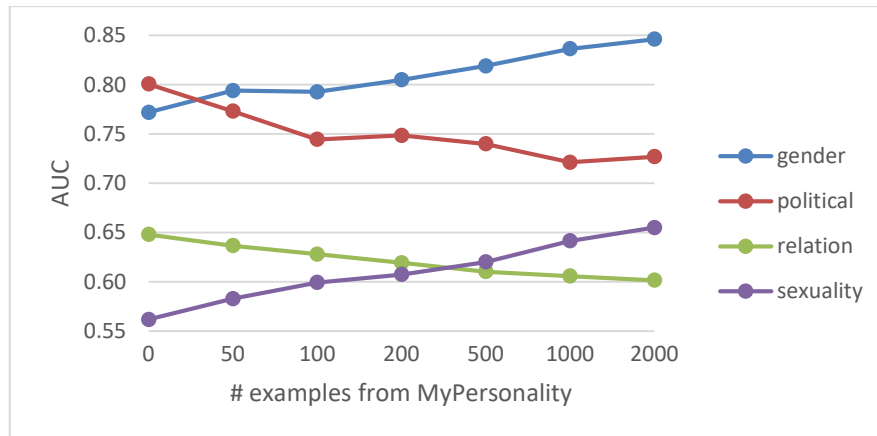


Figure 8. Performance on USEMP as a function of number of training examples from MyPersonality.

3.7 Improvements using data from the pilots

So far, inferences made by the collection-based classifiers were based on models trained on data coming from the pre-pilots. These models were selected based on a comprehensive set of experiments that examined the performance of various types of predictive features and classification algorithms on the prediction of the considered user attributes (details can be found in D6.4). Those experiments demonstrated that several attributes could be predicted with high accuracy based on the pre-pilot data (e.g. political ideology), while most attributes could be predicted significantly more accurately than with random guessing. Provided that these results were obtained using a rather limited set of training data, it is expected that even more accurate predictions can be obtained if more training data is available. However, the results of the previous section demonstrated that the use of additional training data does not always lead to better performance when this data represent a different population of users. To further investigate this issue, and in attempt to further improve the accuracy of the collection-based classifiers, in this section we examine how the use of additional training data from the same population as the one represented by the pre-pilot data, affects the performance of the collection-based models. To this end, we employ the data obtained during the pilots⁷ and repeat the experiments presented in D6.4, contrasting the performance of using only the pre-pilot data (170 data points), to the performance of using the union of the pre-pilot and the pilot data for training (204 data points).

In particular, we conducted experiments using seven types of predictive features (i.e. all types of features used in D6.4 except for the LDA-based ones) and LibLinear’s (Fan, 2008) L2-regularized logistic regression as the classification algorithm, as it provided better results

⁷ We actually use a subset of the pilot data because at the time of writing only the data collected by the Swedish pilot (LTU) was available. The full set of pilot data will be analyzed when available.

(on average) than other state-of-the-art classifiers on this task (see D6.4). In light of the stability issues reported in the previous section with respect to the performance estimates, and in contrast to D6.4 where the evaluation was carried out using a single cross-validation run, the evaluation here is carried out using repeated (10 times) 10-fold cross-validation and accuracy is measured in terms of AUC. Figure 9 shows the average AUC performance (across all target attributes) obtained using each type of features, when only the pre-pilot data is used for training versus when the union of data is used. We see that the performance improves in all cases, suggesting that even a small data augmentation is beneficial when the data come from the same population. When the best performance per attribute is considered (i.e. using any type of features), the average AUC increases from 0.675 (when only pre-pilot data is used) to 0.691 (when all data is used).

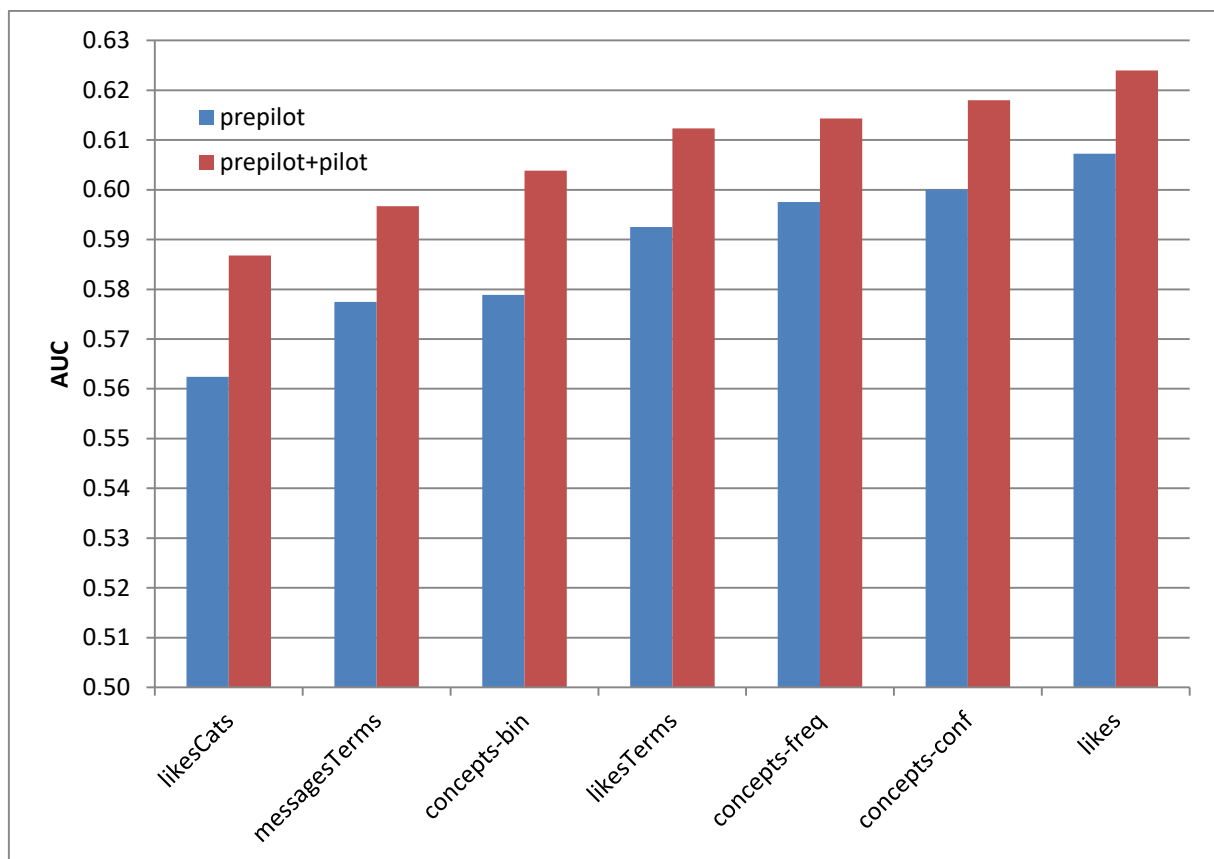


Figure 9: Comparison of classifiers trained on pre-pilot data versus classifiers trained on pre-pilot+pilot data using different types of features.

3.8 Improvements using class rebalancing

Taking a critical view over the performance of the collection-based classifiers, we noticed a potential shortcoming, resulting from the fact that most classification algorithms are tailored to the minimization of the error rate (the percentage of incorrect predictions), ignoring how errors are distributed across different classes. In some learning settings, this focus on the minimization of the total number of classification errors can result in highly skewed error distributions, i.e. some classes being predicted with high accuracy while others are being predicted poorly. A common cause for uneven error distributions is the well-known class imbalance setting (Japkowicz, 2002) which is quite prevalent among the user attributes that

we want to predict (see Table 6). Class imbalance refers to classification problems where different classes have disproportionate priors, i.e. some classes (referred to as majority classes) are significantly more common than others (referred to as minority classes). Under this setting, typical classification algorithms tend to generate models that accumulate their errors on the minority classes because this usually results in lower error rates. As an illustrative example, consider a binary classification problem where 99% of the training examples belong to class A and only 1% of the training examples belong to class B. Assuming that the test set will have a similar class distribution, a naïve classifier that always predicts the majority class will have a very low error rate (1%) but while the TPR (True Positive Rate or Recall) for class A will be 100%, the TPR for class B will be 0%. As described earlier (please see Table 14 in Annex 1), this is the case for specific attributes that the collection-based classifier handles. For instance, consider the prediction of the attribute 'sexual orientation', where out of 168 users that revealed their sexual orientation, 147 users identified themselves as heterosexuals and only 21 as homosexual or bisexual. In turn, the classification model that was built from this dataset, learned to classify all users as heterosexuals as this led to relatively small misclassification error. Obviously, this behavior is problematic because minority classes are usually the most sensitive ones and incorrectly predicting that a minority (sensitive) class user belongs to the majority (insensitive) class is an error that is clearly associated to a higher risk of disclosure, compared to the opposite type of error. Based on this observation, we worked towards the development of methods that attempt to balance the performance (or more accurately TPR) between majority and minority classes. In the following sections we describe and evaluate these methods, after shortly discussing related work and providing some necessary background.

3.8.1 Previous work

Learning under class imbalance has been a topic of active research in machine learning for almost two decades and thus, it is not surprising that a recent survey on the topic (Branco et al., 2015) cites roughly 200 papers. A coarse categorization of the various approaches that have been developed to address this problem is the following:

- *Data manipulation* approaches: This category includes techniques that try to balance the data distribution, usually by means of: a) under-sampling the majority class, e.g. (Drummond & Holte, 2003), b) over-sampling the minority class, e.g. (Japkowicz & Stephen, 2002) and c) generating synthetic minority class examples, e.g. SMOTE (Chawla et al., 2002) and variants.
- *Algorithm adaptation* approaches: This category includes methods that adjust existing learning algorithms so that they can better handle problems with class imbalance. The adaptation is often performed by introducing the notion of prediction cost and associating misclassification of minority class examples with a higher cost. Examples include cost-sensitive adaptations of decision trees (Maloof, 2003), neural networks (Zhou & Liu, 2006), SVMs (Akbari et al., 2004), boosting (Nikolaou, et al., 2016), etc.
- *Prediction post-processing* approaches: This category includes methods that are applied on the outputs of classifiers that are able to predict confidence scores, and attempt to transform these scores into hard predictions in a way that takes class imbalance into account. One way to achieve that is by pushing the decision threshold towards the minority class (Weiss, 2004).

Unfortunately, despite the active research on this topic, the research community has not yet reached a conclusive decision with respect to which category of methods works better and performance is often dependent on the particular problem. Nevertheless, cost-sensitive learning approaches stand out by their wide-adoption and effectiveness (Weiss, 2004). Based on that, as well as the observation (Maloof, 2003) that sampling, modifying the classification threshold, and adjusting the cost matrix, all produce classifiers with similar ROC curves, we focused on approaches that tackle class-imbalance from a cost-sensitive learning perspective. These approaches are described in the following section.

3.8.2 Background and methodology

Before describing our proposed approaches, we first provide some necessary background on cost-sensitive learning. Cost-sensitive learning refers to techniques for optimal decision-making when different classification errors incur different costs (Elkan, 2001). Typically, prediction costs are specified via a square matrix C (known as the *cost matrix*) where each entry $C(i, j)$ corresponds to the cost of predicting class i when the true class is j . Thus, a cost matrix for a multi-class problem with three classes $\{X, Y, Z\}$ looks like this:

		actual		
		X	Y	Z
predicted	X	$C(X, X)$	$C(X, Y)$	$C(X, Z)$
	Y	$C(Y, X)$	$C(Y, Y)$	$C(Y, Z)$
	Z	$C(Z, X)$	$C(Z, Y)$	$C(Z, Z)$

Given such a specification of prediction costs, an example x should then be classified to the class $i \in \{1, \dots, N\}$ that leads to the lowest expected cost:

$$\arg \min_{i \in \{1, \dots, N\}} \sum_{j \in \{1, \dots, N\}} P(j|x) C(i, j) \quad (1)$$

instead of the most probable one:

$$\arg \max_{i \in \{1, \dots, N\}} P(i|x) \quad (2)$$

It is easy to show that when the costs of correct predictions are zero, i.e. $C(i, i) = 0 \forall i \in \{1, \dots, N\}$, and all incorrect predictions have the same cost, i.e. $C(i, j) = a \forall i, j \in \{1, \dots, N\}, i \neq j$, equation (1) coincides with equation (2). Thus, traditional, cost-insensitive learning can be thought of as a special case of cost-sensitive learning.

In the past, several cost-sensitive learning methods have been proposed, e.g. (Domingos, 1999), (Elkan, 2001), (Zadrony et al., 2003), (Tu & Lin, 2010). Here we shortly review two of the most well-known ones, which are later used in the experiments:

- The *plug-in rule* (Elkan, 2001): It consists of employing a typical cost-insensitive learner to estimate the posterior class probabilities and then directly applying equation (1) to output the class that minimizes the expected cost instead of the most probable one. (Elkan, 2001) showed that this simple approach should be preferred over approaches that change the proportions of training examples of different classes.
- *MetaCost* (Domingos, 1999): MetaCost first uses bagging of decision trees to obtain reliable probability estimates for the training examples, then relabels them according to (1), and finally uses the relabeled examples to train a cost-insensitive classifier.

As already discussed, a main implication of learning under class imbalance is that standard classifiers tend to generate models with very low TPR for the underrepresented classes because this behavior gives rise to a lower overall number of classification errors; and this behavior is a result of most classifiers implicitly assuming that all errors have equal costs. Thus, a straightforward way to increase the TPR of the minority classes (thus achieving a more balanced performance) is to come up with an appropriate cost-matrix (one where errors on minority classes are costlier), and then employ any of the existing cost-sensitive learning techniques to solve the problem. A simple method to construct such a cost matrix is by making the cost of failing to correctly classify an instance, inversely proportional to the frequency of the instance's actual class.

More formally, if we denote as $f_i, i \in \{1, \dots, N\}$ the frequency of each class, then we set $C(i, j) = \frac{1}{f_i} \forall i, j \in \{1, \dots, N\}, i \neq j$. We call this approach *inverse frequency*. Other choices include setting $C(i, j) = \frac{1}{m_i} \forall i, j \in \{1, \dots, N\}, i \neq j$, where m_i is a measure of performance on class i (e.g. TPR) as estimated on the training set (by e.g. internal cross-validation) or even using techniques that learn a cost-matrix that optimizes a target performance measure. In preliminary experiments, we found that the simple $C(i, j) = \frac{1}{f_i} \forall i, j \in \{1, \dots, N\}, i \neq j$ strategy leads to comparable or better results than more sophisticated techniques on the USEMP dataset (probably due to the fact that its small size leads other methods to overfitting the training set) and, therefore, we do not consider other cost matrix construction methods here.

3.8.3 Generalizing the plug-in rule

As outlined above, the *plug-inrule* (Elkan, 2001) employs equation (1) to perform cost-sensitive learning. In our analysis, we try to skew posterior distributions according to costs $C(i, j) = C(i) \forall i, j \in \{1, \dots, N\}, i \neq j$ and $C(i, i) = 0 \forall i \in \{1, \dots, N\}$. Under this assumption, equation (1) yields the classification rule:

$$\arg \min_{i \in \{1, \dots, N\}} (1 - P(i|x))C(i) \quad (3)$$

We can observe that the end-result is essentially a simple class-dependent transformation that rebalances the posterior distribution. Thus, we can present a more generalized form:

$$\arg \min_{i \in \{1, \dots, N\}} t(C(i), 1) - t(C(i), P(i|x)) \quad (4)$$

for a suitable confidence function $t(c, w)$ that correlates the misclassification costs c with posterior distribution scores w . Essentially, equation (4) tries to minimize the risk caused by low confidence. The confidence function $t(c, w)$ should be increasing for both its variables, as we would like to boost higher misclassification costs, as well as retain inner-class ordering of confidence levels. Equation (4) is indeed a generalization of our plug-in rule, as presented in equation (3), for $t(c, w) = c \cdot w$.

We can now observe that, as mean classification scores $E_x[P(i|x)]$ increase, respective posterior distributions become skewed towards classes i . In fact, posterior class distributions are approximated by mean classification scores. However, we have previously shown that posterior class distributions are also skewed towards majority classes in imbalanced settings. Hence, we can deduce that mean classification scores increase proportionally to class

frequencies f_i and class performance gains ΔTPr_i . In fact, we can prove that, for normalized (i.e. summing to 1) posteriors $P(i|x)$:

$$\Delta E_x[P(i|x)] = f_i \cdot \Delta TPr_i \quad (5)$$

Now, the outlined risk function can be considered a rebalance towards the un-normalized scores $t(c, w) - t(c, 1) + t(\sup c, 1)$. Therefore, as long as higher costs are assigned for lower class frequencies (e.g. $c = \frac{1}{f}$), subsequent normalized minority class scores increase and we can thus infer that equation (4) indeed improves minority class performance. Unfortunately, increasing normalized scores for minority classes also decreases normalized scores for majority ones, yielding a similar loss for them. Hence, we can understand that equation (4) improves minority class performance at the cost of majority class performance. Since majority classes are less impacted by this process for heavily imbalanced training, our non-conformist outlook indicates a desirable end-result.

To more accurately quantify this desired end-result, we employ the notions of mean TPr and *fairness*. Mean TPr is the (non-weighted) average between all class' performance. Therefore, it gives considerably larger importance to minority classes, as opposed to weighted TPr that leverages performance according to class priors. Following the previous line of reasoning, an increase of mean TPr indicates that minority class performance gains are larger than majority class performance losses, netting a positive tradeoff. Mean TPr is a performance-oriented metric and, as such, does not adequately represent the balance between different class performance. Nevertheless, high values can only be achieved if performance is both high and balanced.

On the other hand, *fairness* solely measures the balance between different classes. It is defined by the *US Equal Employment Opportunity Commition* for binary problems as the ratio of performance between classes. For multi-class problems, we expand this definition to a lower bound of the weighted mean of one-vs-all individual class *fairness* that yields intuitive results while being contained in the range $[0,1]$;

$$Fairness = \frac{1}{\sum_i \sum_{j \neq i} f_i f_j} \sum_i \sum_{j \neq i} f_i f_j \left/ \frac{TPr_i}{TPr_j} + \frac{TPr_j}{TPr_i} \right. \quad (6)$$

As previously outlined, the confidence function $t(c, w)$ needs only be increasing for c and w . However, we may also need a way to parameterize it, in order to retain control over the rebalance process. An easy way to do this this is tweaking the confidence function to:

$$t_a(c, w) = (1 - a)w + at(c, w) \quad (7)$$

This new function retains all properties for $0 < a \leq 1$. For any confidence function $t(c, w)$, we can prove that it improves both mean TPr and *fairness* for a sufficiently small constant a , as long as results are even marginally imbalanced. In the case of our datasets though, performance distributions are so imbalanced that we can safely select even $a = 1$.

The presented rebalance method is indeed a generalization to the plug-in rule. The generalization directly improves fairness between classes, in addition to optimizing the plug-in rule for frequency-based classification costs. The more general framework allows us to perform posterior probability skewing in a *per-sample* basis, even under assumptions that do not necessarily adhere to a cost-sensitive approach. To showcase this ability, we also introduce a heuristic confidence $t(c, w) = w^{1/c}$ ($= w^f$ for $c = \frac{1}{f}$).

Furthermore, within the developed framework we are now able to parameterize rebalancing according to classification certainty. In particular we can select the parameterization quantity for the rebalancing of each sample in equation (7) to be the normalized entropy $a = -\sum_i P(i|x) \log_N P(i|x)$. As entropy is an indicator of uncertainty, now $a = 1$ indicates an uncertain classification and $a = 0$ a completely certain one. Therefore, the parameterized process now lessens the impact of rebalancing if classification is estimated to be correct. Effectively, this process causes the classification to lean towards identifying minority classes under uncertainty (but not otherwise).

3.8.4 Experiments

The following experiments assess the effectiveness of the methods described above on balancing the performance across different classes of binary and multi-class classification problems. In particular, we use the arithmetic mean of the per-class TPRs (amTPR) and the fairness measure described in the previous section. Both measures promote models with a balanced performance across classes. However, in contrast to fairness which focuses exclusively on balance, amTPR accounts for both balance and absolute performance. We focus on imbalanced classification problems and especially on problems where the minority class (or at least one of the minority classes, if more than one exist) is associated with a sensitive piece of information about the user. To this end, we use the USEMP and MyPersonality datasets that both involve classification targets with the aforementioned characteristics and, at the same time, represent two diverse imbalanced learning settings: one where training data is scarce (USEMP) and one where there are plenty of training examples, even for the minority class (MyPersonality). Table 6 shows the target variables that were selected from each dataset, as well as the different classes of each target and the number of training examples from each class.

In both datasets we carried out experiments using two different types of feature: a) likes-based and b) topic-based. Likes-based features correspond to a binary vector where each variable indicates the presence or absence of a like in the set of likes of the user. A different vocabulary is constructed for each dataset, which consists of all likes that appear in the sets of likes of at least two users of that dataset. This resulted in a vocabulary of 3,622 likes for all target attributes of USEMP and vocabularies of 193,934 and 731,146 likes for the targets religion and sexual orientation, respectively, of the MyPersonality dataset (in contrast to USEMP, each target attribute involves different users in MyPersonality). With respect to topic-based features, in the case of USEMP these correspond to the LDA-t (t=30) features described in D6.4, i.e. LDA topics are extracted from the combination of the textual content in the user's posts and in the description, title and about sections of the user's likes. In the case of MyPersonality, topic-based features are again computed with LDA and topics are extracted by treating each user as a document containing words from MyPersonality's like dictionary. Note that likes and LDA-t were the two best-performing features (in terms of AUC) of those tested in D6.4 and in (Spyromitros-Xioufis et al., 2016a).

On each target of each of the two datasets and for each type of feature, we evaluate the ability of the following classification performance balancing methods:

- Cost-sensitive balancing using *inverse frequency* and the *plugin-rule* (CSB-p),
- Cost-sensitive balancing using *inverse frequency* and *MetaCost* (CSB-m),
- Entropy balancing using the heuristic *exponential rule* (EB-e)

Table 6. Classification targets used in experiments along with different classes and number of examples from each class. There are two versions of the targets “Sexual orientation” and “Religion” in MyPersonality, one for each type of predictive features that we

Dataset	Target	Class	# Examples
USEMP	Sexual orientation	Heterosexual	147
		Homo/bi	21
	Health status	Very good/Excellent	97
		Fair/good	63
		Poor	7
	Religion	Atheism	63
		Catholic	34
		Agnosticism	21
		Protestant	15
		Other	13
		Buddhism	5
		Islam	2
		Judaism	1
	Religious practice	No	121
		Yes	22
	Cannabis	No	154
		Yes	16
Alcohol	No	135	
	Yes	25	
Smoking	No	145	
	Yes	25	
MyPersonality	Sexual orientation (LDA)	Straight woman	17,366
		Straight man	13,501
		Gay	684
		Lesbian	520
	Religion (LDA)	Christian	8,820
		Muslim	383
	Sexual orientation (Likes)	Heterosexual	55,826
		Homo/bi	2,266
	Religion (Likes)	Christian	13,378
		Muslim	743

All three methods can be parametrized with any standard classifier able to output a probability distribution (or confidence scores that can be transformed to a probability distribution) for each instance. In this set of experiments, we select two state-of-the-art probabilistic classifiers (that are however tailored to the minimization of misclassification error) to parametrize the proposed error rebalancing methods with: a) *L2-regularized logistic regression* (the LibLinear implementation) and b) *random forest* (with 10 trees). Note that L2-regularized logistic regression and random forest were the two best-performing classifiers (in terms of AUC) of those tested in D6.4 and in (Spyromitros-Xioufis et al., 2016a).

Performance is evaluated using 2-fold cross-validation on MyPersonality, while on USEMP we perform repeated (10 times) 10-fold cross-validation to obtain reliable performance estimates. Tables 7-10 show the performance obtained by each error balancing method (CPR, CSB-p and CSB-m) as well as the performance obtained without balancing (No), on each target of USEMP and MyPersonality in terms of amTPR (left-side) and fairness (right-side), for the four distinct combinations of base classifier and predictive features that we tested, i.e. logistic regression with topics-based features (Table 7), logistic regression with

likes-based features (Table 8), random forest with topics-based features (Table 9) and random forest with likes-based features (Table 10). The last row of each table reports the average performance of each method across all targets of USEMP and MyPersonality. Also note that due to the large dimensionality of the likes-based features on the MyPersonality dataset, only logistic regression could be used as base classifier.

Table 7. Performance of error balancing methods with topics-based features and logistic regression.

Dataset	Target	amTPR				Fairness			
		No	EB-e	CSB-p	CSB-m	No	EB-e	CSB-p	CSB-m
USEMP	sexual or.	0.500	0.498	0.585	0.521	0.000	0.000	0.710	0.549
	health status	0.332	0.258	0.357	0.340	0.089	0.198	0.732	0.514
	religion	0.129	0.030	0.200	0.185	0.062	0.001	0.081	0.132
	rel. practice	0.500	0.491	0.508	0.473	0.000	0.000	0.634	0.475
	cannabis	0.500	0.503	0.537	0.523	0.000	0.009	0.609	0.572
	alcohol	0.500	0.486	0.447	0.458	0.000	0.000	0.642	0.589
	smoking	0.500	0.492	0.510	0.475	0.000	0.000	0.677	0.649
MyPerson	sexual or.	0.415	0.465	0.452	0.489	0.828	0.878	0.876	0.902
	religion	0.504	0.531	0.646	0.717	0.015	0.123	0.532	0.727
Average		0.431	0.417	0.471	0.465	0.110	0.134	0.610	0.568

Table 8. Performance of error balancing methods with likes-based features and logistic regression

Dataset	Target	amTPR				fairness			
		No	EB-e	CSB-p	CSB-m	No	EB-e	CSB-p	CSB-m
USEMP	sexual or.	0.500	0.500	0.481	0.503	0.000	0.000	0.604	0.024
	health status	0.333	0.342	0.392	0.351	0.000	0.057	0.743	0.320
	religion	0.125	0.024	0.168	0.148	0.001	0.000	0.142	0.150
	rel. practice	0.500	0.500	0.489	0.513	0.000	0.000	0.646	0.170
	cannabis	0.500	0.500	0.511	0.497	0.000	0.000	0.561	0.031
	alcohol	0.500	0.500	0.498	0.499	0.000	0.000	0.723	0.583
	smoking	0.500	0.500	0.473	0.469	0.000	0.000	0.673	0.209
MyPerson	sexual or.	0.523	0.580	0.679	0.647	0.093	0.314	0.661	0.552
	Religion	0.546	0.650	0.825	0.751	0.182	0.551	0.929	0.806
Average		0.447	0.455	0.502	0.486	0.031	0.102	0.631	0.316

Table 9. Performance of error balancing methods with topics-based features and random forest.

Dataset	Target	amTPR				fairness			
		No	EB-e	CSB-p	CSB-m	No	EB-e	CSB-p	CSB-m
USEMP	sexual or.	0.498	0.491	0.527	0.520	0.035	0.245	0.585	0.255
	health status	0.336	0.316	0.333	0.333	0.652	0.647	0.753	0.697
	religion	0.114	0.110	0.148	0.138	0.176	0.209	0.239	0.244
	rel. practice	0.495	0.517	0.546	0.514	0.025	0.395	0.676	0.202
	cannabis	0.512	0.556	0.533	0.535	0.038	0.325	0.614	0.172
	alcohol	0.494	0.505	0.505	0.518	0.145	0.351	0.637	0.533
	smoking	0.491	0.493	0.500	0.480	0.000	0.365	0.664	0.108
MyPerson	sexual or.	0.432	0.514	0.533	0.423	0.867	0.951	0.954	0.880
	religion	0.598	0.754	0.803	0.677	0.376	0.827	0.991	0.639
Average		0.441	0.473	0.492	0.460	0.257	0.480	0.679	0.414

Table 10. Performance of error balancing methods with likes-based features and random forest.

Dataset	Target	<i>amTPR</i>				<i>fairness</i>			
		No	EB-e	CSB-p	CSB-m	No	EB-e	CSB-p	CSB-m
USEMP	sexual or.	0.500	0.505	0.508	0.515	0.009	0.103	0.309	0.073
	health status	0.338	0.253	0.291	0.322	0.359	0.342	0.597	0.576
	religion	0.150	0.149	0.147	0.146	0.198	0.245	0.160	0.213
	rel. practice	0.503	0.517	0.477	0.537	0.055	0.488	0.487	0.692
	cannabis	0.499	0.497	0.469	0.436	0.000	0.128	0.539	0.385
	alcohol	0.500	0.511	0.514	0.529	0.022	0.259	0.617	0.465
	smoking	0.498	0.489	0.455	0.482	0.000	0.128	0.562	0.416
Average		0.427	0.417	0.409	0.424	0.092	0.242	0.467	0.403

Looking at the results, we notice that on *fairness*, CSB-p is consistently better than the other error balancing methods as well as no balancing, while on *amTPR*, CSB-p is outperformed only in one case (likes-based features and random forest as the base classifier). When we consider the best performance (for any base classifier and feature) achieved on each target by each method (Table 11), we again see that CSB-p obtains the best average performance followed by CSB-m and EB-e for both measures. In terms of *amTPR*, CSB-p achieves an average of 0.532 which is 16% better than that of no balancing, while in terms of *fairness*, CSB-p achieves an average of 0.704 which is 267% better than that of no balancing. Target-wise, the best performance is always achieved by an error balancing method for both *amTPR* and *fairness*. On both *amTPR* and *fairness*, CSB-p wins on 7 out of 9 targets, followed by CSB-m and EB-e that both have one win each.

Table 11. Maximum performance per target for any base classifier and feature.

Dataset	Target	<i>amTPR</i>				<i>fairness</i>			
		No	EB-e	CSB-p	CSB-m	No	EB-e	CSB-p	CSB-m
USEMP	sexual or.	0.500	0.505	0.585	0.521	0.035	0.245	0.710	0.549
	health status	0.338	0.342	0.392	0.351	0.652	0.647	0.753	0.697
	religion	0.150	0.149	0.200	0.185	0.198	0.245	0.239	0.244
	rel. practice	0.503	0.517	0.546	0.537	0.055	0.488	0.676	0.692
	cannabis	0.512	0.556	0.537	0.535	0.038	0.325	0.614	0.572
	alcohol	0.500	0.511	0.514	0.529	0.145	0.351	0.723	0.589
	smoking	0.500	0.500	0.510	0.482	0.000	0.365	0.677	0.649
MyPerson	sexual or.	0.523	0.580	0.679	0.647	0.867	0.951	0.954	0.902
	religion	0.598	0.754	0.825	0.751	0.376	0.827	0.991	0.806
Average		0.458	0.491	0.532	0.504	0.263	0.494	0.704	0.633

Based on these results, it is clear that all the proposed error balancing methods and particularly CSB-p and CSB-m, can effectively improve the performance in terms of balance-aware performance measures such as *amTPR* and *fairness*.

4 User Perceptions on Predictability of Disclosed Personal Information

The learning experiments presented in D6.4 and earlier in chapter 3 provided us with an idea of the predictability of different types of information. Nevertheless, the users that participated in the pre-pilot, apart from providing us access to their OSN data and answering questions about their personal attributes, also answered questions related to their perceptions about the predictability and the sensitivity of different types of information. Feedback about the perceived predictability was provided by the users with a yes/no answer to the question: “Can this particular type of information be inferred based on your OSN data?”, and feedback about the sensitivity of different types of information was provided in a scale from 1 to 7 with higher values denoting higher sensitivity. In the following, we consider the relationship between the predictability of the different types of personal information and the users’ perception about them. The results that will be presented next have also been presented in the latest Internet Science conference (Spyromitros-Xioufis, 2016a).

D6.4 included a table that compared the actual predictability of the different dimensions – as expressed by the wAUC of the collection-based classifiers - to the perceived predictability of the different dimensions – as indicated by the users’ responses to the corresponding question. This is repeated here in Table 12; it is however extended in order to also take into account the results of the experiments presented in (Kosinski, 2013). Kosinski’s experiments do not consider all dimensions that our experiments do, however, they can provide an additional source of information for estimating the predictability of the dimensions that we include in our comparison. It is noted that users perceive ‘Demographics’ as the dimension that is most predictable (88.4%), and indeed it was found through our study that it is the dimension that can be predicted most accurately. Our conclusions also appear to mostly match those of (Kosinski, 2013). In particular, ‘Demographics’ and ‘Political views’ are identified as the most predictable dimensions in both studies and the ranking of the remaining dimensions is quite consistent (except for Religious views).

Table 12. Comparison of perceived and actual predictability according to our experiments and the experiments of (Kosinski, 2013)

Rank	Perceived predictability	Actual predictability according to our experiments	Actual predictability according to (Kosinski, 2013)
1	Demographics	Demographics	Demographics
2	Location	Political views	Political views
3	Relationship status and living condition	Sexual orientation	Religious views
4	Sexual orientation	Employment / income	Sexual orientation
5	Consumer profile	Consumer profile	Health status
6	Political views	Relationship status and living condition	Relationship status and living condition
7	Personality traits	Religious views	
8	Religious views	Health status	
9	Employment / income	Personality traits	
10	Health status		

We now proceed to take into account how the perceived sensitivity of dimensions correlates with the actual and perceived predictability of dimensions. Note that actual predictability is measured in terms of the wAUC achieved by the collection-based classifiers that have been presented in Chapter 3 of this document and in D6.4 and that perceived predictability is measured by averaging the responses of the users and mapping them in the interval from 0 to 1. Figure 10 shows the relevant results. Let us first focus on the relationship between perceived predictability and sensitivity. With the exception of the 'Religious views' and 'Relationships' dimensions, there appears to be a clear linear relationship between sensitivity and perceived predictability. That is, the more sensitive some dimension is perceived by users, the less predictable it is considered. For instance, 'Demographics', the dimension that is perceived as the easiest to predict (and is actually the most predictable), is considered to be the least sensitive. At the same time, 'Health status', the dimension that is perceived as the least predictable (and is actually among those that are the hardest to predict), is considered as the most sensitive.

Two more observations can be made based on the results shown on Figure 10. The first is that the accuracy of the perceptions of users about the predictability of each dimension tends to vary considerably. Their perception is rather accurate for only some of the dimensions. For instance, users correctly believe that their demographics information is quite predictable (actual predictability is quite high) and also have a quite accurate perception about the predictability of their consumer profile information and factors related to their personality traits. On the other hand, their perception about the predictability of their health related information is rather incorrect. This leads us to the second observation: the actual predictability of the more sensitive dimensions is considerably higher than the perceived predictability. Vice versa, perceived predictability is higher than actual predictability for the less sensitive dimensions (with the exception of 'Religious views').

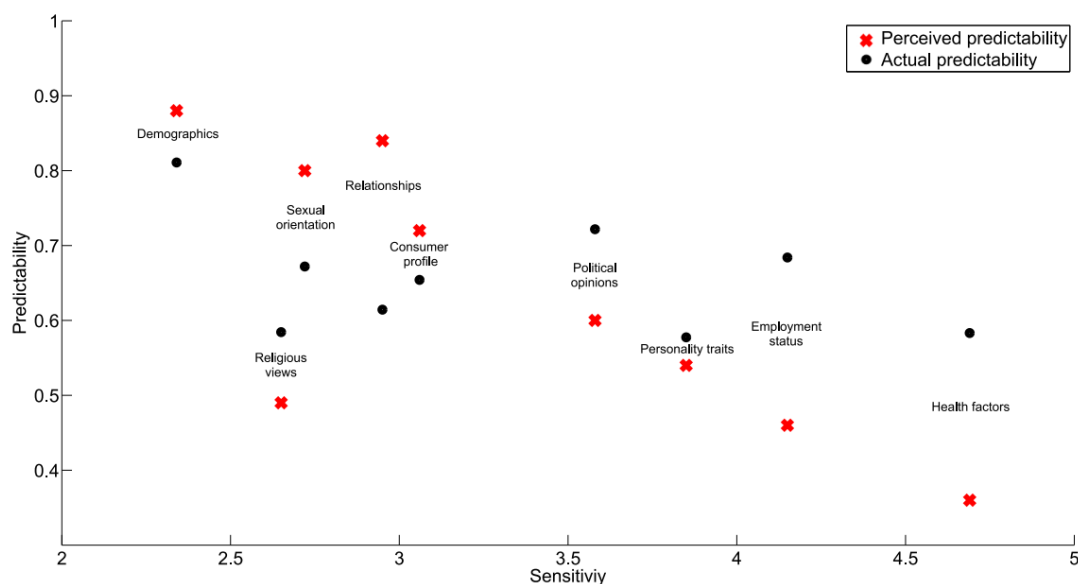


Figure 10. Comparison of perceived and actual predictability of the privacy dimensions with respect to perceived sensitivity.

It is also worth looking at any conclusions that may be reached by looking at the perceptions of individual users and in particular, users that belong to potentially sensitive groups; for

instance, people that have answered that their health is poor or people that are not heterosexuals. We examined whether the sensitivity of particular dimensions differs for users belonging to different classes. We formed a two-way table with one dimension representing the class of the user (e.g. poor/good health) and the other dimension representing the sensitivity of the information. A *X*-square test was performed to examine if the perceptions of different classes of users about the sensitivity of some dimension differ. The test was positive (at the 0.05 level) for the following three dimensions: 'Sexual orientation' (p-value: 0.000003), 'Health factors' (p-value: 0.029) and 'Religious beliefs' (p-value: 0.011). So, for instance, homosexual and bisexual users tend to view the disclosure of information about their sexual profile as more sensitive than heterosexual users. Also, users with good health tend to view the disclosure of information about their health as less sensitive than people with poor health.

To sum up, a number of insights have been extracted with respect to the relationship between actual predictability, perceived predictability and sensitivity. In particular, it appears that users have sometimes largely incorrect perceptions about the predictability of specific types of information. Moreover, the more sensitive a type of information is, the more the users underestimate its predictability. Additionally, the sensitivity of particular types of information seems to be different for users belonging to different classes.

5 Disclosure control assistance

Apart from raising the awareness of users with respect to the disclosure of their personal information, another key goal of WP6 is to assist them to better *control* the disclosure of their personal information. To this end, we have proposed in D6.2 and in D6.4 a policy-based framework that allows users to express their preferences with respect to the disclosure of different types of information. In particular, this framework allows users to define their own disclosure policy by building a set of disclosure rules, each of which is represented as a triplet of the form content-audience-access. The *content* part of the triplet identifies the OSN content posted by the user to which the rule applies (e.g. any content associated with the attribute ‘religious beliefs’), the *audience* part of the triplet identifies the OSN users to which the rule defines that access should be allowed or disallowed (e.g. to the friends of the user that do not belong in the user’s family), depending on the access part of the triplet. For instance, the user may define a rule that defines that “content related to my religious beliefs should not be accessible by anyone apart from my family” or that “content related to my sexual orientation should not be accessible by anyone apart from me”. Eventually, the complete set of rules is used to compute a number of suggestions to the user, in order to change the sharing settings of particular pieces of content. Despite the attractiveness of this approach, there are two main reasons that rendered its implementation problematic:

- First, it is complicated for the average user. As mentioned in (Madejski, 2012): “Access control policies are notoriously difficult to configure correctly, even people who are professionally trained system administrators experience difficulty with the task.” The user would have to undertake the cumbersome task of *manually* defining an exhaustive set of policy rules. Additionally, the way that the policy is applied may not be transparent to the average users, especially in case that there are multiple rules which are applicable to some specific case.
- Second, part of the approach would not be possible to implement due to API restrictions. The main problem is that very limited information about the friends of a user can be obtained using the Graph API. In particular, an application like DataBait can only know the existence of only those friends of the user that also use the application and also very limited information about them can be accessed by the application. Therefore, in most cases, it would be hardly possible to produce any useful matches to the user’s audience definitions.

Motivated by the above, we opted for a simpler, effort-free and more transparent way of generating disclosure settings suggestions. This solution is based on ranking the content posted by the user depending on its contribution to the user’s overall disclosure score (in practice if a particular piece of content is considered by the system to reveal a lot of personal information for the user it is ranked high in the list). Subsequently, the ranked list of content is shown to the users, suggesting to them to reconsider sharing it. Top ranked content will have the highest contribution to the disclosure score and, therefore, either changing its sharing settings or removing it will result in a decrease of the disclosure score. Moreover, the use of the disclosure scoring framework as part of this process allows us to group and present the suggestions according to the associated disclosure dimensions, thereby allowing the user to focus only on the dimension or attributes that he/she thinks are important. By adopting this approach, we produce meaningful disclosure settings suggestions, with the advantage that the user can completely skip the cumbersome task of policy building.

Additionally, since the disclosure score depends on the sensitivity of different types of information and since the sensitivity of different dimensions, attributes and values can be set by the user, the user can affect the ranking of the content based on his/her own sensitivity preferences. This is similar to policy building and can be considered as an even more flexible and intuitive way of expressing the user's disclosure preferences with respect to the different types of information.

To provide an example, in the case of policy building, one could define rules that would cover specific types of information, e.g. all content associated to a user's religious beliefs. Therefore, the presence or absence of rules regarding some specific dimension or attribute would express the relative importance of the different types of information for the user. On the other hand, when the user sets their own sensitivity values, he/she effectively defines the relative importance of the different types of information, albeit in a more detailed and explicit manner, due to the fact that specific numerical values are used.

In addition, building upon WP5 results, we look at an alternative way of suggesting to the user content for which its sharing should be reconsidered. In particular, we use the classification of images into private or public and subsequently recommend to the user to either change the sharing settings or completely avoid sharing those photos that have been classified as private.

Apart from suggesting to the user pieces of content for considering its sharing settings, it was decided to attempt to 'train' DataBait users with respect to the OSN's sharing settings and various information disclosure risks and scenarios. To this end, a number of hints are now shown through popups in the visualization of the disclosure scoring framework. These provide explicit information aiming to assist the user to identify potential threats. Moreover, we have prepared and integrated in DataBait a tutorial that provides both general information about OSN presence as well as specific information about controlling one's own presence in Facebook. This tutorial is integrated to DataBait and is presented to the user together with the aforementioned ranked list of content.

5.1 Training and alerting DataBait users

We now look at the additions that were made in DataBait with the aim of educating the users about the threats associated with the disclosure of different types of information and about controlling their disclosure settings in an OSN environment.

Let us first examine the warnings about the potential threats associated with the different types of information. These are shown as pop-up windows that appear when the user hovers over the node that represents some dimension or attribute. Most of the threats shown had already been identified when coming up with the user attributes that are being considered by DataBait. Nevertheless, in some cases, these were extended with some newly identified threats. It was also decided to make the description of potential threats as short and concise as possible -i.e. thorough details were not provided - with the goal of making them more accessible to users. For instance, the following text is shown to notify users about potential threats involved in disclosing information with respect to nationality:

"In specific cases, the disclosure of the national identity of people may result in racist behavior with direct implications both in the online and offline life of people."

Figure 11 shows a snapshot of the disclosure scoring visualization with a pop-up appearing when the user hovers over an attribute node.

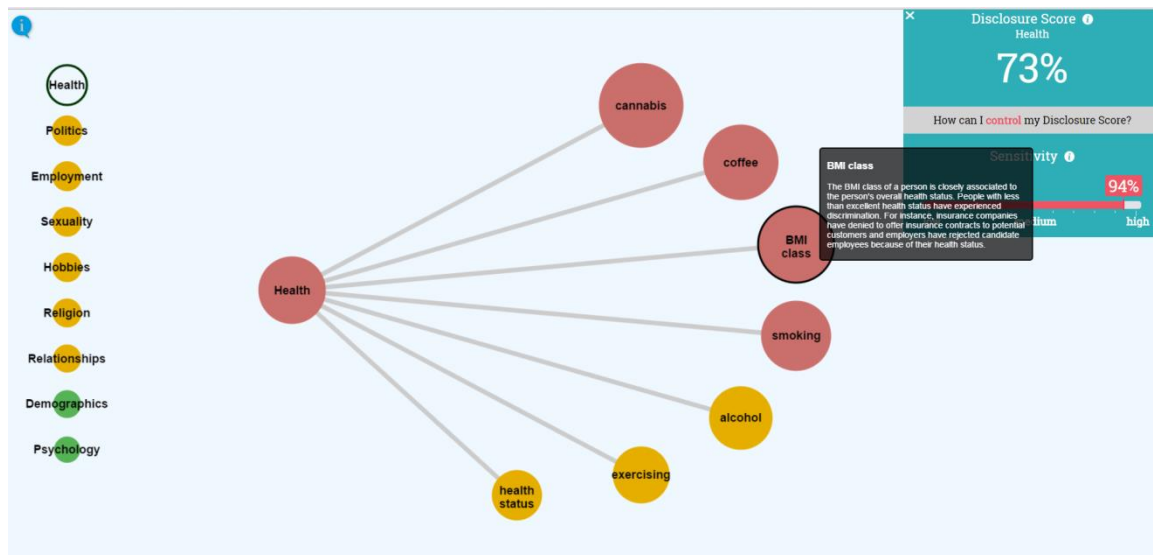


Figure 11. Pop-up window showing potential threats due to the disclosure of information related to a specific user attribute.

Moreover, for specific attributes, e.g. for the psychological attributes, the pop-up window shows a short explanation of what the attribute is about.

As already mentioned, apart from warnings about potential threats, we try to train users with respect to controlling their OSN presence with a tutorial that is provided through DataBait. The tutorial first briefly gives general information on information disclosure on OSNs and then provides specific information about controlling information disclosure on Facebook. It is accessible through an appropriate icon at the control assistance page that provides the list of pieces of content ranked by their contribution to the disclosure score.

The tutorial consists of the following parts:

- 1) An introduction to disclosure control in social networks
- 2) A taxonomy of personal information on social networks
- 3) Sharing settings basics
- 4) Creating friends' lists
- 5) Managing the disclosure of one's profile info
- 6) Image privacy
- 7) Examining the activity log
- 8) View profile as seen by other users
- 9) Blocking other users
- 10) Applications
- 11) Final guidelines

It should be noted that the tutorial is designed so that it is concise and compact and it is also enriched with appropriate snapshots, so that it is more pleasant to read. A snapshot from the tutorial is shown in Figure 12. The full tutorial can be found in Annex 3 of this deliverable.

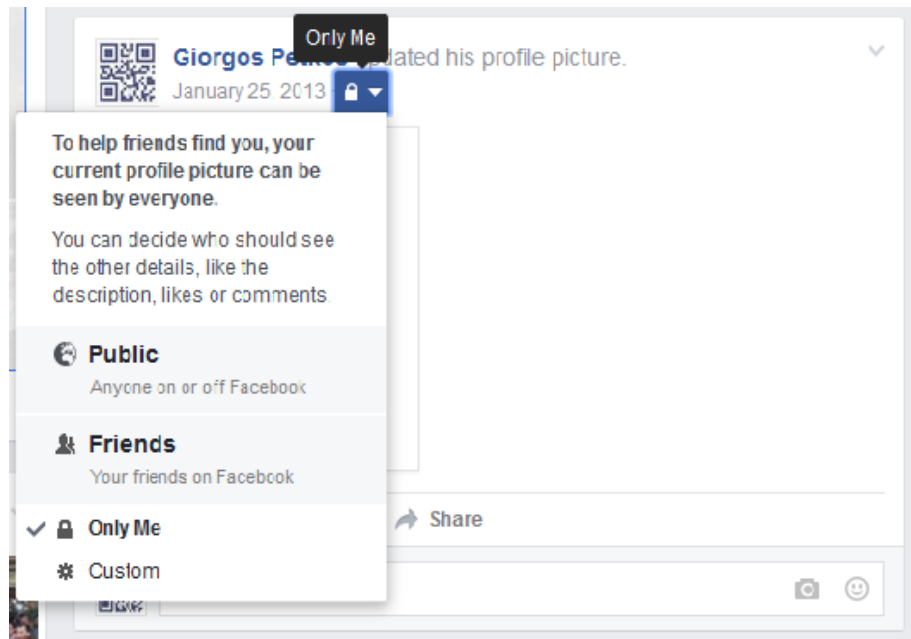


Figure 12. Snapshot from one of the sections of the tutorial

5.2 Sharing suggestions based on the disclosure scoring framework

The implemented user assistance approach is based on the tenet of empowering users to easily identify pieces of shared content that potentially disclose a lot of personal information. To this end, the shared pieces of content are ranked according to their contribution to the disclosure score and the user is prompted to reconsider sharing or to change the sharing settings of those pieces that are ranked highest. It is important to stress though that the users can define their preferences with respect to the disclosure of different types of information by providing their own sensitivity values and in this way they can effectively control the ranking of the list. In the following we look at some details of the relevant module and some challenges that had to be overcome.

The first question that had to be answered in order to proceed with this approach is how to determine the contribution of each piece of content to the disclosure score. Two alternative approaches have been pursued. To start with, it is reminded that the four inference modules that feed data into the disclosure scoring framework are the following:

- The collection-based classifier, operating on the full set of likes, posts and images.
- The likes mapper that handles Facebook likes.
- The URL mapper that considers URL domains included in posts.
- The visual concepts mapper that considers the visual concepts detected in images.

It is clear that for those inference modules that handle individual types of data (likes mapper, URL mapper and visual concepts mapper) the association of particular pieces of content to specific inference results is straightforward. This is not the case for the collection-based classifiers though, as these consider the collection of all OSN data of the user as a whole and it is not straightforward to identify those specific pieces of content that have the highest contribution to the disclosure score.

The initial approach that was implemented in order to associate pieces of content to contributions in the disclosure score considered only the inference results produced by the three inference modules that handle individual types of data. For each of them, the associated disclosure score is calculated as the product of the confidence of the inference, the visibility and sensitivity. Clearly, this is a simplification of the complete scenario, as it ignores the results produced by the collection-based classifier. This simplified approach was integrated to DataBait for the pilots, along with a relevant visualization.

This visualization is shown on the left side of Figure 13 where we see a user’s liked pages ranked by disclosure score. This ranking is produced when the default sensitivity scores are used for all disclosure dimensions. We notice that with the default sensitivity scores, the Health dimension receives the highest disclosure score and, as a result, the highest ranked item is related to this dimension. To showcase the ability that the user has to modify this ranking by modifying the default sensitivity scores of individual dimensions, we show in Figure 14, how the initial ranking of the items changes when we increase the sensitivity of the Hobbies dimension from 61% to 100%. We notice that after this change, the disclosure of the Hobbies dimension increases and this effectively results in items related to this dimension being ranked higher.

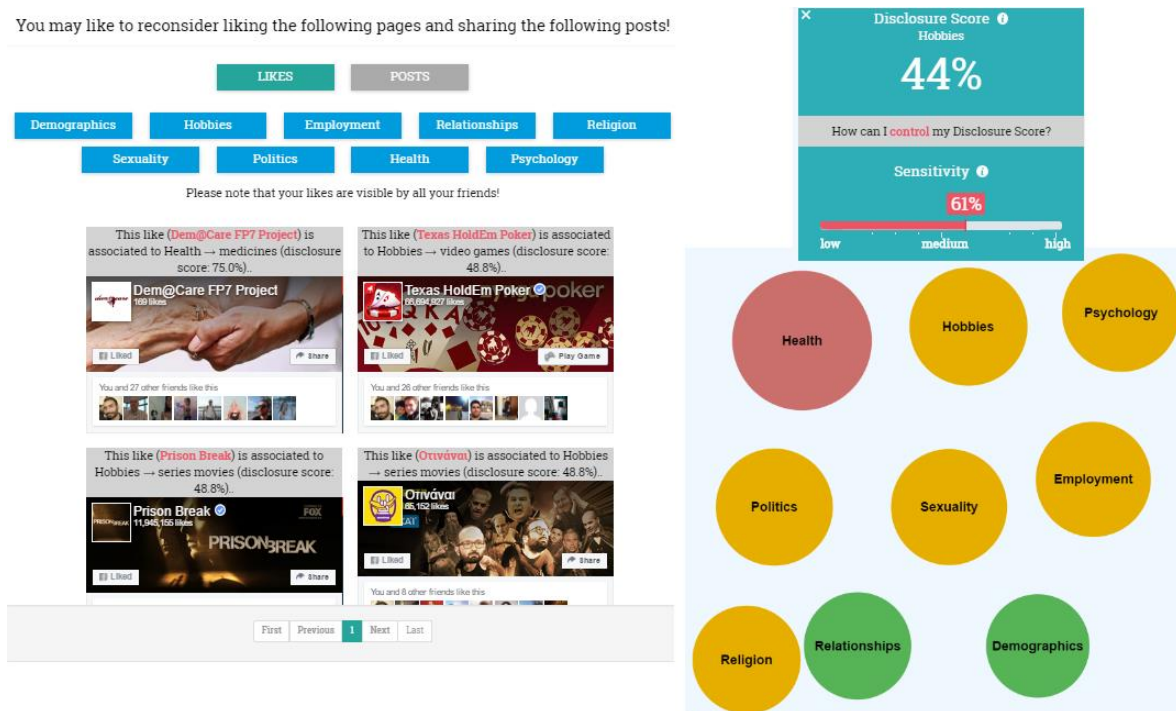


Figure 13. The likes page of the disclosure control assistance module for default sensitivity values.

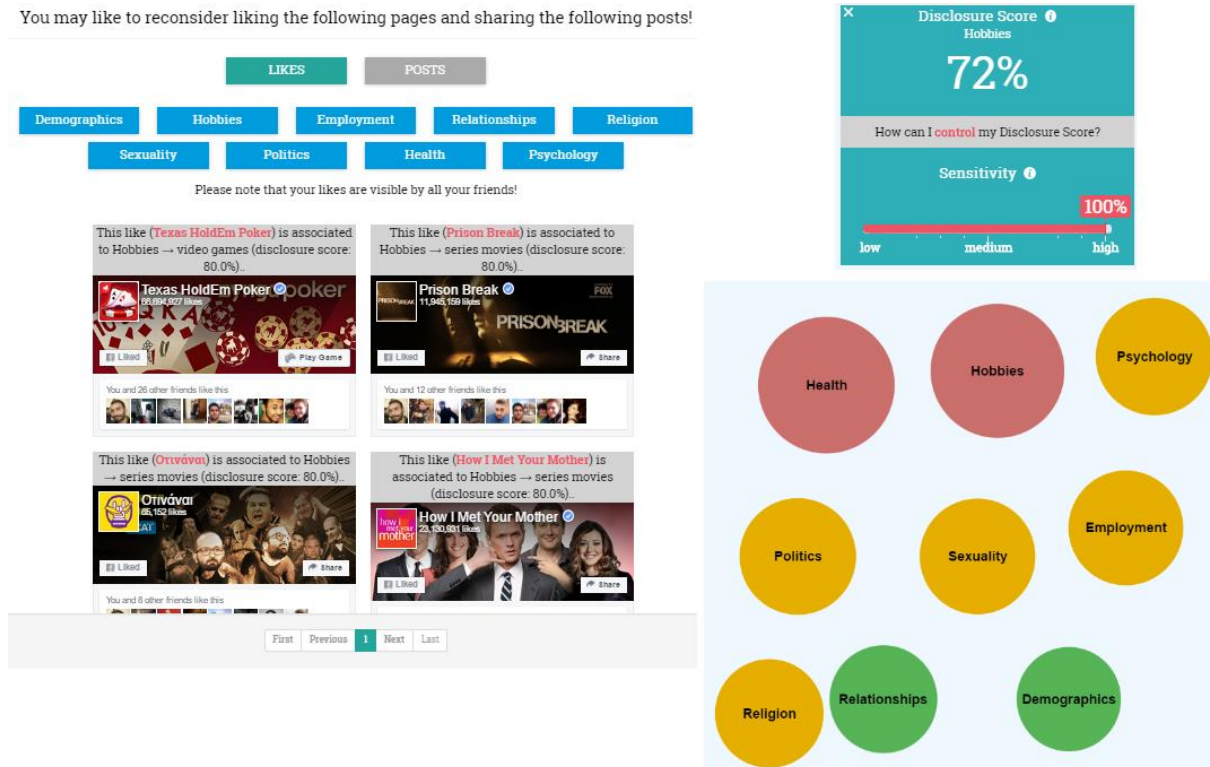


Figure 14: The likes page of the disclosure control assistance module with modified sensitivity values.

The user can opt to view the recommendations of DataBait for the likes, posts (based on the URLs) or the images by clicking the appropriate button at the top of the page. Moreover, the user can decide to focus on results that are associated to specific disclosure dimensions of their choice, all of them or a subset of them. To this end, the user can just select or unselect the appropriate dimensions from the list of dimensions at the second row of buttons at the top of the page. Each of them acts as an on/off button that defines if content associated to the respective dimension will be shown or not. For instance, Figure 15 shows only those likes that are related to the 'health dimension. Results are paginated (since for some users there is a lot of content)

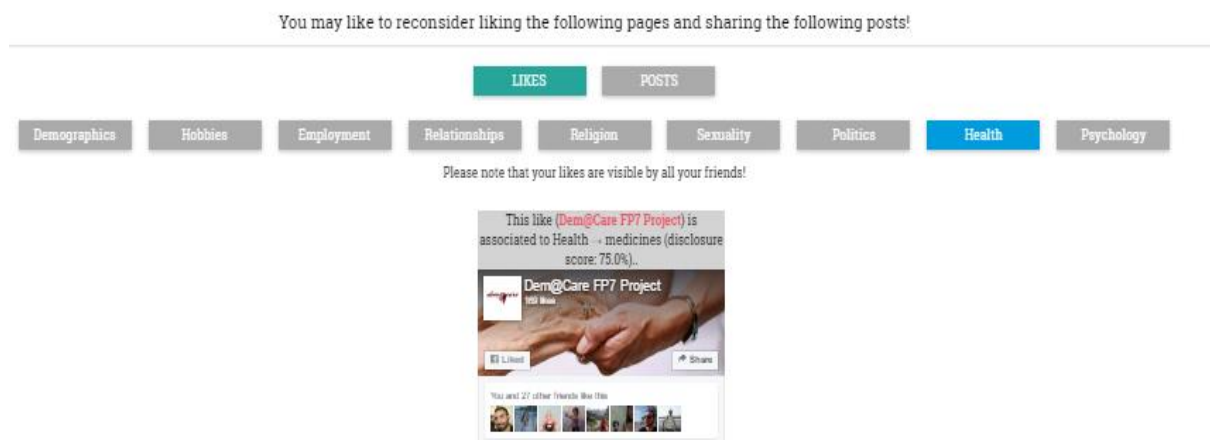


Figure 15. Set of likes filtered according to the associated disclosure dimension

The tutorial on disclosure control in OSNs is accessible through the book icon at the top left corner of the page.

We also implemented an alternative way by which content may be associated to its contribution to the disclosure score. This alternative way attempts to also take into account the inference results produced by the collection-based classifiers. The main challenge is that it is not straightforward to identify specific pieces of content that are responsible for the inference results produced by the collection-based classifier. The approach that has been adopted in order to deal with this problem is to simulate the absence of each piece of content, reproduce the inference results and compute the relevant change in the user's overall disclosure score. More particularly, the following process is carried out:

- The set of disclosure scores of the user is retrieved.
- For each piece of content posted by the user, the following steps are carried out:
 - o A copy of the set of disclosure scores is made.
 - o All support records that point to the currently examined piece of content are removed from the copy of the disclosure scores of the user (this definitely includes all the inference results produced by the collection-based classifiers).
 - o The set of collection-based classifiers is executed again— ignoring the currently examined piece of content - and is fed into the copy of the disclosure scores of the user.
 - o The classifier scores are re-aggregated/re-computed into a new overall disclosure score.
 - o The new overall disclosure score is compared to the initial disclosure score and is stored as the change associated to the removal of the currently examined piece of content.
- The pieces of content are ranked according to the associated change.

Effectively, this procedure simulates the removal of each piece of content and measures the associated change to the overall disclosure score. Clearly, this approach is more principled in the sense that it takes into account all inference results and not only a subset of them. Its disadvantage is that it is computationally intensive, especially for users with a very large amount of posted content. For instance, there are users with more than 3,000 images. This means that the main loop in the above procedure would have to be executed more than 3,000 times, only for the images. Deleting supports and aggregating scores is not a costly procedure. However, running the collection-based classifier is a relatively costly procedure. Various optimizations have been carried out, such as fetching the input data only once and then each time removing only the currently examined piece of content. Nevertheless, this is still a costly procedure. To better perceive the computational cost of this procedure let us first have a look at the distribution of the number of pieces of content that each user has. This distribution is shown in Figure 16.

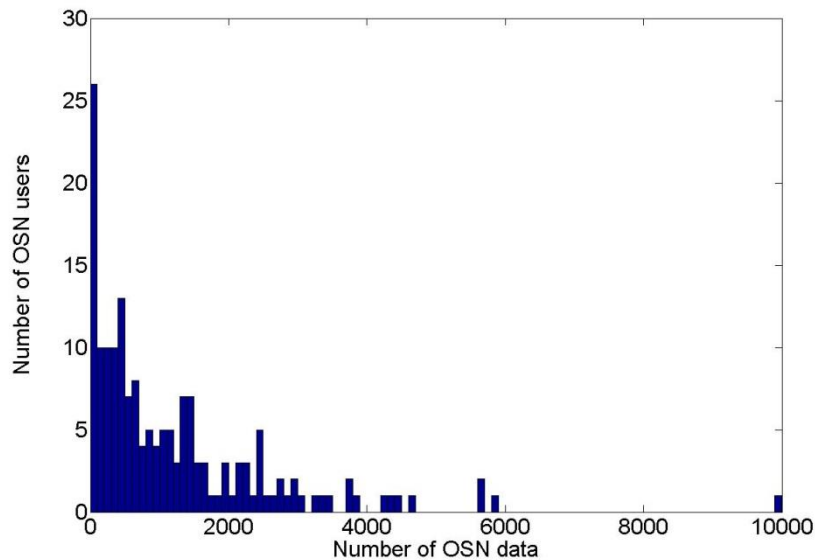


Figure 16. Distribution of the number of pieces of content a user has based on the pre-pilot data.

As can be seen, most users have at most 1,000 to 2,000 pieces of content, with a few having even more. Now, let us see the total cost of the above procedure in relation to the amount of data points under their profile. This is shown in Figure 17.

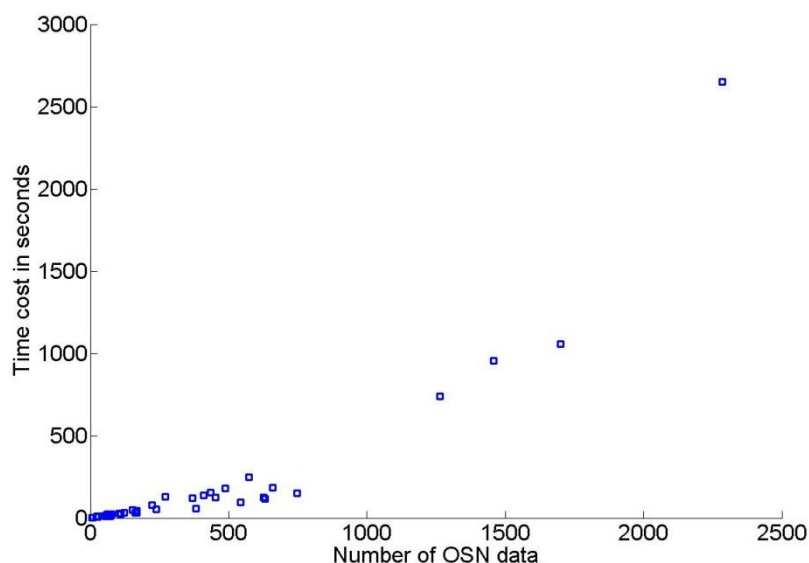


Figure 17. Time cost in seconds of the exhaustive analysis procedure.

The computational cost increases almost linearly for up to 1000 data points and on average, the cost per item is roughly 300ms. Nevertheless, considering that there are quite a few users with more than 1000 items, there is still large computational cost in this procedure.

The above procedure would have to be repeated – computing the new contributions to the overall disclosure score for each posted piece of content. Instead of this costly option, we ended up in a solution, in which every time a new piece of content is added, the change in the overall disclosure score is computed only for the specific item (this is simple, since the new inferences are computed and fed into the disclosure scoring framework anyway) and only repeat the full procedure periodically (e.g. once every week).

5.3 Control assistance based on image privacy

Apart from considering the contribution of the content to the disclosure score of the user, we have also looked into directly identifying potentially sensitive content and notifying the user about it. More specifically, we have looked at the problem of directly classifying images as depicting private or public content. Eventually, images that have been labeled as private are presented to the user, alerting him/her about the potential disclosure of sensitive information. It should be noted that this work has been carried in the context of WP5, but has been developed with the needs of disclosure settings assistance in mind.

The approach recognizes that user perceptions about what types of images may be considered private or public may vary considerably across individuals. Therefore, it accordingly utilizes effective personalization methods. It should be noted that, to the best of our knowledge, (Buschek et al., 2015) is the only other work that considers privacy classification of personal photos. However, (Buschek et al., 2015) evaluate only purely personalized models, assuming that each user provides sufficient amount of feedback. In contrast, our method achieves high performance even at the presence of very limited user-specific feedback by leveraging feedback from other users. Moreover, while (Buschek et al., 2015) use only metadata-based features (location, time, etc.) and simple visual features (colors, edges, etc.), we employ state-of-the-art CNN-based semantic visual features that facilitate comprehensible explanations of the classification outputs.

Importantly, it should be noted that the relevant module was integrated to DataBait and was evaluated during the pilots. A snapshot of the integrated module is shown in Figure 18.

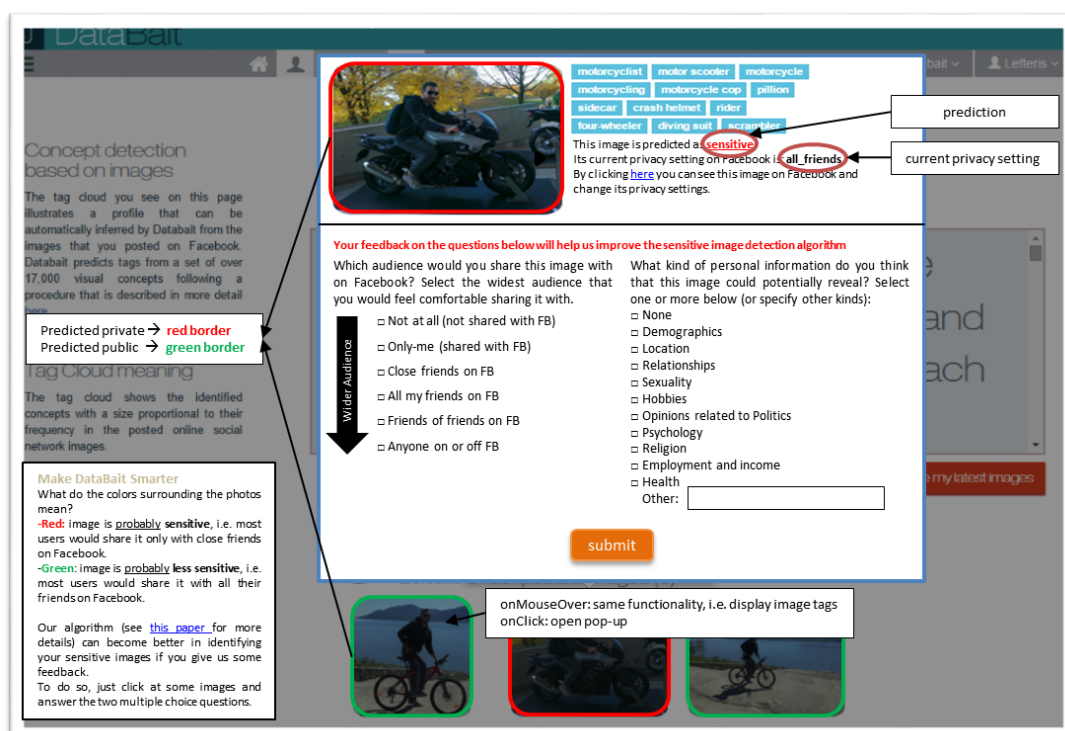


Figure 18: Snapshot of privacy-aware image classification module operating within DataBait.

The module does not only provide suggestions to the user, but it can also receive feedback from the user, so that it takes into account the user's own preferences with the goal of improving performance on new images. For more details on the inner workings of the module and its integration to DataBait please see D5.6.

6 Web Trackers and Do-Not-Track policies

One of the outcomes of the USEMP project is the development of a tool that addresses privacy issues related to a user's Web browsing behaviour. To this end, the DataBait browser plugin allows users to control Web tracking entities that monitor the browsing behaviour of users. In D6.2 we proposed a solution for a Do-Not-Track (DNT) policy, which has been implemented by the DataBait Web tracking tools. In D6.4 we presented an analysis of the evolution of the DataBait plugin and a detailed analysis of initial findings in terms of recommending trackers to be blocked by users. During the last period of the project the work around the DataBait plugin focused on a new mobile friendly version of the DataBait Web tracking tool and tracker, and a URL classification that could provide further insights to users.

6.1 Enhancements on the DataBait plugin

Previous deliverables (D6.2 and D6.4) presented the DataBait browser plugin in detail. The DataBait plugin is a browser extension that allows users to exert fine-grained control over third-party trackers when browsing the Web. The plugin provides a visual representation of the trackers and, through a user-friendly interface, users can select or deselect third-party trackers to be disabled.

In addition, the plugin is fully integrated with the DataBait back-end, storing all users' preferences in regards to blocked trackers. The integration allows the user to transfer their preferences between different sessions in the same browser or even between browsers as long as they log in with their DataBait credentials. Moreover, through the DataBait web application, users can view the history of blocked trackers as well as make changes i.e. block or unblock specific trackers. The data gathered by users can be used for additional analysis as for example recommending to the users, trackers to be blocked (as presented in D6.4) or providing additional information through a URL classification (as will be explained further in section 6.2).

During the last period of the project, enhancements to the plugin were necessary in order to be able to continue supporting the evolutions of the browsers (for example Google Chrome provided an update that disabled main functionalities used by the DataBait plugin). One additional opportunity that was given to the USEMP team during this period is the fully operational availability of the DataBait browser plugin in a mobile browser.

In the last quarter of 2015, Apple made a public launch of the iOS 9, which, among other features, enabled third-party extensions to the integrated mobile browser Safari, finally allowing ad- and content-blocking applications to be transferred to the mobile world in a native way. Taking this opportunity, the USEMP consortium created an iOS version of the DataBait web plugin. With very easy installation, users can have the same functionalities provided by the desktop browser version on their mobile phones.

6.2 Trackers and URL classification

The work that has been carried out on trackers so far aims at informing the user about which trackers observe (part of) their web browsing behavior and which domains each of them knows that the user has visited. Moreover, the developed tool allows the blocking of trackers.

In the following, in order to enrich the information provided to users, we associated – when possible – the domains that the user has visited to specific user attributes and hence disclosure dimensions. Therefore, showing next to the domains that are tracked by each tracker the association of the domain to some user attribute, the user can better perceive the profile that the tracker may have built about him/her.

In this way, users' awareness can be further raised with respect to monitoring of their web browsing behavior by third parties. Especially in the case that the associated user attributes are highly sensitive, they become more aware that their personal information may be disclosed via their web browsing behavior.

In our implementation, the association of URLs to user attributes is carried out with the URL mapper that was presented in chapter 3. It is important to note that just like in the case of posted URLs, it is not always possible to associate a URL domain to some user attribute. For instance, consider the URL domains that correspond to the 20 most frequently tracked domains for the users that installed the trackers plug in (Table 13).

Table 13. Top-20 most visited URL domains for pilot users of the Web browsing plugin.

Tracked URL	Number of users
hwcomms.com	73
facebook.com	62
youtube.com	38
mail.google.com	33
google.be	31
wikipedia.org	27
live.com	25
qualtrics.com	22
docs.google.com	19
surveymonkey.com	19
www.google.com	19
hln.be	17
twitter.com	17
drive.google.com	16
google.se	14
linkedin.com	14
accounts.google.com	13
ltu.se	13
wordpress.com	13
imdb.com	12

Out of those 20 domains, only one can be associated to some privacy dimension: imdb.com can be associated to the “series / movies” attribute under the “hobbies” dimension. In total, from the data available from the pre-pilots, there are 2,061 domains tracked for the 75 users that installed the trackers plug in. Out of those domains, 146 could be mapped to some user attribute, most of them being associated to the consumer profile of users. The full list of the associations made from the available sample data is available in the Annex 4 of this document.

Finally, the URL mapper was actually used in the test server in order to associate domains to user attributes and the relevant visualization was updated. A snapshot is shown in Figure 19.






✓ Website	 enikos.gr	30/11/2015 12:47	18	-	-
✓ Website	 scubadiving.com	30/11/2015 14:21	19	Hobbies	sports
✓ Website	 opap.gr	03/12/2015 15:49	9	Hobbies	betting
✓ Website	 foursquare.com	30/11/2015 14:22	6	-	-
✓ Website	 talassadiving.com	30/11/2015 14:32	4	Hobbies	sports

Figure 19. Snapshot of URL classification integrated with the Web browser plugin.

6.3 Future of DataBait web tracking tool

As people are becoming increasingly privacy conscious, it is evident that research around web tracking and ad- or content-blocking is of high importance. Over the life of the project we have witnessed a number of technology shifts towards providing tools that either train users on privacy issues or help them better protect their personal data. In the web tracking domain, we have seen major companies slowly opening their solutions to be able to cover ad-blocking content (e.g. Apple for mobile browsers) or shifting their solutions to implement do-not-track options (e.g. Google Chrome private browsing).

DataBait not only followed this evolution but also has contributed to both the discussions and the work around tracking protection (see participation in W3C workshop, September 2015). Currently W3C is seeking to standardize both the meaning and the technology of Do-Not-Track and of Tracking Selection Lists. In collaboration with the W3C Tracking Protection Working Group, DataBait web plugin, could become one of the main contributing partners in such effort.

With tracking protection becoming a major issue for the next years, the DataBait web plugin could be further expanded, fully implementing the recommendation system presented in D6.4 and the URL mapper presented here. The open issues and the ideas for the future work of the tracking tool are part of the exploitation plan of the project (D9.7).

7 Conclusions and Next Steps

7.1 Summary of this document

This deliverable presented the work carried out within WP6 in the third year of the project. Important progress has been presented in all the directions of work. In particular:

- A new implementation of the disclosure scoring framework has been developed and successfully integrated to DataBait (Chapter 2).
- Additionally, the visualization of the disclosure scoring framework has been developed - based on the guidelines produced by task 6.3 and feedback received internally from the consortium partners - and integrated to the system (Chapter 2).
- A number of new inference modules were developed (Chapter 3):
 - o The likes mapper that operates on liked Facebook pages.
 - o The URL mapper that examines the domains of posted URLs.
 - o The visual concepts mapper that considers the visual concepts detected in posted images.
- A thorough evaluation of the developed set of inference modules was carried out and used in order to select the inference modules that were eventually used for each attribute in the final system.
- The collection-based classifiers were extended in the following ways (Chapter 3):
 - o It was empirically shown that using external data from the MyPersonality dataset could lead to considerable improvements in terms of classification accuracy for some of the attributes.
 - o Data collected during the pilots were used to achieve further classification accuracy improvements.
- The rebalancing problem was examined in order to improve the models' predictions for minority classes, a problem that is prominent in the USEMP use case scenarios and datasets (Chapter 3).
- The relationship between the users' perceptions about the predictability and sensitivity of different types of information was compared to the actual predictability of the different types of information (as indicated by the performance of our inference modules). It was shown that the more sensitive a piece of information is considered by users, the more users underestimate its actual predictability (Chapter 4).
- To support users in better controlling the disclosure of their information in OSNs, the contribution of each posted content item to the disclosure score is computed. This is used to produce a ranked list, suggesting to users to reconsider sharing those pieces of content that rank highest (Chapter 5).
- Various hints and a tutorial on information disclosure on OSNs are presented to the user with the goal of training them to better control their OSN presence (Chapter 5).
- The work on trackers was extended by associating tracked domains - when possible - to specific user attributes. This provides to the users an idea about the part of their profile of the user that is visible to trackers (Chapter 6).

7.2 Overall WP6 outputs

This being the last deliverable from WP6, it is useful to summarize and discuss the main outputs produced by WP6.

Disclosure scoring framework. The disclosure scoring framework attempts to build an overview of the disclosed profile of a person. In order to formulate the disclosure scoring framework, an extensive review of relevant privacy scoring models was first carried out. Our own formulation extended previous approaches by considering a) an explicitly semantic organization of the different attributes of users (i.e. the disclosure dimensions) and b) the different aspects of information disclosure in an OSN that need to be quantified (please see D6.1, D6.4 and Section 2.1 of this document). Essentially, the development of the disclosure scoring framework was influenced by the existing literature on the subject of privacy scoring. A paper detailing the developed approach was published (Petkos, 2015) and the source code of the approach implementation was made publicly available. The disclosure scoring framework has been successfully integrated in DataBait and tested during the pilots.

Inference modules. The disclosure scoring framework relies on a number of inference modules that analyze the user's OSN data and feed their results into it. In the beginning of the project, various approaches were examined, such as using the set of likes as well as taking into account the set of user links and the principle of homophily (i.e. the principle that people that are close in the OSN will have similar attributes). Eventually, the approaches that were developed early in the project resulted in the development of a module that takes as input the complete set of data posted by a user and makes a number of predictions. The classifiers of this module were initially trained using data from the pre-pilots and we refer to them as *collection-based* classifiers. Due to comments received during the second year review, and recognizing the importance of the inference modules, various extensions were carried out. Eventually, four inference modules were delivered (collection-based classifiers, likes mapper, URL mapper, visual concepts mapper) and moreover additional data were used for training (either external, from the MyPersonality dataset, or internal, from the new data that was obtained during the pilots). Importantly, this extended set of classifiers was evaluated and integrated to DataBait for the pilots. Moreover, the classifiers have been made publicly available as part of the open source package that contains the disclosure scoring framework. Finally, we also examined the problem of class imbalance, which is of increasing importance for the USEMP use cases.

Visualization of the disclosure scoring framework. The design of visualization for the disclosure scoring framework was the subject of task 6.3 and was carried out in an iterative manner, involving repeated cycles of evaluation and improvement, as described in D6.3 and D6.6. The design of the visualization was followed by its actual implementation and integration of the visualization to DataBait. It is important to note that even after the end of the design process that was carried out in collaboration with end users and visualization experts, further feedback was received internally by the consortium and resulted in further improvements. The final version was used in DataBait during the pilots.

Disclosure settings assistance. In the direction of disclosure settings assistance, we opted for a flexible and intuitive solution that is based on the tenet of empowering users to quickly identify pieces of content that could potentially disclose considerable amounts of personal information and be informed on how to act on them. This is achieved through the following:

- Each piece of posted content is ranked according to its impact on the overall disclosure score. The ranked list of content is shown to the user, prompting him/her to reconsider sharing those pieces of content that are ranked highest. Effectively, this utilizes the disclosure scoring framework in order to assist the user to control the sharing of their content. The user can adjust the sensitivity of the different types of information through the disclosure scoring framework visualization and this affects the disclosure score associated to the different pieces of content. Therefore, the user can define their own priorities, essentially a different and simpler kind of policy for the disclosure of different types of personal information. By doing this, the users effectively adapt the ranking of their content to their needs. It is also important to note that two alternative ways of associating content to the disclosure scores have been developed, the first being able to take into account of only the newer inference modules, while the other being able to take into account all inference modules.
- Thorough descriptions of the threats associated with the disclosure of different types of information are shown to the user in order to assist him/her in better perceiving the risks of disclosing specific pieces of content. Additionally, a tutorial about information disclosure in social networks and sharing settings adjustment was developed and integrated in the system. The hints and the tutorial have been developed in order to assist users in taking better decisions about their OSN presence.

Audience influence and personal data value. A number of audience influence indicators were defined and integrated to DataBait. For instance, DataBait, shows the most influenced friends and allows the user to browse their interactions with them. It also shows statistics about the demographics information of the user's audience. Furthermore, an approach was proposed for deriving value estimates for the personal data of OSN users. The proposed approach models the value of personal data as a product of two main factors: a) the online audience of an OSN user, and b) the influence (in terms of reactions and interactions) that a user's OSN posts have on their audience.

Web Trackers. Apart from information disclosure in the context of OSNs, the issue of information disclosure in the context of web browsing behaviour was examined. To this end, web tracker technologies were thoroughly examined and a set of relevant tools were developed and integrated to DataBait. The DataBait web trackers control tools have been extended by characterizing the tracked domains in terms of dimensions and attributes of the disclosure scoring framework.

7.3 Directions for future work

We conclude by discussing some possible directions for future work, along the lines of the subjects considered within WP6:

- As new types of technology emerge, new types of personal data become available. For instance, wearable sensors and Internet of Things sensors have introduced new types of data that reflect a wide variety of real world activities. Considering that most of these real world activities were previously usually not monitored in any way in the user's digital footprint, the capturing and sharing of related data poses new privacy risks. Moreover, apart from user traits that may be directly reflected in such new types of data, this opens new possibilities for the development of more advanced inference algorithms. It is interesting to examine how the concepts and tools that have been

developed within USEMP, such as the disclosure scoring framework, could fit scenarios that involve such technologies and how they could be extended in order to better meet the new challenges.

- Sharing of personal data is an issue with multiple aspects, the main of which are the following:
 - The type of data that is shared and its sensitivity (what).
 - The party that has obtains access to the data (who).
 - The purpose that the data is used for (what for).

Based on this observation, one could envision a framework that would enable users to fully control these three aspects in a transparent manner. Ideally, such a framework would be applicable not only on data stemming from an OSN or from monitoring of web browsing behavior, but for any type of digital footprint. The framework would then have to be part of any piece of software that handles personal data.

Although the development of such a framework may be extremely challenging, one could envision some options for moving forward with it. For instance, the ‘what’ part could be based on work that has been carried out within USEMP (e.g. the work carried out on the disclosure scoring framework and the inference modules is relevant). The ‘who’ part would require the use of robust identification and authentication techniques. Also, secure encryption techniques would also have to be utilized. The ‘what for’ part could be quite challenging though: once the data has been obtained by an authorized party, it is difficult to control how the data is used.

Nevertheless, despite the challenges, the wide adoption of such a framework could have the potential to completely transform the notion of privacy and the perceptions of users about it.

- As mentioned in Chapter 3 and in Chapter 5, although it is straightforward to associate the results produced by the URL mapper, the likes mapper and the visual concepts mapper to some specific piece of data, this is not easy to do with the collection-based classifier. This is the reason why in Chapter 5, in order to measure the contribution of each piece of content to the disclosure score of a user, we resorted to a method in which exhaustively each piece of content is temporarily removed from the data and the associated change in the disclosure score is measured. Nevertheless, associating data items to inference results is important in order to be able to explain inference results to user, thereby increasing their confidence in the produced results. Thus, an interesting direction of work is to develop effective methods that would allow us to identify which data is responsible for the results produced by statistical inference mechanisms.

Annex 1 – Evaluation of the collection-based classifier

This Annex provides detailed evaluation results for the inference modules that have been presented in Chapter 3.

Collection-based classifier

Table 14 lists the evaluation results for the collection-based classifier

Table 14. Performance of the collection-based classifier

Dimension / Attribute	% classif.	Precision	Recall	F-score	Accuracy
Demographics / gender	100%				0.6863
- male		- 0.6711	- 0.9615	- 0.7905	
- female		- 0.8000	- 0.2500	- 0.3809	
Demographics / nationality	100%				0.9294
- Slovak		- -	- 0.0000	- -	
- Russian		- -	- 0.0000	- -	
- Maltese		- -	- 0.0000	- -	
- Danish		- -	- 0.0000	- -	
- Bulgarian		- -	- 0.0000	- -	
- German		- 0.9417	- 1.0000	- 0.9700	
- Belgian		- 0.9104	- 0.9838	- 0.9457	
- Swedish					
Demographics / degree	100%				0.4082
- highschool		- 0.4912	- 0.5090	- 0.5000	
- postgraduate		- 0.4270	- 0.6949	- 0.5290	
- bachelor		- 0.0000	- 0.0000	- -	
Employment / status	100%				0.6358
- employed		- 0.6268	- 0.9545	- 0.7567	
- unemployed		- -	- 0.0000	- -	
- other		- 0.6785	- 0.3064	- 0.4222	
Employment / income	100%				0.4751
- low		- -	- 0.0000	- -	
- medium		- 0.4751	- 1.0000	- 0.6442	
- high		- -	- 0.0000	- -	
Relationship / status	100%				0.4457
- single		- 0.4470	- 0.5757	- 0.5033	
- in relationship		- 0.4285	- 0.5076	- 0.4647	
- married		- 0.7500	- 0.0937	- 0.1666	
Relationship / living situation	100%				0.4882
- with my parents		- 0.2500	- 0.0416	- 0.0714	
- alone or with friends		- 0.4375	- 0.3442	- 0.3853	
- with my own family		- 0.5169	- 0.7625	- 0.6161	

Religion / belief	100%				0.3961
- judaism		- -	- 0.0000	- -	
- agnosticism		- -	- 0.0000	- -	
- islam		- -	- 0.0000	- -	
- atheism		- 0.4233	- 0.9206	- 0.5800	
- catholic		- 0.1764	- 0.0882	- 0.1174	
- protestant		- -	- 0.0000	- -	
- christian (other)		- -	- 0.0000	- -	
- buddhism		- -	- 0.0000	- -	
- other		- -	- 0.0000	- -	
Religion / practice	100%				0.8461
- no		- 0.8461	- 1.0000	- 0.9166	
- yes		- -	- 0.0000	- -	
Psychology / agreeableness	100%				0.8764
- agreeable		- 0.8764	- 1.0000	- 0.9341	
- disagreeable		- -	- 0.0000	- -	
Psychology / conscientiousness	100%				0.8000
- conscientious		- 0.8083	- 0.9854	- 0.0555	
- unconscientious		- 0.3333	- 0.0303	- 0.8881	
Psychology / extraversion	100%				0.7235
- extravert		- 0.7417	- 0.9333	- 0.8265	
- introvert		- 0.5789	- 0.2200	- 0.3188	
Psychology / neuroticism	100%				0.6529
- neurotic		- 0.6691	- 0.8557	- 0.7510	
- stable		- 0.5945	- 0.3333	- 0.4271	
Psychology / openness	100%				0.8647
- open		- 0.8698	- 0.9932	- 0.9274	
- closed		- 0.0000	- 0.0000	- -	
Sexuality / orientation	100%				0.8750
- homo/bi		- -	- 0.0000	- -	
- heterosexual		- 0.8750	- 1.0000	- 0.9333	
Politics / ideology	100%				0.7187
- left		- 0.7187	- 1.0000	- 0.8363	
- centre		- -	- 0.0000	- -	
- right		- -	- 0.0000	- -	
Health / status	100%				0.5209
- poor		- -	- 0.0000	- -	
- good		- 0.0000	- 0.0000	- -	
- very good		- 0.5649	- 0.8969	- 0.6932	
Health / coffee	100%				0.6882
- no		- 0.6250	- 0.0909	- 0.1587	
- yes		- 0.6913	- 0.9739	- 0.8086	
Health / smoking	100%				0.8529
- no		- 0.8529	- 1.0000	- 0.9206	
- yes		- -	- 0.0000	- -	
Health / alcohol	100%				0.7941
- no		- -	- 0.0000	- -	
- yes		- 0.7941	- 1.0000	- 0.8852	
Health / cannabis	100%				0.9058

- no - yes		- 0.9058 - -	- 1.0000 - 0.0000	- 0.9506 - -	
Health / BMI class - healthy - non-healthy	100%	- 0.6258 - 0.4615	- 0.8613 - 0.1875	- 0.7249 - 0.2666	0.6000
Health / exercising - no - yes	100%	- 0.6728 - 0.5000	- 0.9646 - 0.0701	- 0.7927 - 0.1230	0.6647
Hobbies / reading - no - yes	100%	- 0.6016 - 0.5192	- 0.7395 - 0.3648	- 0.6635 - 0.4285	0.5764
Hobbies / series movies - no - yes	100%	- - - 0.7470	- 0.0000 - 1.0000	- - - 0.8552	0.7470
Hobbies / gardening - no - yes	100%	- 0.8941 - -	- 1.0000 - 0.0000	- 0.9440 - -	0.8941
Hobbies / music - no - yes	100%	- 0.5463 - 0.6164	- 0.6543 - 0.5056	- 0.5955 - 0.5555	0.5764
Hobbies / sports - no - yes	100%	- 0.7941 - -	- 1.0000 - 0.0000	- 0.8852 - -	0.7941
Hobbies / shopping - no - yes	100%	- 0.8882 - -	- 1.0000 - 0.0000	- 0.9408 - -	0.8882
Hobbies / travelling - no - yes	100%	- 0.6235 - -	- 1.0000 - 0.0000	- 0.7681 - -	0.6235
Hobbies / hiking - no - yes	100%	- 0.9529 - -	- 1.0000 - 0.0000	- 0.9759 - -	0.9529
Hobbies / camping - no - yes		- 0.9705 - -	- 1.0000 - 0.0000	- 0.9850 - -	0.9705
Hobbies / animals - no - yes	100%	- 0.9058 - -	- 1.0000 - 0.0000	- 0.9506 - -	0.9058
Hobbies / dancing - no - yes	100%	- 0.9176 - -	- 1.0000 - 0.0000	- 0.9570 - -	0.9176
Hobbies / theatre - no - yes	100%	- 0.9647 - -	- 1.0000 - 0.0000	- 0.9820 - -	0.9647

URL mapper

Table 15 lists the evaluation results for the URL mapper.

Table 15. Performance of the URL mapper

Attribute	% classif.	Precision	Recall	F-score	Accuracy
Demographics / gender	0.59%				0.0000
- male		- 0.0000	- 0.0000	- -	
- female		- -	- 0.0000	- -	
<u>Sexuality / orientation</u>	0.58%				1.0000
- Homo/bi		- 1.0000	- 0.0476	- 0.0909	
- heterosexual		- -	- 0.0000	- -	
<u>Hobbies / music</u>	8.82%				0.5333
- no		- -	- 0.0000	- -	
- yes		- 0.5333	- 0.0898	- 0.1538	
<u>Hobbies / reading</u>	4.12%				0.8571
- no		- -	- 0.0000	- -	
- yes		- 0.8571	- 0.0810	- 0.1481	
<u>Hobbies /series movies</u>	5.88%				0.6000
- no		- -	- -	- -	
- yes		- 0.6000	- 0.0472	- 0.0875	
Hobbies/ sports	6.47%				0.1818
- no		- -	- -	- -	
- yes		- 0.1818	- 0.0571	- 0.0869	
Hobbies / shopping	1.76%				0.0000
- no		- -	- 0.0000	- -	
- yes		- 0.0000	- 0.0000	- -	
Hobbies / travelling	2.35%				0.2500
- no		- -	- 0.0000	- -	
- yes		- 0.2500	- 0.0156	- 0.0294	
Hobbies / hiking	0.58%				0.0000
- no		- -	- -	- -	
- yes		- 0.0000	- 0.0000	- -	
Hobbies / motor sports	0.58%				0.0000
- no		- -	- 0.0000	- -	
- yes		- 0.0000	- 0.0000	- -	
<u>Hobbies / animals</u>	1.17%				0.5000
- no		- -	- 0.0000	- -	
- yes		- 0.5000	- 0.0625	- 0.1111	
Hobbies / dancing	0.58%				0.0000
- no		- -	- 0.0000	- -	
- yes		- 0.0000	- 0.0000	- -	

Hobbies / theatre	0.58%	- -	- 0.0000	- -	0.0000
- no		- -	- 0.0000	- -	
- yes		- 0.0000	- 0.0000	- -	
Hobbies / gardening	1.18%	- -	- 0.0000	- -	0.0000
- no		- -	- 0.0000	- -	
- yes		- 0.0000	- 0.0000	- -	

Likes mapper

Table 16 lists the evaluation results for the likes mapper.

Table 16. Performance of the likes mapper

Dimension / attribute	% classif.	Precision	Recall	F-score	Accuracy
<u>Religion / belief</u>					
- judaism		- -	- 0.0000	- -	
- agnosticism		- -	- 0.0000	- -	
- islam		- -	- 0.0000	- -	
- atheism	3.90%	- 0.0000	- 0.0000	- -	0.6666
- christianity		- -	- 0.0000	- -	
- catholic		- 1.0000	- 0.2000	- 0.3333	
- protestant		- 1.0000	- 0.2000	- 0.3333	
- buddhism		- -	- 0.0000	- -	
- other		- -	- 0.0000	- -	
<u>Religion / practice</u>					
- no	9.09%	- -	- 0.0000	- -	0.6153
- yes		- 0.6153	- 0.3636	- 0.4571	
<u>Sexuality / orientation</u>					
- homo/bi	1.79%	- 0.6666	- 0.0952	- 0.1666	0.6666
- heterosexual		- -	- 0.0000	- -	
<u>Politics / ideology</u>					
- left	43.75%	- 1.0000	- 0.4347	- 0.6060	1.0000
- centre		- -	- 0.0000	- -	
- right		- 1.0000	- 0.4444	- 0.6153	
Health / status					
- poor	0.59%	- 0.0000	- 0.0000	- -	0.0000
- good		- -	- 0.0000	- -	
- very good		- -	- 0.0000	- -	
<u>Health / coffee</u>					
- no	15.29%	- -	- 0.0000	- -	0.7692
- yes		- 0.7692	- 0.1739	- 0.2836	
<u>Health / smoking</u>					
- no	2.35%	- 1.0000	- 0.0275	- 0.0536	1.0000
- yes		- -	- 0.0000	- -	
<u>Health / alcohol</u>					
- no	38.82%	- -	- 0.0000	- -	0.8939
- yes		- 0.8939	- 0.4370	- 0.5870	
<u>Health / BMI class</u>					
- healthy	18.18%	- 1.0000	- 0.0099	- 0.0196	0.3333
- unhealthy		- 0.3103	- 0.1406	- 0.1935	

Health / exercising						
- no	8.24%	- -	- 0.0000	- -	- -	0.4285
- yes		- 0.4285	- 0.1052	- 0.1690		
Hobbies / reading						
- no	60.00%	- -	- 0.0000	- -	- -	0.5196
- yes		- 0.5196	- 0.7162	- 0.6022		
Hobbies / series movies						
- no	85.29%	- -	- 0.0000	- -	- -	0.7586
- yes		- 0.7586	- 0.8661	- 0.8088		
Hobbies / gardening						
- no	2.35%	- -	- 0.0000	- -	- -	0.2500
- yes		- 0.2500	- 0.0555	- 0.0909		
Hobbies / music						
- no	82.35%	- -	- 0.0000	- -	- -	0.5285
- yes		- 0.5285	- 0.8314	- 0.6462		
Hobbies / sports						
- no	80.59%	- -	- 0.0000	- -	- -	0.2335
- yes		- 0.2335	- 0.9142	- 0.3720		
Hobbies / shopping						
- no	71.18%	- -	- 0.0000	- -	- -	0.1322
- yes		- 0.1322	- 0.8421	- 0.2285		
Hobbies / travelling						
- no	65.88%	- -	- 0.0000	- -	- -	0.4464
- yes		- 0.4464	- 0.7812	- 0.5681		
Hobbies / hiking						
- no	0.59%	- -	- 0.0000	- -	- -	0.0000
- yes		- 0.0000	- 0.0000	- -		
Hobbies / cooking						
- no	12.35%	- -	- 0.0000	- -	- -	0.3809
- yes		- 0.3809	- 0.1666	- 0.2318		
Hobbies / camping						
- no	3.53%	- -	- 0.0000	- -	- -	0.0000
- yes		- 0.0000	- 0.0000	- -		
Hobbies / motor sports						
- no	25.29%	- -	- 0.0000	- -	- -	0.0930
- yes		- 0.0930	- 0.4444	- 0.1538		
Hobbies / animals						
- no	15.29%	- -	- 0.0000	- -	- -	0.2307
- yes		- 0.2307	- 0.3750	- 0.2318		
Hobbies / dancing						
- no	7.65%	- -	- 0.0000	- -	- -	0.0769
- yes		- 0.0769	- 0.0714	- 0.0740		
Hobbies / theatre						
- no	4.71%	- -	- 0.0000	- -	- -	0.1250
- yes		- 0.1250	- 0.1666	- 0.1428		

Visual concepts mapper

Table 17 lists the evaluation results for the visual concepts mapper

Table 17. Performance of the visual concepts mapper

Attribute	% clasif.	Precision	Recall	F-score	Accuracy
Religion / practice					
- no	9.79%	- -	- 0.0000	- -	0.2142
- yes		- 0.2142	- 0.1363	- 0.1666	
<u>Health / smoking</u>					
- no	66.47%	- 0.8181	- 0.0620	- 0.1153	0.2300
- yes		- 0.1666	- 0.6800	- 0.2677	
<u>Health / alcohol</u>					
- no	35.09%	- -	- 0.0000	- -	0.8650
- yes		- 0.8653	- 0.3333	- 0.4812	
<u>Health / coffee</u>					
- no	24.71%	- -	- -	- -	0.8095
- yes		- 0.8095	- 0.2956	- 0.4331	
Hobbies / dancing					
- no	16.47%	- -	- 0.0000	- -	0.0357
- yes		- 0.0357	- 0.0714	- 0.0476	
Hobbies / camping					
- no	18.82%	- -	- 0.0000	- -	0.0312
- yes		- 0.0312	- 0.2000	- 0.0540	
Hobbies / gardening					
- no	18.24%	- -	- 0.0000	- -	0.0967
- yes		- 0.0967	- 0.1666	- 0.1224	
Hobbies / theatre					
- no	14.71%	- -	- 0.0000	- -	0.0800
- yes		- 0.0800	- 0.3333	- 0.1290	
<u>Hobbies / music</u>					
- no	22.94%	- -	- 0.0000	- -	0.5897
- yes		- 0.5897	- 0.2584	- 0.3593	
Hobbies / animals					
- no	37.06%	- -	- 0.0000	- -	0.1428
- yes		- 0.1428	- 0.5625	- 0.2278	
Hobbies / shopping					
- no	9.41%	- -	- 0.0000	- -	0.1250
- yes		- 0.1250	- 0.1052	- 0.1142	
Hobbies / motor sports					
- no	20.59%	- -	- -	- -	0.1714
- yes		- 0.1714	- 0.6666	- 0.2727	
Hobbies / sports					
- no	29.41%	- -	- -	- -	0.2400
- yes		- 0.2400	- 0.3428	- 0.2823	
Hobbies / reading					
- no	21.18%	- -	- -	- -	0.4166
- yes		- 0.4166	- 0.2027	- 0.2727	

Annex 2 – Mappings from likes categories to user attributes

This Annex lists the mappings from likes categories – as provided by Facebook – to user attributes that were used as part of the likes mapper (Chapter 3):

Library → Hobbies / reading
Movie Theater → Hobbies / series movies
Religion → Religion / practice
Concert Venue → Hobbies / music
Music Video → Hobbies / music
Sport → Hobbies / sports
TV → Hobbies / series movies
Movie Genre → Hobbies / series movies
Record Label → Hobbies / music
Retail and Consumer Merchandise → Hobbies / shopping
Author → Hobbies / reading
Musical Instrument → Hobbies / music
Teens/Kids Website → Demographics / has child
Movie Character → Hobbies / series movies
Album → Hobbies / music
Transport/Freight → Hobbies / travelling
Music Award → Hobbies / music
Theatrical Play → Hobbies / theatre
Clothing → Hobbies / shopping
Recreation/Sports Website → Hobbies / sports
Airport → Hobbies / travelling
TV Channel → Hobbies / series movies
Book Series → Hobbies / reading
Church/Religious Organization → Religion / practice
Actor/Director → Hobbies / series movies
Automotive → Hobbies / motor sports
Video Game → Hobbies / video games
Musician/Band → Hobbies / music

Musical Genre → Hobbies / music
Book Store → Hobbies / reading
Amateur Sports Team → Hobbies / sports
Movie → Hobbies / series movies
Drugs → Health / drugs
Animal Breed → Hobbies / animals
Coach → Hobbies / sports
Athlete → Hobbies / sports
TV Show → Hobbies / series movies
Writer → Hobbies / reading
School Sports Team → Hobbies / sports
Sports League → Hobbies / sports
Shopping/Retail → Hobbies / shopping
Pet → Hobbies / animals
Patio/Garden → Hobbies / gardening
Song → Hobbies / music
Drink → Health / alcohol
Publisher → Hobbies / reading
Episode → Hobbies / series movies
Health/Medical/Pharmaceuticals → Health / medicines
Jewelry/Watches → Hobbies / shopping
Health/Medical/Pharmacy → Health / medicines
Book → Hobbies / reading
Concert Tour → Hobbies / music
Animal → Hobbies / animals
TV/Movie Award → Hobbies / series movies
Wine/Spirits → Health / alcohol
Holiday → Hobbies / travelling
Vitamins/Supplements → Health / medicines
TV Season → Hobbies / series movies
Sports Venue → Hobbies / sports
Pet Supplies → Hobbies / animals
Hotel → Hobbies / travelling
Travel/Leisure → Hobbies / travelling

Book Genre → Hobbies / reading
Automobiles and Parts → Hobbies / motor sports
Baby Goods/Kids Goods → Demographics / has child
Outdoor Gear/Sporting Goods → Hobbies / sports
Sports Event → Hobbies / sports
Pet Services → Hobbies / animals
Sports Team → Hobbies / sports
Music → Hobbies / music
Dancer → Hobbies / dancing
Sports/Recreation/Activities → Hobbies / sports
Tours/Sightseeing → Hobbies / travelling
Movie Studio → Hobbies / series movies
Music Chart → Hobbies / music
Kitchen/Cooking → Hobbies / cooking
Bar → Health / alcohol
TV Network → Hobbies / series movies
Comedian → Hobbies / series movies
Cars → Hobbies / motor sports
TV Genre → Hobbies / series movies

Annex 3 – OSN information disclosure tutorial

This Annex presents the tutorial personal information disclosure in OSNs that has been integrated to DataBait. The tutorial is split in 11 sections; it first discusses some general issues about information disclosure in OSNs and then proceeds to discuss specific details about controlling sharing settings in Facebook. It should be noted that along the tutorial different risks and ways by which information leak may occur are mentioned.

Introduction on disclosure control in OSNs

It is not an exaggeration to say that social networks have transformed the overall Internet landscape! Social networks have affected the way people communicate, are being informed or even make business online. An issue that is sometimes overlooked though is the exposure of personal information through the social networks. Participation at a social network means that a certain amount of data related to the user is accessible from a) other social network users and b) the social network service itself. The disclosure of specific types of information may pose serious threats to the users though. A relevant example that has attracted considerable attention is a [tool](#) that analyzes Twitter accounts in order to identify the physical location of their owners, unveiling a potential vulnerability of the users' residence. In other cases, information about the gender, age, ethnicity, political or religious beliefs, sexual preferences, and financial status of a person have been used for unjustified discrimination, for instance, in the context of personnel selection and for loan approval and pricing based on social media profiles. Within the DataBait tool, risks associated with the disclosure of different types of information are shown through the disclosure scoring framework when you hover over the nodes of the different dimensions and attributes.

It is interesting to note though that people's attitude towards information disclosure differs significantly. For instance, [\[Knijneburg\]](#) identifies three main classes of users with respect to the level of information disclosure in OSNs:

1. Privacy fundamentalists
2. Pragmatists
3. Unconcerned

Privacy fundamentalists avoid sharing any content at all at the social network, pragmatists do share content but are careful about the content that they share and unconcerned users post anything without considering privacy at all. Clearly, unconcerned users have the highest risk; however, pragmatists also run risks. In fact, it appears that most users, regardless of their attitude towards privacy, seem to have difficulties managing their information disclosure. For instance, in a seminal study by [\[Madejski\]](#), 65 users were asked to look for any sharing violations in their OSN profiles, that is to find cases in which they shared content with people that they really would not like to. Indeed, all 65 users found that they had at least one sharing violation. [\[Acquisti\]](#) attributed this behaviour to incomplete information, bounded cognitive ability and cognitive and behavioural biases, which may be caused by difficult to find settings and opt-out defaults. This is where DataBait comes to play and attempts to assist users in taking appropriate decisions about controlling their presence at a social network.

One of the tools provided by DataBait for raising awareness regarding information disclosure and assisting the user to adjust their present online is a list of suggestions for particular pieces of content the sharing settings of which you may need to reconsider. Please note though that due to Facebook API constraints, the DataBait application is not allowed to make direct changes to your sharing settings, and therefore the suggestions, if you decide to follow them, will need to be applied directly to Facebook, rather than through DataBait. Complementary to the tools provided by DataBait, in the following, we provide this tutorial for assisting the users to control their presence at the social network. The first part of the tutorial provides some rather theoretical introduction about the different types of personal data that is shared on social networks. Subsequently, this tutorial discusses in detail the sharing options and tools offered by Facebook for managing privacy. The hope is that this tutorial, in conjunction with the tools offered by DataBait, will substantially assist users to better control their presence at the social network.

A taxonomy of personal information on social networks

Before looking at practical details and guidelines for managing disclosure settings, it is useful to present a taxonomy of user data in social networks. This taxonomy considers the source of data about a user, rather than the type of personal information (as e.g. in the disclosure scoring framework). This will allow us to distinguish between different levels of privacy and to identify the limits to which a social network user can control the disclosure of their information. The taxonomy has been defined by [\[Schneier\]](#). Briefly, Schneier identifies the following six categories of OSN data:

- Service data. This is the set of data that a user explicitly provides to the OSN service. In many cases, this includes the user's legal name, age, gender, etc.
- Disclosed data. This includes the content (messages, status updates, photos, etc.) posted by the user to his own page.
- Entrusted data. This is the content posted by the user to the page of another user. It is similar to disclosed data, with the difference that, in many cases, the user does not have full control of the content, but some other user does.
- Incidental data. This is the content posted about the user by some other user (e.g. when a friend of the user posts a picture depicting the user). Again, this is similar to disclosed data, but again, the user does not have control of the content.
- Behavioural data. This type of data includes the actions of the user in the OSN. For instance, this may include information about which profiles the user visits, what games s/he plays, what pages s/he likes, etc.
- Derived or inferred data. This is data about a user that may be derived from all other types of data, typically by means of algorithmic processes. We will also refer to such kind of data as inferred or inferences.

Schneier's taxonomy identifies that the level of control a user has over the data that concern him/her may vary significantly depending on the above categories. For instance, the user typically has full control over the personal details that he/she explicitly provides to the social network in order to register with the service (service data) and that he/she deliberately posts

to his/her profile (disclosed data). On the other hand, the user's control is limited over data that he/she posts on other people's profiles (entrusted data), data about him/her that is posted from other users (incidental data) and data that has resulted from analysing other data (derived or inferred data).

Based on the above, it is useful to make the following remarks:

- Social network users typically focus on disclosed data and often on entrusted and incidental data. Nevertheless, other types of data are also important for privacy; in particular, inferred data are very likely to disclose potentially sensitive information. Within DataBait, there is a particular focus on inferred data. That is, within the disclosure scoring framework, the user can examine the different types of information that can be inferred about him/her. It should be kept in mind that these inferences come with some uncertainty due to the statistical nature of the related inference models. Nevertheless, even in an application of this scale, the accuracy of the inferences is on average quite accurate and suffices to show the scope of possible inferences.
- The social network service typically has access to all those types of data. In fact, one can identify two different types of privacy: social privacy, where privacy concerns the disclosure of information to the other users of the social network, and institutional privacy, where privacy concerns the disclosure of information to the social network service itself. It is also important to note that the inferences that can be made by the social network service are much more elaborate and accurate than those offered by DataBait, as the social network service has access to a much larger pool of data that it can use to base its predictions on. Additionally, it is important to note that the social network service typically does not seem to forget any data, even when the users delete it. Therefore, it is important to consider before posting anything that data never really completely disappears from a social network.
- Behavioural data may disclose much more information than one may initially think of and much more than other types of data. Behavioural data are typically not observed by third parties, only by the social network service. It should be noted that there are clear indications that the service utilizes this data, nevertheless, it is unclear exactly how.

Some legal issues with respect to OSN providers

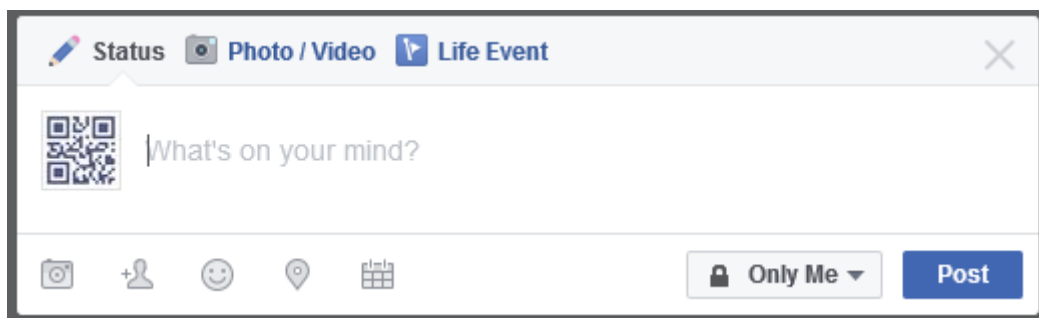
This section of the tutorial discusses some legal issues related to the rights and duties of users and OSN providers. It has been written in collaboration with WP3 and is based on excerpts from D3.10. Since the included content can also be found in D3.10, only the introductory paragraph of this section of the tutorial is included here:

How can users of online platforms know what can be inferred from their online disclosures and behavior, who has access to this information and how it can be used commercially and otherwise? According to EU data protection law, providers of online services who process information relating to individuals have all kind of duties. These duties include, for example, that a data controller only collects and processes data based on one of the legal grounds listed in Art. 6(1) GDPR 2016/679 (for example, a legitimate interest of the controller that is



not overruled by user interests, explicit user consent or a contract the user that necessitates processing), that she does not process data in a way that is unforeseeable (that is, incompatible with the specified, explicit and legitimate purposes set out at the moment of data collection), that she keeps the data safe, secure and up-to-date, and that she deletes them as soon as they are no longer necessary. Data controllers also have certain *informational* duties (see Arts. 12-15 GDPR 2016/679) with regard to their users when they process their data. These informational duties include providing some basic information at the time of data collection (e.g., purpose for collecting the data, contact details of the data controller, persons to whom the data may be disclosed, indication when the data will be deleted, existence of the right of access and rectification) and the duty to provide access to the data upon a user's request. When data processing includes *profiling* of users, these informational duties also entail that the data controller has to inform users about the fact that they are subjected to profiling and provide "meaningful information about the logic involved, as well as the significance and the envisaged consequences of such processing for the data subject"

Sharing settings basics

- The figure below shows the box that appears at the top of a Facebook user's profile and that allows users to post new content.



Please note that the user can post three different types of data depending on which tab of the box is selected: status, photo / video or life event. Particularly important for privacy are the following options:

- By clicking on the  icon, the user can tag other users. By doing this, the user directly provides information about other users. Of course, the user that has been tagged can remove the tag; however, it should be clear that disclosure of our information may sometimes be out of our control, since other users can share information about us.
- By clicking on the  icon, the user can provide his / her location. This action explicitly provides potentially sensitive information about the user. It is often the case also that posts are automatically tagged with location information, regardless of the fact that the user may not have explicitly provided it.
- By clicking on the drop down list next to the post button, the user can set the audience that will have access to the shared content. The options are: public, friends,

only me and custom. It is important to note that selecting one of these options, also changes the default option for future posts. In the 'custom' option, the user can explicitly define which of his friends to share or not to share the content with (please see the next figure).

Custom Privacy [X]

+ Share with

These people or lists

Anyone tagged will be able to see this post.

× Don't share with

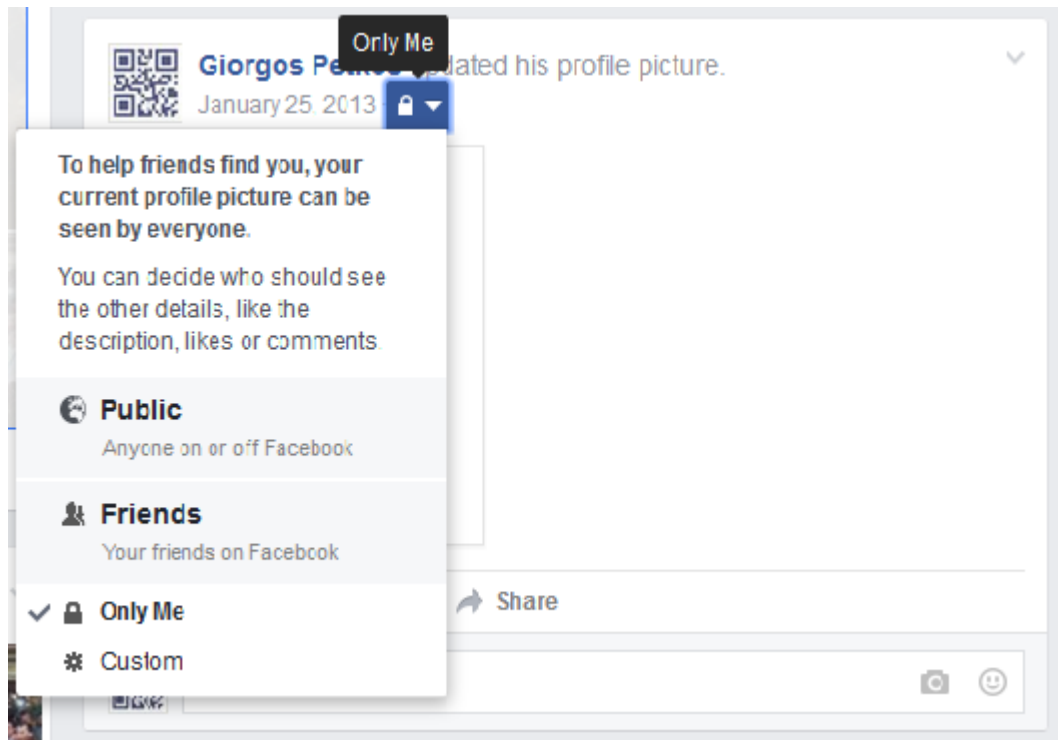
These people or lists

Anyone you include here or have on your restricted list won't be able to see this post unless you tag them. We don't let people know when you choose to not share something with them.

Before posting some content on a social network it is important for a person to ask himself the following questions:

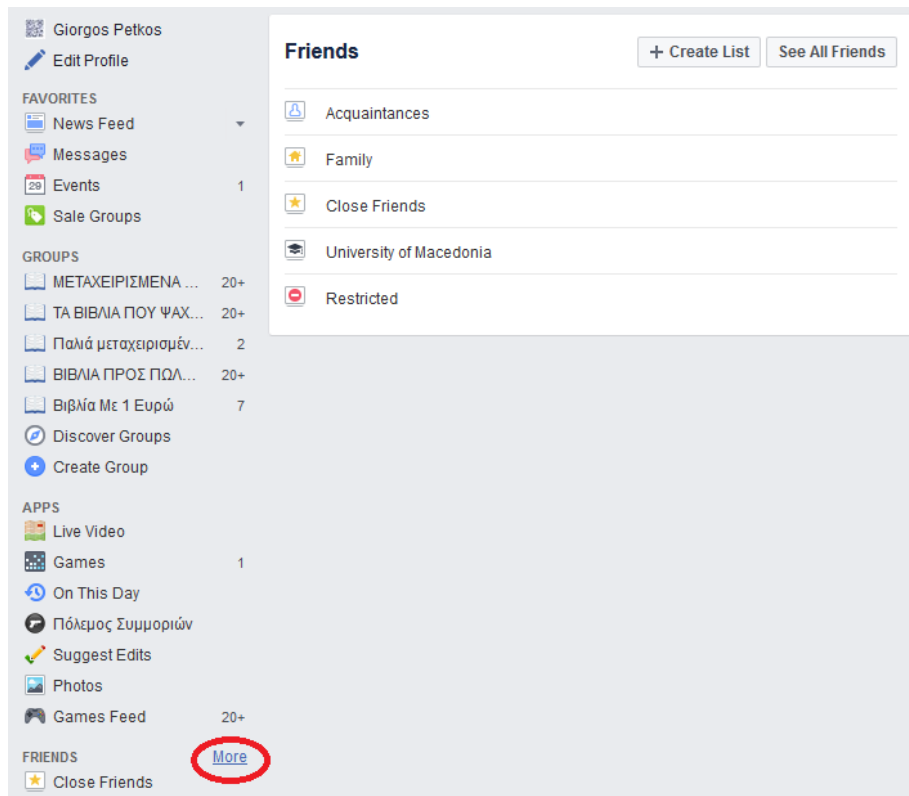
- Is the content I am about to share sensitive or reveals any information about me that I am not be very comfortable about making public?
- Who will be able to see this data? In fact, it is useful for a user to, once in a while, examine their list of friends. This will make them to better perceive their audience. Managing friends is an important issue that we will come back to in the next section.

It is also important to know that privacy settings can also change after some piece of content has been posted. This is shown in the next image.



Creating friends' lists

As mentioned in the previous section, when setting the audience for some new or existing piece of content, the user can define a custom set of users that consists of either a newly defined set of users or a previously defined list of users. Indeed, users can group their friends and create lists of friends, according to the type of relationship that they have with them. This can be done by clicking on the 'friends' link that can be found on the left column of their news feed as shown in the following figure.



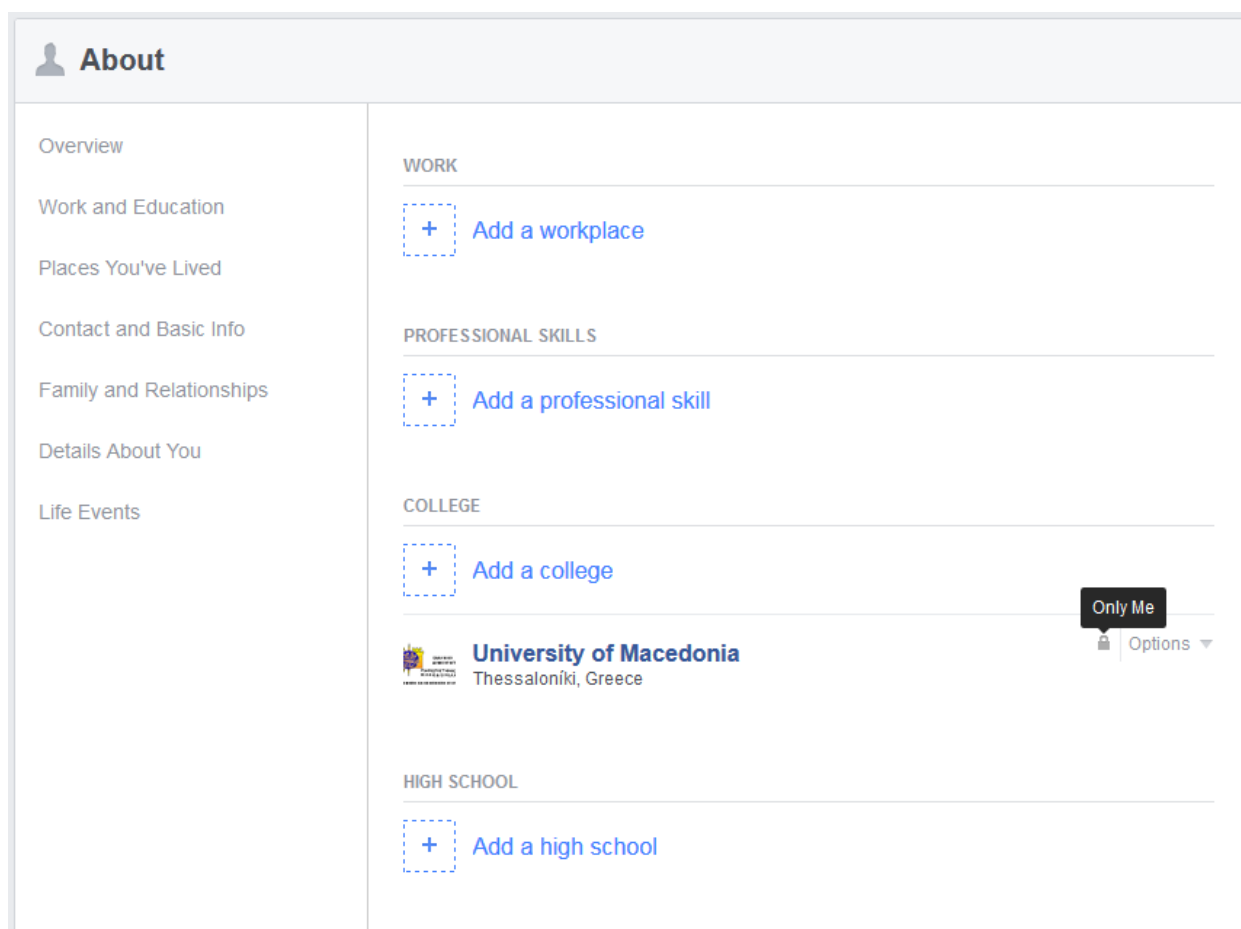
From there, the user can manage their friends lists, e.g. the following figure shows the interface for creating a new list.

 A screenshot of the 'Create New List' dialog box. The dialog box has a title bar with the text 'Create New List' and a close button. Below the title bar, there is a message: 'Create a list of people so you can easily share with them and see their updates in one place.' Underneath the message, there are two input fields. The first is labeled 'List Name' and is empty. The second is labeled 'Members' and contains the placeholder text 'Who would you like to add to this list?'. At the bottom of the dialog box, there are two buttons: 'Cancel' and 'Create'.

Friends lists are a very powerful tool that allows the user to simplify the task of defining their sharing settings. Moreover, friends' lists allow the user to better perceive their audience. That is, by grouping the audience in meaningful sets, it is easier to figure out if some content should not be shared with some other friend.

Managing the disclosure of your profile info

Apart from the disclosure of information through posted content, users also often provide explicit profile information to the social network service that is sometimes disclosed without the user being aware of it. This includes information, such as their mobile phone number, their education, the members of their family, etc. Sometimes also, such information is provided by other users. Nevertheless, the user may prefer to avoid sharing such information. In order to change the sharing settings of profile information of their profiles, users must go to the 'about' section of their profile. As shown in the following image, the different parts of the profile of the user are listed on the left.



The user can then select each category, examine the information under it and change the sharing settings. In the example shown in the image above for instance, information about the college attended by the user is only visible to him / her.

Images privacy

Sharing settings of images in Facebook is also straightforward but has a couple of caveats. The basic point is that photos privacy can be controlled on a per album basis, that is, the user can define sharing settings similar to those of posts for each of their photo albums. One first caveat is that the profile image is always public. This is shown in the following image, in

which it is mentioned that adjusting the audience of the profile image does not change the privacy settings of the image itself, it just changes the access to the comments, likes and description of the image (the image itself is always public). The same holds for the cover photo at the top of the page.

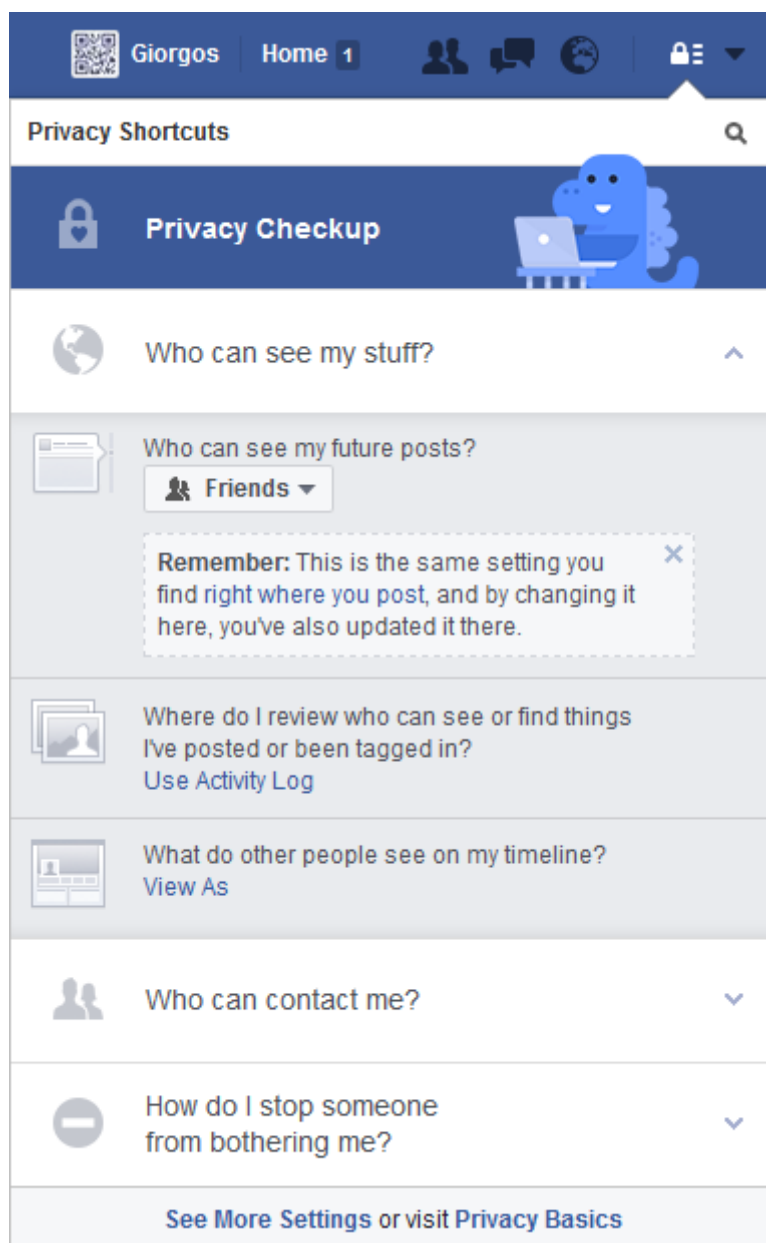


Nevertheless, the user can change the privacy of all other photos in the 'Cover photos' and 'Profile Pictures' albums.

Moreover, unlike other photo albums you create, you can choose an audience for individual photos in your Timeline Photos and Mobile Uploads albums. Each time you post a new photo, you choose who sees that photo using the audience selector.

Examining the activity log

In this and the following sections, we look at some tools offered by Facebook that assist the definition of sharing settings. The first one is the activity log that summarizes recent activity by the user. In order to access it, click on the 'Privacy Shortcuts' icon at the main menu at the top of the page, as shown on the following picture.



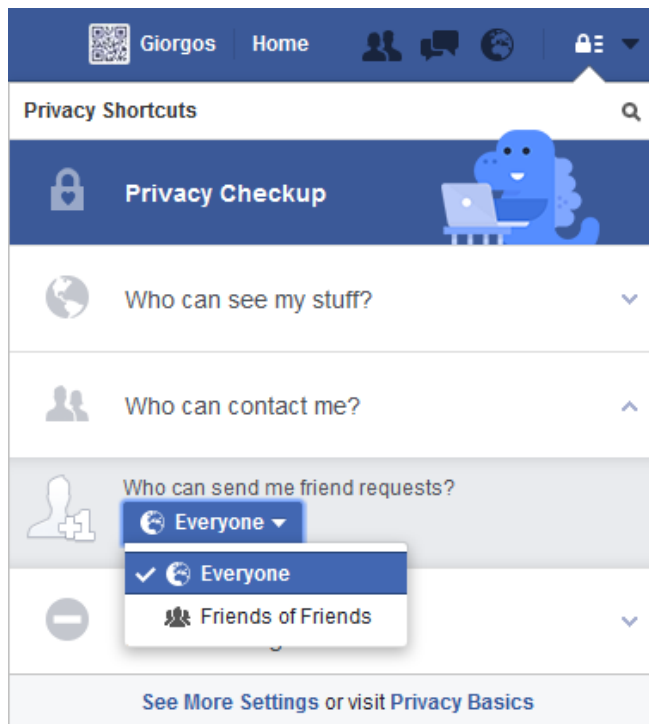
When clicking on the activity log link, the user is shown a list similar to the one shown in the next figure, through which the user can control if the relevant activity will be shown on the user's timeline or not.

View profile as seen by other users

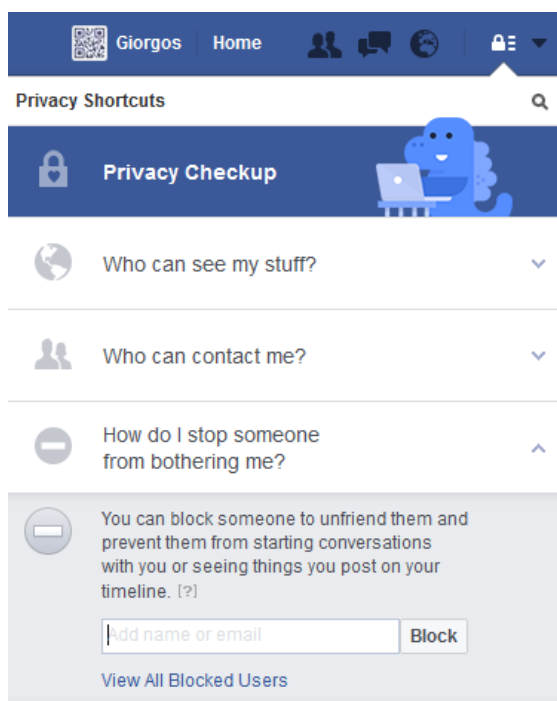
Another very useful tool that is offered by Facebook allows the users to examine their profiles as seen by other users. This function is again accessible through the 'Privacy Shortcuts' icon at the main menu at the top of the page. Once clicked, the user can select a friend (or just select 'public') and accordingly see how his / her profile looks like. This is shown in the following figure.

Blocking other users

Facebook also allows users to protect their privacy by selecting who may contact them. As shown on the following figure, the two options are 'everyone' and 'friends of friends'.



Moreover, users can opt for completely ignoring specific users, as shown on the following figure.



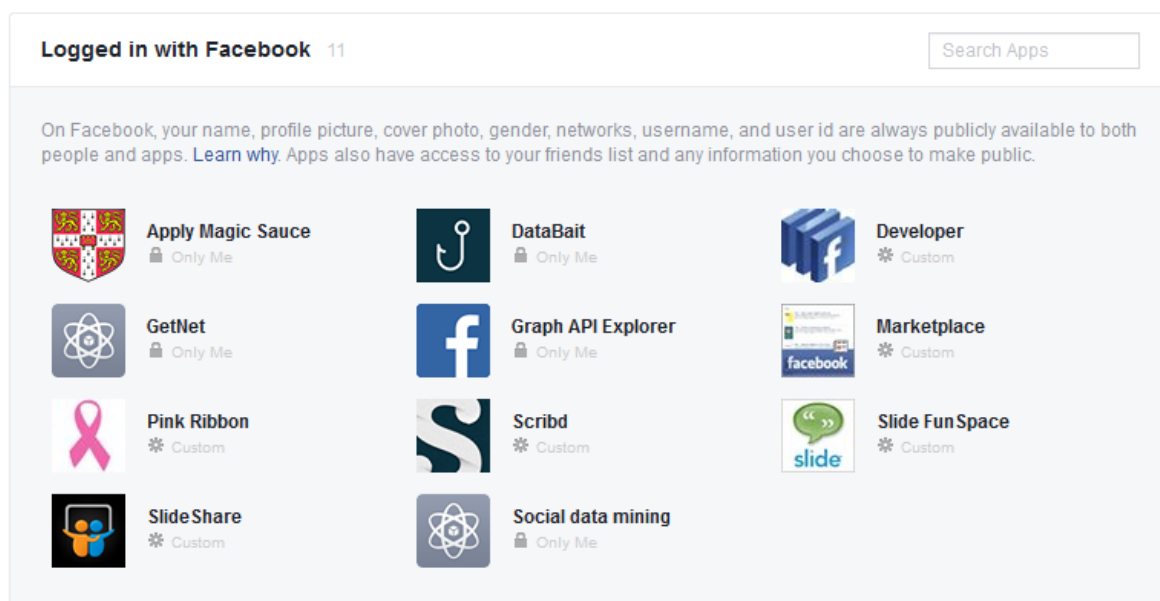
Applications

Facebook applications, like DataBait, obtain access to your Facebook data. That is, when the application is approved by the user, the user grants access rights to the application and the application can then retrieve data related to the user directly from Facebook. The particular access rights obtained by the application determine exactly which data the application can retrieve, e.g. the list of friends, posts, etc. In some cases, the application also has the right to post on the user's profile.

In some cases, applications may be a major threat for the disclosure of personal information. It is not unusual that an application is installed and obtains access to the user's data without the user realizing it. This may happen for instance with some sort of phishing, where the user simply clicks on a random link or button that resembles a like button.

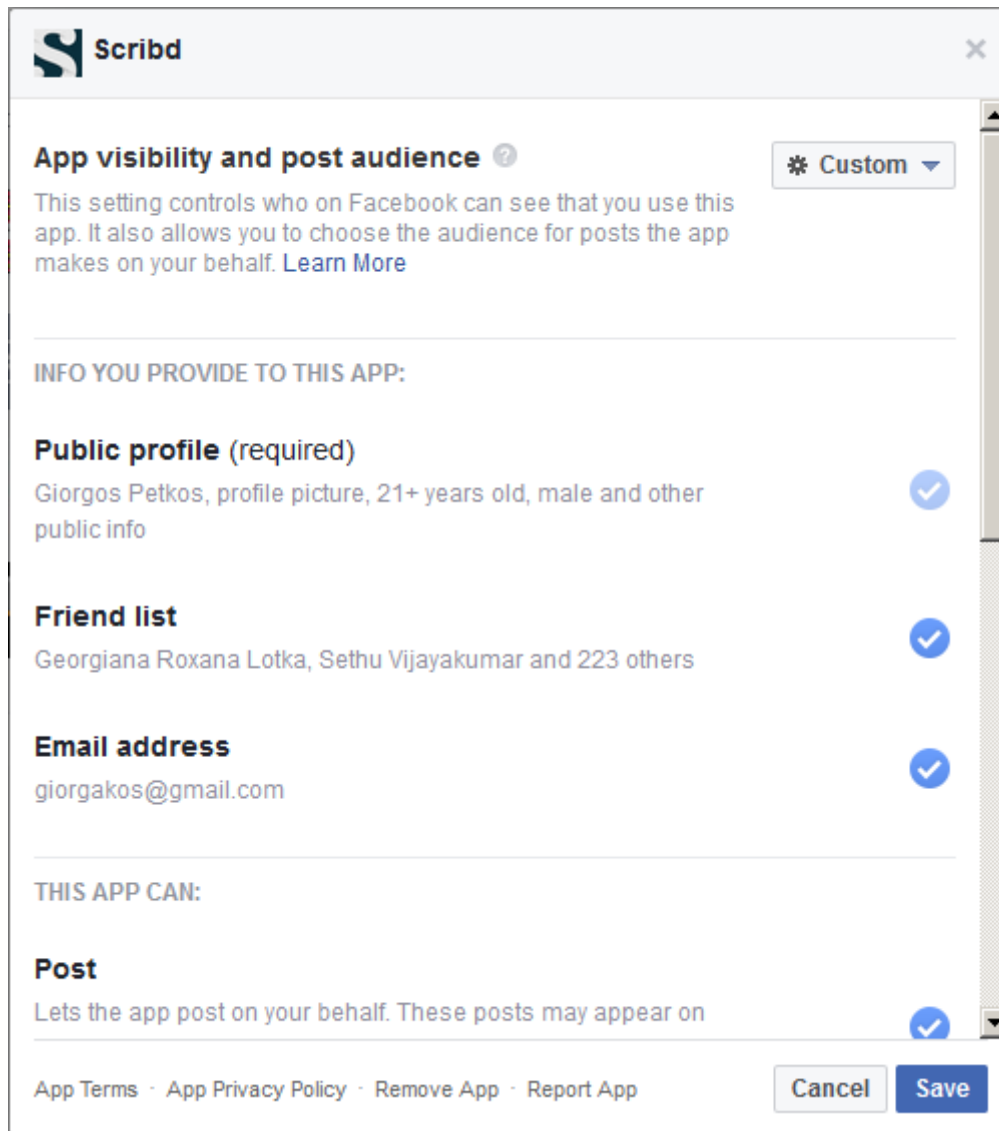
It is advisable that users regularly check the list of their applications and examine the access rights that the applications have. In order to do this, click on 'Apps' on the column on the left of your news-feed and then click on settings. This will show a list of applications similar to that in the figure below.

App Settings

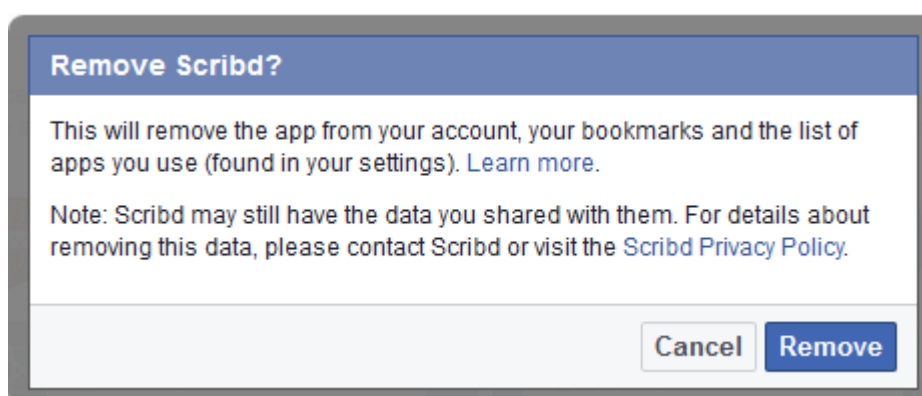


The screenshot shows the 'App Settings' page on Facebook. At the top, it says 'Logged in with Facebook' followed by the number '11'. There is a search bar labeled 'Search Apps'. Below this, a notice states: 'On Facebook, your name, profile picture, cover photo, gender, networks, username, and user id are always publicly available to both people and apps. Learn why. Apps also have access to your friends list and any information you choose to make public.' The main content is a grid of application cards. Each card displays the app's icon, name, and a lock icon indicating its privacy setting. The applications listed are: Apply Magic Sauce (Only Me), DataBait (Only Me), Developer (Custom), GetNet (Only Me), Graph API Explorer (Only Me), Marketplace (Custom), Pink Ribbon (Custom), Scribd (Custom), Slide FunSpace (Custom), SlideShare (Custom), and Social data mining (Only Me).

When you put your mouse over an application, you will see an 'Edit Settings' button. Clicking on it you will see the details of the application and the access rights that it has, just like in the figure below.



Regardless of any changes to the settings of an application though, it is important to keep in mind that once an application gets access to your data, it is unclear what happens with it and control over it may be permanently lost. For instance, please see the following message that is displayed when removing an application.



Final guidelines

This tutorial concludes with a list of some guidelines that should be kept in mind with respect to information disclosure in the social networks:

- Institutional privacy, that is, the disclosure of information to the social network service should not be overlooked. It is not always clear how the social network handles the data.
- Once some content is posted, complete control over it is lost. It is not clear whether the social network ever actually deletes any content, regardless of the fact that you may have deleted it from your profile. This poses further concerns about institutional privacy.
- Moreover, once some content is posted, it may very easily be shared or copied by other users, thus reducing the amount of control that the user has over the content.
- Inferred data is important. More things may be disclosed than one may first think. In addition, the social network can produce very elaborate inferences as it has in its disposal a huge amount of data.
- Content posted about you from other users, incidental data, may reveal a lot about you. In some cases, the user can delete the content (if it is posted in their own profile) or untag themselves. It is important for users to keep track of notifications about other people tagging them and carefully examining posts by other users on their profiles.
- Applications are a major source of information leak. Users should regularly examine the applications that they are using and the access rights that they have.
- It is important to be comfortable with the privacy tools offered by the social network. Often the tools or settings may change without much notice and the user should be constantly try to follow any changes.
- Take advantage of third party tools, like DataBait, that can help you to better control the disclosure of your information!

Annex 4 – Web Plugin URL Mapped Domains

The following table presents the sample data of URLs mapped to specific user attribute.

URL	Dimension	Attribute
pitchfork.com	Hobbies	music
soundcloud.com	Hobbies	music
teslamotors.com	Hobbies	motor sports
imdb.com	Hobbies	movies/series
kinopolis.be	Hobbies	movies/series
netflix.com	Hobbies	movies/series
vier.be	Hobbies	movies/series
amazon.com	Hobbies	shopping
paypal.com	Hobbies	shopping
sporza.be	Hobbies	sports
aliexpress.com	Hobbies	shopping
amazon.co.uk	Hobbies	shopping
amazon.de	Hobbies	shopping
coolblue.be	Hobbies	shopping
vente-exclusive.com	Hobbies	shopping
1dayfly.com	Hobbies	shopping
collectandgo.be	Hobbies	shopping
e-shop.gr	Hobbies	shopping
ebay.com	Hobbies	shopping
fjallraven.se	Hobbies	shopping
flysas.com	Hobbies	travelling
genius.com	Hobbies	music
groupon.be	Hobbies	shopping
ibood.com	Hobbies	shopping
libelle-lekker.be	Hobbies	cooking
openingsuren.com	Hobbies	shopping
qwertee.com	Hobbies	shopping
smulweb.nl	Hobbies	cooking
solo.be	Hobbies	cooking
songteksten.net	Hobbies	music
svtplay.se	Hobbies	movies/series
trakt.tv	Hobbies	movies/series
travelbird.be	Hobbies	travelling
tv.com	Hobbies	movies/series
villagecinemas.gr	Hobbies	shoppingmovies/series
vimeo.com	Hobbies	movies/series
zalando.be	Hobbies	shopping
zwerfkatinleuven.be	Hobbies	animals
abconcerts.be	Hobbies	music
abebooks.com	Hobbies	reading
acnestudios.com	Hobbies	shopping
addnature.com	Hobbies	shopping
adlibris.com	Hobbies	reading
aegeanair.com	Hobbies	travelling
ah.nl	Hobbies	shopping
airport-pickups-london.com	Hobbies	travelling
airportcars-uk.com	Hobbies	travelling

airporttaxi-uk.co.uk	Hobbies	travelling
allmusic.com	Hobbies	music
allrecipes.com	Hobbies	cooking
amadeus.net	Hobbies	travelling
amazon.ca	Hobbies	shopping
asics.com	Hobbies	sports
automotorsport.se	Hobbies	motor sports
azlyrics.com	Hobbies	music
baby.be	Demographics	has baby
barcelona-tourist-guide.com	Hobbies	travelling
beer-deli.gr	Health	alcohol
bet365.com	Hobbies	betting
bet365.gr	Hobbies	betting
bibliohora.gr	Hobbies	reading
bjee.com	Hobbies	sports
blabbermouth.net	Hobbies	music
blueairweb.com	Hobbies	travelling
bluearena.gr	Hobbies	sports
bmw.be	Hobbies	motor sports
bobdylan.com	Hobbies	music
boeken.com	Hobbies	reading
bokus.com	Hobbies	reading
bookland.gr	Hobbies	reading
books-in-greek.gr	Hobbies	reading
booksmania.gr	Hobbies	reading
bookukoo.gr	Hobbies	reading
bristolairport.co.uk	Hobbies	travelling
brusselsairlines.com	Hobbies	travelling
car.gr	Hobbies	motor sports
contra.gr	Hobbies	sports
decathlon.be	Hobbies	sports
discogs.com	Hobbies	music
discshop.se	Hobbies	music
ebay.be	Hobbies	shopping
essmusic.se	Hobbies	music
finnair.com	Hobbies	travelling
fjallraven.com	Hobbies	shopping
game-solution.be	Hobbies	sports
gamereactor.se	Hobbies	video games
goodreads.com	Hobbies	reading
graspop.be	Hobbies	music
graspopmetalmeeting.blogspot.be	Hobbies	music
greekbooks.gr	Hobbies	reading
hardmusic.gr	Hobbies	music
icefilms.info	Hobbies	movies/series
intersport.gr	Hobbies	sports
kasetophono.com	Hobbies	music
kinopolis.com	Hobbies	movies/series
lemoni.gr	Hobbies	reading
livescore.com	Hobbies	sports
luleahockey.se	Hobbies	sports
lyricsfreak.com	Hobbies	music
metal-archives.com	Hobbies	music

metrosport.gr	Hobbies	sports
nakasbookbazaar.gr	Hobbies	reading
napapijri.com	Hobbies	shopping
nintendo.com	Hobbies	video games
opap.gr	Hobbies	betting
papasotiriou.gr	Hobbies	reading
patakis.gr	Hobbies	reading
parts.gr	Hobbies	motor sports
pelgrimroutes.nl	Hobbies	hiking
peru.travel	Hobbies	travelling
pietmoodshop.be	Hobbies	shopping
playfuturama.com	Hobbies	video games
playstation.com	Hobbies	video games
plus4u.gr	Hobbies	shopping
politeianet.gr	Hobbies	reading
ponomusic.com	Hobbies	music
primaverasound.com	Hobbies	music
primaverasound.es	Hobbies	music
protoporia.gr	Hobbies	reading
rocking.gr	Hobbies	music
sas.se	Hobbies	travelling
scubadiving.com	Hobbies	sports
sff.gr	Hobbies	reading
skyscanner.net	Hobbies	travelling
sport24.gr	Hobbies	sport24
spotify.com	Hobbies	music
stephenking.com	Hobbies	reading
stercinemas.gr	Hobbies	movies/series
tabiblia.gr	Hobbies	reading
tabstabs.com	Hobbies	music
talassadiving.com	Hobbies	sports
teleioskiklos.gr	Hobbies	reading
tripadvisor.be	Hobbies	travelling
tripadvisor.com	Hobbies	travelling
tripadvisor.nl	Hobbies	travelling
trivago.be	Hobbies	travelling
turkishairlines.com	Hobbies	travelling
uefa.com	Hobbies	sports
ultimate-guitar.com	Hobbies	music
volkswagenag.com	Hobbies	motor sports
volkswagengroup.se	Hobbies	motor sports
vueling.com	Hobbies	travelling
wintersporters.be	Hobbies	sports
wizzogames.com	Hobbies	video games
wolfclub.be	Hobbies	music
wolfordshop.be	Hobbies	shopping

References

- Akbani, R., Kwek, S., & Japkowicz, N. (2004, September). Applying support vector machines to imbalanced datasets. In *European conference on machine learning* (pp. 39-50). Springer Berlin Heidelberg.
- Branco, P., Torgo, L., Ribeiro, R. A survey of predictive modelling under imbalanced distributions. *arXiv preprint arXiv:1505.01658* (2015).
- Buschek, D., Bader, M., von Zezschwitz, E., Luca, A. Automatic privacy classification of personal photos. In *Human-Computer Interaction – INTERACT, 2015*.
- Domingos, P. (1999, August). Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 155-164).ACM.
- Drummond, C., & Holte, R. C. (2003, August). C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on learning from imbalanced datasets II* (Vol. 11).
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321-357.
- Elkan, C. (2001, August). The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence* (Vol. 17, No. 1, pp. 973-978). LAWRENCE ERLBAUM ASSOCIATES LTD.
- Fan, R., Chang, K., Hsieh, C., Wang, X., Lin, C. (2008). LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learnign Research*.
- Japkowicz, Nathalie, and Shaju Stephen. "The class imbalance problem: A systematic study." *Intelligent data analysis* 6.5 (2002): 429-449.
- Kosinski, M., Stillwell, D., Graepel, T. Private traits and attributes are predictable from digital records of human behavior. *Proceeding of the National Academy of Sciences*, 110 (15): 5802-5805, 2013.
- Madejski, M., Johnson, M., Bellovin, S. (2012).A study of privacy settings errors in an online social network.PERCUM 2012.
- Maloof, M. A. (2003). Learning when data sets are imbalanced and when costs are unequal and unknown. In *ICML-2003 workshop on learning from imbalanced data sets II* (Vol. 2, pp. 2-1).
- Menon, A. K., Narasimhan, H., Agarwal, S., Chawla, S. (2013, June). On the Statistical Consistency of Algorithms for Binary Classification under Class Imbalance. In *ICML* (3) (pp. 603-611).
- Nikolaou, N., Edakunni, N., Kull, M., Flach, P., & Brown, G. (2016). Cost-sensitive boosting algorithms: Do we really need them?. *Machine Learning*,104(2-3), 359-384.
- Petkos, G., Papadopoulos, S., Kompatsiaris, Y. (2015).PScore: a framework for enhancing privacy awareness in online social networks. *MFSec 2015*.
- Spyromitros-Xioufis, E., Petkos, G., Papadopoulos, S., Heyman, R., Kompatsiaris, Y. (2016a).Perceived vs. actual predictability of personal information in social networks.*Internet Science 2016*.

- Spyromitros-Xioufis, E., Papadopoulos, S., Popescu, A., Kompatsiaris, Y. (2016b). Personalized Privacy-aware Image Classification. Proc. International Conference on Multimedia Retrieval (ICMR), New York, USA, June 6-9, 2016.
- Tu, Han-Hsing, and Hsuan-Tien Lin. "One-sided support vector regression for multiclass cost-sensitive classification." Proceedings of the 27th International Conference on Machine Learning (ICML-10). 2010.
- Weiss, Gary M. "Mining with rarity: a unifying framework." ACM SIGKDD Explorations Newsletter 6.1 (2004): 7-19.
- Zhou, Z. H., & Liu, X. Y. (2006). Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge and Data Engineering, 18(1), 63-77.