User Classification Based on Public Reddit Data

Robin De Pril  
Student number: 01305560

Supervisors: Prof. dr. ir. Bart Dhoedt, Prof. dr. ir. Pieter Simoens  
Counsellors: ing. Sam Leroux, Pieter Van Molle

Master's dissertation submitted in order to obtain the academic degree of  
Master of Science in Computer Science Engineering

Academic year 2018-2019
Permission for use of content

“The author gives permission to make this master dissertation available for consultation and to copy parts of this master dissertation for personal use. In the case of any other use, the copyright terms have to be respected, in particular with regard to the obligation to state expressly the source when quoting results from this master dissertation.”

Robin De Pril, 2019
Preface

I want to thank my supervisors and counsellors for their valuable feedback, especially my weekly meetings with ing. Sam Leroux have been crucial to shape this thesis.

Furthermore, I want to thank my parents and partner. Words cannot express my gratitude for their support.

Lastly, I also want to thank some people from the internet, which I will not name, to protect both theirs and my own privacy. They too provided a source of inspiration and support.
Abstract

Reddit is a website with a significant privacy culture. In this work, we investigate whether it is possible to deduce personal information using public data on the Reddit website. More specifically, we focus on age, gender, and nationality. To this end, we first create six labeled datasets. Labels for these datasets are gathered using two techniques using Reddit specific information. This on its own can already be of interest to privacy-minded Reddit users. Next, we construct features for Reddit users based on their comments. We explore two types of features: on one hand we look at the textual content of a user’s comments, and the other hand we construct a graph of Reddit users. We apply a number of baseline methods on this data and compare them to a Graph Convolutional Network. We conclude it is indeed possible to correctly classify a majority of users, but the Graph Convolutional Network did not manage to outperform our strongest baseline method.

Keywords

Privacy, Reddit, User Classification, Machine Learning, Graph Convolutional Networks
User Classification Based on Public Reddit Data

Robin De Pril

Supervisors: Prof. dr. ir. Bart Dhoedt, Prof. dr. ir. Pieter Simoens
Counsellors: ing. Sam Leroux, Pieter Van Molle

Abstract—Reddit is a website with a significant privacy culture. In this work, we investigate whether it is possible to deduce personal information using public data on the Reddit website. More specifically, we focus on age, gender, and nationality. To this end, we first create six labeled datasets. Labels for these datasets are gathered using two techniques using Reddit specific information. This on its own can already be of interest to privacy-minded Reddit users. Next, we construct features for Reddit users based on their comments. We explore two types of features: on one hand we look at the textual content of a user’s comments, and the other hand we construct a graph of Reddit users. We apply a number of baseline methods on this data and compare them to a Graph Convolutional Network. We conclude it is indeed possible to correctly classify a majority of users, but the Graph Convolutional Network did not manage to outperform our strongest baseline method.

Index Terms—Privacy, Reddit, User Classification, Machine Learning, Graph Convolutional Networks

I. INTRODUCTION

The title of this paper reads “User Classification Based on Public Reddit Data”. The goal of this introduction is twofold, we first provide a short introduction to what Reddit is and introduce some Reddit-specific terminology. Secondly, we formulate the goal of this thesis and lay out the structure of this document.

A. What is Reddit?

Reddit is an American social news site, link aggregator, and discussion platform. It was founded in 2005. As of February 2019, it is the 17th most popular website worldwide and the 6th most popular in the United States according to the Alexa web ranking [1].

Reddit is organized around communities, called subreddits. Each subreddit has its own topic, internal culture, and community rules. The name of a subreddit is always preceded by “r/” or “/r/”.

Users on the site are also referred to as a Redditors. A username is preceded by “u/” or “/u/”. To become a user on the site, one has to create an account. Unlike other social networks such as Facebook, a user does not need to register using their real name. While Reddit does encourage users to provide an email address for account recovery, it is not strictly required. As a result, the website has a strong privacy culture.

Users can post content to a subreddit and subsequently leave comments on these posts. The majority of discussions happen on public subreddits. This means that anyone can not only view, but also download and analyze, all the discussions on these subreddits, even without a Reddit account.

B. Goal of this document

This thesis researches which personal information can be deduced from publicly available activity of Reddit users. More specifically, we apply various machine learning methods on comments made by Reddit users. One relatively novel technique we investigate is Graph Convolutional Networks (GCNs). The types of personal information we try to deduce are nationality, gender, and age.

Our contribution includes the creation of six labeled datasets. Labels for these datasets are gathered using two techniques specific to the Reddit website: subreddits with user flairs and subreddits that encourage or enforce standardized post titles.

We hope this work can be helpful to review GCNs on one hand, and on the other hand help raise awareness among privacy-conscious Reddit users about the personal data they make publicly available.

The remainder of this document is split into five sections. First, we provide an introduction to machine learning, review the literature regarding GCNs and review previous attempts at classifying internet users. Secondly, we explore different ways to create labeled datasets from Reddit. Such a dataset consists of Reddit users mapped to a piece of personal information, such as nationality, gender, or age. For each user we also create a set of features based on their public comments. Next, we train and evaluate various baseline machine learning models on our dataset. Thirdly, we apply a GCN model on the datasets and compare the results with those obtained using the baseline methods. Finally, we conclude this thesis and think about future work in this area.

II. BACKGROUND AND RELATED WORK

This chapter reviews the literature and summarizes works related to this thesis. First, we construct an overview of machine learning models, building up to Graph Convolutional Networks. Next, we review literature related to classifying internet users.

A. Graph Convolutional Networks

In the next chapter, we will construct a graph where the nodes correspond to Reddit users, with each node having a label. We will then set some of these labels to unknown. The job of the machine learning model is filling in these missing labels. This is called semi-supervised learning [2]. To this end, we explore Graph Convolutional Networks (GCNs).

GCNs try to extend the ideas of Convolutional Neural Networks (CNNs). CNNs are a family of Artificial Neural
Networks often used for structured graph data, like images. GCNs on the other hand operate on arbitrary graph data. In a classic CNN, a node (read: pixel) is transformed using a weighted sum of itself and its neighbors (read: nearby pixels) [3]. However, in an arbitrary graph, a node (read: Reddit user) has a variable amount of neighbors, and these nodes do not have a particular order [4]. We will therefore need to find a more general way to combine a node’s features with those of other nodes in the graph.

We will use the GCN model proposed by Kipf and Welling [2]. Due to its impressive performance, the model by Kipf and Welling is often used by the research community as the GCN benchmark [4]. We will explain it in more detail here.

Let’s first introduce some notation. A graph is defined using an $N \times N$ adjacency matrix $A$, with $N$ the number of nodes. In the case of machine learning, each node can have multiple features. These are stored in a $N \times D$ matrix $X$, with $D$ the number of features. Finally, the labels are stored one hot encoded as a binary $N \times E$ matrix $Z$, with $E$ the number of number of output classes. $H^{(l)}$ is the output of layer $l$. $H^{(0)} = X$ and $H^{(L)} = Z$.

The most basic graph convolution is replacing each node by the sum of itself and its direct neighbors. In matrix notation, this corresponds to the following, with $I$ the identity matrix with the same dimensions of $A$:

$$(A + I)H^{(l)}$$

Let’s denote $(A + I)$ as $\tilde{A}$ for convenience. We now replaced each node by the sum of itself and its direct neighbors, however it is intuitively clear that it might be better to take an average instead. This can be achieved by normalizing $\tilde{A}$ as follows: $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$, where $\tilde{D}$ the diagonal node degree matrix of $\tilde{A}$. Kipf and Welling however propose to use symmetric normalization instead. Here we replace $\tilde{A}$ with $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$. Let’s denote this symmetrically normalized matrix as $\hat{A}$. The result of $\hat{A}H^{(l)}$ is not a simple average of a node and its neighbors, but a weighted sum, where nodes that have few other neighbors will have a higher contribution. Including weights and an activation function completes the propagation rule proposed by Kipf and Welling:

$$H^{(l+1)} = \sigma(\hat{A}H^{(l)}W^{(l)})$$

An illustration of this process is provided on Figure 1. In this example, we have three input features for each graph node, depicted by three versions of the graph on the left-hand side. The graphs are convoluted, meaning the feature values of each node of the graph are combined with itself and its neighbors. This corresponds to the $\hat{A}H^{(l)}$ operation and is graphically illustrated inside the pale blue boxes. Next, the different features are combined, similar to an MLP, using weights and an activation function. This is illustrated by the arrows between the pale blue boxes. In this example, we have one hidden layer with 3 elements. The process is repeated to obtain an output layer. In this example, there are two different output labels, represented by the two graphs on the right-hand side. Nodes that have the highest value in the top graph of the output layer will be assigned the first label. Nodes that have the highest value in the bottom graph the second one. The weights are trained similar to other Neural Networks, using the backpropagation algorithm [5].

### B. Online privacy and Reddit

Trying to recover personal information from people based on their behavior online is not a new phenomenon. Early examples from the the mid 2000s classify users based on blog posts [6], [7]. In the early 2010s we see examples using data from the popular microblogging platform Twitter [8], [9]. And more recently, research, that like this paper, uses data from Reddit [10].

Besides age, gender, and nationality/location, these papers also include research into political orientation, ethnicity and brand loyalty. Labels were sourced in various ways: sometimes a standardized way of reporting personal information was available, sometimes a regular expression was used to parse a label, and other times manual classification was performed. Features most often are based on word-count, often supplemented with some creative, platform-specific features.

### III. Data collection and analysis

The first challenge is gathering data from the Reddit website. Reddit has a public API, however rate limitations make it troublesome to use this API to collect large historic datasets. Luckily, read-only mirrors of the API exist without these limitations. In this thesis, the mirror at pushshift.io was used. To ensure consistency and reproducibility, all queries have a parameter stating the data must be from before Oct 1st 2018, at 00:00:00 UTC.

First, we will discuss how we can collect Reddit users for which we can find a label. Next, we will download for each user their 500 most recent comments and use those to create features from each user.

#### A. Collection and labeling

In this thesis, we explore two methods specific to the Reddit website to source these labels.

The first method uses the user flairs of a subreddit. In some communities, users can add a piece of extra information next to their username, this is called a user flair. Three such subreddits that support user flairs are r/Europe, r/AskMen, and r/AskWomen. On r/Europe, users can optionally select a flag to show up next to their username, and on r/AskMen and r/AskWomen, people can report their gender.

The second method to source labels exploits subreddits that enforce or encourage standardized post titles. For example, the subreddits r/relationship_advice and r/r4r encourage users to include their age and gender in their posts.

In total, we have created six datasets. One on nationality, where labels are sourced from r/Europe. Three on gender: one with labels from r/AskMen and r/AskWomen, one from [1]https://pushshift.io/api-parameters/
Table I: Number of users per country in the r/Europe dataset.

<table>
<thead>
<tr>
<th>Name</th>
<th>Abbr.</th>
<th>No. of users</th>
</tr>
</thead>
<tbody>
<tr>
<td>United States</td>
<td>UNSA</td>
<td>890</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>UKGB</td>
<td>870</td>
</tr>
<tr>
<td>Germany</td>
<td>GERM</td>
<td>805</td>
</tr>
<tr>
<td>Sweden</td>
<td>SWED</td>
<td>796</td>
</tr>
<tr>
<td>The Netherlands</td>
<td>NETH</td>
<td>778</td>
</tr>
<tr>
<td>Finland</td>
<td>FINL</td>
<td>735</td>
</tr>
<tr>
<td>France</td>
<td>FRAN</td>
<td>474</td>
</tr>
<tr>
<td>Ireland</td>
<td>IREL</td>
<td>466</td>
</tr>
<tr>
<td>Denmark</td>
<td>DENK</td>
<td>459</td>
</tr>
<tr>
<td>Poland</td>
<td>POLA</td>
<td>444</td>
</tr>
<tr>
<td>Portugal</td>
<td>PORT</td>
<td>421</td>
</tr>
<tr>
<td>Romania</td>
<td>ROMA</td>
<td>352</td>
</tr>
<tr>
<td>Norway</td>
<td>NORW</td>
<td>340</td>
</tr>
<tr>
<td>Italy</td>
<td>ITAL</td>
<td>285</td>
</tr>
<tr>
<td>Belgium</td>
<td>BELG</td>
<td>283</td>
</tr>
<tr>
<td>Austria</td>
<td>ASTR</td>
<td>276</td>
</tr>
<tr>
<td>Greece</td>
<td>GREE</td>
<td>247</td>
</tr>
<tr>
<td>Turkey</td>
<td>TURK</td>
<td>245</td>
</tr>
<tr>
<td>Czechia</td>
<td>CZEK</td>
<td>220</td>
</tr>
<tr>
<td>Canada</td>
<td>CANA</td>
<td>211</td>
</tr>
</tbody>
</table>

r/relationship_advice, and one from r/r4r. And two on age: one with labels from r/relationship_advice, and one from r/r4r.

1) r/Europe: r/Europe is a subreddit intended for content related to the European continent.

The first step in order to gather a set of labeled users is downloading the 500 000 most recent comments on the subreddit. From these comments, we found 56 856 unique authors. Of these users, 16 499 have a user flair. We choose to only retain the 20 most popular flairs that correspond to a country. This leaves us with 9597 labeled users, the number of users per label are shown on Table I.

2) r/AskMen and r/AskWomen: r/AskMen and r/AskWomen are two communities with the goal of asking questions to the general male, resp. female, population of Reddit.

Following the same procedure as for the r/Europe dataset, we find 5419 users that picked “Male” as a userflair and 3670 users that picked “Female”.

3) r/relationship_advice: As the name implies, r/relationship_advice is a community where users can ask advice regarding their personal relations. Someone who seeks advice will create a post explaining their situation. Users will often mention the age and gender of the different actors in the title of their post. Since r/relationship_advice does not provide user flairs like the previously mentioned subreddits, we will try to deduce the labels from these titles instead. Some examples of posts on the subreddit:

- “My roommate [21F] keeps cooking food I [19F] cannot stand the smell of...how do I address it?”
- “Me (18M) Not sure what to do about relationship with (18F) girlfriend”

First, 50 000 posts were downloaded. Next, a regular expression was created that attempts to catch the mentions of the author’s gender and age, while having a minimal amount of false positives. Using trial-and-error, this is the resulting regular expression, which is applied in a case-insensitive manner:

```regex
(i\s\d\d\d\d)\s*(\w\b\s\w\b\s\w\b\s\w\b\s\d\d\d\d)\s*(\w\b\s\w\b\s\w\b\s\w\b\s\d\d\d\d)\s*(\w\b\s\w\b\s\w\b\s\w\b\s\d\d\d\d)\s*(\w\b\s\w\b\s\w\b\s\w\b\s\d\d\d\d)
```

This expression might look daunting, but it basically looks for phrases like “I”, “I’m”, “I am”, “I’m a”, “Me”, and “My”, followed by age and gender expressed in the ways shown in the examples above. As such, we find the age and gender of 11 989 unique users. If we look at the activity of these users, we see that a significant portion of these users have made few comments. We choose to discard users who make less than 250 comments. 1905 users remain: 1251 men and 654 women. For age, we choose to split the users in two bins, and pick the threshold in such a way that the bins are as equal in size as possible. For this dataset, the most equal split is at 24, with 900 users being 24 or older, and 1005 being younger.
than 24.

4) r/r4r: r/r4r, short for “Redditor for Redditor”, is a subreddit where users can place personal ads, for example to find a friend or partner. The subreddit enforces the following title format: `<age>\[<r4r>\]<location>-<title>`, where the user has to fill in the parts between angled brackets. `<r4r>` is replaced to indicate the user’s own gender and the gender of the person they are looking for, e.g. a woman looking for a man would write `f4m`. Sadly, location is not expressed in a systematic way, making it non-trivial to use this to create location labels.

We follow a similar strategy like with the r/relationship_advice dataset, using the following regular expression:

```
((0-9)+)\s*\[(\[!4\]\]+)4(\[!4\]\]+)\]
```

We again notice many users with low actively and discard users with less than 250 comments. The result is a gender dataset with 3126 users, of which 2555 male, and age dataset with 3561 users, of which 1690 users 26 years old or older.

B. Feature extraction

As mentioned earlier, we download up to 500 comments per user. We process these comments to create two types of features: textual features on one hand, and graph-based features on the other.

1) Textual features: We extract textual features from the comments by first concatenating all comments made by a user with a tf-idf vectorizer from the scikit-learn machine learning library [11]. Additionally, we choose to remove English stop words and only consider words that are used by at least 5 different users.

2) Graph-based features: Besides interpreting comments purely textual, we can also take their context into account and use them to create a graph structure between the different users.

Each user is represented by a node. For each comment of a user, we know the post the discussion belonged to. If two users take part in the discussion belonging to the same post, we draw an edge between their respective nodes. The weight of the edge is calculated as follows, with $S_{i,j}$ each conversation shared by users $i$ and $j$:

$$\text{weight}_{\text{edge}}(i,j) = \sum_{c \in S_{i,j}} \frac{1}{\text{No. of comments } c}$$

The assumption is that there is a relationship between a user and its neighbors in this graph. The intuition underpinning this hypothesis is that people who have the same gender, age or nationality have a higher chance of having similar interests and thus a higher chance taking part in the same conversations. A simplified version of the graph created using the r/Europe dataset is shown on Figure 2. We can already visually detect some clusters of nationalities in this visualization.

IV. BASELINE METHODS

We have constructed five baseline methods. Two methods that only look at the textual properties of a user’s comments, one method that uses the graph representation of the users, and two methods that combine textual properties with graph information.

For the r/AskMen and r/AskWomen dataset, we only consider features sourced outside those subreddits. This is done because users with the same gender would be close together in the graph just because of the way the data is collected and the nature of these two subreddits.

For each of the datasets, we randomly select 25% of the samples as testing data and the remaining 75% as training data.

As mentioned earlier, two models were trained on textual tf-idf features alone. More concretely, these are a Random Forest (RF) and a Multilayer Perceptron (MLP). For both, we use implementations from the scikit-learn machine learning library [11].

The graph based model is a K-Nearest Neighbors (KNN) model. This model classifies a node by looking at the K labeled nodes with the shortest distance to the node, and chooses the most prevalent label among these “neighbors”.

The fourth and fifth baseline consist of two ensembles: one combining the Random Forrest and the KNN algorithm, and one the MLP with the KNN algorithm. The ensembles are constructed as follows: we apply each model separately, but instead of just returning the predicted label, we ask each of the models to return the probability they attach to each label. The probabilities of these models are then fed into a Logistic Regression classifier to determine the final classification.

A. Results

The results of the different methods are summarized in Table II.

Looking at these results, we remark that the KNN model overall performs the worst, in the case of the age datasets even random. It is also the least advanced model of them all. We notice that creating the ensemble of the RF or MLP model with the KNN model results in a similar score than the models without the KNN model added. Overall, a MLP model trained on the tf-idf features extracted from a user’s comments performs the best.

It is also possible to request the scikit-learn Random Forest implementation which features are the most important. For the r/Europe dataset, we mostly see words in non-English languages. Reddit has various non-English subreddits, often tied to a specific country. From this, we notice that our model has picked up on this usage of foreign languages as a handy way of classifying users by nationality. For the gender datasets, we recognize words that could be stereotypically associated with men, such as words related to gaming, and words related to women, such as “husband”, “boyfriend”, and beauty-related terms. For the age datasets, we recognize words that are associated with certain stages of life, such as “school”, “married”, “work”, “apartment”, and “wife”.
Fig. 2: A visualization of the r/Europe users represented as a graph. This is a simplified version of the graph only containing the edges with weight higher than 1.5. Nodes are colored by nationality.

On Figures 3a and 3b, we see the confusion matrices for r/Europe dataset for the MLP model and the MLP+KNN model. We that in the case of the MLP model, most errors occur for nationalities that share a native language. For example, many Austrians are misclassified as Germans. Similarly, Belgians get confused with Dutch people and to a lesser extend with French people. By supplementing this model with the graph data using the KNN model, we see this model can overcome this hurdle.

V. GRAPH CONVOLUTIONAL NETWORKS

We now test the GCN architecture proposed by Kipf and Welling [2]. This model features a number of hyperparameters: the number of hidden layers, the size of each hidden layer, the learning rate, the dropout rate, the L2 loss, the number of epochs the model should be trained and the number of epoch without improvement for early stopping. We trained the hyperparameters on the r/Europe dataset and then reused those parameters for the other datasets, this is similar to the baselines where we also reused hyperparameters for the different models. The results are included in Table II and the confusion matrix for the r/Europe dataset on Figure 3c.

The GCN does not seem to be working better than our baselines. Overall, it performs worse than the MLP baseline. Looking at the confusion matrix of the r/Europe dataset, we see the common mishaps stemming from nationalities with shared languages. They are less pronounced than in the case of the MLP, but more visible compared to the result of the MLP enhanced with the KNN model.

VI. CONCLUSION

We succeeded in creating six labeled datasets using publicly available Reddit data. This on its own can already be of interest to privacy-minded Reddit users. Next, we also showed that it is possible to classify most users by age, gender, and nationality using their publicly available comments. Applying a Multilayer Perceptron on the tf-idf transform of these comments proves to be a viable approach. We discovered that in some cases, it can be useful to think of the Reddit users as nodes in a graph. For example, when looking at nationality it is interesting to not only look at which words you use, but also at who you talk with. This is particularly helpful to discern between nationalities that share a mother tongue, such as Austrians and Germans. We demonstrated this by creating an ensemble of a Multilayer Perceptron on the tf-idf transform of the comments.
of a user’s comments with a K-Nearest Neighbors algorithm operating on a graph of Reddit users. We compared this to a Graph Neural Network, a model that also takes both these word and graph properties into account. It did not manage to improve upon our rather strong baseline methods.

**REFERENCES**


Contents

List of Figures ......................................................... 15
List of Tables .......................................................... 17

1 Introduction .......................................................... 18
  1.1 What is Reddit? ................................................. 18
  1.2 Goal of this master thesis ..................................... 20

2 Related work ......................................................... 21
  2.1 Machine learning and Artificial Neural Networks ................. 21
    2.1.1 Multilayer Perceptron .................................... 22
    2.1.2 Convolutional Neural Network ........................... 23
    2.1.3 Graph Convolutional Networks ........................... 24
  2.2 Online privacy and Reddit ..................................... 27
    2.2.1 Early examples using blogs (mid 2000s) ................. 28
    2.2.2 Twitter enters the scene (early 2010s) .................. 29
    2.2.3 And now... Reddit (2010s) .............................. 30

3 Data collection and analysis ...................................... 31
  3.1 Collection and labeling ....................................... 32
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.1</td>
<td>Finding your place on r/Europe</td>
<td>32</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Asking the hard questions on r/AskMen and r/AskWomen</td>
<td>33</td>
</tr>
<tr>
<td>3.1.3</td>
<td>Getting help on r/relationship_advice</td>
<td>33</td>
</tr>
<tr>
<td>3.1.4</td>
<td>A/S/L on r/r4r</td>
<td>35</td>
</tr>
<tr>
<td>3.2</td>
<td>Feature extraction</td>
<td>38</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Textual features</td>
<td>40</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Graph-based features</td>
<td>41</td>
</tr>
<tr>
<td>4</td>
<td>Baseline Methods</td>
<td>47</td>
</tr>
<tr>
<td>4.1</td>
<td>Evaluation metrics</td>
<td>47</td>
</tr>
<tr>
<td>4.2</td>
<td>Using the textual content of comments</td>
<td>49</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Random Forest</td>
<td>49</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Multilayer perceptron</td>
<td>50</td>
</tr>
<tr>
<td>4.3</td>
<td>Using graph data</td>
<td>51</td>
</tr>
<tr>
<td>4.4</td>
<td>Combining textual content and graph data</td>
<td>54</td>
</tr>
<tr>
<td>4.5</td>
<td>Conclusion</td>
<td>55</td>
</tr>
<tr>
<td>5</td>
<td>Graph Convolution Networks</td>
<td>61</td>
</tr>
<tr>
<td>6</td>
<td>Conclusion and future work</td>
<td>64</td>
</tr>
<tr>
<td>6.1</td>
<td>Conclusion</td>
<td>65</td>
</tr>
<tr>
<td>Bibliography</td>
<td>66</td>
<td></td>
</tr>
</tbody>
</table>
List of Figures

1.1 A picture of the Reddit website, with some of the elements annotated  

2.1 A visual representation of a Multilayer Perceptron  

2.2 An illustration of how a filter can transform an image in a convolutional layer of a CNN. The displayed filter has learned to recognize top vertical edges.  

2.3 An illustration of a Graph Convolutional Network  

3.1 An example of user flairs on r/Europe: users provide extra information about themselves by picking a flag to display next to their username.  

3.2 Visualization of the regular expression used to extract the age and gender of the authors of posts on r/relationship_advice. Image created using regexper.com, license: CC BY 3.0  

3.3 A histogram showing the age of r/relationship_advice users, zoomed in on the [13 - 50] range, where the vast majority of users are found. Users with more than 250 comments are denoted as active. The red line splits the active users into two groups as evenly as possible.  

3.4 Visualization of the regular expression used to extract the age and gender of the authors of posts on r/r4r. Image created using regexper.com, license: CC BY 3.0.  

3.5 A histogram showing the age of r/r4r users, zoomed in on the [15 - 60] range, where the vast majority of users are found. Users with more than 250 comments are denoted as active. The red line splits the active users into two groups as evenly as possible.
3.6 A histogram showing how many comments were collected per user. We define users with more than 250 comments as active. For the r/relationship_advice and r/r4r datasets, we discard all users with low activity. ........................................ 39

3.7 A visualization of the r/Europe users represented as a graph. This is a simplified version of the graph only containing the edges with weight higher than 1.5. .... 42

3.8 The users collected from r/relationship_advice, represented as a graph. Above the users are marked by gender, below by age. ................................................ 44

4.1 The ROC curves of two fictional models. The curve on the left is the result of a more capable model, while the curve on the right belongs to a model returning random predictions. The ROC AUC score is the area under this curve. .... 48

4.2 Normalized confusion matrices when applying a Random Forest classifier on the different datasets ............................................................... 56

4.3 Normalized confusion matrices when applying a Multilayer Perceptron classifier on the different datasets ....................................................... 57

4.4 Normalized confusion matrices when applying a K-Nearest Neighbors classifier on the different datasets ....................................................... 58

4.5 Normalized confusion matrices when applying an ensemble of a Random Forest and K-Nearest Neighbors classifier on the different datasets. ........ 59

4.6 Normalized confusion matrices when applying an ensemble of a Multilayer Perceptron and K-Nearest Neighbors classifier on the different datasets. ..... 60

5.1 Normalized confusion matrices when applying the GCN classifier on the different datasets .......................................................... 63
List of Tables

3.1 The flair choices of r/Europe users. The choices in bold are included in our dataset ........................................ 45
3.2 The flair choice of r/AskMen and r/AskWoman users. The choices in bold are included in our dataset ......................... 46
3.3 The reported gender of r/relationship_advice users. “m” and “f” are used for “Male” and “Female”, respectively .................. 46
3.4 The abbreviations related to gender used by r/r4r users in post titles. The choices in bold are included in our dataset ............... 46

4.1 The accuracy and ROC AUC scores when training a Random Forest classifier on the various datasets ................................. 50
4.2 The words that the Random Forest classifier attaches the most value to for classifying users .............................................. 51
4.3 The accuracy and ROC AUC scores when training a Multilayer Perceptron classifier on the various datasets ......................... 52
4.4 The accuracy and ROC AUC scores when training a K-Nearest Neighbors classifier on the various datasets ......................... 54
4.5 The accuracy and ROC AUC scores when training an ensemble of a Random Forrest and K-Nearest Neighbors classifier on the various datasets .............................................................. 55
4.6 The accuracy and ROC AUC scores when training an ensemble of a Multilayer Perceptron and K-Nearest Neighbors classifier on the various datasets ....................................................... 55

5.1 The accuracy and ROC AUC scores when training a GCN classifier on the various datasets .................................................. 62
Introduction

The title of this thesis reads “User Classification Based on Public Reddit Data”. The goal of this introduction is twofold, we first provide a short introduction to what Reddit is and introduce some Reddit-specific terminology. Secondly, we formulate the goal of this thesis and lay out the structure of this document.

1.1 What is Reddit?

Reddit is an American social news site, link aggregator, and discussion platform. It was founded in 2005 by Steve Huffman, Aaron Swartz, and Alexis Ohanian. As of February 2019, it is the 17th most popular website worldwide and the 6th most popular in the United States according to the Alexa web ranking [1]. An annotated screenshot of the site is shown on Figure 1.1.

In the following section, we will briefly explain the mechanics behind the site.

Reddit is organized around communities, called subreddits. Each subreddit has its own topic, internal culture, and community rules. The name of a subreddit is always preceded by “r/” or “/r/”. The rules of a subreddit are created and enforced by its moderators, which are volunteer community members. A moderator has the ability to remove content from their subreddit and to prohibit certain users from submitting new content to that subreddit.
1.1. WHAT IS REDDIT?

Users on the site are also referred to as a Redditors. A username is preceded by “/u/” or “/u/”.

To become a user on the site, one has to create an account. Unlike other social networks such as Facebook, a user does not need to register using their real name. While Reddit does encourage users to provide an email address for account recovery, it is not strictly required. Since users are allowed to register as many accounts as they like using various arbitrary usernames, the website has a strong privacy culture. A registered user can perform the following actions on the site:

- Submit a new post to a subreddit. A post can be a text post, an image, a video, a link to another web page, or just a title without any additional content.

- Leave a comment, either directly on a post, or on another comment. As such, conversations on Reddit have a tree structure, whereby the Original Post (abbreviated as OP, sometimes also refers to the author of the post) is the root, and the comments are the nodes.

- Upvote or downvote a post or a comment. Content should be upvoted when it is relevant to the subreddit it was posted to, and downvoted otherwise. Reddit will use these votes to rank posts and comments. Recently made content that receives many positive votes will be ranked the highest, and thus will be featured more prominently.

As mentioned earlier, Reddit has a strong privacy culture. However, the majority of discussions happen on public subreddits. This means that anyone can not only view, but also download and analyze, all the content made on these subreddits, even without a Reddit account.
CHAPTER 1. INTRODUCTION

1.2 Goal of this master thesis

This thesis researches which personal information can be deduced from the publicly available activity of Reddit users. More specifically, we apply various machine learning methods on comments made by Reddit users. One relatively novel technique we investigate is Graph Convolutional Networks (GCNs). The types of personal information we try to deduce are nationality, gender, and age.

Our contribution includes the creation of six labeled datasets. Labels for these datasets are gathered using two techniques specific to the Reddit website: subreddits with user flairs and subreddits that encourage or enforce standardized post titles.

We hope this work can be helpful to review GCNs on one hand, and on the other hand help raise awareness among privacy-conscious Reddit users about the personal data they make publicly available.

The remainder of this document is split into five sections. First, we provide an introduction to machine learning, review the literature regarding GCNs and review previous attempts at classifying internet users. Secondly, we explore different ways to create labeled datasets from Reddit. Such a dataset consists of Reddit users mapped to a piece of personal information, such as nationality, gender, or age. For each user we also create a set of features based on their public comments. Next, we train and evaluate various baseline machine learning models on our dataset. Thirdly, we apply a GCN model on the datasets and compare the results with those obtained using the baseline methods. Finally, we conclude this thesis and think about future work in this area.
“The little league team you coach just won the big game, and you ask them if they want to go out for pizza or for burgers. Each kid starts screaming their preference, and you go with whatever was the loudest. This is basically how a neural network works but on multiple levels. [...] 

The magic part is no one tells them when to scream, it is based on feedback. Your little league team went for burgers, and some of them got sick. Next week, they might not scream for burgers, or might not scream as loudly. They have collectively learned that burgers might not have been a great choice, and are more likely to lean away from the option. 

--Reddit user u/kouhoutek, trying to explain a Neural Network to a five year old.

2 Related work

This chapter reviews the literature and summarizes works related to this thesis. First, we construct an overview of machine learning models, building up to Graph Convolutional Networks. Next, we review literature related to classifying internet users.

2.1 Machine learning and Artificial Neural Networks

In this section, we provide an introduction to the field of machine learning, with a focus on the concepts that are used throughout this work.

Machine learning covers a wide range of techniques and applications. The goal of this thesis is supervised and semi-supervised learning. We will first focus on supervised machine learning. In this branch of machine learning, we have labeled data samples. From this data, we extract a set of features. These are called \( x \). The output labels, or targets, are called \( r \). We assume there is some true function \( f \) that maps possible input features to the correct output label. The goal is to find an approximation to this function, the hypothesis \( h \), using the labeled training samples. The hope is that this hypothesis will generalize to samples outside the training set. Often, the training data is noisy. A machine learning model that tries to perfectly fit this noisy data will most likely overfit. A model that overfits will be overly complex and generalize badly.
More concretely, a learner needs a loss function which tells the model how good its predictions are (lower is better), and an optimization procedure, which is the strategy the model will take in order to minimize this loss function.

We look at classification, which means the output labels will be part of a discrete finite set.

In the remainder of this section, we discuss various variants of Artificial Neural Networks.

### 2.1.1 Multilayer Perceptron

*Artificial Neural Networks* (ANNs) is a class of machine learning models loosely based on the biological network of neurons in the brain. The “vanilla” ANN is a *Multilayer Perceptron* (MLP).

On Figure 2.1 a schematic representation of an MLP with three layers is shown. On the left-hand side are the input neurons. There are as many input neurons as the number of features. On the right, we see the output neurons. There are as many output neurons as there are labels. In between, there are one or more hidden layers. Each neuron has an incoming connection from each neuron from the previous layer. Each such connection has a weight. A Neural Network where the connections do not form a cycle, as is the case for an MLP, is a feedforward Neural Network. The example network has two input features, three output labels and one hidden layer with three neurons. In practice, there are often more neurons in each layer. The input layer and each hidden layer, also have a bias term with fixed value one.

Classification on a trained MLP happens by setting the values of the input neurons to the values of the input features. A neuron in the second layer now receives a value by multiplying each input value by the weight of the incoming connection and summing the results. A non-linear function $\sigma$, called an activation function, is applied to the result. This process is then repeated for the next layers. In the end, the label is determined by the neuron in the final layer with the highest value. The formula to calculate the value of a neuron in mathematical notation, with $a_j^{(l)}$ the value of neuron $j$ in layer $l$ and $w_{i,j}^{(l)}$ the weight in layer $l$ to go from neuron $j$ to neuron $j$:

$$a_i^{(l+1)} = \sigma(\sum w_{i,j}^{(l)} \times a_j^{(l)}) \quad (2.1)$$

All of these weights need to be tuned during the learning phase. We can think of the MLP as a large function that takes all the weights as input parameters and returns one single value: the average cost value when applying this MLP on all training samples. The cost is a metric for how well the model did the classification. We could initialize all the weights randomly and then take the derivative of this large function. Using this derivative, we could find the directions in
2.1. MACHINE LEARNING AND ARTIFICIAL NEURAL NETWORKS

A Convolutional Neural Network (CNN) is an extension of the MLP from the previous subsection. CNNs are often used for image, video, and audio recognition. Besides fully connected layers, whereby a neuron is directly connected with each neuron of the previous layer, a CNN also features other types of layers. One such layer is the convolutional layer, which we will discuss more closely here.

Let’s consider a convolutional layer in the case of image classification. At the top of Figure 2.2, an image of the number seven is shown. This number seven comes from the MNIST database of handwritten digits. The picture is represented by a $28 \times 28$ grid of numbers. A convolutional layer consists out of one or more filters. On Figure 2.2, one $3 \times 3$ filter is shown. This filter...
CHAPTER 2. RELATED WORK

slides across the image, whereby every $3 \times 3$ chunk of the image is multiplied with the filter and then summed together. The values of the filter are determined during the learning phase of the network [9]. The example filter on the Figure responds the most to top vertical edges. A filter like this makes sense from a biological perspective, since the brain also features neurons that fire when seeing certain edges [11].

One convolutional layer can have multiple filters in parallel. For image classification, filters in one of the first layers in the network will most likely learn to respond to basic shapes, like edges in different directions [12]. Subsequent layers combine the results of previous layers, therefore a filter later on in the network might respond to more complex shapes, like noses or eyes. In the case of the MNIST dataset, we can imagine that a filter that responds heavily to vertical edges, and one that responds to diagonal lines going from the bottom left to top right, would be useful to recognize a seven.

Compared to a fully connected layer from an MLP, this corresponds to a layer where each neuron is only connected to a few neurons from the previous layer, namely those corresponding to neighboring pixels. Instead of every connection having a different weight, the weights of the filter are reused each time. This system of filters allows the model to learn to recognize features, independent of where exactly they present themselves in the image.

CNNs can not only be used for 2D data such as an image [12], but also for 1D data such as audio or text [13], or for 3D data, such as video footage [14].

2.1.3 Graph Convolutional Networks

*Graph Convolutional Networks* (GCNs) try to extend the ideas of CNNs to graph data. In fact, the image from the previous section can already be thought of as a graph: an image is a regular 2D grid, where adjacent pixels are neighbors in the graph. With GCNs however, this idea is expanded to arbitrary graphs.

Until now, we described supervised learning where we feed a model different samples and want it to learn a label for each one of them, e.g.: feeding the model complete images each with one label, with the goal of image recognition. Here, we will do something different. In the next chapter, we will construct a graph where the nodes correspond to Reddit users, with each node having a label. We will then set some of these labels to unknown. The job of the model is filling in these missing labels. Compared to image recognition, this would be like taking an image and erasing some of the pixels, where the job of the model is to fill in these missing pixels. This is called semi-supervised learning [15].

In the previous section, we saw that during 2D convolution, a node is transformed using a
Figure 2.2: An illustration of how a filter can transform an image in a convolutional layer of a CNN. The displayed filter has learned to recognize top vertical edges.
weighted sum of itself and its neighbors. These weights are the elements of the filter. However, in an arbitrary graph, a node has a variable amount of neighbors, and these nodes do not have a particular order \[16\]. We will therefore need to find a more general way to combine a node’s features with those of other nodes in the graph.

The first prominent research in this field is by Bruna et al. from 2013 \[17\]. They define convolution using spectral graph theory. The model proposed by Kipf and Welling, also has its mathematical roots in spectral graph theory, but features some simplifications and can therefore be explained without going into this theoretical background \[15\]. Due to its impressive performance, the model by Kipf and Welling is often used by the research community as the GCN benchmark \[16\]. It is the GCN model we will use in this thesis, and we will now explain it in greater detail.

Let’s first introduce some notation. A graph is defined using an \(N \times N\) adjacency matrix \(A\), with \(N\) the number of nodes. In the case of machine learning, each node can have multiple features. These are stored in a \(N \times D\) matrix \(X\), with \(D\) the number of features. Finally, the labels are stored one hot encoded as a binary \(N \times E\) matrix \(Z\), with \(E\) the number of number of output classes. \(H^{(l)}\) is the output of layer \(l\). \(H^{(0)} = X\) and \(H^{(L)} = Z\).

The most basic graph convolution is replacing each node by the sum of itself and its direct neighbors. Replacing a node by a sum of its neighbors can be written as the following matrix multiplications:

\[
AH^{(l)}
\]  \(\text{(2.2)}\)

To also include the value of the node itself, we add the identity matrix to the adjacency matrix:

\[
(A + I)H^{(l)}
\]  \(\text{(2.3)}\)

Let’s denote \((A + I)\) as \(\tilde{A}\) for convenience. We now replace each node by the sum of itself and its direct neighbors, however it is intuitively clear that it might be better to take an average instead. This can be achieved by normalizing \(\tilde{A}\) as follows: \(\tilde{D}^{-1}\tilde{A}\), where \(\tilde{D}\) the diagonal node degree matrix of \(\tilde{A}\). Kipf and Welling however propose to use symmetric normalization instead. Here we replace \(\tilde{A}\) with \(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}\). Let’s denote this symmetrically normalized matrix as \(\hat{A}\). The result of \(\hat{A}H^{(l)}\) is not a simple average of a node and its neighbors, but a weighted sum, where nodes that have few other neighbors will have a higher contribution. Attaching greater importance to nodes that have few neighbors makes sense from an intuitive point of view. Including weights and an activation function completes the propagation rule proposed by Kipf and Welling:
2.2 ONLINE PRIVACY AND REDDIT

\[ H^{(l+1)} = \sigma(\hat{A}H^{(l)}W^{(l)}) \]  

(2.4)

An illustration of this process is provided on Figure 2.3. In this example, we have three input features for each graph node, depicted by three versions of the graph on the left-hand side. The graphs are convoluted, meaning the feature values of each node of the graph are combined with itself and its neighbors. This corresponds to the \( \hat{A}H^{(l)} \) operation and is graphically illustrated inside the pale blue boxes. Next, the different features are combined, similar to an MLP, using weights and an activation function. This is illustrated by the arrows between the pale blue boxes. In this example, we have one hidden layer with 3 elements. The process is repeated to obtain an output layer. In this example, there are two different output labels, represented by the two graphs on the right-hand side. Nodes that have the highest value in the top graph of the output layer will be assigned the first label. Nodes that have the highest value in the bottom graph the second one.

The weights are trained similar to other Neural Networks, using backpropagation.

2.2 Online privacy and Reddit

Trying to recover personal information from people based on their behavior online is not a new phenomenon. A large amount of academic research has been conducted in this area. This
section provides some examples for deducing characteristics from internet users using machine
learning. First, we look at examples from the mid 2000s where users are classified using blog
posts. Next, we present a couple of examples from the early 2010s that use data from the popular
microblogging platform Twitter. And lastly, we look at an example that, like this thesis, uses
data from Reddit.

2.2.1 Early examples using blogs (mid 2000s)

Weblogs, more commonly referred to as blogs, are personal websites where an author writes
diary-style entries. Finding their origin in the 1990’s, they had a rapid uptake during the 2000s
[18]. In this subsection, two examples of user classification are presented that use blogging
information as an input.

The first example is the work by Schler et al. from 2005 [19]. In this research, the authors try
to deduce the gender and age of users of the blogger.com website. This platform allows users to
report their age and gender in a standardized manner. They collected all blogs accessible on a
day in August 2004, which (1) included an age and gender and (2) used at least 200 instances of
common English words. As a result, 71 000 blogs where collected. They trained a Multi-Class
Real Winnow model on this data. As input of the model a vector is constructed, where each
entry is the frequency of a feature in the text, normalized by document length. These features
include word choices as well as the use of certain stylistic features, such as the use of hyperlinks.
The authors report an accuracy of 80.1% in predicting gender. For age prediction, the following
categories where used: 10s (13-17), 20s (23-27) and 30s (33-42). People not fitting into these
categories where omitted for two reasons: on one hand, to create greater distinction between
the categories, and on the other hand, because many blogs are active over the span of multiple
years, blurring the lines between categories even more. They report an accuracy of 43.8% for
predicting these age groups.

The second example is the research by Yan et al. from 2006 [20]. Here data was taken from
another blogging platform, named Xanga. The goal of this work is to classify users by gender.
Starting with scraping the profile of one arbitrary user for links to other blogs, and repeating
this procedure recursively, 3000 users who report their gender were discovered. These users
account for 75 000 individual blog entries. A Naive Bayes model was trained for two different
cases. In the first case, the input feature vector consists of a simple, bag-of-words wordcount.
In the second case this feature vector is supplemented with some more creative features, such as
the background color of the blog, use of fonts and cases, punctuation marks, and emoticons. In
the first case, a precision of 40% and a recall of 54% is reported. In the second case, a precision
of 65% and a recall 71%.
2.2. ONLINE PRIVACY AND REDDIT

2.2.2 Twitter enters the scene (early 2010s)

Twitter is a microblogging platform founded in 2006 [21]. Witnessing rapid growth during the late 2000s and the first half of the 2010s, it surpassed 300 million active users in 2015. After 2015, while still remaining a widely popular platform, growth has slowed [22]. In this subsection, we look at two papers classifying Twitter users using machine learning.

The first paper is from 2010 by Rao et al. [23]. The authors try to classify users on the following attributes: gender, age, regional origin, and political orientation. For each of these attributes, accounts were crawled by starting from an initial set, or seed, and then recursively looking at the followers of this seed. For classifying gender, they crawled 500 men and 500 women. The initial seed included sororities, fraternities, and male and female hygiene products. For age, they collected 1000 users younger than 30 and 1000 users older. The ground truth for these people was determined by looking for users that link to an external profile which does have a standardized way of reporting age, such as a blog or LinkedIn account. For regional origin, the goal was to differentiate between English written by someone in Southern or Northern India. The seeds are posts from three cities in north India, and three in south India. The ground truth was determined by manually annotating the profiles. In total, 500 users per category were retained. Political orientation was again interpreted by looking for users that link to an external profile which does have a standardized way of reporting age, such as a blog or LinkedIn account. For regional origin, the goal was to differentiate between English written by someone in Southern or Northern India. The seeds are posts from three cities in north India, and three in south India. The ground truth was determined by manually annotating the profiles. In total, 500 users per category were retained. Political orientation was again interpreted in a binary way, Republican/Conservative on one side, Liberal/Left/Democrat on the other. The seed included political hashtags, keywords, and organizations like the NRA. The ground truth was again determined by manual inspection, for a total of 200 users per class. Three Support Vector Machine (SVM) models are trained. The first model was trained using sociolinguistic features. These features include things like the use of emoji, ellipses, repeated characters, etc. The second model uses n-gram features. The last model stacks the previous two. The best models report accuracies between 72% and 83% for the various data sets. For example, the stacked model performs best for classifying gender with 72.33%, and the n-gram one for political orientation with 82.84%. The paper also looked at network features, such as the number of followers and followees, and behavioral features, such as response, retweet and tweet frequency, but did not see a significant correlation between these features and the classes.

The second paper is from 2011 by Pennacchiotti et al. [24]. The authors try to infer the following attributes from Twitter users: political orientation, ethnicity, and whether they like the brand Starbucks. Political orientation was again interpreted as binary: Republican or Democrat. The labels where collected using two Twitter directories, WeFollow and Twellow, where users can classify themselves as Republican or Democrat. Both directories are offline at the time this thesis was written. Ethnicity is interpreted as “African-American” or “not African-American”. The labels are collected by parsing the user’s profile for people who mention their ethnicity. And lastly, the authors look at Starbucks fans. The positive examples are people that follow Starbucks on Twitter, the negative examples are people who do not. The authors used Gradient
Boosted Decision Trees as a model. The F-measure of predicting democratic affiliation is 0.915 and for republicans, it is 0.840. F-measure for predicting ethnicity is 0.655 and for Starbucks affiliation it is 0.759.

2.2.3 And now... Reddit (2010s)

Lastly, we take a look at previous work related to Reddit itself. In particular, we review a paper from 2015 about user classification on Reddit.

Fabian et al. \[25\] started with a collection of 660,464 Reddit comments from users posted during nine days in November 2013. From this, they extracted 76,767 unique Reddit users, and downloaded all of the comments of each of those, using the Reddit API. Next, they used a regular expression on these comments to extract labels. One expression looked for nationality by searching for mentions of country names and the respective demonyms. In total, for 15,012 users a nationality was found. These results are grouped by continent to create 6 different labels, ranked by prevalence: Europe, North America, Asia, Australia and Pacific, South and Middle America, and Africa. Similarly, a regular expression was made looking for combinations of “my” in comments together with gender indicators like “husband”, “girlfriend”, or “hubby”. Fabian et al. seem to assume heterosexual relationships. The result was 25,465 labeled users, of which 78.5% are male. The features used for machine learning are obtained by applying a bag-of-words transformation on the user’s comments. Three models where trained on this data: an SVM, an SVM on Latent Dirichlet Allocation (LDA) topics, and a technique called Supervised Latent Dirichlet Allocation (sLDA). The first model did not perform better than random, with an AUC score of 50%. The second model performed better, with an AUC score of 53.8% for the continent dataset and 87.3% for the gender dataset. The last model performed best, reaching an AUC score of 68.2% for the continent dataset and 87.9% for the gender dataset.
Data collection and analysis

In this chapter, we will discuss the collection of the datasets and analyze their properties.

The first challenge is gathering data from the Reddit website. Reddit has a public API, however rate limitations make it troublesome to use this API to collect large historic datasets. Luckily, read-only mirrors of the API exist without these limitations. In this thesis, the mirror at pushshift.io was used. It is frequently used in academic research. PSAW is a Python wrapper around the Pushshift API and is used in this dissertation. To ensure consistency and reproducibility, all queries have a parameter stating the data must be from before Oct 1st 2018, at 00:00:00 UTC.

As described in Section 2.1, to achieve supervised machine learning we need a large number of labeled samples, and for each sample we need features. First, we will discuss how we can collect Reddit users for which we can find a label. Next, we will gather more information of these users and extract features from this data.

https://pushshift.io/api-parameters/
https://github.com/dmarx/psaw
CHAPTER 3. DATA COLLECTION AND ANALYSIS

3.1 Collection and labeling

The first step of creating a dataset is finding a large number of users for which we can deduce a label. The type of information we are interested in are nationality, age, and gender. Reddit does not allow users to add this information in a standardized way to their profile. In this thesis, we explore two methods specific to the Reddit website to source these labels.

The first method uses the user flairs of a subreddit. In some communities, users can add a piece of extra information next to their username, this is called a user flair. For an example of this, see Figure 3.1. Three such subreddits that support user flairs are r/Europe, r/AskMen, and r/AskWomen. On r/Europe, users can optionally select a flag to show up next to their username, and on r/AskMen and r/AskWomen, people can report their gender. At the time of writing, Reddit is undergoing a redesign impacting some of the technical details about how user flairs are handled internally, but at the time the data was collected none of the listed subreddits were significantly affected by this change.

The second method to source labels exploits subreddits that enforce or encourage standardized post titles. For example, the subreddits r/relationship_advice and r/r4r encourage users to include their age and gender in their posts.

In total, we have created six datasets. One on nationality, where labels are sourced from r/Europe. Three on gender: one with labels from r/AskMen and r/AskWomen, one from r/relationship_advice, and one from r/r4r. And two on age: one with labels from r/relationship_advice, and one from r/r4r.

3.1.1 Finding your place on r/Europe

r/Europe is a subreddit intended for content related to the European continent, it’s tagline reads: “Europe: 50 (+6) countries, 230 languages, 743M people... 1 subreddit.”. The subreddit
mainly features news articles, pictures, maps and infographics. As of writing, the subreddit has 2 million subscribers.

The first step in order to gather a set of labeled users is downloading the 500,000 most recent comments on the subreddit. From these comments, we found 56,856 unique authors. Of these users, 16,499 have a user flair. The user flairs on r/Europe both have a CSS class that is used to show the right flag and a human-readable label stating the country. Users can edit this human-readable label as wished, therefore using the CSS classes as labels instead is less noisy. The breakdown of the different labels is shown on Table 3.1. As you can see, some of the popular options do not correspond to one specific country, i.e. EARTH for earth and EURO for Europe. We discard these users. The overall number of different flairs is large and has a long tail. Therefore we chose to only look at users from the 20 most prevalent locations. This leaves us with 9,597 labeled users. These countries are indicated in bold on the table. The table also includes some additional information that might interest the reader, such as the number of users we collected expressed in relation to the total population of their respective country.

### 3.1.2 Asking the hard questions on r/AskMen and r/AskWomen

r/AskMen and r/AskWomen are two communities with the goal of asking questions to the general male, resp. female, population of Reddit. Common themes of questions are relationships, gender, and sex. As of writing, r/AskMen has 1 million subscribers, and r/AskWomen almost 900 thousand. As with the previous dataset, first 500,000 comments were downloaded. Using these comments, we found 9,489 unique users with a flair. The flair options on r/AskWomen are “male”, “female”, “trans”, and “neutral”. On r/AskMen, they are “male”, “female”, “bane”, “trans”, and “agender”. Bane does not refer to a gender, but rather to the comic book character of the same name. It is part of an inside joke on the subreddit. Both subreddits include options for transgender people, but do not refine these options further into trans men or trans woman.

The number of users we found for each of these options are shown on Table 3.2.

As you can see, the vast majority of users of these subreddits have picked either “male” or “female”. For the purpose of this thesis, we will discard the other options, as including these would either make no sense, in the case of “bane”, or would result in a very unbalanced dataset, in the case of “trans”, “neutral”, and “agender”. This leaves us with 9,089 users, which are displayed in bold in the table.

### 3.1.3 Getting help on r/relationship_advice

As the name implies, r/relationship_advice is a community where users can ask advice regarding their personal relations. These relationships are often of a romantic nature, but this is not
required. Someone who seeks advice will create a post explaining their situation. As customary on Reddit, this explanation most often will not contain any real names. However, users will often mention the age and gender of the different actors in the title of their post. Since r/relationship_advice does not provide user flairs like the previously mentioned subreddits, we will try to deduce the labels from these titles instead.

First, 50,000 posts were downloaded. Next, a regular expression was created that attempts to catch the mentions of the author’s gender and age, while having a minimal amount of false positives. These are a couple examples of post titles:

- “My roommate [21F] keeps cooking food I [19F] cannot stand the smell of...how do I address it?”
- “Me (18M) Not sure what to do about relationship with (18F) girlfriend”
- “I am (22/M) still in love with a girl (22) who is my best friend now for almost 5 years”
- “I (23,F) try to love my BF (26,M)”
- “I [M24] just moved for school and my girlfriend [F23] and I don’t know if we can do long distance.”
- “Me M(19) and my colleague F(20) women are confusing”
- “I’m a 42F and very happy and people like to criticize that.”

Using trial-and-error, this is the resulting regular expression, which is applied in a case-insensitive manner:

```
(i[^\s'\"]*(a?m( a)?|me|my)\s*(([0-9]+)\s/,,*([mf])|([mf])(\s/,,*([0-9]+))\s*([mf])(\s/,,*([0-9]+))\s*))
```

This expression might look daunting, but it basically looks for phrases like “I”, “I’m”, “I am”, “I’m a”, “Me”, and “My”, followed by age and gender expressed in the ways shown in the examples above. A visualization of the regular expression is provided on Figure 3.2.

The regular expression has false positives, such as the following title, where the author placed the age and gender before the actors instead of after: “(M19)My (F19)Girlfriend chooses event over my friends”.

Of the 50,000 posts, the expression managed to find the age and gender for 11,989 of them. The results are shown on Table 3.3.
3.1. COLLECTION AND LABELING

Manual inspection learns us that the posts where the expression failed are most often posts where the author does not mention their age and gender at all. There are also examples where the expression failed because it was not intelligent enough, such as in the following cases:

- “[25M] No Friends, and not sure where to begin”
- “My wife and I (31F, 32M) want different things, not sure if to split up because we do love each other. I want to move back home, she wants to stay in the big city.”

The expression was not adapted for these cases because they are either too complex, or would cause too many false positives.

Note, the expression only looks for people who denote their gender as either “f” or “m”, for female or male, respectively. This choice was made as they are the two most common gender-related abbreviations used on the subreddit.

Besides gender, age was also collected. The resulting age distribution of the users is visualized on Figure 3.3. Note that people under 13 are not allowed to use Reddit. Later, we will split these users into two groups, as equally as possible, whereby the resulting label is whether a user is younger or older than a certain value.

We now have two datasets with users from r/relationship_advice: one mapping users to gender, the other to age. Each with 11 989 users. In the following section, the datasets will be refined, to exclude users with low activity.

3.1.4 A/S/L on r/r4r

r/r4r, short for “Redditor for Redditor”, is a subreddit where users can place personal ads, or to use the subreddit’s own words:

Whether you’re looking for platonic or non-platonic friends, gaming buddies, online friends, soulmates, travelmates, smoking buddies, activity partners, friends with benefits, or casual encounters, this is the place to find and seek.

Similar to the previous subsection, we collect 50 000 posts and apply a regular expression to the title to extract age and gender. The subreddit enforces the following title format: <age><r4r>\location<title>, where the user has to fill in the parts between angled brackets. <r4r> is replaced to indicate the user’s own gender and the gender of the person they are looking for, e.g.
CHAPTER 3. DATA COLLECTION AND ANALYSIS

Figure 3.2: Visualization of the regular expression used to extract the age and gender of the authors of posts on r/relationship-advice. Image created using regexper.com. License: CC BY 3.0
3.1. **COLLECTION AND LABELING**

Figure 3.3: A histogram showing the age of r/relationship_advice users, zoomed in on the [13 - 50] range, where the vast majority of users are found. Users with more than 250 comments are denoted as active. The red line splits the active users into two groups as evenly as possible.

A woman looking for a man would write f4m. Sadly, location is not expressed in a systematic way, making it non-trivial to use this to create location labels.

Since this subreddit enforces stricter post formatting than r/relationship_advice, it is possible to extract age and gender using a simpler regular expression:

```
([0-9]+)\s*\[(\[^4\]\]+)4(\[^4\]\]+)\]
```

A visual representation of this expression is shown on Figure 3.4. The regular expression is able to catch other characters than “f” and “m”, and also multiple characters at a time. The result of this is shown on Table 3.4. The character “r” is used to describe a Reddit user in general, regardless of gender. Multiple characters refer to couples of the respective genders. We again opted to only consider the most popular choices, “m” and “f”, indicated in bold on the table. The resulting size of this dataset is 16,366.

The age distribution is shown on Figure 3.5. There is a sharp drop-off before age 18, since the subreddit rules forbid younger users from participating. There are 18,997 unique users in this dataset. We will later split these users into two groups, to reduce the different values of this label to two.

The two datasets created using r/r4r will be refined in the following section to exclude users with low activity.
Figure 3.4: Visualization of the regular expression used to extract the age and gender of the authors of posts on r/r4r. Image created usingregexper.com, license: CC BY 3.0

Figure 3.5: A histogram showing the age of r/r4r users, zoomed in on the [15 - 60] range, where the vast majority of users are found. Users with more than 250 comments are denoted as active. The red line splits the active users into two groups as evenly as possible.

3.2 Feature extraction

In the previous section, we collected six datasets, each containing a collection of Reddit users mapped to a label. Now, we need to create for each user a set of features. These features will be fed into various machine learning models in the next two chapters, with the goal of getting the correct label back.

The first step of creating these features is collecting for each user a large number of comments. With the Pushshift API it is possible to get up to 500 comments per request. With PSAW, the Python wrapper around the Pushshift API we used, it is possible to request more comments, but this is achieved internally by issuing multiple successive requests. We decided to only issue
3.2. FEATURE EXTRACTION

Figure 3.6: A histogram showing how many comments were collected per user. We define users with more than 250 comments as active. For the r/relationship_advice and r/r4r datasets, we discard all users with low activity.

one request per user, and thus only collect their 500 most recent comments. This is a trade-off between the amount of collected data and the speed of collection. However, for labels with a temporal character, such as age and to a lesser extent also gender and nationality, only looking at more recent comments makes sense.

These comments can be made in any subreddit, i.e. for a user in the r/Europe dataset, we used information from r/Europe to find the label, but the comments we use to create the features can come from other subreddits too. When a user hasn’t made a total of 500 comments yet, fewer are collected, 500 per user is thus the upper limit. Figure 3.6 shows how many comments are effectively collected per user. As you can see, r/relationship_advice and r/r4r have many users with a low number of comments. Having a large percentage of users with low activity negatively affects the quality of the dataset, therefore we define users with more than 250 comments as “active users”. For r/relationship_advice and r/r4r we discard all users not marked as active. Making this change, the two r/relationship_advice datasets now have 1905 users each, the r/r4r age dataset 3561 users, and the r/r4r gender dataset 3126. The active users are shown on Tables 3.4 and 3.3, and Figures 3.5 and 3.3.

Now that our datasets have reached their final form, we can split the age related datasets in two, aggregating all ages below a threshold into one label, and all the remaining ones into another. We choose to split the users in only two bins, and pick the threshold in such a way that the bins are as equal in size as possible. For r/relationship_advice, the most equal split is at 24, with 900 users being 24 or older, and 1005 being younger than 24. For r/r4r, it is two years older,
with 1690 users 26 years old or older, and 1598 being younger.

Next, we process these comments to create two types of features: textual features on one hand, and graph-based features on the other.

3.2.1 Textual features

A first way of processing the comments made by a user is by looking at them as pure text. All comments of a user are concatenated and then vectorized using term frequency–inverse document frequency (tf-idf)\(^27\). The goal of vectorizing is to turn the free text into an array of numbers. In tf-idf, each element in the array corresponds to a word written by any of the users. tf-idf is the product of the frequency of a term inside a document and the inverse document frequency of that term. In this context, one document corresponds to all the comments made by a user.

\[
\text{tf} - \text{idf}(t, d) = \text{tf}(t, d) \times \text{idf}(t)
\]  

Let’s illustrate this idea with a small example. Imagine there a 5 users in total. A user \(u\) mentions the word “Belgium” 3 times. One other user mentions Belgium as well. In this case, the term frequency of the word “Belgium” for that user is 3, and the inverse document frequency of the term “Belgium” is \(5/2\), thus \(\text{tf} - \text{idf} (“Belgium”, u) = 15/2\). In this way, a term that is used often by a user will result in a high value, a term that is also used by many other users will result in a lower value. Concretely, we used the tf-idf implementation from the scikit-learn machine learning library\(^28\). This implementation is slightly more complex than the one explained in the simplistic example above. Assume term \(t\), document \(d\), total number of documents \(n\), and \(\text{df}(t)\) the number of documents containing \(t\). The idf used in the example is \(\text{idf}(t) = n/\text{df}(t)\), however it is more common to use a logarithmic scale \(\text{idf}(t) = \log(n/\text{df}(t))\). This would however mean that terms that occur in all documents are completely ignored, this can be solved by adding one to the idf as follows \(\text{idf}(t) = \log(n/\text{df}(t)) + 1\). To prevent division by zero when we look at a term that does not occur in any document, we can pretend there is an extra document that contains all terms, resulting in the final formula\(^29\):

\[
\text{idf}(t) = \log((1 + n)/(1 + \text{df}(t))) + 1
\] 

Additionally, we also remove English stop words and only consider words that are used by at least 5 different users.
3.2. FEATURE EXTRACTION

3.2.2 Graph-based features

Besides interpreting comments purely textual, we can also take their context into account and use them to create a graph structure between the different users.

The idea behind the way we create this user graph is loosely inspired on tf-idf, explained in the previous subsection. Each user is represented by a node. For each comment of a user, we know the post the discussion belonged to. If two users take part in the discussion belonging to the same post, we draw an edge between their respective nodes. The weight of the edge is calculated as follows:

\[
\text{weight}_{\text{edge}}(i, j) = \sum_{c \in S} \frac{1}{\text{No. of comments } c}, \text{for } S \text{ each conversation shared by } i, j
\]  

(3.3)

As a first example, let’s look at the resulting graph obtained from the r/Europe dataset. The resulting graph has 9597 nodes and on average a node is connected to 500 others, with only 2 nodes having no neighbors. The total number of edges is 4 799 396.

The assumption is that there is a relationship between a user and it’s neighbors in this graph. The intuition underpinning this hypothesis is that people who have the same gender, age or nationality have a higher chance of having similar interests and thus a higher chance taking part in the same conversations. To test this intuition, we make a visualization of the r/Europe graph using NetworkX \cite{30}, a Python library for studying graphs. We color the nodes based on the label, and position them using the "Spring layout“ implemented in NetworkX. This layout uses the force-directed layout algorithm by Fruchterman and Reingold \cite{31}. In a nutshell, this algorithm models all nodes as electrically charged steel rings, and the edges as springs connecting these rings. Each iteration, the electrical force will push nodes apart, while the springs will pull them together. As a result, nodes with many connections will be displayed close to each other. Since displaying 9597 nodes and 4 799 396 edges results in a very crowded image, and it is computationally intensive to apply the Fruchterman-Reingold with this many variables, we decided to visualize a simplified version of the graph. First, we removed all edges with a weight value below 1.5. Next, we only look at the largest connected subgraph. This results in a graph with 3889 nodes and 10 350 edges, and is shown on Figure 3.7. We can immediately spot multiple clusters of nodes of the same color drawn close to each other. This supports the idea that users of who report the same nationality on r/Europe, often take part in the same discussions on Reddit, i.e. there is some correlation between nationality and the graph we have created.

As a second example, we take a look at the graph created using the r/relationship_advice data. This dataset contains 1905 nodes and 90 259 edges, good for and average degree of 47.4.
Figure 3.7: A visualization of the r/Europe users represented as a graph. This is a simplified version of the graph only containing the edges with weight higher than 1.5
3.2. **FEATURE EXTRACTION**

Fifteen nodes have no neighbors, all other nodes are connected. The r/relationship_advice datasets are the smallest of all the datasets we collected, so we will plot it in full, with only the 15 unconnected nodes removed. The graph is shown twice on Figure 3.8: once with the nodes colored by gender, once by age category. Due to the high connectivity of the graph, the visualization can best be described as an entangled mess. However, on the graph indicating gender, it can be seen that there is a higher concentration of women on the right hand side. We did not manage to recognize a pattern in the graph with the age label marked. Based on this, we predict that there will be some relationship between gender and the discussions a user joins, and if there also exists such a relationship regarding to age, it will be a weaker one.
Figure 3.8: The users collected from r/relationship_advice, represented as a graph. Above the users are marked by gender, below by age.
Table 3.1: The flair choices of r/Europe users. The choices in bold are included in our dataset.

<table>
<thead>
<tr>
<th>CSS Class</th>
<th>Name</th>
<th>Type</th>
<th>No. of users</th>
<th>Users per million inhabitants</th>
</tr>
</thead>
<tbody>
<tr>
<td>EURO</td>
<td>Europe</td>
<td>Continent</td>
<td>1387</td>
<td>1.87</td>
</tr>
<tr>
<td>UNSA</td>
<td>United States</td>
<td>Sovereign state</td>
<td>890</td>
<td>2.74</td>
</tr>
<tr>
<td>UKGB</td>
<td>United Kingdom</td>
<td>Sovereign state</td>
<td>870</td>
<td>13.15</td>
</tr>
<tr>
<td>GERM</td>
<td>Germany</td>
<td>Sovereign state</td>
<td>805</td>
<td>9.80</td>
</tr>
<tr>
<td>SWED</td>
<td>Sweden</td>
<td>Sovereign state</td>
<td>796</td>
<td>80.32</td>
</tr>
<tr>
<td>NETH</td>
<td>The Netherlands</td>
<td>Sovereign state</td>
<td>778</td>
<td>45.67</td>
</tr>
<tr>
<td>FINL</td>
<td>Finland</td>
<td>Sovereign state</td>
<td>735</td>
<td>133.07</td>
</tr>
<tr>
<td>FRAN</td>
<td>France</td>
<td>Sovereign state</td>
<td>474</td>
<td>7.29</td>
</tr>
<tr>
<td>IREL</td>
<td>Ireland</td>
<td>Sovereign state</td>
<td>466</td>
<td>97.87</td>
</tr>
<tr>
<td>DENK</td>
<td>Denmark</td>
<td>Sovereign state</td>
<td>459</td>
<td>80.06</td>
</tr>
<tr>
<td>POLA</td>
<td>Poland</td>
<td>Sovereign state</td>
<td>444</td>
<td>11.63</td>
</tr>
<tr>
<td>PORT</td>
<td>Portugal</td>
<td>Sovereign state</td>
<td>421</td>
<td>40.76</td>
</tr>
<tr>
<td>ROMA</td>
<td>Romania</td>
<td>Sovereign state</td>
<td>352</td>
<td>17.89</td>
</tr>
<tr>
<td>NORW</td>
<td>Norway</td>
<td>Sovereign state</td>
<td>340</td>
<td>64.09</td>
</tr>
<tr>
<td>ITAL</td>
<td>Italy</td>
<td>Sovereign state</td>
<td>285</td>
<td>4.80</td>
</tr>
<tr>
<td>BELG</td>
<td>Belgium</td>
<td>Sovereign state</td>
<td>283</td>
<td>24.76</td>
</tr>
<tr>
<td>ASTR</td>
<td>Austria</td>
<td>Sovereign state</td>
<td>276</td>
<td>31.60</td>
</tr>
<tr>
<td>GREE</td>
<td>Greece</td>
<td>Sovereign state</td>
<td>247</td>
<td>22.13</td>
</tr>
<tr>
<td>TURK</td>
<td>Turkey</td>
<td>Sovereign state</td>
<td>245</td>
<td>3.03</td>
</tr>
<tr>
<td>CZEC</td>
<td>Czechia</td>
<td>Sovereign state</td>
<td>220</td>
<td>20.72</td>
</tr>
<tr>
<td>CANA</td>
<td>Canada</td>
<td>Sovereign state</td>
<td>211</td>
<td>5.76</td>
</tr>
<tr>
<td>SCOT</td>
<td>Scotland</td>
<td>UK Country</td>
<td>202</td>
<td>37.24</td>
</tr>
<tr>
<td>ENGL</td>
<td>England</td>
<td>UK Country</td>
<td>199</td>
<td>3.58</td>
</tr>
<tr>
<td>EART</td>
<td>Earth</td>
<td>Planet</td>
<td>185</td>
<td>0.02</td>
</tr>
<tr>
<td>CROA</td>
<td>Croatia</td>
<td>Sovereign state</td>
<td>184</td>
<td>43.92</td>
</tr>
<tr>
<td>ESPA</td>
<td>Spain</td>
<td>Sovereign state</td>
<td>163</td>
<td>3.52</td>
</tr>
<tr>
<td>LITH</td>
<td>Lithuania</td>
<td>Sovereign state</td>
<td>157</td>
<td>54.32</td>
</tr>
<tr>
<td>HUNG</td>
<td>Hungary</td>
<td>Sovereign state</td>
<td>154</td>
<td>15.84</td>
</tr>
<tr>
<td>ESTO</td>
<td>Estonia</td>
<td>Sovereign state</td>
<td>148</td>
<td>113.01</td>
</tr>
<tr>
<td>BULG</td>
<td>Bulgaria</td>
<td>Sovereign state</td>
<td>140</td>
<td>19.76</td>
</tr>
<tr>
<td>RUSS</td>
<td>Russia</td>
<td>Sovereign state</td>
<td>136</td>
<td>0.94</td>
</tr>
<tr>
<td>AUST</td>
<td>Australia</td>
<td>Sovereign state</td>
<td>123</td>
<td>5.03</td>
</tr>
<tr>
<td>SLOV</td>
<td>Slovenia</td>
<td>Sovereign state</td>
<td>123</td>
<td>59.14</td>
</tr>
<tr>
<td>SWIT</td>
<td>Switzerland</td>
<td>Sovereign state</td>
<td>120</td>
<td>14.16</td>
</tr>
<tr>
<td>Other</td>
<td>276 flairs with less then 100 users each</td>
<td>3266</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 3.2: The flair choice of r/AskMen and r/AskWoman users. The choices in bold are included in our dataset.

<table>
<thead>
<tr>
<th>Flair</th>
<th>No. of users</th>
</tr>
</thead>
<tbody>
<tr>
<td>male</td>
<td>5419</td>
</tr>
<tr>
<td>female</td>
<td>3670</td>
</tr>
<tr>
<td>bane</td>
<td>133</td>
</tr>
<tr>
<td>trans</td>
<td>84</td>
</tr>
<tr>
<td>neutral</td>
<td>47</td>
</tr>
<tr>
<td>agender</td>
<td>29</td>
</tr>
</tbody>
</table>

Table 3.3: The reported gender of r/relationship_advice users. “m” and “f” are used for “Male” and “Female”, respectively.

<table>
<thead>
<tr>
<th>Gender</th>
<th>No. of users</th>
<th>No. of active users</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>6484</td>
<td>1251</td>
</tr>
<tr>
<td>f</td>
<td>5505</td>
<td>654</td>
</tr>
</tbody>
</table>

Table 3.4: The abbreviations related to gender used by r/r4r users in post titles. The choices in bold are included in our dataset.

<table>
<thead>
<tr>
<th>Gender</th>
<th>No. of users</th>
<th>No. of active users</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>11928</td>
<td>2555</td>
</tr>
<tr>
<td>f</td>
<td>4438</td>
<td>571</td>
</tr>
<tr>
<td>r</td>
<td>299</td>
<td>77</td>
</tr>
<tr>
<td>mf</td>
<td>270</td>
<td>23</td>
</tr>
<tr>
<td>t</td>
<td>177</td>
<td>32</td>
</tr>
<tr>
<td>fm</td>
<td>133</td>
<td>13</td>
</tr>
<tr>
<td>mm</td>
<td>18</td>
<td>4</td>
</tr>
<tr>
<td>w</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>Other</td>
<td>69</td>
<td>12</td>
</tr>
</tbody>
</table>
We have constructed five baseline methods. Two methods that only look at the textual properties of a user’s comments, one method that uses the graph representation of the users, and two methods that combine textual properties with graph information. We discuss these methods and apply them on the datasets constructed in Section 3.

For the r/AskMen and r/AskWomen dataset, we only consider features sourced outside those subreddits. This is done because users with the same gender would be close together in the graph just because of the way the data is collected and the nature of these two subreddits.

For each of the datasets, we randomly select 25% of the samples as testing data and the remaining 75% as training data. We first train the models using the training data, and then use this trained model to classify the testing data. The performance of the models on this testing data is used to evaluate the usefulness of the models. Before we go over to training the models themselves, we will go deeper into the evaluation metrics we used.

### 4.1 Evaluation metrics

In this and the following chapter, we evaluate the various models in three ways: an accuracy score, an Area Under the Receiver Operating Characteristic Curve (ROC AUC) score and by
CHAPTER 4. BASELINE METHODS

Figure 4.1: The ROC curves of two fictional models. The curve on the left is the result of a more capable model, while the curve on the right belongs to a model returning random predictions. The ROC AUC score is the area under this curve.

looking at the confusion matrices. We will define each of them in this section.

Accuracy is the ratio of all test samples that were correctly predicted:

\[
\text{Accuracy} = \frac{\text{No. of correctly predicted samples}}{\text{No. of samples}}
\]

This metric will be less informative for multi-label datasets or unbalanced datasets.

ROC AUC stands for the area under the Receiver Operating Characteristic (ROC) curve. We first define the ROC curve in the case of binary classification. Instead of requesting the model whether a sample is predicted as true or as false, we request the model to return the probability they associate with the label being classified as true. We now vary the threshold for classifying a label as true. The ROC curve is the plot of the true positive rate (TPR) against the false positive rate (FPR) for these various thresholds. For a high threshold such as 1, no samples will be classified as true, regardless of how high their probability is. Both the TPR and the FPR will be zero. For a low threshold such a 0, all samples will be classified as true. Both the TPR and the FPR will be one. Figure 4.1 shows two fictional ROC curves. The first curve belongs to the more capable model, because it has a spot where a high TPR can be achieved with a low FPR. The second curve belongs to a model that just returns random predictions. The ROC AUC score is the area under this curve. A higher score corresponds to a better model, the random model has a score around 0.5. The ROC AUC score is extended to multiple labels by calculating the ROC AUC for each label separately and then taking the average.

A confusion matrix breaks down for each label what the model predicted. On the Y-axis,
the true labels are laid out, and on the X-axis the predicted labels. We decided to normalize the confusion matrix, meaning that each element of the matrix is the fraction of samples with a given true label that is designated a certain predicted label. In other words, each row adds up to 1 and having high values on the diagonal means the model predicted many samples correctly. A confusion matrix is particularly helpful to analyze multi-label datasets. Many examples of confusion matrices are provided in the following sections.

4.2 Using the textual content of comments

The first two baselines only use textual features extracted from the comments, as described in Subsection 3.2.1. More specifically, the two first methods are a Random Forest and a Multilayer Perceptron classifier. For both, we use implementations from the scikit-learn machine learning library.

4.2.1 Random Forest

In this subsection, we will first briefly describe how a Random Forest classifier works, and then review the results obtained after training the Random Forest classifier on the various datasets.

A Random Forest works by combining multiple decision trees, therefore we will first explain how a decision tree works. Take all labeled training examples. Check for which feature variable, and with which value, the samples are split in two the best, where the two splits have the lowest entropy, i.e. the two splits are as homogeneous as possible. Repeat this step for each of the two resulting groups. Keep repeating until each group only has samples of the same label. The result is a flowchart that could be used to perfectly classify the elements of the training set. However, the algorithm as we just presented it, will most likely overfit the training data.

A Random Forest classifier will train multiple “random” decision trees. Instead of training a tree using all the data samples, random samples are picked with replacement. The process is repeated multiple times, in our case 10 times. Since each time other random samples are chosen, the results are 10 different decision trees. The resulting prediction is the plurality vote between the results of the various trees. The scikit-learn implementation builds further on this idea: instead of returning one prediction, each tree returns the probability of each label. The resulting prediction is then the label with the largest combined probability.

The resulting accuracy and ROC AUC scores are shown on Table 4.1, and the confusion matrices
**Table 4.1:** The accuracy and ROC AUC scores when training a Random Forest classifier on the various datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>r/Europe</th>
<th>r/AskMen</th>
<th>r/AskWomen</th>
<th>r/relationship_advice</th>
<th>r/r4r</th>
<th>r/r4r</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>0.73</td>
<td>0.84</td>
<td>0.78</td>
<td>0.85</td>
<td>0.62</td>
<td>0.60</td>
</tr>
<tr>
<td>ROC AUC</td>
<td>0.93</td>
<td>0.90</td>
<td>0.82</td>
<td>0.85</td>
<td>0.65</td>
<td>0.63</td>
</tr>
</tbody>
</table>

It is also possible to request the scikit-learn Random Forest implementation which features are the most important. These are shown on Table 4.2. For the r/Europe dataset, we mostly see words in non-English languages. Reddit has various non-English subreddits, often tied to a specific country. From this, we notice that our model has picked up on this usage of foreign languages as a handy way of classifying users by nationality. For the gender datasets, we recognize words that could be stereotypically associated with men, such as words related to gaming, and words related to women, such as “husband”, “boyfriend”, and beauty-related terms. For the age datasets, we recognize words that are associated with certain stages of life, such as “school”, “married”, “work”, “apartment”, and “wife”.

Looking more closely at the confusion matrix of the r/Europe dataset, we see the most errors occur for nationalities that share a native language. For example, many Austrians are misclassified as Germans. Similarly, Belgians get confused with Dutch people and to a lesser extend with French people. Canadians, get mistaken for Americans and also sometimes for French people. Written Norwegian is similar to Danish, another prevalent confusion. We also see troubles for the very unbalanced r/r4r gender database, where most women are classified as male. For the two age datasets, we see that the model does an okay job at classifying younger people as such, but it predicts nearly at random for older people.

### 4.2.2 Multilayer perceptron

The second baseline is a Neural Network model, more specifically a Multilayer Perceptron (MLP). The model works analogously to the model described in Subsection 2.1.1. The network has one
4.3 Using graph data

A third baseline takes a graph based approach, where the different users are represented by nodes. An edge is drawn between users when they take part in the same discussions. The graph is constructed as described in Subsection 3.2.2. The third baseline consists of a *K-Nearest Neighbors* (KNN) algorithm over this graph. Similar to the previous baselines, we again take a test size of 25% of each data set. The labels of these nodes are set to unknown in the graph, but the nodes itself remain part of the graph structure.

This KNN model is implemented by the author of this thesis. Compared to the previous models, it is a rather simplistic model. We will now briefly describe how our KNN algorithm works. First, the weights of the edges are converted to costs. This is achieved by inverting them. A higher weight indicates a stronger connection, it should therefore be cheaper to follow that path. Next, for each node in our test set, we find the K nearest nodes with a label. These are the K nodes...
Table 4.3: The accuracy and ROC AUC scores when training a Multilayer Perceptron classifier on the various datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>r/Europe</th>
<th>r/AskMen</th>
<th>r/AskWomen</th>
<th>r/relationship_advice</th>
<th>Gender</th>
<th>r/r4r</th>
<th>Gender</th>
<th>Age</th>
<th>r/r4r</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>0.84</td>
<td>0.94</td>
<td>0.92</td>
<td>0.92</td>
<td>0.77</td>
<td>0.72</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROC AUC</td>
<td>0.99</td>
<td>0.98</td>
<td>0.96</td>
<td>0.95</td>
<td>0.86</td>
<td>0.79</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

with a label for which the cost of getting to them is the smallest. Note that the path of getting to these nodes themselves can contain unlabeled nodes. Since all costs are positive this can be implemented efficiently, drawing inspiration from Dijkstra’s algorithm \[37\]. Lastly, if such “neighbors” are found, we choose their most prevalent label as prediction. If the node is not connected to any labeled nodes, either directly or indirectly, no neighbors are found and we simply return the most prevalent label in the whole graph. Pseudocode for this algorithm is shown on Listing \[1\].

We apply this algorithm on the various datasets. We found the optimal number for $K$ to be 27 for r/Europe dataset and choose to reuse this value for the other datasets as well. The resulting accuracy and ROC AUC scores are displayed on Table 4.3 and the confusion matrices on Figure 4.4.

We see that this model performs significantly worse than the previous two. In the case of the age datasets, the predictions are mostly random. And in the case of the heavily unbalanced r/r4r gender dataset, almost all users are predicted to be male. It also does a poor job at labeling women in the r/relationship_advice dataset. However, if we look at the confusion matrix for the r/Europe dataset, we see it is better at distinguishing between users from countries with similar languages: the confusion of Austrians with Germans and Canadians with Americans is now significantly less pronounced.

The KNN model is not a very intelligent algorithm, and for most of the datasets it does not return a very intelligent answer. However, this experiment shows that in the case of the r/Europe dataset, using even a very simple model that takes the graph structure into account has the potential to improve results for samples where the previous models failed.
for each edge: convert weights to costs
for each node in testSet:
    priorityQueue = PriorityQueue()
    nearestNeighbors = []
    visited = []
    priorityQueue.push(priority=0, node=node)
while length(nearestNeighbors) <= k and length(priorityQueue) > 0:
    cost, currentNode = priorityQueue.pop()
    visited.append(currentNode)
    if the label of currentNode is known:
        nearestNeighbors.append(currentNode)
    for neighbor in currentNode.neighbors:
        if neighbor not in visited:
            priorityQueue.push(priority=cost(currentNode, neighbor) + cost, node=neighbor)
    if nearestNeighbors:
        Predicted label of node = most common label of nodes in nearestNeighbors
    else:
        Predicted label of node = most common label in the whole graph

Listing 1: Pseudocode for the K-Nearest Neighbors algorithm
Table 4.4: The accuracy and ROC AUC scores when training a K-Nearest Neighbors classifier on the various datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Accuracy</th>
<th>ROC AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>r/Europe</td>
<td>0.58</td>
<td>0.88</td>
</tr>
<tr>
<td>r/AskMen</td>
<td>0.80</td>
<td>0.88</td>
</tr>
<tr>
<td>r/AskWomen</td>
<td>0.75</td>
<td>0.80</td>
</tr>
<tr>
<td>r/relationship_advice</td>
<td>0.80</td>
<td>0.72</td>
</tr>
<tr>
<td>r/r4r</td>
<td>0.55</td>
<td>0.58</td>
</tr>
<tr>
<td>Gender</td>
<td>0.54</td>
<td>0.55</td>
</tr>
<tr>
<td>Age</td>
<td>0.54</td>
<td>0.55</td>
</tr>
</tbody>
</table>

4.4 Combining textual content and graph data

The fourth and fifth baseline methods consist of two ensembles: one combining the Random Forrest and the KNN algorithm, and one the MLP with the KNN algorithm.

We will now explain in more detail how these ensembles are constructed. First, the training data is shuffled and split into two equal parts. Each of the two input models are trained on the first half of the training data. Next, each of the models are used to classify the remainder of the training data. Instead of just returning the predicted label, we ask each of the models to return the probability they attach to each label. The probabilities of these models are then fed into a Logistic Regression classifier. After this Logistic Regression classifier is trained on these probabilities, the two input models are trained again using all training samples.

The results of the two ensembles are shown on Tables 4.5 and 4.6, and Figures 4.5 and 4.6.

Overall, the ensembles perform similar to the models without the KNN model added. It makes sense that the Logistic Regression attaches the most value to the Random Forest or MLP, since these are more potent models. However the ensemble with the MLP performs exceptionally well for the r/Europe dataset: the common misconceptions are now even less pronounced in the confusion matrix. The usefulness of graph data in the case of the r/Europe dataset is thus in line with the intuition from the previous chapter.

---

4.5. CONCLUSION

We tested a number of baseline models on the various datasets. Overall, a Multilayer Perceptron model trained on the tf-idf features extracted from a user’s comments performs the best. Taking the graph structure between users into account can be useful in certain cases. For example, in the case of the r/Europe dataset, to discriminate between users of different nationality who share the same mother tongue.
Figure 4.2: Normalized confusion matrices when applying a Random Forest classifier on the different datasets.
4.5. CONCLUSION

![Confusion Matrices](image)

Figure 4.3: Normalized confusion matrices when applying a Multilayer Perceptron classifier on the different datasets.
Chapter 4. Baseline Methods

Figure 4.4: Normalized confusion matrices when applying a K-Nearest Neighbors classifier on the different datasets.
Figure 4.5: Normalized confusion matrices when applying an ensemble of a Random Forest and K-Nearest Neighbors classifier on the different datasets.
FIGURE 4.6: Normalized confusion matrices when applying an ensemble of a Multilayer Perceptron and K-Nearest Neighbors classifier on the different datasets.
In this chapter we will examine the GCN architecture proposed by Kipf and Welling [15] and described in Subsection 2.1.3.

This model features a number of hyperparameters: the number of hidden layers, the size of each hidden layer, the learning rate, the dropout rate, the weight decay, the number of epochs the model should be trained and the number of epochs without improvement for early stopping. We will now clarify some of these terms. During training we will, each iteration, randomly disable some of the neurons of the network. As such, the other nodes should be able to “step in” for the missing neurons. This is a regularization technique, i.e. it is used to prevent overfitting. The rate at which we disable neurons is the dropout rate. The weight decay is also used to counter overfitting: it is the scaling factor used for L2 regularization. Put briefly, this penalizes the model when learning large weight values [2].

We trained the hyperparameters on the r/Europe dataset and then reused those parameters for the other datasets. This is similar to the baselines where we also reused hyperparameters for the different models. We started by looking at the values used by Kipf and Welling and the values of the MLP of the previous chapter, and started tuning values from those two starting points.

We first increase the number of epochs to 2000 and the early stopping threshold to 500. The results of our hyperparameter optimization are a learning rate of 0.05, one hidden layer of size 61.
The GCN does not seem to be working better than our baselines. Overall, it performs worse than the MLP baseline. Looking at the confusion matrix of the r/Europe dataset, we see the common mishaps stemming from nationalities with shared languages. They are less pronounced than in the case of the MLP, but more visible compared to the result of the MLP enhanced with the KNN model. We also notice that the GCN did not handle the unbalanced r/r4r age dataset well.
Nationality
r/Europe

Gender
Age

Figure 5.1: Normalized confusion matrices when applying the GCN classifier on the different datasets.
Conclusion and future work

We explored numerous experiments during the year that did not make it into the final dissertation. We will now list some of those and give an indication, based our preliminary or incomplete results, how useful they might be to explore.

We applied the GCN on an early version of the r/Europe dataset, either without the textual features or with an unweighted version of the graph, or both. All of these versions had worse results than the version with weighted edges and features. This makes sense from an intuitive point of view, since in the latter case, there is more information to work with.

We explored additional features, such as time based features. We looked at the time of the day users place their comments. As such, we created a vector of 24 items, indicating the percentage of comments a user made between 0-1 UTC, 1-2 UTC, etc. We concatenated this vector with the result of the tf-idf transform. We also repeated this experiment with quarters instead of hours. Both additions indicate a similar, positive result when used as input features for GCN on the r/Europe dataset. Time based features are thus a good idea to include for nationality classification.

The nodes in the graphs we created have on average a large number of neighbors. We wondered whether this number was maybe too high and the convolution might blur a node’s features too much. We did some preliminary experiments limiting the number of neighbors to their 27
6.1 Conclusion

We succeeded in creating six labeled datasets using publicly available Reddit data. This on its own can already be of interest to privacy-minded Reddit users. Next, we also showed that it is possible to classify most users by age, gender, and nationality using their publicly available comments. Applying a Multilayer Perceptron on the tf-idf transform of these comments proves to be a viable approach. We discovered that in some cases, it can be useful to think of the Reddit users as nodes in a graph. For example, when looking at nationality it is interesting to not only look at which words you use, but also at who you talk with. This is particularly helpful to discern between nationalities that share a mother tongue, such as Austrians and Germans. We demonstrated this by creating an ensemble of a Multilayer Perceptron on the tf-idf transform of a user’s comments with a K-Nearest Neighbors algorithm operating on a graph of Reddit users. We compared this to a Graph Neural Network, a model that also takes both these word and graph properties into account. It did not manage to improve upon our rather strong baseline methods.
Bibliography


