Final Year Project

Inferring User Demographics from Social Media

Author: Christopher Inskip (105957)
Supervisor: Prof. David Weir

University of Sussex, Department of Informatics
Bachelor of Science in Computer Science

2015

Statement of Originality

This report is submitted as part requirement for the degree of Bachelor of Science in Computer Science at the University of Sussex. It is the product of my own labour except where indicated in the text. The report may be freely copied and distributed provided the source is acknowledged.

Christopher Inskip
May 8th 2015

Acknowledgements

First and foremost, I would like to thank my supervisor, Professor David Weir, for both his guidance throughout the project, and for inviting me into such an interesting and challenging area of research. I would also like to thank David Spence for his helpful pointers and discussions over the course of the project. Finally, I would like to express my appreciation to Simon Wibberley and the rest of the Text Analytics Group (TAG), for the development of Method51, and general advice on starting such a project.

Summary

Social networking platforms such as Twitter provide an abundance of user generated content that is of interest to many sectors. Public opinion polling, targeted marketing, and business analytics are just a few applications that can benefit from this large amount of data. However, in order for some applications to make effective use of the data, author demographics are required.

This project explores the techniques required to infer one highly sought-after demographic, namely age, treating the problem as a binary classification task of "under 30" and "over 30". For classification, a Linear Support Vector Machine is employed, trained on features extracted from three publicly available user attributes, specifically, their description field, Tweets, and friends (the accounts that they are following).

One of the major challenges of Twitter analysis is obtaining a high-quality dataset. For this project, an automated annotation technique is designed, using pattern matching to extract explicit declarations of a user's age from their description field. From data collected over a one-week period, the approach was able to construct a dataset containing 62,450 users, with over 97% of the included users being correctly annotated. Although accurate, the approach was shown to result in some bias, most notably a large skew towards the younger age groups.

The experiments explored various combinations of feature sets, and studied the effect that the size of the dataset has on classification performance. The results show that when used alone, friend based features perform competitively to Tweet based features, both obtaining very promising performances. Description based features also showed promise, however, due to the method of creating the dataset, it was found to be paramount to the generalisability of the model that any text used for annotation was not provided to the classifier. Overall, using a combination of all three user attributes obtained the best results, with over 86% classification performance. Due to Twitter's API rate restrictions, current research has largely neglected using friends. However, the results from the dataset size experiments provide evidence that collecting a smaller number of users, but also including friends, may in fact be more effective than building a larger dataset that neglects them.

Contents

1 Introduction

Social networking platforms such as Twitter[1] provide an abundance of user generated content that is of interest to many sectors. Public opinion polling, targeted marketing, and business analytics are just a few applications that can benefit from this large amount of data. However, in order for some applications to make effective use of the data, author demographics are required.

Twitter, unlike other social networks such as Facebook[2], provides very limited demographic meta-data about its users. User attributes such as age, gender, educational attainment, and employment status are not explicitly included as fields. Instead, it is up to the user whether or not to include demographic information, and if so, how to structure it via the use of free-form text fields. Nevertheless, Twitter is of particular interest as the vast majority (over 91% [1]) of its user's profile and communication history is publicly available and freely accessible through Twitter's official API [2].

The focus of this project is to explore techniques of inferring age, a highly sought-after user demographic useful in many applications. Since this task is non-trivial, it is simplified into a binary classification task, employing a linear Support Vector Machine (SVM) to classify users as either "under 30" or "over 30". Current research has shown linguistic features obtained from a user's Tweets and description field to provide promising performance, but using the accounts that a user follows (so called 'friends') has largely been neglected. Therefore, in addition to previously explored linguistic features, this project assesses the value that friend based features have towards inferring age.

One of the major challenges of performing analysis on Twitter is obtaining a high-quality dataset. For this project, an automated annotation technique is designed, using pattern matching to extract explicit declarations of a user's age from their description field. Since the quality of the dataset has a direct reflection on quality of the inference, this project assesses the possible bias that could be caused by the annotation approach, in addition to the amount of incorrectly annotated users included in the dataset.

---

[1] http://twitter.com

[2] http://facebook.com

Additionally, since collecting data from Twitter is subject to API rate restrictions, this project also evaluates how much data is sufficient to build successful inference models, and to which attributes to devote collection time.

Report Structure

Section 2 explains how the project meets the British Computing Society's standards and practices, and explains the ethical responsibilities of undertaking the project. Section 3 gives an overview of the current research in the field, covering the datasets, features, and machine learning models that have currently been explored. Section 4 presents the aims and objectives of the project, providing the main research questions to be tackled. Section 5 goes on to briefly describe the experimentation platform built to aid the undertaking of the project. Section 6 explains the process of dataset creation, with analysis into any dataset bias that may be caused. Section 7 explains the approach to age inference, addressing the age bucketing strategy, feature sets, machine learning model, and how the approach is evaluated. Section 8 presents the experiments and results, with section 9 discussing the results, project limitations, and suggesting areas of future work. Finally, section 10 provides a conclusion to the report.

2 Professional Considerations

Professional and ethical considerations including those specified in the British Computing Society's (BCS) Code of Conduct [18] and Code of Good Practice [19] have been taken into account for this project. The following sections identify the relevant areas in the above documents, and explain how this project will adhere to them.

2.1 Code of Conduct

Public Interest
Labelled datasets of Twitter users could be of potential value to third parties such as advertising or marketing companies. The privacy of any user that has been included in a dataset is of utmost importance. In accordance to section 1.a of the BCS Code of Conduct, all data will be stored securely, and no data that enables a user to be identified will be presented in this project, to the public, or to third parties.

Professional Competence and Integrity
This project is a large part of a final year university degree and will therefore be a challenging experience. However, all of the work that is being carried out is well within the bounds of the course. In accordance to section 2.c of the BCS Code of Conduct, a continuing basis of research and background reading will be performed throughout the project to maintain professional knowledge and competence.

Any alternative viewpoints and honest criticisms of work undertaken on this project will be valued and respected, complying to section 1.e of the BCS Code of Conduct.

As previously expressed, the datasets built to undertake research are of potential value to third parties. In compliance with section 1.g of the BCS Code of Conduct, no offer of bribery or unethical inducement will be accepted.

Duty to Relevant Authority
In compliance to section 3.a of the BCS Code of Conduct, any situation that may give rise to a conflict of interests between this project and the University of Sussex will be avoided. While undertaking work on the project, knowledge of some confidential information such as usernames and passwords is required to facilitate data collection. In accordance to section

3.d of the BCS Code of Conduct, no confidential data will be disclosed or used for personal gain, except with the permission of the University of Sussex, or as required by legislation.

Duty to the Profession
During encounters with members of BCS or other professions that are dealt with on a professional capacity, integrity and respect will be upheld, as declared in section 4.b of the BCS Code of Conduct.

2.2 Good Practices

Although the software developed during this project is not being produced for a client, it is still very important to produce well-written, testable, and ultimately, correct code. Without this quality assurance, the legitimacy of the results collected using the software can not be confirmed. In accordance with section 5.2 of the BCS Code of Good Practice, all code produced will strive to be well structured to facilitate testing and improve maintainability. Section 5.2 of the BCS Code of Good Practice states to follow programming language guidelines. Since the software for this project will be implemented in Python, the PEP 8 guidelines [20] will be adhered to where appropriate.

2.3 Ethical Considerations

Although Twitter encourages researchers to perform analysis on its user's data (and provides many resources[3] which describe how to do so), it is still important to discuss the ethical issues that surround analysing data obtained from Twitter.

Twitter is built around user-generated content, and therefore much of the data collected while undertaking this project will also be user-generated. All data that will be collected is in the public domain, and can be obtained using the official API that Twitter provides. However, users will be unknowingly involved in this study, so as outlined in section 2.1, no user-identifiable data will be presented in this project, to the public, or to third parties.

This particular project is user-centric, and attempts to infer attributes that users have not explicitly provided. The aim of this project is not to reveal attributes about specific users, but instead to reveal in a general sense how data that is publicly and freely available on Twitter can be used to infer

[3] See https://dev.twitter.com/overview/general for links to research projects.

such attributes. However, it must be recognised that the techniques described in this study have the potential be used by third parties in an attempt to infer user attributes of specific Twitter accounts.

The University of Sussex has rated this project as 'low risk' and has approved the undertaking of the research and the manner that it will be conducted.

3 Current Research

Although Twitter is a relatively new platform, research into inferring user attributes on social networks has been rapidly emerging. Related research can be broken down into three main sections: dataset creation, choice of features, and machine-learning approach.

3.1 Datasets

One significant challenge for researchers in the field is the lack of good-quality available training data. When building a dataset, research usually goes down one of two paths, manual annotation or automated annotation.

Manual Annotation
For many applications manual annotation is generally considered to be very accurate, however, it is time consuming, labour intensive, and therefore results in a relatively small dataset.

Some studies approach manual annotation by searching for Twitter profiles that contain external links to more informative sources (e.g. Facebook or LinkedIn) [3, 4]. The information found on these external sites can then help human annotators decide upon an appropriate label. In the case of age, information such as a school grade, or in more hopeful situations, an explicit age can be used [3].

Sometimes 'seed' users are obtained to start the manual annotation process. In one study, seed users were obtained by searching for terms such as "freshman" and "junior" in the user's description field [4]. Ground truth labels for these users are then derived by manually investigating linked external resources as described previously. After the seed users have been annotated, each user's neighbourhood (friends and followers) is explored and annotated in a breadth-first manner. One study argues that only using profiles that provide a URL to more informative social networks leads to bias, stating that it rules out users that do not have/provide such accounts [3].

Automated Annotation
One approach to annotation is to define rules that can extract explicitly defined user attributes contained on external social networks or blogs [5]; this approach essentially automates the manual annotation methods described above. The main difference between the two approaches (asides from time and labour) is that manual investigatory work provides the ability

to be dynamic, as opposed to being constrained to pre-defined rules. [5] found that in general, accuracy increased with the quantity of training data, which resulted in their automatically labelled data providing greater accuracy than their manually labelled data.

A few studies [5, 6] state that when automating the labelling process using regular expressions, a lot of noisy data is obtained. One study argues that when trying to extract a user's age, the age of a user's child may be extracted instead [5]. In studies that use regular expressions [3, 6, 7], only samples of the expressions used in such experiments have been provided, therefore it is unclear how specific the regular expressions were. One study provides an example of a regular expression it used to capture age:

```
(I|i)(m|am|'m)[0-9]+(yo|year old)
```

The above expression, although simplistic, is also grammatically incorrect providing no method to capture the plural, 'years'. Despite being an illustrative example, it provides questions into how specific and well-formed the regular expressions used in these studies actually were.

As the content of Tweets generally reflect current culture and events, inference models may need to be frequently re-trained to keep up with sociocultural evolution. The need for new datasets to be quickly and periodically constructed goes heavily in favour of an automated approach.

When downloading users and their respective Tweets and neighbourhood data (i.e. friends and followers), both manual and automated approaches are subject to Twitter's API rate restrictions. In many cases, the API rate limits have deterred researchers from building datasets that include neighbourhood data. Obtaining neighbourhood data is a higher cost operation (in terms of API calls and therefore time) than obtaining Tweets [9]; this usually means that if neighbourhood data is to be included, a compromise on the size of the dataset is required.

3.2 Features

The performance of a machine-learning model directly reflects the quality of features that it is built from. Social networking platforms provide a multitude of potential features, with the main extension to traditional text-based corpora being that a user's relationship with others can be explored. The types of features that previous research has explored are described below.

User Profile Features

There is a large amount of meta-data contained in a user's profile, most notably the user's biography, name and location fields. In many applications, the expectation would be that a user's bio provides the richest source of demographic information. However, on Twitter, a study has shown that a user's bio is left blank in 48% of users, and many do not provide enough good-quality information as the exclusive provider of features [6].

In addition to the three attributes described above, there are over 40 more pieces of profile data associated with each user [10]; much of which has not been explored in current literature. However, user assigned profile colours have been used as an attempt to infer gender with some success [11], and information about the number of friends, followers and Tweets have also been trialled as features [7, 12].

Linguistic Features

Many studies have found success in inferring age using simple n-gram features extracted from a user's Tweets [3, 4, 5, 7, 8, 12, 13, 14, 15, 16]. Although not Twitter specific, one study found bigrams to yield greater performance at inferring age than unigrams; they suggest this is due to bigrams containing more semantic information [12].

Sociolinguistic features have been demonstrated to differ between age, gender, and social class [17]. One study redefines some of the traditional sociolinguistic cues in order to adapt to the social-networking domain [4]. Cues such as emoticons (e.g. ':)'), affection (e.g. 'xoxo'), laughs (e.g. 'haha' and 'lol'), and excitement (e.g. '!!!') were shown to have different usage in varying age and gender [4].

Demographic Features

Some studies [3, 8, 15] have found that including demographics as features (i.e. including a user's gender when attempting to infer age) has shown to slightly increase performance towards inferring other demographics of interest.

Statistical Features

Extending the feature set to include a range of user and Tweet statistics has also been explored in some studies [7, 8, 12, 16]. Statistical features include the number of Tweets, mentions, hashtags, links, and Retweets, along with ratios of Tweets to Retweets, and the number of friends and followers in the user's neighbourhood.

3.3 Machine Learning Approaches

The majority of research treats age inference as a classification task, however, treating the problem as a regression task has also been explored [2]. When deciding upon classes, many user attributes are self-evident (e.g. gender having two classes, "male" and "female"). However, age permits many possible bucketing strategies, a variety of which have been explored in current research; these classes are outlines below.

- **Two classes:** 18-23 and 25-30. [7]
- **Two classes:** under 30 and over 30. [4]
- **Three classes:** under 20, 20-40, and over 40. [3]
- **Three classes:** 10s, 20s, and 30s. [14]
- **Four classes:** 0-18, 18-24, 24-35, and 35+. [12]
- **Four classes:** 10s, 20s, 30s, and 40s. [5]

As shown above, research generally focuses on distinguishing between users within younger age groups, and tends to ignore distinguishing between age groups past 30. There is a justified reason for focusing on younger age groups, as it has been shown that most changes in language occur at young ages; this makes it hard for both humans and automated systems when classifying within older age groups [3].

Model

The reviewed research generally accepts that for this domain, Support Vector Machines (SVMs) consistently outperform other commonly used models, such as Naive Bayes and Gradient Boosted Decision Trees. For this domain, the dimensionality of features commonly becomes very large (well into the millions [4]); it is therefore important to choose a classifier (such as SVMs) that can cope with the magnitude of such feature sets.

4 Project Aims and Objectives

The project aims to answer four main research questions. To aid research, a framework will be developed, providing dataset annotation to be performed, and allowing experiments to be quickly configured.

4.1 Research Questions

RQ1. How viable is dataset creation based on automatically annotating users using their description field?

Automating the dataset creation process has huge advantages over manual annotation, saving both time and labour efforts, and ultimately resulting in a larger dataset. However, much of the reviewed research is sceptical of the quality of the datasets produced by automated techniques.

In an attempt to answer this question, the amount of noise (i.e. incorrectly annotated users) that the dataset consists of will be assessed. Improvements to the pattern matching techniques will be made where appropriate, helping to improve the quality of the annotation approach, and by extension, the dataset. Viability not only refers to the amount of noise, but also to how representative the dataset is of the actual domain. As such, attempts will be made to identify any biases that the annotation approach includes.

RQ2. Are the accounts that a user follows valuable for age inference?

Most of the current research finds that linguistic features (such as n-grams from Tweets) are informative towards age inference; however, including who the user follows in the feature set has been largely overlooked. Intuitively, the accounts that a user follows could be considered very indicative of their age, since it may directly reflect their personal interests.

RQ3. What effect does combining friend based features with typical linguistic features have on classification performance?

Although the accounts that a user follows (i.e. friends) may be useful towards age inference, linguistic features (such as unigrams and bigrams from a user's Tweets) have already been shown to be relatively successful. Therefore, the effectiveness of combining

different linguistic features with a user's friends will also be explored.

RQ4. What effect does altering the dataset size have on classification performance?
Twitter's API rate restrictions can have a large impact on the amount of data that can be collected within a set timeframe. It is therefore useful to know how much data needs to be collected before successful inference models can be constructed.

4.2 Experimentation Framework Aims

The main aim of the experimentation framework is to provide the ability to:

1. Automatically build annotated datasets using pattern matching techniques.
2. Display statistics about the composition of the constructed datasets.
3. Allow user-defined feature extraction algorithms to be applied to a dataset.
4. Integrate with existing machine learning and natural language processing frameworks to build classification models.
5. Allow experiments to be configured and run in batch.
6. Log the experiment's output, including predictions, results, and a human-readable overview of the experiment.

5 Experimentation Framework

The system is designed purely to aid research, and is therefore not the focus of the project. Nevertheless, the development of the system is still a large part of the project in terms of both time and effort, and is therefore included as part of the report. The following sections provide functional and non-functional requirements, an overview of the system design, and a general description of how it aids experimentation.

5.1 Requirements

The framework shall provide the functionality to:

- Use regular expressions to construct annotated datasets from data stored in a MySQL[4] or MongoDB[5] database.
- Produce datasets that can be stored in memory or a MongoDB database.
- Allow the full array of user meta-data [10], Tweets, friends, and followers to be included in the dataset.
- Perform dataset re-labelling and class size resampling.
- Allow feature extraction algorithms to be defined and applied to a dataset.
- Integrate with the existing machine learning framework, scikit-learn [27], to build classification models.
- Allow experiments to be configured and run in batch, and record the experiment's output. The output shall include predictions, results (e.g. accuracies, precisions, recalls, F-scores), and a human-readable overview of the experiment.

[4] https://www.mysql.com

[5] http://www.mongodb.com

5.2 System Design

5.2.1 Programming Language

Python has been chosen as the language for implementing the system. One of the main reasons to use Python is to integrate with existing and powerful machine learning frameworks such as scikit-learn [27]. However, since the framework is largely modular and user-definable, with a lot of the framework being written per experiment, a highly expressive and non-verbose language such as Python also minimises the amount of code needed to implement features.

5.2.2 Persistent Data Store

MySQL is adequate for storing the raw data obtained from Twitter, but when building and working with a dataset, it does not provide the flexibility that of a document-based database such as MongoDB can provide. Since MongoDB adopts a dynamic schema design, the underlying data structure can be freely modified; this is particularly useful for dataset creation as it allows additional fields such a class labels to be included. An example MongoDB user entry can be seen below in Figure 5.1.

Figure 5.1 – Example MongoDB User Document

```
{
    "id": "123456789",
    "label": "1979",
    "attributes": {
        "name": "John Smith",
        "screen_name": "JSmith"
        "description": "I'm 35 years old and have 2 kids.",
        "lang": "en",
        ...
        "tweets": {
            "1902901321": {
                "text": "So tired, need coffee!!! :O",
                "is_retweet": 0
            },
            ...
        }
        "friends": [
            "123432342",
            "84893758",
            ...
        ]
    }
}
```

5.2.3 Dataset Creation Tools

The first stage of the dataset creation process is to combine the large amount of data collected using Twitter into a single MongoDB database. MySQL or MongoDB databases containing Tweets, users, and friends can all be supplied to the dataset builder, allowing a single (unlabelled) dataset to be created. A high-level diagram of this process can be seen below in Figure 5.2.

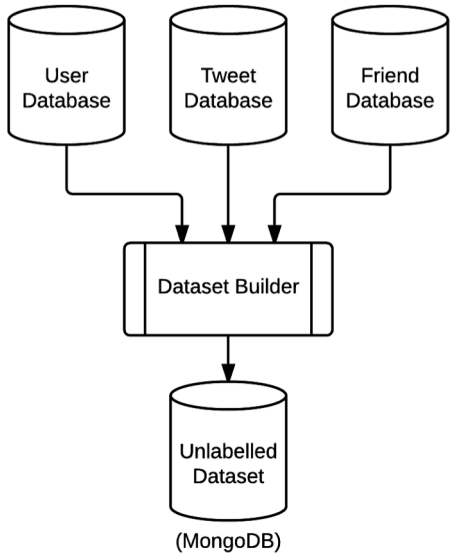

Figure 5.2 – Building an un-annotated dataset from raw Twitter data.

The next stage is to annotate the users via a pattern matching technique. The system uses a pattern matching configuration file using Python regular expressions to filter and annotate the dataset. This configuration file works in conjunction with a user-defined labelling function to aid the annotation and filtering process. This process can be seen below in Figure 5.3.

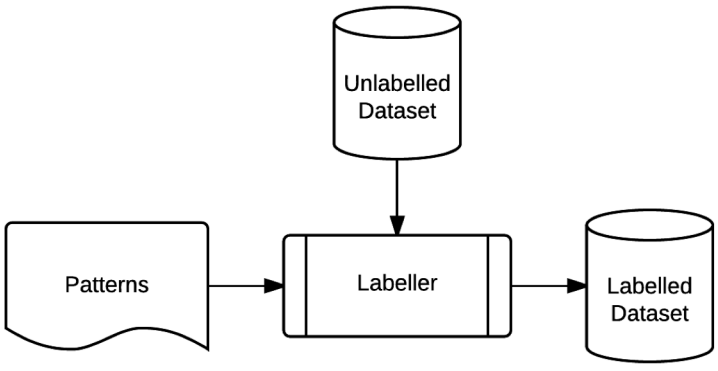

Figure 5.3 – Annotating a dataset using a pattern configuration and a labelling function

An example pattern matching configuration file can be seen below in Figure 5.4. The first two lines indicate that if the pattern is matched then the user should not be included in the dataset. The next two lines indicate that if the pattern is matched then the labelling function should extract the capturing group and treat the extraction as an 'AGE'. The final two lines indicate that if the pattern is matched then the labelling function should extract the capturing group and treat the extraction as a 'DOB'.

Figure 5.4 – Example Pattern Matching Configuration File

```
DO_NOT_INCLUDE
i'?m [0-9][0-9] years old in (dog|cat) years

AGE
i'?m (1-9][0-9]?) years old

DOB
Born in (19[2-9][0-9])
```

5.2.4 Machine Learning Experiment Platform

One of the main reasons to implement the experimentation platform is to provide a simple way of exploring different machine learning approaches without having to write large amounts of boilerplate code for each experiment. The platform allows an experiment to be configured by providing a set of configuration files; these files detail the dataset, classes, features, machine learning model, and type of evaluation and validation to use. The general system schematic can be seen in Figure 5.5 below.

Scikit-learn, a popular machine learning framework, is at the heart of platform, providing the functionality to train and test classifiers, and assess their performance. Although scikit-learn also provides the ability to perform commonly required tasks such data partitioning and feature extraction, when working with complex dataset structures (such as those obtained from Twitter) a lot of additional code is required to set up an experiment.

Twitter provides researchers with a large amount of features to explore. Having a platform that allows user-attributes to be easily processed and trialled as features provides the researcher with a greater inclination to explore the Twitter domain. Bespoke feature extractors can be written and integrated with very few lines of code; an example feature extractor that simply extracts the user's first name can be seen below in Figure 5.6.

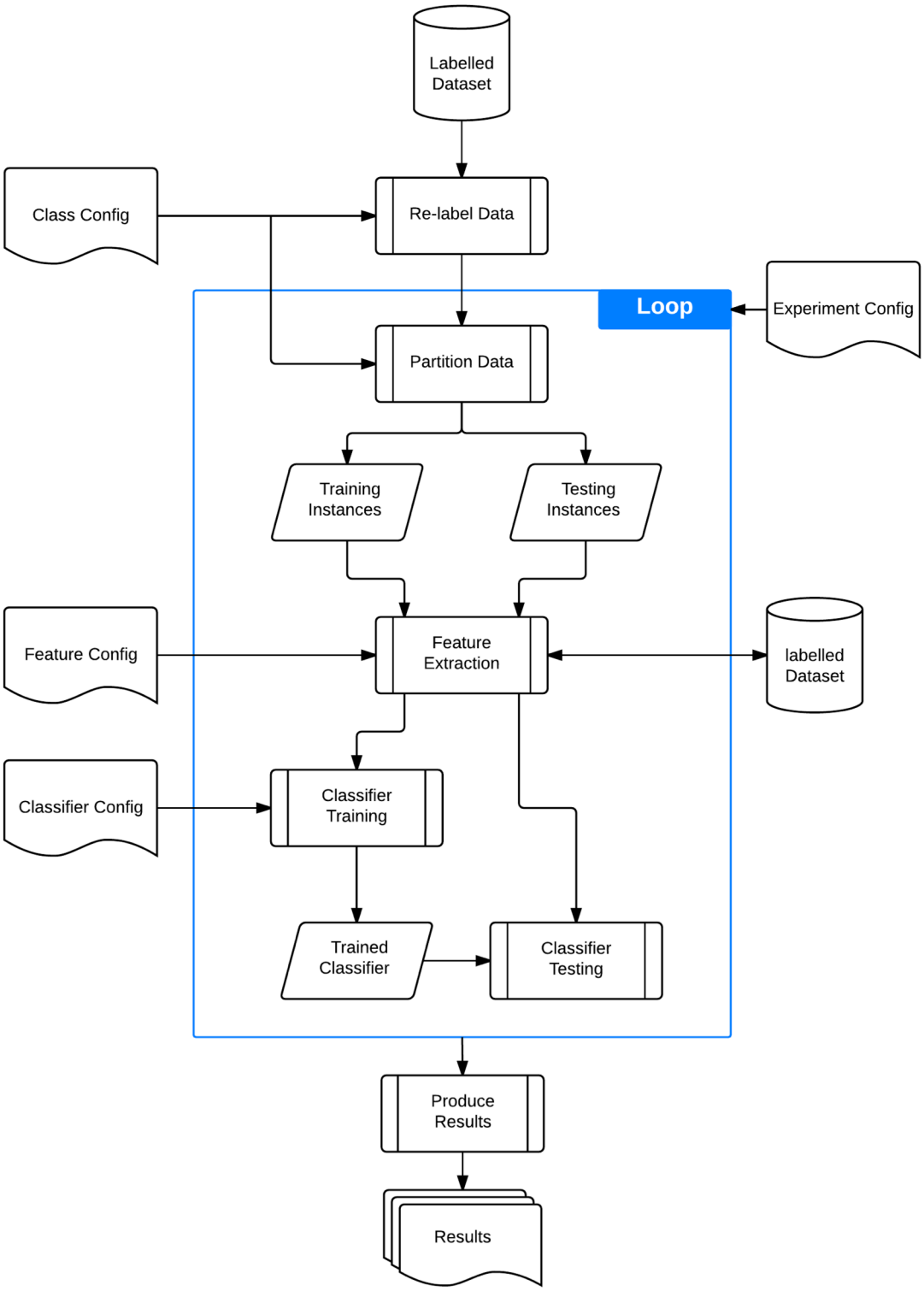

Figure 5.5 – Overall schematic of the experimentation platform

Figure 5.6 – Example First Name Feature Extractor

```
def extract_features(name):
    # Split name by whitespace and use first element
    return ['first_name=' + name.split()[0]]
```

5.2.5 Platform Usage

This section gives a small example of the main configurations required to run a simple experiment. In this example, the experiment uses a user's first name to predict whether they are under or over 18 years old, and a Linear Support Vector Machine, which will have its parameters optimised via a grid search. These configurations are expressed in JavaScript Object Notation (JSON) and can be seen below if Figures 5.7, 5.8, and 5.9.

Figure 5.7 – Example Class Configuration

```
{
 "classes": [
   {"class": "18-", "range_upper": 2004, "range_lower": 1997},
   {"class": "18+", "range_upper": 1996, "range_lower": 1920}
 ]
}
```

Figure 5.8 – Example Feature Configuration

```
[
  {
    "attribute": "name",
    "extractor_module": "firstname_feature_extractor",
    "extractor_function": "extract_features"
  }
]
```

Figure 5.9 – Example Classifier Configuration

```
{
  "classifier": "LinearSVC",
  "params": {
     "penalty": "l2"
  },
  "tuning": [{
      "C": [0.01, 0.1, 1.0],
      "tol": [0.001, 0.01, 0.1]
  }]
}
```

6 Dataset Development

Publicly available datasets for the Twitter domain are relatively scarce, and in terms of an age-annotated dataset containing users, Tweets, and friends, they are non-existent. One of the main reasons for the lack of datasets, is that although Twitter is lenient on data collection, they have strict rules on how the obtained data can be shared [28]. As such, this project requires a dataset to be created.

The process of building the dataset can be broken down into three main stages:

1. Obtaining a large random sample of Twitter users.
2. Filtering users and annotating them with their age.
3. Collecting additional data for each of the annotated users.

Section 6.1 describes the data collection process, explaining how the initial random sample of users is obtained, and how Tweets and friends are collected for the annotated users. Section 6.2 describes how the users are annotated with their ages, and assesses the proportion of incorrectly labelled users present in the dataset. Section 6.3 breaks down the composition of the final dataset, with section 6.4 attempting to identify any bias that the annotation method may have led to.

6.1 Data Collection

To build an annotated dataset, raw Twitter data is required. Users data in the form of their profile, Tweets, and friends are the basis for this project, and to obtain this data, Twitter's streaming [21] and REST [22] APIs need to be used. Although the API is relatively user-friendly and well documented, to simplify the process, data is collected via the application 'Method51' where possible.

Method51 is a software platform that facilitates social media analysis [23]. For this project, a subset of the platform is used to aid user sampling and tweet collection. Unfortunately, at the time of collection it was not possible to collect friends using Method51, so instead, a simple python script was developed and utilised.

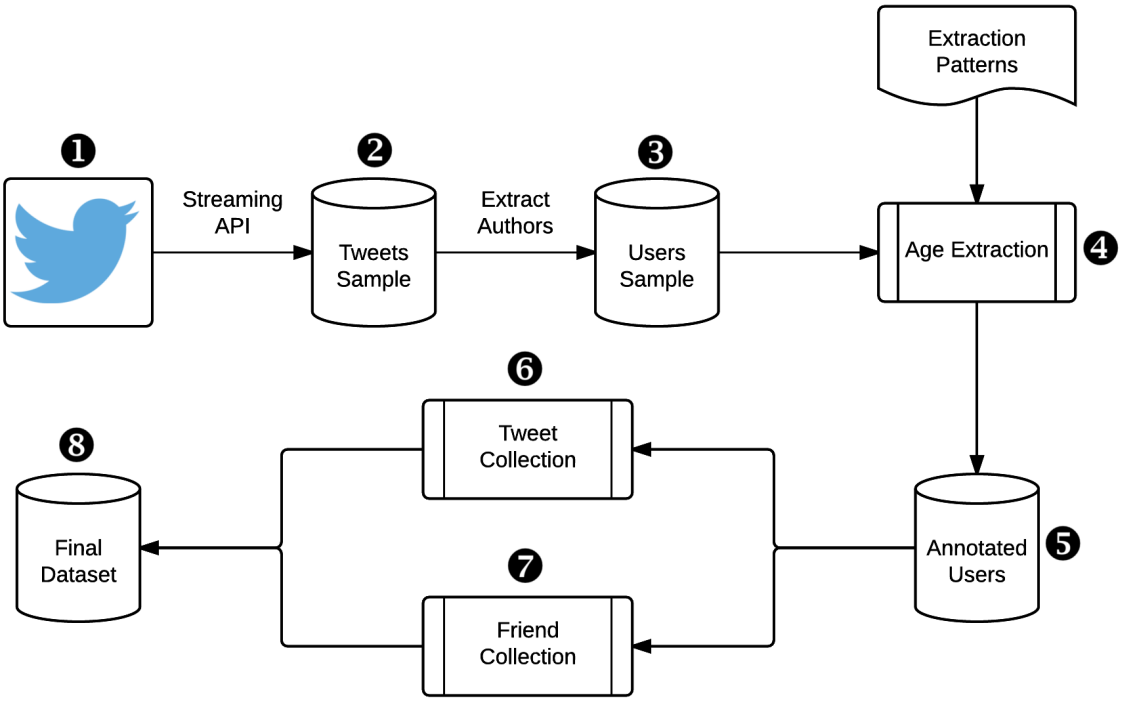

Figure 6.1 — The dataset creation process[6]

Figure 6.1 above shows the process of building the dataset, the first stage of which is collecting a large sample of users. Over the course of a week, Method51 was configured to constantly stream a 1% random sample of Tweets with a filter for the English language ❷. Each tweet contains the author's full Twitter profile, which is extracted and stored, forming a user database of 9,627,866 unique users ❸. An attempt is then made to annotate each user with their age ❹, the details of which is described in section 6.2 below. For each successfully labelled user ❺, up to 200 of their most recent Tweets ❻ and 3,600 of their friends ❼ are collected. This results in the final age-annotated dataset containing 66,450 users that is used throughout the project ❽.

It is worth noting that API rate restrictions have a large impact on the data collection process. With Twitter's current rate limits, requesting Tweets permits 720 users to be processed per hour, and requesting friends permits 60 users to be processed per hour. This means that even building datasets of modest sizes can require several weeks to construct if friends are being included.

[6] Twitter bird logo: https://about.twitter.com/press/brand-assets

6.2 Automated Annotation

To build the annotated dataset, each user's description field is processed in an attempt to find any explicit declarations of their age. To automate the approach, regular expressions are employed to match and extract the user's age or year of birth based on a predefined set of patterns. Pattern design is a critical stage of the automated annotation process, since the quality of the patterns are directly responsible for the quality of the dataset, and by extension, the quality of the machine learning approach. This section describes the process of pattern development for age extraction, and assesses how much noise (i.e. how many incorrectly annotated users) the patterns have included in the dataset.

6.2.1 Methods of Explicitly Declaring Age

After manually analysing the raw data, five main methods that users take to declare their age were identified. These are as follows:

1. Stating how old/young someone is in years. (E.g. "24 years old")
2. Stating a year of birth. (E.g. "Born in 1963")
3. Stating that someone has aged by an amount. (E.g. "aged 39")
4. Stating that they are a particular number. (E.g. "I'm 18")
5. Providing a stand-alone number in their description. (E.g. "25.")

Methods 1-3 are all explicitly age related, and method 4 is semantically linked to the author of the description. However, although in many cases method 5 appears to be age related, there is no clear indication that the number located at the start of the description field is in fact related to their age. For this reason, only patterns based on methods 1-4 are designed.

6.2.2 Designing the Patterns

To design the patterns, three development iterations are carried out. For the first iteration, a set of patterns are designed based on findings 1-4 described above. The quality of the patterns are then assessed by determining the amount of incorrectly annotated users being captured. These patterns are then modified according to the findings from the previous iteration, and the amount of noise is analysed again. This refinement and analysis process is carried out one last time, resulting in the final dataset that will be the basis for model evaluation. All three iterations of patterns can be seen in appendix A.

6.2.3 Methodology

From a preliminary experiment with simplistic patterns, it was found that the dataset may contain close to 100,000 users. In these situations it is implausible to manually assess the noise of the entire dataset, and instead it is common practice to assess performance on a random sample of instances. One problem with taking a random sample over the entire dataset, is that if the dataset is heavily skewed towards a certain age group, the overall noise may be low even if some age groups are entirely comprised of noise. To combat this, a fixed sized random sample can be taken for each decade, resulting in a performance that is less biased by the age distribution of the dataset.

The development and analysis process is as follows. Firstly, the dataset is split into 10 subsets according to decade, spanning from the 1920s to the 2010s. For each of these subsets, 50 users are randomly sampled, and their description fields along with their extracted year of birth are stored for manual analysis. An example for the "1990s" age bracket is as follows:

Correct	19	"I am 19 years old and I go to University."
Correct	17	"I go to college and I'm aged 17."
Incorrect	18	"I can't wait until I'm 18 years old!"
⋮	⋮	⋮

After manually annotating whether or not each user has been correctly labelled, the proportion of noise can be calculated, the patterns can be refined, and the process can be repeated.

6.2.4 Results

First Iteration

Table 6.1 below shows that the average noise per decade is 30.8% and predominantly comes from the highest and lowest age groups. In the 1920s and 1930s subsets, 64% and 56% of the data is noise, respectively. However, the 2010s decade is virtually all noise with 90% of its instances being incorrectly included.

Table 6.1 – Resulting noise for iteration 1

Age Range	Number Correct	Number Incorrect	Noise (%)
1920-1929	18	32	64.0
1930-1939	22	28	56.0
1940-1949	39	11	22.0
1950-1959	42	8	16.0
1960-1969	38	12	24.0
1970-1979	44	6	12.0
1980-1989	48	2	4.0
1990-1999	49	1	2.0
2000-2009	41	9	18.0
2010-2019	5	45	90.0
Average	—	—	**30.8**

Causes of Noise

There are many reasons that noise is being included in the dataset, for the youngest age bracket (2010-2019) the patterns are matching users that state things such as "I'm 2 cool 4 school" and "I'm 1 of a kind". It is worth noting that there were a few users that are in fact under 5 years old! However, many of these accounts seem to be either created on behalf of a child or pet. In terms of refining the patterns, a few modifications can be made, but it is probably more appropriate to exclude this age bracket from the dataset entirely.

For the oldest age brackets (1920-1929 and 1930-1939), much of the noise is coming from users stating things like:

- "I act like I'm 90 years old"
- "I'm 85 years old, stuck in a 20 year olds body."
- "Carer for my 87 year old mother."
- "Maybe I'll figure out what I'm doing when I'm 80"
- "bitter 82 yr old trapped in a 43 yr old's body"

From this analysis, it is evident that many of the users can be turned into usable data. This can be achieved by capturing the second part of the description instead of the first for descriptions such as "82 yr old trapped in a 43 yr old's body". It is also apparent that a large amount of noise just requires specific corner cases to be taken into account, such as in the following examples:

- "Programming since I was 14 years old"
- "Broke my leg jumping down the stairs when I was 12 years old"
- "I'm in love with a 37 year old"
- "10 yr old daughter"

Second Iteration

Table 6.2 below shows that the average noise per decade has been reduced by around 5% between the first and second iterations. However, there are still many cases of noise that can be dealt with.

Table 6.2 – Resulting noise for iteration 2

Age Range	Number Correct	Number Incorrect	Noise (%)
1920-1929	19	27	58.7
1930-1939	30	20	40.0
1940-1949	39	11	22.0
1950-1959	46	4	8.0
1960-1969	48	2	4.0
1970-1979	45	5	10.0
1980-1989	46	4	8.0
1990-1999	48	2	4.0
2000-2009	44	6	12.0
2010-2019	5	45	90.0
Average	—	—	**25.7**

Causes of Noise

The majority of the noise still occurs in the limits of the dataset. This iteration uncovers a number of cases were the patterns can be refined for the next iteration. An example of such noise can be seen below.

- "I'm 70 years old in cat years"
- "I'm 50 shades of perfect"
- "... before I'm 50 years old"
- "knees of an 80 year old"
- "started a business by age 16"
- "I'm 72 inches tall"

Third Iteration

As shown in Table 6.3 below, the averaged noise across all decades has been reduced again by a further 7.4%, resulting in a final un-weighted noise of 18.3%.

Table 6.4 below shows the noise when weighted according to the actual age distribution in the dataset, which suggests an overall noise of 2.38%. As previously stated, the 2010 decade contains a large amount of noise, much of which is tedious to filter out using pattern matching techniques. By removing all instances that are in this bracket (i.e. all users labelled under five years old), per decade noise can be reduced from 18.3% to 12.6%, resulting in the noise of the final dataset being 2.17%.

Table 6.3 – Resulting noise for iteration 3

Age Range	Number Correct	Number Incorrect	Noise (%)
1920-1929	20	13	39.4
1930-1939	38	12	24.0
1940-1949	47	3	6.00
1950-1959	45	5	10.0
1960-1969	44	6	12.0
1970-1979	48	2	4.00
1980-1989	46	4	8.00
1990-1999	50	0	0.00
2000-2009	45	5	10.0
2010-2019	15	35	70.0
Average	—	—	**18.3**

Remaining Causes of Noise

There are still a few improvements that can be made, however three development iterations has allowed the patterns to exclude much of the generalisable noise.

- "A 12 year old scotch and a good beer is very important to me"
- "Providing education to students aged 11 - 19"
- "Carer to Frank (90 year old war veteran)"

Table 6.4 – Weighted noise for iteration 3

Age Range	Weighted Noise (%)
1920-1929	0.0199
1930-1939	0.0260
1940-1949	0.0126
1950-1959	0.0442
1960-1969	0.107
1970-1979	0.0733
1980-1989	0.539
1990-1999	0.00
2000-2009	1.35
2010-2019	0.214
Sum	**2.38**

6.3 Dataset Composition

Figure 6.2 below shows the dataset's age distribution. A table of frequencies for each year of birth is provided in appendix B. Table 6.5 below outlines relevant attributes of the dataset, such as number of users, Tweets and friends.

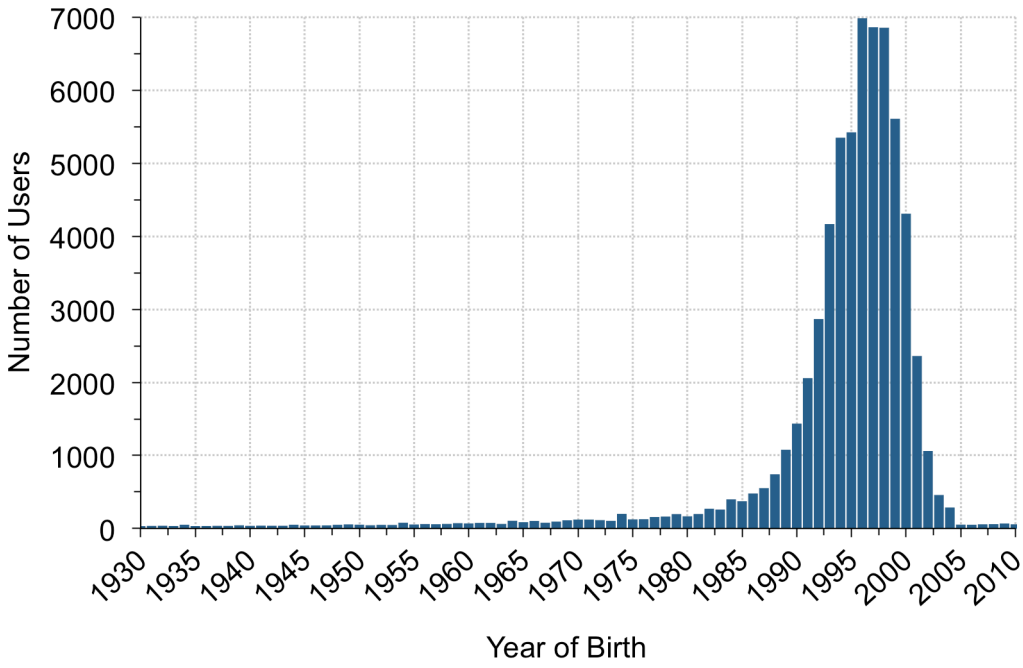

Figure 6.2 – Age distribution of the final dataset

Table 6.5 – Attributes about the dataset

Dataset Attribute	Value
Number of annotated users	62,450
Total number of Tweets	11,815,835
Avg. number of Tweets per user	189
Number of original Tweets	7,822,421
Number of Retweets	3,993,414
Total Tweet vocabulary	6,955,132
Total number of friends	41,277,051
Avg. number of friends per user	660
Total number of unique friends	14,909,005

6.4 Dataset Bias

One of the main aims of the project is to assess whether automated annotation via pattern matching is a viable solution for dataset creation. Since inference models will be trained and evaluated on the dataset, the dataset should be as close to a representative sample of the domain as possible.

Figure 6.2 above shows that the age distribution of the dataset is heavily skewed towards the younger age groups. Unfortunately, the true age distribution of Twitter is not known, so instead, independent social media research carried out by PEW [29] is commonly used as a guideline for the Twitter population. Intuitively, it seems reasonable that Twitter is a platform highly dominated by younger users, however, PEW's research finds the skew to be much less prominent. Figure 6.3 below shows the differences in age-distribution between the annotated dataset and that found by PEW's research.

As shown by Figure 6.3 below, there is a large difference between PEW's age distribution and the annotated dataset's, particularly in the oldest age groups. One possible reason for this difference is due to the labelling method. Since the technique of pattern matching requires users to explicitly express their age, as users get older they may be less inclined to do so.

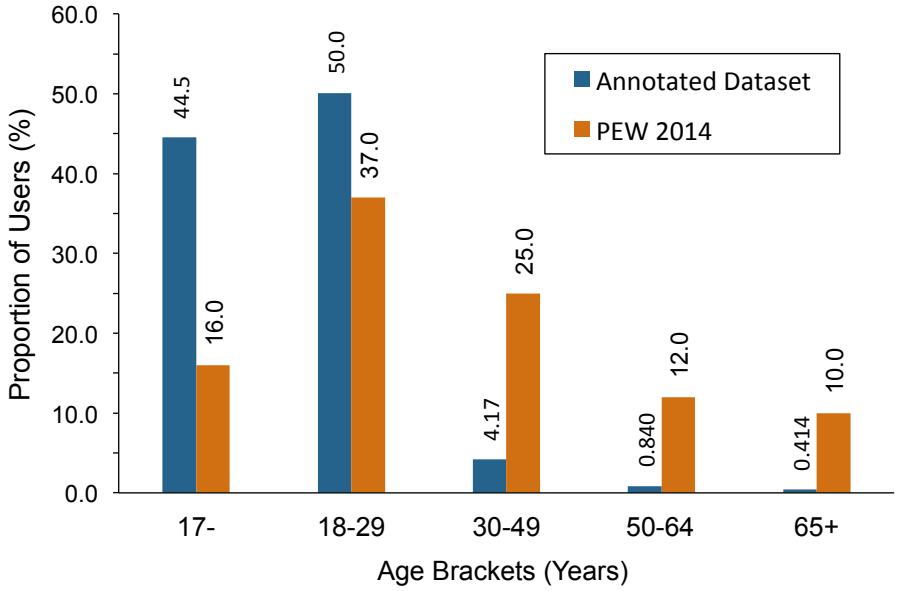

Figure 6.3 — Comparison between the age distribution of the annotated dataset and PEW's Twitter demographic research

However, it is not the percentage of users matched that is the true concern, since a small percentage can still translate into a large amount of Twitter data. Instead, it is the 'types' of users being included that could cause a potential problem. Since users in the dataset have all shown the inclination to reveal their age, they may also be uncharacteristically revealing about other personal attributes. These types of users may not be representative of the true Twitter population, potentially causing classifiers to heavily rely on features that are not often seen outside of the dataset, and ultimately resulting in poor generalisation. The labelling method also inherently means that all users in the dataset have actively filled in their description field. However, out of all users in the initial sample, 17% of them had empty descriptions; this is another potential cause for overly optimistic results.

The sampling period is also another area of bias. Since users were collected via streaming Tweets over a span of one week, it is probable that the dataset mainly contains the types of users that tweet frequently. Additionally to this, events during the collection period (i.e. world events,

TV shows, celebrity birthdays) may encourage certain types of users to be more active on Twitter. Due to the relatively short collection period, these aspects may have had a large affect on the types of users included in the dataset.

7 Approach

As with the majority of current research, this project simplifies the task of age inference into a classification problem. The dataset is partitioned into classes of "under 30" and "over 30" as described in section 7.1. For the classification model, Linear Support Vector Machines (SVMs) are employed, with their hyperparameters optimised via a grid-search, as described in section 7.2. For each experiment, the model is trained on feature sets based on a user's friends, Tweets, and description field; the details of such are described in section 7.3. To validate the results, a cross-validation approach known as Repeated Random Sub-Sampling Validation is used, training on 70% of the data, testing on the remaining 30%, and repeating the process 10 times; this validation method is detailed in section 7.4.

7.1 Age Bucketing Strategy

As previously discussed in section 3, current research has attempted many different approaches to bucketing users by their age. Since PEW's research suggests that the true distribution of Twitter users is centred at age 30 [29], the users will be bucketed into classes of under 30 and over 30. However, since the dataset is heavily skewed towards the younger ages, partitioning the dataset at age 30 results in a huge class imbalance of 58,752 under 30s and 3,698 over 30s. To address the class imbalance, the majority class will be undersampled to match the minority class, allowing the class sizes to line up with that seen by PEW. To help avoid wasting the majority of the annotated data, the classifier validation process known as Repeated Random Sub-Sampling Validation described below will be used.

The main statistics about the resampled dataset are shown in Table 7.1 below. Since a random sample of the majority class is being used for each training/testing iteration, an asterisk (*) is used to represent the average taken over 10 random samples.

Table 7.1 – Statistics for the under 30s/over 30s dataset

Dataset Attribute	Under 30s	Over 30s
Number of users	3,698	3,698
Total number of Tweets	699,122*	699,739
Total number of friends	2,398,529*	2,914,837

7.2 Machine Learning Model

As discussed in section 3, current research has generally found Support Vector Machines (SVMs) to provide the best performance on this domain. For this task, a Linear SVM is used. A Linear SVM is a geometric model, meaning that instances are represented as points is space. Training the model involves finding a hyperplane that *optimally* separates the instances that belong to different classes. This optimal separation is defined as a hyperplane that provides a maximal margin between itself and the instances of different classes. The instances that influence the hyperplane are called support vectors, with the number of support vectors being controlled by the regularization parameter. The regularization parameter has been shown to heavily influence the model's performance [24], therefore, when training the model, a grid search using 3-fold cross-validation on the training data will be performed, helping to find the optimal values.

There are two main reasons why SVMs have been employed for this task. Firstly, SVMs can handle both a very large and very sparse feature space; this is extremely desirable since the feature dimensionality will be very large for this domain. Secondly, SVMs are typically resistant to overfitting, even when the number of features overshadows the number of training instances.

7.3 Features

7.3.1 Friend Features

In this context, friends refer to the Twitter accounts that a user is following. Friend features are simply the IDs of these Twitter accounts; no other information about these accounts is included. No pre-processing or feature selection is performed on these feature sets, which means that accounts that are only being followed by a single user are still included. This is done for two reasons; firstly, it minimises the risk of knowledge about the testing set leaking into the model, and secondly, although these accounts will not be useful for this dataset, they may be useful when applied in practice, so keeping them allows the SVM's ability to cope with the scale of friend features to be tested.

7.3.2 Description Features

When extracting features from the description field, it is crucial that the document is pre-processed to remove all text that was used to annotate the instance. Without removing this text, the classifier may find that the text

used for annotation provides near perfect classification. However, since users that reveal their age in this way represent less than 1% of Twitter, a model built using these features is highly unlikely to generalise. Text from the description field is lowercased and then tokenised using the "Twokenizer", a Twitter specific tokeniser aimed to tokenise text such as emoticons, hashtags, and URLs more appropriately [25]. From the tokenised text, unigrams, bigrams, and a combination of the two are trialled as features. Each feature is uniformly weighted, however, the raw feature frequency extracted from each user's description field is also provided to the model.

Stopwords are commonly removed when using text based features, however, the notion of a stopword is not clear for this setting, since even tokens such as "the" may show significant variation between age groups. Therefore, no stopword removal is performed for this task. As with friend features, no feature selection is performed on the description features. These decisions will result in a greater number of features, however, since the SVM can handle a large feature dimensionality, the SVM is given the power to decide how to utilise such features.

7.3.3 Tweet Features

Similarly to description features, for Tweet based features, each Tweet is lowercased and then tokenised using the Twokeniser. Unigrams, bigrams, and a combination of the two are trialled as features, with no stopword removal or feature selection taking place. Since only the description field is used to annotated the users, no such pre-processing steps are required for the Tweets. As with the description features, each Tweet feature is uniformly weighted, with the raw feature frequency extracted from each user's Tweets also being provided to the model.

7.4 Model Evaluation

7.4.1 Performance Metrics

To measure the classifier's performance, a standard F_1 score will be used. F-measures were introduced to give a performance metric that takes both precision and recall into account. An F_1 score gives both precision and recall equal weighting, and is therefore considered as the harmonic mean between precision and recall. The formula for the metric is shown below, with a perfect classification score obtained at 1, and a worst score obtained at 0.

$$F_1 = 2 \cdot \frac{(precision \cdot recall)}{(precision + recall)}$$

7.4.2 Repeated Random Sub-Sampling Validation

When performing model evaluation, a technique called repeated random sub-sampling validation is used. This is a similar approach to typical strategies such as k-fold cross validation, but allows greater flexibility when partitioning the dataset. It works by randomly splitting the dataset into training and testing data, building and optimising the model on the training data, and then evaluating its performance on the testing data. This process is repeated and then an average is taken.

The experiments in this project undersample the majority class, a technique allowing the dataset to contain an equal number of instances in each class. Cross validation strategies such as k-fold are not ideal in this situation, since the initially undersampled class is used for all fold combinations. Repeating the experiment may cause differing results due to the undersampled class consisting of a different initial sample of instances. In this scenario, repeated random sub-sampling validation is therefore preferable to k-fold since each random sub-sample can be drawn from the original dataset. Not only does this increase the consistency and reproducibility of the experiments, but it also allows the size of training-testing splits to be independent of the number repetitions. When performing experiments, the dataset will be split into 70% for training and 30% for testing, and 10 repetitions will be performed. It is important to note that the same 10 random samples will be used for each experiment, via the use of seed states.

8 Results

8.1 Baseline Performance

Since both classes have been resampled to contain the same number of instances, a model that always predicts users as "over 30", always predicts users as "under 30", or predicts randomly, will manage to predict around 50% correctly. Therefore, the baseline performance for this task is 0.5.

8.2 Independent Feature Sets

For the first set of experiments, friend, description, and Tweet based features are trialled independently from one another. The results for the individual feature sets can be seen below in Table 8.1, and the dimensionalities of the respective feature sets can be seen below in Table 8.2. A preliminary experiment was performed to see the effect of not removing text used for annotation. Using description unigrams the model obtained near performance of 0.96; the most informative features for this model can be seen in appendix C.

Table 8.1 – Performance (± standard deviation) obtained from independent feature sets

Feature Set	**F_1 Score (3 s.f.)**
Friends	0.832 (± 0.00796)
Description unigrams	0.791 (± 0.00647)
Description bigrams	0.738 (± 0.00678)
Description unigrams + bigrams	0.795 (± 0.00782)
Tweet unigrams	0.820 (± 0.00712)
Tweet bigrams	0.833 (± 0.00445)
Tweet unigrams + bigrams	0.835 (± 0.00537)

Straight away, the value of using friend features can be seen, with a very similar performance obtained from Tweets. Interestingly, unigrams are superior for description based features, whereas bigrams are superior for Tweet based features.

For both Tweet and description based features, combining their unigram and bigram counterparts did not significantly boost classification performance, with an increase of 0.2% and 0.4% for Tweet and description

feature sets respectively. Since combining both unigrams and bigrams greatly increases model complexity yet only results in a very small performance gain, description bigrams and tweet unigrams will not be used in the remainder of the experiments.

Table 8.2 – Average training dimensionalities of independent feature sets

Feature Set	Number of Features
Friends	2,071,189
Description unigrams	16,092
Description bigrams	54,597
Description unigrams + bigrams	70,689
Tweet unigrams	896,397
Tweet bigrams	4,136,572
Tweet unigrams + bigrams	5,032,969

The top 10 most informative features for friends, description unigrams, and Tweet bigrams can be seen below in Tables 8.3, 8.4, and 8.5, respectively. Since each friend feature is simply the Twitter ID of the account being followed, presenting such information is not very insightful; instead, the account's screen name and full name that has been provided on Twitter is presented.

Table 8.3 – Top 10 most informative friend features

Under 30	Over 30
@pewdiepie (Felix Kjellberg)	@listia (Listia)
@justinbieber (Justin Bieber)	@cnnbrk (CNN Breaking News)
@ddlovato (Demi Lovato)	@BBCBreaking (BBC Breaking News)
@UberFacts (UberFacts)	@ConanOBrien (Conan O'Brien)
@ArianaGrande (Ariana Grande)	@Oprah (Oprah Winfrey)
@vine (Vine)	@Pink (P!nk)
@MileyCyrus (Miley Ray Cyrus)	@ShareThis (ShareThis)
@KSIOlajidebt (KSI)	@jimmyfallon (Jimmy Fallon)
@selenagomez (Selena Gomez)	@GeorgeTakei (George Takei)
@deadmau5 (deadmau5)	@stephenfry (Stephen Fry)

It is apparent that nearly all of the indicative friends are 'verified' Twitter accounts. Twitter verifies the authenticity of accounts that are "highly sought users in music, acting, fashion, government, politics, religion,

journalism, media, sports, business and other key interest areas." [26], and it is therefore not surprising that these accounts are found to be indicative of a user's age.

Table 8.4 – Top 10 most informative unigram description features

Under 30	Over 30
student	married
girl	male
taken	father
youtuber	kids
instagram	wife
❤	retired
directioner	work
college	man
:d	husband
youtube	children

Table 8.5 – Top 10 most informative bigram Tweet features

Under 30	Over 30
when you	my daughter
about to	ha ha
me on	what is
me :	a great
when your	i thought
follow me	you are
rt for	my kids
come to	.. by
, you're	my son
right now	my wife

8.3 Combined Feature Sets

The next set of experiments combines the feature sets that were previously trialled independently. To avoid a combitorial explosion of comparisons, and for the performance reasons stated previously, only combinations of friends, description unigrams, and Tweet bigrams are explored as feature sets. The resulting performance of the combined feature sets can be seen below in

Table 8.6, with the dimensionalities of respective feature sets shown below in Table 8.7.

Table 8.6 – Performance (± standard deviation) obtained from combined feature sets

Feature Set	F_1 Score (3 s.f.)
Friends + Description Unigrams	0.863 (± 0.00729)
Friends + Tweet Bigrams	0.855 (± 0.00659)
Description Unigrams + Tweet Bigrams	0.844 (± 0.00391)
Friends + Description Unigrams + Tweet Bigrams	0.864 (± 0.00803)

Supplementing the Tweet bigrams with description unigrams boosts performance by 1.1%, resulting in a reduction in error of 6.57%. Although bigram features from Tweets were found to be better than unigram features from descriptions, when adding friends into the feature set description unigrams resulted in better performance. Ultimately, the use of all three feature set combinations results in the best performance of 0.864; this amounts to a 19.0% reduction in error over using friends alone.

Table 8.7 – Average training dimensionalities of combined feature sets

Feature Set	Number of Features
Friends + Description Unigrams	2,087,281
Friends + Tweet Bigrams	6,207,761
Description Unigrams + Tweet Bigrams	4,152,664
Friends + Description Unigrams + Tweet Bigrams	6,223,853

8.4 Impact of Dataset Size

The following set of experiments assess how changing the dataset size effects classification performance. To achieve this, the number of users in each class is restricted, ranging from dataset sizes of 200 users (100 in each class), to 6,400 users (3,200 in each class). The method of model evaluation remains the same, with 70% of the reduced dataset being used for training and the remaining 30% being used for testing.

The resulting performances for feature sets using friends, description unigrams, and Tweet bigrams are shown in Figures 8.1, 8.2, and 8.3, respectively. Varying the dataset size using the combination of all feature sets is shown in figure 8.4. Shaded areas of the graphs represent the standard deviation, with a smaller shaded area around the point representing greater result stability. For comparison, figure 8.5 shows all four dataset size experiments on the same graph, but without standard deviation. The precise statistics used to generate the graphs are provided in appendix C.

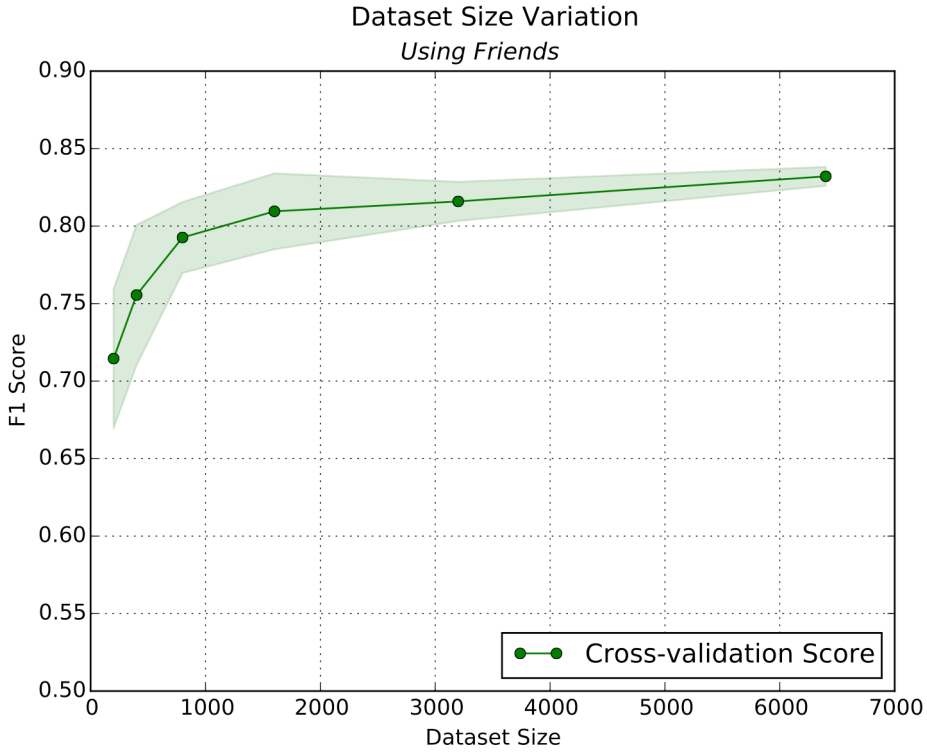

Figure 8.1: Varying the dataset size using friend features

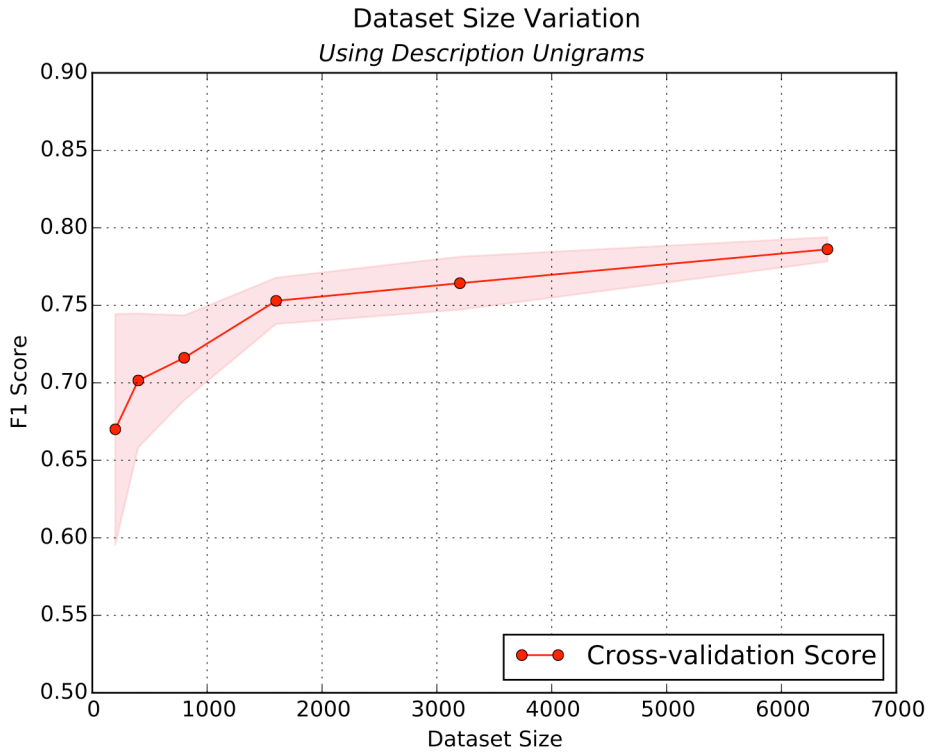

Figure 8.2 – Varying the dataset size using description unigram features

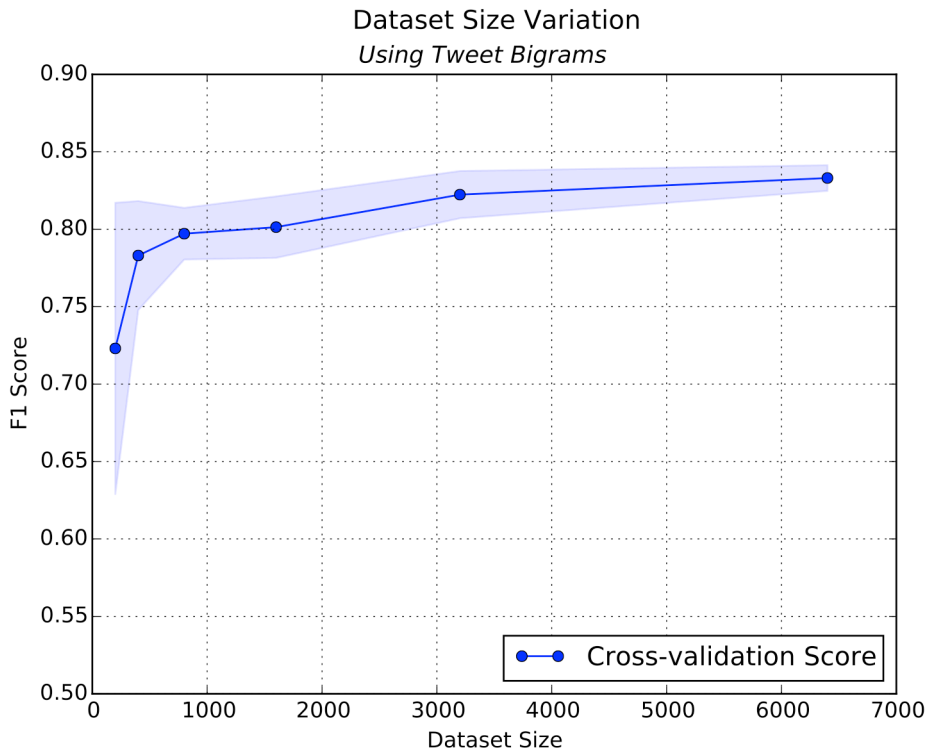

Figure 8.3 – Varying the dataset size using Tweet bigram features

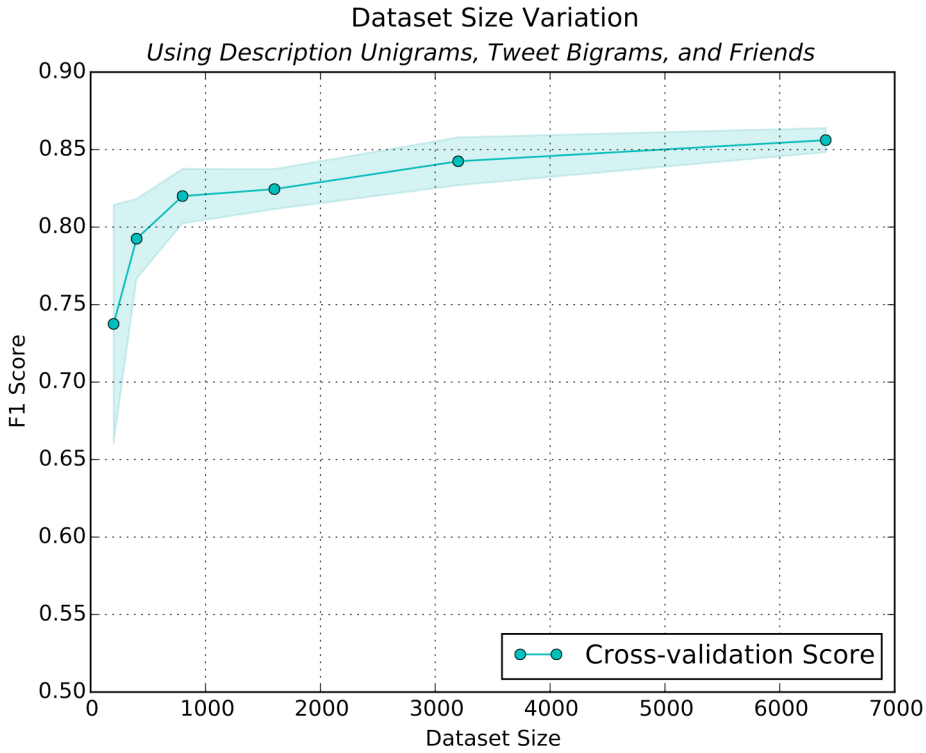

Figure 8.4 – Varying the dataset size using the combination of friend, description unigram, and Tweet bigram features

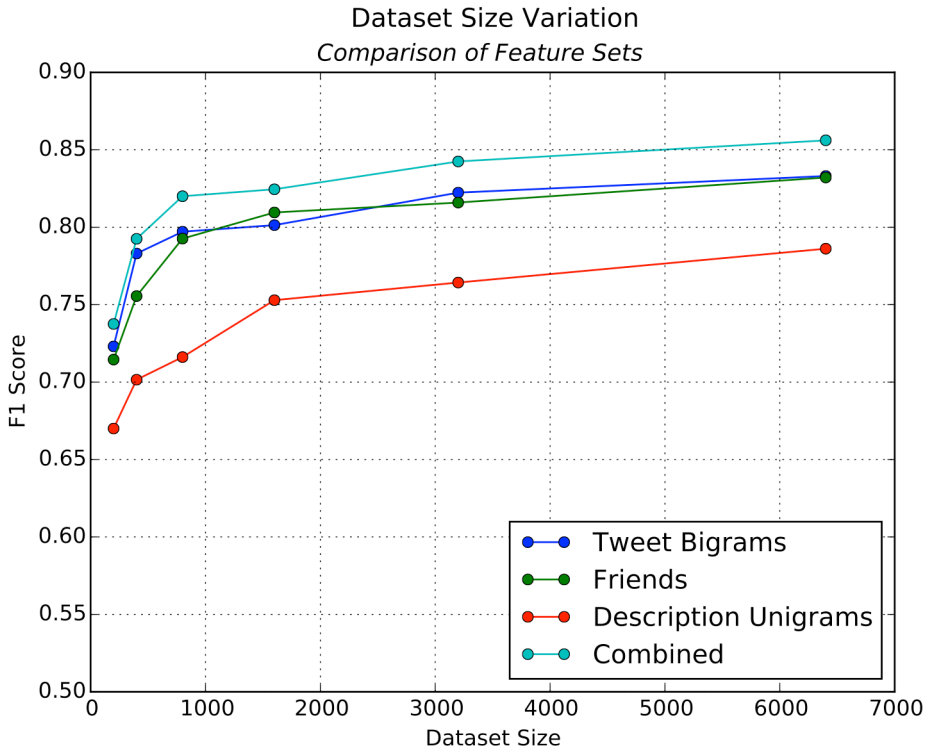

Figure 8.5 – Comparison of feature sets when the dataset size is varied

9 Discussion

RQ1. How viable is dataset creation based on automatically annotating users using their description field?

The analysis performed in section 6 provides some confidence in the annotation approach, with the pattern matching technique creating a dataset that contains around 2% overall noise. The annotation method was shown to produce a strong skew towards the younger age groups, and comparing with research performed by PEW suggests that the skew is not representative of the true population of Twitter. However, it is not fair to make a direct comparison, since PEW's research is based solely on US citizens, and their method of acquiring data is not based on the activity of such accounts, just whether or not a user has signed up to Twitter.

Regardless of PEW's findings, the skew does appear unrepresentative, and there are many possible reasons for this. The most obvious reason is that as users get older, they may be less inclined to present their age. Older age groups may find that their age is an irrelevant piece of information to include, whereas younger age groups may see their age as an attribute that helps define them. Unfortunately, there is little that can be done to adapt the pattern matching technique to accommodate this issue.

Another reason for the skew is that the users were sampled over a period of one week. This sampling period leads to two major biases. Firstly, the collection method favours users that are more active on Twitter, with users that have an average Tweeting frequency of over one week likely to be ignored. Secondly, current events may have encouraged a particular group of users to be more active during the collection period. Even simply wishing a celebrity a Happy Birthday, or reacting to sports results could have caused a surge of certain users to be highly active during the collection period. In hindsight this is fairly obvious, and it is clear that sampling over a longer period of time, or aggregating data from multiple sampling periods would have been a more suitable approach.

The annotation process also results in all users having informative description fields, which is clearly unrepresentative as 17% of users in the collected data had blank descriptions. Therefore, training a model from features based on a user's description field may result in poor generalisation.

Nevertheless, the value of automated annotation must not be dismissed, with a large dataset of 62,450 users being created from only one week's

worth of data, and requiring no human intervention. Even if this dataset is deemed to have bias due to the short collection period, a new dataset collected over a longer period of time can be quickly annotated using the same set of patterns. Additionally, since the annotation approach provides such large numbers of annotated users, the distribution could be resampled to match the true age distribution of Twitter.

RQ2. Are the accounts that a user follows valuable for age inference?

As the results in section 8.2 show, friend features are found to be highly indicative of age, resulting a classification performance of 0.83. The performance obtained from using friend features surpasses that obtained from description features (0.79), and rivals the performance obtained from Tweets features (0.83), which in current research is generally seen to provide the best performance.

Although for many applications it is a Tweet's author that is of interest, some applications are interested in users regardless of whether or not they Tweet. Highly desirable tasks, such as determining a brand's follower demographic, are likely to require the classification of *passive* accounts, i.e. users that mainly use Twitter to keep informed about their interests. Tweets have shown to be effective for age inference, but clearly, inference can only be performed if a user has written them. Since friend based features are prominent amongst both active and passive users, they can be considered a highly valuable attribute.

However, although friend based features are highly desirable, obtaining a user's friends is subject to much higher API rate restrictions than Tweets, making them less feasible for use in large scale classification. For example, if a brand has a modest following of 100,000 users, obtaining the 200 most recent Tweets for each user requires six days, whereas obtaining their friends requires over two months. To increase user classification throughput for applications where passive users are still of interest, a collection back-off approach could be applied, only collecting friends when the number of Tweets is not adequate.

RQ3. What effect does combining friend based features with typical linguistic features have on classification performance?

Using friends alone results in a respectable performance of 0.83. However, combining friend features with both description based features and Tweet based features results in performance of 0.86 an increase in performance of 3%; an overall error reduction of 19%. Supplementing description based

features with friend features results in a performance of over 0.86, and supplementing Tweet based features resulted in a performance of over 0.84. Surprisingly, although description based features performed worse than Tweet features when used independently, when friend features are included into the feature sets, the reverse is found. However, it is unclear why this is the case. One possible reason is than description based features are more indicative than Tweet based features, but due to descriptions providing much less text than Tweets, some users may not provide enough features for an accurate prediction. Therefore, when using both descriptions and friends, there is more chance for a user to provide indicative features.

Additionally including Tweets in the friend and description based feature set only increased performance by 0.1%, a very small gain for the increased model complexity. However, it is important to note that the description may provide overly optimistic performance due to dataset bias, and therefore combining all feature sets may be necessary for the model to provide both optimal performance and generalisability.

The informative description features shown in Table 8.4 are fairly intuitive, however the top features for the over 30s seem more robust to sociocultural evolution, with a much smaller number of features based on current bands, websites, and technology. Unlike the description features, the most informative Tweet based bigrams in Table 8.5 are not so intuitive, perhaps further backing up the theory about description features being more indicative. It is not clear why features for the under 30s such as "about to" or "when you" are included, one suggestion is that since the Tweets also includes Retweets, a popular Tweet containing these bigrams may have been Retweeted by a number of users under 30.

For Tweet based features, bigrams surpassed unigrams, however, for description based features the opposite was found. Suggestions were made in literature that bigrams perform better than unigrams as they capture more semantic information [12]. However, since bigrams introduce greater feature sparseness, and description fields are much shorter than the collection of Tweets, bigrams from the description may only be appropriate with more training data.

RQ4. What effect does altering the dataset size have on classification performance?

As is common within machine learning, increasing the size of the dataset size shows a performance increase, with a typical learning curve that starts to converge. However, what is interesting with this this domain is that

Twitter's API rate restrictions greatly effect how fast, and ultimately, how much data can be collected. Since collecting friends is a factor of 12 slower than collecting Tweets, in real world applications development speed may be favoured over overall model accuracy, and is therefore hard to compare. All attributes however, exhibit a rapid decrease in performance gain as the dataset grows. With the current rate limits in place, Figures 8.1 through 8.5 suggest that spending extra weeks or months worth of data collection may only provide minimal performance gain after a dataset contains a just few thousand users.

Nevertheless, Figure 8.5 shows that with a dataset containing only 1,600 users (800 per class) but using all of three feature sets, provides almost identical performance to a dataset containing 6,400 users (3,200 per class) that only uses Tweets. However, collecting Tweets for 6,400 users is still over three times fasters than collecting all attributes for the 1,600 users that are required to obtain comparable performance. Comparisons in this way are difficult, especially due to the dataset being unrepresentative and favouring users that Tweet frequently. As previously mentioned, users that have very few Tweets, or even no Tweets at all, are still of interest for some applications. In these situations, friends provide a crucial attribute and should therefore not be ignored. Instead, being tactical about which attributes to collect for certain situations is definitely necessary. Relating back to the findings, if the application is known to deal with infrequent Tweeters, then Figure 8.5 suggests that it may be more appropriate to collect a smaller dataset that covers more user attributes.

9.1 Future Work

Due to both the projects time constraints, and the task's non-triviality, there many areas for future work. Firstly, to help address the dataset bias, data could be collected over a longer period of time, work into reshaping the dataset to better fit the *true* age distribution, and additional measures such as removing a random sample of description fields could be performed.

The classes "under 30" and "over 30" may not be considered the most useful in practice. Future work could include attempting this task with a more useful age bucketing strategy, or perhaps more appropriately, treating it as a regression task.

For this project, the dataset was not focused towards a specific geographic region. For many applications, it may be more useful to build datasets

targeted towards a specific country. Future work could include further filtering the dataset by user provided location or time-zone.

This project managed to report on three of the main user attributes available, exploring the much neglected possibility of using who a user follows. However, due to time constraints, a vast number of attributes were not explored, and therefore the exploration of such features is left for future work.

Dataset bias causes the generalisability of the model to be questioned. Future work could include assessing the performance of the model on a smaller, manually annotated dataset.

Although a comparison of machine learning models was not within the scope of this project, other models should not be neglected. Support Vector Machines have many nice properties, but since simplicity, interpretability, and speed of model training may be major factors in real world applications, future work should also explore the effectiveness of models such as Naive Bayes and Logistic Regression.

10 Conclusion

This project deployed a technique of automatically creating an age annotated dataset, and explored three main user attributes for performing age inference.

The automated annotation process shows promise in accuracy, with the approach only annotating around 2% of users in the dataset incorrectly. Additionally, the approach shows success in providing a large dataset to work from, with only a single week's worth of data resulting in a dataset containing 62,450 users. However, it was also found that the annotation approach resulted in many biases, most notably, a very large skew towards the younger age groups, which was mainly put down to the idea that older users are less inclined to explicitly provide their age.

Feature sets based on friends, Tweets, and the user's description field were all shown to be encouraging indicators of age. Simply having the knowledge of who a user follows showed performance competitive with Tweet based features. Description based features were also found to be of use, and when using all three feature sets in conjunction, a top performance of over 86% could be achieved. When used independently, Tweet based features slightly outperformed friend based features. However, it was argued that friend based features are of more use, since they additionally allow inference to be performed on users that have little or no Tweets. Additionally, pre-processing steps to remove all text used in the annotation process were found to be paramount; without doing so showed clear signs of the model exhibiting poor generalisation.

The impact that the dataset size has on performance was also explored. The exploration revealed that as the size of the dataset increases, performance gain rapidly slows, suggesting that data collection efforts past a few thousand users may be unnecessary. Additionally, findings suggest that although collecting friends incurs much higher rate restrictions than Tweets, building a smaller dataset that also incorporates a user's friends may be more effective than a larger dataset than ignores them; especially in applications that require the classification of infrequent Tweeters.

Overall, age inference was found to be a non-trivial task, with creating a representative dataset a major challenge in itself. Nevertheless, the project managed to successfully explore three main user attributes, including the user's friends, which has been widely neglected in current research. The findings provide additional confidence that a user's publicly provided data can be very indicative of their age, and show that even relatively small-scale datasets can provide satisfactory results.

11 References

[1] Mislove, A., Lehmann, S., Ahn, Y. Y., Onnela, J. P., & Rosenquist, J. N. Understanding the Demographics of Twitter Users. *ICWSM*. **11**. 2011.

[2] Twitter. *Twitter API Documentation*. Accessed: 06/05/15. Available from: https://dev.twitter.com/overview/documentation

[3] Nguyen, D., Gravel, R., Trieschnigg, D., & Meder, T. "How Old Do You Think I Am?": A Study of Language and Age in Twitter. *Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media.* 2013. pp. 439-448.

[4] Rao, D., Yarowsky, D., Shreevats, A., & Gupta, M. Classifying Latent User Attributes in Twitter. *Proceedings of the 2nd international workshop on Search and mining user-generated contents*. ACM. 2010. pp. 37-44.

[5] Ito, J., Hoshide, T., Toda, H., & Uchiyama, T. What is he/she like?: Estimating Twitter User Attributes from Contents and Social Neighbors. *In Advances in Social Networks Analysis and Mining (ASONAM) IEEE/ACM International Conference on.* IEEE. 2013. pp. 1448-1450.

[6] Pennacchiotti, M. & Popescu, A. A Machine Learning Approach to Twitter User Classification. *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media.* 2011. pp. 281-288.

[7] Zamal, A. F., Liu, W., & Ruths, D. Homophily and Latent Attribute Inference: Inferring Latent Attributes of Twitter Users from Neighbors. *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media.* 2012. pp. 387-390.

[8] Nguyen, D., Smith, A. N., & Rosé, P. C. Author Age Prediction from Text using Linear Regression. *In Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities.* Association for Computational Linguistics. 2011. pp. 115-123.

[9] Twitter. *Twitter's API Rate Limits*. Accessed: 06/05/15. Available from: https://dev.twitter.com/rest/public/rate-limits

[10] Twitter. *Twitter User Meta-Data*. Accessed: 06/05/15. Available from: https://dev.twitter.com/overview/api/users

[11] Alowibdi, S. Jalal., Buy, A. Ugo., & Yu, Philip. Language Independent Gender Classification on Twitter. *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining.* 2013. pp. 739-743.

[12] Zheng, L., Yang, K., Yu, Y., & Jin, P. Predicting Age Range of users over Microblog Dataset. *International Journal of Database Theory and Application.* **6** (6). 2013. pp. 85-94.

[13] Jones, R., Kumar, R., Pang, B., & Tomkins, A. "I Know What You Did Last Summer" — Query Logs and User Privacy. *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. ACM. 2007. pp. 909-914.

[14] Schler, J., Koppel, M., Argamon, S., & Pennebaker, J. Effects of Age and Gender on Blogging. *In AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs.* 2006. **6**. pp. 199-205.

[15] Peersman, C., Daelemans, W., & Vaerenbergh, V. L. Predicting Age and Gender in Online Social Networks. *In Proceedings of the 3rd international workshop on Search and mining user-generated contents.* ACM. 2011 pp. 37-44.

[16] Santosh, K., Bansal, R., Shekhar, M., & Varma, V. Author Profiling: Predicting Age and Gender from Blogs. *Notebook for PAN at CLEF 2013.*

[17] Macaulay, R. K. S. *Talk that counts: Age, Gender, and Social Class Differences in Discourse.* New York: Oxford University Press. 2005.

[18] British Computer Society. *Code of Conduct*. 2011. Accessed: 06/05/15. Available from: http://www.bcs.org/upload/pdf/conduct.pdf

[19] British Computer Society. *Code of Good Practice*. Accessed: 06/05/15. Available from: http://www.bcs.org/upload/pdf/cop.pdf

[20] Python Software Foundation. *PEP 8 Code Guidelines*. Accessed: 06/05/15. Available from: http://legacy.python.org/dev/peps/pep-0008/

[21] Twitter. *Twitter's Streaming API*. Accessed: 06/05/15. Available From: https://dev.twitter.com/streaming/overview

[22] Twitter. *Twitter's REST API*. Accessed: 06/05/15. Available from: https://dev.twitter.com/rest/public

[23] Wibberley, S., Reffin, J., & Weir, D. Method51 for Mining Insight from Social Media Datasets. *Proceedings of COLING 2014, the 25th International*

Conference on Computational Linguistics: System Demonstrations. 2014. pp. 115-119.

[24] Burges, C. J. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*. 1998. pp. 121-167.

[25] O'Connor, B., Krieger, M., & Ahn, D. TweetMotif: Exploratory Search and Topic Summarization for Twitter. *In ICWSM*. 2010.

[26] Twitter. *FAQs About Twitter's Verified Accounts*. Accessed: 06/05/15. Available from: https://support.twitter.com/articles/119135-faqs-about-verified-accounts

[27] Pedregosa, F., Varoquaux, G., Gramfort, A., et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research.* **12**. 2011. pp. 2825-2830.

[28] Twitter. *Twitter's Developer Agreement & Policy*. Accessed: 06/05/15. Available from: https://dev.twitter.com/overview/terms/agreement-and-policy

[29] Duggan, M., Ellison, N.B., Lampe, C., Lenhart, A,. & Madden, M. *Social Media Update 2014*. Pew Research Center. 2015. Accessed: 06/05/15. Available from: http://pewinternet.org/2015/01/09/social-media-update-2014

12 Appendices

Appendix A: Pattern Matching Configurations

The regular expressions have been split across multiple lines for formatting purposes. In practice, each pattern is written on one line and contains no whitespace.

Development Iteration 1

```
AGE
((([^\w]i)|(^i))((\'?m)|(\sam)))?
(?P<AGE>([^\w][1-9][0-9])|^([1-9][0-9]))\s?
(((yr|year)s?\s?((old)|(of\sage)|(young)))|(y\/o)|(y\.o))

AGE
(([^\w]i)|(^i))((\'?m)|(\sam))\s
(?P<AGE>[1-9][0-9]?)($|\s|[^\w^%^\'](?$|\s))

DOB
(([^\w]born)|(^born))\s([io]n\s)?
(?P<DOB>(19[2-9][0-9])|(\'[0-9][0-9])|(200[0-4]))

AGE
(([^\w]aged?)|(^(aged?)))\s?:?\s(?P<AGE>[1-9][0-9])
```

Development Iteration 2

```
AGE
(in\sa)(?P<AGE>([^\w][1-9][0-9])|^([1-9][0-9]))\s?
(((yr|year)s?\s?((old)|(of\sage)|(young)))|(y\/o)|(y\.o))

DO_NOT_INCLUDE
(when|until|was|with\sa)((([^\w]i)|(^i))((\'?m)|(\sam)))?
(([^\w][1-9][0-9])|^([1-9][0-9]))\s?
(((yr|year)s?\s?((old)|(of\sage)|(young)))|(y\/o)|(y\.o))

DO_NOT_INCLUDE
(my)(([^\w][1-9][0-9])|^([1-9][0-9]))\s?
(((yr|year)s?\s?((old)|(of\sage)|(young)))|(y\/o)|(y\.o))\
s(son|daughter|baby|grandson|granddaughter|grandchild|moth
er|father|grandmother|grandfather|mom|mum|dad)

DO_NOT_INCLUDE
(when|until)(([^\w]i)|(^i))((\'?m)|(\sam))\s
([1-9][0-9]?)($|\s|[^\w^%^\'](?$|\s))

DO_NOT_INCLUDE
(([^\w]i)|(^i))((\'?m)|(\sam))\s([1-9][0-9]?)
($|\s|[^\w^%^\'](?$|\s))(seconds|shades|percent|feet|foot|i
nches|pounds|stone)
```

```
AGE
((([^\w]i)|(^i))((\'?m)|(\sam)))?
(?P<AGE>([^\w][1-9][0-9])|^([1-9][0-9]))\s?
(((yr|year)s?\s?((old)|(of\sage)|(young)))|(y\/o)|(y\.o))

AGE
(([^\w]i)|(^i))((\'?m)|(\sam))\s
(?P<AGE>[1-9][0-9]?)($|\s|[^\w^%^\'](?$|\s))

DOB
(([^\w]born)|(^born))\s([io]n\s)?
(?P<DOB>(19[2-9][0-9])|(\'[0-9][0-9])|(200[0-4]))

AGE
(([^\w]aged?)|(^(aged?)))\s?:?\s(?P<AGE>[1-9][0-9])
```

Development Iteration 3

```
AGE
(in\sa)(?P<age>([^\w][1-9][0-9])|^([1-9][0-9]))\s?
(((yr|year)s?\s?((old)|(of\sage)|(young)))|(y\/o)|(y\.o))

DO_NOT_INCLUDE
(to|after|with|of\san?|in\san?|my|before|it\sis|those\sund
er|by|you.?re?|like|sometimes|when|till?|until|was|with\sa
|of\s(an|a)(\s\w*)?)(\s(a|an|am))?((([^\w]i)|(^i))((\'?m)|
(\sam)))?(([^\w][1-9][0-9])|^([1-9][0-9]))\s?
(((yr|year)s?\s?((old)|(of\sage)|(young)))|(y\/o)|(y\.o))

DO_NOT_INCLUDE
(of\san?|in\san?|to|after|my|before|by|than|care\sof|carer
\s(to|for)|caring\sfor)(\s?(my|our|a))?
(([^\w][1-9][0-9])|^([1-9][0-9]))\s?
(((yr|year)s?\s?((old)|(of\sage)|(young)))|(y\/o)|(y\.o))\
s(son|daughter|baby|grandson|granddaughter|grandchild|moth
er|father|grandmother|grandfather|mom|mum|dad)

DO_NOT_INCLUDE
(to|after|before|by|like|sometimes|when|till?|until)(([^\w
]i)|(^i))((\'?m)|(\sam))\s
([1-9][0-9]?)($|\s|[^\w^%^\'](?$|\s))

DO_NOT_INCLUDE
(to|after|when|by|before|till?|until)
(([^\w]aged?)|(^(aged?)))\s?:?\s([1-9][0-9])

DO_NOT_INCLUDE
(([^\w]aged?)|(^(aged?)))\s?:?\s
([1-9][0-9])('?s|\sin\s(dog|cat)\syears)
```

```
DO_NOT_INCLUDE
(([^\w]i)|(^i))((\'?m)|(\sam))\s([1-9][0-9]?)
($|\s|[^\w^%^\'](?$|\s))(in\s(dog|cat)\syears|years\s(older
|younger)|seconds|flavors|shades|percent|feet|foot|ft|inch
es|pounds|kg|stone|away|much|cool|times|of|decades)

AGE
((([^\w]i)|(^i))((\'?m)|(\sam)))?
(?P<AGE>([^\w][1-9][0-9])|^([1-9][0-9]))\s?
(((yr|year)s?\s?((old)|(of\sage)|(young)))|(y\/o)|(y\.o))

AGE
(([^\w]i)|(^i))((\'?m)|(\sam))\s
(?P<AGE>[1-9][0-9]?)($|\s|[^\w^%^\'](?$|\s))

DOB
(([^\w]born)|(^born))\s([io]n\s)?
(?P<DOB>(19[2-9][0-9])|(\'[0-9][0-9])|(200[0-4]))

AGE
(([^\w]aged?)|(^(aged?)))\s?:?\s(?P<AGE>[1-9][0-9])
```

Appendix B: Dataset Age Frequencies

Year of Birth	Frequency
1930	2
1931	6
1932	8
1933	4
1934	21
1935	3
1936	4
1937	7
1938	6
1939	14
1940	7
1941	9
1942	8
1943	8
1944	21
1945	11
1946	11
1947	12
1948	20
1949	26
1950	22
1951	15
1952	19
1953	18
1954	49
1955	25
1956	31
1957	28
1958	33
1959	43
1960	39
1961	46
1962	46
1963	33
1964	76
1965	56
1966	73
1967	49
1968	64
1969	82
1970	92

Year of Birth	Frequency
1971	92
1972	85
1973	75
1974	170
1975	95
1976	97
1977	127
1978	133
1979	167
1980	137
1981	168
1982	240
1983	229
1984	369
1985	343
1986	449
1987	523
1988	712
1989	1048
1990	1406
1991	2029
1992	2838
1993	4139
1994	5322
1995	5396
1996	6958
1997	6835
1998	6829
1999	5582
2000	4282
2001	2333
2002	1031
2003	429
2004	257
2005	23
2006	21
2007	27
2008	29
2009	38
2010	27

Appendix C: Additional Experiment Data

Table C.1 – Top 20 most informative unigram description features without the removal of text used for annotation

Under 30	Over 30
17	30
18	32
19	35
20	40
21	33
16	36
22	38
24	37
23	34
15	44
14	43
26	52
13	42
25	39
27	45
28	50
12	48
11	47
21yrs	39
17years	41

Table C.2 – Results from varying the dataset size using description unigrams features, where the dataset contains an equal number of under 30s and over 30s

Dataset Size	F_1 Score (3 s.f.)
200	0.677 ($\pm$ 0.0704)
400	0.703 ($\pm$ 0.0424)
800	0.717 ($\pm$ 0.0704)
1600	0.734 ($\pm$ 0.0150)
3200	0.765 ($\pm$ 0.0168)
6400	0.786 ($\pm$ 0.00778)

Table C.3 – Results from varying the dataset size using Tweet bigram features, where the dataset contains an equal number of under 30s and over 30s

Dataset Size	F_1 Score (3 s.f.)
200	0.727 (± 0.0907)
400	0.784 (± 0.0342)
800	0.797 (± 0.0167)
1600	0.801 (± 0.0198)
3200	0.822 (± 0.0152)
6400	0.833 (± 0.00829)

Table C.4 – Results from varying the dataset size using friend features, where the dataset contains an equal number of under 30s and over 30s

Dataset Size	F_1 Score (3 s.f.)
200	0.725 (± 0.0382)
400	0.760 (± 0.0425)
800	0.794 (± 0.0220)
1600	0.811 (± 0.0235)
3200	0.817 (± 0.0122)
6400	0.833 (± 0.00602)

Table C.5 – Results from varying the dataset size using friend, description unigram, and Tweet bigram features, where the dataset contains an equal number of under 30s and over 30s

Dataset Size	F_1 Score (3 s.f.)
200	0.740 (± 0.0757)
400	0.793 (± 0.0252)
800	0.820 (± 0.0174)
1600	0.823 (± 0.0129)
3200	0.843 (± 0.0154)
6400	0.856 (± 0.00782)

Appendix D: Work Log

Summer 2014

Literature research and started work on the machine-learning framework to aid Twitter experiments.

30/10/14

First official supervisor meeting, we shared some ideas with David Spence, a PhD student who is starting similar research to my project. I've sent over some relevant literature.

01/10/14

The framework now supports feature extraction from the description field. Working on extending feature extraction from all user meta-data available (name, statuses count, friends count, etc.). This will eventually also need to work on the user's Tweets, followers, and friends.

03/10/14

Datasets can now be built to contain any user meta-data that is stored in the database. As the types of meta-data has changed from being solely text based to containing numerical data such as the number of followers, different ways of feature extraction needs to be developed.

06/10/14

Feature extraction can now be performed on any of the user meta-data. Simple feature extractors can be written as Python functions and then specified in a config file.

07/10/14

Core framework functionality has been implemented, the next stage is to develop the system to include the extended meta-data (Tweets, friends, and followers). Started work on the project proposal.

10/10/14

Continuing work on the project proposal.

12/10/14

Finished project proposal.

14/10/14

Handed in the project proposal.

26/10/14

Working on the interim report outline. Started Tweet collection for users in the dataset using Method51.

28/10/14
Handed interim report outline to supervisor. Tweet collection had a few bugs, so Method51 has been updated an I have been set-up on a more stable server. The Tweets database will contain approximately 14,000,000 Tweets when collection is finished.

02/11/14
Finished draft interim report.

04/11/14
Finished final interim report and submitted.

11/11/14
The Tweets database has been fully obtained, however, 4,000 users have changed their accounts to private between the time of user collection, and the time of tweet collection. The dataset now contains, around 12 million Tweets and 66,000 users.

18/11/14
The datasets require 20GB+ of RAM to use stored in memory, either the dataset size needs to be reduced, or a database needs to be used. The MySQL database they are already in can be used, however, it is stored remotely so query speed is a concern, and network reliability during a experiment may cause negative side effects. Storing the MySQL database locally is one option, however, the schema of the database is not ideal for this machine learning task. MongoDB seems like the best option.

26/11/14
The dataset has been migrated to MongoDB and the framework has been adapted to accommodate. Iterating over the 66,000 users and obtaining around 200 of their Tweets takes approximately two minutes. This is a lot slower than memory, but is still a very short period of time and will not be the bottle neck of the experiments.

03/12/14
Performing experiments using a split at age 30 and undersampling the majority class. However, the experimental set up had a flaw and when the experiment was run multiple times slightly different results were being displayed (even with 10 fold CV). This is due to the majority class being randomly undersampled. To provide reproducibility, I've implemented the ability to use seed states.

11/12/14
Although implementing a seed for the random sampling allows experiments to be reproduced, running the same experiment with a different initial seed could potentially result in dramatically different results. I've implemented a method called Repeated Random Sub-Sampling Validation instead, which draws a new random sample for each iteration.

18/12/15
Started assessing regular expression noise, refining the regexes would benefit from a more sophisticated filtering systems, where users can be easily excluded if a specific pattern is matched.

02/01/15
Added the functionality to filter users by more sophisticated pattern matching and filtering, and finished all three iterations of pattern development and noise analysis.

06/01/15
The next stage is to collect friends, however as Method51 is not currently prepared to collect friends, I'm working on a Python script to do the job.

14/01/15
Written the friend collection script and started collecting. This will take a few weeks!

26/01/15
Looking into PEW's Twitter demographic research.

01/02/15
Unit testing the framework.

09/02/15
More unit testing, obtaining 100% test coverage for most modules.

18/02/15
Major refactoring of the whole system.

27/02/15
Writing additional unit and integration tests.

10/03/15
Added friends into the dataset. The dataset is now fully populated.

16/03/15
Started poster design.

20/03/15
Initial poster finished.

24/03/15
Finished final poster and submitted.

02/04/15
Experiments are pushing the boundaries of my system, for some experiments 16GB of RAM is just not enough. Looking into memory optimisations.

07/04/15
Python does not have block level scoping, performing manual variable dereferencing seemed to bring all experiments into my system's capabilities.

11/04/15
Report structuring.

18/04/15
Running set of independent feature set experiments, and analysing results. Full set of experiments is taking many days to run, even on a 4GHz 8-core machine! Grid-searching an SVM is not fun...

24/04/15
Running set of combined feature set experiments, and analysing results. Another few days of non-stop computation.

27/05/15
Report writing and generating tables and graphs for report.

29/04/15
Running set of dataset size variation experiments, and analysing results.

02/05/15
Near final draft.

04/05/15
Final draft.

06/05/15
Finalised report.