APPROVAL SHEET

Title of Thesis: PREDICTING LATENT DEMOGRAPHIC ATTRIBUTES OF TWITTER USERS

Name of Candidate: Georgiy Frolov
MS Computer Science 2016

Thesis and Abstract Approved : ______________________
Dr. Tim Oates
Professor
Department of Computer Science and
Electrical Engineering

Date Approved: ______________________

ABSTRACT

Title of dissertation: PREDICTING LATENT
DEMOGRAPHIC ATTRIBUTES
OF TWITTER USERS

Georgiy Frolov, MS Computer Science, 2016

Thesis directed by: Dr. Tim Oates, Professor
Department of Computer Science and
Electrical Engineering

Social media websites such as Twitter, Facebook, and LinkedIn aggregate large amounts of textual data. There is a wealth of user information that can be inferred from this, that is potentially useful in advertising, analytics, sentiment analysis, etc. It is estimated that over 60% of people in the US have a Twitter account, and a significant portion of US population is comprised of immigrants. As social media have become common place, people are willingly posting their personal information such as their name, age, location, alma mater, etc. This makes it possible to use text classification methods to accurately determine demographic profiles.

This thesis focuses on extracting latent demographic information from social media data. Previous works have attempted to determine user's race and ethnicity, while our work focuses on using posts on Twitter (tweets), to determine whether a user is an immigrant or a native US citizen. The method uses ethnic name distribution among immigrant and native populations to find and collect users in the United States, and their tweets across three race groups: Asian, Latino, and Caucasian/White. We use supervised machine learning approach to predict the immigration status of a user by examining the textual content of tweets, using Multinomial Naive Bayes, Support Vector Machines, Logistic Regression, k-Nearest Neighbors, and Decision Trees. We investigate methods for improving the performance of algorithms and determine how number of features affects the accuracy of the built models. Additionally we evaluate which features have more weight in classifying users, and attempt to discover latent topical patterns in the data corpus using Latent Dirichlet Allocation.

PREDICTING LATENT DEMOGRAPHIC ATTRIBUTES OF TWITTER USERS

by

Georgiy Frolov

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, Baltimore County in partial fulfillment
of the requirements for the degree of
Master of Science
2016

Dedication

This thesis is dedicated to

my mother, Anjelika Frolova, for her immeasurable love and support

&

my father, Aleksandr Frolov, for his care and dedication to our family.

Acknowledgments

I would like to acknowledge first and foremost my advisor, Dr. Tim Oates, who was very supportive and patient throughout the process for which I am very grateful. I would also like to thank my family and relatives, who continued to encourage and support my education. And last but not least I wanted to express my gratitude to management at my work, who have been very understanding and helpful in finishing my graduate degree.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

During the past decade, the use of social media has expanded and is no longer limited to simply exchanging messages between users. Users may share, view photos and videos, establish business/professional connections, publicize their creative ideas and thoughts, organize meetings with others based on common interests along with many other social networking services. With an ever increasing user base that spends approximately 2 hours a day on social media alone, it offers an opportunity for improving personalization and for advertisers to promote new products, offers and other types of ads [17]. The presence of users on social media has dramatically increased in the past decade, with over 50% increase in the United States alone as shown in Figure 1.1. A significant portion of the US population is comprised of immigrants, which accounts for approximately 80 million people(including their second generation children). Some social media platforms such as Facebook or LinkedIn provide an option for users to specify demographic details about themselves such as origin, gender and education, while other sites such as Twitter contain incomplete or often misleading biographical information about its users. To provide most relevant advertising and personalization it requires knowledge of user attributes. Recent work had shown that using text classification methods can accurately determine user's attributes such as age [15, 32], political orientation [27], and user's ethnicity

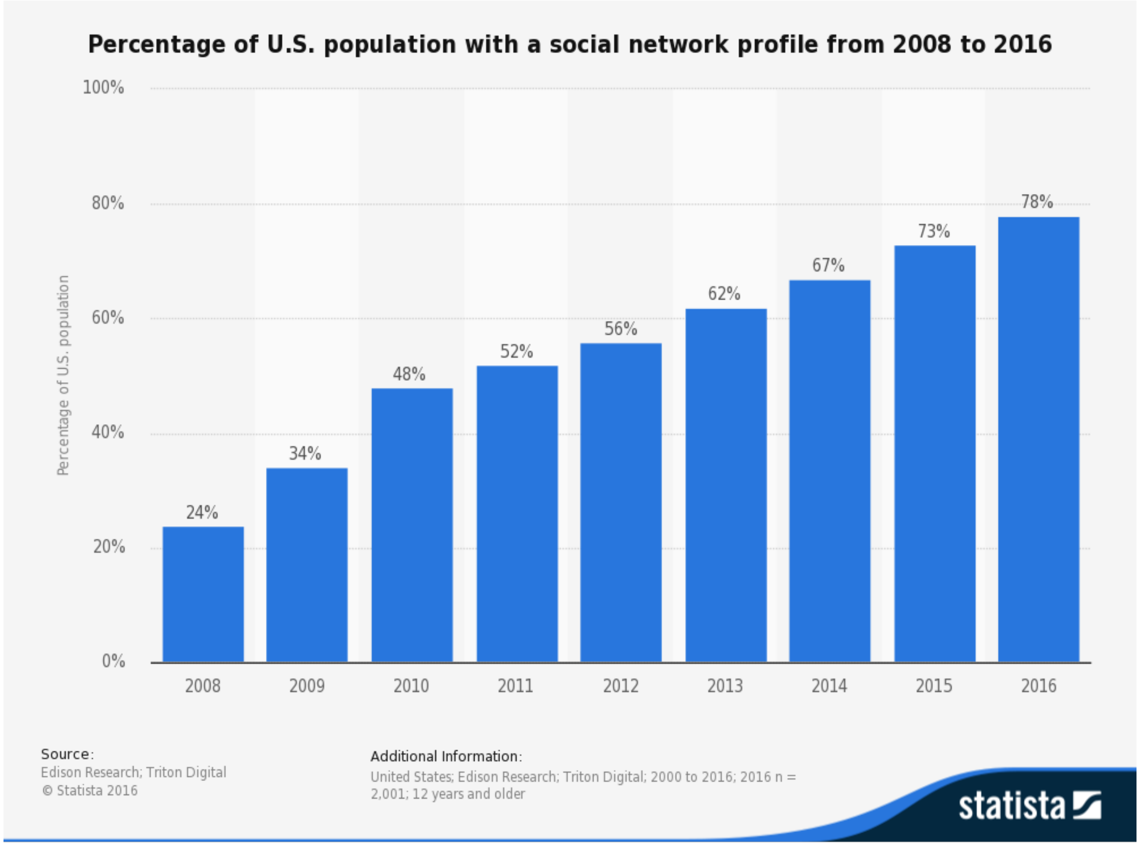


Figure 1.1: Percentage of US Population with a social network profile. Figure borrowed from Statista Inc. [39].

[7]. In this thesis we address the task of classifying tweets through application of supervised machine learning algorithms to predict the immigration status of a Twitter user.

1.1 Social Media, Twitter

The term social media includes any platform that allows users to communicate with one another, and share ideas. There are various types of social media websites: photo/video sharing websites such as Instagram and YouTube, interest based websites such Pinterest, work networking websites such as LinkedIn and Meetup, and social networking websites such as Facebook or Twitter just to name a few. With registration free and open for most social media, they are available to connect anyone with the rest of the world.

Twitter is a popular social networking service that enables its users to send short messages called ”tweets” either publically (in most cases) or as private messages. It is frequently used by people to share their thoughts, ideas, or for celebrities and/or politicians to attract new fans, post updates about their life and discuss ”trending topics”. With over 320 million monthly active users, it is in the top 10 of the most visited websites in the world according to Alexa rank [2].

Twitter is a unique social platform because unlike other social media sites such as Facebook and LinkedIn, which are mostly used for direct communication between users, Twitter also serves as micro blogging service, that provides a way to share user's everyday thoughts, opinions, fears and interests. Since the growth of popularity of Twitter, it is frequently used for data mining purposes both in real time and offline data analysis. Tweets that are coming in real time are used to detect if terrorist acts or natural disasters are occurring [12]. For example, tweets in Japan are closely monitored for information about earthquakes as people can post about them on Twitter way ahead of underground sensors [36]. Similarly by studying the posts of individual user, one may infer various attributes about a user by combining text contents with self reported data [31]. Some of the attributes are often not reported such as educational level, gender, or political orientation, however if they can be extracted, it would enable one to build a social profile with user's interests, ethnic origin, etc. which is very useful in marketing, advertising and even security.

1.2 Demographics of the US and Twitter

The use of social media has recently experienced a growth surge as the internet became a common commodity in practically every household in the US and the world. The number of social media users had increased by over a billion since the beginning of 2010 [37]. As the prices of mobile processors went down and the rise of smartphones starting with the iPhone ® in 2007, the number of social media users boomed because a stationary desktop was no longer required.

The number of Twitter users in the United States has grown substantially over the last few years. It now accounts for 21% of the entire Twitter population [41]. At the beginning of 2016, the population of the United States was estimated to be at over 323 million and growing [45]. The number of total registered users on Twitter (active and inactive) is estimated to be greater than 1 billion, thus approximately 60% of the United States population may be on Twitter [35].

A significant portion of the US population comprises of first and second generation Americans. There are over 42 million immigrants living in the United States, whether as naturalized citizens, permanent residents, refugees and asylees, international students and others; adding the US-born children of immigrants means that approximately 80 million people or one-quarter of the US population is either of the first or second generation [20]. Immigration remains a critical issue in the United

States because it has a significant impact on the economy, workforce, education and national security. Historically American nation was built on immigration and is a major source of population growth and cultural change.

However Twitter profiles do not provide sufficient information about their users to be able to identify demographic statistics such as immigration status. This presents an opportunity to use the linguistic content of user tweets and available profile information in order to extract latent attributes such as a user's origin. For the purposes of this study the term ”immigrant” (also known as foreign born) will be reserved for people who reside in the United States, with no US citizenship at birth. Thus this population includes naturalized citizens, refugees, legal aliens (work visas, or students), permanent residents (green cards), or illegal aliens. And the term ”native” (native born) will be reserved for US Born citizens (also known as natural born citizens). This includes people who were born in one of the 50 US states, or a US territory, or have at least one US citizen parent. A term user's origin is used throughout this research and it is intended to signify whether a user is a native US born citizen or an immigrant.

1.3 Objective of research

Signing up for twitter only requires an email or a phone number since the majority of active users on Twitter are using mobile devices [41]. There are additional fields that a user may specify such as location, description and name, although they are not required. The user profile on Twitter is sparse compared to the metadata available on Facebook or LinkedIn, which contain important features such as gender, age, education, favorite books or music and others. These and other features are very useful for customizing behavior of ads, search results and other personalization services.

The Twitter API provides access to largely unstructured data with a limited demographic profile of its users. This requires developing methods that would be able to use the available data to classify the origin of the user. We investigate previously unexplored latent demographic attribute of Twitter users, that focuses specifically on user's origin. We collect a dataset of diverse users and their tweets to study the relationships between the content of their tweets, and their immigration status. The details on data collection, pre-processing and manual annotation are described in Chapters 5 and 6. We treat the task of identifying a user's origin as a binary classification problem and apply supervised machine learning algorithms to build models that can determine whether a user is native born or an immigrant.

The objective of this research is to demonstrate that given a text document that contains user's tweets, supervised machine learning method can effectively predict the immigration status of a given user.

1.4 Thesis Contributions

The contributions of this thesis are as follows:

- We introduce a method for collecting the data by using ethnic related surnames that fit demographic groups of immigrant and native users.

- We present a novel dataset collected from unstructured Twitter data source that comprises of native US born and immigrant users that live in the US, which has not been gathered before.

- We demonstrate several methods that use linguistic content of tweets in order to extract latent demographic attribute, specifically user's immigration status, which has not been previously explored.

- We experimentally demonstrate our approach in classifying immigration status of Twitter users.

1.5 Practical significance of work

Currently, data mining plays an important role which is not only limited to search engines, but social media as well. By being able to accurately identify user's interests or origin, it would be possible to not only market "interesting" products, but also identify social trends, have a more complete picture of the population and even be helpful in national security, because people are often more observable on social media than in real life. The results of this research can be further extended for use in the following areas:

- Use in advertising : Targeted advertising requires knowledge of user's preferences, and by building an extended social profile these services would be able to present advertisements that would appeal to users and generate an increased "click through rate" (i.e. the number of users who open a presented page, email, or advertisement).

- Demographic profiling / census purposes : Potential underrepresentation of foreign-born immigrants due to a language barrier may contribute to inaccurate data on immigration statistics. One of the advantages of Twitter is that it supports over 30 languages that allows practically any person in the US to use it. Being able to extract a user's origin on Twitter would make it possible to identify under reported groups such as foreign-born immigrants.

- Use in marketing : Depending on a user's origin, service providers would be able to identify specific topics of interests of users such as products, brands, movies and others. The marketing efforts thus can be shifted according to the population demographics.

- Use in analytics : Expanding social profiles allows us to determine the demographic situation in the country with a greater accuracy. By classifying a user we may run additional analytical tools to gain insight into how user's usage and habits vary among different groups. This would be useful for policy makers to introduce new laws and policies that are more relevant and beneficial for the given community.

- Opinion mining : sentiment analysis assists companies and advertisers to get feedback from their customers about products/services, and adjust them accordingly to the needs of customers. It also helps social scientists understand how the general mind state differs between the communities. Sentiment analysis is of great importance for political analysts for predicting election results and can be used in combination with user classification to identify weak and strong points of candidates among social groups. Relevant to 2016 elections, immigration is one of the key topics discussed by the candidates and certain remarks by candidates may lead to strong disapproval from first and second generation of immigrants.

The remainder of this thesis is organized as follows. Chapter 2 covers research that has been previously performed in extracting latent attributes and the methods used. Chapter 3 outlines the methods that were used in this work. In Chapter 4 we describe the architecture of the system, the process of data collection and processing, as well as validation of data. Chapter 5 provides a detailed report on the collected datasets and associated statistics. Chapter 6 discusses the experiments conducted and methods that were used to build models. It provides and evaluates the results obtained from the models, followed by a conclusion and brief discussion about potential improvements and future work.

Chapter 2

Literature Survey

This chapter briefly summarizes previous work that has been completed in related areas. It provides an overview of methods that has been commonly used and how text classification was used to determine latent user attributes that are typically not provided or not explicitly available to access.

2.1 Related Works

Burger et al. [2011] explores identifying latent demographic features of online users, specifically gender, by treating it as a binary classification problem. They initially collected a massive dataset which consisted of approximately 18.5 million Twitter user profiles along with 213 million tweets. Similar to our research, their dataset did not contain demographic attributes which they were trying to classify, specifically gender and age, which required using external methods to label users' gender. Several experiments using Support Vector Machines, Naive Bayes and Balanced Winnow2 were run on the final dataset that contained 4.1 million tweets and over $180,000$ users achieving a maximum of 92.0% accuracy from the Balanced Winnow2 algorithm (using all fields). Further analysis of using different features has found that using only tweet text field provides an accuracy of 76%.

Rao et al. [2010] worked on investigating the performance of stacked-SVM-based classification algorithms over a rich set of novel features attempting to classify user attributes including gender, age, political orientation (Republican vs. Democrat) and regional origin (Southern or Northern India). Using Support Vector Machines they built several models that focused on specific sets of features, including social linguistic, and n–gram features.

Zamal et al. [2012] explored the idea that individuals with similar attributes tend to stick together in online social networks to detect gender, political orientation and age of a user. By adding neighborhood data from a user's friend list, they were able to boost the accuracy by 3% to 5%. Utilizing the idea of this study, we collected a list of followers for labeled users which were more likely to contain users from the same ethnic group and help increase the size of our training dataset.

Rao et al. [2011] presents minimally supervised hierarchical Bayesian model for detecting latent attributes of social media users, focusing specifically on gender and ethnicity classes in Nigeria. Using name sites, they compiled a name dictionary of names and their corresponding gender and ethnicity. Then, utilizing Facebook Graph API, wall posts were collected from a number of political figures and people that commented on their profile. They presented three different hierarchical Bayesian approaches and evaluated their performance using names and user posts, separately and in conjunction. In our efforts, we use US Census name data combined with tweet dataset to determine user's origin based on the user-generated textual content on Twitter. That is given a user's set of tweets we predict whether or not he or she is an immigrant or native US citizen.

Pennacchiotti and Popescu [2011] attempt to infer implicit user profile features such as political orientation and ethnicity on Twitter. The general model that they created considered the following :

- Account profile details such as name, description, location and others
- User's tweeting behavior such as average number of tweets sent or frequency of URLs posted
- Linguistic content of messages, that is words occurring in his or her posts
- Social network or "who you tweet" examined the people or "friends" that a user sends messages to or follows

Their results showed that while linguistic features may achieve high performance result, using a model with additional set of features boosts the accuracy of the model.

A study that heavily influenced the current research was completed by Chang et al. [2010] which utilizes census data to predict the ethnicity of users on Facebook using Bayesian probabilistic approach. Applying the U.S. Census Bureau's data on frequency of surnames distributed by ethnicity, they build a model using a combination of unsupervised and supervised (census name statistics) machine learning techniques. Chang and others also faced an issue where it is impossible to obtain ground truth with a dataset that does not contain a feature characteristic that we are interested in. To validate the proposed model, a dataset of approximately $77,000$ MySpace users was obtained which consisted of self reported name and ethnicity for each user. Several versions of the proposed model were tested against MySpace dataset by varying data used for training :

- Last name : use only last name
- First name : use only first name
- Census : simple census based model
- Internet : uses estimated ethnic breakdown of internet households

First or last name based models were able to achieve results much closer to the ”ground truth” (self reported ethnicity) compared to census or internet models which tend to overestimate the size of the White population and underestimate minorities. The resulting model was applied to a larger set of Facebook users located in the US and analyzed various relationships among the ethnic groups such as number of friends, religion, relationships, videos shared and others.

2.2 Previously used methods

Several machine learning algorithms (and their derivations) were used previously for attribute classification of users including Support Vector Machines and Stacked SVM, Naive Bayes, Balanced Winnow2, and Logistic Regression. We explore some of these methods and use supplementary methods as described in Chapter 3.

2.3 Points that were not addressed

Previous works have attempted to predict the ethnic background of users on social media websites. One of the main issues that was encountered is the lack of ground truth data, requiring to seek outside resources to annotate the training sets. Related to the current research Rao et al. [2010] tried to predict the regional origin of users by selecting a set of Twitter posts from three South and three North Indian cities that were tweeted in English. Based on the surveyed literature, certain classification problems remain to be explored such as determining a user's origin/hometown, determining if a language that a user is authoring posts is his or her native language and whether a user who tweets from a specific country a native or an immigrant. In this thesis we will explore the topic of determining if the user who tweets from the US is an immigrant or a US native citizen.

2.4 Conclusion of Related Works

Our work utilizes the results obtained from the model built by Chang et al. [2010], that suggests that first and last names are good indicators of a person's ethnicity, and by using the ethnic distribution reported by the US Census Bureau a dataset of names was built based on which a social network was queried for tweets. We extend the work of Rao et al. [2010] as we investigate the problem of classifying a user's origin based on data from an entire country. A detailed approach and system design are described in the forthcoming chapters.

Chapter 3

Background Methods

In the preceding chapter we gave an overview of previous works and methods that were used for social media text classification. This chapter describes assumptions and constraints, and gives an overview of machine learning algorithms that were used in this research.

3.1 Assumptions and constraints

Before proceeding forward it is important to discuss assumptions and constraints that this work is predicated on.

As mentioned in Chapter 1, Twitter does not have an option to disclose detailed demographic user attributes except for user's name, location and unstructured description field. We thus rely on self reported profile features of Twitter users to determine a user's location. Initially we attempted to use the geo-tagging feature that records the exact location of user at the time he or she sends a tweet. However it was found that a small number of users have this feature enabled thus reducing the dataset. Instead we use a location attribute where users can specify their location. Since the entered location data is not validated by Twitter, it may contain imaginary places which were excluded from the dataset by verifying it against a list of known locations. The dataset collected represents users that currently report a location within the United States or a US territory.

User's origin or hometown is not reported by Twitter API, thus the ground truth whether a Twitter user is an immigrant or not cannot be established from the user's profile. Therefore to determine the immigration status of a user we examine profile description and the content of tweets for information about birthplace and/or current immigration status.

Twitter provides support for over 30 languages and allows users to use any characters in their user name including names in different locales and special characters such as ”lee eras wang ˆ_ˆ” or ”Ξo̲ 85”. Certain characters can be converted into standard ASCII characters, thus allowing identical names written in different locales such as ”Josè” and ”Jose” to be matched, while other user names with special characters such as ”:D @_@” were not included in the dataset. Tweets in English or normalized to Latin characters were considered. Tweets in other languages such as Chinese, Hindi and so on were not considered due to encoding and translation accuracy, thus excluding any Unicode characters.

Twitter accounts are not limited to personal use and are frequently utilized for marketing, news and political purposes. Celebrities such as actors, musicians and politicians make up a small portion of twitter users, however they have a high number of posts and followers. Some of the personal accounts remain inactive after a user creates an account with a few tweets or none at all. Other accounts exist for the sole purpose of retweeting posts from someone else, which can frequently be a bot or a fake account setup to spam users and spread advertisements. For the purpose of this project, only personal accounts which are active (or have been active) and contain at least 3000 posts (including retweets) were included in the dataset.

Twitter posts may include URLs, photos, videos, or twitter generated links to shorten posts longer than 140 characters. Because such posts contain external information that is not directly posted by the user, this text was excluded. Subsequently a user's profile also contains retweets from other users which were excluded, because it contains someone else's tweet.

Hashtag "#" is a way to specify a topic on Twitter. It may contain a single word or a combination of multiple words written as one (with a "#" prefix). While hashtags frequently contain misspelled words or acronyms, they often contain a message that is useful to infer the type of user such as "#newuscitizen" or "#freshofftheboat" and others. Hence, hashtags were included as part of the dataset with the "#" sign removed.

By default, user Tweets are public and visible to everyone. However, at any time users have an option to set their profile to private only, hiding their tweets and allowing only approved accounts to view the tweets. For the purpose of this work users with private profiles were disregarded due to an inability to collect their tweets.

3.2 Classifiers

Supervised machine learning methods have been frequently used for text classification. Depending on the features of a dataset such as size, or density of words, some algorithms may require more time or resources to build models. We began by exploring Naive Bayes and Support Vector Machines approach which was used by Rao et al. [2010] and Burger et al. [2011], and also consider Logistic Regression. Additionally we selected k-Nearest Neighbors and Decision Trees algorithms.

3.2.1 Multinomial Naive Bayes

There are various Bayesian classifiers that derive from Bayes' Rule (or Bayes' Theorem) in probability theory and statistics. Naive Bayes is a simple ("naive") supervised learning method that is frequently used for text classification problems. It relies on the assumption that the features in a dataset are mutually independent given the class label, which is often not the case (such as position of words in a sentence), but it still tends to perform very well under this assumption.

Consider a fixed set of classes $C = \{c_1, c_2, .., c_j\}$ and document X, then the Bayes' Theorem states that the probability of a class c given the training instance X, i.e. the posterior probability of class c, can be determined in terms of the prior probability of class c, the prior probability of training instance X, and conditional probability of the training instance given the class c:

$$P(c \mid X) = \frac{P(X \mid c)\, P(c)}{P(X)} \tag{3.1}$$

where,

- $P(c)$ = prior probability of class c

- $P(X)$ = prior probability of training instance X

- $P(c \mid X)$ = posterior probability of c given X

- $P(X \mid c)$ = conditional probability of X given c

For this research, we have two classes - ”Immigrant” and ”Native”. The one that has the highest probability of occurring wins the case and gets assigned the corresponding class. The training data D of m labeled documents X represented as a word vector $X = \{x_1, x_2, ..., x_n\}$ is used to estimate $P(X)$ and $P(c)$. Given that we assume that the probability of each word occurring in a document X is independent we can re-write equation 3.1 as:

$$P(c \mid X) = P(X \mid c)P(c) = P(x_1, x_2, \ldots, x_n \mid c)P(c) \tag{3.2}$$

Multinomial Naive Bayes is used for multinomial data, meaning there are more than two possible outcomes (a generalization of the binomial distribution). It is especially effective in text classification where data is represented as a vector of word counts, as in our case using a bag of words model. For each user, all tweets are combined into a sequence of words and we assume that the word position is independent of one another. To classify, we assume that there are a fixed number of classes, $c \in \{1, 2, 3, \ldots, m\}$, each with a fixed set of multinomial parameters. The parameter vector for a class c is $\vec{\theta_c} = \{\theta_{c_1}, \theta_{c_1}, \ldots, \theta_{c_n}\}$, where n is the size of an entire vocabulary (all possible words collected from a dataset), $\sum_i \theta_{c_i} = 1$ and θ_{c_i} is the probability that word i occurs in that class. Then the likelihood of observing a collection of tweets is a product of the parameters of the words that appear in tweets,

$$P(T \mid \vec{\theta_c}) = \frac{(\sum_i f_i)!}{(\prod_i f_i)!} \prod_i (\theta_{ci})^{f_i} \tag{3.3}$$

where f_i is the frequency count of the word i in a collection of tweets T. By assigning a prior distribution over the set of classes, $P(T \mid \vec{\theta_c})$, we can arrive at the minimum-error classification rule which selects the class with the largest posterior probability [34]

$$l(T) = argmax_c[\log P(\vec{\theta_c}) + \sum_i f_i \log \theta_{ci}] \tag{3.4}$$

$$= argmax_c[b_c + \sum_i f_i w_{ci}] \tag{3.5}$$

where b_c is the threshold term and w_{ci} is the class c weight for word i.

Despite its simplicity and "naive" assumption that every pair of features is independent of each other given the class label, it often outperforms more complicated algorithms in both the time it takes to build a model and accuracy [10, 22].

3.2.2 k Nearest Neighbors

K-Nearest neighbors (kNN) is a simple lazy learner algorithm that attempts to predict the label based on the classification of other instances that are nearby in the vector space. Unlike the previous methods such as Naive Bayes, it does not rely on computing prior probabilities. However it does require going through an entire training set to find the k nearest neighbors to the current instance. Depending on the number k, it looks at a specific number of neighbors and then uses the majority to vote for the new label. The distance to the nearest points can be determined by various distance functions such as Manhattan Distance, Hamming Distance and Cosine Similarity.

The value of parameter k must be chosen properly for the following reasons:

- k too small results in overfitting, meaning that it will fit the neighborhood too much by potentially capturing noise.
- k too large results in underfitting, meaning that model becomes too smooth and does not capture structure in the data.

Figure 3.1 demonstrates the effect of choosing value of k on the classification of a new instance. It present two classes : green square as Class 1 and blue diamond as Class 2. We want to predict the class label for the red star by examining its nearest neighbors. If the value of k is 3, then the output label would be Class 2, because there are 2 blue diamonds and 1 green square. If we choose the value of k to be 5, then it will assign Class 1 due to majority of green squares.

In our experiments we evaluated the various value of k from 1 to 15. Based on the results obtained, the optimal value of k appears to be 2 however for values greater than 2 the accuracy of the model did not change significantly, but did deteriorate slightly as value of k increased.

3.2.3 Support Vector Machines

SVMs find a linear classifier, called the maximum hyperplane, that can divide the data into two, thus allowing to distinguish between two classes. It tries to separate two classes as far as possible to prevent any overlap, which would lead to

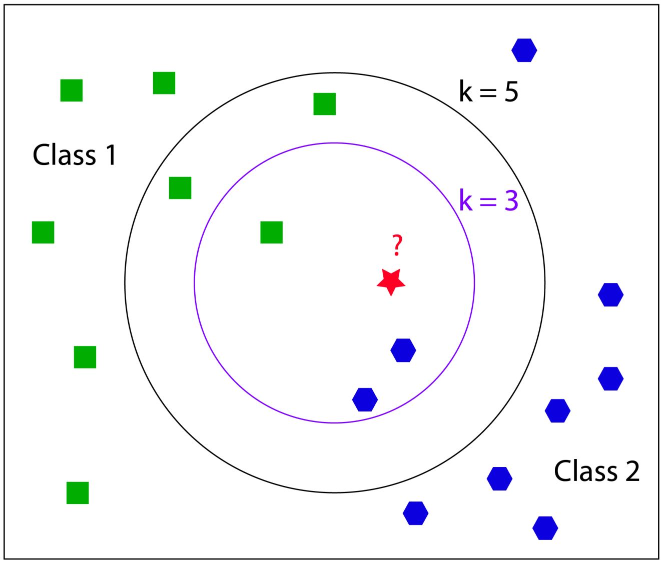


Figure 3.1: Classifying new item using kNN. Depending on the value of k, the class of red star can change. Figure borrowed from Perseus Documentation [28].

incorrect results. If the data is linearly separable then we can select two hyperplanes and maximize the distance between them. Then for two classes $\{+1, -1\}$ these hyperplanes can be defined as:

$$w * x - b = +1 \tag{3.6}$$

$$w * x - b = -1 \tag{3.7}$$

where x is a set of points x, w a vector perpendicular to the hyperplane commonly referred to as the weight vector, and b is an intercept term. In order to determine the maximum margin hyperplane (maximum distance between hyperplanes) we need to minimize the $||w||$.

An example of how to separate circles and squares using SVMs is shown in Figure 3.2, which presents an optimal way to linearly separate data, however there

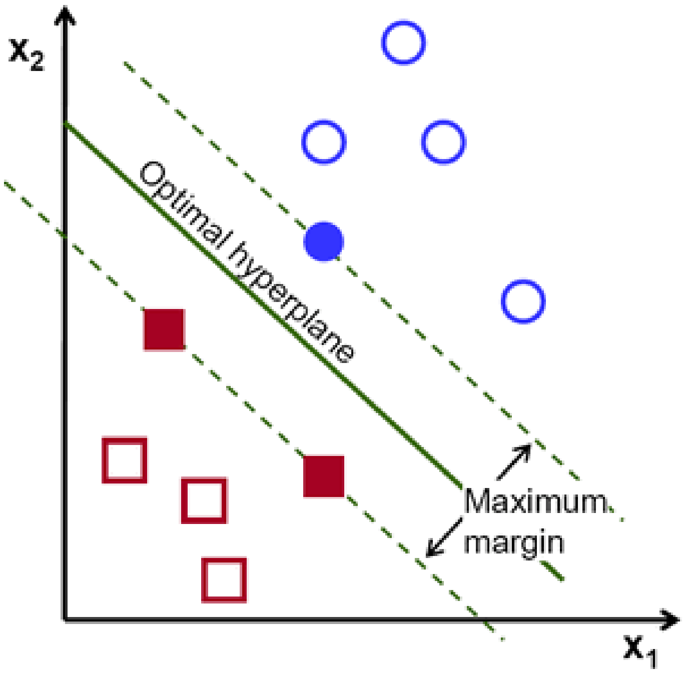


Figure 3.2: Using SVM to provide linear separation with hyperplane and maximal margin. Figure borrowed from OpenCV [25].

are infinitely many more ways that can separate this data. The dotted lines represent the boundaries of the maximum margin hyperplane. For some types of data it is not possible to separate the data with a straight line as shown in Figure 3.3, however the original space can be transformed via a kernel function to a higher dimensional space where data can be separated. We use polynomial kernel in our experiments, however there are other kernel methods available including linear kernel, rbf kernel and so on.

One of the advantages of SVMs is that they handle high dimensional input space very well, which is especially useful for text classification since it has many features. Unlike Bayesian classifiers, SVM is computationally intensive and requires

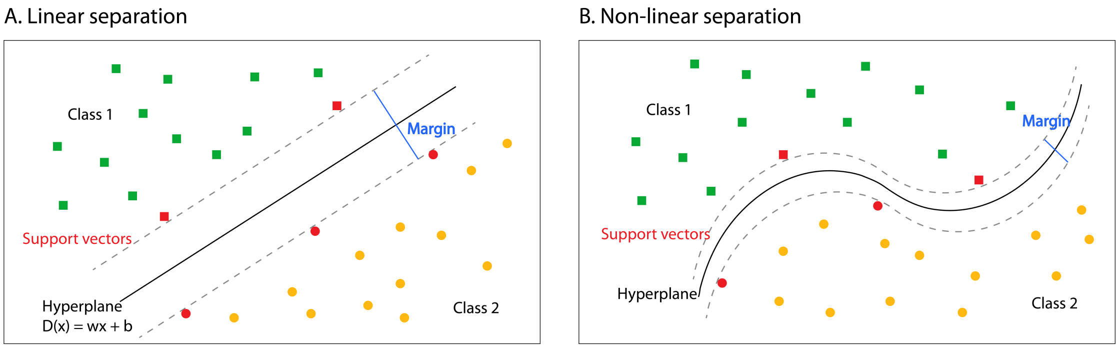


Figure 3.3: Linear separation vs. Non-linear separataion usin SVM. Figure borrowed from Perseus Documentation [29].

a lot of resources for discretization, normalization and repetitive dot product operations.

3.2.4 Decision Trees

Decision tree is a non-parametric supervised learning method that is used to predict the value of a target variable by creating simple decision rules inferred from training data. The process of classification of an instance begins from the root of the tree to some leaf node that contains the class label of an instance. Each internal node in a tree presents a test of some attribute of the instance and the branch descending corresponds to the attribute value. Figure 3.4 illustrates a decision tree that can be used to predict whether a person will play outside based on the weather attributes.

There are numerous implementations of the decision tree algorithms and for this project we used J48 method which is a Java implementation in Weka of C4.5

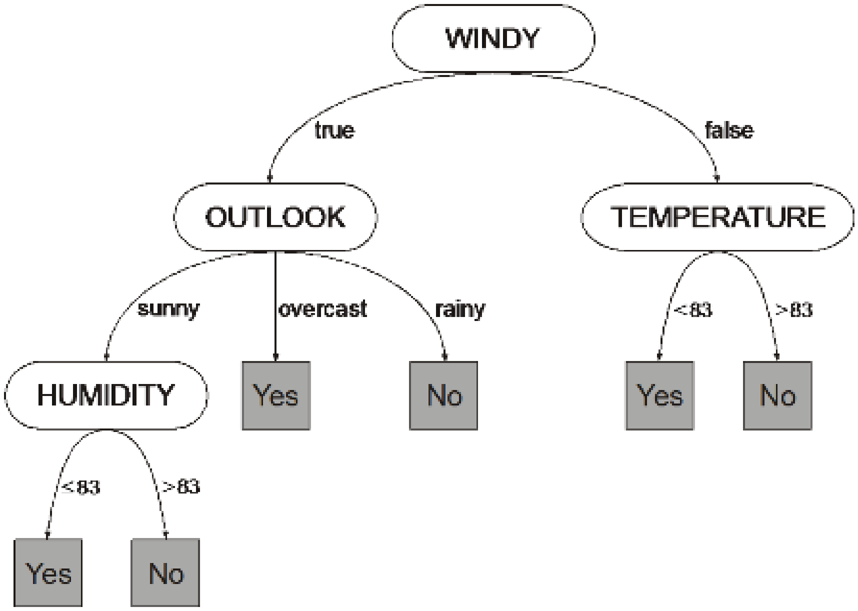


Figure 3.4: Decision Tree model to determine whether one should play outside. Figure borrowed from Wikipedia [46].

algorithm. It uses top down recursive divide-and-conquer strategy. At the root node, it selects an attribute to split the tree and creates a branch for each possible attribute value. This splits the instances into subsets for each possible branch and then the procedure is repeated recursively for each branch using only instances that reach that branch. It aims to get the smallest tree, therefore at the root node it attempts to find an attribute that produces the greatest information gain i.e. amount of information gained by knowing the value of the attribute.

3.2.5 Logistic Regression

Logistic regression is a type of exponential or log linear classifier which works by producing linear combination of weighted features from the input. The goal of logistic regression is given an observation X, where $X = \{X_1, X_2, \ldots, X_n\}$, compute the probability of label y, $P(y \mid X)$. It combines a weight vector with observations to determine the label. There are two components that define logistic regression, a weight vector β_i and observations X_i where β_0 represents a bias term:

$$P(h = Immigrant \mid X) = \frac{1}{1 + exp[\beta_0 + \sum_i \beta_i X_i]} \tag{3.8}$$

$$P(h = Native \mid X) = 1 - P(h = Immigrant \mid X) \tag{3.9}$$

When the exponential functions are plotted, it results in a sigmoid that looks like an S, that is always bounded between 0 and 1. When applied to text classification task we want to determine the probability that a tweet vector belongs to an "Immigrant" or "Native" class. The Naive Bayes classifier is a special case of logistic classifier, where all the weights are set independently, while the logistic regression sets the weights together. For example, if the words "health" and "care" are useful predictors and occur in a tweet more than once, Naive Bayes will assign strong weights, so their increased correlation will be double-counted. However logistic regression accounts for correlation and will reduce the weights to compensate for repetition.

3.2.6 Topic Modeling

For a collection of documents, topic modeling aims to uncover the hidden thematic structure and patterns that might be present in the collection. Topic models are a suite of algorithms that are used to organize, search and understand large archives of texts, images and other data that would be hard to annotate manually

[4]. We attempt to discover latent topical patterns that are contained in the data corpus. Then for each document in the corpus we annotate the topics according to those patterns, i.e. pick specific topics that are common to this document from a pool of discovered topics. For example, for a science journal a list of common words to appear would include ”genome”, ”human”, ”evolution”, etc. We can also analyze how the topic words change over time by examining their probability. Additionally connections between topics can be observed such as two related words ”ancient” and ”found” would be connected based on the frequency of occurrence in a document.

Latent Dirichlet Allocation (LDA) is a probabilistic model that generates topics based on word frequency from a set of documents [5]. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics, where each topic is modeled as an infinite mixture over underlying set of topic probabilities. The underlying intuition of LDA is that documents exhibit multiple topics, where each topic is a distribution over terms in a vocabulary. Different topics have different words with various probabilities, with one word may be present in multiple topics.

In this research we apply LDA to discover hidden patterns and topics in user's tweets. Examining the results that we obtain may assist in annotating a user based on the topic probabilities.

Chapter 4

System Design

A modular approach for the design of the system was chosen, so that its components can be changed, yet still allowing the system to complete its task. System components are as follows : user collection, tweet collection, data pre-processing and classification which are shown in Figure 4.1. Data collection is mostly automated and can to be run in parallel, thus improving processing time.

4.1 Data Collection

Twitter provides an API that allows integration with web services and applications. There are two types of APIs that are available : Streaming and REST APIs. The Streaming API provides an access to the global stream of public tweets in near real-time over a persistent connection and is limited to recent tweets. The REST API provides a way to read and write tweets, access user's profile information, modify account settings, get number of friends and followers, and access historical Twitter data, although it is rate-limited (see Section 4.6.1 for API limits). The Search API, which is a part of REST API, can be used to build search queries to find relevant statuses and users. However, Twitter imposes limits on Search API restricting searches on data that is older than one week. Since we were interested in collecting all tweets for users, we used REST APIs for this research.

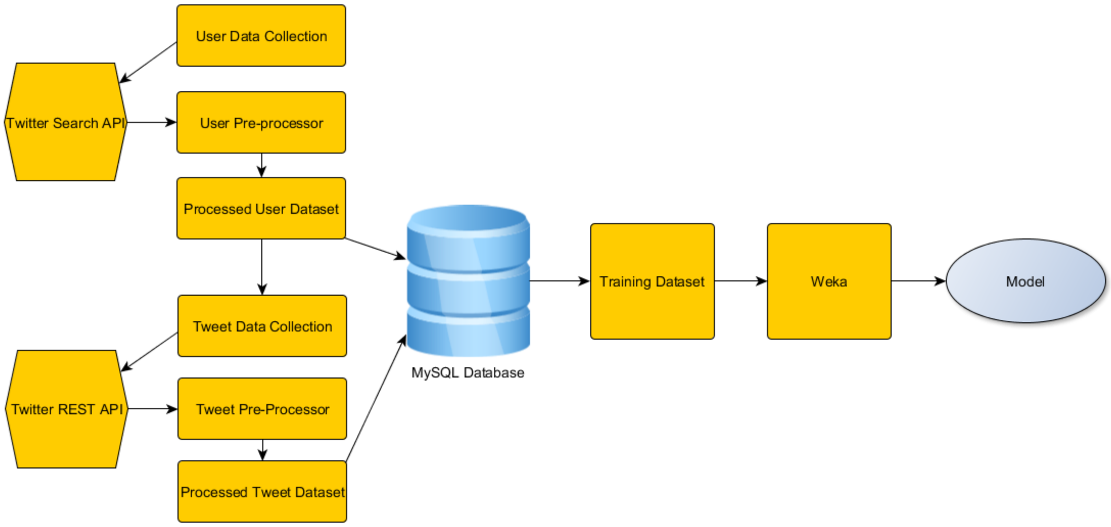


Figure 4.1: System Design

With over 320 million active users on Twitter, approximately 88% include public profiles allowing to read the timeline. While Facebook does contain more demographic attributes for each user, and a couple of years ago made all user profiles public, the posts on user timelines are frequently limited by the privacy settings or contain links and videos rather than text posts. Due to privacy settings, neither Twitter nor Facebook releases a user's email address for public access which would provide a link to connect the tweet data with Facebook's extended demographic profile, and allow to collect a larger training dataset.

4.2 Selection of Twitter users

There are over 67 million active Twitter users in the United States, which would make the task of collecting an entire Twitter user base extremely complex

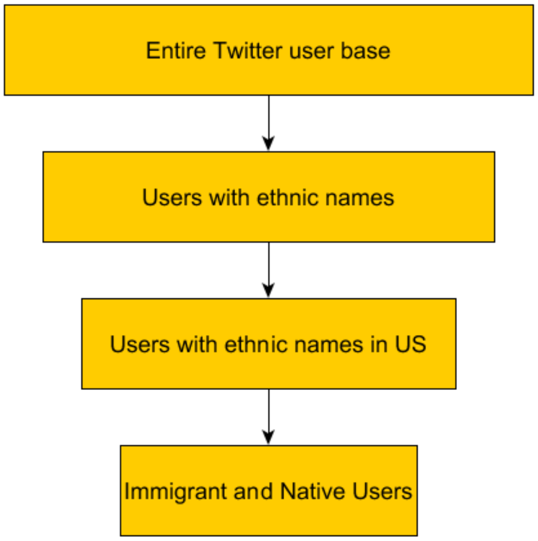


Figure 4.2: Process of selecting Twitter users

both in time and resources. Since we are interested in immigrant and native US population, we generated a list of names for Asian, Latino and Caucasian races by combining first and last names to form a potential user name. This narrows down the amount of users to collect and increases the chances of finding the target population. The process of user selection is demonstrated in Figure 4.2. For each full name, one or more matching user profiles were collected. While Twitter does not offer exact matches for names, it returned users relevant to the search query. Then for each user, we collected the most recent tweets that were posted by the user up to a maximum of $3,200$ tweets per user. Details on user and tweet dataset collection and pre-processing are specified in Chapter 5.

4.3 Data pre-processing

The dataset collection was split into two steps : Twitter user collection and Tweets collection (per user). There are two datasets that were collected: a user dataset and a tweet dataset (based on user dataset). Both datasets were collected in JSON format and the python built in JSON library was used to parse the files combined with a collection of regular expressions that was used to process text data. After the user dataset was collected, the profile information was normalized by removing accents and punctuation, and removing extraneous profile parameters. Next duplicates were removed and filtered by location (excluding unmatched locations). The tweet dataset was processed in a similar matter with an addition of removing URLs, hashtags, punctuation, and so on. A detailed explanation of data processing is included in Chapter 5.

4.4 Validation of data

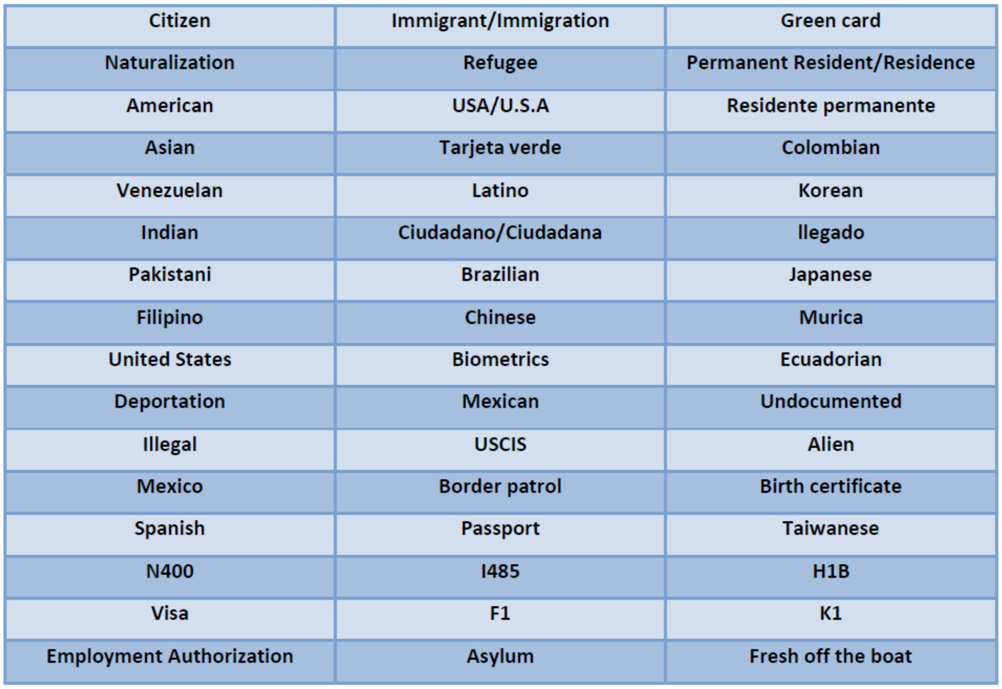
One of the most important challenges is verifying whether the user is an "immigrant" or not. Since Twitter does not provide this demographic information, the training set for "immigrant" and "native" users had to be annotated manually. We rely on the assumption that user's self reported posts regarding their immigration status are accurate.

After the tweet dataset was collected and processed, a set of keywords and phrases was created that identifies candidates for immigrant class label. The phrases in Table 4.1 were used to select users based on matching tweets from collected dataset. We used the matching tweets to determine whether a user is an immigrant. For some words such as "immigrant" we used its root to capture more words that are related such as "immigration", "immigrate", "immigrating", "migrant", "emigrant" and so on. The content of matching tweets was then manually examined to make sure the user is not posting a news article headline or other unrelated information.

To establish whether a user is a native US citizen, we manually examined user's profile attributes including profile description, pictures, location and content of tweets. The profile pictures were used to establish user's race, discarding any users without profile pictures. Then profile description was analyzed for information about a hometown or user's origin, occupation, hobbies, family information, etc. The content of tweets was reviewed by selecting a random sample of tweets to discover tweets containing information on user's origin. We rely on the assumption that personal twitter accounts that contain a profile picture of a Caucasian person, with a reported location within US, is a native US born citizen. If a user's description contained keywords such as "native New Yorker" or "born and raised in California", they were included in the training set labeled "Native", although not required to be labeled as "Native".

4.5 Software used

This section gives a brief overview of the main software, languages, and libraries that were used to implement the system architecture presented in Figure 4.1.



Citizen	Immigrant/Immigration	Green card
Naturalization	Refugee	Permanent Resident/Residence
American	USA/U.S.A	Residente permanente
Asian	Tarjeta verde	Colombian
Venezuelan	Latino	Korean
Indian	Ciudadano/Ciudadana	llegado
Pakistani	Brazilian	Japanese
Filipino	Chinese	Murica
United States	Biometrics	Ecuadorian
Deportation	Mexican	Undocumented
Illegal	USCIS	Alien
Mexico	Border patrol	Birth certificate
Spanish	Passport	Taiwanese
N400	I485	H1B
Visa	F1	K1
Employment Authorization	Asylum	Fresh off the boat

Table 4.1: Search words/phrases used for annotation of ”Immigrant” class label

Platform

Data collection, pre-processing and testing environment was setup on a 64-bit Intel $i5$ processor running a Windows operating system.

XAMPP

XAMPP is a cross platform web server solution package that includes the following:

- Apache: HTTP Server where web applications can be hosted. Local instances are supported.

- MariaDB (formerly MySQL) : open source relational database management system

- PHP : server side scripting language

It allows to quickly setup a local web server with a database as a testing environment. XAMPP provides an easy to use GUI where developers can quickly adjust the settings.

PHP

PHP is a server side scripting language that is primarily used for web development. Due to its tight integration with MySQL (see below), it was chosen to import JSON datasets into the database. It contains necessary JSON and SQL libraries to parse data files and execute data manipulation queries with MySQL database.

MariaDB / MySQL

MySQL is an open source database management system that is frequently used for web-based applications. Due to acquisition by Oracle, a community developed forked called MariaDB was created to continue development under the GNU GPL license. MySQL support was officially replaced with MariaDB in XAMPP distribution at the end of 2015.

Python

For this project Python *v*2.7 was chosen as it is a versatile, portable and robust programming language. It is a hybrid language, thus it can be used for writing standalone scripts that perform specific task and allows for quick debugging. One of the main reasons why this version of Python was chosen over version 3.*x* is due to Twitter authorization libraries being compatible only with version 2.7.*x*. Additionally, there are extensive libraries available for natural language and JSON processing.

In addition to standard available libraries such as ”json”, ”sys”, ”datetime” and others, the following external libraries were used for the purpose of this project:

- NLTK : Natural Language ToolKit (NLTK) is an Open Source suite of libraries that was used to process the tweet dataset.
- OAuth : Open standard for authorization (OAuth) is an authentication protocol that allows applications to act on behalf of users without sharing password through access tokens.
- Gensim : Open source topic modeling toolkit which contains implementation of TF-IDF, Latent Dirichlet Allocation and other modeling algorithms.

Weka

The Waikato Environment for Knowledge Analysis (Weka) is an open source software environment that was developed using Java at the University of Waikato, New Zealand. It contains a collection of visualization tools and machine learning algorithms together with a user interface that were suitable for the research goals of text processing and classification. Weka can be used to customize existing machine learning methods and import your own algorithms to build models. Data can be imported by using an ARFF file, which is a text file that contains metadata about the dataset attributes and labeled training data.

4.6 Limitation / Issues

After collecting tweets for approximately 50 thousand users with ethnic names, we uncovered lots of users that didn't contain enough data to label them as immigrant or native. These users had names that identify them with a certain ethnicity group, but neither their description nor content of their tweets revealed whether or not they are an immigrant, thus excluding a great number of users and their tweets from training dataset.

Twitter is mostly used on mobile devices and frequently users post misspelled words. Even though most smartphones are equipped with an "auto-correct" mode, some words are deliberately misspelled such as "night" is frequently spelled as "nite". Thus grammatical mistakes further complicated analyzing tweets for matching keywords to establish user's origin.

4.6.1 Twitter API Limits

Twitter imposes rate limits per user and per application. Rate limits are split into 15 minute intervals with every request requiring an authentication. Depending on the type of requested data, each interval allows either 15 requests or 180 requests. The types of requests that were used in this research are below :

- https://api.twitter.com/1.1/users/search.json : Provides a search interface to public user accounts. Users can be queried by name and other criteria such as location, company, description, and number of followers. The search does not support exact matches. Twitter allows 180 requests per interval. Each request returns only up to 20 potential user results per page with up to 50 pages for each search query.

- https://api.twitter.com/1.1/users/show.json : Returns information about a user that was specified by his or her user_id and screen_name. Twitter API allows up to 180 requests per each interval. Each request returns all available information, allowing to collect up to 180 users every 15 mins.
- https://api.twitter.com/1.1/statuses/user_timeline.json: Returns a sample of the most recent Tweets, a maximum of 3200, which includes retweets in the total count. Retweets may be excluded from the sample, however, it will not increase the total number of user statuses returned. Twitter API allows up to 180 requests per user and 300 requests per application. Each request returns up to 200 statuses where the total count includes retweets, replies and deleted/suspended content.

These rate limits significantly slowed down the data collection specifically for user collection. However, each developer account can create virtually an unlimited number of applications that have separate authentication keys can work in parallel. By creating multiple applications, data was aggregated in parallel, thus reducing the overall amount of time required for data collection.

When searching for users using "GET users/search" function from the REST API, it only returns the first 1000 matches and exact matches are not supported, meaning that for a queried name "Kevin Li" it may return users that match either "Kevin" or "Li" or any name that contains search query such as "Lily". This resulted in obtaining users that do not belong to their ethnic category. For example, after collecting the user dataset, it was found that actor/comedian Kevin Hart's profile was also collected as part of the "Asian" group set.

Chapter 5

Dataset

This chapter provides details about datasets that were collected. It describes the methods that were used to collect and process the datasets.

5.1 Type of data obtained

There are two datasets that were used for training classifiers : user and tweet datasets. The user dataset contains a list of all users that we collected along with their profile information. The tweet dataset was built by using the list of user IDs from the user dataset.

5.1.1 User Dataset

Before collecting the tweet dataset we must first know which user accounts we are interested in. Initially the Search API was used to query a list of keywords provided in Table 4.1, which yielded limited amounts of tweets due to the Search API only searching against a sample of recent Tweets published in the past 7 days. The majority of the results returned were news accounts or retweets related to elections, movies or some other trending topic such as:

"Superman is a working class immigrant who uses his power for the public good. Batman is a rich semi-despot. course sanders likes superman"

Therefore a user dataset was required to collect tweets that contained both immigrants and natives. To address the task of ethnicity identification, we used the data from US Census Bureau to create a list of the most common surnames for each race.

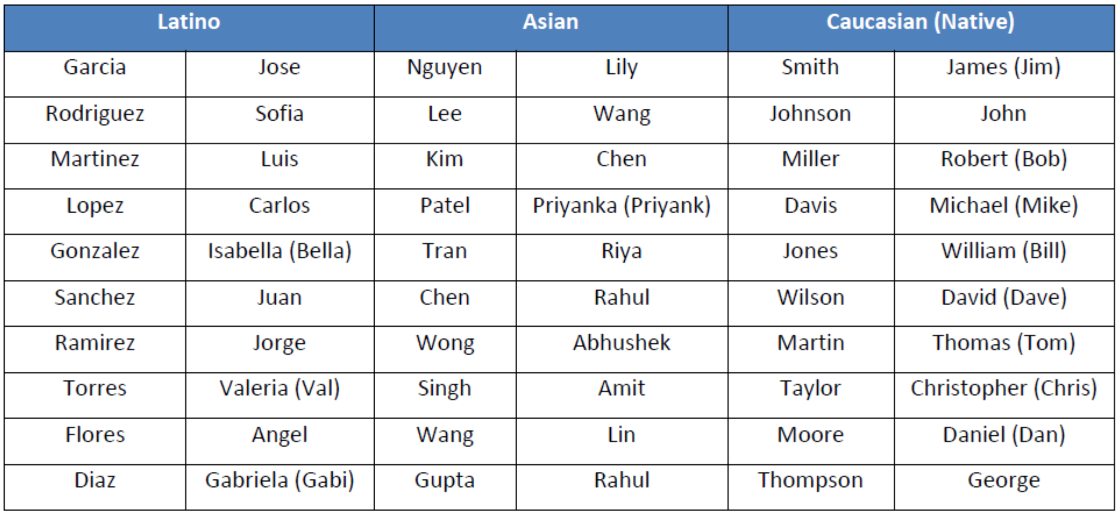
As of 2014, the race and Hispanic origin[1] of immigrants in the US is distributed as follows:

- Caucasian/White - 47.5%
- Latino origin[2] - 45.7%
- Asian - 26.2%
- Other race - 14.8%
- Black or African American - 8.7%
- Two or more races - 2.2%
- American Indian and Alaska Native - 0.4%
- Native Hawaiian and other Pacific Islander - 0.3%

Thus Asians and people of Latino origin make up the majority of the immigrants in the United States. Even though "Caucasian/White" immigrants comprise 47.5% of US immigrant population, they were excluded from immigrant user

[1] It is important to note that the concept of race is separate from the concept of Hispanic origin, thus percentages of Latino origin should not be combined with race categories

[2] Latino origin not included in race total



Latino		Asian		Caucasian (Native)	
Garcia	Jose	Nguyen	Lily	Smith	James (Jim)
Rodriguez	Sofia	Lee	Wang	Johnson	John
Martinez	Luis	Kim	Chen	Miller	Robert (Bob)
Lopez	Carlos	Patel	Priyanka (Priyank)	Davis	Michael (Mike)
Gonzalez	Isabella (Bella)	Tran	Riya	Jones	William (Bill)
Sanchez	Juan	Chen	Rahul	Wilson	David (Dave)
Ramirez	Jorge	Wong	Abhushek	Martin	Thomas (Tom)
Torres	Valeria (Val)	Singh	Amit	Taylor	Christopher (Chris)
Flores	Angel	Wang	Lin	Moore	Daniel (Dan)
Diaz	Gabriela (Gabi)	Gupta	Rahul	Thompson	George

Table 5.1: Top 10 Most common surnames and first names (with nicknames) for Latino, Asian and Caucasian groups

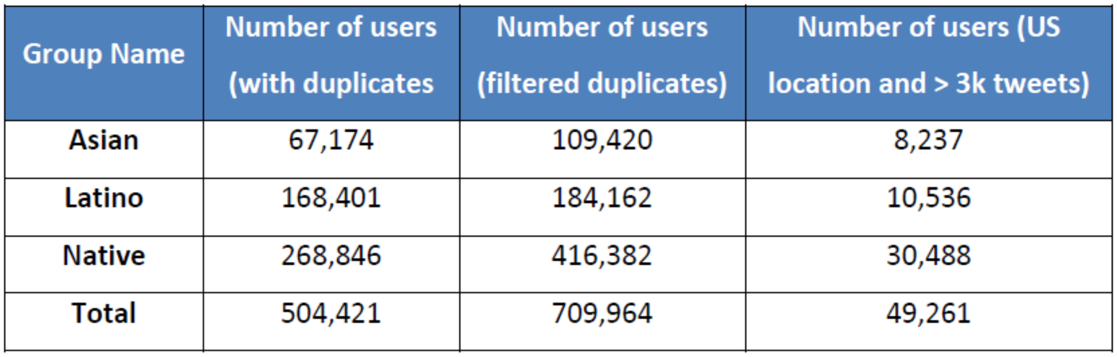
dataset since there are a lot of countries where people identify themselves as ”Caucasian/White”, resulting in a large variety of unique and/or rare surnames with some of them overlapping with the native Caucasian population. For a sample of ”Native” users we used the surnames that most frequently occur for people that identify themselves as ”Caucasian/White” which accounts for 62.1% of the US population.

Based on the research by Chang et al. [2010] we used most common first names for each race and combined them with every last name thus producing a list of full names, which was subsequently used to create a query and use the REST API to gather user dataset. Table 5.1 provides the top 10 first and last name choices for each ethnic group. This provided access to a larger sample of tweets for each user including older tweets up to the first posts for some users (due to API limits listed in Chapter 4.6.1).

The initial user dataset contained a large number of duplicates since the REST API does not support exact searches and the same list of first names was applied to each surname for a corresponding ethnicity group. The de-duped dataset was then filtered to include only those users that report a location within the US. We compared a user's self reported location with a list of 50 US states and their most populated cities as reported by the US Census Bureau, excluding those users who do not report any location or report a location outside the United States and/or non-existing places such as ”Instagram”, ”Planet Earth” and so on. Questionable locations such as ”NYC” were manually analyzed to determine whether it is an acronym or a slang term for a city and/or state (e.g ”Cali” means ”California”), which expanded the list of acceptable locations. Finally, we selected only users that had 3000 or more posts (which includes retweets in the total count), because more active users tend to post more updates about their personal life events. We demonstrate in Table 5.2 how the amount of users reduced significantly after processing and filtering.

5.1.2 Tweet dataset

After the user dataset was filtered by location, we used the obtained user IDs to collect the maximum amount of tweets per user as allowed by API rate limits. Retweets were excluded from the tweet dataset because they are not posted by the user and do not add any value to classifying. For Asian and Latino users, tweets were frequently written in a language other than English, making the data very noisy.

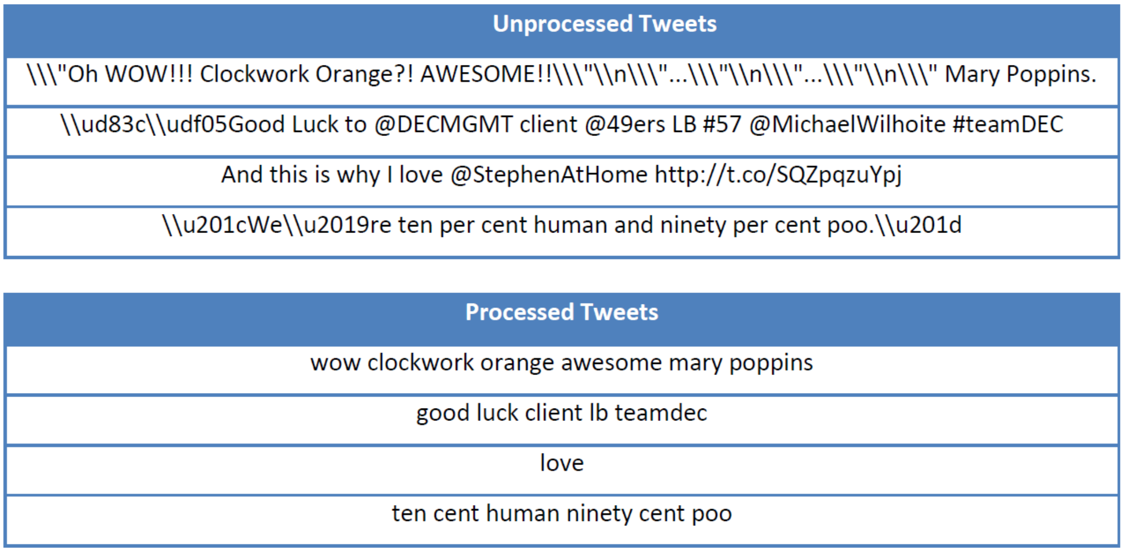


Group Name	Number of users (with duplicates	Number of users (filtered duplicates)	Number of users (US location and > 3k tweets)
Asian	67,174	109,420	8,237
Latino	168,401	184,162	10,536
Native	268,846	416,382	30,488
Total	504,421	709,964	49,261

Table 5.2: User group statistics. Asian and Latino – Immigrant group. Native – US born users

Thus we used the python module ”unicode” which transforms Unicode characters into their closest ASCII representation. Then non-English tweets were normalized into corresponding ASCII characters such as ”à, è, ì, ò, ù” were converted to ”a, e, i, o, u” respectively. Tweets, although originally started as short (140 character) text, were soon expanded to include photos, videos, retweets and links which is noisy data and would not contribute to a classifier. Thus the following processing rules were applied and Table 5.3 shows the comparison of tweets before and after processing:

- URLs : links that were included in the text part of the tweet were removed
- Usernames : @username were removed
- Hashtags : ”#” was removed
- Special characters : any non alphanumeric characters were removed



Unprocessed Tweets
\\\"Oh WOW!!! Clockwork Orange?! AWESOME!!\\\"\\n\\\"...\\\"\\n\\\"...\\\"\\n\\\" Mary Poppins.
\\ud83c\\udf05Good Luck to @DECMGMT client @49ers LB #57 @MichaelWilhoite #teamDEC
And this is why I love @StephenAtHome http://t.co/SQZpqzuYpj
\\u201cWe\\u2019re ten per cent human and ninety per cent poo.\\u201d

Processed Tweets
wow clockwork orange awesome mary poppins
good luck client lb teamdec
love
ten cent human ninety cent poo

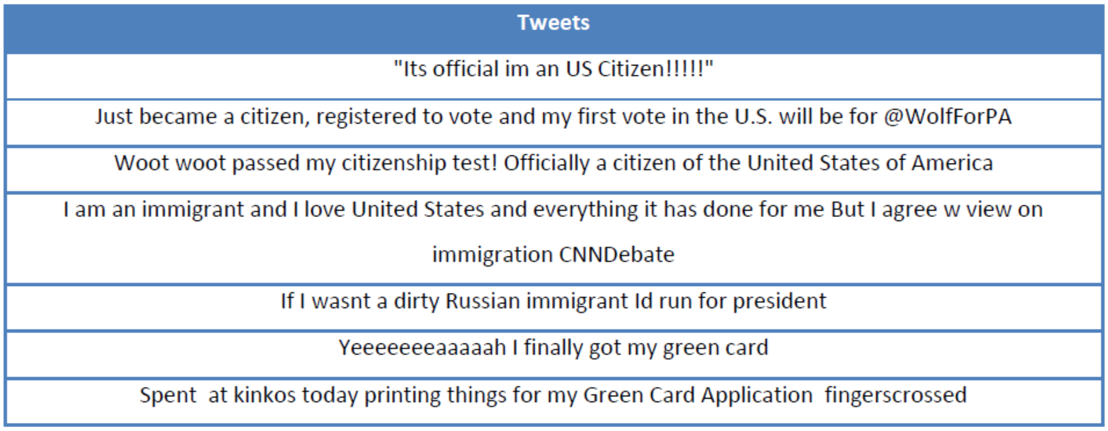
Table 5.3: Tweets before and after processing

- Unicode characters : the REST API returns Unicode characters as escaped character code such as ”\\u2019” which may include words in other languages. Unicode characters were removed only leaving ASCII characters.

- Stop words : certain words such as ”a”, ”for”, ”the” introduce noise into the data without contributing to classifier and thus were removed.

- Upper case : The same words in upper and lower case are treated differently by classifier thus increasing the complexity and leading to overfitting. Thus the corpus of the tweet dataset was converted to lowercase.

Next we were tasked to evaluate Asian and Latino tweets along with corresponding user description to determine which users were immigrants. Since no ground truth data is available to compare this data against to, we selected a sample of users whose tweets closely matched the keywords and/or phrases presented. The



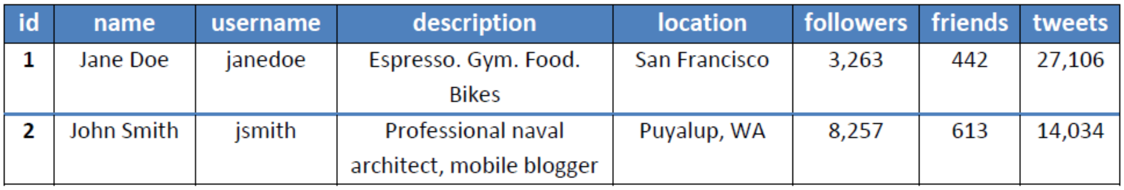
Tweets
"Its official im an US Citizen!!!!!"
Just became a citizen, registered to vote and my first vote in the U.S. will be for @WolfForPA
Woot woot passed my citizenship test! Officially a citizen of the United States of America
I am an immigrant and I love United States and everything it has done for me But I agree w view on immigration CNNDebate
If I wasnt a dirty Russian immigrant Id run for president
Yeeeeeeeaaaaah I finally got my green card
Spent at kinkos today printing things for my Green Card Application fingerscrossed

Table 5.4: Sample of tweets matching "Immigrant" evaluation keywords

results we obtained were mixed because users that were immigrants contained a lot of tweets about immigration but not directly matching the keywords or discussing a topic, thus slowing down the process of selecting "Immigrant" users for training dataset. Table 5.4 demonstrates tweets that contain a text relevant to a user's immigration status and therefore were labeled as an "Immigrant" class. The process of selecting users for the "Native" class was different since keywords were no longer applicable to this group. To determine if a user belongs to the "Native" class, we manually analyzed the contents of user accounts by reading through the description and selecting only users that had a profile picture as described in Chapter 4.4.

5.1.3 Database import

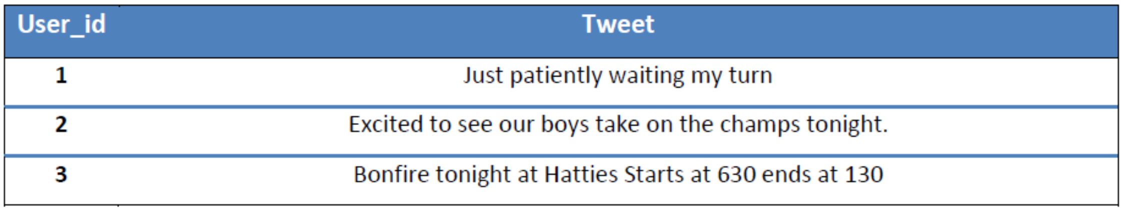
User and tweet datasets were initially collected into JSON files that were kept in the file system. As the number of users grew, it took more time to maintain and process files manually which prompted setting up the database to import the data.



id	name	username	description	location	followers	friends	tweets
1	Jane Doe	janedoe	Espresso. Gym. Food. Bikes	San Francisco	3,263	442	27,106
2	John Smith	jsmith	Professional naval architect, mobile blogger	Puyalup, WA	8,257	613	14,034

Table 5.5: User Table Sample

A database provides several advantages over file system such as fast I/O operations, data is kept organized by the database management system (DBMS) and support for multithreaded applications (for future work). XAMPP provides a test environment for web based applications which includes PHP and MariaDB (formerly MySQL). Owing to a tight integration between MariaDB and PHP the process of creating importing scripts for users and tweets was seamless. A database was designed to keep track of users and tweets. The users table (see Table 5.5) keeps track of each profile attribute in a separate column with user ID serving as Primary_Key (unique ID). Tweets are stored in two tables: split and combined. Split tweets table contains each tweet in a separate row with a corresponding user_id. Combined tweets table stores all tweets for a specific user in one row requiring the data to be stored as BLOB data type due to its size. An example of the data stored in the split tweet table is shown in Table 5.6. Having a dedicated database allows to mix and match feature attributes with ease such as number of friends and/or followers as a new feature attribute, because each feature is stored as a separate column.



User_id	Tweet
1	Just patiently waiting my turn
2	Excited to see our boys take on the champs tonight.
3	Bonfire tonight at Hatties Starts at 630 ends at 130

Table 5.6: Tweets Table Sample

5.2 What could not be obtained

Information about user’s immigration status is not public information, however Facebook users can specify the ”Hometown” and ”Current City” in their profile details. Matching Twitter users with Facebook account information would help create a larger training set because a user’s self reported hometown establishes ground truth in that case. However, neither Facebook nor Twitter provides identifying information that would tie a Twitter account to a Facebook such as email address.

5.3 Limitations/Issues

The Twitter REST API provides an option to set a flag ”include_rts” to false which will strip any native retweets from a user’s timeline, which was set to true, because retweets are not user’s original posts. However many web services, iOS and Android apps, and even certain devices provide native Twitter integration that lets applications post ”status updates” to a user’s account on behalf of a user. Thus these generated statuses are still classified as ”user tweet” (not a retweet) and include generated texts such as ”I just posted a photo to facebook”, ”just reached level 80 as Dursey on World of Warcraft ", "This week I walked 24859 steps so far" and others. An effort was made to remove these posts from the dataset because they do not add value to the classifier, however for some users, it may still contain these tweets.

5.4 Ethics/Privacy

By default, a user's profile is set to public, thus his or her profile can be indexed by the search engine and the user timeline containing all available tweets can be queried directly through the Twitter API including text, photos, videos and links to other websites. However Twitter still returns profile information for protected (private) profiles, excluding their tweets. Thus to make sure that no private profile information was accessed, protected profiles were excluded from the user dataset and tweet dataset. No additional personal information other than provided by the Twitter API was obtained for collected users. Chapter 7.2 opens up a discussion on security and privacy issues associated with information on twitter.

Chapter 6

Experiments and Results

We performed experiments using supervised learning classifiers in Weka because it offers extensive libraries of classification algorithms, report generation and attribute selection filters.

6.1 Preprocessing Training Data in Weka

Weka requires the training data to be converted into ARFF (Attribute Relation File Format) file which represents an ASCII text file that describes a list of instances sharing a set of attributes. We generated the ARFF files by reading user tweets from the database and combining them in a single string for each user. ARFF files contain information about the attributes and data that is labeled for each training instance. The training set contains a total of 200 instances, with 100 users for each class, "Immigrant" and "Native". Multiple files were created to test classifiers for the following options:

- Text–only vs. text–with–numbers : After tweets were processed as described in Chapter 5, numbers were retained in tweets to allow comparison of classifiers for tweets with and without numbers such as :

  " I cant believe its January 01 2015*" vs. " I cant believe its January"*

- Top n features in tweets : We select the top n number of tweets for each user to be included to train the classifier. The n values include $\{50, 100, 500, 1000, 1500, 2000, All\}$ tweets. By varying the number of n we experiment with what is the minimum number of tweets are required before the classifier accuracy begins to deteriorate.

- Stemming dataset : We apply stemming algorithm to reduce words to their root form and compare the performance against non-stemmed dataset.

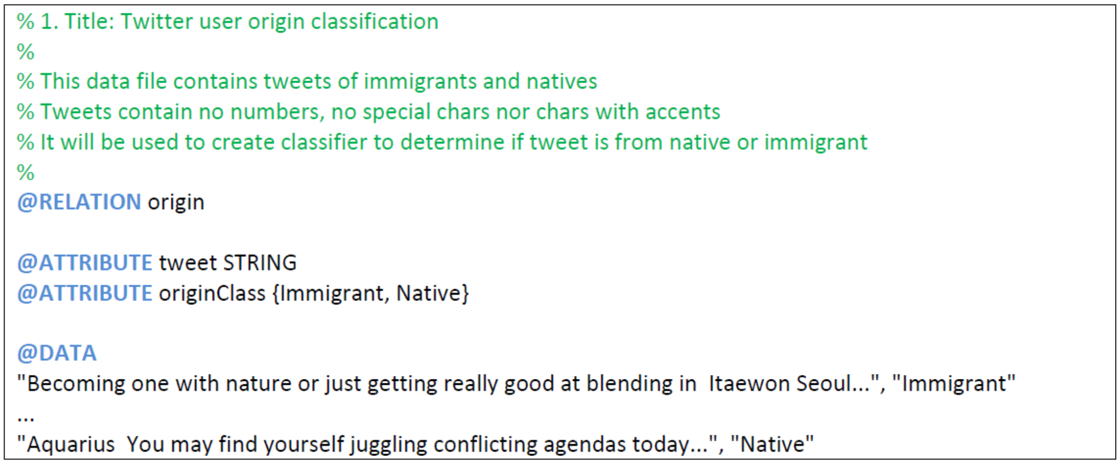
- Including stopwords: stopwords were initially excluded from the tweet dataset. We evaluate how the presence and absence of stopwords affects the accuracy of models.

- TF–IDF model: We analyze how using tf–idf scheme vs. word occurrence affects the performance.

- Removing keywords: We remove keywords listed in Table 4.1 from tweet dataset, that were used to identify candidates for ”Immigrant” class label, and compare the accuracy against dataset that includes them.

Next, the training ARFF file such as shown in Figure 6.1 was loaded into Weka to perform further processing of data.

The ”StringToWordVector” filter performs the conversion of string attributes in a document into attributes that represent word occurrence information from the text contained in the string, which is commonly known as bag of words model. The text document may contain just one sentence, hundreds or even thousands of



```
% 1. Title: Twitter user origin classification
%
% This data file contains tweets of immigrants and natives
% Tweets contain no numbers, no special chars nor chars with accents
% It will be used to create classifier to determine if tweet is from native or immigrant
%
@RELATION origin

@ATTRIBUTE tweet STRING
@ATTRIBUTE originClass {Immigrant, Native}

@DATA
"Becoming one with nature or just getting really good at blending in  Itaewon Seoul...", "Immigrant"
...
"Aquarius  You may find yourself juggling conflicting agendas today...", "Native"
```

Figure 6.1: Sample ARFF file used for classification

sentences (thousands of tweets per user in our case) which would demand a lot of resources and be computationally inefficient.

6.2 Evaluation parameters

In order to determine how well the classifiers had performed, we used the following metrics :

- Accuracy : measures the proportion of correctly classified instances

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{6.1}$$

- Precision : measures the proportion of returned documents that are correct

$$Precision = \frac{TP}{TP + FP} \tag{6.2}$$

- Recall : measures how many true positives the system selected

$$Recall = \frac{TP}{TP + FN} \tag{6.3}$$

- F-measure : measures the geometric mean of precision and recall

$$\text{F-Measure} = \frac{2 * Precision * Recall}{Precision + Recall} \tag{6.4}$$

where

TP = True Positives, number of positive examples that were labeled as such.

FP = False Positives, number of negative examples that were labeled positive.

TN = True Negatives, number of negative examples that were labeled as such.

FN = False Negatives, number of positive examples that were labeled as negative.

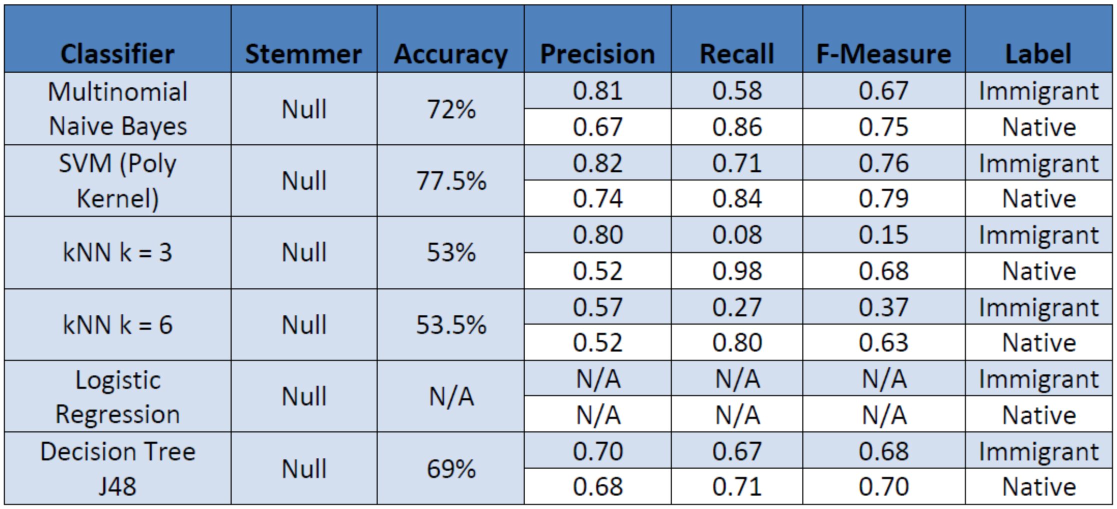
6.3 Initial Results

After applying the "StringToWordVector" filter, the generated dictionary contained over $140,000$ attributes (distinct words). The generated vector with word occurrences is sparse for all users because it's based on the entire dictionary. The initial results were conducted using the complete feature set. The results presented in Table 6.1 show that classifying documents with thousands of tokens, makes the classification problem very hard and requires a lot of processing power. For multinomial Naive Bayes the classifier accuracy did not exceed 72%. SVM model performed better achieving 77.5% outperforming the rest of classifiers. This can be attributed to the use of kernel trick that maps original space into a higher dimensional space to provide linear separation of data, which is useful for datasets with large number of attributes. For kNN we cross validated the k value from 1 to 15 with k equals 6 being the optimal producing 53.5% even though other values of k did not result in significantly lower results. Given the number of attributes we had, it contained noisy data, and since kNN algorithm works by computing the distance between all features, it could have selected less meaningful attributes. The Logistic Regression model could not be built due to the large number of tokens, and amount of memory required that exceeded test system's capabilities. We used the initial results presented in Table 6.1 to compare performance against models in the remaining experiments.

The surplus of attributes produced poor results and required to be filtered. After examining the produced attributes, we noticed that not all words were actual words because of spelling mistakes, hashtag topics and slang terms such as "aaaaaaa", "hf", 'zzzzz", and "hmhmhmhmfjesf". Further, some words occur less often than the others and would not contribute to classifier accuracy. Thus we applied dimensionality reduction filters that select a subset of attributes and are aimed improve the accuracy of models.

6.4 Dimensionality Reduction

Text classification requires analyzing vast amounts of documents that contain rich feature sets (dictionaries). These dictionaries however include features that are inherently noise data and increase the complexity of the classification task. Dimen-



Classifier	Stemmer	Accuracy	Precision	Recall	F-Measure	Label
Multinomial Naive Bayes	Null	72%	0.81	0.58	0.67	Immigrant
			0.67	0.86	0.75	Native
SVM (Poly Kernel)	Null	77.5%	0.82	0.71	0.76	Immigrant
			0.74	0.84	0.79	Native
kNN k = 3	Null	53%	0.80	0.08	0.15	Immigrant
			0.52	0.98	0.68	Native
kNN k = 6	Null	53.5%	0.57	0.27	0.37	Immigrant
			0.52	0.80	0.63	Native
Logistic Regression	Null	N/A	N/A	N/A	N/A	Immigrant
			N/A	N/A	N/A	Native
Decision Tree J48	Null	69%	0.70	0.67	0.68	Immigrant
			0.68	0.71	0.70	Native

Table 6.1: Model evaluation without attribute selection (All features)

sionality reduction is a technique that aims at reducing the high dimensionality of a text document by introducing a new feature space. It can be divided into:

- Feature extraction : creates new features from the original feature set by mapping a high dimensional space to a space with less dimensions. There are various feature extraction methods such as Principal Component Analysis (PCA) or Singular Value Decomposition (SVD).

- Feature selection : selects a subset of specific words/features based on their computed quality metric. The definition of feature selection is:

  Given a feature set $x = \{x_i \mid i = 1, \ldots, N\}$, *find a subset* $x_M = \{x_{i1}, x_{i2}, \ldots, x_{iM}\}$, *where* $M < N$, *such that it optimizes a function* $J(Y)$.

Unlike feature extraction which produces a new set of attributes and possibly results in losing some information about the original set in the process, feature selection just chooses some values from a set of features and can make use of class information. While Weka supports both methods, feature selection was chosen over feature extraction because it was shown to be successful for Yang and Pedersen [1997], where removal of 98% of unique terms actually led to an improved text categorization classifier accuracy.

To apply feature selection on the training data, Weka requires to setup configuration options: evaluator and search algorithm. For the purposes of this research we examined the following evaluators that are used to measure the quality of the attribute or a set of them:

- Information Gain : (IG) measures the number of bits of information obtained for class prediction by knowing the presence or absence of a term in a document; or expected reduction in entropy caused by portioning the examples according to this attribute. Given a set of training attributes S, the information gain, $Gain(S, A)$, of an attribute A is defined as:

  $$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v}{|S|} Entropy(S_v) \qquad (6.5)$$

  where $Values(A)$ is the set of all possible values for attribute A and S_v is the subset of S for which attribute A has value v.

- Chi square test ($\chi^2 - test$) : Frequently used in statistics to test the independence of two events. In feature selection it is used to determine whether the occurrence of a selected attribute is independent of the class label. A high value of χ^2 indicates that the hypothesis that two events are independent is incorrect and thus attribute and class label are dependent and thus attribute should be selected. The χ^2 score can computed using the following formula:

$$\chi^2(D,t,c) = \sum_{e_t \in \{1,0\}} \sum_{e_c \in \{1,0\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}} \tag{6.6}$$

- Gain Ratio : Related to information gain, but instead it evaluates the worth of an attribute by measuring the ratio between the information gain and the intrinsic value $IV(A)$:

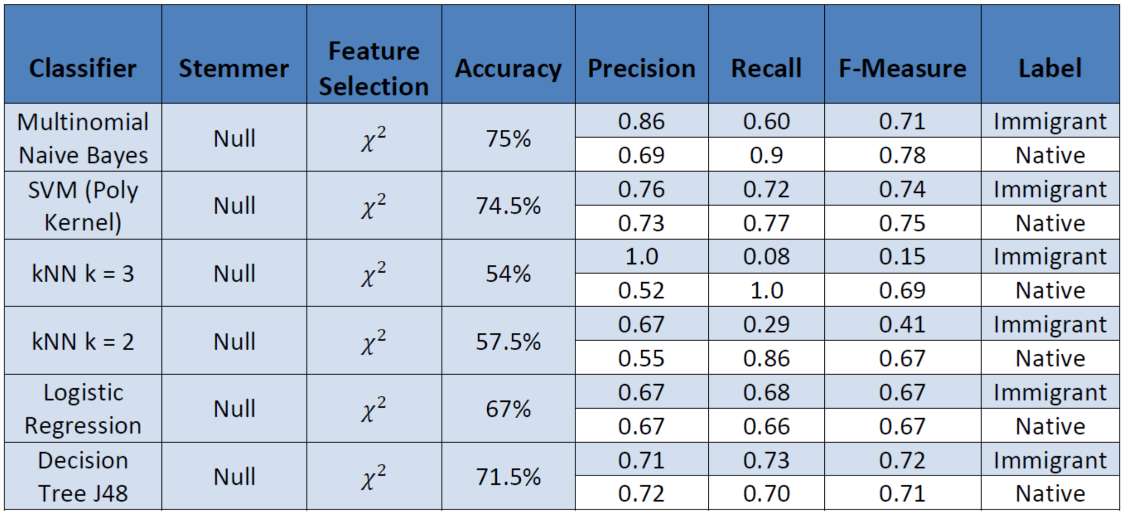
$$IV(A) = -\sum_{v \in Values(A)} \frac{|S_v|}{S} log(\frac{|S_v|}{S}) \tag{6.7}$$

$$GainRatio(Class, Attribute) = \frac{Gain(A)}{IV(A)} \tag{6.8}$$

The search algorithm Ranker, which ranks the attributes according to the individual evaluations, was then used to select the attributes. We compared the performance between the three feature selection evaluators above, and found that the subsets generated are largely similar and with no significant differences in accuracy we continued the experiments using chi square test.

6.5 Evaluation of classifiers

We trained 5 different classifiers, Multinomial Naive Bayes, SVM (with polynomial kernel), kNN, Logistic Regression and Decision Tree. We used 10 fold cross validation for each of the classifier to prevent overfitting, as it demonstrated the optimal results among classifiers when experimented with k values from 5 to 15 in k-fold cross validation.



Classifier	Stemmer	Feature Selection	Accuracy	Precision	Recall	F-Measure	Label
Multinomial Naive Bayes	Null	χ^2	75%	0.86	0.60	0.71	Immigrant
				0.69	0.9	0.78	Native
SVM (Poly Kernel)	Null	χ^2	74.5%	0.76	0.72	0.74	Immigrant
				0.73	0.77	0.75	Native
kNN k = 3	Null	χ^2	54%	1.0	0.08	0.15	Immigrant
				0.52	1.0	0.69	Native
kNN k = 2	Null	χ^2	57.5%	0.67	0.29	0.41	Immigrant
				0.55	0.86	0.67	Native
Logistic Regression	Null	χ^2	67%	0.67	0.68	0.67	Immigrant
				0.67	0.66	0.67	Native
Decision Tree J48	Null	χ^2	71.5%	0.71	0.73	0.72	Immigrant
				0.72	0.70	0.71	Native

Table 6.2: Model evaluation with attribute selection (Text–only)

6.5.1 Effect of feature selection

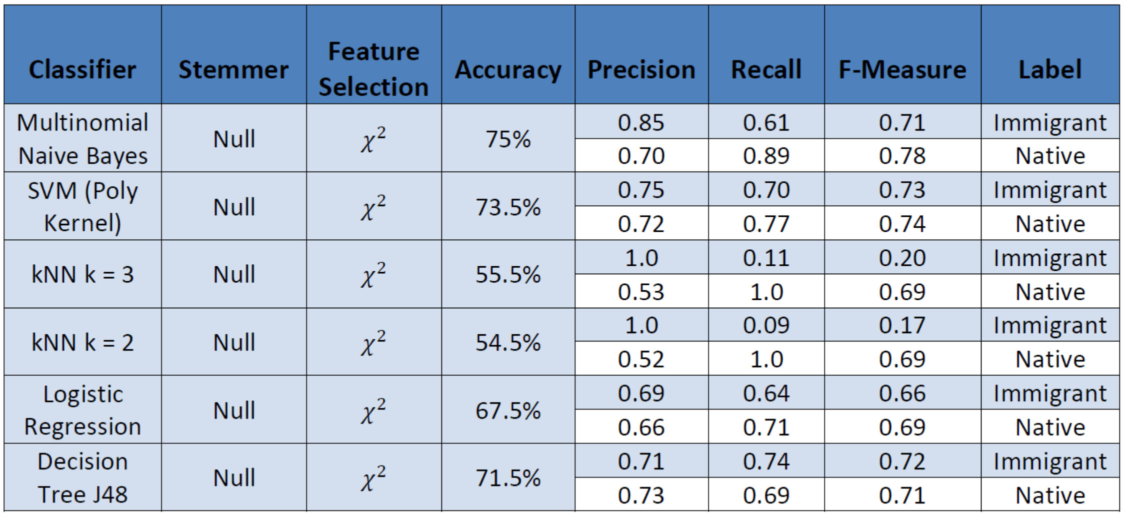
After applying feature selection, a great number of attributes were removed from the feature set which led to an accuracy boost of 4% for kNN, 3% for Multinomial Naive Bayes, and 1.5% for Decision Tree when compared to initial results without feature selection. The accuracy of SVM however was decreased by 3%. In case of logistic regression, Weka was not able to produce a model that uses all 140,000 attributes, however after attribute selection the number of attributes was reduced to approximately 4,500 which allowed to generate a model with an accuracy of 67%. We found that some models may benefit from removing over 97% of attributes, which can be attributed to removing noisy attributes, and we will continue performing feature selection for those in future experiments.

We can see in Table 6.2 that kNN model's accuracy improved by 4% after feature selection due to some of the noise reduced, however it still did not achieve accuracy over 57.5%. We experimented with different distance functions such as Manhattan, Euclidian and Chebyshev, however the accuracy did not improve. Low accuracy can be attributed to large number of features, given that kNN uses all features when it computes distances. To address this issue Han et al. [2001] proposed using weight adjust k-Nearest Neighbor Classification (WAKNN) algorithm that iteratively assigns weights and adjusts them based on the improvement in the objective function. The feature weights are then used in similarity measure calculation resulting in important features contributing more in the similarity measure.

While at the first glance removing attributes may seem counter intuitive, dimensionality reduction reduces the time and processing power required to build a model, allows to visualize the data in low dimensions, and decreases the chance of overfitting due to elimination of noisy data.

6.5.2 Effect of including numbers in dataset

We compared the performance of classifiers on two datasets with and without numbers in the textual content. We found that including numbers retained nearly identical accuracy for Multinomial Naive Bayes, Logistic Regression, Decision Tree, and reduced the accuracy of SVM model by 4% as shown in Tables 6.2 and 6.3. The size of feature selection set was increased by approximately 400, which did not affect the probabilities of previous features since Naive Bayes assumes that features are independent of one another. The performance of kNN was decreased by 3%, which is likely due to an increased number of attributes in feature set.



Classifier	Stemmer	Feature Selection	Accuracy	Precision	Recall	F-Measure	Label
Multinomial Naive Bayes	Null	χ^2	75%	0.85	0.61	0.71	Immigrant
				0.70	0.89	0.78	Native
SVM (Poly Kernel)	Null	χ^2	73.5%	0.75	0.70	0.73	Immigrant
				0.72	0.77	0.74	Native
kNN k = 3	Null	χ^2	55.5%	1.0	0.11	0.20	Immigrant
				0.53	1.0	0.69	Native
kNN k = 2	Null	χ^2	54.5%	1.0	0.09	0.17	Immigrant
				0.52	1.0	0.69	Native
Logistic Regression	Null	χ^2	67.5%	0.69	0.64	0.66	Immigrant
				0.66	0.71	0.69	Native
Decision Tree J48	Null	χ^2	71.5%	0.71	0.74	0.72	Immigrant
				0.73	0.69	0.71	Native

Table 6.3: Model evaluation with attribute selection (Text–with–numbers)

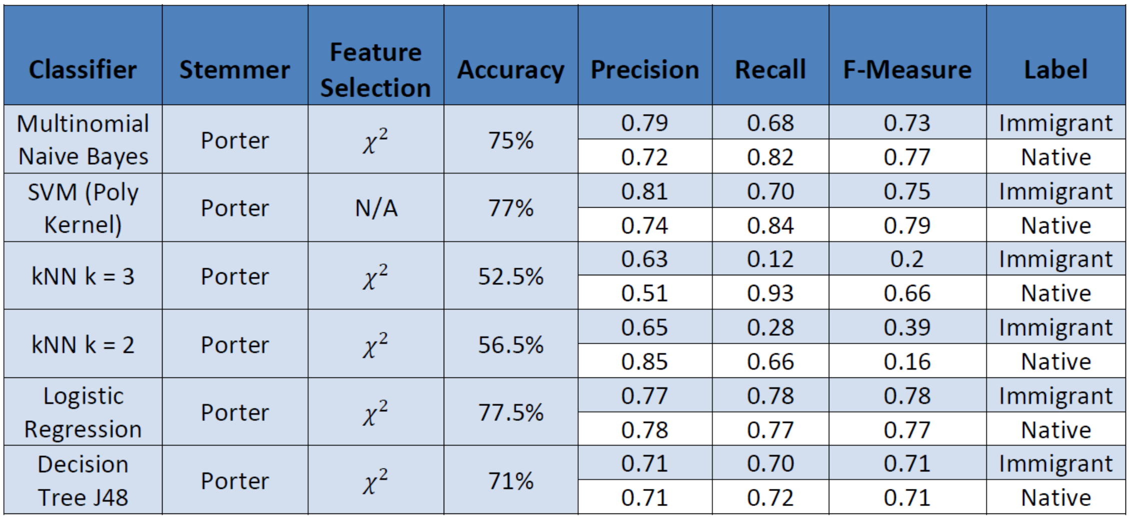
6.5.3 Effect of using stemmer

Stemming is a process of reducing a word to its stem or root form. It is frequently used to make the training data more dense, because previously different features such as ”rider”, ”riding” are reduced to their common root ”ride”. There are different stemming algorithms including KStem, Porter and others, which vary in how aggressively the words are stemmed. We used Porter stemming algorithm, written by Martin Porter, that reduced the original feature set (excluding numbers) from 140k to 116k, which can be considered an aggressive approach. Stemming may provide an advantage though by combining features with common roots and increase their importance/weight in a classifier. Comparison of Tables 6.2 vs. 6.4 and Tables 6.3 vs. 6.5 shows that overall stemming the dataset did not substantially improve the accuracy of models, and reduced accuracy of Decision Tree by 2%. However Logistic Regression model received a significant increase in accuracy improving from 67.5% to 78%. Given that Logistic Regression performance benefits from stemming, we used stemmed dataset when evaluating Logistic Regression models in the following experiments.

The current training set includes Latino users whose tweets frequently contained text in Spanish that was normalized into ASCII characters as described in Chapter 5.1.2. Thus there are tweets in foreign language which would occur frequently among "Immigrants" such as "los", "la" and "como". Figure 6.3 demonstrates a visualized decision tree produced from a model that contains multiple Spanish words at the root and other internal nodes. We compared the top features between the stemmed and non-stemmed dataset and noticed that there was an increased presence of Spanish words, however Porter stemmer was created based on English dictionary and thus would be inept at stemming non-English words. Thus for datasets with combined languages, it might be beneficial to use multiple stemming algorithms.

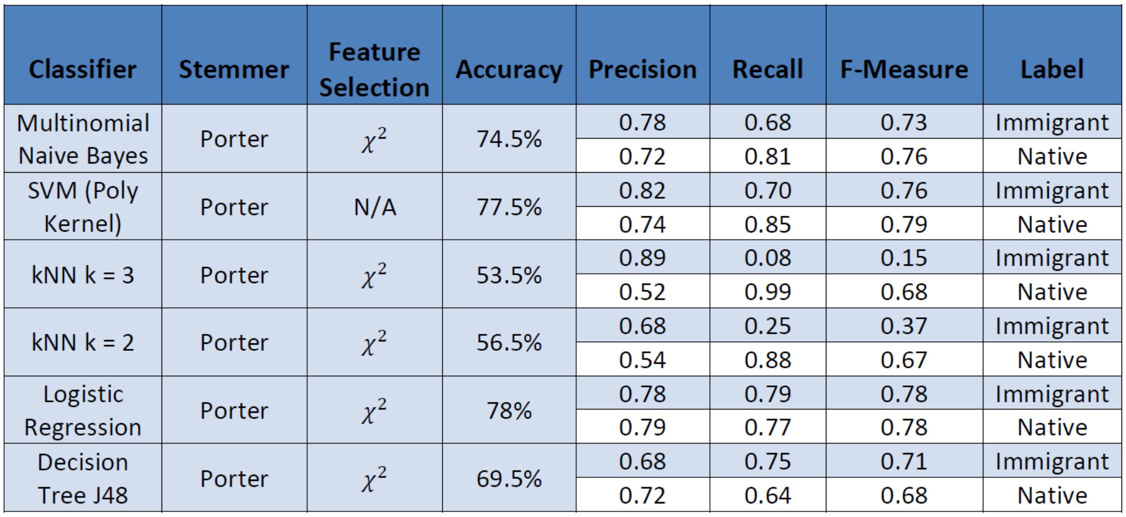
6.5.4 Effect of varying number of features

After evaluating the performance of algorithms using all features and with feature selection, we experimented with varying the number of features used, by selecting the top n features from the subset of features produced by feature selection. We varied the number of n from 50 to 2000 and noticed that for some algorithms it improved the accuracy while hurting the others. We previously mentioned that the model produced by kNN classifier resulted in low accuracy compared to others which



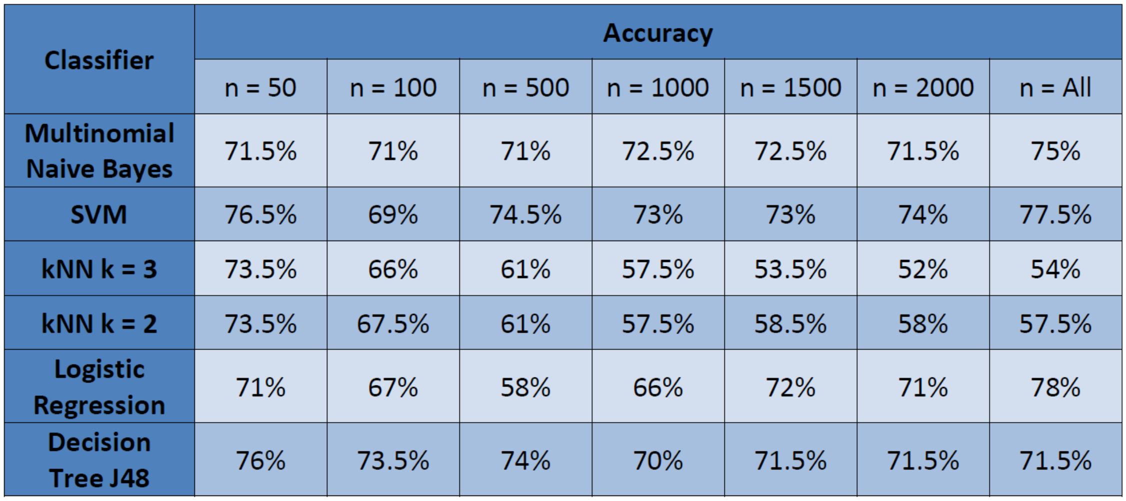
Classifier	Stemmer	Feature Selection	Accuracy	Precision	Recall	F-Measure	Label
Multinomial Naive Bayes	Porter	χ^2	75%	0.79	0.68	0.73	Immigrant
				0.72	0.82	0.77	Native
SVM (Poly Kernel)	Porter	N/A	77%	0.81	0.70	0.75	Immigrant
				0.74	0.84	0.79	Native
kNN k = 3	Porter	χ^2	52.5%	0.63	0.12	0.2	Immigrant
				0.51	0.93	0.66	Native
kNN k = 2	Porter	χ^2	56.5%	0.65	0.28	0.39	Immigrant
				0.85	0.66	0.16	Native
Logistic Regression	Porter	χ^2	77.5%	0.77	0.78	0.78	Immigrant
				0.78	0.77	0.77	Native
Decision Tree J48	Porter	χ^2	71%	0.71	0.70	0.71	Immigrant
				0.71	0.72	0.71	Native

Table 6.4: Model evaluation with attribute selection and stemming (Text–only)



Classifier	Stemmer	Feature Selection	Accuracy	Precision	Recall	F-Measure	Label
Multinomial Naive Bayes	Porter	χ^2	74.5%	0.78	0.68	0.73	Immigrant
				0.72	0.81	0.76	Native
SVM (Poly Kernel)	Porter	N/A	77.5%	0.82	0.70	0.76	Immigrant
				0.74	0.85	0.79	Native
kNN k = 3	Porter	χ^2	53.5%	0.89	0.08	0.15	Immigrant
				0.52	0.99	0.68	Native
kNN k = 2	Porter	χ^2	56.5%	0.68	0.25	0.37	Immigrant
				0.54	0.88	0.67	Native
Logistic Regression	Porter	χ^2	78%	0.78	0.79	0.78	Immigrant
				0.79	0.77	0.78	Native
Decision Tree J48	Porter	χ^2	69.5%	0.68	0.75	0.71	Immigrant
				0.72	0.64	0.68	Native

Table 6.5: Model evaluation with attribute selection and stemming (Text–with–numbers)



Classifier	Accuracy						
	n = 50	n = 100	n = 500	n = 1000	n = 1500	n = 2000	n = All
Multinomial Naive Bayes	71.5%	71%	71%	72.5%	72.5%	71.5%	75%
SVM	76.5%	69%	74.5%	73%	73%	74%	77.5%
kNN k = 3	73.5%	66%	61%	57.5%	53.5%	52%	54%
kNN k = 2	73.5%	67.5%	61%	57.5%	58.5%	58%	57.5%
Logistic Regression	71%	67%	58%	66%	72%	71%	78%
Decision Tree J48	76%	73.5%	74%	70%	71.5%	71.5%	71.5%

Table 6.6: Varying number of features used by classifier from $n = 50$ to $n = 2000$ and all features[2].

we attributed to large number of attributes. As we can see from Table 6.6, that the performance of kNN improved dramatically, achieving 19.5% accuracy improvement by using smaller subset of attributes. Figure 6.2 illustrates how the number of features used, affects the performance of kNN model, improving performance for both values of k as number of features is decreased. This experiment demonstrates that noise can significantly affect the performance of algorithms that are sensitive to noise data such as kNN. For other classifiers that are not as sensitive to high dimensional data such as Multinomial Naive Bayes and SVM, the performance dropped when using $n = 50$ but as we increased the number of features, so did the accuracy. For Multinomial Naive Bayes, reduced set of features affected the accuracy, because given a test instance with a word such as ”development” that is not present in feature set, the $P(”development”|c) = 0$, because it has not occurred in feature set, regardless of its frequency and importance in test set.

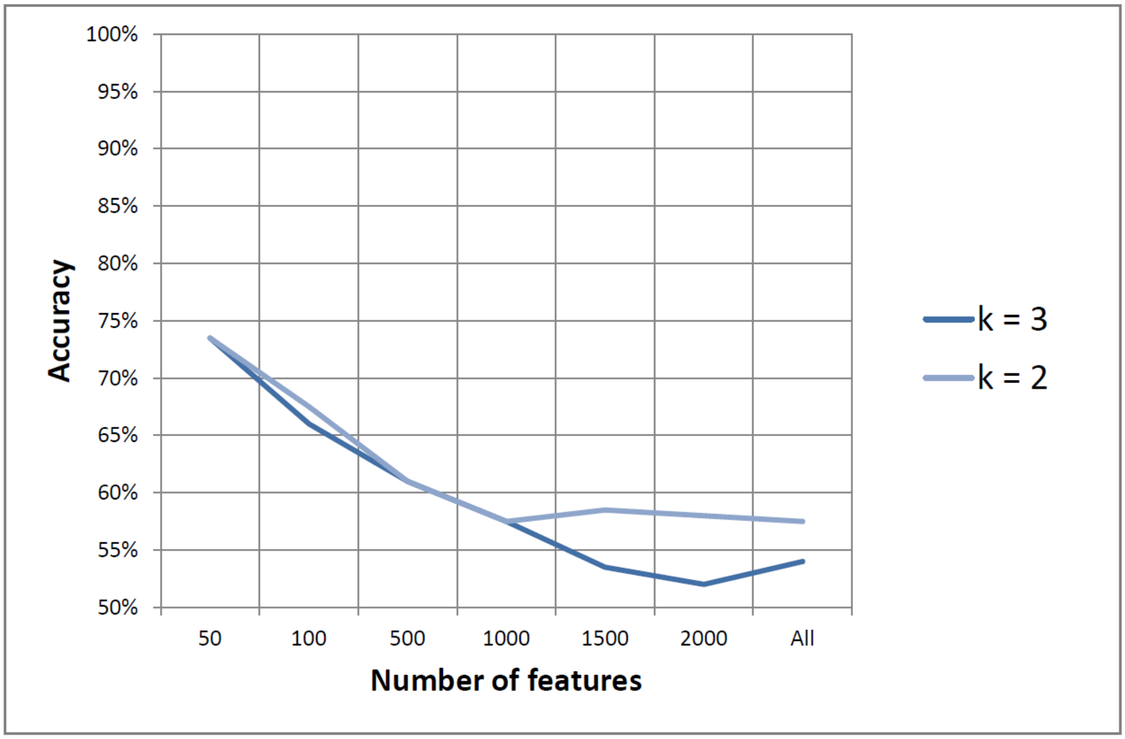


Figure 6.2: Effect of changing the number of features on the performance of kNN

6.5.5 Effect of including stopwords

As part of pre-processing step in Chapter 5.1.2, stopwords such as ”a”, ”the”, ”why”, and others were removed, since they can introduce noise and not contribute to the classifier. However given the nature of tweets, their occurrence may play an important role, especially if their use differs between immigrants and natives. We re-ran the experiments using the best performing configurations for each classifier to see if the performance improves. Table 6.7 demonstrates that stopwords may be

[2] $n = All$ means using all features produced by feature selection. For $n = All$, SVM uses all features of text-only dataset, since feature selection reduces its performance. Logistic Regression uses stemmed dataset that includes numbers. Remaining classifiers use text-only dataset with feature selection.

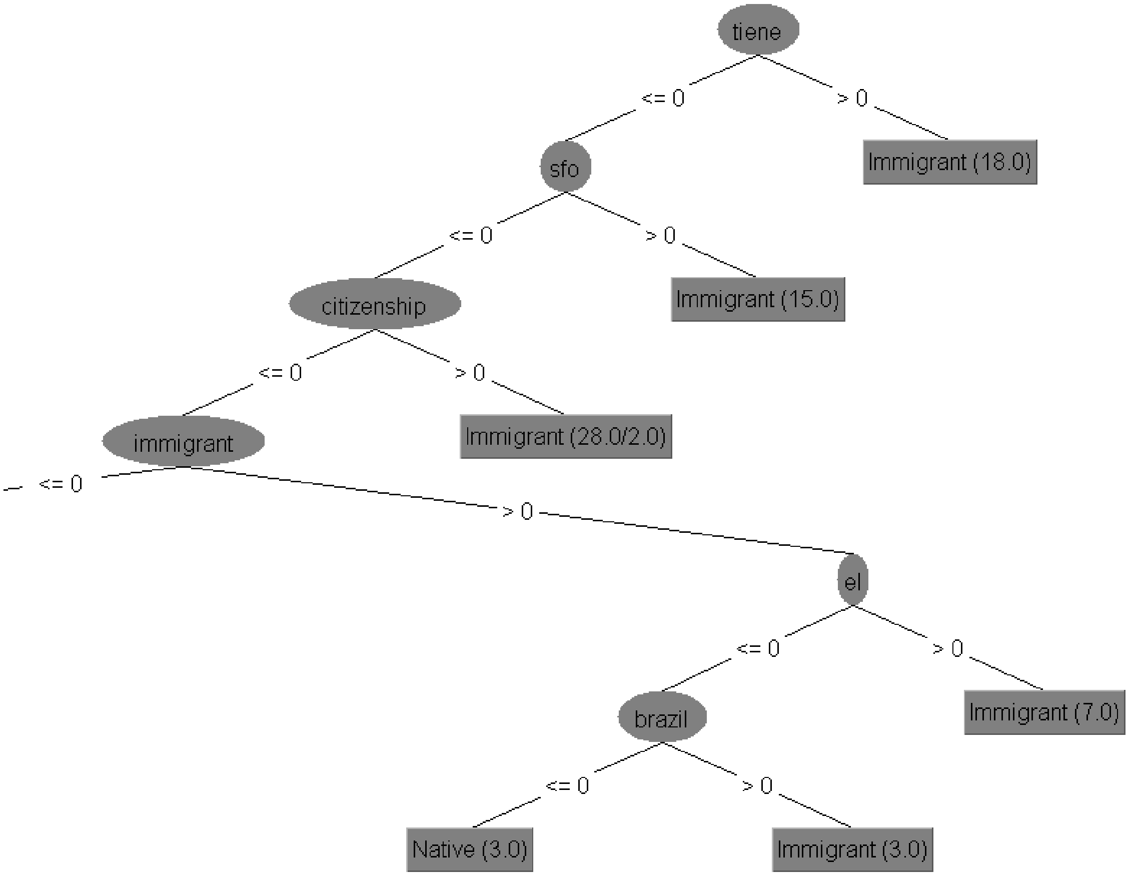
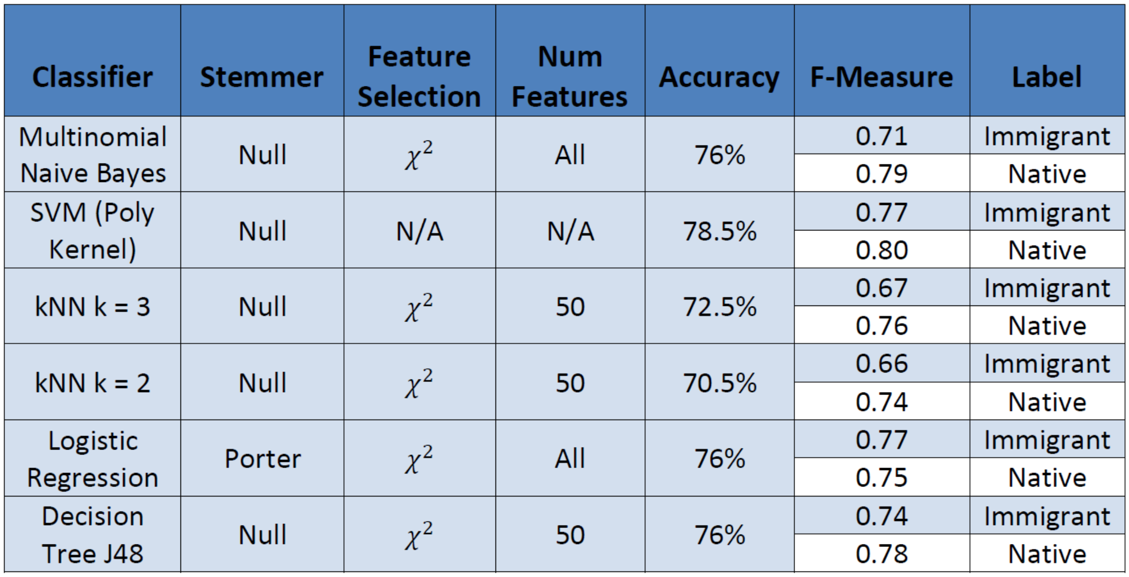


Figure 6.3: Partially visualized decision tree produced after applying feature selection and using top 50 features.



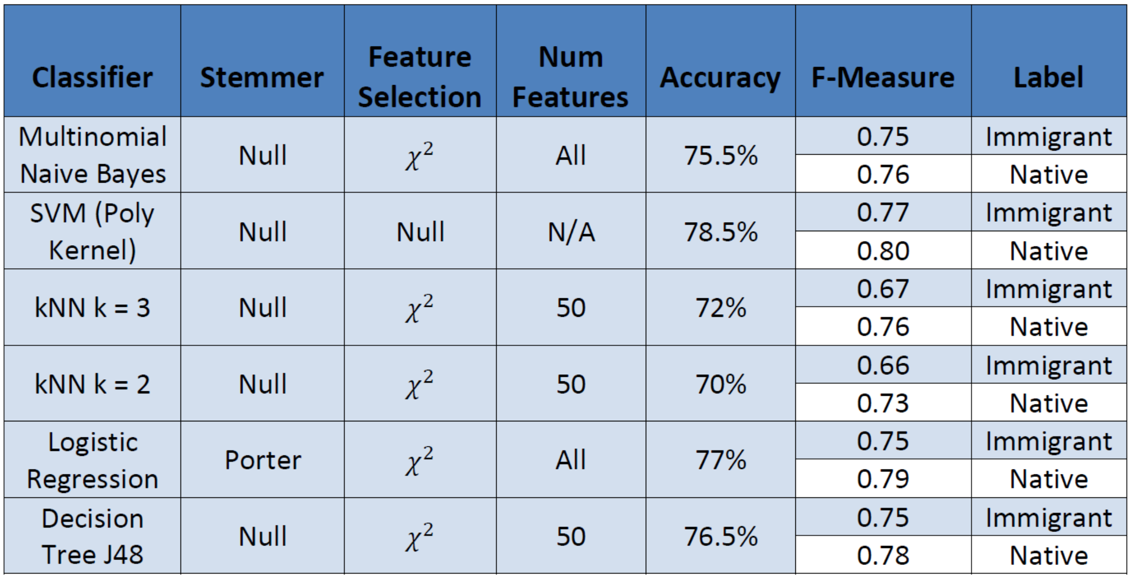
Classifier	Stemmer	Feature Selection	Num Features	Accuracy	F-Measure	Label
Multinomial Naive Bayes	Null	χ^2	All	76%	0.71	Immigrant
					0.79	Native
SVM (Poly Kernel)	Null	N/A	N/A	78.5%	0.77	Immigrant
					0.80	Native
kNN k = 3	Null	χ^2	50	72.5%	0.67	Immigrant
					0.76	Native
kNN k = 2	Null	χ^2	50	70.5%	0.66	Immigrant
					0.74	Native
Logistic Regression	Porter	χ^2	All	76%	0.77	Immigrant
					0.75	Native
Decision Tree J48	Null	χ^2	50	76%	0.74	Immigrant
					0.78	Native

Table 6.7: Effect of including stopwords in the dataset.

beneficial for some classifiers such as Multinomial Naive Bayes and SVM where accuracy was increased by 1%. The performance of Decision Trees remained unchanged since including stopwords did not change the nodes of the generated decision tree.

6.5.6 Effect of using TF–IDF model

Term Frequency – Inverse Document Frequency (tf–idf) is a way to determine the importance of words in a document (collection of tweets) based on how frequently words appears across multiple documents. tf–idf weight is commonly used in information retrieval and is very useful in document classification. The tf–idf weight of a term is the product of the term frequency, $tf(t, d)$, which is the raw frequency of a term t in a document d, and inverse document frequency, idf, which is how rare or common the term is across all the documents. The tf–idf can be



Classifier	Stemmer	Feature Selection	Num Features	Accuracy	F-Measure	Label
Multinomial Naive Bayes	Null	χ^2	All	75.5%	0.75	Immigrant
					0.76	Native
SVM (Poly Kernel)	Null	Null	N/A	78.5%	0.77	Immigrant
					0.80	Native
kNN k = 3	Null	χ^2	50	72%	0.67	Immigrant
					0.76	Native
kNN k = 2	Null	χ^2	50	70%	0.66	Immigrant
					0.73	Native
Logistic Regression	Porter	χ^2	All	77%	0.75	Immigrant
					0.79	Native
Decision Tree J48	Null	χ^2	50	76.5%	0.75	Immigrant
					0.78	Native

Table 6.8: Effect of including tf-idf model.

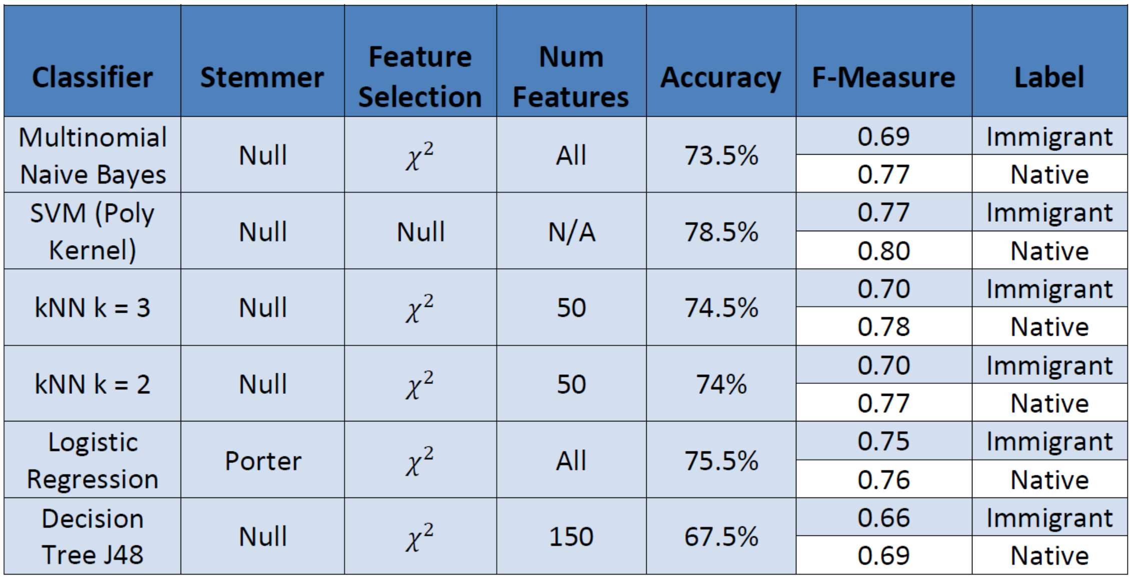
calculated using the following formula :

$$tfidf(t,d) = tf(t,d) \times \log \frac{N}{df_t} \tag{6.9}$$

where df_t is number of documents in the collection of documents of size N that contain term t. We compared the performance using tf and idf only and tf–idf with no significant differences in accuracy. The results shown in Table 6.8 illustrate that using tf–idf weights did not substantially improve the accuracy of models. The accuracy of kNN was reduced by 3.5% for $k = 2$, however it slightly improved accuracy of Decision Tree to 76.5%.

6.5.7 Effect of removing keywords

During the annotation of class labels for users described in Chapter 4.4, we used keywords and phrases in Table 4.1 to identify candidates for immigrant class.



Classifier	Stemmer	Feature Selection	Num Features	Accuracy	F-Measure	Label
Multinomial Naive Bayes	Null	χ^2	All	73.5%	0.69	Immigrant
					0.77	Native
SVM (Poly Kernel)	Null	Null	N/A	78.5%	0.77	Immigrant
					0.80	Native
kNN k = 3	Null	χ^2	50	74.5%	0.70	Immigrant
					0.78	Native
kNN k = 2	Null	χ^2	50	74%	0.70	Immigrant
					0.77	Native
Logistic Regression	Porter	χ^2	All	75.5%	0.75	Immigrant
					0.76	Native
Decision Tree J48	Null	χ^2	150	67.5%	0.66	Immigrant
					0.69	Native

Table 6.9: Effect of removing keywords that were used to identify candidates for immigrant users.

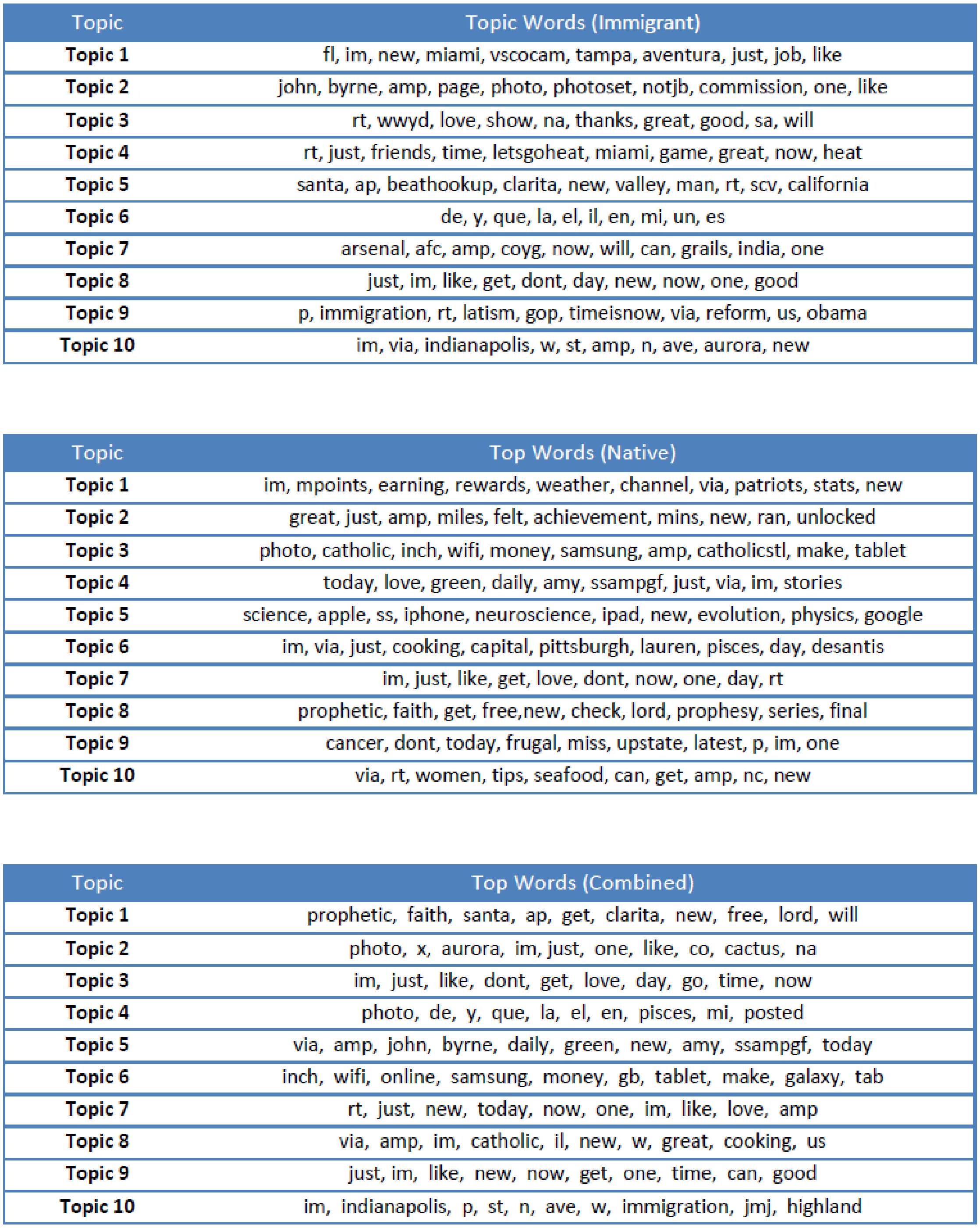
Since only users with tweets that contain those specific keywords were selected may add additional weight to them due to their required presence (at least one of keywords). Thus we excluded keywords listed in Table 4.1 from the training datasets and ran the experiments based on the best performing configuration for each classifier. From Table 6.9 we can see that overall removing keywords did not significantly reduce the accuracy of classifiers. The accuracy of Multinomial Naive Bayes and Logistic Regression was reduced by 1.5% and 2.5% respectively. The performance of Decision Tree reduced significantly because keywords such as "immigrant" and "citizenship" were removed and as we can see from Figure 6.3, these attributes were important nodes used for classification. However the accuracy of kNN improved by 1%, due to removing a certain number of attributes.

6.6 Topic Modeling

Latent Dirichlet Allocation can be used to generate topics based on word frequency in a document. We use an implementation of LDA available in the Gensim [33] toolkit to perform model estimation from a training corpus. We begin by tokenizing the document by matching any word character excluding stop words and building a dictionary. We did not perform word stemming due to an increased presence of foreign words. Next we converted the dictionary into a bag of words converting the corpus into a list of vectors with series of tuples (term ID, term frequency) such as

$[(0, 1), (1, 2), (2, 2), (3, 1), (4, 3), (5, 0)]$

Finally we apply the LDA model, choosing 10 topics for each class (Immigrant, Native) given that the tweet dataset is large. Additionally we build another LDA model for combined tweet dataset to see how common topics change in a mixed dataset. We compared the difference between datasets with and without numbers which produced similar topics and suggests that having numbers did not significantly affect the topic weight. For the two class labels, we can see in Table 6.10 that a few topics overlap between the classes such as "im" which stands for "I'm" which frequently occurs in Twitter since posts are coming directly from user's perspective. Additionally we previously mentioned in Chapter 5.3 the issue with presence of auto generated posts such as "vscocam" in Table 6.10, which are tweeted on behalf of a user through apps/services and their occurrence was more frequent in user tweets than previously anticipated.



Topic	Topic Words (Immigrant)
Topic 1	fl, im, new, miami, vscocam, tampa, aventura, just, job, like
Topic 2	john, byrne, amp, page, photo, photoset, notjb, commission, one, like
Topic 3	rt, wwyd, love, show, na, thanks, great, good, sa, will
Topic 4	rt, just, friends, time, letsgoheat, miami, game, great, now, heat
Topic 5	santa, ap, beathookup, clarita, new, valley, man, rt, scv, california
Topic 6	de, y, que, la, el, il, en, mi, un, es
Topic 7	arsenal, afc, amp, coyg, now, will, can, grails, india, one
Topic 8	just, im, like, get, dont, day, new, now, one, good
Topic 9	p, immigration, rt, latism, gop, timeisnow, via, reform, us, obama
Topic 10	im, via, indianapolis, w, st, amp, n, ave, aurora, new

Topic	Top Words (Native)
Topic 1	im, mpoints, earning, rewards, weather, channel, via, patriots, stats, new
Topic 2	great, just, amp, miles, felt, achievement, mins, new, ran, unlocked
Topic 3	photo, catholic, inch, wifi, money, samsung, amp, catholicstl, make, tablet
Topic 4	today, love, green, daily, amy, ssampgf, just, via, im, stories
Topic 5	science, apple, ss, iphone, neuroscience, ipad, new, evolution, physics, google
Topic 6	im, via, just, cooking, capital, pittsburgh, lauren, pisces, day, desantis
Topic 7	im, just, like, get, love, dont, now, one, day, rt
Topic 8	prophetic, faith, get, free,new, check, lord, prophesy, series, final
Topic 9	cancer, dont, today, frugal, miss, upstate, latest, p, im, one
Topic 10	via, rt, women, tips, seafood, can, get, amp, nc, new

Topic	Top Words (Combined)
Topic 1	prophetic, faith, santa, ap, get, clarita, new, free, lord, will
Topic 2	photo, x, aurora, im, just, one, like, co, cactus, na
Topic 3	im, just, like, dont, get, love, day, go, time, now
Topic 4	photo, de, y, que, la, el, en, pisces, mi, posted
Topic 5	via, amp, john, byrne, daily, green, new, amy, ssampgf, today
Topic 6	inch, wifi, online, samsung, money, gb, tablet, make, galaxy, tab
Topic 7	rt, just, new, today, now, one, im, like, love, amp
Topic 8	via, amp, im, catholic, il, new, w, great, cooking, us
Topic 9	just, im, like, new, now, get, one, time, can, good
Topic 10	im, indianapolis, p, st, n, ave, w, immigration, jmj, highland

Table 6.10: Top 10 topics with 10 words per Topic for "Immigrant", "Native" and combined users

Chapter 7

Conclusion

7.1 Conclusion

This research explored identifying latent demographic attributes on social media platforms focusing specifically on Twitter users and their immigration status. We designed a system that collected a large scale dataset which included user profile, and tweets for users that reside in the United States. The dataset we collected contained approximately 50 thousand users which contained Asian, Latino and Caucasian users. After manually inspecting the users, we selected a subset of users that contained two categories : "Immigrant" and "Native". The initial set of features contained a large set of attributes (words) that was reduced by 97.1% using feature selection. After applying multiple machine learning classifiers including Multinomial Naive Bayes, Perceptron, Logistic Regression, SVM and Decision Trees, we built a model that was able to achieve 78.5% accuracy using Support Vector Machine and 78% accuracy using Logistic Regression, which was verified by 10–fold cross validation. We found that the accuracy of classifiers can be increased through an application of feature selection, stemming, or increasing feature set such as including stopwords. Depending on the classifier one must take into an account the specifics of the algorithms such as sensitivity to noise data and memory requirements and apply correct performance improving techniques.

7.2 Future Work

Throughout the work, a number of improvements were implemented such as moving data files to a database or removing @user mentions altogether. However a certain number of improvements can be considered for future work. In Chapter 2 we reviewed the work by Zamal et al. [2012] that used neighboring accounts to boost the accuracy of their model. We could additionally consider sampling followers and/or friends : as we examined multiple Twitter users we found that people tend to stay in touch (follow) people who are relatives, friends, or like-minded individuals with similar interests or backgrounds. Thus by including followers and friends it is more likely to include users that match the criteria for the training set (in our case fellow immigrants).

After examining individual tweets, we noticed that immigrant users tend to frequently tweet in other languages besides English, which are most likely their own native languages. During the processing of tweet dataset in Chapter 5.1.2 Unicode characters were removed from the tweets. Hence most of the tweets are English words that are commonly used by both immigrants and native citizens, which results in overlap of features and could further complicate the model and/or reduce its accuracy. Therefore it would be beneficial to at least maintain an indicator that a foreign word was used, as an additional feature which could improve the performance of a classifier.

Additionally building a richer set of social network features can be valuable which was demonstrated by Pennacchiotti and Popescu [2011] and Zamal et al.

[2012]. Even though Twitter API does not provide demographic information about a user, we could include the following profile features : total number of tweets, number of followers, number of friends, user's first and last name, description, and location.

The process of annotating training data involved manual inspection of tweets and descriptions that matched certain keywords. User descriptions however frequently contain links to personal blogs and/or other social media profiles that might contain more demographic information about a user's origin. For example, Facebook allows users to specify "Places You've Lived" including a separate sub-option for "Hometown" that can be collected. Additionally the list of keywords and key phrases in Table 4.1 that were used to identify immigrant candidates can be expanded by including foreign slang words that would yield more matching results for training data.

For the purpose of this research we focused on two race groups, Asian and Latino, that made up the majority of immigrant population in the United States. However, by including more race groups such as "Black or African American", "Caucasian/White", "American Indian and Alaska Native", and "Native Hawaiian and Other Pacific Islander" we would encompass a far greater user base on Twitter, and increase the size of training data that could lead to building a more accurate and generalized model.

This work can also be applied to other social media websites, specifically Facebook that provides information about user's origin. Facebook's Graph Search API allows to build natural language queries to easily find foreign born living in the US users such as ”people from Colombia who live in United States” However based on preliminary analysis, user’s profile do not contain as many self authored posts as on Twitter, which might make it challenging to build a model.

User’s tweets frequently contain URLs and @user mentions which may contribute to the classification model. During the data processing stage URLs and user mentions (@JaneDoe) were stripped from the training dataset completely. Instead of removing them altogether, we could replace URL address with a ”URL” signifying that a URL was used and replace ”@JaneDoe” mentions with a standardized ”at_user” word which signifies that a user mention was used.

The Twitter API limits slow down the data collection process, however by running an increased number of applications, data could be collected at a faster rate. We could consider creating multiple number of applications that would run in parallel (more than 10) in order to increase the data collection rate.

7.3 Use in other applications/fields

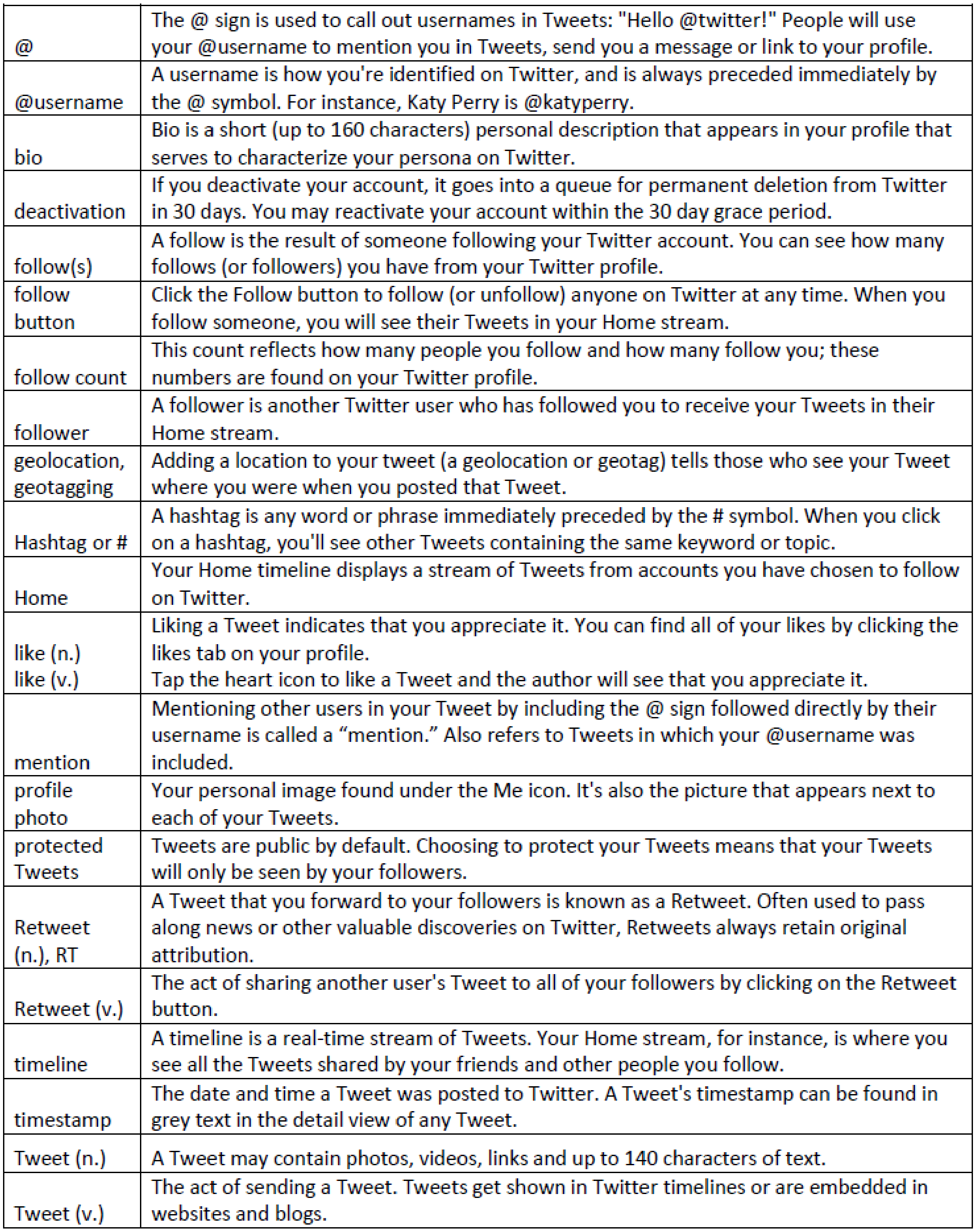
Latent attribute extraction is not limited to demographic attributes such as user’s ethnicity, age, or gender. Prior work has focused on determining user’s age group, political inclination and gender. Text classification allows to uncover user’s preferences such as devotion to certain products/brands or emerging trends and can be applied in the following fields:

- Marketing and personalization : By predicting latent attributes of users, it allows to present products and/or services to specific target audience that would have more interest in it than others which would reduce operating costs and increase revenue.

- Legal Investigation / Legal : Sentiment analysis could be used to determine public opinion on certain topic or used to infer emerging conflicts or epidemics and is explored by Colbaugh and Glass [2011].
- Improve user experience on Twitter or other participating website by providing content related to user's interests, culture and/or language. In addition to public surveys, population data can be sampled from social media that can act as supplemental data for under–reported areas.

Appendix A

Twitter Terminology



@	The @ sign is used to call out usernames in Tweets: "Hello @twitter!" People will use your @username to mention you in Tweets, send you a message or link to your profile.
@username	A username is how you're identified on Twitter, and is always preceded immediately by the @ symbol. For instance, Katy Perry is @katyperry.
bio	Bio is a short (up to 160 characters) personal description that appears in your profile that serves to characterize your persona on Twitter.
deactivation	If you deactivate your account, it goes into a queue for permanent deletion from Twitter in 30 days. You may reactivate your account within the 30 day grace period.
follow(s)	A follow is the result of someone following your Twitter account. You can see how many follows (or followers) you have from your Twitter profile.
follow button	Click the Follow button to follow (or unfollow) anyone on Twitter at any time. When you follow someone, you will see their Tweets in your Home stream.
follow count	This count reflects how many people you follow and how many follow you; these numbers are found on your Twitter profile.
follower	A follower is another Twitter user who has followed you to receive your Tweets in their Home stream.
geolocation, geotagging	Adding a location to your tweet (a geolocation or geotag) tells those who see your Tweet where you were when you posted that Tweet.
Hashtag or #	A hashtag is any word or phrase immediately preceded by the # symbol. When you click on a hashtag, you'll see other Tweets containing the same keyword or topic.
Home	Your Home timeline displays a stream of Tweets from accounts you have chosen to follow on Twitter.
like (n.) like (v.)	Liking a Tweet indicates that you appreciate it. You can find all of your likes by clicking the likes tab on your profile. Tap the heart icon to like a Tweet and the author will see that you appreciate it.
mention	Mentioning other users in your Tweet by including the @ sign followed directly by their username is called a "mention." Also refers to Tweets in which your @username was included.
profile photo	Your personal image found under the Me icon. It's also the picture that appears next to each of your Tweets.
protected Tweets	Tweets are public by default. Choosing to protect your Tweets means that your Tweets will only be seen by your followers.
Retweet (n.), RT	A Tweet that you forward to your followers is known as a Retweet. Often used to pass along news or other valuable discoveries on Twitter, Retweets always retain original attribution.
Retweet (v.)	The act of sharing another user's Tweet to all of your followers by clicking on the Retweet button.
timeline	A timeline is a real-time stream of Tweets. Your Home stream, for instance, is where you see all the Tweets shared by your friends and other people you follow.
timestamp	The date and time a Tweet was posted to Twitter. A Tweet's timestamp can be found in grey text in the detail view of any Tweet.
Tweet (n.)	A Tweet may contain photos, videos, links and up to 140 characters of text.
Tweet (v.)	The act of sending a Tweet. Tweets get shown in Twitter timelines or are embedded in websites and blogs.

Table A.1: Twitter Glossary

Bibliography

[1] Alexa Internet Inc. (2016a). Alexa - Top Sites by Category: Computers/Internet/On the Web/Online Communities/Social Networking.

[2] Alexa Internet Inc. (2016b). Alexa Top 500 Global Sites.

[3] Baffour, B., King, T., and Valente, P. (2013). The Modern Census: Evolution, Examples and Evaluation. *International Statistical Review*, 81(3):407–425.

[4] Blei, D. M. (2012). Probabilistic Topic Models. *Commun. ACM*, 55(4):77–84.

[5] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3:993–1022.

[6] Burger, J. D., Henderson, J., Kim, G., and Zarrella, G. (2011). Discriminating Gender on Twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1301–1309, Stroudsburg, PA, USA. Association for Computational Linguistics.

[7] Chang, J., Rosenn, I., Backstrom, L., and Marlow, C. (2010). ePluribus: Ethnicity on Social Networks. In *Proceedings of the Fourth International Conference on Weblogs and Social Media (ICWSM-10)*, Washington DC. AAAI Press.

[8] Cheng, Z., Caverlee, J., and Lee, K. (2010). You Are Where You Tweet: A Content-based Approach to Geo-locating Twitter Users. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 759–768, New York, NY, USA. ACM.

[9] Colbaugh, R. and Glass, K. (2011). Agile Sentiment Analysis of Social Media Content for Security Informatics Applications. In *EISIC*, pages 327–331. IEEE Computer Soecity.

[10] Domingos, P. and Pazzani, M. (1997). On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. *Machine Learning*, 29(2-3):103–130.

[11] Frnkranz, J. (2009). Decision-Tree Learning.

[12] Ghazaleh Beigi, X. B. H. (2015). An Overview of Sentiment Analysis in Social Media and Its Applications in Disaster Relief.

[13] Han, E.-H. S., Karypis, G., and Kumar, V. (2001). *Advances in Knowledge Discovery and Data Mining: 5th Pacific-Asia Conference, PAKDD 2001 Hong Kong, China, April 16–18, 2001 Proceedings*, chapter Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification, pages 53–65. Springer Berlin Heidelberg, Berlin, Heidelberg.

[14] Hargittai, E. and Litt, E. (2011). The tweet smell of celebrity success: Explaining variation in Twitter adoption among a diverse group of young adults. *New Media & Society*, page 1461444811405805.

[15] Jones, R., Kumar, R., Pang, B., and Tomkins, A. (2007). "I Know What You Did Last Summer": Query Logs and User Privacy. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, pages 909–914, New York, NY, USA. ACM.

[16] Liu, W. and Ruths, D. (2013). Whats in a Name? Using First Names as Features for Gender Inference in Twitter. In *2013 AAAI Spring Symposium Series.*

[17] Mander, J. (2015). Daily time spent on social networks rises to 1.72 hours.

[18] Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval.* Cambridge University Press, New York, NY, USA.

[19] Migration Policy Institute (2014). State Demographics Data - US.

[20] Migration Policy Institute (2015). State Demographics Data - US.

[21] Mislove, A., Jrgensen, S. L., Ahn, Y.-Y., Onnela, J.-P., and Rosenquist, J. N. (2011). Understanding the Demographics of Twitter Users. *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, pages 554–557.

[22] Mitchell, T. M. (1997). *Machine Learning.* McGraw-Hill, Inc., New York, NY, USA, 1 edition.

[23] Mohammady, E. and Culotta, A. (2014). Using County Demographics to Infer Attributes of Twitter Users. In *Proceedings of the Joint Workshop on Social Dynamics and Personal Attributes in Social Media*, pages 7–16, Baltimore, Maryland. Association for Computational Linguistics.

[24] Murthy, D., Gross, A., and Pensavalle, A. (2016). Urban Social Media Demographics: An Exploration of Twitter Use in Major American Cities. *Journal of Computer-Mediated Communication*, 21(1):33–49.

[25] OpenCV (2014). Svm optimal hyperplane.

[26] Paynter, G. (2008). Attribute-Relation File Format (ARFF).

[27] Pennacchiotti, M. and Popescu, A.-M. (2011). A Machine Learning Approach to Twitter User Classification. In *Fifth International AAAI Conference on Weblogs and Social Media.*

[28] Perseus Documentation (2015a). k nearest neighbors.

[29] Perseus Documentation (2015b). Svm high dimension.

[30] Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, 1(1):81–106.

[31] Rao, D., Paul, M. J., Fink, C., Yarowsky, D., Oates, T., and Coppersmith, G. (2011). Hierarchical Bayesian Models for Latent Attribute Detection in Social Media. In Adamic, L. A., Baeza-Yates, R. A., and Counts, S., editors, *ICWSM*. The AAAI Press.

[32] Rao, D., Yarowsky, D., Shreevats, A., and Gupta, M. (2010). Classifying Latent User Attributes in Twitter. In *Proceedings of the 2Nd International Workshop on Search and Mining User-generated Contents*, SMUC '10, pages 37–44, New York, NY, USA. ACM.

[33] Rehurek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *IN PROCEEDINGS OF THE LREC 2010 WORKSHOP ON NEW CHALLENGES FOR NLP FRAMEWORKS*, pages 45–50.

[34] Rennie, J. D. M., Shih, L., Teevan, J., and Karger, D. R. (2003). Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In *In Proceedings of the Twentieth International Conference on Machine Learning*, pages 616–623.

[35] Sacca, C. (2015). *What Twitter Can Be.*

[36] Sakaki, T., Okazaki, M., and Matsuo, Y. (2010). Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 851–860, New York, NY, USA. ACM.

[37] Statista Inc. (2015). Number of worldwide social network users 2010-2019 Statistic.

[38] Statista Inc. (2016a). Monthly active U.S. Twitter users 2015 Statistic.

[39] Statista Inc. (2016b). U.S. population with a social network profile 2016 Statistic.

[40] Truong, B., Caragea, C., Squicciarini, A., and Tapia, A. H. (2014). Identifying valuable information from twitter during natural disasters. *Proceedings of the American Society for Information Science and Technology*, 51(1):1–4.

[41] Twitter Inc. (2015a). About public and protected Tweets.

[42] Twitter Inc. (2015b). API Rate Limits.

[43] Twitter Inc. (2015c). The Twitter glossary.

[44] US Census Bureau (2015). Population estimates, July 1, 2015, (V2015).

[45] US Census Bureau (2016). Population Clock.

[46] Wikipedia (2015). Decision tree with numeric and nominal attributes.

[47] Word, D. L., Coleman, C. D., Nunziata, R., and Kominski, R. (n.d.). Demographic Aspects of Surnames from Census 2000.

[48] Yang, Y. and Pedersen, J. O. (1997). A Comparative Study on Feature Selection in Text Categorization. pages 412–420. Morgan Kaufmann Publishers.

[49] Young-Woo, S. (n.d.). InfoGain.

[50] Zamal, F. A., Liu, W., and Ruths, D. (2012). Homophily and Latent Attribute Inference: Inferring Latent Attributes of Twitter Users from Neighbors. In *Sixth International AAAI Conference on Weblogs and Social Media*.

[51] Zong, J. and Batalova, J. (2016). Frequently Requested Statistics on Immigrants and Immigration in the United States.