An exploration of gender indicative feature data on Twitter

ML1819 Research Assignment 2

*Team 41* - *Task 107 - How well can the gender of Twitter users be predicted?*

*Word count - TODO*

[Github Source Code](https://github.com/beakeyd/ML1819--task-107--team-41)

[Github Contributors](https://github.com/beakeyd/ML1819--task-107--team-41/graphs/contributors)

.

TODO Update Contribution Screenshots

|  |  |  |  |
| --- | --- | --- | --- |
| Samuel McCann Trinity College Dublin Ireland #15318105 mccanns3@tcd.ie | | Sean McDonagh Trinity College Dublin Ireland #15319517 semcdona@tcd.ie | David Beakey Trinity College Dublin Ireland #15335531 d[beakey@tcd.ie](mailto:beakeyd@tcd.ie) |
|  |

TODO – UPDATE CREDITS - We worked on the project everyday over the course of a week together. Everything that we have submitted, in terms of code, was worked on together.

# 1 Introduction

Twitter has become a key platform for companies and public figures to engage with their consumer base. Understanding your demographic of users has become a priority of businesses to better interact with them.

Privacy has become more of a concern nowadays on the internet. It is harder for companies to scrape relevant data on their demographics from publicly available information on twitter profiles, such as gender.

We wish to discover how well we can predict the gender of twitter users based off their publicly available profile information.

# 2 Related Work

John D.Burger et al [1] managed to classify a user's gender with an accuracy rate of 92%. They used a Winnow2 linear classifier on text features such as Screen Name, Profile Description and Tweets. This showed success, so we shall try build off those features. To build their ground truth they followed the blogs of users and obtained their gender there. TODO Possibly delete this line, and just mention how the ground truth for our data was obtained.

Clay Fink [2] also examined this problem in detail. They determined gender by searching twitter users Facebook pages todo same as above. They achieved an accuracy rate of 80.6% with a combined feature vector of unigrams, linguistic inquiry word count, and hashtags and using a SVMLight linear classifier.

# 3 METHODOLOGIES

## 3.1 Dataset

We built and gathered our own dataset. We started with a data set consisting of User Id’s and an associated gender for that Id from an existing research paper on gender classification [3] Confirm this paper is correct. The ground truth for this dataset was built by following blog and external social media links associated with their twitter profiles to discover the real gender. We wrote a script to scrape publicly available data from the Twitter API for each user Id to give us an initial dataset of 12000 instances.

## 3.2 Pre-Processing

We first cut out features that we deemed to intuitively be irrelevant to our classification problem. We deleted features that were set to default values for most of the dataset. We also deleted features that had too many instances of missing data to interpolate.

After pruning some of these initial features, we then balanced the data set to a 50/50 split of male and female. We didn’t want the classifier just classifying always Male to achieve a higher accuracy.

We were left with a dataset of 6000 users.

Categorical features were given numerical representations. Male for example was mapped to 0, Female mapped to 0.

Numerical features were given a min-max normalization between 0 and 1. We normalized this ourselves rather than use a library call.

Users with Non-english tweets were not considered. We didn’t think there were enough instances of non-english users to classify correctly. Unicode characters inside of text features were also removed. TODO more info on text prep?

## 3.3 Algorithms

All algorithms and machine learning processes were done using the Scikit-Learn machine learning library in Python.

We compare three models.

LinearSVC classifier trained on both numerical and textual features.

Naïve Bayes classifier trained only on textual features.

KNeighbours model trained on numerical features.

## 3.4 HyperParameter Selection

We ran our models through a various selection of hyperparameters and feature sets to obtain optimal performance of each model.

TODO visualizations of heatmaps for models

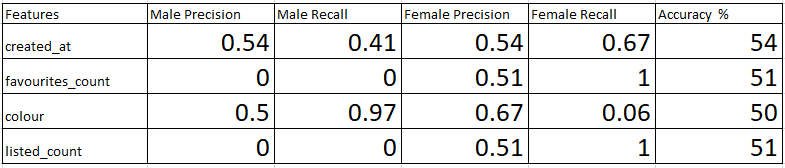
## 3.5 Evaluation

We evaluated our results of each model through a K Fold cross validation method. We’ve used 4 folds, as 10 folds proved to be too time consuming in training each model under different hyperparameters with varying feature sets.

TODO Visualizations?

# 4 Results and Discussions

Table 1 - Numerical Features



You can see from the overall accuracy in Table 1, as well as the Precision and Recall numbers, that the prediction rates for the models which used individual numerical features alone are poor. You can see for example, the model Figure 3 which has favourites\_count as its singular feature always predicts Female. This predication is no better than random chance. On their own, these individual features are not good indicators of gender.

Figure 2 - Twitter Account Creation Date

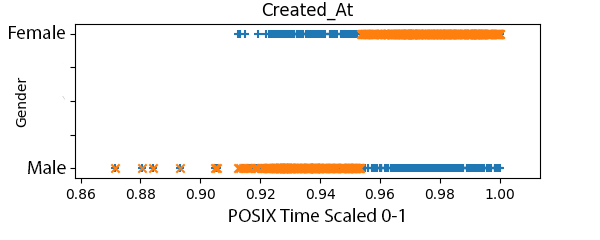
A plot of the Created\_At data Figure 2 shows where the classifier is defining the split in the data. Blue crosses on the graph represent the test data used in the model, while orange X’s represents the predictions on the test data. X Values are POSIX, time scaled between 0 and 1. Accounts created before 2017 are predicted to be Male, while accounts created after are predicted to be female. This confirms that this model is doing more than just predicting everything to be all Female like in the favourites\_count Figure 3 singular feature model, which is below.

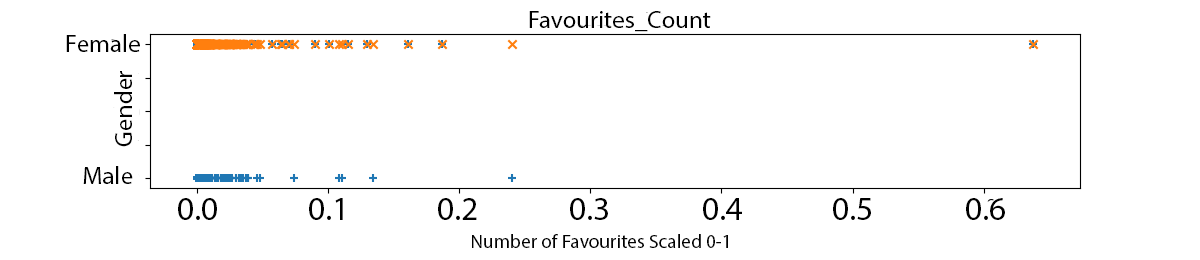
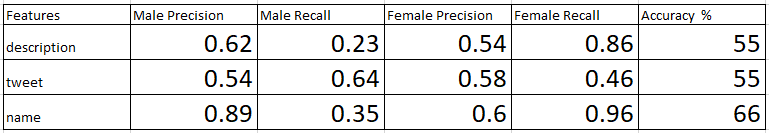
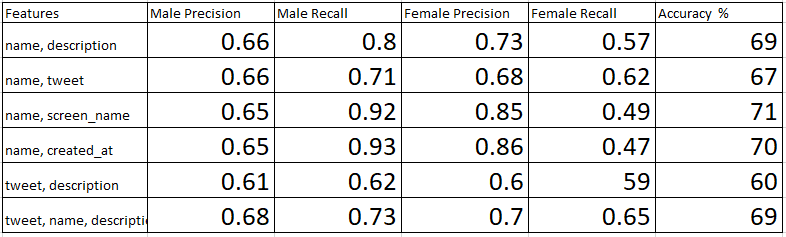
Figure 3 - Number of Favourites

Table 2 - Text Features



The models that uses text-based features Table 2 show slightly better results. Name being the strongest identifier so far, with an accuracy of 66%.

Table 3 - Combined Features



However, as we combine the features used in the singular feature based models, the prediction accuracy rises, as high as 70% Table 3. Following John D. Burger’s et al [1] success with using the name, tweet, description and screen\_name, we chose to combine this type of data in various combinations to feed as features into our models. Those combinations that returned high accuracy are shown in the above table Table 3

We believe that these results aren’t strong enough to used as a classifier in a commercial or professional environment. Our best model, which used name and screen\_name data Table 3, was able to reach an accuracy of 71%, which is less than the 92% reported by John D. Burger et al [1] and the 80% reported by Clay Fink et al [2].

Hence, name is the most indicative of gender, and when combining feature data into a single model, name and screen\_name are the most indicative.

# 5 Limitations and Outlook

The main limitation that we encountered, was that a major part of our data was not linearly separable (particularly the numeric data). A polynomial kernel was used to cater for this fact, but still the predication accuracy remained low.

Text classification showed better results over numerical features. An exploration into classifying users based on their tweets could offer more promising results

Number of K folds reduced – too long to run.

No test set used

TODO – add reference comparing our results to previous results

# References

|  |  |
| --- | --- |
| [1] | J. D. Burger, J. Henderson, G. Kim and G. Zarrella, "Discriminating Gender on Twitter," The MITRE Corporation, Massachusetts, 2011. |
| [2] | C. Fink, J. Kopecky and M. Morawski, Inferring Gender from the Content of Tweets:, Maryland: ICWSM, 2012. |
| [3] | W. Liu and D. Ruths, "What’s in a Name? Using First Names," *AAAI Spring Symposium - Technical Report,* pp. 10-16, 2013. |
| [4] | dbaghern, "A Deep Dive Into Sklearn Pipelines," 30 October 2017. [Online]. Available: https://www.kaggle.com/baghern/a-deep-dive-into-sklearn-pipelines. |
| [5] | M. Usman, "Text Classification with Python and Scikit-Learn," 27 August 2018. [Online]. Available: https://stackabuse.com/text-classification-with-python-and-scikit-learn/. |