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) |
|  |

# 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

## 4.1 LinearSVC

## 4.2 KNeighbours

## 4.3 Naïve Bayes

## 4.4 Conclusions

TODO – add reference comparing our results to previous papers results from references

# 5 Limitations and Outlook

We didn’t use as many folds as we would have liked to. Training and testing the models for different hyperparameter settings on different feature set combinations proved to take too long. There may be some bias in our results depending on the validation sets used in the different K Fold validation procedures.

In gathering our results, we didn’t use a test set. All our numbers come from an averaging procedure on the results from K Fold Cross Validations. Testing our final models on a test set may give a more accurate result as to how they would perform in a real-world scenario.

Only LinearSVC used the complete feature set of both textual and numerical features. We may have seen better results if the Naïve Bayes or KNeighbours model was adapted to use both types of features as well. Currently they only use either Numerical or Textual features, not both.

# 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/. |