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 in order 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. Therefore our research question is: **What features of a twitter user profile are particularly indicative of their gender?**

# 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.

Clay Fink [2] also examined this problem in detail. They determined gender by searching twitter users Facebook pages so as to determine ground their ground truth. 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 a dataset used by an existing research paper on gender classification [3] . The ground truth for this paper was generated by a gender association formula which classified users based on how often their name was linked to a certain gender on a U.S census. 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 model just predicting always Male in order to achieve a higher accuracy.

The text that followed hashtags were extracted, along with the number of hashtags used in a tweet. Each of these data types were made into separate features.

Following all the above pruning/additions, we were left with a dataset of approximately 6000 instances.

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.

## 3.3 Algorithms

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

We compare the use of 3 algorithms, to classify gender.

LinearSVC classifier trained on both numerical and textual features.

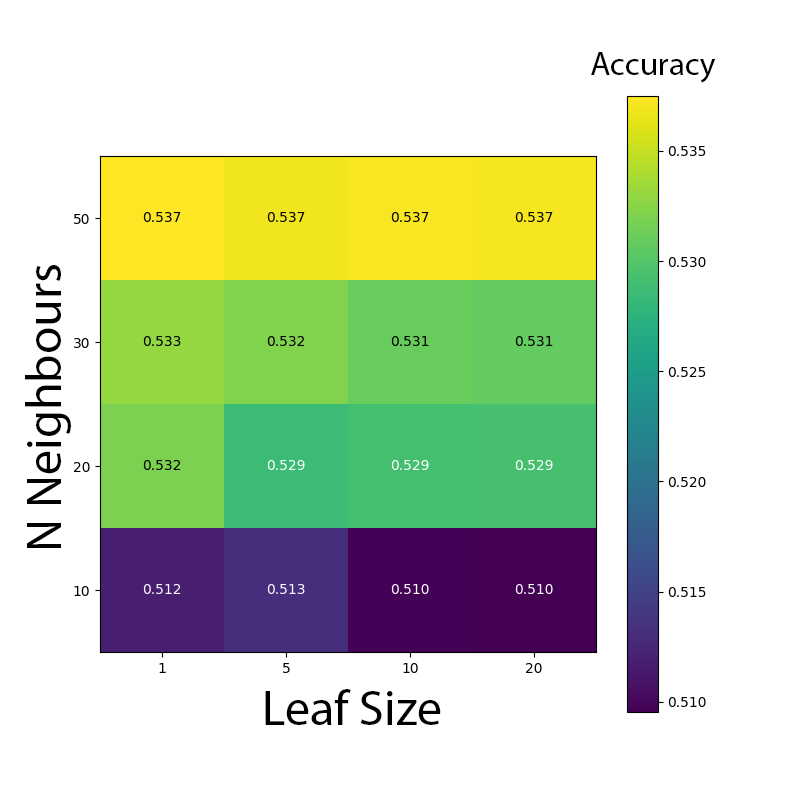
Naïve Bayes classifier trained only on textual features.

KNeighbours model trained on numerical features.

## 3.4 Hyperparameter Selection

For each algorithm, we created models using various sets of features and hyperparameters, in order to find the model which gives the optimal performance/accuracy.

Figure 1

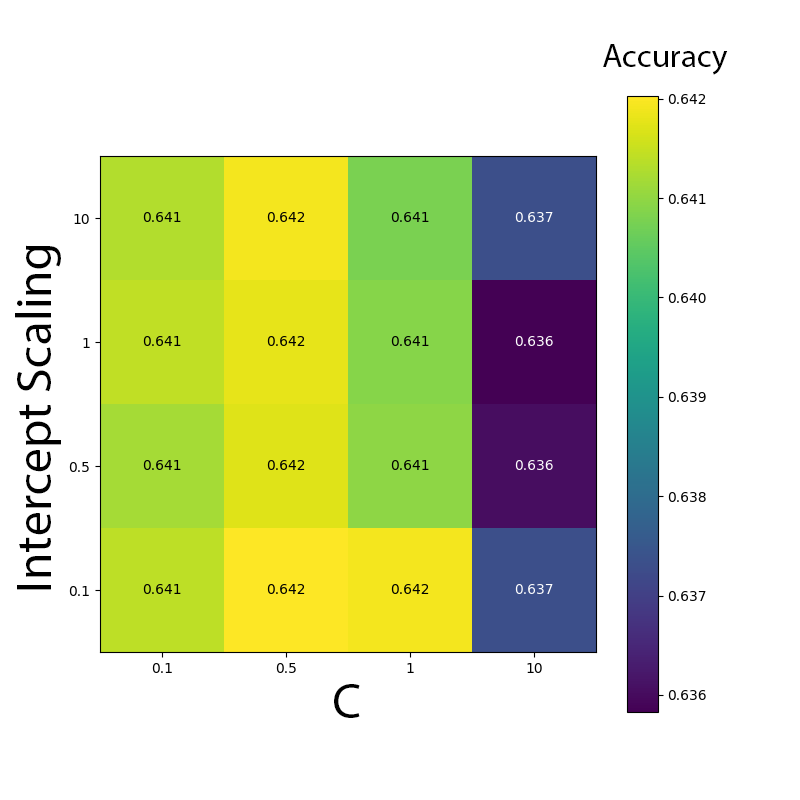


We ran each of our models under various Hyperparameter settings to arrive at the best accuracy. We plotted the heatmap of accuracy values of each model with its various Hyperparameters. In our results, the model was trained and evaluated with the parameter settings that had the highest accuracy from these heatmaps.

You can see from Figure 1 a variation of approximately 2% depending on the selected Leaf Size and N Neighbours in this KNN example.

Another example selecting the C and Intercept Scaling parameters for SVC in Figure 2

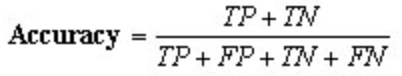
Figure 2

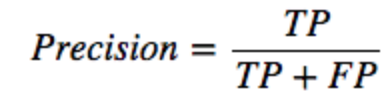


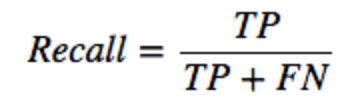
## 3.5 Evaluation

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.

The following formulae were used to calculate; Accuracy, Precision and Recall;







TP – Total number of True Positives (correctly predicted male)

TN – Total number of True Negatives (correctly predicted female)

FP – Total number of False Positives (falsely predicted male)

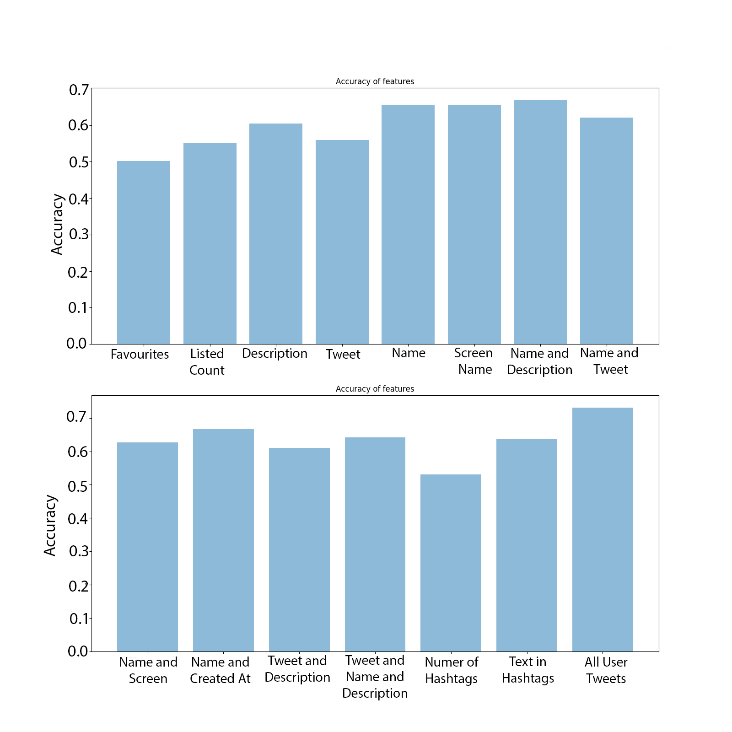
FN – Total number of False Negatives (falsely predicted female)

# 4 Results and Discussions

The following results were obtained through an averaging procedure of accuracy values from K Cross Validation.

## 4.1 LinearSVC

Figure 3



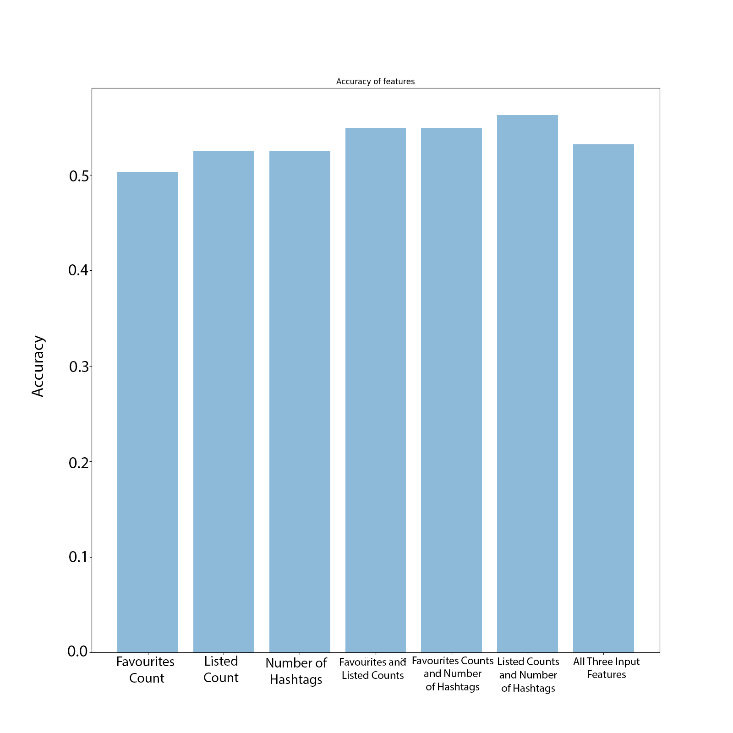
LinearSVC was the algorithm we used to measure accuracy on both textual and numerical features. In general, the singular features on their own didn’t show any promising results.

However, when we started combining textual and numerical feature input sets, we found the accuracy to rise above 60% in most cases.

Similar to our results in the Naïve Bayes model in 4.3 Naïve Bayes, classifying the gender based on 100 of the user tweets proved to be the most successful feature with an accuracy of TODO

## 4.2 KNeighbours

Figure 4



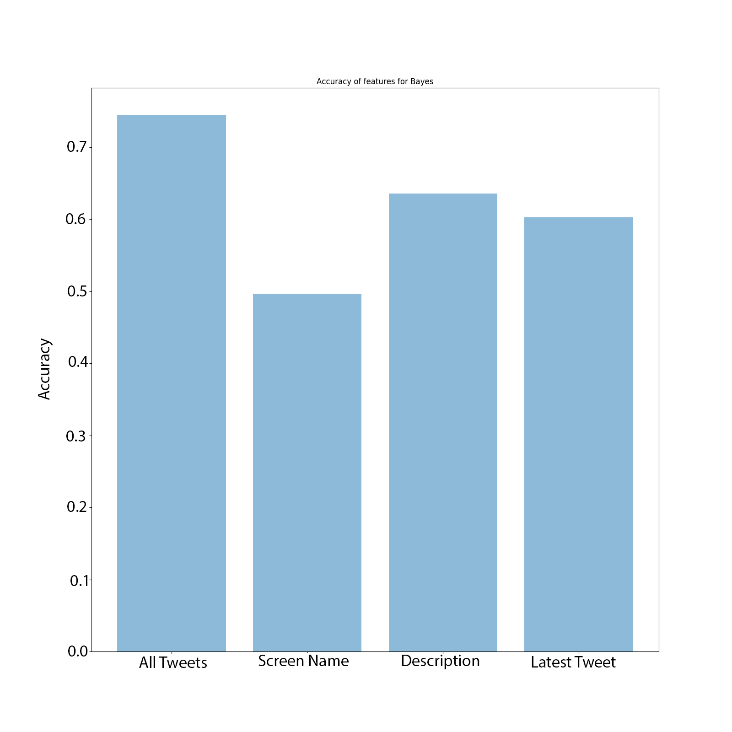
A KNN model was built for each numerical feature, as well as a combination of numerical features. It showed us that either KNN was a bad algorithm to use in classifying the gender from these numerical features, or the numerical features themselves are not indicative of gender.

The numbers for these numerical features in Figure 3are very similar to the accuracy numbers seen in the Linear SVC for the numerical features. For example, Favorites count in both models Figure 3 and Figure 4 is approximately 50%, so we can rule out KNN being a bad algorithm for these particular use cases. The numerical features just don’t seem to be a good metric for predicting gender.

## 4.3 Naïve Bayes

We trained and evaluated our Naïve Bayes model with individual textual features using 10-Fold Cross Validation.

Figure 5



The user tweets proved to be the best indication of gender over all other input features and algorithms with an accuracy of 75%.

The Screen Name proved to not be indicative of the gender under these circumstances. This surprisingly contrasted with the accuracy obtained with a LinearSVC algorithm for the same feature.

We could possibly achieve higher results if we trained a model on a combined feature set of User Tweets, Profile Description and their Pinned Tweet.

For more details on performance, we have a table of results in the appendix.

## 4.4 Conclusions

Generally singular input features are a poor metric for determining gender. Only when we started combining features did we see results of above 60%. We have a table of all our results in the appendix section 6.1 Tables. Generally, if a prediction accuracy was below 60%, it was considered by us to just be a little bit better than random chance, but still an unusable estimate.

Our best performing metric was an Naïve Bayes model that classified 100 user tweets. We found that the largest identifying factor between classifying a twitter user to be male or female was in fact the content and writing style of their tweets. The hashtags they use, the emoji they use etc.

From related works [1] [2] our best performing metric of 75% still doesn’t compare to the reported 92% [1] and 80.6% [2].

# 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.

Generally, we saw a trend where more input features gave a higher accuracy. We didn’t experiment past a combination of 3 input features in this research assignment. We could potentially increase our prediction rates with a better combination of input features.

## 6 Appendix

## 6.1 Tables of Results

Legend:

Green Rows = Best Performing Features

Orange Rows = Decent Performing Features

Red Rows = Poor Performing Features

LinearSVC Single Features

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Accuracy | Precision | Recall |
| Favorites Count | 0.5 | 0.44 | 0.99 |
| List Count | 0.52 | 0.59 | 0.36 |
| Profile Description | 0.6 | 0.47 | 0.64 |
| Tweet | 0.56 | 0.58 | 0.55 |
| Name | 0.65 | 0.83 | 0.4 |
| Screen Name | 0.66 | 0.83 | 0.39 |
| Number of Hashtags | 0.53 | 0.56 | 0.45 |
| Text in Hashtags | 0.63 | 0.7 | 0.48 |

LinearSVC Combined Features

|  |  |  |  |
| --- | --- | --- | --- |
| Features | Accuracy | Precision | Recall |
| Name and Description | 0.67 | 0.7 | 0.58 |
| Name and Tweet Accuracy | 0.62 | 0.63 | 0.61 |
| Name and Screen Name | 0.66 | 0.82 | 0.41 |
| Name and Account Creation Date | 0.67 | 0.81 | 0.44 |
| Tweets and Creation Date | 0.61 | 0.61 | 0.58 |
| Tweets and Name and Description | 0.64 | 0.65 | 0.6 |

KNN – Single Features

|  |  |  |  |
| --- | --- | --- | --- |
| FeatureSet | Accuracy | Precision | Recall |
| Favourites Count | 0.51 | 0.51 | 0.52 |
| Listed Count | 0.52 | 0.56 | 0.60 |
| Hashtag  Count | 0.52 | 0.53 | 0.56 |

KNN – Combined Features

|  |  |  |  |
| --- | --- | --- | --- |
| Favourites Count and Listed Count | 0.55 | 0.56 | 0.47 |
| Favourites Count and Hashtag Count | 0.55 | 0.55 | 0.49 |
| Listed Count and Hashtag Count | 0.56 | 0.56 | 0.53 |
| Favourites Count, Listed Count and Hashtag Count | 0.54 | 0.58 | 0.35 |

Naïve Bayes

|  |  |  |  |
| --- | --- | --- | --- |
| FeatureSet | Accuracy | Precision | Recall |
| All User Tweets | 0.75 | 0.77 | 0.69 |
| Screen Name | 0.49 | 0.27 | 0.71 |
| Name | 0.65 | 0.71 | 0.71 |
| Profile Description | 0.61 | 0.62 | 0.6 |

# 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/. |