GENDER PREDICTION ON TWITTER

INTRODUCTION

Social Media Platforms tries to complete user registration as fast as possible. This might be a boon to the users but not for companies who needs to target people for marketing and advertising based on Gender. In this project, we have used character-based n-gram for usernames along with word-based n-gram for description as well as tweets to train the model. We have also utilized favorites, retweet count, side bar colors and link colors for determining gender. We achieved an accuracy of 72.68% after ensembling using majority votes technique.

RELATED WORK

Study of relation between gender and language is extensive [4]. Username is one of the fields where the gender can be easily identified. Mohsen and Giovanni [3] used Microsoft Discussion Graph tool (DGT) to identify the names in the screen name. There were three classifiers (Image, Name and text classifier used in this paper to predict the gender. Liu and Ruths [5] introduced a gender-name association score for all names in Census data. The gender score is computed with the gender distribution of each name and it shows how often the name is given to male or female. Slater and Feinman [6] reported that the gender of a person can be identified using number of syllables, number of consonants, number of vowels and ending character. Jalal and Ugo [7] used language independent features like background colors, side bar colors to predict the gender of the profile. Among all the papers we saw, there wasn't any ensembling done for favorites, tweet_count, color, text, description and username with character-based n-gram. In this project, we will try to merge all these attributes to see a better accuracy.

MODEL/ALGORITHM/METHOD

For our analysis, we found that the text, description, tweets, link colors, side bar colors, retweet count, favorites, created date, profile image URL, username fields from the dataset might be helpful in determining the gender. We have filtered the duplicate records along with records having confidence level not equal to one. We have not used profile image URL as most of the images was deactivated after the user changed/ removed their picture. In the top 100 URLs, we were able to retrieve only 37 images. We had considered to retrieve the color histogram from the image URL and this will be part of our future work.

Method for text, description columns

Data preprocessing is the first step that we will be doing for most of the columns. We have removed URLS, non-ascii characters from the tweets. Feature extraction is done for the document using CountVectorizer from ScikitLearn. We have also used one more bag of words model scheme provided by ScikitLearn named TFidVectorizer. In a text document, there may be many frequent unimportant words like "the, and, are" present. TfidfVectorizer takes the text documents as input and it not only gives importance to the most frequent words but also to the rare words in document. In this project, we have compared the results with the feature extraction performed with 1-gram through 3-gram.

Specific genders might be interested towards use of more punctuation and stopwords. This has led us to train the model with/without these characters. After doing all the above steps described, the data may contain noise (irrelevant features) and these features when included might reduce the accuracy of the model. Hence, it is good to do feature selection before passing it on to classifiers. We followed two methods in this project for feature selection, one is to do selection based on maximum word frequency feature in CountVectorizer and TfidfVectorizer and second one using chi square.

The purged data is passed on to the classifier. We used Random Forest, Multinomial Naïve Bayes, Bernoulli Naïve Bayes to train the model and compared the results of these classifiers. Multinomial Naïve Bayes with feature selection done using Chi2 performed better among all the compared classifiers. Bernoulli Naïve Bayes also excelled when feature extraction done using TfidfVectorizer. We followed the same steps for the description field also. There was an increase in accuracy when we merged both text and description column.

Method for username column

As we all know that data preprocessing is the prior step before sending the data to the classifiers, we will be removing numbers from the username. We have also compared the results with/ without the numbers in the username. In this project, we used character n-gram for feature extraction. Because higher order n-grams gives the correlation of different characters within a text, we have used 3-gram to 5-gram. We know that few of the users will not keep their name in the username. It was surprising to see that this method, produced better accuracy than text + description.

Preprocessed username was sent to Multinomial Naïve Bayes, Bernoulli Naïve Bayes and Random Forest. The performance was better for the Multinomial Naïve Bayes like what happened for text and description. When we followed one of the ensembled method Majority voting for evaluating, we got a sharp increase in accuracy with the text + description and username.

Method for color, retweet count, favorites and created_year

In data preprocessing, there were records where there were no colors present. We checked the color count of the sidebar color column (Fig 1) and we found that the top value is the default value for all genders. Thus, we replaced all the sidebar colors with improper color hexacodes to default color. This is also followed for link color column.

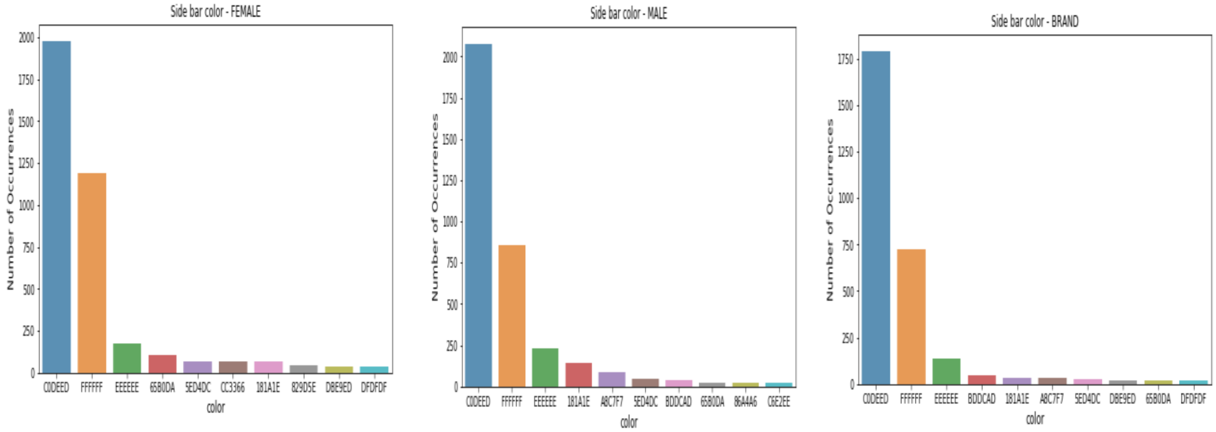


Fig a) Color count of Side bar color based on Gender

After all the required preprocessing is done, the color hexacodes are converted to RGB color with each component going to a separate column (i.e. R to one column, G to another column and B to new column). The hardest part of this approach is to that most of the people would be choosing default color irrespective of gender. This data is passed to Random Forest, AdaBoost and KNN and GaussianNB classifiers. Comparison of these results are covered in the evaluation section. We combined retweet count, favorites, color and profile_created_ year to train the model. This resulted in increase in accuracy when compared to color alone.

RESULTS

First, we checked for proper n-gram value for text, description and username using Multinomial Naïve Bayes. After selecting the required parameter from different combinations, we tested it on various classifiers with the n-gram value having better accuracy. Table1. shows us accuracies of various n-gram ranges used for parameter selection. Each gender may be inclined towards a special character, punctuations or a stop word. So, we decided to verify the accuracies for most of the combination available. From the accuracies which we attained, the text with the n-gram range (1,1) and without any punctuations in it was better performing. In the case of description, the performance was good when the text did not have any punctuations, stopwords and @screenname words in it. The username accuracy was the best of all with 66.2% when it was untouched.

		word			char					
	n-gram	(1-1)	(1-2)	(1-3)	(1-2)	(1-3)	(1-4)	(2-4)	(3-5)	(3-7)
Without Punctuation	text	59.25	56.95	56.49	52.3	56.3	57.25	56.25	57.73	56.38
	description	60.8	59.04	58.81	50.28	56.71	56.58	58.81	60.34	59.88
	username	NA	NA	NA	56.87	62.36	65.9	64.57	65.06	64.19
Without stopwords	text	56.95	57.25	56.68	53.4	56.5	56.57	57.44	57.71	56.6
	description	58.9	60.8	61.02	51.27	58.24	58.24	59.91	59.97	60.03
without username in tweet	text	58.05	58.32	56.72	51.62	55.8	56.72	51.62	58.05	0.43
Without Punctuation/ stopwords/ username	text	54.66	54.44	54.2	53.71	53.7	54.89	54.85	53.64	53.75
	description	60.83	60.8	61.1	57.25	57.25	58.85	58.55	59.61	59.69

	username	NA	NA	NA	61.55	61.86	63.2	64.72	64.6	65.1
With Punctuation/ stopwords/ username	text	56.95	57.25	56.68	53.4	56.57	57.44	57.44	57.71	56.6
	description	61.03	60.8	61.02	51.27	58.24	59.2	59.1	59.77	60.03
	username	NA	NA	NA	58.2	62.51	65.6	66.05	66.2	66

Table 1: Accuracies for various n-gram ranges for username, text and description

After selecting the filtering conditions from the above table, we will be using chi square for feature selection. Fig.2 and Fig. 3 shows us the accuracies of feature selection done using chi square with Count vectorizer and TfidfVectorizer. In fig 2., the plotting shows us that the accuracy staying stable after 5000. Hence, we chose 5100 chi2 returned features for tweet, 5100 features for description and 15100 features for username. When TfidfVectorizer used, we selected the top 3100 chi2 returned features for tweet, 2600 for description and 4100 features for username.

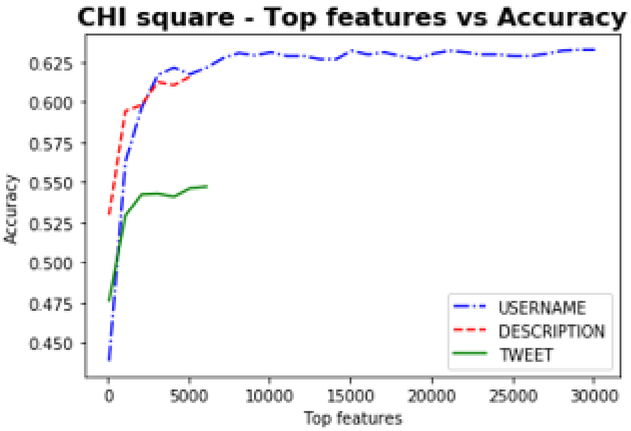


Fig. 2. Accuracies of model after Feature selection using Chi Square and Count Vectorizer

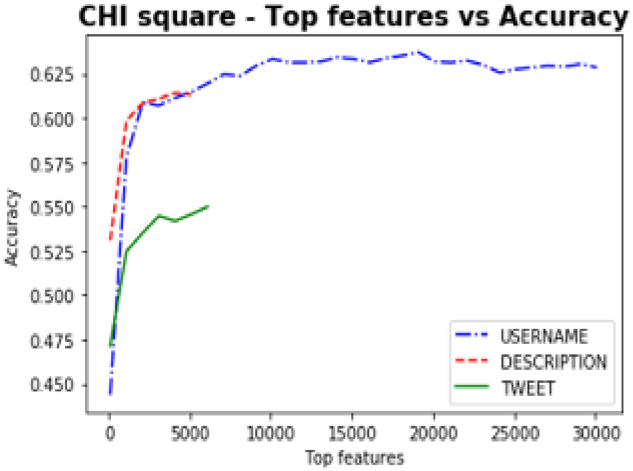


Fig. 3. Accuracies of model after Feature selection using Chi Square and TfidfVectorizer

After all data preprocessing, feature extraction and feature selection done, we started comparing various models.

The table 2 shows the accuracies of various models for color, retweet_count, favorites, created_year. The created _year field was created from created column. For colors, we tried with different classifiers like Random Forest, Decision Tree, KNN and Gaussian NB. As expected, Random Forest, Decision Tree and KNN performed better for colors. When we combined the favorites, retweet count, created year and colors, there was a sharp increase in the accuracy. We did an exhaustive search for the parameter values using GridSearchCV. The parameter values for the RandomForest was n_estimators as 9, max_depth as 8.

	Accuracy(10-folds)				
	Random Forest	Decision Tree	AdaBoost	KNN	GaussianNB
colors (Link/sidebar colors)	44.6	44.1	NA	43.13	38.33
favorites, retweet count, created_year + colors	56.45	NA	55.75	51.96	41.71
favorites, retweet count, created_year, created_month + colors	56.55	NA	55.34	50.89	41.88
favorites, retweet count + colors	54.47	NA	53.74	49.95	41.33

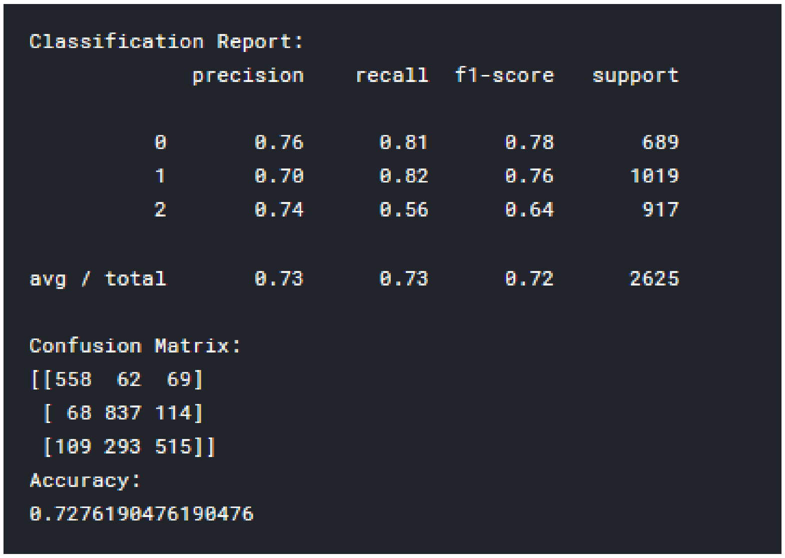
Table 3. Accuracies for the color, retweet, favorites, created_year model

The below table (Table 4) shows the accuracies for different classifiers. Maximum accuracy of 72.68% can be seen for Multinomial Naïve Bayes using CountVectorizer, when all the fields mentioned in the table are used. The parameter used for Multinomial NB is alpha 1.015 and the parameter value used for Random Forest is 'max_depth': 9, 'n_estimators': 8, 'random_state': 2}. BernoulliNB worked best with default parameter.

	BernoulliNB with CountVectorizer	MultinomialNB with CountVectorizer	BernoulliNB with TfidfVectorizer	Random Forest with Count Vectorizer
text	58.2	58.23	58.2	58.1
text + description	64.03	64.34	64.01	58.13
username	64.19	64.6	64.15	60.45
text + description + username	69.48	71.39	69.33	65.18
text + description + username + favorites, retweet count, created_year, created_month + colors	70.476	72.68	70.4	66.47

Table 4: Accuracies for classifiers used in text classification

We followed majority voting for merging all the best models we retrieved in the above steps. The multiplicative value of predict_proba and precision_score is calculated for each classifier and then the cumulative sum gives us the accuracy score of the model. The below image summarizes the final output with Classification report containing precision, recall and f-1 score. We can also see confusion matrix for all genders with the rows in the order of Brand, Female and Male.



```
Classification Report:
             precision    recall  f1-score   support

          0       0.76      0.81      0.78       689
          1       0.70      0.82      0.76      1019
          2       0.74      0.56      0.64       917

avg / total       0.73      0.73      0.72      2625

Confusion Matrix:
[[558  62  69]
 [ 68 837 114]
 [109 293 515]]
Accuracy:
0.7276190476190476
```

Fig. 4 Confusion matrix, Classification Report, Accuracy [0 – Brand, 1 – Female, 2 – Male]

CONCLUSION AND FUTURE WORK

Our first contribution in this project is to build a model for gender classification using text, description and username. Using character n-gram for username, improved the accuracy of the model to a greater extent. Our next contribution was in creating models using color, retweet count, created_year and favorites. We have plans in future to enhance the accuracy by considering images in the dataset. It would be interesting to look for the histogram values of each gender profile image.

REFERENCE:

1) Juergen Mueller, Gerd Stumme. 2016. Gender Inference using Statistical Name Characteristics in Twitter
2) Marco, Fernando, Joao Paulo Carvalho. Twitter gender classification using user unstructured information
3) Mohsen Sayyadiharikandeh, Giovanni Luca Ciampaglia, and Alessandro Flamming. Cross. domain gender detection in Twitter
4) J. Holmes and M. Meyerhoff, The handbook of language and gender, 2008.
5) W. Liu and D. Ruths. What's in a name? using first names as features for gender inference in twitter. In Analyzing Microtext AAAI 2013 Spring Symposium, pages 10–16. AAAI, 2013.
6) A. S. Slater and S. Feinman. Gender and the phonology of north american first names. Sex Roles, 13(7-8):429– 440, 1985
7) Jalal S. Alowibdi1, 2013. Language Independent Gender Classification on Twitter