CIS 787: Project Report

Name: Jyoti Bansal SU ID: 267844676

Gender Prediction on Twitter

1. Introduction:

With a rapid expansion in the use of social media like Twitter in the last few years, there has been a creation of unprecedented amount of user-generated text. These social media networks play a significant role in the day-to-day life of people, companies and organizations. They are considered as a medium of interacting with new people, sharing one's achievements, knowledge etc. This results in the generation of unprecedented amount of user data. One of the key findings observed during the research is that Twitter does not collect the gender information as do the other social media sites, which makes "Gender Prediction on Twitter" an interesting concept to understand the difference in the writing patterns between males and females. This fascinating observation intrigued me to develop a machine learning model via this project that would help identify the gender of the author based on different features like combination of tweets and user description or features like user name, sidebar color, profile link color and image. In this study, I test the accuracy of my model by using TF-IDF vectorizer, n-grams model, SelectKBest feature selection model and find the best fit. Various classification algorithms are employed out of which the best results were obtained from LinearSVC model with an accuracy of 69% based on tweets and description attribute. I have implemented a separate model to predict the gender of the author based on remaining features of the dataset like link_color, sidebar_color, etc. that yields 55% accuracy using RandomForest classifier.

In this paper, I will only explain in detail of how the classification model is built using the text and description attributes to predict the gender. However, the results of both the individual models has been enumerated in "Results section".

2. Prior Work:

Initially, there has been a lot of research already conducted in this domain. [1] There was a competition held on Kaggle on the topic called, "Gender Prediction on Twitter", where the dataset which I have used for this project, had been hosted. Multiple people have shared their kernels and discussed about the importance of different attributes in the prediction of gender. Some experts have provided deep insight in the classification models like Logistic Regression which proved to be accurate in prediction of gender.

Author in paper [3], emphasized on the usage of n-gram tokens to be used as the feature set based on which classification models like Perceptron model and Naïve Bayes model proved to fetch an accuracy in the range of (90 – 100) %. Author in report [2], also generated tokens using n-grams and used the reduced feature set based on which he created a Naive Bayes model providing a combination of different attributes.

3. Model/Algorithm/Method:

3.1 Data Preprocessing:

To start with, the dataset provided contained 20,000 records of tweets from unique users containing 25 different features. I first implemented a classification model to predict the gender using only the tweets and description attributes from the dataset of each user. I then implemented a separate classification model to predict the gender based on features like link_color, sidebar_color, profile_yn, profile_yn:confidence, tweet_count, etc.

The records which had "profile_yn" value set to "no" and blank records in the "text" and "description" attribute were removed from the dataset. As a result, the dataset size was reduced from 20079 to 16251. Using the

LabelEncoder class from sklearn library, I encoded the categorical target variable which labelled brand as “1”, female as “1”, male as “2”. Moreover, the instances whose gender was “unknown” in the Gender attribute column of the dataset were replaced by the most frequent value in that column. The following table summarizes the final distribution of the tweets across different values of the target variable:

Gender	Total Count (Before preprocessing)	Total Count (After preprocessing)
Male	6223	5496
Female	6700	6427
Brand (Organization)	5942	4328
Unknown	1117	0

Table 1: Distribution of the gender attribute

After analyzing the text of tweets and description provided in the dataset, I observed that the data contains some extremely common words which do not provide any significance in predicting the gender of the user. The data also contained the usage of many special characters, URL’s, blank spaces, characters not in English, etc. which would only reduce the accuracy in prediction of gender. As a result, I performed a cleaning step in which I removed all the special characters from the text, the blank spaces, the URL’s from the data and converted all the words which had an apostrophe to their root word. (eg: ‘s was converted to is, ‘ve was converted to have, etc.) I also imported the stopwords list provided by the nltk library and added some words to that list which seemed to be irrelevant after analyzing the dataset and removed them as a part of this cleaning process.

Once the data was cleaned, I observed that there were many similar words with similar meanings but were written in different forms. For eg. The words Deliver, delivered, delivering all have similar meanings but were used differently in different tweets. To convert these words to their root dictionary words while keeping their meaning same, I first generated tokens from the sentences and then used the WordNetLemmatizer class of the nltk library on these tokens to create a corpus of both tweets and description attribute of the dataset.

The first 2 diagrams display the top 20 words used by females and males in their tweets and description. The next 2 diagrams display the top 10 sidebar colors used by both females and males in their profile on Twitter. Similar results can be displayed for link color attribute.

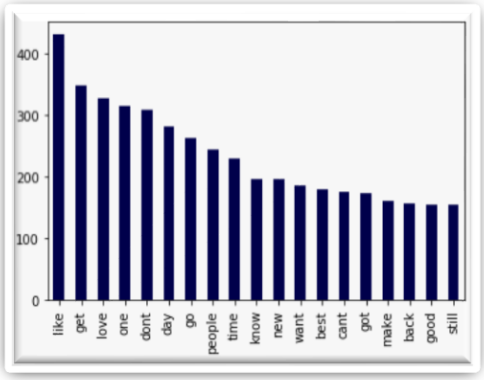

Fig 1: Top 20 Female Words

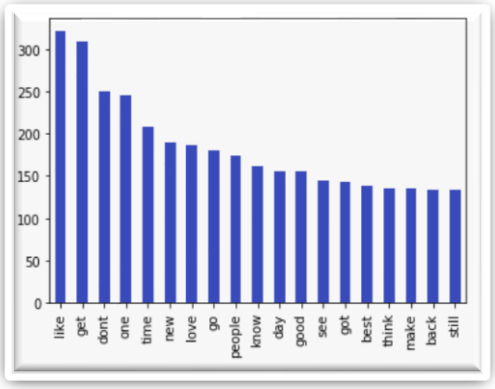

Fig 2: Top 20 Male Words

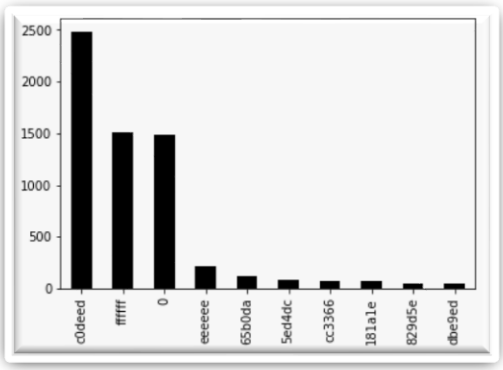

Fig 3: Top 20 Female Sidebar colors

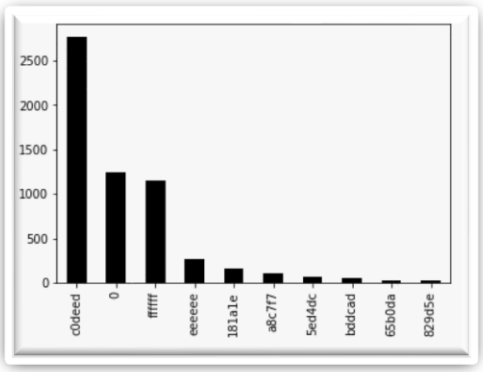

Fig 4: Top 20 Male Sidebar colors

3.2 Feature Creation:

To predict the gender of the author based on their tweets and description on Twitter, we need to analyze which words are more frequently used by which category of users. To accomplish this, I used TF-IDF vectorizer technique which is used to weigh all the words in the content and assign importance to that word based on the number of times it was used by the user in their tweets or description. Thus, it helped me analyze the relevant keywords for each set of users. With the assistance of TF-IDF vectorizer, I created a corpus of the top 30,000 words from a combination of tweets and description attribute based on the weights assigned to it by the vectorizer. I thus built a Compressed Sparse Feature Matrix (x) that would map each record to the frequency of each of the token appearing in it. The shape of my feature matrix resulted in (16251, 30000).

Weights provided by TF-IDF vectorizer to some of the common words and some rare words used by most users is as follows:

Word	Weight	Word	Weight
Love	3.3681907186859861	05 05	10.002824076547672
Get	3.5359020495114906	20 cm	10.002824076547672
Like	3.5484119859181633	as like crazy	10.002824076547672
Don’t	3.8033296159355285	asterisk	10.002824076547672
Life	3.8043453600553638	attorney woman	10.002824076547672
Time	3.8640176915417621	artist musician	9.597358968439508

Table 2: TF-IDF Weight distribution

To represent tweets and description text, I employed n-grams model to create a token of 1-gram to 5-gram tokens. The range of n-values for different n-grams to be created were passed as a parameter to the TF-IDF vectorizer that extracted the n-grams and assigned them weights on basis of their occurrence. Since, higher orders of n-grams disclose the actual correlation of different words, feature matrix was created using 1-gram to 5-grams to represent each tweet. However, the only pitfall of using higher order of n-grams is that as the value of n increases, the number of features also increase exponentially resulting in a very large sparse matrix. I have therefore created a vocabulary of maximum 30,000 words which have all possible n-grams some of which can be seen in the below wordcloud.

Fig 3: WordCloud

3.3 Feature Selection:

It was after reducing the size of feature matrix to 30,000 attributes, I observed that the size of the matrix is still too large for me to run my classifier on and predict the gender. Hence, I employed feature selection methods on the matrix to reduce the dimensionality of my dataset. Out of all the feature selection methods that I tried with my dataset, "SelectKBest" using the chi2 method, gave the best results combined with different classifiers in-order to predict the gender of the user.

By passing the chi2 as a score function, SelectKBest will compute the chi2 statistic to evaluate the probability of each individual attribute with respect to each of the target variable. The array of scores returned by this method helps SelectKBest to select the "k" top features thereby reducing the dimensionality of the dataset.

I performed an analysis on the optimal value of "k" that should be used for my dataset. Based on the diagram shown below, the optimal value for k turned out to be 8000. Hence, 8000 features were recorded for all 16251 instances that was then used as the training set on which multiple gender classification algorithms were deployed to predict the gender of the author.

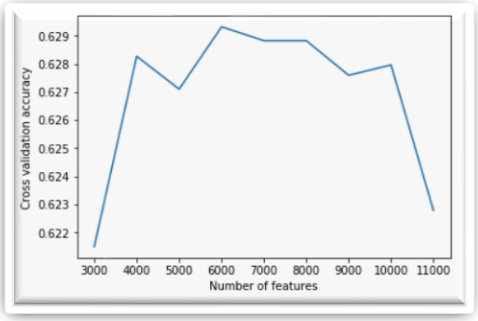

Fig 4: Feature Selection Analysis

3.4 Models:

Classification is a supervised learning process used for predicting the class/target variable of given data points. Based on the training dataset provided to the classifier, it understands and trains itself of how the given input variables are mapped to a class.

However, there are multiple classification algorithms developed which are used based on the application and nature of dataset. In this study, I employed Multinomial Naïve Bayes, Random Forest, Logistic Regression, SVM,

KNN and ensemble classifiers to analyze which model can better predict the gender of the author. Out of all the models used, SVM gives the highest accuracy and has turned out to be the most accurate predictor whereas on the other hand, KNN and Random Forest provide less accurate results.

Multinomial Naïve Bayes:

Multinomial Naïve Bayes classifier uses a probabilistic model for prediction, which requires the multiplication of the probabilities of each feature, based on the assumption that all features are independent of one another. The algorithm calculates a probability from the occurrence of each feature, with regards to the true class of the instance. The class with the maximum probability is considered to the prediction of the corresponding instance.

Logistic Regression:

Logistic regression algorithm provides a probabilistic view of regression and uses a linear equation with independent predictors to predict a value. However, the predicted value can range from +∞ to -∞. Since we want the output of the algorithm to be in the range (0,2) as per our dataset, we make use of logit function and squash the output of the predictor to lie in range from (0,2).

Support Vector Machines:

Linear Support vector classifier returns the best hyperplane that divides the data into different sections. Once, the hyperplane is obtained, we feed some features to the classifier and obtain the predicted class.

Ensemble Classifier:

Ensemble models combine the decisions from multiple models to improve the overall performance of the classification model. In this study, I have used the "max voting" technique of the ensemble classifier in which predictions made by each model is considered as a vote. The prediction obtained from majority of the models would then be used as the final prediction.

4. Results and Experimentations:

This section of the paper provides an analysis of the performance of different classification algorithms and their ability to predict the gender of the author accurately. The performance of these models is measured by metrics like accuracy, f1-measure, precision, recall.

Once the top 8000 important features were selected by the feature selection model, the feature matrix was divided into train and test set in the ratio of 80:20 along with the corresponding class labels. The above-mentioned classification algorithms were deployed on the train set to create the training model. Once the training model was ready, I executed the model on the test set and obtained the following results for each of them:

Classification Model	Prediction time (in secs)	Accuracy (%)	F1-Measure	Precision	Recall
Multinomial Naïve Bayes	0.06s	67.3	0.67	0.70	0.67
Logistic Regression	0.056s	66.49	0.66	0.67	0.66
Linear SVC	0.053s	69.5	0.69	0.70	0.70
Random Forest	0.347s	58.9	0.57	0.64	0.59
Ensemble (Voting Classifier)	0.842s	68.5	0.68	0.70	0.69

Table 3: Classification Model Analysis (Tweets and Description)

Classification Model	Prediction time (in secs)	Accuracy (%)	F1-Measure	Precision	Recall
Multinomial Naïve Bayes	0.008s	46.33	0.45	0.46	0.46
Logistic Regression	0.004s	45.3	0.45	0.45	0.45
Random Forest	0.127s	55.96	0.56	0.55	0.56
Ensemble (Voting Classifier)	0.136s	49.63	0.48	0.49	0.50

Table 4: Classification Model Analysis (Remaining Features)

The ensemble voting classifier results are obtained by taking the majority votes of only Multinomial Naïve Bayes and Linear SVC model.

Based on the results of Table 3, it is observed that all the models provide approximately the same accuracy in predicting the target variable. However, the clear winner amongst all is LinearSVC model with 69% accuracy. This suggests that the data can be linearly separated based on which the target variable can be predicted accurately. However, the results of Random Forest classifier and KNN classifier indicate that these two models did not perform well and should not be used in prediction of gender based on tweets and description.

Based on the results of Table 4, it is observed that amongst all the models, Random Forest Classifier yields the best accuracy of 55% in predicting the gender of the author based on the remaining attributes of the dataset.

I also computed the confusion matrix which is considered as an error matrix where the columns represent the instances in a predicted class and the rows represent the instances In an actual class. Once the target labels of the test set are predicted by running the classification algorithm, I obtained the following confusion matrix of the predicted labels and actual labels for each model.

		Predicted Class Labels		
Actual Class Labels		Brand	Female	Male
	Brand	579	149	132
	Female	84	1009	219
	Male	79	334	666

Table 4: Confusion Matrix for LinearSVC

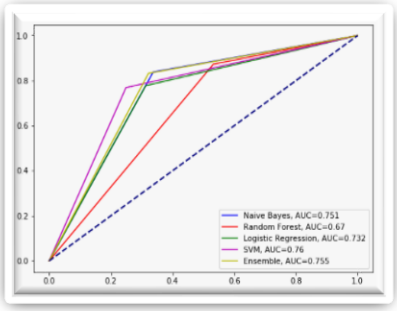

Fig 5: ROC Curve of classifiers for tweets and description

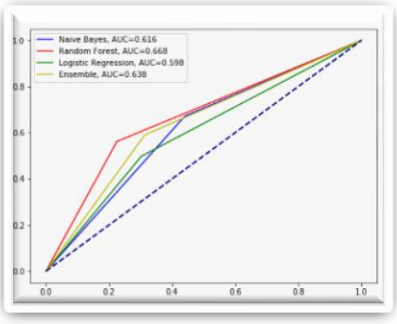

Fig 6: ROC Curve of classifiers for remaining features

From the experiments and the ROC curve (Fig. 5) we can observe that the Naïve Bayes, Logistic Regression and SVM have approximately similar accuracies, all having reasonable training and testing time. Thus, we can select any one of those as our final classifier or select ensemble classifier of these 3 classifiers to predict the gender based on tweets and description attributes.

From the experiments and the ROC curve (Fig. 6) we can observe that the Random Forest Classifier provides the best results amongst all. Thus, we can select Random Forest Classifier to predict the gender based on attributes like link_color, sidebar_color, tweet_count, tweet_location, fav_number etc.

5. Conclusion:

With the speedy growth in the use of social networks, particularly Twitter, there is an enormous amount of user generated text produced on daily basis. Because of the identity concealment on the Internet, many times the tweets posted by the user become the only data source for gender identification. However, data crawled and collected from social media sites like Twitter are often restricted by the text limit of 140 characters which increases the difficulty in analysis and identification of gender.

In this study, we attempt to identify user genders based on their tweets and description on Twitter. To achieve this, I have used a novel approach of creating and representing each tweet as a vector based on 1-gram through 4-gram features. Although higher orders of n-gram provide more understanding about the tweets and description of the user, they also require exponentially more features to be used creating a sparse feature matrix. To extract the informative features and improve the classification and runtime of our algorithms, we first fetched only the top 15,000 n-grams using a TF-IDF vectorizer of both tweets and description attributes. Based on this feature matrix, we ran SelectKBest feature selection algorithm through which the top 8000 features were selected thereby reducing the dimensionality of the dataset. To evaluate the efficiency of these selected features for gender identification on Twitter, I employed Multinomial Naïve Bayes, Linear SVC, Logistic Regression, Random Forest, KNN and Ensemble classifiers. Out of all these, best results were obtained from the LinearSVC classifier with 69% accuracy and 70% precision.

I also tried to understand the significance and contribution of other features in the dataset to predict the gender of the author. To accomplish this, I removed all the null values from the dataset, replaced the unknown values in the gender attribute with the most frequent value in that column, used a label encoder across all the attributes, selected the top 7 features using reduced feature elimination method and lastly applied different classification algorithms to analyze the best model which predicts the gender of the author accurately. Out of all the classifiers, RandomForest provided the best results with an accuracy and precision of 55%.

6. Future Work:

For future work, we need to increase the size of the dataset by crawling more tweets so that we can get more unique tokens from the tweets and description. In addition, we can find more ways to clean the data more efficiently that can help increase the accuracy of the model. We can perform more analysis on the significance of the remaining attributes of the dataset in prediction of gender. Currently, the model developed is not very accurate, thus we can work with some different feature selection and classification models that can help improve the accuracy of model. Furthermore, we can think of predicting other demographical features besides gender like the age, occupation, religion, marital status, location, etc. using the same set of features.

7. Appendix:

All the coding of the project was done in Python. I have included the dataset and all the .py and .ipynb files. I have also included the stopwords file used in the code. There is a separate file of analysis.py which contains the code snippets of all the analysis done during the project.

1) tweetClassifier_model.py
2) tweetClassifier_model.ipynb
3) userProfileClassifier_model.py
4) userProfileClassifier_model.ipynb
5) analysis.py
6) stopwords.txt
7) gender-classifier.csv (dataset)

8. References:

1) https://www.kaggle.com/crowdflower/twitter-user-gender-classification

2) https://github.com/vjonnala/Gender-Classification-using-Twitter-Feeds

3) “Gender Prediction on Twitter Using Stream Algorithms with N-Gram Character Features”, Zachary Miller