**Project 3- Team 3 Analysis for the name gender classifier**

To build the best name gender classifier, the name package was downloaded from nltk using “nltk.download(name)”. Then, the male and female names were loaded in “male.txt” and “female.txt” respectively. After shuffling the names so they appeared randomly, the first 10 names were printed out in a dataframe format. There were a total of 7,944 names. These names were then divided into three parts: the ***first 500*** becoming the **test set** (to be used as a final test), the following ***500*** names becoming the **dev-test set** (to be used for error analysis), and the **training set** being the **last 6944 names**.

**Models Used to Predict the Gender of Names:**

* **Baseline (using the last letter of the names)**

After all the preparations, the Text book’s code was used to help build the basic structure for the name gender classifier (called the gender\_feature). All words were changed to lower case and the last letter of the names was printed out. Using the dev-test set, the result showed an accuracy of 75% in predicting the gender of names.

* **Baseline + Last Trigram (using the last letter and the last 3 letters)**

The last Trigram was added to the above classifier in hopes of increasing the accuracy. For example, given the name “Melodie” the last trigram would be “die” and hopefully this, in addition to the last letter being “e”, would allow the correct prediction of the name’s gender as “female.” This addition to the model increased the accuracy to 77%. Logically, this was to be expected since the last letter alone would make it difficult to determine gender (ex. “Melodie” is female but “Eddie” is male and yet they both end with “e”).

* **Baseline + Last Trigram+ First Trigram + First Four Grams (adding on to the previous model the first 3 letters and the first 4 letters)**

Because combining the last letter and last trigram resulted in an increase in accuracy, other combinations were then added (in particular, the first the 3 letters and the first 4 letters of a name). This lead the model’s accuracy to rise to 85%.

* **Baseline + Last Trigram+ First Trigram + First Four Grams + Entropy**

The addition of the “Entropy” feature to the classifier did not improve the result. Instead, the accuracy decreased to 82%. This may make sense when taking into consideration Type I and Type II error. Before “Entropy”, we have 85% accuracy that does not mean that it is actually 85% in the reality. Maybe there are “errors” existing in our classifier when it fetches for the gender and name. There is some “off” information contained in our dataset.

For Type I error, for instance, under the category of male names, our classifier identified some female names as male names.

For Type II error, for example, under the category of male names, our classified identified the male names as female names and put them under the female category.

The entropy feature helps us to correct some of our false positive and false negative. Our accuracy level is more reliable.

* **Entropy and Model performance**

We use the sample code from text book to test how the entropy works. By the textbook, the formula of Entropy is

For example for the line of entropy(‘male’, ‘female’, ‘male’, ‘male’), to calculate it, we will have entropy = label for males and label for female= = .811278

Then we will do the entropy for input labels of names, and we get about .9510 for our test. It is close to 1 and it is considered high. When entropy is high, the input values have highly varied labels. For our case, entropy of labels in the name gender classifier depends on the ratio of male to female names. When the entropy is high, there are many labels with a medium frequency. Probability of labels and log of probability of labels are not small. It means that our data set has a relatively balanced gender labels for names.

Then we run the classifier to show that how many names misplaced in our classifiers. There are 89 names in total that we are misplaced. Around 18% error in our classifier. Then, we use “most\_informative\_features” to show the ratio of female and male for different last letter and last trigram. It gives us an informative idea about the prediction.

When we did some statistics for male and female, we get the mean for male is 2.32 with standard deviation is 0.35 and the mean for female is 2.27 with standard deviation is 0.37. Standard deviation is relatively small that is good sign for our data set. It means that the data set varies less. For example, some names such as “Alex”, “Francis” and so on, they may be with female labels and male labels, because they are unisex. With mean 2.32 for male, it means that for names with same first trigram/four gram, last trigram and last letter, around 2.32 names under the male category and 2.27 names under the female category. The histograms for both genders also show that they are approximately normal distribution.

* **Testing with real data**

The data used to test the classifier’s prediction capabilities was the male vs. female top 1000 names from the US social Security Administration for each year since 1880. From the plot, it was easy to tell that before 1900, the classifier still had a good estimator with an accuracy of more than 80%. However, this dropped significantly in 1920 and after 2000. Despite the decline, the accuracy percentage remained above 78% which could still be considered a good estimator.