ECS763P/U Assignment 3: EastEnders Gender Identification

You are provided with the following data: a set a training set (**training.csv**) and test set (**test.csv**) consisting of lines from scripts of the television soap opera EastEnders (from the 2008 series of episodes). These are two comma-separated csv files of the format:

line, character, gender

Where the character column indicates which character said the line in the show and the gender column indicates which gender that character is. There are no headers in the data, and the data is randomized in order (it is not in dialogue order from the scripts).

Gender Classification (30 marks)

Using the training data, use the skills learnt throughout the course (in the statistical methods, syntax and semantics lectures) to train a classification function which takes a line as input and classifies the gender of the character of the line (**but NOT using the character column label as a feature)**.

For developing your classifier, you may either split the data into a training and validation set within the main training data or use cross-validation - either is fine, but you must not develop using the test data or look at the test data in the test file. Once you have optimized your gender classifier, report your final results from applying the trained classifier to the **test data in terms of weighted f-score and a classification report** to show accuracy on both genders.

You can begin by adapting the code you developed in **Assignment 1, or with the specimen answer notebooks linked to in your individual feedback to this task** (though be wary of the differences in the input data, such as the .csv files not having headers). You can begin by training an SVM classifier as per Assignment 1, though you can explore using other classifiers you're familiar with, including classifiers you've encountered in the module so far such as (Complement) Naïve Bayes. There are some unique challenges to this data in terms of dealing with empty lines (where your classifier must still try to classify, as these empty lines may come up at test time) and non-standard English characters, challenges which are realistic in a real-world application.

Build on this starting point by using other methods learnt in the class to improve on your classifier by providing different features other than just the words (or pre-processed lemmas/stems). For example, using grammatical features can include POS tags from a **POS tagger** (like that provided to you in **Assignment 2**), or use a **parser** to get parse trees, production rules, number of special constituents, e.g. NP, VP, PP, dependency relations, e.g. nsubj, mod, det, etc. as binary features. You could go beyond the syntactic features by using a **named entity recognizer** or **sentiment analyzer** to get extra semantic features. You can also characterize other features for speech styles by looking at the presence of scripted pauses denoted by '…' and other markers.

Your submission notebook (.ipynb) or python (.py) file should contain the following:

i) Evidence of developing your model trying out different pre-processing, features and/or classifiers/parameters thereof either in cross-validation or with heldout data.
ii) Your best model applied to the test data including a classification report and weighted f-score on the test data.

Note on the Data:

These are the conditions on which you have access to the scripts:

The data is supplied under the signed agreement signed between the BBC and QMUL
The data is supplied for the purpose of training data for the agreed NLP Masters module at QMUL in November/December 2019, January 2020. If the data is to be used for any other purpose this has to be agreed in writing.
If there is a publication arising from use of this data the BBC should be provided with a chance to review the publication with respect to BBC sensitivities around the data
The data has to be stored securely, only passed on by Julian Hough to specified students.
Each student must agree to the following terms:
The data must not be passed to anyone else or placed online for others to access (e.g. not on a website , Github, etc)
The results of any processing of this data may not be placed online for public access
The data must be deleted after the end of the project.

General Mark Scheme:

The details of the marking scheme is as follows:

10% for preprocessing.
30% for feature engineering.
30% for a sound evaluation and performance of the classifier (including baselines if possible).
30% for the quality of the results presentation/exploration.