# Assignment 03: Text Classification in Scikit-Learn

Muhammad Awais, SP19-PCS-012

## Introduction

Text classification is one of the fundamental task in the natural language processing to assigning tags to the text according to its content. The author gender identification problem is also a text classification problem with two classes (male and female), when text is submitted it assigned a class on the basis of its content and style. Each person has its own writing style and content; this is refer to as their author profile. There have been over 1000 stylistic features proposed to date in the literature to identifying the authorship. Moreover, content discussed in the text is also refer as their author profile. To design a classifier we need to extract features from the text that are consistent for author of the same gender. Therefore, in this work we investigate multiple text documents and extract features from them that potentially divide authors into male or female.

## Methods for Text Classification

The text classification methods can be classified into two types of methods: stylometry based method and content based method. In this work, these two methods are explored.

### Stylometry Based Method

Phycology researchers have revealed that writing style of each author is unique features that he/she can be used consciously or unconsciously. These wring styles develop a pattern to identifying the gender by extracting the stylistic features from text. These features may language dependent or independent. In this work, lexical character based stylometry features for text classification has been used. It depends on sequence of characters and character count are used as features, i.e., space count, digit count, special characters count, small and large alphabets counts, brackets count etc. A simple Python script is written named "**Stylometry_Based_Classification.IPYNB**" that read the dataset and count the features to extract these 20 lexical character based features and save it in the CSV file named **"Stylometry_Based_Features.cvs"**. Moreover, the performance is evaluated for individual lexical features as well and 20 files for each feature named **"Stylometry_Based_Features*_featurename*.cvs"** are also created. After that, the model is trained and tested by using two different machine learning algorithms.

### Content Based Method

Text is a composition of characters and combination of characters make words sequence. Therefore, the sequence and order of the words carry useful information about its author. To extract content based features generally two approaches are used: word n-gram model and character n-gram model. In word n-gram model the text is tokenized into tokens (words) and features are extracted which consists of the adjacent string of tokens. Whereas, in character n-gram the text is tokenized into characters and string of adjacent tokes are considered as a feature by applying sliding window operations throughout the string. In this work, both models have been used. For word n-gram models the value of 'n' varied from 1 to 3, whereas, for character n-gram the value of 'n' varied from 3 to 10. The following steps have been done for the text based classification: Loading Dataset, Preprocessed the data, Convert textual data to numerical data, Feature selection, Training and testing.

### Feature Extraction

A simple Python script is written named "**Content_based_text_classification.IPYNB**" that read the dataset, preprocessed it (removing URLs, unwanted spaces, numbers, special characters, single characters, converting into lower case and lemmatization) and extract the features by using word n-gram model and character n-gram model. At first, the simple unigram representation of features has been done and created a binary features file named as **"Simple_unigram_representation.csv".** After that, the word n-gram feature representation is implemented by considering the length of **"n"** vary from 1 to 3. Similarly, for character n-gram representation of features the value of **"n"** is considered varying from 3 to 10. These extracted features with their corresponding TFIDF values are saved in files named "**Word_N_gram_Features.csv**" for word n-gram features and "**Chracter_N_gram_Features.csv**" for character n-gram features.

### Convert textual data to numerical data

Machine cannot understand the raw data, it can only understand the numbers. Therefore, the machine learning algorithms also deal with numbers. So, it is necessary to convert the textual data into numeric data. To perform this task we used ***TfidfVectorizer*** object from the ***sklearn.feature_extraction.text***. TFIDF is calculated by multiplying the Term Frequency (TF) of particular word with Inverse Document Frequency (IDF) of that term. Three important parameters of ***TfidfVectorizer*** include ***max_features***, ***ngram_range***, and ***analyzer***. The ***max_features*** parameters sets a limit on the maximum number of words that this ***TfidfVectorizer*** considers. We set this to a value of 1500 for our task. The ***analyzer*** controls how to break the strings: words or individual characters. We used both word and char values for this parameter. Finally, we set the ***ngram_range*** to 1 to 3 if the value of analyzer is set to be word and 3 to 10 in case of ***analyzer=char-wb***. After this step is completed, we now move towards the actual classification part by first splitting all our data into the training and test sets.

### Training and Testing

We used 10-fold cross validation approach to split the data set with multiple subsets and select any pair for the estimation of the performance using stylometry and content based features. We also used the ***train_test_split*** module from the ***sklearn.model_selection*** library. We set the ratios of the division so that our data is divided into corresponding ratio automatically. This split approach used after feature selection method. Before that, the algorithms are trained by using 10-fold cross validation approach which returns the classifier with highest accuracy score. We divide the data into 20% test set (85 instances) and 80% training set (340 instances). This utility also shuffles or randomizes the data automatically. To improve the accuracy of the classification and reduce the time needed by the algorithms to produce the results, we need to select the most distinguishing features from our data.

### Feature Selection Methods

The textual data consists of the large amount of text and the features extracted by applying feature extraction models construct a large feature space. It is exhaustive for most of the classifiers to deal with such large feature space. To mitigate this issue, multiple feature selection methods are used to extract the most discriminating features from the large feature space and remove the redundant features. In this work, we used n-gram models for feature extraction. Whereas, for feature selection we have used **Recursive Feature Elimination** (**RFE**) approach. It gives external estimator that assign weights to the features to select smaller to smaller set of features. We have used LinearSVC as an external estimator in feature selection. We also set the **step** parameter of RFE to 1 so that it eliminates a single feature in each iteration. After feature selection, we are now ready for the final and real task of text classification.

## Experimental Setup

In this section, we will discuss the experimental setup used for the text classification by applying the content based methods and stylometry based methods on PAN-AP-16 twitter corpus.

### Techniques

For stylometry based features, we applied 20 distinct lexical character based features. Moreover, we have applied word n-gram and character n-gram models for content based method. We set the range of 'n' 3-10 for character n-gram model, whereas 1-3 for word n-gram model. It is worth noted that, the data is first preprocessed for content based approaches.

### Dataset

For experimentation purpose, we have used PAN-AP-16 twitter corpus for gender identification which consists of 425 author profiles, 215 female profiles and 210 male profiles.

### Evaluation Methodology

The author’s gender identification task is treated as the supervised document classification task. We deal the problem as a binary classification problem because our goal is to distinguish between two classes: male and female. Furthermore, we have used K-fold cross validation approach for better estimation of performance with stylometry based and content based approach. Two different machine learning algorithms were used for this classification task including Random Forest and linear Support Vector Classifier (LinearSVC) of the Support Vector Machine classifier. The numeric values obtained from TFIDF after feature engineering methods are the input of these classifiers. In this work, the RFE method is used to select the most distinct features from the extracted features.

### Evaluation Measures

The performance of the algorithms are evaluated on the basis of accuracy and confusion matrix. The data is nearly balanced therefore the accuracy is the better evaluation measure and too look deep insight in the performance of algorithm, we used confusion matrix as well.

## Results and Analysis

In this task the performance of the classifiers are compared with the base line approach called Most Common Category (MCC). The accuracy of MCC is calculated by assigning the most common category and according to this the **MCC accuracy is 0.50**.

For the actual training and classification or prediction, we use the ***RandomForestClassifier*** and the ***LinearSVC*** machine learning algorithms. For the RandomForestClassifier, we set the parameters **n_estimators = 1000**, for the number of trees in the forest, and **random_state = 0** for the random selections of features to remain the same during each iteration, so that our classification results remain consistent. For the **LinearSVC** algorithm, we used default parameters.

### Results Using Stylometry Based Method

After applying the algorithms we calculate the prediction by using stylometry based method. The detailed performance of the algorithms before and after applying feature selection method are illustrated in Table 1. It is clear from the Table 1, that the feature selection have an impact on the results, and Random Forest outperforms the LinearSVC classifier in terms of accuracy. The best result is obtained when we consider all the stylistic features. This shows that the combination of stylistic features improve the performance of the algorithms. Moreover, the best accuracy of 0.61 is achieved on the “No. of Commas” feature in case of stylometry based method. This means that it the most discriminating feature among all other features. The deep insight of the classifier is clear from the confusion matrix given in Figure 1 and 2. It is clear from the figure 1 that how much instances are truly classified by the machine learning classifier, i.e. the 24 instances are truly classified as females because “0” is for female and 1 is for male and 7 instances are misclassified as males, but in actual they are females. Same is for figure 2.

*Table 1 Detail performance of stylometry based method*

| **Features** | **Without Feature Selection** | | **With Feature Selection** | |
|---|---|---|---|---|
| | **Classifier** | **Accuracy** | **Classifier** | **Accuracy** |
| | **MCC** | **0.50** | **MCC** | **0.50** |
| All features | LinearSVC | 0.57 | LinearSVC | 0.62 |
| | Random Forest | **0.61** | Random Forest | **0.77** |
| No. of Commas | Random Forest | 0.61 | - | - |

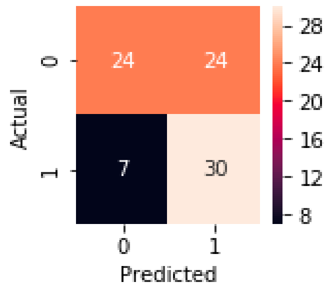

*Figure 1 Confusion matrix of LinearSVC Classifer*

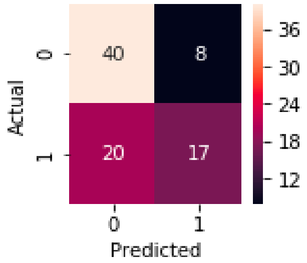

*Figure 2 Confusion matrix of RandomForest*

**Results Using Content Based Method**

Best results of this method for word and character n-grams are presented in Table 2. The highest accuracy of 0.74 and 0.76 for word and character gram respectively, is achieved after applying features selection method. This implies that the feature selection not only reduced the feature space and computation complexity, but it also improves the accuracy by shortlisting the most discriminating features. Moreover, it is also clear from the results that the character n-grams are the most discriminating features for gender identification on this dataset. That is why, in stylometry and in character n-gram the accuracy results are not drastically change because both are depends on lexical character based features. Furthermore, it is also seems that the accuracy decrease with the increase in word or character n-grams. This is because the dataset is not too large that the large length words are frequently used. Therefore, it can be deduce from the results, that for this type of application of author gender identification the character n-gram is effective model to use. The deep insight of the classifier is clear from the confusion matrix given in Figure 3 and 4. It is clear from the figure 3 that how much instances are truly classified by the machine learning classifier, i.e. the 29 instances are truly classified as females because “0” is for female and 1 is for male and 19 instances are misclassified as males, but in actual they are females by Random Forest classifier using word n-gram. Same is for figure 4.

*Table2 Detail performance of content based method*

| **Features** | **Without Feature Selection** | | **With Feature Selection** | |
|---|---|---|---|---|
| | **Classifier** | **Accuracy** | **Classifier** | **Accuracy** |
| | **MCC** | **0.50** | **MCC** | **0.50** |
| Word n-gram | **LinearSVC** | **0.73** | LinearSVC | 0.71 |
| | Random Forest | 0.66 | **Random Forest** | **0.74** |
| Character n-gram | **LinearSVC** | **0.75** | LinearSVC | 0.71 |
| | Random Forest | 0.65 | **Random Forest** | **0.76** |

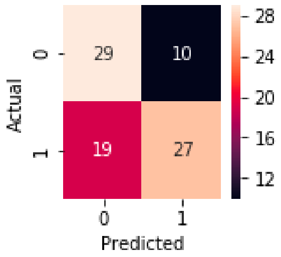

*Figure 3 Confusion matrix of Random Forest Classifier for Word n-gram*

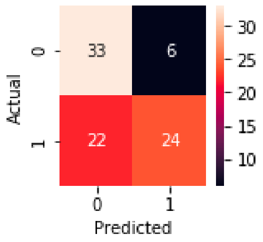

*Figure 4 Confusion matrix of Random Forest Classifier for Character n-gram*