Dobre Mihaela Beatrice

Romanian sub-dialect identification

I gathered the dataset and information required for the problem from Kaggle. The task is to train a model on tweets in order to build a model for a in-genre binary classification by dialect task, in which a classification model is required to discriminate between the Moldavian (label 0) and the Romanian (label 1) dialects. The training data is composed of 7757 samples. The validation set is composed of 2656 samples. All samples are preprocessed in order to replace named entities with a special tag: NE. Each line in the data samples have two columns based on tab separated values: the first column shows the ID of the data sample and the second column is the actual data sample.

In the first place, the text must be parsed to remove words. The words need to be encoded as integers or floating point values for use as input to a machine learning algorithm, called feature extraction (or vectorization). The scikit-learn library offers easy-to-use tools to perform both tokenization and feature extraction of the text data and for that I used TfidfVectorizer which will tokenize documents, learn the vocabulary and inverse document frequency weightings. I chose to approach the char n-grams features.

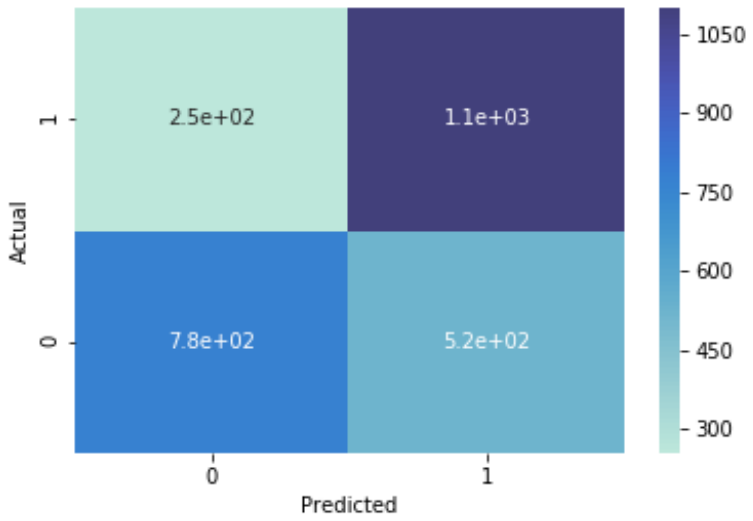
The first method I used is Naïve Bayes Classifier and more exactly the Multinomial Naïve Bayes Classifier. Naive Bayes is a classification algorithm for binary and multi-class classification problems and the probabilistic model of naive Bayes classifiers is based on Bayes' theorem, and the adjective *naive* comes from the assumption that the features in a dataset are mutually independent. In practice, the independence assumption is often violated, but naive Bayes classifiers still tend to perform very well under this unrealistic assumption.

For the second method, I also tried the Complement Naïve Bayes classifier because it was designed to correct the "severe assumptions" made by the standard Multinomial Naive Bayes classifier. It is particularly suited for imbalanced data sets.

For this two methods, I implemented a Grid Search and a Cross Validation function to find the right hyperparameters for the TfidfVectorizer (max_features, number of n-grams) and the best alpha hyperparameter. For MultinomialNB, I obtained the best F1 score for maximum of 120000 features, best n-grams were (5,7) and best alpha was 0.2. For ComplementNB the best number of features and the best number of n-grams were also 120000 and (5,7) and best alpha was 0.05.

Multinomial Naïve Bayes classification report and confusion matrix :

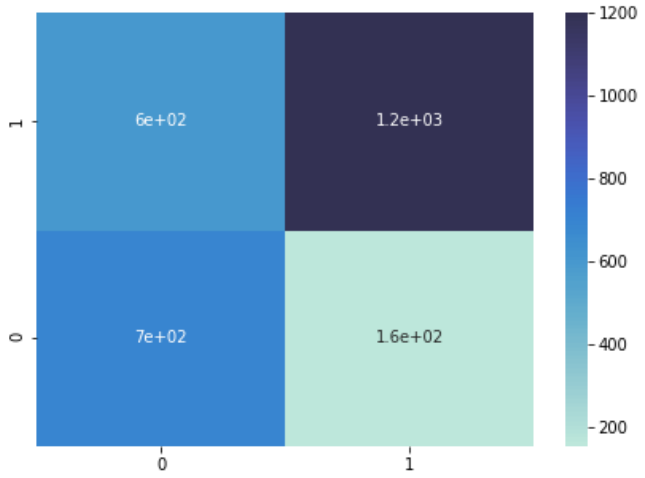
	precision	recall	f1-score	support
0	0.75	0.60	0.67	1301
1	0.68	0.81	0.74	1355
accuracy			0.71	2656
macro avg	0.72	0.70	0.70	2656
weighted avg	0.71	0.71	0.70	2656



The F1 score on our validation_texts was: 0.7389261744966442
The macro F1 score on our validation texts was : 0.7026534817594714

Complement Naïve Bayes classification report and confusion matrix:

	precision	recall	f1-score	support
0	0.75	0.62	0.68	1301
1	0.69	0.80	0.74	1355
accuracy			0.71	2656
macro avg	0.72	0.71	0.71	2656
weighted avg	0.72	0.71	0.71	265

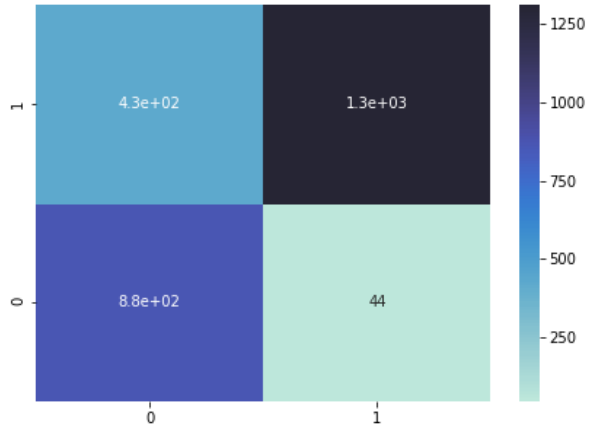


The F1 score on our validation_texts was: 0.7347498286497601
The macro F1 score on our validation texts was : 0.7057207789865342

I also tried many values for the hyperparameters “min_df” and “max_df” with (5, 7) char n-grams and obtained better results. The best result for Naïve Bayes was with min_df = 0.0006 and max_df = 0.1.

Multinomial Naïve Bayes classification report and confusion matrix:

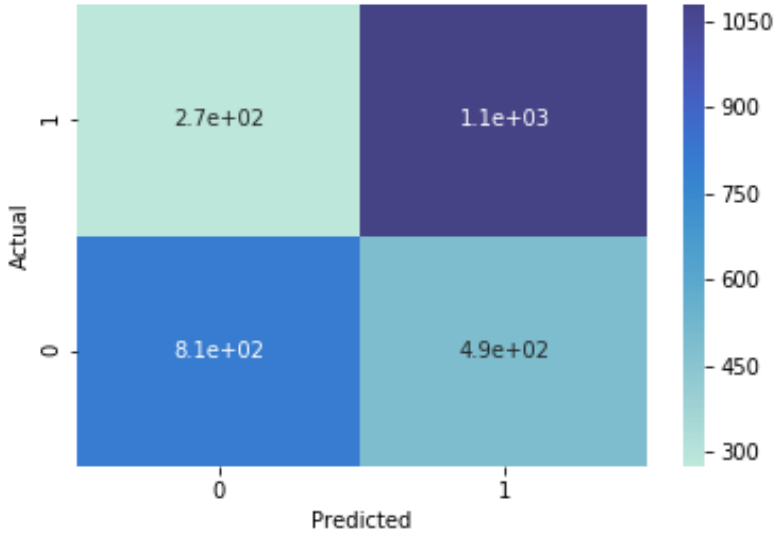
	precision	recall	f1-score	support
0	0.76	0.60	0.67	1301
1	0.68	0.81	0.74	1355
accuracy			0.71	2656
macro avg	0.72	0.71	0.71	2656
weighted avg	0.72	0.71	0.71	2656



The F1 score on our validation_texts was: 0.7421848739495799
The macro F1 score on our validation texts was : 0.7069931644031169

Complement Naïve Bayes classification report and confusion matrix:

	precision	recall	f1-score	support
0	0.82	0.54	0.65	1301
1	0.67	0.89	0.76	1355
accuracy			0.71	2656
macro avg	0.74	0.71	0.70	2656
weighted avg	0.74	0.71	0.71	2656

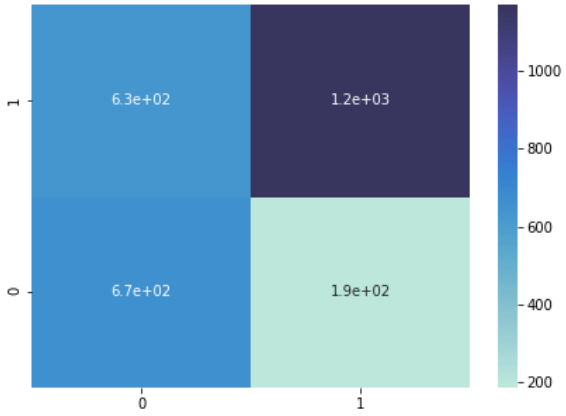


The F1 score on our validation_texts was: 0.7602153943617358
The macro F1 score on our validation texts was : 0.704469646136784

The third method is Support Vector Machine (SVM). SVM is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. The idea of SVM is simple: the algorithm creates a line or a hyperplane which separates the data into classes. According to the SVM algorithm we find the points closest to the line from both the classes. These points are called support vectors. Now, we compute the distance between the line and the support vectors. This distance is called the margin. Our goal is to maximize the margin. The hyperplane for which the margin is maximum is the optimal hyperplane. In this case I will use a SVM with an 'linear' kernel, a regularization parameter of 0.4 and a gamma set to auto value. This parameters were found by performing a GridSearch method from sklearn library.

SVM classification report and confusion matrix:

	precision	recall	f1-score	support
0	0.78	0.51	0.62	1301
1	0.65	0.86	0.74	1355
accuracy			0.69	2656
macro avg	0.72	0.69	0.68	2656
weighted avg	0.71	0.69	0.68	2656



The F1 score on our validation_texts was: 0.7408111533586819
The macro F1 score on our validation_texts was: 0.674983301338328

Furthermore, I implemented a vote function. Every model makes a prediction (votes) for each test instance and the final output prediction is the one that receives more than half of the votes. The three models above voted and the accuracy has improved.

In my opinion, the chosen way of vectorization is very important and it contributes increasing the accuracy of the model.

As a conclusion, most of the models performed acceptable taking in account the dataset with considerable difference of features between samples and the fine differences between samples of the two classes. Multinomial Naïve Bayes performed slight better with the right hyperparameters on the test samples. Complement Naïve Bayes was weaker on the test samples even if on the validation samples was better. The SVM approach did not have excellent results. Moreover, the voting helped some predictions to increase their score.