Machine Learning Challenge

Marco Bragoni, Jeroen Oskam, Paul Schreiber & Andras Tűű

Feature engineering

To begin, the datasets were thoroughly reviewed to see which variables would be most relevant for creating a good model. After doing research, it became clear that the text in the abstract was a good feature to use as a variable in the model. The assumption was that each author has a particular writing style and that each author writes on different subjects and thus uses different words in the abstract.

Before a model was created, the text in the abstract was pre-processed and cleaned. First, 'word_tokenizer' was used to extract tokens from the strings in the abstract, this is an important step of pattern recognition. The following cleaning steps were done by the removal of stop words and the use of lemmatization. Lemmatization is the process of converting words to its base form, so that they can be analysed as a single item.

Finally, a Multilabel Binarizer is fitted to the pre-processed abstract to transform textual information into numerical features. As of now, research could be done on a suitable learning algorithm.

Learning algorithm

The algorithm to apply to this data needs to be a classification algorithm. The goal is to identify authors of new observations based on the abstract. After trying several different algorithms, like Gaussian Naïve Bayes, K-Nearest Neighbours and Support Vector Machines, it became clear that Support Vector Machine was the best performing classification algorithm. The objective of the SVM algorithm is to fit the data and return a 'best-fit' hyperplane that categorises the data. In particular, the Support Vector Classifier (SVC) is implemented, meaning that the hyperplane categorises data linearly.

To apply the algorithm to the data, the training data is first divided into a training and validation set. After splitting the data, the model was trained on the training data and tested on the validation set. The score of the model was evaluated by calculating the accuracy of the predictions.

Hyperparameter tuning

The linear SVC has been set with default options.

Discussion

During this project, several learning algorithms were tested. All team members were simultaneously implementing a model using the same pre-processing steps (tokenization and lemmatization) to eventually achieve a good accuracy. On the one hand, a language model based on trigrams of POS tags was attempted to be built, but here only an accuracy of 1.4% was achieved. The same model based on word bigrams achieved a better accuracy (11.3%). On the other hand, a CountVectorizer method was attempted, only an accuracy of 3.7% was achieved with a Gaussian Naïve Bayes. Changing the CountVectorizer with the TF|IDF vectorizer increased the accuracy to 6%.

Eventually the Multilabel Binarizer in combination with the SVC algorithm achieved an accuracy of 11.7% on the validation data of the training dataset. Implementing the model on the test dataset and submitting the predictions to Codalab, returned an accuracy of 19.6%. This result is 4.6% above the provided baseline. With nearly 1 in 5 authors correctly predicted, it can be concluded that the model has a relatively good predictive performance. Future research could combine some of the different models to achieve an even better accuracy.

- The name of the account on CodaLab: andrastuu

Work specification

Marco	● Literature research ● Pre-processing data ● Creating language model with POS ● Applying classifiers on different models
Jeroen	● Literature research ● Creating model with CountVectorizer and TF|IDF ● Applying classifiers on different models ● Writing report
Paul	● Literature research ● Creating language model with POS ● Creating language model with bigrams ● Applying classifiers on different models
Andras	● Literature research ● Create Multilabel Binarizer ● Applying classifiers on different models ● Making final prediction

References

Linear SVC Machine learning SVM example with Python (n.d)
https://pythonprogramming.net/linear-svc-example-scikit-learn-svm-python/

Python – Lemmatization Approaches with Examples (2022, November 7)
https://www.geeksforgeeks.org/python-lemmatization-approaches-with-examples/

Li S. (2020, April 21) *Multi Label Text Classification with Scikit-Learn*
https://towardsdatascience.com/multi-label-text-classification-with-scikit-learn-30714b7819c5