In this challenge we were given two datasets, ‘train’ and ‘test’ with information about different papers. They included features like the abstract, title, venue and the year of publication. In the first dataset the author of each publication was included. In the second dataset, however, the author was excluded. We attempted to predict the author of each article from the second dataset by creating a machine learning model that will be explained through this report.

Methodology

As a starting point, exploratory data analysis consists of identifying and getting insights into the features. While doing so, we looked into different ways to engineer features to best suit our needs. As a result of this, we decided to use the abstract and the title as our main features.

We combined the title and the abstract into one feature and pre-processed it to extract new features from it. This process consisted of two parts. First, we pre-processed data with a function built by us called ‘lem’. This function was used to make the text lowercase, and to delete punctuation and stop words. Then, we used the lemmatizer from NLTK to bring the words to their meaningful base to make them more analyzable, so different words with the same base would be counted as the same word (GeeksforGeeks, 2022). This process was used in both the training and test dataset.

The pre-processed text features were tokenized using the count vectorizer from NLTK. We set the parameter max_features to 31,000 to reduce the dimensions and set ‘ngrams_range’ from 1 to 3 as part of our hyperparameter tuning. The variables ‘venue’ and ’year’ were one-hot encoded with handle_unkwon=’ignore’ parameter to include values without appearance in the test set with value “0”. Then, in both the training and test set, we used fit_transform and transform, respectively.

After testing other models, we decided to use MultinomialNB based on the results obtained while testing, and the suggestion of Kumar et al. (2021c), as they found that it outperformed other learning algorithms. Within the model, the only hyperparameter specified was alpha (0.01) to regularize the model and prevent the model from overfitting or underfitting (Pedregosa et al., 2011).

Conclusion

Our approach was focused on the likelihood of our features appearing in other abstracts; the accuracy of the model is 20.95%. This means for around 1367 papers the author was accurately predicted. Although this accuracy score might appear low instinctively, it is more representative of real world prediction. Therefore we do not consider this an unsatisfactory score. We are also taking into consideration that this is our first encounter with Natural Language Processing.

Codalab account name

Alex_rivera_

Work division

Anthonijsz, Romy	Research prediction models, investigate the SEA nmf model in particular, which did not prove to be fruitful due to its complexity.
Drenth, Joya	Research and attempt different prediction models, implement LDA however didn't manage to apply it to our dataset effectively. Test different hyperparameters.
Mitsas, Nikolas	Implement the data preprocessing , lemmatizer , count vectorizer and multinomialNB after research for other models too, like tf_idf with linearSVC and cosine similarity matrix with several algorithms which worked, but unfortunately not good enough to use it.
Rivera, Alejandro	Analyzing similar applications of text authentication in python, researching and trying to implement xgbooster to the model but didn't work for ours. Documentation and formatting.

Bibliography

GeeksforGeeks. (2022, 22 augustus). *Python | Lemmatization with NLTK*. https://www.geeksforgeeks.org/python-lemmatization-with-nltk/

Jarmul, K., Bowne-Anderson, H. & Roy, Y. (2022). *Introduction to Natural Language Processing in Python*. DataCamp. https://app.datacamp.com/learn/courses/introduction-to-natural-language-processing-in-python

Kumar, S., Sharma, A., Reddy, B. K., Sachan, S., Jain, V. & Singh, J. (2021c). An intelligent model based on integrated inverse document frequency and multinomial Naive Bayes for current affairs news categorisation. *International Journal of System Assurance Engineering and Management*, *13*(3), 1341–1355. Https://doi.org/10.1007/s13198-021-01471-7

Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

Misheva, V. & DataCamp. (2022). Sentiment Analysis in Python. Data Camp. https://www.datacamp.com/courses/sentiment-analysis-in-python

Banik, R. B. & DataCamp. (2022). Feature Engineering for NLP in Python. DataCamp. https://www.datacamp.com/courses/feature-engineering-for-nlp-in-python