Machine Learning Challenge - Group 5

Anne-Sophie Klijn
Leonard Sugg
Kursat Gonc
Emiel Verhoeven

Feature engineering

Firstly, we handled the missing values within the dataset. The feature *'editor'* misses over 95% of its' values, resulting in a covariance coverage that is too low. Therefore, we excluded this variable. Missing values within the remaining features were filled with the string *' '* to allow for text processing and vectorizing. To prevent bias in the model, duplicate records based on the *'title'* column were removed. Then, we computed the feature *'ENTRYTYPE'* as a dummy variable using one-hot-encoding, resulting in 3 new binary variables. The features *'title'*, *'abstract'*, *'publisher'* and *'author'*, are of textual nature. That's why we used the TF-IDF Vectorizer, which transforms text into numerical vectors while excluding high frequency words, allowing our models to work with text data (Scikit-learn, 2023; Stecanella, 2019). To streamline and automate this process for both models, we used a ColumnTransformer and put the ColumnTransformer in the pipeline for each model. Finally, the preprocessed dataset was split into X and y, containing all predictor variables and the publishing year, respectively.

Learning algorithms

We created an ensemble algorithm that consists of two regressors: the Ridge Regression and the Stochastic Gradient Descent. For combining the two predictions, we have used equal weights of 0.5 for each algorithm. The Ridge Regression had a MAE of 3.52 on the validation set, and the Stochastic Gradient Descent had a MAE of 3.41. When we combined the two predictions, the MAE decreased to 3.40 (Appendix). Even though we have tested Multinomial Naive Bayes - which seemed to work well - we have not included it in our predictions because a classifier is not suitable for a regression problem. Other algorithms like Lasso Regression or Random Forest Regression resulted in a higher MAE or were too computationally expensive.

Hyperparameter tuning

To tune our model hyperparameters, we used a grid search with a 5-fold cross validation. We split the training set into a training and validation set. Based on the MAE of the validation set, we have adjusted the parameters to output the lowest MAE. After validating, but before submitting to CodaLab, we have trained the model on the entire training set.

Discussion of the performance of your solution

The averaged MAE on our validation set was 3.40. In our submission to Codalab, it returned an MAE of 3.34. This means that our model generalizes well to the test data. For further improvements, overor undersampling should be tested since the distribution of year in the training data is skewed. Feature engineering such as word embeddings might have improved the performance of the model. However, we were not able to find a suitable dictionary. General dictionaries were too computationally expensive. An alternative for this would be to include the length of each value (length of the title, how many authors, etc).

Specification of contribution

Date	**Names**	**Contribution**	**Time spent**
13-11-2023	Leonard, Kursat, Anne-Sophie	Zoom meeting & make a plan	30 min
17-11-2023	Everyone	Data preparation, EDA, try first algorithm	6 hours
23-11-2023	Everyone	Tested KFold, testing different word embeddings for *title*	5 hours
29-11-2023	Everyone	Initialize 3 best algorithms, tune parameters and combine them	5 hours
30-11-2023	Anne-Sophie	Optimize Ridge Regression, write report	1 hour
30-11-2023	Leonard	Tuning LogisticRegression and SGDRegressor	3 hours
1-12-2023	Everyone	Choose final models, combine ensemble method	3 hours
1-12-2023	Leonard, Emiel, Anne-Sophie	Write report, test ensemble with Poisson algorithm	2 hours
6-12-2023	Everyone	Write report	2 hours

References

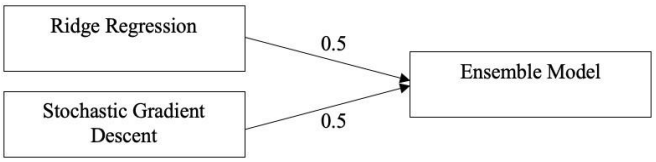
Scikit-learn developers. (Accessed 2023). *Comparing a HashingVectorizer to a DictVectorizer.* https://scikit-learn.org/stable/auto_examples/text/plot_hashing_vs_dict_vectorizer.html#sphx-glr-auto-examplestext-plot-hashing-vs-dict-vectorizer-py

Stecanella, B. (2019, May 10). *Understanding TF-ID: A simple introduction*. MonkeyLearn Blog. https://monkeylearn.com/blog/what-is-tf-idf/

Zach. (2022, June 1). *K-fold cross validation in Python (step-by-step)*. Statology. https://www.statology.org/k-fold-cross-validation-in-python/

Appendix

Figure 1: Visualization of the ensemble model



CodaLab Group_5