Prediction of year of publication

Feature engineering

Our code applied a range of feature engineering techniques to the original 6 features. We first combined editor and author into editor_author and extracted new features about the number of authors and the number of editors. From “edit_author” the textual features “title” and “publisher”, we extracted the word unigrams in the names, titles and abstracts, subsequently vectorising the first using count vectorizer and the rest with TF-IDF vectorizer, because for titles and abstracts the most frequent words often provide least information, which TF-IDF considers. For entrytype and publisher, we used one hot encoder for feature transformation as they are categorical.

Learning algorithm(s)

We have tried different learning algorithms,including Ridge/Lasso regression, RidgeCV , LassoCV ,RandomForestRegressor,Deep Neural Network, K-fold cross validation.etc.After many trials, we finally chose the learning algorithm based on RandomForestRegressor and on this basis, we applied feature selection and model selection. First ,we fit all the training data using the RandomForestRegressor model. Secondly,We implemented feature selection through SelectFromModel in ‘sklearn.model_selection’ library to get the new feature (X_new) that was well filtered (the shape of train data reduced from (65914,140589) to (65914,4893)). Thirdly , we implemented model selection through cross_val_score in ‘sklearn.model_selection’ library, to fit a new RandoForestRegressor with X_new and find the best model through cross validation.

Hyperparameter tuning

After the initial selection of the RandomForest model, we encountered computational challenges attributable to the high dimensionality of the transformed data. In response, we adjusted the hyperparameter ngram_range of the TfidfVectorizer from (1, 3) to (1, 1), effectively reducing the dimensionality of the data from 4,464,048 to 140,591.This modification facilitated more efficient model execution without compromising the essential information encoded in the data. Moreover, to enhance the predictive capabilities of the model, we tuned the hyperparameter n_estimators of the RandoForestRegressor from 50 to 100.

Discussion of the performance of your solution

The performance of our solution can be discussed in terms of the Mean Absolute Error (MAE) on both the training and testing sets, as well as the results from 5-fold cross-validation. The training MAE is 1.16, and the testing MAE is 3.22. We can see that there is overfitting of the model, but the testing MAE is still relatively low, suggesting that our model generalizes well and is able to make accurate predictions on new data. The final test MAE in Codalab is 3.08, which is consistent with the testing MAE and significantly lower than the baseline model’s MSE of 5.61. This indicates that our choice of algorithm and hyperparameter tuning is effective.

Group number: 34

Codalab account: Group34

Division of work:

Chunxi Zhao (SNR 2121768 ANR 938301): a) dataset exploration and feature extraction, responsible for cleaning the textual data and creation of the number of the editors and authors b) experiment with different ways of feature engineering, including CountVectoriser, Tfidfvectoriser, Glove c) experimented with various models including RidgeCV, RandomForest, XGBoost, LightBGM

Jinshan Su(SNR:2111602,ANR:168627):a) Test dataset exploration. b)Experiment with different models including RidgeCV, LassoCV, RandomforesRegressor and Bert transfer learning and also apply hyperparameter tuning . c)Explore the feature selection approach and eventually apply SelectFromModel in our learning algorithm.

Kexin He(SNR:2110423, ANR:707986): a) Test dataset exploration. b)Experiment with different models including RandomforestRegressor, XGBoost and linearSVC and also apply hyperparameter tuning . c)Explore the feature selection approach and apply K-fold cross validation in our learning algorithm.

Yujia Xie(SNR: 2109369, ANR: 886344): a) Testing baseline model. b) Language recognition function. c) Testing different models (RandomforesRegressor and RandomizedSearchCV) and apply hyperparameter tuning.

References or appendices

Article on ColumnTransformer method:
https://scikit-learn.org/stable/modules/generated/sklearn.compose.ColumnTransformer.html
Article on Feature selection method:
https://scikit-learn.org/stable/modules/feature_selection.html#feature-selection

Article on TfidfVectorizer method:
https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
Article on RandomForestRegressor method:
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
Article on Natural Language Processing with Python:
https://www.dataquest.io/blog/natural-language-processing-with-python/

*ChatGPT provided guidance on technical aspects during the project