Predicting the Year of Publication for Scientific Papers

Preprocessing: Missing data was handled with the SimpleImputer from sklearn, with default parameters and strategy=‘constant’, as ‘most frequent’ yielded worse results. Predicting missing 'publisher' data was attempted with RandomForestClassifier (accuracy 0.72) but yielded lower model performance than the 'constant' imputation strategy. The 'editor' feature (97.76% missing values) was dropped, and duplicate rows were removed based on 'title' feature. Assuming a regression problem, 'year' was cast to int64, and other features cast from lists to strings. Outliers in the 'year' feature were identified through boxplot inspection, but kept in the training data as removing them caused overfitting. No influential leverage points were found using Cook's distance.

Feature engineering: Experimented with encoding vectorizers: TFidfVectorizer, CountVectorizer and HashingVectorizer from sklearn; the latter being the one selected. After manually testing different values for ngram_range the chosen values were: ngram_range of (1, 2) for ‘title’, ‘publisher’, ‘author’; ngram_range of (1,1) for ‘ENTRYTYPE’ and ‘abstract’. Experimented with removing the most/ least frequent words in ‘abstract’ and ‘title’, however not implemented due to being highly computationally expensive. Attempted removing common stop words with minimal impact on MAE, therefore also not used. Data was split into 80% training, 20% validation set (better than 25%) and recombined after model evaluation. The 'year' feature was clipped at max=2023. Predictions were cast from float to int64.

Learning algorithms: Ridge Regression resulted in the lowest MAE on the validation set (=3.2) as compared the all other models (Lasso Regression, Support Vector Regression, ElasticNet Regression, Decision Tree Regression, RidgeCV, Stochastic Gradient Descent, Gradient Boosting Regressor, AdaBoostRegressor, BaggingRegressor, MLPRegressor, BaggingRegressor (SVR), Random Forest Regressor), considering computational and memory constraints. RandomForestRegressor was one of the best model candidates, different parameter values were tested: n_estimators=10 and max_depth=40 provided the best results (validation MAE=3.55). Exploring larger values for parameters exceeded computational capabilities, addressed by TruncatedSVD for dimensional decomposition before testing Random Forest (with default parameters) and MLP (with default hidden layers, and with 2 and 3 layers). Memory constraints only allowed n_components=90 for TruncatedSVD. The above experiments still led to MAE higher than that of the Ridge model.

Hyperparameter tuning: RandomizedSearchCV was selected over the more computationally expensive GridSearch for finding the best solver and alpha parameters, still leading to the default parameters.

Discussion on performance of solution: After experimentation with different prediction models the Ridge Regression proved to be simple, computationally accessible and fast, as well as resulting in the lowest MAE score.

References

https://scikit-learn.org/stable/index.html