Introduction:

This report details the computational learning experiments conducted for predicting the publication year of scientific papers based on their metadata. The process involves feature engineering, the implementation of a learning algorithm and a discussion on the performance of the developed solution.

Feature Engineering:

1. **Data Preprocessing:**

-Preprocess textual columns by lowercasing them, transforming ENTRYTYPE column to categorical.

-In line with tests performed by applying different learning algorithms, feature ablation analysis demonstrated that the **'abstract'**, **'publisher'** and **'title'** are crucial features. Where **'author'** and **'ENTRYTYPE'** have less impact though we still keep them.

-Build new composite features such as **'title_publisher_low'** with the goal to enhance the model's performance, as well used to perform further data enrichments.

-Replace Null values with empty strings

-Detect text language using langdetect library. To validate the detected language function performed on col: **'lang_tit_publ_low'**, **'title_low'**, **'abstract_low'.** Best result obtained using col: **'title_publisher_low'**

-Extract number of authors per instance '**author_number'.**

-Split the dataset into training and validation sets, with stratification based on **'year'**, making sure to preserve a representative distribution of data from different years.

2. **Topic Modeling:**

-With the aim to better classify the different instances into distinct topics, we implemented Latent Direchlet Allocation (LDA) using col. **'title_low'** focusing on the instances labeled as English under **'lang_tit_publ_low'.** Performed lemmatization of the title_low column and generated bigrams and trigrams. Applied Term Frequency-Inverse Document Frequency (**TF-IDF**) vectorization aiming in removing not relevant most frequent words.

-LDA baseline model selected 10 topics that we see not overlapping between each other. (Figure 1)

3. **TF-IDF Transformation:**

Applied Term Frequency-Inverse Document Frequency (TF-IDF) vectorization separately to the

'title_low', 'publisher_low', 'abstract_low' columns, while OneHotEncoder was applied to the categorical 'ENTRYTYPE', 'lda_topic', 'lang_tit_publ_low'.

Learning Algorithm:

Having tested different linear models Lasso, LinearRegression. We selected Ridge model, mainly due to Its interpretability and performance being a resource efficient algorithm compared to other models like Random Forest Regression.

Hyperparameter Tuning:

We use the standard Ridge algorithm from Scikit-learn library with all the default parameters.

Performance observation:

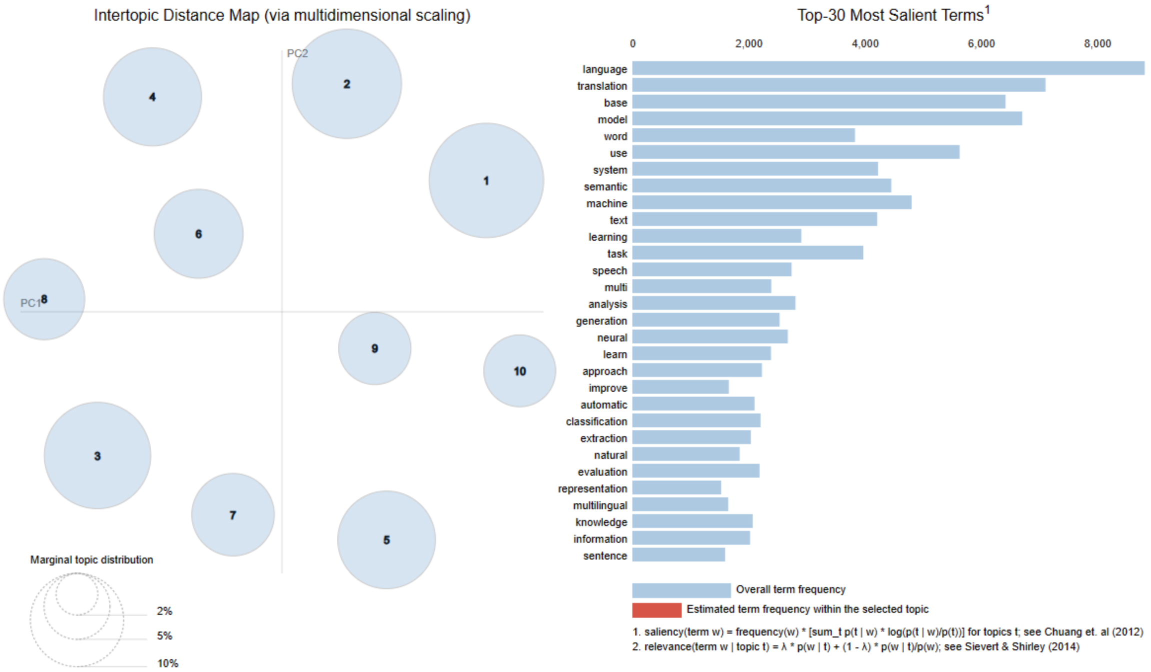
Training process: model was done on 75% of the train data with a split of 25% for validation, stratifying based on ‘year’ to ensure that the distribution of the target variable is similar in both the training and testing sets. Keeping a resource efficient approach where the model can be trained in a few minutes on any standard modern device (4GB RAM).

Model Evaluation: We performed the model evaluation on the validation set using Mean Absolute Error (MAE) with a reported value of 3.90.

Discussion:

The model developed provided a reasonable performance on predicting the target variable based on our results from the validation set without compromising the explainability of the results and being resource efficient. With potential room for improvement in example having a Subject Matter Expert (SME) on the content of the papers a better LDA topic model could be achieved to support the predictiveness of the model together with further hyperparameter tunning. In case we would not focus on the efficiency and further explainability of the model outcome, using non linear models such as Random Forest Regression, NN would provide better results as was proven with our final model.

Figure 1: LDA Topic model



References:

scikit-learn. (n.d.). Ridge regression. scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html#sklearn.linear_model.Ridge