Word Complexity Estimation

This paper explains the process of predicting how difficult words will be for native and non-native speakers, using regression forms which is a set of statistical processes for estimating the associations between dependent variables “outputs” and independent variables “inputs” which called “predictors or features”. This is the difference between regression and classification, as the first is used to predict some values and the other is to classify known and previously explained values.

The task is about guessing how much a specific sentence or word is tricky to understand for people either they are native or not, and the complexity is calculated by adding the number of annotators who marked the word as difficult and divide it by the total number of annotators.

In this case, there are two approaches to get the ratio of difficulty, either using the full test or just depending on the target word, of course after cleaning, vectorizing and making some preprocessing before starting. There are some beliefs that reading the entire text can facilitate the process of understanding words and content in general instead of reading just one sentence, but the results show the opposite, as the percentage was lower in the second method than in the first.

Building the model require several libraries like Scikit Learn , Numpy , Pandas etc.

The implementation at the beginning was on the train dataset in case it contains needed labels to enable to estimate the test file labels.

Different regression models are used like simple linear regression, multiple, polynomial and others. As it is shown in the code file each form of regression gives dissimilar results and thus the value of mean absolute error vary from one to another. The best results are gotten from SVR model, the second one is random forest regression model.

The forms description and hyperparameters were used are presented below with the value of mean absolute error “MAE” in each one:

1. **Simple linear regression**

   It is a linear regression model with a single explanatory variable. That is, it concerns two-dimensional sample points with one independent variable and one dependent variable.

   Hyperparameters used was the default ones (fit_intercept=True, normalize='deprecated', copy_X=True, n_jobs=None, positive=True).

2. **Multiple linear regression**

   It differs from simple linear in case it uses two or more independent variables to predict the dependent variable, but in this case the results are similar to the previous model since just one variable used to predict one variable.

3. **Polynomial regression**

   It is a special case of linear regression where we fit a polynomial equation on the data with a curvilinear relationship between the target variable and the independent variables. It is not used in the model.

4. **Support vector regression**

   It creates a hyperplane or set of hyperplanes in a high-dimensional space, which can be used for regression and classification.

   Hyperparameters: (kernel = 'rbf', epsilon = 0.0005, gamma=500.0). and the rest are default.

5. **Decision tree regression**

   It is a supervised machine learning algorithm that can be used for both classification and regression problems. A decision tree is simply a series of sequential decisions made to reach a specific result.

   Hyperparameters: (ccp_alpha=0.000001, random_state=100, min_samples_leaf=40, min_samples_split=10, splitter='best').

6. **Random forest regression**

   The Random Forest Algorithm combines the output of multiple (randomly created) Decision Trees to generate the final output.

   Hyperparameters: (n_estimators = 7000, random_state = 1000), the others are still on default.

The resulting mean absolute error value in a 5-fold cross-validation is shown in the table below:

Regression Model	MAE value
1- Simple linear regression	0.114034718159207
2- Multiple linear regression	0.114034718159207
3- Polynomial regression	Not used
4- Support vector regression	0.077516271892923
5- Decision tree regression	0.096512221993151
6- Random forest regression	0.094013953562240

The table shows that the linear model is the worst one in this case, it gives 0.11 which is the highest value between all the others.

Cross-validation

Sometimes called rotation estimation or out-of-sample testing, is any of various similar model validation techniques for assessing how the results of a statistical analysis will generalize to an independent data set. Cross-validation is a resampling method that uses different portions of the data to test and train a model on different iterations. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice.

The 5-fold cross-validation was implemented on all models, it gave close results and it is represented in the table below.

Regression Model	MAE value
1- Simple linear regression	0.11246484966589083
2- Multiple linear regression	Not Counted
3- Polynomial regression	Not used
4- Support vector regression	0.076240738170416
5- Decision tree regression	0.09424268189589007
6- Random forest regression	0.0916650899158791