**Kaggle Project 10 – Feedback Prize English Language Learning**

**Prepared by: Dwaipayan Mukherjee (2211569, AIA – AIML)**

**About the Project:**

The dataset presented here (the ELLIPSE corpus) comprises argumentative essays written by 8th-12th grade English Language Learners (ELLs). The essays have been scored according to six analytic measures: **cohesion**, **syntax**, **vocabulary**, **phraseology**, **grammar**, and **conventions**. Each measure represents a component of proficiency in essay writing, with greater scores corresponding to greater proficiency in that measure. The scores range from 1.0 to 5.0 in increments of 0.5. Our task is to predict the score of each of the six measures for the essays given in the test set.

**Steps that have been followed:**

1. Necessary Imports:

The necessary imports of basic libraries like NumPy, Pandas, Matplotlib and Seaborn are done.

1. Understanding the given data:

The train data has been loaded into Pandas dataframe to understand it and work on it. The attributes on which each essay is scored has been taken into a list.

1. Exploratory Data Analysis:

First, the distribution of each of the attributes in the data has been printed out using histograms.

Next, the median, standard deviation and outliers of the attributes are explored using boxplots.

1. Next the **nltk** library is imported and stopwords are downloaded. The English stopwords are downloaded from the stopwords corpus.
2. Three new columns are created, namely, **tokens**, **text\_len** and **stop\_words\_len**. The tokens column is created using nltk tokenizer on the full text. The text\_len column is the length of tokens and the stop\_words\_len column is the length of stopwords in the full text.
3. A heatmap is created on the attributes as well as the text\_len and stop\_words\_len columns to explore the correlation.
4. The TF-IDF Vectorizer is imported for feature extraction. FeatureUnion is imported for building the pipeline and Decision Tree Regressor is imported for the model.
5. The feature and the targets are separated in the train dataset.
6. Since we are evaluating the essays, I think it could be reasonable to introduce ngram\_range to TfidfVectorizer. After visualizing the decision tree from the previous versions, I realize that the decision tree mostly split on stop words. However, it's not reasonable to remove stop words because we are evaluating essays.

To resolve this, I would use 2 feature extractors:

1. TD-IDF extractor with ngram\_range=(3,6), which can be helpful for grammar or cohesion classes
2. TD-IDF extractor with stop words extraction, which helps introduce more information in the decision tree
3. The Decision Tree Regressor is used for the training purpose.
4. Next, we load the test data and then use the feature extractor pipeline.
5. The trained model is used to predict on the small test dataset.