ASSIGNMENT: NLP based problem to estimate the difficulty level of kid's stories.

Dataset Insights:

The dataset contains 556 diverse .txt files which include empty files, duplicates, and some biased files. Word length ranges from 0 to 2000+ per document.

Exploratory Data Analysis:

- It shows that the data is very diverse and most of the documents between are 40 to 800 words per document.
- More than 50% of the words are stopwords so, removing stopwords can give more strength to n-grams.
- Total word count without stopword (**328989**), and with stopwords(**172752**)
- 3 random samples help us to understand whether the data distribution is the same
- In this problem statement, stopword removal works better than lemmitization. As lemmitization returns the base word, this can reduce the difficulty weightage in the document.
- TF-IDF helps to find the word weightage, and it is a good feature extraction technique to find the word difficulty or word importance
- The distance matrix is used to find the duplicates and closest documents
- The distance matrix also shows that the documents can be clustered as the features extracted by TF-IDF shown the closeness in documents
- Nearly 100 documents are considered to be outliers which are basically duplicates, and more or less than 5% of data

Cleaning:

- From the inference of EDA. It is clearly shown that ~100 documents are considered to be outliers
- Removed outlier documents from the data for better modeling

Training:

- **Baseline model:**
  - All the TF-IDF extracted features are considered in the baseline model (with stopwords)
  - Plotting shows that the features are evenly distributed but the predictions are not that great
- **Model V2 and V3:**
  - V2 and V3 are quite similar but little better than Baseline. In these n-gram range is increased, and features are more focused on the complexity of documents
  - V3 performance was better than baseline and can be used for predictions
- **Model V4:**
  - In V4 stopwords removed and the performance is not better than V3
  - V4 predictions show that the predictions are more word difficulty oriented

    - The drawback is 2 3 clusters have the same center, which is mixing up the predictions
- **Model V5:**
    - Features extracted by TF-IDF are mostly zeros so the maximum features are zeros which might be showing their dominancy over other features
    - V5 is the worse model which is completely divided into 2 clusters
- **Model V6:**
    - V6 is quite similar to V5 but it is having n-gram range (1-3) and n_components reduced to 50
    - 50 PCA components worked better than 300
    - In this model understood the difficulty level of words but still not able to make cluster based on document length
- **Model V7:**
    - V6 and V7 are the finest models
    - In V7 length features also introduced which tries to maintain the length and document difficulty but end up showing the trade-off between length and difficulty level of documents

Summary:

Hyperparameter tuning can help in reducing trade-off between length and difficulty level of documents

In these models, TF-IDF features are basically works well for document difficulty and POS tags and length which is more focused on word length of documents. By resolving trade-off which might need more time and more dataset can give remarkable performance

Some of the documents were very noisy

Future work:

- By increasing dataset model V7 might show better performance
- We can try using average word embedding to give more weightage to the difficulty level as we can see V7 is slightly dominant towards the length and size
- Semi-Supervised learning can help in increasing performance but it is little complex to write semi-supervised model (lot of manual efforts needed)
- Instead of PCA, we can try using other dimensionality reduction techniques which might give better results like autoencoders, matrix factorization, etc