Text-Summarization Report

Name- Pranshu Kala

Data Collection

- **Dataset Chosen:** Inshorts News Summary Dataset
- **Purpose:** The dataset was selected for its suitability to be trained on available hardware.
- **Loading Status:** Successfully loaded the selected dataset and saved it in the Data Folder.

Data Preprocessing and Exploratory Data Analysis (EDA)

Data Extraction

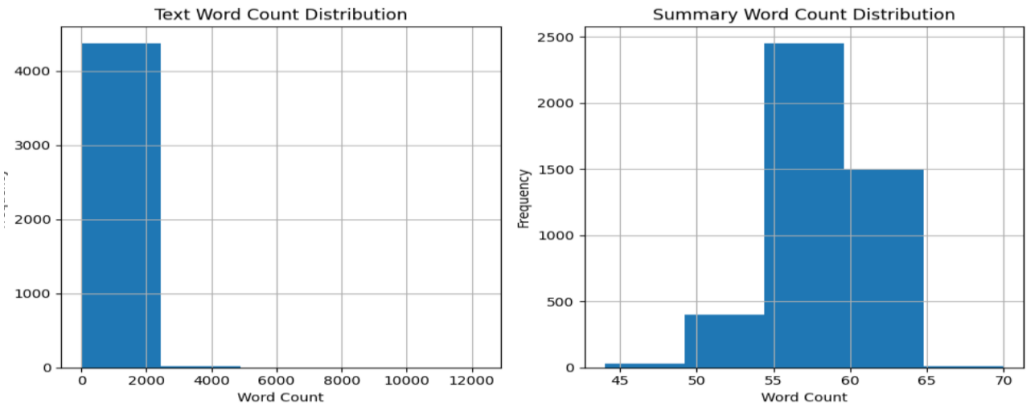
- **Columns Extracted:** Extracted the required columns from the master dataset to focus on input text and target summaries.

NLP Preprocessing

- **Techniques Applied:** Applied various NLP preprocessing techniques such as:
  - Tokenization
  - Lemmatization
  - Removing stopwords, punctuation, special characters, and white spaces

Data Visualization

- **Visualization:** Visualized the data distribution between the input and target columns to understand the dataset characteristics better.



Data Saving

- **Saved Preprocessed Data:** The preprocessed dataset was saved for further processing and model training.

Model Building

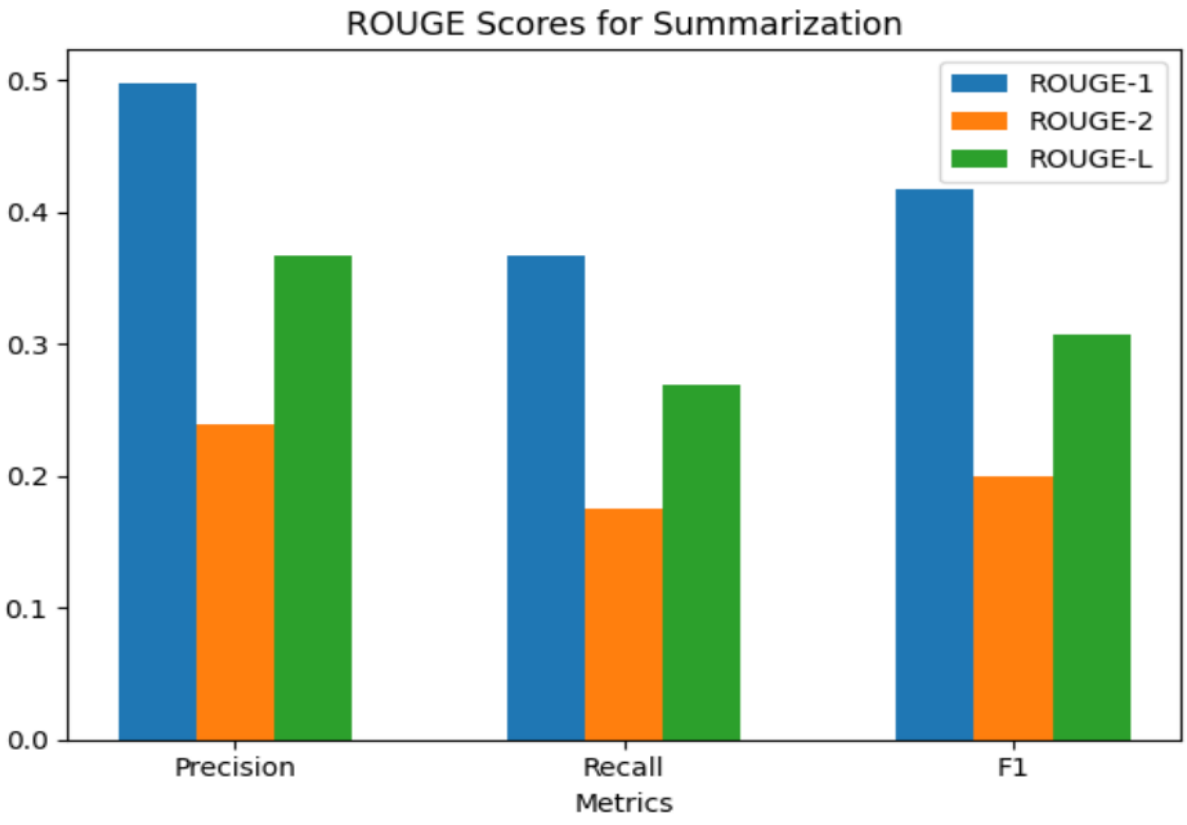
Abstractive Summarization

Model Selection

- **Model Chosen:** T5-small model for abstractive summarization due to its efficiency and capability in handling such tasks.

Pre-fine Tuning

- **Initial Results:** Pre-fine tuning results were stored in `Summarization_Model/model.ipynb`.



Fine Tuning

- **Process:** Fine-tuned the T5-small model on the selected dataset.
- **Evaluation:** Evaluated the fine-tuned model's performance and stored the results in `Summarization_Model/evaluation.ipynb`.

In [15]:
```
#final ROUGE scores for the model

rouge_calc(pred, list(test['target_text']))
```

Out[15]:
```
{'Rouge_1': 0.4200905043366553,
 'Rouge_2': 0.2237638177289328,
 'Rouge_L': 0.3491328918546062}
```

Model Size

- **Final Model Size:** Approximately 800 MB after fine-tuning.

Extractive Summarization

Preprocessing

- **Techniques Applied:** Performed extractive summarization preprocessing, which included:
    - Removing stopwords, punctuation, special characters, and white spaces

Algorithm Selection

- **Algorithm Chosen:** Text Rank Algorithm, implemented through the `py-summa` library.

Initial Results

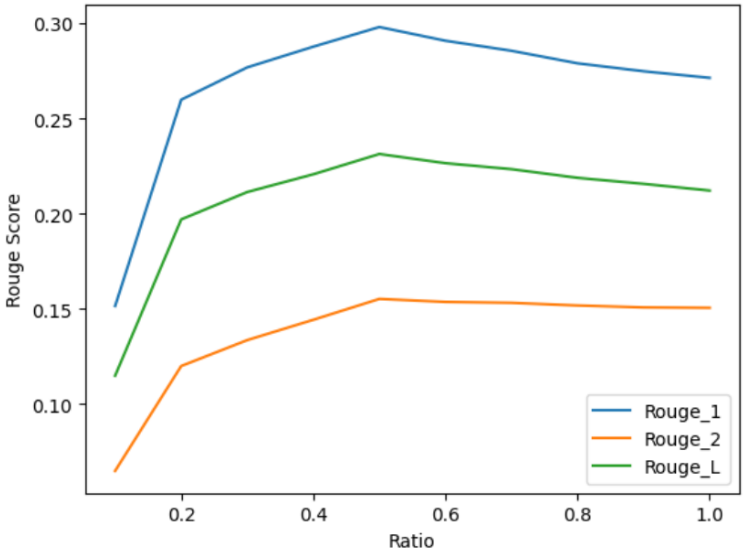
```
rouge_calc(list(df['generated_summary']), list(df['summary']))
```

```
{'Rouge_1': 0.2598418492010122,
 'Rouge_2': 0.12002228311211788,
 'Rouge_L': 0.19699924793967336}
```

Optimization

- **Optimization Process:** Optimized the summary-to-input text length ratio to achieve better summarization results.

```
# we find that the best ratio is between 0.2 to 0.4  for this dataset after that it declines
```



Model Interface

Implementation

- **Library Used:** Streamlit library was used to build the model interface.
- **Implementation Details:** Implemented the interface in `Model_Interface/app.py`.

Visualization

- **Results Visualization:** Visualized both abstractive and extractive summarization results on the interface to compare their effectiveness and usability.