Exploratory Data Analysis: The goal was to understand and develop intuition about each feature. The following steps are taken -

1. Each feature is analyzed using descriptive statistics and visualized through distribution plots.
2. The relationship between each feature and the target variable 'Impact' is explored to assess predictive power.
3. Missing values, outliers, sparsity and anomalies are identified to assess data issues and potential preprocessing needs.

Data Pre-processing:

1. **Text Cleaning -** Standardized the text data by removing punctuation, special characters, and stop words. The text was lemmatized to reduce words to their base form. Missing values were backfilled with relevant data.
2. **Feature Engineering -** Created new features, including the title and description length, authors count, and the year and month of publication and identified the primary author based on the highest average impact.
3. **Word2Vec Embeddings-** Generated Word2Vec embeddings for the cleaned title and description (combined) and added these embeddings as new features to the dataset.
4. **Encoding-** Categorical features such as publisher, category, and main author were encoded using label encoding.
5. **Feature Selection-** Performed correlation analysis and used feature importance techniques to remove redundant and weak features, improving the model's performance.

Modelling:

1. **Modelling Choice -** Different techniques were tried as described below-
    a. **CatBoost Regressor -** Chosen for efficiently handling high-cardinality categorical data through ordered label encoding and symmetric tree formation, which helps reduce overfitting.
    b. **Multi-layer perceptron -** The main idea behind using an MLP architecture was to leverage the embedding layer for categorical data and capture non-linear relationships in the data.
    c. **Spark GBDT regressor -** Developed a Spark app using the GBDT regressor model -
        i. The app includes functionality for loading data, preprocessing, and performing cross-validation to evaluate the model's performance.
        ii. It also allows experimentation with different numbers of worker cores to observe their impact on training time and model performance.
        iii. However, the full data preprocessing pipeline couldn't be implemented in Spark due to issues with Word2Vec, which couldn't be resolved in the given time.

iv. Tried better algorithms like CatBoost and LightGBM in PySpark, but these libraries don't have native support in PySpark, and their external integrations caused compatibility issues that couldn’t be resolved within the timeframe.

v. To avoid memory issues in the Spark app, the data was sampled by 50%.

vi. There was no significant difference in training time when the number of workers varied, possibly due to issues with the Spark setup on the local system.

Key Improvement Suggestions:

1. **BERT-based Embeddings**- Utilize BERT or other transformer-based models to generate more contextually rich embeddings for the title and description, which can capture deeper semantic relationships than traditional methods like Word2Vec.
2. **Feature Expansion**- Incorporate additional features such as sales data, user reviews, book language, book format, author popularity, and publication house reputation. These features could enhance the model's predictive performance by providing more informative inputs.