Repository link: https://github.com/DonCheetos/DS_Group_B4

Fanfiction books prediction model

Task2:

Identifying Your Business Goals

Background
Fanfiction websites such as https://fanfiction.net host vast collections of user-contributed stories, offering basic categorization such as genres, character tags, and pairings. Despite this, the overwhelming volume of content poses a challenge for users to find high-quality or relevant stories quickly. This lack of efficient discovery mechanisms results in user dissatisfaction and reduced engagement. By leveraging advanced metadata analysis, machine learning, and text processing techniques, it is possible to enhance content discoverability.

Business Goals

1. **Advanced Tagging Functionality**: Create a tagging system that can automatically classify stories into detailed genres and categories based on metadata, summaries, and other attributes.
2. **Story Recommendation System**: Build a system capable of recommending similar stories based on user-selected inputs, utilizing metadata and content analysis.
3. **Improved User Experience**: Reduce the time and effort users need to find interesting stories, increasing their satisfaction.

Business Success Criteria

- **Accurate Tagging**: The system should correctly classify stories into genres and categories with a high degree of accuracy.
- **Relevant Recommendations**: The recommendation system should suggest stories that align well with user preferences, evaluated through similarity metrics.

Assessing Your Situation

Inventory of Resources

- **Data**:
  - 6.7GB of metadata containing attributes such as `Title`, `Summary`, `Category`, `Genre`, `word_count`, `Rating`, etc.
  - 111.5GB of story content for text analysis.
- **Hardware**: Two PCs available for data processing and model development.

Requirements, Assumptions, and Constraints

- **Requirements**:
  - High processing power and sufficient storage to handle the large dataset.
  - Advanced text-processing techniques for analyzing story summaries and metadata.
- **Assumptions**:
  - Metadata is accurate, complete, and contains meaningful patterns that can inform tagging and recommendations.
  - The dataset includes a diverse representation of stories across genres and categories.
- **Constraints**:
  - Computational limitations may require cloud-based resources or distributed processing.
  - Randomness or ambiguity in the data (e.g., unclear or overlapping categories) may limit model accuracy.
  - Limited development time may constrain the scope of initial features.

Risks and Contingencies

1. **Large Number of Categories and Genres**:
   - **Risk**: High memory and processing requirements for analyzing numerous categories.
   - **Contingency**: Process data incrementally by focusing on one category or genre at a time. Filter excess data to reduce processing load.
2. **Ambiguity in Categories and Genres**:
   - **Risk**: Stories may fit into multiple categories, making clear classification difficult.
   - **Contingency**: Implement multi-label classification models to handle overlapping categories and define similarity thresholds for determining dominant genres.
3. **Incomplete or Incorrect Metadata**:
   - **Risk**: Metadata errors may reduce model accuracy.
   - **Contingency**: Apply data-cleaning methods to handle missing or erroneous metadata. Use textual summaries as a fallback for classification.

Terminology

- **Metadata**: Information about other data, such as `Title`, `Summary`, `word_count`, and `Rating`.
- **Multi-label Classification**: A machine learning task where multiple labels (e.g., genres) are assigned to a single data point (e.g., a story).

Costs and Benefits

- **Costs**:
  - Time required for development and data preprocessing.

    - Computing power for training models and handling large-scale data.
- **Benefits**:
    - Reduced manual effort in tagging stories.
    - Enhanced user satisfaction through easier story discovery.

Defining Your Data-Mining Goals

Data-Mining Goals

1. **Extract Insights from Metadata**: Identify patterns in metadata to understand popular genres, categories, and story attributes.
2. **Classify Stories**: Develop machine learning models to categorize stories into appropriate genres and categories using metadata, textual summaries, and other features.
3. **Build Recommendation System**: Create a recommendation model to suggest similar stories based on content and metadata analysis.

Data-Mining Success Criteria

- **Tagging Accuracy**: The model should achieve high precision and recall in assigning genres and categories.
- **Recommendation Relevance**: The system should produce highly relevant recommendations as evaluated by user feedback or similarity metrics.
- **Correlation Analysis**: Demonstrate well-defined correlations between story attributes (e.g., `word_count`, `Rating`, `Summary`) and genres/categories.

Task3

Gathering Data

Data Requirements

To achieve the business and data-mining goals, we need data that supports accurate tagging, categorization, and recommendations. Specifically:

- **Metadata**:
  - Attributes: `Title`, `Summary`, `Category`, `Genre`, `word_count`, `chapter_count`, `Age rating`, `Published`, `Updated`, `Status`, `Language`.
  - Usage: Essential for training classification models, understanding patterns, and generating recommendations.
- **Story Content**:
  - Full text of stories.
  - Usage: Enables advanced text processing and similarity-based recommendations.

Data Availability

- **Metadata**: 6.7GB of structured metadata is readily available, providing detailed attributes for each story.
- **Story Content**: 111.5GB of unstructured text data is available for analysis. Text files include summaries, chapters, and full story content.

Selection Criteria

- **Relevance**: Data must contribute directly to tagging, categorization, or recommendation tasks. Attributes such as `word_count`, `Summary`, and `Category` are prioritized.
- **Completeness**: Only records with non-missing critical fields (`Summary`, `word_count`, `Category`, `Genre`) are selected.
- **Usability**: Data with meaningful patterns and correct labeling will be preferred. Duplicate or irrelevant records (e.g., empty stories) will be excluded.

Describing Data

The dataset includes:

1. **Metadata Table**:
   - **Size**: 6.7GB.
   - **Structure**: Tabular data with 15+ fields.
   - **Fields**:
     - **`Title`**: Story title, textual.

        - **`Summary`**: Story summary, unstructured text.
        - **`Category`**: Primary category of the story (e.g., "Movies", "Games"), categorical.
        - **`Genre`**: One or more genres (e.g., "Romance", "Sci-Fi"), multi-label categorical.
        - **`word_count`**: Total word count, numerical.
        - **`chapter_count`**: Total number of chapters, numerical.
        - **`Age rating`**: Story rating by users, numerical or categorical.
        - **`Published`**, **`Updated`**: Datetime fields.
        - **`Language`**: Language of the story, categorical.
        - **`Status`**: Completion status (e.g., "Complete", "In Progress"), categorical.
        - **`Story URL`**, **`Author URL`**: Links to story and author profiles.
2. **Story Content Files**:
    - **Size**: 111.5GB.
    - **Structure**: Unstructured text files.
    - **Content**:
        - Full story content, split into chapters.
        - Useful for deeper textual analysis (e.g., NLP).

Exploring Data

Exploratory Data Analysis (EDA)
Initial exploration focuses on understanding the structure and variability of data:

1. **Distributions**:
    - **`word_count`**: Range varies significantly (e.g., from flash fiction to full-length novels). Median and mean values provide insights into typical story lengths.
    - **`chapter_count`**: Stories range from one-shot to multi-chapter sagas.
2. **Text Analysis**:
    - Summaries: Analyzed for average word count, common phrases, and variability.
    - Genres and Categories: Checked for frequency and overlap, identifying dominant combinations.
3. **Date Fields**:
    - Analyzed the `Published` and `Updated` fields for temporal trends (e.g., peak publication periods).
4. **Correlations**:
    - Early analysis identifies potential relationships between `word_count`, `chapter_count` and `Age rating`.

Initial Findings:

- Some categories and genres are highly imbalanced, with a few being overrepresented.
- Metadata fields like `Chapters` and `word_count` exhibit patterns that can inform models.
- Text content includes noise (e.g., user notes, unrelated content).

Verifying Data Quality

Quality Checks Performed:

1. **Completeness**:
    - Missing values in critical fields (e.g., `word_count`, `Summary`) identified and logged.
    - Stories with empty summaries or categories flagged for removal or imputation.
2. **Consistency**:
    - Checked for duplicate records based on `Title`, `Author`, and `Published` date.
    - Ensured uniform formatting of datetime fields (`Published`, `Updated`).
3. **Relevance**:
    - Verified alignment of `Category` and `Genre` with story content (e.g., no "Sci-Fi" tag on non-sci-fi stories).
4. **Balance**:
    - Analyzed genre distribution to ensure models are not biased toward overrepresented classes.

Issues Identified:

- **Missing Data**: 652658 (9.7%) entries lack values. Imputation strategies will be explored for these but `Genre` is essential for training and may result in dropping incomplete records.
- **Imbalance**: Certain categories(e.g., "Harry Potter") dominate, requiring oversampling or weighted loss functions during training.
- **Text Noise**: Summaries include unrelated information (e.g., author notes or disclaimers), requiring preprocessing.

Planned Resolutions:

- Delete unusable records.
- Use advanced text-cleaning techniques to preprocess summaries and story content.
- Normalize numerical attributes (e.g., `word_count`, `chapter_count`) for model input.

Task 4.

Planning your project

Tasks (Corresponding numbers of tasks)	Expected time cost by hours (All Members)
1.	12
2.	10
3.	20
4.	12
5.	10
Total	64

Project Plan

1. **Data Cleaning and Preparation**
    - **Description**:
        - Handle missing values (e.g., remove rows with empty or invalid values for critical fields like `Summary` and `Category`).
        - Remove infrequent outliers that might skew the analysis.
        - Standardize text fields (e.g., lowercase, remove punctuation) and handle noise in metadata.
    - **Methods and Tools**:
        - Python (Pandas, NumPy for preprocessing), Seaborn/Matplotlib for data visualization.
        - Regex for text cleaning.
    - **Time Allocation**:
        - Kaur Lõhmus: 6 hours
        - Maksim Kelus: 6 hours
2. **Data Selection and Feature Engineering**
    - **Description**:
        - Exclude irrelevant fields (e.g., `Publisher`) that do not contribute to predictions.
        - Engineer new features like text embeddings (Bag of Words, TF-IDF, or Word2Vec).
    - **Methods and Tools**:

        - Scikit-learn for feature extraction (TF-IDF, Bag of Words).
        - Gensim for Word2Vec embeddings.
    - **Time Allocation**:
        - Kaur Lõhmus: 5 hours
        - Maksim Kelus: 5 hours
3. **Model Selection and Training**
    - **Description**:
        - Experiment with models such as K-Nearest Neighbors (KNN), Random Forest, Logistic Regression, and deep learning models.
        - Train and validate models using cross-validation.
    - **Methods and Tools**:
        - Scikit-learn, TensorFlow/PyTorch.
        - GridSearchCV or RandomizedSearchCV for hyperparameter tuning.
    - **Time Allocation**:
        - Kaur Lõhmus: 10 hours
        - Maksim Kelus: 10 hours
4. **Model Assessment**
    - **Description**:
        - Evaluate model performance using metrics such as accuracy, precision, recall, and F1 score.
        - Compare performance across different models to select the best one.
    - **Methods and Tools**:
        - Scikit-learn for metrics calculation.
        - Matplotlib for visualizing confusion matrices and performance metrics.
    - **Time Allocation**:
        - Kaur Lõhmus: 6 hours
        - Maksim Kelus: 6 hours
5. **Model Deployment**
    - **Description**:
        - Apply the trained model to predict missing genres in the dataset.
        - Format and integrate predictions into a good looking graph.
    - **Methods and Tools**:
        - Downloadable Jupyter Notebook and trained model.
        - Pandas for formatting predictions.
    - **Time Allocation**:
        - Kaur Lõhmus: 5 hours
        - Maksim Kelus: 5 hours

Important Comments

- Cross-validation ensures robust model evaluation, reducing the risk of overfitting.
- Collaboration and task distribution among team members are balanced to ensure timely completion.

Total Time Commitment:

- Kaur Lõhmus: 32 hours
- Maksim Kelus: 32 hours