**Problem Statement: Building a Contextual Recommendation System for News Articles**

**Objective**

Create a recommendation system that suggests relevant news articles to users based on the contextual similarity of article content. The system will employ a **Word2Vec** model to generate word embeddings and determine similarity between articles based on content.

**Steps to Solve the Problem**

1. **Preprocess and Prepare the Text Data**:
   * **Objective**: Clean the news articles by removing stop words, punctuation, and other irrelevant text components.
   * **Tasks**:
     + Tokenize the text of each article.
     + Remove common stop words and punctuation to focus on meaningful words.
     + Prepare a corpus of cleaned, tokenized words for each article.
2. **Train or Use a Pre-trained Word2Vec Model**:
   * **Objective**: Represent each article in a way that captures semantic meaning.
   * **Tasks**:
     + Use a pre-trained Word2Vec model (e.g., Google News model) or train a Word2Vec model on the cleaned corpus if your dataset is domain-specific.
     + Generate embeddings for each word in each article using Word2Vec.
     + Obtain a single embedding vector for each article by averaging the Word2Vec embeddings of all words in the article.
3. **Calculate Article Similarities and Build Recommendations**:
   * **Objective**: Recommend articles with the highest similarity scores to the user.
   * **Tasks**:
     + Use cosine similarity to calculate similarity scores between articles based on their Word2Vec-averaged embeddings.
     + For each article, identify the top n articles with the highest similarity scores as recommendations.
     + Display or store these recommendations for a user-friendly recommendation experience.

**Expected Output**

* A recommendation system that can suggest contextually similar news articles for a given article, enabling readers to discover more articles that align with their interests based on content similarity.

**Problem Statement: Sentiment Analysis for Amazon Product Reviews Using Word2Vec Embeddings**

**Objective**

Develop a sentiment analysis model to classify Amazon product reviews as **positive** or **negative** based on their textual content. The system will use a **pre-trained Word2Vec model** to generate embeddings that capture the semantic meaning of reviews and use these embeddings to classify the sentiment.

**Steps to Solve the Problem**

1. **Preprocess the Amazon Review Data**:
   * **Objective**: Clean and prepare the text of each review for embedding generation.
   * **Tasks**:
     + Load Amazon product review data, focusing on reviews and their sentiment labels (positive or negative).
     + Preprocess each review by lowercasing, removing punctuation, stop words, and irrelevant characters.
     + Tokenize each review to create a cleaned, tokenized corpus ready for embedding generation.
2. **Generate Word Embeddings Using a Pre-trained Word2Vec Model**:
   * **Objective**: Represent each review as a vector by leveraging word embeddings.
   * **Tasks**:
     + Use a pre-trained Word2Vec model (e.g., Google News Word2Vec or Amazon review-specific Word2Vec model if available) to generate embeddings for each word in each review.
     + Compute a single embedding vector for each review by averaging the Word2Vec embeddings of all words in the review.
     + Store these averaged vectors as representations of each review for downstream sentiment classification.
3. **Train a Sentiment Classification Model Using Embeddings**:
   * **Objective**: Use the embeddings to classify reviews as positive or negative.
   * **Tasks**:
     + Split the dataset into training and testing sets.
     + Train a classifier (e.g., logistic regression, SVM, or neural network) using the averaged Word2Vec vectors of the reviews as input and their sentiment labels as output.
     + Evaluate the model on the test set using metrics like accuracy, F1-score, and ROC-AUC to measure performance.
     + Fine-tune the classifier and analyze misclassifications for improvement.

**Expected Output**

* A trained sentiment analysis model that can accurately predict the sentiment of Amazon product reviews. The model should generalize well to new reviews by using semantic information from the pre-trained Word2Vec embeddings.