Report on Problem 2: Text Classification of BBC Articles

1. Preprocessing Steps:

Step 1: Tokenization

i. Description: Tokenization is the process of breaking down the text into individual tokens or words.

ii. Implementation: NLTK's word_tokenize function is used to tokenize the text data.

Example: The sentence "Machine learning is awesome!" would be tokenized into ['Machine', 'learning', 'is', 'awesome', '!'].

Step 2: Lowercasing

i. Description: All tokens are converted to lowercase to ensure consistency in the text data.

ii. Implementation: The lower() method is applied to each token.

Example: ['Machine', 'learning', 'is', 'awesome', '!'] becomes ['machine', 'learning', 'is', 'awesome', '!'].

Step 3: Stopword Removal

i. Description: Common stopwords like 'and', 'the', 'is', etc., are removed as they do not carry significant meaning.

ii. Implementation: NLTK's English stopwords corpus is used to filter out stopwords.

Example: After removing stopwords, ['machine', 'learning', 'is', 'awesome', '!'] might become ['machine', 'learning', 'awesome', '!'].

Step 4: Lemmatization

i. Description: Lemmatization reduces words to their base or dictionary form, aiding in standardizing the text data.

ii. Implementation: NLTK's WordNet lemmatizer is used to lemmatize tokens.

Example: Words like 'awesome', 'awesomeness', 'awesomely' might all be lemmatized to 'awesome'.

2. Featurization Methods:

Step 5: TF-IDF Vectorization

i. Description: Term Frequency-Inverse Document Frequency (TF-IDF) converts text data into numerical features, capturing the importance of words in documents.
ii. Implementation: The TfidfVectorizer from scikit-learn is used to compute TF-IDF scores for each word.

Example: TF-IDF scores are calculated based on how often a word appears in a document (TF) and how rare it is across all documents (IDF).

3. Summary:

Tokenization: Breaking down the text into tokens.
Lowercasing: Ensuring uniformity by converting tokens to lowercase.
Stopword Removal: Eliminating common but irrelevant words.
Lemmatization: Standardizing words to their base form.
TF-IDF Vectorization: Converting preprocessed text into numerical features.

The preprocessing steps prepare the text data by tokenizing, lowercasing, removing stopwords, and lemmatizing, ensuring it is clean and standardized for analysis. TF-IDF vectorization then transforms the preprocessed text into a numerical representation suitable for machine learning algorithms.