5.1 Data Sources and Collection  
  
For building the news category classifier, data was sourced from three reputable publishers: ABC News, CBS News, and NBC News. These publishers were chosen for their reliability and diverse coverage across news categories. Data collection was facilitated using libraries such as BeautifulSoup and Selenium to scrape articles' metadata, including publish date, headline, content, and category. The scraping process was iterative, utilizing breadth-first search (BFS) to systematically collect data from each publisher. The resulting data were stored in separate CSV files: abc\_news\_articles.csv, cbs\_news\_articles.csv, and nbc\_news\_articles.csv.  
  
  
  
5.2 Data Preprocessing and Feature Extraction  
  
The collected data underwent several preprocessing steps to ensure quality and consistency for modeling purposes. These steps included:  
  
1. Merging data from the three CSV files.  
2. Removing rows with missing values in the "Category" and "Published Date" columns.  
3. Dropping rows where both the "Article Content" and "Headline" columns were missing.  
4. Replacing NaN values in the "Article Content" column with corresponding values from the "Headline" column.  
5. Handling duplicate records.  
6. Standardizing text data through lowercase conversion, punctuation removal, and stop-word removal in headlines.  
7. Performing lemmatization and tokenization to enhance feature representation and reduce text complexity.  
8. Establishing a mapping system to ensure uniformity of category labels.  
9. Conducting exploratory data analysis (EDA) to understand dataset characteristics.  
10. Addressing class imbalance through oversampling.  
11. Utilizing TF-IDF vectorization for feature extraction.  
12. Splitting the dataset into training and testing sets in an 80:20 ratio.

5.3 Classification Algorithm and Performance Metrics  
  
The logistic regression algorithm was chosen for its simplicity, efficiency, and effectiveness in binary classification tasks. Logistic regression is well-suited for this task due to its ability to model the probability of a categorical outcome based on predictor variables.  
  
Performance evaluation was conducted primarily using the F1 score, a metric that balances precision and recall. F1 score is particularly useful in this context as it provides a single measure of a model's accuracy, considering both false positives and false negatives. Given the goal of accurately classifying news categories, F1 score serves as a robust evaluation metric.  
  
  
  
5.4 Model Insights: Strengths and Limitations  
  
**Strengths:**  
- The model demonstrates proficiency in classifying international and US news categories, benefiting from sufficient and diverse dataset coverage.  
- Logistic regression offers simplicity and interpretability, making it easy to implement and understand.  
  
**Limitations:**  
- The model may suffer from overfitting due to class imbalance and could generalize poorly to new categories.  
- Performance may vary across different news categories, with some categories being more accurately predicted than others.  
- Further improvements could be achieved with additional data, especially for underrepresented categories, and efforts to balance the dataset.  
  
In conclusion, while the logistic regression model shows promise in classifying news categories, there's room for enhancement through addressing data imbalances and expanding dataset coverage. Continued refinement and iteration can lead to a more robust and reliable classifier for news categorization tasks.  
  
  
  
To run the entire process of collecting news articles, preprocessing the data, training the model, and testing it, follow these steps:  
  
1. Run `abc.py` to collect articles from ABC News and generate the file `abc\_news\_articles.csv`.  
2. Run `cbs.py` to collect articles from CBS News and generate the file `cbs\_news\_articles.csv`.  
3. Run `nbc.py` to collect articles from NBC News and generate the file `nbc\_news\_articles.csv`.  
4. Open `news category classifier - preprocessing and model training.ipynb` in a Jupyter Notebook environment or any compatible Python environment to preprocess the data and train the model.  
5. Execute the cells in the notebook to perform data preprocessing, feature extraction, model training, and evaluation. Save the trained model as `logistic\_regression\_model.pkl`.  
6. After training the model, run `news\_categorizer.py` to use the trained model for classification and test its performance.  
  
Ensure that all necessary dependencies and libraries are installed before running the scripts and notebooks. You may need to install libraries such as BeautifulSoup, Selenium, pandas, scikit-learn, and any other dependencies required for the project.