README for Assignment 2: Text Classification Using XGBoost

Overview

This project involves the development of a text classification model to classify text data into different categories using the XGBoost algorithm. The notebook demonstrates data preprocessing, feature extraction, model training, evaluation, and visualization techniques. The aim is to achieve high accuracy by employing robust machine learning methods.

Table of Contents

---

1. Setup

- Python version: 3.11.3
- IDE: Jupyter Notebook
- Libraries:
  - pandas version: 2.2.3
  - numpy version: 2.0.2
  - nltk version: 3.9.1
  - sklearn version: 1.5.2
  - matplotlib version: 3.9.3
  - seaborn version: 0.13.2
  - wordcloud version: 1.9.4
  - xgboost version: 2.1.3

pip install pandas numpy nltk scikit-learn matplotlib seaborn wordcloud xgboost.

2. Libraries Used

General Libraries for Reading and Manipulation.

- pandas: For handling and manipulating data.
- numpy: For numerical operations.

For Text Preprocessing.

- sklearn.feature_extraction.text.CountVectorizer: For converting text to numerical features.
- nltk.corpus.stopwords: For removing stopwords.
- nltk.stem.PorterStemmer: For stemming words.
- re: For text cleaning and regular expressions.
- nltk: For general text processing utilities.

Model Training and Evaluation.

- xgboost.XGBClassifier: The XGBoost classifier.
- sklearn.model_selection.train_test_split: For splitting the dataset into training and testing sets.
- sklearn.model_selection.StratifiedKFold: For cross-validation.
- sklearn.metrics: For accuracy, precision, recall, F1-score, confusion matrix, and precision-recall curves.

Data Visualization.

- matplotlib.pyplot: For creating static, interactive visualizations.
- seaborn: For enhanced data visualization.

Text Analysis.

- wordcloud.WordCloud: For generating word clouds.
- collections.Counter: For counting word frequencies.

Ignore Warnings.

- warnings: To suppress warnings for cleaner outputs.

---

3. Dataset Description

- The dataset is assumed to be a CSV file named text.csv containing columns:
    - text: The input text.
    - label: The corresponding labels for classification.

4. Steps Performed

1. Reading the Dataset
    - The dataset is read using pandas.read_csv, and the structure is inspected using head() and info().
2. Data Preprocessing
    - Lowercasing all text.
    - Removing non-alphabetic characters.
    - Removing stopwords using NLTK.
    - Stemming words using Porter Stemmer.
3. Feature Analysis
    - Calculating word count statistics (total, average, max, min).
    - Determining vocabulary size.
    - Generating bar plots for the 20 most frequent words.
    - Visualizing text length distribution with histograms.
    - Visualizing some relation between the words and the emotions.
4. Model Training and Testing
    - Feature extraction using Bag-of-Words with CountVectorizer.
    - Training an XGBoost model (XGBClassifier) with the processed features and labels.
    - Testing the model on unseen data.
5. Model Evaluation
    - Generating a classification report with precision, recall, F1-score, and accuracy.
    - Visualizing the confusion matrix using seaborn.heatmap.
    - Plotting precision-recall curves for each class.
    - Performing cross-validation to calculate the mean and standard deviation of metrics.
6. Possible Improvements
    - Feature Engineering:
        - Using TF-IDF for feature extraction.
        - Experimenting with n-grams.

- Hyperparameter Tuning:
    - Using Grid Search or Randomized Search for parameter optimization.
    - Experimenting with regularization parameters (`lambda`, `alpha`) to reduce overfitting.

---

5. Results

- The XGBoost model achieved high accuracy and robust performance across precision, recall, and F1-score metrics.
- Key visualizations include:
    - Confusion matrix.
    - Precision-recall curves.
    - Distribution of text lengths.

---

6. How to Run the Notebook

1. Ensure downloading all the libraries.
2. Place the dataset (text.csv) in the same directory as the notebook.
3. Run the cells sequentially to preprocess the data, train the model, and evaluate the results.
4. Review the outputs and visualizations for insights.

Personal Information

Diego Armando Salinas Lugo

Ds24353@essex.ac.uk

2401168