Text Classification Project

Team members:

Mayer Hamdi Armaniuos

Aliaa Mahmoud Shawky

Yousef Shams Eldin

Class 10

Dataset

The AG News Classification dataset is a collection of news articles from the AG's corpus of news articles, which is a collection of more than 1 million news articles gathered from various news sources around the world. The AG News Classification dataset contains a subset of 120,000 news articles from four different categories: World, Sports, Business, and Science/Technology.

Each news article in the dataset is labeled with one of the four categories, and the goal of the dataset is to train a machine learning model that can correctly classify new news articles into one of these four categories.

The AG News Classification dataset is commonly used as a benchmark dataset for text classification tasks, and it has been used in many research papers to compare the performance of different machine learning algorithms and techniques for text classification.

The dataset is available for download from various sources, including the UCI Machine Learning Repository and the Kaggle platform.

Machine learning algorithm

Logistic regression is a supervised learning algorithm used for binary classification problems. In logistic regression, the goal is to model the probability of an event occurring, given some input features. The output of the logistic regression model is a probability score between 0 and 1, which can be thresholded to make binary predictions.

In text classification, logistic regression can be used to predict the category or class of a text document based on its content. The input features for the logistic regression model are usually derived from the text of the document, such as the frequency of occurrence of certain words or phrases in the document. These input features are often represented as a vector of numbers.

Once the input features are prepared, the logistic regression model can be trained using a labeled dataset of text documents. During training, the model learns to assign higher probabilities to documents in the correct category and lower probabilities to documents in other categories. The objective function used to train the logistic regression model is usually the cross-entropy loss function, which measures the difference between the predicted probabilities and the true labels of the training examples.

After training, the logistic regression model can be used to predict the category of new text documents. The model takes the input features of the new document, computes a probability score for each category, and then selects the category with the highest probability score as the predicted category for the document.

Logistic regression is a simple and interpretable algorithm that can be trained quickly on large datasets. It is also robust to noise and can handle sparse input features, which makes it a popular choice for text classification tasks. However, logistic regression may not perform as well as

The dataset is available for download from various sources, including the UCI Machine Learning Repository and the Kaggle platform.

Preprocessing

The code is performing some basic text preprocessing steps to prepare a dataFrame of news articles for text classification using a bag of words approach. Here are the details of the individual steps:

Creating copy of the original DataFrame to work with, so that the original DataFrame is not modified.

Creating a CountVectorizer object with a maximum vocabulary size of 5000. The CountVectorizer is a feature extraction technique that converts text documents into a matrix of word counts.

Using a regular expression to replace each non-alphabetic character with a space. This step helps to remove special characters and symbols from the text.

Converting the text to lowercase and splits it into a list of words.

Performing word stemming on each word in the list, using the PorterStemmer object created earlier. It also removes stop words from the list. Stop words are common words like "the" and "and" that are often removed from text data because they don't carry much meaning.

Creating a bag of words representation of the preprocessed text data using the CountVectorizer object created earlier. The fit_transform() method of the CountVectorizer object first learns the vocabulary of the text data and then converts the text data into a matrix of word counts. The resulting matrix is a sparse matrix of shape (n_samples, n_features), where n_samples is the number of documents and n_features is the size of the vocabulary (in this case, 5000). The toarray() method is then called to convert the sparse matrix to a dense matrix, which is assigned to the text column of the preprocessed_df DataFrame.

Finally the dataframe is split into train (90%) and test (10%) splits, after that the training takes place.

Results and accuracy

Due to training using large dataset that consists of more than 10 thousand rows (2000 epochs), the model has some good results.

Accuracy: 89.60%

Precision: 89.63%

Recall: 89.60%