Documentation

Approach

The provided code is a text classification task to predict whether a given tweet is about a real disaster or not. The approach involves the following steps:

Data Preprocessing:

The function preprocess(text) is used to clean the input text. It removes URLs, non-alphanumeric characters, and converts the text to lowercase. It also tokenizes the text, applies lemmatization, and filters out stopwords using the NLTK library.

The function prepare\_df(path) reads the CSV file into a Pandas DataFrame, drops irrelevant columns like 'keyword' and 'location', and applies the preprocessing function to the 'text' column, creating a new 'preproc' column containing the preprocessed text.

Exploratory Data Analysis:

The function visualizes the value counts per target class using a count plot.

The most common words per target class are visualized using horizontal bar plots.

Text Vectorization:

The TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer is used to convert the preprocessed text into numerical features.

The training data is split into training and validation sets using the train\_test\_split() function.

Handling Class Imbalance:

The Synthetic Minority Over-sampling Technique (SMOTE) is used to address the class imbalance problem. It oversamples the minority class to balance the data distribution.

Model Selection and Training:

Three classifiers are used: Logistic Regression, Naive Bayes, and Support Vector Machine (SVM).

Each model is trained on the training data, and accuracy, precision, recall, and F1-score are calculated on the validation set.

Cross-validation is performed to get an estimate of the models' performance on different folds of the training data.

Model Evaluation:

The confusion matrix and classification report are used to evaluate the models' performance on the validation set.

The model with the best recall value is selected for predicting the test set.

Test Set Prediction:

The selected model (SVM) is used to predict the target class for the test set.

The predicted results are stored in the 'test\_set' variable.

Insights

Data Preprocessing: The data preprocessing step is essential for text classification tasks. It involves removing noise, converting text to lowercase, tokenization, and removing stopwords. The NLTK library provides useful tools for these tasks.

Class Imbalance: The dataset is imbalanced, with more non-disaster tweets than disaster tweets. To address this issue, SMOTE is used to oversample the minority class (disaster tweets) to achieve a balanced dataset, which can improve the model's performance.

Model Selection: Three classifiers (Logistic Regression, Naive Bayes, and SVM) are tested. The selection is based on the recall metric since it is essential to correctly predict the disaster tweets to ensure timely response and assistance.

SVM Performance: The SVM model shows the best recall values for both classes, making it the selected model for this particular dataset.

Conclusion

In this task, we performed text classification to predict whether a tweet is about a real disaster or not. We preprocessed the data, handled class imbalance using SMOTE, and evaluated different models. SVM was selected as the best model based on its recall values for both classes. Finally, we used the SVM model to predict the target class for the test set.