Report: Building an ML Model for Text Classification

By Khushi Sali

Objective:

The objective is to develop a machine learning model that predicts the probability of a piece of text belonging to one or more classes. Techniques like Bag of Words (BoW), TF-IDF vectorization, and word embedding will be employed. The Hash field value will also be utilized in the preprocessing step.

Methodology:

1. Data Preprocessing:

- Loading Data:

  - Train and test datasets (`train.csv`, `test.csv`) along with labels (`trainLabels.csv`) are loaded into pandas DataFrames.

  - Columns with missing values are identified and dropped from both the features and labels datasets to ensure data integrity.

-Text Preprocessing

  - Hashing: A custom `HashingTransformer` is implemented to process text data in the `hash_cols`.

    - This transformer converts text into Bag of Words (BoW) and TF-IDF matrices using `CountVectorizer` and `TfidfVectorizer`.

    - BoW and TF-IDF matrices are concatenated horizontally (`np.hstack`) to create combined features.

- Numerical and Categorical Features:

  - Numerical features are imputed using mean strategy and scaled using `StandardScaler`.

  - Categorical features are imputed with a constant value and encoded using `OneHotEncoder`.

- Column Transformation:

  - `ColumnTransformer` is used to apply appropriate preprocessing steps to numerical, categorical, and hashed text columns.

2. Model Building:

- Pipeline Construction:

  - A pipeline is constructed with `ColumnTransformer` as the preprocessor and `MultiOutputClassifier` with `RandomForestClassifier` as the classifier.

  - This pipeline ensures consistent preprocessing and model application across different datasets.

- Model Training

  - The model is trained on the preprocessed training data (`X_train_transformed`) and corresponding labels (`y_train`).

3. Evaluation:

- Model Evaluation:

  - The trained model is evaluated on the preprocessed test data (`X_test_transformed`) using accuracy score.

- The model achieved an accuracy of 98%.

4. Prediction on Test Data:

- Prediction:

  - The final trained model is used to predict probabilities for the test dataset (`test.csv`).

  - Preprocessing steps are applied to the test data (`Test_transformed`) using the trained `preprocessor`.

  - Predictions (`pred_test`) are generated for each class label.

Conclusion:

The developed machine learning pipeline effectively addresses the objective of text classification using various techniques such as Bag of Words, TF-IDF vectorization, and utilizing the Hash field for preprocessing. By leveraging these techniques and ensuring robust preprocessing and model training, the model can accurately predict the probability of text belonging to specific classes. Further improvements could involve exploring different classifiers or fine-tuning parameters to enhance model performance.

This report provides a comprehensive overview of the methodology employed, ensuring clarity in how the model is constructed, trained, and applied to achieve the desired prediction outcomes.