Text Classification on Customer Review Dataset

Introduction

Text classification is the process of categorizing text into predefined labels based on its content. For this task, we focus on classifying customer reviews into multiple categories such as "Design," "Build Quality," "Performance," "Screen," "Price," "Camera," and "Battery." Given that the dataset is multi-label in nature, meaning each review can belong to one or more categories, this introduces complexities in model training and evaluation.

The dataset contains customer reviews on various products, with labels corresponding to different product features and attributes. The goal of the project was to build a robust classification model to accurately predict the categories based on the review content.

Upon analyzing the dataset, we experimented with the following approaches:

Approach 1: BERT (Bidirectional Encoder Representations from Transformers)

BERT is a state-of-the-art transformer-based model for natural language processing tasks. Due to its deep architecture, BERT generally provides excellent results for text classification tasks, especially with large datasets. However, due to the smaller size of the dataset and its potential to overfit, we observed that BERT underperformed. The model showed poor generalization, and in some cases, the accuracy was significantly low, especially with smaller amounts of data.

Therefore, after careful consideration, we decided not to pursue BERT for this particular project.

Approach 2: Deep Learning Neural Networks

While deep learning neural networks such as CNNs and LSTMs have shown great promise in text classification tasks, we faced similar challenges when applied to this dataset. The small size of the dataset combined with multi-label classification resulted in overfitting issues, despite attempting different methods of regularization.

We also considered using techniques like SMOTE (Synthetic Minority Over-sampling Technique) and random sampling for data augmentation. However, these methods were not effective in dealing with multi-label classification, further complicating the use of deep learning neural networks. Consequently, we opted not to pursue deep learning neural networks due to these limitations.

Approach 3: Machine Learning Algorithms

In comparison to BERT and deep learning models, machine learning algorithms offered a more stable and reliable approach for this task. We observed that machine learning models performed better in terms of accuracy, precision, recall, and F1 score, making it the optimal choice for this project.

Below are the steps followed in implementing machine learning algorithms:

Machine Learning Workflow

1. Data Cleaning

The first step involved preprocessing the review text data to ensure the quality of the input for the machine learning models. The following text preprocessing techniques were applied:

- **Tokenization**: Splitting the text into individual words or tokens.
- **Removal of Special Characters**: Special characters, punctuations, and unnecessary symbols were removed to clean the data.
- **Stemming**: Words were reduced to their root form to ensure consistency (e.g., "running" → "run").
- **Lowercasing**: All text was converted to lowercase to ensure uniformity.

These preprocessing steps helped standardize the text data and prepare it for further analysis.

2. One-Hot Encoding on the Category Column

Since the task was a multi-label classification problem, the target labels (categories) were one-hot encoded. This transformed each category into a binary vector, where each column represents a category, and the values are either 0 or 1 depending on whether the review belongs to that category.

We also attempted data augmentation using SMOTE, but due to the multi-label nature of the task, it did not work as expected. Therefore, we did not use SMOTE for model training.

3. Data Visualization

To gain insights into the dataset, we performed data visualization using various plots, including:

- **Bar charts**: To understand the distribution of categories across the reviews.
- **Line plots**: To analyze trends in reviews and labels over time.

These visualizations helped us better understand the data distribution, identify class imbalances, and formulate strategies for model training.

4. Embedding the Reviews

To prepare the reviews for model input, we used **Langchain embeddings** to convert the text reviews into numerical vector representations. This method captures the semantic meaning of the text and allows the machine learning models to process the information effectively.

5. Model Building

For model training, we employed several supervised machine learning algorithms, including:

- **Logistic Regression**
- **Support Vector Machine (SVM)**
- **K-Nearest Neighbors (KNN)**
- **Random Forest Classifier**
- **LightGBM (LGB)**
- **CatBoost**

We utilized **k-fold cross-validation** to evaluate the models and perform hyperparameter tuning. Among all the models tested, **CatBoost** showed the best performance in terms of accuracy, precision, and recall.

6. Model Deployment

After selecting the best-performing model, **LightGBM (LGB)**, we deployed it for inference. The model was trained and validated, achieving satisfactory results in terms of both classification accuracy and generalization.

7. Model Validation

We validated the model using a separate test dataset (`test.xlsx`), which contained only review texts. The reviews were first converted into word embeddings using the same embedding method as during training. The trained **LightGBM model** was then used to predict the categories for each review in the test dataset.

8. Output Generation

The final output was generated as an Excel file, where each review was paired with its predicted categories. This output was useful for practical applications, such as automatically tagging customer reviews in a system.

Results

The final results after model validation demonstrated the effectiveness of the **LightGBM** model. Key performance metrics included:

- **Accuracy**: High accuracy in predicting the correct categories.
- **Precision**: The model was able to correctly identify positive categories with a high degree of precision.
- **Recall**: The model captured a significant proportion of the relevant categories, as evidenced by a good recall score.
- **F1 Score**: A balanced F1 score that indicates the model's effectiveness in classifying the reviews.

Best Model: LightGBM

Based on cross-validation results and performance metrics, **LightGBM** emerged as the best model for this task. It outperformed the other algorithms in terms of all evaluation metrics and was chosen for deployment.

Conclusion

In this text classification task on a customer review dataset, we explored multiple approaches to classify reviews into multiple categories. Despite the potential of BERT and deep learning models, machine learning algorithms, particularly **LightGBM**, proved to be the most effective for this specific dataset.

The steps taken included data cleaning, embedding, model training, and evaluation using k-fold cross-validation, followed by model validation on a separate test set. The final output, an Excel file containing reviews and their predicted categories, provides valuable insights for further analysis.

The success of the **LightGBM** model highlights the importance of selecting the right approach based on dataset size and characteristics, and in this case, traditional machine learning techniques offered the most reliable results.