Week 9 Deliverables

Orkun

2 July 2024

Team Member's Details

- **Group Name:** Orkun
- **Name:** Orkun Kınay
- **Email:** orkunkinay@sabanciuniv.edu
- **Country:** Turkey
- **College/Company:** Sabancı University
- **Specialization:** NLP

Problem Description

The goal of this project is to develop a hate speech detection model using Twitter data. The dataset consists of tweets labeled as hate speech or non-hate speech, which will be used to train and evaluate machine learning models. The primary objective is to accurately classify tweets and mitigate the spread of hate speech on social media platforms.

GitHub Repo Link

https://github.com/orkunkinay/Hate-Speech-Detection/tree/main/data_preprocessing

Data Cleaning and Transformation

Handling Missing Values

Since there are no missing values in the dataset, no imputation was necessary.

Handling Outliers

Outliers in text length were handled using two methods:

- **Method 1: Remove Outliers** Outliers based on text length were removed. The threshold for outliers was determined using the Interquartile Range (IQR) method.
- **Method 2: Cap Outliers** Outliers were capped at the threshold value to reduce their impact on the model.

Text Data Transformation

The text data was cleaned and transformed using the following steps:

- Removed URLs, mentions, and special characters using regular expressions.
- Converted all text to lowercase.

- Tokenized the text and removed stop words.
- Lemmatized the tokens to reduce them to their base forms.

The text data was further transformed into numerical features using BERT embeddings.

Handling Imbalanced Data

Imbalanced data was handled using two techniques:

- **Technique 1: Oversampling** The minority class was oversampled by randomly sampling with replacement from the minority class to match the majority class.
- **Technique 2: SMOTE** A custom implementation of Synthetic Minority Over-sampling Technique (SMOTE) was used to generate synthetic samples for the minority class.

Model Training and Evaluation

The models were trained and evaluated using the following configurations:

- **Base Model:**
  - Validation Accuracy with Base Data: 95.06%
- **Oversampling:**
  - Validation Accuracy after Oversampling (removed outliers): 99.48%
  - Validation Accuracy after Oversampling (capped outliers): 99.78%
- **SMOTE:**
  - Validation Accuracy after SMOTE (removed outliers): 99.30%
  - Validation Accuracy after SMOTE (capped outliers): 99.22%