Emotion Recognition on Twitter

Introduction

The goal of this project is to predict emotion labels for tweets collected from Twitter. By applying data mining techniques, we aim to classify tweets into distinct emotion categories based on their textual content.

Preprocessing

The dataset consists of raw Twitter data in JSON format, which is transformed into a structured DataFrame for better handling. Each record contains the following fields:

- **Tweet ID**: A unique identifier for each tweet.
- **Hashtags**: Zero or more hashtags associated with the tweet.
- **Text**: The main content of the tweet.

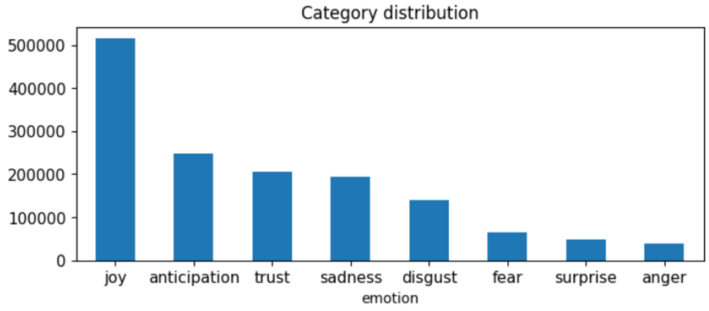
The dataset is then split into training and test subsets based on tweet identifications. Emotion labels provided for training data are appended to the corresponding records, preparing the data for feature engineering.

Data Overview

	Number of records
train	1455563
test	411972

Emotion category distribution

```
emotion
joy             516017
anticipation    248935
trust           205478
sadness         193437
disgust         139101
fear             63999
surprise         48729
anger            39867
Name: count, dtype: int64
<Axes: title={'center': 'Category distribution'}, xlabel='emotion'>
```



Feature Engineering

Hashtags, which may provide valuable contextual information, are appended to the text column for simplicity. This consolidated text is then used as the input for the prediction model.

Model Development

The primary model is built using the following approach:

1. TF-IDF Transformation: The text data is converted into numerical features using TF-IDF with a maximum feature size of 10,000.
2. Logistic Regression: The TF-IDF features are used to train a logistic regression model. This approach yielded the best result among the methods explored.

Alternative Approaches

Several additional methods were implemented and evaluated:

1. **TF-IDF with Naive Bayes**
   - TF-IDF Transformation: Text data is transformed using the same TF-IDF approach.
   - Modeling: Features are used to train a Naive Bayes classifier.

2. **Custom-Trained Word2Vec**
   - Preprocessing: Tokenized the text data (including appended hashtags).
   - Word2Vec Training: Trained a Word2Vec model on the tokenized text.
   - Modeling: Used the resulting embeddings as features for logistic regression.

3. **Pretrained Word2Vec (GloVe)**
   - Preprocessing: Same as the custom-trained Word2Vec approach.
   - Embedding: Leveraged pre-trained embeddings from the GloVe Twitter 25 model.
   - Modeling: Fed these features into a logistic regression model.

4. **Deep Learning Approach**
   - Embedding: Utilized embeddings from the custom-trained Word2Vec model.
   - Neural Network: Developed a neural network with:
     - Input shape: 100 (embedding dimension).
     - Output shape: 8 (number of emotion categories).

- Training: Model trained for 25 epochs with a batch size of 32.

Performance

Model	Private Score	Public Score
TFIDF + Logistic Regression	**0.39201**	**0.40772**
TFIDF + Naïve Bayes	0.36590	0.38450
Custom-Trained Word2Vec + Logistic Regression	0.35152	0.36778
Pretrained Word2Vec + Logistic Regression	0.28284	0.28593
Deep Learning	0.39066	0.40741

Conclusion

This project explored multiple approaches to predict emotion labels for tweets, ranging from traditional machine learning methods to deep learning. The logistic regression model with TF-IDF features provided the most reliable result.