Emotion Dataset

Statistical Methods

I explored multiple methods using traditional ML techniques to predict the confidence scores of various classes of emotions. Some of the methods I pursued are:
Supporting Vector Machines

ElasticNet Regression: ElasticNet regression combines ridge and lasso regression, which adds both L1 and L2 penalties to the least squares objective function. It combines the strengths of both methods effectively handles multicollinearity and selects relevant features.

K-Nearest Neighbors (KNN): KNN is a non-parametric method that predicts the class of a data point by averaging the labels of its nearest neighbors. It is simple to implement and suitable for small to medium-sized datasets.

Decision Trees: Decision trees partition the feature space into regions based on feature values, and each region corresponds to a prediction. Decision trees are interpretable and can handle both numerical and categorical features. However, they are prone to overfitting.

I used TF-IDF with Linear regression

Model Used

TF-IDF (Term Frequency-Inverse Document Frequency): TF-IDF is a statistical method used to evaluate the importance of a word in a document relative to a collection of documents. It measures the frequency of a term in a document relative to its frequency in the entire document corpus.
Linear Regression: Linear regression models model the relationship between independent variables (TF-IDF features) and a dependent variable (emotion labels).

Reasons

Reasons for Using TF-IDF with Linear Regression:

Interpretability: TF-IDF features are interpretable and provide insights into the importance of individual terms in predicting emotions.
Efficiency: TF-IDF is computationally efficient and can easily handle large text datasets.
Simplicity: Linear regression is simple to implement and understand, making it suitable for baseline models.

Outcome

The Root Mean Square of the model is 0.3. Since it's less than 0.5 hence the model performs very well.

Deep Learning Methods

There are many Deep Learning methods available like Neural Networks, LSTMs, and Bidirectional LSTMs. I have used BERT because it has been trained on a large corpus of data hence it has very relevant encodings for each word.

BERT is a state-of-the-art deep learning model developed by Google. It belongs to the Transformer architecture, which relies on self-attention mechanisms to capture relationships between words in a sentence. BERT is pre-trained on a large corpus of text data using two unsupervised learning tasks: masked language modeling and next-sentence prediction. This pre-training allows BERT to learn rich contextual representations of words, making it highly effective for a wide range of NLP tasks, including text classification, named entity recognition, question answering, and sentiment analysis.

Reasons for using BERT

Representation Learning: Deep learning models automatically learn feature representations from data, eliminating the need for manual feature engineering.
Hierarchical Feature Extraction: Deep learning models can capture hierarchical representations of data, enabling them to learn complex patterns and relationships.
Scalability: Deep learning models can scale to large datasets and complex tasks, making them suitable for real-world applications with high-dimensional data.
State-of-the-Art Performance: Deep learning models like BERT have achieved state-of-the-art performance on various NLP benchmarks, outperforming traditional machine learning methods in many cases.

Method

I used the BERT-uncased model because of the limited amount of parameters hence generating encoding takes less time. After generating expected confidence scores, I compared them with actual confidence scores from the test dataset and calculated the RMSE.

Result

RMSE was 0.2 which is below 0.5 hence it is good