EMOLNT

Predicting text intensity

Kavish shah

6352055045
shahkavish406@gmail.com

INTRODUCTION

The EmoInt task involves predicting the intensity of emotions in textual data. The goal is to develop models that can accurately estimate the emotional intensity expressed in given sentences. In this report, we present our approach and findings using both statistical and deep learning models.

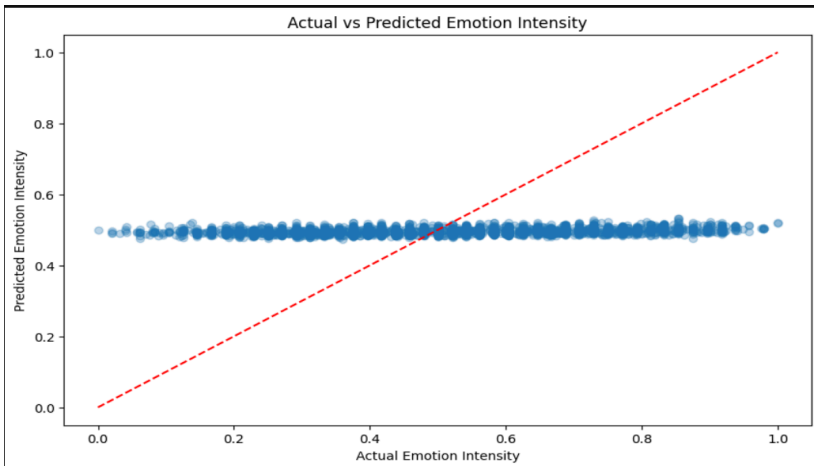
Statistical Model Methodology

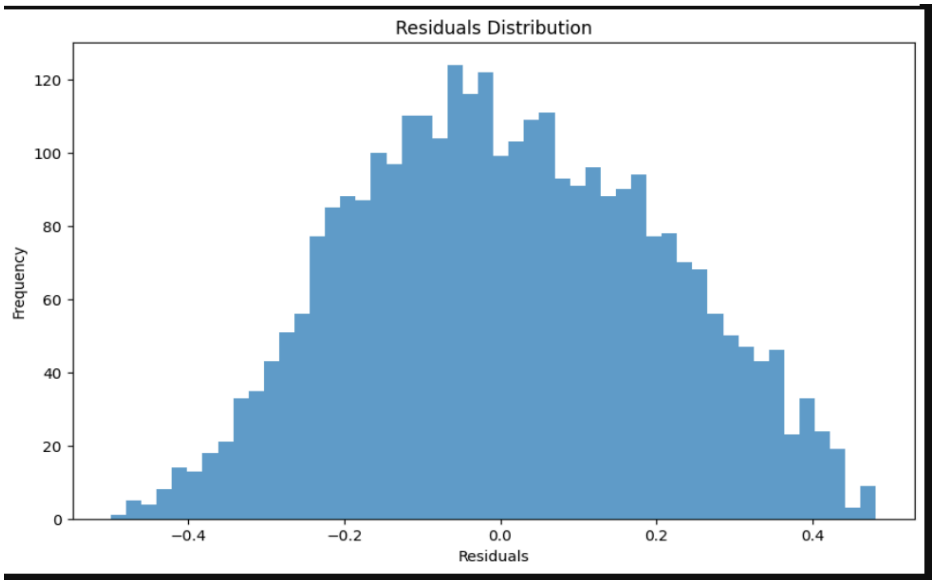
1) Data preprocessing- We first loaded the training and testing datasets for four emotions: anger, joy, fear, and sadness. The datasets were concatenated to form a unified training and testing set.
2) Data cleaning-Text data was cleaned by removing URLs, mentions, hashtags, and non-alphabetic characters, and converting to lowercase.
3) Feature Extraction- TF-IDF vectorization was used to transform the cleaned text into numerical features.

4) **Model Training** -We explored several regression models using Grid Search to find the best hyperparameters for each model. The models included Linear Regression, Ridge Regression, Lasso Regression, Decision Tree, Random Forest, Support Vector Regression (SVR), and Gradient Boosting.The best performing model was identified, and its parameters were saved for future use.

5) **Evaluation** - The best model was evaluated on the test dataset, and performance metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R^2) were calculated.

6) **Visualization**- A scatter plot of actual vs. predicted values and a residual histogram were created to visualize model performance.

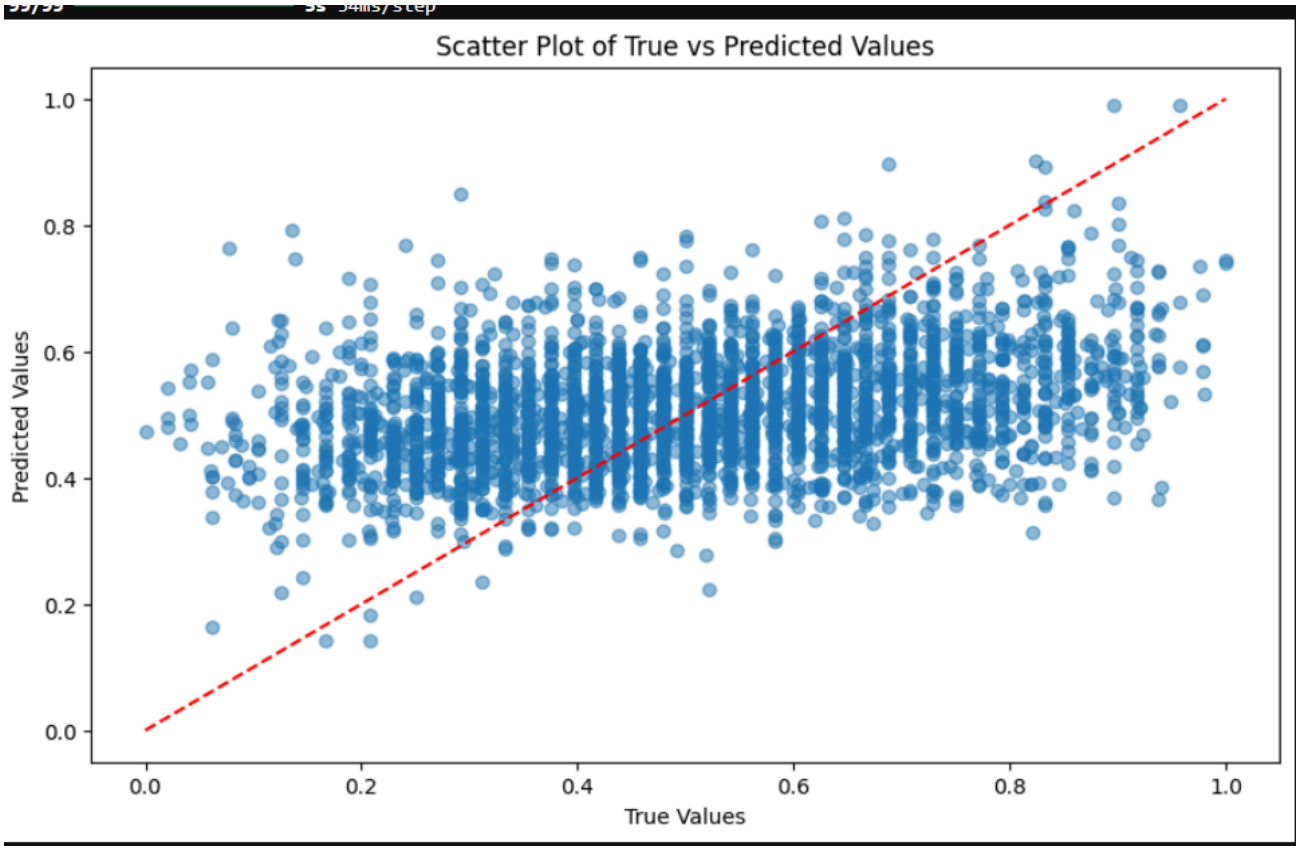


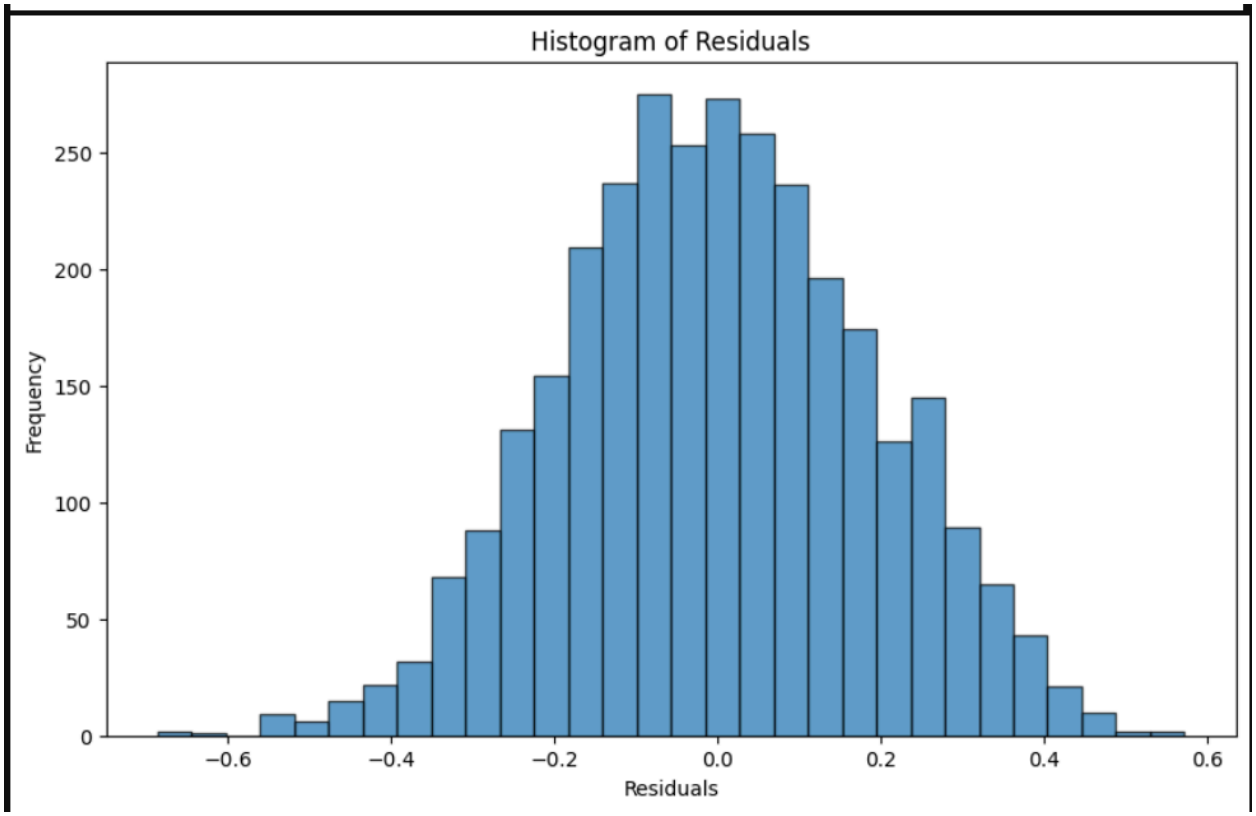


Conclusion-The statistical model achieved a mean squared error of 0.038, mean absolute error of 0.1622, and an R^2 score of 0.020 on the test dataset. The Ridge Regressor was identified as the best performing model. The residual analysis showed that the model predictions were generally close to the actual values, with some degree of variance.

Deep learning Model

1) Data Preparation- The same dataset was used as in the statistical model. The data was concatenated, cleaned, and preprocessed similarly, but with a focus on preparing it for a deep learning model.
2) Tokenization and Padding- The text data was tokenized and converted into sequences. The sequences were then padded to ensure uniform length.
3) Embedding Matrix- A pre-trained Word2Vec model was used to create an embedding matrix. This matrix helps in initializing the embedding layer of the model with pre-trained word vectors.
4) Model Architecture- A bidirectional LSTM model was used to capture the sequential nature of the text data. The architecture consisted of an embedding layer initialized with the embedding matrix, followed by a bidirectional LSTM layer, dense layers, and dropout for regularization.
5) Training the Model - The model was trained on the training data with a validation split to monitor performance on the validation set.
6) Evaluation and Visualization-The model was evaluated on the test data, and predictions were made. Scatter plots and histograms were used to visualize the performance and residuals.


Scatter Plot of True vs Predicted Values
Predicted Values
True Values
0.0
0.2
0.4
0.6
0.8
1.0


Histogram of Residuals
Frequency
Residuals
0
50
100
150
200
250
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6