Analysis of Zulip chat messages

Roman Karpenko
roman.karpenko@ut.ee
Master's
Tartu Ulikool

Abstract

This is a report on the analysis of Zulip chat

1 About the analysis

We analyzed the Data Thinking Zulip chat data to predict the sender ID based on the text content. We performed preprocessing on the text data, such as tokenization, and converted it into numerical format using embeddings. Then, we built and evaluated logistic regression and linear regression models.

The code snippet above consists of the following steps:

Preprocessing: Tokenization of the text data into sentences. Further preprocessing, such as removing stop words, could also be applied if needed.

Embeddings: The text data is converted into numerical format using Word2Vec embeddings. These embeddings allow the model to capture the semantic relationships between words.

Data Splitting: The dataset is divided into training and test sets with an 80-20 split.

Logistic Regression: We trained a logistic regression model on the training data and evaluated its performance on the test data using accuracy. Logistic regression is used for binary or multiclass classification tasks, and its equation can be represented as:

y = 1 / (1 + exp(-(b0 + b1 * x1 + ... + bn * xn)))

Linear Regression: We trained a linear regression model on the training data and evaluated its performance on the test data using mean squared error (MSE). Linear regression is used for continuous target variables, and its equation can be represented as:

y = b0 + b1 * x1 + ... + bn * xn

The goal of this analysis was to predict the sender ID of a message based on the text content using logistic and linear regression models. The logistic regression model aims to classify the sender ID directly, while the linear regression model attempts to predict a continuous value that can be mapped to the sender ID. The choice of these models helps us understand and compare the performance of classification and regression approaches for this task.

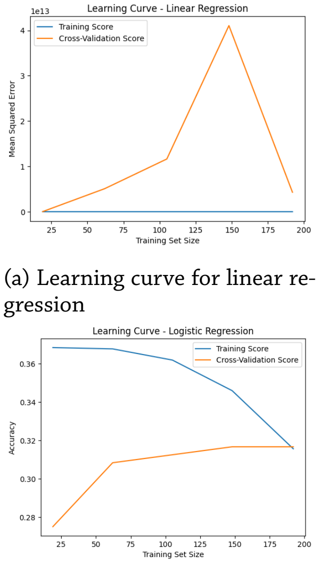
As for the visualizations, we created a residual plot for linear regression, which shows the residual errors between the true and predicted values. It helps us identify any patterns in the errors, such as non-linearity, heteroscedasticity, or presence of outliers. We also created learning curves for both logistic and linear regression models to visualize the performance of these models as the training set size increases. These curves help us assess overfitting or underfitting and the overall model generalization.

For the embeddings visualization, we used the t-SNE (t-distributed Stochastic Neighbor Embedding) dimensionality reduction technique to project the high-dimensional Word2Vec embeddings into a 2D space. This allowed us to visualize the embeddings and observe the relationships between words in the Data Thinking Zulip chat data.

The t-SNE algorithm is a nonlinear dimensionality reduction technique that maintains the relative pairwise distances between points in the original high-dimensional space while mapping them to a lower-dimensional space. It's particularly effective for visualizing complex datasets like word embeddings, where words with similar meanings tend to form clusters in the lower-dimensional space.

In our analysis, we first computed the Word2Vec embeddings for the first word in each message. Then, we used t-SNE to transform these embeddings into a 2D representation. Finally, we created a scatter plot of the transformed embeddings, with the points colored according to the sender ID. This visualization helped us explore the structure and relationships within the embeddings, as well as observe any potential patterns or clusters related to the sender IDs.

2 Figures



(a) Learning curve for linear regression

(b) Learning curve for logistic regression

Figure 1: Learning curves for linear and logistic regression models

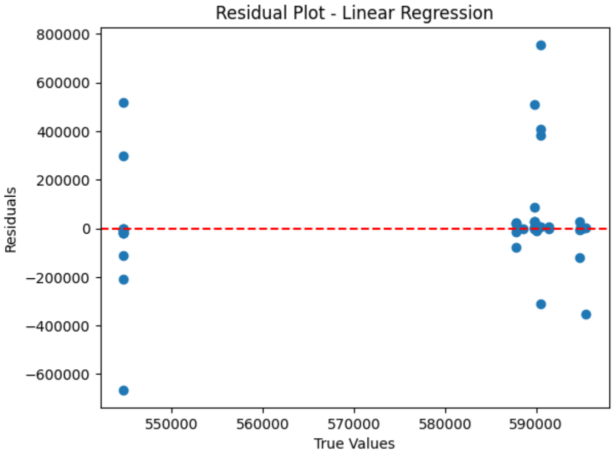


Figure 2: Residual plot for linear regression

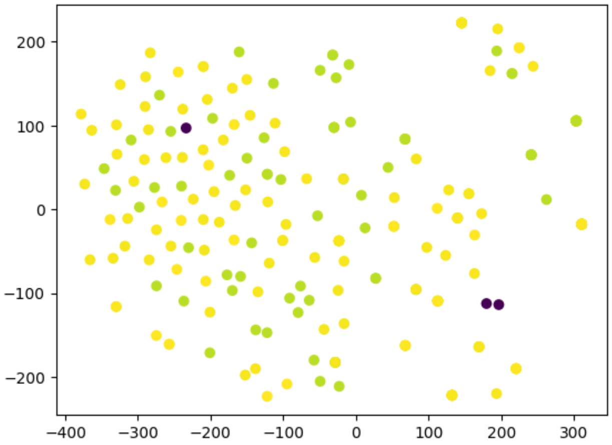


Figure 3: t-SNE visualization of Word2Vec embeddings