Amazon ML Challenge 2023 Report

BITS.BARD

Team member Names:

1. Bhaswanth Ayapilla
2. Aman Jain
3. Rakshith Dasari
4. Devesh S

Introduction:

In this report, we predicted the product's length using ridge regression in conjunction with other pre-processing methods. To update the outliers, we applied the 1.5*IQR technique, and we filled in the missing numbers with the word "Missing". We used a CountVectorizer to turn the preprocessed data into tokens, and then we fit-transformed the vectorizer, the result was a sparse matrix. Finally, we fitted the model to the training data and predicted the Product Lengths for the test data using Ridge regression with solver = "auto," alpha = 15, and max_iter = 60. NumPy,

Tools: Pandas, Re, sklearn are some of the common Python libraries we used.

Data Pre-processing:

The data was pre-processed to weed out any extraneous information and format it for use. The 1.5*IQR method was used to update outliers by substituting the mean. Values that were missing have the word "Missing" in their place. The text data was cleaned of punctuation, stop words, emoticons, html elements, and URLs. After changing the text's case to lowercase, the data was stemmed.

Tools: Punctuation from string library and stop words from nltk were used.

Vectorization:

To create tokens, the preprocessed data was fed through a CountVectorizer. Each unique word in the text is compiled into a vocabulary by the CountVectorizer, which then counts how many times it appears throughout the text. Aside from undesired terms like stop words, this phase also eliminates them.

Tools: CountVectorizer, TfidfVectorizer, Tokenizer are some of the sklearn libraries used.

Model Fitting:

After vectorization, we fitted the model to the training data using Ridge regression with solver = "auto," alpha = 15, and max_iter = 60. By including a penalty term to the least squares regression goal function, the regularisation approach of ridge regression helps avoid overfitting. To avoid the model overfitting the data, the hyperparameters alpha and max_iter were adjusted to 15 and 60, respectively.

Tools: Ridge from sklearn library

Model Prediction:

We predicted the Product Lengths for the test data using the trained model. Before being fed through the CountVectorizer to create the token matrix, the test data underwent the same pre-processing as the training data. The product lengths for the test data were then predicted using the model.

Tools: Ridge from sklearn library