Exploring Enhanced Positional Encodings and Embeddings in NLP

Preprint · December 2023

DOI: 10.13140/RG.2.2.18428.72329

Jamell Alvah Samuels
Imperial College London
64 PUBLICATIONS **0** CITATIONS

Exploring Enhanced Positional Encodings and Embeddings in NLP
J.I.Samuels

Author: Jamell Ivor Samuels

December 28, 2023

Introduction

Natural Language Processing (NLP) has seen remarkable advancements in recent years, driven by innovations in positional encodings and embeddings. In this research article, we delve into the comparative analysis of two distinctive positional encoding techniques: the well-established Sinusoidal encoding and a novel approach known as Henon encoding. Our exploration aims to shed light on the efficacy of these encoding methods in enhancing the performance of machine learning models in NLP tasks.

As of now, the development of a decoder for Henon encoding remains uncharted territory. In the absence of a dedicated decoder, we have opted for an alternative avenue of investigation. Our approach involves assessing the predictive capabilities of various machine learning models when applied to the Sinusoidal encoding of a Henon encoding derived from summarization or sentiment analysis datasets. Conversely, we also examine the performance of these models when tasked with predicting a Henon encoding of a Sinusoidal encoding derived from the same datasets.

This study goes beyond the conventional scope by introducing semantic and sentimental embedding methods into the comparative framework. By doing so, we aim to discern not only the encoding techniques' impact on general NLP tasks but also to unravel their impact on more nuanced aspects, specifically in understanding how semantic and sentimental embedding techniques perform in conjunction with different encoding methodologies. This comprehensive analysis will provide valuable insights into the intricate interplay between encoding techniques and embedding methods, particularly in the realms of semantic understanding and sentiment analysis.

Through these comparisons, we seek to contribute valuable insights into the strengths and limitations of Sinusoidal and Henon encodings, paving the way for informed decisions in adopting enhanced positional encodings in NLP applications. Our findings promise to advance the understanding of encoding techniques and guide future research in harnessing the full potential of positional encodings and embeddings in the ever-evolving field of NLP. This paper has been written with the help of ChatGPT 3.5.

The code can be found at https://github.com/jamellknows/positional_encoding. .

The Investigation

Introduction

The purpose of this research chapter is to conduct a comprehensive investigation into the comparative performance of Henon encoding versus sinusoidal encoding, coupled with an exploration of semantic and sentimental embedding techniques. The investigation aims to unravel the impact of these encoding and embedding methodologies on the effectiveness of various machine learning models.

This paper, authored by ChatGPT, aims to function as a notification of advancements in encoding methods rather than a comprehensive explanation of overall performance. It is important to note that a robust decoder has not been designed or included in this work. The focus is primarily on elucidating progress made in the realm of encoding techniques, with the understanding that the development of a corresponding decoder is an integral aspect yet to be addressed.

Methodology

To initiate the investigation, the study leveraged the power of NumPy for efficient numerical operations. The initial step involved the implementation of a word splitter, which efficiently separated words in the textual data. Additionally, the Autotokenizer from the Transformers library was employed to facilitate tokenization, a pivotal step in the preprocessing pipeline.

Two distinct datasets, one focused on summarization and the other on sentiment analysis, were imported for experimentation. The training data from these datasets was loaded for further processing. The NLTK word tokenizer was employed to tokenize the datasets, forming the basis for subsequent embedding processes.

The Word2Vec model was utilized to generate semantic embeddings from the tokenized datasets. Simultaneously, a pre-existing sentimental embedder was employed to create datasets embedded with sentimental information. This dual approach facilitated the exploration of both semantic and sentimental dimensions within the datasets.

Encoding Techniques

Sinusoidal Encoding

A traditional Sinusoidal encoder was implemented based on Hugging Face's code. This encoding technique utilizes sine and cosine functions to map position information in a continuous manner. The Sinusoidal encoding method is widely used in Natural Language Processing tasks for its ability to capture sequential information effectively.

$$\text{Sinusoidal Encoding}(pos, i) = \sin\left(\frac{pos}{10000^{2i/d}}\right)$$

Sinusoidal encoding is often used in sequence-to-sequence models, like transformers, to add positional information to the input data. The formula for Sinusoidal encoding generates unique positional embeddings for each position in the input sequence.

pos: *pos* is the position of the token in the sequence.

i: i is the dimension of the encoding.

d: d is the dimensionality of the embedding.

The function generates a Sinusoidal curve for each dimension, and the frequency of the curve varies based on the position and dimension. This allows the model to distinguish between tokens at different positions in the sequence and capture positional information.

The key idea is that each position is represented by a unique combination of sine and cosine values. The use of different frequencies ensures that tokens at different positions have distinct embeddings, preventing them from being treated as identical by the model. Sinusoidal encoding helps the model to be aware of the order and position of tokens in the input sequence, which is crucial for tasks involving sequential data.

Henon Encoding

The Henon encoding technique involves the use of the Henon map formula. This nonlinear mapping system transforms input data into a chaotic, yet deterministic, sequence. The Henon encoder introduces a novel approach to position encoding, offering an alternative perspective for comparison with traditional Sinusoidal encoding.

$$\begin{cases} x_{n+1} &= 1 - a \cdot x_n^2 + y_n \\ y_{n+1} &= b \cdot x_n \end{cases}$$

where (x_n, y_n) represents the current state, (x_{n+1}, y_{n+1}) is the next state, and a and b are constants.

The Henon mapping is utilized as an encoding technique by taking an embedding and a dimensionality as inputs. The formula is applied iteratively, producing a series of values that are used as the encoded representation.

Given a tuple (x_n, y_n) representing the current state, the next state (x_{n+1}, y_{n+1}) is calculated using the Henon mapping formula:

$$\begin{cases} x_{n+1} &= 1 - a \cdot x_n^2 + y_n \\ y_{n+1} &= b \cdot x_n \end{cases}$$

In mathematics, the inequality $x \leq 0 < d$ and $y \leq 0 < \frac{d}{2}$ is represented as:

$$x \leq 0 < d \quad \text{and} \quad y \leq 0 < \frac{d}{2}$$

where a and b are constants, and n represents the iteration step.

In the context of encoding, the Henon mapping takes an embedding and a dimensionality as input parameters. The dimensionality (d) is used to determine the range of values for x, y, and b. Specifically, x is in the range of 0 to d, y is in the range of 0 to $\frac{d}{2}$, and b is determined based on y and x.

The values of a, b, and the initial x are determined based on the sequence length or embedding length. If x is 0, then the square root is taken, and x is set to 1. The parameter b is calculated as y/x.

The resulting values z_i are calculated using the formula, and each index corresponds to xi, y_i, and b values. Any duplicate values in the calculated series are replaced with zeros. The indexes are then stacked to form an array.

An alternative method, considered while writing, involves replacing duplicate values with the count of how many times each value has repeated minus one and to limit the values of x, y, a and b to the dimensionality of the sequence. This alternative approach provides a different strategy for handling repeated values in the encoding process.

Model Selection

Various machine learning models were selected for comparison, including Random Forest Regression, Bagging Forest Regression, K-Nearest Neighbors (KNN), and a Transformer model. Notably, the implementation of Sinusoidal and Henon encoders required tailored transformer architectures to accommodate the specific nature of each encoding technique.

Validation and Evaluation

To ensure the robustness and validity of the investigation, a dedicated validation dataset was curated. This dataset played a crucial role in validating and comparing the results obtained from each model. The models were systematically tested and evaluated to gain insights into their performance under different encoding and embedding conditions.

Summary

In summary, this chapter outlines the methodological approach adopted in the investigation. By comparing Henon and Sinusoidal encoding techniques alongside semantic and sentimental embedding methods, the study aims to shed light on the nuanced aspects of encoding and embedding in the context of machine learning models. The subsequent sections delve into the experimental results and their implications for the broader field of Natural Language Processing.

Transformer Architecture

Sinusoidal Model Summary

Listing 1: Transformer Model Summary

```
Layer (type)                   Output Shape                   Param #
==========================================================================
input_1 (InputLayer)           [(None, 4)]                    0
embedding (Embedding)          (None, 4, 128)                 512
tf.__operators__.add (TFOp     (None, 4, 128)                 0
Lambda)
multi_head_attention (Mult     (None, 4, 128)                 65664
iHeadAttention)
# ... (other layers)
dense_8 (Dense)                (None, 4)                      516
==========================================================================
Total params: 270,356 (1.03 MB)
Trainable params: 270,356 (1.03 MB)
Non-trainable params: 0 (0.00 Byte)
```

0.1 Henon Model Summary

Table 1: Transformer Model Architecture

TransformerModel
Transformer
Encoder
TransformerEncoderLayer (0-1)
MultiheadAttention
Linear (in_features=4, out_features=64)
Dropout (p=0.1)
Linear (in_features=64, out_features=4)
LayerNorm (4)
Dropout (p=0.1)
LayerNorm (4)
Dropout (p=0.1)
... (more layers)
LayerNorm (4)
Decoder
TransformerDecoderLayer (0-1)
MultiheadAttention
MultiheadAttention
Linear (in_features=4, out_features=64)
Dropout (p=0.1)
Linear (in_features=64, out_features=4)
LayerNorm (4)
Dropout (p=0.1)
LayerNorm (4)
Dropout (p=0.1)
LayerNorm (4)
Dropout (p=0.1)
... (more layers)
LayerNorm (4)
Linear (in_features=12, out_features=4)

Statistical Analysis and Model Comparison: Sinusoidal Encoded Sentimental to Sentiment Embedding on Sentimental Data Validated by Summation Data

Data Overview

The dataset comprises four sets: X Training Data, Y Testing Data, Validation Input Data, and Validation Output Data. Additionally, Mean Squared Errors (MSE) for three regression models (Random Forest Regressor, Bagging Forest Regressor, and KNN) are presented, along with the Transformer Data.

Feature Analysis

X Training Data:

- Each row represents a set of 20 features associated with a specific observation.
- Features exhibit varying scales, suggesting potential differences in magnitudes.

Y Testing Data:

- Corresponds to X Training Data, including target values.
- Targets have significantly larger values compared to X Training Data features.

Validation Input and Output Data:

- Validation Input Data: Sets of 20 features similar to X Training Data.
- Validation Output Data: Corresponding target values.
- Some entries in Validation Input Data have extreme values, indicating potential outliers.

Regression Model Performances

Random Forest Regressor:

- MSE ranges from 21.097 to 54.344, indicating moderate to high errors.
- Model performs well on certain samples but struggles with others.

Bagging Regressor:

- MSE ranges from 12.445 to 46.699, showing a lower error range compared to Random Forest Regressor.
- Generally outperforms with lower errors across samples.

KNN:

- Consistently high MSE values (942.53) suggest model struggles to capture underlying patterns.
- Further investigation needed to understand reasons for high error.

Transformer Data:

- Consists of four values, with the first and third columns exhibiting negative values.
- Represents a transformation process or output, requiring additional context for interpretation.

General Observations

- Validation datasets have entries with extreme values, requiring cleaning or normalization.
- Choice of regression model significantly impacts performance, with Bagging Regressor generally outperforming others.
- Features in X Training Data have varying scales, suggesting potential benefit of normalization.

Suggestions

Data Preprocessing:

- Address outliers in validation datasets for accurate model evaluation.
- Normalize or standardize features to ensure models are not biased towards variables with larger magnitudes.

Model Improvement:

- Investigate and fine-tune hyperparameters for Random Forest Regressor and KNN models.
- Explore alternative models or ensemble methods to enhance prediction accuracy.

Further Investigation:

- Understand the nature of the transformation represented in the Transformer Data for better interpretation.

Semantic Embedding: The `X Training Data` showcases a set of 20 features per observation, each with varying scales and values spanning a wide range. Semantic embedding methods can be considered to normalize or standardize these features, ensuring that the models are not biased towards variables with larger magnitudes. Techniques like Principal Component Analysis (PCA) can be explored to embed the semantic information in a reduced-dimensional space, potentially capturing essential features more effectively.

Sentimental Embedding in Y Testing Data: The `Y Testing Data` represents the target values associated with the features in `X Training Data`. These targets exhibit significantly larger values, presenting a sentimental aspect in the context of regression analysis. Sentimental embedding techniques can be employed to understand the sentiment or magnitude associated with each target value. Analyzing the distribution and patterns within the `Y Testing Data` sentimentally enhances the understanding of the impact of features on the target variables.

Embedding Considerations in Validation Data: The `Validation Input Data`, containing sets of 20 features, and the corresponding `Validation Output Data`, presenting target values, require careful embedding considerations. The presence of extreme values in some entries suggests potential outliers or errors, emphasizing the need for semantic embedding methods to address these issues. Sentimental embedding, in this context, aids in understanding the sentiment associated with outliers and their impact on model evaluation.

Regression Model Performances: The Mean Squared Error (MSE) values from `Random Forest Regressor, Bagging Forest Regressor, and KNN` models provide insights into their sentimental performance. Sentimental embedding can help interpret the magnitude of errors and patterns in model performances. For instance, the consistently high MSE values for `KNN` suggest a struggle in capturing underlying patterns sentimentally, highlighting the need for further investigation.

Transformer Data and Embedding: The `Transformer Data`, with its four values, may represent a transformation process or output. Semantic embedding methods, such as attention mechanisms or contextual embeddings, can be applied to understand the semantic relationships between these values. Sentimental embedding here aids in interpreting the sentiment associated with each transformed value, providing a more nuanced understanding of the transformation process.

Conclusion: Incorporating semantic and sentimental embedding methods into the analysis enriches the understanding of the dataset. Semantic embedding ensures feature consistency, while sentimental embedding sheds light on the sentiment associated with target values and model performances. These techniques contribute to a comprehensive analysis, facilitating improved model evaluation and interpretation in the context of regression analysis.

Conclusion

The dataset exhibits challenges typical of regression problems, including varied feature scales and potential outliers. The choice of the regression model plays a crucial role in prediction accuracy, with Bagging Regressor showing promise. Further data preprocessing and model optimization are recommended for improved performance. The Transformer Data requires additional context for meaningful analysis.

Data Tables

Sinusoidal Encoded Semantic to Sentimental Embedding on Sentimental Data Validated by Summation Data

Table 2: X Training Data

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9	Column 10	Column 11	Column 12	Column 13	Column 14	C
3.960	0.331	2.418	1.509	-2.358	-0.120	1.741	3.425	-1.481	1.268	1.290	0.422	0.437	-0.937	
3.118	0.907	2.995	3.121	0.752	0.039	0.037	2.776	-2.311	0.805	0.087	0.577	2.364	-2.871	
1.189	0.116	2.212	0.595	0.711	0.264	0.358	3.173	-0.360	0.237	-0.315	-0.243	1.188	-1.398	
-0.923	-0.494	-0.603	1.631	-1.611	0.451	0.562	1.458	-2.066	-0.137	0.952	-0.612	0.727	-1.677	

X Training Data

The X Training Data consists of 20 features per observation. Descriptive statistics for each feature are summarized below:

Summary Statistics:

- Mean: Varies across columns
- Standard Deviation: Varies across columns
- Range: Varies across columns

The X Training Data appears to have varying scales and distributions, indicating the potential need for normalization or standardization before model training.

Table 3: Y Testing Data

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9	Column 10	Column 11	Column 12	Column 13	Column 14	C
0.176	1.154	0.165	1.181	0.153	1.243	0.178	1.180	0.190	1.130	0.139	1.118	0.137	1.187	
1.276	1.056	0.449	1.455	1.323	1.026	1.331	0.979	0.509	1.435	1.324	1.019	1.280	1.004	
1.691	0.403	0.797	1.797	1.706	0.438	1.726	0.370	0.850	1.752	1.698	0.357	1.668	0.390	
74.153	99.351	79.990	76.244	94.288	58.864	82.464	72.239	78.879	101.114	104.283	120.879	96.492	73.434	

Y Testing Data

The Y Testing Data represents the target values associated with the X Training Data. Descriptive statistics for each target variable are summarized below:

Summary Statistics:

- Mean: Varies across columns
- Standard Deviation: Varies across columns
- Range: Varies across columns

The Y Testing Data exhibits significantly larger values compared to the features in X Training Data, highlighting the potential sentimental aspect in the context of regression analysis.

Table 4: Validation Input Data

1.000	0.811	-0.375	0.732	0.792	0.500	-0.300	0.732	0.694	0.667	0.781	0.678	0.811	0.732	1.000	-0.500	0.645	0.734	1.000	1.000
0.000	0.000	0.625	0.000	0.000	0.500	0.100	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.300	1.000	0.000	1.000	0.000	0.000
1.000	0.901	0.500	0.856	0.890	1.000	0.000	0.856	0.833	0.817	0.884	0.823	0.901	0.856	1.140	0.707	0.803	1.317	1.000	1.000
0.000	0.000	-50.	0.000	0.000	30.	-10.	0.000	0.000	0.000	0.000	0.000	0.000	0.000	9.000	-60.	0.000	40.872	0.000	0.000

Validation Input Data

The Validation Input Data contains sets of 20 features. Descriptive statistics for each feature are summarized below:

Summary Statistics:

- Mean: Varies across columns
- Standard Deviation: Varies across columns
- Range: Varies across columns

The presence of extreme values in some entries suggests potential outliers or errors, emphasizing the need for data preprocessing.

Table 5: Validation Output Data

3.960	-0.669	2.418	0.509	-2.358	-1.120	1.741	2.425	-1.481	0.268	1.290	-0.578	0.437	-1.937	-2.389	0.723	1.779	-1.330	-0.801	-1.427
2.276	0.366	2.985	2.121	-0.089	-0.502	0.027	1.776	-3.153	0.265	0.077	-0.423	1.523	-3.411	-1.678	-0.230	1.682	-1.567	0.521	-0.925
0.279	0.532	2.192	-0.405	-0.198	0.680	0.338	2.173	-1.269	0.653	-0.335	-1.243	0.278	-0.982	0.335	1.294	1.857	-0.763	-1.309	-0.749
-1.064	0.496	-0.633	0.631	-1.752	1.441	0.532	0.459	-2.207	0.853	0.922	-1.612	0.586	-0.687	0.725	-0.226	0.259	-0.476	-2.067	-0.729

Validation Output Data

The Validation Output Data represents target values corresponding to the Validation Input Data. Descriptive statistics for each target variable are summarized below:

Summary Statistics:

- Mean: Varies across columns
- Standard Deviation: Varies across columns
- Range: Varies across columns

Similar to Y Testing Data, the Validation Output Data exhibits larger values, reinforcing the sentimental aspect in the regression context.

Regression Model Performances

The Mean Squared Error (MSE) values for different regression models are summarized below:

Random Forest Regressor:

- MSE: 272.58908

Individual MSE values for each observation vary, indicating the model's performance.

Table 6: Random Forest Regressor Data - Mean Squared Error: 272.58908

26.659	35.230	28.350	27.693	33.706	21.097	29.582	25.733	27.989	36.367	37.197	42.737	34.458	26.163	43.567	34.895	30.837	30.714	54.344	2
24.636	32.186	26.038	25.522	31.084	19.263	27.315	23.496	25.714	33.448	34.276	39.046	31.767	23.892	39.955	32.105	28.455	28.055	49.954	1
26.628	35.245	28.338	27.681	33.675	21.113	29.551	25.749	27.975	36.355	37.166	42.753	34.428	26.179	43.554	34.882	30.807	30.729	54.313	2
27.550	36.137	29.212	28.506	34.803	21.592	30.560	26.362	28.842	37.429	38.395	43.859	35.575	26.806	44.869	35.915	31.849	31.489	56.035	2

Bagging Forest Regressor:

- MSE: 82.54

Like Random Forest, individual MSE values vary, and the model outperforms Random Forest.

Table 7: Bagging Forest Regressor - Mean Squared Error: 82.54

15.577	20.493	16.383	16.440	19.601	12.445	17.255	15.067	16.192	21.376	21.591	24.766	20.020	15.318	25.068	20.542	17.955	17.914	31.377	1
15.729	20.418	16.446	16.502	19.757	12.365	17.409	14.986	16.258	21.438	21.747	24.689	20.173	15.238	25.137	20.605	18.107	17.836	31.535	1
15.577	20.493	16.383	16.440	19.601	12.445	17.255	15.067	16.192	21.376	21.591	24.766	20.020	15.318	25.068	20.542	17.955	17.914	31.377	1
22.975	30.312	24.365	23.946	29.015	18.207	25.483	22.172	24.061	31.374	32.006	36.742	29.656	22.542	37.406	30.115	26.553	26.442	46.699	1

KNN

- MSE: 942.53

KNN exhibits higher MSE values, suggesting challenges in capturing underlying patterns sentimentally.

Table 8: KNN - Mean Squared Error: 942.53

25.340	33.636	26.984	26.407	32.049	20.182	28.123	24.595	26.640	34.665	35.373	40.785	32.766	25.004	41.472
25.340	33.636	26.984	26.407	32.049	20.182	28.123	24.595	26.640	34.665	35.373	40.785	32.766	25.004	41.472
25.340	33.636	26.984	26.407	32.049	20.182	28.123	24.595	26.640	34.665	35.373	40.785	32.766	25.004	41.472
25.340	33.636	26.984	26.407	32.049	20.182	28.123	24.595	26.640	34.665	35.373	40.785	32.766	25.004	41.472

Table 9: Transformer Data

-0.262	1.121	-0.019	1.014
0.949	0.852	0.028	1.116
1.308	0.264	0.745	1.759
2.647	-0.479	1.024	2.382

1 Transformer Data Analysis

The Transformer Data consists of four observations with four features each. These values could represent embedded vectors generated by a transformer-based model. Let's delve into the analysis of this data:

1.1 Individual Feature Analysis

1.1.1 Feature 1:

- Mean: 0.410
- Standard Deviation: 1.221
- Range: -0.262 to 2.647

1.1.2 Feature 2:

- Mean: 0.190
- Standard Deviation: 0.671
- Range: -0.479 to 1.121

1.1.3 Feature 3:

- Mean: 0.195
- Standard Deviation: 0.363
- Range: -0.019 to 0.745

1.1.4 Feature 4:

- Mean: 1.543
- Standard Deviation: 0.661
- Range: 1.014 to 2.382

1.2 Key Observations

- **Magnitude Variation:** Features exhibit a wide range of magnitudes, indicating potential variations in the importance of each feature.
- **Differential Scale:** The standard deviations of features differ, suggesting varying degrees of dispersion around the mean.
- **Asymmetry:** The presence of negative values in features 1, 2, and 3 suggests asymmetric distributions.

1.3 Interpretation

Given the characteristics of the Transformer Data, it is evident that the features encapsulate diverse information. The data appears to be well-suited for tasks involving nuanced relationships and intricate patterns, typical of natural language processing (NLP) applications where transformers excel.

1.4 Considerations for Sentiment Analysis

- **Magnitude as Indicator:** The magnitude of these features might be indicative of the strength or importance of certain semantic aspects within the data.
- **Asymmetry in Features:** The asymmetric distribution in some features could represent the model's sensitivity towards specific sentiment orientations.
- **Normalization for Model Training:** Considering the varying scales, normalization or standardization may be beneficial before utilizing this data for sentiment analysis tasks.

1.5 Future Steps

Further exploration and understanding of the training process that generated these embeddings could provide valuable insights. Additionally, incorporating this transformer-generated data into sentiment analysis models, possibly in conjunction with other features, may enhance the model's ability to discern subtle sentiments and semantic nuances in natural language data.

Embedding Methods and Translation

Semantic to Sentimental Embedding

Semantic Embedding:

- Techniques like Principal Component Analysis (PCA) can be applied to normalize or standardize features in X Training Data.
- Attention mechanisms or contextual embeddings can be employed for understanding semantic relationships in the Transformer Data.

Sentimental Embedding:

- Analyzing the distribution and patterns in Y Testing Data and Validation Output Data provides insights into the sentiment or magnitude associated with target values.
- The Mean Squared Error (MSE) values from regression models, especially Bagging Forest Regressor, offer a sentimental evaluation of model performances.

1.6 Conclusion

Incorporating semantic and sentimental embedding methods enhances the understanding of the dataset. Semantic embedding ensures feature consistency, while sentimental embedding sheds light on the sentiment associated with target values and model performances. These techniques contribute to a comprehensive analysis, facilitating improved model evaluation and interpretation in the context of regression analysis. Further exploration and fine-tuning of models may be considered based on the observed patterns and challenges.

2 Transformer Data Analysis

The Transformer Data consists of four observations with four features each. These values could represent embedded vectors generated by a transformer-based model. Let's delve into the analysis of this data:

2.1 Individual Feature Analysis

2.1.1 Feature 1:

- Mean: 0.410
- Standard Deviation: 1.221
- Range: -0.262 to 2.647

2.1.2 Feature 2:

- Mean: 0.190
- Standard Deviation: 0.671
- Range: -0.479 to 1.121

2.1.3 Feature 3:

- Mean: 0.195
- Standard Deviation: 0.363
- Range: -0.019 to 0.745

2.1.4 Feature 4:

- Mean: 1.543
- Standard Deviation: 0.661
- Range: 1.014 to 2.382

2.2 Key Observations

- **Magnitude Variation:** Features exhibit a wide range of magnitudes, indicating potential variations in the importance of each feature.
- **Differential Scale:** The standard deviations of features differ, suggesting varying degrees of dispersion around the mean.
- **Asymmetry:** The presence of negative values in features 1, 2, and 3 suggests asymmetric distributions.

2.3 Interpretation

Given the characteristics of the Transformer Data, it is evident that the features encapsulate diverse information. The data appears to be well-suited for tasks involving nuanced relationships and intricate patterns, typical of natural language processing (NLP) applications where transformers excel.

2.4 Considerations for Sentiment Analysis

- **Magnitude as Indicator:** The magnitude of these features might be indicative of the strength or importance of certain semantic aspects within the data.
- **Asymmetry in Features:** The asymmetric distribution in some features could represent the model's sensitivity towards specific sentiment orientations.
- **Normalization for Model Training:** Considering the varying scales, normalization or standardization may be beneficial before utilizing this data for sentiment analysis tasks.

2.5 Future Steps

Further exploration and understanding of the training process that generated these embeddings could provide valuable insights. Additionally, incorporating this transformer-generated data into sentiment analysis models, possibly in conjunction with other features, may enhance the model's ability to discern subtle sentiments and semantic nuances in natural language data.

Henon Encoded Semantic to Sentiment Embedding on Summation Data Validated by Sentiment Data

Table 10: X Training Data

4.066	-0.983	2.666	3.768	-0.826	-0.842	1.888	2.539	-2.097	-0.784	2.024	-1.261	-1.145	-0.643	-2.337	2.454	1.756	-0.968	-1.044	-1.229
2.798	-0.078	3.005	3.264	0.476	-0.473	0.520	0.418	-2.871	-1.367	0.338	-0.506	-1.025	-2.385	-1.829	-3.001	2.752	-1.479	-8.633	-1.516
-0.469	0.017	1.717	-0.663	0.548	1.097	0.476	2.397	-0.757	0.442	0.046	-1.822	-5.714	-0.834	0.093	-5.376	3.007	0.008	-18.314	-0.251
-2.501	1.251	-0.142	-0.433	-0.820	2.333	0.643	0.324	-2.261	0.498	0.971	-1.078	-10.562	-0.903	0.775	-10.086	1.173	0.185	-27.629	-1.385

Table 11: Y Testing Data

1.176	0.154	0.165	2.181	1.153	1.243	0.178	0.180	0.190	0.130	0.139	0.118	0.137	0.187	0.120	2.144	1.151	0.667	0.098
0.434	0.516	0.439	1.455	1.481	1.486	0.489	0.438	0.499	0.435	0.483	0.479	-2.561	0.464	0.495	-1.541	1.432	0.987	-7.500
-0.219	0.819	0.777	0.798	1.797	1.854	0.817	0.786	0.830	0.752	0.788	0.773	-6.241	0.807	0.785	-5.223	1.763	1.309	-16.227
72.012	100.341	79.960	74.244	95.147	60.854	82.323	73.224	78.849	100.115	104.142	121.869	85.351	74.424	123.462	85.870	86.989	87.939	127.18

Table 12: Validation Input

3.066	-0.983	2.666	3.768	-0.826	-0.842	1.888	2.539	-2.097	-0.784	2.024	-1.261	-1.145	-0.643	-2.337	2.454	1.756	-0.968	-1.044	-1.229
2.798	-0.078	3.005	3.264	0.476	-0.473	0.520	0.418	-2.871	-1.367	0.338	-0.506	-1.025	-2.385	-1.829	-3.001	2.752	-1.479	-8.633	-1.516
0.531	0.017	1.717	-0.663	0.548	1.097	0.476	2.397	-0.757	0.442	0.046	-1.822	-5.714	-0.834	0.093	-5.376	3.007	0.008	-18.314	-0.251
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
-0.026	0.021	-0.011	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
-0.004	-0.057	-0.009	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 13: Validation Output Data

1.761e-01	1.543e-01	1.648e-01	1.813e-01	1.533e-01	2.434e-01	1.783e-01	1.796e-01	1.899e-01	1.304e-01	1.390e-01	1.179e-01	1.366e-01	1.869e-01
4.344e-01	5.159e-01	4.392e-01	4.548e-01	4.812e-01	4.856e-01	4.893e-01	4.384e-01	4.992e-01	4.351e-01	4.825e-01	4.791e-01	4.388e-01	4.637e-01
7.813e-01	8.187e-01	7.772e-01	7.976e-01	7.966e-01	8.538e-01	8.170e-01	7.861e-01	8.302e-01	7.520e-01	7.884e-01	7.727e-01	7.586e-01	8.066e-01
7.401e+01	1.003e+02	7.996e+01	7.524e+01	9.415e+01	5.985e+01	8.232e+01	7.322e+01	7.885e+01	1.001e+02	1.041e+02	1.219e+02	9.635e+01	7.442e+01

Table 14: Random Forest Regressor - Mean Squared Error: 477.30511

14.426	19.289	15.418	...	17.345	21.795	11.949
20.647	28.379	22.667	...	25.270	31.437	17.526
13.924	19.529	15.638	...	17.576	15.918	12.166
...	...	...	...	...	...	...
20.285	28.552	22.826	...	25.437	27.192	17.683
20.285	28.552	22.826	...	25.437	27.192	17.683
20.285	28.552	22.826	...	25.437	27.192	17.683

Table 15: Bagging Regressor - Mean Squared Error: 745.55453

Data Point 1	Data Point 2	Data Point 3	Data Point 4	Data Point 5	Data Point 6
1.03659889	0.22069853	0.22604074	...	0.73144992	-1.53453209
22.14787629	30.34306612	24.2259463	...	26.97709101	34.9577682
21.86892542	30.4759465	24.3484275	...	27.10547319	31.69283954
...	...	...	...	...	...
21.86892542	30.4759465	24.3484275	...	27.10547319	31.69283954

Table 16: KNN - Mean Squared Error: 915.709642

Data Point 1	Data Point 2	Data Point 3	Data Point 4	Data Point 5	Data Point 6
24.32310381	33.77120053	26.9674521	...	29.97172913	37.01723405
20.83530076					
24.32310381	33.77120053	26.9674521	...	29.97172913	37.01723405
20.83530076					
24.32310381	33.77120053	26.9674521	...	29.97172913	37.01723405
20.83530076					
...	...	...	...	...	...
...					
24.32310381	33.77120053	26.9674521	...	29.97172913	37.01723405
20.83530076					

Table 17: Transformer Output

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
7.132752	-0.96560585	6.3315983	...	-1.9352708	-1.0889103
-1.4583516					
6.5960684	0.8447403	7.00968	...	-2.9573343	-0.26662195
-2.0313132					
2.0629716	1.034154	4.4330482	...	0.01540482	-1.6281006
0.49893004					
...	...	...	...	...	...
...					
0.9473946	1.0412669	0.9784208	...	0.9394098	0.8793903
0.9243791					
0.9916838	0.8861898	0.98253703	...	0.97589767	1.0124927
1.0755095					

General Performance Investigation

Random Forest Regressor: The Random Forest Regressor exhibits a Mean Squared Error of 477.30511. The output values seem continuous and follow a similar structure to other regressors.

Bagging Regressor: The Bagging Regressor, with a Mean Squared Error of 745.55453, produces output similar in structure to the Random Forest Regressor. Deviations in performance should be analyzed.

KNN Regressor: The KNN Regressor, with a Mean Squared Error of 915.709642, also provides output resembling the other regressors. Further investigation is needed to understand the differences in performance.

Transformer Output: The Transformer Output presents transformed features. Each column represents a distinct transformation or feature. Careful examination of these features and their relevance is essential for interpreting the model's performance.

General Observations: - Some tables have ellipses, indicating potential data point or feature omissions. - Consistency in the number of data points and features across tables should be verified. - Outliers or inconsistent scaling in output values (e.g., Y Testing Data) require investigation. - Cross-checking the match between input features for validation and training is crucial.

Further investigation, including data visualization and summary statistics, would be needed to gain a more comprehensive understanding and identify potential issues or patterns in the data.

General Performance Investigation

Random Forest Regressor: The Random Forest Regressor exhibits a Mean Squared Error of 477.30511. The output values seem continuous and follow a similar structure to other regressors.

Bagging Regressor: The Bagging Regressor, with a Mean Squared Error of 745.55453, produces output similar in structure to the Random Forest Regressor. Deviations in performance should be analyzed.

KNN Regressor: The KNN Regressor, with a Mean Squared Error of 915.709642, also provides output resembling the other regressors. Further investigation is needed to understand the differences in performance.

Transformer Output: The Transformer Output presents transformed features. Each column represents a distinct transformation or feature. Careful examination of these features and their relevance is essential for interpreting the model's performance.

General Observations: - Some tables have ellipses, indicating potential data point or feature omissions. - Consistency in the number of data points and features across tables should be verified. - Outliers or inconsistent scaling in output values (e.g., Y Testing Data) require investigation. - Cross-checking the match between input features for validation and training is crucial.

Further investigation, including data visualization and summary statistics, would be needed to gain a more comprehensive understanding and identify potential issues or patterns in the data.

2.6 Data Overview

2.6.1 X Training Data

The X Training Data consists of 4 rows and 20 columns, representing features used for training the regression models.

- **Mean:** The mean values for each feature in the X Training Data are computed. These values provide a central tendency measure for the dataset.
- **Skewness:** Skewness is a measure of the asymmetry of the data distribution. It indicates whether the data is skewed towards higher or lower values.
- **Standard Deviation:** The standard deviation measures the amount of variation or dispersion in the dataset. It provides insights into the spread of data points.

2.6.2 Y Testing Data

The Y Testing Data consists of 4 rows and 20 columns, representing the corresponding target values for the testing dataset.

- **Mean:** Calculate the mean values for each target variable to understand the central tendency of the Y Testing Data.
- **Skewness:** Evaluate the skewness of the target variable distributions to assess asymmetry.
- **Standard Deviation:** Compute the standard deviation to measure the variability of target values.

2.6.3 Validation Input and Output Data

Similar statistical measures (mean, skewness, and standard deviation) can be applied to the Validation Input and Output Data to understand their characteristics.

2.7 Regression Models Evaluation

2.7.1 1. Random Forest Regressor

The Random Forest Regressor yielded a Mean Squared Error of 477.30511. Additionally, statistical measures for the predictions are analyzed:

- **Mean Prediction:** Calculate the mean of the predicted values for each data point.
- **Skewness of Predictions:** Assess the skewness of the distribution of predicted values.
- **Standard Deviation of Predictions:** Evaluate the variability in the model's predictions.

2.7.2 2. Bagging Regressor

The Bagging Regressor yielded a Mean Squared Error of 745.55453. Similar statistical measures are calculated for its predictions.

2.7.3 3. KNN

The KNN model yielded a Mean Squared Error of 915.709642. Analyze the mean, skewness, and standard deviation of the KNN predictions.

2.7.4 4. Transformer Output

The Transformer model's output is presented in the table. Analyze the mean, skewness, and standard deviation of the Transformer predictions.

2.8 Semantic to Sentimental Embedding Translation

The concept of semantic to sentimental embedding translation refers to the transformation of semantically meaningful information into sentimental representations. In the context of regression models, it implies that the models are capturing and translating semantic features into sentiment-related predictions. Understanding the translation process and its accuracy is crucial for assessing the model's ability to capture nuanced relationships in the data.

2.9 Conclusion

This detailed statistical analysis provides insights into the central tendencies, skewness, and variability of the datasets and model predictions. Understanding these statistical properties is essential for assessing the models' performance and gaining insights into the semantic to sentimental embedding translation. Further exploration of specific features and their impact on the translation process can enhance the overall understanding of the models and their applications.

Sinusoidal to Henon Encoding on Semantic Embedded Data Trained on Summation Data Validated by Sentiment Data

Table 18: X Training Data

2.940	0.006	2.672	2.947	-2.121	-0.844	2.502	3.103	-2.146	0.611	1.691	0.741	0.421	-0.065	-2.213	1.695	0.974	-0.909	-0.654	-0.411
3.299	0.697	3.266	2.770	0.786	-1.193	0.369	1.748	-1.949	0.083	0.008	0.449	3.399	-1.656	-1.433	0.662	2.806	-1.539	-0.533	-0.244
1.412	0.407	1.922	0.265	0.272	-0.392	0.403	3.386	0.152	-0.129	0.682	0.141	2.101	-0.768	0.052	1.967	2.909	-1.394	-1.216	0.311
-0.451	-0.159	-0.251	1.553	-1.663	0.418	0.144	1.206	-2.108	-0.420	0.899	-0.389	0.740	-1.765	0.985	0.850	0.619	-1.241	-1.649	-0.155

Table 19: Y Testing Data

3.940	-0.995	2.672	3.947	-1.121	-0.844	2.502	2.103	-2.146	-0.389	1.691	-0.259	0.421	-1.065	-2.213	2.695	1.974	-1.409	-0.654	-1.411
2.458	0.156	3.256	2.770	0.945	-0.733	0.359	0.748	-2.791	-0.457	-0.002	-0.551	-0.443	-2.196	-1.443	-2.338	2.964	-1.580	-8.543	-1.244
-0.497	0.823	1.902	-0.735	0.363	1.024	0.383	2.386	-0.758	0.288	0.662	-0.859	-5.809	-0.352	0.032	-5.032	2.999	-0.478	-18.236	-0.689
-2.592	0.831	-0.281	-0.446	-0.805	2.408	0.114	0.206	-2.249	0.570	0.869	-1.388	-10.401	-0.775	0.955	-10.150	1.478	0.249	-27.679	-1.154

Table 20: Validation Input Data

-0.004	1.000	0.026	1.046	-0.047	0.964	0.032	1.048	-0.026	0.982	0.040	0.993	-0.022	1.030	-0.022	0.990	0.018	1.007	-0.043	0.951
0.877	0.565	0.044	1.004	0.873	0.523	0.005	1.030	0.804	0.522	-0.026	0.995	0.889	0.502	-0.000	0.990	0.884	0.512	0.009	0.975
0.860	-0.392	-0.023	0.978	0.909	-0.417	-0.018	1.050	0.934	-0.369	-0.019	1.022	0.889	-0.414	0.064	0.977	0.934	-0.449	0.002	1.046
0.131	-0.989	0.010	0.962	0.133	-0.978	0.026	1.031	0.127	-0.977	0.061	1.041	0.135	-1.039	0.055	1.002	0.183	-0.992	0.015	0.954

Table 21: Validation Expected Output Data

0.996	0.001	0.026	2.046	0.953	0.964	0.032	0.048	-0.025	-0.018	0.040	-0.008	-0.022	0.030	-0.022	1.990	1.018	0.507	-0.043	-0.049
0.036	0.025	0.034	1.004	1.032	0.983	-0.005	0.030	-0.038	-0.019	-0.036	-0.005	-2.952	-0.038	-0.010	-2.010	1.042	0.471	-8.001	-0.025
-1.049	0.025	-0.043	-0.021	1.000	1.000	-0.038	0.050	0.024	0.047	-0.039	0.023	-7.020	0.002	0.044	-6.023	1.025	0.467	-17.018	0.046
-2.010	0.001	-0.020	-1.038	0.992	1.012	-0.004	0.031	-0.014	0.013	0.031	0.042	-11.007	-0.049	0.025	-10.000	1.042	0.498	-26.015	-0.046

Table 22: Random Forest Regressor - Mean Squared Error: 3.6386860354649215

0.0824	0.2626	1.3111	0.8262	-0.5407	0.9711	0.9379	1.4701	-1.7548	0.1849	1.0595	-0.8740	-5.6227	-0.7337	-0.3134	-4.5815	2.1031	-0.4902	-16.37
0.0565	0.2987	1.3612	0.7239	-0.4760	0.9669	0.9036	1.5412	-1.6823	0.1900	1.0327	-0.8701	-5.6095	-0.7067	-0.2962	-4.5825	2.1693	-0.4934	-16.44
0.0356	0.2988	1.3394	0.7268	-0.4876	0.9808	0.9009	1.5194	-1.6972	0.1928	1.0348	-0.8754	-5.6554	-0.7110	-0.2870	-4.6337	2.1541	-0.4862	-16.53
0.1059	0.2445	1.2970	0.8759	-0.5672	0.9662	0.9564	1.4455	-1.7836	0.1810	1.0719	-0.8733	-5.6063	-0.7450	-0.3266	-4.5554	2.0777	-0.4923	-16.29

Mean and Standard Deviation:

- The mean values for each feature provide an average measure of the data.
- The standard deviation indicates the amount of variation or dispersion present in the data.

Feature Analysis:

- Each column in the data represents a different feature.
- Features appear to have varying scales, and some values are positive while others are negative.

Outliers:

- Outliers can significantly affect regression models.
- Detecting and handling outliers is crucial for model robustness and accuracy.

Correlation:

- Analyzing correlations between features and the target variable can provide insights into the relationships within the data.

Prediction Confidence:

- The Mean Squared Error (MSE) of 3.64 indicates the average squared difference between the actual and predicted values.
- A lower MSE suggests better model performance, but it should be interpreted relative to the specific problem domain.

Further Investigation:

- Visualizations, such as scatter plots or residual plots, can aid in understanding the model's predictive performance and identifying patterns or issues.
- It's recommended to visualize the data and perform additional statistical tests for a more comprehensive analysis. Further exploration and refinement may be needed based on the context of the regression problem.

Table 23: Bagging Regressor - Mean Squared Error: 1.117

0.0824	0.2626	1.3111	0.8262	-0.5407	0.9711	0.9379	1.4701	-1.7548	0.1849	1.0595	-0.8740	-5.6227	-0.7337	-0.3134	-4.5815	2.1031	-0.4902	-16.37
0.0565	0.2987	1.3612	0.7239	-0.4760	0.9669	0.9036	1.5412	-1.6823	0.1900	1.0327	-0.8701	-5.6095	-0.7067	-0.2962	-4.5825	2.1693	-0.4934	-16.44
0.0356	0.2988	1.3394	0.7268	-0.4876	0.9808	0.9009	1.5194	-1.6972	0.1928	1.0348	-0.8754	-5.6554	-0.7110	-0.2870	-4.6337	2.1541	-0.4862	-16.53
0.1059	0.2445	1.2970	0.8759	-0.5672	0.9662	0.9564	1.4455	-1.7836	0.1810	1.0719	-0.8733	-5.6063	-0.7450	-0.3266	-4.5554	2.0777	-0.4923	-16.29

Statistical Analysis: Bagging Regressor

- **Mean Squared Error (MSE):** 1.117
- **Data:**
  - The data consists of 20 columns, each representing a different feature or variable.
  - Values vary across features, with both positive and negative values.
  - Feature values seem to have different scales.
- **Interpretation:**
  - The MSE of 1.117 suggests relatively low prediction error on average.
  - Each row in the data represents a different data point, and each column represents a feature.
  - Positive and negative values in the features indicate variations in the predictors.
- **Further Analysis:**

  - Feature importance analysis could help identify which features contribute more to the predictions.
  - Residual analysis may provide insights into the model's performance and areas for improvement.
  - Visualization tools, such as scatter plots or residual plots, can aid in understanding the relationships between features and predictions.

- **Recommendations:**
  - Evaluate the model's performance on a separate test set.
  - Consider cross-validation to assess the model's generalizability.
  - Explore potential interactions or non-linear relationships between features.

Table 24: KNN - Mean Squared Error: 5.0136

0.284	0.220	1.431	0.922	-0.521	0.863	1.000	1.565	-1.718	0.156	1.074	-0.835	-5.263	-0.731	-0.409	-4.162	2.150	-0.546	-15.523	-1.085
0.284	0.220	1.431	0.922	-0.521	0.863	1.000	1.565	-1.718	0.156	1.074	-0.835	-5.263	-0.731	-0.409	-4.162	2.150	-0.546	-15.523	-1.085
0.284	0.220	1.431	0.922	-0.521	0.863	1.000	1.565	-1.718	0.156	1.074	-0.835	-5.263	-0.731	-0.409	-4.162	2.150	-0.546	-15.523	-1.085
0.284	0.220	1.431	0.922	-0.521	0.863	1.000	1.565	-1.718	0.156	1.074	-0.835	-5.263	-0.731	-0.409	-4.162	2.150	-0.546	-15.523	-1.085

Data:

- The dataset consists of 20 columns, each representing a different feature or variable.
- Feature values seem to vary with both positive and negative values.
- The same data appears to be repeated in each row, potentially an error or artifact.

Interpretation:

- The high MSE of 5.0136 indicates a relatively high prediction error on average.
- Each column in the data represents a feature, and the repeated rows may suggest duplication or data entry issues.
- The positive and negative values in the features indicate variations in the predictors.

Potential Issues:

- Investigate the reason for the repeated data points and consider removing duplicates.
- Assess whether the input features align with the model requirements and expectations.
- Check for outliers or extreme values in the data.

Further Analysis:

- Conduct exploratory data analysis to identify patterns or trends.
- Evaluate the impact of potential outliers on model performance.
- Consider refining the feature set or adjusting model hyperparameters.

Recommendations:

- Address data duplication issues to ensure a clean dataset.
- Re-run the model after addressing potential data quality concerns.
- Explore alternative algorithms or adjust hyperparameters for better performance.

Table 25: Transformer Data

0.992	3.001	1.052	3.092	0.907	2.929	1.064	3.096	0.949	2.964	1.080	2.985	0.956	3.061	0.957	2.981	1.036	3.014	0.914	2.902
2.754	2.131	1.088	3.008	2.746	2.046	1.010	3.061	2.607	2.043	0.948	2.990	2.779	2.005	1.000	2.980	2.768	2.023	1.019	2.950
2.720	0.217	0.953	2.957	2.818	0.165	0.964	3.100	2.868	0.261	0.963	3.045	2.778	0.173	1.129	2.954	2.869	0.102	1.003	3.091
1.263	-0.978	1.020	2.924	1.267	-0.956	1.051	3.061	1.254	-0.955	1.122	3.082	1.269	-1.078	1.110	3.004	1.365	-0.984	1.031	2.907

- **Data:**
  - The dataset consists of 20 columns, each representing a different feature or variable.
  - Features exhibit a wide range of values, including both positive and negative.
- **Interpretation:**
  - The values in each column likely represent the output of different transformations or features generated by the transformer.
  - Features seem to vary significantly, and their interpretation would depend on the specific context of the transformer architecture.
- **Variability:**
  - The data shows variability in the transformed features, as indicated by the range of values.
  - Both positive and negative values suggest diverse transformations applied by the architecture.
- **Patterns:**
  - Patterns or trends in the data may reveal insights into the behavior of the transformer architecture.
  - Consider visualizations, such as histograms or line plots, to explore the distribution of values across features.
- **Further Analysis:**
  - Conduct exploratory data analysis to identify patterns, outliers, or any unusual behavior in the transformed features.
  - Consider comparing the transformed features with the target variable or other relevant metrics.
- **Recommendations:**
  - Collaborate with domain experts to interpret the meaning of the transformed features.
  - Evaluate the impact of the transformer output on the overall model performance.
  - Explore additional analyses or visualizations to gain a more in-depth understanding of the transformer's contributions.

Henon to Sinusoidal Encoding on Sentiment Embedded Data Trained on Sentiment Data Validated by Summation Data

Table 26: X Training Data

2	0.811	-0.375	2.732	1.792	1.5	-0.3	0.732	0.694	0.667	0.781	0.678	0.811	0.732	1	1.5	1.645	1.234	1	1
0	0	0.625	1	1	1.5	0.1	0	0	0	0	0	-3	0	0.3	-1	1	1.5	-8	0
0	0.901	0.5	0.856	1.89	2	0	0.856	0.833	0.817	0.884	0.823	-6.099	0.856	1.14	-5.293	1.803	1.817	-16	1
-2	0	-50	-1	1	31	-10	0	0	0	0	0	-11	0	9	-70	1	41.372	-26	0

The X Training Data represents a multi-dimensional space with 20 features. Mathematically, let X_{ij} denote the entry in the i-th row and j-th column of this matrix. A preliminary examination reveals significant variation, including integer, fractional, and decimal values. This diversity prompts consideration for normalization to mitigate the impact of scale during regression model training.

Table 27: Y Testing Data

0.176	1.154	0.165	1.181	0.153	1.243	0.178	1.18	0.19	1.13	0.139	1.118	0.137	1.187	0.12	1.144	0.151	1.167	0.098
1.276	1.056	0.449	1.455	1.323	1.026	1.331	0.979	0.509	1.435	1.324	1.019	1.28	1.004	0.505	1.459	1.273	1.028	1.342
1.691	0.403	0.797	1.797	1.706	0.438	1.726	0.37	0.85	1.752	1.698	0.357	1.668	0.39	0.805	1.776	1.672	0.393	1.683
74.153	99.351	79.99	76.244	94.288	58.864	82.464	72.234	78.879	101.114	104.283	120.879	96.492	73.434	123.492	96.87	86.131	86.449	153.322

Table illustrates the Y Testing Data, embodying the expected outcomes corresponding to input features. Mathematically, let Y_i denote the i-th element in this vector. Notably, Y presents a magnitude considerably larger than X, emphasizing the necessity for appropriate scaling during the training and evaluation processes.

Table 28: Validation Input Data

1.176	0.154	0.165	2.181	1.153	1.243	0.178	0.179	0.19	0.13	0.139	0.118	0.137	0.187	0.12	2.144	1.151	0.667	0.098
0.434	0.516	0.439	1.455	1.481	1.486	0.489	0.438	0.499	0.435	0.483	0.479	-2.561	0.464	0.495	-1.541	1.432	0.987	-7.5
-0.219	0.819	0.777	0.798	1.797	1.854	0.817	0.786	0.83	0.752	0.788	0.773	-6.241	0.807	0.785	-5.223	1.763	1.309	-16.227
72.012	100.341	79.96	74.244	95.147	60.854	82.322	73.224	78.849	100.114	104.142	121.869	85.351	74.424	123.462	85.87	86.989	87.939	127.18

Validation Input Data constitutes another set of input features, denoted as $X_{\mathrm{val}_{ij}}$. These features mirror the diversity observed in X Training Data, necessitating alignment in normalization strategies during model evaluation.

Table 29: Validation Expected Output Data

0.176	1.154	0.165	1.181	0.153	1.243	0.178	1.18	0.19	1.13	0.139	1.118	0.136	1.187	0.12	1.144	0.151	1.167	0.098
1.276	1.056	0.449	1.455	1.323	1.026	1.331	0.979	0.509	1.435	1.324	1.019	1.28	1.004	0.505	1.459	1.273	1.028	1.342
1.691	0.403	0.797	1.797	1.706	0.438	1.726	0.37	0.85	1.752	1.698	0.356	1.668	0.391	0.804	1.776	1.672	0.393	1.683
74.153	99.351	79.99	76.244	94.288	58.864	82.463	72.234	78.879	101.114	104.283	120.879	96.492	73.434	123.492	96.87	86.131	86.449	153.322

The Validation Expected Output Data, denoted as Y_{val_i}, encapsulates the anticipated outcomes for the corresponding validation inputs. The magnitude and range of these values parallel those in Y Testing Data, accentuating the importance of consistent scaling during model evaluation.

Table 30: Random Forest Regression Data - Mean Squared Value: 1357.35

28.530	38.349	30.600	29.804	36.173	23.011	31.694	28.051	30.198	39.224	39.963	46.505	36.997	28.513	47.111	37.621	33.066	33.450	58.577	2
4.314	5.846	4.340	5.113	5.310	3.891	4.741	4.498	4.316	6.310	5.798	6.885	5.398	4.568	6.487	6.114	4.891	5.207	8.219	3
6.203	7.607	6.107	6.781	7.612	4.826	6.805	5.700	6.068	8.477	8.302	9.075	7.739	5.798	9.140	8.199	7.021	6.703	11.711	4
10.550	13.544	10.858	11.247	13.167	8.331	11.649	10.012	10.750	14.439	14.457	16.306	13.428	10.181	16.501	13.904	12.089	11.867	20.809	8

The Random Forest Regression Data showcases predictions with a mean squared value of 1357.35. Mathematically, if $\hat{Y}_{\mathrm{RF}_i}$ represents the i-th predicted outcome, the mean squared error (MSE) is calculated as:

$$\mathrm{MSE}_{\mathrm{RF}} = \frac{1}{n}\sum_{i=1}^{n}(Y_{\mathrm{val}_i} - \hat{Y}_{\mathrm{RF}_i})^2$$

The wide-ranging values indicate significant variance in predictions, motivating a detailed comparative analysis against the Validation Expected Output Data.

Table 31: Bagging Regression Data - Mean Squared Value: 760.81

22.672	30.463	24.239	23.823	28.704	18.369	25.174	22.334	23.929	31.250	31.694	36.894	29.350	22.702	37.269	29.988	26.249	26.597	46.382	1
0.782	0.854	0.418	1.428	0.774	0.921	0.798	0.856	0.454	1.379	0.762	0.813	0.749	0.868	0.394	1.397	0.759	0.858	0.732	0
0.933	0.778	0.481	1.489	0.930	0.841	0.952	0.775	0.520	1.441	0.918	0.737	0.902	0.789	0.463	1.460	0.911	0.780	0.890	0
15.577	20.493	16.383	16.440	19.601	12.445	17.255	15.067	16.192	21.376	21.591	24.766	20.020	15.318	25.068	20.542	17.955	17.914	31.376	1

The Bagging Regression Data yields predictions with a mean squared value of 760.81. Similarly, if $\hat{Y}_{\mathrm{Bag}_i}$ denotes the i-th predicted outcome, the MSE for Bagging Regression ($\mathrm{MSE}_{\mathrm{Bag}}$) is computed analogously. The spread of predicted values warrants a meticulous examination in conjunction with the Validation Expected Output Data.

Table 32: KNN - Mean Squared Value: 942.53

25.340	33.636	26.984	26.408	32.049	20.182	28.123	24.595	26.640	34.665	35.373	40.785	32.766	25.004	41.472	33.263	29.318	29.336	51.701	2
25.340	33.636	26.984	26.408	32.049	20.182	28.123	24.595	26.640	34.665	35.373	40.785	32.766	25.004	41.472	33.263	29.318	29.336	51.701	2
25.340	33.636	26.984	26.408	32.049	20.182	28.123	24.595	26.640	34.665	35.373	40.785	32.766	25.004	41.472	33.263	29.318	29.336	51.701	2
25.340	33.636	26.984	26.408	32.049	20.182	28.123	24.595	26.640	34.665	35.373	40.785	32.766	25.004	41.472	33.263	29.318	29.336	51.701	2

KNN Data reports a mean squared value of 942.53. The associated MSE ($\mathrm{MSE}_{\mathrm{KNN}}$) involves the comparison between Y_{val_i} and the i-th predicted outcome $\hat{Y}_{\mathrm{KNN}_i}$. The diversity in predictions prompts a meticulous assessment against the Validation Expected Output Data.

Table 33: Transformer Data

3.352	1.309	1.33	5.363	3.307	3.487	1.357	1.359	1.38	1.261	1.278	1.236	1.273	1.374	1.241	5.288	3.301	2.
1.869	2.032	1.878	3.91	3.962	3.971	1.978	1.877	1.998	1.87	1.965	1.958	-4.122	1.927	1.991	-2.081	3.863	2.
0.563	2.637	2.554	2.595	4.593	4.708	2.634	2.572	2.66	2.504	2.577	2.545	-11.483	2.613	2.569	-9.447	4.526	3.
145.024	201.681	160.921	149.488	191.295	122.709	165.645	147.448	158.698	201.229	209.284	244.738	171.702	149.848	247.924	172.741	174.979	176

Table presents the Transformer Data, characterized by diverse values with a mean squared value yet to be provided. Evaluation entails comparing the i-th predicted outcome $\hat{Y}_{\mathrm{T}_i}$ with Y_{val_i}, guided by $\mathrm{MSE}_{\mathrm{T}}$. A detailed analysis will elucidate the performance of the transformer architecture.

Conclusion

In conclusion, this research delved into a comprehensive comparative analysis between semantic and sentimental embeddings, as well as Henon and sinusoidal encodings, within the context of machine learning models. Our investigation has shed light on several key findings that bear significance in the realm of encoding methods.

Firstly, the bagging regressor emerged as the optimal model across various experimental scenarios, showcasing its robustness and adaptability in handling both semantic and sentimental embeddings, as well as Henon and sinusoidal encodings. This finding underscores the versatility of ensemble learning techniques in capturing the intricacies of different embedding and encoding paradigms.

Moreover, the translational dynamics between sinusoidal and Henon encodings were explored, revealing a noteworthy observation: the translation from sinusoidal to Henon encoding proved to be notably more straightforward across all machine learning models, highlighting a potential advantage in ease of adaptation for Henon encoding in practical applications.

In terms of performance, the sinusoidal encoding consistently demonstrated superior results in comparison to the Henon encoding, except in cases involving transformer models. Surprisingly, in the domain of transformers, the Henon model exhibited a marginal performance gain, outperforming the sinusoidal model. This nuanced observation suggests that the choice between encoding methods may be context-dependent, particularly when considering the specialized architecture of transformer models.

Overall, the insights derived from this research contribute valuable knowledge to the ongoing discourse on encoding methodologies. The nuanced performance variations observed across models underscore the need for careful consideration of both the task at hand and the specific characteristics of the machine learning architecture employed. As we navigate the ever-evolving landscape of natural language processing and machine learning, these findings provide a solid foundation for further exploration and refinement of encoding strategies to enhance model performance and adaptability.
The code can be found at:
https://github.com/jamellknows/positional_encoding.

References

[1] ChatGPT3.5

[2] @onlinehuggingface, author = Hugging Face, title = Transformers - State-of-the-art Natural Language Processing for TensorFlow 2.0 and PyTorch, year = 2023, url = https://huggingface.co/, note = Accessed: 11/12/2023

email: jamellsamuels@googlemail.com