Artificial Intelligence and Electrical & Electronics Engineering: AIEEE Open Access ISSN: 3069-0706

PRIME OPEN ACCESS

Volume 1, Issue 2

Research Article

Date of Submission: 21 October, 2025
Date of Acceptance: 11 November, 2025
Date of Publication: 21 November, 2025

Text2Price: Deep Learning Model for Predicting Electronic Product Prices from Descriptive Text Sequences

Nashwan Al Majmar, Waheeb Al Marhabi* and Akram Alsubari

Computer Science and Information Technology Science Faculty, Ibb University, and Department of Computing, Aljazeera University, Ibb, Yemen

***Corresponding Author:**
Waheeb Al Marhabi, Computer Science and Information Technology Science Faculty, Ibb University, and Department of Computing, Aljazeera University, Ibb, Yemen

Citation: Marhabi, W. A., Majmar, N. A., Alsubari, A. (2025). Text2Price: Deep Learning Model for Predicting Electronic Product Prices from Descriptive Text Sequences. *Art Intelligence and Ele & Electronics Eng: AIEEE Open Access, 1*(2), 01-09.

Abstract
This study investigates deep learning models for predicting electronic product prices through Text Sequence (Text2Price). It examines the performance of these models in price prediction, factors influencing predictions, the model's comprehension of numerical and expressive text, and the efficacy of the developed price prediction model. The primary aim is to create a model skilled at forecasting product prices using textual sequences containing product names, brands, and features. Methodologically, the research employs the T5-BESD model, a transformer-based architecture trained on a dataset of 22,000 electronic products from Amazon. Data preprocessing involves cleaning it and Integrate features to create Text Sequence. In the model definition phase, a custom neural network architecture, T5Regressor, predicts product prices from textual descriptions. This architecture comprises a transformer-based language model (T5) and a linear regression layer. The T5 model comprehends and encodes the input text, while the linear regression layer predicts the numerical output (price). The linear regression layer involves a transformation with a weight matrix W and bias vector b. Additionally, the L1 loss, used for training, measures the absolute difference between predicted and true values. In the initialization and setup phase, critical components, including the optimizer (Adam_W), learning rate scheduler, and loss function, are initialized. The learning rate scheduler dynamically adjusts the learning rate during training, incorporating a warm-up phase. Results demonstrate a consistent accuracy improvement from 38.48% to 54.86% over five epochs, with test accuracy reaching 52.38%. Mean Squared Error decreases from 45057.29 to 19783.88, indicating enhanced prediction accuracy, and Mean Absolute Error drops from 66.87 to 47.34, reflecting reduced disparities between predicted and actual values. The research concludes by providing insights into the T5-BESD model's effectiveness, emphasizing the importance of comprehensive data and suggesting avenues for improvement.

Keywords: Deep Learning Models, T5regressor, Text2price, Product Price Prediction

Introduction
The electronic commerce (e-commerce) landscape, particularly in the domain of electronic products, has witnessed unprecedented growth fueled by technological advancements and changing consumer preferences. With the proliferation of online retail platforms such as Amazon, eBay, and Tesco, the competition in this space has become increasingly fierce. A pivotal aspect of this competition is the pricing strategy adopted by sellers, making accurate price predictions a crucial element for success. This research endeavors to explore the application of deep learning models in predicting the prices of electronic products, employing a novel approach known as text-to-price (text2price). By converting textual descriptions into price predictions, this research aims to bridge the gap between product specifications and accurate

pricing, considering factors such as the model's comprehension of numerical information and expressive language within the text.

The primary objective of this paper is to provide a comprehensive understanding of the effectiveness of deep learning models in predicting electronic products prices. This involves constructing a model that can forecast prices based on textual descriptions, amalgamating product names, brands, and features. The research seeks to contribute valuable insights into the capabilities, limitations, and potential enhancements of text-to-price models in the context of e-commerce. While acknowledging the potential of deep learning models, it is imperative to recognize the limitations faced in terms of data accessibility. The study relies on a text-to-text model designed for textual sequences, underscoring the need for more extensive commercial data to achieve optimal results. The literature review explores the evolution of e-commerce giants like Amazon and the critical role played by pricing in consumer decision-making. It highlights the shift towards data-driven decision-making in various domains and emphasizes the relevance of machine learning techniques, particularly in predicting product prices for informed consumer choices and optimized seller profits.

The methodology section delineates the systematic approach employed for data collection, preprocessing, and text sequencing. Leveraging the Amazon Seller Scraper tool, the research we collected data on 22,000 electronic products, with subsequent steps ensuring data quality and relevance for accurate modeling. As we delve into the core of the research, the model training section outlines the steps involved in training the T5-BESD model. Pre-training on a large text corpus, fine-tuning with task-specific datasets, and the definition of a custom neural network architecture are key components in this phase. The performance results are presented, offering a comprehensive overview of the model's accuracy, mean squared error, and mean absolute error across 5 training epochs. In conclusion, this research contributes to the growing body of knowledge in e-commerce pricing strategies by investigating the potential of text-to-price models. The insights gained from this study may pave the way for advancements in predicting electronic product prices, facilitating more informed decision-making for both consumers and sellers in the ever-evolving e-commerce landscape.

Research Question and Limitations

Can a deep learning model effectively predict the prices of electronic products by converting textual descriptions (text2price), considering factors influencing predictions, the model's comprehension of numbers and expressive letters in text, and assessing the accuracy and efficiency of the developed price prediction model?

The main purpose of this research paper was to provide an insightful view on the use of deep learning models for predicting electronic product prices. This was achieved by constructing a model to forecast the price of a product based on a textual sequence representing the specifications and features of the product. To aid in predicting the price of a product, a descriptive text was generated for each product by combining its name, brand, and product features.

These textual sequences were then utilized by the model to recognize and leverage them for predicting prices of new products. There were certain limitations in this study, with the most significant one being the access to product data. In this study, for the purpose of price prediction, we dealt with a textual sequence containing numerical and expressive characters. Since the model employed is a text-to-text model specifically designed for working with textual sequences, there was a need for access to more commercial data in order to achieve better results.

Literature Review

In this section, we review relevant previous works collected in this field:

Recently, online retail stores, known as E-Shops, have been increasingly popular among consumers. Giant corporations like Amazon, and dominate this market by offering a wide array of products, attributing much of their success to online transactions, doorstep delivery, and competitive pricing. In PostgreSQL tables, data is stored encompassing information on products, manufacturers, categories, and prices [1]. In the realm of e-commerce, pricing stands out as the paramount consideration. Typically, individuals base their product selections predominantly on cost. Data has evolved into a crucial tool across various domains. When strategically employed, data can furnish extensive insights into the intricacies of e-commerce dynamics. Precise predictions regarding e-commerce prices can result in substantial cost savings. The significance of data extends across diverse domains, revealing valuable information about the e-commerce landscape when applied judiciously [2].

Forecasting the costs of electronic gadgets such as smartphones, tablets, LEDs, laptops, scanners, printers, digital cameras, and so forth represents a crucial and captivating challenge. Moreover, the application of machine learning (ML) for predicting product prices has become imperative in contemporary times. Furthermore, consumers can make more informed buying choices, and sellers can optimize their profits by leveraging these methodologies [3]. The primary objective of businesses is to maximize profits. In pursuit of this aim, the continual revision and anticipation of sales prices hold fundamental significance for every company, automating the prediction and adjustment of prices can significantly enhance productivity [4]. Past studies commonly utilized three forecasting models the Vector Error Correction Model, Artificial Neural Network, and Autoregressive Integrated Moving Average. Forecasting extensively employed time series and machine learning techniques [5].

Machine learning techniques offer a more accurate means of predicting software costs compared to traditional estimation methods. Methods such as the K-Nearest Neighbors (KNN) algorithm, Cascade Neural Networks (CNN), and Elman Neural Networks (ENN) have been implemented and demonstrated the capability to forecast the costs involved in constructing or developing software engineering projects. Notably, the K-Nearest Neighbors algorithm has exhibited superior accuracy in predicting the required costs for software engineering projects when compared to Cascade Neural Networks and Elman Neural Networks (ENN) [6]. It is feasible to forecast stock prices with an exceptionally high degree of precision through meticulously crafted predictive models. Moreover, it has been determined that the effectiveness of a predictive model relies on the selection of variables incorporated in constructing the model, the algorithms employed, and the optimization techniques applied to the model The researchers introduced five predictive models for forecasting stock prices - two models based on convolutional neural networks (CNNs) and three models based on long short-term memory (LSTMs) networks [7].

Future research avenues could explore multi-step commodity price forecasting by adjusting the incorporating data for a more extensive dataset [8]. I found some previous research that covers a similar topic to the topic of my paper. Specifically, there was a paper titled "Text2Price: Deep Learning for Price Prediction," where the authors designed a model to predict product price using title, supplier, category, and description information. They used deep learning methods based on RNN and CNN and compared their performance. According to the results, LSTM-based models achieved more accurate predictions [9].

Methodology

The followed methodology includes the following step and as shown in the figure 1.

- **Data Collection:** Collecting information about electronic products and pricing data from the Amazon market involves gathering data about electronic products available on the Amazon website, with a focus on price information and all available features.
- **Data Preprocessing:** This involves cleaning the data and removing unnecessary or duplicate data, as well as identifying relevant features related to product price prediction. Then, forming the text chains where each product's text chain consists of merging features followed by dividing the data into training and testing data sets, which will be used in machine learning algorithms.
- **Model Building:** Defining a T5Regressor class as a subclass to configure a T5 model and regression layer. Configuring the T5Regressor model with a specified leakage rate and using AdamW optimizer with learning rate and epsilon value.

Data Collection

Collecting data is a very tedious and time-consuming task. Web scraping is a technique that aims to address this issue. Web scraping or web data mining is used to extract unstructured data from websites and convert it into structured data and store it in a central local database or spreadsheet for retrieval or analysis in the future, Web scraping is a methodology that allows researchers to extract data from multiple websites and combine them into a single spreadsheet or database, which Simplifies the data analysis process [10].

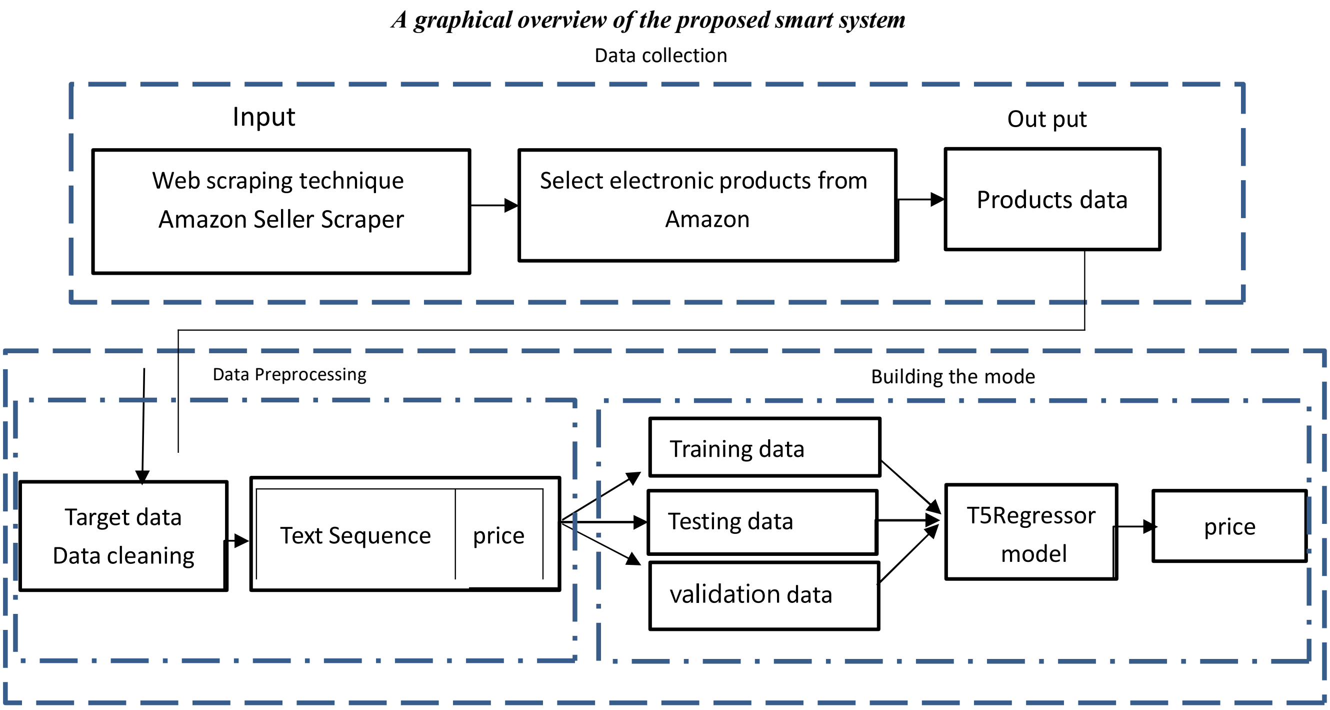

Figure 1: Methodology

Additionally, researchers explored the use of crowdsourcing marketplaces like Amazon Mechanical Turk to enhance Big Data projects, providing guidelines on best practices and demonstrating the strengths and limits of crowdsourcing in social science research [11]. To collect the data, we were systematically used Amazon Seller Scraper tool to gather information about 24,974 electronic products from the Amazon marketplace. This meticulous approach ensured the acquisition of a dataset consisting of 34 features, representing a diverse range of product attributes, The Amazon Seller Scraper tool facilitates the extraction of valuable information such as product titles, descriptions, prices, customer ratings, seller information, and other relevant details.

Data Availability

The CSV data used to support the findings of this study have been deposited in the waheebo/Text2Price repository on GitHub. The data file is named training_testing.csv and can be obtained through the following link: [https://github.com/waheebo/Text2Price.git].

Data Preprocessing

Pre-processing of collected product data involves various techniques and methods to improve the quality and usefulness of the data. Attribute reduction based on rough set theory is a technique used in pre-processing data to reduce the number of attributes or variables in a dataset [12]. This helps eliminate unnecessary attributes and reduce data dependency. Another technique is rank transformation, which can be used to overcome sensitivity to units/scales used to measure/represent data [13]. We worked with the Python programming language and executed it in the Google Colab environment. We utilized techniques such as Pandas and NumPy, which provide powerful functions for loading, cleaning, and analyzing data. I used these libraries to perform data filtering, column transformation, and handling missing values. Additionally, for text analysis, processing, and performing tasks such as text tokenization and word classification, we utilized the NLTK library.

Target Data

The objective is to transform text into corresponding prices. Therefore, we designate the product price as the label for text sequences describing product specifications. This involves crafting text sequences with diverse titles or labels, ensuring each product is associated with a distinct price.

Data Cleaning

Ensuring data quality is essential, which is why a thorough data we cleaned process was conducted. The first step involved removing columns that were deemed irrelevant to the price prediction task. This helped simplify the dataset by eliminating any unnecessary variables that may not contribute to the accuracy of the predictions. Additionally, products lacking price information were excluded from the dataset, as the focus was specifically on products with available pricing data. This streamlined dataset is more suitable for analysis and modeling as it contains the necessary information for accurate price prediction. By removing irrelevant columns and excluding products without price information, the dataset is now better prepared for further analysis and modeling techniques. This process involved systematically removing unnecessary symbols or characters that could potentially introduce noise or inconsistencies in the data. Special attention was given to preserving numerical information as it plays a crucial role in representing product features that impact pricing accurately and other numbers and letters because they represent important parts of the text strin During the data cleaning stage, mentioned techniques were employed to handle missing values, outliers, and inconsistencies.

Missing values were either imputed or excluded based on the specific context and their impact on the analysis. Outliers, which are extreme values deviating significantly from the expected range, were identified and handled separately, depending on their relevance to the analysis. This involved verifying the accuracy of information across columns or different sources to ensure coherence and accuracy. After applying these preprocessing steps, the dataset was narrowed down to a focused set consisting of 20,498 electronic products and 11 features are ['ID', 'price', ' Name ',' brand ',' features_0',' features_1',' features_2',' features_3',' features_4',' features_5',' features_6',' stars '], The table 1 illustrates the features' characteristics of the optimized dataset we selected to represent the inputs for the price prediction model.

features	**Description**
ID	Product index
price	Product price
Name	The product name, which is the text found below the image of each product on an Amazon page, represents a link to open the product specifications page.
brand	The product label/tag that the manufacturer uses to distinguish this product.
features_0 to features_6	Text strings representing the features that distinguish the product from others
stars	Product preference and likability

Table 1: Description Features Table

Text Sequencing

To create comprehensive inputs for the subsequent model, a text sequencing process was utilized. This involved merging features ['Name ',' brand ',' features_0',' features_1',' features_2',' features_3',' features_4',' features_5',' features_6',' stars'] in the one feature to form a single text sequence. By combining these features, the resulting sequenced text aims to provide a rich and comprehensive representation of product information. This unified representation was designed to provide the necessary context and details for the T5-BESD model, which will be used for further analysis and prediction tasks. The text sequencing process allows the model to consider multiple aspects of the product, such as its name, brand, and various features, in a cohesive manner. By merging these elements into a single text sequence, the model can leverage the aggregated information to make more accurate predictions or derive meaningful insights related to pricing or other relevant factors. The resulting sequenced text serves as valuable input for the subsequent model, enabling it to effectively process and analyze the aggregated information and generate insightful outputs, after this stage, we have the final form of the dataset that will serve as inputs for the price prediction models.

It consists of text Sequencing representing features, with corresponding product prices. Final features are ['ID', 'price', ' Name ']. We have included a clear example of a data sample in Table 2, which represents a wireless headphone product. This table displays the data with the Eleventh feature after preprocessing but before generating the text sequences. In Table 3, the data is shown for the wireless headphone product mentioned earlier. However, this table exhibits the data after combining the features and creating a text sequence consisting of eight features, while ensuring that the product price remains as the label.

ID	price	Name	Brand	Features_0	Features_1	Features_2	Features_3	Features_4	Features_5	stars
12	235.32	soni wh1000xm4 wireless premium nois cancel overhead headphon mic phonecal alexa voic control black wh1000xm4	soni	industrylead nois cancel dual nois sensor technolog	next level music edgeai co develop sony music studio tokyo	30hour batteri life quick charg 10 min charg 5 hour playback	wh1000xm4 touch sensor control paus play skip track control volum activ voic assist answer phone call	speaktochat technolog automat reduc volum convers	superior call qualiti precis voic pickup	4.7

Table 2: Before Creating the Text Sequence

ID	price	Text
12	235.32	soni wh1000xm4 wireless premium nois cancel overhead headphon mic phonecal alexa voic control black wh1000xm4 soni industrylead nois cancel dual nois sensor technolog next level music edgeai co develop sony music studio tokyo 30hour batteri life quick charg 10 min charg 5 hour playback wh1000xm4 touch sensor control paus play skip track control volum activ voic assist answer phone call speaktochat technolog automat reduc volum convers superior call qualiti precis voic pickup 4.7

Table 3: After Creating the Text Sequence

Data Splitting

To evaluate the performance of the model and ensure its ability to generalize, the cleaned dataset was randomly divided into three subsets: training, testing, and validation. The dataset was split in such a way that 89.70% of the data, equivalent to 18,393 products, was allocated for training the model. This large training set allows the model to learn patterns and relationships from a diverse range of examples, enhancing its ability to make accurate predictions. Additionally, 10% of the data, equivalent to 2,044 products, was allocated for testing the trained model. This separatetesting set serves as an independent dataset to evaluate the model's performance and assess its ability to generalize well to unseen data.

By evaluating the model on a distinct testing set, it provides insights into the quality of the model's performance on new and unseen instances. Furthermore, a small portion of the data, approximately 0.30% (61 products), was allocated for validation. This validation set is used to fine-tune the model and make any necessary adjustments or improvements based on its performance during training. It helps improve the model's hyper parameters and ensures that it performs well on both the training and testing sets.By randomly splitting the cleaned dataset into these three subsets, it ensures a robust evaluation of the model's generalization capabilities. This approach helps assess the model's performance on unseen data, fine-tune its parameters, and make any necessary adjustments before deploying it in real-world scenarios.

Proposal Model

We have Pretrained the model on a large corpus of text data using a language modeling objective. This helps the model learn general language understanding. Next, finetune the pretrained model on a task-specific dataset by formulating the task as a text-to-text problem. The input to the model is the task description or prompt, and the output is the desired task-specific output. During fine-tuning, the model learns to generate the correct output given the input prompt. Finally, evaluate the finetuned model on a validation set to measure its performance. Iterate this process by adjusting hyper parameters and training for more epochs if necessary to improve the model's performance [14]. Training the T5 BESD model to predict prices involves several steps. First, the model needs to be pre-trained on a large dataset [15]. This pretraining helps improve the model's performance on price prediction tasks. Next, the model can be fine-tuned using a machine learning algorithm, such as linear regression, with data that includes various components and features [16]. This fine-tuning process helps the model learn the relationships between these variables. By modifying the data and increasing accuracy and efficiency, the model can provide a fair price estimation [17].

Data Loading

train_dataloader and test_dataloader we are created using the create_dataloaders function , to convert your training and test data into dataloaders.Figure 2 illustrates the workflow of the our proposal model, starting from the loading of electronic product data, preprocessing data, defining and building the model, then tuning and optimizing the model for predicting products prices.

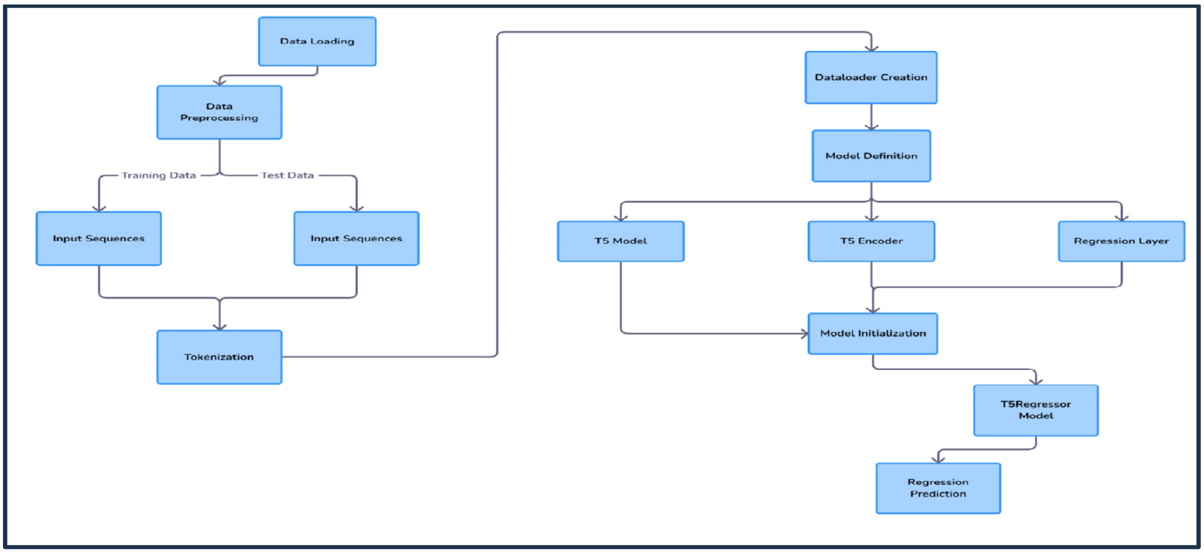

Figure 2: T5 Regressor Model

Model Definition

The neural network's primary task is to convert text strings representing product specifications into the final product price. This is accomplished using two key equations.The first equation is the Forward Pass Equation:

logits = W * last_hidden_state + b.

This equation represents the core computation of the neural network. The input text strings, undergo a linear transformation using a set of learned weights (W) and biases (b). The last_hidden_state is a vector or tensor that encodes the input text information. The matrix multiplication W * last_hidden_state applies the learned weights to the input features, while the addition of the bias term b allows the model to learn more complex relationships in the data. The result, logits, is the raw output of the neural network before applying any activation function.

The second equation is the Linear Regression Layer Equation:

y_pred = W * x + b.

This equation represents the computation performed by the final linear regression layer of the neural network model. The input x is typically the output of the previous layer, which can be the same as the last_hidden_state from the Forward Pass Equation. The weights (W) and biases (b) in this equation are the same as those used in the Forward Pass Equation. The matrix multiplication W * x applies a linear transformation to the input data, similar to the Forward Pass Equation. The addition of the bias term b allows the model to capture any constant offset in the output. The result, y_pred, represents the predicted product prices, which can be used for further processing, such as computing the loss function.

Model Initialization

In the initialization and setup phase, critical components—including the optimizer, a variant of the Adam optimizer called Adam_W, a specialized choice for training neural networks—along with the learning rate scheduler and loss function are initialized. The learning rate scheduler is instrumental in dynamically adjusting the learning rate during training, incorporating a warm-up phase. The total number of training .T5 models, such as Spam-T5, outperform baseline techniques and other large language models (LLMs) in scenarios with limited training samples [18]. For morphologically rich languages like Slovene, T5 models (SloT5) are useful for generative tasks but lag behind in classification tasks compared to monolingual models like SloBERTa [19]. In detecting grammatical errors in Bangla, the T5 model achieves low Levenshtein Distance but requires post-processing for optimal performance [20]. T5 models also demonstrate multilingual skills, performing well in conducting multi-class classification on non-English languages without parameter updates ,In terms of sentence embeddings, T5 models produce superior results compared to BERT-based embeddings and achieve state-of-the-art performance on semantic textual similarity tasks [21].

After defining the core equations that make up the neural network model, the next step is to initialize the model parameters. This is a crucial step that sets the foundation for the model's learning and performance. The initialization of the model parameters, specifically the weights W and biases b, is important because it can significantly impact the model's ability to converge during training and the quality of the final predictions. The pre-trained T5 model is used as the encoder because it has been trained on a large corpus of text data and has learned powerful text representations. By using this pre-trained model as the encoder, the model can leverage these learned representations to extract relevant features from the input text (product specifications) and use them to predict the target variable (product price).

The initialization of the model happens in the __init__ method of the T5Regressor class The T5Config is loaded from the pre-trained 't5-base' model, which provides the necessary configuration for the T5 model. The T5Model is then loaded from the pre-trained 't5-base' checkpoint using the loaded configuration. This initializes the weights of the T5 encoder with the pre-trained values freeze_t5 is set to True, the parameters of the T5 encoder are frozen, meaning they will not be updated during the training process. This can be useful we want to finetune only the regressor head and keep the T5 encoder fixed, The regressor head is defined as a simple nn.Sequential module, consisting of a dropout layer and a linear layer. The input size of the linear layer is set to 768 (the output size of the T5 encoder), and the output size is set to 1 (the target variable, which is the product price).

Training Loop

In the heart of the training process lies the training loop, a critical component iterating over the dataset for a 5 number of epochs. Within each epoch, the model transitions to training mode, and the training data is processed in batches. Key steps in each iteration encompass zeroing gradients to prevent accumulation, making predictions (logits) on input data, computing the loss (using the L1 loss function), L1 Loss Equation:

loss = Σ |y_pred - y_true|

calculating custom accuracy to assess model performance, backward pass for gradient computation, and updating model parameters. Furthermore, the learning rate scheduler dynamically adjusts the learning rate, enhancing training effectiveness. The backward pass and optimization phase involves calculating gradients with respect to the model parameters, a crucial step in optimizing the model. The subsequent optimization step utilizes the Adam_W optimizer to update the model parameters, employing equations that consider gradient moments and a small constant epsilon:
Adam_W Update Step:

$$\mathbf{Param} = \mathbf{param} - \mathbf{lr}\,\frac{\boldsymbol{m\_t}}{\sqrt{\boldsymbol{v-t}}+\boldsymbol{epsilon}}$$

This phase plays a pivotal role in refining the model's parameters, facilitating effective learning during the training process Where:

- Param is the parameter being updated.
- lr is the learning rate.
- m_t is the moving average of the first moment of the parameter.
- v_t is the moving average of the second moment of the parameter.
- Epsilon is a small value added to the square root of v_t to avoid division by zero.

Training Evaluation

During the training evaluation phase, following each epoch, the model transitions to evaluation mode for assessing its performance on a separate test dataset. This involves running the model on the test data and computing various metrics, including mean squared error (MSE), mean absolute error (MAE), and custom accuracy.

Additionally, metrics such as MSE and MAE are calculated, with MSE computed as :
MSE = MEAN((predicted_prices - batch_labels_original)^2) and MAE as :
MAE= MEAN(|predicted_prices - batch_labels_original|).

This comprehensive evaluation process provides insights into the model's generalization capabilities and performance on unseen data.

Performance Results

Supervised machine learning employments calculations for cases to deliver common theories, and after that makeThis table provides a concise overview of the model's performance across the 5 training epochs, including accuracy, mean squared error (MSE), and mean absolute error (MAE).

Epoch	Metric		
	Accuracy	**MSE**	**MAE**
1	38.48%	45057.29	66.87
2	39.40%	31273.60	59.37
3	46.65%	21847.38	53.69
4	54.20%	21264.59	49.55
5	54.86%	19783.88	47.34

Table 4: Results Table

Accuracy

Begins at 38.48% in the first epoch, Shows a fluctuating pattern but generally improves, reaching 54.86% by the fifth epoch. Indicates that the model is learning and adapting to the training data.

Mean Squared Error (MSE)

Starts at a high value of 45057.29 in the first epoch, Exhibits a decreasing trend, reaching 19783.88 by the fifth epoch, Lower MSE values indicate better performance in terms of minimizing prediction errors.

Mean Absolute Error (MAE)

Begins at 66.87in the first epoch, Decreases over epochs, reaching 47.34 in the fifth epoch, Indicates a reduction in the average absolute differences between predicted and actual values.

Conclusion

This study explores the intricate link between descriptive text and electronic product prices, unveiling the Text2Price model with the T5-BESD architecture. Results confirm the efficacy of deep learning in predicting e-commerce prices, showcasing adaptability and learning with a boosted accuracy of 54.86%. The model excels in predicting unseen data with 52.38% accuracy, demonstrated by decreasing trends in Mean Squared Error (MSE) and Mean Absolute Error (MAE) to 19783.88 and 47.34, respectively, highlighting its proficiency in minimizing prediction errors. Acknowledging limitations from limited product data access, the study advocates for a broader dataset to enhance accuracy. Contextualized in the e-commerce landscape, the literature review emphasizes pricing dynamics' strategic importance, placing Text2Price in data-driven decision-making and competitive pricing, drawing insights from industry leaders like Amazon. In summary, this research advances predictive modeling in e-commerce, showcasing the transformative potential of deep learning. The Text2Price model, innovatively predicting text-to-price, signifies progress in decision-making. In the evolving e-commerce space, the study explores integrating technology for optimized pricing, emphasizing the Text2Price model's role in empowering stakeholders with data-driven insights in the dynamic world of electronic commerce.

References

1. Salvatore Carta, Andrea Medda, Alessio Pili , Diego Reforgiato Recupero * and Roberto Saia(2018)"Forecasting E-Commerce Products Prices by Combining an Autoregressive Integrated Moving Average (ARIMA) Model and Google Trends Data.
2. DUYKU, E., GUZEL, M., BOSTANCI, E., & ASKERZADE, I. (2022). A MACHINE LEARNING BASED APPROACH FOR PRICE ESTIMATION. Proceedings of IAM, 11(1), 50-61.
3. Hasnain, M., Sajid, A., Ayeb, A., & Awan, A. (2023). Predicting the price of used electronic devices using machine learning techniques. International Journal of Computing and Related Technologies, 4(1), 13-19.
4. Menculini, L., Marini, A., Proietti, M., Garinei, A., Bozza, A., Moretti, C., & Marconi, M. (2021). Comparing prophet and deep learning to ARIMA in forecasting wholesale food prices. Forecasting, 3(3), 644-662.
5. Mingxue Ma, Vivian W. Y. Tam, Khoa N. Le & Robert Osei-Kyei(2023)" A systematic literature review on price forecasting models in construction industry".
6. Abdulmajeed, A. A., Al-Jawaherry, M. A., & Tawfeeq, T. M. (2021, May). Predict the required cost to develop Software Engineering projects by Using Machine Learning. In Journal of Physics: Conference Series (Vol. 1897, No. 1, p. 012029). IOP Publishing.
7. Jaydip Sen , Sidra Mehtab , Gourab Nath "Stock Price Prediction Using Deep Learning Models" Deep Learning Developers' Conference (DLDC'2), October 29 - 30, 2020 At: Bangalore. INDIA, Volume: Vol 1, Issue 3, Pages: 34 – 40
8. Realtek Ulf Geir Indahl ,Master's Thesis Norwegian university of life sciences(2021)"Non-ferrous metal price forecasting with Recurrent Neural Networks – How do they perform when forecasting multiple timesteps ahead?" 30ECTS

9. Saraçlar, B., Kuyumcu, B., & Delil, S. (2022). Text2Price: Deep Learning for Price Prediction. Artificial Intelligence Theory and Applications, 2(2), 28-38.
10. SCM de S Sirisuriya (2023) " Importance of Web Scraping as a Data Source for Machine Learning Algorithms – Review" IEEE 17th International Conference on Industrial and Information Systems (ICIIS).
11. Porter, N. D., Verdery, A. M., & Gaddis, S. M. (2020). Enhancing big data in the social sciences with crowdsourcing: Data augmentation practices, techniques, and opportunities. PloS one, 15(6), e0233154.
12. Lin, S. H., Kuijpers, O., Peterhansl, S., & Pollmann, F. (2023). Distributive pre-training of generative modeling using matrix-product states. arXiv preprint arXiv:2306.14787.
13. Nurrahman, F., Wijayanto, H., Wigena, A. H., & Nurjanah, N. (2023). Pre-processing data on multiclass classification of anemia and iron deficiency with the XGBOOST method. Barekeng: Jurnal Ilmu Matematika dan Terapan, 17(2), 0767-0774.
14. Singh, A. P., & Singh, G. (2019). Analysis of Amazon product reviews using big data-apache pig tool. Int. J. Inf. Eng. Electron. Bus, 1, 11-18.
15. Yang, P. J., Chen, Y. T., Chen, Y., & Cer, D. (2021). Nt5?! training t5 to perform numerical reasoning. arXiv preprint arXiv:2104.07307.
16. Carmo, D., Piau, M., Campiotti, I., Nogueira, R., & Lotufo, R. (2020). Ptt5: Pretraining and validating the t5 model on brazilian portuguese data. arXiv preprint arXiv:2008.09144.
17. Labonne, M., & Moran, S. (2023). Spam-t5: Benchmarking large language models for few-shot email spam detection. arXiv preprint arXiv:2304.01238.
18. Ulčar, M., & Robnik-Šikonja, M. (2023). Sequence-to-sequence pretraining for a less-resourced Slovenian language. Frontiers in Artificial Intelligence, 6, 932519.
19. H.A.Z. Sameen Shahgir , Khondker Salman Sayeed.(2023)," Bangla Grammatical Error Detection Using T5 Transformer Model",arXiv:2303.10612v1 [cs.CL] 19 Mar 2023.
20. Genta Indra Winata, Andrea Madotto , Zhaojiang Lin, Rosanne Liu , Jason Yosinski , Pascale Fung, (2021)."Language Models are Few-shot Multilingual Learners",arXiv:2109.07684v1 [cs.CL] 16 Sep 2021.
21. Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, Yinfei Yang,(2021)," Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models".