DYNAMIC PRICING ANALYSIS FOR INDUSTRY4.0 USING NEURAL NETWORK

[1]Chetana Tukkoji, [2]Sandhya Guruswamy, [3]Spoorthi T, [4]Boosi Shyamala, [5]Archana S Nadhan, [6]Sandhya Rani Nallola, [7]Saritha A. K

[1,3,4,5,7] Department of Computer Science & Engineering, GITAM School of Technology- Bengaluru
[2] School of Computing & Information Technology, Reva University – Bengaluru
[6] Research Scholar, Department of Computer Science & Engineering, GITAM School of Technology- Bengaluru

ABSTRACT

The Work We Do is Feature Engineering with Dynamic Pricing. Over the past five years, online shopping has grown significantly, increasing demand for and competition among superior products. To ensure greater profit, it is important to understand the appropriate price for a product. Continuous study and price forecasting based on the variables that influence a product's price can be quite helpful in this regard. Over the past decade, improvements in machine learning and natural language analysis have introduced a wealth of new prospects and methods for price prediction. The most effective, modern categorization technique under the umbrella of machine learning is called feature engineering. We have derived the desirable qualities from a humungous data set and used applied DNN (Deep Neural Network) on them. An extensive and detailed literature survey was conducted which helped in accumulating the knowledge of recent technologies, tools, and algorithms in the field of Dynamic Pricing, Feature Engineering, and Natural Language Processing, this made the process of detailing the project plan and construction of end-to-end algorithm design easier.

Keyword: *Dynamic Pricing, Feature Engineering, DNN, Machine Learning, Natural Language.*

1. INTRODUCTION

According to IBEF, the Indian E-commerce sector would grow from $46.2 billion in 2020 to $111.40billion in 2025. With a growing number of people preferring to shop online rather than in traditional stores, it's becoming a rapidly growing industry of business to keep in mind as you plan your career. Ecommerce,or"electroniccommerce,"refersto theelectronicpurchaseandsellingofgoodsands ervicesthroughtheinternet.

On national and world wide level, e-commerce is essential in the market place. Pricing products sold on e-commerce sites is difficult as they have a large and wide range of products and pricing should be done

Accordingly and should be done in such a way that all persons involved in the chain are benefitted. In industry, the purpose of dynamic pricing is to discover the highest price that customers are willing to pay. Dynamic pricing is also known as demand pricing in the business. Sort of price discrimination used to try to maximize revenue depending on the willingness to pay of different market.

Dynamic pricing allows you to examine each customer's demand curve. Because the demand curve can more accurately represent the minimum and maximum price clients are prepared to pay for various products the procedure becomes easier. Machine learning and deep learning are being used widely in this field as we have plenty of data to build a good model and new algorithms are being used to continuously change (update) prices based on demand, supply, ratings, etc.

In this project, we mainly used Feature engineering is the process of creating new features. Featureengineeringisthepracticeofextractingf eaturesfromadomainutilizingdomainknowled ge.Andcharacteristics from raw data, which is we have extracted the features that influence the price of a product. DNN (Deep Neural Networks) are Artificial Neural Networks with multiple layers which include the Embedding layer, Flattening layer, GRU layer, dense layer, and dropout layer between the input and output layer.

They all perform their own actions which are elaborated on in the below sections. All the extracted features from feature engineering are further used for training a Deep Neural Network. After training our data set on these Neural Networks we obtained predicted prices of the cross-validation set and calculated Root mean squared logarithmic error and it turned out to be very less which indicates our model has performed well enough. The model generated has been used to predict the prices of the testing set and the results are indicated below.

2. LITERATUREREVIEW

a. Motivation

Dynamicpricinghasbeenapopularpricediscriminationtacticusedby E-Commercevendorstomaximize profits by charging varying rates for very similar or essentially the same items or services based on the amount of money the client is ready to spend. While the retailers consistently try to maximize theirprofitsandgainmarketsharebyusingmultiplestrategieslikepromotions, discounts, incentives, complementary products, providing after-sales services, buyers in the present day are well informed and have ease of access in terms of comparing prices and features of a product across multiple platforms with just a few clicks. This grants them a better bargaining position and more selectiveness while deciding upon the final product to purchase.

Leading to enhanced availability of consumer demand data and technology developments that benefitted both buyers and sellers, e-commerce marketplaces have been developing at an unparalleled rate. As the Ecommerce industry grows more competitive, merchants must be able to readily monitor their competitors and the market as a whole and modify pricing to make the same items enticing to customers while preserving profit margins. As a consequence, dynamic pricing is being employed more frequently, which helps to develop better customer strategies and, in turn, aids the firm's success.

b. Research Gap

In practice, describing the need ahead of time is typically challenging. A new dynamic pricing strategy based on PSO-trained neural networks was provided in a research study that explored the particle swarm optimization (PSO) algorithm for training neural networks, then brought the PSO-trained neural network into e-commerce. It can determine the best dynamic price and be utilized in e-commerce with several quantities and demanding levels at the very same time. (Liang Peng &Haiyun Liu, 2007)

Time-series models have long played a key role in marketing since so much data is in the case of the time period, such as item revenues for pricing dynamics. Multiple time-series analyses of competitive marketing behavior explored the important importance of competition in marketing and offered vector auto-regressive methods (VARMA) as a valuable tool.

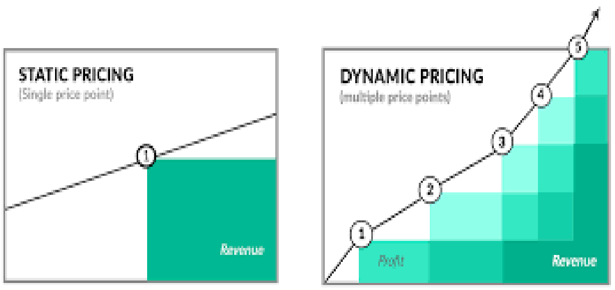


Figure 1. Static Vs Dynamic Pricing

A machine Learning Cognitive Approach for Predicting Buy by Online Clients based on Dynamic Pricing suggests combining three methods: identifying client categories, determining suitable pricing for them, and predicting their likelihood of buying within that budget range. (Faehnle &Guidolin, 2021).

The modern advances in Computation and Reinforcement Learning enabled many sophisticated approaches to be used for dynamic problems. In Kephart, J.O., Hanson, J.E., Greenwald, A.R.: Dynamic pricing by software agents, a price bot is developed using Q-learning in order to properly adjust the price in response to changes in the market state. (Kephart J, Hanson J, & GreenwaldA, 2018)Ma, S.; Fildes, R.; Huang, T. made another major contribution, this time adder.

Addressing the issue of high dimensionality in retail. Demand forecasting employing high-dimensional data, with a focus on SKU retail sales forecasting using intra-and inter-category promotional data. The authors of this research used a multistage Lasso regression to construct a four-step frame work for dealing with high complexity (Shaohui,Robert, &Tao, 2016)

3. PROBLEMSTATEMENT

Pricing for an online product is important because it affects many aspects of the e-commerce business namely marketing, finance, and marketing. A product with the wrong price can lead to the death of your company while a product with the right price may position you as the main competitor in your area.

As millions of products, discounts, and vendors will be added, flexible prices should be made by considering various factors. A good flexible pricing strategy allows you to make prices faster and on average, all while understanding the consequences of your changes. But it is not an easy task to calculate all the products properly and to do it yourself is an impossible task. So there is a need to create an algorithm for pricing these products.

As we have vast amounts of data and many features to be considered, machine learning and deep learning come into play. While pricing us need an algorithm that will optimize prices by taking all factors into consideration and deep learning does that.

The Figure 1 Static Vs Dynamic Pricing tells that Dynamic Pricing is when the price of a product depends on demand while supply is constant in static pricing the price of the product does not depend on demand.

a. Objectives

i. It may be used to increase sales:
Dynamic pricing is frequently seen as a tool for businesses to raise prices. This may be true to some extent, however, this practice may also be used to lower prices.

ii. It may be used to increase profits:
If rivals are providing things or services at a much lower price, dynamic pricing can be utilized to increase profits. If you know what your customers want ahead of time, you may adjust the pricing of items depending entirely on their purchase styles.

iii. It can increase demand:
Because open space equals no money, dynamic pricing is frequently employed in events. If the room is available on the day of the event, some customers may be offered a discounted fee. This helps you to optimize your earnings while also giving you access to all of the money you may have at the moment.
This procedure may be found in the service sector, transportation choices, and other businesses with varying demand levels.

iv. It allows for demand-driven pricing:
Imagine making a fortune farming tomatoes. During the summer months, when there is abundant of sunlight and rainfall, it is much simpler to grow. This implies that producing crops is less expensive and customers save

money. Tomatoes are cultivated in cages in greenhouses during the winter. You must pay for additional light and water, in addition to the greenhouse's cost. To represent the additional labor necessary to bring tomatoes to market, we employ dynamic pricing.

v. It gives you a better understanding of client behavior: Dynamicpricingmakescalculatingthedemand curveforeachclientmucheasier.Thesegraphsd epictthelowestand highest prices that a customer is willing to pay.

4. MODEL OF THE PROPOSE SYSTEM

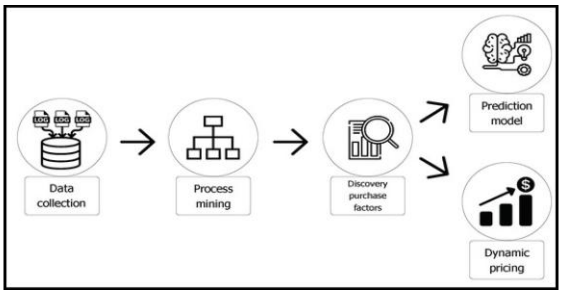


Figure 2. Dynamic Pricing Model

The above Figure2. The dynamic Pricing Model indicates the process of making the project.

i. Data Collection: collecting the required dataset and useful columns.
ii. Process Mining: After being corrected for incorrect data and missing values, the data is evaluated using a process mining algorithm to find platform user models and patterns.

iii. Discovery Purchase Factors: The events are examined in order to determine the most importantaspectsthatimpactauser'spurchasing decisions.Thepurposeofthisstageistodetermin etherelationship between the information available when purchasing a product and the product's sale.

iv.PredictionModel: Machinelearningalgorit hmsaretrainedontheprimaryaspectsthatimpact auser'spurchasing decisions in order to forecast whether or not such a user will buy the product.

v. Dynamic Pricing: The found characteristics are utilized to create dynamic pricing strategies with the goal of boosting the number of items purchased and an e-commerce company's overall income.

4. SYSTEMARCHITECTURE

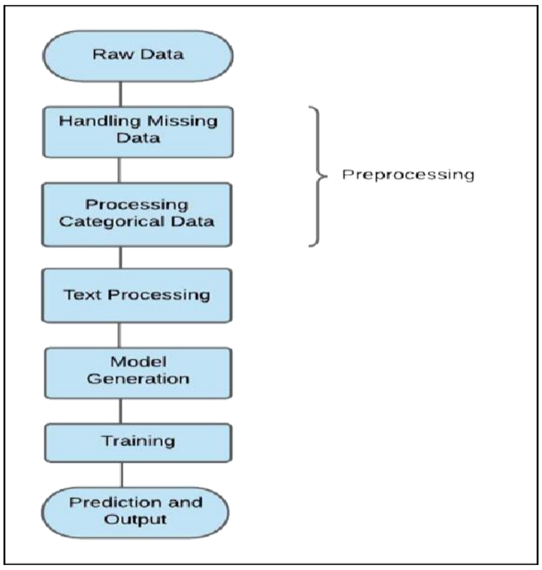


Figure 3: System Architecture

The above Figure 3 system architecture of our project tells step by step process of the implementation. The data used here comes from a Japanese e-commerce company called Mercari. This is an open database sent to Kaggle and separated by train.tsv and test.tsv files. There are 693,358 rows and 7 columns in thetest.tsv file and 1,482,534 rows and 8 columns in the train.tsvfile. Files contain the following listing of product features.

- train_id or test_id - the id of the listing the products
- name - the title of the products

- item_condition_id-the condition of the items provided by the seller.
- category_name - category of the listing the products
- brand_name- brand of the product.
- Price-the price that the item was sold for. This is the target variable that we predicted. The unit is USD($).This column doesn't exist intest.tsv.
- Shipping-1 if the shipping fee is paid by the seller and 0if the shipping fee is paid by the buyer.tem_description - the full description of the item.

The architecture has following steps:

i. Checked for null and missing values, and filled some of these while some were removed usingfillna() and dropna() respectively.
ii. Categorical data were converted into numeric values using LabelEncoder().
iii. TextPreprocessingisdoneusingTokenizer() to clean the data and converting into a model readable format.
iv. Feature Scaling is performed using MinMaxScaler() which helps to normalize the independent features of the data within a particular range.

5. METHODOLOGY USED

5.2.1. Deep Neural Network

Figure 4 illustrates my point. A deep neural network is an artificial neural network (ANN) with several layers between the input layer, hidden layer, and output layer (DNN). Neurons, synapses, weights, and functions are all basic components of neural networks, which come in a range of forms and sizes.

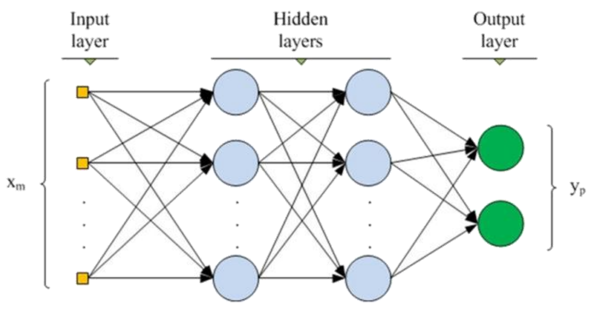


Figure 4: DNN Architecture

The Neural Network we used contains 5 embedding layers, 3 flatten layers, 2 GRUs, 3 dense, and 2dropout layers.

Embedding Layer:
The embedding layer of a neural network is the initial hidden layer that allows us to encode each word into a fixed-length vector. Instead of only 0s and 1s, the result is a dense vector with actual values. Because word vectors have a fixed length, we can easily express words while minimizing their dimensionality.

Flatten layer:
Flattening is the process of converting data into a one dimensional array for usage in the subsequent layer, as seen in Figure 5. We flattening the outputs of the convolutional layers to create a single long feature vector. It's also connected to the final categorization model, known as a fully-connected layer.

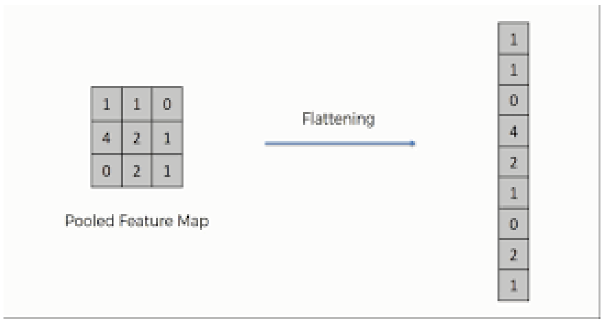


Figure 5: Flatten Layer

GRU Layer:
GRUs (Gated Recurrent Units) is a more advanced version of recurrent neural networks. To tackle the vanishing gradient problem of a normal RNN, GRU uses the update and reset gates. In essence, there are two vectors that specify what data should be transmitted to the output and how much of the previous memory should be retained. They are remarkable in that they can be trained to remember information from a long time ago without it being washed away with time or information that is unimportant to prediction being removed.

Dense Layer:
Adenselayeristhemostbasiclayerinanyneuralnetworkthatisdeeplyconnectedtoitsprecedinglayer, meaning the layer's neurons are connected to every neuron of the preceding layer and from below Figure6 each neuron receives input from all of the preceding Layer's neurons, thus the term dense and fully connected layer.

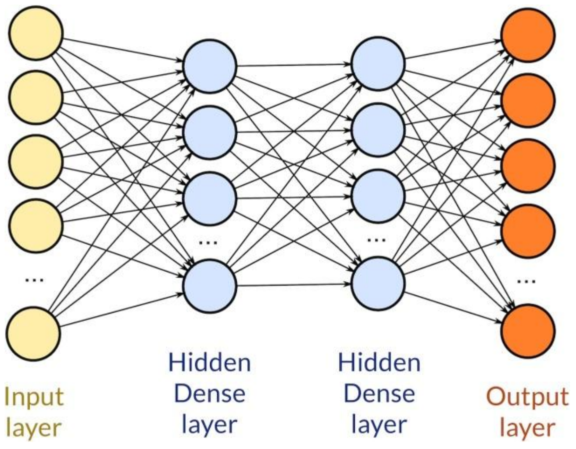


Figure 6: Dense Layer

6. IMPLEMENTATION

a. Data Handling

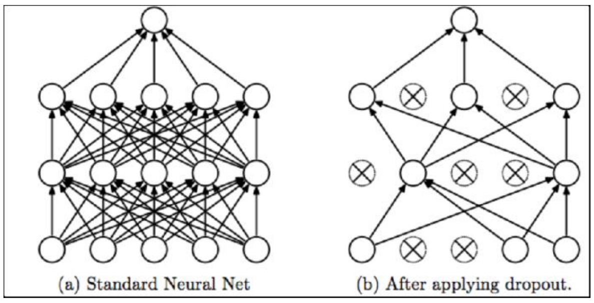
Installthelibrariesi.e.,numpy,matplolib,keras,sckitlearn,pandasfrompythonusingpipcommandandimport those libraries in jupyter notebook and also define a function for RMSLE to find the error value after training the model
Now read the datasets train and test from our system which we have downloaded from kaggle.

b. Preprocessing

Raw data is usually incomplete and has inconsistent formatting. Data preprocessing is the process of converting raw data into a comprehensive format for using Machine Learning methods, involving the manipulation or reduction of data before it is used to ensure or improve performance.

c. Handling Missing Data

The dataset we found has many missing



values and is incomplete.

Figure 7: GRU Layer

So we have used the pandas library to eliminate or fill the missing values and make the data consistent. The function fillna() has been used to fill the blanks in columns category_name, brand_name, and item_description with the word "missing".

d. Handling Categorical Variables

Category_name and brand_name are categorical variables, so each category can be assigned a number in order to make the classification easier. We accomplished this using the function LabelEncoder().

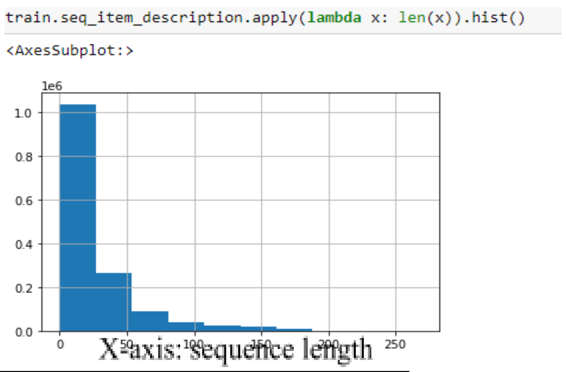
e. Text Processing

Text processing is the process of automating the analysis of electronic text that uses Natural Language Processing. This makes machine learning models suitable to obtain structured information about the text, which they can apply for text analysis, transformation, or generation. Here, we have used the functionTokenizer()fromkeraslibrarytoconvertsentencesintolowercase,splitthewordsinit,andnumberthemto form text sequences to make it easier for analysis.

f. Scaling

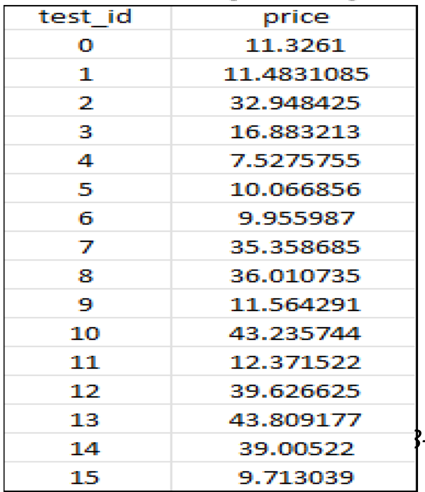
Feature scaling is a method employed for standardizing a range of independent variables or data features. It is also called data normalization and is a technique for normalizing the independent characteristics present in the data in a set range. It is usually performed during the data preprocessing step. Feature scaling is a way of reducing the values of all of our dataset's independent features to the same range.
Here we have used the MinMaxScaler()functionfromtheScikitlearnlibrarytoreshapethedatasetsothatit contains 11 columns. The final shapes of training and validation setsare(1467709,11)and(14826,11) respectively.

g. Padding Sequences



```
train.seq_item_description.apply(lambda x: len(x)).hist()
<AxesSubplot:>
```

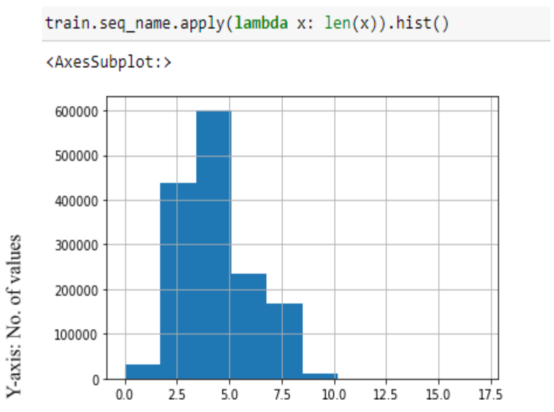
X-axis: sequence length



test_id	price
0	11.3261
1	11.4831085
2	32.948425
3	16.883213
4	7.5275755
5	10.066856
6	9.955987
7	35.358685
8	36.010735
9	11.564291
10	43.235744
11	12.371522
12	39.626625
13	43.809177
14	39.00522
15	9.713039

We must choose the sequence length column for each of our sequences. The model training will take an eternity if it is too long. We run the danger of truncating crucial information if it's too short. Before we make this decision, we should visualize the sequence length distribution.

The below code will plot the sequence length graphical distribution for the seq_name column in a histogram representation.



```
train.seq_name.apply(lambda x: len(x)).hist()
<AxesSubplot:>
```

X-axis: sequence length

The below code describes the sequence length graphical distribution for the seq_item_description column in a histogram representation.
We use the Keras sequence processing pad sequences() function to pad sequences to the same length for each column in the code below

TRAININGAMODEL

This will be a multi-input model for those inputs

- name: sequences created from texts
- item_desc: sequences created from texts.
- brand: Integerization of texts
- category: Integerization of texts
- category_name: sequences created from texts
- item_condition: integers
- shipping: integers with 1 or 0

$$RMSLE=\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\log\left(y_i+1\right)-\left(\widehat{y+1}\right)\right)^2}$$

Except for "shipping," all input will be sent to the embedding layer.

We'll need to provide you with Embedding layers for the following inputs. Whole integers (representing particular words) are converted into dense vectors by embedding layers. It takes integers as input, searches them up in an internal dictionary, and returns the vectors that go with them. Using a dictionary to look anything up is a good idea.

Embedded sequences, like other forms of repeating networks, will feed the GRU layer, which is useful for learning patterns in data sequences.

Data from sequential embedded layers will be flattened to two dimensions each Flatten layer. The layers will then be joined to a huge two-sided tensor, which will include "transmission." Following a few dense layers, the last layer thicker take as "linear" retreat to make up to the wrong values, such as specifying none for the activation parameter.

In the end, the mean absolute error on the cross-validation set is 0.0942

TESTINGA MODEL

We trained the model using parameters batch_size=20000andepochs=5.

RMSLE (Root Mean Squared Logarithmic Error)

As demonstrated in Figure 9, it's the Root Mean Squared Error of the log-transformed anticipated and log-transformed actual values. RMSLE adds 1 to both actual and projected values before taking the natural logarithm to avoid computing the natural log of likely 0 (zero) values. As a result, the function can be used if the actual or projected data contains zero-valued elements.

i. When dealing with goals that develop exponentially, such as population counts, average product sales over time, and so on.

ii. We don't want to penalize significant differences when both the predicted and actual values are large

.

iii. RMSLE is favored since we are concerned with percentage errors rather than absolute quantities of errors.

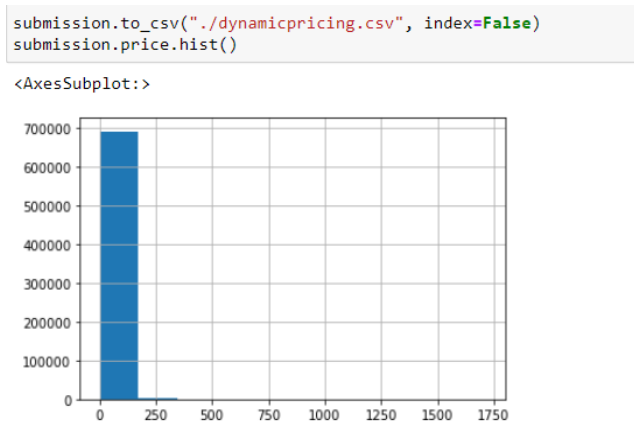
We have obtained an RMSLE (RootMeanSquaredLogarithmic Error) of 0.4895. As more data is collected and the number of features is increased, the supervised learning model will produce much more accurate results.

7. RESULTS ANALYSIS

a. Predicting Values

Inthisblockofcodethemodelpredictedvalueswill bestoredintovariablepreds.Thenfrompreds, test_id and price of the products are truncated and added to the submission variable.

Here we are converting the submission variable values into a CSV file called dynamic pricing.csv and representing a histogram view for it

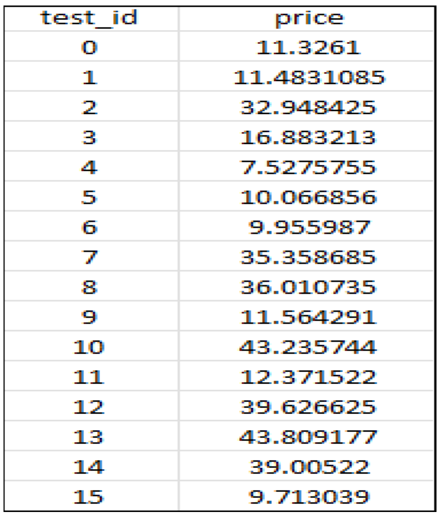


```
submission.to_csv("./dynamicpricing.csv", index=False)
submission.price.hist()
```

```
<AxesSubplot:>
```

a. Output

Now the data stored in submission is our final output. It will be converted to a csv file. The output file consists of the continuous predicted prices for the corresponding test_id. From Figure 10, we can see the final predicted dynamic prices of each product.



test_id	price
0	11.3261
1	11.4831085
2	32.948425
3	16.883213
4	7.5275755
5	10.066856
6	9.955987
7	35.358685
8	36.010735
9	11.564291
10	43.235744
11	12.371522
12	39.626625
13	43.809177
14	39.00522
15	9.713039

Figure 10: Final Output

8. CONCLUSION AND FUTURE WORK

Dynamic pricing in response to market price changes is becoming a hallmark of e-commerce.Determiningthesellingpriceofap roductisadifficulttaskforsellerstokeeponthe market.Theroleofdynamic pricing is to determine the selling price so that the seller can generate more revenue.

In this paper, we proposed a neural network framework for dynamic pricing on e-commerce platforms. This framework has been trained and tested on a large dataset from the Japanese e-commerce company Mercari. Common frameworks can be used in different industries in online mode and can be adapted to specific applications. The proposed framework was developed using machine learning, data mining, and deep learning techniques to predict online customer buying behavior by choosing the right price based on dynamic pricing increase. Our approach uses neural networks to automatically adjust the selling price of a product to increase sales.

FUTUREWORK

The performance of this architecture can be further improved by increasing the injection rate, reducing packet size, adding packet normalization using LSTMs, bidirectional RNNs, stacked RNNs, and implementing density layers or more RNN outputs.

Pricing structures use only product-specific features to describe the state of the environment. When pricing in a more customizable scenario, you can consider more kinds of features, such as promotions or membership pricing or pricing as an add-on significant research concerns will be looked into in the future. For example, our prediction model may be expanded to include external elements that undoubtedly impact consumers' purchase decisions, such as competitive strategies and prices. Furthermore, we can apply our methodology to other sales domains by illustrating how sophisticated forecasting techniques combined with dynamic pricing strategies may boost ticket sales and income.

REFERENCES

1. Yongli Wang, "Dynamic Pricing Considering Strategic Customers", International Conference onLogistics Informatics and Service Sciences (LISS), pp. 1-5, 2016.
2. Rajan Gupta and Chaitanya Pathak / Procedia Computer Science 36 (2014) 599 – 605
3. Garbarino, E., & Lee, O. F. (2003). Dynamic pricing in internet retail: effects on consumer trust.Psychology & Marketing, 20(6), pp. 495-513.
4. Dimitris Bertsimas and Georgia Perakis. Dynamic pricing: A learning approach. In Mathematical andcomputationalmodels for congestion charging,pp. 45–79. Springer, 2006
5. Rajan Gupta and Chaitanya Pathak / Procedia Computer Science 36 (2014) 599 – 605
6. Garbarino, E., & Lee, O. F. (2003). Dynamic pricing in internet retail: effects on consumer trust.Psychology & Marketing, 20(6), pp. 495-513.
7. Dimitris Bertsimas and Georgia Perakis. Dynamic pricing: A learning approach.

In Mathematical andcomputationalmodels for congestion charging,pp. 45–79. Springer, 2006

8. AfecheandB.Ata.Bayesiandynamicpricinginqueueingsystemswithunknowndelaycostcharacter-istics.Manufacturing & Service Operations Management, 15(2):292–304,2013
9. AdidaandG.Perakis.Dynamicpricingandinventorycontrol:uncertaintyandcompetition.Operations Research, 58(2):289–302, 2010b
10. Aoki.Onsomepriceadjustmentschemes.AnnalsofEconomicandSocialMeasuremen, 3(1):95–115,1974.