Product Pricing Suggestions

Venkata Anil Kumar Thota

012525734

San Jose state university

**Contents:**

[Abstract 3](#_Toc8587405)

[Introduction 4](#_Toc8587406)

[Purpose 5](#_Toc8587407)

[Literature review 6](#_Toc8587408)

[Research Methods 13](#_Toc8587409)

[Anticipated Results 16](#_Toc8587410)

[Discussions 17](#_Toc8587411)

[Recommendations 18](#_Toc8587412)

[References 19](#_Toc8587413)

**Abstract**

This research suggests the price of the product for online selling company, Mercari. Using this type of predictive analysis will be easier for the seller to sell their products. Exploratory data analysis is performed on the dataset and some managerial, organizational and strategic insights are drawn. This study compares the traditional machine learning algorithms (Multiple Linear regression, XGBoost) with Deep neural networks and implements these algorithms on Mercari dataset. RMSLE (Root mean squared logarithmic error) is used as an evaluation metrics in this research. Python and Jupyter notebook are used to perform the Exploratory data analysis and data modeling.

**Introduction**

Pricing the products is the main keystone determination because it influences every point of the business. The price that is set by the seller will affect the profit margin. In general, higher prices gives higher profit, but it also leads to lower sales which will completely wipeout the profits.

Due to the ever-changing technology E-commerce has become one of the dynamically evolving industry. Now-a-days most of the companies which are in the field of E-commerce are exploring the ways to offer suggested prices to people selling items through their website. Companies like Flipkart, Snapdeal, amazon is capable of building algorithms for their customers. This type of guidelines would help the seller to estimate their product price rather than searching for the prices of similar products online.

**Tools and skills:**

Programming language used in this paper to perform EDA and fit the Machine learning model on the data is Python. Python has great data handling capacity. It is open source and can interact with almost all the third-party programming languages and platforms. Jupyter Notebook an open sourced web application is used to write the python code. Instead of writing whole program we can write few lines of code and run them one at a time.

Tableau, one of the most powerful visualization tools is used for Exploratory data analysis. It is very fast and user-friendly tool for real time data analysis.

**Terminology:**

Corpus: collection of documents

Term Frequency: frequency of a term in s document

Inverse Document Frequency (IDF): Distribution of term over a corpus

Overfitting: When a model learns noise in the training data, which will negatively impact the performance of the model on test data.

Regularization: It shrinks the coefficients estimates towards zero.

**Purpose**

Online selling applications are facing problems while suggesting the price to the sellers because pricing depends on various factors. My aim is to address this problem by automatically suggesting the price to the products based on the condition, descriptions, brands of the product. This can be achieved by developing an algorithm which takes these features as an input and gives the price of the product as output. Business analytics brings different kinds of models to fit this addressed problem.

Decision makers think that the dynamic prediction using the words as input variables. But with the evaluation of the analytical techniques handling text analysis in prediction of prices became easy.

There are many online research papers and articles in the field of dynamic price prediction which are taken as the primary resource in this paper. Kaggle competitions for price predictions provided the dataset with metadata. Many Articles in online gave an idea of what evaluation metrics to be used to find the performance of the developed model.

**Literature review**

**Dataset:**

Source of the dataset is from Kaggle which is sponsored by Mercari. Mercari is an E-commerce company where users can buy or sell their products. It is operating in Japan and united states.

It contains two csv files.

1)Train data and

2)Test data.

There are 29999 records in train and test dataset. Metadata for the dataset is explained in the following table.

|  |  |  |
| --- | --- | --- |
| Feature | Metadata | Type |
| train\_id | Id of the product/listing in train data | Numeric |
| name | The Name of the product/listing | Categorical |
| category\_name | Category of the product/listing | Categorical |
| brand\_name | The brand of the product/listing | Categorical |
| Item\_condition\_id | The condition of the items given by the seller | Numeric |
| shipping | It shows whether the shipping is paid by seller or buyer | Categorical |
| item\_description | The description of the product/listing | Categorical |
| price | The price that the item is sold for. Values are mentioned in Dollars. This column doesn’t exist in test data since this column is the target variable. | Numeric |

**Data Analysis:**

Before fitting the model on the train set we need make several hypotheses by exploring the patterns in the data. Below are the results for the analysis performed on the dataset.

**Variation in prices:**

The first question that was in my mind when I started Exploring data is the range of prices in training data. Descriptive statistics of the price column are explained below:

The minimum price is $0

The maximum price is $1506

The median price in the dataset is $17

The 75 percentile of the price is $29.

**Condition of the products:**

There is column in the dataset named “item\_Condition\_id”. This column contains the values from 1 to 5. Item with condition id 1 represents that the product condition is very good and the item with condition id represents that the product condition is very bad. The statistics of each condition id is given in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| Item condition id | Frequency | Median Price ($) | Mean Price ($) |
| 1 | 13040 | 17.0 | 26.5 |
| 2 | 7511 | 17.0 | 27.5 |
| 3 | 8727 | 16.0 | 27.2 |
| 4 | 665 | 15.0 | 22.9 |
| 5 | 56 | 19.5 | 25.7 |

From the analysis we can see that items with condition id 1 and condition id 2 have 50% of their prices below $17.But their mean price is around $27. The frequency of items with condition id 4 and 5 are very less when compared with others.

**Shipping condition of the Products:**

We need to check whether the price is impacted by the shipping condition. Shipping attribute contains two values: 0 and 1.

If the shipping fee is paid by seller then the shipping is denoted by 1 and it is represented by 0, if the shipping fee is paid by buyer. In the given dataset there are 55% of “0” and 45% of “1”. The average prices of group1 is 22.5 and group 0 is 30.5. So, we can conclude that the items for which the shipping fee is paid by the seller has lower average price when compared with the other group.

**Expensive Brands:**

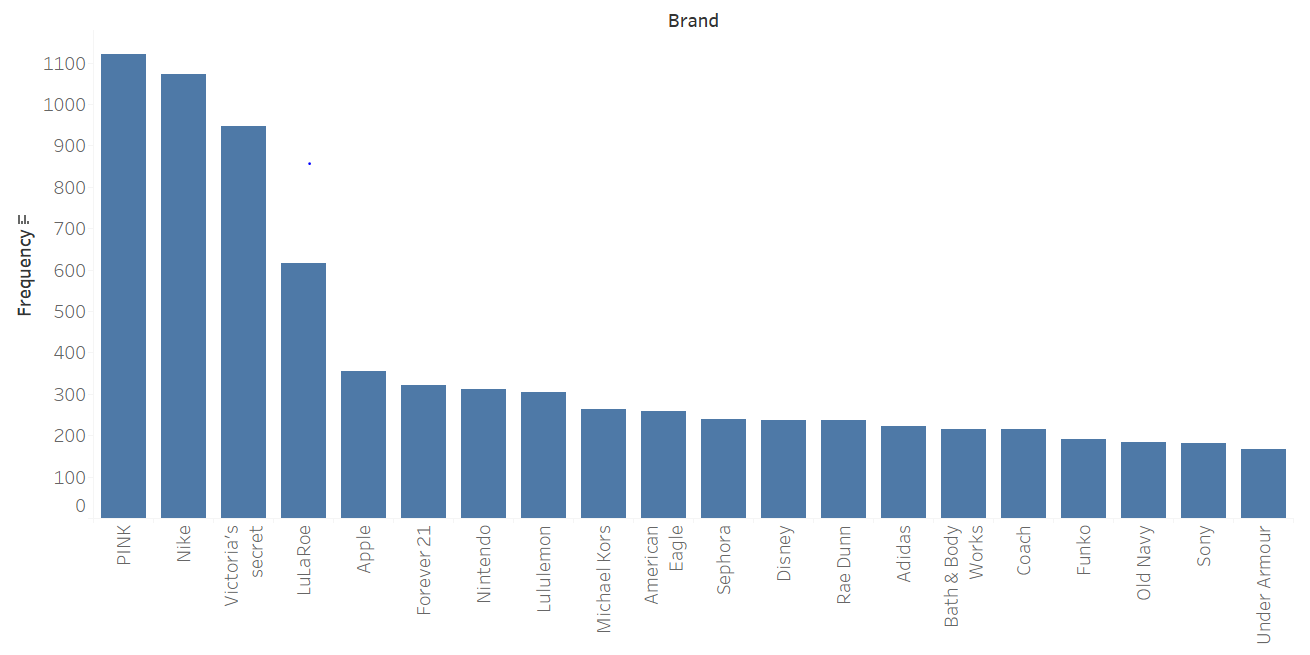
Upon exploring the top expensive brands based on mean and median price. Below are the statistics.

|  |  |  |  |
| --- | --- | --- | --- |
| Brand Name | Mean Price ($) | Median Price ($) | Standard deviation |
| Acacia swimwear | 26.5 | 17 | 0.40 |
| Razer | 27.5 | 17 | 0.41 |
| Nike | 26.9 | 17 | 0.40 |
| Victoria’s secret | 28.3 | 18 | 0.39 |
| Smashbox | 26.0 | 16 | 0.30 |
| Target | 23.0 | 18 | 0.56 |
| Scholastic | 29.6 | 21.5 | 0.40 |

Using the information in the above table we can say that the standard deviation is high for all the expensive brands in the training set.

**Top 20 Brands:**

Below bar chart shows the top 20 brands based on the occurrence ordered in descending order. Comparing above table and below bar chart, we can see that Nike and Victoria’s secret brands are most expensive and most used brands.



While performing Exploratory data analysis on brands of the product, it was found that there are missing values. Around 43% brand names are missing among 29999 records. Upon observing the other features for these missing values, analyzed that brands names for these records are mentioned in name of the product. By cleaning these records missing values percentage was dropped to 0.5.

**Methods:**

This part of literature review focus on the Traditional machine learning algorithms used by various researchers till now to address the problem.

**Multiple linear Regression:**

Based on several features Multiple linear regression helps to predict an outcome. Interrelationships between the variables can be explained with this type of model. Regression problem mainly with text as the feature needs creation of dummy variables. In regression the dependent variable should be normally distributed, in this case price column in the dataset in right skewed. By applying logarithmic to the price values, it is transformed to Normally distributed variable. The prediction equation of Multiple linear regression is given below:

Y = a + b1(item\_condition\_id1) + b2(item\_condition\_id2) + b3(shipping)+b4(category1)+……………………

Y – dependent variable.

a – intercept

b1, b2...bn – slope

The drawback of the Multiple linear regression algorithm is that it is not accurate as most predictive algorithms. In this type of regression model data is fit in N-dimension space instead of curvy surface which will fit the training data more accurately. So, this type of model often introduces bias when results are predicted with test data.

**XGBoost:**

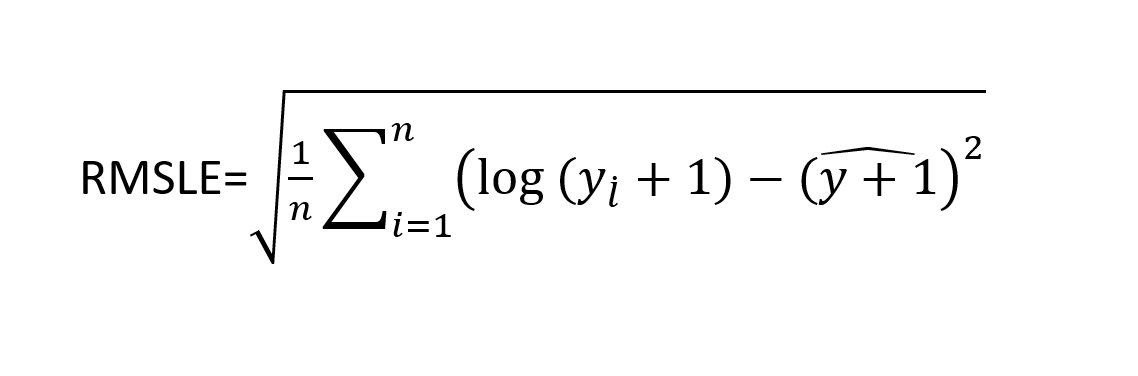
XGBoost is one of the extensions of gradient boosting. It stands for extreme gradient boosting. It is more efficient and runs faster when compared with other gradient boosting algorithms. For dynamic pricing predictions XGBoost can be well modelled. L1 and L2 regularization helps the model to penalize and avoid overfitting. The model limits input only to numerical variables in matrix format. This helps to manage different type of sparsity patterns in the data as there are huge number of brand names and categories in Mercari dataset. XGBoost formula for prediction is given below:

O(a) = L(a) + Ω(a)

L is the loss function for training data. It gives the range that the training data can predict the model accurately. Loss function used here is the traditional least squares regression. Ω is the regularization parameter.

**RMSLE:**

In general, the metric used to evaluate the regression tasks is Root Mean Squared Error (RMSE). As the price variable in the dataset is following long tailed distribution logarithmic is applied to price variable and the metric chosen to evaluate is Root Mean Squared Logarithmic Error (RMSLE). RMSE and RMSLE are used to find the difference between the predicted values using the model and the actual values. Both variance and bias are incorporated with Mean squared error (MSE). RMSE is the square root of MSE in case of unbiased estimator, on the other hand RMSLE is the difference between log of predicted values and actual values.



The RMSLE values obtained for Multiple Linear regression and XGBoost are given below:

|  |  |  |
| --- | --- | --- |
| Algorithm | RMSLE performance | |
| Train set | Test set |
| Multiple Linear Regression | 0.61 | 0.61 |
| XGBoost | 0.50 | 0.51 |

XGBoost algorithm performed well when compared with Multiple linear regression. XGBoost is able to handle around 4000 plus Brands and nearly 1000 categories as numeric factors, so it greatly helped in reducing RMSLE value. XGBoost weights the predictors and keeps the decision tree away from errors. But these models lack interpretability that support pricing decisions. Hence these models are very hard to interpret.

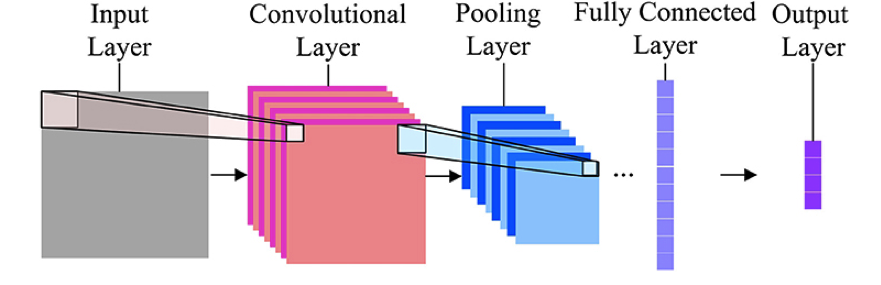
**Research Design and Methods**

This part of the paper focus on introducing neural networks and explains the architecture of the Convolutional Neural Network model to decrease the RMSLE values obtained from the Traditional machine learning algorithms.

Considering the tasks like Natural language, speech, vision and image identification neural networks exhibited its superiority among all the algorithms. Neural networks scale effectively with data when compared with classical algorithms. It also gives higher accuracies. Traditional algorithms require feature engineering. It includes normalizing data w.r.t dimensionality reduction. Whereas it is not necessary in the case of network. Data can be directly fed in to network, which makes the interpretation quite easy. In this paper research discusses the implementation of model using TensorFlow with CNN (Convolutional Neural networks) layers.

**CNN (Convolutional Neural networks):**

CNN is an artificial neural network. There are several layers in CNN. It takes input and will process the data before transforming it. Layers include a wide range of convolutional layers with kernels, fully connected layers, pooling layers and activation functions. By seeing the below diagram, it can be better understood.



In CNN, convolutional layers, pooling layers, fully connected layers and normalization layers are termed as hidden layers. Because these layers act as the activation functions. Pooling layer reduces the dimensionality of the input. To select the max value from the region we can use max pooling and for min value in the selected region min pooling is used.

**Implementation of CNN for Mercari dataset:**

The attributes name and item\_description are textual and these are the most important predictors of price. As users use different writing style, sequence of words and various semantics it is very hard to find the patterns in these attributes. NLP (Natural Language processing) is used as the solution for machine learning to find the patterns in these attributes. ELI5 framework helps to debug and improve text tokenization. It also analyzes the model feature importance. Vectorization should be performed for item description. So, famous technique TF-IDF is used. TF-IDF stands for term frequency – inverse document frequency. For each text in the description it generates a vector. The size of the vector is defined by corpus. In vectorization the bi-grams are used to add semantics with small phrases.

For CNN, text length is very important for variables which are given as text inputs. For product name first 20 words are fixed and for item description it will take up to 70 words. Special tokens were padded against shorter texts and sequences. There are many words which occur less than 40 times. These rare words are replaced by special tokens. For neural networks to do a better job categorical variable can be transformed to a different form. So, for shipping and item\_condition variables one hot encoding transformation was applied. Whereas for name of the brand and category of the product dense vectors were generated. For better representation to neural networks feature scaling was applied to description of the item column. Categorical features, condition and shipping are directly given as input.

Now the parameters should be tuned to increase the accuracy and avoid overfitting. Different parameters that can be tuned in CNN are explained below.

Number of Epochs: It defines number of times the model should learn the entire training dataset. If the number of epochs is very large after a certain point the accuracy starts decreasing. So, it is suggested to tune epoch value between 10 and 100.

Batch Size: It is one more hyperparameter that can be tuned to reduce the false predictions. Before updating the model parameters, it goes through the sample data. Popular Sample size include 32,128 and 256.

Learning Rate: It defines the learning advancement of the model. It is called the mother of all the hyperparameters.

Kernel width: It define the capacity of the model. It should be defined with two dimensional values.

Dropout: Dropout is used to avoid overfitting. In general, it is added to the hidden layers in the network. Values in the range of 0.4 to 0.75 is a better choice to test with. Kernel and bias initializers are used to reset the weights with random values.

**Obstacles:**

Handling missing values was the first obstacle in this paper. Because more than 43% of the brand names are missing. Brand names are filtered from the name of the product manually. Text analysis is the second obstacle that was faced in preprocessing data. Item descriptions has many different patterns, it was hard to analyze the descriptions. The last one is the computational time, as the dataset contains more than 20000 records fitting model on the training set took a lot of time.

**Anticipated Results**

Convolutional Neural Networks outperformed traditional machine learning algorithms in RMSLE on the test set, which was 0.42. The results obtained are easy to interpret. These results of the model are very helpful for Mercari as they want to find price drivers and predict the resulted price to the current price drivers.

**Impacts and improvements:**

Predicting the price of the products based on several kinds of features will be easier for sellers to know how buyers value the products. This is the major impact resulted in this research. XGBoost model is overfitting on the training data which will impact the predictions of test data. This problem is resolved after using CNN. Tuning regularization parameters in CNN avoided overfitting on the train data. GPU was used for processing which reduced processing of about 10 times.

**Discussions**

Understanding customer behavior and preferences becomes easy by performing this type of predictive analytics. To gain profits in dynamic markets, business must look ahead of their customers. Based on the shopping history and preferences of the user business can develop a recommendation to sellers. Adding more to that if the price of the product is more when compared with market place, sellers will end up getting loss. So Mercari will be the solution for those sellers.

**Insights:**

From the distribution of the brands, the main thing that can be inferred is that women and beauty products take up more than 50% of the distribution. Majority of the categories falls under Electronics. From the analysis, users are listing mainly beauty products and electronics. Now-a-days success of the online business depends on the amount of the people who use the application. Mercari got a splendid chance to increase the views of their application/website by implementing price prediction for product listings.

**Tools and Skills:**

Data preprocessing and data modeling tools are very important to save time in developing the code. Research paper used python libraries Numpy, pandas for data wrangling and data cleaning. Matplotlib for data visualization. Scikit-learn is used to build traditional models and find the performance of the models by computing classification report and confusion matrix. TensorFlow and keras are used to build convolutional neural network.

**Recommendations**

**Future study:**

This research helped me to learn a lot about parallelization and Natural language processing. Computational speed and memory errors are the main issues when we run the CNN model. To overcome these drawbacks, I want to use spark clusters or servers with large GPU’s as they have more advantages than others. Implementing LSTM (long short-term memory) in TensorFlow could resolve the limitations of CNN that is used to address the price prediction problem. LSTM is called recurrent neural network that helps to find the behavior in the description of product. Word embedding and hidden layers which are implemented in CNN can also be used in LSTM. In next quarter the research could be carried by exploring above mentioned deep learning models.

**New Trends:**

There are two trends that I am interested, the first one is the IOT (internet of things). Many organizations are moving to IOT as this gives more ways to collect, analyze and process the data. The second one is Predictive analysis, because this helps the business to know about customers future actions.

# **References**

Alex. (2019). *How to “farm” Kaggle in the right way*. Retrieved from https://towardsdatascience.com/how-to-farm-kaggle-in-the-right-way-b27f781b78da

Girish. (2018). *Mercari Price suggestion Challenge*. Retrieved from https://github.com/girishkathireddy/Mercari-Price-Suggestion-Challenge

Jesus. (2018). Retrieved from https://towardsdatascience.com/understanding-hyperparameters-optimization-in-deep-learning-models-concepts-and-tools-357002a3338a

Joshua D. McKenney, Y. J. (2018). *Competitive Approach for Online Price Prediction*. Retrieved from http://matthewalanham.com/Students/2018\_MWDSI\_XGBoost\_A%20Competitive%20Approach%20for%20Online%20Price%20Prediction.pdf

Li, S. (2018). Retrieved from https://towardsdatascience.com/machine-learning-for-retail-price-suggestion-with-python-64531e64186d

Lopes, H. (2018). *Data Science Trends for 2019*. Retrieved from https://towardsdatascience.com/data-science-trends-for-2019-11b2397bd16b

Mercari. (2018). Retrieved from https://www.kaggle.com/c/mercari-price-suggestion-challenge

Moreira, G. (2018). Retrieved from https://medium.com/unstructured/how-i-lost-a-silver-medal-in-kaggles-mercari-price-suggestion-challenge-using-cnns-and-tensorflow-4013660fcded

Zhang, W. (2018). *Online Compact Convexified Factorization Machine*. Retrieved from https://arxiv.org/pdf/1802.01379.pdf