EMBARC: Embeddings for Multilevel Product Analysis and Review Classification

Natrajan Parameswaran
Ganesh Kumar Koyya
Sai Sumana Brungi
Shyam Sunder Mundrika

1

Abstract:

- EMBARC (Embeddings for Product Analysis and Review Classification) is a comprehensive framework that combines the power of BERT embeddings, autoencoders and Optuna framework to generate embeddings at different levels on textual and numerical data in the dataset for analyzing product categorical prices and customer reviews. This approach aims to improve the accuracy of a regression model that helps in price prediction of a product category.

2

Motivation

- Accurate price prediction is of utmost importance in industries such as e-commerce and retail, as it directly impacts sales and profitability. Traditional methods overlook the insights hidden in textual data, such as product reviews, which greatly influence consumer purchasing decisions.
- Leveraging embedding techniques like BERT and the Optuna, we aim to create embeddings of reviews, price, rating, and number of ratings. The integration of these embeddings into the prediction model is expected to provide businesses with a more accurate understanding of the relationship between these factors, enabling them to optimize pricing strategies and ultimately improve sales performance.

3

ML Problem specification

- We are using amazon dataset from kaggle to create embeddings on different levels like ratings using auto encoders and reviews_title using Bert to improve the price pattern prediction of various product category algorithm. The dataset has salient features of numeric data like actual price, discounted price, ratings and ratings count which is used in training the model.
- Finally, Optuna framework is used to do hyperparameter tuning to find out the best hyperparameter setting for embeddings creation using auto encoders. Columns irrelevant to the price prediction like product_id, user_id, image, product_link and those columns having more than 10% NaN values are excluded in the prediction problem. Also features like Reviews_title is used to create textual embeddings.

4

Data specification:

- **Dataset:** https://www.kaggle.com/datasets/karkavelrajaj/amazon-sales-dataset
- **Features description:**
- product_id - Product ID
- product_name - Name of the Product
- category - Category of the Product
- discounted_price - Discounted Price of the Product
- actual_price - Actual Price of the Product
- discount_percentage - Percentage of Discount for the Product
- product_link - Official Website Link of the Product

•rating - Rating of the Product
•rating_count - Number of people who voted for the Amazon rating
•about_product - Description about the Product
•user_id - ID of the user who wrote review for the Product
•user_name - Name of the user who wrote review for the Product
•review_id - ID of the user review
•review_title - Short review
•review_content - Long review
•img_link - Image Link of the Product

5

Data specification:

```
# Displaying the correlation matrix
print(correlation_matrix)
```

```
                     discount_percentage  discounted_price  actual_price  \
discount_percentage             1.000000         -0.242298     -0.117855
discounted_price               -0.242298          1.000000      0.961910
actual_price                   -0.117855          0.961910      1.000000
rating                         -0.155679          0.121132      0.122467

                       rating
discount_percentage -0.155679
discounted_price     0.121132
actual_price         0.122467
rating               1.000000
```

```
RangeIndex: 1465 entries, 0 to 1464
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   product_id           1465 non-null   object
 1   product_name         1465 non-null   object
 2   category             1465 non-null   object
 3   discounted_price     1465 non-null   object
 4   actual_price         1465 non-null   object
 5   discount_percentage  1465 non-null   object
 6   rating               1465 non-null   object
 7   rating_count         1463 non-null   object
 8   about_product        1465 non-null   object
 9   user_id              1465 non-null   object
 10  user_name            1465 non-null   object
 11  review_id            1465 non-null   object
 12  review_title         1465 non-null   object
 13  review_content       1465 non-null   object
 14  img_link             1465 non-null   object
 15  product_link         1465 non-null   object
dtypes: object(16)
memory usage: 183.2+ KB
```

6

Design and Milestones:

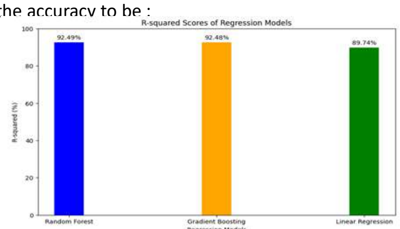
- Data Preprocessing:
  - Of the initial 16 columns we dropped 8 columns that are unused and left with 8 features.
  - On the features we refined the values by removing symbols like (₹, %,',') and converting the value into numeric (float) format.
  - Drop all N/A values.
  - For the column 'Category' we extracted the first 2 subcategories into a new feature.
  - For the 'Review Title' we extracted the first 6 words and even removed special characters that might have been added.

Design and Milestones

- We first used Optuna framework for hyperparameter setting, Based on its selection embeddings were created embeddings at Price, Rating levels and we used BERT language model to create textual embeddings on Review_title. The 2 embeddings columns, embeddings and embedding_review were then merged into our dataset and then price prediction was done in un use case data.

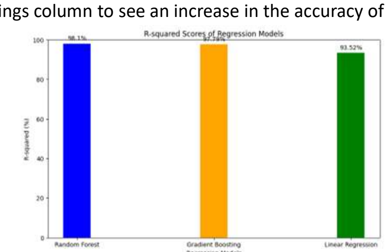
Design and Milestones

- We performed 3 supervised prediction techniques namely, random forest, gradient boosting and linear regression on the use case data noting the accuracy to be :



Design and Milestones

- Then we performed the same on the use case data along with embeddings column to see an increase in the accuracy of the three models :



Repository / Archive:

- Steps to run:
  - Import the [dataset]() into google drive.
  - Run the files in colab.
  - https://colab.research.google.com/drive/1igWHMG1WkWLS4DX_Nr7hEUUbPhMyir5C?usp=sharing
  - https://colab.research.google.com/drive/1uSWxQwFG4jbE7UIaxam5m5SLy6Il2CCX?usp=sharing

Code:

Resources and Related Projects:

- Asudani, D.S., Nagwani, N.K. & Singh, P. Impact of word embedding models on text analytics in deep learning environment: a review. Artificial Intelligence Review (2023). https://doi.org/10.1007/s10462-023-10419-1
- The above article guided us on BERT usage.

- https://towardsdatascience.com/vector-representation-of-products-prod2vec-how-to-get-ridof-a-lot-of-embeddings-26265361457
- We used the above as a reference to see how price pattern prediction can be done on several product categories. Here they have used prod2vec but in our case we are using Textual and Numerical Embeddings.

13

What's next?

- Optimizing the BERT embeddings creation using powerful GPUs.
- Can create embedding on larger real time datasets for future analysis.

14