Kaggle AI Challenge "Product repurchase prediction"

Cesar Anasco [*1] **DoPhan BaoKhang** [*1]

Abstract

Carrefour's 2027 strategic plan emphasizes data-driven transformation offering an educational data science challenge in partnership with the University of Bordeaux. Several analysis and 2 models were implemented to predict the products that Carrefour customers will repurchase using their history of frequent purchases. The report focuses on the implementation of different approaches to predict purchases by ranking custom products recommended to each customer.

1. Introduction

The objective of this kaggle competition is to develop a product recommendation system to predict the items customers are most likely to repurchase using the Frequent Purchase Carousel, a feature available on Carrefour's website.

As competitors we where assigned the challenge to implement methods to predict the first transaction for a given customer based on their historical purchase data and the associated product information. From this we assign a ranking between 1 and 10, 1 being the most likely one to be bought, in order for the carrefours website to recommend these ranked products and improve customer satisfaction.

The project presents three datasets:

- Product information detailing items available for sale.
- A training dataset containing the purchase history of 100,000 customers from 2022 to 2023.
- A validation dataset with the transaction history of 80,000 customers in 2024.

The evaluation metric used in this competition is called "Hit Rate @10", which corresponds to the accuracy of the predicted products inside the ground-truth test set. In other words, out of the 10 recommended products, how many are actually purchased by the customer.

The approaches implemented in this report include the use of data analysis and the information gained for it, weighted coefficients for relevance recommendations, fine tuning of large language models for context generation of products and deep neural network for purchase history analysis and predictions.

2. State of the art

In the context of purchase prediction and recommendation systems, several state-of-the-art models have demonstrated their potential, providing advances in Machine Learning and Deep Learning.

From this, several models were considered fpr this work as follows:

1. Sequence Models:
    - Hierarchical Recurrent Neural Networks (RNN) and Point Process model (Bjørnar Vassøy, 11–15, 2019): ideal for complex session-based recommendations where both inter-session dependencies and time gaps between sessions are crucial, such as predicting next-session interactions and return-time.
    - RNNs with Long Short-Term Memory (LSTM) layer model (Graves, 2013): flexible option for sequential recommendation tasks where longer interaction histories are available and temporal dependencies are essential but do not require a session-level hierarchy. It captures sequential dependencies effectively but does not inherently separate sessions.
    - Autoencoder Gated Recurrent Unit (GRU) model (Xin Chen, 2013): Efficient for real-time, contextually relevant recommendations that need to quickly respond to immediate patterns in transactional data, but may need fine-tuning for longer-term dependencies.

[*]Equal contribution [1]Université Jean Monnet. Correspondence to: Cesar Anasco <cesar.anasco@etu.univ-st-etienne.fr>, DoPhan BaoKhang <first2.last2@www.uk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

2. Attention-Based

    - BERT: Bidirectional Encoder Representation from Transformer (BERT) is a pre-processing mechanism that, based on representations of the general language, is intended to fulfill the requirements of more specific conditions like transfer learning. It is a language model based on the transformer architecture (Vaswani, 2017) and uses the attention mechanism for next token prediction and classification.
    - Models like BERT allow you to learn this intense pre-training process and have it tested in tests like these components are working. This process of adjusting to new information is known as Fine Tuning (fine tuning) (Gao et al., 2019).

3. CNN-Based Model:

    - CASER (Convolutional Sequence Embedding Recommendation) (Jiaxi Tang, 2018): Uses CNN layers on both horizontal and vertical relationships in user-item sequences and able to effectively captures sequential patterns in shorter sequences without the need for recurrent connections, but may not capture bidirectional dependencies as effectively as attention-based models.

After extensive research and evaluation, BERT and CASER were selected as the most suitable models for this project because of their complementary strengths and proven effectiveness in similar applications. By integrating these models or using them in a comparative framework, the project aims to achieve robust and scalable purchase prediction.

3. Methodology

3.1. Data Analysis

The analysis initially focused on the relationship between purchase quantity and recency, and later we incorporated frequency for deeper insights.

3.1.1. Analysis of the top most recently purchased products of each customer using Quantity and Recency

The 2 main factors that we used for this part are:

- Quantity: The number of items purchased in each transaction.
- Recency: The time elapsed since the last purchase for a specific product or customer which can be calculated by: $Recency = Lastday - Firstday$

The equation (1) combines the key behavioral factors — quantity and recency — into a single Score to rank items for recommendation for each customer. To assign a relative importance (via weights α), reflecting their contribution to predicting future purchases.

$$Score = \alpha.Quantity + (1 - \alpha).Recency \quad (1)$$

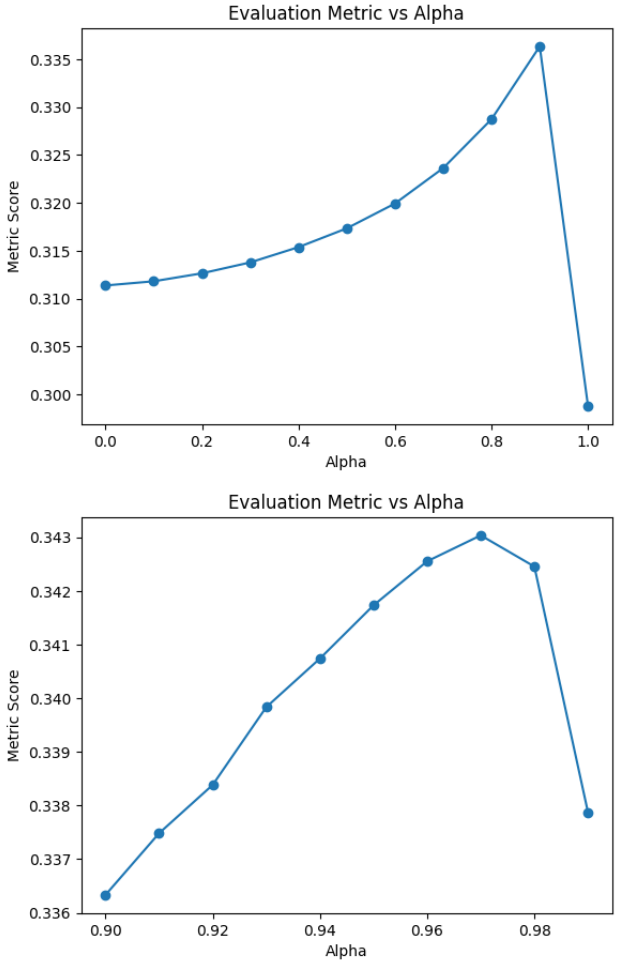

Figure 1. Tuning α

We also found that in Figure 1, at $\alpha = 0.97$ we're able to achieve the highest hit rate score which represent the contribution of the factors into the final prediction.
Customers seem to prefer items they purchase in larger quantities, even if those purchases were not recent. This simple equation may not fully capture the complexity of customer behavior, which is why this approach relies heavily on Quantity.

3.1.2. Incorporating Frequency

Since Frequency is known for its effectiveness in recommendation task, we thought it would be good to see how it performs with Quantity and Recency.

- Frequency: The number of transactions a customer makes in a given period.

And so we updated the equation as following:

$$Score = \alpha.Quantity + \beta.Frequency + \gamma.Recency \quad (2)$$

$$\alpha + \beta + \gamma = 1 \quad (3)$$

The equation (2) represents a weighted scoring mechanism used to rank or evaluate items based on 3 key factors: Quantity, Frequency, and Recency; and it must satisfy equation (3) to ensure a balanced combination; after that we use the Score as a composite metric that combines these factors to rank products for recommendation.

After tuning as in Figure 2, we found that by ranking the recommendation items with the resulted Score from α = 0.03, β = 0.87, γ = 0.1, it gave us the highest hit rate score of 0.35694 as in our submission on the Kaggle competition. From this result, we can observe clearly that Frequency is the most influential factor in predicting repurchase likelihood, accounting for 87% of the weight.
This suggests that customers are more likely to repurchase items they purchase frequently, regardless of Quantity or Recency. This equation is more nuanced because it incorporates 3 factors, allowing for a richer representation of customer behavior.

3.1.3. Summary of Findings

1. Achieve a higher score than the baseline.
2. Acquire useful information to enhance the performance of the ML models.

3.2. Data Preprocessing

3.2.1. Products Embedding with Bert

The baseline provided by the competition does not use in any form the product dataset information, and it was for competitors to use this data source in a effective way. As described in Appendix A, the datasets for products have a large amount of associated information for each product, most of the columns being used as a binary description of whether the product is defined by the specific column.

The processing of the product data set started with the selection of certain features. Given the large amount of textual information, the restructure started by creating a single text

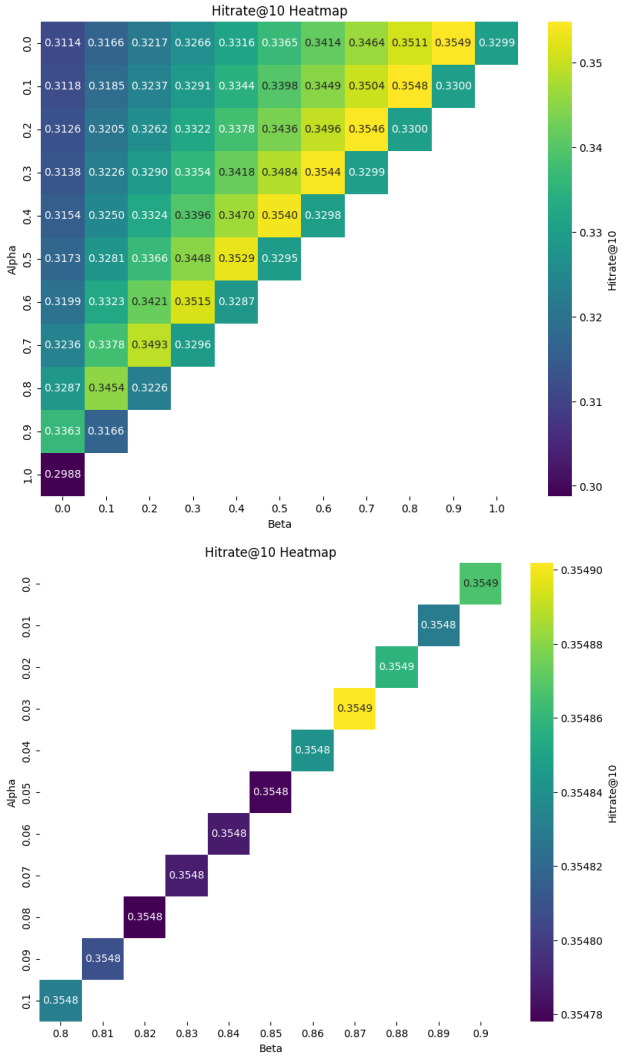

Figure 2. Tuning α, β

defining each product by the selection of specific features (not all columns were added to this description). As seen in Appendix B the process started by using all columns in English, since the pre-trained LLM "Bert-Uncased" was based on English text (Behera & Dash, 2022). The next step was to unify all selected features as seen in Appendix C, this special structure used token separation that is understood by the Bert model and creates more meaningful embeddings. As of the transformer architecture released in 2018 (Vaswani, 2017), Bert is based on learned embedding that are trained and updated based on the information context (Wang, 2024).

The embedding creation had three important steps:

- Restructuring of the column features as mentioned in the previous paragraph.

- Prediction of the "Shelf Level 1" feature on each product by fine-tuning the LLM.
- Final embedding creation with fine tuned Bert model.

The results from this process gave us a 768 length of embedding numpy array done to 82966 products. This is used as a dictionary when the product is encountered inside the historic training data rows described in the next section.

3.2.2. Historic Data Feature Engineering

This training data contains the transaction history of 100,000 customers, as described in Appendix A. It is divided into 10 parts, each containing the transactions of 10,000 customers for the years 2022 and 2023. The analysis described in section 3.1 was added as the first feature engineered score, these associated a top of recommended products to each customer on the dataset. Listed are the next new columns added to the train data corpus:

- Date related columns: we extracted the month, day and year from the original date column.
- Average Quantity: average number of items a client purchases per transaction.
- Promo Ratio: proportion of purchases made by a specific customer that included products on promotion by transaction.
- Days Since Last Purchase: days that have passed since the customer's last purchase.
- Unique Products : The total number of distinct products a customer has purchased over time.

Since the objective prediction from this data corpus is based on the relevance of each transaction made by each customer, we created new rows of non-relevant examples with two types of sampling:

- Easy Negatives: Negatives that are different or selected away from the target and don't provide much learning signal. The model can easily distinguish them, leading to faster convergence, but to a suboptimal representation.
- Hard Negatives: Negatives that are more similar to the target force the model to learn finer distinctions. This improves the quality of embeddings and decision boundaries. In this case, hard negatives were selected by evaluating the similarities of the embeddings obtained from the product descriptions.

3.3. Models Implementation

3.3.1. Bert with FNN

The complete dataset for model training can be summarized with Figure 3. The columns that were left were the ones listed in section 3.2.2, the relevant binary column and the added products embedding explained in section 3.2.1.

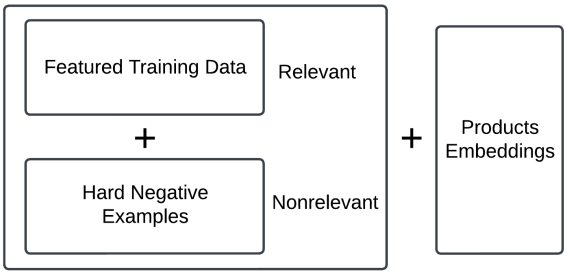

Figure 3. Complete Data Corpus

There where two architectures implemented for this data corpus:

- LSTM: is recommended for finding patterns in sequential data. As described in section 2, is well sited for detecting patters in customer cycle of purchases.
- FNN: Given that we based on feature relationships, Fully Connected Neural Networks (FNN) can identify correlations between features like purchase frequency, recency, and promotions. Features used inside our corpus.

The code implemented on FNN used TensorFlow and Keras for a binary classification task. The training dataset is split into training and validation with 20% of the data is reserved for validation. The model consists of an input layer that matches the number of features in the dataset, followed by three dense layers with 256, 128, and 64 neurons, respectively. Each layer employs the ReLU activation function, with a L2 regularizer to prevent overfitting. The output layer uses a single neuron with a sigmoid activation function, producing probabilities for binary classification.

The model is trained using binary cross-entropy loss, and the result of the prediction done on the training model involve the evaluation of the top 50 products the customer has frequently bought and assigning the probabilities obtained from the model to each product and get the top 10 values closest to 1.

3.3.2. Caser

1. Initial basic implementation: from scratch based on the paper (Jiaxi Tang, 2018)
   Training setup:

- Loss function: Cross-entropy loss.
- Optimization function: Adam with default learning rate.
- Train 1 file at a time, 10 times in total for all 10 files

Result: Evaluate the resulted models on the test dataset and obtain the Hit rate @10 score of 0.

2. Adding relevant features and Final_score:
Selecting features:

   - Customer-level attributes: loyalty card, frequency...
   - Item-level attributes: product category (frozen, fresh,...)

   By adding the embedding layers of these additional features, concatenate to the corresponding original embedding of customers or items and modify the fully connected (FC) layers to process the combined feature set and finally at the output of the model, we also add the Final_score as leverage in the prediction of each customer, the resulted model trained on 1 file achieved the hit rate @10 score of 0.001.

3. Negative sampling:

   - The negative samples have the same sector and shell level with the most frequently bought products of the customer. The score gained from this method is 0.1013.
   - The negative samples are the popular products that are in the top 50 most frequently bought products of all customers. The score gained from this method is 0.102.

4. Experimental evaluation

The experiments done on the test dataset after training, presented in Table 2, began with LSTM as a starting point with a prediction of quantity which had a low rate. The next implementation included FNN by adjusting and adding negative samples and the feature relevance prediction, which improved the hit rate obtained. At the end of these approaches the maximum number of customers trained inside the FNN was 40000 with their transaction history.

5. Conclusion

We achieved the highest score of 0.356 using a weighted scoring analysis based on three key factors: Frequency, Recency, and Quantity. This method demonstrated its effectiveness in aligning with customer purchasing behavior. However, as a non-learning, heuristic-based approach, it inherently has a ceiling on its performance and adaptability to more complex patterns.

Model	Training Data	Prediction	Clients	Hit
LSTM	No Negatives	Quantity	1000	0.02
FNN	No Negatives	Quantity	3000	0.12
FNN	Easy Negatives	Relevance	3000	0.25
FNN	Easy Negatives	Relevance	20000	0.18
FNN	Hard Negatives	Relevance	40000	0.19
CASER	Hard Negatives	ProductId	80000	0.10

Table 1. Comparison of models results HitRate

In contrast, the results from our experiments with the CASER and Bert-FNN model illustrate iterative improvements through the incorporation of additional features, innovative sampling strategies, and architectural enhancements. These experiments show that deep learning models have significant potential for further development and refinement. While the data analysis method currently outperforms the deep learning approach, its limitations become apparent when considering scalability and adaptability to evolving customer behaviors. The deep learning model, despite its initial performance gap, shows promise as it can continually improve through fine-tuning, advanced feature integration, and optimization techniques.

6. Percentage of Involvement

Name	Percentage
DO PHAN Bao Khang	50%
Cesar Anasco	50%

Table 2. Percentage of involvement of each group member

References

Behera, S. K. and Dash, R. Fine-tuning of a bert-based uncased model for unbalanced text classification. In *Advances in Intelligent Computing and Communication: Proceedings of ICAC 2021*, pp. 377–384. Springer, 2022.

Bjørnar Vassøy, Massimiliano Ruocco, E. d. S. d. S. E. A. Time is of the essence: a joint hierarchical rnn and point process model for time and item predictions. *The Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19)*, 11–15, 2019.

Gao, Z., Feng, A., Song, X., and Wu, X. Target-dependent sentiment classification with bert. *Ieee Access*, 7:154290–154299, 2019.

Graves, A. Generating sequences with recurrent neural networks. 2013.

Jiaxi Tang, K. W. Personalized top-n sequential recommendation via convolutional sequence embedding. 2018.

Vaswani, A. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Wang, M. L. *Fine-Tuning BERT for Sentiment Analysis*. PhD thesis, UCLA, 2024.

Xin Chen, Alex Reibman, S. A. Sequential recommendation model for next purchase prediction. *Computer Science Information Technology (CS IT), 13(10), 141-158*, 2013.

A. Datasets Descriptions

train_data.csv

This dataset contains two years (2022 & 2023) of historical transactions for 100,000 Carrefour customers. It has 10 columns:

- **date**: Date of the transaction.
- **transaction_id**: ID of the transaction.
- **customer_id**: Customer ID.
- **product_id**: Product purchased.
- **has_loyality_card**: Flag indicating whether the customer has a loyalty card.
- **store_id**: Store where the purchase was made.
- **is_promo**: Flag indicating whether there was a discount on the product.
- **quantity**: Quantity purchased of the product.
- **format**: Ecommerce activity format.
- **orderChannelCode**: Indicates whether the online activity was made through the website or mobile app.

products_data.csv

This dataset contains detailed information about the products. The following columns are relevant to this project:

- **product_id**: Product name.
- **product_description**: Product description.
- **department_key**: Department key.
- **class_key**: Class key.
- **subclass_key**: Subclass key.
- **sector**: Sector name.
- **brand_key**: Brand name.
- **shelf_level1**: Top-level shelf category.
- **shelf_level2**: Second-level shelf category.
- **shelf_level3**: Third-level shelf category.
- **shelf_level4**: Fourth-level shelf category.
- **bio**: Indicates whether the product is organic.
- **sugar_free**: Indicates whether the product is sugar-free.
- **aspartame_free**: Indicates whether the product is aspartame-free.
- **gluten_free**: Indicates whether the product is gluten-free.
- **halal**: Indicates whether the product is halal.
- **casher**: Indicates whether the product is kosher.
- **eco_friendly**: Indicates whether the product is eco-friendly.
- **local_french**: Indicates whether the product is locally produced in France.
- **artificial_coloring_free**: Indicates whether the product is free of artificial coloring.
- **taste_enhancer_free**: Indicates whether the product is free of taste enhancers.
- **naturality**: Naturality score.
- **antibiotic_free**: Indicates whether the product is antibiotic-free.
- **reduced_sugar**: Flag indicating whether the product has reduced sugar content.
- **vegetarian**: Flag indicating whether the product is vegetarian.
- **pesticide_free**: Flag indicating whether the product is pesticide-free.
- **grain_free**: Flag indicating whether the product is grain-free.
- **no_added_sugar**: Flag indicating whether the product has no added sugar.
- **salt_reduced**: Flag indicating whether the product has reduced salt content.
- **nitrite_free**: Flag indicating whether the product is nitrite-free.
- **fed_without_ogm**: Flag indicating whether the animals were fed without GMOs.
- **no_added_salt**: Flag indicating whether the product has no added salt.
- **no_artificial_flavours**: Flag indicating whether the product has no artificial flavors.
- **porc**: Indicates whether the product contains pork.
- **vegan**: Indicates whether the product is vegan.

- **frozen**: Indicates whether the product is frozen.
- **fat_free**: Flag indicating whether the product is fat-free.
- **reduced_fats**: Flag indicating whether the product has reduced fat content.
- **fresh**: Flag indicating whether the product is fresh.
- **alcool**: Flag indicating whether the product contains alcohol.
- **lactose_free**: Flag indicating whether the product is lactose-free.
- **phenylalanine_free**: Flag indicating whether the product is phenylalanine-free.
- **palm_oil_free**: Flag indicating whether the product is palm oil-free.
- **ecoscore**: Ecoscore.
- **produits_du_monde**: Flag indicating whether the product is an international product.
- **regional_product**: Flag indicating whether the product is a regional product.
- **national_brand**: Flag indicating whether the product is a national brand.
- **first_price_brand**: Flag indicating whether the product is a first-price brand.
- **carrefour_brand**: Flag indicating whether the product is a Carrefour brand.

test_data.csv

This dataset contains the actual purchases of the first 80,000 customers in 2024. It has 3 columns:

- **transaction_id**: ID of the transaction.
- **customer_id**: Customer ID.
- **product_id**: ID of the purchased product.

B. Translated features from products dataset

The three following columns where translated from french to english using the translation library ”googletrans 4.0.2 ” (**?**). Inside each square we can see the example values for shelf levels 1 and 2, being the more extensive ones, and the sector column having only 7 values to transalte.

Shelf Level 1

```
{   'Boissons': 'Drinks' ,
    'Epicerie salée': 'Savory Groceries',
    'Fruits et Légumes' : 'Fruits and Vegetables'
    "Laits : 'Milk'
    ...
}
```

Shelf Level 2

```
{   'Boissons': {
    'Colas' : 'Colas',
    'Thés Glaces' : 'Iced Teas',
    'Sirops et Sodas' : 'Syrups and Sodas'}
    ...
}
```

Sector

```
{   'PGC': 'Fast-Moving Consumer Goods',
    'PRODUITS FRAIS TRANSFORMATION': 'Fresh product processing',
    'BAZAR': 'Bazar',
    'EPCS': 'Electro Photo Cine Sound',
    'TEXTILE': 'textile',
    'ACTIVITES PERIPHERIQUES': 'Peripheral Activities'}
```

Figure 4. Translated column to english in products dataset

C. Structured text before fine tuning and context embedding process

The following graph shows the structured text provided to the ”bert-uncased” large language model (Behera & Dash, 2022). The embedding was generated without the Shelf Level 1, given that this column was the one predicted to get more meaningful embedding from the text generation. We also make use of the ”[SEP]” special token used with Bert to separate each column value inside the sentence and analyze multiple segments at the same time (Wang, 2024).

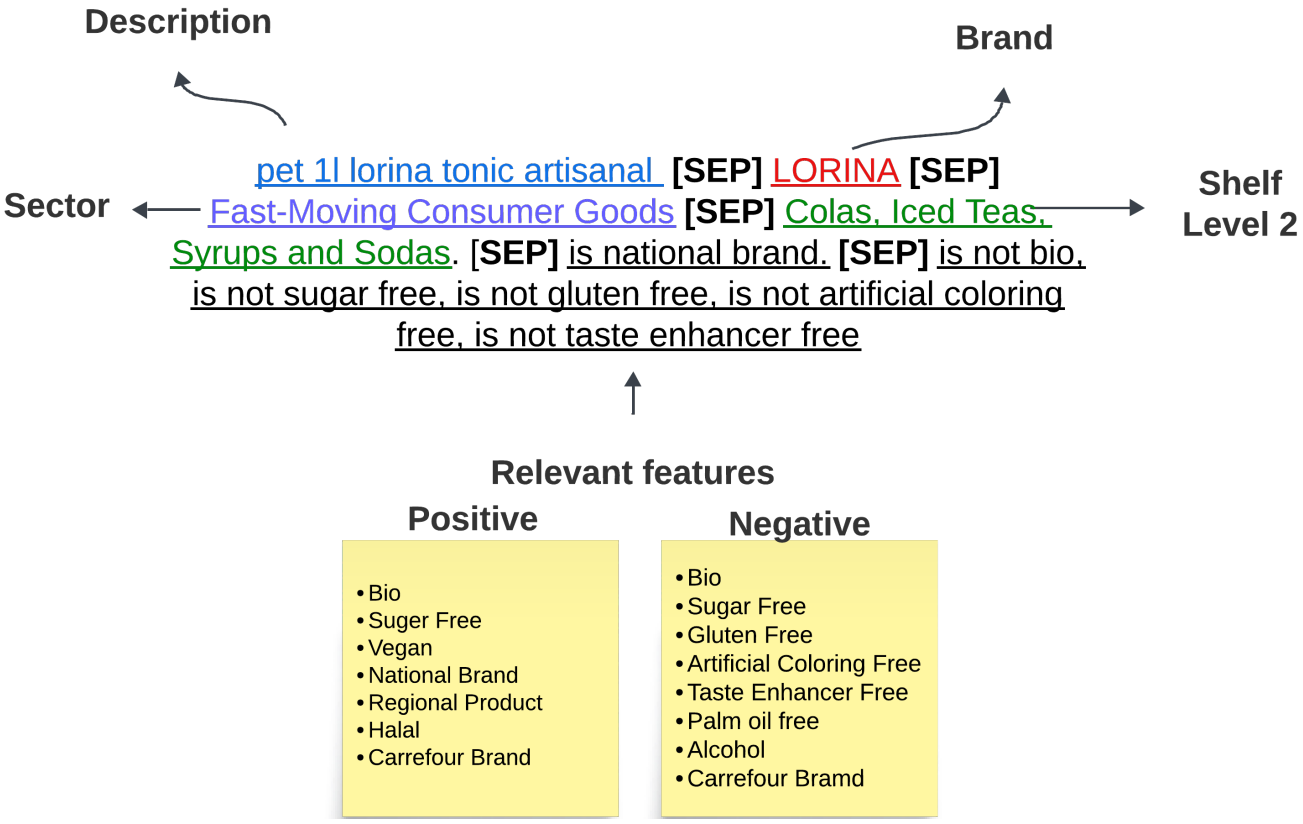

Figure 5. Structured text for product features embedding generation

D. Feature Engineering analysis

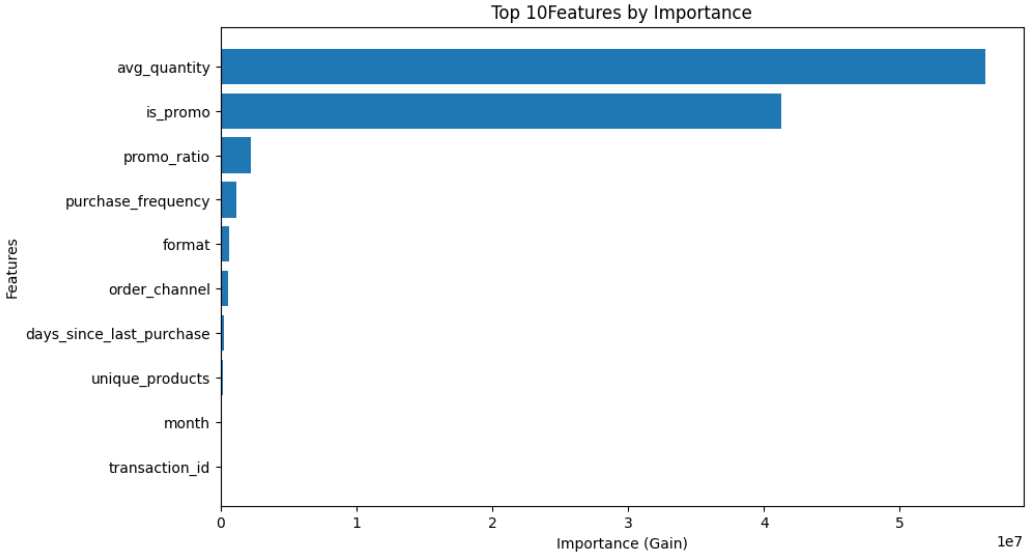

Figure 6. Feature importance on training data

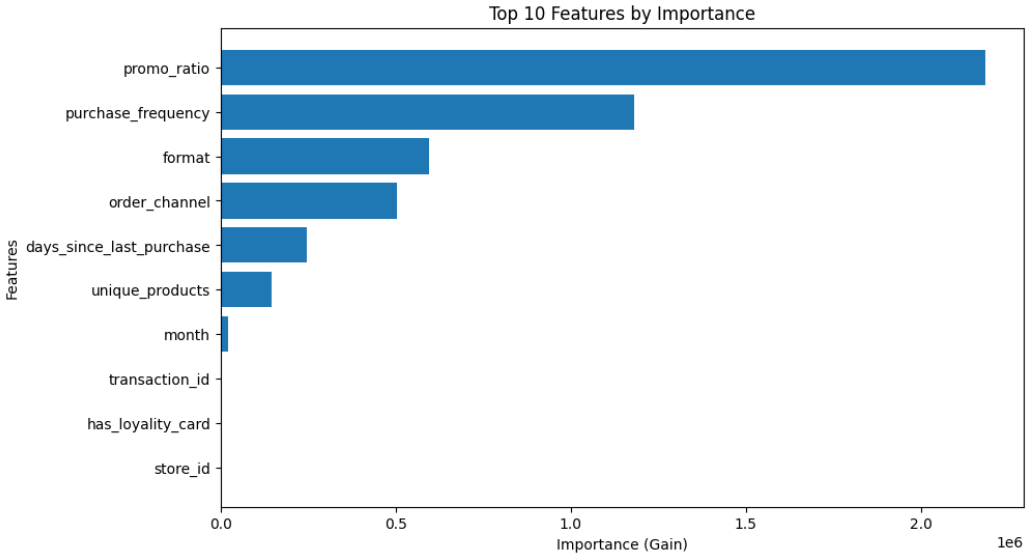

Figure 7. Feature importance on training data without top 2 more important features