Product Recommendation System

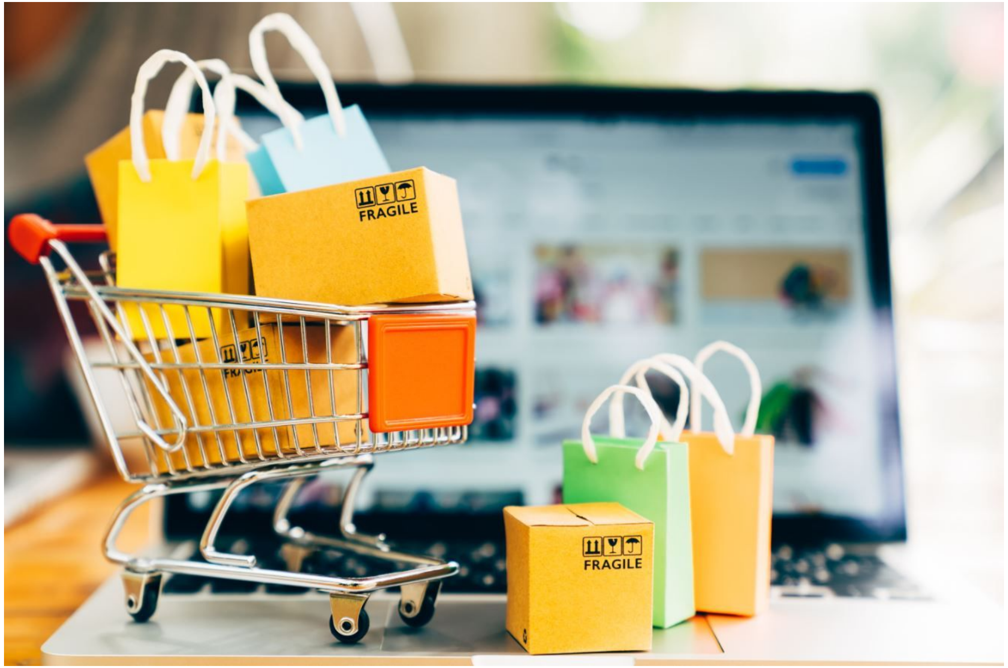
Team 4

Agenda

1. Problem
2. Motivation
3. Objective
4. Previous approach.
5. Proposed approach.
6. Detailed design of solution.
7. Extensibility.
8. Future development.

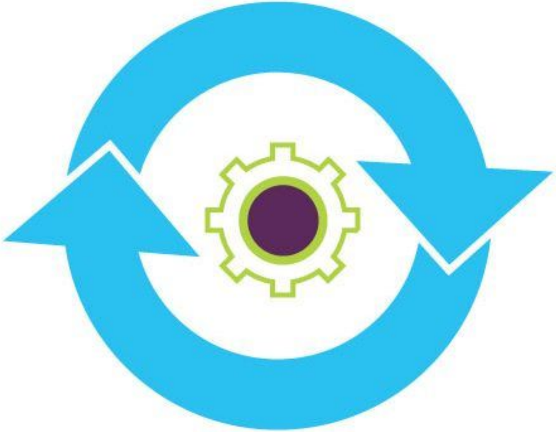
1. Problem

Online shopping has become more and more popular

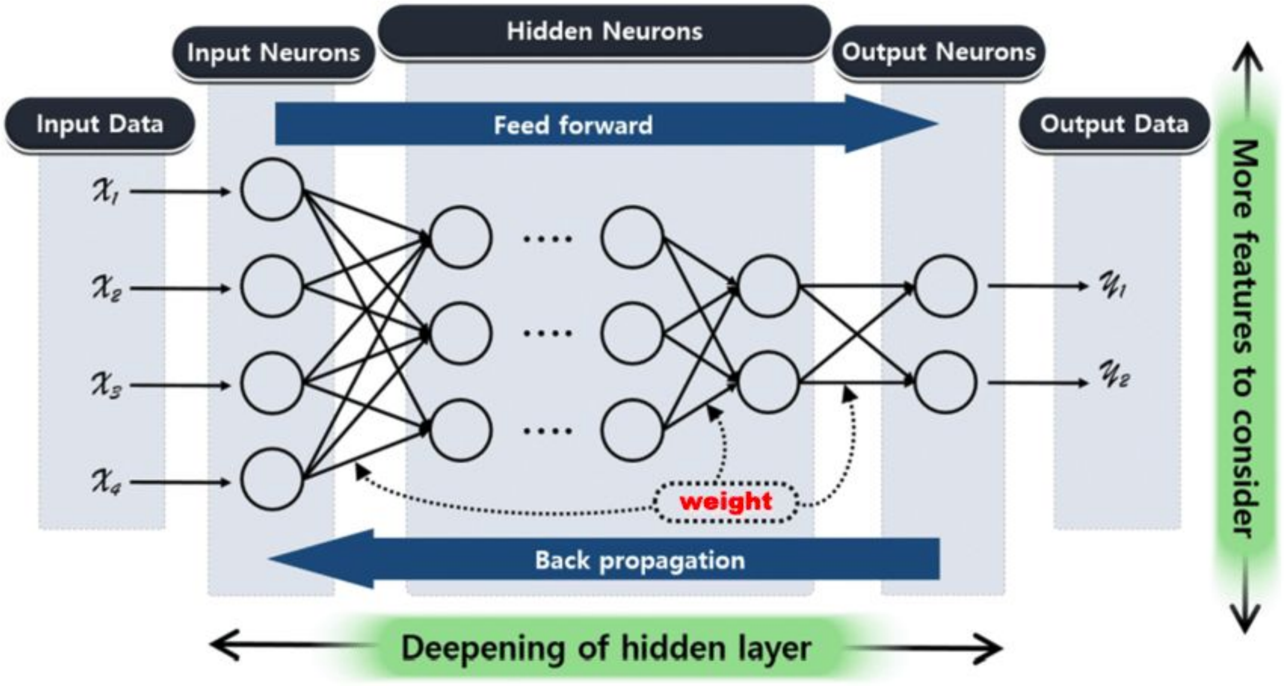


2. Motivation

Many small business cannot maintain a team for developing a ML classifier on their own



3. Objective



Hidden Neurons
Input Neurons
Output Neurons
Input Data
Feed forward
Output Data
x_1
x_2
x_3
x_4
y_1
y_2
weight
Back propagation
Deepening of hidden layer
More features to consider

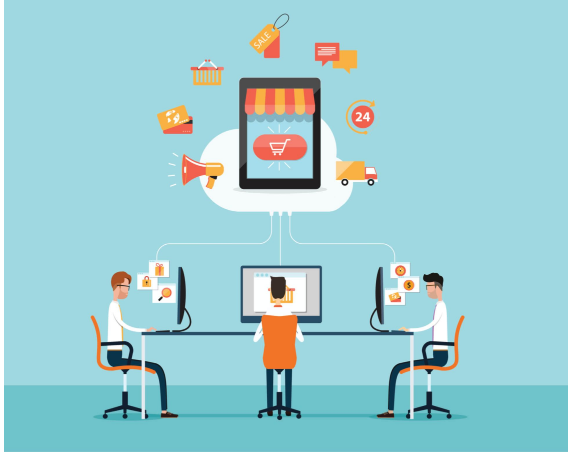
3. Objective

- Portable deep learning model for product recommendation with high reusability

- Reduce cost of feature engineering and data-driven customization.

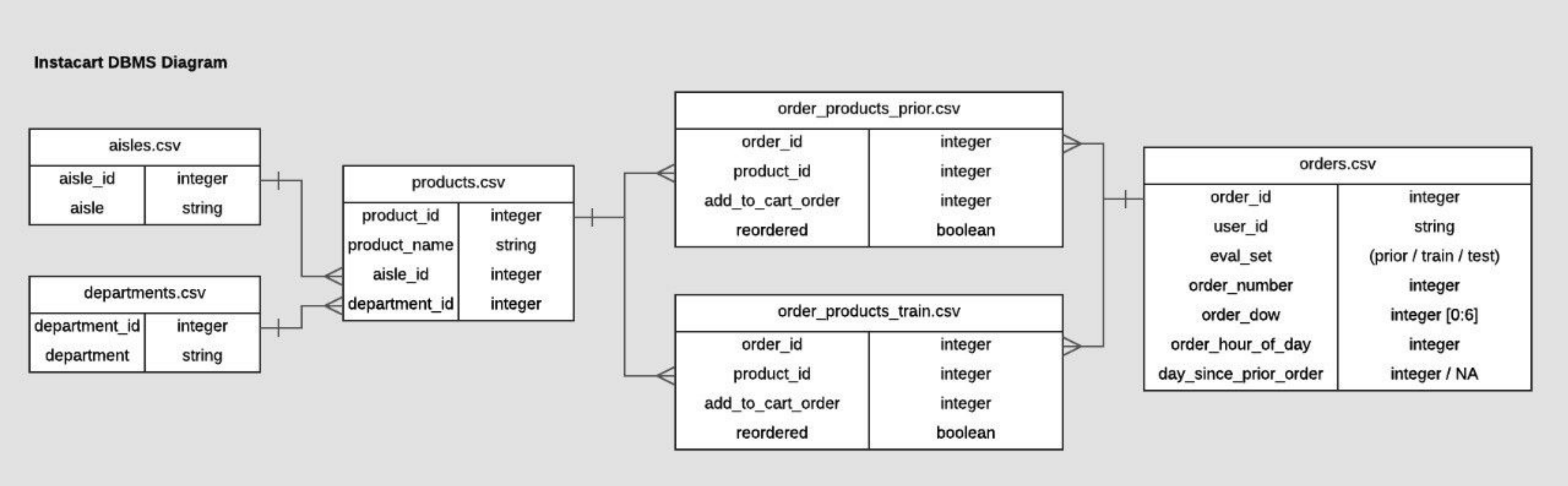
- Simple input-output pineline. No need a whole dedicated team for developing ML model

Target: small online shopping providers



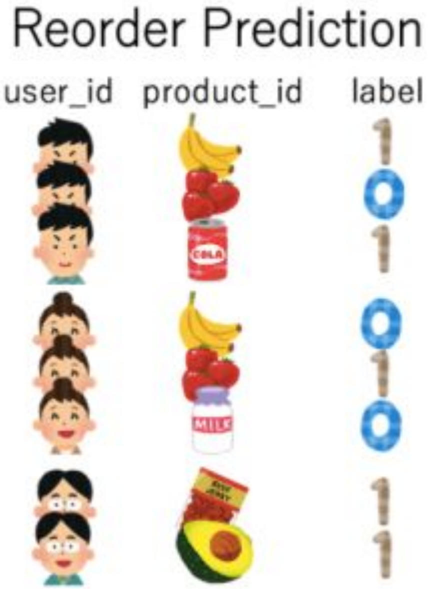
4. Previous Approach

Dataset: Using dataset and evaluation scheme from Kaggle competition: **Instacart Market Basket Analysis**



4. Previous Approach

- Recommendation-based problem, also multi-label classification problem

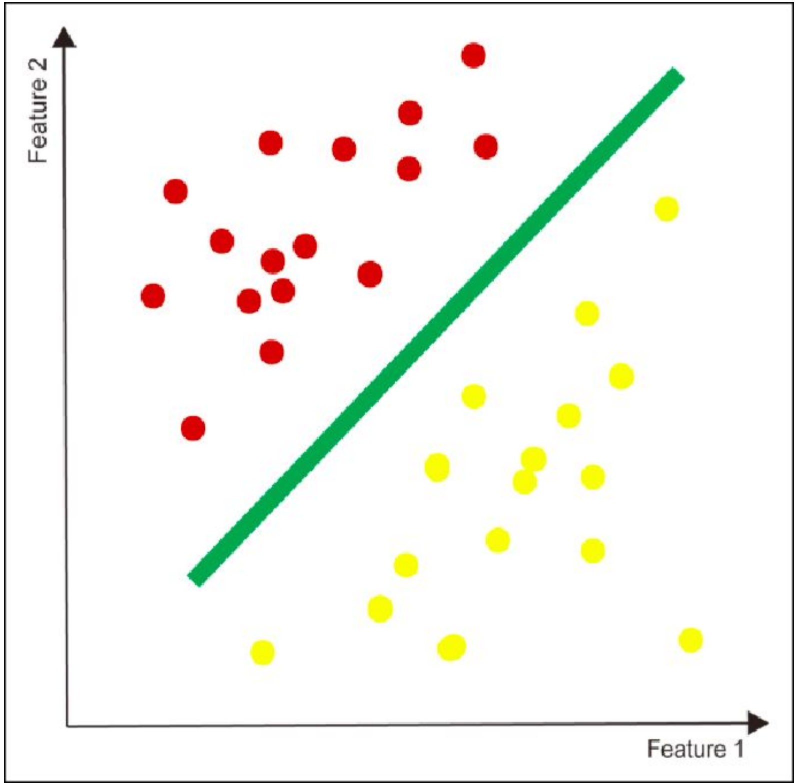


4. Previous Approach

Binary classification problem

$$L = -\frac{1}{M}\sum_{1}^{M} y^{i}.\log p(y^{i}) + (1-y^{i}).\log p(1-y^{i})$$

$$F1\ score = 2 * \frac{Precision\ * Recall}{Precision + Recall}$$



4. Previous Approach

3.2. **Variable threshold** with feature engineering and modeling

- Product only features:
    - Product_reorder_rate
    - Average_pos_cart (average position of product in the cart)
    - Aisle_reorder_rate
    - Department_reorder_rate
- User only feature
    - User_reorder_rate
    - User_unique_products
    - User_total_products
    - User_avg_cart_size
    - User_avg_days_between_orders
    - user_reordered_products_ratio

4. Previous Approach

3.2. **Variable threshold** with feature engineering and modeling

- User-product features:
  - How frequently user order the product
  - How frequently user reorder the product
  - Average position of product in cart ordered by user
  - Number of placed orders since the product was last ordered
  - Max_streak: number of continuous order that contain the product

4. Previous Approach

3.2. **Variable threshold** with feature engineering and modeling

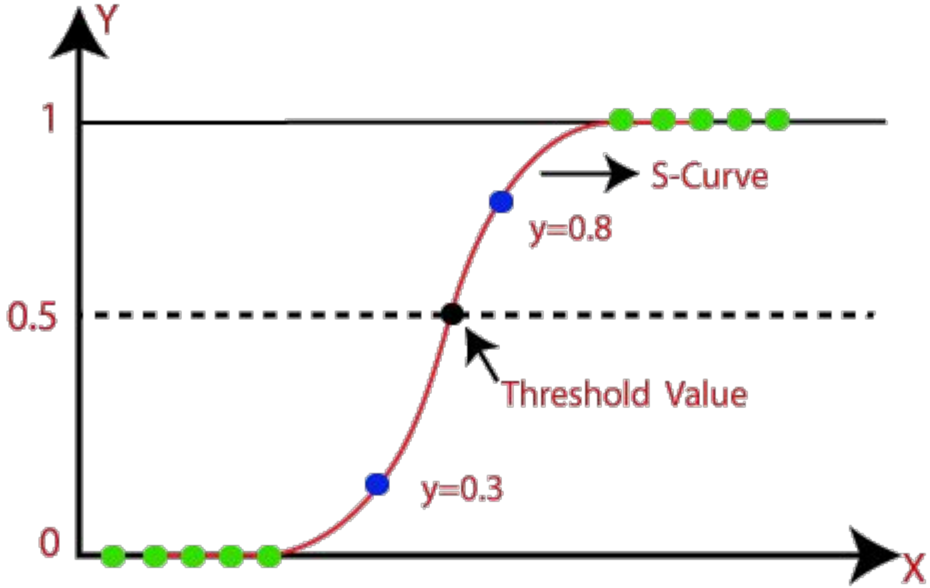
- Misc features
  - Reorder frequency of product in certain hour of day
  - Reorder frequency of product in certain day of week
  - Product reorder frequency with certain gap between 2 order (days)

** Finally, we join all these feature into 1 dataset for model input (vectors of information)

4. Previous Approach

4.1. Fixed threshold

- Pick a fixed threshold for training and predicting items



4. Pre

4.1. Fixed

- Runı

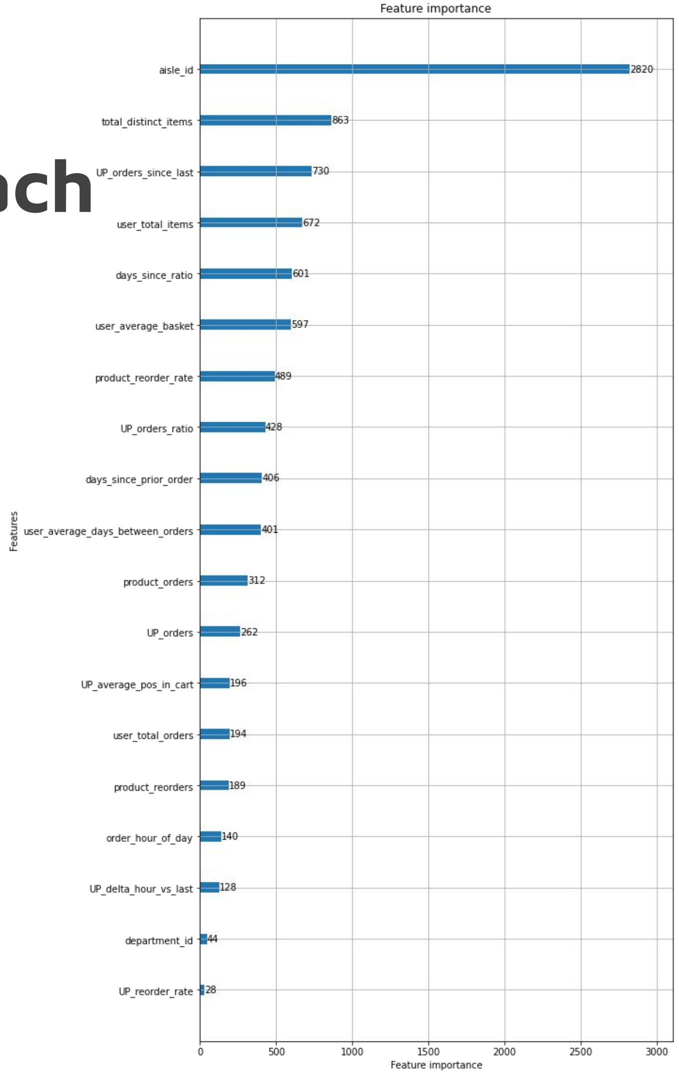
```
Gridsearch for CV:
Threshold :0.1 - Mean F1 : 0.39482614483927203
Threshold :0.11 - Mean F1 : 0.40821274814817676
Threshold :0.12 - Mean F1 : 0.41921852708056134
Threshold :0.13 - Mean F1 : 0.4283893438097734
Threshold :0.14 - Mean F1 : 0.43585812272191155
Threshold :0.15 - Mean F1 : 0.441635249877309
Threshold :0.16 - Mean F1 : 0.44591584719058164
Threshold :0.17 - Mean F1 : 0.44899811372540854
Threshold :0.18 - Mean F1 : 0.45027007949260434
Threshold :0.19 - Mean F1 : 0.4510747802415275
Threshold :0.2 - Mean F1 : 0.4508590730312771
Threshold :0.21 - Mean F1 : 0.4496127850119758
Threshold :0.22 - Mean F1 : 0.4479275352903583
Threshold :0.23 - Mean F1 : 0.44544772835112206
Threshold :0.24 - Mean F1 : 0.4420615597988679
Threshold :0.25 - Mean F1 : 0.43846782715130017
Threshold :0.26 - Mean F1 : 0.4340069375869638
Threshold :0.27 - Mean F1 : 0.4291958342288445
Threshold :0.28 - Mean F1 : 0.42408857866299154
Threshold :0.29 - Mean F1 : 0.4185846766993442
Threshold :0.3 - Mean F1 : 0.41263509780995516
Threshold :0.31 - Mean F1 : 0.406413121136945
```

4. Previous Approach

4.1. Fixed threshold

Score: 0.37628

Private score: 0.37683



4. Previous Approach

4.2. **Variable threshold** with feature engineering and modeling

- Classification flow

User A → will buy Product B → in his next order C → reordered(1/0) ?

User ID	Product ID	Product only features	User only features	User product features	Misc features	Future Order ID	reordered ?

4. Previous Approach

4.2. **Variable threshold** with feature engineering and modeling

- Choosing the thresholds => multiple threshold (0.18, 0.19, 0.2)

```
Gridsearch for CV:
Threshold :0.1 - Mean F1 : 0.39482614483927203
Threshold :0.11 - Mean F1 : 0.40821274814817676
Threshold :0.12 - Mean F1 : 0.41921852708056134
Threshold :0.13 - Mean F1 : 0.4283893438097734
Threshold :0.14 - Mean F1 : 0.43585812272191155
Threshold :0.15 - Mean F1 : 0.441635249877309
Threshold :0.16 - Mean F1 : 0.44591584719058164
Threshold :0.17 - Mean F1 : 0.44899811372540854
Threshold :0.18 - Mean F1 : 0.45027007949260434
Threshold :0.19 - Mean F1 : 0.4510747802415275
Threshold :0.2 - Mean F1 : 0.4508590730312771
Threshold :0.21 - Mean F1 : 0.4496127850119758
Threshold :0.22 - Mean F1 : 0.4479275352903583
Threshold :0.23 - Mean F1 : 0.44544772835112206
Threshold :0.24 - Mean F1 : 0.4420615597988679
Threshold :0.25 - Mean F1 : 0.43846782715130017
Threshold :0.26 - Mean F1 : 0.4340069375869638
Threshold :0.27 - Mean F1 : 0.4291958342288445
Threshold :0.28 - Mean F1 : 0.42408857866299154
Threshold :0.29 - Mean F1 : 0.4185846766993442
Threshold :0.3 - Mean F1 : 0.41263509780995516
Threshold :0.31 - Mean F1 : 0.406413121136945
```

4. Previous Approach

4.2. **Variable threshold** with feature engineering and modeling

- Combine the products result to maximize F1-score => dynamic programming

$$f_{\beta;k} = \sum_{k_1=0}^{k} (1 + \beta^{-2}) k_1 P(s_{1:n} = k_1) s(k, k\beta^{-2} + k_1)$$

4. Previous Approach

4.2. **Variable threshold** with feature engineering and modeling

- Choosing the models:
  - Logistic Regression
  - Decision tree classifier
  - Random forest classifier
  - MLP model
  - XGB
  - Catboost

4. Previous Approach

4.2. **Variable threshold** with feature engineering and modeling

- Some models comparison:

Model	logloss	Final accuracy
Logistic Regression	0.255	0.374
Decision Tree Classifier	0.2509	0.295
Random Forest Classifier	0.2518	0.378
MLP	0.2434	0.383
XGB	0.2434	0.391
Catboost	0.243	0.394

5. Related work with our methods (DL):

- With **dependency (long-term) nature** of dataset, we propose to use sequential deep learning model to solve this problem.
- Long-term prediction can be solved effectively with sequential model and widely applicable.

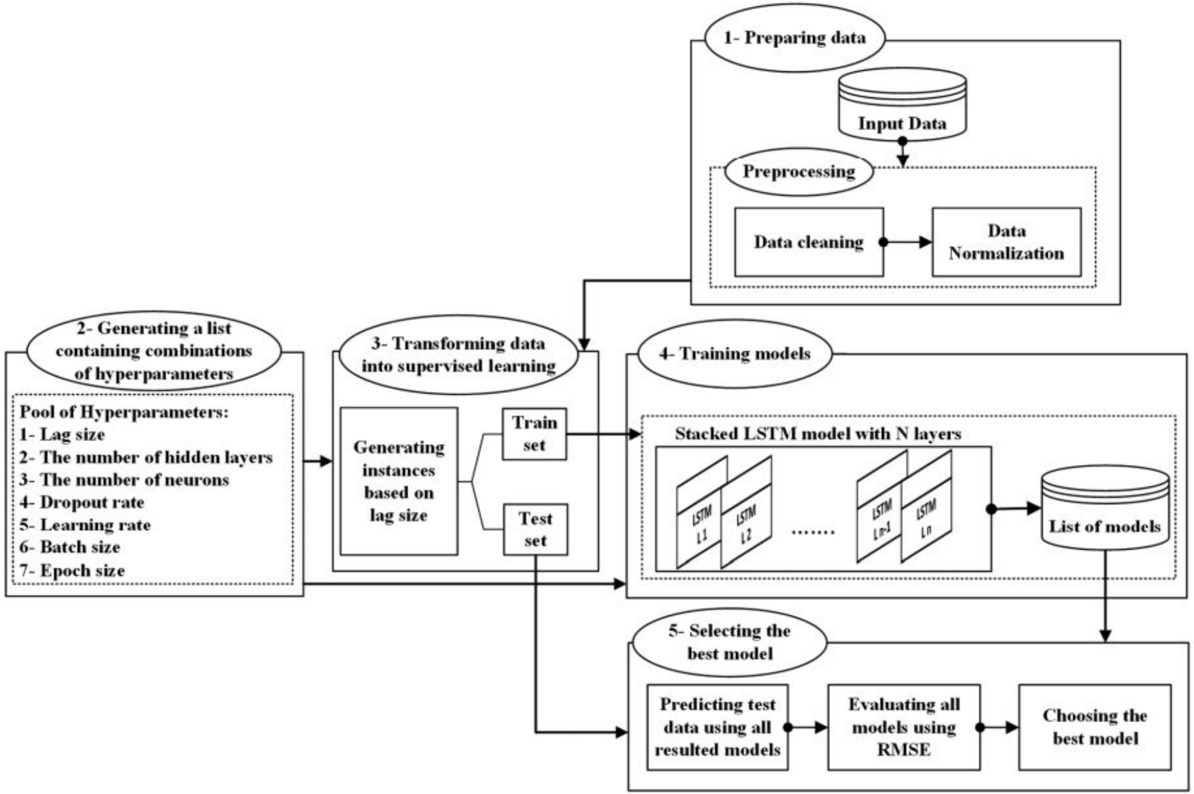
5. Related work with our methods (DL):

An optimized model using LSTM network for demand forecasting

by H Abbasimehr · 2020 · Cited by 128 - Elsevier Computers & Industrial Engineering Volume 143, May 2020, 106435

An optimized model using LSTM network for demand forecasting of a furniture company. Author “propose a method based on a multi-layer LSTM network by using the grid search approach. The proposed method searches for the optimal hyperparameters of the LSTM network.”

5. Related work with our methods (DL):



5. Related work with our methods (DL):

For evaluation of all models we use RMSE and SMAPE performance measures (Martínez et al., 2018). SMAPE is defined by:

$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|\hat{y}_t - y_t|}{\frac{|\hat{y}_t| + |y_t|}{2}} \quad (9)$$

RMSE is expressed a

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)}$$

Table 9
The performance results of all methods on 5 datasets in terms of SMAPE.

Applied methods	D1	D2	D3	D4	D5
ARIMA	0.1257	0.1310	0.1966	0.2535	0.1222
ETS	0.0829	0.1704	**0.1631**	0.2630	0.1212
SVM	0.1259	**0.1169**	0.2182	0.2816	0.1122
ANN	0.1020	0.1299	0.2712	0.3629	0.1138
LSTM	0.1127	0.1441	0.2765	0.3235	0.1208
Proposed method	**0.0730**	0.1295	0.1801	**0.2513**	**0.1085**

5. Related work with our methods (DL):

Evolving Deep CNN-LSTMs for Inventory Time Series Prediction

by N Xue · 2019 · Cited by 23 - 2019 IEEE Congress on Evolutionary Computation (CEC)

In this work, author utilise hybrid deep learning models for inventory forecasting: real world production planning of highly perishable foods and staff scheduling in an environment with highly variable customers demand. . According to the highly nonlinear and non-stationary characteristics of inventory data, the models employ Long Short-Term Memory (LSTM) to capture long temporal dependencies and Convolutional Neural Network (CNN) to learn the local trend features

5. Related work with our methods (DL):

B. Particle Swarm Optimisation

PSO is a population-based meta-heuristic introduced by Eberhart and Kennedy [31] that has been successfully applied to many parameter value optimisation methods. PSO is initialised with a group of random particles (i.e. solutions, population members) and then the algorithm searches for optima by updating generations of particles. In each iteration of PSO, each particle is updated within its given bounds by following two "best" values. The first one is the best solution that the particle and its neighbours have achieved so far (local best). The other one is the best solution found so far by any particle in the whole population (global best). Based on these two best values, a particle updates its velocity and

C. Differential Evolution

DE optimises a problem by maintaining a population of candidate solutions and creating new candidate solutions by combining existing ones according to its simple formulae, and then keeping whichever candidate solution has the best score or fitness on the optimisation problem at hand. Usually, DE generates new parameter vectors by adding the weighted difference between two other vectors to a third vector in the population. If the resulting vector yields a lower fitness value than a predetermined member in the population, the newly generated vector replaces the predetermined vector that was compared in the next generation; otherwise, the old one is retained [33]. This basic principle has been extended to many variants of DE.

A DE algorithm can be marked as DE/x/y/z, where x denotes how the differential mutation base is chosen, y denotes the number of vector differences added to the base vector and

Formula

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

MAE = mean absolute error

y_i = prediction

x_i = true value

n = total number of data points

TABLE VI: Experiments results

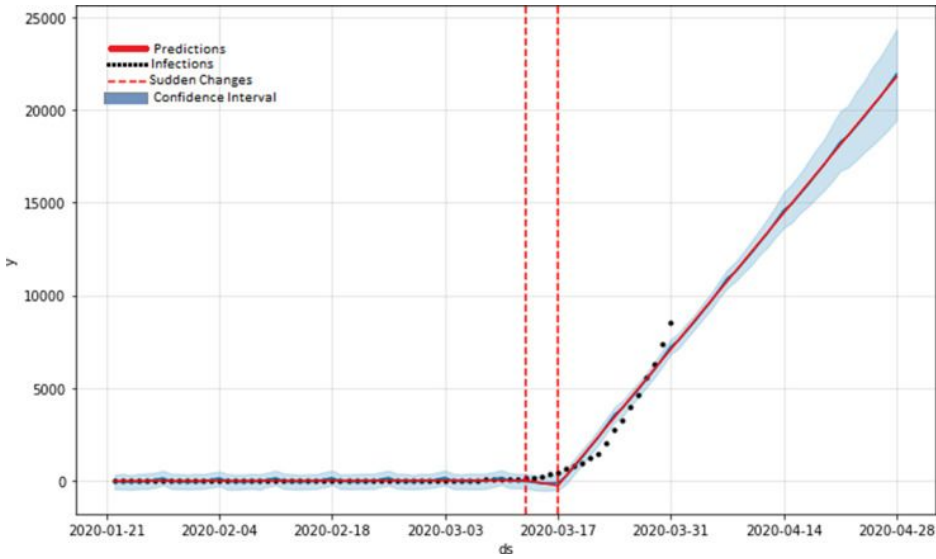
	SARIMA	DE/best/1/bin	PSO	DE/best/1/exp
I1	7.30	5.28	5.30	**4.43**
I2	9.82	9.41	9.45	**8.44**
I3	6.38	4.63	4.55	**4.32**
I4	6.11	5.32	5.22	**4.98**
I5	6.07	4.57	4.62	**4.40**
I6	4.88	3.10	3.15	**2.81**
I7	4.70	3.25	3.48	**3.20**
I8	43.14	36.57	36.33	**35.66**
I9	25.12	19.76	19.70	**19.54**
I10	18.88	**14.55**	14.82	14.69
AVG	13.24	10.64	10.66	**10.25**

5. Related work with our methods (DL):

Time series forecasting of COVID-19 transmission in Canada using LSTM networks

by VKR Chimmula · 2020 · Cited by 533

In this paper we presented the Long short-term memory (LSTM) networks, a deep learning approach to forecast the future COVID-19 cases. To solve spatio-temporal components simultaneously: addressed the above problem by modifying the internal connections. In our modified LSTM cells, We have established the alternative connections between the input and output cells.



6. Detailed design of solution

6.1. **High-level idea**

- Which products are reordered by an user in the next order?
  - The probability that the user would order a specific product.
  - The probability that product belong to a specific aisle.
  - The probability that product belong to a specific department.
  - The size of the next order. } user-order size

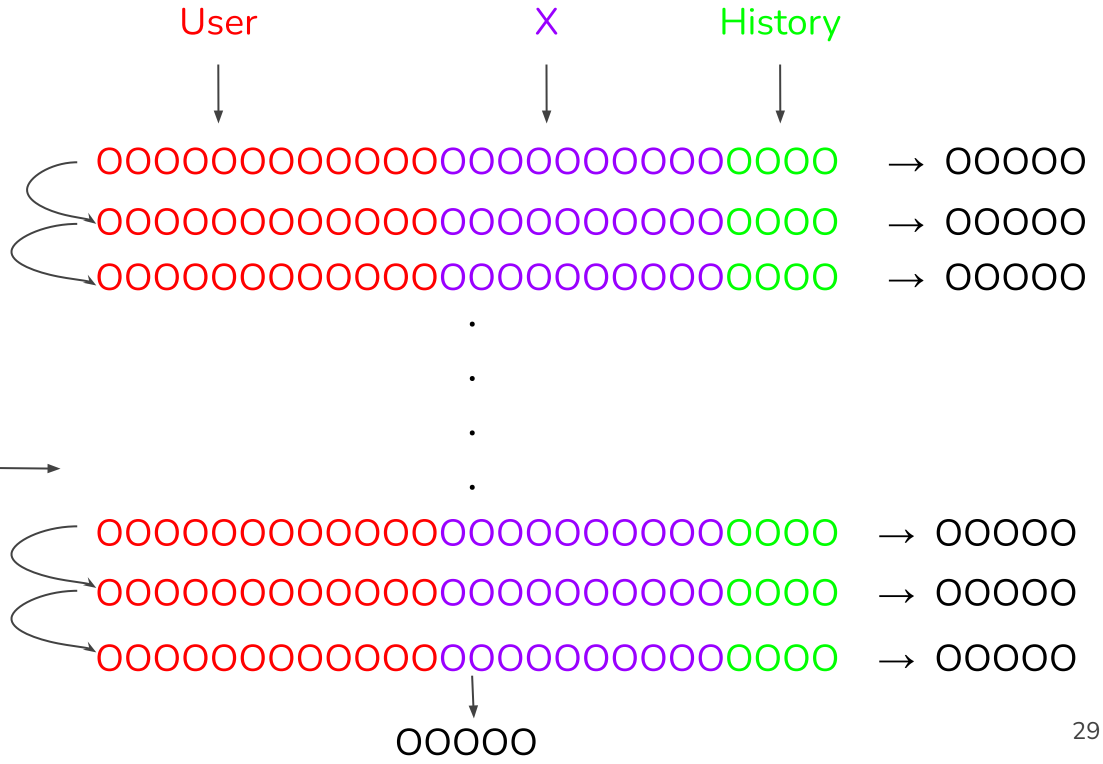
user-product

6. Detailed design of solution

6.2. **Sequential model.**

- User - X
  - Product
  - Aisle
  - Department
  - Order size
- Models:
  - RNN
  - LSTM
  - CNN
  - Transformer

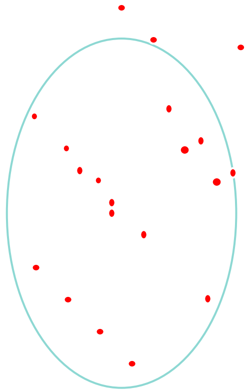
Up to 100 orders



6. Detailed design of solution

6.3. **Mixture model.**

- Anomaly generation (prediction).
- Bernoulli .
- Gaussian.



6. Detailed design of solution

6.4. **Input sequence.**

User-Product	User-Aisle	User-Department	User-Order size
UserID ProductID AisleID DepartmentID Ordered in history Index in order history Order dow in history Order hour in history Days since prior order Order size in history Reorder size in history Order number in history Number of orders Product Name Product Name length Eval set Label	UserID AisleID DepartmentID Ordered in history Index in order history Order dow in history Days since prior order Order size in history Order number in history Number of products from aisle Number of orders	UserID DepartmentID Ordered in history Index in order history Order dow in history Days since prior order Order size in history Order number in history Number of products from department Number of orders	UserID Ordered in history Index in order history Order dow in history Days since prior order Order size in history Reorder size in history Order number in history Number of orders

6. Detailed design of solution

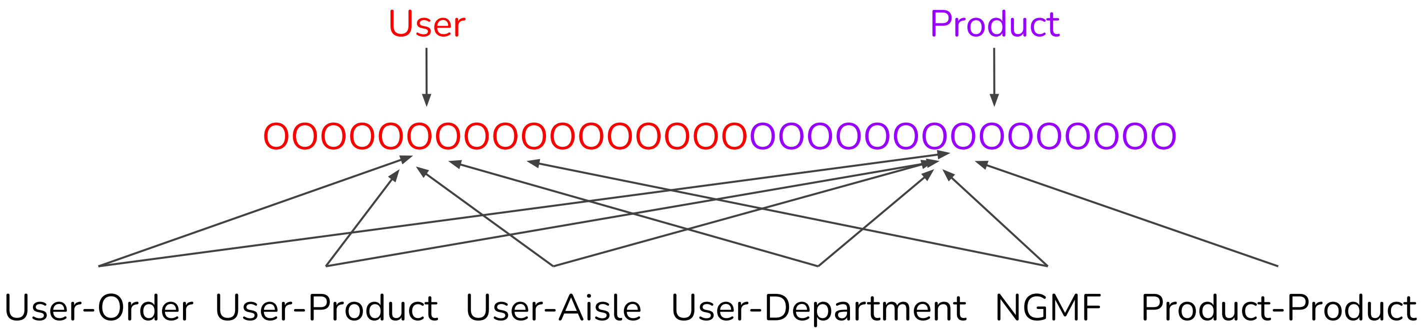
6.5. **Additional models for representation learning.**

- User-Product based on order counts (Non-negative matrix factorization).
- Product-Product model based on whether they are both in an order.

6. Detailed design of solution

6.6. **Output of representative learning models.**

- Vectors contains information of users, products
- Fed into a machine learning algorithm to conclude reorder probability.



6. Detailed design of solution

6.7. **Final-layer models.**

- 0.2*Neural network + 0.8*LightGBM/Catboost.
- F1 maximization.

6. Detailed design of solution

6.8. **Training process.**

- Model selections.
  - RNN (X)
  - LSTM (O).
  - CNN_LSTM (O).
  - Transformer (Pending)
- Dataset.
  - Train on 500k orders (30k users)

6. Detailed design of solution

6.9. **Result.**

On 500k orders,

- Previous approach with 500k dataset.
  - Average F1: 0.2978-0.3239
- Proposed approach.
  - Average F1: LSTM-0.3112; CNN+LSTM-0.3250 .

6. Detailed design of solution

6.10. **Explanation.**

Why the proposed method did not significantly outperform the conventional method.

- Overfitting of deep learning models (X, as relatively similarities in loss of validation and training set).
- Conventional methods have effectively learnt the dataset for predictions (previous solutions were optimized).
- Tabular dataset is still best solved by conventional machine learning methods (adapted widely)

6. Detailed design of solution

6.11. **Pros and cons.**

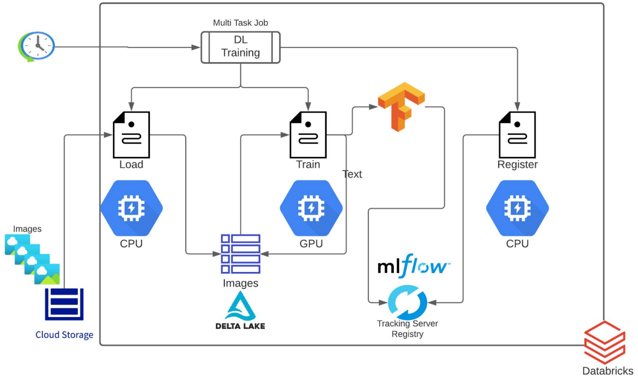
- Pros:
- Fully filter out labour works of feature engineering process.
- Widely applicable to multiple relevant problems.
- Cons:
- Require large computation resources, time to train models.

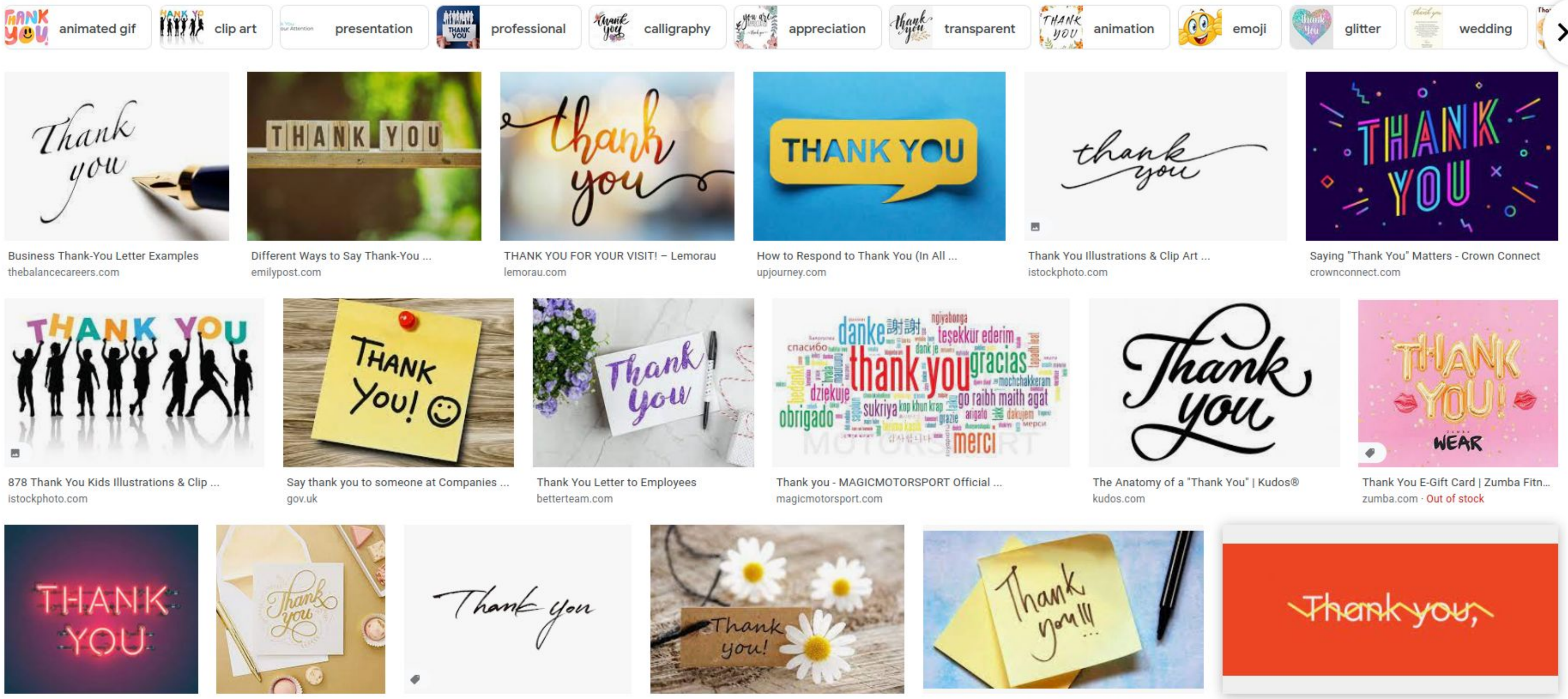
7. Extensibility

- Build a code-free system for all SME supermarket and ecommerce.
- Apply to other sets of products (fashions, accessories, electronic devices).

8. Future Development

- Train with all datasets
- Using transformer architecture (benefits of parallel and attention mechanisms).
- Produce end-to-end reusable pipeline.




animated gif
clip art
presentation
professional
calligraphy
appreciation
transparent
animation
emoji
glitter
wedding
Business Thank-You Letter Examples
thebalancecareers.com
Different Ways to Say Thank-You ...
emilypost.com
THANK YOU FOR YOUR VISIT! – Lemorau
lemorau.com
How to Respond to Thank You (In All ...
upjourney.com
Thank You Illustrations & Clip Art ...
istockphoto.com
Saying "Thank You" Matters - Crown Connect
crownconnect.com
878 Thank You Kids Illustrations & Clip ...
istockphoto.com
Say thank you to someone at Companies ...
gov.uk
Thank You Letter to Employees
betterteam.com
Thank you - MAGICMOTORSPORT Official ...
magicmotorsport.com
The Anatomy of a "Thank You" | Kudos®
kudos.com
Thank You E-Gift Card | Zumba Fitn...
zumba.com · Out of stock