Product Matching

...

Surbhi Prasad

Motivation

- E-commerce websites have products that are sold by multiple sellers
- Each seller can upload different images and descriptions of the same product.
- These images and descriptions can be highly diverse.
- Identifying these duplicates is essential to improve user experience as well as to optimize resources.

Dataset

- We use a combination of 2 sources to obtain this data:
  - Kaggle Shopee competition data
    - ~ 32500 observations
  - Data scraped from ebay
    - ~ 6000 observations
- Each observation has
  - Each observation has the following attributes:
    - Image (1000 x 1000 RGB image)
    - Product description
    - Label Group (unique product ID)
    - Posting ID (unique observation ID)
    - Image pHash

Dataset

posting_id	image	image_phash	title	label_group
train_1000516097	b85fa528f92075181...	bac9c1363ac9e836	RINNAI Kompor Gas...	2155806999
train_1000574656	c3db137ba71aad476...	f481c39ecb789172	TAS IMPORT FASHIO...	1653532917
train_1000653457	222856ba7c2b88e81...	eed1b58ab545950a	topi jaring full ...	1503972976
train_1000697977	bfa96a6e065417a73...	b8388c38e3c74e6d	SHAMPO PANTENE 75...	3291104203
train_1000782282	d72a8087c4ff746a4...	bf8699938c962595	MOMMYCLOPEDIA 567...	1130629555
train_1000804730	83bbb79e8e2c6318a...	d6aec151632f2bb0	Karet Rambut Wani...	994676122
train_1000931038	0811ace2de57cdc70...	d0a55a3f655a25ca	Bokoma Massage / ...	378982106
train_1000976599	aa41f0a56be8b55ae...	ff1515c5315a2be0	HOLISTICARE ESTER...	3902160400
train_1000990874	b3b5f015ce5753b0e...	ebea8a5b8bc2254c	Soft Case Tpu Mot...	1095455866
train_1001207739	23100e23b1a96cf8f...	b4d6c2f8c1229bcb	READY STOK BANYAK...	2625326568

only showing top 10 rows

Dataset

The images and descriptions shown here all belong to the same product

title
GROSIR LVN COLLAGEN / COLAGEN STROBERI PEMUTIH...
DISTRIBUTOR LVN COLLAGEN STROBERI / COLAGEN 1 ...
TERMURAH LVN COLLAGEN STROBERI 1 BOX 10 SACHET
LVN COLLAGEN - ORIGINAL TERMURAH - LVN STROBER...
LV.N COLLAGEN 1 BOX "BELI ECER HARGA GROSIR" O...
LVN STRAWBERRY (isi 10 Bgks)
[ORIGINAL] LVN COLLAGEN / POMEGLOW COLLAGEN 1 ...
LVN Collagen / Stroberi eco 1box (10 sachet)
LVN COLLAGEN / STROBERI
LVN COLLAGEN / STROBERI
Lvn Collagen / Lvn Strawberry Original BPOM RI...
LVN COLLAGEN LVN STROBERI ORIGINAL 100% 1BOX I...

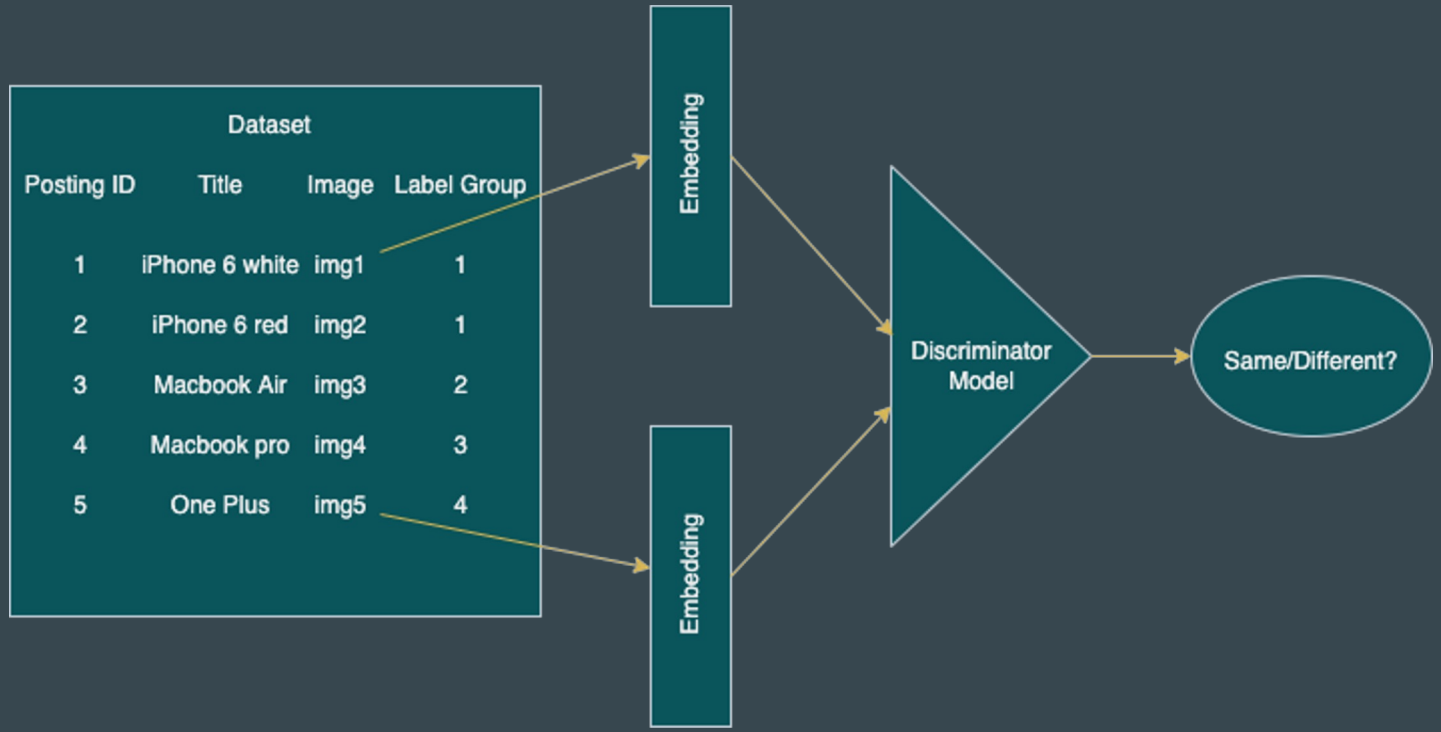
Analytic Goals

Our aim is to identify duplicated products among ~33k products sold by multiple sellers on Shopee. As the dataset has more than ~11k distinct class labels of products,multiclass classification model wasn't feasible , our solution is based on finding similar products(deduplication) by different approach of vector embeddings of products.

Modelling approach

- Make meaningful vector representations of each product. This is achieved by using
  a. Embeddings of title text
     i. TFIDF
     ii. RoBERTa (Multilingual)
  b. Embeddings of image
     i. ResNet18
  c. Joint Embeddings (text + image)

Modelling Flowchart



Supervised Models Approach

- Pair Classification:
  - Make pairs of observations
    - Arrange some of the pairs to be from matching products (positive pairs)
    - Remaining pairs from non-matching products (negative pairs)
    - 1:1 Negative Sample is selected for maintaining class balance.
  - Train a binary classification model to take a pair and predict whether both observations belong to the same unique product or not.
  - Cosine Similarity (used in Logistic) is calculated for all pairs and set as feature for model to find right distance.
  - The models can be:
    - Logistic regression
    - Neural Network
  - For any new observation, compare it with K nearest observations and classify.

Modelling approach

- Using these representations, we can approach the problem in 2 ways:
  - Find Nearest Neighbors and choose majority class
  - Classify pairs of observations into ‘same product’ vs ‘different products’
- Nearest neighbors approach:
  - Given an observation, find its (approximate) K nearest-neighbors using Locality Sensitive Hashing techniques:
    i. BucketRandomProjection LSH
    ii. MinHash LSH
  - Out of these K neighbors, choose the most frequent product class.

Experiments and results

1. Neural Networks:
    a. Architecture:
        i. Layers:
            1. Linear
            2. Batch Normalization
        ii. Activation Functions:
            1. Sigmoid
            2. Rectified Linear Unit activations (ReLU)
            3. Leaky ReLU
        iii. Binary Cross-Entropy loss function (with logits)

    b. Hyperparameters:
        i. Learning rate: [0.1, 0.001, 0.0001]
        ii. Batch Size: [8, 16]
        iii. Weight decay: [0, 10^{-4}, 10^{-2}]
        iv. Early Stopping (limit of 4 non-improvement epochs)

Experiments and results

c. Results:
   a. Best Hyperparameters found:
      i. Learning Rate: 10^{-5}
      ii. Batch size 16
      iii. Weight Decay 0.0001
   b. Best performance:

Dataset	Loss	Accuracy
Training	0.60	0.57
Validation	0.61	0.66

Experiments and results

2. Logistic Regression:
   a. Architecture:
      i. Cluster:
         1. Group-3 GPU- 64GB, 16 Cores, DBR 10.3 ML,Spark 3.21
         2. Since embeddings were computationally expensive to run(~800 features)

      ii. Algorithms for word Embeddings:
         1. Spark NLP - version: 3.4.2
         2. BERT - Roberta- Multilingual Embeddings
         3. Cosine Similarity

   b. Parameters:
      i. regParam = 0.01
      ii. maxIter = 10

Experiments and results

c. Results:

Dataset	Accuracy
Training	0.659
Validation	0.657

Experiments and results

3. Approximate KNN (Locality Sensitive Hashing) :
    a. Architecture:
        i. Cluster:
            1. Group-3 GPU- 64GB, 16 Cores, DBR 10.3 ML,Spark 3.21
    b. Algorithms Used:
        i. Resnet18 Embeddings
        ii. TFIDF for word features
        iii. Cosine Similarity
        iv. BucketRandomProjectionLSH
        v. MinHash LSH
    c. Parameters:
        i. K = 5

Experiments and results

c. Results:

Methods	Dataset	Accuracy
TFIDF with cosine similarity Train	Training	0.52
	Validation	0.44
BucketRandomProjectionLSH on Resnet18 pertained image embedding	Training	0.38
	Validation	0.36
BucketRandomProjectionLSH on Resnet18 pertained image embedding + tfidf features Train	Training	0.83
	Validation	0.74
MinHash LSH on product description	Training	0.62
	Validation	0.56

Performance Comparisons

- Runtime on GPU-enabled cluster:
  - Neural Networks
    - Hyperparameter Tuning: ~ 108000 seconds
    - Training : ~ 6500 seconds
  - Logistic Regression
    - Cosine Similarity: ~ 1200 seconds
    - Fitting Logistic Model: ~ 7200 seconds
  - Approximate KNN
    - BucketRandomLSH model training: ~80 seconds
    - BucketRandomLSH approxSimilarity - 600 seconds
    - TFIDF Cosine Similarity: ~480 seconds
    - MinHashLSH model training~ 240 seconds

Lessons Learned

- Training of word and image embeddings together showed better results than individual training.
- Caching and repartitioning helps boost the performance.
- Spark helps improve model training on distributed data.
- Using GPU helps to train models faster.
- All approaches that require cartesian product of the data are very expensive.

Conclusion

Training complex data like images that has large quantity is a huge task for a normal Machine Learning model. In such scenarios, pyspark can help boost the performance by distributing data on different nodes. In our model, usage of pyspark helped boost the performance for image classification. Moreover, by using image and text embedding together, we got our best performance accuracy of 0.73 on validation set. Hence, for problems like deduplication, it is always a good idea to get the embeddings of images and texts separately and then use them together in another model to get the best results.

References

- ResNet model for image embeddings [Link]
- Locality Sensitive Hashing on spark [Link]
- k-Nearest Neighbors algorithm [Link]
- Text Classification with Spark NLP [Link]
- Siamese Networks for face similarity [Link]
- RoBERTa Model [Link]