CS230

Sales Prediction Based On Product Titles and Images with Deep Learning Approaches

Haishan Gao
hsgao@stanford.edu

Zhaoqiang Bai
baizq871@stanford.edu

Jingqian Li
jqli0201@stanford.edu

Abstract

Over the past few years, online shopping gradually became the mainstream shopping method. More and more local retailers chose to start their businesses on e-commerce platforms. However, few can survive due to the competitive pressure from the big companies and the entry barriers. The motivation of this project is to identify product listing strategies, primarily visual and textual presentation, that can help retailers to raise their product sales. To achieve that, we build a neural network architecture to predict product sales from text, image and other product listing features that retailers can control. In particular, we use pre-trained VGG16 model for image data and TF-IDF model for text data in our architecture. The accuracy of the model is 73.91%, far above the accuracy of human raters, which is 55.33%.

1 Introduction

E-commerce platforms emerged in the recent couple of decades as a mainstream channel for vendors to sell their products. Due to the non-tactile nature of the online products, to attract customers and promote their sales, e-commerce vendors rely more on an alternative group of presented visual and textual information such as product images and titles. From a business intelligence perspective, it is of practical importance to find the best strategy for both visual and textual presentation. In this project, we designed and trained a deep learning model to predict sales using product titles and images, which allows us to identify the best strategy of product listings that lead to sales success. The motivation of solving this problem using deep learning is: first, unlike cognitive tasks where humans can excel easily, the task of identifying the hidden patterns in high-sale product texts and images is not straightforward to human eyes; second, machines have the potential to learn the trend behind the large amount of e-commerce data that is too large for humans to process.

2 Related Work

Traditional sales prediction models use time series analysis to forecast sales for stores. With the rapid development and deployment of machine learning techniques, recent studies start to use deep learning models to predict sales and have shown very high accuracy [1]. In our project we mainly consider the influence of two features on sales: image and text. Previous research has shown that textual

CS230: Deep Learning, Fall 2021, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

information of a product, such as descriptions, can have significant impact on sales: seasonal, polite, authoritative and informative descriptions can lead to higher sales [2]. In "Creating the best first impression: Designing online product photos to increase sales", the authors investigate the effect of product photos on sales of clothing products and they find that the influence is differentiated between women's and men's clothing and men's clothing is more susceptible [3].

3 Dataset

The dataset is from Kaggle "Sales of Summer Clothes in E-commerce Wish" [4] and it consists of 1.5k products with 43 columns including listings, ratings and sales performance information. The two primary features from the dataset we will study are product image and product title. Aside from image and text, we select some of the rest features to form metadata features.

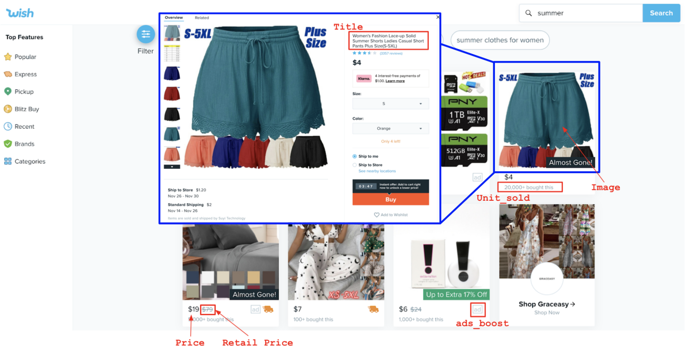


Figure 1: *Wish* Product Listing with Keyword "Summer"

3.1 Features

We read images from "product_pricture_url" and the dimension is $200 * 200 * 3$. Based on our survey, the resolution can be at most reduced to $128 * 128 * 3$ to ensure the visual details of the products can be observed clearly. For text data, we remove stop words, remove non-alphabetic terms, apply lowercase and use WordNetLemmatizer to lemmatize the terms. Since the project goal is to recommend vendor product listing strategies, other than images and titles, we keep only the features vendors can personalize such as price, shipping fees, ads boost etc., and remove the features vendors cannot customize such as user rating, origin country etc. "discount_ratio" is added as a feature by $Price_{display}/Price_{original}$ to investigate the effectiveness of attracting customers by discounts. We also convert the color strings in "product_color" to their corresponding RGB values. We then use Random Forest to rank the feature importance:

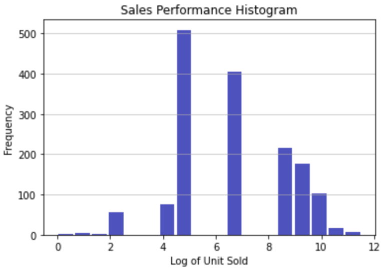
Table 1: Feature Importance excluding Image and Text Data Using Random Forest

Feature	Importance	Feature	Importance
countries_shipped_to	0.20	badges_count	0.03
discount_ratio	0.16	uses_ad_boosts	0.03
log(price)	0.16	urgent	0.02
log(retail_price)	0.16	badge_fast_shipping	0.01
shipping_option_price	0.06	badge_local_product	0.01
product_color	0.06	inventory_total	0.00
badge_product_quality	0.03	shipping_is_express	0.00

3.2 Labels

The sales information "unit_sold" in the dataset is obtained by scraping displayed sentences "N bought this" on the product listing page. We label the dataset with binary values "high sales" and "low sales" based on the value of "unit_sold". We define the the sales prediction problem as binary classification, for the purpose of selecting the optimal architecture of image and text models.

$$y = \begin{cases} 1, & \log(\text{unit\_sold}) \geq 6 \\ 0, & \text{otherwise} \end{cases}$$



4 Models and Results

4.1 Baseline Models

In our features baseline model, we use the 16 selected features only as the input and Random Forest as the prediction model. The average accuracy is about 0.6540. To study the relative impacts of images and texts, we set up image and text baseline models, respectively. The baseline image model has a simple CNN structure of three layers of [CONV-16, DROPOUT-0.3, MAXPOOLING] followed by a FLATTEN layer, then a dense layer with RELU activation function. The result has relatively low bias with 0.84 train accuracy but high variance with 0.61 dev accuracy. The baseline text model uses TF-IDF to encode text data and the network architecture is CONV-24, DROPOUT, MAXPOOLING followed by a FLATTEN layer, then a dense layer with sigmoid as the final output for prediction class. We observe 0.84 train accuracy and 0.67 dev accuracy.

4.2 Image Models

Pre-trained Network We then move on to use VGG-16 network [5] in Keras [6], which is a pre-trained version on more than ten million images from the ImageNet database [7]. We apply transfer learning and keep all the weights of the convolutional blocks frozen. Only the parameters in the fully-connected blocks were trained on our own dataset. The original VGG-16 output layer of 1000 classes is modified to just 2 classes for our binary classification task.

In comparison with the image baseline model, the VGG-16 network improves the dev accuracy from 0.6022 to 0.6497. The exact Bayes error of predicting sales from product images is unknown and intuitively it is difficult for a person to judge good/bad sales merely from product images, so we regard 0.65 accuracy acceptable. Noted that the VGG16 model has lower training accuracy than the baseline CNN. We attribute this to the fact that the VGG networks employ regularization techniques like dropouts, which randomly ignores some number of layer outputs during training. On the contrary, all features are used at test time and the model is more robust and can lead to higher dev accuracy.

Data Augmentation To mitigate the problems that may arise from limited dataset size (around 1.5k), We further apply the image augmentation technique to artificially expand the size of our training dataset. 50,000 images are generated through manipulations such as shifts, flips, crops, rotation, and zooms. We then use the the augmented dataset to train and validate our models. However, the training and validation accuracy is not improved.

4.3 Text Models

Embedding Methods We start by the Doc2Vec model to represent each title as a vector and we observed that the inclusion of features will improve model accuracy significantly. Using only title information as input has limited performance. Therefore, we decided to include selected features when training the models. The second model is using TF-IDF as the embedding of text sentences. This modification increased our validation accuracy using the same model architecture. Finally, we use Sentence Transformers [8] to encode product tile, which presented worse performance than using TF-IDF models.

Data Augmentation To achieve better performance with limited data, we use Easy Data Augmentation [9] to augment the number of labelled data. This paper introduced four ways to augment data set.

First is to replace synonym with randomly choose words. Second is to insert random synonym of a random word in each sentence. Third is to randomly swap two words in the sentence and repeat n times. The last method is to randomly remove a word with probability p. Using these four methods we successfully increase our training data from 1,537 to 15,731, approximately by ten times. However, increasing the amount of training data doesn't improve the performance of either a logistic regression binary classifier, or a neural network. The best model performance with augmented training data still stays around 72% dev accuracy.

4.4 Ensemble Methods

In addition to product titles, we use product tags to help the training process. However, adding the information of tags doesn't improve model performance and slows down the training process since the input space is larger. We also explore the idea of combing weak learners with a better model. We first use TF-IDF model to encode titles, then pass the embeddings into a logistic regression classifier to make the prediction. The output probabilities are concatenated later with other features as the next level model input. However, using a similar model architecture as before, this change makes the model performance a bit worse than just using TF-IDF embedding with features as input. We only get around 0.7 dev accuracy. One of the possibilities is we don't have enough data to build independent components for any ensemble methods.

4.5 Combined Model

After our exploration on each modality of the model, we integrate all the features with our best tries into one combined model. For image data, we apply transfer learning using a pre-trained VGG-16 model as we discussed in the "Image Models" section. Following the fully-connected FLATTEN layer are two [RELU, BATCHNORM, DROPOUT] blocks that reduce the image data into a 128 by 1 vector. For text data, we decide to use TF-IDF model for text representation by comparing the performances of multiple word representation models. The title of each example is transformed into a 1169 by 1 vector through TF-IDF model and passed through two [CONV1D, MAXPOOLING, DROUPOUT] blocks followed by a FLATTEN layer. The output is then passed to a dense layer with RELU activation function that outputs a 128 by 1 vector. The model then concatenates the image and text vectors with the metadata feature vector, passes the concatenated vector through three layers of [RELU, BATCHNORM] and finally a dense layer with sigmoid for the binary classification.

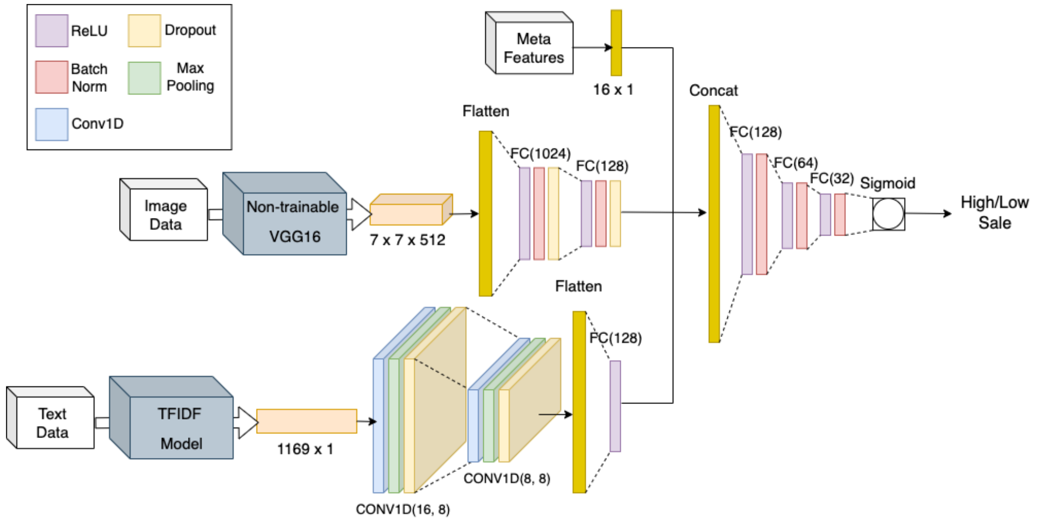


Figure 2: Neural Network Architecture of the Combined Model

Hyperparameter Tuning We tune the following hyperparameters of the model based on the validation accuracy after 200 epochs. The Adam optimizer is used in our training and we find that the default parameters performed the best: $lr = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-7}$. The resolution of the input image to the pre-trained VGG-16 model is $(224 * 224 * 3)$. The dropout rates are all set to be 0.2. The number of fully connected layers that train the concatenated vector is 3. We use $batch\_size = 16$ to train the model to ensure the maximized GPU efficiency and the convergence of the training.

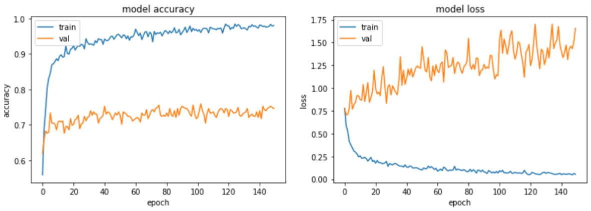


Figure 3: Diagrams of Combined Model Accuracy and Loss

Regularization and Result The train accuracy is 97.83% and the validation accuracy is 73.91%. We observe that the model is overfitting the training set as the validation loss does not decrease and the validation accuracy does not increase after certain epochs. We have explored various regularization approaches including L2 norm on weights, data augmentation and higher dropout rates. Although we can reduce the variance, i.e. the gap between train and validation accuracy, we can't achieve a better validation accuracy. Our hypothesis is that the current accuracy, 73.91% is a bottleneck for our defined problem.

Table 2: Comparison of neural network model performance on image and text dataset

Models	Train accuracy	Dev accuracy
(Features; baseline) Random Forest	N/A	0.6540
(Image; baseline) Simple CNN	0.8338	0.6142
(Image) VGG-16	0.6022	0.6497
(Text; baseline) TF-IDF	0.8372	0.6701
(Text + Features) Doc2vec	0.6811	0.6396
(Text + Features) TF-IDF	0.7693	0.7259
(Text + Features) Sentence Transformers	0.6716	0.6333
(Combined) VGG-16, TF-IDF	0.9783	0.7391

5 Result Analysis

Human Rating We randomly select around 200 samples and perform human grading. Our grading guideline is to only use product description and pictures, pick around 60% products as things we would like to buy and filter out the other 40%. Surprisingly the accuracy of human rating is 55.33%, which means our models has outperformed human raters.

Text Feature Analysis We observe that models with text input have better performance in general, and TF-IDF embedding works better than contextual embedding. Therefore, we believe product titles have effect on the product sales. By comparing common word pairs between high sales product and low sales product, we find the titles of high-sale products have more phrases that describe the characteristics of the product or are associated with women apparels; while low sales product tend to have more terms related to sports or functionalities. Please see Appendix Figure 4 for more information. In terms of title length, high sales product has slightly longer titles in general. But no evidence shows title length impacts the sales directly.

Image Feature Analysis The models with only image as the input don't perform as well as those with the text input. There might be two reasons for this: 1) The VGG model is pre-trained on the ImageNet dataset, which is designed for significantly different tasks (e.g., animal classification) than commercial products classification and analysis; 2) The correlation between product sales and images is weak.

6 Conclusion

From the experiment results, we conclude that text input is more important than image input or other features. We observe that numerical statistic embedding method TF-IDF performs better than contextual embedding. Such finding indicates that including specific keywords in the product title can increase the product sales. This aligns with the fact that most e-commerce platforms retrieve products mainly based on keyword matching. The Bayes error rate remains unknown for our project due to the inherent difficulty of this task. Thus, why data augmentation has limited improvement on our project might not attribute to the robustness of our model, but to the fact it's reaching the Bayes error rate. There are several directions we think can be explored more. First of all, the size of the dataset is small for training deep learning models. One should try collecting more data for training. Second, one can use an image network model that is pre-trained with commercial product images instead of ImageNet.

7 Contribution

Haishan is responsible for pre-processing the data, analyzing the text and image data, evaluating the feature importance, training the baseline models for image and meta features, and training the combined deep learning model.

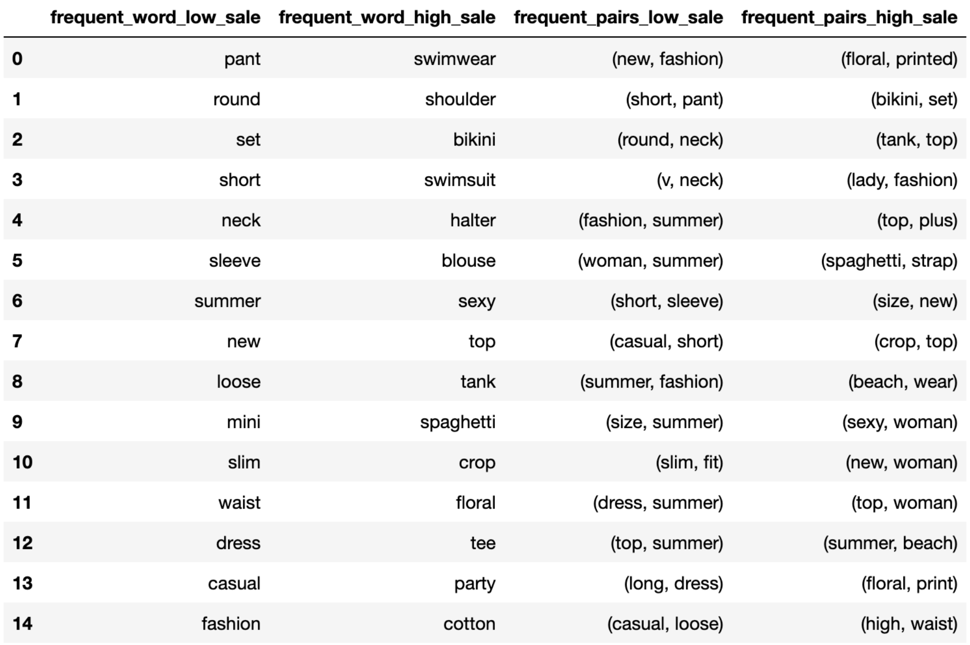
Zhaoqiang is responsible for setting up and training the image-only models.

Jingqian is responsible for pre-processing text data, analyzing different text embedding methods, training different models for text only, and text with meta features only, also experimenting text data augmentation and ensemble methods of modeling for text with meta features.

All authors contributes to conception of project topics, analysis of the results and writing up the reports.

8 Appendix

All codes can be found in https://github.com/jqli0201/etsy-analysis.



	frequent_word_low_sale	frequent_word_high_sale	frequent_pairs_low_sale	frequent_pairs_high_sale
0	pant	swimwear	(new, fashion)	(floral, printed)
1	round	shoulder	(short, pant)	(bikini, set)
2	set	bikini	(round, neck)	(tank, top)
3	short	swimsuit	(v, neck)	(lady, fashion)
4	neck	halter	(fashion, summer)	(top, plus)
5	sleeve	blouse	(woman, summer)	(spaghetti, strap)
6	summer	sexy	(short, sleeve)	(size, new)
7	new	top	(casual, short)	(crop, top)
8	loose	tank	(summer, fashion)	(beach, wear)
9	mini	spaghetti	(size, summer)	(sexy, woman)
10	slim	crop	(slim, fit)	(new, woman)
11	waist	floral	(dress, summer)	(top, woman)
12	dress	tee	(top, summer)	(summer, beach)
13	casual	party	(long, dress)	(floral, print)
14	fashion	cotton	(casual, loose)	(high, waist)

Figure 4: Ranking of Text Frequency

References

[1] Kaneko, Y., Yada, K. (2016). A Deep Learning Approach for the Prediction of Retail Store Sales. 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), 531-537.

[2] Pryzant, R., Chung, Y., Jurafsky, D. (2017). Predicting Sales from the Language of Product Descriptions. eCOM@SIGIR.

[3] Xia, H., Pan, X., Zhou, Y., Zhang, Z. (2020). Creating the best first impression: Designing online product photos to increase sales. Decision Support Systems, Volume 131.

[4] Mabilama, J. M.. (2021) Sales of summer clothes in E-commerce Wish, Version 4. Retrieved November 1, 2021 from https://www.kaggle.com/jmmvutu/summer-products-and-sales-in-ecommerce-wish.

[5] Simonyan, K., Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR, abs/1409.1556.

[6] Keras. https://keras.io//

[7] ImageNet. http://www.image-net.org

[8] Sentence Transformers. https://huggingface.co/sentence-transformers

[9] Wei, J., Zou, K. (2019). EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. ArXiv, abs/1901.11196.