September 2022

# Systems and Methods for Machine Learning Brand Identification Models

Roy Palacios

Pallavi Satyan

# SYSTEMS AND METHODS FOR MACHINE LEARNING BRAND IDENTIFICATION MODELS

## Introduction

Traditionally, machine learning processes for identifying brands have low precision rates and require handcrafted formulae which need scratch retraining upon replacement or edit. As such, a traditional machine learning process for brand identification may be able to identify and link users with brands but result in imprecise matching with a high computational cost. There remains a need for a high precision model that only requires retraining sparsely.

## Summary

Generally, the present disclosure is directed to a high precision machine learning model trained to identify brands. In particular, in some implementations, the systems and methods of the present disclosure can include or otherwise leverage one or more machine-learned models to create signals for top product categories at the brand/merchant level and brand signals from custom Brand Ownership Service based on external digital brand signals such as trademark data from the USPTO, the EUIPO, and/or other trademark registries.

Computer-implemented systems and methods are provided for brand identification, including the extension of a brand identification pipeline to support machine learning and inference for Brand Entity candidate proposals.  By way of example, the algorithm may take a merchant and identify their most popular consumer good category and conduct an aggregation ranking across all products provided. The system may also identify competing merchants, acting under different brands, and rank the goods against competing goods from different brands.

The system takes inputs including global trademark data, internally created signals for top products and signals from an internal Brand Ownership Service. The outputs include a refactored

and simplified Machine Learning training pipeline, a top products signal, an internal Brand Ownership Service signals, and generated truth data for internal training purposes. The current model proposed improves on previous custom-built models both in performance and specificity. The model reduces its size and saves on computational cost while increasing precision of matching, for end user purposes, upwards of 80% accuracy.

The initial stage of the brand identification pipeline has two components including a workflow which reads the initial merchant and branded trademark information and goes on to aggregate brand identification signals. Such signals are then processed by a subsequent workflow which refines the brand pairings with external signals from the Brand Ownership Service and exports the refined pairs to a placer. The Brand Ownership Service is the mechanism that verifies brand existence and ownership using USPTO and EUIPO trademark data as well as other internal mechanisms. Finally, the pipeline concludes with the outputs fed into a pipeline to export the brand pairings to their appropriate display.

The two primary inputs used by the service are category signals at the merchant/brand level and signals that provide some inference about the ranking of those particular products. For example, the system may determine if a brand from the merchant is popular based on sales data, and other externally provided information that can signal consumer behavior around the product.

## Detailed Description

As described above, the present disclosure is directed to a high precision machine learning model trained to identify brands. In particular, in some implementations, the systems and methods of the present disclosure can include or otherwise leverage one or more machine-learned models to create signals for top product categories at the brand level, create a brand signal from a custom

Brand Ownership Service based on merchant information, trademark data, and other transaction related information.

Direct to consumer (DTC) brands are constantly evolving due to new business models and product categories. Estimations for the number of DTC brands that may be processed by a merchant center do not have great accuracy and new sources of discovering brand information are becoming more available. To handle the input data and increase identification accuracy, machine learning models may be used to scale the identification effort. The machine learning model attempts to predict the likelihood of a raw brand associated with a specific merchant center account to be accepted as a brand entity and replace the current hand-crafted formula. The model relies on ground-truth data composed of a class label and a list of signals and features such as brand signals from a custom Brand Ownership Service based on external digital brand signals such as trademark data from both the USPTO and the EUIPO.

Brand Verification occurs in multiple forms including voluntary participation by vendors verifying their brands with verified URLs and automated authentication of URLs upon merchant participation. In the first stage a workflow aggregates a large amount of “raw data” from the sources described and more, and filters that data applying hard constraints to it. Merchant/brand pairs are filtered by high-level criteria such as ‘who sells what product’ and ‘how similar are products.’ That data flows into a different pipeline which receives the output of the previous step, enriches and scores that data, and makes it available to downstream users. Said data is then shuffled off to a distribution pipeline. The model achieves a high confidence precision rate of 87.5% -- improving on the previous model’s 75% precision rate. Model use will be directed primarily to three categories: (1) consumer shopping; (2) merchant shopping; and (3) shopping data.

Figure 1 depicts a block diagram of an example machine-learned model according to example implementations of the present disclosure. As illustrated in Figure 1, in some implementations, the machine-learned model is trained to receive input data of one or more types and, in response, provide output data of one or more types. Thus, Figure 1 illustrates the machine-learned model performing inference.

The machine-learned model can be or include one or more of various different types of machine-learned models. In particular, in some implementations, the machine-learned model can perform classification, regression, clustering, anomaly detection, recommendation generation, and/or other tasks.

In some implementations, the machine-learned model can perform various types of classification based on the input data. For example, the machine-learned model can perform binary classification or multiclass classification. In binary classification, the output data can include a classification of the input data into one of two different classes. In multiclass classification, the output data can include a classification of the input data into one (or more) of more than two classes. The classifications can be single label or multi-label.

Another example ensemble technique is stacking, which can, in some instances, be referred to as stacked generalization. Stacking includes training a combiner model to blend or otherwise combine the predictions of several other machine-learned models. Thus, a plurality of machine-learned models (e.g., of same or different type) can be trained based on training data. In addition, a combiner model can be trained to take the predictions from the other machine-learned models as inputs and, in response, produce a final inference or prediction. In some instances, a single-layer logistic regression model can be used as the combiner model.

Another example ensemble technique is boosting. Boosting can include incrementally building an ensemble by iteratively training weak models and then adding to a final strong model. For example, in some instances, each new model can be trained to emphasize the training examples that previous models misinterpreted (e.g., misclassified). For example, a weight associated with each of such misinterpreted examples can be increased. One common implementation of boosting is AdaBoost, which can also be referred to as Adaptive Boosting. Other example boosting techniques include LPBoost; TotalBoost; BrownBoost; xgboost; MadaBoost, LogitBoost, gradient boosting; etc.

Furthermore, any of the models described above (e.g., regression models and artificial neural networks) can be combined to form an ensemble. As an example, an ensemble can include a top level machine-learned model or a heuristic function to combine and/or weight the outputs of the models that form the ensemble.

In some implementations, multiple machine-learned models (e.g., that form an ensemble) can be linked and trained jointly (e.g., through backpropagation of errors sequentially through the model ensemble). However, in some implementations, only a subset (e.g., one) of the jointly trained models is used for inference.

Referring again to Figure 1, and as discussed above, the machine-learned model can be trained or otherwise configured to receive the input data and, in response, provide the output data. The input data can include different types, forms, or variations of input data. As examples, in various implementations, the input data can include global trademark data, internally created signals for top products, and signals from an internal Brand Ownership Service.

In some implementations, the machine-learned model can receive and use the input data in its raw form. In some implementations, the raw input data can be preprocessed. Thus, in addition or alternatively to the raw input data, the machine-learned model can receive and use the preprocessed input data.

In some implementations, preprocessing the input data can include extracting one or more additional features from the raw input data. For example, feature extraction techniques can be applied to the input data to generate one or more new, additional features. Example feature extraction techniques include edge detection; corner detection; blob detection; ridge detection; scale-invariant feature transform; motion detection; optical flow; Hough transform; term frequency, inverse document frequency, etc.

As another example preprocessing technique, portions of the input data can be imputed. For example, additional synthetic input data can be generated through interpolation and/or extrapolation.

Referring again to Figure 1, in response to receipt of the input data, the machine-learned model can provide the output data. The output data can include different types, forms, or variations of output data. As examples, in various implementations, the output data can include a simplified Machine Learning training pipeline, the top products signal and internal Brand Ownership Service signals, and generated truth data for internal training purposes.

As discussed above, in some implementations, the output data can include various types of classification data (e.g., binary classification, multiclass classification, single label, multi-label, discrete classification, regressive classification, probabilistic classification, etc.) or can include various types of regressive data (e.g., linear regression, polynomial regression, nonlinear regression, simple regression, multiple regression, etc.). In other instances, the output data can include clustering data, anomaly detection data, recommendation data, or any of the other forms of output data discussed above.

In some implementations, the output data can influence downstream processes or decision making. As one example, in some implementations, the output data can be interpreted and/or acted upon by a rules-based regulator.

Figure 3 depicts the proposed model in detail according to the present disclosure. As illustrated in Figure 3, in some implementations, the machine-learned model is trained to receive input data of one or more types and, in response, provide output data of one or more types. Thus, Figure 3 illustrates the machine-learned model performing inference.

Thus, the present disclosure provides systems and methods that include or otherwise leverage one or more machine-learned models to enable machine learning models to identify brand categories for commercial or general use based on limited input signals indicative of brand origin, product popularity, and other identification and ranking metrics for branded goods and services. Any of the different types or forms of input data described above can be combined with any of the different types or forms of machine-learned models described above to provide any of the different types or forms of output data described above.

The systems and methods of the present disclosure can be implemented by or otherwise executed on one or more computing devices. Example computing devices include user computing devices (e.g., laptops, desktops, and mobile computing devices such as tablets, smartphones, wearable computing devices, etc.); embedded computing devices (e.g., devices embedded within a vehicle, camera, image sensor, industrial machine, satellite, gaming console or controller, or home appliance such as a refrigerator, thermostat, energy meter, home energy manager, smart home assistant, etc.); server computing devices (e.g., database servers, parameter servers, file servers, mail servers, print servers, web servers, game servers, application servers, etc.); dedicated, specialized model processing or training devices; virtual computing devices; other computing devices or computing infrastructure; or combinations thereof.

Thus, in some implementations, the machine-learned model can be stored at and/or implemented locally by an embedded device or a user computing device such as a mobile device. Output data obtained through local implementation of the machine-learned model at the embedded device or the user computing device can be used to improve performance of the embedded device or the user computing device (e.g., an application implemented by the embedded device or the user computing device). As one example, Figure 2 illustrates a block diagram of an example computing device that stores and implements a machine-learned model locally.

In other implementations, the machine-learned model can be stored at and/or implemented by a server computing device. In some instances, output data obtained through implementation of the machine-learned model at the server computing device can be used to improve other server tasks or can be used by other non-user devices to improve services performed by or for such other non-user devices. For example, the output data can improve other downstream processes performed by the server computing device for a user computing device or embedded computing device. In other instances, output data obtained through implementation of the machine-learned model at the server computing device can be sent to and used by a user computing device, an embedded computing device, or some other client device. For example, the server computing device can be said to perform machine learning as a service.

In yet other implementations, different respective portions of the machine-learned model can be stored at and/or implemented by some combination of a user computing device; an embedded computing device; a server computing device; etc.

Computing devices can perform graph processing techniques or other machine learning techniques using one or more machine learning platforms, frameworks, and/or libraries, such as, for example, TensorFlow, Caffe/Caffe2, Theano, Torch/PyTorch, MXnet, CNTK, etc.

Computing devices can be distributed at different physical locations and connected via one or more networks. Distributed computing devices can operate according to sequential computing architectures, parallel computing architectures, or combinations thereof. In one example, distributed computing devices can be controlled or guided through use of a parameter server.

In some implementations, multiple instances of the machine-learned model can be parallelized to provide increased processing throughput. For example, the multiple instances of the machine-learned model can be parallelized on a single processing device or computing device or parallelized across multiple processing devices or computing devices.

Each computing device that implements the machine-learned model or other aspects of the present disclosure can include a number of hardware components that enable performance of the techniques described herein. For example, each computing device can include one or more memory devices that store some or all of the machine-learned model. For example, the machine-learned model can be a structured numerical representation that is stored in memory. The one or more memory devices can also include instructions for implementing the machine-learned model or performing other operations. Example memory devices include RAM, ROM, EEPROM, EPROM, flash memory devices, magnetic disks, etc., and combinations thereof.

Each computing device can also include one or more processing devices that implement some or all of the machine-learned model and/or perform other related operations. Example processing devices include one or more of: a central processing unit (CPU); a visual processing unit (VPU); a graphics processing unit (GPU); a tensor processing unit (TPU); a neural processing unit (NPU); a neural processing engine; a core of a CPU, VPU, GPU, TPU, NPU or other processing device; an application specific integrated circuit (ASIC); a field programmable gate array (FPGA); a co-processor; a controller; or combinations of the processing devices described above. Processing devices can be embedded within other hardware components such as, for example, an image sensor, accelerometer, etc.

Hardware components (e.g., memory devices and/or processing devices) can be spread across multiple physically distributed computing devices and/or virtually distributed computing systems.

The technology discussed herein makes reference to servers, databases, software applications, and other computer-based systems, as well as actions taken and information sent to and from such systems. The inherent flexibility of computer-based systems allows for a great variety of possible configurations, combinations, and divisions of tasks and functionality between and among components. For instance, processes discussed herein can be implemented using a single device or component or multiple devices or components working in combination. Databases and applications can be implemented on a single system or distributed across multiple systems. Distributed components can operate sequentially or in parallel.

In addition, the machine learning techniques described herein are readily interchangeable and combinable. Although certain example techniques have been described, many others exist and can be used in conjunction with aspects of the present disclosure.

Thus, while the present subject matter has been described in detail with respect to various specific example implementations, each example is provided by way of explanation, not limitation of the disclosure. One of ordinary skill in the art can readily make alterations to, variations of, and equivalents to such implementations. Accordingly, the subject disclosure does not preclude inclusion of such modifications, variations and/or additions to the present subject matter as would be readily apparent to one of ordinary skill in the art. For instance, features illustrated or described as part of one implementation can be used with another implementation to yield a still further implementation.

A brief overview of example machine-learned models and associated techniques has been provided by the present disclosure. For additional details, readers should review the following references: *Machine Learning A Probabilistic Perspective* (Murphy); *Rules of Machine Learning: Best Practices for ML Engineering* (Zinkevich); *Deep Learning* (Goodfellow); *Reinforcement Learning: An Introduction* (Sutton); and *Artificial Intelligence: A Modern Approach* (Norvig).

## Figures

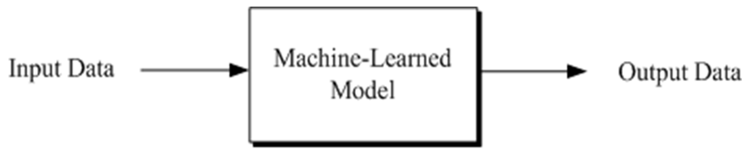

Figure 1

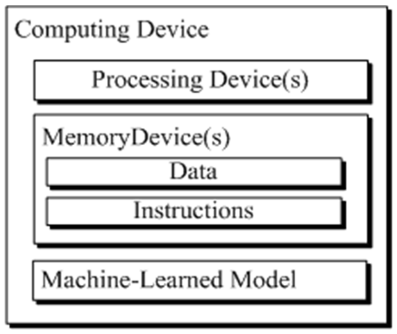

Figure 2

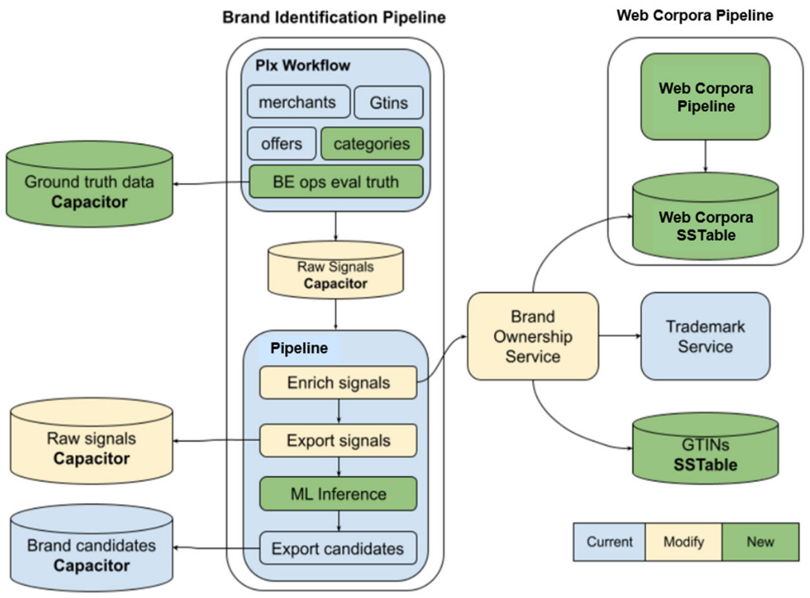

Figure 3

## Abstract

A high precision machine learning model trained to identify brands is described. One or more machine-learned models are trained to create signals for top product categories at the brand/merchant level and brand signals from a custom Brand Ownership Service based on external digital brand signals such as trademark data from the USPTO, the EUIPO, and/or other trademark registries.