USE CASES OF MACHINE LEARNING IN DEXFREIGHT'S DECENTRALIZED LOGISTICS PLATFORM

Abstract

In this document, we describe in high-level use cases and strategies to implement machine learning algorithms to provide platform users with benefits such as intuitive selection of loads, carriers, and shippers, trip chaining and revenue maximization, cost prediction, and more. Machine learning algorithms can only be performed on archived and real-time data. In such cases, users will "opt-in" to allow dexFreight to store and access their historical data.

The document provides a strategic vision for deployment of machine learning within the platform and in no way intended to disclose specific details of algorithm design and implementation. More information on algorithm designs will be made available in our code repository during specific calls for bounty programs.

Primary Author

Rajat Rajbhandari, Ph.D. | Co-founder dexFreight

Contributors and Reviewers

Ricardo Escobar

Release date: July 27, 2018

Release version: 1.0

Website: www.dexfreight.io

Contact information: info@dexfreight.io

LinkedIn: www.linkedin.com/dexfreight

Telegram: https://t.me/dexFreight

Disclaimer

- This paper is for informational purposes and describes vision, plan, and implementation strategies for the dexFreight platform.
- The paper's purpose is not to offer or solicit any securities or to invest in any financial instruments related to dexFreight.
- The paper is subject to change at any time without notification. We do not guarantee, represent or warrant any of the statements in the light paper, because they are based on our current beliefs, expectations, and assumptions, about which there can be no assurance due to various anticipated and unanticipated events that may occur.
- Blockchain, cryptocurrencies and other aspects of our technology are in their infancy and will be subject to many challenges and risks.
- The light paper has been prepared to the best of our knowledge. However, it should not be relied upon for any future actions including but not limited to financial or investment related decisions.
- dexFreight founders, advisors or affiliates shall not be liable for any losses that arise in any way due to the use of this document or the contents contained herein.
- The content, both written and graphics may be historical or forward-looking and therefore should not be relied upon.
- The content is based on assumptions and therefore uses words such as 'expects,' 'intends,' 'will,' 'can,' 'should' or similar expressions.
- The assumptions drawn in this document are based on past trends and data from third parties and other sources, which were believed to be reasonable at the time they were made. However, they still involve unknown risks and uncertainties, as it is impossible to predict anything outside of our immediate control including economic factors.

Version	Release Date	Revision	By
1.0	July 27, 2018	First public release in www.dexfreight.io	dexFreight Team

Table of Contents

Machine Learning and Artificial Intelligence

Machine learning (ML) algorithms help identify natural patterns in observed data, build models that explain the context of the data and predict future observation without having to explicitly program rules. In simple terms, machine learning algorithms "learn" from new data to improve itself to improve the accuracy of analytics and prediction. Hence, it requires a feedback loop to compare the predicted value with the actual observed value. This difference is then applied to the algorithm to "improve" during the next iteration.

Machine learning and Artificial Intelligence (AI) are often used interchangeably. However, they are not the same thing. A recent Forbes article described the difference between the two - AI is the broader concept of machines being able to carry out tasks in a way that we would consider "smart." And, machine learning is a current application of AI based on the idea that we should just be able to give machines access to data and let them learn for themselves[1].

More Than Just A Buzzword

Use of machine learning in logistics and supply chain is nothing new. A recent Business Insider article mentioned a McKinsey study from 2017 that found early adopters with a proactive AI strategy enjoyed profit margins greater than 5% compared to companies in the same sector that had not adopted AI[2].

At dexFreight, it is imperative that ML is part of overall strategy to build a decentralized logistics marketplace. Our mission is to bring benefits of blockchain as well as ML technology to logistics companies of all shapes and sizes. In many ways, both ML and blockchain will be ubiquitous in the functioning of the platform.

We see a lot of instances where machine learning in logistics is thrown as a buzzword in whitepapers, trade magazines, and presentations. Through this document, we want to demonstrate to our potential customers and investors that machine learning is not just a buzzword. Instead, it is a critical component of the platform.

This document describes in high-level several use cases that we will build into the platform by ourselves and third-party developers through bounty programs.

1 https://www.forbes.com/sites/bernardmarr/2016/12/06/what-is-the-difference-between-artificial-intelligence-and-machine-learning/#3aeb45e92742
2 http://www.businessinsider.com/ai-supply-chain-logistics-report-2018-1

Key Benefits To Platform Community

We believe machine learning provides the following benefits to the platform users, including shippers, intermediaries, and carriers as well as third-party service providers:

- Intuitive and simplified workflow that saves time and resource for users while making intelligent decisions to find carriers, loads, and other services.
- Multi-objective optimization as a planning tool to maximize revenue and minimize risks before committing to move loads and hire carriers.
- Short-term prediction of key decision-making variables such as transportation costs to select better price quotes and compete in new markets.
- Monitor and detect undesirable patterns in blocks in public blockchain where transaction and document hashes reside and smart contracts are deployed by the platform.
- Open data faucets and bounty programs for machine learning experts to build better ML algorithms.

Automated Matching And Recommending Loads To Carriers

With millions of loads in the marketplace, and in the absence of automated matching and recommendation system, a carrier will be overwhelmed with a lengthy list of loads, most of which it will not be able to move.

For carrier's benefit, intuitive matching and recommendation system will save them tremendous amount of time when choosing loads they want to bid on. Also, they can choose loads that they ought to pick that will generate the most revenue and/or appropriately fit equipment and capacity they have.

Presenting a short list of loads to a carrier to bid on requires multiple levels of filtering to narrow down the list of loads:

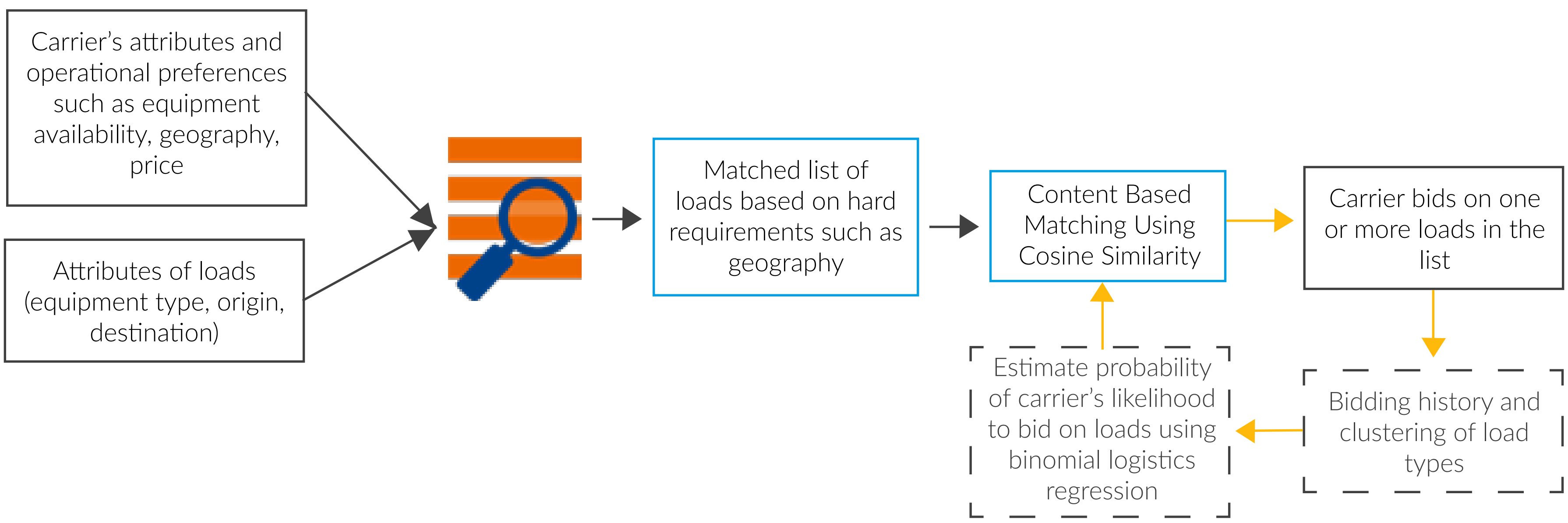
Level 1: Filter and extract a subset of loads based on carrier's hard requirements and pre-defined preferences (load type, equipment type, geographic locations for pick up and drop off.)

Level 2: Use Cosine Similarity or similar selection algorithm to narrow the search and create a smaller subset of loads to present to the carrier.

Assuming only a small set of carriers in the platform respond to price quote on a load, a simple Cosine Similarity algorithm can be performed on x = {x1,... , xn) and y = {y1,... yn) vectors where x1,...xn represent load attributes and y1,...yn represent carrier attributes[3]. The result is a scalar value that measures the orientation of both vectors representing similarity of both. Meaning, the result will show carriers which loads closely resemble their own requirements, and it will show shippers/brokers which carriers have attributes that closely match load requirements.

Carrier is then presented with a smaller set of loads to bid on.

Carrier bids on 1 or more loads, which is then sent to carrier's bidding history.
In next iteration or refresh, after Level 2 matching, binomial probability of carrier to bid on individual loads is estimated using multi-variable binomial logistics regression , as illustrated in the figure below using historical data. Probability of load being selected p(Y) = 1 or 0 is determined with multiple independent variables (x1= equipment type, x2 = price range, x3= geographical radius,... ,xn) using regression on historical data.



Once a shipper chooses a carrier, and it moves a load and shipper is satisfied with the service, the shipper will most likely hire the carrier again. That means next time around with similar kind of load; the shipper will prefer that this carrier appears on top of the recommendation list. Hence, there should be a feedback loop as to which carriers were presented to the shipper and which carrier(s) did the shipper chose to do business with. This loop is then fed into binomial or binary logistics regression model. The model will estimate the likelihood of a carrier being chosen by a shipper and presented to the shipper in the top of the list.

[3] http://blog.christianperone.com/2013/09/machine-learning-cosine-similarity-for-vector-space-models-part-iii/
[4] http://www.med.mcgill.ca/epidemiology/joseph/courses/EPIB-621/logistic2.pdf

Trip Chaining And Revenue Maximization

Truck fleet owners including many owner-operators pick up and drop off multiple loads from more than one origin and destination in a single round trip. Such a trip can last for days and weeks. Dispatchers back in the office assist in finding loads. Sometimes drivers themselves find suitable loads while on the road.

Screening of loads during outbound trips is mostly based on equipment type needed, price, and location. Because this is a manual process that relies heavily on driver and dispatcher's experience, selection of loads may or may not result in maximum benefit (i.e., net revenue, operating cost) to the driver or company. While the first few loads may be planned, remaining loads are not planned or ill-planned (ad hoc.) Hence, trucking companies cannot be certain about selecting the right combination of loads to maximize revenue since it is an ad-hoc process.

We propose a feature that will allow drivers and dispatchers to select an optimal set of loads for a single outbound and inbound trip from an exhaustive list in dexFreight marketplace. The feature has the following benefits:

- This optimal selection of loads is based on constraints such as number of days out, equipment type, fuel cost, etc. and provide maximum per trip revenue. It allows companies to estimate revenue for trips before drivers leave their base.

- The tool automatically finds appropriate loads, loading and unloading points based on user's vehicle type, out days, pricing, etc.

- Then it creates one or more chains (loading and unloading points) based on user-defined constraints along with estimated revenue and other variable costs.

- Users can select an appropriate chain and track progress while on-route.

Warning: It is a recommendation tool that assists companies to find optimal number of loads based on maximum revenue. Conditions on the ground (e.g., load not ready to pick up) can change in which case optimization module needs to be run again.

Finding an optimal number of loads in a set geographic space (e.g., driving distance from the carrier's origin) is a classic vehicle routing problem, but with many practical constraints.

One such constraint is the time window in which drivers have to pick up and drop off loads. The objective function is to either minimize cost (or maximize revenue) by selecting a set of routes S (R1, R2, ..., Rl) such that each customer (load) will be covered by exactly one route Ri as proposed by Chang and Chen (2007)[5]. Other constraints include maximum round-trip time (set by the driver or company) and time window for individual loads.

This is not a classic machine learning problem. But driver or dispatcher's selection bias can be used in a feedback loop to improve the problem.

[5] Chang, Y., and Chen, L., Solve the Vehicle Routing Problem with Time Window via a Genetic Algorithm, Discrete and Continuous Dynamical Systems Supplement, AIM Sciences, 2007. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.359.9152&rep=rep1&type=pdf

Transportation Cost Validation

In the short term, carriers need to know if the price they are bidding at the moment and in the future is competitive and rational. Our goal is to validate carrier's bidding price based on historical price attributed to corridor, shipment type, equipment type, and likely volatility in price in next few days. It ensures that prices being proposed to shippers or freight forwarders are not out of range.

These price predictions are intended only for recommendation purposes and come with X% confident interval. Historical data sets are aggregated using support vector clustering[6], the median price range is determined, and provided to the carrier as a recommended price.

Near Real-Time Monitoring Of Smart Contracts

Since the platform is going to push millions of transaction and document hashes via smart contracts to the public blockchain, dexFreight will be a "caretaker" of those smart contracts on behalf of platform users. Meaning, users of the platform will still assume that dexFreight is responsible for preventing attacks on smart contracts deployed by the platform.

Other purposes of smart contract monitoring are following:

- Check if a hash of transaction or document was mined or if a transaction was successful and present it to the user.

- Monitor short and long-term trends of block validation time and gas prices. Slower block time and increasing gas prices will severely impact the platform's performance, which will impact traction.

- Keep an off-chain log of which smart contract resides in which block number and which transactions reside in the block.

Blockchain explorers such as Etherscan are useful to manually scan blocks where smart contracts reside. Such explorers are a good start to check smart contracts occasionally. But are not appropriate for real-time monitoring, because they have limitations on the number of API calls, and callers have no control over what these explorers log.

We will maintain our own node(s) of public blockchain to perform a faster query of transaction blocks, instead of relying on a third-party API.

Classical time series algorithms such as exponential smoothing, moving average, Box-Jenkins methods can be used to identify seasonality, monitor sudden change in value, and predict short-term values of gas prices, block time, etc. Performance measures of predicted value versus actual value can be used as a feedback to improve the algorithm.

[6] https://en.wikipedia.org/wiki/Support_vector_machine#Support_vector_clustering_(SVC)

Data Faucets And Bounty Programs

Through bounty programs, we encourage third-party developers and data analysts to help us improve machine learning algorithms used within the platform. This requires that we disclose limitations of existing algorithms and define target performance measures for improvement – mostly significant improvement in mean square error and other performance measures are desired. For program participants, data faucets will be made available through which anonymized data will be made available to design and test algorithms.