UNIVERSITY OF CALIFORNIA - SAN DIEGO
DEPARTMENT OF DATA SCIENCE

DSC180A - Quarter 1 Project
Flock Freight - Section A07
Offer Acceptance in the Freight Industry

Keagan Benson, Nima Yazdani, Benson Duong, Radu Manea

Fall Quarter
2022

Abstract

Flock Freight is a company that engages with a variety of carriers and shipments, operating as a marketplace for freight carriers to submit bids on shipping orders that need to be fulfilled, a function commonly referred to as a "freight broker." Other companies remunerate Flock Freight to transport their shipments, which are subsequently outsourced to freight carriers. The company often encounters complications related to the bid acceptance procedure. Some shipment orders receive multiple offers, and minimizing the cost of orders is crucial to the enterprise since it maximizes profits.

The objective of this project is to present a machine learning technique that will establish the optimal stopping point for the acceptance and reneging of delivery offers by carriers for Flock Freight's orders. Time is a valuable resource, and any time spent being indecisive implies a loss of carrier clients. Therefore, it is critical to provide a model and process that is dependable and scalable to Flock Freight's requirements.

1 Introduction

1.1 Background

Flock Freight operates as an intermediary within the shipping industry, serving as a trusted "freight broker" for both shipping carriers and shippers. The company specializes in facilitating the transportation of freight across the country by connecting multiple shippers' loads with a carrier as a "pool" in what is known as a "Shared-Truckload" (STL). Shippers provide Flock Freight with orders, which carriers then bid on to secure the load. Flock Freight ultimately decides which carrier to hire for a specific order and pays them to deliver the shipment.

The decision-making process of accepting or rejecting a carrier's bid can be quite complex, as the goal is to maximize margin while being mindful of potential future offers that may yield better results. Moreover, the consideration of STLs adds another layer of complication, as new orders may arise that can be combined with existing ones after a carrier has already been chosen for the original order.

Optimizing the acceptance of bids is crucial for increasing margins and reducing costs. To achieve this, Flock Freight leverages information on current and historical orders and associated bids to develop a model that follows the optimal stopping problem. This model takes into account contextual factors surrounding bid reception, as well as order details, to predict the optimal bid price.

1.2 Review of Prior Work

Flock Freight operates as an intermediary in the shipping industry, providing a platform that connects shipping carriers and shippers for the purpose of transporting freight across the country. As a "freight broker," Flock Freight acts as a trusted intermediary that specializes in

connecting multiple shippers' loads with a carrier as a "pool" through what is known as a "Shared-Truckload" (STL).

To accomplish this, Flock Freight receives orders from shippers, which carriers will bid on to take the load. Flock Freight then decides which offer to accept for a certain order and pays the carrier to deliver the shipment. However, the decision to accept or reject a carrier's offer can be challenging, as it requires balancing the goal of maximizing margins with the uncertainty of future offers that may be more profitable.

Previous work on the optimal stopping problem has shown that this is a common issue across industries and contexts. For instance, a well-known scenario is the secretary problem, in which an administrator seeks to hire the best secretary out of "n" applicants for the position. The administrator must make a decision immediately following each interview and can only gather information from previously interviewed candidates. The 1/e stopping rule is commonly used to solve this problem, where e is the base of natural logarithms. Applying this rule to the scenario above, if we have 10 randomly selected applicants, we'd interview the first 10/e applicants and record the best applicant. We'd then interview the remaining applicants stopping at the first applicant who beats our recorded "best applicant". If we never find a better applicant and are interviewing the last applicant, we must select the final offer.

Applying this concept to freight brokering, the "applicants" are the offers received from carriers for a particular order. However, the number of offers for a given order is unknown, making it challenging to apply the secretary approach without first building a model to predict the number of offers. We must create a model determining what we classify as a good offer based on historical offer data. This metric for what a good and bad offer is must also take into consideration the variation of offer costs and the number of expected future offers, which both require their own sub-models.Ultimately, the goal is to develop an optimal approach to offer acceptance that maximizes margins while reducing costs.

1.3 Data Description

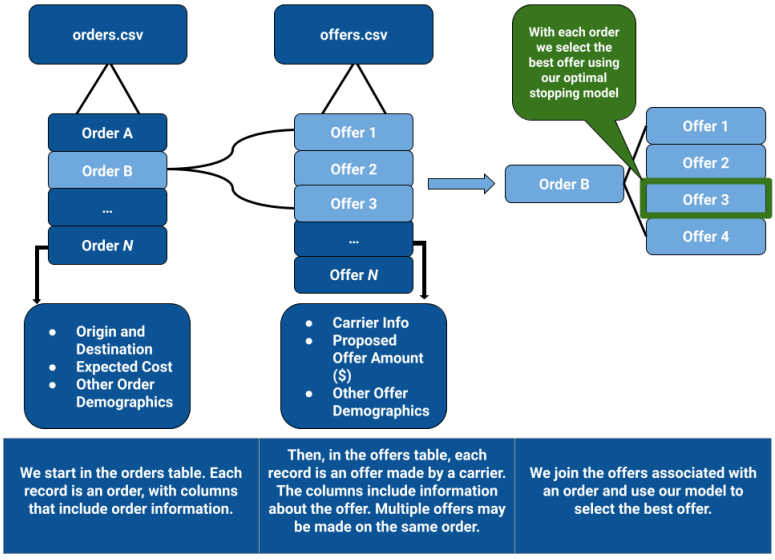
The full dataset includes 2 tables which contain: A) the orders, and descriptive information about it (Order Time, Pickup Deadline, accommodating conditions for its mode of delivery such as transport mode, refrigeration, and hazardness, etc); B) The offers by carriers to deliver said orders - this would be a many-to-one relationship, with the reference number column (assuming it's a singleton list) being the foreign key. The offers table includes mostly information such as the rate of the offer, whether it is pooled or not, whether it was selected, and whether it was uncovered. We also created a supplemental dataset that maps 3 digit zip code identifiers to latitude and longitude coordinates.

Table 1: Orders Data Dictionary

Column Name	Description
REFERENCE_NUMBER	Unique ID for the order
ORDER_DATETIME_PST	Date and time of order in Pacific standard time
PICKUP_DEADLINE_PST	Date and time order must be picked up from origination in Pacific standard time
DELIVERY_TIME_CONSTRAINT	Type of delivery time scheduling constraint
ORIGIN_3DIGIT_ZIP	The first three digits of the origination location ZIP Code
DESTINATION_3DIGIT_ZIP	The first three digits of the destination location ZIP Code
APPROXIMATE_DRIVING_ROUTE_MILEAGE	Approximate number of driving miles from origination to destination
PALLETIZED_LINEAR_FEET	The length and weight of the shipment converted to the percent amount of the truck filled
FD_ENABLED	Customer paid for upgraded service with delivery deadline and no transfer of truck (hub and spoke system not allowed)
EXCLUSIVE_USE_REQUESTED	Customer paid for shipment to be delivered on its own truck (cannot be pooled)
HAZARDOUS	The shipment is hazardous material (cannot be pooled)
REEFER_ALLOWED	The shipment can go on a refrigeration truck
STRAIGHT_TRUCK_ALLOWED	The shipment can go on a straight truck
LOAD_BAR_COUNT	The number of load bars required by the load
LOAD_TO_RIDE_REQUESTED	Delivery service without hub stops
ESTIMATED_COST_AT_ORDER	Flock Freight's estimated cost to fulfill the order (estimated at the time of order)
TRANSPORT_MODE	The type of shipment (FTL, LTL, PTL)

Table 2: Offers Data Dictionary

Column Name	Description
CARRIER_ID	Unique ID for the carrier providing the offer
REFERENCE_NUMBER	Set of order reference numbers the offer would deliver (more than one for a pool)
CREATED_ON_HQ	Date and time offer was submitted in Pacific standard time
RATE_USD	Amount carrier will be paid if offer is accepted
OFFER_TYPE	"pool" for two orders pooled together, or "quote" for one order
SELF_SERVE	Boolean field designating carrier made offer through the app without representative intervention
IS_OFFER_APPROVED	Boolean field designating if Flock Freight approved carrier's offer (carrier must still confirm contract)
AUTOMATICALLY_APPROVED	Boolean field designating if Flock Freight approval was done without representative intervention
MANUALLY_APPROVED	Boolean field designating if Flock Freight approval was done with representative intervention
WAS_EVER_UNCOVERED	Boolean field designating if agreed contract to deliver load was ever broken (e.g. carrier truck broke down)
COVERING_OFFER	Boolean field designating Flock Freight and carrier agreed contract together to deliver load
LOAD_DELIVERED_FROM_OFFER	Boolean field designating this offer was the offer to deliver load
RECOMMENDED_LOAD	Boolean field designating the load (set of order references numbers) was sent to the carrier as a recommended load



2 Methods

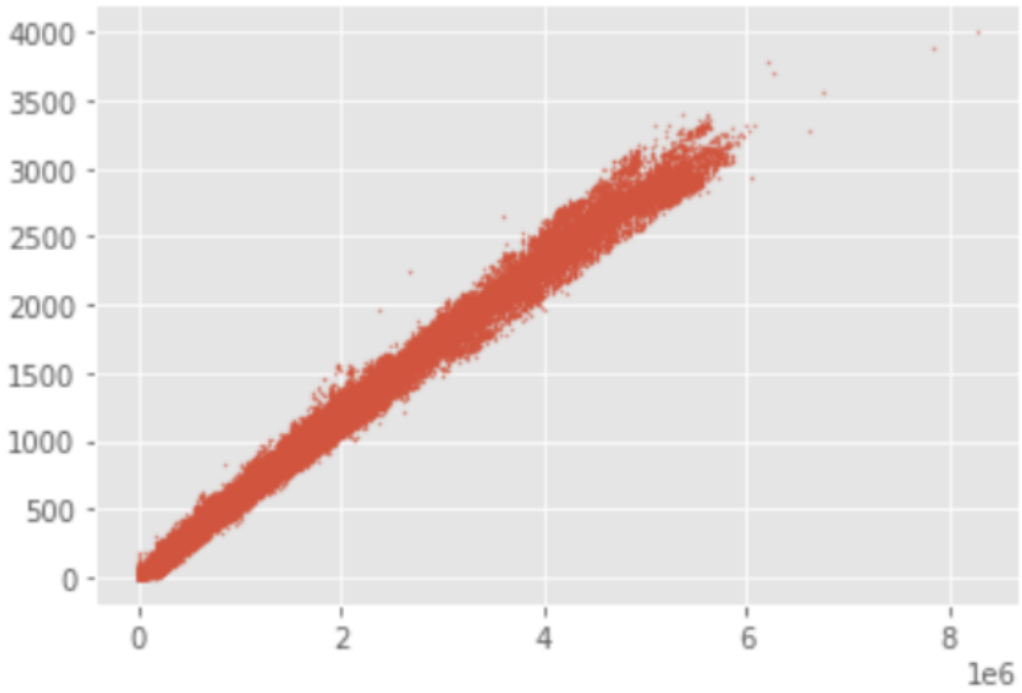
2.1 Exploratory Data Analysis

2.1.1 Geospatial Analysis

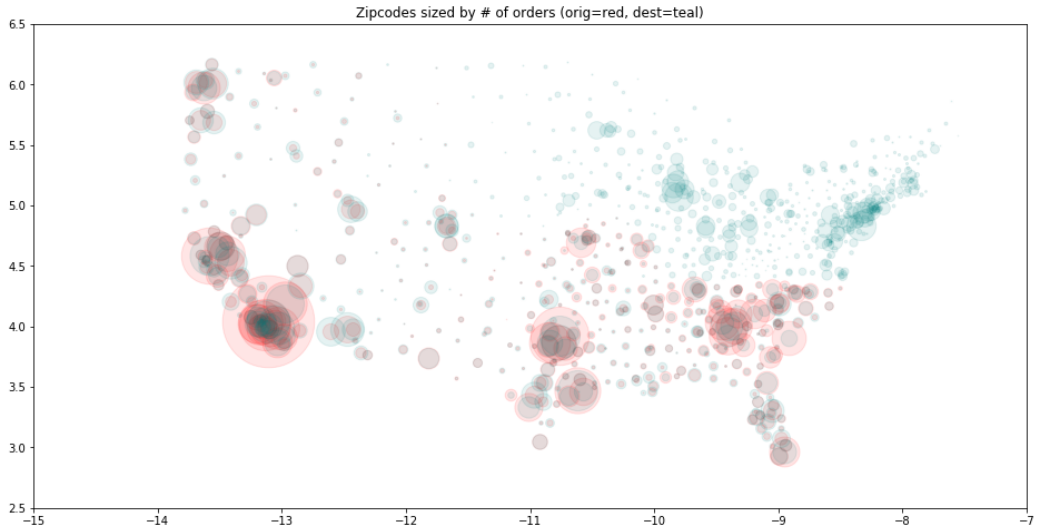
An automated python script first uses BeautifulSoup to web scrape from a government census website to download the latest 2020 census shapefile for zip code tabulation areas, (which is a geospatial data frame that maps each zip code to geographic polygonal areas) for data

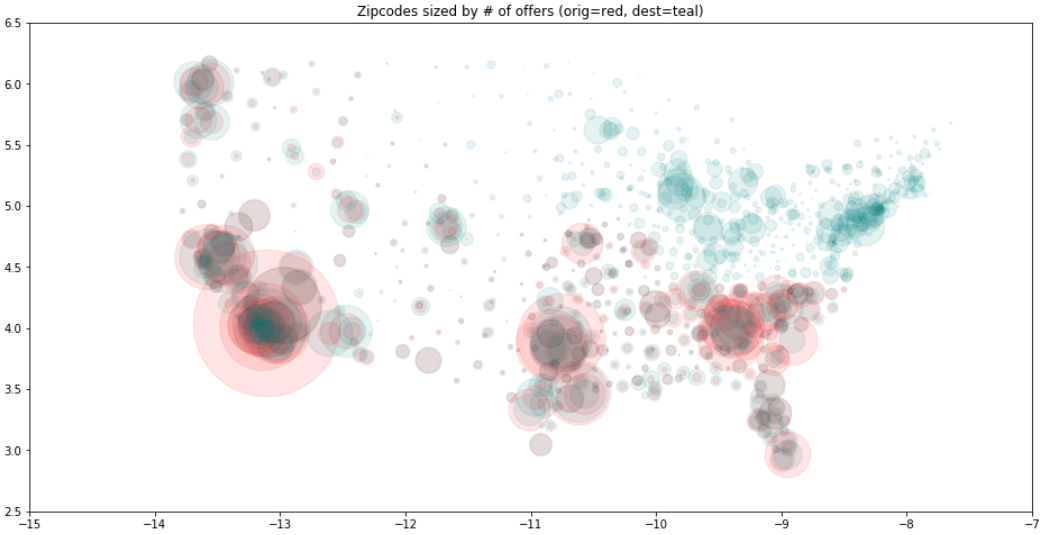
preparation with geopandas; Geopandas will be used to reproject the shapefile to pseudo-mercator. The original data only provides the first 3 (zero padded) digits identifier of the zip codes, whereas the shapefile zip codes are 5 digits; for this reason, the 5-digit zip code column had its 2 last trailing digits dropped (making it usable as a one-to-one foreign key), and Dissolve (a spatial group-by operation in Geopandas) was used to group up the zip code areas into unified polygons. The centroids of each zip code tabulation area will represent the x, y coordinates.

To verify this geospatial data preparation actually produced accurate results, we calculated the euclidean distances between the new x/y coordinates of the destinations and origins, and plotted them against the pre-existing column "Approximate Driving Route Mileage"; the resulting scatterplot is a nearly perfect straight diagonal line, attesting that it is reliable.

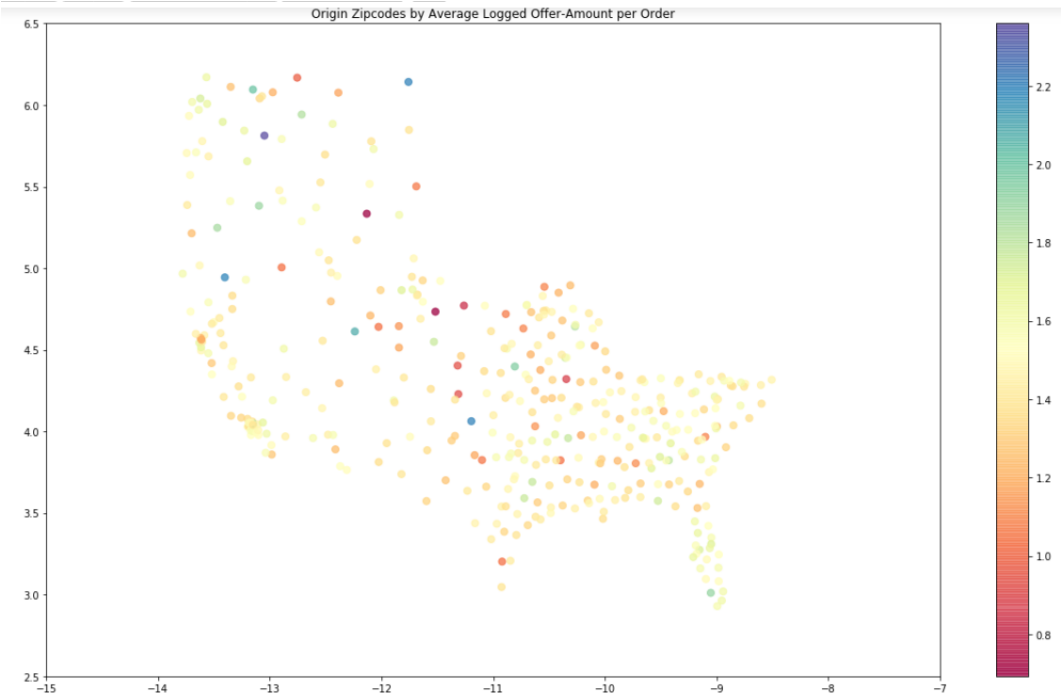


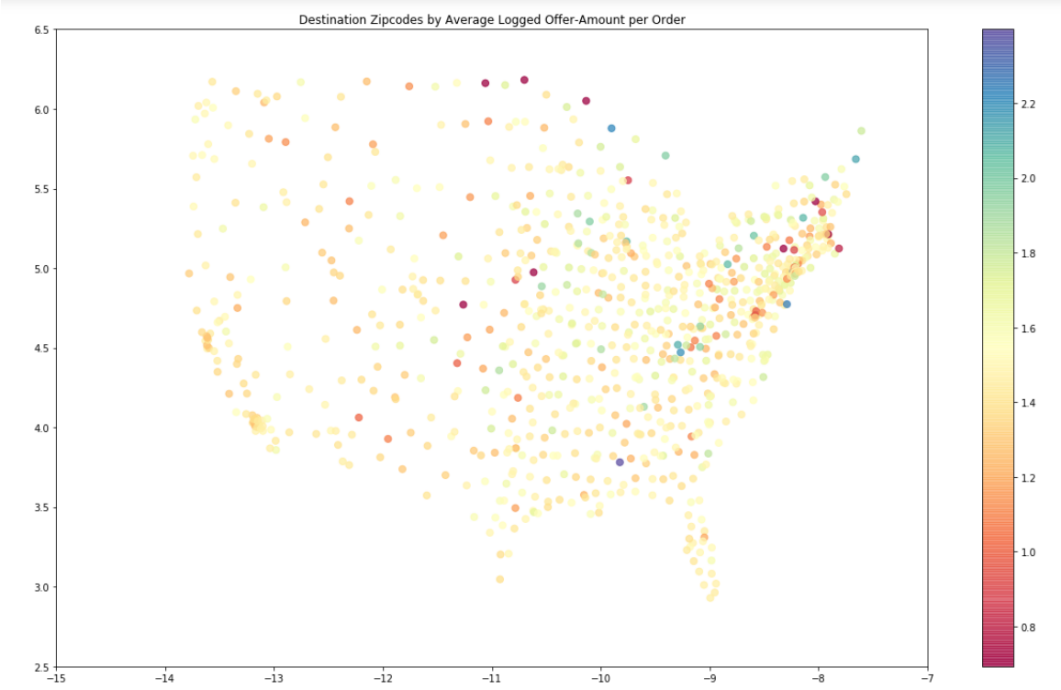
By plotting the orders in terms of their origin and destination zip codes, most scatter plots only show a population bias. The eastern half of the US tends to be more heavily populated than the western half. This makes sense since population centers tend to be logistic hubs, but is otherwise not useful.





The 2 following plots show a scatterplot of zipcode nodes colored according to the logged average amount of offers they see per order, (done for origin zipcodes and destination zipcodes).



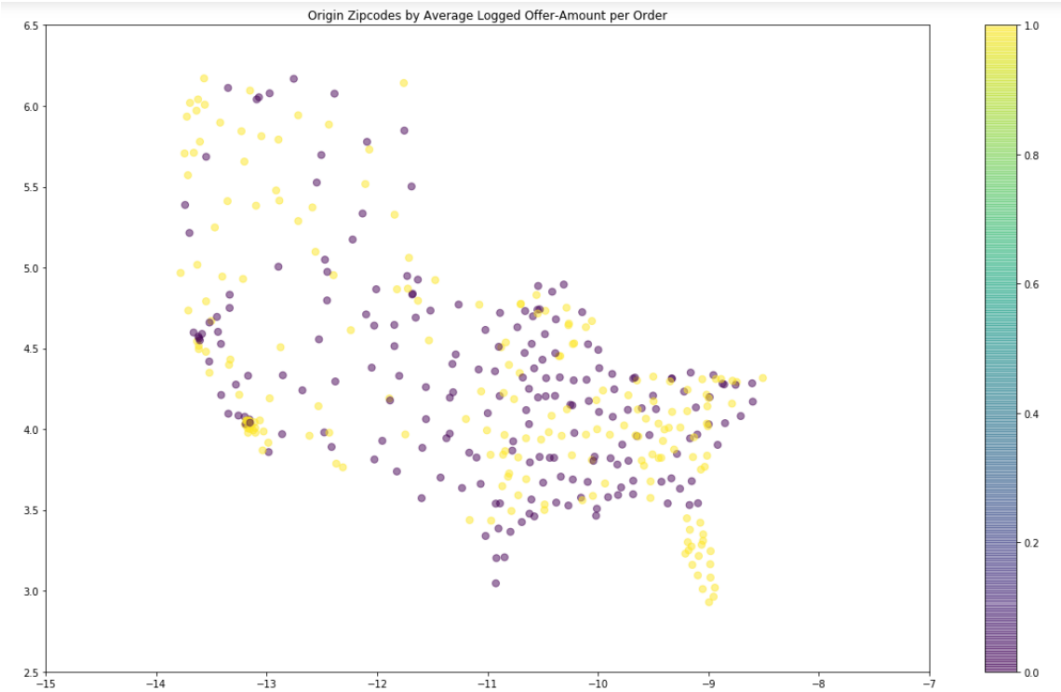


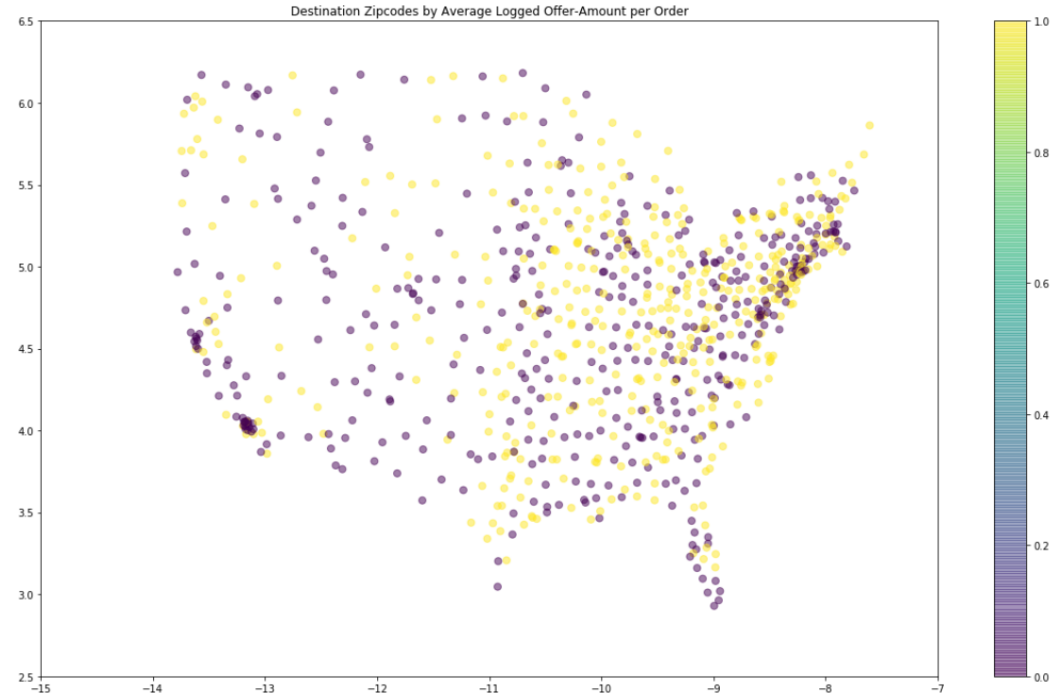
The next 2 plots are binarized versions (above or below the median) of the earlier 2, to make contrast easier to see.

* Clearly, the eastern half of the US and the urban west coast has higher population density, and there is an equal balance of both high and low offer amounts. So utilizing geographic features for these regions for offer amount prediction might not be helpful.

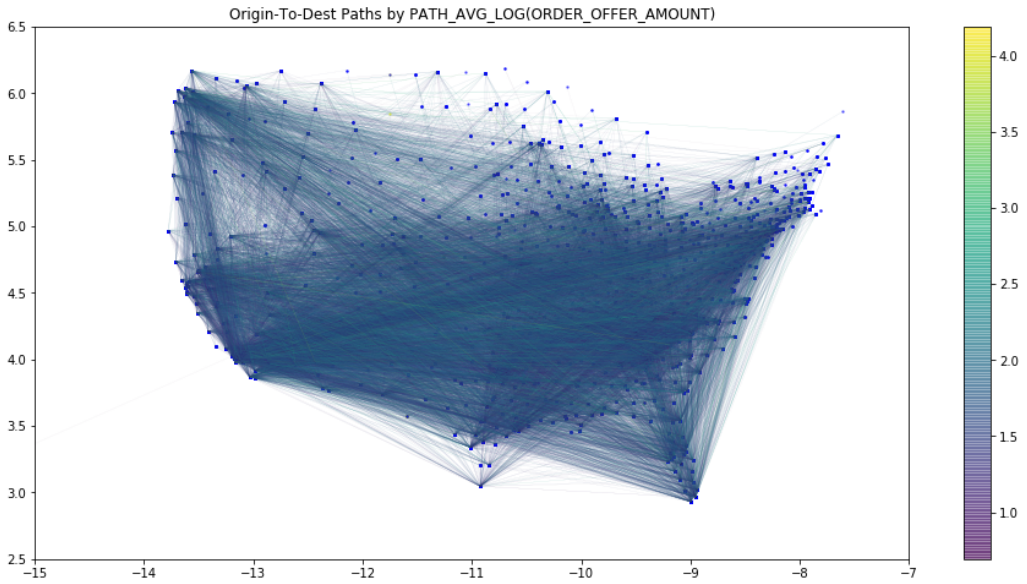
* Orders with an origin zipcode in the Southwest seem to have seem to have a mostly low (below-average) offer amount.

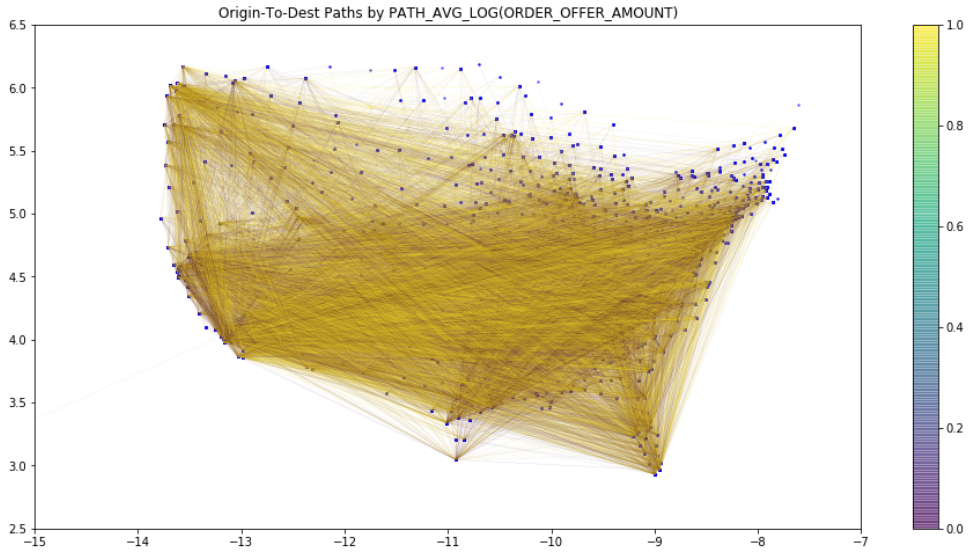
* Orders with an origin zipcode in Florida seem to have a confidently high (above-average) offer amount.



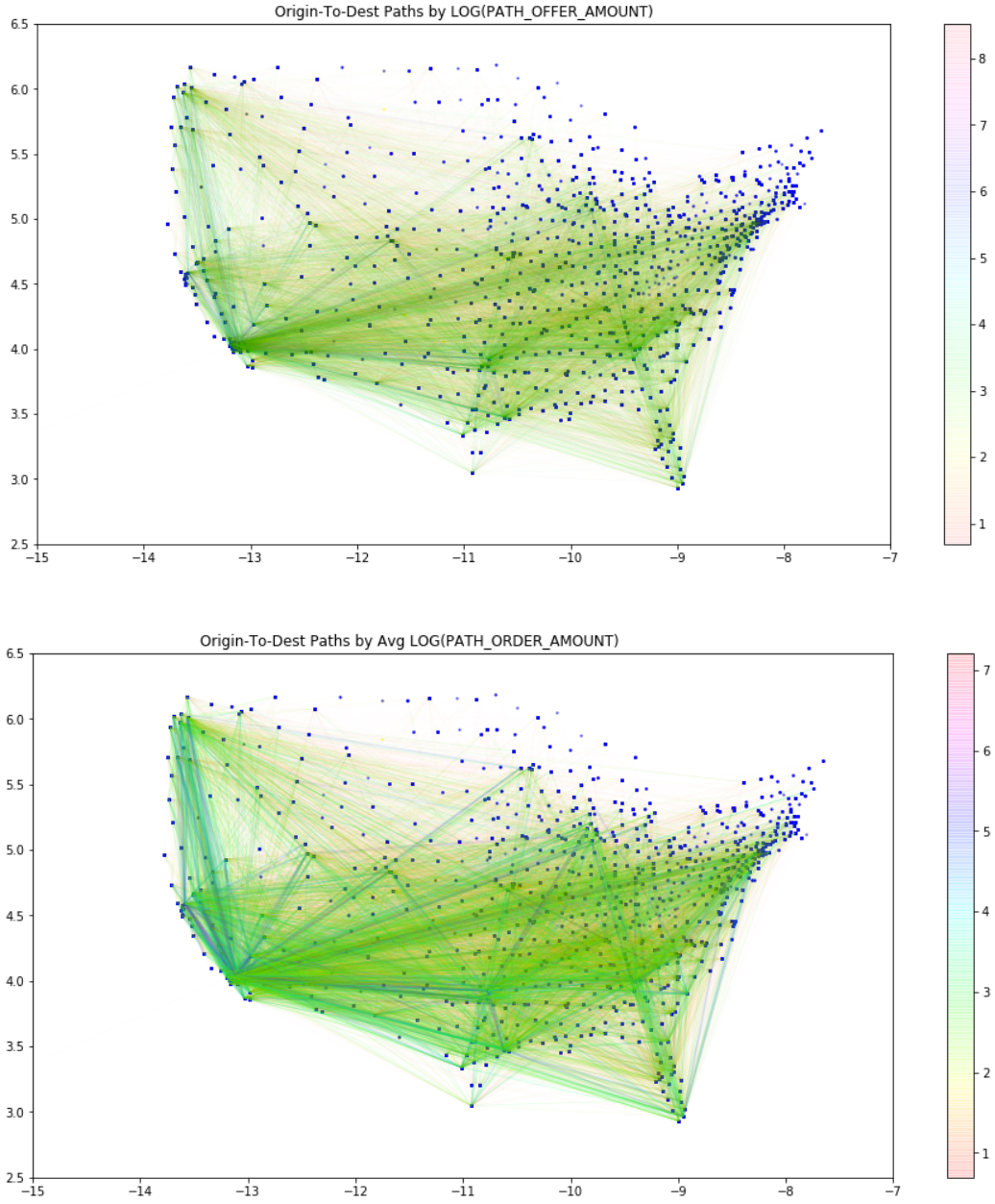


The 2 following plots also try to visualize logged average amount of offers per order, but now it's for shipping routes instead of Nodes. The plot is too difficult to see any differences, so the 2nd plot binarizes the values (below or above the median).

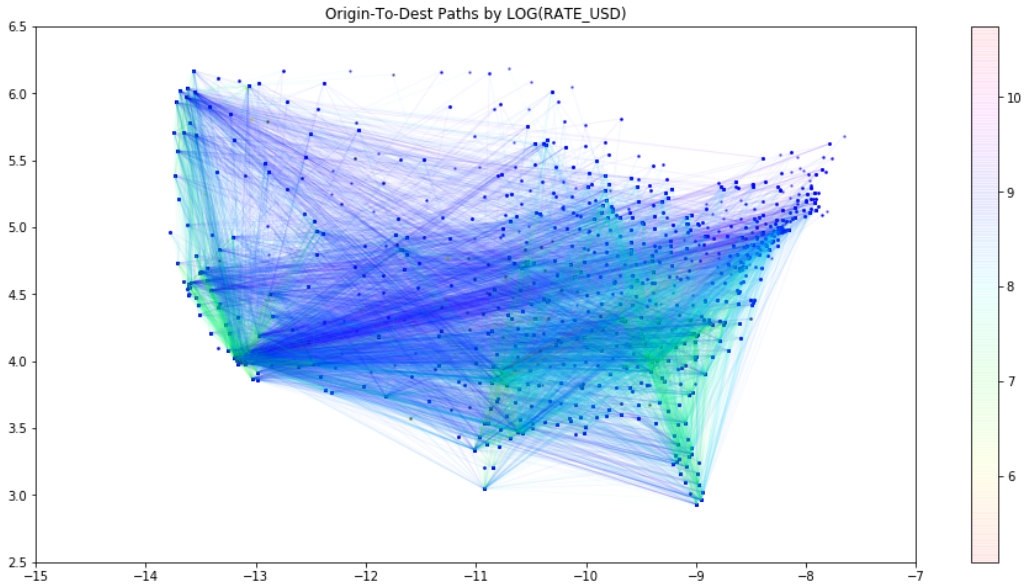


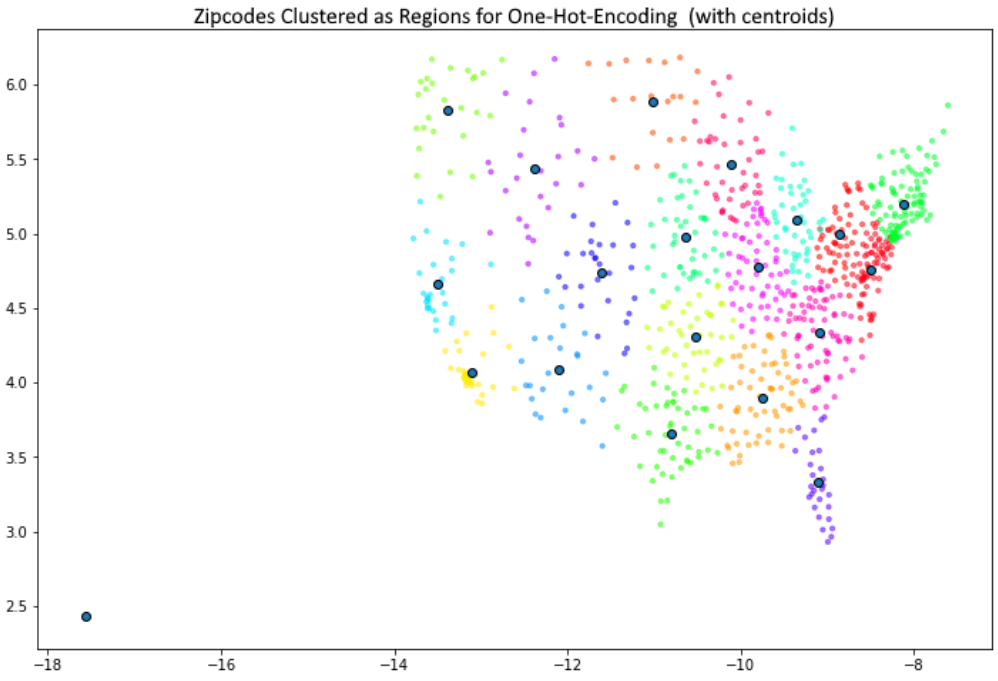


The network of connections between the origins and destinations for orders is also plotted; Side note: some zip code nodes seem fully isolated, but they do in fact have a link ; the links are just at a very low opacity (0.1). What this can mean: Paths with origin / destination zip code nodes that are connected to the cities, will have more orders. In addition, the area in what seems to be between the Upper South and Mid Atlantic have noticeable activity with orders. Shipping between Los Angeles and New York is also very heavy. This could be useful in our prediction model for number of offers.



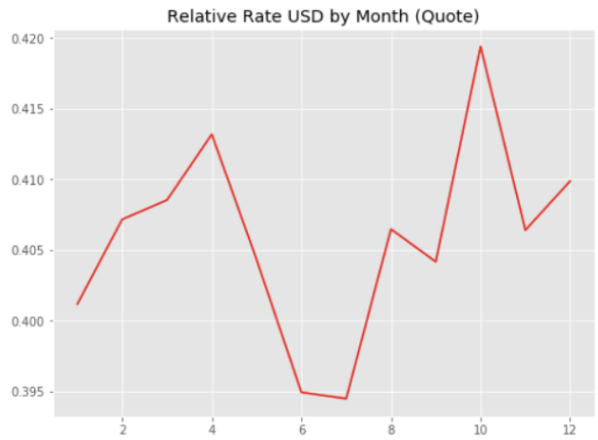
This last figure (Shipping Paths by Log(Rate USD)) shows that several shipping corridors are cheaper than others. These include most deliveries entering and exiting Florida (From Chicago to Florida? And Dallas to Chicago?), and deliveries along the west coast between Southern California to Seattle.

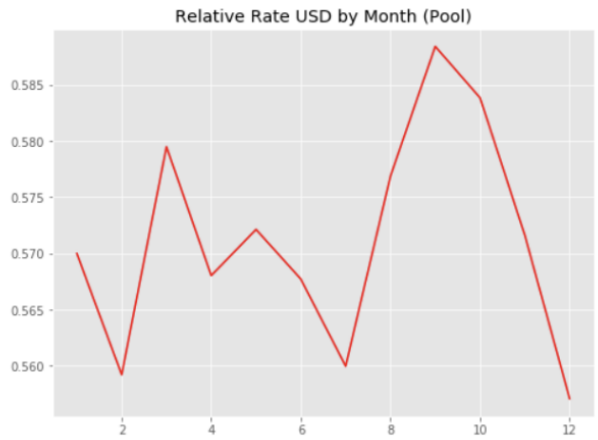




We also tried to use unsupervised ML, specifically grouping the zipcodes into clusters. This was done by applying K-means clustering (N=20) onto 3 columns: the zipcode's X and Y coordinates, and a 3rd dimension to symbolize density (We went with the number of offers seen per zipcode), allowing the clustering procedure to extract metropolitan areas. This is going to be used as a way to one-hot-encode the zipcodes as generalized metropolitan clusters, 20 columns rather than hundreds of zipcodes.

2.1.2 Time Data Analysis



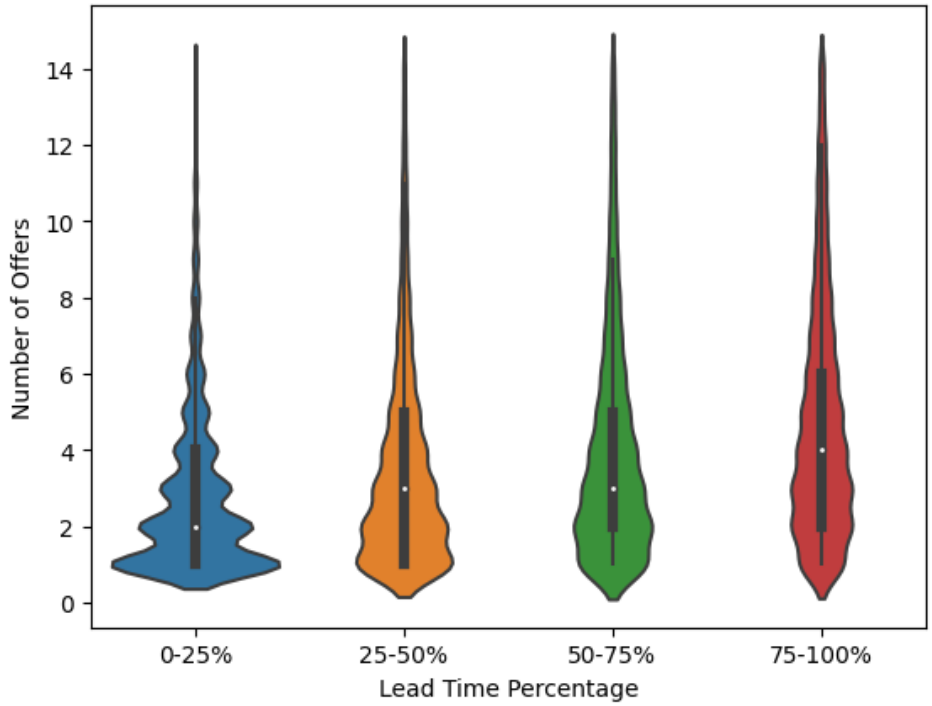


When comparing the metrics of different business days, it was found that Thursday had unusual statistical significance in having the cheapest of rates (relative to the other offers for an order). In other words, we logged the RATE USD column of the joined tables, then z-scored it (not in terms of the whole table, but to each group of offers per order), then binarized it so that being $>= 0$ means above-average (since z-scoring re-centers the mean at 0), and in doing so, found that Thursday has the lowest percentage of above-average offer rates compared to other business days; even with further hypothesis testing (Chi-Square, T-Testing), Thursday continues to pop up as statistically significant. If we put the 2 last columns of the table below through a chi-square hypothesis test, the resulting p-value is 0.0010375. This could be a promising area of potential use for future modeling.

Day	% of Lognorm(Rate)>0	# of Lognorm(Rate)>0	# of Lognorm(Rate)<=0
Friday	0.451306	65670	79841
Monday	0.454253	76642	92079
Thursday	0.448788	74901	91995
Tuesday	0.453000	77902	94067
Wednesday	0.455491	76220	91116

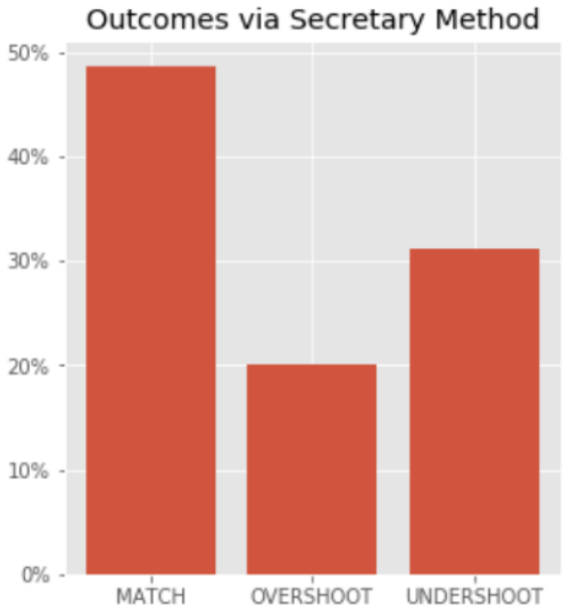
2.1.3 Lead Time Data Analysis

The number of offers an order receives depends on how long into the lead time the order had been when an offer was accepted. Lead time is the difference in time between when an order is placed and when it needs to be picked up. When Flock accepts the offer early in the lead time, it does not accurately reflect the number of offers the order will receive as well as if the order was accepted later into the lead time. When looking at the violin graph, the earlier into the lead time (0-25%), the number of offers are a lot lower than later into the lead time (75-100%). We can use this information to weight our samples when we train the estimating number of offers model.



2.2 The Secretary Method

Secretary Method Outcomes - The following analysis shows that using the raw secretary approach by itself is still a very reliable method: There should be 3 types of outcomes of the secretary method: 1) Match - the method happens to land exactly on the best offer; 2) undershoot - the method stops on an offer better than anything before the 1/e mark, but is not in fact the best one yet; 3) Overshooting - the method had long passed the best offer already- meaning the best offer was actually before the 1/e mark, and the method has no choice but to stop on the last offer, regardless if that last offer is good or bad; Overshooting is the worst outcome; As seen by this barplot for the secretary method, there's a 0.8 chance that either a perfect match or undershooting occurs; While undershooting is not as good as a match, it's a far better position to be in than overshooting - so this is still a good deal.



3 Model

Our Offer Acceptance model comprises of 3 sub-models. They are prediction models that take in the data of an order, and outputs 3 columns representing the average rate of an order, the standard deviation of said average rate of an order, and the number of offers the order gets. These 3 columns are the features of the final model that classifies whether or not an incoming offer for an order is acceptable or not.

* This model setup also has the benefit of letting you adjust the range of tolerance for the threshold (without needing to retrain the model) by multiplying the standard deviation with some positive constant.

* The column for the predicted amount of offers allows more carefulness when rejecting offer: if the amount is high, we are free to gamble with rejecting incoming offers until we get one that is below the threshold.

The Number of Offers sub-model uses order characteristics such as estimated cost, pickup and drop off location, distance apart, size of load and truck requirements to predict the number of offers that order will receive. We can use lead-time for sample-weight adjustment during the training. This will allow the model to use weight the samples higher that were accepted later into the lead time.

The Rate Average Model uses linear regression and is very accurate at 85% to 90%. The Rate Standard Deviation model performed poorly when any sort of regression method was used, so the model task was re-simplified as ordinal classification for the time being: the standard deviation is now split in tiers, and the tier cut-off values serve as the "standard deviations".

For example, if the value 0.3 splits the observed distribution standard deviation in the training set into 50%-50% (i.e. median), then stdev's in the range 0 <= x <= 0.3 are labeled Low and represented by 0. And those 0.3 <= x <= inf are labeled High and represented by 0.3.

Both the Rate Average Model and Rate Standard Deviation model mostly uses the same features, with closely identical input data frames. The features are only allowed to come from the orders dataset.

4 Results

Our resulting offer acceptance model consists of a classifier which determines whether an offer is a good offer. We classified good offers as offers which flock freight accepted and the offers which had the lowest cost of their order.The features the logistic regression model was trained on were the offer rate and the outputs of the sub models, the estimated number of offers, the estimated cost, and the estimated standard deviation. Our model was able to select offers with an average price 4.44% lower than what Flock Freight had selected!

5 Discussion

While the aim was to develop a model that accepts lower-cost offers compared to those historically accepted by Flock Freight, the findings demonstrate that there is considerable room for improvement. The constraints imposed by the secretary method do not entirely align with the real-world problem under consideration. For instance, the inability to reconsider a previously "rejected" offer in our model does not accurately represent the decision-making process in a real-world scenario, where multiple offers can be assessed simultaneously.

Furthermore, the introduction of STLs plays a crucial role in offer acceptance, which has not been accounted for in the final models. Future work should explore the possibility of incorporating pooling, as well as refining the estimation of costs by incorporating and extending geographical features identified in the EDA. The same applies to predicting the number of offers received.

It is important to reevaluate the sub-models of our pipeline and consider whether other models and features could contribute to more accurate classifications. Additionally, the prediction of the number of offers received should be revisited to account for potential future offers that may have been prematurely disregarded. Although these questions were addressed during the model's development, the answers obtained highlight the need for further refinement, reframing of the problem, and abandoning the base secretary method in favor of a purely trained model approach. Enhancements can be made by improving the prediction of a "good offer" basis and refining the prediction of the number of offers and the variation in price for an order.

6 References

Thomas S. Ferguson. "Who Solved the Secretary Problem?." Statist. Sci. 4 (3) 282 - 289, August, 1989. https://doi.org/10.1214/ss/1177012493

Hill, Theodore. “Knowing When To Stop” American Scientist, 6 February, 2017 https:www.americanscientist.org/article/knowing-when-to-stop

https://www.census.gov/geographies/mapping-files/ time-//series/geo/cartographic-boundary.2020.html