Exploring Methods To Predict Pickup Locations With Most Pickups Destined To A Drop-off Location

Lang (Ron) Chen
Student ID: 1181506
Github repo

August 25, 2022

1 Introduction

In recent years, the Taxi industry has been significantly challenged by the rise of app-based rideshare service competitors like Uber[1]; the COVID pandemic, hitting the tertiary sector of the economy hard, has only inflicted further pain. Hence, creating business value out of data has never been as important for taxi companies, with the vast amounts of records from decades in operation being a potent advantage unmatchable by their more youthful competitors.

This report aims to explore methods which New York taxi companies can **use historical data to predict the Pickup Location with the most demand destined to a given Drop-off Location (at a given point in time)**.

Potential benefits from successful models include better allocations of resources (i.e., ability to better match demand and supply through advance direction of taxis to different zones), and also help taxi drivers plan their last service of each shift: if a driver can pick up a passenger travelling towards the same Drop-off zone as their home or depot, it makes drivers better off by reducing their fuel costs in this difficult time where energy prices are record high.

To achieve this, pre-processing will first be undertaken to remove outliers and aggregate data by location and time, before creating two related but different data sets with discrete and continuous labels respectively. Features will then be engineered and selected for model-building, and the best model for each data set (classifier and predictor respectively) will be analysed and compared to derive industry recommendations.

2 The Data Set

The primary data used in this research was provided by the New York Taxi and Limousine Commission (TLC)[2], with data of both yellow and green taxis from 2009 to present (2022) available publicly. Each trip is recorded as a row in the data, with 19 attributes of which the key ones relevant to this research topic are 'Time of Pickup', 'Time of Drop-off' (both precise to second) and [TLC defined] 'Pickup' and 'Drop-off Zone IDs'.

This research used green and yellow taxis' data from January to June 2016 (inclusive), totalling 78,394,032 trips (rows). The time frame chosen was fixed relatively short because longer time periods would significantly increase data volume and burden the limited computational resources and time allocated to this research when performing analysis and model building. Controlling the time frame

also suppresses the impact of potential confounding factors such as the change in demand between zones with passing time. A pre-COVID time frame was chosen because the pandemic drastically increases data volatility and hence the complexity of modelling; so as an exploratory research topic, it is best to first investigate whether the stated aims could be successfully implemented in a less-volatile setting before modelling the more recent and volatile data.

External weather and holiday data sets were also used, and will be discussed during feature engineering.

3 Data Pre-processing

3.1 Outlier and Erroneous inputs removal

The raw data set contains erroneous inputs and records which may distort analysis and modelling, hence their removal during data cleaning is essential. In total, 1,302,354 rows were dropped as outliers or erroneous inputs, accounting 1.66% of the original data.

Invalid pickup or drop-off location

As modelling and analysis is conditioned on Pickup (PUL) and Drop-off locations (DOL), it is essential all records contain valid values (i.e., integer between 1 and 263 inclusive). Any rows with invalid value in the 'PULocationID' and 'DOLocationID' columns were dropped.

Trips with non-positive durations

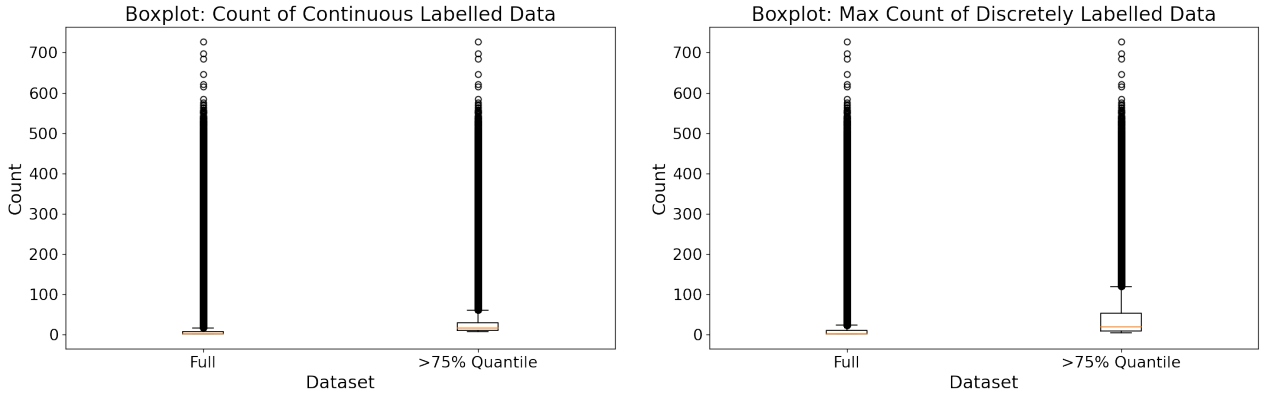
By common sense, a valid trip must have occupied strictly positive time, and trips with 0 seconds is likely a cancelled trip that should be considered erroneous. Although trips taking less than a minute are also likely redundant and would distorts analysis, it is too difficult to ascertain an alternative non-arbitrary cut-off and hence the decision to discard only non-positive duration trips was maintained.

3.2 Aggregation and Labelling

Since the research goal relies only on counts of trips made between zones, with disregard for individual trip details apart from the two LocationIDs and pickup time, the outlier-less data was aggregated by 'PULocationID', 'DOLocationID' and 'Three-Hour Blocks' (3HrB). This granularity of time was chosen because intuitively each three-hour block of the day can be mapped to a typical reason for travel (i.e., 0-3 Clubbers Go Home; 3-6 Early Morning Travel; 6-9 Morning Peak; 9-12 Morning Calm; 12-15 Lunch Break; 15-18 Evening Peak; 18-21 Evening Entertainment; 21-24 Clubbers Go Out).

To build supervised learning models, labels must be generated; two different labelling approaches were taken to allow both classifier and predictor models to be attempted. Data for (continuous) predictors labelled each {DOL, PUL, 3HrB} combination simply by the counts observed. Meanwhile, data for (discrete) classifiers labelled each {DOL, 3HrB} combination by the PUL with most frequent pickups. If the maximum was tied, then create multiple instances with same attribute values but different labels and in evaluation, if the classifier successfully outputted one of the tied labels, it would be considered a correct prediction. Whilst this may artificially inflate accuracy, the validation and test data set also contains duplicated instances which - if classified wrong - would add multiple 'incorrect classifications' to the count and weigh down accuracy. Hence, this special accuracy calculation method catering for the characteristics of this data set should not significantly bias the true accuracy and can be treated just as normal accuracy.

Following aggregation, discretely labelled data had 565,425 rows whilst continuous 9,377,239.



(a) Boxplot of label distribution for continuously labelled data before and after reduction

(b) Boxplot of max count of labels for discretely labelled data before and after reduction

Figure 1: Boxplots of discretely and continuous labels, before and after dropping instances with bottom 75% label values. However, this method only partially mitigates the low-value-cluster problem.

3.3 Fill Non-Observations or Continue to Reduce Instances?

Due to the method of aggregation, {PUL, DOL, 3HrB} combinations with no trips observed were not represented in the continuously labelled data, whilst for discretely labelled data if no taxis made trips to a certain DOL in a 3-hour period, then that {DOL, 3HrB} combination would not be present in data.

Filling unrepresented continuous labelled data attribute combinations with label count = 0 and discretely labelled data attribute combinations with a new ZoneID = 0 (representing "no max PUL" present) was attempted. However, this method was found to be inappropriate as the filled rows severely distorted model-training for both discrete and continuous labels: classifiers assimilated to a 0R model labelling all inputs as Zone 0, whilst predictors always outputted near-0 values. Hence, this pre-processing step was abandoned.

Following this experiment, the distribution of the original trip counts (for continuously labelled data) and trip counts of the maximum PUL (for discretely labelled data) was reviewed, leading to the observation that most "counts"/"max PULs' counts" were close to 0 and could effectively be considered noise (see figure 1a and 1b's left box), as learning to predict or classify when the count is i.e., 1 or 2 creates little business value. The decision was hence made to discard any rows where "count"/"max PUL's count" was below each data set's 75% quantile of 7 and 10 respectively, leaving 2,424,270 continuously labelled instances and 202,253 discretely labelled instances.

While neither figure 1a nor figure 1b demonstrate clear resolution of the aforementioned problem, with both 'after' boxplots still highly negatively skewed, it strikes a balance between completely mitigating this problem and retaining enough samples for training and evaluation.

3.4 External Dataset and Feature Engineering

As this stage of pre-processing, the aggregated data contains no predictive features bar the categorical PULocationID, DOLocationID, and 3HrB. Hence, all classifiers will only pick the most frequent label based on each {DOL, 3HrB}, and predictor models likewise but with the mean count of trips in each {DOL, PUL, 3HrB}. Hence, New York Daily Weather data[3] (containing 28 attributes including 'snowfall', 'rainfall', 'windgust' and 'min', 'max', 'mean temperature') and New York Public Holidays

data[4] (binary Boolean variable for public holidays) were introduced to provide extra predictive variables. Daily granularity was deliberately chosen for weather data because when deployed, models can only classify/predict using forecasted weather data, propagating forecast errors into model outputs; and using less granular data which reduces model sensitivity to noise in weather should limit the negative impact.

Furthermore, another binary attribute denoting weekday was generated using Pickup Datetime, denoting each instance as either 'weekday' or 'weekend'. All categorical attributes (i.e., 'Holidays', 'DOL', 'PUL' for continuously labelled data, '3HrB' etc.) were one-hot encoded.

3.5 Train Validation Test Split

TrainValTest split was performed chronologically as models will predict future data in deployment, so testing models trained on earlier data using later data should better emulate how models will perform when put to real-world use. Jan1-May10 data was used as training data; May11-Jun5 validation and Jun5-Jun30 test.

4 Preliminary Analysis

4.1 0R/null model, distribution of values and preliminary challenge identification

The null classifier model (0R) which always classifies as the most frequently occurring training label (PULocationID) '79', yields an accuracy of 8.1% for training set, meaning 8.1% of training labels was location 79. The 0R validation accuracy is 8.5% for and test 7.7%.

The main challenge with the classification models will be to get enough predictive features to account for 263 different labels, and having enough representation of each PUL label for each subset of data grouped by DOL. Failure of either conditions will likely cause underfit and poor classification.

The null model for the continuously labelled dataset always predicts instances as the mean of counts at 26.35, returning Root Mean Squared Error (RMSE) of 29.58 for the Training set, meaning the average prediction is off by 30 trips per hour. The validation RMSE is 27.78 and test RMSE 27.24.

From the box plot of the continuous label values, it is evident that there are many values that would be considered outliers according to the Q3 + 1.5*IQR upper limit. Although all original labels under the 75% quantile have been discarded, the challenge of instances with labels clustered at lower values perseveres for fitting good predictive models.

4.2 F-Test and Feature Selection

Whilst there were 33 available variables (32 in the case of discretely labelled data because PUL is the label), feature selection must be conducted to only build models using predictive/correlated attributes to avoid overfitting.

The F-test, which measures statistical significance of including one feature in model building via comparing the accuracy/residual deviation (for discrete and continuous respectively) of the null model against a model trained on just that feature, was used to perform feature selection.

Features chosen for discretely labelled data were:

weekdaylabel	3HrB	holidays	snowdepth	snow
temp	min temp	max temp	DOLocationID	dew

Features chosen for continuously labelled data were:

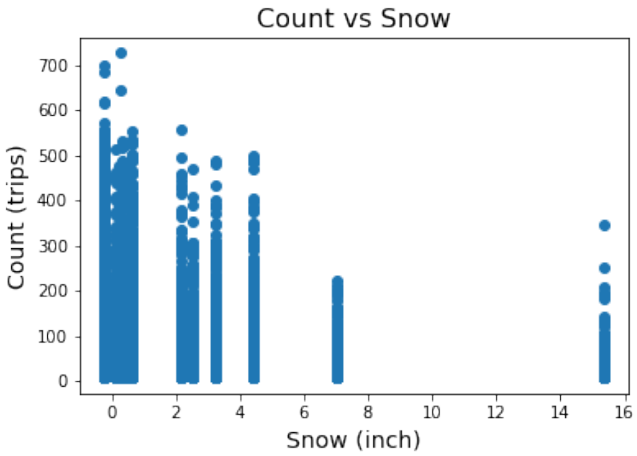


Figure 2: Scatter Plot for Snow and Counts; the negative relationship is not significant due to conditional dependence

weekdaylabel	snowdepth	snow	DOLocationID	PULocationID	windgust
max temp	windspeed	temp	min temp	uvindex	

The difference in the F-test selected attributes make intuitive sense: continuous predictors predict the actual counts per {DOL, PUL, 3HrB}, so weather attributes that significantly affect comfort such as 'windgust' and 'uvindex' are correlative as people alter their travelling patterns/behaviours due to intrusive conditions. Both these attributes do not feature in discretely labelled data because it affects the underlying counts of all PULs, and so will less impact which PUL contains maximum counts of trips going to a DOL and hence correlate less to severe weather attributes. The additional attributes selected for the discretely labelled data set in '3HrB' and 'holidays' - both time related - is intuitively more significant compared to continuously labelled data because different demographic (i.e., workers or leisure seekers), presumably from different PUL, would travel to the same DOL at different times of the day; whereas, the correlation of time attributes to the continuous labels are likely less significant because not every zone's demand is impacted by time in the same way; this henceforth alludes an inherent impasse in feature selection for this research: conditional dependence.

Conditional dependence describes an attribute being independent (non-predictive) to the label over the full data set, but when conditioned to holding another attribute at fixed value, dependence surfaces. The F-test tests each feature standalone, and in this research problem the discrete labels are conditioned on DOL, whilst continuous labels conditioned on {DOL, PUL}. Conducting F-tests assumes the distribution or relationship of labels to attributes are consistent for each DOL/{DOL, PUL}, but this is unreasonable. As seen in figure 2, although snow was one of the higher scoring attributes on the F-test, its negative correlation is rather weak with only a -0.012 Pearson's coefficient. An immediate solution would be to partition the data by DOL/{DOL, PUL} and select features individually, but this would need be repeated 263 times for the discretely labelled and 69169 (263^2) times for the continuously labelled data set, which even if is acceptable in terms of allocated resources makes any model built difficult to interpret or discussed. Hence, this research continued with the results of F-test feature selection whilst maintaining vigilance of the implications of this conditional dependence flaw.

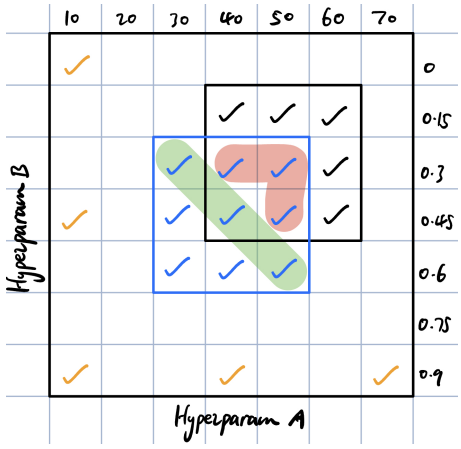


Figure 3: A demonstration of "YangZhou" on d=2 hyperparameter plane: Initialises by train-scoring all combinations in the blue square; then, say the mean difference Welch-test for highlighted red 'block' against green 'block' returns statistical significance, "YangZhou" will proceed to train-score the previously unexplored combinations in the black square; it will stop when no direction is statistically significant and 'cruise' algorithm will search the combinations with orange ticks. The highest scoring searched combination will be returned once the algorithm is terminated.

5 Model Building and Tuning

The best model types for this problem was Logistical Regression for classification and Random Forest Regression for prediction.

Hyperparameter tuning for this project relied mainly on a self-designed heuristic/greedy "YangZhou" Class Tuner, but also the traditional 'Brute Force' search over all hyperparameter combinations for smaller scale tuning. In simplistic terms, "YangZhou" assumes the combinations of hyperparameters exists in a d-dimensional discrete mathematical field (d being the number of hyperparameters). It will first train and get scores for a small 3^d block of hyperparameter combinations before relying on t-Welch statistical tests to determine which direction to move in the hyperparameter combination field to conduct the next set of train-scoring, maximising likelihood for each train-scoring to produce higher validation dataset accuracy/R^2 scores than the previously train-scored combinations. As greedy algorithms can get stuck in local optimums, "YangZhou" implements an end 'cruise' system that train-scores a combination within every 3^d field-space where no combination has been tested, restarting the initial search algorithm at the combination that the 'cruise' found where the validation data set score was statistically not significantly different enough from the current maximum score. Figure 3 provides a brief example.

5.1 Classifier: Logistical Regression (LogR)

The best hyperparameter combination for (one-vs-rest) Logistical Regression was {C = 10, penalty = l2} (tuned completely by brute force given it had only 2 dimensions), with training accuracy 67.2%, validation 64.3% and test 64.4%. This is significant improvement from the 0R, which had accuracies all below 10%. Particularly, it is more than acceptable for a 263 label classification problem, being a 169 times improvement from the random baseline accuracy of 1/263 = 0.38%.

With training score exceeding validation by only 2.9%, overfit would seem to only be slight; and given test accuracy even exceeds the validation's, this LR poses as a strong solution to the research problem.

5.2 Predictor: Random Forest Regressor (RFR)

Regression Trees differ from common Linear Regression in that it builds a decision tree to partition instances before conducting Linear Regression on each leaf; a Random Forest Regressor builds multiple different trees based on random sampling that generates one ultimate prediction by taking the mean of each regression tree's prediction, weighted by each's R^2 score. This more complex regression technique largely resolves the problems identified in the preliminary analysis in the clustering of lower values (and hence heteroskedasticity) by fitting multiple regressions, each only on behavioural-alike data points.

The best hyperparameter combination for RFR was {max_samples_in_bootstrap = 0.75, max_depth = 60, #estimators = 50}. As the inputs to the models DOL, PUL are one-hot-encoded as attributes, max features in bootstrap were left at 1 - it was deemed that allowing the inputs themselves to potentially not be used in a base tree is inappropriate for the research goal, even if it improves the R^2 score.

The training R^2 score is 0.589, validation 0.553 and testing 0.551, with the 0.036 difference evidence of minimal overfit. Nonetheless the validation accuracy (used in analysis instead of test set to allow comparability with the best classifier) is close to the typical industry accepted R^2 standard of 0.6 and is almost deployable. The training RMSE is 18.95, validation 18.57 and test 18.24, making significant improvement compared to the null model.

5.3 Comparison and Contrast

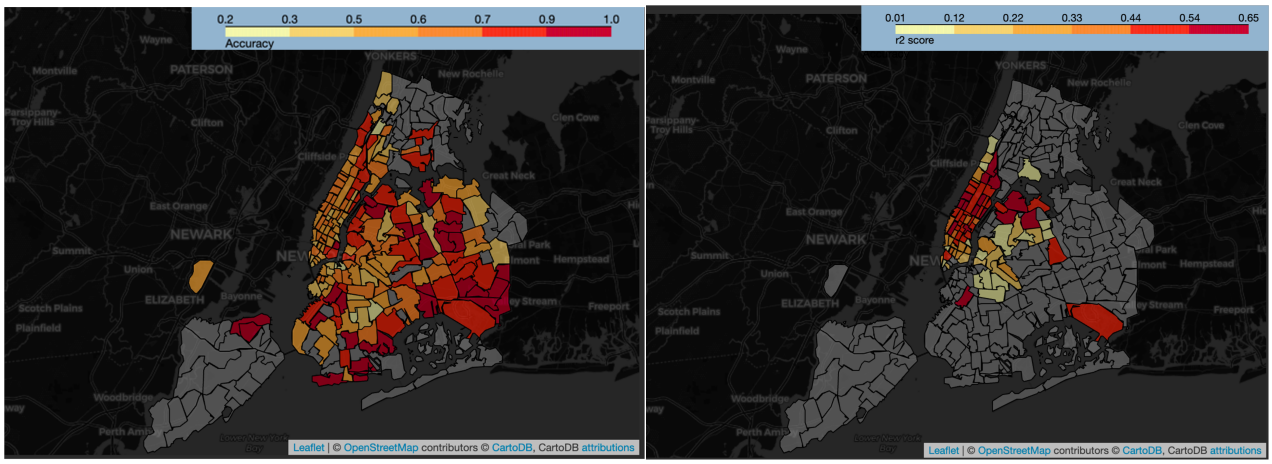
Both candidate models' test performance bring them close to - if not already - deployable in the industry. Particularly being results of an exploratory research, these models at the very least serve as a proof of concept and demonstrates that a successful solution is achievable for this research problem using the available data, and is a strong starting point for future improvements.

Assuming the two scores are relatively comparable given both under normal circumstances take values between [0, 1] (R^2 may be negative if model fit is worse than null), the classifier candidate would pose as the dominant solution for this problem. However, this does not diminish the value of the predictor candidate, as, in practice, a successful model outputting continuous labels is more versatile than one outputting discrete, with classifiers only providing one value whilst predictors capable of returning multiple zones with high expected demand. This in itself is highly valuable as it spreads the risk of an 'all or nothing' choice to multiple likely-accurate values, of which taxi drivers could also choose a zone that has a slightly lower predicted values but is closer to their current location. The existence of two well-performing solutions with different characteristics provides a diverse range of use and benefits for both the TLC and drivers, and hence both models remain valuable even though in rough comparison the classifier seem to outperform the regressor.

As analysis of performance in individual zones showed that the overall continuous predictor made some zones worse than 0R (negative R^2 on test data set), those zones were exempted from analysis and in deployment simply should refrain from being predicted for. Zones in the discretely labelled data set with less than 10 test instances were also excluded from analysis as the small sample sizes make accuracy scores unreliable.

In comparing the analysable - and hence useable - zones by observing figure 4, both models exhibit similar patterns in being deployable in Manhattan and parts of Queens/Brooklyn whilst basically un-useable in Staten Island, parts of the Bronx and Long Beach - all on the outskirts of the city. This is likely because taxi services saw higher demand in the Central Business District and significantly lesser moving away from the city centres.

On a micro-level, 21 zones had 100% test accuracy, and 36 more had test scores above 80%. The



(a) Accuracy Score Heatmap for Logistic Regression (b) R^2 Score Heatmap for Random Forest Regressor

Figure 4: Heatmaps of scores for Classifier and Predictor; grey zones are exempted from analysis

lowest accuracy on a zone which was tested (as some zones had no test instances) was 18.8% - still more than double the 0R of 7.7%. Meanwhile, only 84 zones performed better than just predicting the zone's mean using the predictor. Zones 7, 42, 236, 129 surpassed the 0.6 R^2 benchmark with zone 7 peaking at R^2 of 0.654. Whilst these 4 zones all achieved higher than 75% test accuracy on the classifier, with all 200 zone-7 test instances accurately predicted, there is little evidence to suggest the two models learnt similarly to achieve their strong performances given 3 of the predictor's best zones yielded less than 80% accuracy and hence were outside the 78% percentile of top accuracy zones.

6 Recommendations for Industry

Both models are basically ready-made for industry deployment, with the predictor perhaps requiring slight improvements. In particular, its ability to extrapolate onto instances that had lower values (which were discarded in section 3.3) must be tested, at least to ensure they are not systematically overestimated.

The LogR could immediately be made available to taxi drivers by creating a mobile application interface to host the models and database to help them be more likely to find passengers heading to the same DOL as their depot/home in the last trip of their shift, and the zone's test accuracy should be displayed alongside the classification output as a measure of confidence. When drivers are destined to RFR-useable zones, the RFR should be activated to provide more choices for drivers.

The LogR could also be used by the TLC to build (using also accurate estimated trip durations which the TLC should be capable of generating) a topological model that helps drivers plan entire shifts in advance so they can most likely finish their last trip in their home/depot's DOL. Moreover, the RFR could contribute to a much more sophisticated model (albeit serving predominately the TLC) by building a Continuous Time Markov Chains every three hours with zones as Markov Chain states and Transition Rates derived from expected trips to better capture the demand for taxis in New York. This too makes drivers better off by reducing expected wait times between Drop-off and next Pickup, thus increasing efficiency.

Overall, results from this research should readily make both taxi drivers and the TLC better off, and improved models using this research as a starting point should only enhance this effect.

References

[1] Forbes. *Dislocation And Its Discontents: Ride-Sharing's Impact On The Taxi Industry.* https://www.forbes.com/sites/michaelgoldstein/2018/06/08/uber-lyft-taxi-drivers/. Accessed: 2022-08-10.

[2] TLC. *TLC Trip Record Data.* https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page. Accessed: 2022-07-26.

[3] Visual Crossing. *Weather Data Services.* https://www.visualcrossing.com/weather/weather-data-services. Accessed: 2022-08-01.

[4] Office Holidays. *Federal Holidays In New York.* https://www.officeholidays.com/countries/usa/new-york/2016. Accessed: 2022-08-04.