A Tree-Based Model for Activity Based Travel Models and Feature Selection

Lisa Kuwahara, Sophia Lau, Ruiqin (Max) Li

1. Introduction

In a previous study, Deloitte Consulting LLP developed a method of creating city simulations through cellular location and geospatial data. Using these simulations of human activity and traffic patterns, better decisions can be made regarding modes of transportation or road construction. However, the current commonly used method of estimating transportation mode choice is a utility model that involves many features and coefficients that may not necessarily be important but still make the model more complex. Our goal is to use a tree-based approach - in particular, XGBoost - to identify just the features that are important for determining mode choice so that we can create a model that is simpler, robust, and easily deployable. By the end of the quarter, we plan to have a working model to predict mode choice and a written paper to report our results and analyses.

2. Hypothesis

The primary reason for choosing a tree-based model is due to its built-in hyperparameters for handling overfitting. These hyperparameters will be important in ensuring that our model is general enough to be applied to different scenarios while maintaining its accuracy. Therefore we need a good technique to choose the best values for our hyperparameters.

Currently, most models use grid search or random search to pick hyperparameter values; however, these methods are exhaustive and inefficient. Our proposal is to use Bayesian optimization, a method that involves choosing a value from a distribution based on its past

evaluations. Since Bayesian optimization learns from its past iterations, it is able to find the optimal hyperparameter values faster than grid or random search.

Our hypothesis is that our XGBoost classifier for travel mode choice, tuned with Bayesian optimization, will be able to achieve the same or higher accuracy than existing utility models on the same Chicago dataset and maintain a similar performance on other datasets, including imbalanced datasets and specific subpopulations.

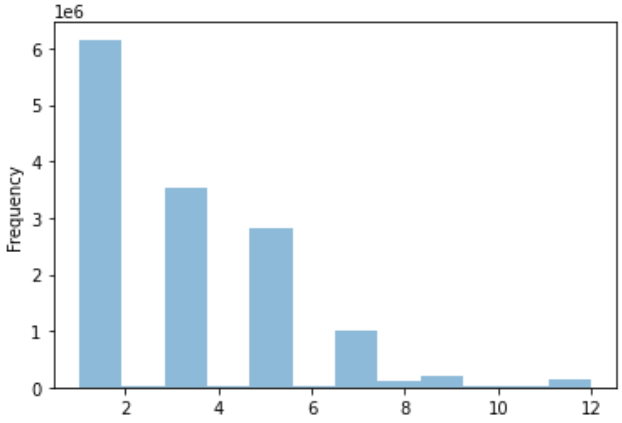
3. Datasets

The datasets we use consists of two dataframes: trips and utilityvars. The first dataframe, trips comes from a csv that previously predicted the travel mode choice each activity took, linked to activity and person IDs.

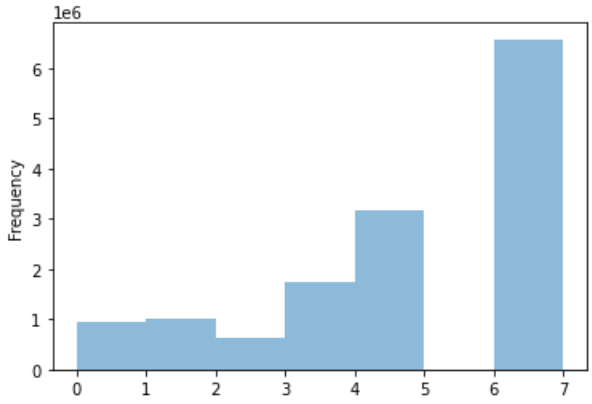
The utilityvars data provides us information about each unique trip, which includes activityid, age, gender, autosuf, numhouseholdpersons, income, oduden, oempden, ototint, dempden, sovdrivetime, hovdrivetime, tolldrivetime, tollcostsov, tollcosthov2, tollcosthov3, walkttime, walktotransitutility, drivetotransitutility, parkingcost, parkingwalktime, sovcost, hovcost, tollcost, tourpurpose, firststop, laststop, zerototalstops, and the variable we are predicting, targettripmode. Tourmode was removed from our dataset because it provides the same information as our targettripmode. We plotted some of the variables to get insight into the distributions of our data and the activities we have.

The targettripmode has 12 different trip modes, and its distribution is shown in the figure below. The modes range from 1 to 12, and the most common trip modes are 1, 3, and 5, which are Drive Alone Free, HOV2 Free, HOV3 Free. The least common trip modes are 6, 10, 4:

HOV3 Pay, Park and Ride, HOV2 Pay. This shows that most travelling is done through driving and without paying. In most cases, it is rare that people pay when driving.

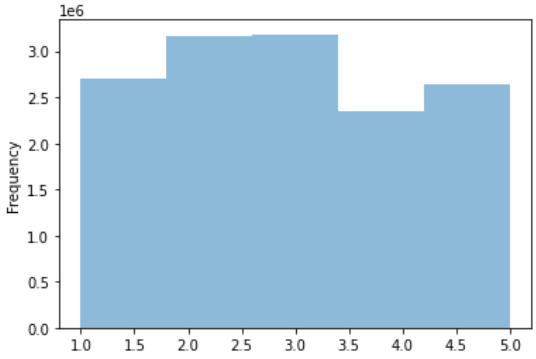


Another variable we plotted a histogram of is age. Most activities consist of people of age groups categorized as 4, 6, 7, which are people of ages 25 and above, with more of the older age range. The gap we see between 5 to 6 is because our data on age does not have the category of 5.



Another descriptive variable we have is of income. The income group 3 is the highest, which has the average income of $60k to $100k. The lowest group for income is group 4, which is the average income of $100k to $150k. Most households in our dataset have an income that is

around average, and in general, of lower income groups. Income subpopulations can be interpreted in our model because this can affect travel modes.



4. Techniques

4.1 XGBoost

XGBoost, abbreviated from Extreme Gradient Boosting, is an ensemble tree method that builds upon the results from simple decision trees, using gradient descent to boost the weak learners, to make better decisions as it increases in depth. The objective function is defined as follows:

$$Obj = \sum_{i=1}^{n} L(\widehat{y_i}, y_i) + \sum_{i=1}^{k} R(f_i)$$

where L is the loss function, is the predicted label and is the actual label, and R(f) is the penalty function for complexity.

The hyperparameters of the classifier are described below:

Parameter	**Default**	**Description**
booster	gbtree	choose which booster to use
verbosity	1	verbosity of printing messages
nthread	max	number of paralleled threads used

learning-rate	0.3	step size shrinkage used in update
gamma	0	min loss reduction required to make a split
max_depth	6	maximum depth of a tree
min_child_weight	1	minimum sum of weights required in a child
max_delta_step	0	maximum allowed tree weight estimation
subsample	1	fraction of observations to be randomly samples
colsample_bytree	1	subsample ratio of columns for each tree
reg_lambda	1	L2 regularization term on weights
reg_alpha	0	L1 regularization term on weights
tree_method	auto	tree construction algorithm
scale_pos_weight	1	balance of positive and negative weights
max_leaves	0	maximum number of nodes to be added
objective	reg:squarederror	loss function to be minimized
eval_metric	rmse	metric used for validation data
seed	0	random number seed

4.2 Bayesian optimizer

Since there are so many hyperparameters in the XGBoost classifier, using grid search for tuning would take a long time and would limit us to a set number of values to test. Instead, we will use a Bayesian optimizer with mean AUC score as the evaluation indicator, similarly to the one used in the heart disease article, to find the value for each hyperparameter that will yield the highest accuracy on the validation set.

Bayesian Optimization is capable of effectively optimizing object functions which are costly to evaluate. It takes into account past evaluations when selecting the next hyper-parameter to set. It selects hyper-parameter combinations in an informed way, concentrating on the most promising areas of search space, so it generally needs less iteration to get the best hyper-parameter combination.

The generic Bayesian optimization algorithm (acquisition function: EI) below:

For t = 1,2,3… repeat:

- By optimizing the acquisition function over the Gaussian process, find the next sampling point: $X_t = argmax_x u(x|\mathcal{D}_{1:t-1})$
- By evaluating the objective function f, obtain a possibly noisy sample $y_t = f(X_t) + \epsilon_t$
- Add new sample (X_t, y_t) to the previous samples $\mathcal{D}_{1:t} = \{\mathcal{D}_{1:t-1}, (X_t, y_t)\}$ and update the Gaussian process

5. XGBoost Model Implementation

We split the data 80/20 into train and test sets, resulting in a train set of 9600 trips and a test set of 2400 trips for our uniform subsample. Since our data is a multiclass problem, we used the multi softprob function as our objective function and AUC as our evaluation metric.

6. Results

7.1.1 Evaluation (curves, confusion matrix, metrics)

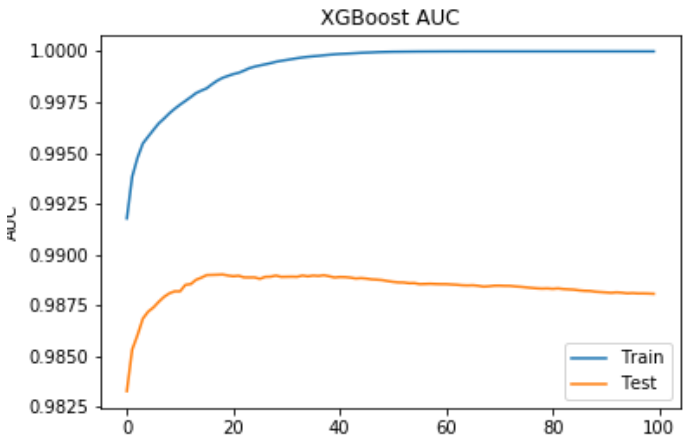
After we finished the model, we created a validation pipeline to estimate the performance of our model and sought the potential problems in our model, like overfitting or imbalanced sample.

For measurement of correctness, we referred to evaluation metrics such as accuracy, sensitivity, F1-score. The feedback indicates that our model did a great job on the sample data, but also causes a great deal of concern. In the next weeks, we would diagnose the problem and adjust our model accordingly.

Current metrics:

Metric name	Score
Accuracy	0.8808
Sensitivity	0.8833
Precision	0.8840
F1-score	0.8836

We also created the AUC curve for our model performance in the train/test dataset.



To check whether our model performed badly on specific labels, we output the confusion matrix and normalized it.

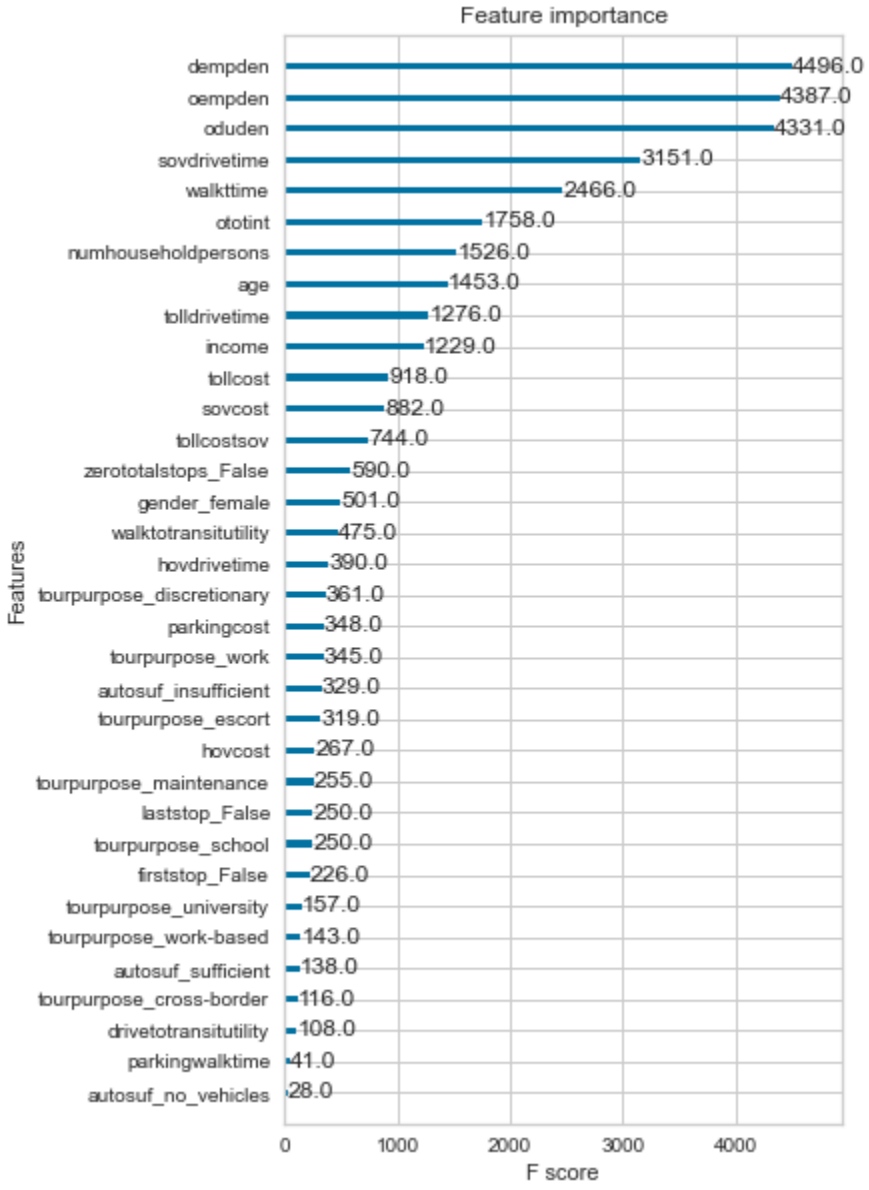
0.803	0.015	0.103	0	0.061	0.003	0	0	0.012	0.003	0	0
0.010	0.869	0	0.054	0.006	0.061	0	0	0	0	0	0
0.029	0	0.747	0.015	0.132	0.003	0.047	0	0.018	0	0.009	0
0.003	0.065	0.045	0.763	0.006	0.113	0.003	0	0	0	0.003	0
0.018	0	0.047	0.003	0.868	0.012	0.029	0	0.015	0	0.009	0
0	0.006	0.003	0.053	0.056	0.878	0.003	0	0.003	0	0	0
0.018	0	0.048	0	0.063	0	0.816	0	0.051	0	0.003	0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0.006	0	0.012	0.003	0.012	0	0.959	0.003	0.006	0
0	0	0	0	0	0	0	0	0	1	0	0
0	0	0.003	0.006	0	0	0	0	0	0	0.991	0
0	0	0	0	0	0	0	0	0	0	0	1

7.1.2 Evaluation on Income Groups

7.2 Feature Fine-tuning

We retrieved the importance scores for all our features in order to create a new model using only the important features. Based on the graph below, we observed that employment density at the origin and destination regions (dempden, oempden) were the two most important features in determining trip mode, followed by dwelling unit density at the origin region (oduden). This may be due to the fact that areas with a higher employment or residential density are likely more urbanized and have more public transportation options available, such as trains, subways, and buses. We intend to use all the features as or more important than the single

occupancy vehicle cost (sovcost) in our optimized model and observe how well it predicts trip mode in comparison to the model trained on all the features.



7. Appendix: Proposal

1. Introduction

In a previous study, Deloitte Consulting LLP developed a method of creating city simulations through cellular location and geospatial data. Using these simulations of human activity and traffic patterns, better decisions

can be made regarding modes of transportation or road construction. However, the current commonly used method of estimating transportation mode choice is a utility model that involves many features and coefficients that may not necessarily be important but still make the model more complex. Our goal is to use a tree-based approach - in particular, XGBoost - to identify just the features that are important for determining mode choice so that we can create a model that is simpler, robust, and easily deployable. By the end of the quarter, we plan to have a working model to predict mode choice and a written paper to report our results and analyses.

2. Hypothesis

The primary reason for choosing a tree-based model is due to its built-in hyperparameters for handling overfitting. These hyperparameters will be important in ensuring that our model is general enough to be applied to different scenarios while maintaining its accuracy. Therefore we need a good technique to choose the best values for our hyperparameters.

Currently, most models use grid search or random search to pick hyperparameter values; however, these methods are exhaustive and inefficient. Our proposal is to use Bayesian optimization, a method that involves choosing a value from a distribution based on its past evaluations. Since Bayesian optimization learns from its past iterations, it is able to find the optimal hyperparameter values faster than grid or random search.

Our hypothesis is that our XGBoost classifier for travel mode choice, tuned with Bayesian optimization, will be able to achieve the same or higher accuracy than existing utility models on the same Chicago dataset and maintain a similar performance on other datasets, including imbalanced datasets and specific subpopulations.

3. Datasets

The data we use in the next quarter consists of three dataframes: household_data, person_data, and mode_choice_variables, which cover the details and background of a potential trip.

The household data give us information about its location, income level, and building/unit type.

	hhid	household_serial_no	taz	mgra	hinccat1	hinc	hworkers	veh	persons	hht	bldgsz	unittype	version	poverty
0	2	0	3407	10	5	210480	1	0	2	1	2	0	0	15.970
1	3	0	3458	16	5	236389	0	0	2	3	2	0	0	15.728
2	4	0	3379	27	4	121977	2	0	2	2	2	0	0	8.353
3	5	0	3358	3	4	104512	2	0	2	7	2	0	0	7.157
4	6	0	3385	20	3	68513	0	0	5	3	2	0	0	2.674

The person data tell us about the basic background of individuals in each household, including age, sex, race, educational attainment, and employment status.

	hhid	perid	household_serial_no	pnum	age	sex	miltary	pemploy	pstudent	ptype	...	occen5	occsoc5	indcen	weeks	hours	rac1p	hisp	version
0	1	2	0	2	75	1	0	1	3	1	...	0	11-1021	0	1	35	1	1	0
1	2	3	0	1	85	2	0	3	3	5	...	0	00-0000	0	5	0	1	1	0
2	2	4	0	2	67	1	0	1	3	1	...	0	11-1021	0	1	35	1	1	0
3	3	5	0	1	10	2	0	4	1	7	...	0	00-0000	0	5	0	1	1	0
4	3	6	0	2	60	2	0	3	3	4	...	0	00-0000	0	5	0	1	1	0

The mode choice data focus on the information about traveling. It contains key points, such as the purpose of the tour, the standard of living in the origin area, and the time and money spent on the trip in different choices (different combinations of vehicle types and road types). The methods of transportation include single occupancy vehicle, multiple occupancy vehicle, and walk while the road types include open road and toll road. It can help us to confirm the actual travel mode that a specific individual would like to choose.

	activityid	age	gender	autosuf	numhouseholdpersons	income	oduden	oempden	ototint	dempden	...	parkingwalktime	sovcost	hovcost
0	3018709	6	False	2	2	1	8.3047	70.117912	12.0	32.273338	...	0.0	495.225231	495.225231
1	8809200	6	False	2	6	3	10.1895	47.538746	3.0	13.567157	...	0.0	462.137117	462.137117
2	11332534	4	False	2	1	2	13.3932	21.983759	4.0	51.544109	...	0.0	342.291418	342.291418
3	6130767	6	False	2	3	2	13.2740	6.568953	3.0	8.480003	...	0.0	310.186071	310.186071
4	16693912	7	True	1	2	5	5.5589	10.919112	12.0	32.705276	...	0.0	1.514632	1.514632

4. Techniques

4.1 XGBoost

XGBoost, abbreviated from Extreme Gradient Boosting, is an ensemble tree method that builds upon the results from simple decision trees, using gradient descent to boost the weak learners, to make better decisions as it increases in depth. The objective function is defined as follows:

$$Obj = \sum_{i=1}^{n} L(\widehat{y_i}, y_i) + \sum_{i=1}^{k} R(f_i)$$

where L is the loss function, is the predicted label and is the actual label, and R(f) is the penalty function for complexity.

The hyperparameters of the classifier are described below:

Parameter	Default	Description
learning_rate	0.3	Shrink the weights on each step
n_estimators	100	Number of trees to fit.
objective	binary: logistic	logistic regression for binary classification
booster	gbtree	Select the model for each iteration
nthread	max	Input the system core number
min_child_weight	1	Minimum sum of weights
max_depth	6	Maximum depth of a tree.
gamma	0	The minimum loss reduction needed for splitting
subsample	1	Control the sample's proportion
colsample_bytree	1	Column's fraction of random samples
reg_lambda	1	L2 regularization term on weights
reg_alpha	0	L1 regularization term on weights

4.2 Bayesian optimizer

Since there are so many hyperparameters in the XGBoost classifier, using grid search for tuning would take a long time and would limit us to a set number of values to test. Instead, we will use a Bayesian optimizer with mean AUC score as the evaluation indicator, similarly to the one used in the heart disease article, to find the value for each hyperparameter that will yield the highest accuracy on the validation set.

Bayesian Optimization is capable of effectively optimizing object functions which are costly to evaluate. It takes into account past evaluations when selecting the next hyper-parameter to set. It selects hyper-parameter combinations in an informed way, concentrating on the most promising areas of search space, so it generally needs less iteration to get the best hyper-parameter combination.

The generic Bayesian optimization algorithm (acquisition function: EI) below:

For t = 1,2,3… repeat:

- By optimizing the acquisition function over the Gaussian process, find the next sampling point: $X_t = argmax_x u(x|\mathcal{D}_{1:t-1})$
- By evaluating the objective function f, obtain a possibly noisy sample $y_t = f(X_t) + \epsilon_t$
- Add new sample (X_t, y_t) to the previous samples $\mathcal{D}_{1:t} = \{\mathcal{D}_{1:t-1}, (X_t, y_t)\}$ and update the Gaussian process

5. Quarter 1 Project: Heart Disease Dataset

To familiarize ourselves with the algorithm that we will use in the next quarter capstone project, we replicated a paper *An optimized XGBoost based diagnostic system for effective prediction of heart disease*. The analysis is carried out on Cleveland heart disease dataset taken from UCI online machine learning and data mining repository. The dataset contains 303 patients who potentially had heart disease in distinct levels and their living features like age, sex, and etc. The goal of the authors is to correctly predict whether a patient has heart disease, regardless of whether the symptoms are severe or not. The proposed XGBoost model in the paper has a high accuracy of 91.8% and performed better than other tree models. We noticed, however, that the 13 features the authors used were based on other studies rather than their own results, so we chose to keep all the features, excluding ones that did not provide useful values (e.g. “id”, features with only one value), in our replication model to find important features manually.

Although the background of the replication task is different from that of our project in the next quarter, it provides us with a robust XGBoost tree as our classification model. Many techniques that we have been taught in the replication task could be applied to the travel model: One-hot encoding, hyper-parameter tuning by Bayesian Optimization, feature selection, and etc. We get to know about different evaluation metrics to evaluate the tree model, and the standard is also suitable for our future classifier.