Modeling Ridership from Transactional Data in the King County Metropolitan Transit System

Hannah Murphy, Alexander Van Roijen, Maggie Stark, Jacob Warwick [1]
University of Washington, Seattle, Washington, USA

March 14, 2020

submitted in partial fulfillment of the requirements for the degree of

Masters of Science in Data Science

Abstract

How can metropolitan bus systems use incomplete transactional data to create actionable estimates of ridership? Many municipal transit agencies offer riders a variety of fare payment options including cash, payment cards, passes, and apps. Because transactional data from diverse sources are hard to aggregate, and because some riders board buses without paying, transit agencies have only been able to estimate demand via in-person rider census, or automated person counting devices. For logistical and cost reasons, many transit agencies, including King County Metro, cannot utilize automated person counting across every bus route and all times of day, instead collecting data from a random sample of routes within a study period. It is impossible to answer detailed questions about specific routes, directions, and times of day where person-count estimates are missing. Enabling transit agencies to answer these questions will allow them to better estimate the impact of delays and schedule changes on riders, and give planners better tools when proposing new routes and schedules.

We use automated person count data from the King County Metro system to create a machine learning model that accurately estimates actual ridership using the incomplete transactional data from ORCA cards. This model can estimate total actual ridership across all routes, directions, and times of day, regardless of whether the bus was outfitted with automated person counting technology. To achieve this, we first created a data processing pipeline which aggregates data from automated person counting, vehicle location records, and ORCA transactions, then assessed various machine learning models to see which offered the best performance. The best performing model was a shallow, wide neural network utilizing the sigmoid function in the first layer with a linear output neuron. We found the optimal width of the first layer to be between two to four times the number of features. We hypothesize that each neuron in the first layer acted as an encoder for a latent feature, and demonstrate that this network has optimized to a sophisticated solution capable of accurately estimating ridership in previously unseen observations with high accuracy. We propose that this model structure could be applied to other transit systems to create accurate, actionable ridership estimates out of incomplete transactional data.

[1] We extend our thanks to Mark Hallenbeck and Dmitri Zyuzin at the Washington State Transportation Center (TRAC), for their collaboration, warmth, and expertise, as well as to our capstone faculty member, Megan Hazen, for her guidance and support.

1. Introduction

Bus route planning and optimization is a contributor to the economic success and equity of urban areas and their connected suburbs. [2] In 2018, Seattle was the second fastest growing city among the 50 most populous cities in the United States, with a metropolitan transit system that served 122 million passengers that year. [3, 4] Despite the city's rapid growth, its publicly released metrics of bus ridership are only available in aggregated form, as systemwide monthly or yearly totals. [5] Although the adoption and widespread use of the ORCA payment card system has created a central store of transactional data across the entire metro system, this only accounts for a fraction of ridership. No models of total ridership currently exist to utilize the available transactional data for predictive and analytical purposes.

This paper documents an attempt to create such a model in partnership with the Washington State Transportation Center (TRAC), and King County Metro. First, we discuss the goals and motivations for the project, and then describe our work creating a data pipeline, training set, and modeling ridership. Next, we describe the process by which we arrived at our final model, analyze the performance and structure of that model, and finally draw conclusions about the way this model, and models like it, could be used in the future.

2. Motivation and Problem Statement

Increasing population in Seattle along with massive highway, light rail, and overall transportation infrastructure projects is increasing congestion into a period known as "period of maximum constraint"[6]. Public transportation, particularly the massive King County Metro (KCM) bus transit system, is crucial in reducing this congestion. However, KCM lacks detailed, accurate measurements of systemwide bus ridership beyond the transactional data from the ORCA farecard system, which is used by a skewed demographic of riders. This project creates a model of total ridership across the KCM bus system based on time of day, route information, weather, and ORCA transactions.

The primary goal of this project is to create a model of total ridership by route by day, with a stretch goal of modeling ridership across all times of day. The model should have predictive power, meaning that for a specific route identifier (a bus number and a direction), number of ORCA transactions, and some information about the day of week / time of day, we should be able to predict total ridership summed over all the stops on that route. Ideally, we would like to extend this model in several other ways:

- Predict usage of specific within-day trips for each route,
- Predict boardings at the per-stop level,
- Use the number of ORCA taps from previous buses on the same route to predict ORCA taps on the next one,

[2] Brockerhoff, 2000
[3] King County Metro, 2019
[4] Balk 2019
[5] King County Metro, 2019
[6] Mike Linbolm, 2018

- Add weather data to the model, allowing planners to estimate the effect of weather events on ridership

We were provided 38.9 million ORCA transactions over two survey periods conducted in the mid-winter and mid-summer of 2019. On approximately 60% of the stops in the set, we were provided additional readings from automated person counters (APC). Neither ORCA transactions nor APC counts provide a perfect picture of ridership: some riders use cash instead of ORCA transactions, and APC has several issues that can contribute to inaccurate counts. However, APC is thought to be the best available unbiased estimate for ridership.

This model, and models like it, could enable transit agencies to make micro and macro planning decisions based on predictive models, and better understand the impact of station closures, headway changes, and weather events. Someday, these models could even be used to dynamically adjust the transit system's operations based on predicted need in real time.

3. Background

Predicting transit ridership is not a new idea, so there are plenty of examples of statistical methodologies for this kind of work. In the following paragraphs we summarize the approach of four different studies which attempted to apply data science methodologies to transit data.

In 2015, four authors from George Mason university used transactional data to create predictive time-series models on the Washington, D.C. metro system. [7] Using ingress and egress data at 15-minute intervals, they were able to use a "bag of words" approach - dimensionality reduction, clustering, and a K-nearest-neighbors classification model - to predict traffic flows for each station in the afternoon, based on that station's performance in the morning, with high accuracy. The authors compared their approach to a simplistic deep learning model, but found that the nearest neighbors approach performed better in cases where the input was out of the standard weekday commute pattern. This paper suggests that we might be able to use PCA and naive clustering as a quick way to understand similarities between routes and stops. The result of that clustering could be a very effective predictive variable in a further model. It also demonstrates that PCA can be an effective, computationally simple, and conceptually easy way to find similarities between time-series.

Lijuan Liu and Rung-Ching Chen, in a collaborative effort between Xiamen University of Technology and Chaoyang University of Technology from China and Taiwan respectively, developed a novel method using a stacked auto encoder (SAE) and a deep neural network (DNN) to predict bus ridership. [8] Though they summarize the challenge of predicting ridership on other forms of public transit, including taxis, light rail, and air travel, they focus on bus ridership as it is not often studied. They created four different models, each using a different response variable to model passenger flow through four stations of the Xiamen Bus Rapid Transit (BRT): ticket holders vs. card holders, and direction of travel. Their model inputs were average passenger flow for that time of day, along with information on holidays, datetime, and direction. They used each of these models to predict a different aspect of passenger flow at each of the four stops. The use of SAE followed by a DNN was meant to engineer better features for the

[7] Truong et al 2018

[8] Liu et al 2017

DNN. SAEs promise to find, in an unsupervised manner, labels that indicate how similar certain data points are to others. The authors demonstrated that in almost all cases, their approach was superior to both random forests and DNNs without SAE. This suggests that a SAE / DNN model could also have predictive power on our dataset.

Another group, in 2013, used weather, ticketing and bus arrival data from Lisbon, Portugal to create a Gaussian Process (GP) regression model to predict bus ridership. [9] The authors were able to predict weekly and daily ridership fluctuation based off of historical records as well as additional items such as weather and demographic information. They compared this model to a model they called the baseline that calculated the average number of passengers over a given period of time at a given stop. The results of their study showed a statistically significant improvement of prediction accuracy of the GP regression model over the baseline model. This research suggests that we should be able to apply a GP regression model to our data because we will be using similar data (passenger counts, bus location data, and weather) and the model is flexible enough to add new parameters for better estimating.

A 2019 paper published by a group from the University of Hong Kong explored an Adapted Geographically Weighted Lasso (Ada-GWL) model for station level metro ridership. Ada-GWL is a network-based spatial extension of an ordinary least squares model (OLS) with coefficient penalization (Lasso). Their ridership data was collected using an Automatic Fare Collection (AFC) system which is used throughout their metro network and accurately measures passenger boardings and exits allowing them to successfully model for day of week and time of day.[10] They approached time of day using rush hour versus non-rush hour which might be something we can bring into our model as it also helps account for slight variations in riders commutes. Their model also used features to account for geographical, socio-economics patterns like land use, rail network structure, station's degree of centrality, local population, etc. These are not features we currently have but we might be able to get them from other public records as a way of improving our model or noting future work opportunities.

4. Methods

4.1 Data source

ORCA transactions and APC data were collected from two six-week studies conducted in the mid-winter and mid-summer of 2019. The ORCA passenger count data was collected via card readers on buses and roadsides, while APC sensors automatically collected ingress and egress estimates at the whenever the bus doors opened. APC sensors were distributed on roughly 60% of trips in the system during the study periods, so that every trip in the system was covered by an APC sensor multiple times during the study period.

The APC data has known data quality issues, due to sensing problems like passengers being too close to the sensors, boarding and deboarding when the buses are full to let passengers off, and opening the doors multiple times at the same stop. Additionally it is only available on 60% of trips in the study period. However, it is the only data we have that estimates total ridership including non-ORCA riders, so it has become the quantity we are attempting to model.

[9] Bhattacharya et al 2013

[10] He et al 2019

We were provided with two sources of data. The first, which we have been calling the "APC" file, was a preprocessed file provided to us by Washington State Transportation Center (TRAC), which used Automatic Vehicle Location (AVL) data in concert with APC data to create a table where each line represented a stop on a particular route. The infrastructure and code used to generate this are located at TRAC.

The second file, also provided by TRAC, was a transactional table of ORCA taps. This file contained the same set of route and stop identifiers found in the APC file, as well as many other pieces of vital information. The ORCA payment system is a product of a third-party vendor and its taps are reported to the system along with a device identifier for the receiver on the vehicle (or within a roadside post). Resolving the correct vehicle, route, and direction for each tap is a complex process and was accomplished by a combination of vendor hardware and software, King County Metro infrastructure, and data processing at TRAC.

The experimental periods were between January 7 and March 3, 2019 and the between July 1 and August 31, 2019. Notably, Seattle experienced unusual snowfall during the week of February 3rd through 10th. Preliminary investigation showed an unusual pattern of reduced transit usage during this time; some of which was due to incorrect coding around snow routes. Our goal was to model normal system-wide ridership, so we omitted 10 days (February 3rd through 13th) where it seemed likely that King County Metro was operating on a reduced or snow schedule.

4.2 Pipeline

We created a data pipeline based on exploratory data analysis and conversations with our sponsor. The pipeline first filters the data to the relevant dates, routes, and trips. We removed snow days in the winter dataset as well as any non-bus routes (for example, light rail and water taxi) and routes not operated by King County Metro (the collected data also contained observations from Sound Transit, Pierce Transit, and Community Transit). We also limited our dataset to trips where APC estimates were available. Next, we aggregated the ORCA data so that each row represented one "stop" instead of one transaction. This proved tricker than anticipated, as system inaccuracies sometimes attributed ORCA taps to the wrong direction on a route. After several rounds of refinement, we ultimately aggregated by business date, stop ID, and trip ID, which let us avoid the issue of resolving cardinal direction or inbound/outbound status. "Business date" was the operational date of the trip; for trips that began shortly before midnight, we saw stops logged where "transaction date" and "business date" differed. In all cases, we created aggregates using "business date" to create logically consistent estimates. In this step, we created columns for total orca taps, the fraction of orca taps from cards in various forms of regional reduced fare programs such as seniors, low income individuals, and disabled people. We also created a feature for the fraction of ORCA taps associated with the University of Washington.

Next we performed a similar aggregation on the APC/AVL file. Although each line in this file nominally represented a "stop," we found rows with duplicate metadata, but different APC counts, from cases where the driver opened the doors multiple times to pick up additional passengers. To handle this, we aggregated the APC file so that each line uniquely represented the total APC boardings at each stop. We also removed outliers where APC reported more than 150 boardings at the same stop.

Next, we merged the APC and ORCA data. Because some routes and stops had no ORCA transactions, we used a left join to merge the ORCA data into the APC data, which was guaranteed to contain a complete list of all stops on all routes during the study period.

In initial versions of the pipeline, we also merged in publicly available weather data collected from the National Weather Service station at Boeing Field, though this was later removed when it was not found to contribute substantially to the model.

The merged data was then further aggregated by route, direction, and time in blocks of 15 minutes, 30 minutes, and 1 hour. Each line in the resulting dataset represented the total APC and ORCA boardings across all trips that started within the time block. For example, a single line might represent the total number of ORCA taps and APC boardings for all southbound route 40 trips that originated between 2:15 and 2:30pm on a particular day. We removed lines where the total ORCA taps exceeded the APC boardings, eliminating APC sensor faults from the model. At this point we also created some additional features:

- Day-of-week, indexed from 0 to 6,
- Indicators of cardinal direction
- Whether the trip occurred on a weekend
- The hour of day that the trip originated
- Whether the trip was RapidRide or not

The final product also included categorical data listing the general location of the route as well as the neighborhood that the route started and ended in. [11] We included this information to better identify the geographic regions that KCM served. For example, Route 301 serves the North King County region by starting in Downtown Seattle and ending at the Aurora Village Transit Center.

Finally, at each level of aggregation we randomly split the data into 80% training, 10% cross-validation, and 10% test sets. The test sets were not used until final evaluation.

4.3 Model selection

We attempted a variety of modeling methods to generate a rough impression of the characteristics of the problem space. These models included PCA-and-KNN, Gaussian process regression, linear regression, clustered support vector machines, gradient boosted trees, and a deep neural network. In this initial round, we used data from the winter study only, as summer data was not yet available. We selected mean absolute error (MAE) and mean absolute percentage error (MAPE) as the shared performance indicators for this project; after our first modeling pass we realized that MAPE was not defined (and uninformative) for routes and aggregations where APC counted zero boardings. While this was rare, we found that some cases did exist, and therefore shifted to mean absolute error. The MAE of the model is measured in the difference between the number of predicted and actual boardings for a specific route, direction and time block.

Linear regression was the least powerful estimator, with a mean absolute error of 9.1, while the gradient boosted tree model (XGBoost) was the top performer with a mean absolute error of 6.6. The deep

[11] "List of King County Metro bus routes"

neural network was only slightly more effective than linear modeling. Across all modeling methods, weather data only marginally improved the accuracy of the model. The early success of a tree-based model was not unexpected, as we expected that different routes would behave very differently across various times of day. Similarly, the SVMs attempt to partition the input space into predictive regions could explain the relative success of that model. At the time, we thought that deep learning was less effective because our data was insufficiently sized, and we predicted that deep learning could outperform standard methods when trained on many more examples.

As a result, we decided to concentrate on further improving our SVM and XGBoost models, with the option of re-examining different shapes of neural networks as time allowed. As summer data became available, we also decided to investigate whether we could use both winter and summer data within the same model.

4.4 Model refinement

At this stage, we updated our pipeline to remove weather data (as its predictive power was low) and to process both winter and summer datasets. Subsequent analysis showed that APC counts across all routes and directions were extremely similar when comparing the summer and winter datasets, so we decided to combine the summer and winter data, and add an indicator variable so the model could learn any behavioral differences between the two seasons. After merging the summer and winter data, we resampled new training, cross-validation, and test sets.

Next, we performed manual hyperparameter optimization to see if we could improve the XGBoost and SVM models. Some generalization was achieved with XGBoost by creating shallower trees and tuning various other parameters. Ultimately, depth 4 boosted trees with eta at 0.1 and RMSE loss were used to achieve the results shown in Figure 1. Our initial SVM used a linear kernel and achieved a MAE of 17. Tuning the parameters made no significant improvements. It is likely that a different kernel, like a sigmoid adding a small non-linearity, would have performed better, however the time complexity of these kernels do not scale to a training set of our size. Instead, working off the theory that groups of trips or routes behave similarly, we tried to improve our SVM with clustering. We did a K-means cluster on the training set then trained a SVM for each cluster. As we increased our number of clusters from 1 to 3, we saw our MAE drop to 11.4. We may have seen continued improvement by adding more clusters, but we were already seeing more promising results from the other models.

Additionally, we tried a variety of neural network structures, activations, and optimizers to try and improve the performance of the model. Our working theory was that the problem space was discretized, with some routes and times of day behaving in similar ways - for example, common commuter routes during peaks hours on weekdays, vs. other kinds of routes at other times of day. To this end we examined a variety of dense network shapes with many layers between four and sixty four neurons each, with the thought that a relatively small number of parameters was appropriate to the size of our dataset, but that higher-order interactions might be detectable by a deep network. Ultimately, none of these models produced MAEs under 12, many above 20, which significantly underperformed our best XGBoost and SVM models.

Next, we performed K-means clustering on the training set and used the inertia measure and the elbow method to identify an ideal number of clusters, which we found to be 6. Then, we trained a dense network to reproduce those classifications, which it did with acceptable (>90%) accuracy. Then we added three dense layers to the end of that network and trained it on the prediction task. Unfortunately, this model also underperformed, with a mean absolute error of about 21. We also repeated this experiment with 24 clusters with the hope that the additional information encoded into the classifier layers might yield better predictions; unfortunately this also did not prove to be true. By reducing the number of clusters and layers, the best prediction accuracy we saw using this method was 14.

Out of the many neural networks trained over the course of this investigation, we found that the shallow networks seemed to be performing best. Following that lead, we trained a series of networks that were each shallower and simpler than the last, ultimately arriving at a network with two layers: 64 sigmoid neurons in the first layer, and one linear output neuron in the second. This led to a surprisingly good performance of 7.4 mean absolute error on the cross-validation set, which was in the range of the best models so far.

At this point, we performed a grid search across the type of activation function in the first layer, and found that sigmoid was the best performing activation, even compared to tanh. Linear activation functions were not able to attain nearly the same performance; the nonlinearity in the first layer proved to be powerful. Then we performed grid search on the number of neurons in the first layer. To our surprise, the model's performance increased as the first layer grew from 64 neurons all the way up to 600 neurons, and decreased thereafter. Given that the training set had about 450 features - most one-hot encoded categorical values - we hypothesize that the extensive first layer was using the input weights to understand interactions between categories. After optimization, the best-performing network attained a mean absolute error of 6.39 on the cross-validation set at a 15-minute level of aggregation, which outperformed all other methods. Additionally, this model was relatively simple and easy to fit on commodity hardware in minutes.

At this point we had performed hyperparameter optimization on an XGBoost model, a SVM, and the neural network, and we compared their relative performance on the combined (summer and winter) datas at the 15- and 30-minute aggregation levels. The chart below shows the result of that comparison.

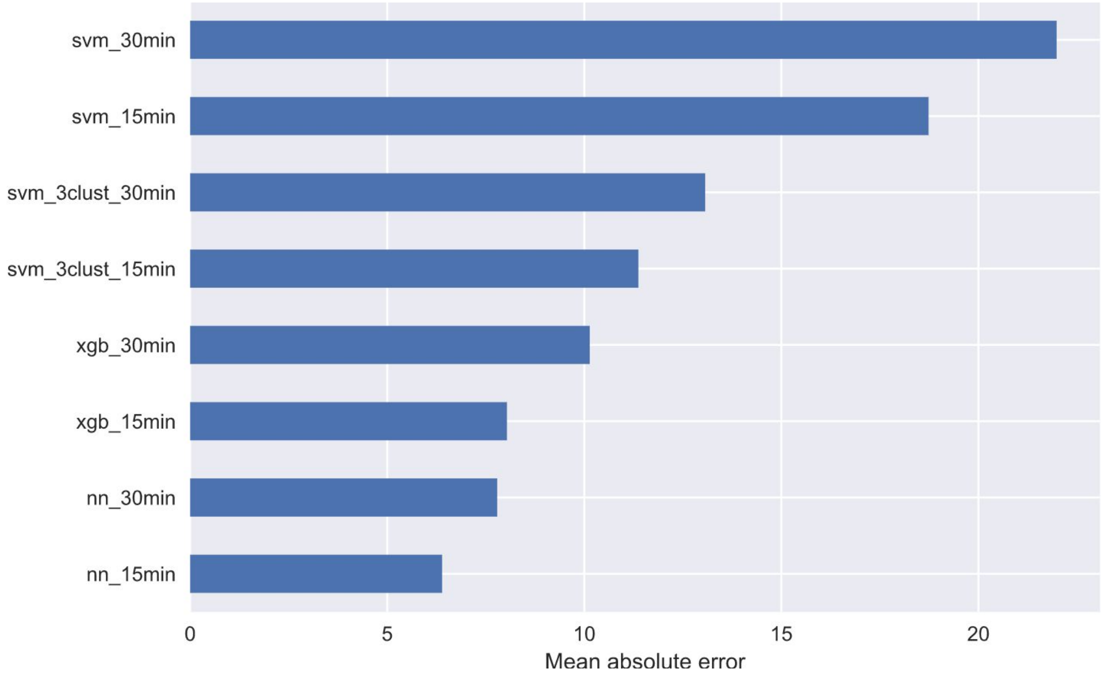


Figure 1: Comparative performance of XGBoost, SVM, and neural network models on the cross-validation set, at both 15- and 30-minute aggregation levels

Additionally, we identified interesting subsets within our data and evaluated the performance of every model within those subsets. Examples include: high, medium, and low frequency routes; morning and evening peak commute times; region; and north-south vs east-west routes. Across all subgroups, the neural network consistently outperformed other models. The worst performing subgroup was almost always RapidRide. At this point, we decided to devote our time and effort to further optimizing the neural network model.

4.5 Model finalization

Next, in consultation with our sponsor we removed RapidRide routes from our dataset. While RapidRide (BRT) is important and widely used in the region, and we would like to be able to someday model RapidRide ridership, RapidRide customers tap their ORCA cards onto roadside posts before they board buses, which makes it much harder for TRAC to link those taps to a specific bus and direction. The data are messy and TRAC suspects that other systematic errors might be present in the RapidRide data. Our theory was that the "noise" in the RapidRide data was confusing the model's ability to predict other high-frequency routes. To test this, we compared the performance of two models on high-frequency, non RapidRide routes, which are perhaps the most important subgroup to model accurately since they represent high frequency, high utilization routes. We found that the model's predictions for the high-frequency subgroup improved from a mean absolute error of 9.32 to 5.98 (cross-validation performance, 15-minute dataset) when we removed RapidRide data from the training and evaluation sets.

In previous versions of the model, all times were encoded as categorical variables. For example, the period between 6:00 and 6:15 AM was categorized as "06_15", and in the neural network's one-hot encoded training set, we would observe a "1" in the equivalent column for that category. In the final version of the model, we additionally encoded the hour as a numeric variable, on the hope that the model might be able to achieve some more predictive power by differentiating time periods in a continuous numeric fashion as well as a categorical one.

Next, we used Bayeisan optimization to further tune the learning rate, dropout rate, and number of neurons in the first layer. This method is known to be effective in situations where there are many parameters to tune, but each evaluation is computationally expensive. We used cross-validation performance as optimizer's cost function, which led to some interesting behavior; because we enacted early stopping whenever the cross-validation performance increased, the optimizer found solutions that slowed down the learning rate considerably, because it was able to attain slightly better performance with more epochs at lower learning rates, without "overshooting" the best solution to the point of overfitting,

Interestingly, with a numeric "hour" variable, the best-performing size of the first-layer jumped to 1020, with a learning rate of 0.16 (SGD / Nesterov with a decay of 1e-6) and 23.5% dropout between the first and the output layers. This model had a cross-validation mean absolute error of 5.5074. We then evaluated it against the test set, which was totally unseen by any model to this point, and observed a mean absolute error of 5.5077. This became our best performing model, and we have saved its structure, weights, and the Python objects used to encode and categorical variables and scale the numeric variables on the github repository associated with this project.

Interestingly, we tried combining the training and cross-validation sets to train a variant of this model, in the hopes that the additional data could further improve our performance. When evaluated against the test set, this model did not perform as well as the best model trained on the training set alone; however, we did not use Bayesian optimization with this version of the model, in the goal of avoiding overfit to the test set. While it's likely that after further optimization, this model might yield even better performance, in the absence of any other held-out data, running an optimization algorithm using evaluations against our test set seemed like it would put us at too much of a risk for overfitting, which we would not be able to subsequently detect.

5. Results

With our final model selected and predictions obtained, we again performed a subgroup analysis to identify the times and routes where the mode performed best and worst. In general, the model performed worst on east-west routes, and during the evening peak, when ridership is high and models' residuals were larger, with a mean absolute error of approximately 7. The model performed best during weekend peak hours, and in the North King County region, where the MAE was approximately 3.6.

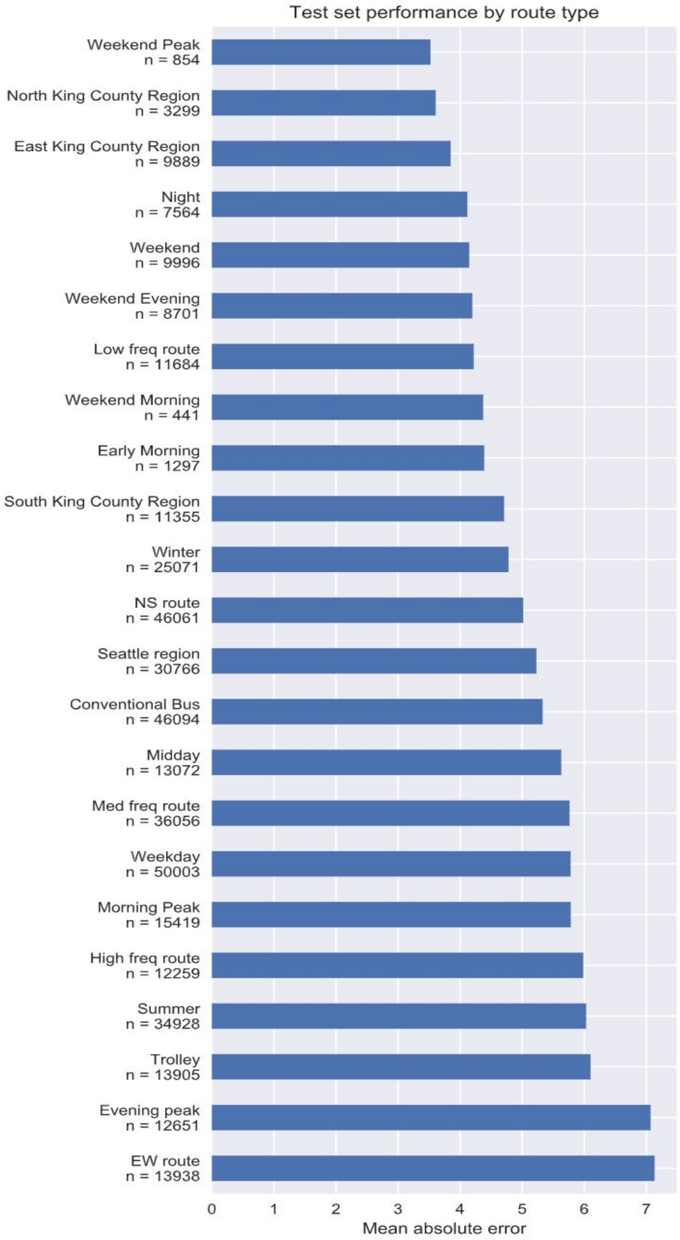


Figure 2: Model performance by subgroup, 15-minute aggregation, test set

We visualized the residuals on weekends and weekdays separately, given that they have a demonstrably different ridership pattern, in the below figures:

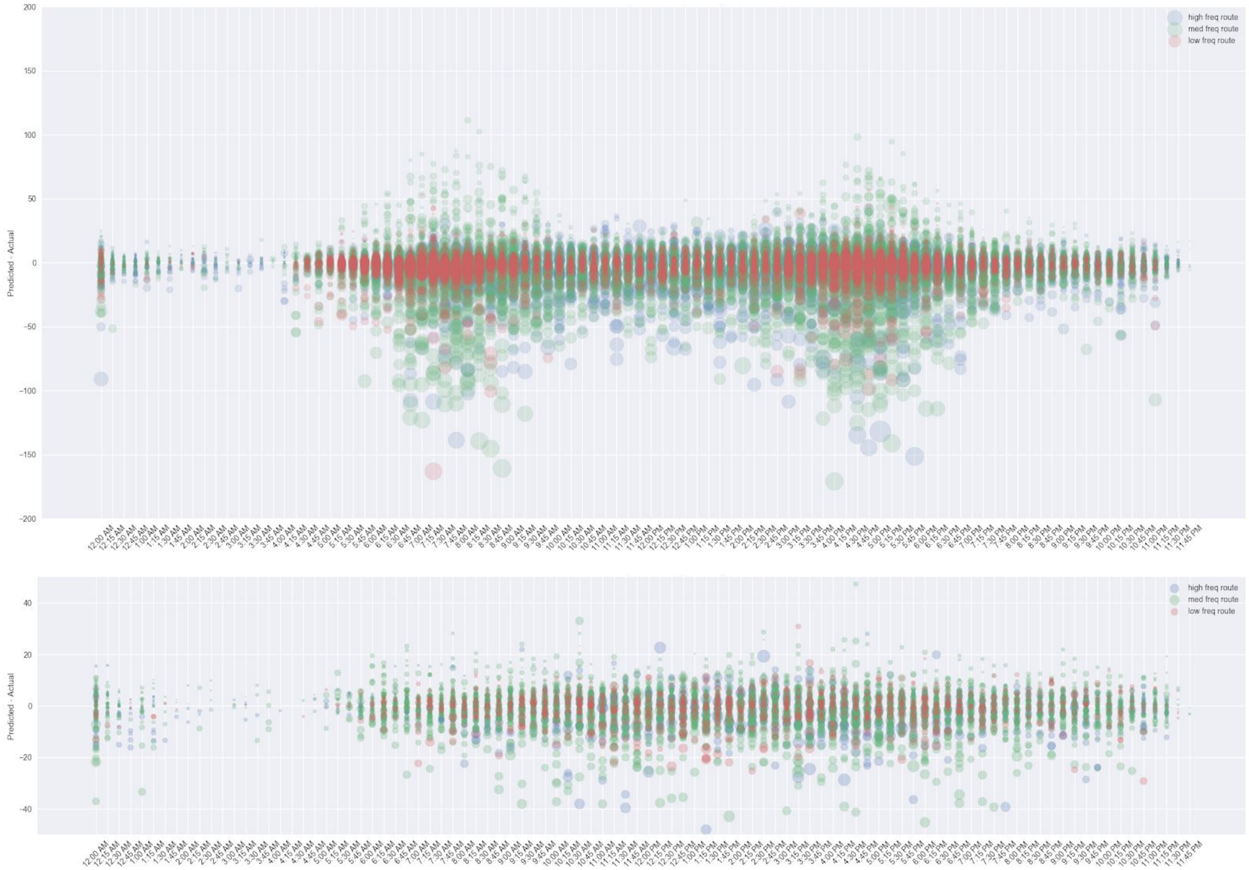


Figure 3: Test-set residuals (15-minute aggregation) plotted over time of day, weekday (upper figure) and weekend (lower figure). [12] High, medium, and low frequency routes are encoded in blue, green, and red respectively; points are sized by the number of APC boardings (the dependent variable).

In the residual plots, we observe a negative bias during peak hours; the model tends to underpredict the number of riders during the peak times for high-frequency routes. This could be the result of sensing problems, as APC estimates can become inflated under crowded conditions, or it could be a result of specific routes that the model has difficulty predicting.

5.1 Model analysis

To try and further understand the performance of this model, we analyzed its structure. Our working hypothesis is that each neuron in the first layer learned a set of weights that created a unique combination of the categorical and numeric features, representing a latent feature. In fact, this is similar to "autoencoding." By subtracting the bias terms from the first layer input weights and re-applying the sigmoid function, we were able to assess and visualize the effect of each input feature in each neuron on the model's predictions. Here we selected the top 50 positively and negatively contributing neurons and

[12] These can be viewed in greater detail on our github repository: weekday, weekend.

plotted the effect of each feature on the neuron's prediction.

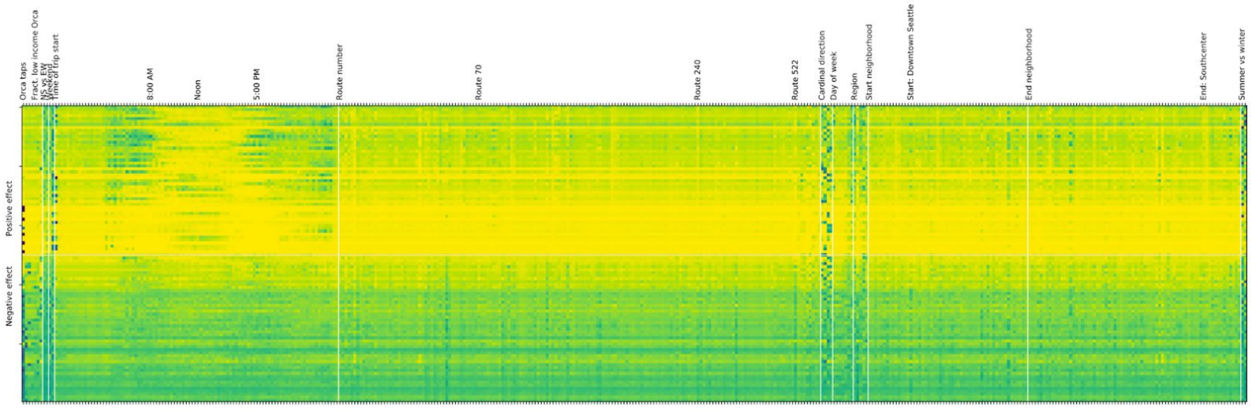


Figure 4: Prediction contributions by feature among the top 50 positively and negatively weighted first-layer neurons. [13]

From this, we can observe the model learned to differentiate peak times on weekdays and weekends, and subtle interactions between start and end neighborhoods, route number and direction, and season. The morning and evening peaks are clearly visible among the top-contributing neurons, and the weekend peak is similarly visible. While this analysis is not conclusive, it suggests that the model is performing as intended, in using a variety of factors to predict actual ridership from ORCA transactions.

In addition to understanding the model's structure, we evaluated our model against its ability to predict the total ridership by day of the week. Currently, TRAC attempts to estimate ridership by day of week, but they would like a better method for doing this. Our method averaged the number of boardings by day of the week. Using Monday as an example, we totalled the number of boardings across all Mondays, and divided it by the number of Mondays in the data (15 Mondays) to get the average number of boardings on a Monday. With this method of evaluating the model, we typically underpredict the average number of boardings by day by 5%.

[13] This can be viewed in greater detail on our github repository.

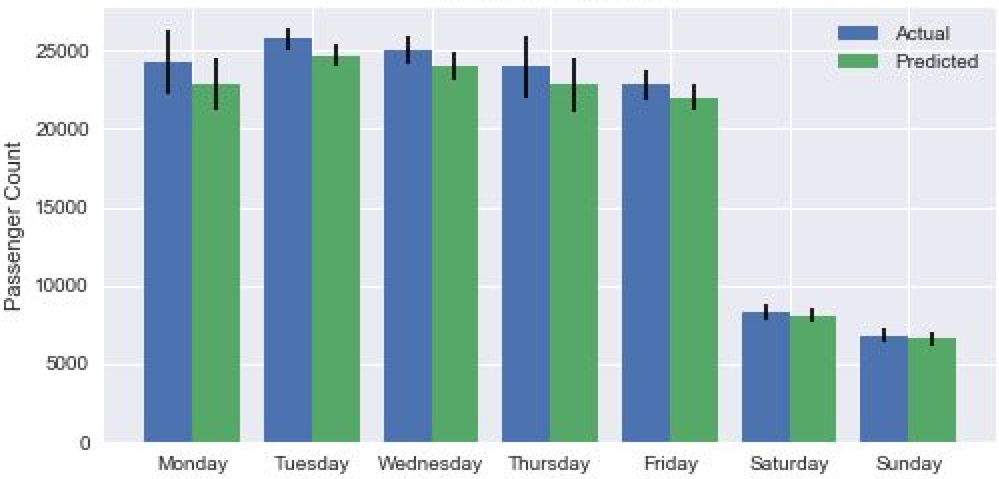


Figure 5: Actual vs Predicted Total Ridership by Day of Week (15 minute - test set aggregation) with a 95% confidence interval around the mean ridership by day.

After understanding the predictive power of our model, we also wanted to perform a bias analysis of our model. For each route we calculated the average difference (predicted - actual), and plotted the distribution by frequency. We then performed a one-way ANOVA test to understand if the mean difference of our frequency clusters were the same. This yielded a p-value of 0.005, supporting that the frequency clusters are treated differently by the model. From here we did a multiple comparison of means to understand how the clusters are different, which showed that the high frequency routes are different than the low and medium frequency clusters. Looking at the distribution below, this is likely because there were few high frequency routes for the model to learn from. It is also possible that high frequency routes behave in significantly different ways, due to their reduced headways and/or potential bus overcrowding at peak times.

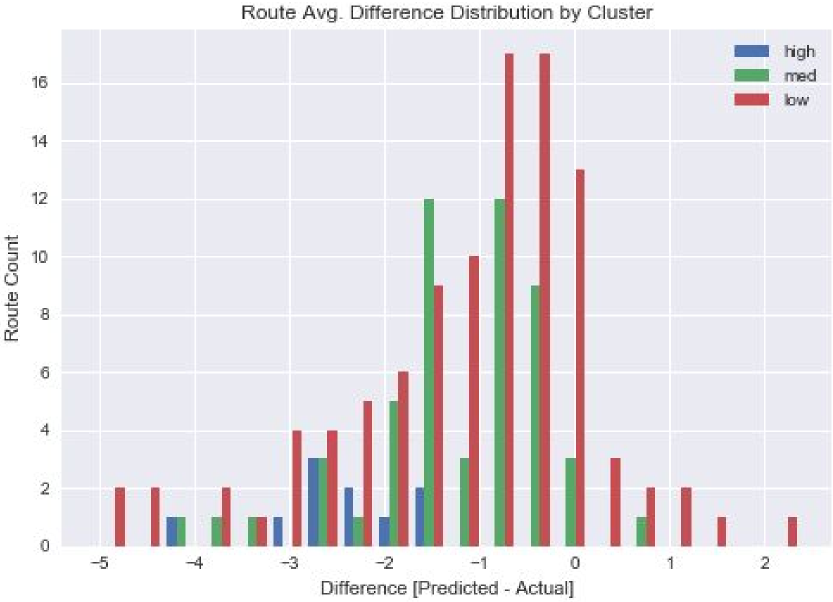
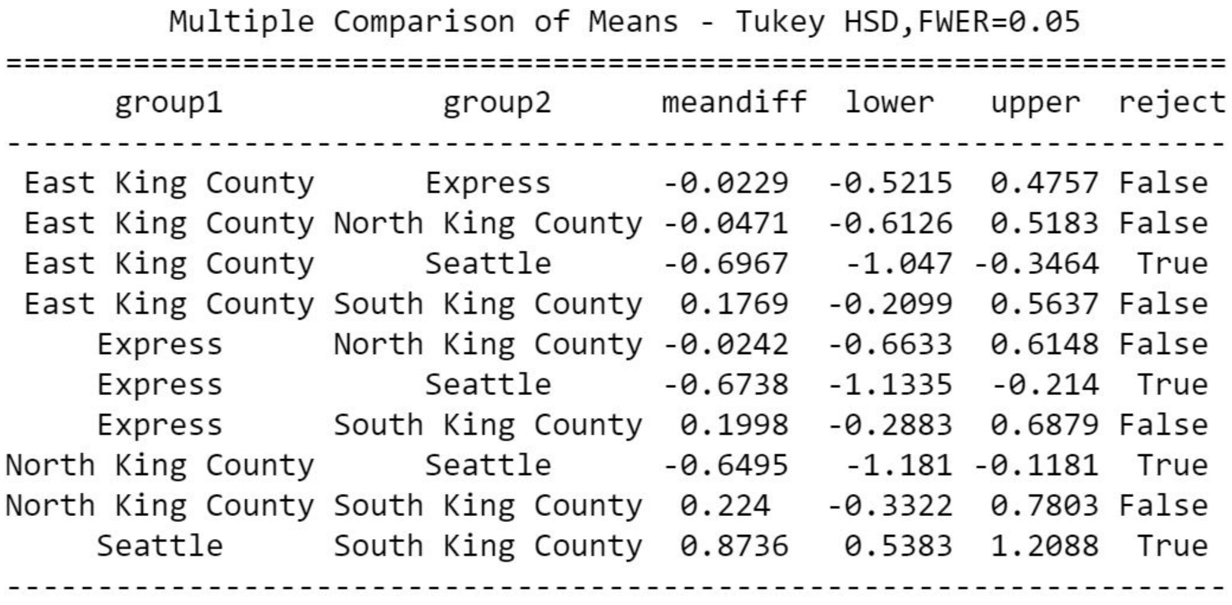


Figure 6: Average residual distribution by route frequency (15 minute aggregation, test set)

Then we performed the same analysis steps with region (Express, North King County, South King, County, East King County, and Seattle), as understanding the regional bias of this model is important. This analysis suggests that the model underpredicted in Seattle, in a way it didn't in other regions. Seattle is also the only region that contains high frequency routes. We removed high frequency routes from the analysis to determine if there was a causal factor, but again found that Seattle's mean to be different than the other regions.



```
         Multiple Comparison of Means - Tukey HSD,FWER=0.05
==================================================================
     group1          group2      meandiff  lower   upper  reject
------------------------------------------------------------------
 East King County      Express      -0.0229  -0.5215  0.4757 False
 East King County North King County -0.0471  -0.6126  0.5183 False
 East King County      Seattle      -0.6967   -1.047 -0.3464  True
 East King County South King County  0.1769  -0.2099  0.5637 False
     Express      North King County -0.0242  -0.6633  0.6148 False
     Express           Seattle      -0.6738  -1.1335  -0.214  True
     Express      South King County  0.1998  -0.2883  0.6879 False
North King County      Seattle      -0.6495   -1.181 -0.1181  True
North King County South King County  0.224   -0.3322  0.7803 False
     Seattle      South King County  0.8736   0.5383  1.2088  True
------------------------------------------------------------------
```

Figure 7: Multiple Comparison of Means for route regions (15 minute - test set aggregation)

6. Conclusion

We have demonstrated that we can accurately model actual bus ridership from ORCA transactional data along with some basic features regarding time of day, season, and start and end locations for various bus routes. The relative effectiveness of SVMs and Boosted Trees compared to simple neural networks imply a relatively intuitive understanding how various ORCA taps interact with our other features. Overall, we achieved an effective solution to imputing bus ridership on routes without APC.

The next steps for this project involve using this model for planning and optimization of bus routes. Additionally, once TRAC is able to provide their estimations for total ridership, it would be beneficial to compare the method we developed and the one currently in use. We also see an opportunity for improving the model by using different datasets to address various opportunities in improving transit in the King County region. Additional data includes:

- RapidRide - this was excluded due to road site taps being difficult to assigned to a route
- Additional transit agencies such as Community Transit, Pierce Transit, and Sound Transit
- Additional season and years. We only had samples from Winter and Summer of 2019
- Additional socio-economic and geographic data

In general, our team set out to determine if it was possible to model ridership across an entire transit system using only transactions and route metadata. We propose that these models are not only possible but, using the neural network structure we have identified, can be made relatively simple and do not require vast investments in hardware. We hope that by using this model and models like it, city and transit planners will be able to better understand and plan for transit needs in our quickly growing city.

References

Balk, Gene. “Big-City Growth Slows across U.S. - but Seattle Still Ranks No. 2 in 2018.” The Seattle Times. The Seattle Times Company, May 22, 2019. https://www.seattletimes.com/seattle-news/data/big-city-growth-slows-across-u-s-but-seattle-still-ranks-no-2-in-2018/.

Bhattacharya, Sourav, Arto Klami, Santi Phithakkitnukoo, Marco Veloso, Petteri Nurmi, and Carlos Bento. “Gaussian Process-Based PredictiveModeling for Bus Ridership.” In *UbiComp13 Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing; September 8-12, 2013, Zürich, Switzerland*, 1189–97. ACM, 2013. http://www.ubicomp.org/ubicomp2013/adjunct/adjunct/p1189.pdf.

Brockerhoff, Martin. An urbanizing world. Population Reference Bureau, 2000.

Liu, Lijuan, and Rung-Ching Chen. “A Novel Passenger Flow Prediction Model Using Deep Learning Methods.” *Transportation Research Part C: Emerging Technologies* 84 (2017): 74–91. https://doi.org/10.1016/j.trc.2017.08.001.

“Ridership Annual Performance Measures.” Ridership Annual Measures - Accountability Center - King County Metro Transit - King County. King County Metro, August 1, 2019. https://kingcounty.gov/depts/transportation/metro/about/accountability-center/performance/ridership/annual.aspx.

Truong, Robert, Olga Gkountouna, Dieter Pfoser, and Andreas Züfle. “Towards a Better Understanding of Public Transportation Traffic: A Case Study of the Washington, DC Metro.” *Urban Science* 2, no. 65 (July 2018): 1–21. https://doi.org/10.3390/urbansci2030065.

He, Yuxin, Yang Zhao, and Kwok Leung Tsui. “An Adapted Geographically Weighted Lasso (Ada-GWL) Model for Estimating Metro Ridership.” Cornell University, April 2, 2019. https://arxiv.org/abs/1904.01378v1.

Mike Lindbolm. “If You Think Seattle Traffic Is Bad Now, Just Wait until All These Projects Start.” *The Seattle Times*, The Seattle Times Company, 7 Jan. 2018, projects.seattletimes.com/2018/one-center-city/.

"List of King County Metro bus routes" *Wikipedia: The Free Encyclopedia*. Wikimedia Foundation, Inc. 20 September 2019. Web. 2 Feb. 2020, en.wikipedia.org/wiki/List_of_King_County_Metro_bus_routes

Author Biographies

Hannah Murphy

Hannah is an Industrial Engineer at Boeing Commercial Airplanes specializing in Production System Analysis and Design where she develops various methods to analyze and optimize the manufacturing of commercial airplanes. Hannah earned a Bachelors of Science in Industrial Engineering at the University of Washington, and is currently pursuing a Masters of Science in Data Science at the University of Washington.

Alex Van Roijen

Alex is a second year full time MSDS student with past internships and research positions at the national renewable energy laboratory (NREL), Air Force Research Laboratory (AFRL), IBM, Microsoft, QueBit, and WSDOT. Alex brought in this project after working on a project on the I-405 and analyzing what kinds of drivers use the High Occupancy Toll lane on the I-405. Alex looks forward to working on more data science projects that have immediate real world impacts.

Maggie Stark

Maggie is a Masters of Science in Data Science prospect at the University of Washington, where she also earned a Bachelors of Science in Physics. She currently works as a professional data analyst focused on optimizing the supply chain, with past professional analytical work focused on developing Arctic ice melt models. Maggie is passionate about using data and the scientific theory to better understand environmental and social issues.

Jacob Warwick

Jacob earned a Bachelors of Science in Statistics and a Bachelors of Arts in Political Science from the George Washington University, where he pursued graduate level coursework in statistical modeling and Bayesian inference. He has professional experience modeling issue support in the U.S. political electorate, and has worked as a software engineer at Adaptive Biotechnologies for the last four years. At the University of Washington he has pursued elective coursework in high performance computing and bioinformatic algorithms.