Predictive Analytics for Metro Bus Arrival Times

UNIVERSITY of WASHINGTON

Vicky Tseng, Aman Ankit, Joey Moran, Brad Luong, Gol-Dann Slater, Muse Wu, and Kai Tan

King County METRO — Moving forward together

With special thanks to advisors Dr. Yi Mi, Al-Ahad Ekram, Elizabeth Love, and Patricia Buchanan

Problem

King County Metro (KCM) bus arrival time estimations are often inaccurate.

Even though apps like OneBusAway and Google Maps are widely used, they still rely on sourcing their data from the King County Metro data centers.

Current System

The current system uses a linear model which calculates the arrival time by using the distance to the bus stop and speed of the bus. The bus relays its location and speed to the transit data center every 90 seconds, which sometimes can be unreliable.

Time to arrive = Distance remaining / Current speed

122+ million	**Annual ridership on Metro Transit**
23%	**Of buses are late**
4.7 million	**New Seattle transit riders in 2017**
2/3	**Of non-riders are more likely to ride with real-time updates[1]**

1.https://www.kingcounty.gov/~/media/depts/transportation/metro/accountability/reports/2018/2018-rider-non-rider-survey-final.pdf

Constraints

While we can neither enforce a change in driver behavior nor change real-world events that influence the arrival time, we can improve the experience by increasing the transparency of bus arrivals and communicating the updates to the rider in real-time.

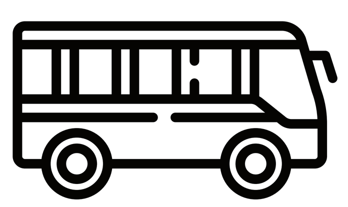
Objective

Given the constraints and problem, our objective is to ***improve the data quality at King County Metro data center by building a scalable machine learning model that can accurately and dynamically predict when the bus will arrive to any bus stop.***

Data Collection and Cleaning

What data was collected?

Bus telemetry data *(from KCM)*



- *scheduled vs. actual times for arrivals/departures*
- *passenger onboarding/offboarding count*
- *nominal and actual meters travelled*

Traffic data *(from SDOT)*

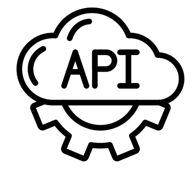
- *traffic volume averages and peak traffic in the AM and PM for each street*

Street attribute data *(from SDOT)*

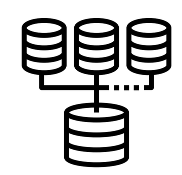
- *road slope, width, length, speed limit, material*
- *highway or not highway*
- *one-way or not one-way*

How did we clean the data?

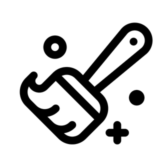
We created a Python program...



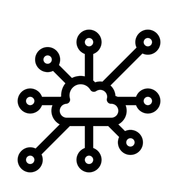
...to automatically source SDOT traffic and attribute via API



...to combine telemetry, traffic, and street attribute data



...for data cleaning and transformation



...with scalability in mind

Feature Selection

Methodology

Filter Based Feature Selection (FBFS)

What is FBFS?

Identifies features (variables) in the dataset that have the greatest ability to predict the target column: elapsed travel time between two stops.[2]

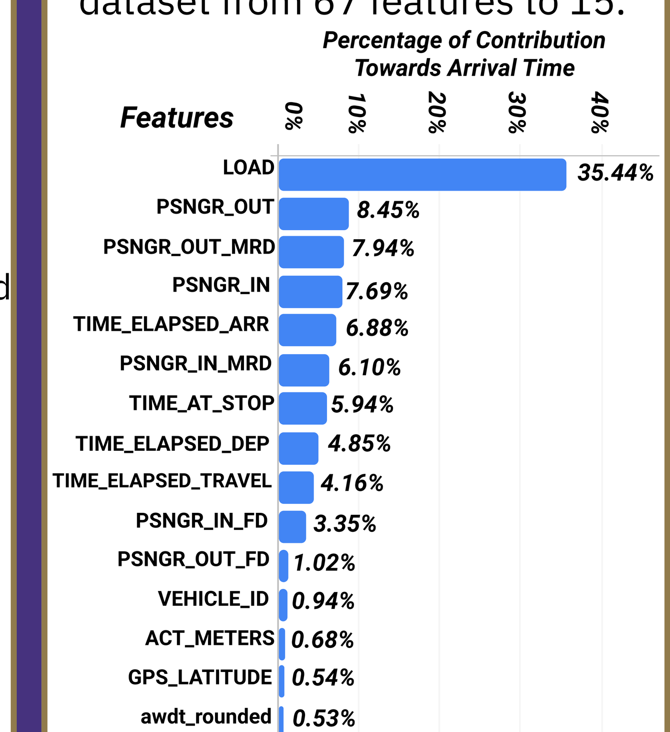
Why use FBFS?

Using identified features allows the model to

- have higher performance and accuracy
- limit the required computational resources

FBFS Results

Using the Mutual Information scoring criteria we filtered our dataset from 67 features to 15.



2.https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/filter-based-feature-selection

Clustering

Methodologies

1. Dynamic Time Warping (DTW)
2. Permutation Distribution Clustering (PDC)

What is DTW and PDC?

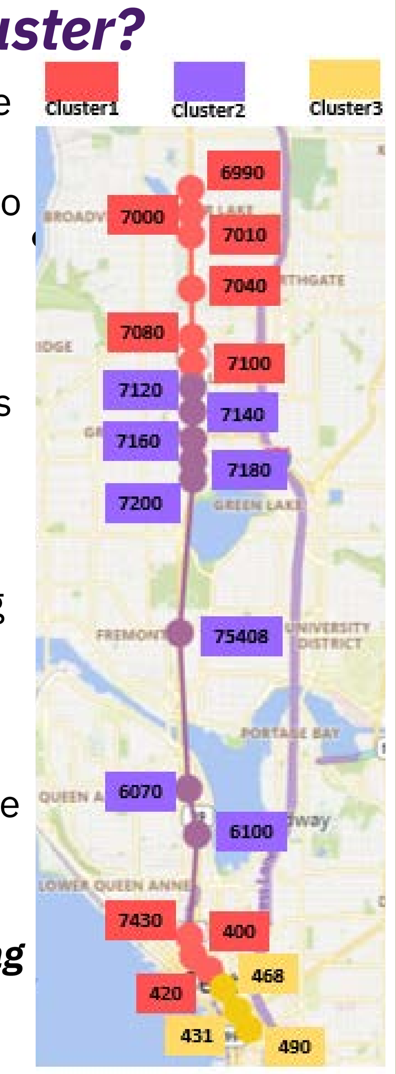
finds similarity between two time series

DTW	PDC
1. aligns time series graphs using warping algorithm 2. calculates the **DTW** distances	1. finds the permutation distributions for time series graphs 2. calculates the **Hellinger** distances

3. outputs a comparison distance matrix to then perform agglomerative hierarchical clustering

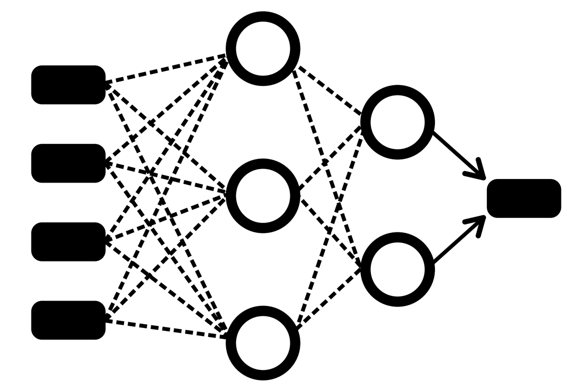
Why cluster?

Along any route there are many different regions, from downtown streets to the highways to the neighborhoods. We did not want to use predetermined segments but instead wanted the clustering algorithm to self-generate segments using the highest scoring variable from FBFS (LOAD) ***to provide absolute objective results***. Dividing the route into clusters allows for the model to treat these roads differently ***resulting in a better predicton accuracy***.

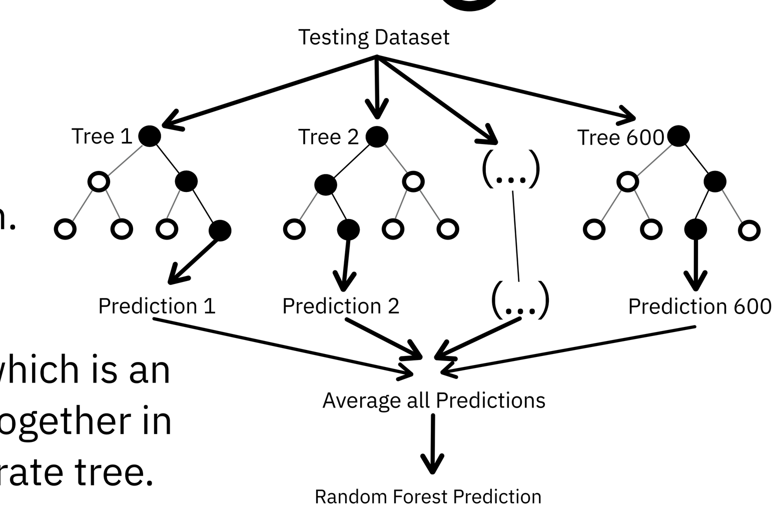


Predictive Modeling

Methodologies



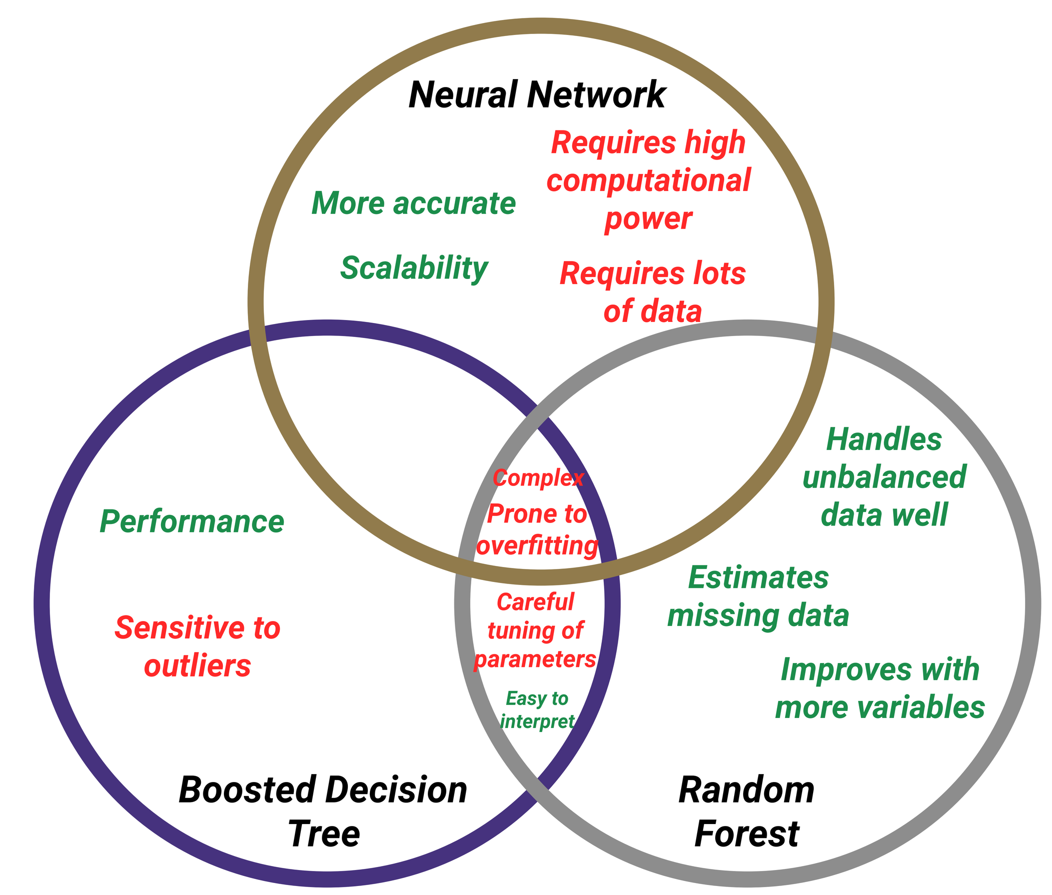
1. Neural Network Regression (NNR)
Mimicking biological neurons, neural networks develop layers of simple operations. With lots of data, these operations are trained to link to one another and strengthen correct chains of logic resulting in a prediction.



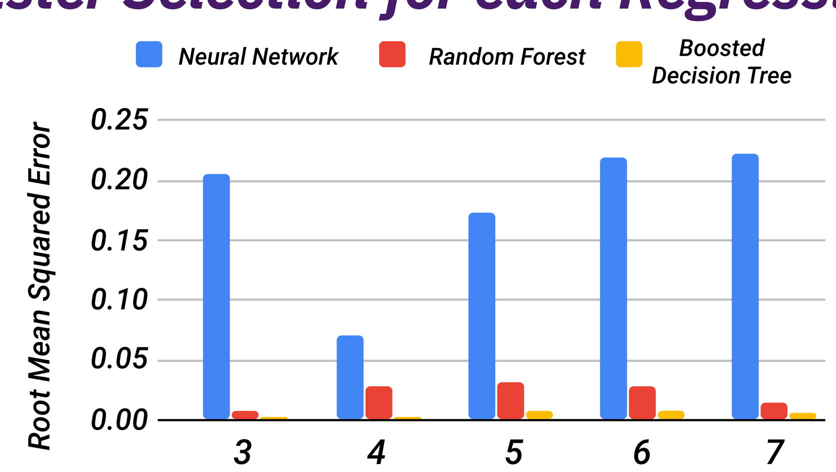
2. Random Forest (RF)
Averages the results of many decision trees to make a prediction.

3. Boosted Decision Tree (BDT)
Same as RF but adds "boosting," which is an algorithm that adds weaker trees together in order to create a larger, more accurate tree.

Which model should we use?



Cluster Selection for each Regression

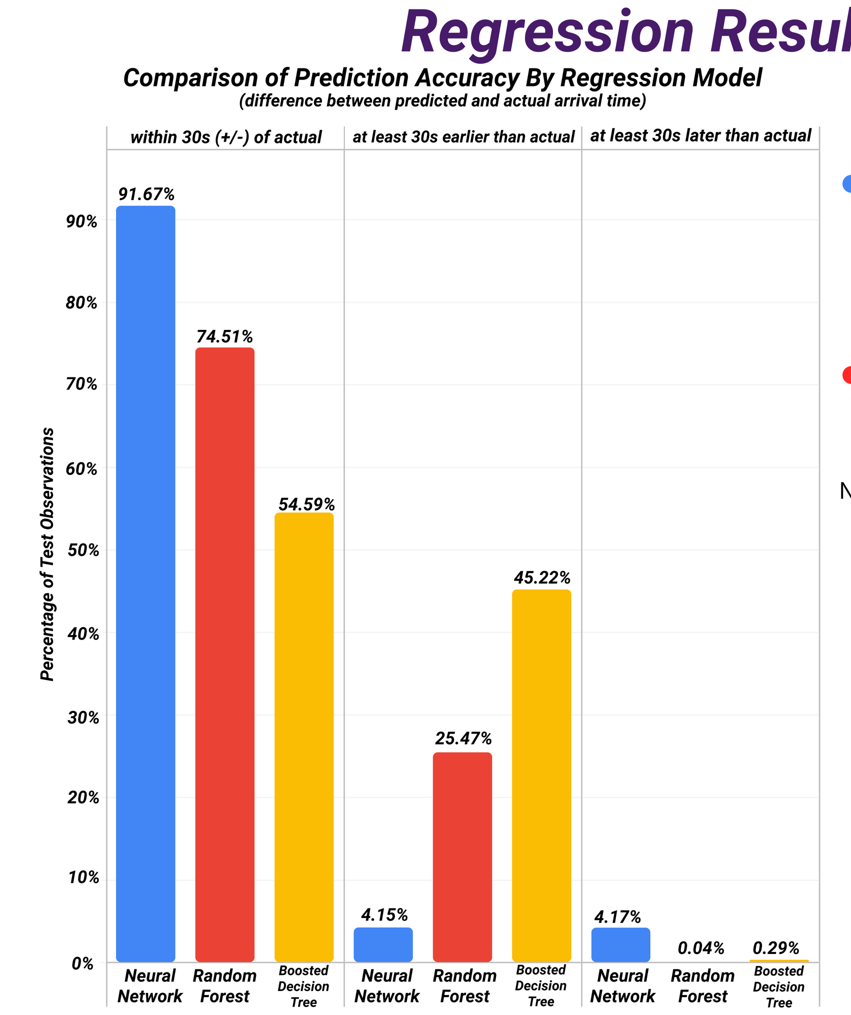


Measure of model accuracy (lower is better) using

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted_i - Actual_i)^2}{N}}$$

This graph is only used to determine which cluster to select for which model. It does not translate to measure performance on unseen data. We choose the cluster set with the lowest RMSE for each regression.

Regression Results Incorporating 3 Models



Our results indicate that

- ***More than 95%*** of bus travel time within one minute and at least 30s earlier than actual for Neural Network.
- ***More than 99%*** for both Random Forest and Boosted Decision Tree.

Note: Results was performed on the validation set, which had around 91,000 data points.

Impact

74+%	**More accurate than current system**
22.9 million+	**More riders given on-time estimates**
6722	**Public Transit Systems in the US can incorporate our work in their data centers[3]**

3.https://www.apta.com/wp-content/uploads/Resources/resources/statistics/Documents/FactBook/2018-APTA-Fact-Book.pdf

Moving Forward

From our final report, code documentation, and models, our project will be applied to other routes within KCM service.

Current System — No utilization of ML → **Methodology** — Create a pipeline that retrains new ML models annually using our methodology → **Ideal System** — Arrival times are always up to date