Predicting Duration of Bike Rental

Jessica Lauzon, Jayant Mukhopadhaya
{jlauzon, jmukho}@stanford.edu
CS229 Machine Learning, Stanford University

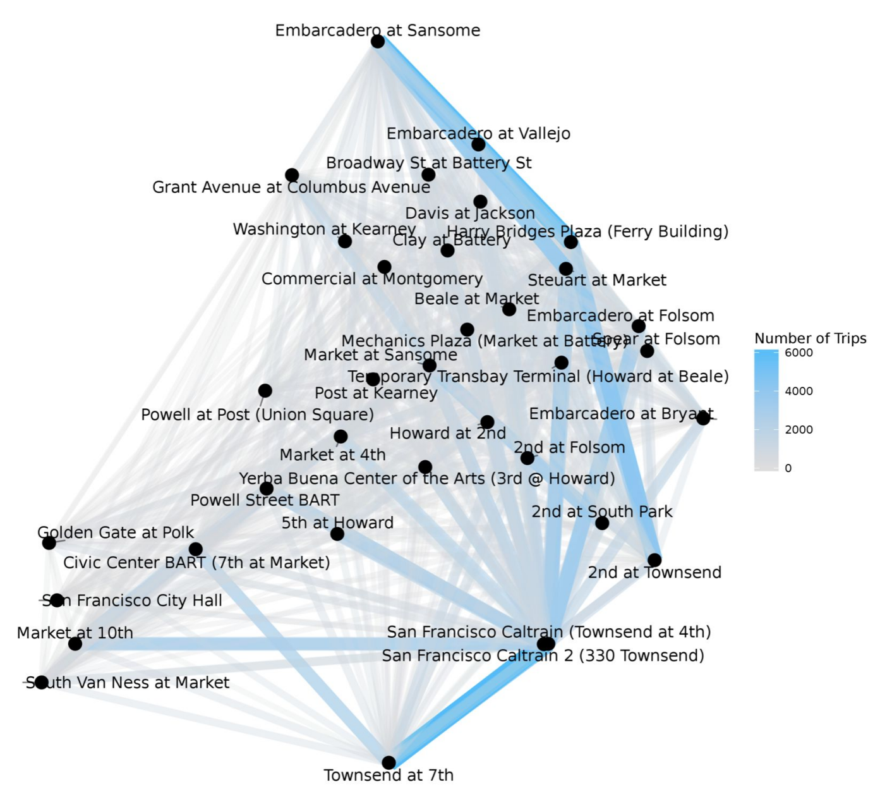
Motivation

Cycling is a cheap and environmentally friendly mode of transportation. Bike sharing projects aim to provide this convenience by renting bikes in various, easily accessible locations. These locations have limited numbers of bikes and limited numbers of docking stations. By predicting trip duration based on a variety of factors, we hope to inform the future installation and expansion of the bike sharing programs.

Data

The dataset came from the Kaggle website for the SF Bike Share Project. The project rents bikes for trips lasting no longer than an hour.

The data (2015) contains rental station information, bike availability, weather, individual bike trip data, and **duration** of each trip, which was used as the ground truth. In total, 397688 rows and 11 columns of data were found to be useable.



https://www.kaggle.com/benhamner/sf-bay-area-bike-share

Feature Selection

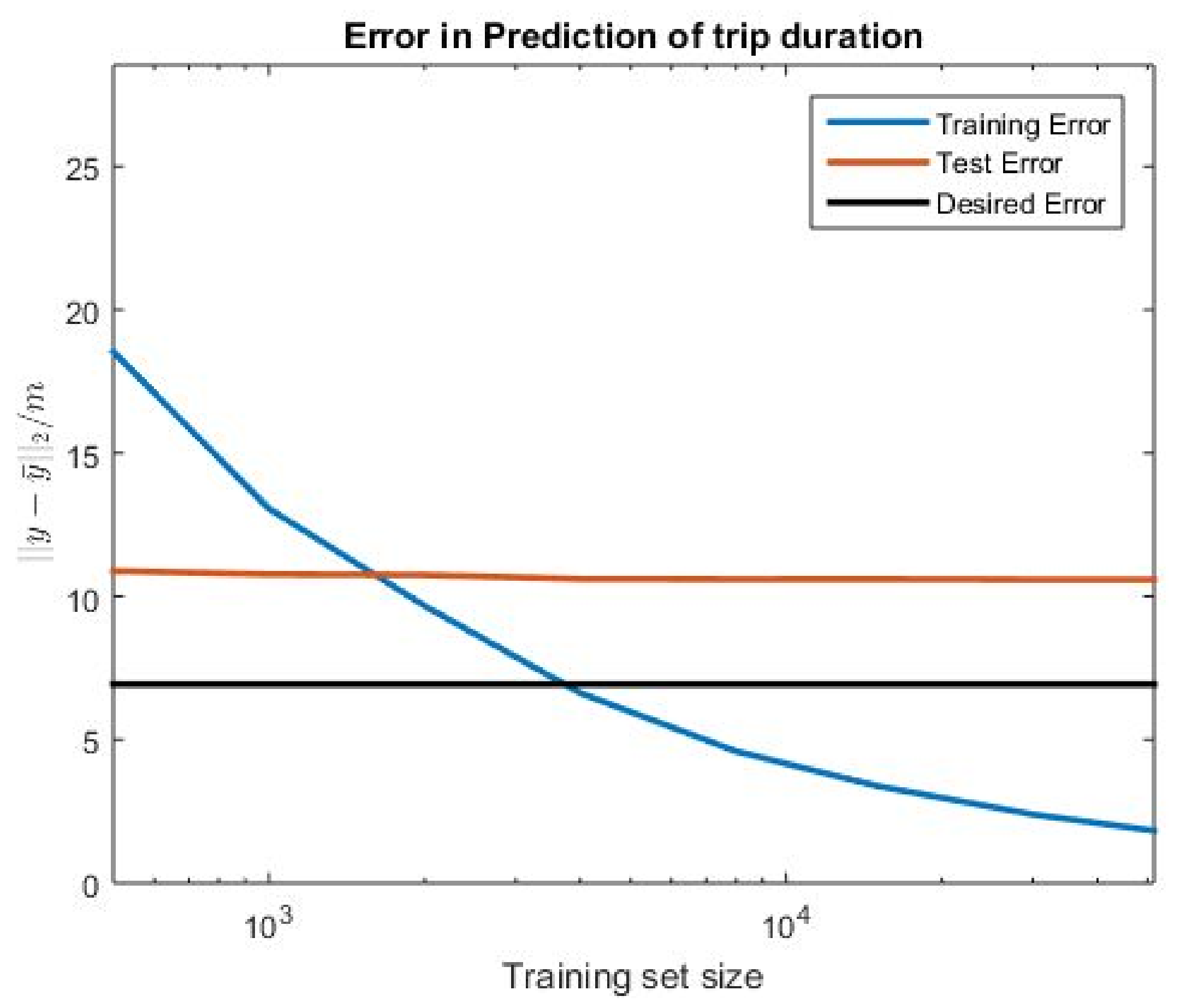
Features were selected based on what would reasonably affect the duration of a bike ride. This includes weather, start time, day of the week, and the proximity of the rental station to other stations.

For linear regression, to ensure a linear relationship between the features and the output, much of the data needed to be preprocessed. This meant classifying some features into "buckets," combining features such as station locations into usable information, and so on.

For the neural net, while the data needed to be in numeric form, the linear relationship was not necessary and the buckets were condensed into single features, with the exception of the truth value for purposes of classification.

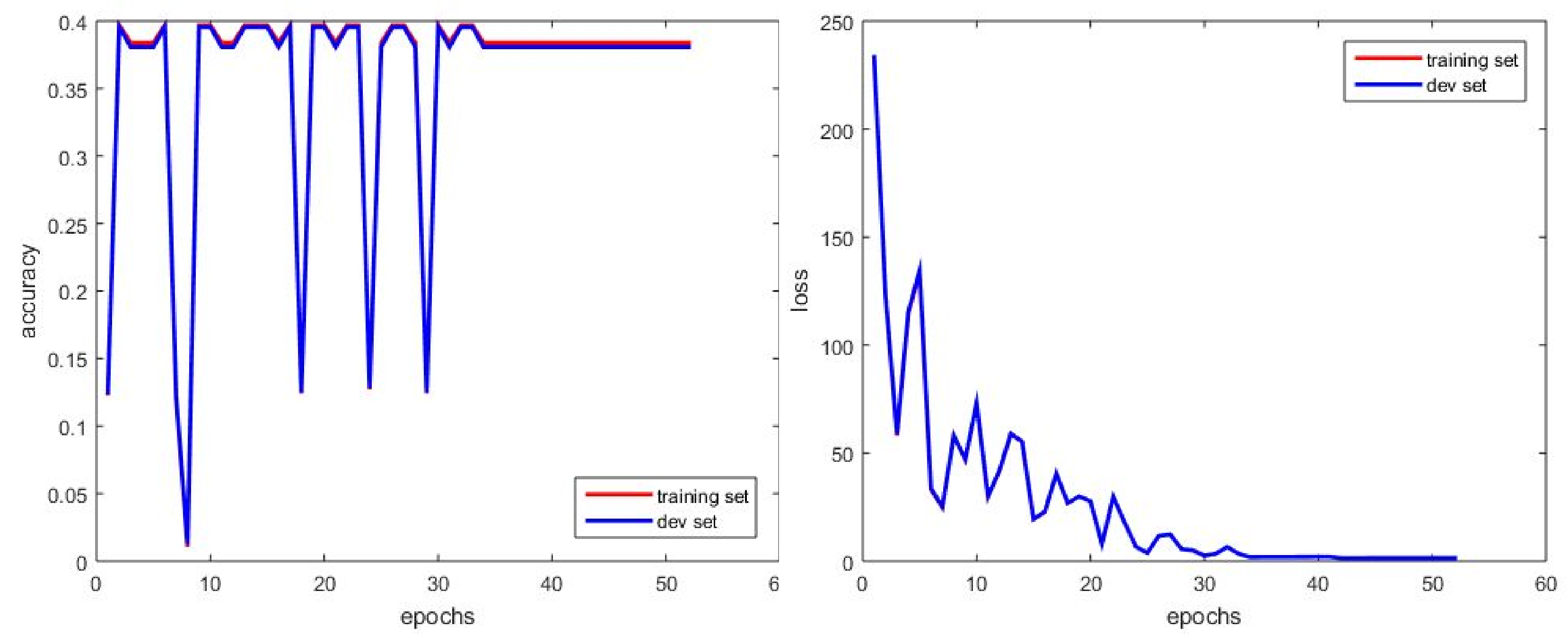
Results from Linear Regression

The aim was to reduce our test error below 1% of the average trip duration, which in this case is about 7 seconds. The error is calculated as the L2 norm of the difference between the true duration, and the predicted duration. In the plot below, we see the training set error reduce with larger training sets. The Dev error does not seem to improve with training set size. The final test error was 10 seconds, which is higher than the desired error. This prompted an attempt to use regularization to bring down the test error. Unfortunately, that did not improve the performance of the model. Using higher order feature mapping would be the next step in improving this model.



Results from Neural Networks

To avoid discarding data by projecting onto a lower dimensional space, a neural network was implemented. The quantity of interest, the ride duration, was bucketed into 5 minute increments, and a softmax regression output layer was used. Different architectures were attempted but a simple 1 hidden layer, with 300 units was used. From the plots, we see the loss reduce with Epochs, but the accuracy does not improve. This is because the neural network outputs the same result for each data point.



Methodology

Linear Regression

- The projection of the raw data into a lower dimensional space to ensure linearity led to identical instances of examples. To keep the parameter matrices well-conditioned, we only kept unique instances.
- The train, dev, and test set proportions were 80%, 10%, and 10% respectively.
- The closed form solution of Linear Regression was used:

$$\theta = (X^T X)^{-1} X^T \vec{y}$$

Deep Neural Net Classification

- Implemented in TensorFlow and MATLAB.
- We chose to structure the neural net as a classification problem and binned the duration truth values into eleven 5-minute intervals.
- Classification was chosen instead of regression due to easier debugging of the network in TensorFlow.
- Train, dev, and test set: also 80%, 10%, and 10%.

Conclusion

The results for both models show that further improvement is needed.

- In general, usage of the raw data could be explored by adjusting features and assumptions about the inclusion and exclusion of certain parameters.
- Linear regression may be improved by using weights (weighted linear regression).

Neural Net:
The model was subject to underfitting, which meant that the predictions were the same for all training examples. The steps that were taken to mitigate this were:

- A large percentage of the training data had a single truth value (about 50%), so under-sampling of the over-represented class was tested.
- The number of hidden units was significantly increased in order to capture more details from the data.
- The number of epochs was also increased.

Despite these steps, underfitting was not resolved. Further exploration of those strategies may be a good starting point for future work.