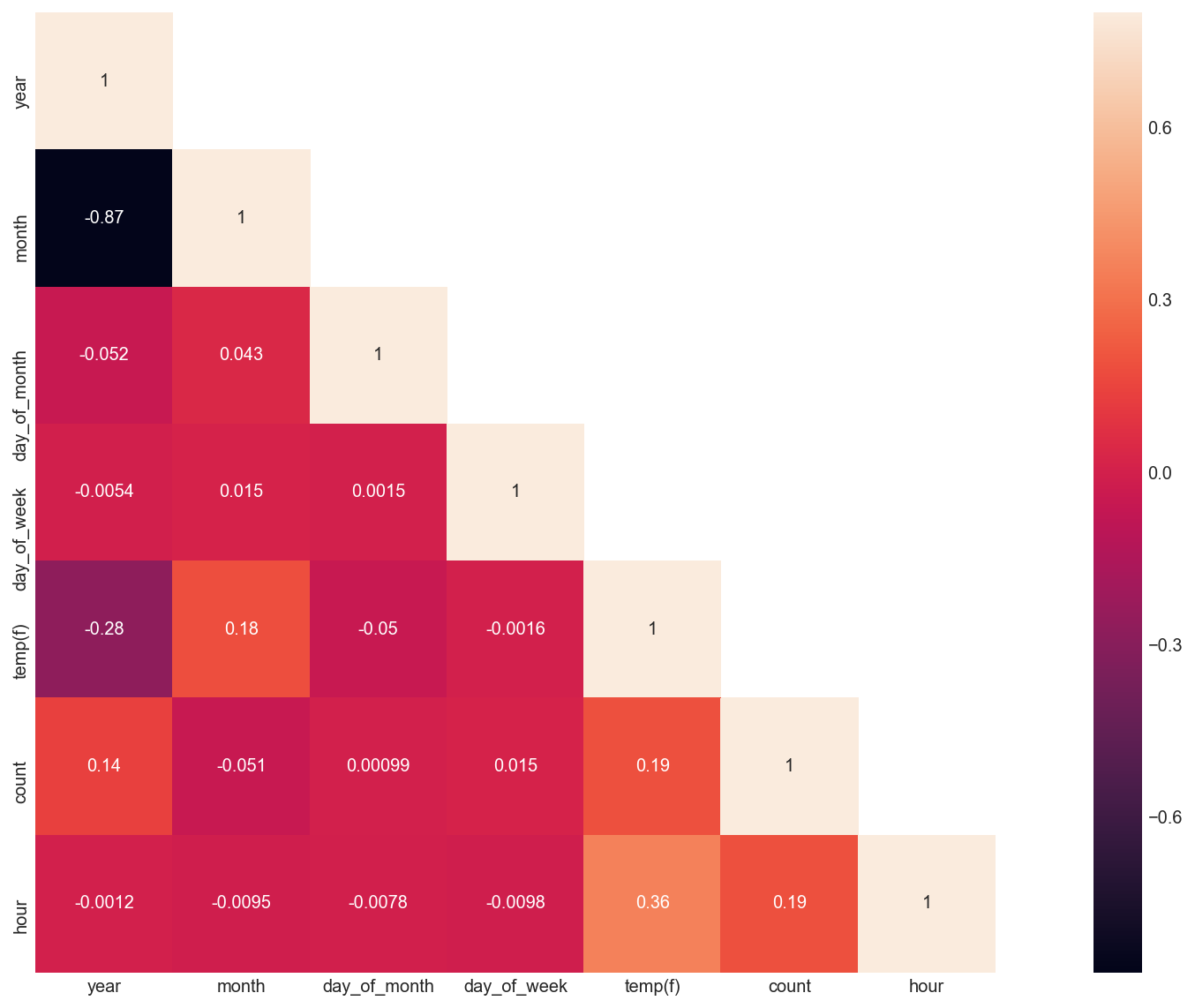
**Neural Networks applied to bike sharing usage forecasting**

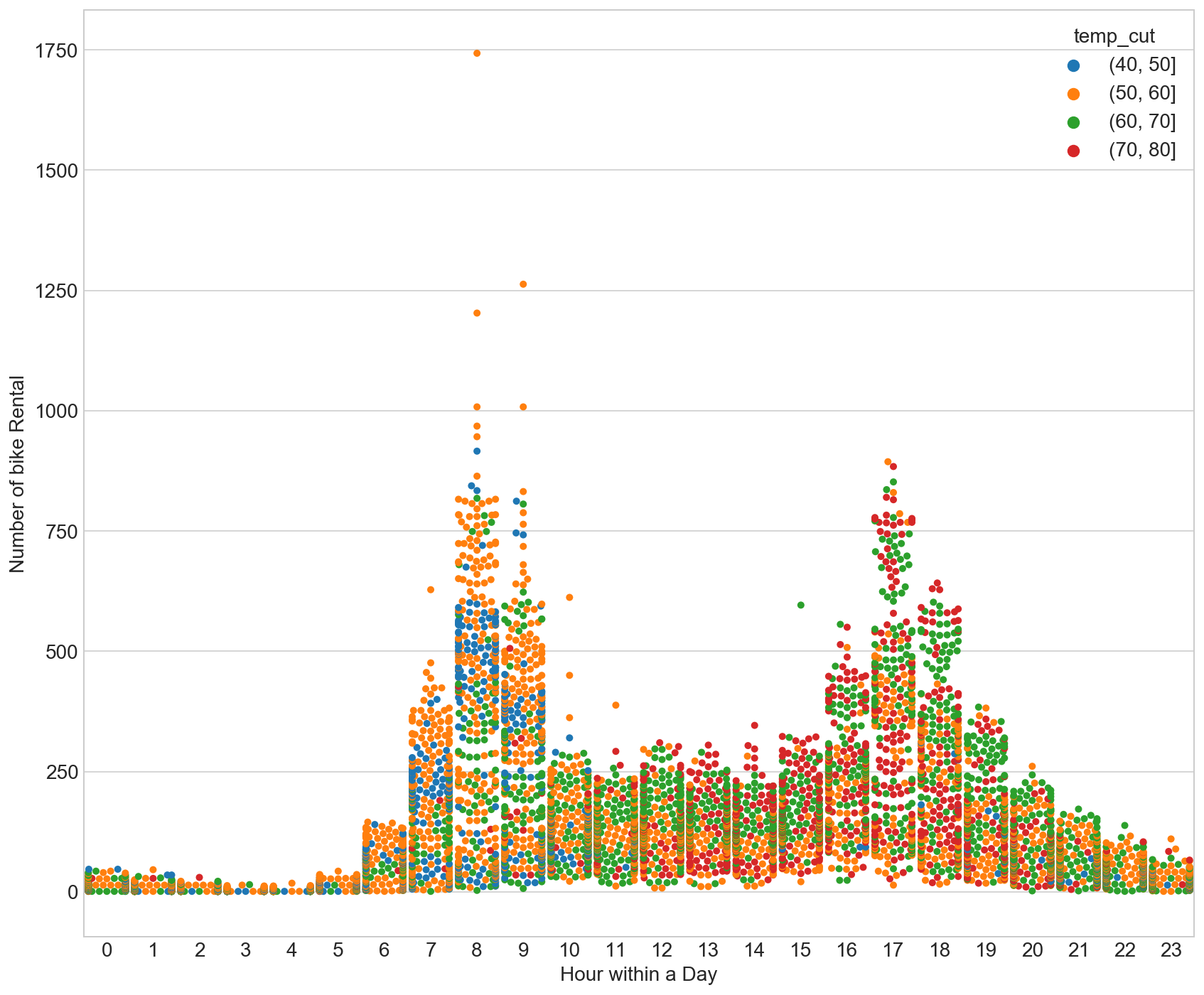
Used to predict bike usage in San Francisco on a per hour basis.

## Pre-analysis data visualization

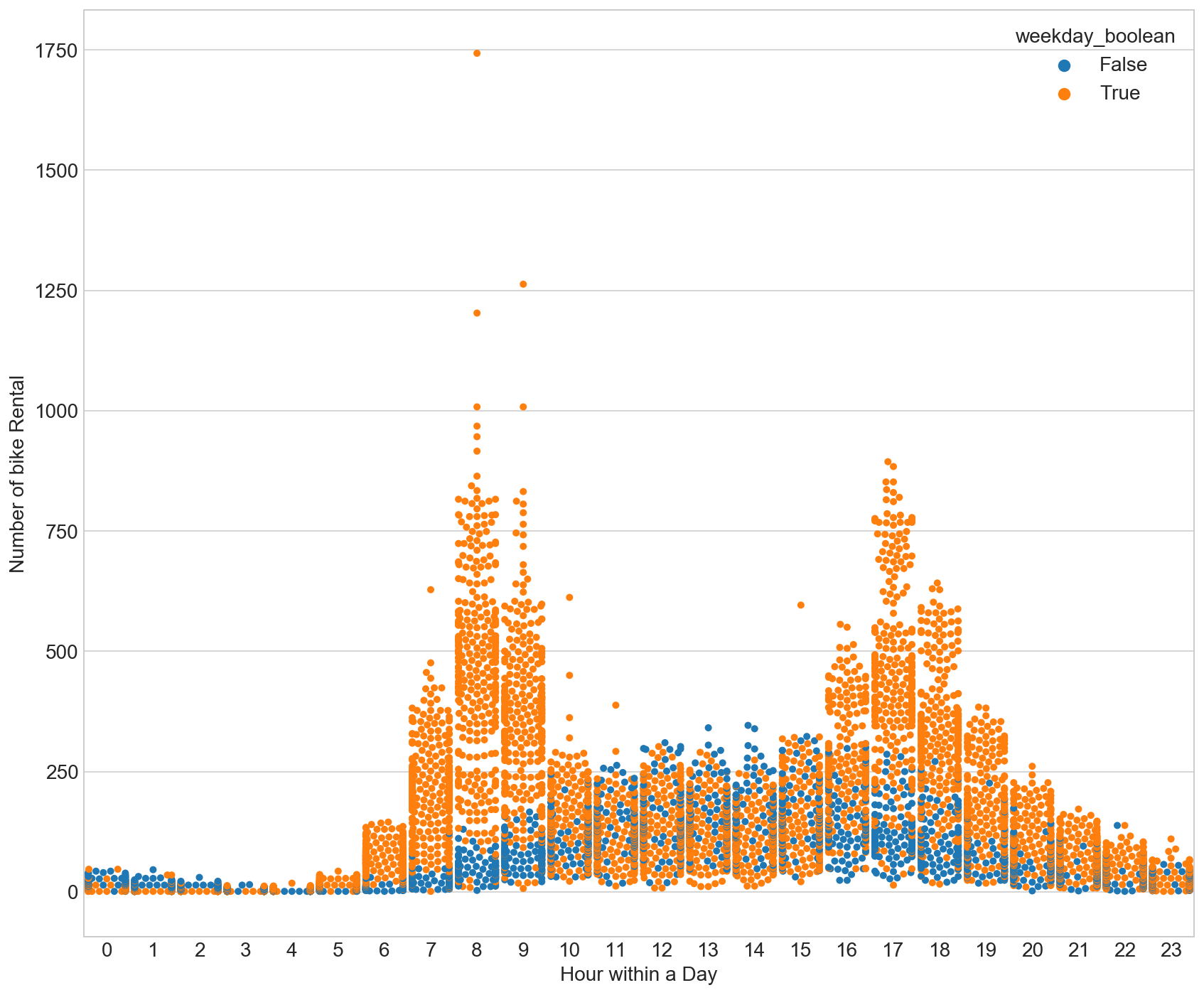
### Correlation analysis between all the variables:



### Hour vs bike usage with temp ranges



### Hour vs bike usage comparison between weekdays and weekend days



## Neural Network details:

Data was converted from a time series (each row represents a date with a periodicity of one hour) into a supervised learning problem, where each row (observation) included the values for the previous observation. After that, a neural network with a hidden LSTM (Long Short Term Memory) layer was used to build the model from the training data. LSTMs are a type of Recurrent Neural Network. This is a useful approach to time series forecasting with multiple variables (pressure, temperature, etc. in our case)**.**

## Notes:

* Data cleaning was vital for good results
* For instance, added average values for pressure and temperature for missing values and remove rows with duplicated indexes (important since these are timestamps)
* For simplicity, weather was categorized into rain/no rain
* An integer for the day of the week was added to the dataset

## Possible improvements:

* Further neural network tuning (nodes, layers, batches, etc.)
* Fill missing rows (timestamps missing)
* OneHot Encode for the day of the week, instead of using integers
* We are not considering seasonality, day (of the week), hour. This could be addressed with [Seasonal Adjustment](https://en.wikipedia.org/wiki/Seasonal_adjustment), or Deseasonalizing.

## **TEST RESULTS**

4 weeks of training data

1 week of test+validation data

4 Inputs (per hour):

* Weekday (integer)
* Pressure
* Rain (yes/no)
* Temperature

1 Output:

* Bike Usage Count (per hour)

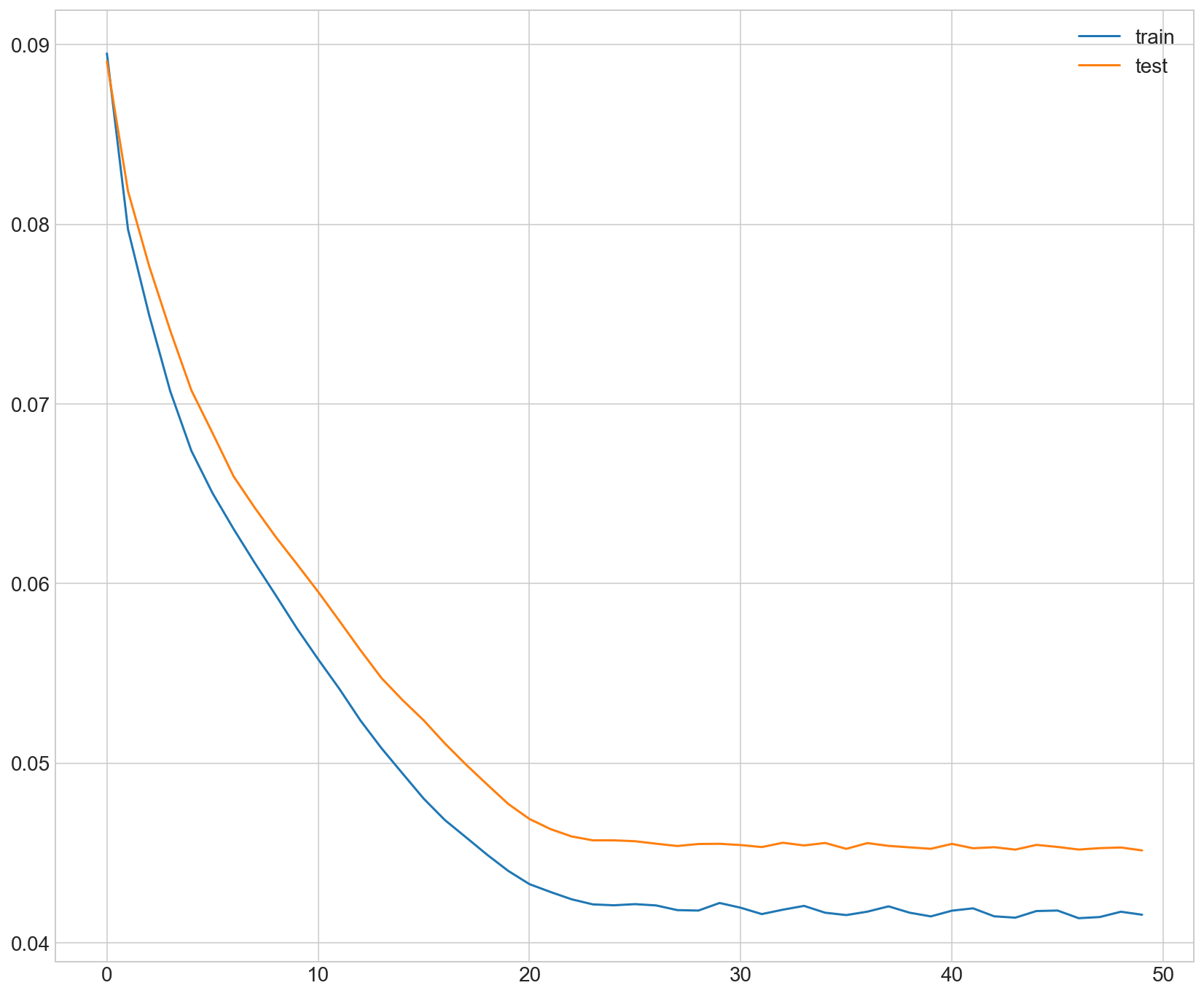
Test RMSE: 137.026

Root Mean Square Error: If \hat{y}_i is the predicted value of the i-th sample, and y_i is the corresponding true value, then the mean squared error (MSE) estimated over n_{\text{samples}} is defined as:



Normalized RMSE (RMSE/(y\_max-y\_min)): 0.160

Training and test loss:



Prediction vs actual bike usage (last week of June 2018)

