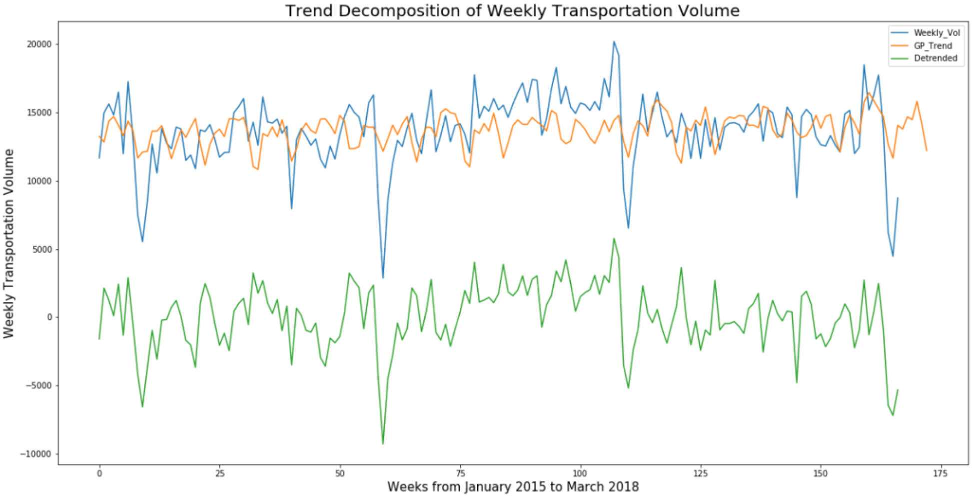
Demand Forecasting

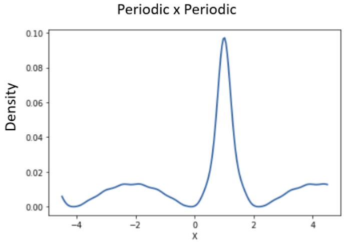
Sam Edds

Seaonality and Trend Detection: Results



How Gaussian Processes work

1. Compute all possible covariances (based on distance of points if stationary)
2. Covariance calculation: Take nearby points and treat as n instances of a normal variable
3. Reweights points based on the kernel chosen
4. The more narrow the kernel, the heavier nearby instances are weighted
5. Example: Squared Exponential: Weights decay exponentially in a symmetric fashion
6. Our final kernel choice below:



Seasonality and Trend Detection: Model Description

Final model parameters in red

1. **Metric for comparison**: Mean Absolute Deviation for 1-step and 10-step ahead prediction. Averaged across both predictions.
2. **Kernel weighting combinations tested** (Kernel 1 x Kernel 2, all included additive Linear and Bias kernels): Periodic Exponential x Periodic Exponential, Periodic Exponential x Matern52, RBF x Matern52
3. **Length scale bandwidth tested**: Kernel 1: 1/2/3, Kernel 2: 9/10/11, Linear: 1/ 2

Model Comparisons

Goal: Forecast n weeks ahead for detrended transportation volumes

1. **Metric for comparison**: Mean Absolute Deviation for 1-step and 10-step ahead prediction
2. **Linear Regression**
3. **Support Vector Regression**
   3.1 Kernels: Sigmoid, RBF
   3.2 Gamma values: .01 , .05, .07
   3.3 C loss penalty: 10, 100, 1000

Cross-Validation: Linear Regression has lower MAD and better fit

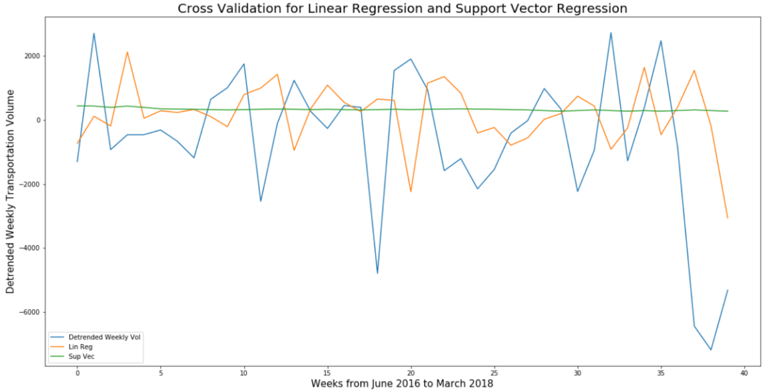


Table 1: Linear Regression

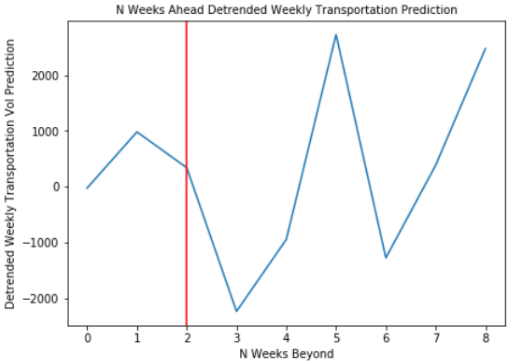
mean_abs_dev	Vars	n_steps_ahead	n_step_avg_abs_dev
1580.85	1 Week Shift and 4 Week Shift	0	1635.68
1690.50	1 Week Shift and 4 Week Shift	10	1635.68

Table 2: Support Vector Regression

mean_abs_dev	kernel_gamma_C	n_steps_ahead	n_step_avg_abs_dev
1666.35	['rbf', 0.07, 1000.0]	0	1756.45
1666.35	['rbf', 0.05, 1000.0]	0	1756.45
1846.55	['rbf', 0.01, 1000.0]	10	1756.45
1846.55	['rbf', 0.07, 1000.0]	10	1756.45
1846.55	['rbf', 0.05, 1000.0]	10	1756.45

Model Chosen: Linear Regression

Predicting 6 steps ahead from the red line



Augmentations

With additional time I would do the following:

1. While data are detrended, there is still high correlation 10+ weeks back. I would explore different lags.
2. Explore more kernel options to improve detrending
3. Explore additional methods to predict n weeks ahead data (e.g. Random Forests) and different lags ahead