Predicting Sludge Volume Index at a Wastewater Treatment Plant

Problem Identification

Background

Efficient removal of sludge from wastewater to enable the safe discharge of clean water into nearby water bodies is a fundamental goal at wastewater treatment plants (WTPs). This process of separation is quantified using the Sludge Volume Index (SVI). SVI plays a crucial role in assisting WTP operators in assessing the efficacy of the activated sludge process. Elevated SVI values may signal underlying issues such as the overgrowth of filamentous bacteria, which can severely impair the separation of solids and liquids, consequently limiting the maximum flow capacity that the facility can effectively treat. Conversely, a low SVI indicates optimal settling conditions. Effective management of SVI not only ensures the smooth operation of the treatment process but also conserves biomass and results in the production of clear effluent, thus enhancing overall treatment efficiency and regulatory compliance. This metric serves as a valuable tool for troubleshooting operational issues and guiding adjustments to process parameters, ultimately optimizing wastewater treatment performance.

Objective

SVI can change slowly over time in response to changes in waste water and plant operations. In this project, I developed machine learning models to predict SVI using a dataset comprising daily wastewater measurements spanning five years at a fictitious WTP. The primary objective was to establish models capable of forecasting SVI using explanatory variables that can be modified by a WTP operator. By identifying and analyzing these controllable factors, the model can provide valuable insights to optimize treatment processes and achieve more efficient sludge settling, ultimately enhancing the overall performance of the wastewater treatment facility.

Data Wrangling

Data Acquisition

Data was provided by the WTP manager in an excel file and consisted of 2,282 rows and 76 columns. The data contained two rows of column headers with one being the WTP metric name and the other indicating whether the metric is controllable by the operator.

Data Cleaning

Non-controllable metrics can have an impact on SVI, but the operator cannot control them and their inclusion in a model is of less value. Therefore, I removed these metrics from the dataset.

The remaining data came formatted in ‘object’ data type which could not be used in calculations. The index contained the sample dates which I changed to a ‘datetime64’ format while I changed the metric data to ‘float64’ format.

I investigated data frequency by subtracting one day from the next. This revealed that data had been collected daily and no days were missing. I also checked the data for duplicate rows and none existed.

The dataset does not include units unfortunately. Confirming consistency of units for a metric is important for data integrity. If found, inconsistent units and associated values can be converted to the appropriate unit. In addition, considering units and values in combination with knowledge of the metric is also helpful when determining the presence of outliers.

I plotted the data in line graphs to investigate data gaps and calculated the percentage of NaNs to determine the usability of metrics. I then removed the metrics with a very high percentage of NaNs and/or with very large temporal data gaps from the dataset (Table 1, Figures 1-3). Typical methods for imputting these large percentages or filling large gaps would not be reliable and may confound a future model.

In order to group the metric data and look for patterns in the remaining NaNs, I added columns for days of the week and months of the year. Finding patterns could be helpful in deciding how to input the missing data. I noted that ‘SVI’, ‘MLSS’, and ‘RAS TSS’ were collected on weekends far less frequently than weekdays. In addition, ‘SE NH3’ was collected on only Sunday through Thursday. If this were a real dataset, I would contact the WTP manager to ask if the ‘SE NH3’ data had possibly been shifted up one

row in the dataset and should be corrected. Otherwise, I did not find any other patterns in the NaNs.

Table 1. Metrics removed from dataset.

Metric	Data gaps	NaN (%)
Aeration Basins in Service	No data recorded	100
MLSS (Hach)	Data record stops half way through the data period	67
MLSS Avg. (Hach)	Data record stops half way through the data period	67
MLVSS	Data is too sparse	87
RAS VSS	Data is too sparse	87
lbs_P_Removed	Data is too sparse	87
lbs_P_per_gal_Alum	Data is too sparse	89
ND AER Avg MLR Flow	Data record stops half way through the data period	52
ND AER Avg Jet Mix MLR Flow	Data record stops half way through the data period	52
ND AER Total Avg MLR Flow Per Basin	Data record stops half way through the data period	52
ND AER Total MLR Flow	Data record stops half way through the data period	52
Avg ND Aeration DO Cell 4 AVG	Data collection only recently began	77
Avg ND AER Avg Airflow Cells 1_6	Data collection started half way through the data period	48
Avg ND AER Avg Airflow Cells 7_10	Data collection started half way through the data period	48
Avg ND RAS Pump 1 Flow	Data record stops half way through the data period	76
Avg ND RAS Pump 2 Flow	Data record stops half way through the data period	67
Avg ND RAS Pump 3 Flow	Data record stops half way through the data period	66
Avg ND Total RAS_RAS PUMPS 1_3	Data record stops half way through the data period	52

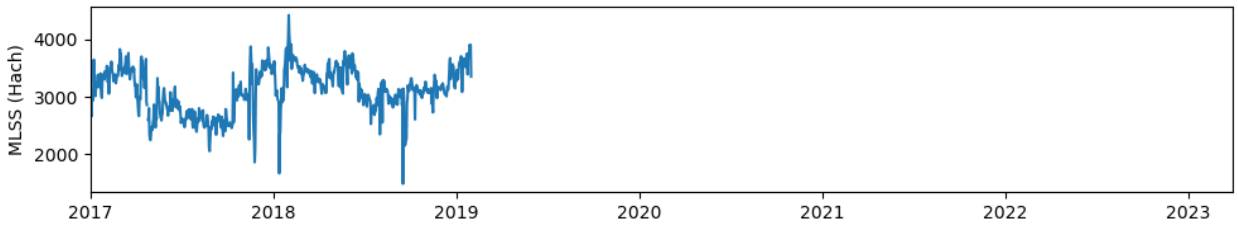


Figure 1. Example of metric data that stops half way through the data period.

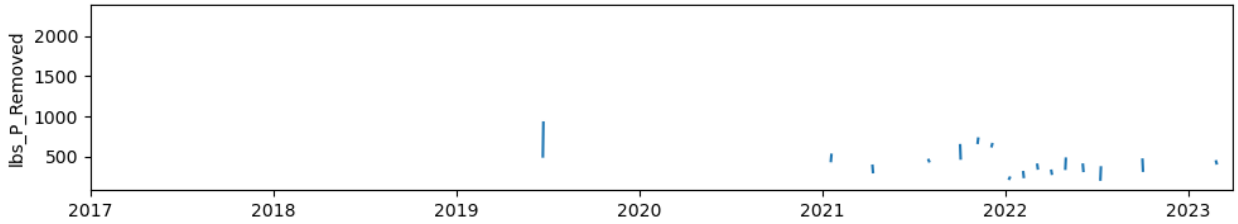


Figure 2. Example of metric data that is very sparse through the data period.

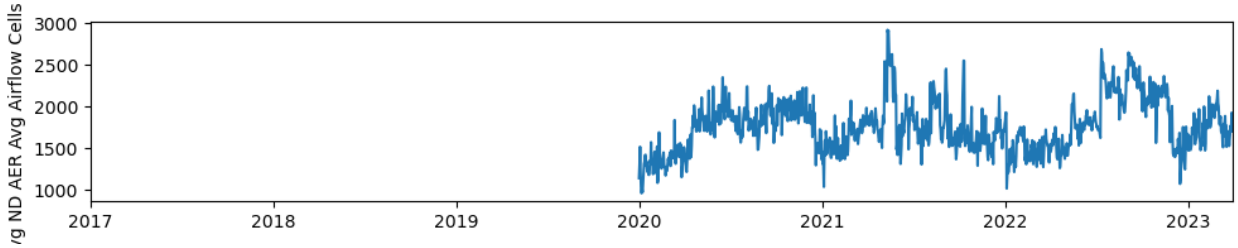


Figure 3. Example of metric data that starts half way through the data period.

As a last step of data organization, I imputed the missing values. For this step, I performed linear interpolation which calculates intermediate values between two known data points on a straight line. This method was more appropriate than others, such as imputing mean or median, as this is time series data and values are related to each other. Imputing was performed in both forward and backward directions from each NaN value.

Exploratory Data Analysis

Data Distribution

I began EDA by examining the distribution of the data. I described the metrics for min, max, and mean; plotted histograms (Figure 4) of the metrics, and ran a Shapiro-Wilk test for each metric. These assessments indicated none of the metrics were normally distributed. Therefore, the median values were used to explore the data further.

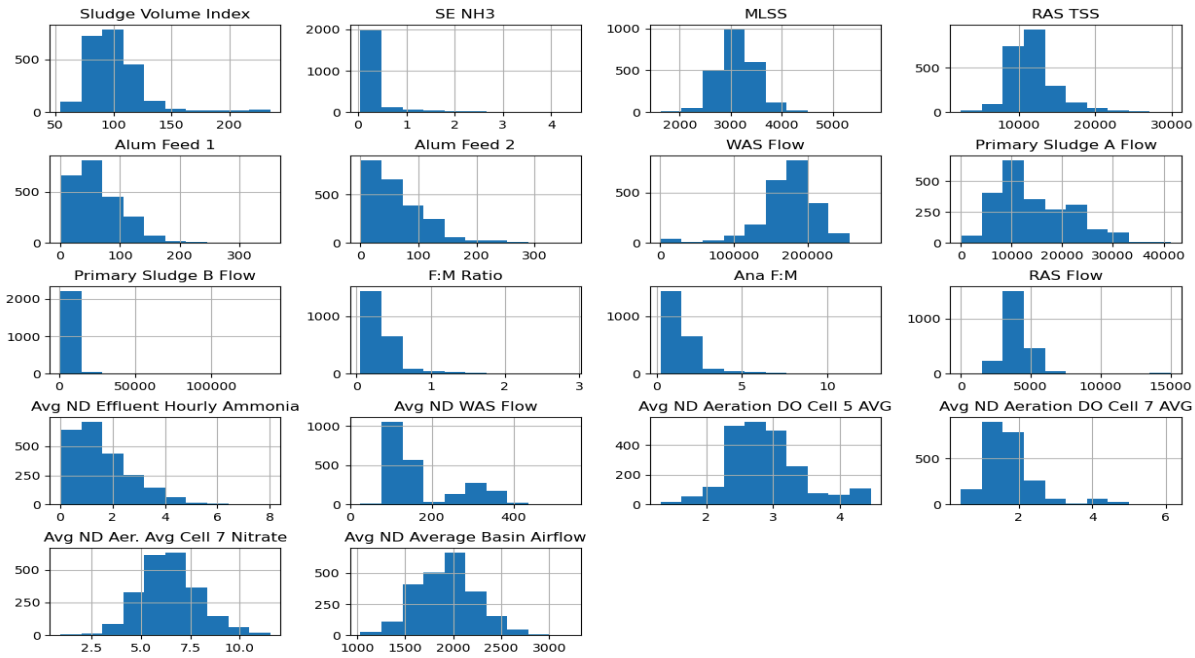
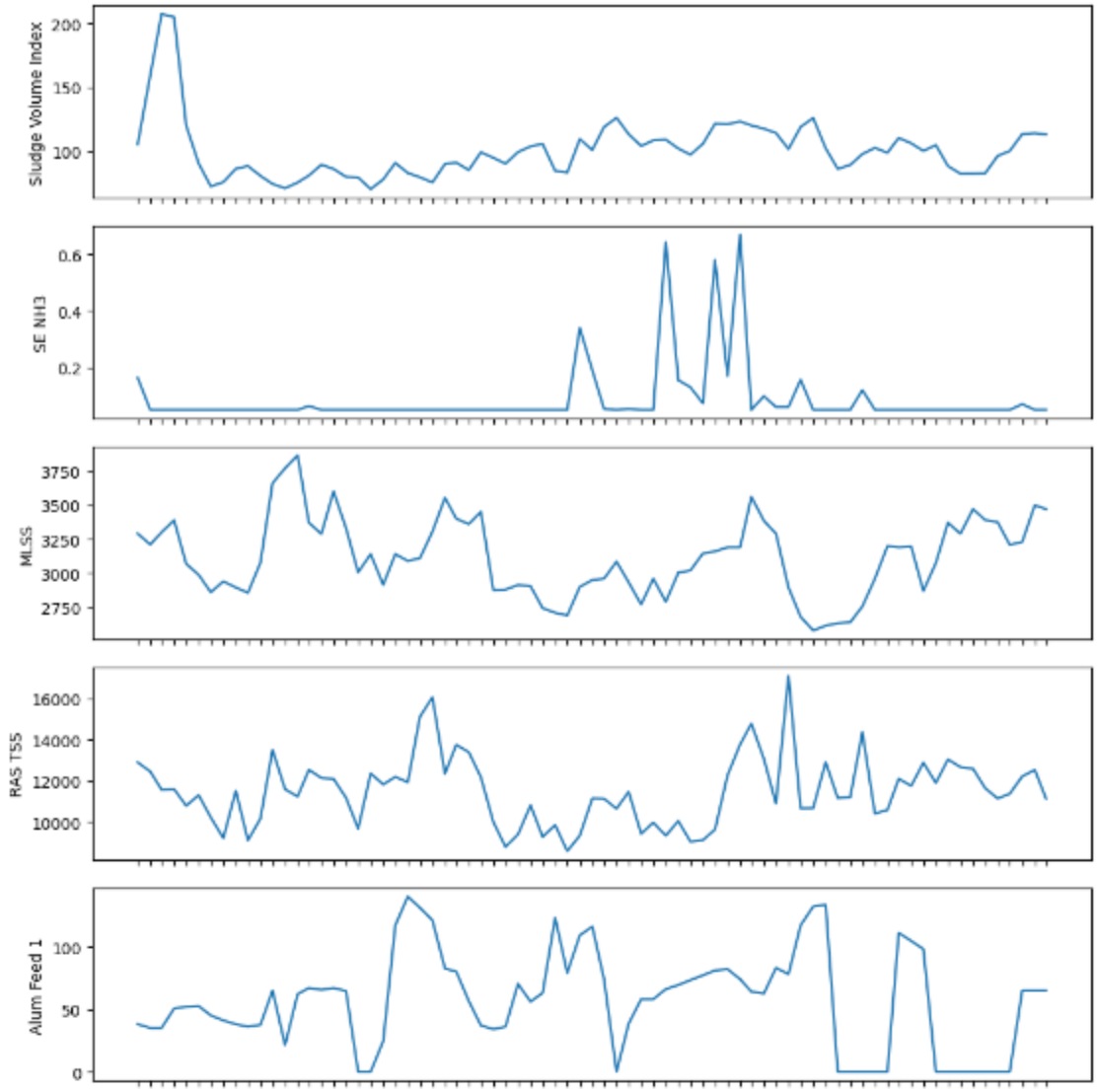
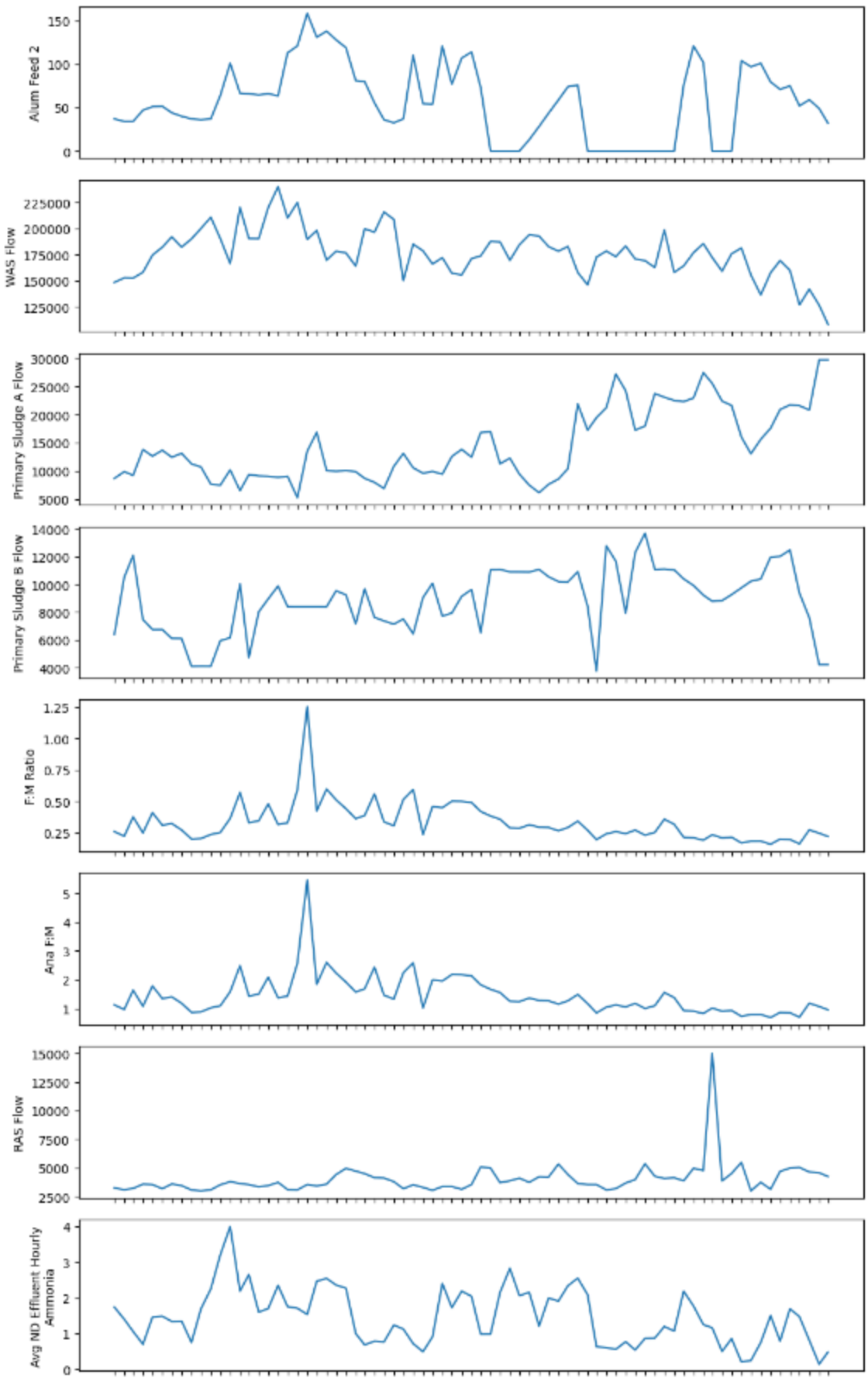


Figure 4. Histograms of metric data.

Visualization

I again plotted line graphs of the metrics but this time in a “smoother” format (Figure 5). First, I added a year column to the dataset, grouped data by year and month, and calculated the median. These “smoothed” figures with no data gaps were easily visually assessed for trends (directional, seasonal) and stationarity. Most obviously, I discovered that the ‘ANA F:M’ and ‘F:M Ratio’ graphs appeared exactly the same. A decreasing trend in ‘WAS flow’ and increase in ‘Primary Sludge A Flow’ from mid 2018 to 2023 was identified. In addition, many metrics had relatively large values in 2017 and 2018. A few metrics display seasonality. In addition, some metrics displayed minimal time dependent variance but, overall, metrics appeared to be stationary.




Alum Feed 2
150
100
50
0
WAS Flow
225000
200000
175000
150000
125000
Primary Sludge A Flow
30000
25000
20000
15000
10000
5000
Primary Sludge B Flow
14000
12000
10000
8000
6000
4000
F:M Ratio
1.25
1.00
0.75
0.50
0.25
Ana F:M
5
4
3
2
1
RAS Flow
15000
12500
10000
7500
5000
2500
Avg NO Effluent Hourly Ammonia
4
3
2
1
0

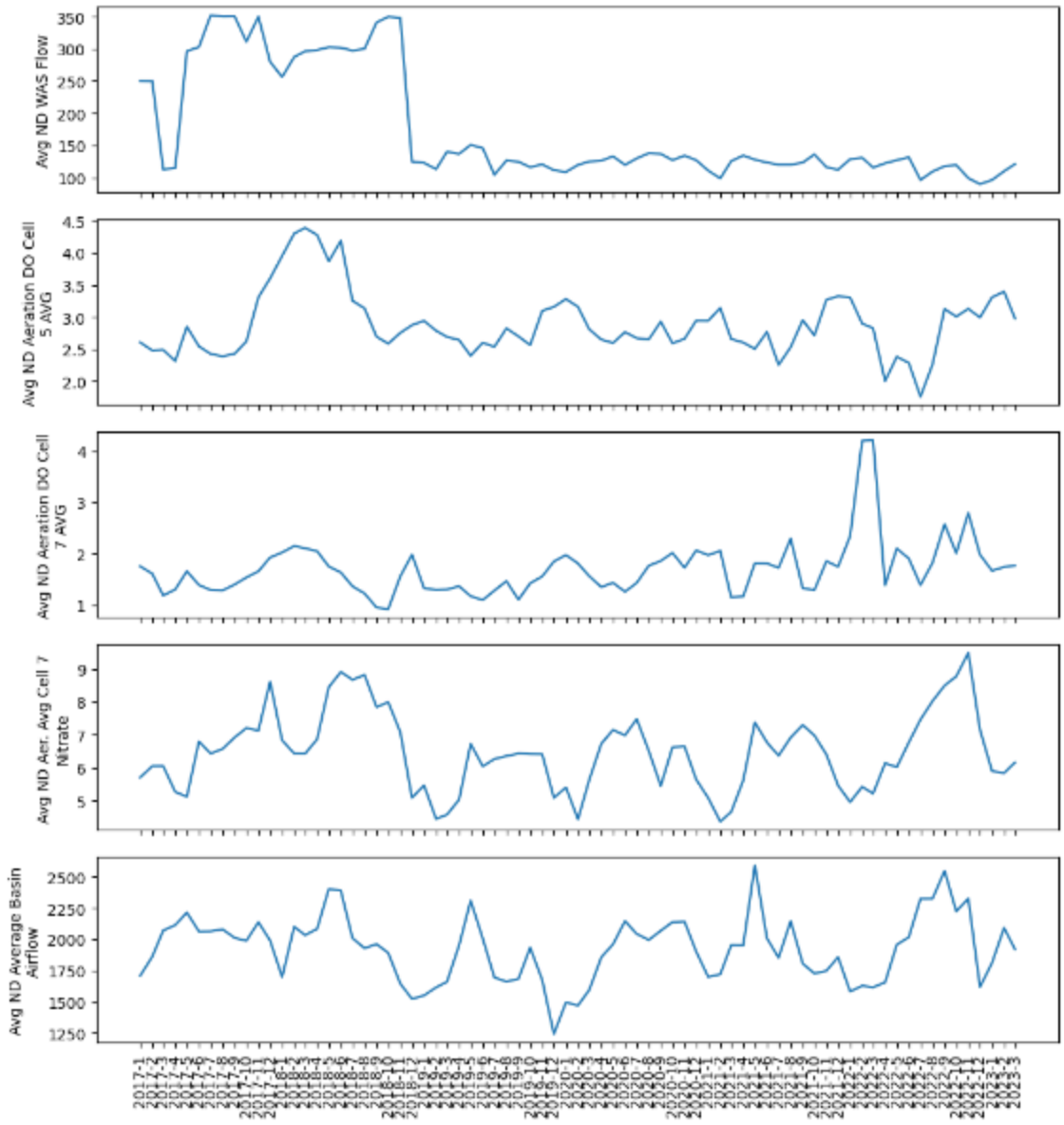


Figure 5. Plots of monthly median metrics.

I also created boxplots for each metric to investigate data distribution. Many metrics had extreme values, however, the majority were not greater than three standard deviations from the median which is a common threshold for removing outliers. In addition, I have no reason to believe that any of these values were erroneous.

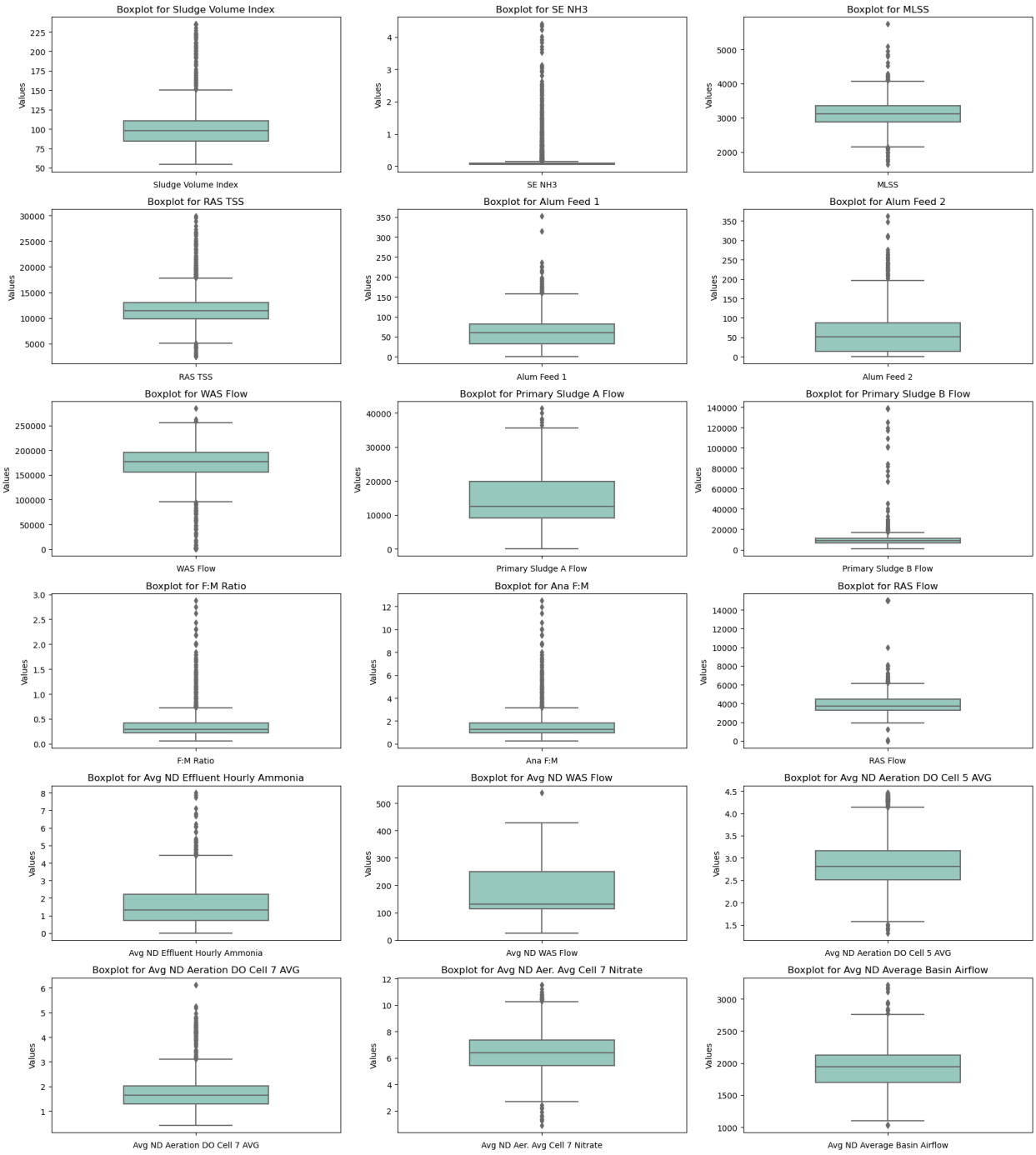


Figure 6. Box plots of metric data.

Correlations

I created a correlation heatmap between metrics to investigate linear relations between metrics, specifically SVI (Figure 7). Heatmaps are an effective way to visualize many correlations at once, making it easy to identify correlations and patterns in correlations.

The heatmap showed weak correlation between SVI and other metrics. Between all other metrics, ‘ANA F:M’ and ‘F:M Ratio’ are perfectly correlated as expected. As ‘F:M Ratio’ measures both aerobic and anaerobic treatment, this perfect correlation would indicate that no aerobic treatment is occuring. I, therefore, removed ‘F:M Ratio’ from the dataset as it adds no more information to the model. In addition, ‘Primary sludge flow A’ is positively correlated with year and ‘Avg ND WAS Flow’ is negatively correlated with ‘year’ but only moderately.

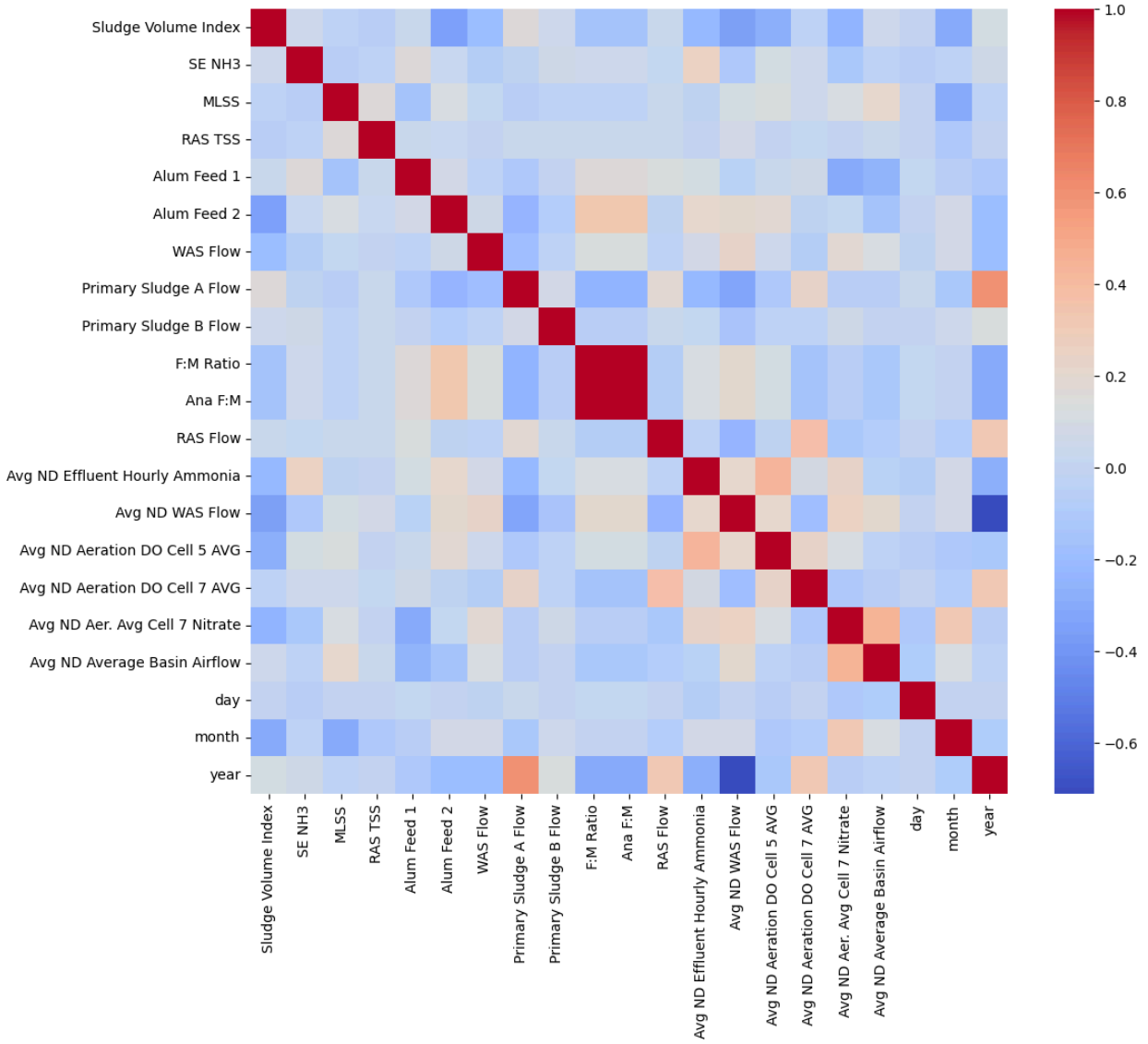


Figure 7. Correlation heat map of metric data.

Lastly, I created a pairplot to further investigate linear relationships and also other relationships between SVI and other metrics. Some of the data for ‘Alum feed 1’ and ‘Alum feed 2’ appear to have a strong linear relationship but much of the data did not. Overall, these scatter plots uncover few additional relationships in the data.

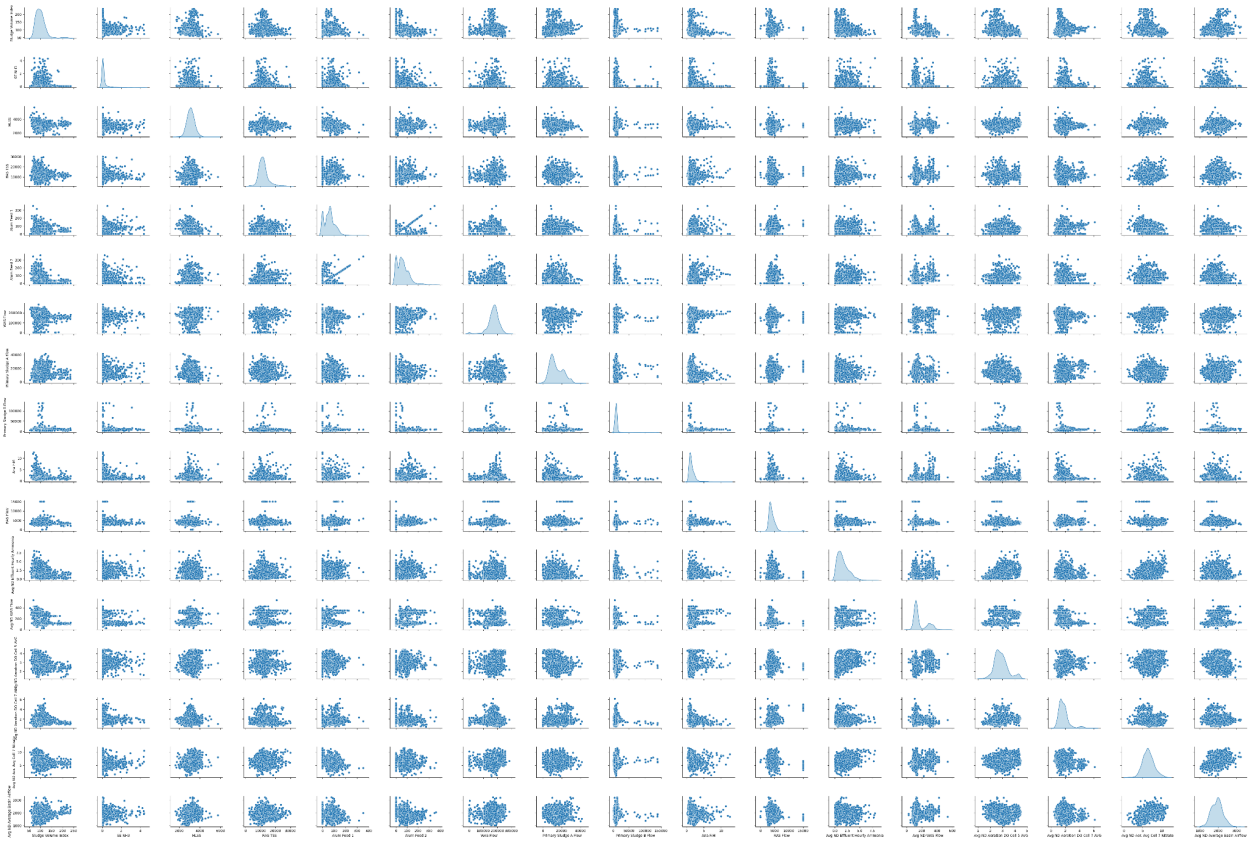


Figure 8. Pairplot between SVI and other metrics.

Pre-processing and Training Data Development

Feature Engineering

As part of feature engineering, I used existing data to calculate additional WTP metrics based on domain knowledge (Table 2). These metrics are commonly recorded by WTPs.

I then performed differencing on the original and these new metrics to compute the differences between consecutive data points. This technique is often performed for time series models and can transform a non-stationary time series into a stationary series and remove trends and seasonality. Overall, differencing can be beneficial because it stabilizes the mean and variance of the data, making it easier to identify underlying patterns and relationships.

Creating lagged versions of data is also a common technique performed for time series models and incorporates temporal relationships involving past values into the model. SVI tends to change over the course of weeks so I created one through seven day lags

which was done simply by copying and shifting the columns in the dataframe the appropriate number of days.

Table 2. Metrics calculated for dataset.

New Metric	**Calculation**
Total Alum Feed	Alum Feed 1 + Alum Feed 2
Total Primary Sludge Flow	Primary Sludge A Flow + Primary Sludge B Flow
Ave ND Aeration DO in Cells	(Avg ND Aeration DO Cell 5 AVG + Avg ND Aeration DO Cell 7 AVG) / 2
MVLSS	MLSS x 0.75
MCRT	MVLSS / (Activated Sludge Flow + RAS Flow)
Sludge Age	MCRT / (1 + F:M Ratio)

Dummy features were created in past steps when I added day, month, and year data to the dataset. Otherwise, the data was not categorical and dummy variables were not necessary.

Pre-processing

I defined multiple versions of X and y for use in my models. I did this for the X data to try different combinations of independent variables in the models. This also reduced the size of the dataset which prevented the curse of dimensionality and overfitting. These data consisted of original data (including the additional WTP metrics I calculated) and specific orders of differenced and lagged data containing and not containing SVI. I didn't create every possible combination of these data groups, but rather a range so that I could hone in on specific combinations when running the models. For instance, I made a data set of data with a one day lag, a set with four day lag, and with seven day lag. That way, a model performing, for instance, well with one and four day lag data but poorly with seven day lag data, would direct me to try the model with two or three day lag data. As for y data, I defined a dataset of original SVI and SVI shifted forward one day. The latter was created to avoid any potential data leakage.

Modeling

Models capable of making predictions from time series data were essential for this project. I chose to explore linear regression, random forest, and ARIMAX models.

For each model, rather than splitting, training, and testing once, cross-validation was performed to divide the data into 5 subsets and training and testing were performed five

times using a different subset for testing each time. I specifically used TimeSeriesSplit() for my cross-validation because this function is specifically designed for time series data. Each fold is a superset of the previous one, ensuring that the model is trained on data that retains the temporal nature of time series data. Predictions were then made and each fold was scored and averaged for a final score. This process avoids overfitting the model to the data and thus makes it more generalized for future data.

The scores I used were root mean squared error (RMSE) and adjusted R squared (R2adj) were calculated. RMSE was calculated to measure the error between the model's predictions and the actual values. R2adj was calculated to measure the proportion of the variance in SVI that is explained by the features and adjusted to account for the large number of predictors in the model. I chose to adjust the R2 because the R squared value of a model can increase as predictors are added to a model even if they are not contributing to the explanatory power of the model. Adjusting the R-squared corrects this.

Other than being performed with cross-validation, the three models were mostly out of the box. The exception being that the ARAMAX model included exogenous features, in contrast to the simpler ARIMA model which only uses the target variable. Before fitting this model, I used auto ARIMA to choose the most appropriate p, d, and q hyperparameters for the model.

Each of the models were run with combinations of my different X and y versions. The linear regression and random forest models used the Xs that included differenced and lagged SVI data while the ARIMAX model used the Xs that did not include differenced and lagged SVI. The ARIMAX model does not need to be fed differenced and lagged versions of the target variable because it creates its own. All models were run with original SVI and SVI shifted as the y.

Linear Regression

The RMSE and R2adj scores (Table 3) from running linear regression models with the various data indicated that the models:

1. Perform poorly with the original data and a lag of 4 and 7 days,
2. Are not improve by differencing the data,
3. Perform moderately well with shifted y and a lag of 1 day data (considering a mean SVI of 101 and max of 235),
4. Appear to suffer from data leakage when y is shifted and lag of 1 day is used,
5. Perform better when y is shifted than when it is not in combination with lag data.

This information indicated that I should run models with a combination of shifted y data and lag of 2 days and lag of 3 days. The models perform:

1. Moderately well with lag of 2 days,
2. Poorly with a lag of 3 days.

Overall, the highest performing linear regression model used shifted y data and a lag of 1 day. The RMSE indicated that the error between the model's predictions and the actual values was small. Adjusted R squared, on the other hand, indicated that the model explains only 16% of the variance in the target variable.

Table 3. Scores for the three models run to predict Sludge Volume Index (SVI). The lagged and differenced data used for the ARIMAX model did not include SVI. RMSE = root mean square error, R2adj = adjusted r squared, y = target data, sludge volume index, shifted y = y data shifted one day forward, original x = unaltered features including those calculated, 1st diff X = first-order differenced X, *t* day lag X = X data lagged by *t* days.

	Linear Regression		Random Forest		ARIMAX	
Data	RMSE	R2adj	RMSE	R2adj	RMSE	R2adj
y, original X	31.13	-10.05	24.91	-8.83	17.77	-1.19
Shifted y, original X	31.55	-10.26	26.11	-10.35	17.10	-1.04
y, 1st diff X	22.94	-19.88	18.55	-5.96	17.77	-2.98
Shifted y, 1st diff X	22.94	-19.90	18.59	-6.21	17.10	-2.70
y, 1 day lag X	8.76	0.16	8.34	0.44	17.77	-1.50
Shifted y, 1 day lag X	2.00	1.0	0.54	0.97	17.10	-1.32
y, 4 day lag X	11.95	-0.64	11.51	-0.05	17.77	-1.50
Shifted y, 4 day lag X	11.47	-0.54	10.94	0.06	17.10	-1.32
y, 7 day lag X	13.64	-1.48	12.52	-0.23	17.77	-1.50
Shifted y, 7 day lag X	12.87	-1.00	11.86	-0.13	17.10	-1.32
Shifted y, 2 day lag X	8.79	0.14	–	–	–	–
Shifted y, 3 day lag X	12.87	-1.00	–	–	–	–

Random Forest

The RMSE and R2adj scores (Table 3) from running random forest models with the various data indicated that the models:

1. Perform poorly with the original data and a lag of 4 and 7 days,

2. Are only slightly improve by differencing the data, although they still perform poorly,
3. Perform better with shifted y data than with non shifted y data in combination with lagged data,
4. With a lag of one day, perform moderately well with not shifted y and very well with shifted y.

This information indicated that I did not need to run the model again with different features because I already found a very good model.

Overall, the random forest model using non shifted y data and a lag of 1 day performed best. The RMSE indicated that the error between the model's predictions and the actual values was very small. In addition, the R2adj indicated that the model explains almost 97% of the variance in the target variable.

ARIMAX

Lastly, the RMSE and R2adj scores (Table 3) from running ARIMAX models with the various data indicated that the models:

1. Perform poorly with all data,
2. Are not improve by differencing the data,
3. Perform slightly better with shifted y values than not shifted.

This information indicated that I did not need to run the model again with different features because no additional data is expected to produce a better ARIMAX model. Overall, these scores indicate that ARIMAX models do not provide a good fit to the data and perform worse than a simple mean prediction model as indicated by the negative R2adj scores.

Conclusion

The random forest model using shifted SVI data and features lagged by one day produced the highest performing model to predict SVI at the WTP.