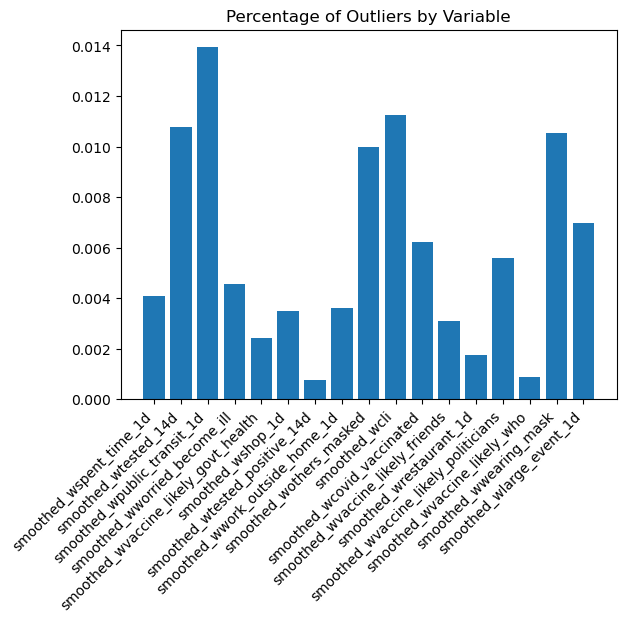
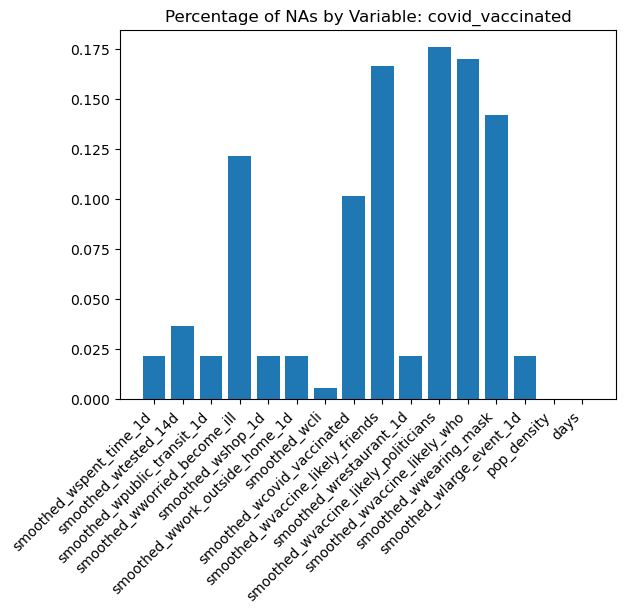
In this iteration of our exploratory data analysis, we began by exploring the number of outliers in each feature. We found the number of outliers for each feature to be less than 2% of the total, so we removed all outliers from the data set:



Next, we converted categorical variables into variables we could use in our analysis. First, using publicly available data from the US Census Bureau, we encoded the ‘geo\_value’ feature into the county and state, then used that information to create a new variable representing the population density of the state in which the given county resides. We also converted the ‘time\_value’ feature into an integer feature representing the number of days since 1 January 2021.

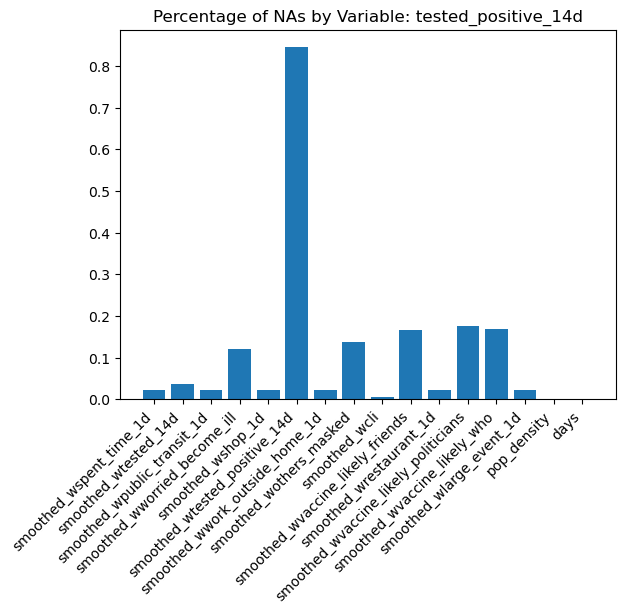
After this, we split the dataset into 2 datasets, one for each of the two target variables. Additionally, we conducted correlation analysis, finding that the features ‘smoothed\_wwearing\_mask’ and ‘smoothed\_wothers\_masked’ were 0.84 correlated and the features ‘smoothed\_wvaccine\_likely\_who’ and ‘smoothed\_wvaccine\_likely\_govt\_health’ were 0.91 correlated. To avoid multi-collinearity, when we split the datasets, we only kept the features in each of these pairs which was more correlated with the target variable of that dataset.

Following this, we analyzed the number of missing variables in each dataset. In the dataset associated with the target variable ‘smoothed\_w\_covid\_vaccinated’ (referred to as CV from here on), we found that the target variable was missing close to 20% of its values.

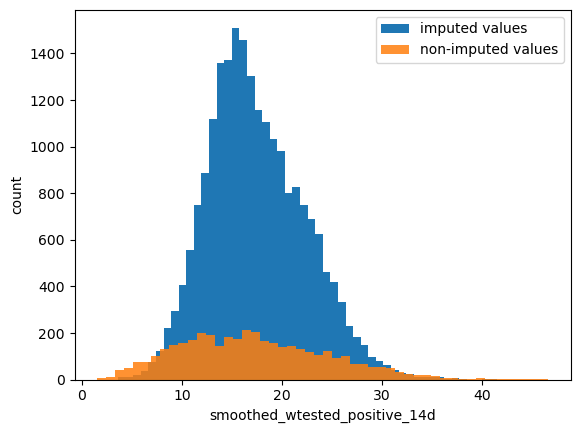


We resolved this by dropping all observations missing this target variable, then imputing the feature mean to all observations still missing values after doing this.

In the dataset associated with the target variable ‘smoothed\_w\_tested\_positive\_14d’ (referred to as TP from here on), we found that over 80% of the target values were missing.



Rather than drop these missing values, we decided to attempt to impute values based on our available data. First, we imputed the mean value to all non-target variables. Then, we fit a zero mean Gaussian Process with Matern ½ kernel to the data which was not missing the target values. We then imputed the mean value from this fitted Gaussian Process to the missing values. A histogram of the resulting values is shown below:



Of note, the imputed values are centered at the mean of the non-imputed values, and exhibit a similar (though not identical) variance to the non-imputed values.

In order to account for the ways in which this imputation method may be flawed, we only used these imputed values for training, not for testing. For the TP dataset, our testing dataset consisted of 20% of the non-imputed data, while our training dataset consisted of the remaining 80% of the non-imputed data along with the imputed data. For the CV dataset, we split the data 80/20 between training and testing.

**Neural Network Analysis**

We attempted to predict both target variables using a neural network. We used a 3 layer perceptron with 128 nodes per layer. We used RMSE as our loss function to ensure comparability with our other methods and a learning rate of 0.0001, training for 50 epochs.

For the TP dataset, our initial attempt using the imputed data produced the following training and testing losses: With a final testing loss of 5.7503.

Without the imputed data, however, the neural network performed even worse, creating the final training and testing losses:



with a final testing loss of 6.4398.

We finally attempted an approach similar to pre-training, using the first 25 epochs to train using the mixture of imputed data, and then using the next 25 epochs to train on only the non-imputed training data. This performed only marginally better, achieving a final testing loss of 5.6028.

The neural network performed slightly better on the CV dataset, creating the following training and testing loss:

With the final test loss of 5.1819.

Overall, it seems that a neural network is poorly suited to this problem due to the lack of data available.