West Nile Virus Prediction

What can the city of Chicago do to properly allocate mosquito spray (when and where) throughout the city to minimize the spread of the West Nile Virus?

Introduction:

The West Nile Virus is very commonly spread to humans through bites of infected mosquitos. Around 20% of infected people develop severe symptoms that can lead to death. The first human cases of the West Nile Virus were reported in Chicago in 2002 and by 2004, The Chicago Department of Public health had a control program in place to help control the outbreak of the virus. Analyzing the given trap, weather and spray data, my goal is to create a model to help determine when and where in the city to spray to help mitigate the spread of the virus.

Data Wrangling:

The data provided by Kaggle contained various files including train.csv, test.csv, spray.csv, weather.csv as well as an .rds and a .txt file to use for generating graphs including the map of the city of Chicago. For this project, I did not use the given test.csv file as I will instead be doing my own train-test split from train.csv. The spray data will not be useful since when and where to spray is what I am trying to figure out.

The train dataset was organized in a way that if the number of mosquitos exceeded 50, it would be split into another row. One of the main tasks was to condense all of these rows into a single data entry. There were also a few discrepancies between the Address, Trap, and Latitude and Longitude columns. There were two separate traps (T009 and T035) that had multiple latitude and longitude values so I had to reference the latitude and longitude coordinates of the addresses in question on google maps to properly match each address with its accurate coordinates. I also dropped a few columns that were redundant, such as Address and Address_street_name.

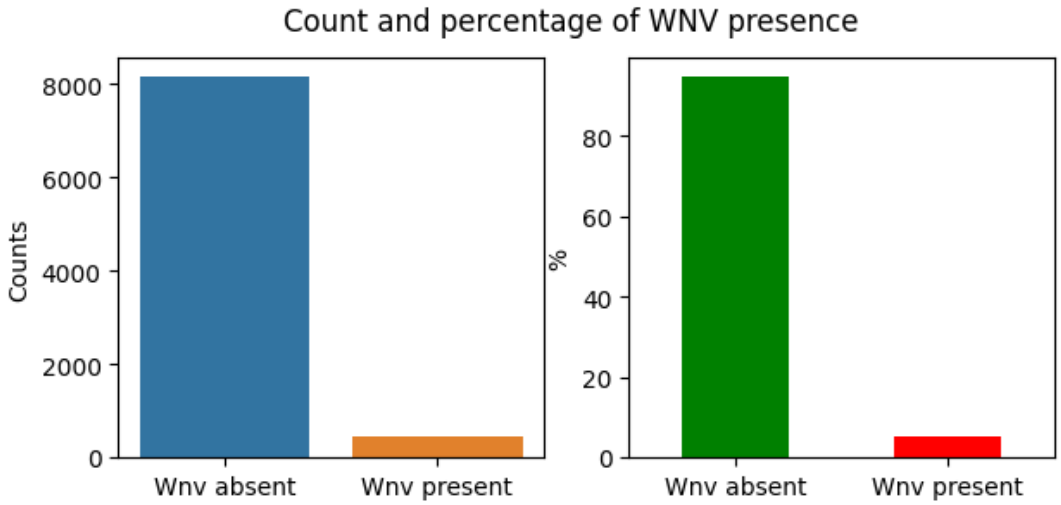
All of the weather data was collected at two stations. Since each station recorded data on the same days, I took the average reading from both stations and dropped the Station column. The majority of the cleaning for this data frame involved cleaning missing values and converting each feature into an appropriate data type.

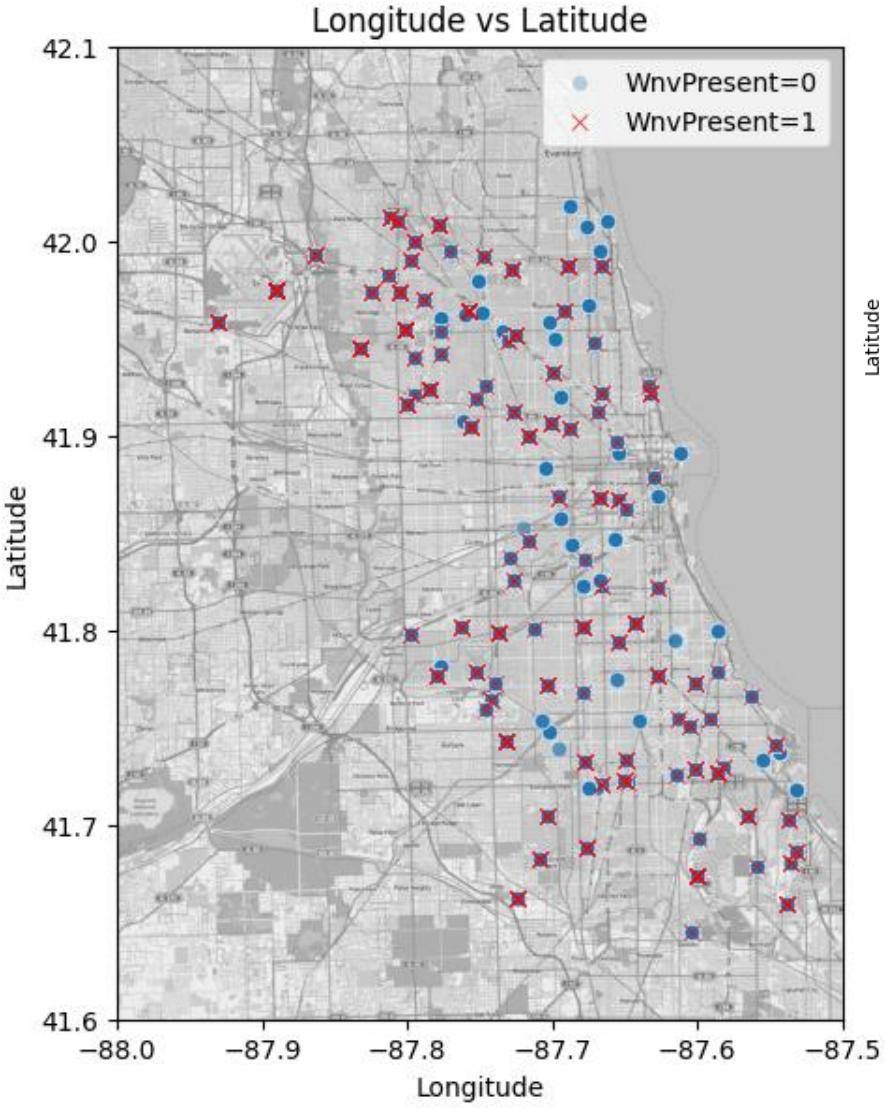
The next step was to create a few new features that could be useful later. These columns include DaytimeLength, RelativeHumidity, DewPointDepression, WetBulbDepression, Day_of_week, day_of_month, Week, Month, Year, and Season, which could be created from the already existing columns.

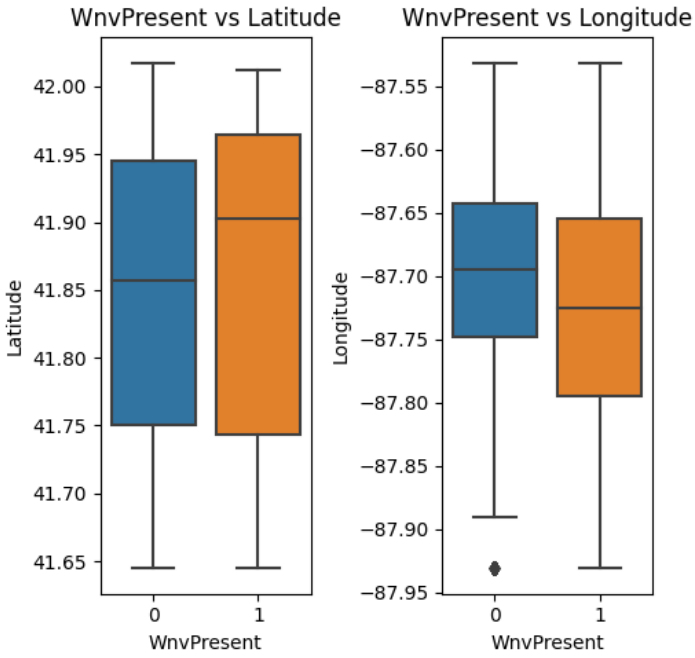
Finally, the train and weather data frames were merged into a single data frame on the Date column.

Exploratory Data Analysis:

There were a few notable discoveries found while exploring each column. The first thing I wanted to do was examine the balance of the classes of my target feature. I found that the data was highly imbalanced with only around 5% of the data indicating the presence of the virus.

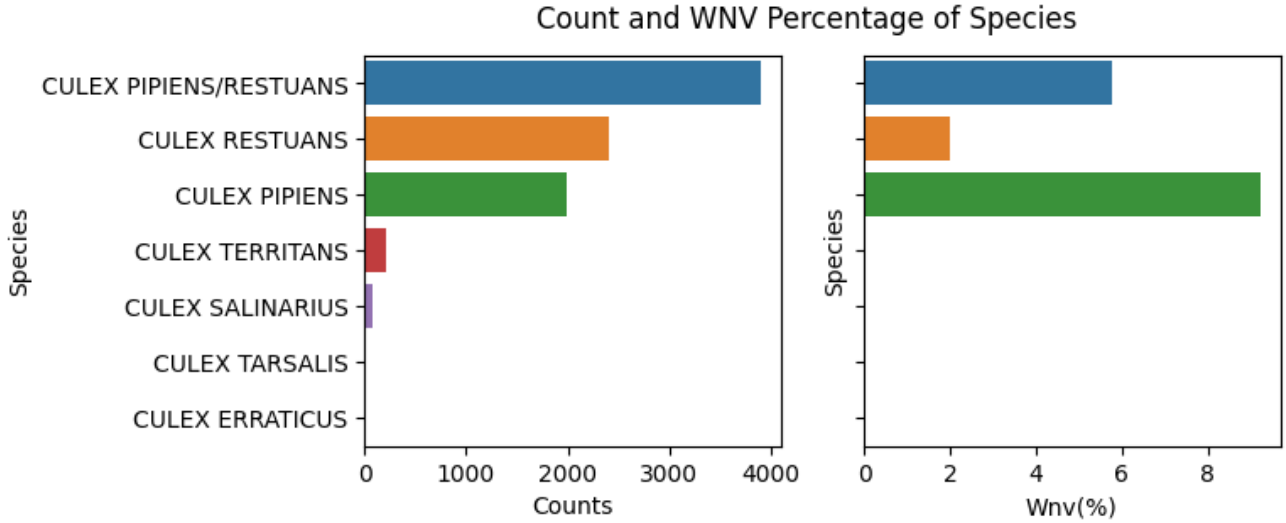




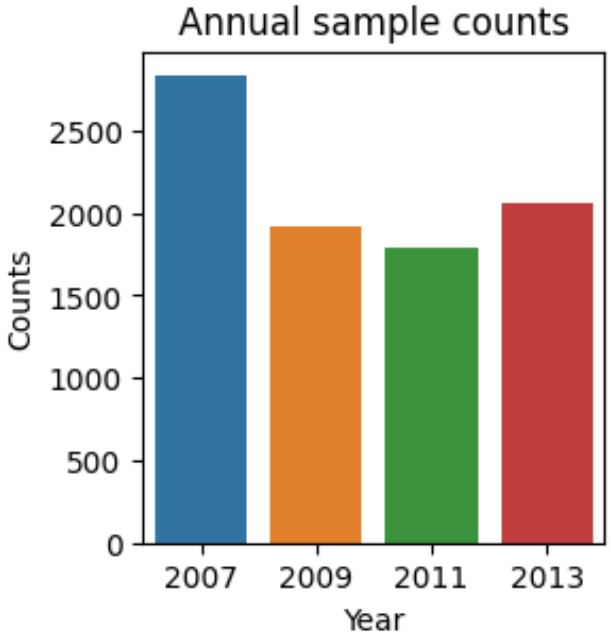


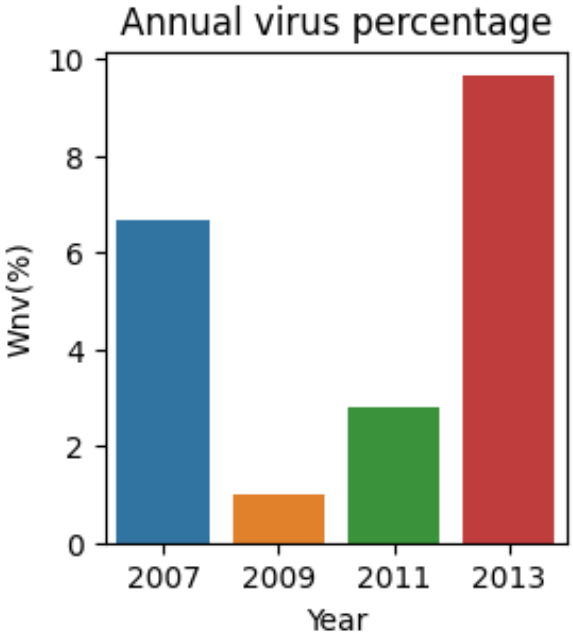
I began analyzing and creating plots for each column to find any notable patterns. Since the one of the end goals is to determine where in the city the virus is most likely to show up, the first columns I took a closer look at was the Latitude and Longitude columns. Combined, these two columns would give an indication about a pattern in location with regards to the presence of the virus. On the plot on the left, each red cross indicates a trap where the virus was present in at least one entry while a blue circle indicates a trap that had at least one entry where the virus was absent. This plot gave no clear indicating pattern for the presence of the virus with regards to location. However, the spread of the data that showed the presence of the virus had a higher Latitude mean and a lower Longitude mean that the data that indicated the absence of the virus.

While analyzing the Species column, I found that out of the five detected species of mosquitos found at the traps, only two of them were found to be carrying the virus, with the species of culex pipiens showing the highest percentage while not being the most common species of mosquito found in the traps.

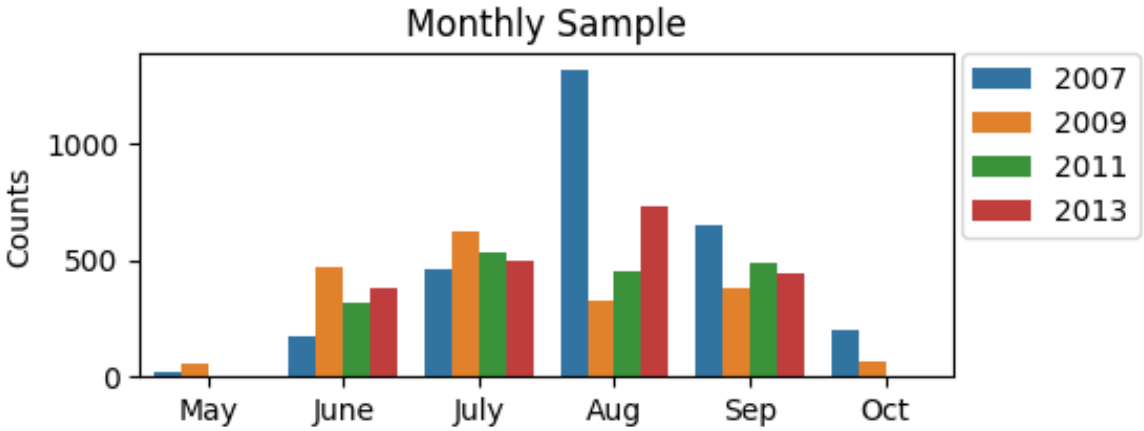


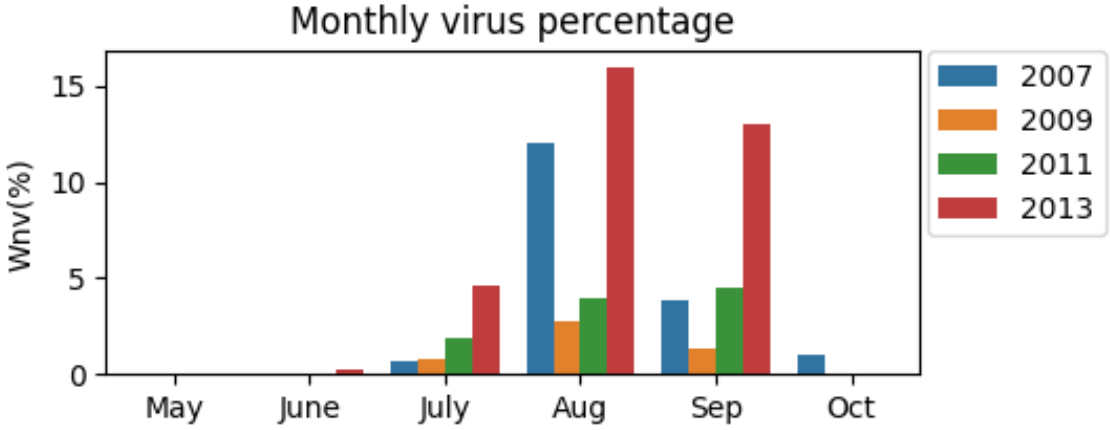
The next step was to find any indication with the seasonality of the virus to see if the virus was more prevalent in certain times. Out of the four years that data was collected from, (2007, 2009, 2011, 2013) The virus was significantly more present in 2013 compared to the other 3 years.





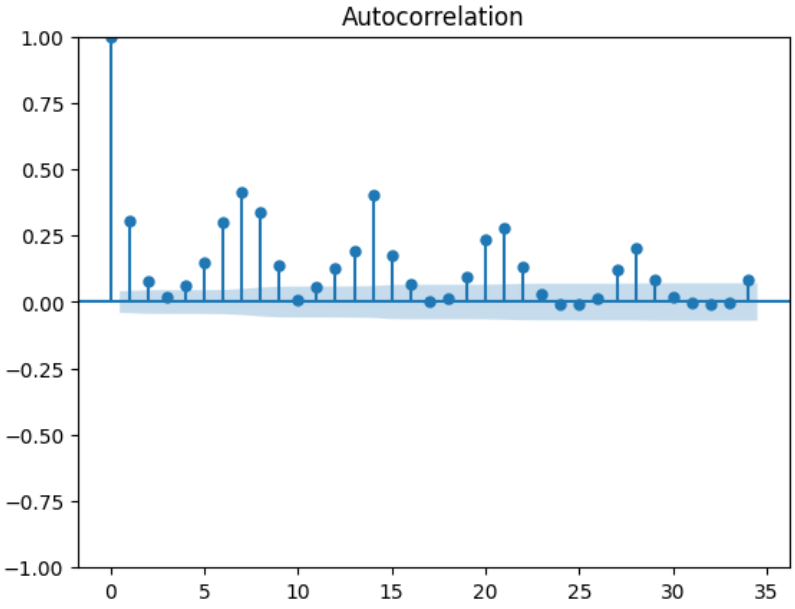
The virus was also very prevalent during August and September of 2007 and 2013, but not clear indicator that there is a general trend of the presence of the virus in those months since our data is very limited to only four years.





Feature Engineering:

Since there are multiple samples collected each day from multiple locations, the virus observation is resampled in the daily time frame in terms of daily virus percentage in order to perform time series analysis. By examining the autocorrelation of the daily virus percentage, the correlation of observation found 7, 14 and 21 days ahead was found to be significantly higher than the other days. These lags were then added as separate features in the dataframe.



Features, such as yearly, seasonal, monthly and weekly percent were also added to the dataframe. There were also a handful of features that were very highly correlated and therefore redundant. These redundancies were removed from the dataset.

Three features, Species, Season, and Month_Name were one-hot-encoded. The species and the months that did not show the virus at all were removed from the dataset.

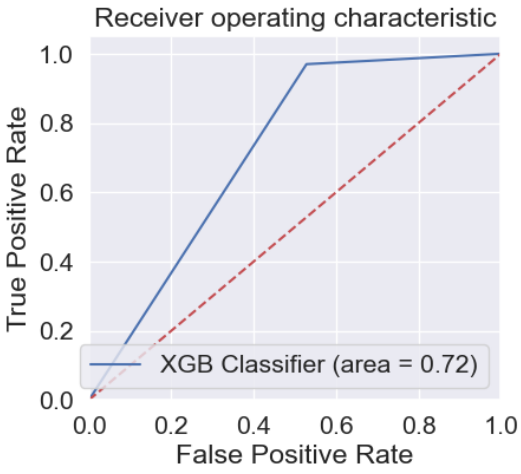
Preprocessing and Training:

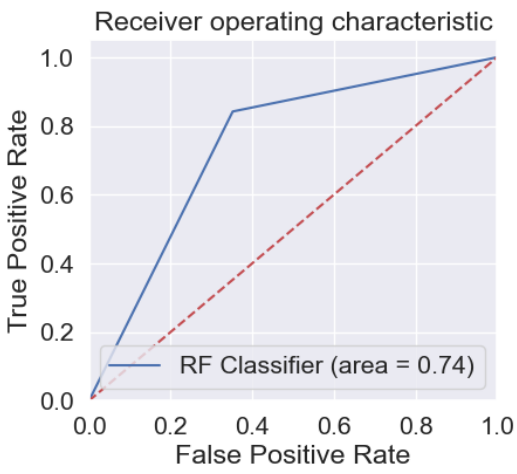
As stated earlier, the data was heavily imbalanced. To address this issue, undersampling of the majority class was conducted to balance the classes of the target feature. A 70-30 train/test split was conducted on the balanced dataset.

To find the importance of each feature, the weight of evidence was conducted to find the information value (IV) of each feature. Features with an IV statistic between 0.1 and 0.8 were selected since features with an IV less than 0.1 would not provide any use and any feature with an IV greater than 0.8 likely lead to biased results. Furthermore, features with a variance inflation factor (VIF) greater than 5 were also removed from the dataset to the degree of multicollinearity. This reduces the number of features to 13. It is important to note that there are no remaining features indicating location.

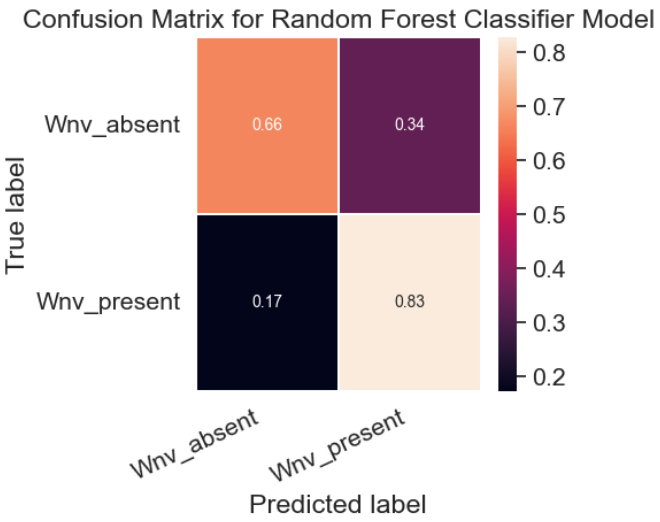
Modeling and Evaluation:

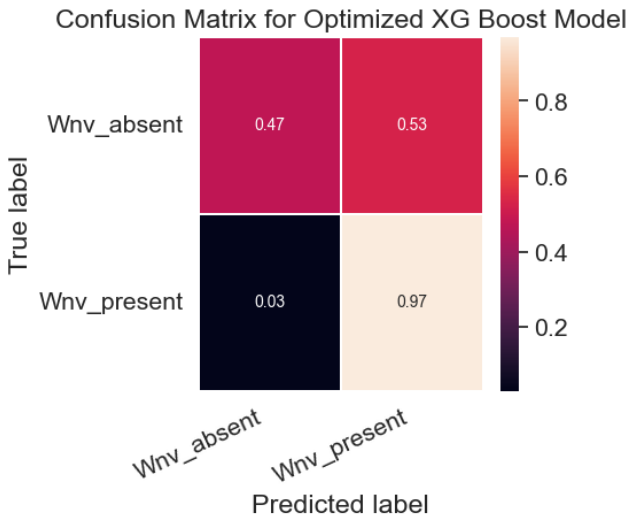
The remaining dataset was then used to fit two potential models, a random forest classifier and a extreme gradient boosting algorithm classifier (XGBoost). The grid search method with five-fold cross-validation and roc_auc scoring function was used on both models to find the optimal hyperparameters for each model. Both models indicated that the seven and fourteen day lag features were strong determining factors in determining the presence of the virus.

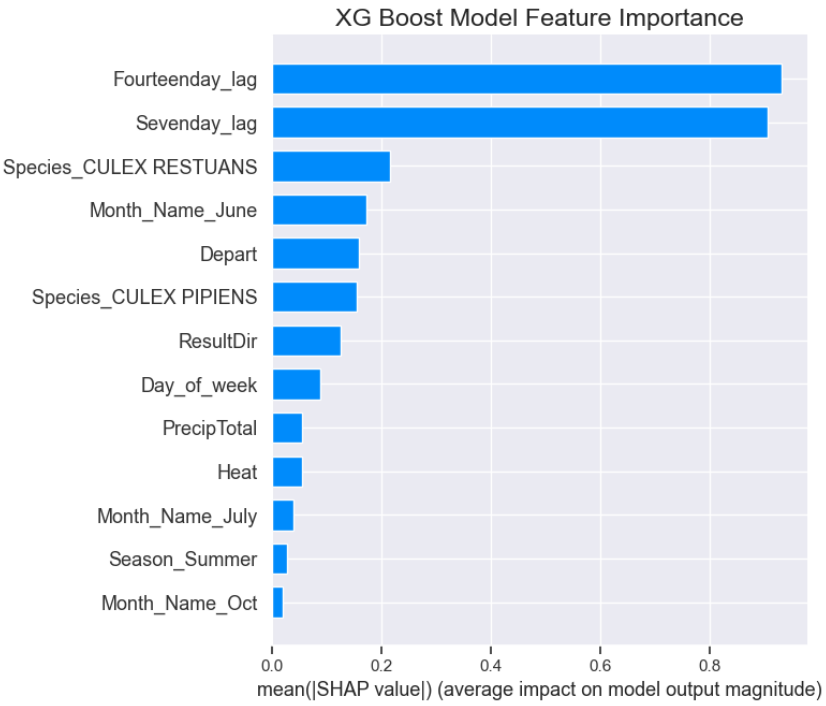




The best random forest model had a score of 0.80 while the best SGBoost model had a score of 0.78. When these two models were then tested on our test data, the random forest model had a score of 0.74 while the XGBoost had a score of 0.72. To further compare the models, confusion matrices were created for each model. Since the accuracy of correctly







classifying the presence of the virus is much more important than correctly classifying its absence and the goal is to create a model that correctly predicts the presence of the virus, we want to find a model that has a high recall score. A high precision is not as important as a high recall in this given situation. The random forest model had a recall score of 0.83 while the XGBoost model had a recall score of 0.97, which means the XGBoost model was much more effective at correctly identifying the presence of the virus.

Conclusion:

The model should be able to correctly predict the presence of the virus since the cost of incorrectly classifying the presence of the virus could lead to an outbreak. Although the random forest model had a slightly higher AUC score than the XGBoost model, The XGBoost model was much better at correctly predicting the presence of the virus with a recall score of 0.97 compared to the recall of 0.83 of the random forest model.

The model indicated that the fourteen and seven day lag value of daily virus percentage were the most important in classifying the virus. With these results, the City of Chicago and the Chicago Department of Public Health should modify their control program to take preventive measures as soon as a fourteen day lag value is positive. It would also be recommended to do the same with the seven day lag values. There was no indication that location was a strong indicator for the presence of the virus.