Chicago - West Nile Virus Prediction
By Eric Kwon

Introduction

The West Nile Virus is a mosquito-transmitted virus. Most people who catch the infection do not develop symptoms and have mild fevers and headaches. However, some people develop a life threatening illness that can affect the spinal cord and brain.

The West Nile Virus was first detected in September 2001 in the city of Chicago and the Illinois Department of Public Health has since maintained a disease surveillance system to monitor the spread of the virus. In this project, we analyze data from the city to assess the current strategy and make data-driven recommendations to further combat the spread of the virus.

> Problem Statement
> Parse data provided by the city of Chicago and create multiple machine learning classifiers to predict the occurrence of the West Nile Virus. Findings from this model should help understand what features should be targeted for mitigation.

Executive Summary

Data provided (train.csv, spray.csv, weather.csv) was explored, analyzed, cleaned, lagged, and ran through several classification models. The Gradient Boost Classifier was found to provide the best performance.

A Shaply analysis from this model further confirms a high correlation between the mosquito population and the West Nile Virus – the best way to decrease the presence of the West Nile Virus is to decrease the mosquito population. Spray data and lagged spray data did not have enough of an appreciable effect and should be further improved.

It is recommended that the City of Chicago increase spray locations to better impact the mosquito population.

Problem Overview

---

1. Start with Problem

   The West Nile Virus causes a potentially fatal fever and is spread by mosquitoes. Using publicly available weather data, mosquito spray data, and mosquito population data from traps, we analyze the current situation and make recommendations for the City of Chicago.

2. Convert Problem into a Classification Problem

   Mosquito data from traps either have the virus (1) or do not (0) which therefore leads to a binary classification problem

3. Solve the Classification Problem with Multiple Methods

   Train multiple machine learning models to identify the most important features

4. Derive Insights from the Best Performing Model

   Perform a Shaply analysis to select the highest impact features

5. Make Actionable and Data-Driven Decisions

   Use findings from the model to make the most effective and sustainable decision in reducing the west nile virus population

Data Wrangling and EDA

Data Definitions

1. Spray.csv
   Contains date of when mosquito spraying was performed, the location o the spray, and overall, gives information on the frequency of sprays
2. Train.csv
   Contains trap data which has information on the location of the trap, the presence of WNV, and the number of mosquitos detected
3. Weather.csv
   Contains dated weather data

EDA of “Train.csv”

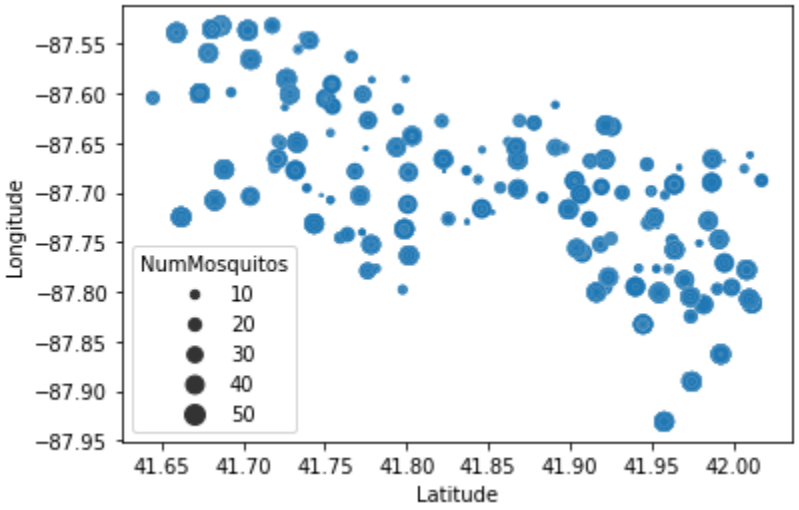
1. **Examine the dataframe:**
   - .info() method to observe column dtypes

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10506 entries, 0 to 10505
Data columns (total 12 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Date                    10506 non-null  object
 1   Address                 10506 non-null  object
 2   Species                 10506 non-null  object
 3   Block                   10506 non-null  int64
 4   Street                  10506 non-null  object
 5   Trap                    10506 non-null  object
 6   AddressNumberAndStreet  10506 non-null  object
 7   Latitude                10506 non-null  float64
 8   Longitude               10506 non-null  float64
 9   AddressAccuracy         10506 non-null  int64
 10  NumMosquitos            10506 non-null  int64
 11  WnvPresent              10506 non-null  int64
dtypes: float64(2), int64(4), object(6)
memory usage: 985.1+ KB
```

   - .isna().sum() method chain to tally up the occurrences of nan values

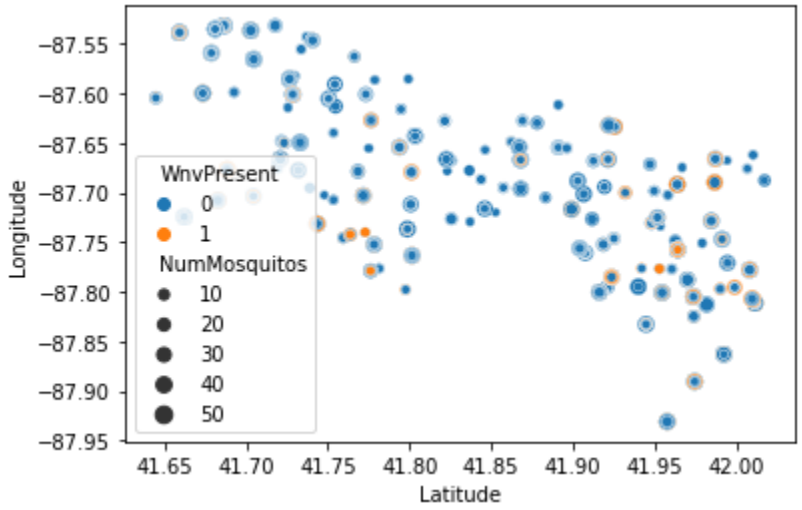
```
Date                        0
Address                     0
Species                     0
Block                       0
Street                      0
Trap                        0
AddressNumberAndStreet      0
Latitude                    0
Longitude                   0
AddressAccuracy             0
NumMosquitos                0
WnvPresent                  0
dtype: int64
```

2. **Use seaborn to visualize the location and magnitude of mosquito data**



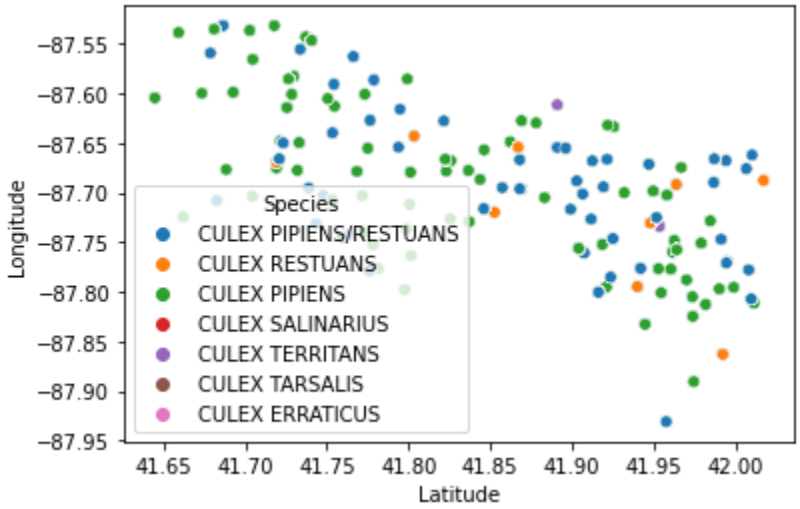
Findings: Mosquito counts are varied and span a wide large vicinity

3. **Use seaborn to visualize the location and magnitude of WNV data**



Findings: The presence of the West Nile Virus is concentrated in specific locations and affects a minority of the entire mosquito population.

4. **Use seaborn to visualize the location and magnitude of data by species**



Findings: Species data is likely not a major factor in the detection of the West Nile Virus.

5. **Groupby Species, WNV, and NumMosquitos - relative population**

```
Species                 WnvPresent
CULEX ERRATICUS         0            1.000000
CULEX PIPIENS           0            0.849365
                        1            0.150635
CULEX PIPIENS/RESTUANS  0            0.892512
                        1            0.107488
CULEX RESTUANS          0            0.971533
                        1            0.028467
CULEX SALINARIUS        0            1.000000
CULEX TARSALIS          0            1.000000
CULEX TERRITANS         0            1.000000
Name: NumMosquitos, dtype: float64
```

6. **Groupby Species, WNV, and NumMosquitos - Absolute**

```
Species                 WnvPresent
CULEX ERRATICUS         0            0.000052
CULEX PIPIENS           0            0.280971
                        1            0.049830
CULEX PIPIENS/RESTUANS  0            0.437985
                        1            0.052748
CULEX RESTUANS          0            0.168574
                        1            0.004939
CULEX SALINARIUS        0            0.001074
CULEX TARSALIS          0            0.000052
CULEX TERRITANS         0            0.003777
Name: NumMosquitos, dtype: float64
```

7. **Data Wrangle Train.csv**
   - Convert date column to a datetime object. This will allow the extraction of month which will be a categorical column
   - Combine columns into a single ‘zipcode’ column using the geolocator geopy api

     Columns to drop:

     ['Address', 'Block', 'Street', 'Trap', 'AddressNumberAndStreet', 'Latitude', 'Longitude', 'AddressAccuracy']

8. **Data Wrangle spray.csv**
   - Convert location columns to a single zipcode column

- Create a two new columns in the spray dataframe: ‘spray-2w’ and ‘spray-1m’
    - These columns will tally up the occurrences of sprays within a 2w or 1m window from the current date.

9. Data Wrangling weather.csv

1. Convert date column to a datetime object
2. Focus on station 1 data
3. Replace “CodeSum” column into multiple categorical columns
4. Normalize Sunset/Sunrise columns which represent the time of occurrence
5. Drop columns and rows with missing values which are labeled ‘M’
6. Impute missing Average Temperature values by finding the mean of the max temp and min. Temp.
7. Lag the weather data. We assume the weather conditions 7d, 2w, and 1mo prior could have an effect on the mosquito population

10. Combine and Merge to Final DataFrame

1. Merge train.csv, spray.csv, and weather.csv with a SQL-like inner join based on the date. Keep lagged dates in mind
2. Uses the pandas ‘.get_dummies’ method for one-hot encoding of categorical columns: ['Species','zipcode','month','Tavg']
3. Create a correlation matrix and only keep correlated features which are defined as to have a minimum correlation threshold of 0.01
4. Check the VIF for each feature and prune such that the VIF value is limited to a maximum of 10.0
5. Use a RandomUnderSampler to focus the model on the presence of the West Nile Virus which is understood to be be a ~10% minority in the dataset

KNN Classifier

We perform a grid search of:

- nearest neighbors between 1-30
- uniform (all equally weighted) vs distance (closer points have higher influence)

Results:

```
From Confusion Matrix:
True Negatives: 1515
False Positives: 898
False Negatives: 20
True Positives: 109
```

	No WNV	WNV Present	accuracy	macro avg	weighted avg
precision	0.986971	0.108242	0.638867	0.547606	0.942377
recall	0.627849	0.844961	0.638867	0.736405	0.638867
f1-score	0.767477	0.191901	0.638867	0.479689	0.738268
support	2413.000000	129.000000	0.638867	2542.000000	2542.000000

Random Forest Classifier

We perform a *Random Search* of:

```
      # Number of trees in random forest
      n_estimators = [int(x) for x in np.linspace(start = 10, stop = 80, num =
10)]
      # Number of features to consider at every split
      max_features = ['auto', 'sqrt']
      # Maximum number of levels in tree
      max_depth = [2,4]
      # Minimum number of samples required to split a node
      min_samples_split = [2, 5]
      # Minimum number of samples required at each leaf node
      min_samples_leaf = [1, 2]
      # Method of selecting samples for training each tree
      bootstrap = [True, False]
```

Results:

```
Fitting 10 folds for each of 10 candidates, totalling 100 fits
From Confusion Matrix:
True Negatives: 1782
False Positives: 631
False Negatives: 26
True Positives: 103
```

	No WNV	WNV Present	accuracy	macro avg	weighted avg
precision	0.985619	0.140327	0.741542	0.562973	0.942723
recall	0.738500	0.798450	0.741542	0.768475	0.741542
f1-score	0.844350	0.238702	0.741542	0.541526	0.813615
support	2413.000000	129.000000	0.741542	2542.000000	2542.000000

We observe an improvement over KNN

Logistic Regression

Results:

```
From Confusion Matrix:
True Negatives: 1828
False Positives: 585
False Negatives: 34
True Positives: 95
```

	No WNV	WNV Present	accuracy	macro avg	weighted avg
precision	0.981740	0.139706	0.756491	0.560723	0.939009
recall	0.757563	0.736434	0.756491	0.746999	0.756491
f1-score	0.855205	0.234858	0.756491	0.545031	0.823724
support	2413.000000	129.000000	0.756491	2542.000000	2542.000000

Gradient Boosting Classifier

We perform a *Random Search* of:

```
params = { "n_estimators":[5,50,250,500],
           "max_depth":[1,3,5,7,9],
           "Learning_rate":[0.01,0.1,0.5,1,10,100]}
```

Results:

```
From Confusion Matrix:
True Negatives: 1805
False Positives: 608
False Negatives: 27
True Positives: 102
```

	No WNV	WNV Present	accuracy	macro avg	weighted avg
precision	0.985262	0.143662	0.750197	0.564462	0.942553
recall	0.748031	0.790698	0.750197	0.769365	0.750197
f1-score	0.850412	0.243147	0.750197	0.546779	0.819595
support	2413.000000	129.000000	0.750197	2542.000000	2542.000000

Ada Boosting Classifier

We perform a Randomized Search:

```
parameters = {'base_estimator__max_depth':[i for i in range(1,10,1)],
              'base_estimator__min_samples_leaf':[5,10],
              'n_estimators':[10,50,250,1000],
              'learning_rate':[0.01,0.1,1,10]}
```

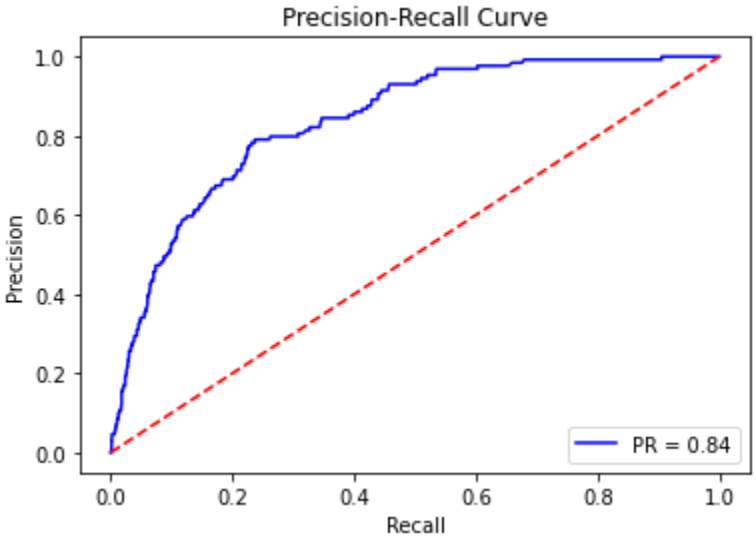
Results:

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
From Confusion Matrix:
True Negatives: 1783
False Positives: 630
False Negatives: 28
True Positives: 101
```

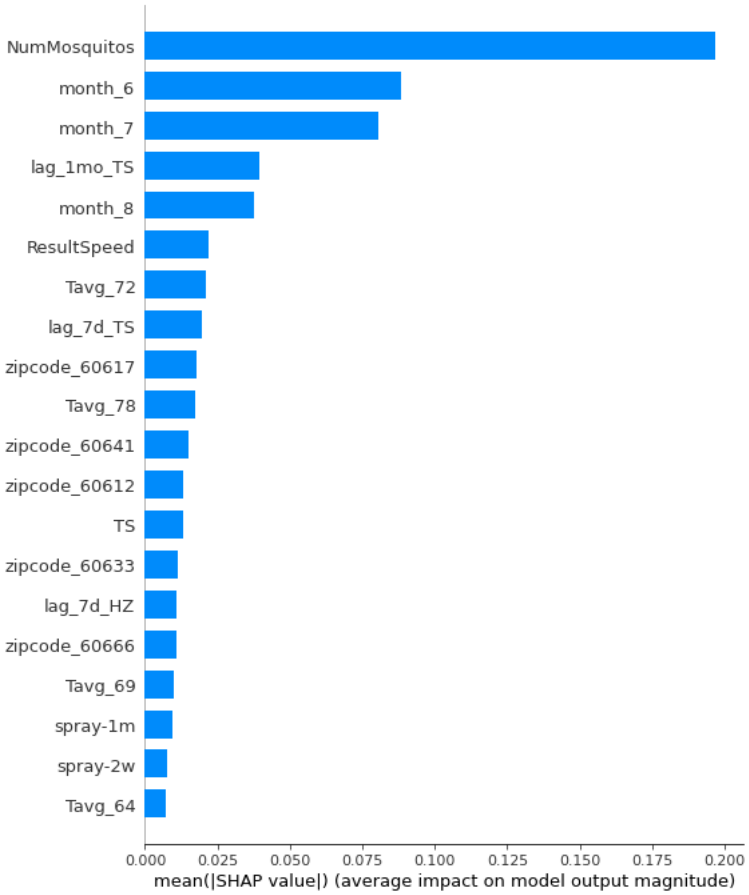
	No WNV	WNV Present	accuracy	macro avg	weighted avg
precision	0.984539	0.138167	0.741149	0.561353	0.941588
recall	0.738914	0.782946	0.741149	0.760930	0.741149
f1-score	0.844223	0.234884	0.741149	0.539554	0.813301
support	2413.000000	129.000000	0.741149	2542.000000	2542.000000

Best-Performing Classifier: Gradient Boosting

ROC-Curve



Shaply Analysis



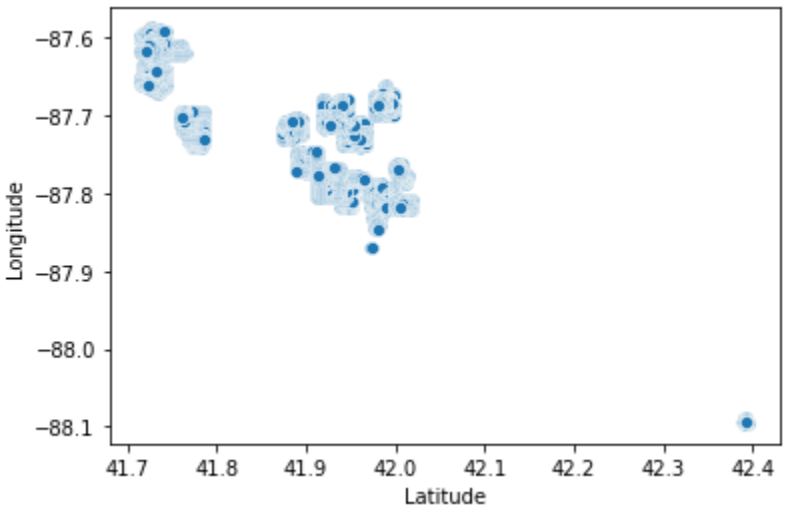
Summary

AUC Score = 0.84
Indicating strong model performance

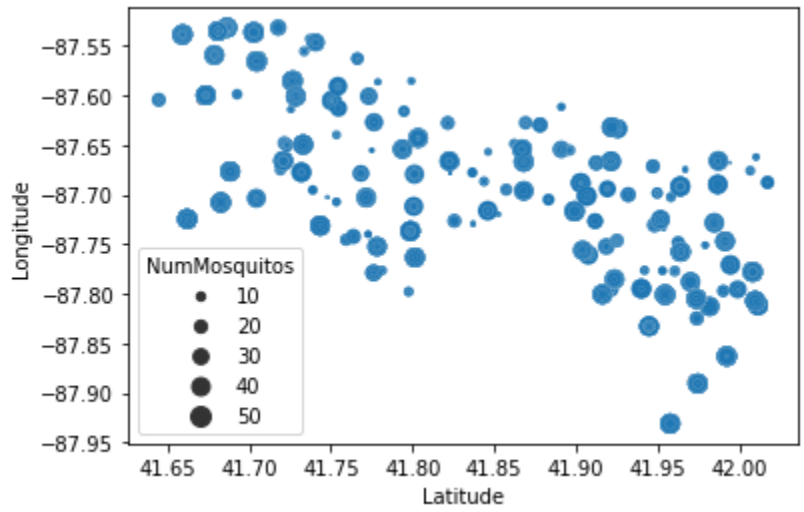
Highest impact features are related to the number of mosquitos, the months of June, July, and August, and whether or not a thunderstorm occurred within the past month 1

We observe the impact of sprays in the past month and past two weeks however the SHAP values are low and plot the spray data itself. We plot spray data and trap data:

Scatter plot of Spray Locations



Scatter plot of Trap Locations



Conclusion

We see from our highest performing classifier model, features related to the population of mosquitos are the highest determining factors in the prediction of the presence of the WNV.

Unexpectedly, we observe low impact from the occurrence of sprays within the past 2 weeks and 1 month. After plotting the spray data with trap locations, we see that the spray locations are highly concentrated and do not cover as large an area as the trap locations.

From the classifier model and the resultant Shapl analysis, it is hence reasonable to assume that distributing mosquito spray locations across a wider area should result in greatly reducing the presence of the West Nile Virus. This is therefore the main recommendation for the City of Chicago.

Possible Improvements for Further Study

To improve the model performance, the following suggestions could be done to enable greater modelling accuracy:

- The model currently utilizes the weather that occurred on a day that is 7 days, 2 weeks, or 1 month prior. It could be beneficial to have a feature that counts the number of occurrences of rain related weather events in these time frames.
- Incorporate data on disease cases in birds and other animals within the area