PREDICTING WEST NILE VIRUS

Group members: Pei Qing, Daniel, Vidhu, Lynn

PROBLEM STATEMENT

- Build a model capable of predicting the outbreak of West Nile Virus in the Chicago locale
  - Executed through predicting the appearance of WNV in the mosquitos themselves (not in humans)
  - This model can be an input for further studies on interaction between infected mosquitos and human / environmental variables
  - Also as an input to the local government to understand which are the high risk areas that require spraying

- Understand factors that have the probability of increasing the probability of the occurrence of West Nile Virus
  - This can be observed through the feature importance of variables in the model
  - Useful for forecasting future outbreak of West Nile Virus which can allow the government to execute early prevention

OUR UNDERSTANDING OF WEST NILE VIRUS

Mosquito Activity

- *More active during hot and humid periods*
- *This will lead to increasing transmission between bird, mosquito and human*

Mosquito Breeding

- *Breeds in stagnant, standing fresh water*
- *Can survive in stasis for >10 years, and will hatch in the right conditions*
- *Ideal conditions to hatch, breed and attack is a consistent 10℃*
- *Mean hatching count is the highest when eggs are held at 22℃ -27℃*

⇒ Prior years' mosquito populations can lead to future outbreaks, especially when there are long periods of drought or cold weather

Presence of infected birds

- *Mosquitoes will need to feed on an infected bird before becoming a disease vector*
- *Preferred species are the American crow, corvids and American robin*

IMPLICATIONS ON FEATURE ENGINEERING

- Built in several weather features into our model, which includes point and rolling averages of humidity, temperature, precipitation, as well as length of day (period between sunrise and sunset)

```
df_weather_A['daylight_hours'] = df_weather_A['Sunset'] - df_weather_A['Sunrise']
df_weather_B['daylight_hours'] = df_weather_B['Sunset'] - df_weather_B['Sunrise']
df_weather_A['daylight_hours'] = df_weather_A['daylight_hours'].apply(lambda x: x.total_seconds()/3600)
df_weather_B['daylight_hours'] = df_weather_B['daylight_hours'].apply(lambda x: x.total_seconds()/3600)
```

```
#The entire life cycle, from an egg to an adult, takes approximately 8-10 days. We'll use 8.
df_weather_A['past_week_tavg'] = df_weather_A['Tavg'].rolling(window = 8).mean()
df_weather_A['past_week_precip'] = df_weather_A['PrecipTotal'].rolling(window = 8).sum()
df_weather_A['past_week_humid'] = df_weather_A['humidity'].rolling(window = 8).mean()
df_weather_B['past_week_tavg'] = df_weather_B['Tavg'].rolling(window = 8).mean()
df_weather_B['past_week_precip'] = df_weather_B['PrecipTotal'].rolling(window = 8).sum()
df_weather_B['past_week_humid'] = df_weather_B['humidity'].rolling(window = 8).mean()
```

```
df_weather_A['past_mth_tavg'] = df_weather_A['Tavg'].rolling(window = 30).mean()
df_weather_A['past_mth_precip'] = df_weather_A['PrecipTotal'].rolling(window = 30).sum()
df_weather_A['past_mth_humid'] = df_weather_A['humidity'].rolling(window = 30).mean()
df_weather_B['past_mth_tavg'] = df_weather_B['Tavg'].rolling(window = 30).mean()
df_weather_B['past_mth_precip'] = df_weather_B['PrecipTotal'].rolling(window = 30).sum()
df_weather_B['past_mth_humid'] = df_weather_B['humidity'].rolling(window = 30).mean()
```

Limitation of the data

- We do not have data for where there are high population of the preferred species of birds that have the West Nile virus to be used as input into our model as potential high risk areas
- This could be approximated by using areas with high density of trees (e.g. forests)

DATA CLEANING PROCESS

Train and test set

Train: 10506 rows, 12 columns

Test: 116293 rows, 11 columns

- *'Species' and 'Trap'* columns made into dummy variables
  - For the Species labeled *'Culex Pipiens/Restuans'*, a '1' was assigned to both *'Culex Pipiens*' and *'Culex Restuans'* columns
- A *'bias'* columns was assigned to represent a high occurrence of West Nile Virus at the particular trap across time
- Assigned weather data to each row through determining the nearest Station by long-lat distance

```
def ungroup_mosq(df):
    for row in range(df.shape[0]):
        if df.loc[row,'CULEX PIPIENS/RESTUANS'] == 1:
            df.loc[row,'CULEX PIPIENS'] = 1
            df.loc[row,'CULEX RESTUANS'] = 1
    df = df.drop(columns = ['CULEX PIPIENS/RESTUANS','Species'],inplace=True)

ungroup_mosq(clean_train)
ungroup_mosq(clean_test)
```

```
def trap_bias(df):
    form = lambda x: np.sum(x)/float(x.count())*100
    transformed = df[['Trap','WnvPresent']].groupby('Trap').agg(form)
    return transformed.reset_index().rename(columns={'WnvPresent':'Bias'})
```

Weather

2944 rows, 22 columns

- Has **7415 null values** throughout the dataset
  - Filled missing values marked 'M' and '-' with a null value to represent missingness
  - Filled values marked 'T' with 0.01 as a dummy value to represent trace amounts
- Dropped columns that are related to winter seasons as they are not important to our analysis
  - Dropped *'Depth', 'Water1', 'SnowFall'*
- Cleaned up *'Sunset' and 'Sunrise'* datasets as time data is not properly stated

```
#Station 1: CHICAGO O'HARE INTERNATIONAL AIRPORT Lat: 41.995 Lon: -87.933 Elev: 662 ft. above sea level
#Station 2: CHICAGO MIDWAY INTL ARPT Lat: 41.786 Lon: -87.752 Elev: 612 ft. above sea level

nearest_station = []
for row, date in enumerate(df_train_1['Date']):
    house_loc = (df_train_1['Latitude'][row], df_train_1['Longitude'][row])
    station_1_loc = (41.995, -87.933)
    station_2_loc = (41.786, -87.752)
    dist_1 = haversine(house_loc, station_1_loc)
    dist_2 = haversine(house_loc, station_2_loc)

    #if distance to station 1 > distance to station 2, append station 2 since it's nearer
    if dist_1 > dist_2:
        nearest_station.append(2)

    else:
        nearest_station.append(1)

df_train_1['Station'] = nearest_station
```

Spray

14835 rows, 4 columns

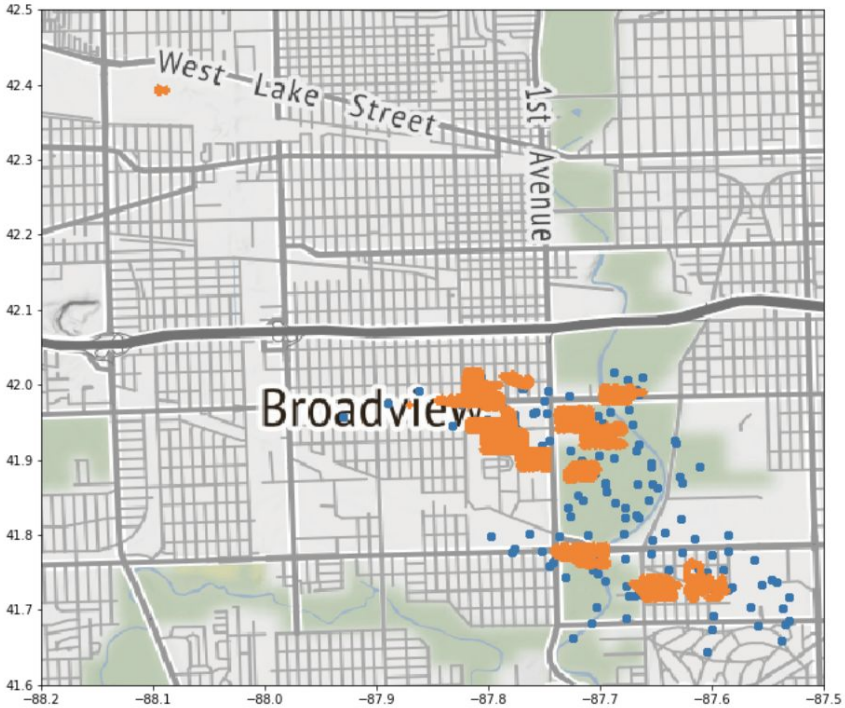
- Dropped **541 duplicate rows**
- Created extra features to take into account the length of effectiveness of each spray and the covered area

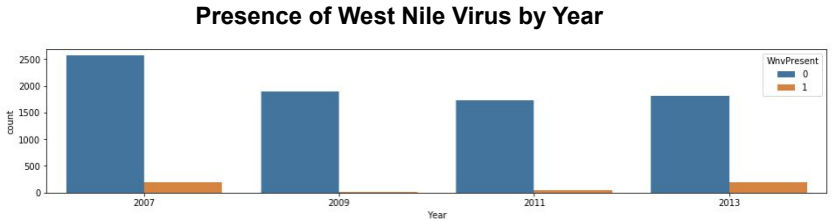
```
    if time == 1860:
        time = datetime.datetime.strptime(str('1900'), "%H%M")

    elif time == 1760:
        time = datetime.datetime.strptime(str('1800'), "%H%M")

    elif time == 1660:
        time = datetime.datetime.strptime(str('1700'), "%H%M")

    else:
        time = datetime.datetime.strptime(str(time), "%H%M")

    return time
```

FEATURE ENGINEERING

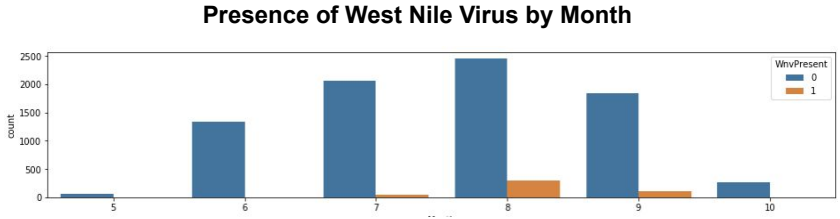
Mosquito Activity	Temperature	1-Day T_Avg, 3-Day T_Avg, Heat, Temperature Difference (Tmax - Tmin)
	Humidity	1-Day T_Avg, 3-Day T_Avg, Heat
	Precipitation	1-Day PrecipTotal, 3-Day PrecipTotal, Weather Code (DZ/TS, etc)
	Others	Clustering by Num Mosquitos, Mosquito Species
Mosquito Breeding	Temperature	1 week T_avg, 1 month T_avg, annual T_avg
	Precipitation	1 week PrecipTotal, 1 month PrecipTotal, annual PrecipTotal
	Wind Speed	
WNV Vectors	Location	Unsupervised clusters / distance from key traps (e.g. airport, lakeside)

EDA



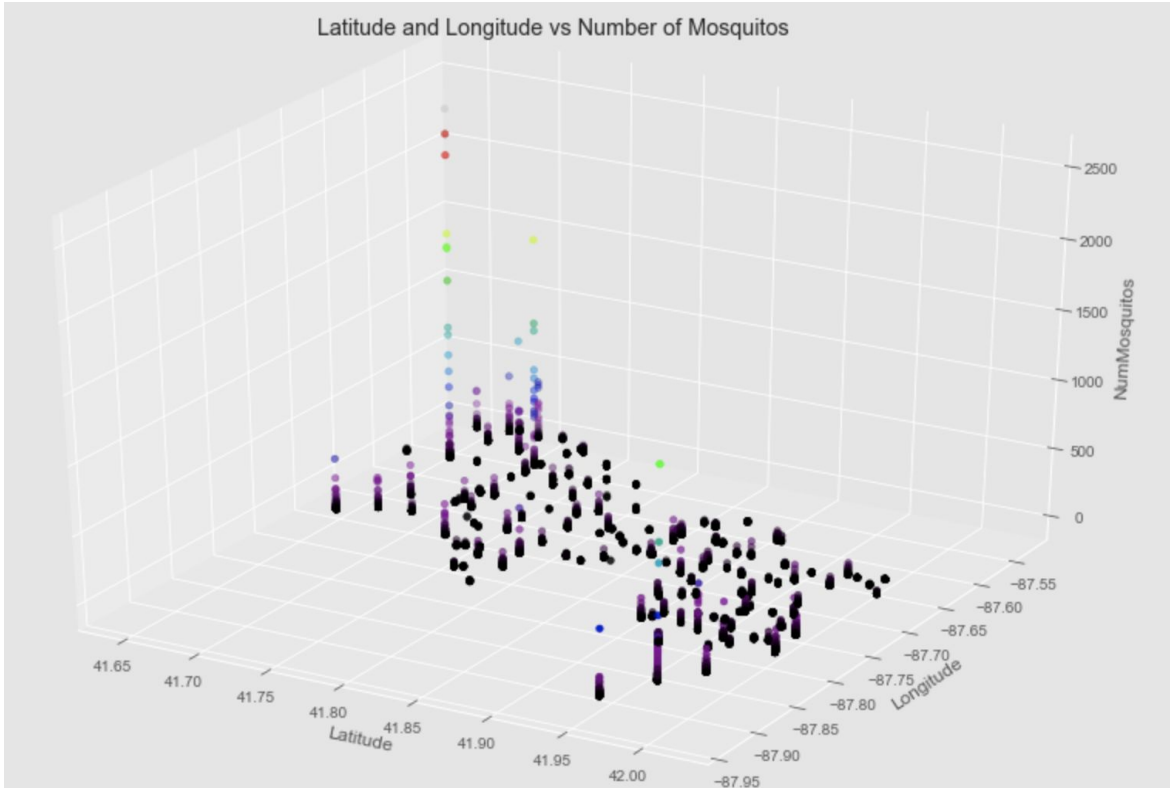


- High amounts of West Nile Virus in 2007 and 2013, compared to 2009 and 2011



- Highest occurrence of West Nile Virus is in the month of August, followed by September and July

No. of mosquitos by Latitude and Longitude



In this aggregated data, there appears to be a large mosquito cluster in the northern area

DATA CLUSTERING

```
geo_cluster_model(MeanShift(cluster_all = True, n_jobs = -1), 'meanshift_mosq', df_os,
                  features = ['Latitude', 'Longitude', 'NumMosquitos'])
```

```
Number of classes: 28
Silhouette Score: 0.8513958634888431
```

	NumMosquitos		WnvPresent	
	mean	count	mean	count
meanshift_mosq_clusters				
0	18.687679	13339	0.494340	13339
1	215.440295	384	0.934896	384
2	371.531713	66	0.954545	66
3	807.700004	23	0.956522	23
4	472.900505	17	0.882353	17
5	1630.985158	28	1.000000	28
6	737.541351	18	1.000000	18
7	551.618466	24	1.000000	24
8	666.965861	18	1.000000	18
9	1013.917849	16	1.000000	16
10	874.896086	13	1.000000	13
11	2324.687469	12	1.000000	12
12	1404.423496	7	1.000000	7
13	944.620502	8	1.000000	8
14	1453.300302	7	1.000000	7
15	1345.540758	8	1.000000	8
16	1148.523184	7	1.000000	7
17	1082.462722	4	1.000000	4
18	1809.160801	7	1.000000	7
19	1709.370442	6	1.000000	6
20	1545.157703	10	1.000000	10
21	1220.896996	4	1.000000	4
22	1916.257712	6	1.000000	6
23	2190.586182	4	1.000000	4
24	2124.259994	4	1.000000	4
25	2395.067911	4	1.000000	4
26	2507.166071	5	1.000000	5
27	2025.210810	4	1.000000	4

The Meanshift clustering does the best job amongst DBSCAN and K-means at separating the WNV areas from the noise.

The pink areas represent areas with low mosquito counts.

DATA RESAMPLING

- Dataset was very unbalanced, with only **457** cases of West Nile Virus out of a total **8475 rows**
  - This would have given us a base case accuracy of ~6%
- Data was rebalanced by up-sampling the minority dataset (where West Nile Virus is present) to match the size of the majority dataset to create a new training set

```
from sklearn.utils import resample
#rejoin train data on index so it can be downsampled to match classes
traindata = X_train.merge(pd.DataFrame(y_train), how = 'left', right_index = True, left_index = True)

#separate minority and majority classes
train_majority = traindata[traindata['WnvPresent'] == 0]
train_minority = traindata[traindata['WnvPresent'] == 1]

#upsample minority class
train_minority_upsampled = resample(train_minority,
                                    replace = True,
                                    n_samples = train_majority.shape[0],
                                    random_state = 42)
#combine classes
train_data_upsampled = pd.concat([train_majority, train_minority_upsampled])
#split back into X_train and y_train
X_train = train_data_upsampled.drop(columns = 'WnvPresent')
y_train = train_data_upsampled['WnvPresent']
```

SUMMARY OF MODELS

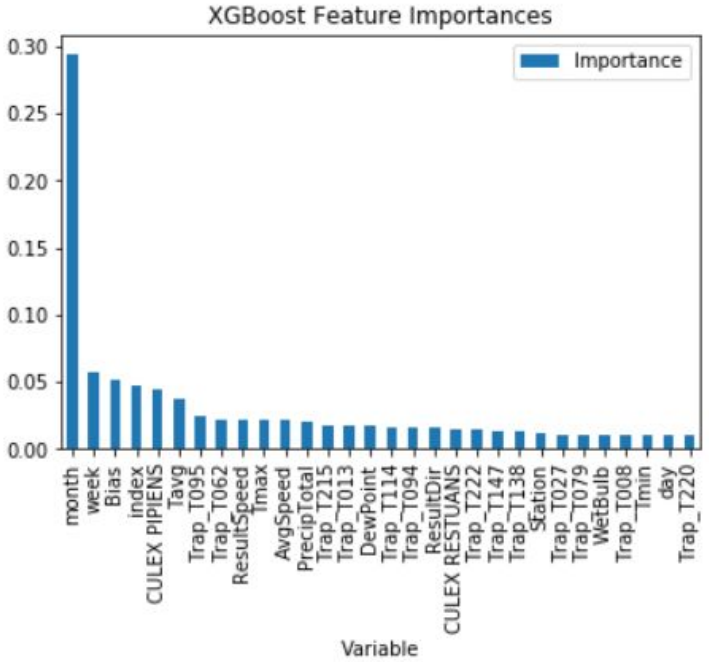
Model	AUC-ROC Score	Recall Score	Kaggle Score	Notes
Logistic Regression	0.833	80.17%	0.70	Used SMOTE
GAM	0.85	1.5%	0.80	No resampling done, recall score is poor
CART (Xgb)	0.90	76.4%	0.77	Resampling on minority class
Clustered CART (Xgb)	0.89	18%	0.77	No resampling

- Logistic Regression and the XGBoost performed best in terms of recall
- GAM performed best in terms of creating a generalizable model (good test AUC ROC, good kaggle AUC ROC)

FEATURE IMPORTANCES (Log Reg vs XGBoost)

Log Reg Feature Importances

feature	coef
PrecipTotal_x	-36.181092
humidity	-8.371265
Species_CULEX TERRITANS	-5.255557
Species_CULEX SALINARIUS	-4.753338
Species_CULEX ERRATICUS	-3.584884
Species_CULEX TARSALIS	-2.104017
daylight_hours	-2.089021
Depart	-0.480403
past_3_precip	-0.415673
DZ	-0.402270

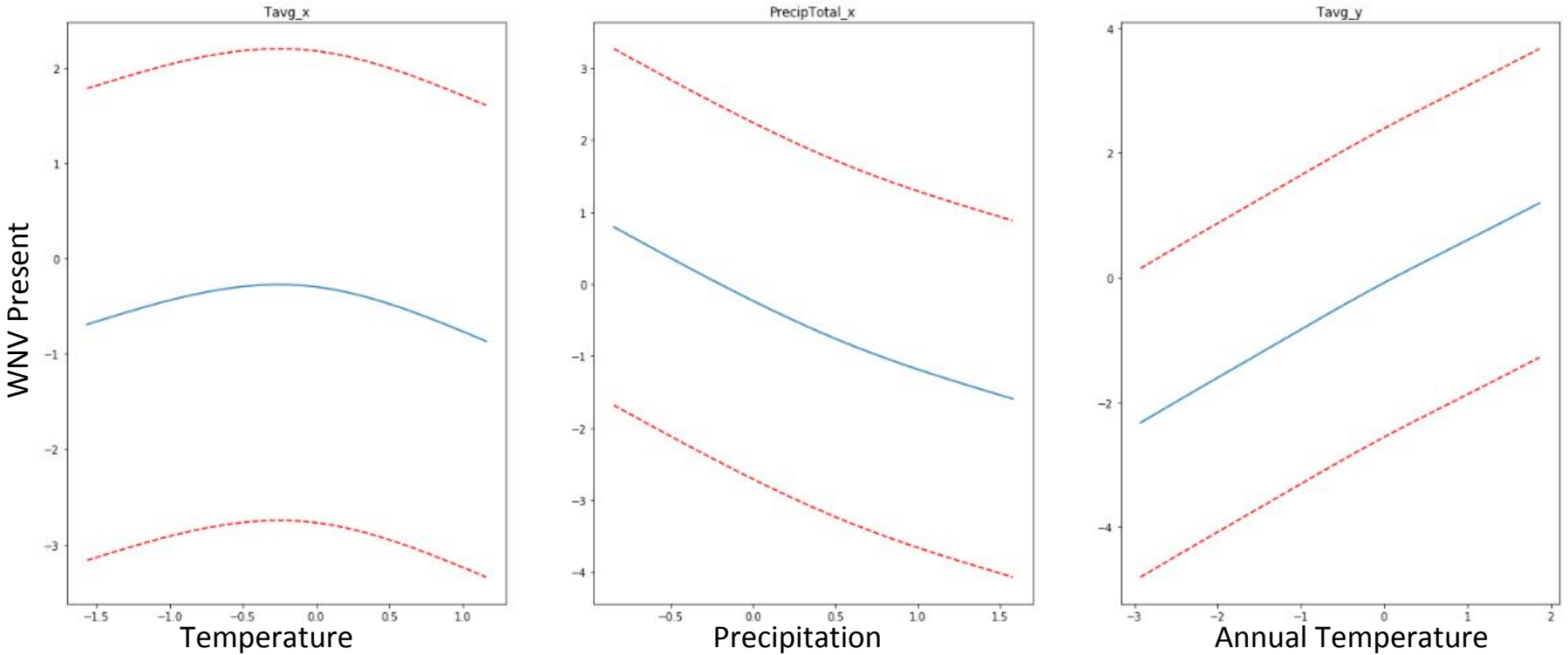


CLUSTERING BASED MODELS

- Cluster based on various unsupervised methods (KMeans, DBScan, MeanShift)
- Predict Number of Mosquitos in Test Set
- Fit KNN Model to train set, and fit to test set to get cluster classes in test set
- Run a bagged model with XGBoost (split by years) to predict WNV

- Bagging performed poorly; simple model based on combined years returned an AUC-ROC Score between 0.74 and 0.79, but the bagged model returned between 0.55 and 0.65.

FEATURE IMPORTANCES (Generalized Additive Model)



CONCLUSION

Weather is a strong predictor of West Nile Virus.

Temperature is important..

…especially temperature over periods of time (*e.g. average over as week, month, and year)*

→ Suggest that parties monitoring mosquito populations track temperatures over time and compare them to historical averages to get a sense of whether the conditions encourage mosquito breeding and activity.

High humidity levels creates a great environment for mosquito activity…

…especially if it is particularly humid for ~3 days in a row

Mosquitos love to breed at high precipitation levels

→ Efforts should be made to wipe out mosquito breeding grounds when high precipitation levels are detected

Stay away from the airport

T900 (the airport) is a large mosquito epicenter. Proximity to T900 correlates with a higher incidence of WNV and mosquito count.

On the whole, we find that weather is a strong predictor of WNV. In particular, we note the following:

1) Temperature is important - but what's important are the temperatures over a period of time (average over a week, month, and year). Higher than average year-round temperatures lead to mosquito outbreaks. We suggest that parties monitoring mosquito populations track temperature at these levels (week/month/year), and compare them to historical averages to get a sense of whether the conditions are right for mosquito breeding and activity.

2) Humidity: High humidity levels create ripe condition for mosquitos. A rolling three day average of humidity is a strong predictor of mosquito activity, and if humidity levels are high, people can be warned to use mosquito repellant or stay away from high risk areas.

3) Precipitation: High precipitation facilitates mosquito breeding. When precipitation levels are high, efforts should be made to wipe out mosquito breeding grounds (stagnant pools of water).

4) For Chicago in particular, T900 (the airport) is a large mosquito epicentre. Proximity to T900 correlates with a higher incidence of WNV and mosquito count. This may be due to a combination of geographic features and man made features, which create an environment conducive to mosquitos.