Capstone Project:
Predict West Nile virus in mosquitos across the city of Chicago
Machine Learning Engineer Nanodegree

Kiyoko Ikeuchi
August 12, 2016

Definition

Project Overview

According to CDC, West Nile virus (WNV) is most commonly transmitted to humans by mosquitoes. The risk of being infected with WNV can be reduced by using insect repellent and wearing protective clothing to prevent mosquito bites. There are no medications to treat or vaccines to prevent WNV infection. Fortunately, most people infected with WNV will have no symptoms. About 1 in 5 people who are infected will develop a fever with other symptoms. Less than 1% of infected people develop a serious, sometimes fatal, neurologic illness.

In 2002, the first human cases of WNV were reported in Chicago. By 2004 the City of Chicago and the Chicago Department of Public Health (CDPH) had established a comprehensive surveillance and control program that is still in effect today.

A Kaggle competition, which was completed on April 22, 2015, asks to predict when and where different species of mosquitos will test positive for WNV. Kaggle is a platform for predictive modelling and analytics competitions on which companies and researchers post their data and statisticians and data miners from all over the world compete to produce the best models [reference]. Data was provided by the Chicago Department of Public Health.

Problem Statement

The goal is to predict the probability of WNV being present in mosquitos. The model will be derived from the input data including 1) Number of mosquitos in a trap (Maximum 50 per entry. Excess of 50 is a new entry), 2) Mosquitos species, 3) Date of capture (Every week from Monday through Wednesday, from late-May to early-October), 4) Location of the traps, and 5) Weather information for the given day. Training data consists of data from 2007, 2009, 2011 and 2013. For each data point, the model should predict the real-valued probability of the virus presence, rather than present or absent.

Ability to predict when and where different species of mosquitos will test positive for WNV will help the City of Chicago allocate resources in preventing transmission of this virus more efficiently and effectively. The results of these tests will determine when and where the city will spray airborne pesticides to control adult mosquito populations.

Training dataset is provided with the trap and mosquito information. Virus presence is labeled 0 (absent) or 1 (present). Weather dataset was extracted from NOAA at 2 weather stations in Chicago. The general strategy is to combine the training data and the weather by day. This combined input data will be split into a training dataset and test dataset. The prediction model will be learned by fitting the training dataset with statistical algorithms, including Naïve Bayes, Decision Tree Classifier and Support Vector Machine (SVM). The model will predict the virus presence probability based on mosquito information, capture time and location, and the day's weather. The prediction will be real values between 0 and 1. Model validity is ensured by cross-validating with the test dataset. Furthermore, cross-validation will be performed on multiple data splits. The final performance will be the average of multiple cross-validations.

Metrics

In accordance with the Kaggle requirements, model performance will be measured by area under the Receiver Operating Characteristic (ROC) curve between the predicted probability that WNV is present and the observed outcomes. ROC would be a good metric especially for the problem with a class skew of the applied data set [reference].

ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity, or recall in machine learning. The area under the curve is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one [reference].

In problems such as detecting the rare occurrence of a virus or a disease, the true positive rate is much lower than the true negative rate in this problem. A Higher score would indicate optimal TPR. The area under the curve of cumulative distribution function would further quantify the TPR against FPR. A score of 0.5 would mean that the model is as good as random guesses.

Analysis

Data Exploration

Training data provided by Kaggle includes data from 2007, 2009, 2011 and 2013. There are 12 columns and 10,506 rows. Sample data points are as below:

	Date	Address	Species	Block	Street	Trap	AddressNumberAndStreet	Latitude	Longitude	AddressAccuracy	NumMosquitos	WnvPresent
0	2007-05-29	4100 North Oak Park…	CULEX PIPIENS/RESTUANS	41	N OAK PARK AVE	T002	4100 N OAK PARK AVE, Chicago, IL	41.954690	-87.800991	9	1	0
1	2007-05-29	4100 North Oak Park…	CULEX RESTUANS	41	N OAK PARK AVE	T002	4100 N OAK PARK AVE, Chicago, IL	41.954690	-87.800991	9	1	0
2	2007-05-29	6200 North Mandell Avenue …	CULEX RESTUANS	62	N MANDELL AVE	T007	6200 N MANDELL AVE, Chicago, IL	41.994991	-87.769279	9	1	0
3	2007-05-29	7900 West Foster …	CULEX PIPIENS/RESTUANS	79	W FOSTER AVE	T015	7900 W FOSTER AVE, Chicago, IL	41.974089	-87.824812	8	1	0
4	2007-05-29	7900 West Foster…	CULEX RESTUANS	79	W FOSTER AVE	T015	7900 W FOSTER AVE, Chicago, IL	41.974089	-87.824812	8	4	0

Above revealed redundant address information. ‘Address’ and ‘AddressNumberAndStreet’ can be dropped and be represented in ‘Block’, ‘Street’ and ‘Trap’. The table below shows the Statistics for the remaining numerical columns. There is no apparent outlier point. ‘AddressAccuracy’ can also be dropped because >75% of the data have a score >= 8. It is also observed that the virus is present in <25% of the data. True positive is unbalanced.

	Block	Latitude	Longitude	AddressAccuracy	NumMosquitos	WnvPresent
count	10506	10506	10506	10506	10506	10506
mean	35.68	41.84	-87.70	7.82	12.85	0.05
std	24.34	0.11	0.10	1.45	16.13	0.22
min	10	41.64	-87.93	3	1	0
25%	12	41.73	-87.76	8	2	0
50%	33	41.85	-87.69	8	5	0
75%	52	41.95	-87.63	9	17	0
max	98	42.02	-87.53	9	50	1

Weather dataset was also provided for 2007 through 2014. Data description was provided by NOAA and its description indicated that the missing values are denoted as ‘M’, ‘-‘ and a blank space. Weather dataset was loaded into pandas data frame with missing data being replaced by NaN. Weather dataset consists of 22 columns and 2944 data points. Columns include 'Station', 'Date', 'Tmax', 'Tmin', 'Tavg', 'Depart', 'DewPoint', 'WetBulb', 'Heat', 'Cool', 'Sunrise', 'Sunset',

'CodeSum', 'Depth', 'Water1', 'SnowFall', 'PrecipTotal', 'StnPressure', 'SeaLevel', 'ResultSpeed', 'ResultDir', and 'AvgSpeed'.

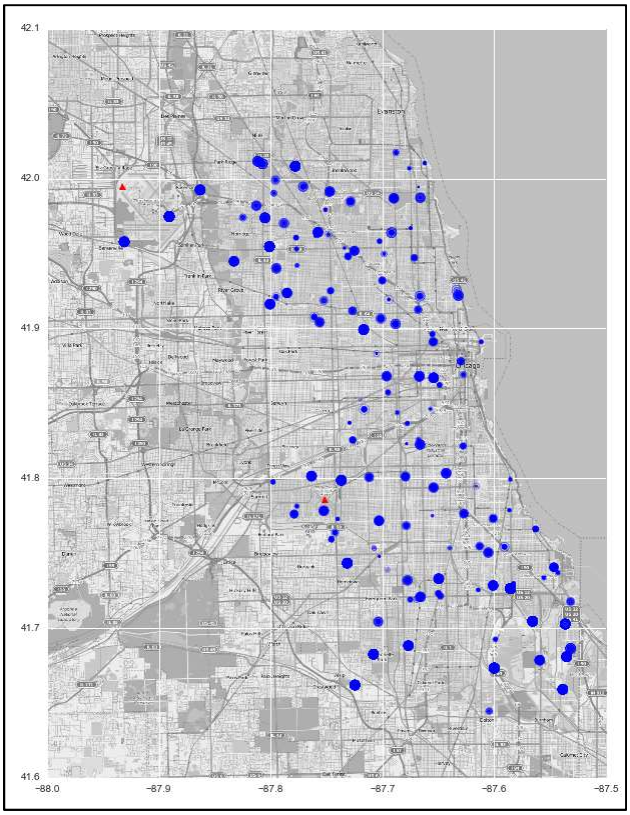
Data types were evaluated for both training and weather datasets. Training dataset is all integers or real numbers except Date, Species, Street and Trap. This will need to be addressed during the data preprocessing. Weather data also contains non-integer/real numbers in Date, CodeSum, SnowFall, and PrecipTotal. In order to address these items, missing values were inspected by the weather station. The table below shows all columns with missing values. CodeSum and Water1 are missing many, if not all, from both stations. These columns will be excluded. Missing data from Depart, Sunrise, Sunset, Depth, and SnowFall were all from Station2. In fact, no data is available for Station2. This is most likely because the values don't expect to differ from Station1. These values will be excluded after the weather dataset is split by Station and re-combined by Date. 2 PrecipTotal points are due to 'T' for trace. They were replaced with 0.0. Other small number of missing data were imputed with the median values. Data distribution was checked to ensure that the imputation did not affect the overall data distribution.

Station	Tavg	Depart	WetBulb	Heat	Cool	Sunrise	Sunset	CodeSum	Depth	Water1	SnowFall	PrecipTotal	StnPressure	SeaLevel	AvgSpeed
1	0	0	3	0	0	0	0	805	0	1472	0	0	2	5	0
2	11	1472	1	11	11	1472	1472	804	1472	1472	1472	2	2	4	3

Common ID for train and weather datasets is Date. This column was converted to DateTime data type. Furthermore, Year, Month and Day attributes were added as new features to distinguish the effect by year, month or day.

Exploratory Visualization

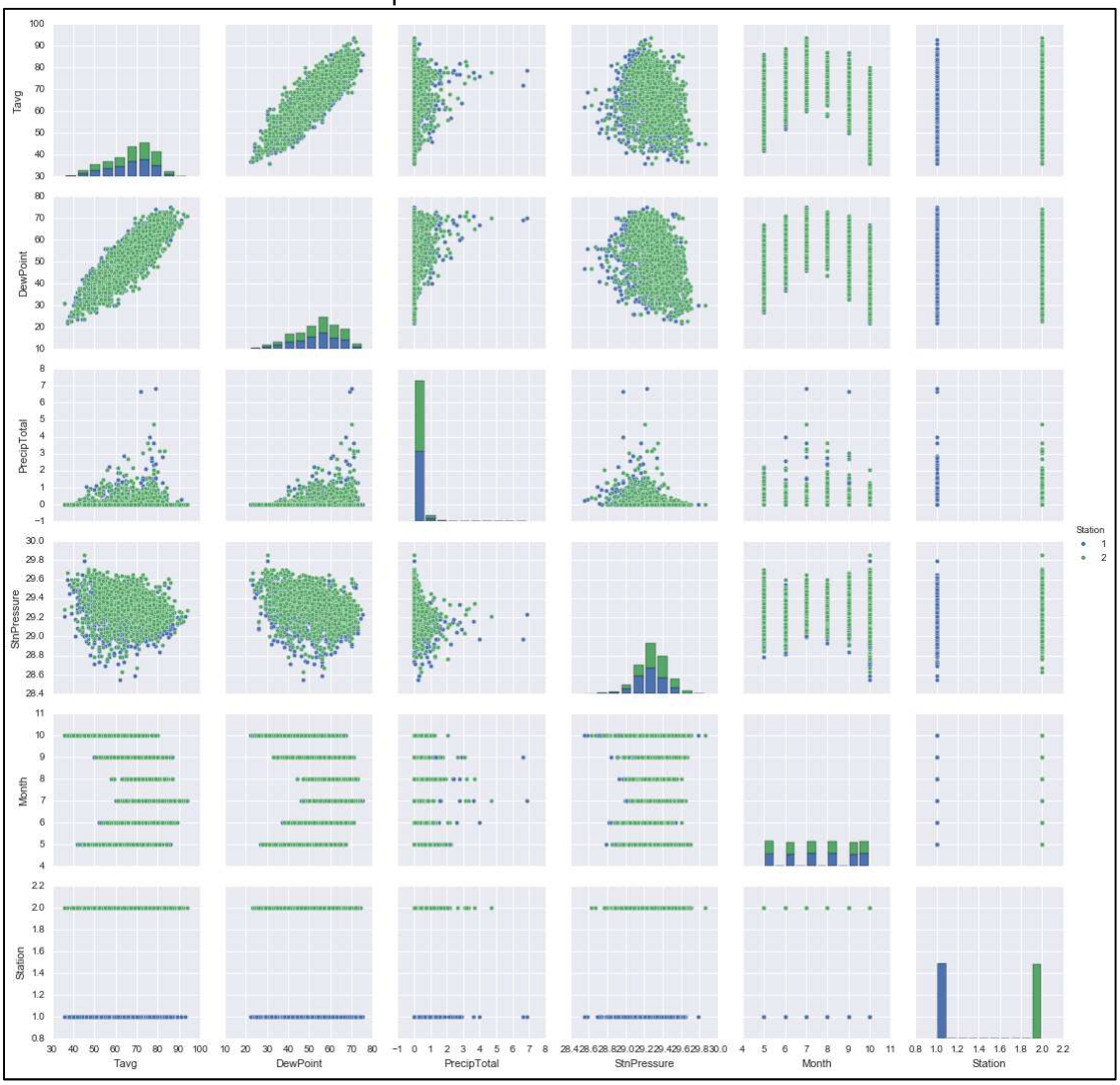
First, a map was created with the location of Trap and Weather Stations. Traps marked in blue circles are sized based on the number of mosquitos. The map provides spatial information about the location of the traps and mosquito concentration. Traps are scattered throughout the city. Number of mosquitos seem to be randomly distributed throughout the city. Red triangles are the weather stations. They are located at North-West corner and center-West of the city. The distance between the station is ~17 miles. Although it is not expected to show much difference, station information will be preserved for the analysis.



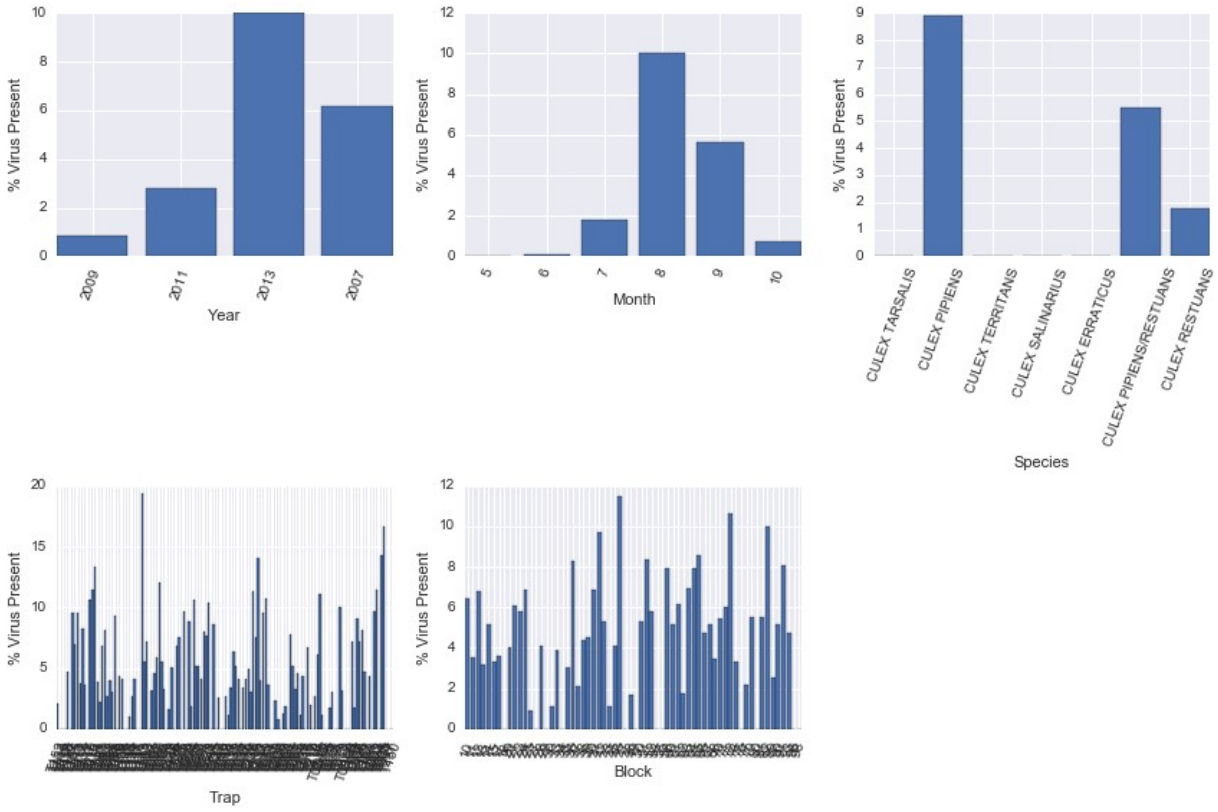
Blue = Trap location. Bubble size indicates number of mosquitos
Red = Weather Stations

Next, weather dataset was examined. The purpose was to see the value range and data distribution, as well as correlation from a column to others. Included in the pairplot below are the temperature aspect of weather data and their monthly variation by the weather stations. It confirms that there are no apparent outlier points in the parameters plotted. Furthermore, data is evenly distributed among the stations. Temperature and DewPoint are highly correlated to one another. They also show slight variation by month. However, precipitation and pressure do not correlate to other parameters or the month. This indicates that the temperature-related parameters can potentially be reduced to fewer features.

Pairplot of selected weather features



Finally, virus occurrence was explored in the training data. This provides the initial indication of the parameters affecting the virus presence. Histograms below show % positive WNV by various features. Virus presence showed noticeable variation by year, month and species. Virus presence was reduced in 2009 and 2011 from 2007 but showed a surge in 2013. Also, more mosquitos were infected in August and September. It also seems that there are some species that are more prone to infection than others. Infected mosquitos found in traps and the street blocks showed large variation between 0-20%, although the previous map did not show a trend by location.



Algorithms and Techniques

Gaussian Naive Bayes (NB) model is a supervised learning algorithm based on Bayes' rule. The model learns most probably hypothesis given data and domain knowledge. "Naive" arises from the independence between feature pairs. In Gaussian NB, the likelihood of the features is assumed to be Gaussian. It can be applied to classification problem modeled from labeled data. The model is being used in text classification, spam filtering, hybrid recommender system, and online applications [reference].

The advantage is speed. Because each distribution can be independently estimated as a one-dimensional distribution, it requires a small amount of training data to estimate the necessary parameters. This, in turn, helps to alleviate problems stemming from the curse of dimensionality. However, the model tends to perform poorer than other models [reference].

Gaussian NB was considered to handle noisy input variables. Specifically, virus presence by trap and its location were noisy in the initial investigation discussed previously. NB is robust to noisy data because the noise is averaged out when estimating the conditional probabilities from all dataset [reference].

Decision Trees are a non-parametric supervised learning method used for classification and regression. The model makes a prediction by learning decision rules inferred from the data features. Decision tree algorithm is used in wide variety of industries, from medical diagnosis to manufacturing [reference].

The advantages are that the model is simple and versatile to input data. It is fast and easy to visualize. It can also handle mixed data types and un-preprocessed inputs. The disadvantage is that it can overfit the data, especially if the data is noisy [reference].

Decision Trees was considered because the input data is processed minimally. It is also simple and fast. In case Decision Trees result in overfitting, Random Forest Classifier was also considered to reduce its susceptibility. It is an ensemble of a bagging of decision tree learners. The model fits a decision tree classifier on a subset of the dataset and uses averaging to mitigate overfitting.

Support Vector Machine (SVM) is another supervised learning algorithm. SVM models the labeled data into categories separated by the maximum margin. Input data will be mapped in the model and it predicts its category based on the location from the margin. The Large margin is achieved by the hyperplane that has the largest distance to the nearest training data point of any class. Large margin means less generalization error or underfitting. It can be applied to classification or regression problems.

The advantage is that it is effective in high dimensional spaces. It is also memory-efficient and can work with non-linear classification using the kernel trick. The disadvantage of support vector machines is that it is not efficient with large noisy data because noise reduces the margin. For these reasons, SVM performs well with text characterization. It is high dimensional separable features [reference]. Other applications are handwritten digit recognition, image based gender identification, and topic-drift in page-ranking algorithms [reference].

SVM was considered because the input data has a high dimension with 40+ features. There may be difficulties with this algorithm due to noise. Also, because some features are not easily separable, for example, trap locations.

Input data containing mosquito test results and weather information will be split into a training dataset and test dataset. The prediction model will be learned from the training dataset using aforementioned statistical algorithms. The model will be cross-validated using the test dataset. In order to ensure the model validity, performance will be determined from the average of multiple cross-validation scores.

Benchmark

The table below shows the summary of the public leaderboard on Kaggle. Out of 1305 teams in Kaggle, scores ranged from 0.38 to 0.89. Mean and Median were 0.72 and 0.74 respectively. These were obtained by the prediction for the separate test dataset for the years, not in the training dataset. This project will use part of training dataset to cross-validate the performance. This may incur discrepancy in the result. Nonetheless, the goal is to achieve above 0.74.

count	1305
mean	0.72
std	0.09
min	0.38
25%	0.69
50%	0.74
75%	0.79
max	0.89

Methodology

Data Preprocessing

Data preprocessing was necessary based on the earlier data exploration. It was noted that the training dataset contained non-numerical columns. Additionally, Depart, Sunrise, Sunset, Depth and SnowFall columns in the weather dataset were available only from Station1.

First, Station1 and 2 data in the weather dataset were combined by Date. As a result, each date will have one set of weather information. Station1 will have "_x" added to the column name. Likewise, Station2 columns will have "_y".

Above weather data was then merged with the training dataset by Date. In the merged dataset, 'Date', 'Sunrise_y', 'Sunset_y', 'Depart_y', 'SnowFall_y', 'Depth_y', 'Year_y', 'Month_y', 'Day_y' are redundant and were dropped.

In order to address non-numerical features, label encoder was used to keep the feature size reasonable. Especially since there are 136 traps on 128 streets, feature sizes will be large if we use one-hot encoder. Label encoder assigns continuous integer and it can be misleading. However, the algorithms described previously should not be affected by the numerical sequence of the label encoder. Finally, "WnvPresent" column was defined as labels and all other columns as features.

Implementation

As mentioned in the previous section, algorithms applied were: Gaussian Naïve Bayes, Decision Tree classifier, Random Forest classifier and SVM. Features and labels were defined as described in the previous data preprocessing section. Data was split into 80% training data and 20% test data. Machine learning algorithms were fit to the training dataset to model prediction. Test dataset was used to cross-validate the model. The performance of the model is determined from the cross-validation result. The model should fit the training dataset just as much as unseen labels in the test dataset. In order to determine more accurate performance score, cross-validation was conducted 10 times on a different set of test data. The final score was the average of those 10 runs. Scores were reported as ROC AUC, according to the designated metric for this problem.

Gaussian Naïve Bayes and Random Forest Classifier performed similarly, while Decision Tree Classifier and SVM performed more than 0.1 points lower. The fact that the Random Forest Classifier performed better than the Decision Tree Classifier suggest that the model is prone to overfitting. SVM took in order of minutes to fit. It also scored poorly and it was excluded from the further analysis.

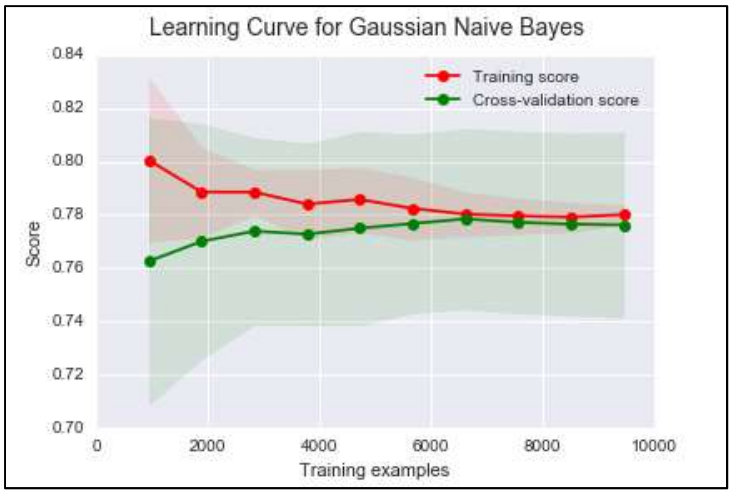
Algorithm	Score
Gaussian Naïve Bayes	0.78
Decision Tree classifier	0.64
Random Forest classifier	0.77
SVM*	0.61

* computation time in order of minutes; excluded from the main code.

Refinement

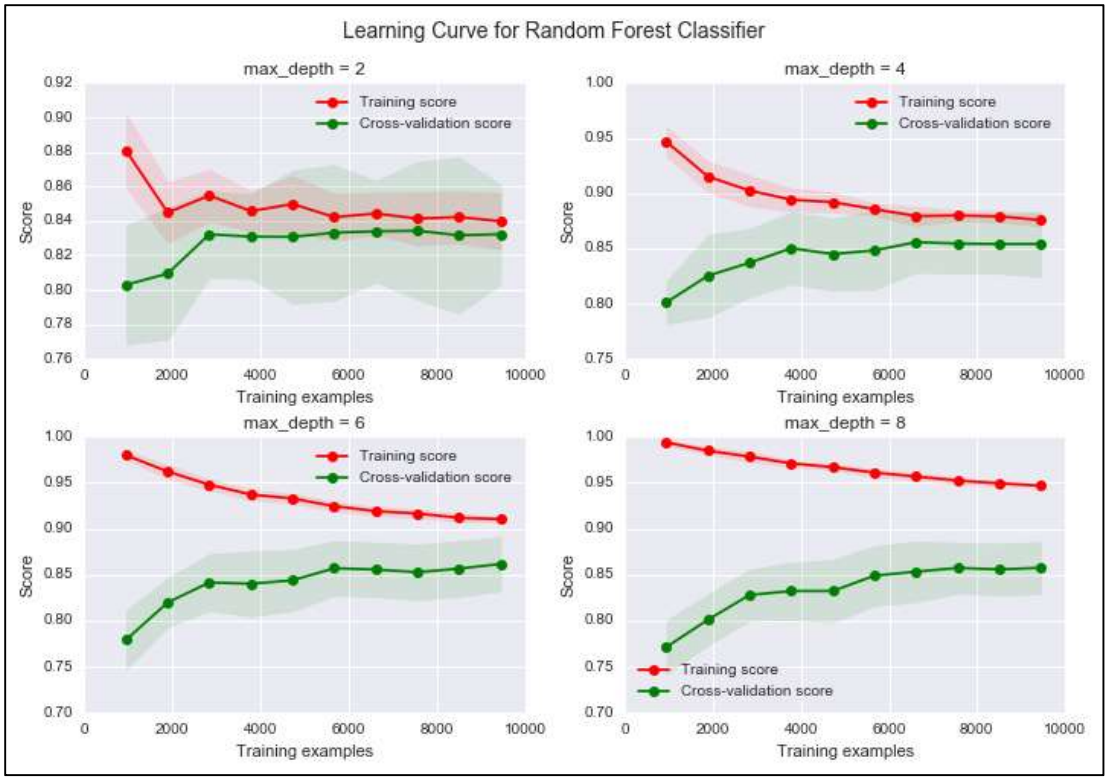
Learning curve determines cross-validated training and test scores for different training set sizes. A cross-validation generator splits the whole dataset k times in training and test data. Subsets of the training set with varying sizes will be used to train the estimator and a score for each training subset size and the test set will be computed [reference].

Below is the learning curve showing how the training size affects the Gaussian Naïve Bayes model performance. The line is the average over 10 runs for each training subset size. Training size ranged from 10 to 100% of the total training data points. The shaded region is the standard deviation of the mean.



The cross-validation score increased slightly up to 60% of the training size, then plateaued. Standard Deviation for the cross-validation is large for all training sizes. Since both scores plateaus at ~0.78, adding training data points above 60% will not improve the model. Furthermore, the model has a bias which is preventing both scores to be higher. Overall, the impact of the training size was minimal.

Random Forest Classifier showed the similar performance as the Naïve Bayes model. The model performance was investigated over the model complexity through the maximum depth of the tree. The purpose was to see if there is any opportunity to perform better than Naïve Bayes. Later doesn't have many parameters that can be tuned further. Graphs below show the learning curves for 4 depths between 2 and 8.



At low depth, the model suffers from some bias because both training score and CV converge to 0.84. As the depth increases to 4 or 6, the CV score attained the score above 0.85. At the maximum depth of 8, the model is overfitted. Training score 0.95 while the CV data remains at ~0.85. This indicates that the model is prone to overfitting. With the parameter tuning, the Random Forest Classifier can achieve the score above 0.85, which is in the 75 percentile of the benchmark. The optimum performance is expected with the maximum depth of 4 or 6 with 70% training size.

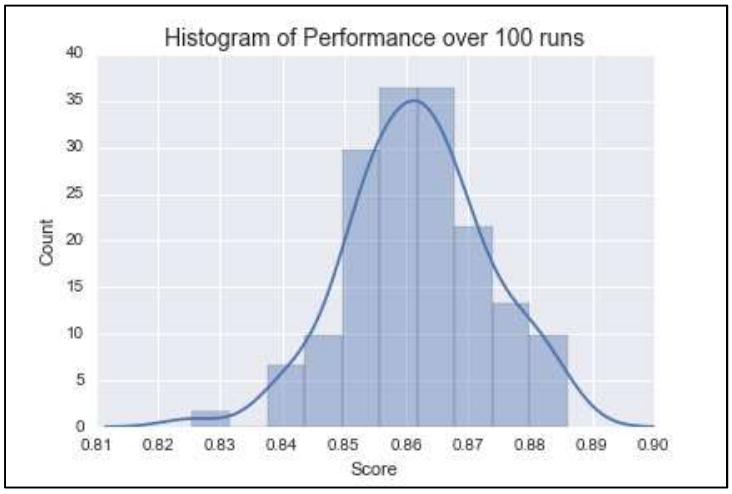
The performance improvement was investigated by tuning feature size, Random Forest maximum depth, and the minimum number of samples required to split. 15, 25, 35 and 44 (all) features, the maximum depth of 4, 6, and 8, and the minimum sample count of 2, 4, 6 and 8 were passed into the pipeline. Through grid search algorithm, the best parameter combination was found. The best score of 0.86 was obtained with 25 features, maximum depth at 6, and the minimum sample of 6. A minimum sample of 4 also showed the score of 0.86 and the performance is optimized with the minimum sample at either 4 or 6.

Results

Model Evaluation and Validation

The final model is fitted with Random Forest Classifier with 25 features, maximum depth of 6, and the minimum sample to split at 6. Validating with 10-fold shuffle split with 70% training data resulted in the average ROC AUC score of 0.86.

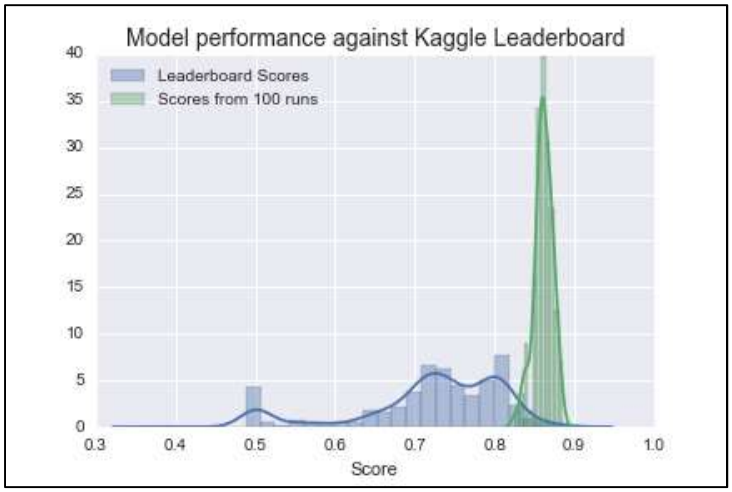
In order to validate the model robustness, the score was collected over 100 runs. Below is the histogram of 100 scores. Scores are normally distributed with the mean at ~0.86. Therefore, the model is reasonably repeatable.



Justification

The expected model performance is 0.86. This is considered good in general, as indicated in http://gim.unmc.edu/dxtests/roc3.htm.

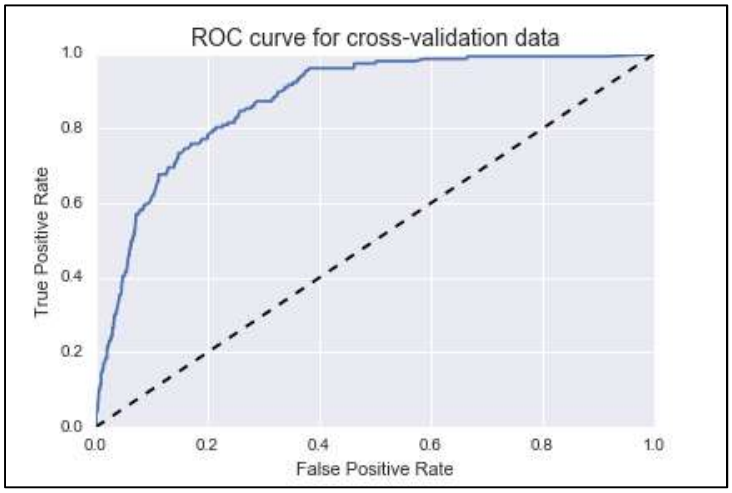
With respect to Kaggle scores from 1305 participating teams, the score is in the 75 percentile. Below is the score distribution comparison between Kaggle's leaderboard scores and 100 runs from this model. The model consistently scores on the high-end of the Kaggle scoreboard distribution.


Model performance against Kaggle Leaderboard
Leaderboard Scores
Scores from 100 runs
40
35
30
25
20
15
10
5
0
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Score

Conclusion

Free-Form Visualization

Below is the ROC curve for the model validation. This confirms the cross-validation score of 0.86. Dotted line shows the AUC score of 0.5, which is equivalent to random guess where the true positive rate equals the false positive rate. This model is much more likely to predict the true positives than the false positives.



Reflection

Mosquito and the weather information from 2007, 2009, 2011 and 2013 were used to model the WNV presence in Chicago. Based on 10,000+ data from 136 traps with <25% true positive rate, the model was generated using Random Forest Classifier. Most of the available data was preserved to model the virus presence with the performance of 0.86.

It is pleasantly surprising that the machine learning algorithm can predict WNV presence with a reasonable performance with fairly minimal data processing. The machine learning algorithm power and potential came through in this project.

Improvement

Most of the features were preserved in this model in order to avoid wrong assumption leading to a loss of information. The grid search optimization selected 25 out of 44 features. Furthermore, decision tree parameters were optimized at the maximum depth of 6 and the minimal samples of 6.

Previous learning curves for the Random Forest Classifier for various maximum depth indicated bias on one end and variance on another. In order to achieve higher model performance, improvements beyond parameter tuning need to be made.

The features can be consolidated further. Specifically, ‘Block’, ‘Street’, ‘Latitude’, ‘Longitude’ and ‘Trap’ can potentially be combined or simplified. An additional idea for feature engineering could be an application of feature reduction with such algorithm as PCA. This may help enable SVM with reasonable modeling time. SVM was removed from contention due to the time required. The model can, then, be further tuned with SVM parameters.