UWPCE DataScience3 KaggleWNVP Project Report

Kaggle Team 3

2 June 2015

Contents

0. Introduction

0.1 Background

WNVP: "West Nile Virus Prediction"

URL of Kaggle Competition.

Goal of Competition: "Predict West Nile virus in mosquitos across the city of Chicago"

Team Goals:

1. Experiment with modelling alternative on a real-world dataset.
2. Learn how to participate in a Kaggle competition.
3. Non-Goal: win the competition, or even score highly.

0.2 Team Members

- Bethene Britt
- Andrew Ewing
- Gregory Hogue
- Patrick Leahy
- Linghua Qiu
- Chris Ross
- Robert Russell
- Jim Stearns

1. Data Preparation

1.1 Obtain Original Datasets from Kaggle Website

Download the training, test, spray, and weather data from the Kaggle web site page for West Nile Virus Prediction.

Please see Appendix for R (and Python) code that:

- General Setup: Clears environment, sets working directory, loads libraries, define utility functions
- Downloads the data from the Kaggle site and unzips them.

```
stopifnot(allKaggleFilesArePresent())
print("All unzipped Kaggle datasets found in PWD. Proceeding.")
```

```
## [1] "All unzipped Kaggle datasets found in PWD. Proceeding."
```

1.2 Data Preparation

- Read in test and train csv-format files into data frames.
- Make the train and test datasets have the same attributes:
  - Train: Convert the WnvPresent column from numeric to factor with levels "Yes" and "No".
  - Test: Remove Id attribute.
  - Train: Remove NumMosquitos attribute. Potentially useful, but not available in Test dset.
  - Test: Add a WnvPresent factor column, all with "No" level.
  - Both: Remove the block attributes of little use:
    - Address, Block, Street, AddressNumberAndStreet, AddressAccuracy.
  - Both: Add bit vectors of each of the levels of the Species factor (a new column for each of the factor levels, with a zero or 1 value). Leave the Species as well.
  - Both: Convert date into date format, add "Year", "Month", and "Week" factor attributes.

```
train_df <- read.csv(paste0(dataSubDir, "/", wnvpTrainFilename))
test_df <- read.csv(paste0(dataSubDir, "/", wnvpTestFilename))
# Quick sanity check: got right number of records?
stopifnot(nrow(train_df) == wnvpTrainFileNRecs)
stopifnot(nrow(test_df) == wnvpTestFileNRecs)

# WnvPresent. Train: convert to factor. Test: add as factor, default value of "No".
train_df$WnvPresent <- factor(train_df$WnvPresent, labels=c("No", "Yes"))
WnvPresent <- factor("No", levels=c("No","Yes"))
test_df <- cbind(test_df, WnvPresent)

# Test: Remove Id attribute.
test_df$Id <- NULL
# Train: Remove NumMosquitos attribute. Potentially useful, but not available in Test dset.
train_df$NumMosquitos <- NULL

# Both: Remove the block attributes of little use:
attrsToRemove <- c("Address", "Block", "Street", "AddressNumberAndStreet", "AddressAccuracy")
train_df <- train_df[,!names(train_df) %in% attrsToRemove]
test_df <- test_df[,!names(test_df) %in% attrsToRemove]

# For creation of factor attributes, temporarily combine train and test into one dataset
# so that factor levels are the same when both are written out as separate files.
# Keeps Weka happy.
# Add a temporary column distinquishing train from test dataset entries.
train_df$DsetType <- "Train"
test_df$DsetType <- "Test"

combined_df = rbind(train_df, test_df)

# Both (in Combined): Add bit vectors for Species, one column for each factor level
# TODO: "UNSPECIFIED CULEX" needs attention.
combined_df <- with(combined_df, cbind(model.matrix( ~ 0 + Species, combined_df), combined_df))

# Both (in Combined): Convert date into date format,
# add "Year", "Month", and "Week" factor attributes.
# as.Date() tries %Y-%m-%d by default, but what the heck, explicitly state the format.
combined_df$Date <- as.Date(combined_df$Date, format="%Y-%m-%d")

combined_df$Year <- as.factor(format(combined_df$Date, "%Y"))
combined_df$Month <- as.factor(format(combined_df$Date, "%m"))
combined_df$Week <- as.factor(format(combined_df$Date, "%U"))

# Move temporary dsetType and date-related attributes to left, leaving WnvPresent last
combined_df <- moveColsToFirst(combined_df, c("DsetType", "Date", "Year", "Month", "Week"))

# Do not remove the Species attribute - not all models will use the bit vectors.
#train$Species <- NULL
#test$Species <- NULL
```

1.3 Merging Weather Data with Trap Observations

Distance of Trap from the Two Weather Stations

- Both (in Combined): Calculate the distance (using lat/long) of the trap from the two weather stations, adding attributes with the value in kilometers. Patience: This takes a while (~5 minutes).
- Both (in Combined): Add a nearest weather station attribute.

Using function *distCosine* in R Geosphere Package to calculate distance on a sphere.

```
# Station 1: O'Hare
station1LongLat <- c(-87.933, 41.995)

# Station 2: Midway
station2LongLat <- c(-87.752, 41.786)

# Patience. This takes a while (~5 minutes)
for (i in 1:nrow(combined_df)) {
    combined_df$Station1DistKm[i] <- distCosine(
        c(combined_df$Longitude[i], combined_df$Latitude[i]), station1LongLat) / 1000
    combined_df$Station2DistKm[i] <- distCosine(
        c(combined_df$Longitude[i], combined_df$Latitude[i]), station2LongLat) / 1000
}
combined_df$NearestStation <- ifelse(combined_df$Station1DistKm <= combined_df$Station2DistKm, 1,
```

- Both (in Combined): Merge in temperature and wind data from nearest station on that date.

```
weather_df <- read.csv(paste0(dataSubDir, "/", wnvpWeatherFilename), stringsAsFactors=FALSE)
colsToKeep <- c("Station", "Date", "Tmax", "Tmin", "Tavg", "AvgSpeed")
weatherData <- weather_df[,names(weather_df) %in% colsToKeep]
weatherData$Date <- as.Date(weatherData$Date, format="%Y-%m-%d")
# Tmax and Tmin come in as type int. Tavg, however, comes in as chr.
weatherData$Tavg <- as.integer(weatherData$Tavg)
```

```
## Warning: NAs introduced by coercion
```

```
# So does AvgSpeed.
weatherData$AvgSpeed <- as.numeric(weatherData$AvgSpeed)
```

```
## Warning: NAs introduced by coercion
```

```
combinedWithWeather <- merge(combined_df, weatherData,
                             by.x=c("Date", "NearestStation"), by.y=c("Date", "Station"),
                             all.x=TRUE)

combinedWithWeather <- moveColsToLast(combinedWithWeather, "WnvPresent")
stopifnot(nrow(combinedWithWeather) == (wnvpTrainFileNRecs + wnvpTestFileNRecs))

# Weather data does have some NA fields, but not the subset merged into train and test dset
```

```
# Throw an exception if that ever proves not to be the case.
stopifnot(sum(is.na(combinedWithWeather$Tmin)) == 0)
stopifnot(sum(is.na(combinedWithWeather$Tmax)) == 0)
stopifnot(sum(is.na(combinedWithWeather$Tavg)) == 0)
stopifnot(sum(is.na(combinedWithWeather$AvgSpeed)) == 0)
```

1.4 Write train and test files, complete (all attributes), to CSV

- Write the train and test datasets - including weather data - as CSV.

```
stopifnot(sum(combinedWithWeather$DsetType == "Train") == wnvpTrainFileNRecs)
stopifnot(sum(combinedWithWeather$DsetType == "Test") == wnvpTestFileNRecs)

trainWithWeather <- combinedWithWeather[combinedWithWeather$DsetType == "Train",]
stopifnot(nrow(trainWithWeather) == wnvpTrainFileNRecs)
trainWithWeather$DsetType <- NULL

write.csv(trainWithWeather,
          paste0(workingSubDir, "/", "train", "Master", ".csv"),
          eol = '\n')
str(trainWithWeather)
```

```
## 'data.frame':    10506 obs. of  24 variables:
##  $ Date                          : Date, format: "2007-05-29" "2007-05-29" ...
##  $ NearestStation                : num  1 1 1 1 1 1 1 2 2 2 ...
##  $ Year                          : Factor w/ 8 levels "2007","2008",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Month                         : Factor w/ 6 levels "05","06","07",..: 1 1 1 1 1 1 1 1 1 1 ..
##  $ Week                          : Factor w/ 20 levels "21","22","23",..: 1 1 1 1 1 1 1 1 1 1 .
##  $ SpeciesCULEX ERRATICUS        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpeciesCULEX PIPIENS          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpeciesCULEX PIPIENS/RESTUANS: num  1 0 0 1 0 1 1 0 1 0 ...
##  $ SpeciesCULEX RESTUANS         : num  0 1 1 0 1 0 0 1 0 1 ...
##  $ SpeciesCULEX SALINARIUS       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpeciesCULEX TARSALIS         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpeciesCULEX TERRITANS        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpeciesUNSPECIFIED CULEX      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Species                       : Factor w/ 8 levels "CULEX ERRATICUS",..: 3 4 4 3 4 3 3 4 3 4
##  $ Trap                          : Factor w/ 149 levels "T001","T002",..: 2 2 7 14 14 95 90 34
##  $ Latitude                      : num  42 42 42 42 42 ...
##  $ Longitude                     : num  -87.8 -87.8 -87.8 -87.8 -87.8 ...
##  $ Station1DistKm                : num  11.81 11.81 13.55 9.25 9.25 ...
##  $ Station2DistKm                : num  19.2 19.2 23.3 21.8 21.8 ...
##  $ Tmax                          : int  88 88 88 88 88 88 88 88 88 88 ...
##  $ Tmin                          : int  60 60 60 60 60 60 60 65 65 65 ...
##  $ Tavg                          : int  74 74 74 74 74 74 74 77 77 77 ...
##  $ AvgSpeed                      : num  6.5 6.5 6.5 6.5 6.5 6.5 6.5 7.4 7.4 7.4 ...
##  $ WnvPresent                    : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
```

```
testWithWeather <- combinedWithWeather[combinedWithWeather$DsetType == "Test",]
stopifnot(nrow(testWithWeather) == wnvpTestFileNRecs)
trainWithWeather$DsetType <- NULL

write.csv(testWithWeather,
          paste0(workingSubDir, "/", "test", "Master", ".csv"),
          eol = '\n')
str(testWithWeather)
```

```
## 'data.frame':    116293 obs. of  25 variables:
##  $ Date                          : Date, format: "2008-06-11" "2008-06-11" ...
##  $ NearestStation                : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ DsetType                      : chr  "Test" "Test" "Test" "Test" ...
##  $ Year                          : Factor w/ 8 levels "2007","2008",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Month                         : Factor w/ 6 levels "05","06","07",..: 2 2 2 2 2 2 2 2 2 2 ..
##  $ Week                          : Factor w/ 20 levels "21","22","23",..: 3 3 3 3 3 3 3 3 3 3 .
##  $ SpeciesCULEX ERRATICUS        : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ SpeciesCULEX PIPIENS          : num  0 0 1 0 0 0 0 0 0 0 ...
##  $ SpeciesCULEX PIPIENS/RESTUANS: num  1 0 0 0 0 0 0 0 1 0 ...
##  $ SpeciesCULEX RESTUANS         : num  0 1 0 0 0 0 0 0 0 1 ...
##  $ SpeciesCULEX SALINARIUS       : num  0 0 0 1 0 0 0 0 0 0 ...
##  $ SpeciesCULEX TARSALIS         : num  0 0 0 0 0 1 0 0 0 0 ...
##  $ SpeciesCULEX TERRITANS        : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ SpeciesUNSPECIFIED CULEX      : num  0 0 0 0 0 0 1 0 0 0 ...
##  $ Species                       : Factor w/ 8 levels "CULEX ERRATICUS",..: 3 4 2 5 7 6 8 1 3 4
##  $ Trap                          : Factor w/ 149 levels "T001","T002",..: 2 2 2 2 2 2 2 2 7 7 .
##  $ Latitude                      : num  42 42 42 42 42 ...
##  $ Longitude                     : num  -87.8 -87.8 -87.8 -87.8 -87.8 ...
##  $ Station1DistKm                : num  11.8 11.8 11.8 11.8 11.8 ...
##  $ Station2DistKm                : num  19.2 19.2 19.2 19.2 19.2 ...
##  $ Tmax                          : int  86 86 86 86 86 86 86 86 86 86 ...
##  $ Tmin                          : int  61 61 61 61 61 61 61 61 61 61 ...
##  $ Tavg                          : int  74 74 74 74 74 74 74 74 74 74 ...
##  $ AvgSpeed                      : num  10 10 10 10 10 10 10 10 10 10 ...
##  $ WnvPresent                    : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
```

1.5 Write train and test files, complete (all attributes), to ARFF

```
write.arff(trainWithWeather,
          paste0(workingSubDir, "/", "train", "Master", ".arff"),
          eol = '\n', relation="WNVPTrainDataset")
str(trainWithWeather)
```

```
## 'data.frame':    10506 obs. of  24 variables:
##  $ Date                          : Date, format: "2007-05-29" "2007-05-29" ...
##  $ NearestStation                : num  1 1 1 1 1 1 1 2 2 2 ...
##  $ Year                          : Factor w/ 8 levels "2007","2008",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Month                         : Factor w/ 6 levels "05","06","07",..: 1 1 1 1 1 1 1 1 1 1 ..
##  $ Week                          : Factor w/ 20 levels "21","22","23",..: 1 1 1 1 1 1 1 1 1 1 .
```

```
##  $ SpeciesCULEX ERRATICUS       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpeciesCULEX PIPIENS         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpeciesCULEX PIPIENS/RESTUANS: num  1 0 0 1 0 1 1 0 1 0 ...
##  $ SpeciesCULEX RESTUANS        : num  0 1 1 0 1 0 0 1 0 1 ...
##  $ SpeciesCULEX SALINARIUS      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpeciesCULEX TARSALIS        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpeciesCULEX TERRITANS       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpeciesUNSPECIFIED CULEX     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Species                      : Factor w/ 8 levels "CULEX ERRATICUS",..: 3 4 4 3 4 3 3 4 3 4
##  $ Trap                         : Factor w/ 149 levels "T001","T002",..: 2 2 7 14 14 95 90 34
##  $ Latitude                     : num  42 42 42 42 42 ...
##  $ Longitude                    : num  -87.8 -87.8 -87.8 -87.8 -87.8 ...
##  $ Station1DistKm               : num  11.81 11.81 13.55 9.25 9.25 ...
##  $ Station2DistKm               : num  19.2 19.2 23.3 21.8 21.8 ...
##  $ Tmax                         : int  88 88 88 88 88 88 88 88 88 88 ...
##  $ Tmin                         : int  60 60 60 60 60 60 60 65 65 65 ...
##  $ Tavg                         : int  74 74 74 74 74 74 74 77 77 77 ...
##  $ AvgSpeed                     : num  6.5 6.5 6.5 6.5 6.5 6.5 6.5 7.4 7.4 7.4 ...
##  $ WnvPresent                   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
```

```
write.arff(testWithWeather,
           paste0(workingSubDir, "/", "test", "Master", ".arff"),
           eol = '\n', relation="WNVPTestDataset")
str(testWithWeather)
```

```
## 'data.frame':    116293 obs. of  25 variables:
##  $ Date                         : Date, format: "2008-06-11" "2008-06-11" ...
##  $ NearestStation               : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ DsetType                     : chr  "Test" "Test" "Test" "Test" ...
##  $ Year                         : Factor w/ 8 levels "2007","2008",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Month                        : Factor w/ 6 levels "05","06","07",..: 2 2 2 2 2 2 2 2 2 2 ..
##  $ Week                         : Factor w/ 20 levels "21","22","23",..: 3 3 3 3 3 3 3 3 3 3 .
##  $ SpeciesCULEX ERRATICUS       : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ SpeciesCULEX PIPIENS         : num  0 0 1 0 0 0 0 0 0 0 ...
##  $ SpeciesCULEX PIPIENS/RESTUANS: num  1 0 0 0 0 0 0 0 1 0 ...
##  $ SpeciesCULEX RESTUANS        : num  0 1 0 0 0 0 0 0 0 1 ...
##  $ SpeciesCULEX SALINARIUS      : num  0 0 0 1 0 0 0 0 0 0 ...
##  $ SpeciesCULEX TARSALIS        : num  0 0 0 0 0 1 0 0 0 0 ...
##  $ SpeciesCULEX TERRITANS       : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ SpeciesUNSPECIFIED CULEX     : num  0 0 0 0 0 0 1 0 0 0 ...
##  $ Species                      : Factor w/ 8 levels "CULEX ERRATICUS",..: 3 4 2 5 7 6 8 1 3 4
##  $ Trap                         : Factor w/ 149 levels "T001","T002",..: 2 2 2 2 2 2 2 2 7 7 .
##  $ Latitude                     : num  42 42 42 42 42 ...
##  $ Longitude                    : num  -87.8 -87.8 -87.8 -87.8 -87.8 ...
##  $ Station1DistKm               : num  11.8 11.8 11.8 11.8 11.8 ...
##  $ Station2DistKm               : num  19.2 19.2 19.2 19.2 19.2 ...
##  $ Tmax                         : int  86 86 86 86 86 86 86 86 86 86 ...
##  $ Tmin                         : int  61 61 61 61 61 61 61 61 61 61 ...
##  $ Tavg                         : int  74 74 74 74 74 74 74 74 74 74 ...
##  $ AvgSpeed                     : num  10 10 10 10 10 10 10 10 10 10 ...
##  $ WnvPresent                   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
```

2. Data Exploration and Analysis

TBD

3. Features for Modeling

TBD

4. Kaggle Submissions

Insert table of submissions here

4.1 Reproduction of May 17 Pat Leahy Submittal (Score 0.59642)

Perform any subsetting here so that train and test formats look the same to Weka.

```
colsToKeep=c("Latitude", "Longitude", "Month", "Tmin", "Tmax", "Tavg", "WnvPresent")
combinedForModeling <- combinedWithWeather[,names(combinedWithWeather) %in% colsToKeep]
stopifnot(nrow(combinedForModeling) == (wnvpTrainFileNRecs + wnvpTestFileNRecs))
```

Undersample: use all the WnvPresent==True samples. Randomly select an equal number of False samples. Use that for the test data set.

```
curModelIdx <- "01"
trainRecs <- combinedForModeling[1:wnvpTrainFileNRecs,]
oversample_df <- trainRecs[trainRecs$WnvPresent=="Yes",]
nFalseObservationsToUse <- nrow(oversample_df)
wnvNotPresent <- trainRecs[trainRecs$WnvPresent=="No",]
oversample_df <- rbind(oversample_df,
                       wnvNotPresent[sample(nrow(wnvNotPresent), nFalseObservationsToUse),])

write.arff(oversample_df,
           paste0(workingSubDir, "/", "train", curModelIdx, ".arff"),
           eol = '\n', relation="WNVPTrainDataset")
str(oversample_df)
```

```
## 'data.frame':    472 obs. of  7 variables:
##  $ Month     : Factor w/ 6 levels "05","06","07",..: 3 3 3 3 3 3 4 4 4 4 ...
##  $ Latitude  : num  41.7 41.7 41.7 41.7 41.7 ...
##  $ Longitude : num  -87.5 -87.6 -87.6 -87.6 -87.6 ...
##  $ Tmax      : int  85 83 83 83 83 83 92 92 92 92 ...
##  $ Tmin      : int  69 70 70 70 70 70 69 69 69 69 ...
##  $ Tavg      : int  77 77 77 77 77 77 81 81 81 81 ...
##  $ WnvPresent: Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
```

```
testRecs <- combinedForModeling[(wnvpTrainFileNRecs+1):(wnvpTrainFileNRecs+wnvpTestFileNRecs),]
stopifnot(nrow(testRecs) == wnvpTestFileNRecs)
write.arff(testRecs,
           paste0(workingSubDir, "/", "test", curModelIdx, ".arff"),
           eol = '\n', relation="WNVPTestDataset")
str(testRecs)
```

```
## 'data.frame':    116293 obs. of  7 variables:
##  $ Month     : Factor w/ 6 levels "05","06","07",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ Latitude  : num  41.9 41.9 41.9 41.8 41.8 ...
##  $ Longitude : num  -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ Tmax      : int  89 89 89 89 89 89 89 89 89 89 ...
##  $ Tmin      : int  64 64 64 64 64 64 64 64 64 64 ...
##  $ Tavg      : int  77 77 77 77 77 77 77 77 77 77 ...
##  $ WnvPresent: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
```

May 28: Additional Predictors (Species bit vectors, NumMosqitos)

Note: Dataset rolls over to a new record if number of mosquitos reaches 50. TODO: Same date, same lat/long, same Species: combine the records.

```
XXXXXXXXX
colsToKeep=c("Latitude", "Longitude", "Month", "Tmin", "Tmax", "Tavg", "WnvPresent")
colsToKeep=c(colsToKeep, "SpeciesCULEX ERRATICUS", "SpeciesCULEX PIPIENS",
             "SpeciesCULEX PIPIENS/RESTUANS", "SpeciesCULEX RESTUANS", "SpeciesCULEX SALINARIUS",
             "SpeciesCULEX TARSALIS", "SpeciesCULEX TERRITANS", "SpeciesUNSPECIFIED CULEX")
colsToKeep=c(colsToKeep, "NumMosquitos")
combinedForModeling <- combinedMaster[,names(combinedMaster) %in% colsToKeep]
stopifnot(nrow(combinedForModeling) == (wnvpTrainFileNRecs + wnvpTestFileNRecs))
```

Undersample: use all the WnvPresent==True samples. Randomly select an equal number of False samples. Use that for the test data set.

```
XXXXXXXXX
curModelIdx <- "03"
trainRecs <- combinedForModeling[1:nTrainingRecs,]
oversample_df <- trainRecs[trainRecs$WnvPresent=="Yes",]
nFalseObservationsToUse <- nrow(oversample_df)
wnvNotPresent <- trainRecs[trainRecs$WnvPresent=="No",]
oversample_df <- rbind(oversample_df, wnvNotPresent[sample(nrow(wnvNotPresent), nFalseObservation

write.arff(oversample_df,
           paste0(workingSubDir, "/", "train", curModelIdx, ".arff"),
           eol = '\n', relation="WNVPTrainDataset")
str(oversample_df)

testRecs <- combinedForModeling[(nTrainingRecs+1):(wnvpTrainFileNRecs+wnvpTestFileNRecs),]
stopifnot(nrow(testRecs) == wnvpTestFileNRecs)
write.arff(testRecs,
           paste0(workingSubDir, "/", "test", curModelIdx, ".arff"),
           eol = '\n', relation="WNVPTestDataset")
str(testRecs)
```

NumMosquitos degraded score. Left Species bit vector.

Jun-2 #3: Additional Predictor (Week). Over-sample WnvPresent

Change: Add "Week". Use SMOTE in R to oversample WnvPresent.

```
XXXXXXXXX
curModelIdx <- "05"
colsToKeep=c("Latitude", "Longitude", "Month", "Tmin", "Tmax", "Tavg", "WnvPresent")
colsToKeep=c(colsToKeep, "SpeciesCULEX ERRATICUS", "SpeciesCULEX PIPIENS",
             "SpeciesCULEX PIPIENS/RESTUANS", "SpeciesCULEX RESTUANS", "SpeciesCULEX SALINARIUS",
             "SpeciesCULEX TARSALIS", "SpeciesCULEX TERRITANS", "SpeciesUNSPECIFIED CULEX")
combinedForModeling <- combinedMaster[,names(combinedMaster) %in% colsToKeep]
stopifnot(nrow(combinedForModeling) == (wnvpTrainFileNRecs + wnvpTestFileNRecs))

nTrainingRecs = nrow(train_df)
nTotalRecs = wnvpTrainFileNRecs + wnvpTestFileNRecs
stopifnot(nrow(combinedMaster) == nTotalRecs)

trainRecs <- combinedMaster[1:nTrainingRecs,]
stopifnot(nTrainingRecs == wnvpTrainFileNRecs)
write.csv(trainRecs,
          paste0(workingSubDir, "/", "train", curModelIdx, ".csv"),
          eol = '\n')
str(trainRecs)

testRecs <- combinedMaster[(nTrainingRecs+1):nTotalRecs,]
stopifnot(nrow(testRecs) == wnvpTestFileNRecs)
write.csv(testRecs,
          paste0(workingSubDir, "/", "test", curModelIdx, ".csv"),
          eol = '\n')
str(testRecs)

# Write the combined master - train and test - as a CSV file.
# Good starting point for future experiments.
write.csv(combinedMaster,
          paste0(workingSubDir, "/", "trainAndTest", curModelIdx, ".csv"),
          eol = '\n')
str(combinedMaster)

trainRecs <- combinedForModeling[1:nTrainingRecs,]

write.arff(trainRecs,
          paste0(workingSubDir, "/", "train", curModelIdx, ".arff"),
          eol = '\n', relation="WNVPTrainDataset")
str(trainRecs)

testRecs <- combinedForModeling[(nTrainingRecs+1):(wnvpTrainFileNRecs+wnvpTestFileNRecs),]
stopifnot(nrow(testRecs) == wnvpTestFileNRecs)
write.arff(testRecs,
          paste0(workingSubDir, "/", "test", curModelIdx, ".arff"),
          eol = '\n', relation="WNVPTestDataset")
```

```
str(testRecs)
```

June 2: Generalized Additive Model (GAM), adding Average Temperature and Speed (Andy Ewing)

Submitted by Andy Ewing. Used a sample from the Kaggle WNVP forum: baby steps: breach 0.71 with GAM

Added weather data: daily average temperature and wind speed.

```
source("src/CurDate.R")
```

```
## [1] "/Users/jimstearns/GoogleDrive/Learning/Courses/UWPCE-DataScience/Course3_DataAtScale/Kagg
## [1] "Wed Jun  3 22:58:05 2015"
## [1] "This is a sourced file"
```

5. Modeling Strategy

TBD. Unclear: discuss modeling strategy here, or in discussion of submittals?

6. Ensemble Model Opportunities

Appendix

A.1 General Setup: Clear environment, set working directory, load libraries, utilities

```
# Clear the working environment of variables, data, functions
rm(list=ls())

# Set working directory for this Kaggle project. Default: pwd.
#kaggleProjHomeDir <- "."
kaggleProjHomeDir <- "/Users/jimstearns/GoogleDrive/Learning/Courses/UWPCE-DataScience/Course3_Da
setwd(kaggleProjHomeDir)
getwd()

#install.packages("rPython")  # For download from web site with login/pwd.
library(rPython) # For calling python function to download file w/login+pwd
# Package for writing Weka ARFF file format
stopifnot(require("foreign"))
library("foreign")
# Package for calculating great circle distances
stopifnot(require("geosphere"))
library("geosphere")

# Return a data frame with the named column(s) moved to last position.
# Here, it will be the output classification, WnvPresent.
moveColsToLast <- function(df, colsToMove) {
```

```
    df[c(setdiff(names(df), colsToMove), colsToMove)]
}

moveColsToFirst <- function(df, colsToMove) {
    df[c(colsToMove, setdiff(names(df), colsToMove))]
}
```

Dataset download and unpacking

This R and Python code downloads the WNVP datasets from Kaggle. Some setup is required:

- One's Kaggle username and password must be defined as environment variables where R is running.
- Easiest way to set environment variable for R: Create (add to) ~/.Renviron file (kaggleUsername="XXXX" and kagglePassword="YYYY").

Alternatively, files can be downloaded manually.

```
wnvpTrainFilename <- "train.csv"
wnvpTestFilename <- "test.csv"
wnvpWeatherFilename <- "weather.csv"
wnvpSprayFilename <- "spray.csv"
kaggleDatasets = c(
    wnvpTrainFilename,
    wnvpTestFilename,
    wnvpWeatherFilename,
    wnvpSprayFilename)
dataSubDir <- "input"  # Kaggle convention
workingSubDir <- "working" # Kaggle convention: massaged datasets - and output - go here.

wnvpTrainFileNRecs <- 10506 # Observation records in training file. Excludes header record.
wnvpTestFileNRecs <- 116293 # Records in test file supplied by Kaggle. Submission record cnt mu
# If download from Kaggle required, and user and pwd are empty (default),
# then user will be prompted for these two values.
kaggleUsername <- ""
kagglePassword <- ""

allKaggleFilesArePresent <- function() {
    filesAllFound <- TRUE
    for (file in kaggleDatasets) {
        if (!file.exists(paste0(dataSubDir, "/", file))) {
            print(paste("Error: could not find unzipped Kaggle file in PWD:", file))
            filesAllFound <- FALSE
        }
    }
    return(filesAllFound)
}

downloadMissingKaggleFiles <- function() {
    python.load("src/UrlFileDownloaderWithLogin.py")
```

```
    kaggleUsername = Sys.getenv("kaggleUsername")
    kagglePassword = Sys.getenv("kagglePassword")
    if (kaggleUsername == "" || kagglePassword == "") {
        print("Please assign kaggleUsername and kagglePassword environment variables.")
        print("Place in ~/.Renviron entries such as kaggleUsername='YourName'.")
    }
    stopifnot(!(kaggleUsername == ""))
    stopifnot(!(kagglePassword == ""))

    wnvpKaggleDataUrl <-
        "https://www.kaggle.com/c/predict-west-nile-virus/download/"

    for (file in kaggleDatasets) {
        if (file.exists(file))
            next

        urlOfZip <- paste0(wnvpKaggleDataUrl, file, ".zip")
        print(urlOfZip)
        # Use a python method to download from URL with login and password.
        # Download to subdirectory "input" and filename w/o the .zip suffix.
        python.call("Download", urlOfZip,
                    kaggleUsername, kagglePassword ,
                    paste0(dataSubDir, "/", file, ".zip"))
    }
}

unzipDownloadedFiles <- function() {
    for (file in kaggleDatasets) {
        zippedFile <- paste0(dataSubDir, "/", file, ".zip")
        print(paste0("Unzip: ", zippedFile))
        if (file.exists(zippedFile)) {
            if (file.exists(file)) {
                print(sprintf("Warning: removing existing file %s\n", file))
                file.remove(file)
            }
            unzip(zippedFile, exdir=dataSubDir)
            print(sprintf("Unzipped: %s\n", zippedFile))
        }
    }
}

if (!allKaggleFilesArePresent()) {
    print(paste("Not all needed Kaggle datasets are present in PWD;",
                "attempting to download from Kaggle web site."))
    downloadMissingKaggleFiles()
    unzipDownloadedFiles()
}
```

File UrlFileDownloaderWithLogin.py:

```
__author__ = 'jimstearns'
```

```
""" Download a file at a URL at a web site that requires a user name and password.
"""

import logging
import os       # File utilities

# Python package "requests": "Python HTTP for Humans" by Kenneth Reitz. Current version: 2.7.0.
# Documented at http://docs.python-requests.org/en/latest/
# To install from the command line: "pip install requests"
# (On Mac, sudo may be required. Also pip2.7 instead of pip, depending on default Python version)

import requests # Http GET, POST

def Download(url, username, password, local_filename):
    # Login to web site such as Kaggle and retrieve the data. Use POST rather than GET as as to
    # send login info in body of HTTP request rather than in query string portion of URL.

    # Limitation: when used by Python version < 2.7.9, an "InsecureRequestWarning" is generated.
    # TODO: Fix. Details: https://urllib3.readthedocs.org/en/latest/security.html#insecureplatfor
    # Workaround: log warnings to file, not stdout.
    logging.captureWarnings(True)

    if (os.path.exists(local_filename)):
        os.remove(local_filename)

    # This won't get the file, but use the return value URL in a follow-on POST:
    r = requests.get(url)

    login_info = {'UserName': '{0}'.format(username), 'Password': '{0}'.format(password) }
    print(login_info)
    r = requests.post(r.url, data = login_info)
    print("POST (w/login info): {0}\n".format(r.status_code))

    # Write the data to a local file one chunk at a time.
    chunk_size = 512 * 1024 # Reads 512KB at a time into memory
    with open(local_filename, 'wb') as fd:
        for chunk in r.iter_content(chunk_size): # Reads 512KB at a time into memory
            if chunk: # filter out keep-alive new chunks
                fd.write(chunk)

    if (os.path.exists(local_filename)):
        return(True)
    else:
        return(False)
```

Miscellaneous

Prepare Weka results as ARFF file as submittal file to Kaggle as CSV.

File PrepareWekaArffResultsForKaggleCsvSubmittal:

```
# Script to read in ARFF file created by Weka modeler,
```

```
# strip all attributes except the predicted classification (here, "WnvPresent"),
# add an Id column with a sequence number equal to the record number; and
# write as a CSV file.

wnvpTestFileNRecs <- 116293 # Records in test file supplied by Kaggle. Submission record cnt must
arffInFilename <- "working/testClassified02"
testClassified_df <- read.arff(paste0(arffInFilename, ".arff"))
stopifnot(nrow(testClassified_df) == wnvpTestFileNRecs)

Id <- seq(1:wnvpTestFileNRecs)
colsToKeep <- c("predicted WnvPresent")
testClassified_df <- cbind(Id, testClassified_df[names(testClassified_df) %in% colsToKeep])
names(testClassified_df) <- c("Id", "WnvPresent")
# Write "No" as 0 and "Yes" as 1
testClassified_df$WnvPresent <- ifelse(testClassified_df$WnvPresent == "No", 0, 1)
str(testClassified_df)

write.csv(testClassified_df, paste0(arffInFilename, ".csv"), row.names=FALSE)
```

NOT USED

Distance function by "Curlew"

```
# From https://conservationecology.wordpress.com/2013/06/30/distance-between-two-points-in-r/
# Calculate distance in kilometers between two points
earth.dist <- function (long1, lat1, long2, lat2)
{
    rad <- pi/180
    a1 <- lat1 * rad
    a2 <- long1 * rad
    b1 <- lat2 * rad
    b2 <- long2 * rad
    dlon <- b2 - a2
    dlat <- b1 - a1
    a <- (sin(dlat/2))^2 + cos(a1) * cos(b1) * (sin(dlon/2))^2
    c <- 2 * atan2(sqrt(a), sqrt(1 - a))
    R <- 6378.145
    d <- R * c
    return(d)
}
```