Appendix R Code

```
##
# Yu Pei
# May 2015
# Kaggle Nile virus competition
# https://www.kaggle.com/c/predict-west-nile-virus
# Biostatistic MS comprehensive exam
##

setwd('~/Documents/Kaggle/nile_virus/scripts/')
options(scipen = 999) # no scientific number
library(randomForest)
library(geosphere)
library(lubridate)
library(AUC)
library(ggmap) # Map viz
library(FNN) # For knn.reg imputation
library(plyr)
library(gbm)
library(zoo)
library(ggplot2)
library(xtable)

## Load in data and generate features
station1 = c(41.995, -87.933)
station2 = c(41.786, -87.752)

#spray = read.csv('../input/spray.csv') not used right now
weather = read.csv('../input//weather.csv')
dat = read.csv('../input/train.csv')
dat = dat[,c(1, 3, 8, 9, 11,12)] # ignore address etc.
dat$Date = ymd(dat$Date)
dat$WnvPresent = as.factor(dat$WnvPresent)
weather$Date = ymd(weather$Date)
dat$Station =
  ifelse(distHaversine(dat[,c('Latitude', 'Longitude')], station1) <
         distHaversine(dat[,c('Latitude', 'Longitude')], station2),
          1, 2)

### Processing data

temp = merge(dat, weather)
temp$month = month(temp$Date)
temp$week = week(temp$Date)
```

```
temp$year = year(temp$Date) ## Seems like recent year has higher Wnv rate
temp$PrecipTotal = as.numeric(as.character(temp$PrecipTotal))
temp$PrecipTotal[is.na(temp$PrecipTotal)] = 0
temp$WetBulb = as.numeric(as.character(temp$WetBulb))
temp$WetBulb[is.na(temp$WetBulb)] = 0
temp$StnPressure = as.numeric(as.character(temp$StnPressure))
temp$StnPressure[is.na(temp$StnPressure)] = 0
temp$AvgSpeed = as.numeric(as.character(temp$AvgSpeed))
temp$AvgSpeed[is.na(temp$AvgSpeed)] = 0
train = temp[, c('week', 'month','Species' ,'Latitude',
                 'Longitude', 'Tmax', 'Tmin', 'DewPoint',
                 'PrecipTotal' ,'StnPressure','AvgSpeed',
                 'ResultDir', 'WetBulb', "NumMosquitos")]
y = as.factor(temp$WnvPresent)
train.date = temp$Date
## Spraying correction
# train$Date = temp$Date
# train$Spray = FALSE
# train$postSpray = FALSE
# for(i in 1:nrow(spray)){
#   if(i %% 1000 == 0){print(i)}
#   train$postSpray[train$Date - spray$Date[i] > 0 & train$Date -
#                                       spray$Date[i] < 8
&
#                 distHaversine(train[ ,c('Latitude', 'Longitude')],
#                       spray[i,c('Latitude', 'Longitude') ]) < 700] = TRUE
# }
# spray = train$Spray
# postspray = train$postSpray
###



glmcv = function(train, y, yr = 2007, td = train.date){

  holdout = train[year(td) ==yr, ]
  holdout.y = y[year(td) ==yr]
  train.cv = train[year(td) != yr,]
  train.y = y[year(td) != yr]
  w = ifelse(train.y == 1, 15, 1)
  dd = data.frame(model.matrix(train.y ~., train.cv))
  dd$y = train.y
  fitted = glm(y ~ ., data = dd,family = binomial(), weights = w)

  impute.train = train.cv[, c( 'week', 'Longitude', 'Latitude','Species')]
  impute.train$Species = as.numeric(impute.train$Species)
```

```
  impute.test = holdout[, c('week', 'Longitude', 'Latitude', 'Species')]
  impute.test$Species = as.numeric(impute.test$Species)
  impute.y = train.cv$NumMosquitos
  ## Tune K
  knnmodel = knn.reg(train = impute.train, test = impute.test, y = impute.y,
                     k = 7)
  holdout$NumMosquitos = knnmodel$pred
  holdout = data.frame(model.matrix(~., holdout))
  ypred = predict(fitted, newdata = holdout, type = 'response')
  auc(roc(ypred, holdout.y))
}


### CV
rfcv = function(train, y, yr = 2007, nt = 1500, knn.k = 7, td = train.date){
  #Split data
  holdout = train[year(td) ==yr, ]
  holdout.y = y[year(td) ==yr]
  train.cv = train[year(td) != yr,]
  train.y = y[year(td) != yr]

  # Simulate NumMosquitos imputation
  n = nrow(train.cv)
  counts = n/table(train.y)
  impute.train = train.cv[, c( 'week', 'Longitude', 'Latitude','Species')]
  impute.train$Species = as.numeric(impute.train$Species)
  impute.test = holdout[, c('week', 'Longitude', 'Latitude', 'Species')]
  impute.test$Species = as.numeric(impute.test$Species)
  impute.y = train.cv$NumMosquitos
  ## Tune K
  knnmodel = knn.reg(train = impute.train, test = impute.test, y = impute.y,
                     k = knn.k)
  holdout$NumMosquitos = knnmodel$pred

  set.seed(999)
  fitted = randomForest(train.cv, train.y, ntree = nt,
                        classwt = counts, importance = TRUE)
  cv.pred = predict(fitted, newdata = holdout,type = 'prob')[, 2]
  auc(roc(cv.pred, holdout.y))
}

## Tune GBM
gbmcv = function(train, y, yr = 2007, nt = 1700,
                 knn.k = 7, minnode = 3, dp = 3, td = train.date)
{
  holdout = train[year(td) ==yr, ]
```

```
  holdout.y = y[year(td) ==yr]
  train.cv = train[year(td) != yr,]
  train.y = y[year(td) != yr]

  impute.train = train.cv[, c( 'week', 'Longitude', 'Latitude','Species')]
  impute.train$Species = as.numeric(impute.train$Species)
  impute.test = holdout[, c('week', 'Longitude', 'Latitude', 'Species')]
  impute.test$Species = as.numeric(impute.test$Species)
  impute.y = train.cv$NumMosquitos
  ## Tune K
  knnmodel = knn.reg(train = impute.train, test = impute.test, y = impute.y,
                     k = knn.k)
  holdout$NumMosquitos = knnmodel$pred

  dd = cbind(train.cv, train.y)
  dd$train.y = as.character(train.y)
  set.seed(1000)
  gbmfit = gbm(train.y ~ ., data = dd, distribution = 'adaboost',
               interaction.depth = dp, n.minobsinnode = minnode,
               shrinkage = 0.001,
               n.trees = nt, train.fraction = 1)
  gbmpred = predict(gbmfit, newdata = holdout, n.trees = nt, 'response')
  auc(roc(gbmpred, holdout.y))

}



# Processing Test data
test = read.csv('../input/test.csv')
id = test[,1]
test = test[,c(1, 2, 4, 9, 10)]
test$Date = ymd(test$Date)
#weather$Date = ymd(weather$Date)
test$Station =
  ifelse(distHaversine(test[,c('Latitude', 'Longitude')], station1) <
           distHaversine(test[,c('Latitude', 'Longitude')], station2) ,
         1, 2)
temp2 = merge(test, weather)
temp2$month = month(temp2$Date)
temp2$week = week(temp2$Date)
temp2$year = year(temp2$Date)
temp2$PrecipTotal = as.numeric(as.character(temp2$PrecipTotal))
temp2$PrecipTotal[is.na(temp2$PrecipTotal)] = 0
temp2$WetBulb = as.numeric(as.character(temp2$WetBulb))
```

```
temp2$WetBulb[is.na(temp2$WetBulb)] = 0
temp2$StnPressure = as.numeric(as.character(temp2$StnPressure))
temp2$StnPressure[is.na(temp2$StnPressure)] = 0
temp2$AvgSpeed = as.numeric(as.character(temp2$AvgSpeed))
temp2$AvgSpeed[is.na(temp2$AvgSpeed)] = 0

temp2 = temp2[order(temp2$Id),]
test = temp2[, c('week', 'month','Species' ,'Latitude',
                 'Longitude', 'Tmax', 'Tmin', 'DewPoint',
                 'PrecipTotal' ,'StnPressure','AvgSpeed',
                 'ResultDir', 'WetBulb')]
test$Species = as.character(test$Species)
test[test$Species == 'UNSPECIFIED CULEX', 'Species'] = "CULEX ERRATICUS"
test$Species = as.factor(test$Species)


###
#Attempt to use KNN to impute NumMsquitos,
#CV validate that it is a strong predictor.
impute.train = train[, c( 'week', 'Longitude', 'Latitude', 'Species')]
impute.train$Species = as.numeric(impute.train$Species)
impute.test = test[, c('week', 'Longitude', 'Latitude', 'Species')]
impute.test$Species = as.numeric(impute.test$Species)
impute.y = train$NumMosquitos
knnmodel = knn.reg(train = impute.train, test = impute.test, y = impute.y,
                   k = 7)
###

test$NumMosquitos = knnmodel$pred
ypred = predict(fitted, newdata = test, type = 'prob')

w = ifelse(y == 1, 15, 1)
dd = data.frame(model.matrix(y ~., train))
dd$y = y
fitted = glm(y ~ ., data = dd,family = binomial(), weights = w)

glmtest = data.frame(model.matrix(~ ., test))
ypred = predict(fitted, newdata = glmtest, type = 'response')

## RF
#Calculate weights
n = nrow(train)
counts = n/table(y)
set.seed(1000)
fitted = randomForest(train, y, ntree = 1500,
                      classwt = counts, importance = TRUE)
```

```
## Fit GBM
dd = cbind(train, y)
dd$y = as.character(y)
set.seed(1000)
gbmfit = gbm(y ~ ., data = dd, distribution = 'adaboost',
             interaction.depth = 4, n.minobsinnode = 1, shrinkage = 0.001,
             n.trees = 1700, train.fraction = 1)
gbmpred = predict(gbmfit, newdata = test, n.trees = 1700, 'response')

### Write out solution
submission = data.frame('Id' = id, 'WnvPresent' = ypred)

submission = data.frame('Id' = id, 'WnvPresent' = ypred[, 2])
submission[test$Species == 'CULEX ERRATICUS', 2] = 0
submission[test$Species == 'CULEX SALINARIUS', 2] = 0
submission[test$Species == 'CULEX TARSALIS', 2] = 0
submission[test$Species == 'CULEX TERRITANS', 2] = 0
write.csv(submission, 'glm_for_report.csv', row.names = F, quote = F)




###### Spatial exploration
library(sp)
library(raster)
oneday = test[test$Date == ymd('2014-10-02'),]
oneday = oneday[, -1]
oneday$NumMosquitos = 1
oneday$WnvPresent = -1


aa = dat[dat$Date == ymd('2013-09-26 UTC'),]
aa$Station = NULL

bb = rbind(oneday, aa)

mapdata = readRDS('../mapdata_copyright_openstreetmap_contributors.rds')
measurement_sites_plot <- ggmap(mapdata) +
  geom_point(aes(x=Longitude, y=Latitude, size = NumMosquitos,
                 color = WnvPresent), data=bb)


dd = as.character(unique(dat$Date))
for(d in dd){
  aa = dat[dat$Date == ymd(d),]
```

```
  aa$Station = NULL
  bb = rbind(oneday, aa)
  bb$WnvPresent = as.factor(bb$WnvPresent)
  siteplot = ggmap(mapdata) + geom_point(aes(x=Longitude, y=Latitude,
                                  size = NumMosquitos, color = WnvPresent),
                                  data=bb) +
          ggtitle(as.character(d))
  print(siteplot)
  trash = readline()
  if(trash == 'q')break
}
##########


#### Explore Species relation to Wnv
table(train$Species[y == 1])/sum(y == 1)
table(train$Species[y == 0])/sum(y == 0)
## Looks like there are some proportion difference wrt CULEX PIPIENS

#### Explore weather relation to Wnv


###### Explore possibility of grouping Lat Lon into regions(less priority)


##### Based on test data pattern to recover the NumMosquitos(spatial kriging?)


##### Explore the time-series dependency To add new feature
yy = split(train, train$Latitude)
test$id = 1:nrow(test)
yy = split(test, test$Latitude)
myfun = function(i){
  print(i)
  j = 7
  if(nrow(yy[[i]]) <5){
    j = 3
  }
  jj = 1:j
  f7 = rep(1/j, j)
  tt = yy[[i]][, c('Tmax', 'Tmin', 'StnPressure', 'AvgSpeed')]
  rmean = rollapply(tt, width = j, mean, by.column = T, partial = TRUE)
  colnames(rmean) = paste0(colnames(rmean), '.mean')
  rsd = rollapply(tt, width = j, sd, by.column = T, partial = TRUE)
  colnames(rsd) = paste0(colnames(rsd), '.sd')
  rslope = rollapply(tt, width = j, FUN = function(z)coef(lm(z ~ jj))[2],
```

```
                    fill = 1, by.column = T)
  colnames(rslope) = paste0(colnames(rslope), '.slope')
  cbind(yy[[i]][,c('id')], rmean, rsd, rslope)
}



##### Plotting for report
## Histogram of Wnv percentage by weeks
pdat = cbind(train$week, as.numeric(as.character(temp$WnvPresent)))
pdat = data.frame(pdat)
names(pdat) = c('week', 'WnvPresent')
pdat2 = ddply(pdat, 'week', summarise,
              percentage = sum(WnvPresent)/length(WnvPresent))
ggplot(pdat2) +
  geom_histogram(aes(x = week, y = percentage),stat = 'identity')

## Species effect
tab = tapply(as.numeric(as.character(temp$WnvPresent)),temp$Species,
             function(x)sum(x)/length(x))
tab2 = tapply(as.numeric(as.character(temp$WnvPresent)),temp$Species,length)
tab = data.frame(tab)
names(tab) = 'Proportion of WNV Cases'
tab$Counts = tab2
xtable(tab)

### Results
res = data.frame(matrix(0, nrow= 4, ncol = 3))
yy = c(2007, 2009, 2011, 2013)
for(i in 1:4){
  res[i, 1] = glmcv(train, y, yr = yy[i])
  res[i, 2] = rfcv(train, y, yr = yy[i])
  res[i, 3] = gbmcv(train, y, yr = yy[i])
  print(paste('round', i, 'finished'))
}
rownames(res) = yy
colnames(res) = c('GLM', 'RandomForest', 'Adaboost')

## Effect of K on GBM
res = data.frame(matrix(0, nrow= 4, ncol = 10))
yy = c(2007, 2009, 2011, 2013)
for(i in 1:4){
  for(j in 1:10){
    res[i, j] = gbmcv(train, y, yr = yy[i],knn.k = j)
  }
```

```
  print(paste('round', i, 'finished'))
}
rownames(res) = yy
colnames(res) = as.character(1:10)
keffect = apply(res, 2, mean)
keffect = data.frame('K' = 1:10, 'MeanAUC' = keffect)
ggplot(keffect) + geom_line(aes(x= K, y = MeanAUC))

### Tree number effect
res = data.frame(matrix(0, nrow= 4, ncol = 4))
yy = c(2007, 2009, 2011, 2013)
ntree = c(200, 400, 800)
for(i in 1:4){
  for(j in 1:2){
    res[i, j] = rfcv(train, y, yr = yy[i],nt = ntree[j])
  }
  print(paste('round', i, 'finished'))
}
treeeffect = apply(res, 2, mean)
treeeffect = data.frame('NumTree' =
                          c(200, 400, 800, 1200, 1500,
                            1600, 1700, 1800, 2000),
                        'MeanAUC' = treeeffect)
ggplot(treeeffect) + geom_line(aes(x= NumTree, y = MeanAUC))
### Get row counts
# xx = ddply(train, c('Latitude', 'Species', 'Date'),
#  summarise, N = length(Tmax))
# yy = ddply(test, c('Latitude', 'Species', 'Date'),
# summarise, N = length(Tmax))
# train = merge(train, xx)
# test = merge(test, yy)
# aa = ddply(dat, c('Date', 'Species', 'Latitude'),
# summarize, N = length(Station))
# aa = aa[order(aa$Latitude), ]
# bb = split(aa, aa$Species)
# xx = sapply(1:7, function(i)rollapply(bb[[i]][,'N'],
#  width = 3, sum, partial = T))
# xx = unlist(xx)
# aa$N2 = xx
# dat = merge(dat, aa)
###
```