Kiriakos\_MSDS\_664\_X70\_Week 8 Deep Learning

Cathy Kiriakos

April 29, 2020

## Week 8: Deep Learning using the UCI Machine Learning Repository: Breast Cancer Data Set

This week we will take a dive into deep learning using the UCI Machine Learning Breast Cancer data set and the R deep learning package H20 to train our data set.H20 is a popular machine learning application that has the ability to leverage many machine learning algorithms including Bayes, linear and logistic regression, and K-means clustering.

We will first start by installing H20 from <http://h2o-release.s3.amazonaws.com/h2o/rel-zahradnik/2/index.html> as the cran release was incompatable with R version 4.0.

Now we will load our data set downloaded from UCI <https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original)>

Attribute Information:

1. Sample code number: id number
2. Clump Thickness: 1 - 10
3. Uniformity of Cell Size: 1 - 10
4. Uniformity of Cell Shape: 1 - 10
5. Marginal Adhesion: 1 - 10
6. Single Epithelial Cell Size: 1 - 10
7. Bare Nuclei: 1 - 10
8. Bland Chromatin: 1 - 10
9. Normal Nucleoli: 1 - 10
10. Mitoses: 1 - 10
11. Class: (2 for benign, 4 for malignant)

bc <- read.csv("C:/Users/Cathy/Downloads/breast-cancer-wisconsin.csv", header = TRUE, strip.white=TRUE, col.names = c("Sample code number","Clump Thickness","Uniformity of Cell Size:","Uniformity of Cell Shape","Marginal Adhesion", "Single Epithelial", "Bare Nuclei", "Bland Chromatin", "Normal Nucleoli","Mitoses", "Diagnosis"))  
head(bc)

## Sample.code.number Clump.Thickness Uniformity.of.Cell.Size.  
## 1 1002945 5 4  
## 2 1015425 3 1  
## 3 1016277 6 8  
## 4 1017023 4 1  
## 5 1017122 8 10  
## 6 1018099 1 1  
## Uniformity.of.Cell.Shape Marginal.Adhesion Single.Epithelial Bare.Nuclei  
## 1 4 5 7 10  
## 2 1 1 2 2  
## 3 8 1 3 4  
## 4 1 3 2 1  
## 5 10 8 7 10  
## 6 1 1 2 10  
## Bland.Chromatin Normal.Nucleoli Mitoses Diagnosis  
## 1 3 2 1 2  
## 2 3 1 1 2  
## 3 3 7 1 2  
## 4 3 1 1 2  
## 5 9 7 1 4  
## 6 3 1 1 2

Now to, drop the sample number and call structure to gain some insight into our data:

bc<-subset(bc,select = -c(Sample.code.number))  
str(bc)

## 'data.frame': 698 obs. of 10 variables:  
## $ Clump.Thickness : int 5 3 6 4 8 1 2 2 4 1 ...  
## $ Uniformity.of.Cell.Size.: int 4 1 8 1 10 1 1 1 2 1 ...  
## $ Uniformity.of.Cell.Shape: int 4 1 8 1 10 1 2 1 1 1 ...  
## $ Marginal.Adhesion : int 5 1 1 3 8 1 1 1 1 1 ...  
## $ Single.Epithelial : int 7 2 3 2 7 2 2 2 2 1 ...  
## $ Bare.Nuclei : chr "10" "2" "4" "1" ...  
## $ Bland.Chromatin : int 3 3 3 3 9 3 3 1 2 3 ...  
## $ Normal.Nucleoli : int 2 1 7 1 7 1 1 1 1 1 ...  
## $ Mitoses : int 1 1 1 1 1 1 1 5 1 1 ...  
## $ Diagnosis : int 2 2 2 2 4 2 2 2 2 2 ...

For some reason the bare.Neuclei was imported as chr, so we need to make a quick conversion to integer:

bc$Bare.Nuclei <- sapply(bc$Bare.Nuclei,as.integer)

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion  
  
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion  
  
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion  
  
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion  
  
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion  
  
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion  
  
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion  
  
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion  
  
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion  
  
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion  
  
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion  
  
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion  
  
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion  
  
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion  
  
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion  
  
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

str(bc)

## 'data.frame': 698 obs. of 10 variables:  
## $ Clump.Thickness : int 5 3 6 4 8 1 2 2 4 1 ...  
## $ Uniformity.of.Cell.Size.: int 4 1 8 1 10 1 1 1 2 1 ...  
## $ Uniformity.of.Cell.Shape: int 4 1 8 1 10 1 2 1 1 1 ...  
## $ Marginal.Adhesion : int 5 1 1 3 8 1 1 1 1 1 ...  
## $ Single.Epithelial : int 7 2 3 2 7 2 2 2 2 1 ...  
## $ Bare.Nuclei : int 10 2 4 1 10 10 1 1 1 1 ...  
## $ Bland.Chromatin : int 3 3 3 3 9 3 3 1 2 3 ...  
## $ Normal.Nucleoli : int 2 1 7 1 7 1 1 1 1 1 ...  
## $ Mitoses : int 1 1 1 1 1 1 1 5 1 1 ...  
## $ Diagnosis : int 2 2 2 2 4 2 2 2 2 2 ...

Ok now that we have uniformity we can move on to our data investigation and get some summary stats for a baseline as shown below:

summary(bc)

## Clump.Thickness Uniformity.of.Cell.Size. Uniformity.of.Cell.Shape  
## Min. : 1.000 Min. : 1.000 Min. : 1.000   
## 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 1.000   
## Median : 4.000 Median : 1.000 Median : 1.000   
## Mean : 4.417 Mean : 3.138 Mean : 3.211   
## 3rd Qu.: 6.000 3rd Qu.: 5.000 3rd Qu.: 5.000   
## Max. :10.000 Max. :10.000 Max. :10.000   
##   
## Marginal.Adhesion Single.Epithelial Bare.Nuclei Bland.Chromatin   
## Min. : 1.000 Min. : 1.000 Min. : 1.000 Min. : 1.000   
## 1st Qu.: 1.000 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 2.000   
## Median : 1.000 Median : 2.000 Median : 1.000 Median : 3.000   
## Mean : 2.809 Mean : 3.218 Mean : 3.548 Mean : 3.438   
## 3rd Qu.: 4.000 3rd Qu.: 4.000 3rd Qu.: 6.000 3rd Qu.: 5.000   
## Max. :10.000 Max. :10.000 Max. :10.000 Max. :10.000   
## NA's :16   
## Normal.Nucleoli Mitoses Diagnosis   
## Min. : 1.00 Min. : 1.00 Min. :2.000   
## 1st Qu.: 1.00 1st Qu.: 1.00 1st Qu.:2.000   
## Median : 1.00 Median : 1.00 Median :2.000   
## Mean : 2.87 Mean : 1.59 Mean :2.691   
## 3rd Qu.: 4.00 3rd Qu.: 1.00 3rd Qu.:4.000   
## Max. :10.00 Max. :10.00 Max. :4.000   
##

We can see we have some incomplete obserations with the n.a. counts, in the bare Nuclei we will go ahead and remove all rows that have na values;at a loss of 16 observations this will not skew our data.

bc <- na.omit(bc)

Lastly, I’ll convert the diagnoses to string and the actual diagnosis for ease for reading the results:

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

bc<-bc %>%  
 mutate(Diagnosis = ifelse(Diagnosis=="2", "benign", ifelse(Diagnosis=="4","malignant", NA)))  
head(bc)

## Clump.Thickness Uniformity.of.Cell.Size. Uniformity.of.Cell.Shape  
## 1 5 4 4  
## 2 3 1 1  
## 3 6 8 8  
## 4 4 1 1  
## 5 8 10 10  
## 6 1 1 1  
## Marginal.Adhesion Single.Epithelial Bare.Nuclei Bland.Chromatin  
## 1 5 7 10 3  
## 2 1 2 2 3  
## 3 1 3 4 3  
## 4 3 2 1 3  
## 5 8 7 10 9  
## 6 1 2 10 3  
## Normal.Nucleoli Mitoses Diagnosis  
## 1 2 1 benign  
## 2 1 1 benign  
## 3 7 1 benign  
## 4 1 1 benign  
## 5 7 1 malignant  
## 6 1 1 benign

summary(bc)

## Clump.Thickness Uniformity.of.Cell.Size. Uniformity.of.Cell.Shape  
## Min. : 1.000 Min. : 1.000 Min. : 1.000   
## 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 1.000   
## Median : 4.000 Median : 1.000 Median : 1.000   
## Mean : 4.441 Mean : 3.154 Mean : 3.218   
## 3rd Qu.: 6.000 3rd Qu.: 5.000 3rd Qu.: 5.000   
## Max. :10.000 Max. :10.000 Max. :10.000   
## Marginal.Adhesion Single.Epithelial Bare.Nuclei Bland.Chromatin   
## Min. : 1.000 Min. : 1.000 Min. : 1.000 Min. : 1.000   
## 1st Qu.: 1.000 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 2.000   
## Median : 1.000 Median : 2.000 Median : 1.000 Median : 3.000   
## Mean : 2.833 Mean : 3.236 Mean : 3.548 Mean : 3.446   
## 3rd Qu.: 4.000 3rd Qu.: 4.000 3rd Qu.: 6.000 3rd Qu.: 5.000   
## Max. :10.000 Max. :10.000 Max. :10.000 Max. :10.000   
## Normal.Nucleoli Mitoses Diagnosis   
## Min. : 1.000 Min. : 1.000 Length:682   
## 1st Qu.: 1.000 1st Qu.: 1.000 Class :character   
## Median : 1.000 Median : 1.000 Mode :character   
## Mean : 2.872 Mean : 1.604   
## 3rd Qu.: 4.000 3rd Qu.: 1.000   
## Max. :10.000 Max. :10.000

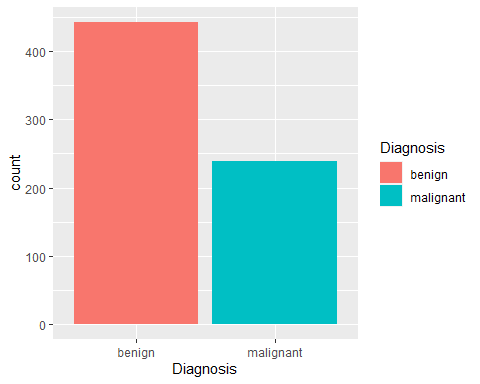
head(bc)

## Clump.Thickness Uniformity.of.Cell.Size. Uniformity.of.Cell.Shape  
## 1 5 4 4  
## 2 3 1 1  
## 3 6 8 8  
## 4 4 1 1  
## 5 8 10 10  
## 6 1 1 1  
## Marginal.Adhesion Single.Epithelial Bare.Nuclei Bland.Chromatin  
## 1 5 7 10 3  
## 2 1 2 2 3  
## 3 1 3 4 3  
## 4 3 2 1 3  
## 5 8 7 10 9  
## 6 1 2 10 3  
## Normal.Nucleoli Mitoses Diagnosis  
## 1 2 1 benign  
## 2 1 1 benign  
## 3 7 1 benign  
## 4 1 1 benign  
## 5 7 1 malignant  
## 6 1 1 benign

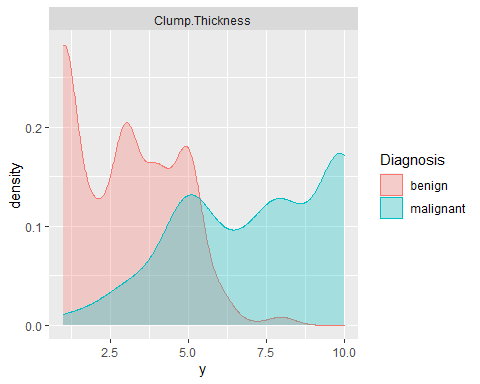
For our analysis, out independent variable will be our Diagnosis variable. Now we can get a quick idea of the amount of malignant and benign diagnoses:

We can see in the data there are 457 benign cases, and 241 malignant cases. So out of 698 diagnoses we;ve got 65% that are benign and so roughly there are 35% that are malignant which seems pretty high.

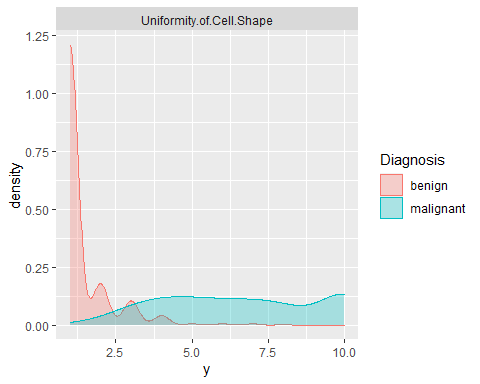
library(ggplot2)  
ggplot(bc, aes(x = Diagnosis, fill = Diagnosis))+  
 geom\_bar()

 We can get a look at clump thickness vs. diagnosis below:

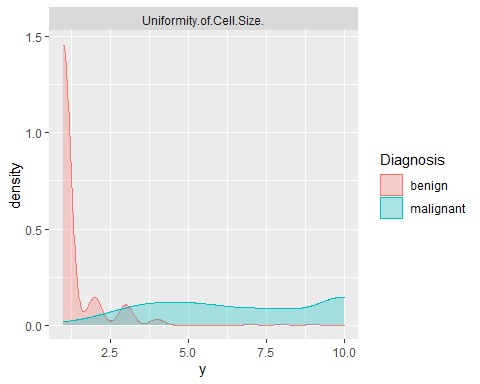
library(tidyr)  
gather(bc, x, y, Clump.Thickness) %>%  
 ggplot(aes(x = y, color = Diagnosis, fill = Diagnosis)) +  
 geom\_density(alpha = 0.3) +  
 facet\_wrap( ~ x, scales = "free", ncol = 3)

 We can get this same view looking at the relationship to Uniformity.of.Cell.Shape

gather(bc, x, y, Uniformity.of.Cell.Shape) %>%  
 ggplot(aes(x = y, color = Diagnosis, fill = Diagnosis)) +  
 geom\_density(alpha = 0.3) +  
 facet\_wrap( ~ x, scales = "free", ncol = 3)

 We can get this same view looking at the relationship to Uniformity.of.Cell.Size.

gather(bc, x, y, Uniformity.of.Cell.Size.) %>%  
 ggplot(aes(x = y, color = Diagnosis, fill = Diagnosis)) +  
 geom\_density(alpha = 0.3) +  
 facet\_wrap( ~ x, scales = "free", ncol = 3)

 With a quick view of our data, I could assume that clump Thickness has a strong relationship to diagnoses. Now that we’ve cleaned up the data and got a pulse using exploratory analysis. For the correlations in H20, I went ahead and converted our results to binary; where malignant = 1 and benign = 0 as shown below:

bc1 <- bc %>%  
 mutate(Diagnosis = ifelse(Diagnosis=="benign", 0, ifelse(Diagnosis=="malignant",1, NA)))  
head(bc)

## Clump.Thickness Uniformity.of.Cell.Size. Uniformity.of.Cell.Shape  
## 1 5 4 4  
## 2 3 1 1  
## 3 6 8 8  
## 4 4 1 1  
## 5 8 10 10  
## 6 1 1 1  
## Marginal.Adhesion Single.Epithelial Bare.Nuclei Bland.Chromatin  
## 1 5 7 10 3  
## 2 1 2 2 3  
## 3 1 3 4 3  
## 4 3 2 1 3  
## 5 8 7 10 9  
## 6 1 2 10 3  
## Normal.Nucleoli Mitoses Diagnosis  
## 1 2 1 benign  
## 2 1 1 benign  
## 3 7 1 benign  
## 4 1 1 benign  
## 5 7 1 malignant  
## 6 1 1 benign

Initializing H20

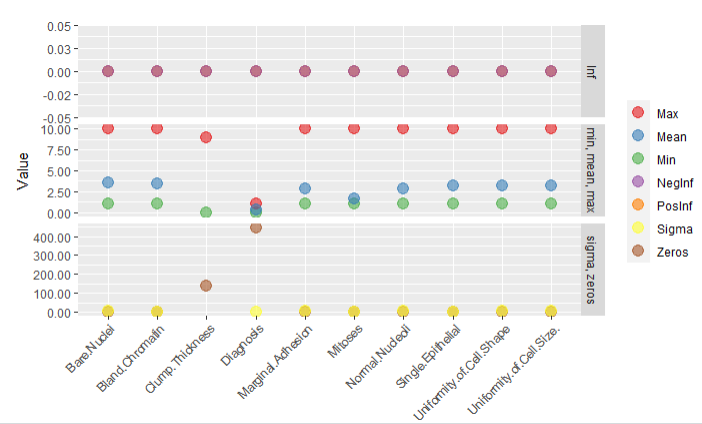
h2o.init(nthreads = -1)

Changing our binary breast cancer data into an H20 frame:

bc1 <- as.h2o(bc1)

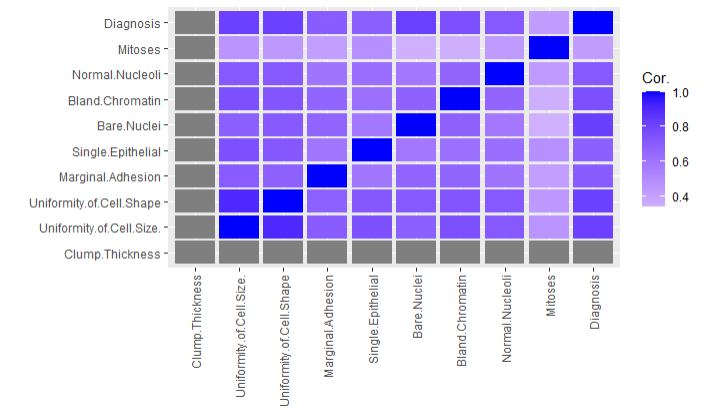
The below plot showing summary stats using H20, I’ve used guidance from the website below: [1]<https://www.shirin-glander.de/2018/06/intro_to_ml_workshop_heidelberg/> Its helpful in gainin some insight into our data

h2o.describe(bc1) %>%  
 gather(x, y, Zeros:Sigma) %>%  
 mutate(group = ifelse(x %in% c("Min", "Max", "Mean"), "min, mean, max",   
 ifelse(x %in% c("NegInf", "PosInf"), "Inf", "sigma, zeros"))) %>%   
 ggplot(aes(x = Label, y = as.numeric(y), color = x)) +  
 geom\_point(size = 4, alpha = 0.6) +  
 scale\_color\_brewer(palette = "Set1") +  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust = 1)) +  
 facet\_grid(group ~ ., scales = "free") +  
 labs(x = "Feature",  
 y = "Value",  
 color = "")+  
 scale\_y\_continuous(  
 labels = scales::number\_format(accuracy = 0.01))



Next we will get a heat map, of our corrlations for another visualization on the data [1], showing us that uniformity of cell size, uniformity of cell shape and bare.Nuclei are correlated higher to a positive diagnosis.

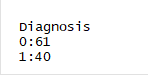
library(reshape2) # for melting  
  
bc1[, 1] <- h2o.asfactor(bc1[, 1])  
  
cor <- h2o.cor(bc1)  
rownames(cor) <- colnames(cor)  
  
melt(cor) %>%  
 mutate(Var2 = rep(rownames(cor), nrow(cor))) %>%  
 mutate(Var2 = factor(Var2, levels = colnames(cor))) %>%  
 mutate(variable = factor(variable, levels = colnames(cor))) %>%  
 ggplot(aes(x = variable, y = Var2, fill = value)) +   
 geom\_tile(width = 0.9, height = 0.9) +  
 scale\_fill\_gradient2(low = "white", high = "blue", name = "Cor.") +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust = 1)) +  
 labs(x = "",   
 y = "")

Next We will start by training our data set

splits <- h2o.splitFrame(bc1,   
 ratios = c(0.7, 0.15),   
 seed = 1)  
  
train <- splits[[1]]  
valid <- splits[[2]]  
test <- splits[[3]]  
  
response <- "Diagnosis"  
features <- setdiff(colnames(train), response)

Now we can get a look at those splits on training set:

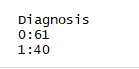
summary(as.factor(train$Diagnosis), exact\_quantiles = TRUE)



Showing that our benign contains 312, and our malignant has 167

We will do the same for our validation and testing sets:

summary(as.factor(test$Diagnosis), exact\_quantiles = TRUE)



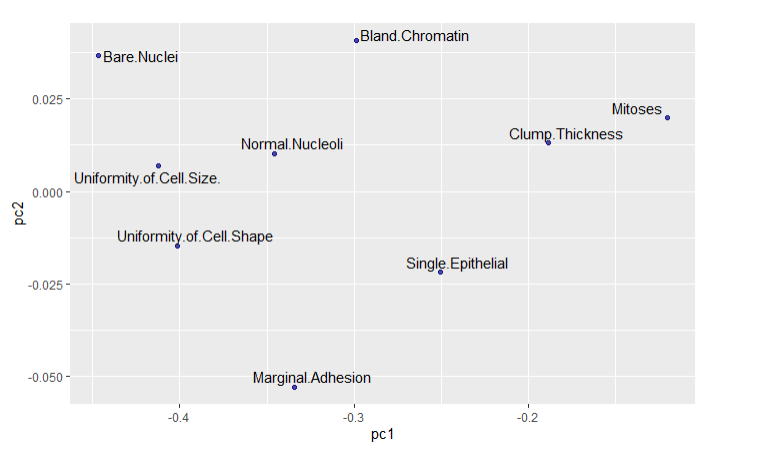
summary(as.factor(valid$Diagnosis), exact\_quantiles = TRUE)

Now we will conduct some priciple component analysis on our sets:

pca <- h2o.prcomp(training\_frame = train,  
 x = features,  
 validation\_frame = valid,  
 transform = "NORMALIZE",  
 impute\_missing = TRUE,  
 k = 3,  
 seed = 42)

Below we’re assigning eigen vector values to our priciple component analysis, and get a view of our results. Showing a relationship between mitoses, bare neuclei bland comatin and a positive diagnoses.

ev <- as.data.frame(pca@model$eigenvectors)  
ev1 <- ev[-c(1:8),]  
ev1$label <- features  
  
library(ggrepel)  
ggplot(ev1, aes(x = pc1, y = pc2, label = label)) +  
 geom\_point(color = "navy", alpha = 0.7) +  
 geom\_text\_repel()



We will next perform a random forest classification using H20. Its important to understand the functionality of our hyperparameters, the description below was obtained from the h20 website at: <https://www.h2o.ai/blog/hyperparameter-optimization-in-h2o-grid-search-random-search-and-the-future/>

Nearly all model algorithms used in machine learning have a set of tuning “knobs” which affect how the learning algorithm fits the model to the data. Examples are the regularization settings alpha and lambda for Generalized Linear Modeling or ntrees and max\_depth for Gradient Boosted Models. These knobs are called hyperparameters to distinguish them from internal model parameters, such as GLM’s beta coefficients or Deep Learning’s weights, which get learned from the data during the model training process.:

hyper\_params <- list(  
 ntrees = c(25, 50, 75, 100),  
 max\_depth = c(10, 20, 30),  
 min\_rows = c(1, 3, 5)  
 )  
  
search\_criteria <- list(  
 strategy = "RandomDiscrete",   
 max\_models = 50,  
 max\_runtime\_secs = 360,  
 stopping\_rounds = 5,   
 stopping\_metric = "AUC",   
 stopping\_tolerance = 0.0005,  
 seed = 42  
 )

Now we will create our RF Grid, its important to note that we can change our model by assigning a different value to our algorithm function (h2o.random Forest,h2o.gbm for Gradient Boosting Trees )

rf\_grid <- h2o.grid(algorithm = "randomForest",   
 x = features,  
 y = response,  
 grid\_id = "rf\_grid",  
 training\_frame = train,  
 validation\_frame = valid,  
 nfolds = 25,   
 fold\_assignment = "Stratified",  
 hyper\_params = hyper\_params,  
 search\_criteria = search\_criteria,  
 seed = 42  
 )

sorted\_grid <- h2o.getGrid(grid\_id = "rf\_grid", sort\_by = "mse")  
print(sorted\_grid)  
best\_model <- h2o.getModel(sorted\_grid@model\_ids[[1]])  
summary(best\_model)

H2O Grid Details

================

Grid ID: rf\_grid

Used hyper parameters:

- max\_depth

- min\_rows

- ntrees

Number of models: 108

Number of failed models: 0

Hyper-Parameter Search Summary: ordered by increasing mse

max\_depth min\_rows ntrees model\_ids mse

1 20 1.0 100 rf\_grid\_model\_56 0.025688642426378847

2 30 1.0 100 rf\_grid\_model\_61 0.025688642426378847

3 10 1.0 100 rf\_grid\_model\_62 0.025688642426378847

4 20 1.0 100 rf\_grid\_model\_92 0.025688642426378847

5 30 1.0 100 rf\_grid\_model\_97 0.025688642426378847

---

max\_depth min\_rows ntrees model\_ids mse

103 30 5.0 75 rf\_grid\_model\_18 0.07336108073186036

104 20 5.0 75 rf\_grid\_model\_31 0.07336108073186036

105 10 5.0 75 rf\_grid\_model\_6 0.07336108073186036

106 30 5.0 100 rf\_grid\_model\_22 0.07386276119556708

107 10 5.0 100 rf\_grid\_model\_29 0.07386276119556708

108 20 5.0 100 rf\_grid\_model\_7 0.07386276119556708

Model Details:

==============

H2ORegressionModel: drf

Model Key: rf\_grid\_model\_56

Model Summary:

number\_of\_trees number\_of\_internal\_trees model\_size\_in\_bytes min\_depth max\_depth mean\_depth min\_leaves max\_leaves mean\_leaves

1 100 100 31320 5 10 7.39000 13 28 20.21000

H2ORegressionMetrics: drf

\*\* Reported on training data. \*\*

\*\* Metrics reported on Out-Of-Bag training samples \*\*

MSE: 0.02674448

RMSE: 0.1635374

MAE: 0.06105604

RMSLE: 0.1118893

Mean Residual Deviance : 0.02674448

H2ORegressionMetrics: drf

\*\* Reported on validation data. \*\*

MSE: 0.02584387

RMSE: 0.1607603

MAE: 0.05926471

RMSLE: 0.1113517

Mean Residual Deviance : 0.02584387

H2ORegressionMetrics: drf

\*\* Reported on cross-validation data. \*\*

\*\* 25-fold cross-validation on training data (Metrics computed for combined holdout predictions) \*\*

MSE: 0.02568864

RMSE: 0.1602768

MAE: 0.06099513

RMSLE: 0.111182

Mean Residual Deviance : 0.02568864

Cross-Validation Metrics Summary:

mean sd cv\_1\_valid cv\_2\_valid cv\_3\_valid cv\_4\_valid cv\_5\_valid cv\_6\_valid cv\_7\_valid cv\_8\_valid

mae 0.060337756 0.030708145 0.08461539 0.04095238 0.035714287 0.0236 0.0972 0.05882353 0.0574 0.09535714

mean\_residual\_deviance 0.02497411 0.021258991 0.050707694 0.004466667 0.006740476 0.0029 0.08046 0.020988235 0.014101 0.047138393

mse 0.02497411 0.021258991 0.050707694 0.004466667 0.006740476 0.0029 0.08046 0.020988235 0.014101 0.047138393

r2 0.8859988 0.10068533 0.78576 0.97811335 0.9628431 0.981875 0.63023895 0.90809697 0.9435056 0.76902187

residual\_deviance 0.02497411 0.021258991 0.050707694 0.004466667 0.006740476 0.0029 0.08046 0.020988235 0.014101 0.047138393

rmse 0.14111435 0.07260658 0.22518368 0.06683312 0.08210041 0.05385165 0.28365472 0.14487317 0.11874763 0.21711378

rmsle 0.09645599 0.051508248 0.14385742 0.050616544 0.057168737 0.03641498 0.19541219 0.10432523 0.07080189 0.15578645

cv\_9\_valid cv\_10\_valid cv\_11\_valid cv\_12\_valid cv\_13\_valid cv\_14\_valid cv\_15\_valid cv\_16\_valid cv\_17\_valid cv\_18\_valid

mae 0.07913043 0.005 0.015 0.11846154 0.08068841 0.04142857 0.055925924 0.094444446 0.028947368 0.08764706

mean\_residual\_deviance 0.04566087 9.0E-5 0.0017875 0.0426 0.028095683 0.009228571 0.029122222 0.057533335 0.0037631579 0.04525294

mse 0.04566087 9.0E-5 0.0017875 0.0426 0.028095683 0.009228571 0.029122222 0.057533335 0.0037631579 0.04525294

r2 0.8082968 0.999625 0.99046665 0.820015 0.8761449 0.9623167 0.8483564 0.66712856 0.98382735 0.81835973

residual\_deviance 0.04566087 9.0E-5 0.0017875 0.0426 0.028095683 0.009228571 0.029122222 0.057533335 0.0037631579 0.04525294

rmse 0.21368404 0.009486833 0.042278837 0.20639767 0.16761766 0.096065454 0.17065234 0.23986107 0.061344583 0.21272738

rmsle 0.15039274 0.0054865982 0.031116107 0.11618284 0.13143545 0.062157225 0.12458951 0.17239398 0.033979885 0.16139354

cv\_19\_valid cv\_20\_valid cv\_21\_valid cv\_22\_valid cv\_23\_valid cv\_24\_valid cv\_25\_valid

mae 0.0885 0.095 0.0371875 0.0555 0.051944446 0.063725494 0.01625

mean\_residual\_deviance 0.037865 0.03048 0.009651562 0.023495 0.0121125 0.017286928 0.002825

mse 0.037865 0.03048 0.009651562 0.023495 0.0121125 0.017286928 0.002825

r2 0.8422292 0.866022 0.95882 0.88811904 0.95155 0.9243042 0.9849333

residual\_deviance 0.037865 0.03048 0.009651562 0.023495 0.0121125 0.017286928 0.002825

rmse 0.19458932 0.17458522 0.098242365 0.1532808 0.1100568 0.13147977 0.05315073

rmsle 0.11459191 0.119457446 0.05401474 0.12444912 0.059770294 0.08771916 0.047885723

Scoring History:

timestamp duration number\_of\_trees training\_rmse training\_mae training\_deviance validation\_rmse validation\_mae validation\_deviance

1 2020-05-03 13:18:50 29.520 sec 0 NA NA NA NA NA NA

2 2020-05-03 13:18:50 29.522 sec 1 0.26491 0.07018 0.07018 0.28440 0.08333 0.08088

3 2020-05-03 13:18:50 29.524 sec 2 0.24423 0.06316 0.05965 0.22822 0.06618 0.05208

4 2020-05-03 13:18:50 29.525 sec 3 0.25774 0.07465 0.06643 0.20412 0.07353 0.04167

5 2020-05-03 13:18:50 29.527 sec 4 0.24596 0.07226 0.06050 0.19687 0.06740 0.03876

---

timestamp duration number\_of\_trees training\_rmse training\_mae training\_deviance validation\_rmse validation\_mae validation\_deviance

96 2020-05-03 13:18:50 29.830 sec 95 0.16278 0.06075 0.02650 0.16356 0.05996 0.02675

97 2020-05-03 13:18:50 29.835 sec 96 0.16319 0.06095 0.02663 0.16342 0.05974 0.02671

98 2020-05-03 13:18:50 29.840 sec 97 0.16330 0.06101 0.02667 0.16354 0.05973 0.02675

99 2020-05-03 13:18:50 29.847 sec 98 0.16345 0.06113 0.02671 0.16240 0.05937 0.02637

100 2020-05-03 13:18:50 29.852 sec 99 0.16329 0.06098 0.02666 0.16226 0.05976 0.02633

101 2020-05-03 13:18:50 29.857 sec 100 0.16354 0.06106 0.02674 0.16076 0.05926 0.02584

Variable Importances: (Extract with `h2o.varimp`)

=================================================

Variable Importances:

variable relative\_importance scaled\_importance percentage

1 Uniformity.of.Cell.Shape 3081.577881 1.000000 0.322439

2 Uniformity.of.Cell.Size. 2338.143555 0.758749 0.244650

3 Bare.Nuclei 1540.237183 0.499821 0.161162

4 Bland.Chromatin 866.415039 0.281160 0.090657

5 Normal.Nucleoli 660.613098 0.214375 0.069123

6 Single.Epithelial 525.331055 0.170475 0.054968

7 Clump.Thickness 332.633820 0.107943 0.034805

8 Marginal.Adhesion 157.178864 0.051006 0.016446

9 Mitoses 54.963165 0.017836 0.005751

Viewing the best random forest model sorted by our mean squared error, we can see that we have a well-suited model that would tell us again that uniformity of cell shape and size are highly related to a positive diagnosis.