Liver Disease Prediction Report

Ganesh Shelke

07/06/2019

Introduction

This project shows use of machine learning and XGBoost (eXtreme Gradient Boosting) algorithm to predict liver disease risk in patients.

Objective

Patients with Liver disease have been continuously increasing because of excessive consumption of alcohol, inhale of harmful gases, intake of contaminated food, pickles and drugs. This dataset was used to evaluate prediction algorithms in an effort to reduce burden on doctors using XGBoost (eXtreme Gradient Boosting) algorithm.

The Dataset

The data that we will be using has been sourced from https://www.kaggle.com/uciml/indian-liver-patient-records. This data set contains 416 liver patient records and 167 non liver patient records collected from North East of Andhra Pradesh, India. The "Dataset" column is a class label used to divide groups into liver patient (liver disease) or not (no disease). This data set contains 441 male patient records and 142 female patient records. Any patient whose age exceeded 89 is listed as being of age "90".

##Libraries required:

Libraries we have used are: caret, xgboost, methods

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
```

Data analysis

Exploration

Let's have a look at the features present in our dataset

```
##  [1] "Age"                         "Gender"
##  [3] "Total_Bilirubin"             "Direct_Bilirubin"
##  [5] "Alkaline_Phosphotase"        "Alamine_Aminotransferase"
##  [7] "Aspartate_Aminotransferase" "Total_Protiens"
##  [9] "Albumin"                     "Albumin_and_Globulin_Ratio"
## [11] "Dataset"
```

The last feature, 'Dataset', contains the label. 1 indicates a liver patient (disease) and 2 indicates a non-liver patient (no disease)

The first 6 observations are:

```
head(data)
```

```
##   Age Gender Total_Bilirubin Direct_Bilirubin Alkaline_Phosphotase
## 1  65 Female             0.7              0.1                  187
## 2  62   Male            10.9              5.5                  699
## 3  62   Male             7.3              4.1                  490
## 4  58   Male             1.0              0.4                  182
## 5  72   Male             3.9              2.0                  195
## 6  46   Male             1.8              0.7                  208
##   Alamine_Aminotransferase Aspartate_Aminotransferase Total_Protiens
## 1                       16                         18            6.8
## 2                       64                        100            7.5
## 3                       60                         68            7.0
## 4                       14                         20            6.8
## 5                       27                         59            7.3
## 6                       19                         14            7.6
##   Albumin Albumin_and_Globulin_Ratio Dataset
## 1     3.3                      0.90       1
## 2     3.2                      0.74       1
## 3     3.3                      0.89       1
## 4     3.4                      1.00       1
## 5     2.4                      0.40       1
## 6     4.4                      1.30       1
```

A quick view into the entire data set using 'summary' function

```
summary(data)
```

```
##       Age           Gender    Total_Bilirubin  Direct_Bilirubin
##  Min.   : 4.00   Female:142   Min.   : 0.400   Min.   : 0.100
##  1st Qu.:33.00   Male  :441   1st Qu.: 0.800   1st Qu.: 0.200
##  Median :45.00                Median : 1.000   Median : 0.300
##  Mean   :44.75                Mean   : 3.299   Mean   : 1.486
##  3rd Qu.:58.00                3rd Qu.: 2.600   3rd Qu.: 1.300
##  Max.   :90.00                Max.   :75.000   Max.   :19.700
##
##  Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase
##  Min.   :  63.0       Min.   :  10.00          Min.   :  10.0
##  1st Qu.: 175.5       1st Qu.:  23.00          1st Qu.:  25.0
##  Median : 208.0       Median :  35.00          Median :  42.0
##  Mean   : 290.6       Mean   :  80.71          Mean   : 109.9
```

```
##  3rd Qu.: 298.0       3rd Qu.:  60.50          3rd Qu.:  87.0
##  Max.   :2110.0       Max.   :2000.00          Max.   :4929.0
##
##  Total_Protiens     Albumin      Albumin_and_Globulin_Ratio
##  Min.   :2.700   Min.   :0.900   Min.   :0.3000
##  1st Qu.:5.800   1st Qu.:2.600   1st Qu.:0.7000
##  Median :6.600   Median :3.100   Median :0.9300
##  Mean   :6.483   Mean   :3.142   Mean   :0.9471
##  3rd Qu.:7.200   3rd Qu.:3.800   3rd Qu.:1.1000
##  Max.   :9.600   Max.   :5.500   Max.   :2.8000
##                                  NA's   :4
##     Dataset
##  Min.   :1.000
##  1st Qu.:1.000
##  Median :1.000
##  Mean   :1.286
##  3rd Qu.:2.000
##  Max.   :2.000
##
```

We see that the data is clean except one feature having missing values: Albumin_and_Globulin_Ratio.

Data Cleaning

Replacing NAs in Albumin_and_Globulin_Ratio by it's mean value i.e. 0.9470639

The next step is to replace non-numeric values with numeric values for feature Gender.

Now check if there are any features which are highly co-related to each other and then remove them and retain the other features.

```
tmp <- cor(data)
tmp[!lower.tri(tmp)] <- 0
data.new <- data[,!apply(tmp,2,function(x) any(x > 0.8))]
data = data.new
names(data)
```

```
##  [1] "Age"                          "Gender"
##  [3] "Direct_Bilirubin"             "Alkaline_Phosphotase"
##  [5] "Alamine_Aminotransferase"     "Aspartate_Aminotransferase"
##  [7] "Total_Protiens"               "Albumin"
##  [9] "Albumin_and_Globulin_Ratio" "Dataset"
```

We can see here that Total_Bilirubin has been eliminated from the list of features since it was very highly correlated with an existing feature.

Now we convert the label variable into the 0 or 1 values that XGBoost expects for binary classification

```
data$Dataset <- data$Dataset - 1
```

Data Splitting

Let us split the available data into train and test sets with a 75:25 ratio.

```
sample_size <- floor(0.75 * nrow(data))
set.seed(123)
train_ind <- sample(seq_len(nrow(data)), size = sample_size)
train <- data[train_ind, ]
test <- data[-train_ind, ]
train_label <- as.numeric(train$Dataset) #labels: if the person has liver disease or not
test_label <- as.numeric(test$Dataset)
```

Training the XGBoost model

converting train and test sets to Formal class dgcMatrix (sparse numeric matrices) to use the XGBoost. DMatrix is an internal data structure used by XGBoost which is optimized for both memory efficiency and training speed.

```
train <- as(as.matrix(train[ , -which(names(train) %in% c("Dataset"))]), "dgCMatrix")
test <- as(as.matrix(test[ , -which(names(test) %in% c("Dataset"))]), "dgCMatrix")
dtrain <- xgb.DMatrix(data = train, label=train_label) #External pointers of class 'xgb.DMatrix'
dtest <- xgb.DMatrix(data = test, label=test_label)
watchlist <- list(train=dtrain, test=dtest)
```

Let's train the XGBoost Model.

We use xgb.train() function. It is an advance function to train XGBoost model Parameters used: max.depth: maximum depth of tree eta: controls learning rate (lower eta implies more robust but slow model) nthread: number of threads nrounds: lower eta implies larger values for nrounds early_stopping_rounds: If set to an integer k, training with a validation set will stop if the performance doesn't improve for k rounds.

```
xgbModel <- xgb.train(data = dtrain, max.depth = 100, eta = 0.001,
                      nthread = 2,  nround = 10000,
                      watchlist=watchlist, objective = "binary:logistic", early_stopping_rounds = 300)
```

```
## [1]  train-error:0.116705    test-error:0.321918
## Multiple eval metrics are present. Will use test_error for early stopping.
## Will train until test_error hasn't improved in 300 rounds.
##
## [2]  train-error:0.112128    test-error:0.308219
## [3]  train-error:0.116705    test-error:0.301370
## [4]  train-error:0.118993    test-error:0.321918
## [5]  train-error:0.116705    test-error:0.308219
## [6]  train-error:0.118993    test-error:0.321918
## [7]  train-error:0.116705    test-error:0.308219
## [8]  train-error:0.121281    test-error:0.308219
## [9]  train-error:0.121281    test-error:0.308219
## [10] train-error:0.121281    test-error:0.308219
## [11] train-error:0.121281    test-error:0.308219
## [12] train-error:0.121281    test-error:0.308219
## [13] train-error:0.121281    test-error:0.308219
## [14] train-error:0.121281    test-error:0.308219
## [15] train-error:0.121281    test-error:0.308219
## [16] train-error:0.121281    test-error:0.308219
## [17] train-error:0.121281    test-error:0.308219
## [18] train-error:0.121281    test-error:0.308219
```

```
## [19] train-error:0.121281    test-error:0.308219
## [20] train-error:0.121281    test-error:0.308219
## [21] train-error:0.121281    test-error:0.308219
## [22] train-error:0.121281    test-error:0.308219
## [23] train-error:0.121281    test-error:0.308219
## [24] train-error:0.121281    test-error:0.308219
## [25] train-error:0.121281    test-error:0.308219
## [26] train-error:0.121281    test-error:0.308219
## [27] train-error:0.121281    test-error:0.308219
## [28] train-error:0.121281    test-error:0.308219
## [29] train-error:0.121281    test-error:0.308219
## [30] train-error:0.121281    test-error:0.308219
## [31] train-error:0.121281    test-error:0.308219
## [32] train-error:0.121281    test-error:0.308219
## [33] train-error:0.121281    test-error:0.308219
## [34] train-error:0.121281    test-error:0.308219
## [35] train-error:0.121281    test-error:0.308219
## [36] train-error:0.121281    test-error:0.308219
## [37] train-error:0.121281    test-error:0.308219
## [38] train-error:0.121281    test-error:0.308219
## [39] train-error:0.121281    test-error:0.308219
## [40] train-error:0.121281    test-error:0.308219
## [41] train-error:0.116705    test-error:0.308219
## [42] train-error:0.118993    test-error:0.308219
## [43] train-error:0.118993    test-error:0.308219
## [44] train-error:0.121281    test-error:0.308219
## [45] train-error:0.116705    test-error:0.308219
## [46] train-error:0.114416    test-error:0.321918
## [47] train-error:0.114416    test-error:0.321918
## [48] train-error:0.116705    test-error:0.328767
## [49] train-error:0.116705    test-error:0.328767
## [50] train-error:0.116705    test-error:0.328767
## [51] train-error:0.114416    test-error:0.328767
## [52] train-error:0.112128    test-error:0.328767
## [53] train-error:0.112128    test-error:0.328767
## [54] train-error:0.112128    test-error:0.328767
## [55] train-error:0.109840    test-error:0.328767
## [56] train-error:0.109840    test-error:0.315068
## [57] train-error:0.109840    test-error:0.315068
## [58] train-error:0.109840    test-error:0.315068
## [59] train-error:0.109840    test-error:0.315068
## [60] train-error:0.109840    test-error:0.315068
## [61] train-error:0.109840    test-error:0.315068
## [62] train-error:0.109840    test-error:0.315068
## [63] train-error:0.109840    test-error:0.315068
## [64] train-error:0.109840    test-error:0.315068
## [65] train-error:0.109840    test-error:0.315068
## [66] train-error:0.109840    test-error:0.315068
## [67] train-error:0.112128    test-error:0.315068
## [68] train-error:0.112128    test-error:0.321918
## [69] train-error:0.112128    test-error:0.321918
## [70] train-error:0.114416    test-error:0.321918
## [71] train-error:0.114416    test-error:0.321918
## [72] train-error:0.112128    test-error:0.321918
```

```
## [73] train-error:0.112128    test-error:0.328767
## [74] train-error:0.112128    test-error:0.328767
## [75] train-error:0.112128    test-error:0.321918
## [76] train-error:0.112128    test-error:0.321918
## [77] train-error:0.112128    test-error:0.321918
## [78] train-error:0.112128    test-error:0.321918
## [79] train-error:0.109840    test-error:0.321918
## [80] train-error:0.109840    test-error:0.328767
## [81] train-error:0.107551    test-error:0.321918
## [82] train-error:0.107551    test-error:0.321918
## [83] train-error:0.107551    test-error:0.328767
## [84] train-error:0.107551    test-error:0.328767
## [85] train-error:0.105263    test-error:0.328767
## [86] train-error:0.098398    test-error:0.328767
## [87] train-error:0.098398    test-error:0.321918
## [88] train-error:0.098398    test-error:0.315068
## [89] train-error:0.100686    test-error:0.321918
## [90] train-error:0.105263    test-error:0.315068
## [91] train-error:0.105263    test-error:0.315068
## [92] train-error:0.102975    test-error:0.315068
## [93] train-error:0.102975    test-error:0.315068
## [94] train-error:0.105263    test-error:0.315068
## [95] train-error:0.105263    test-error:0.315068
## [96] train-error:0.105263    test-error:0.315068
## [97] train-error:0.105263    test-error:0.315068
## [98] train-error:0.105263    test-error:0.315068
## [99] train-error:0.105263    test-error:0.315068
## [100]    train-error:0.105263    test-error:0.315068
## [101]    train-error:0.102975    test-error:0.315068
## [102]    train-error:0.102975    test-error:0.315068
## [103]    train-error:0.102975    test-error:0.315068
## [104]    train-error:0.102975    test-error:0.315068
## [105]    train-error:0.102975    test-error:0.315068
## [106]    train-error:0.102975    test-error:0.315068
## [107]    train-error:0.102975    test-error:0.315068
## [108]    train-error:0.102975    test-error:0.315068
## [109]    train-error:0.102975    test-error:0.315068
## [110]    train-error:0.102975    test-error:0.315068
## [111]    train-error:0.102975    test-error:0.315068
## [112]    train-error:0.102975    test-error:0.315068
## [113]    train-error:0.100686    test-error:0.308219
## [114]    train-error:0.100686    test-error:0.308219
## [115]    train-error:0.100686    test-error:0.315068
## [116]    train-error:0.100686    test-error:0.315068
## [117]    train-error:0.102975    test-error:0.321918
## [118]    train-error:0.102975    test-error:0.328767
## [119]    train-error:0.102975    test-error:0.328767
## [120]    train-error:0.102975    test-error:0.328767
## [121]    train-error:0.102975    test-error:0.335616
## [122]    train-error:0.102975    test-error:0.335616
## [123]    train-error:0.102975    test-error:0.335616
## [124]    train-error:0.102975    test-error:0.335616
## [125]    train-error:0.102975    test-error:0.335616
## [126]    train-error:0.102975    test-error:0.335616
```

```
## [127]    train-error:0.102975    test-error:0.335616
## [128]    train-error:0.105263    test-error:0.335616
## [129]    train-error:0.102975    test-error:0.335616
## [130]    train-error:0.105263    test-error:0.342466
## [131]    train-error:0.105263    test-error:0.335616
## [132]    train-error:0.105263    test-error:0.335616
## [133]    train-error:0.105263    test-error:0.335616
## [134]    train-error:0.105263    test-error:0.342466
## [135]    train-error:0.105263    test-error:0.342466
## [136]    train-error:0.105263    test-error:0.342466
## [137]    train-error:0.102975    test-error:0.342466
## [138]    train-error:0.102975    test-error:0.335616
## [139]    train-error:0.105263    test-error:0.342466
## [140]    train-error:0.105263    test-error:0.342466
## [141]    train-error:0.102975    test-error:0.342466
## [142]    train-error:0.105263    test-error:0.342466
## [143]    train-error:0.105263    test-error:0.342466
## [144]    train-error:0.105263    test-error:0.342466
## [145]    train-error:0.102975    test-error:0.342466
## [146]    train-error:0.102975    test-error:0.342466
## [147]    train-error:0.105263    test-error:0.342466
## [148]    train-error:0.105263    test-error:0.342466
## [149]    train-error:0.105263    test-error:0.342466
## [150]    train-error:0.105263    test-error:0.342466
## [151]    train-error:0.105263    test-error:0.342466
## [152]    train-error:0.105263    test-error:0.342466
## [153]    train-error:0.109840    test-error:0.342466
## [154]    train-error:0.105263    test-error:0.342466
## [155]    train-error:0.105263    test-error:0.342466
## [156]    train-error:0.109840    test-error:0.342466
## [157]    train-error:0.109840    test-error:0.342466
## [158]    train-error:0.109840    test-error:0.342466
## [159]    train-error:0.109840    test-error:0.342466
## [160]    train-error:0.109840    test-error:0.342466
## [161]    train-error:0.109840    test-error:0.342466
## [162]    train-error:0.109840    test-error:0.342466
## [163]    train-error:0.109840    test-error:0.342466
## [164]    train-error:0.109840    test-error:0.342466
## [165]    train-error:0.109840    test-error:0.342466
## [166]    train-error:0.109840    test-error:0.342466
## [167]    train-error:0.109840    test-error:0.342466
## [168]    train-error:0.109840    test-error:0.342466
## [169]    train-error:0.109840    test-error:0.342466
## [170]    train-error:0.109840    test-error:0.342466
## [171]    train-error:0.109840    test-error:0.342466
## [172]    train-error:0.109840    test-error:0.342466
## [173]    train-error:0.109840    test-error:0.342466
## [174]    train-error:0.109840    test-error:0.342466
## [175]    train-error:0.107551    test-error:0.342466
## [176]    train-error:0.112128    test-error:0.342466
## [177]    train-error:0.112128    test-error:0.342466
## [178]    train-error:0.112128    test-error:0.342466
## [179]    train-error:0.109840    test-error:0.342466
## [180]    train-error:0.109840    test-error:0.342466
```

```
## [181]    train-error:0.109840    test-error:0.342466
## [182]    train-error:0.109840    test-error:0.342466
## [183]    train-error:0.109840    test-error:0.342466
## [184]    train-error:0.109840    test-error:0.342466
## [185]    train-error:0.109840    test-error:0.342466
## [186]    train-error:0.109840    test-error:0.342466
## [187]    train-error:0.109840    test-error:0.342466
## [188]    train-error:0.109840    test-error:0.342466
## [189]    train-error:0.109840    test-error:0.342466
## [190]    train-error:0.112128    test-error:0.335616
## [191]    train-error:0.109840    test-error:0.335616
## [192]    train-error:0.112128    test-error:0.335616
## [193]    train-error:0.112128    test-error:0.335616
## [194]    train-error:0.112128    test-error:0.335616
## [195]    train-error:0.112128    test-error:0.335616
## [196]    train-error:0.114416    test-error:0.335616
## [197]    train-error:0.112128    test-error:0.335616
## [198]    train-error:0.112128    test-error:0.335616
## [199]    train-error:0.112128    test-error:0.335616
## [200]    train-error:0.112128    test-error:0.335616
## [201]    train-error:0.112128    test-error:0.335616
## [202]    train-error:0.114416    test-error:0.335616
## [203]    train-error:0.112128    test-error:0.335616
## [204]    train-error:0.112128    test-error:0.328767
## [205]    train-error:0.114416    test-error:0.328767
## [206]    train-error:0.114416    test-error:0.328767
## [207]    train-error:0.114416    test-error:0.328767
## [208]    train-error:0.114416    test-error:0.328767
## [209]    train-error:0.114416    test-error:0.328767
## [210]    train-error:0.114416    test-error:0.328767
## [211]    train-error:0.114416    test-error:0.328767
## [212]    train-error:0.114416    test-error:0.328767
## [213]    train-error:0.114416    test-error:0.328767
## [214]    train-error:0.114416    test-error:0.328767
## [215]    train-error:0.112128    test-error:0.328767
## [216]    train-error:0.112128    test-error:0.328767
## [217]    train-error:0.112128    test-error:0.328767
## [218]    train-error:0.112128    test-error:0.328767
## [219]    train-error:0.112128    test-error:0.328767
## [220]    train-error:0.112128    test-error:0.328767
## [221]    train-error:0.112128    test-error:0.328767
## [222]    train-error:0.112128    test-error:0.328767
## [223]    train-error:0.112128    test-error:0.328767
## [224]    train-error:0.112128    test-error:0.328767
## [225]    train-error:0.112128    test-error:0.328767
## [226]    train-error:0.112128    test-error:0.328767
## [227]    train-error:0.112128    test-error:0.328767
## [228]    train-error:0.112128    test-error:0.328767
## [229]    train-error:0.112128    test-error:0.328767
## [230]    train-error:0.112128    test-error:0.328767
## [231]    train-error:0.109840    test-error:0.328767
## [232]    train-error:0.109840    test-error:0.328767
## [233]    train-error:0.109840    test-error:0.328767
## [234]    train-error:0.109840    test-error:0.328767
```

```
## [235]    train-error:0.109840    test-error:0.328767
## [236]    train-error:0.109840    test-error:0.328767
## [237]    train-error:0.107551    test-error:0.328767
## [238]    train-error:0.107551    test-error:0.328767
## [239]    train-error:0.107551    test-error:0.328767
## [240]    train-error:0.107551    test-error:0.328767
## [241]    train-error:0.107551    test-error:0.328767
## [242]    train-error:0.107551    test-error:0.328767
## [243]    train-error:0.105263    test-error:0.328767
## [244]    train-error:0.105263    test-error:0.328767
## [245]    train-error:0.105263    test-error:0.328767
## [246]    train-error:0.105263    test-error:0.328767
## [247]    train-error:0.105263    test-error:0.328767
## [248]    train-error:0.102975    test-error:0.328767
## [249]    train-error:0.100686    test-error:0.328767
## [250]    train-error:0.100686    test-error:0.328767
## [251]    train-error:0.100686    test-error:0.328767
## [252]    train-error:0.100686    test-error:0.328767
## [253]    train-error:0.100686    test-error:0.328767
## [254]    train-error:0.098398    test-error:0.328767
## [255]    train-error:0.098398    test-error:0.328767
## [256]    train-error:0.098398    test-error:0.328767
## [257]    train-error:0.096110    test-error:0.328767
## [258]    train-error:0.096110    test-error:0.328767
## [259]    train-error:0.096110    test-error:0.328767
## [260]    train-error:0.096110    test-error:0.328767
## [261]    train-error:0.096110    test-error:0.328767
## [262]    train-error:0.096110    test-error:0.328767
## [263]    train-error:0.096110    test-error:0.328767
## [264]    train-error:0.096110    test-error:0.328767
## [265]    train-error:0.096110    test-error:0.328767
## [266]    train-error:0.096110    test-error:0.328767
## [267]    train-error:0.096110    test-error:0.328767
## [268]    train-error:0.096110    test-error:0.328767
## [269]    train-error:0.096110    test-error:0.328767
## [270]    train-error:0.096110    test-error:0.328767
## [271]    train-error:0.096110    test-error:0.328767
## [272]    train-error:0.093822    test-error:0.335616
## [273]    train-error:0.093822    test-error:0.335616
## [274]    train-error:0.093822    test-error:0.335616
## [275]    train-error:0.093822    test-error:0.335616
## [276]    train-error:0.093822    test-error:0.335616
## [277]    train-error:0.093822    test-error:0.335616
## [278]    train-error:0.091533    test-error:0.335616
## [279]    train-error:0.091533    test-error:0.335616
## [280]    train-error:0.091533    test-error:0.335616
## [281]    train-error:0.091533    test-error:0.335616
## [282]    train-error:0.091533    test-error:0.335616
## [283]    train-error:0.091533    test-error:0.335616
## [284]    train-error:0.091533    test-error:0.335616
## [285]    train-error:0.091533    test-error:0.328767
## [286]    train-error:0.091533    test-error:0.328767
## [287]    train-error:0.091533    test-error:0.328767
## [288]    train-error:0.091533    test-error:0.328767
```

```
## [289]    train-error:0.091533    test-error:0.328767
## [290]    train-error:0.091533    test-error:0.328767
## [291]    train-error:0.091533    test-error:0.328767
## [292]    train-error:0.091533    test-error:0.328767
## [293]    train-error:0.091533    test-error:0.328767
## [294]    train-error:0.091533    test-error:0.328767
## [295]    train-error:0.091533    test-error:0.328767
## [296]    train-error:0.091533    test-error:0.328767
## [297]    train-error:0.089245    test-error:0.328767
## [298]    train-error:0.089245    test-error:0.328767
## [299]    train-error:0.089245    test-error:0.328767
## [300]    train-error:0.089245    test-error:0.328767
## [301]    train-error:0.089245    test-error:0.328767
## [302]    train-error:0.089245    test-error:0.328767
## [303]    train-error:0.089245    test-error:0.328767
## Stopping. Best iteration:
## [3]  train-error:0.116705    test-error:0.301370
```

Predictions

We remove labels from the full data set and use the model we trained to predict the labels, or in other words, predict if the patients have liver disease or not.

```
fulldata <- as(as.matrix(data[ , -which(names(data) %in% c("Dataset"))]), "dgCMatrix")
test_pred <- predict(xgbModel, newdata = fulldata)
```

Results

A confusion matrix is used to analyse the results of XGBoost model.

```
confusionMatrix(as.factor(round(test_pred)), as.factor(data$Dataset))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 388  67
##          1  28 100
##
##                Accuracy : 0.837
##                  95% CI : (0.8045, 0.8661)
##     No Information Rate : 0.7136
##     P-Value [Acc > NIR] : 2.377e-12
##
##                   Kappa : 0.5714
##
##  Mcnemar's Test P-Value : 9.670e-05
##
##             Sensitivity : 0.9327
##             Specificity : 0.5988
##          Pos Pred Value : 0.8527
```

```
##          Neg Pred Value : 0.7812
##              Prevalence : 0.7136
##          Detection Rate : 0.6655
##    Detection Prevalence : 0.7804
##       Balanced Accuracy : 0.7657
##
##        'Positive' Class : 0
##
```

We get an accuracy of 88.51% with sensitivity of 93.27%. Though there are some false positives (where we incorrectly predicted liver disease in a healthy patient), the rest of the values indicate that the model is perforiming well overall, and that if we had more training data (especially for non-liver patients) we could increase the accuracy and reduce the false positives.

Analysis of what feautres were the most important to our model when it made the prediction of whether a patient has liver disease or not.

```
xgb.importance(colnames(fulldata), model = xgbModel)
```

```
##                       Feature       Gain      Cover  Frequency
## 1:       Alkaline_Phosphotase 0.20084921 0.21913435 0.20782938
## 2:           Direct_Bilirubin 0.19534923 0.17166063 0.02862155
## 3:                        Age 0.18406963 0.17454623 0.19453421
## 4: Aspartate_Aminotransferase 0.09612948 0.13403000 0.12279568
## 5:             Total_Protiens 0.08653312 0.03828992 0.13239775
## 6:   Alamine_Aminotransferase 0.08291753 0.10564068 0.09371249
## 7:                    Albumin 0.06174611 0.06242364 0.09168129
## 8: Albumin_and_Globulin_Ratio 0.05289526 0.05439954 0.08152525
## 9:                     Gender 0.03951043 0.03987501 0.04690241
```

Conclusion

We have trained the model to diagnose whether a patient has liver disease or not based on a set of available data points. Our model achieved an accuracy of 83.7% which is pretty good.

The dataset we used was not much big, with more data we can have more accurate predictions.

Acknowledgements

This dataset was uploades originally on the UCI ML Repository and is downloaded from https://www.kaggle.com/uciml/indian-liver-patient-records.

We have used XGBoost library for training the prediction model. Official documentation can be found here: https://github.com/dmlc/xgboost

Also an introductory blog on XGBoost algorithm: https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-a