Liver Diease Predictor

Kumar

08/04/2020

1. Overview

1.1.Introduction

This project is a requirement of Data Science Certification program offered by Harvard on Edx platform. The goal of the project is to apply the knowledge gained in machine learning techniques from the courses in the program. The the topic for the project can be chosen by learners as per their own choice.Among others, the program recommends Kaggle as a good place to look for datasets.

1.2.Liver Disease Predictor

Upon a reviw of datasets available on Kaggle.Com, the topic "Indian Liver Patient Records" was chosen.The data is an extract from Lichman's (2013) work at the University of California, Irvine. The dataset contains details such as age, sex and liver enzyme & protein levels and teh presence or absence of liver disease in 583 patients in Andhra Pradesh state of India.

There are two main reasons for choosing this topic. The first reason is that given the fast paced and rather unhealthy modern lifestyle, liver disease is common ailment which afflicts millions across the globe and yet due to the seemingly innocuous symptoms, this condition goes undetected in patients for prologed periods of time, ultimately causing serious physiological damages and even fatalities.Secondly, India, in particular, is an interesting study in the prevalence of liver disease among the population because of the rather renowned and celebrated culture of intake of delicious and yet fat-rich and spicy foods, as part of regular diet.

2. Data Set

The data set, "indian_liver_patient.csv",has been downloaded to the local drive from the site URL: "https://www.kaggle.com/uciml/indian-liver-patient-records".

The .csv is imported into R Studio and an initial examination is done as follows:

```
indian_liver_patient <- read.csv("~/course_tests/Liver Disease
Predictor/indian_liver_patient.csv")
head(indian_liver_patient)

##   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
```

The data set is available in ready to use tidy format and is only missing 4 data points under the column "Albumin_and_Globulin_Ratio". These have been manually set to the average value of 0.947 of the remaining data in teh column.

Before examining the above data set any further, the below libraries are included to enable the use of common functions for anlyses.

```
if(!require(tidyverse)) install.packages("tidyverse", repos =
"http://cran.us.r-project.org")

## Loading required package: tidyverse

## Warning: package 'tidyverse' was built under R version 3.6.1

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

## -- Attaching packages ---------------------------------------------------
------------------------------------------------------ tidyverse 1.2.1 --

## v ggplot2 3.1.1     v purrr   0.3.2
## v tibble  2.1.1     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## Warning: package 'tidyr' was built under R version 3.6.1

## Warning: package 'readr' was built under R version 3.6.1

## Warning: package 'purrr' was built under R version 3.6.1
```

```
## Warning: package 'dplyr' was built under R version 3.6.1

## Warning: package 'forcats' was built under R version 3.6.1

## -- Conflicts ---------------------------------------------------------
------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-
project.org")

## Loading required package: caret

## Warning: package 'caret' was built under R version 3.6.1

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

The dimensions of the data set can be studied as follows:

```
nrow(indian_liver_patient)

## [1] 583

ncol(indian_liver_patient)

## [1] 11
```

The data contains details details of 10 variables and liver disease diagnosis for 583 patients.Out of these 10 vaiables, 7 represent liver enzymes & proteins concentrations, whereas the remaining 3 represent patient's Albumin to Globulin ratio, age and sex. The diagnosis (disease or non disease) is reprsented by a factor variable named "Dataset", which takes value '1' if disease is present and '2', otherwise.

The goal of the model is to split the data set into test and training sets, train a suitable model using training set and finally predict the dignosis outcomes on the test set using the model while ensuring that the predoctions are as close to the actual values of "Dataset" in the test set.

3. Method

Before fitting the regression model, it is helpful to study some of the features of the data set, such as the prevalence of male gender vs female, patients' age groups and live emzymes' concentrations and their correlation with the presence of liver disease.

3.1.Gender

The below code summarizes gender ratio among the patients in the data set.Colum “ratio” below represents the ratio of each gender in the data set.

```
indian_liver_patient%>%group_by(Gender)%>%summarize(ratio= n()/nrow(.))
```

```
## # A tibble: 2 x 2
##   Gender ratio
##   <fct>  <dbl>
## 1 Female 0.244
## 2 Male   0.756
```

The data set comprises of 76% male population and 24% female. Now the below code provides gender based analysis on the occurrence of disease among the two genders.

```
indian_liver_patient%>%group_by(Gender)%>%summarize(occurrence =
mean(Dataset==1))
```

```
## # A tibble: 2 x 2
##   Gender occurrence
##   <fct>       <dbl>
## 1 Female      0.648
## 2 Male        0.735
```

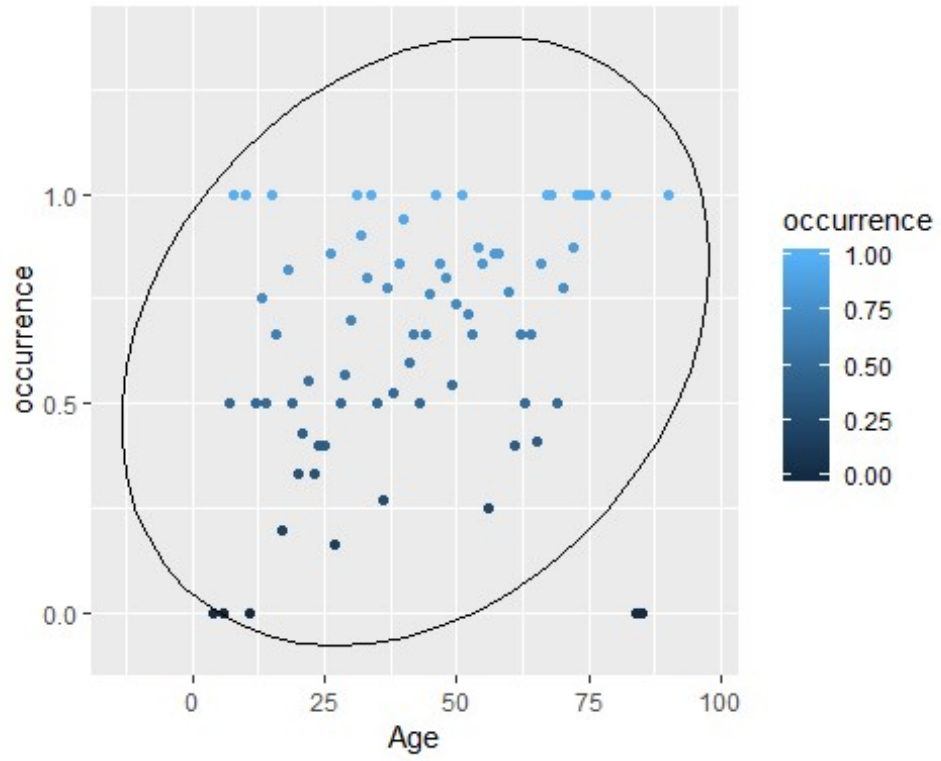
So, it is seen that female patients are only slightly less likely than a male patient to have liver disease.

3.2.Age

Age is another important variable which could possibly have an impact on the occurrence of disease in a patient.

The impact of age on the occurrence of disease is studied via the below scatter plot:

```
indian_liver_patient%>%group_by(Age)%>%summarize(occurrence =
mean(Dataset==1))%>%ggplot(aes(Age,occurrence,
col=occurrence))+geom_point()+stat_ellipse(type="norm")
```



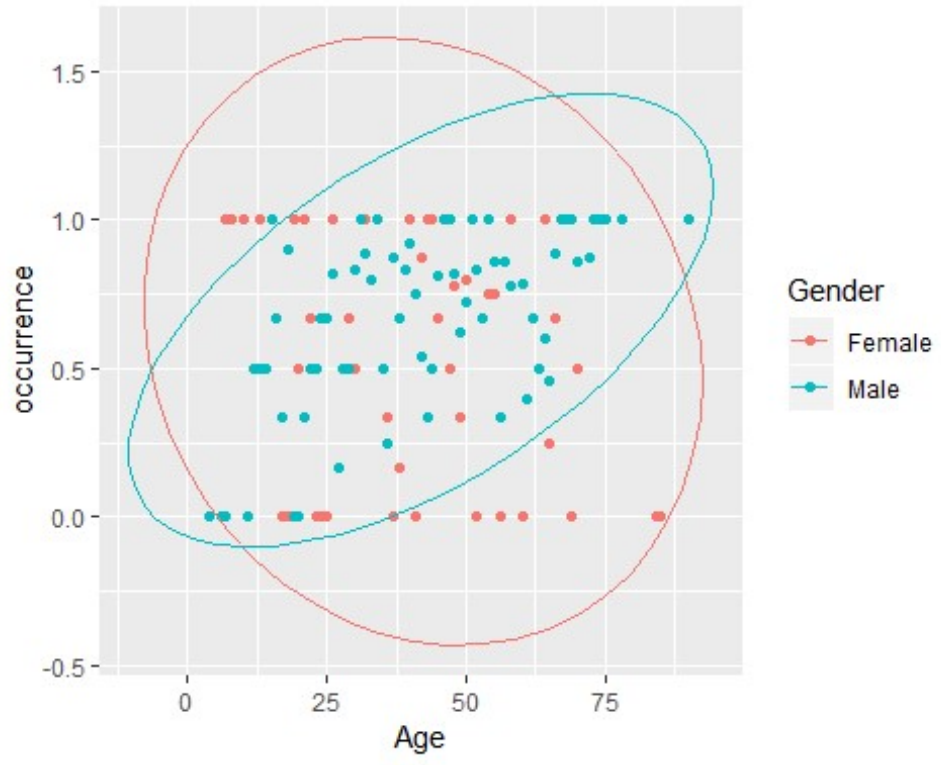
The correlation between Age and occurrence of disease is calculated as follows:

```
indian_liver_patient%>%group_by(Age)%>%summarize(occurrence =
mean(Dataset==1))%>%summarize(r=cor(Age, occurrence))
```

```
## # A tibble: 1 x 1
##       r
##   <dbl>
## 1 0.267
```

A further study of correlation between age and occurrence of liver disease is done by looking at female and male patients separately.

```
indian_liver_patient%>%group_by(Gender,Age)%>%summarize(occurrence=mean(Datas
et==1))%>%ggplot(aes(Age,occurrence,group=Gender,
col=Gender))+geom_point()+stat_ellipse(type="norm")
```



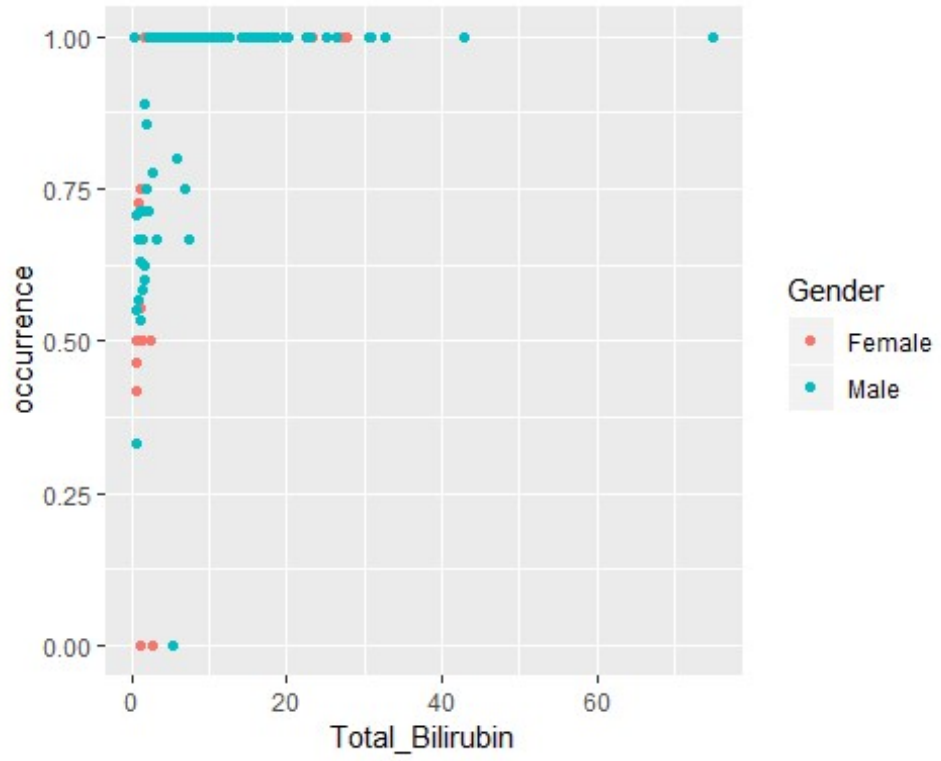
Female gender seems to have a negative correlation of age w.r.t occurrence of liver disease whereas men have a positive correlation between age and liver disease.

3.3.Liver Enzymes

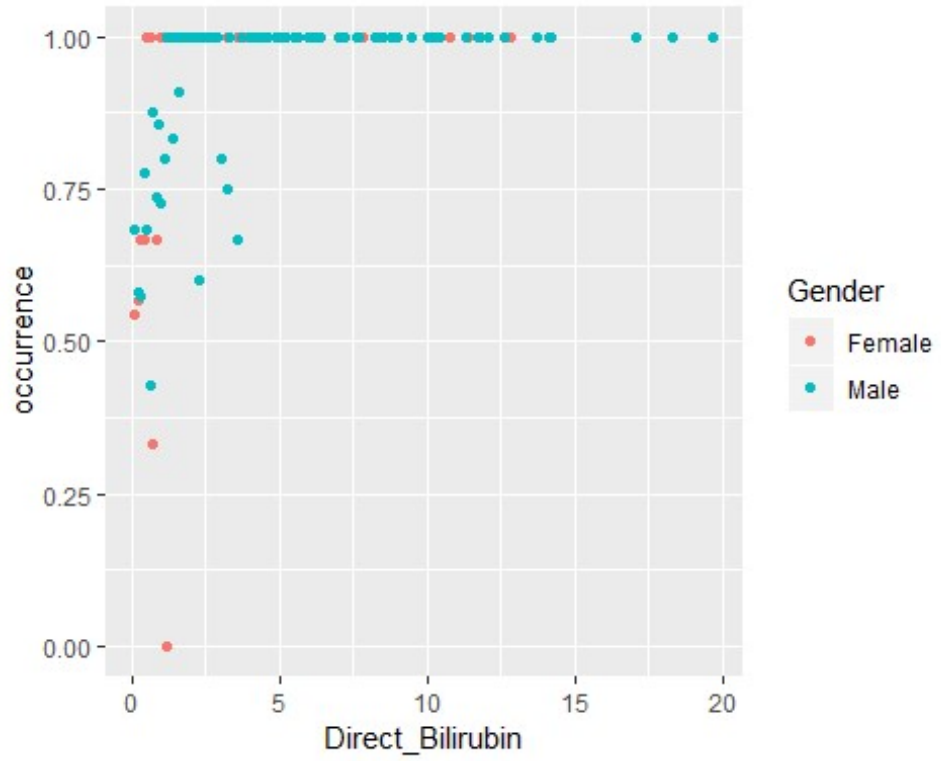
There are seven liver enzyme concentration variables and one enzymes' ratio variable.

The correlation of various enzymes with the occurrence of disease can be visually studied as follows:

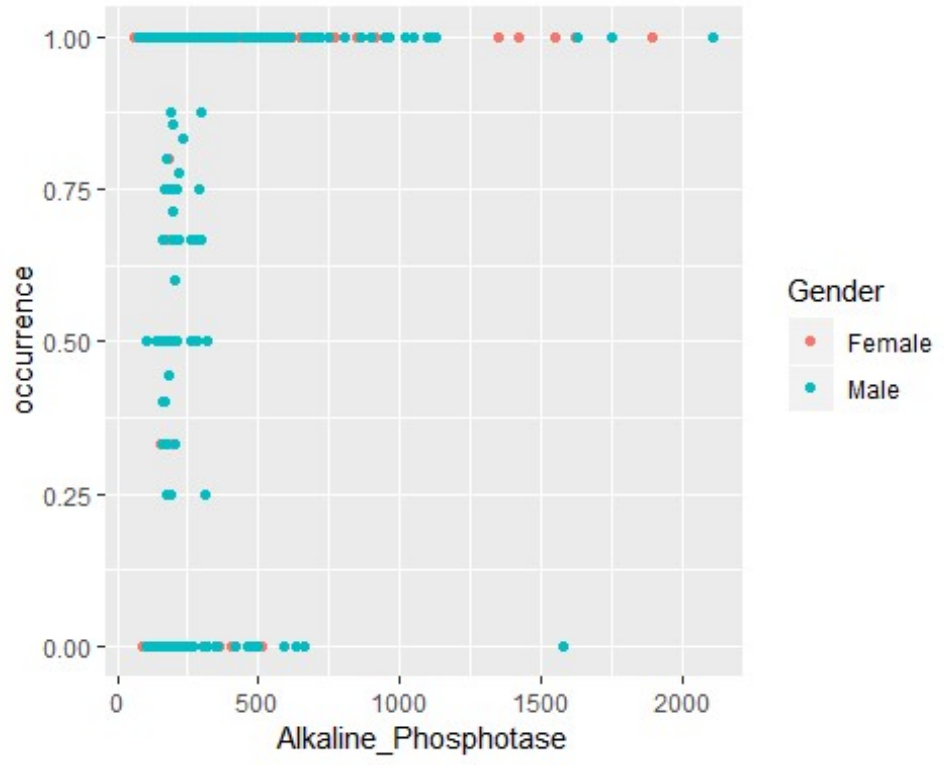
```
indian_liver_patient%>%group_by(Gender,Total_Bilirubin)%>%summarize(occurrenc
e=mean(Dataset==1))%>%ggplot(aes(Total_Bilirubin,occurrence,col=Gender))+geom
_point()
```



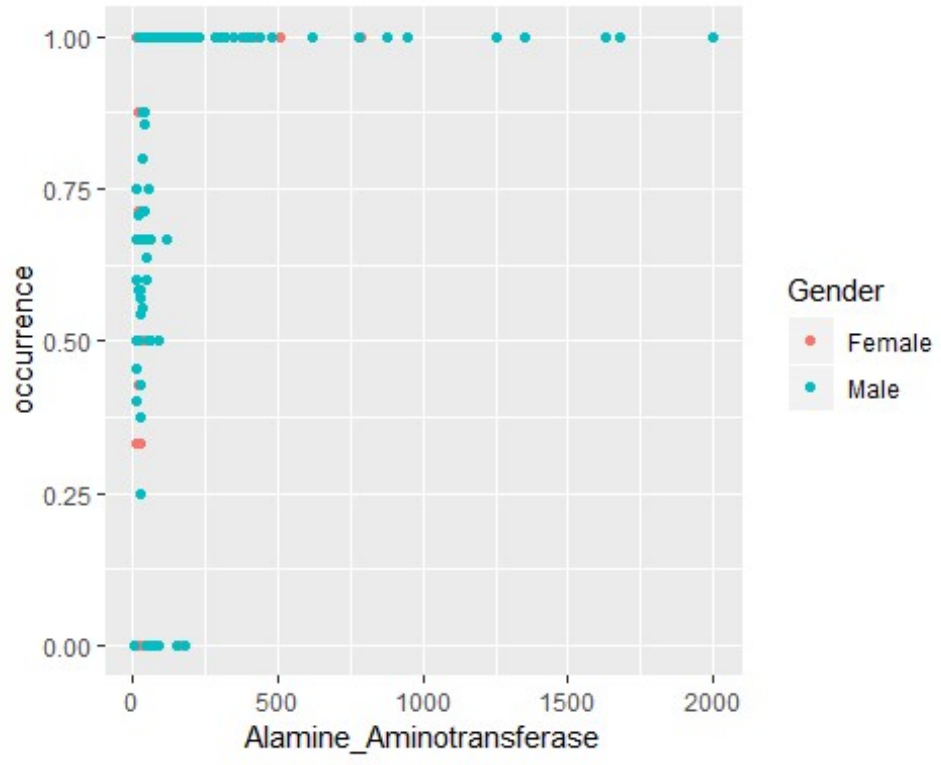

```
indian_liver_patient%>%group_by(Gender,Direct_Bilirubin)%>%summarize(occurren
ce=mean(Dataset==1))%>%ggplot(aes(Direct_Bilirubin,occurrence,col=Gender))+ge
om_point()
```



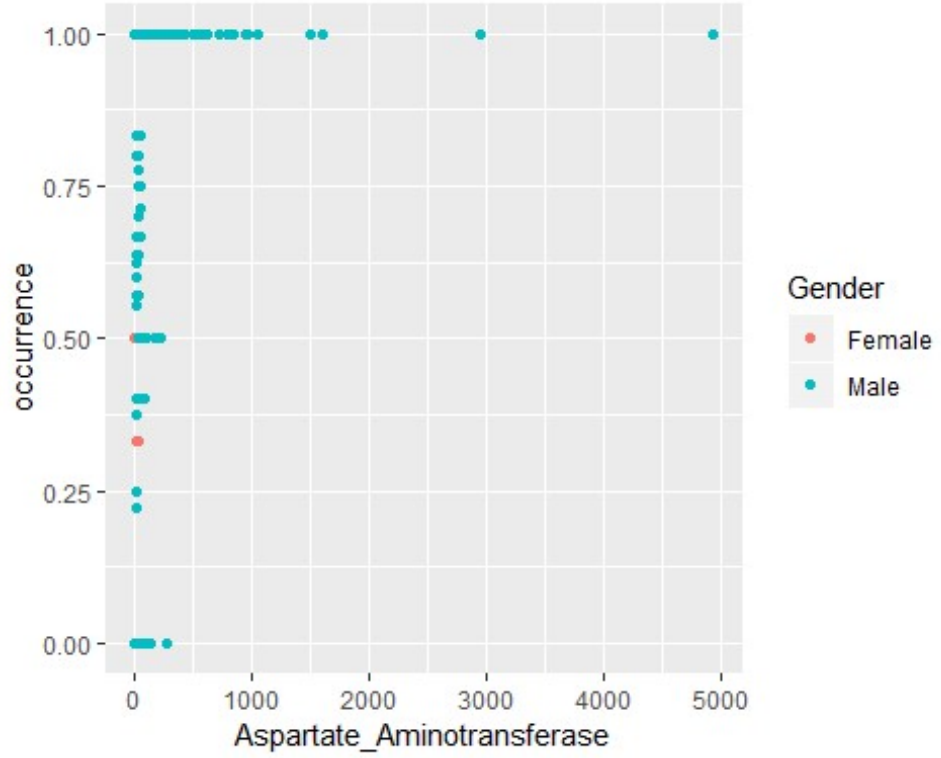
```
indian_liver_patient%>%group_by(Gender,Alkaline_Phosphotase)%>%summarize(occu
rrence=mean(Dataset==1))%>%ggplot(aes(Alkaline_Phosphotase,occurrence,col=Gen
der))+geom_point()
```



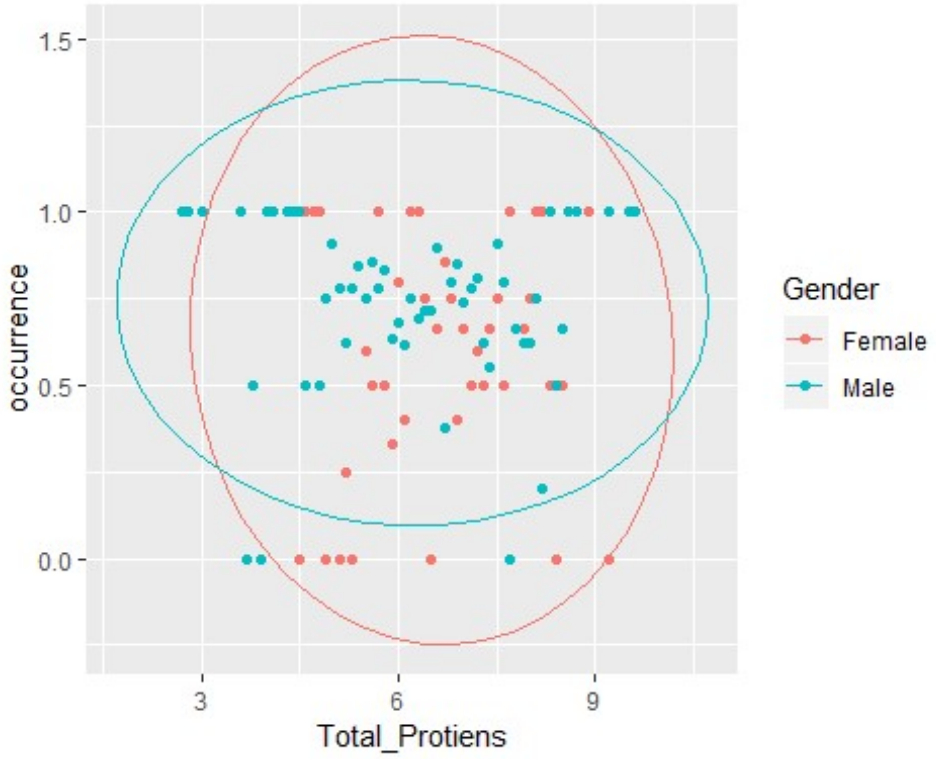

```
indian_liver_patient%>%group_by(Gender,Alamine_Aminotransferase)%>%summarize(
occurrence=mean(Dataset==1))%>%ggplot(aes(Alamine_Aminotransferase,occurrence
,col=Gender))+geom_point()
```



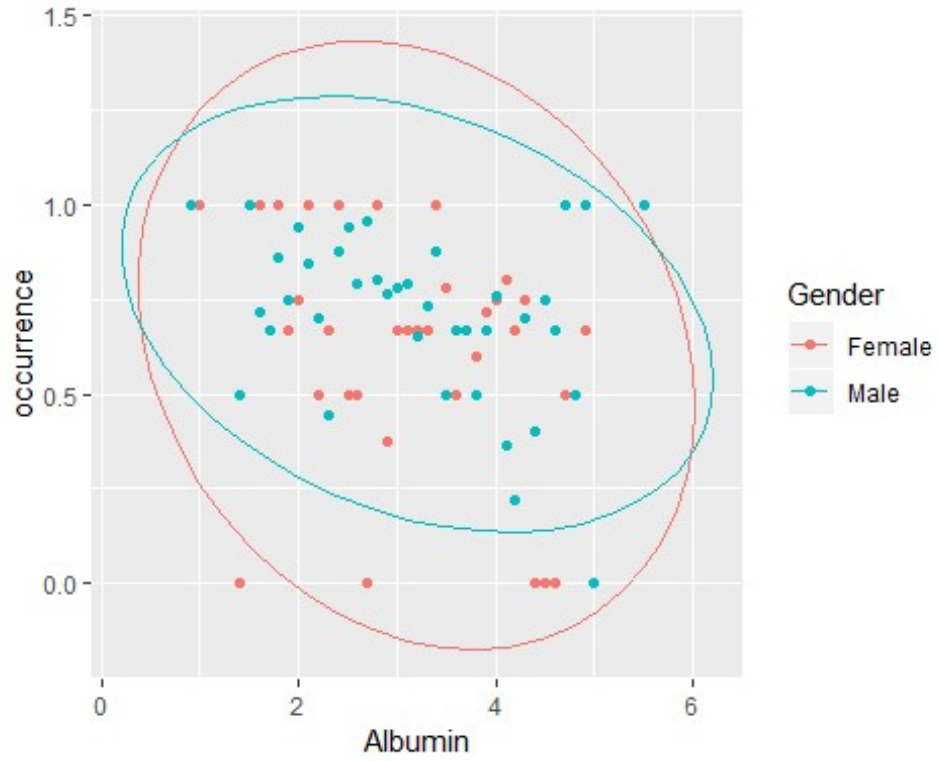

```
indian_liver_patient%>%group_by(Gender,Aspartate_Aminotransferase)%>%summariz
e(occurrence=mean(Dataset==1))%>%ggplot(aes(Aspartate_Aminotransferase,occurr
ence,col=Gender))+geom_point()
```



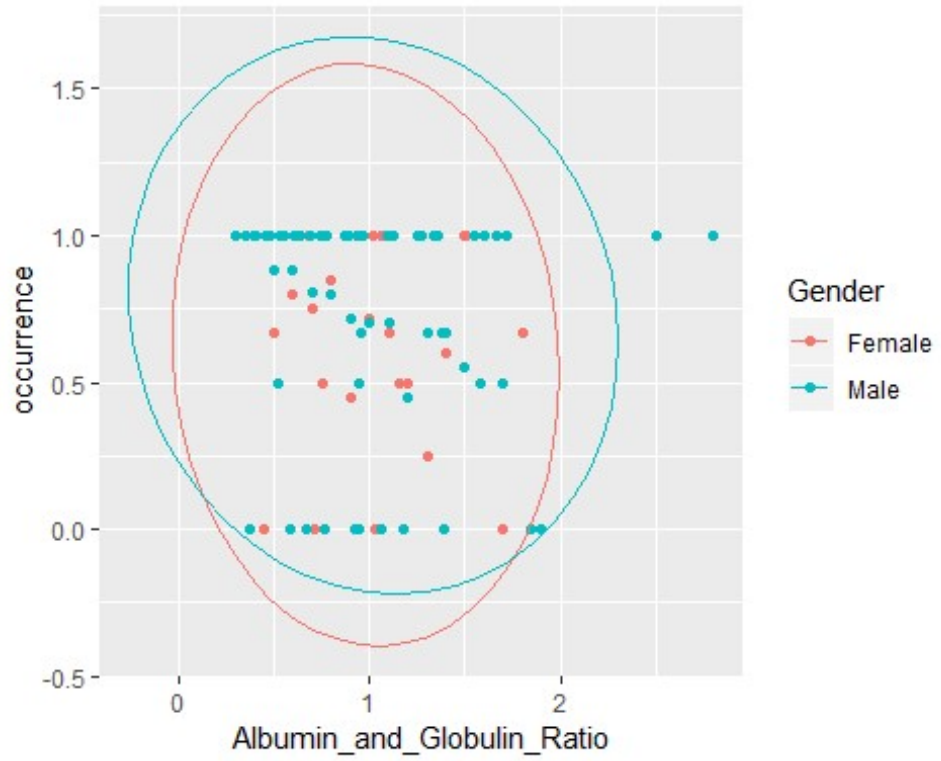
```
indian_liver_patient%>%group_by(Gender,Total_Protiens)%>%summarize(occurrence
=mean(Dataset==1))%>%ggplot(aes(Total_Protiens,occurrence,col=Gender))+geom_p
oint()+stat_ellipse(type="norm")
```




```
indian_liver_patient%>%group_by(Gender,Albumin)%>%summarize(occurrence=mean(D
ataset==1))%>%ggplot(aes(Albumin,occurrence,col=Gender))+geom_point()+stat_el
lipse(type="norm")
```




```
indian_liver_patient%>%group_by(Gender,Albumin_and_Globulin_Ratio)%>%summariz
e(occurrence=mean(Dataset==1))%>%ggplot(aes(Albumin_and_Globulin_Ratio,occurr
ence,col=Gender))+geom_point()+stat_ellipse(type="norm")
```



From the scatter plots, it is clear that, whereas values of Total_Bilirubin (TB), Direct_Bilirubin (DB),Alkaline_Phosphotase(AP), Alamine_Aminotransferase (ALT) and Aspartate_Aminotransferase (AST) show positive correlation with the presence of liver disease, Total_Protiens (TP), Albumin (AB) and Albumin_and_Globulin_Ratio (AGR) show negative correlation with the occurrences of disease in patients.

No gender bias is observed in the occurrence of liver disease among patients. From the scatter plots, it can aso the inferred that whereas enzymes like TB, DB and AP have positive correlation with each other, these have negative correlations with other variables like TP or AB.

The above assertions can be verified as below:

```
indian_liver_patient%>%summarize(r=cor(Total_Bilirubin,Alkaline_Phosphotase))

##           r
## 1 0.2066688

indian_liver_patient%>%summarize(r=cor(Total_Bilirubin,Albumin))

##            r
## 1 -0.2222504
```

4. Data Partition

From the above analyses, it can be safely assumed that the variables presented in the data set do show correlation with the diagnosis of liver disease in patients. So, it should be possible to train and test a machine learning model on the data set and then use the fitted model to predict diagnosis of liver disease among patients in real world situations.

The indian_live_patient dataset is split into training set “train” and testing set “test” as below. With only 583 patients in the dataset, the training set needs to be large enough to allow for sufficient data points for fitting an accurate model. Hence the training set gets 85% data whereas the test set gets 15%.A smaller test set would increase the risk of over or under estimation of model’s accuracy as it may not be a true representative of the real data set.

```
set.seed(1,sample.kind = "Rounding")

## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used

index<-
createDataPartition(indian_liver_patient$Dataset,times=1,p=0.15,list=FALSE)
train<-indian_liver_patient%>%slice(-index)
test<-indian_liver_patient%>%slice(index)
```

5. Models

Several models have been covered as a part of this program. Key among them are summarized below:

GLM

Genarlized Linear model is variant of the linear model, wherein a tranformation function is used for enabling linear modelling of predictors. This overcomes some of the limitations of linear modelling by also constraining the predicted values to practical limits.

kNN

k-Nearest Neighbors model utilizes the concept of bin smoothing for fitting regression model, by assigning to the predicted values, the class which is more abundantly present within the neighborhood defined by the user via the parameter k.

QDA

Quadratic Discriminant Analysis is a naive bias approach of modelling which assumes that Pr(x|Y=1) and Pr(x|y=0) are multivariate normal. However the disadvantage of this method is the need to calulate a large number of parameters - means, standard deviations and correlations, for estimating probability distributions of variables x as the number of predictors increases. Moreover, the assumption of normal districbution of x may not always be true

LDA

Linear Discriminant Analysis reduces the numder of parameters required in QDA approach by assuming same correlation across all classes.

Loess

Local Weighted Estimate improves upon bin smoother method by using user defined windows or spans and modellimg the data points within those spans via straight lines or curves. The degree parameter allows user to either fit straight lines or curves within each span. The higher the degree, the wigglier the curve.

Random Forest

Random Forest approach utilizes the concept of decision tree by averaging several decision trees.It uses bootstrap aggregation or bagging technique wherein it generates many predictors at each step and then averages prediction of several trees.Bootstrap feature ensures that the different trees are selected.

5.1.Fitting the Model

Caret package's train function is used to test all of the models decribed above and to shortlist the best technique(s). Further finetuning of the accuracy is done by tuning the model parameters.

A vector "models" of the above models is created as follows:

```
models<-c("glm", "knn", "qda","lda", "gamLoess", "rf")
```

The below code is run to apply all above models to the liver patients' dataset and store the training outcomes in the list variable "fits".

```
set.seed(1,sample.kind = "Rounding")
fits <- lapply(models, function(model){
    print(model)
    train(factor(Dataset) ~ ., method = model, data = train)
})
```

```
## [1] "glm"
## [1] "knn"
## [1] "qda"
## [1] "lda"
## [1] "gamLoess"

## Loading required package: gam

## Loading required package: splines

## Loading required package: foreach

##
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':
##
##     accumulate, when

## Loaded gam 1.16.1

## [1] "rf"
```

```
names(fits) <- models
```

Now, fitted models are used for predicting occurrence of disease, represented by variable y_hat, as follows:

```
y_hat<-sapply(fits,function(fits){
  predict(fits,test,type = "raw")
})
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.5, degree = 1)"]], z, w, span =
## 0.5, : eval 90

## Warning in gam.lo(data[["lo(Age, span = 0.5, degree = 1)"]], z, w, span =
## 0.5, : upperlimit 85.405

## Warning in gam.lo(data[["lo(Age, span = 0.5, degree = 1)"]], z, w, span =
## 0.5, : extrapolation not allowed with blending

## Warning in gam.lo(data[["lo(Total_Bilirubin, span = 0.5, degree = 1)"]], :
## eval 75
```

```
## Warning in gam.lo(data[["lo(Total_Bilirubin, span = 0.5, degree = 1)"]], :
## upperlimit 43.012

## Warning in gam.lo(data[["lo(Total_Bilirubin, span = 0.5, degree = 1)"]], :
## extrapolation not allowed with blending
```

As expected, y_hat is a matrix with below dimensions as there are 88 patients in test set and there are 6 different models.

```
tibble(nrow(y_hat),ncol(y_hat ))

## # A tibble: 1 x 2
##   `nrow(y_hat)` `ncol(y_hat)`
##           <int>         <int>
## 1            88             6
```

The accuracy of predicted values for each model is calculated and stored in variable "accuracy" as follows:

```
accuracy<-0
for (i in seq(1,6,by=1)){
  accuracy[i]= mean(factor(y_hat[,i])==factor(test$Dataset))
}
print(tibble(models,accuracy))

## # A tibble: 6 x 2
##   models   accuracy
##   <chr>       <dbl>
## 1 glm         0.727
## 2 knn         0.705
## 3 qda         0.568
## 4 lda         0.739
## 5 gamLoess    0.705
## 6 rf          0.705
```

So, lda, glm, rf, knn and gamLoess models yield accuracies above 70%. Out of these, kNN and rf provide the option of tuning or further improving accuracy.

5.2.Tuning of kNN model

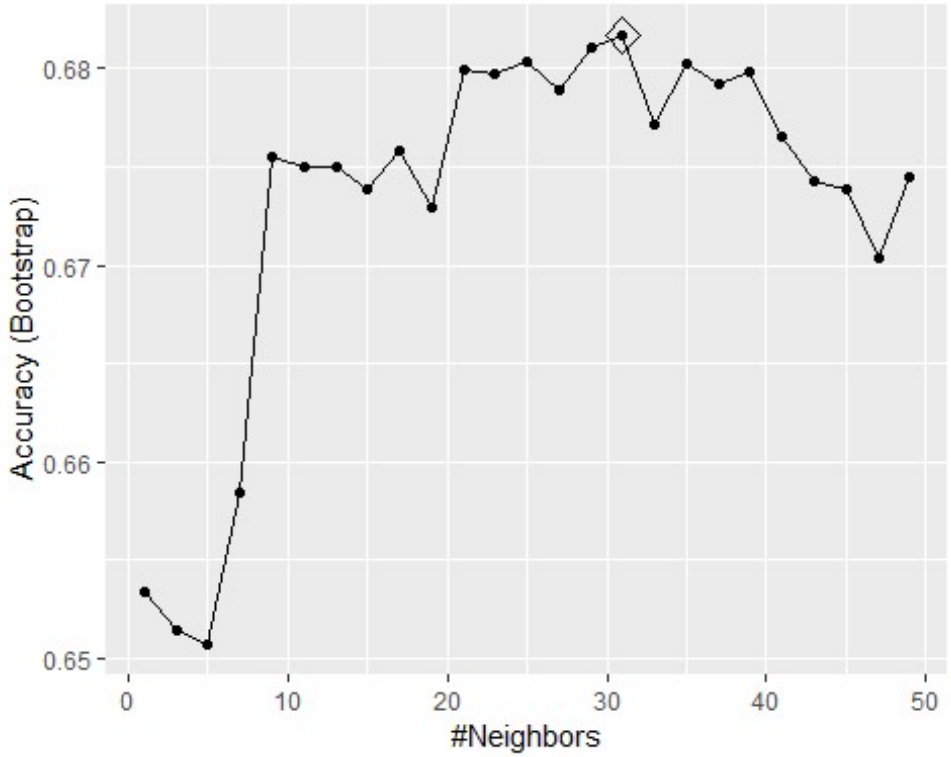
The parameter k in the kNN model represents the number of neighbors to be used for predicting the outcomes. By default, the model uses the tuneGrid parameter k=5,7,9, for fitting the model with the highest accuracy. The search for the value of k which maximizes accuracy can be expanded to a much larger set such as k=seq(1,50,by=2).

A model train_knn is trained on the train set using below code.

```
set.seed(1,sample.kind = "Rounding")

## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

```
train_knn <- train(factor(Dataset) ~ ., method = "knn",
data = train, tuneGrid = data.frame(k = seq(1, 50, by=2)))
ggplot(train_knn, highlight = TRUE)
```



The parameter value which maximises accuracy is obtained by the below code:

```
train_knn$bestTune

##     k
## 16 31
```

So, now the best kNN model "fit_knn" is fitted and is used to predict presence of liver disease on test set. The accuracy on test set is obtained as follows:

```
fit_knn<-knn3(factor(Dataset)~.,data=train,k=train_knn$bestTune)
y_hat_knn<-predict(fit_knn,test,type="class")
accuracy_knn<-mean(factor(y_hat_knn)==factor(test$Dataset))
accuracy_knn

## [1] 0.7386364
```

Using the tuning parameter, the accuracy of kNN model's predictions has been improved to 73.86%, which is the same as that of LDA.

5.3.Tuning of Random Forest model

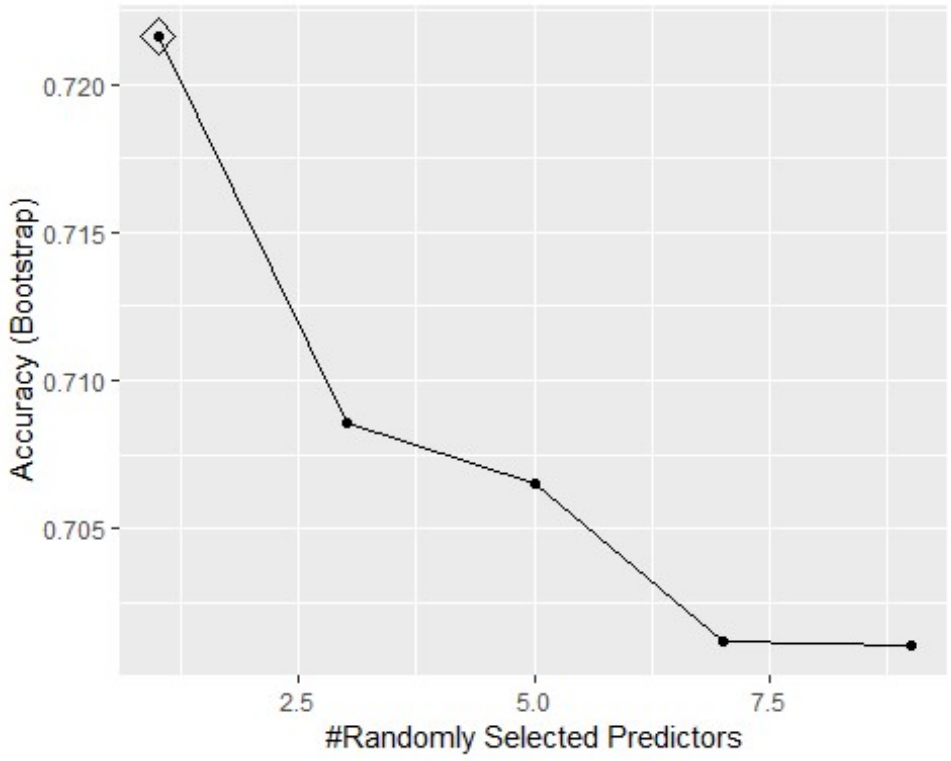
The random forest model provided an accuracy of 70.45% on the test set. The possibility of improving result further via tuning is looked into. The "mtry" parameter allows to try

different numbers of predictor at each split in a tree. The default, in this case, is 10/3 = 3. The following values of mtry are tried, to maximize accuracy.

mtry = seq(1,10, by=2).

As the size of training data set is small, there is no need to change the default 25-fold cross validation. "ntree" is also maintained at default (i.e. 500). As usual, a model train_rf is trained on the train set as below.

```
train_rf<-train(factor(Dataset)~.,method =
"rf",data=train,tuneGrid=(data.frame(mtry=seq(1,10,by=2))))
ggplot(train_rf, highlight = TRUE)
```



The mtry value which maximizes accuracy on training set is:

```
train_rf$bestTune

##   mtry
## 1    1
```

So the use of 1 predictor variable at each split maximizes the accuracy on the training set.

Now the best model"fit_rf" is fitted to the train set and then used for prediction of disease on the test set as follows:

```
fit_rf<-
train(factor(Dataset)~.,method="rf",data=train,tuneGrid=data.frame(mtry
=train_rf$bestTune$mtry))
y_hat_rf<-predict(fit_rf,test,type="raw")
```

```
accuracy_rf<-mean(y_hat_rf==test$Dataset)
accuracy_rf

## [1] 0.7613636
```

So, tuning can help improve the default accuracy of RandomForest algorithm on the training set by nearly 6% from 70% to 76%. The importance of the different variables on the fitted model can be seen below:

```
varImp(fit_rf)

## rf variable importance
##
##                            Overall
## Alkaline_Phosphotase        100.00
## Alamine_Aminotransferase     97.43
## Aspartate_Aminotransferase   94.16
## Age                          83.46
## Total_Bilirubin              76.30
## Albumin                      68.07
## Total_Protiens               67.76
## Direct_Bilirubin             63.69
## Albumin_and_Globulin_Ratio   57.27
## GenderMale                    0.00
```

The above aligns with the earlier observations that AP, ALT, AST, Age and TB show stronger positive correlation with occurrence of disease, wheras the rest of the predictors, age inlcuded show very feeble correlation with the presence of disease.

6. Results

All machine learning algorithms applied on “indian liver patient” data set, aprt from QDA, provide greater than 70% accuracy with the default model settings in Caret package. LDA and GLM yield higher accuracies than the rest, but still below 75%. Upon tuning the parameters of kNN, the accuracy on test set becomes better than that yielded by GLM (72.73%) and equals the accuracy of LDA (73.86%).However, with tuning, the most drastic improvement in accuracy is achieved in Random Forest algorithm which uses only one predictor variable at each split. In fact, the accuracy of the tuned Random Forest model is more than 75%.

7. Conclusions

The outcomes of nearly all machine learning prediction models reaffirm the observations from the scatter plots which show correlation between occurrence of liver disease in patients and various factors such as age and live emzymes & proteins.One of the major challenges faced while modelling is the relatively small size of data set, which necessitates judicious selection of train and test set sizes. Even though ample data points are needed for training an accurate model, very small test set size could also grossly over or under state overall test accuracy.A key effect of the relatively small training set size is seen in QDA model, which performs significantly worse than LDA in terms of overall accuracy on the

test set.On the other hand, the small size of training set helped save some time in fitting the models without reducing the default cross validation settings or the default ntree parameter (in Random Forest model). Overall, Random Forest model stands out in terms of the improvement in accuracy which can be brought about by finetuning parameters. A key limitation of the model is that it is not trained on behavioural patterns of patients, such as alcohol intake, smoking habits, eating habits and physical exercise routines. Even though these factors will likely be correlated with the levels of liver enzymes and proteins, their inclusion should help in impoving the predictive power of these models.