proteinfunctionclass.R

Alfredo Baione mat. s279328

2023-01-20

```
rm(list=ls())


library(purr)
library(readxl)
library(ggplot2)
library(caret)

setwd("C:\\Users\\user\\Desktop\\polito\\Bioquants\\materiale\\Benso\\Assignments\
\Assignment")
df=read_excel("protein_data.xlsx")


#Preprocessing

df2=df;
df2=unique(df2)
df2 <- df2[rowSums(is.na(df2)) == 6, ]
df2 %>% keep(~all(is.na(.x))) %>% names

## [1] "Absorption"

df2<-df2[,-4]

names(df2) <- make.names(names(df2), unique=TRUE)

df2$Catalytic.activity= as.numeric(ifelse(is.na(df2$Catalytic.activity),0,1))
df2$Cofactor= as.numeric(ifelse(is.na(df2$Cofactor),0,1))
df2$DNA.binding=as.numeric(ifelse(is.na(df2$DNA.binding),0,1))
df2$Activity.regulation=as.numeric(ifelse(is.na(df2$Activity.regulation),0,1))
df2$Kinetics=as.numeric(ifelse(is.na(df2$Kinetics),0,1))
df2$Pathway=as.numeric(ifelse(is.na(df2$Pathway),0,1))

rowSums(df2[4:9])

##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

colSums(df2[4:9])
```

```
##  Catalytic.activity            Cofactor         DNA.binding Activity.regulation
##                  55                  11                  12                   9
##            Kinetics             Pathway
##                   2                  38

data.frame(colnames(df2))

##         colnames.df2.
## 1               Entry
## 2       Protein.names
## 3            Sequence
## 4  Catalytic.activity
## 5            Cofactor
## 6         DNA.binding
## 7 Activity.regulation
## 8            Kinetics
## 9             Pathway

df2<-df2[!(df2$Kinetics==1),]
df2=df2[,-8]


df2$Function=NA
df2 <-
df2[order(df2$Activity.regulation,df2$Catalytic.activity,df2$Cofactor,df2$DNA.bind
ing,df2$Pathway,decreasing=TRUE),]
df2$Function[1:9]='Activity regulation'
df2$Function[10:64]='Catalytic activity'
df2$Function[65:75]='Cofactor'
df2$Function[76:87]='DNA binding'
df2$Function[88:125]='Pathway'

df2$Activity.regulation<-as.factor(df2$Activity.regulation)
df2$Catalytic.activity<-as.factor(df2$Catalytic.activity)
df2$Cofactor<-as.factor(df2$Cofactor)
df2$DNA.binding<-as.factor(df2$DNA.binding)
df2$Pathway<-as.factor(df2$Pathway)
df2$Function<-as.factor(df2$Function)

df2$Sequence <- factor(df2$Sequence)
df2$Sequence <- as.integer(df2$Sequence) - 1

attach(df2)
```

```r
#Processing

set.seed(55555)

ind<-createDataPartition(Function,p=0.7,list=FALSE)

train<-df2[ind,]
test <- df2[-ind,]


fitControl <- trainControl(method ="LOOCV")

model1<-train(Function~Sequence, data=train, method = "nb", trControl=fitControl)
model1
```

```
## Naive Bayes
##
## 90 samples
##  1 predictor
##  5 classes: 'Activity regulation', 'Catalytic activity', 'Cofactor', 'DNA
binding', 'Pathway'
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 89, 89, 89, 89, 89, 89, ...
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.4111111   0.01242236
##    TRUE      0.3777778  -0.07554417
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were fL = 0, usekernel = FALSE and adjust
##  = 1.
```

```r
model2<-train(Function ~ Sequence,  data=train, method =
"C5.0",trControl=fitControl,tuneLength=2)
model2
```

```
## C5.0
##
## 90 samples
##  1 predictor
##  5 classes: 'Activity regulation', 'Catalytic activity', 'Cofactor', 'DNA
binding', 'Pathway'
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
```

```
## Summary of sample sizes: 89, 89, 89, 89, 89, 89, ...
## Resampling results across tuning parameters:
##
##   trials  model  winnow  Accuracy   Kappa
##    1      rules  FALSE   0.4111111  -0.006753905
##    1      rules   TRUE   0.4333333   0.016288041
##    1      tree   FALSE   0.4000000  -0.030971574
##    1      tree    TRUE   0.4333333   0.016288041
##   10      rules  FALSE   0.4111111  -0.006753905
##   10      rules   TRUE   0.4333333   0.016288041
##   10      tree   FALSE   0.4000000  -0.030971574
##   10      tree    TRUE   0.4333333   0.016288041
##
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were trials = 1, model = rules and winnow
##  = TRUE.
```

```
model3<-train(Function ~ Sequence,  data=train, method =
"rf",trControl=fitControl,tuneLength=1)
model3
```

```
## Random Forest
##
## 90 samples
##  1 predictor
##  5 classes: 'Activity regulation', 'Catalytic activity', 'Cofactor', 'DNA
binding', 'Pathway'
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 89, 89, 89, 89, 89, 89, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.2888889  -0.02272727
##
## Tuning parameter 'mtry' was held constant at a value of 1
```

```
model4<-train(Function~ Sequence,  data=train, method =
"nnet",trControl=fitControl)
```

```
model4
```

```
## Neural Network
##
## 90 samples
##  1 predictor
##  5 classes: 'Activity regulation', 'Catalytic activity', 'Cofactor', 'DNA
binding', 'Pathway'
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
```

```
## Summary of sample sizes: 89, 89, 89, 89, 89, 89, ...
## Resampling results across tuning parameters:
##
##   size  decay  Accuracy   Kappa
##   1     0e+00  0.3666667  -0.076600210
##   1     1e-04  0.3888889  -0.048284625
##   1     1e-01  0.4000000  -0.013767209
##   3     0e+00  0.3888889  -0.004260499
##   3     1e-04  0.4111111   0.025137952
##   3     1e-01  0.4111111   0.016089109
##   5     0e+00  0.4111111   0.017507724
##   5     1e-04  0.3888889   0.019607843
##   5     1e-01  0.4111111   0.008728180
##
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were size = 3 and decay = 0.1.
```

```
nb.pred <- predict(model1,test)
dt.pred <- predict(model2,test)
rf.pred <- predict(model3,test)
nn.pred <- predict(model4,test)
```

```
cm1 <- confusionMatrix(nb.pred, test$Function)
cm2 <- confusionMatrix(dt.pred, test$Function)
cm3 <- confusionMatrix(rf.pred, test$Function)
cm4 <- confusionMatrix(nn.pred, test$Function)
```

```
accuracy1 <- cm1$overall[1]
accuracy2 <- cm2$overall[1]
accuracy3 <- cm3$overall[1]
accuracy4 <- cm4$overall[1]
```

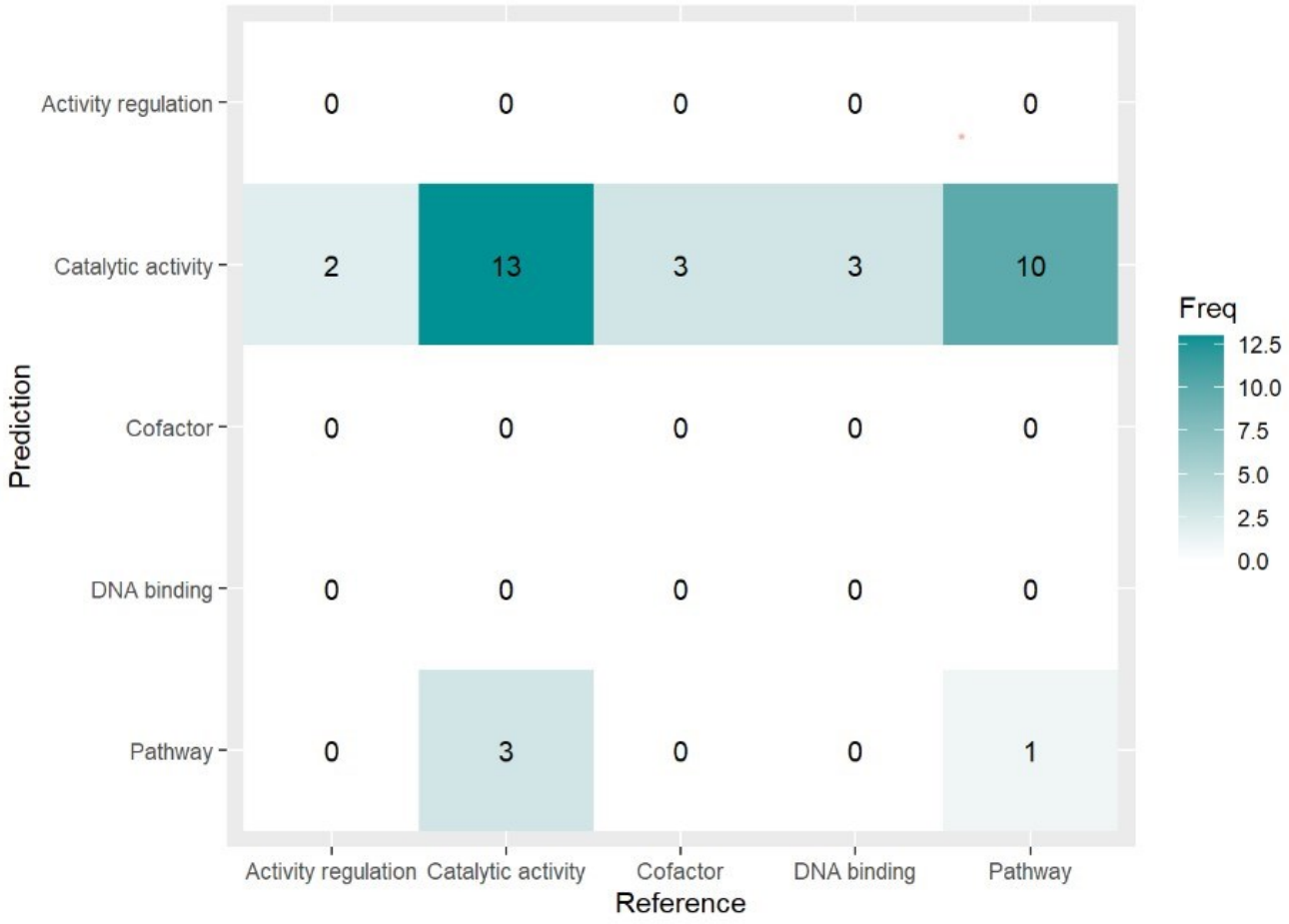
```
recall1<- mean(cm1$byClass[,1])
recall2<- mean(cm2$byClass[,1])
recall3<- mean(cm3$byClass[,1])
recall4<- mean(cm4$byClass[,1])

precision1<-mean(cm1$byClass[,3])
precision2<-mean(cm2$byClass[,3])
precision3<-mean(cm3$byClass[,3])
precision4<-mean(cm4$byClass[,3])
```

```
metricstab<-
matrix(c(accuracy1,accuracy2,accuracy3,accuracy4,recall1,recall2,recall3,recall4,p
recision1,precision2,precision3,precision4),ncol=3)
```

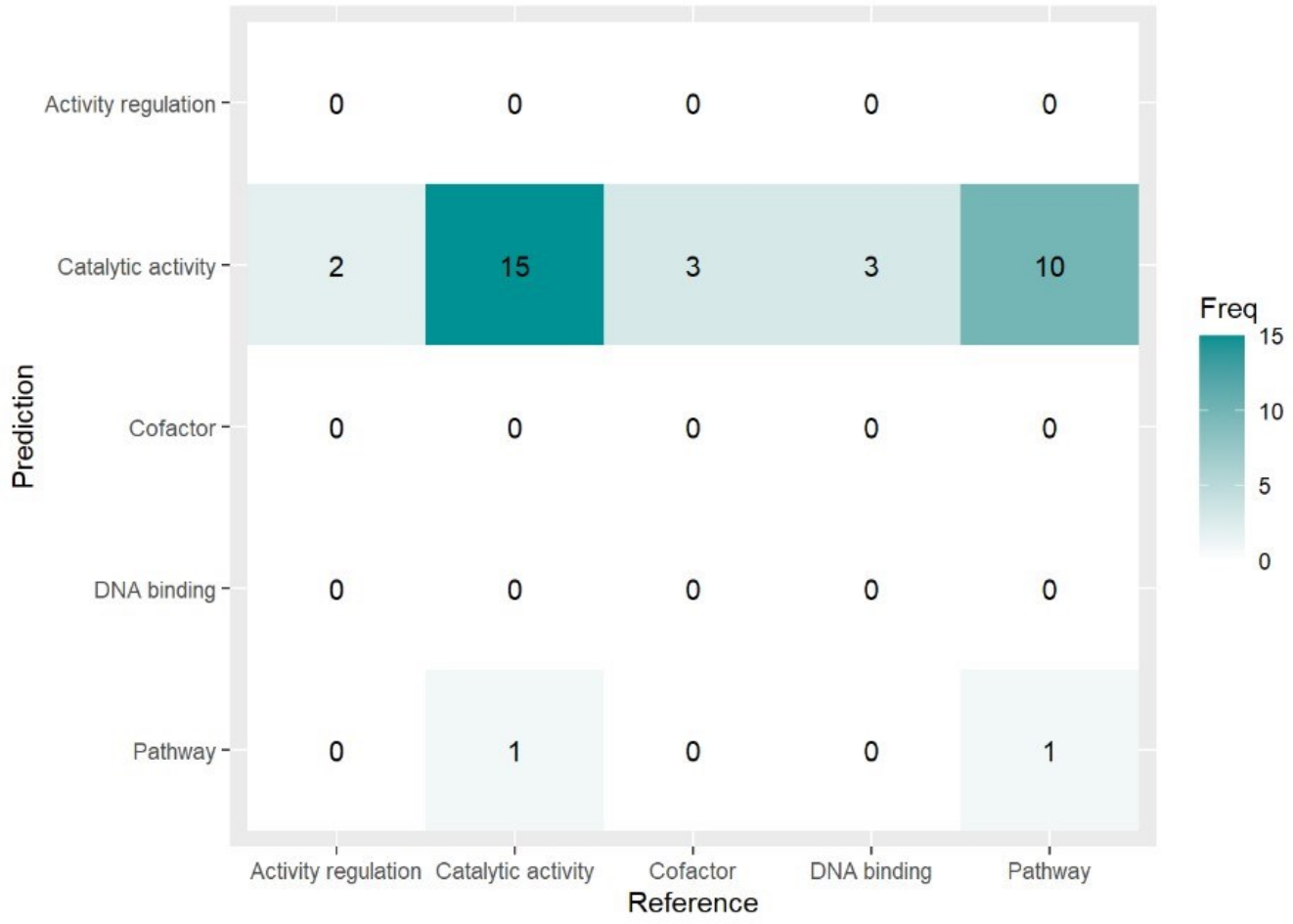
```
colnames(metricstab)<-c('Accuracy','Recall','Precision')
rownames(metricstab)<-c('nb','dt','rf','nn')
metricstab<-as.table(metricstab)
metricstab
```

```
##     Accuracy    Recall Precision
## nb 0.4000000 0.1806818
## dt 0.4571429 0.2056818
## rf 0.5428571 0.3742424 0.4091729
## nn 0.4000000 0.1806818
```

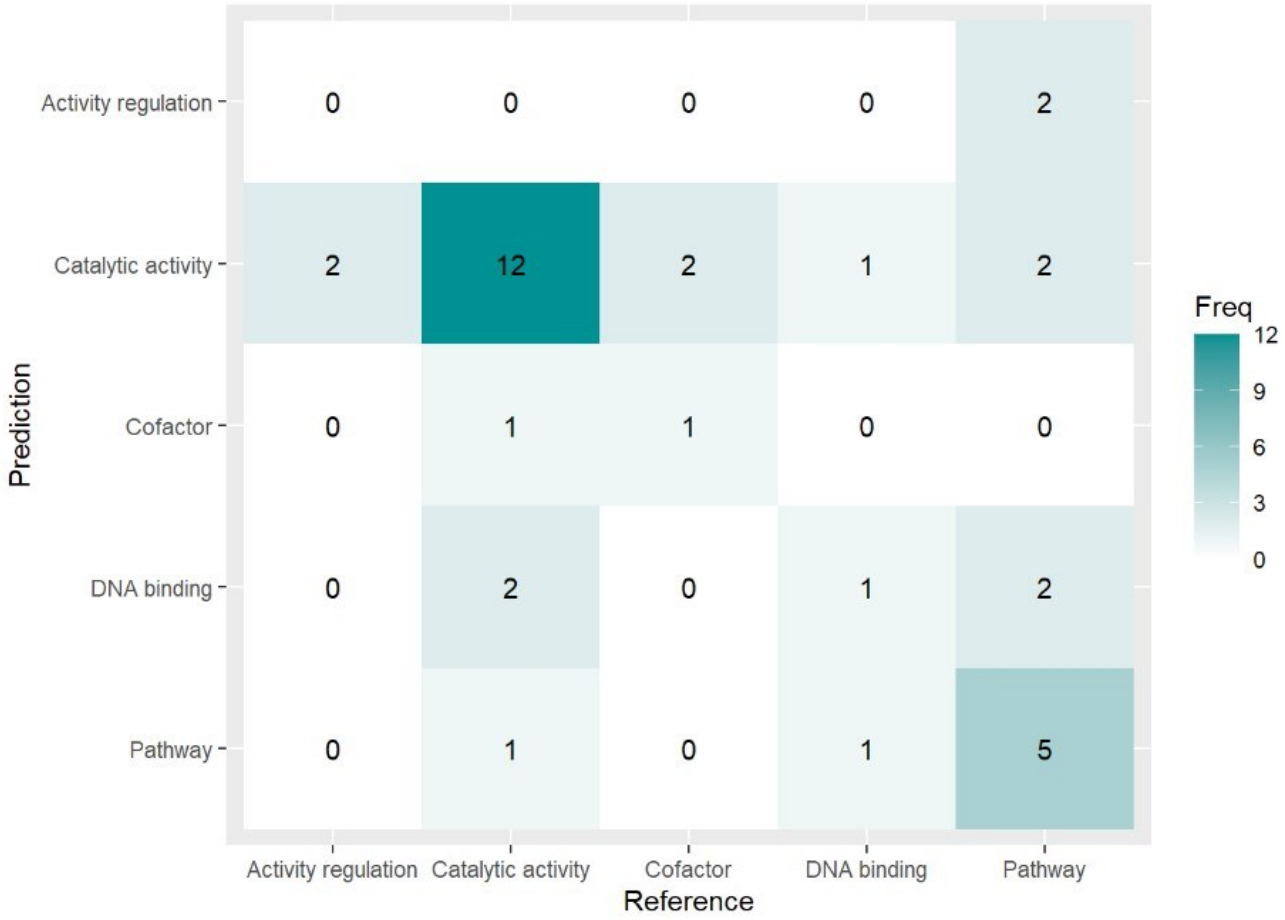
```
plt1<-as.data.frame(cm1$table)
plt1$Prediction <- factor(plt1$Prediction, levels=rev(levels(plt1$Prediction)))
ggplot(plt1, aes(Reference,Prediction,fill=Freq)) +
  geom_tile() + geom_text(aes(label=Freq)) +
  scale_fill_gradient(low="white", high="#009194") +
  labs(x = "Reference",y = "Prediction") +
  scale_x_discrete(labels=c("Activity regulation","Catalytic
activity","Cofactor","DNA binding","Pathway")) +
  scale_y_discrete(labels=c("Pathway","DNA binding","Cofactor","Catalytic
activity","Activity regulation"))
```



```
plt2<-as.data.frame(cm2$table)
plt2$Prediction <- factor(plt2$Prediction, levels=rev(levels(plt2$Prediction)))
ggplot(plt2, aes(Reference,Prediction, fill= Freq)) +
  geom_tile() + geom_text(aes(label=Freq)) +
  scale_fill_gradient(low="white", high="#009194") +
  labs(x = "Reference",y = "Prediction") +
  scale_x_discrete(labels=c("Activity regulation","Catalytic
activity","Cofactor","DNA binding","Pathway")) +
  scale_y_discrete(labels=c("Pathway","DNA binding","Cofactor","Catalytic
activity","Activity regulation"))
```




```
plt3<-as.data.frame(cm3$table)
plt3$Prediction <- factor(plt3$Prediction, levels=rev(levels(plt3$Prediction)))
ggplot(plt3, aes(Reference,Prediction, fill= Freq)) +
  geom_tile() + geom_text(aes(label=Freq)) +
  scale_fill_gradient(low="white", high="#009194") +
  labs(x = "Reference",y = "Prediction") +
  scale_x_discrete(labels=c("Activity regulation","Catalytic
activity","Cofactor","DNA binding","Pathway")) +
  scale_y_discrete(labels=c("Pathway","DNA binding","Cofactor","Catalytic
activity","Activity regulation"))
```




```
plt4<-as.data.frame(cm4$table)
plt4$Prediction <- factor(plt4$Prediction, levels=rev(levels(plt4$Prediction)))
ggplot(plt4, aes(Reference,Prediction, fill= Freq)) +
  geom_tile() + geom_text(aes(label=Freq)) +
  scale_fill_gradient(low="white", high="#009194") +
  labs(x = "Reference",y = "Prediction") +
  scale_x_discrete(labels=c("Activity regulation","Catalytic
activity","Cofactor","DNA binding","Pathway")) +
  scale_y_discrete(labels=c("Pathway","DNA binding","Cofactor","Catalytic
activity","Activity regulation"))
```


Activity regulation
Catalytic activity
Cofactor
DNA binding
Pathway
Prediction
0 0 0 0 0
2 13 2 3 10
0 0 0 0 0
0 0 0 0 0
0 3 1 0 1
Activity regulation
Catalytic activity
Cofactor
DNA binding
Pathway
Reference
Freq
12.5
10.0
7.5
5.0
2.5
0.0