

Predicting BiodegradabilityChallenge

Table of Contents

Report Background	1
Introduction	6
Data Description:.....	7
Feature Selection Results.....	7
Comparison of Methods.....	9
Conclusion.....	10

Report Background

This report was prepared for **ChemsRUs** by *YOUR NAME HERE*

This report embeds all of the R code necessary to produce the results described in this report. If non-R programs are used, then summarize the results here.

*NOTE: THIS IS A TEMPLATE FOR THE REPORT WITH INSTRUCTIONS FROM ASSIGNMENTS INCLUDED FOR YOUR CONVENIENCE. **DELETE THE INSTRUCTIONS GIVEN IN ITALICS IN YOUR FINAL REPORT NOTEBOOK.** THE NOTEBOOK SHOULD BE AS YOU WOULD GIVE CHEMS-R-US. CODE FRAGMENTS ARE PROVIDED. YOU CAN ADD OR DELETE R CODE BLOCKS AS NECESSARY. THERE IS SOME SAMPLE CODE AT THE END OF THIS NOTEBOOK. IT SHOULD BE REMOVED BEFORE SUBMISSION.*

We split the code into 90% train and 10% validation.

```
# Prepare biodegradability data
# get training data
cdata.df <- read.csv("~/MATP-4400/data/chems_train.data.csv", header=FALSE)
# get external testing data
tdata.df <- read.csv("~/MATP-4400/data/chems_test.data.csv", header=FALSE)
# get feature names and add them to columns
featurenames <- read.csv("~/MATP-4400/data/chems_feat.name.csv", header=FALSE)
colnames(tdata.df) <- featurenames$V1
colnames(cdata.df) <- featurenames$V1
# get class as factors
class <- as.factor(read.csv("~/MATP-4400/data/chems_train.solution.csv", header=FALSE)$V1)
```

scale the data

```

sc_tr <- scale(cdata.df, center = TRUE, scale = TRUE)
means <- attr(sc_tr, 'scaled:center') # get the mean of the columns
stdevs <- attr(sc_tr, 'scaled:scale') # get the std of the columns
cdata_scaled.df <- scale(cdata.df, center=means, scale=stdevs)#scale tst
using the means and std of tr'

sc_tr <- scale(tdata.df, center = TRUE, scale = TRUE)
means <- attr(sc_tr, 'scaled:center') # get the mean of the columns
stdevs <- attr(sc_tr, 'scaled:scale') # get the std of the columns
tdata_scaled.df <- scale(tdata.df, center=means, scale=stdevs)#scale tst
using the means and std of tr'

#ss will be the number of data in the training set
n <- nrow(cdata.df)
ss<- ceiling(n*0.90)
# Set random seed for reproducibility
set.seed(300)
train.perm <- sample(1:n,ss)
trainlabel <- class[train.perm]
train <- cdata.df[train.perm, ] #The training data is just the training
rows
validation <- cdata.df[-train.perm, ] # Using -train gives us all rows
except the training rows.
classtrain<-class[train.perm] #solution train
classval <-class[-train.perm] # solution val

#glm original test
train.df <-cbind(train,classtrain)
lrfit <- glm(classtrain~., data=train.df,family = "binomial")

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

# Predict validation
ranking_lr<-predict(lrfit,validation,type="response")
head(ranking_lr)

##           3           36           39           54           62
64
## 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 2.023438e-07
1.000000e+00

# Create the actual validation class labels
# there are quicker ways to do, this but this illustrates the logic
temp <- ranking_lr > 0.5
temp[temp==TRUE]<-1
temp[temp==FALSE]<- -1
classval_lr <- as.factor(temp)

confusion.matrix<-table(classval,classval_lr)
classval[1:10]<-1

```

```
kable(confusion.matrix, type="html", digits = 2, caption="Actual versus Predicted Class")
```

Actual versus Predicted Class

```
      -1    1
-1  56   13
1    7   29
```

True Positive Rate or Sensitivity

```
Sensitivity<-
confusion.matrix[1,1]/(confusion.matrix[1,1]+confusion.matrix[1,2])
```

True Negative Rate or Specificity

```
Specificity<-
confusion.matrix[2,2]/(confusion.matrix[2,1]+confusion.matrix[2,2])
```

```
BalancedAccuracy<- (Sensitivity+Specificity)/2
```

```
Sensitivity
```

```
## [1] 0.8115942
```

```
Specificity
```

```
## [1] 0.8055556
```

```
BalancedAccuracy
```

```
## [1] 0.8085749
```

lda test with all features and unscaled data

```
lda.fit <- lda(train,trainlabel,prior=c(1,1)/2)
predict_train<-predict(lda.fit,validation,type="response")$class
train.pred <- as.vector(predict_train)
train.pred[train.pred=="lived"] <- -1
train.pred[train.pred=="died"] <- 1
train.pred <- as.factor(train.pred)
```

Table command counts the actual versus the predicted labels for the training data. The result is stored in a confusion matrix

```
classval[1:10]<-1
confusion.matrix<-table(classval,predict_train)
kable(confusion.matrix, type="html", digits = 2)
```

```
      -1    1
-1  59   10
1    6   30
```

```
accplus<-confusion.matrix[1,1]/(confusion.matrix[1,1]+confusion.matrix[1,2])
```

```
accneg<-confusion.matrix[2,2]/(confusion.matrix[2,1]+confusion.matrix[2,2])
```

Calculate the balanced accuracy

```
balanced_accuracy<-(accplus+accneg)/2
```

```

accplus
## [1] 0.8550725

accneg
## [1] 0.8333333

balanced_accuracy
## [1] 0.8442029

```

feature selection I make the threshold p value in to 0.01 and 23 more features are being included now 44 features count into classification

```

# Calculate the PCA
#my first way of selecting features
my.pca0 <- princomp(cdata.df)
l<-loadings(my.pca0)
s1<-l[,1]
s2<-l[,2]
s1 <- abs(s1)>0.00025
s1[s1==TRUE]<-1
s1[s1==FALSE]<- 0
s1 <- as.factor(s1)
features1<-matrix(0,nrow=(ncol(train)),ncol=1)
i<-1
size <-0
for(val in as.vector(s1)){
  features1[i,] = val
  if(val==1){
    size <- size +1
  }
  i <- i+1
}
features1 <-as.numeric(features1)
#number of selected features
size

## [1] 35

#second way of selecting features
summlrweights<-coef(summary(lrfit))[,1]
summlrweights<-summlrweights[-1] #drop the threshold entry
# get the p-values from the fourth column of summary
lrpvalues<-coef(summary(lrfit))[,4]
lrpvalues<-lrpvalues[-1]

keepfeatures<-lrpvalues<0.005
#these are the features we are keeping
sum(keepfeatures)

```

```
## [1] 29
```

use scaled data with the application of LDA

```
# Create a formula that uses only the features names in keepfeature
train_scaled <- cdata_scaled.df[train.perm, ]
validation_scaled <- cdata_scaled.df[-train.perm, ]
trainfs.df <- cbind(train_scaled[,keepfeatures],classtrain)

lda.fit2 <- lda(train_scaled,trainlabel,prior=c(1,1)/2)
#threshold value
thresh <- ((lda.fit2$means[1,] +lda.fit2$means[2,])/2)%*%lda.fit2$scaling
#prediction
train.pred2 <- predict(lda.fit2,validation_scaled)$class
summary(train.pred2)

## -1  1
## 65 40

train.pred2 <- as.vector(train.pred2)
train.pred2[train.pred2=="lived"] <- -1
train.pred2[train.pred2=="died"] <- 1
train.pred2 <- as.factor(train.pred2)

confusion.matrix<-table(classval,train.pred2)
kable(confusion.matrix, type="html",digits = 2,caption="Actual versus
Predicted Class")
```

Actual versus Predicted Class

```
      -1  1
-1  59 10
1    6 30
```

```
# True Positive Rate or Sensitivity
Sensitivity<-
confusion.matrix[1,1]/(confusion.matrix[1,1]+confusion.matrix[1,2])
# True Negative Rate or Specificity
Specificity<-
confusion.matrix[2,2]/(confusion.matrix[2,1]+confusion.matrix[2,2])
BalancedAccuracyfs<- (Sensitivity+Specificity)/2
Sensitivity

## [1] 0.8550725

Specificity

## [1] 0.8333333

BalancedAccuracyfs

## [1] 0.8442029
```

Comparison with the logistic regression (original Chem-R-US method) the logistic regression method shows a much more accurate balanced accuracy LDA method fail to calculate the died or(un-biodegradable value), but it shows a higher sensitivity to the bio-degradable value which is over 80 percent while the logistic regressions shows similar accuracy. But over all, logistic regression shows higher value

```
trainfs.df0 <- cbind(train[,keepfeatures],classtrain)
trainfs.df0 <-as.data.frame(trainfs.df0)
lrfitfs <- glm(classtrain~., data=trainfs.df0,family = "binomial")
# Predict validation
ranking_lrfs<-predict(lrfitfs,validation[,keepfeatures],type="response")

# Predict the test data. We use ranking_lrtest since the contest is based on
the AUC.
# ranking_lrtest<-predict(lrfit,tdata.df,type="response")
tdata_scaled.df <-as.data.frame(tdata_scaled.df)
ranking_lrtest<-predict(lrfitfs,tdata.df[,keepfeatures])
ranking_lrtest<-as.numeric(ranking_lrtest)
# no need to convert to 0 and 1, can need ranking for AUC.
write.table(ranking_lrtest,file = "classification.csv", row.names=F,
col.names=F)

#selection features file
# Here is the mean prediction file for submission to the website
# features should be a column vector of 0 and 1. 1=keep feature, 0 = don't
# Set the ones we want to keep to 0
features1 <-as.matrix(features1)
features2<-matrix(0,nrow=(ncol(train)),ncol=1)
# Set the ones we want to keep to 0
features2[keepfeatures]<-1
write.table(features1,file = "selection.csv", row.names=F, col.names=F)

#This automatically generates a compressed (zip) file
system("zip -u MyEntry.csv.zip classification.csv")
system("zip -u MyEntry.csv.zip selection.csv")
```

Introduction

Provide an overview of your report

Chems-R-Us has created an entry to the challenge at <https://competitions.codalab.org/competitions/22892> based on logistic regression. Their entry is in the file 'FinalProjChemsRUs.Rmd' Based on the information in the leaderboard under bennek, their entry is not performing feature selection well. The approaches tried by Chems-R-Us were logistic regression with feature selection based on the coefficients of logistic regression with p-values used to determine importance.

The purpose of this report is to investigate alternative approaches that achieve high AUC scores on the testing set while correctly identifying the relevant features.

Data Description:

Provide a basic description of the data which includes: 1. number of attributes 2. number of points in each class 1) the train class has 950 samples (train.perm) the val class has 105 samples (classval) totally there are 1055 samples in cdata.df there are 400 samples in tdata.df for PCA feature-selection way, I have found 46 features while 44 were found by elevating the threshold probability

*Describe the data preparation The data preparation should include:

- Read in the external train and external test datasets.
- Divide the external training set into an internal train and internal validation set using an 90% and 10% split.
- Chems-R-Us did not scale any data in their entry. But you can add centering and scaling if desired.

Handy HINTS if you would like to scale: The following code scales data in matrix tr and then applies the same scaling to matrix tst.

```
sc_tr <- scale(tr, center = TRUE, scale = TRUE) # scale tr means <- attr(sc_tr, 'scaled:center')
# get the mean of the columns stdevs <- attr(sc_tr, 'scaled:scale') # get the std of the
columns sc_tst <- scale(tst, center=means, scale=stdevs) #scale tst using the means and std
of tr' *
```

I have scaled the test data set and train data set and attribute them with new variable names : tdata_scaled.df, cdata.df and validation_scaled # Baseline Results:

*Investigation of alternative using all the features. ChemsRUs has asked you to evaluate how LDA and logistic regression performs on this problem using all the features. Divide the training data into 90% train and 10% validation splits. Set seed(300) before you split so you get the same train and validation splits. Train LDA and logistic regression on the training data and evaluate how well it does on the validation data. Compute the balanced accuracy and the AUC for the train and test results. Compare the results between the two models. lda method with all features | | -1| 1| |:-|:-|:-| | -1 | 59| 10| |1 | 6| 30| [1] 0.8550725 [1] 0.8333333 [1] 0.8442029 logistic method with all features | | -1| 1| |:-|:-|:-| | -1 | 56| 13| [1] | 7| 29| [1] 0.8115942 [1] 0.8055556 [1] 0.8085749 the lda methods shows a more accurate balanced predictability : 84.42% while the original general method gives a percentage of around 80.8 percent

Feature Selection Results

Create an approach for selecting the relevant features. Describe your approach. Describe the features that you selected. Create a PCA biplot comparing the two classes with these two

classes. Create a classifier using LDA and logistic regression using these features. Evaluate how they perform on the validation sets in term of balanced accuracy and AUC. Compare your results with your prior results. Discuss your findings. I have applied two ways of selecting features 1) I use PCA to find out the correlation between variables and the data sample with function of loading The bigger the loading value (the bigger projection value on the PCA components) is the more correlation between variable and the result is. It is 0f 60 percent of AUC score 2) I have elevated the threshold probability to 0.005 to hold more features with 57 percent of AUC score. PCA seemes to be a better way for features selection
Challenge Prediction

My challenge ID is Qianxy23 with an AUC score of 86% for prediction and 60% for feature selection.

Pick your best shot classification and feature selection methods and then enter the contest. Provide your scores in the text. Discuss your results and the strengths and weaknesses of the different approaches. There should be a separate entry for each participant. Make sure that between all of your teams entries three different classification methods are used and three different feature selection methods are used. Using PCA with another method counts as a different classification method.

The contest can be found here:

<https://competitions.codalab.org/competitions/22892>

*Enter the contest. Prepare you entry by making the classification.csv and selection.csv and zipping them into a single file. See FinalProjChemsRUs.Rmd for an example and discussion of the format of the file.

Provide a csv file with your predictions of the biodegradability of each data point in chems_test.csv. Chems-R-Us will use this to independently verify the quality of your results. These predictions should be given as a csv files with on column containing the prediction (1 or -1) for each points in chemstest.csv. Provide a csv file with your prediction of which features are real. The feature predictions should be given as a csv files with on column containing the prediction (1 or 0) indicating if each of the 168 features should be included.

Create the files:

- classification.csv: Test target values (437 lines x 1 column)
- selection.csv: Solution indicating which variables are real and which are fake (168 lines x 1 column)

Your submission must be a zip archive containing the following files: -

classification.csv, your predicted labels for test dataset. It should include plus or minus one values, one for each test sample, representing the class label predictions. -

selection.csv, representing the features you selected as real or fake (ie 0 or 1).

Comparison of Methods

- Create a table comparing the different results for the methods you tried. Discuss which method performed best for feature selection, and which method worked best for prediction. What method would you recommend overall? Why?*

```
train.pred<-as.numeric(train.pred)
roc_rose <- plot(pROC::roc(classval,train.pred), print.auc = TRUE,col =
"blue",main="LDA vs GLM without scaling and feature selection")

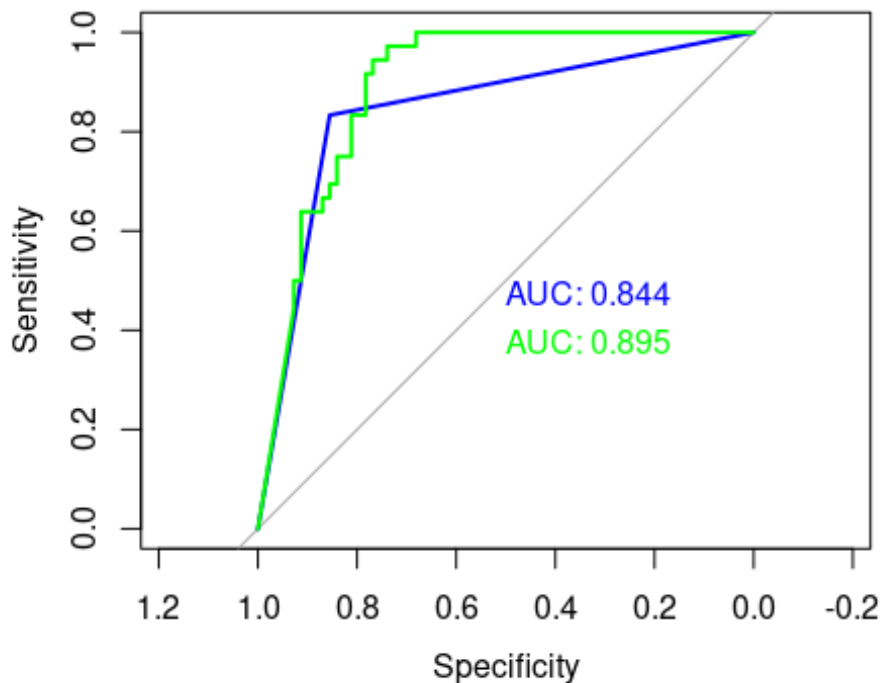
## Setting levels: control = -1, case = 1

## Setting direction: controls < cases

# This prints the other curve
roc_rose <- plot(roc(classval, ranking_lr), print.auc = TRUE,
col = "green", print.auc.y = .4, add = TRUE)

## Setting levels: control = -1, case = 1
## Setting direction: controls < cases
```

LDA vs GLM without scaling and feature selection



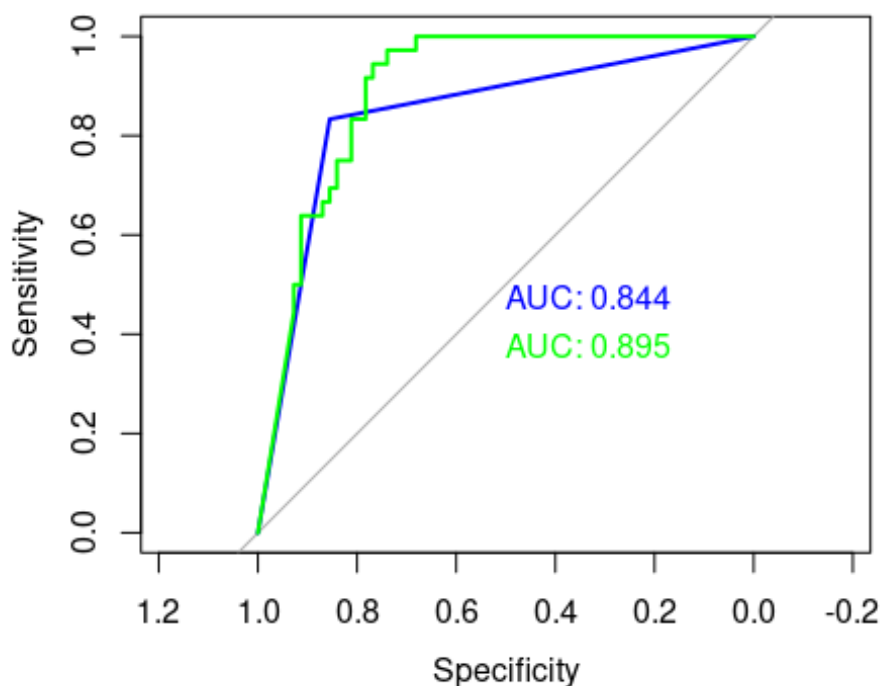
```
#comparison between LDA scaled and selected-feature method and GLM method
without scaling and selected
train.pred2<- as.numeric(train.pred2)
roc_rose <- plot(pROC::roc(classval,train.pred2), print.auc = TRUE,col =
"blue",main="FS scaled data with LDA VS GLM without scaling or FS")

## Setting levels: control = -1, case = 1
## Setting direction: controls < cases
```

```
# This prints the other curve
roc_rose <- plot(roc(classval, ranking_lr), print.auc = TRUE,
                 col = "green", print.auc.y = .4, add = TRUE)

## Setting levels: control = -1, case = 1
## Setting direction: controls < cases
```

FS scaled data with LDA VS GLM without scaling or



Additional

Analysis

Provide an additional analysis and/or visualization that may be insightful to Chems-R-Us. Use your imagination, extra credit for creativity here! Discuss the insights your analysis provides. Be sure to title any figures! Comment your code so all can understand what you are doing. Feel free to use any R code from class or from the web.

Conclusion

Provide a conclusion which summarizes your results briefly and adds any observations/suggestions that you have for Chems-R-Us about the data, model, or future work.

the validation test shows a preferred way of LDA to original GLM procedures. However, the size of the selection features seem do not influence that much to the genral result. And so does the way of scaling the data or not. What I recommend is to use the same unit or same kind of data features being provided (e.g all numeric), then it might help for the elevation of the accuracy of predication and selection of features