Urinary Biomarkers for Pancreatic Cancer

Log Theme09 - Introduction Machine Learning

Lisa Hu
414264
Bio-Informatics
Hanzehogeschool Groningen, ILST

Dave Langers (LADR) & Bart Barnard (BABA)

October 27, 2022

Contents

1 Data description **2**

2 Reading the data **2**

3 Manipulate the data **4**

3.1 REG1A vs. REG1B 5

3.2 Log transformation 5

4 Analyse the data **7**

4.1 Boxplots 7

4.2 Correlation matrix 9

4.3 PCA 10

5 Machine Learning **11**

5.1 Model exploration 11

5.2 Metrics 11

5.3 Weka: Model Exploration 12

1 Data description

The data can be found on kaggle.com: Urinary biomarkers for pancreatic cancer The files are saved as `Data.csv` and `Documentation.csv` for easier access.

The following packages were used:

- ggplot2
- tidyr
- dplyr
- readr

2 Reading the data

We first want to create an insight of our data:

```
dataset <- read.csv("../data/Data.csv")
codebook <- read_delim("../data/codebook.txt", delim = "|")
pander(codebook[1:4], booktabs = T, caption = "Data values", split.tables = 100)
```

Table 1: Data values

Name	Full Name	Type	Unit
sample_id	Sample ID	chr	-
patient_cohort	Patient's Cohort	chr	-
sample_origin	Sample Origin	chr	-
age	Age of subject	dbl	-
sex	Sex of subject	chr	-
diagnosis	Diagnosis	dbl	-
stage	Stage	chr	-
benign_sample_diagnosis	Benign Sample's Diagnosis	chr	-
plasma_CA19_9	Blood plasma CA19-9	dbl	U/ml
creatinine	Creatinine	dbl	mg/ml
LYVE1	LYVE1	dbl	ng/ml
REG1B	REG1B	dbl	ng/ml
TFF1	TFF1	dbl	ng/ml
REG1A	REG1A	dbl	ng/ml

```
pander(codebook[c(1,5)], booktabs = T, caption = "Description",
       justify = c("right", "left"), split.tables = 100)
```

Table 2: Description

Name	Description
sample_id	Unique string identifying each subject
patient_cohort	Cohort 1 = previously used samples; Cohort 2 = newly added samples
sample_origin	BPTB: Barts Pancreas Tissue Bank, London, UK; ESP: Spanish National Cancer Research Centre, Madrid, Spain; LIV: Liverpool University, UK; UCL: University College London, UK
age	Age in years
sex	M = male; F = female

Name	Description
diagnosis	1 = control (no cancer); 2 = benign hepatobiliary disease; 3 = PDA (pancreatic cancer)
stage	The stage of the disease (IA, IB, IIA, IIB, III, IV)
benign_sample_diagnosis	The diagnosis for those with a benign diagnosis
plasma_CA19_9	Blood plasma levels of CA19-9 monoclonal antibody, usually elevated when pancreatic cancer
creatinine	Urinary biomarker of kidney function
LYVE1	Urinary levels of Lymphatic Vessel Endothelial Hyaluronan receptor 1
REG1B	Urinary levels of Regenerating Family Member 1 Beta
TFF1	Urinary levels of Trefoil Factor 1
REG1A	Urinary levels of Regenerating Family Member 1 Alpha

The information given in the codebook originates from the `Documentation.csv`. This file was given with the data file and can be found on the website.

3 Manipulate the data

A lot of the rows contain empty strings instead of NA, which has to be fixed first. Besides that, the columns `sample_id` , `patient_cohort`, `sample_origin`, and `benign_sample_diagnosis` in the dataset significant value for the analysis and are therefor dropped. A column `diagnosis_group` was added for a comparison test.

```r
# Change the empty strings to NA
dataset[dataset == ""] <- NA

# Remove unnecessary columns
drop <- c("sample_id", "patient_cohort", "sample_origin", "benign_sample_diagnosis")
dataset <- dataset[,!(names(dataset) %in% drop)]

# Group the samples
dataset <- dataset %>%
  mutate(
    ## Factor for order of age
    diagnosis_group = factor(
      dplyr::case_when(
        diagnosis == 1 ~ "Control",
        diagnosis == 2 ~ "Benign",
        stage == "I" ~ "I-IIA",
        stage == "IA" ~ "I-IIA",
        stage == "IB" ~ "I-IIA",
        stage == "II" ~ "I-II",
        stage == "IIA" ~ "I-IIA",
        stage == "IIB" ~ "I-II",
        stage == "III" ~ "III-IV",
        stage == "IV" ~ "III-IV" ),
      level = c("Control", "Benign", "I-IIA", "I-II", "III-IV")
    )
  )

# Perform the tests
REG1A <- dunnTest(dataset$REG1A ~ dataset$diagnosis_group)
REG1B <- dunnTest(dataset$REG1B ~ dataset$diagnosis_group)

# Create a nice format to show the correct comparisons
comparison <- t(cbind(REG1A$res[c(2:5,7,8),c(1,4)], REG1B$res[c(2:5,7,8),4]))
colnames(comparison) <- comparison[1,]
comparison <- comparison[-1,]
comparison <- apply(comparison, 2, as.numeric)
rownames(comparison) <- c("REG1A", "REG1B")
comparison[comparison > 0.05] <- "ns"

dataset <- dataset[,!(names(dataset) %in% "REG1A")]
```

3.1 REG1A vs. REG1B

Although performance between the two is similar, a Kruskal-Wallis test with Dunn's multiple comparisons shows that REG1B outperforms REG1A when the control and benign samples are compared to the I-IIA PDAC samples. Therefor, REG1B was used further on in the experiments and REG1A is dropped.

3.2 Log transformation

A summary of the data shows very high maximum values, but rather low medians. A log-transformation is applied to correct this.

```r
pander(summary(dataset), split.table = 100)
```

Table 3: Table continues below

age	sex	diagnosis	stage	plasma_CA19_9	creatinine	LYVE1	REG1B	TFF1	diagnosis_group
Min. :26.00	Length:590	Min. :1.000	Length:590	Min. : 0.0	Min. :0.05655	Min. : 0.000129	Min. : 0.0011	Min. : 0.005	Control/Benign:391
1st Qu.:50.00	Class :character	1st Qu.:1.000	Class :character	1st Qu.: 8.0	1st Qu.:0.37323	1st Qu.: 0.167179	1st Qu.: 10.7572	1st Qu.: 43.961	I-IIA : 27
Median :60.00	Mode :character	Median :2.000	Mode :character	Median : 26.5	Median :0.72384	Median : 1.649862	Median : 34.3034	Median : 259.874	I-II : 75
Mean :59.08	NA	Mean :2.027	NA	Mean : 654.0	Mean :0.85538	Mean : 3.063530	Mean : 111.7741	Mean : 597.869	III-IV : 97
3rd Qu.:69.00	NA	3rd Qu.:3.000	NA	3rd Qu.: 294.0	3rd Qu.:1.13948	3rd Qu.: 5.205037	3rd Qu.: 122.7410	3rd Qu.: 742.736	NA
Max. :89.00	NA	Max. :3.000	NA	Max. :31000.0	Max. :4.11684	Max. :23.890323	Max. :1403.8976	Max. :13344.300	NA
NA	NA	NA	NA	NA's :240	NA	NA	NA	NA	NA

```r
log.data <- log(dataset[5:9] +1)
dataset[5:9] <- log.data
```

The samples are then grouped by diagnosis for easier access of the different samples. Table 5 shows the different amounts of samples per diagnosis and the amount of which are also blood samples. After the blood samples are seperated the column can be dropped.

```r
# Different diagnosis and blood groups
control <- subset(dataset, diagnosis == 1)
benign <- subset(dataset, diagnosis == 2)
pdac <- subset(dataset, diagnosis == 3)
blood <- subset(dataset, plasma_CA19_9 >= 0)

# Drop the "plasma" column
dataset <- dataset[,-5]

# Demographics
demograph <- data.frame(c(sum(control$sex == "F"), sum(control$sex == "M")),
                        c(sum(benign$sex == "F"), sum(benign$sex == "M")),
                        c(sum(pdac$sex == "F"), sum(pdac$sex == "M")))

blood.demo <- data.frame(c(sum(blood$sex == "F" & blood$diagnosis == 1),
                           sum(blood$sex == "M" & blood$diagnosis == 1)),
                         c(sum(blood$sex == "F" & blood$diagnosis == 2),
                           sum(blood$sex == "M" & blood$diagnosis == 2)),
                         c(sum(blood$sex == "F" & blood$diagnosis == 3),
                           sum(blood$sex == "M" & blood$diagnosis == 3)))

colnames(blood.demo) <- c("Control", "Benign", "PDAC")
colnames(demograph) <- c("Control", "Benign", "PDAC")
demograph <- rbind(demograph, blood.demo)
rownames(demograph) <- c("Female total", "Male total", "Female blood", "Male blood")

pander(demograph, booktabs = T, caption = "Demographic of the samples",
       justify = c("left", "center", "center", "center"))
```

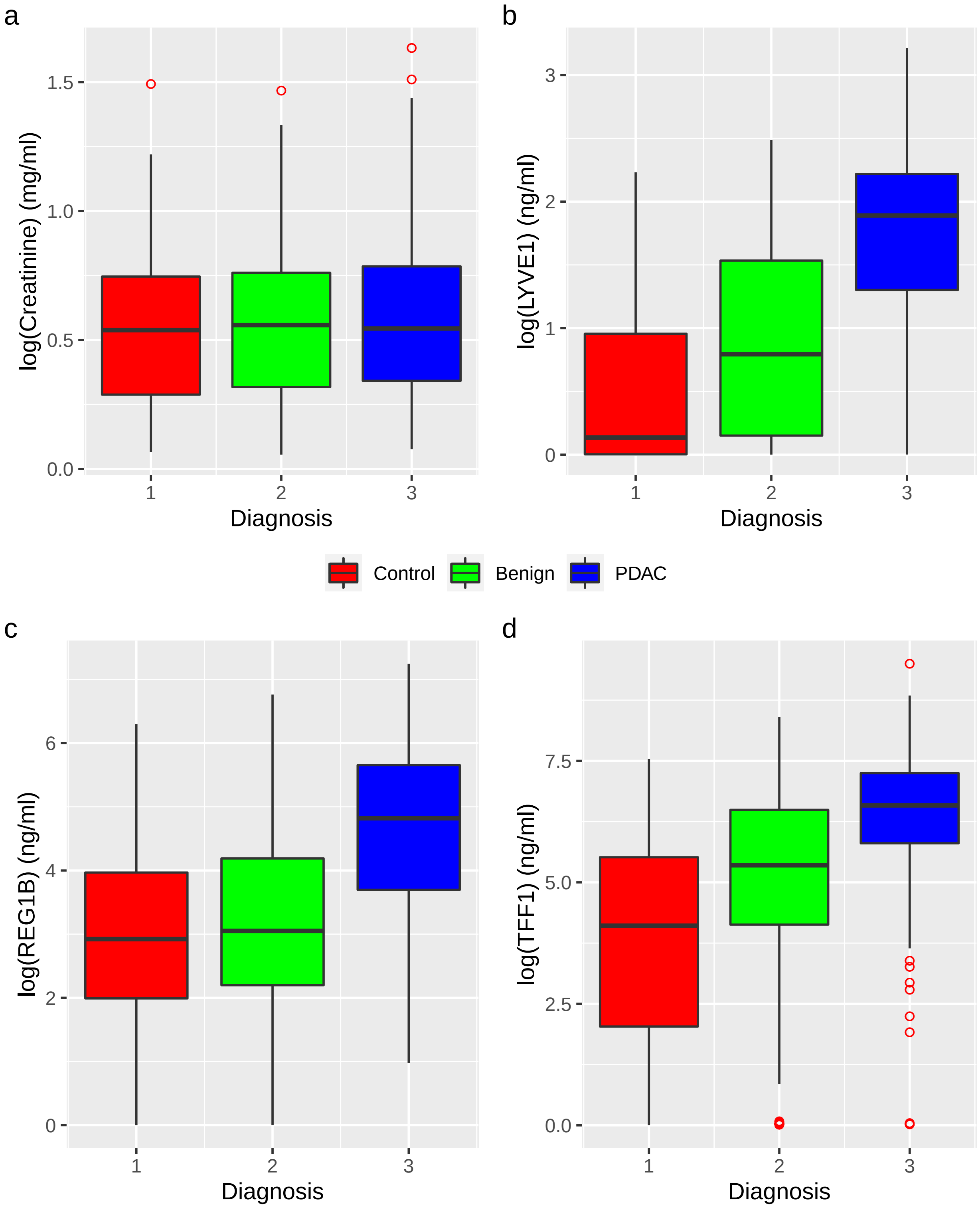
Table 5: Demographic of the samples

	Control	Benign	PDAC
Female total	115	101	83
Male total	68	107	116
Female blood	58	57	64
Male blood	34	51	86

4 Analyse the data

4.1 Boxplots

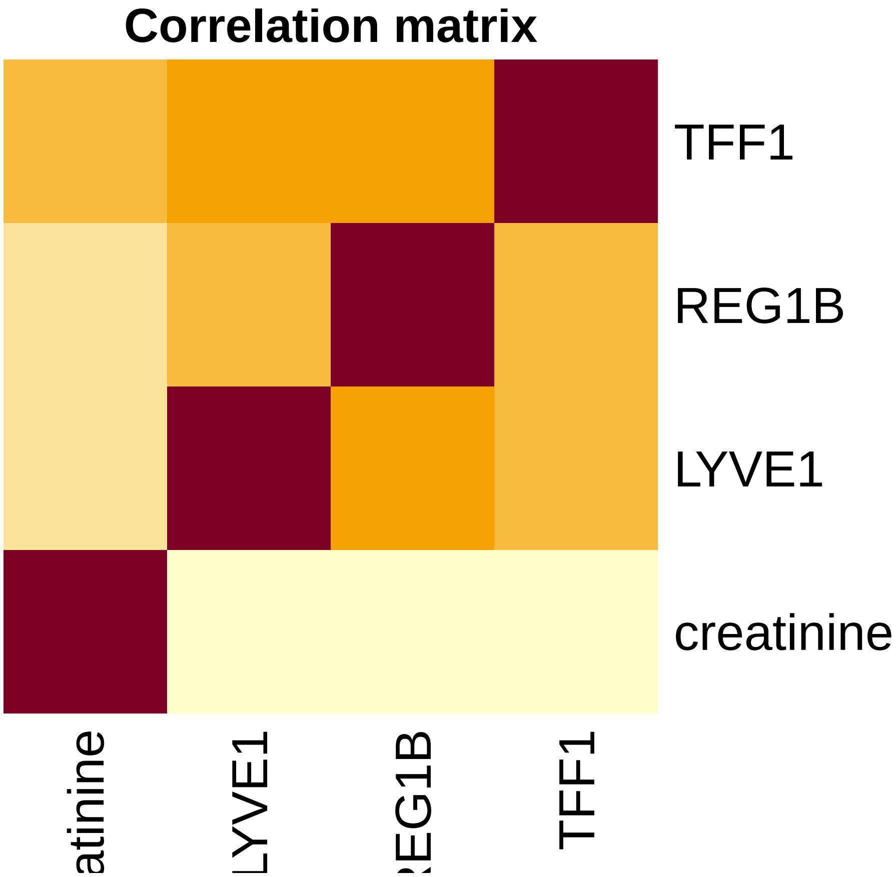
```r
# Boxplot function
create.plots <- function(y.values, y.label, plt.tag) {
  list(ggplot(data = control, aes(x = diagnosis, y = !!sym(y.values))) +
    geom_boxplot(outlier.color = "red", outlier.shape = 1, aes(fill = "Control")) +
    geom_boxplot(data = benign, outlier.color = "red", outlier.shape = 1,
                 aes(fill = "Benign")) +
    geom_boxplot(data = pdac, outlier.color = "red", outlier.shape = 1,
                 aes(fill = "PDAC")) +
    labs(x = "Diagnosis", y = y.label, tag = plt.tag) +
    scale_fill_manual(values = c("red", "green", "blue"),
                      limits = c("Control", "Benign", "PDAC"),
                      name = ""))
}

# Create the boxplots for the different columns
y.values <- names(dataset[5:8])
y.labs <- c("log(Creatinine) (mg/ml)", "log(LYVE1) (ng/ml)", "log(REG1B) (ng/ml)",
            "log(TFF1) (ng/ml)")
plt.tag <- c("a", "b", "c", "d")
plts <- mapply(create.plots, y.values, y.labs, plt.tag)

# Grid and print the plots
p1 <- ggarrange(plotlist = plts[1:2], ncol = 2,
                common.legend = TRUE, legend = "bottom")
p2 <- ggarrange(plotlist = plts[3:4], ncol = 2,
                common.legend = TRUE, legend = "none")
my.grid <- ggarrange(p1, p2, nrow = 2)
print(annotate_figure(my.grid))
```



The outliers are not localized in a specific diagnosis group, but rather spread over the groups.

4.2 Correlation matrix

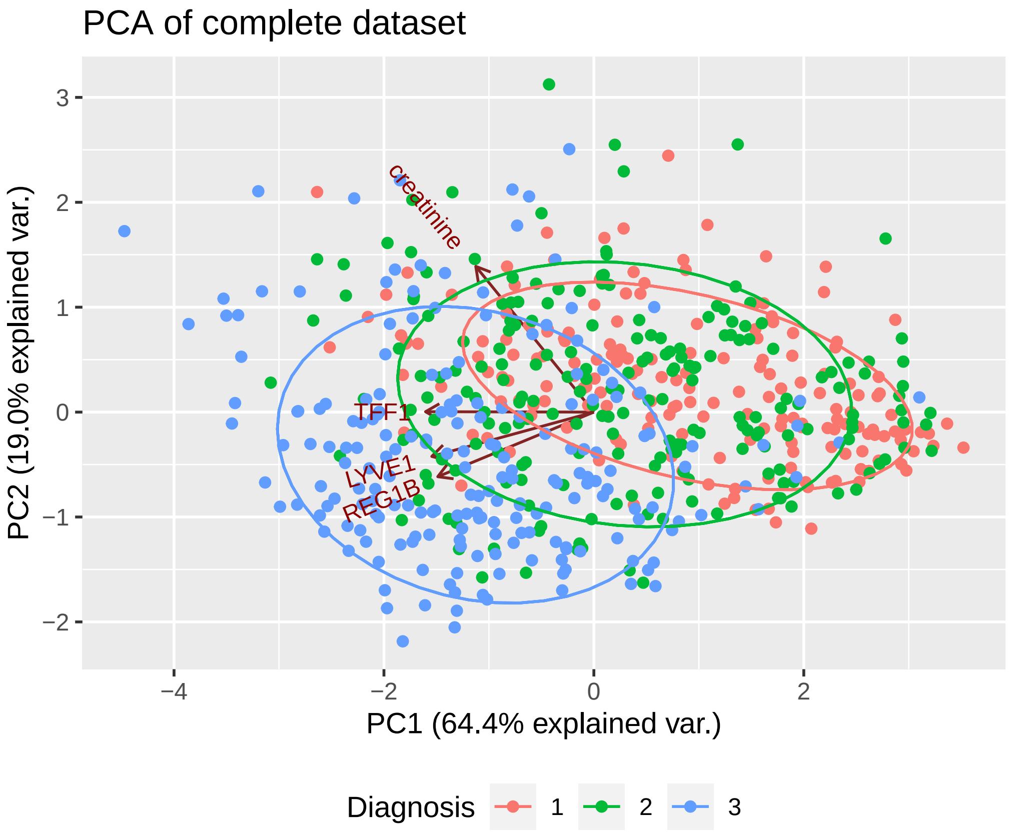
```
cor_matrix <- cor(dataset[,5:8])
heatmap(cor_matrix, scale = "column", Colv = NA, Rowv = NA, main = "Correlation matrix")
```



The heatmap shows that there is not much correlation between creatinine and the other variables. The other outstanding one has to be the TFF1 biomarker, being the most correlated variable to others.

4.3 PCA

```
pca <- prcomp(dataset[,5:8], center = TRUE, scale. = TRUE)
ggbiplot(pca, obs.scale = 1, var.scale = 1, groups = factor(dataset$diagnosis),
         ellipse = TRUE, circle = FALSE) +
  ggtitle("PCA of complete dataset") +
  guides(color = guide_legend(title = "Diagnosis")) +
  theme(legend.position = "bottom")
```



While the control and benign group show relative distance from the PDAC group, there is a still a lot of overlapping samples with the benign and PDAC groups. As earlier concluded from the heatmap, the creatinine biomarker does not show much relativeness with the other biomarkers. LYVE1 is nicely in between the TFF1 and REG1B biomarkers. Every point close tho the origin have values close to the mean for all variables.

5 Machine Learning

5.1 Model exploration

For the exploration of the model, the data is cleaned it contains only the biomarkers and the classification labels (Control, Benign, I-II, and III-IV).

```r
cleaned <- read.csv("../data/cleaned_data.csv")
pander(head(cleaned))
```

plasma_CA19_9	creatinine	LYVE1	REG1B	TFF1	diagnosis_group
2.542	1.041	0.6383	3.988	6.485	Control
NA	0.6794	1.111	4.559	5.349	Control
2.079	0.5768	0.1359	4.638	6.136	Control
2.197	0.5313	0.002801	4.12	4.969	Control
2.303	0.1947	0.0008592	4.198	3.74	Control
NA	0.6142	0.003387	4.145	4.107	Control

To set a baseline, the data is run through different types of algorithms in Weka:

Table 7: Results algorithm comparison in Weka, '*' = significantly worse; 'v' = significantly better

Algorithm	**Speed**	**Accuracy**	**Sensitivity**	**Specificity**	**ROC**	**FNR**
ZeroR	0.00s	35.25%	0.00	1.00	0.50	1.00
OneR	0.00s	41.19% v	0.48 v	0.78 *	0.63 v	0.52 *
NaiveBayes	0.00s	50.07% v	0.57 v	0.82 *	0.81 v	0.43 *
SimpleLogistics	0.04s v	54.17% v	0.51 v	0.86 *	0.82 v	0.49 *
SMO	0.01s v	51.17% v	0.39	0.89 *	0.78 v	0.61 *
IBk	0.00s	39.25%	0.44 v	0.89 *	0.66 v	0.56 *
J48	0.01s v	47.63% v	0.60 v	0.79 *	0.79 v	0.40 *
RandomForest	0.17s v	53.22% v	0.65 v	0.82 *	0.84 v	0.35 *

These results show a relative low sensitivity and high FNR. Some algorithms have a low ROC value, low sensitivity and low accuracy: OneR, IBk, and J48 are not further analysed. OneR will be kept to set a baseline. To apply the CostSensitiveClassifier, the confusion matrix of every algorithm should be known:

Table 8: Confusion matrix per algorithm

NaiveBayes

a	b	c	d	classified as
105	60	6	12	a = Control
74	96	21	17	b = Benign
1	23	43	35	c = I-II
3	10	35	49	d = III-IV

SimpleLogistics

a	b	c	d	classified as
94	85	2	2	a = Control
49	143	11	5	b = Benign
3	29	42	28	c = I-II
2	17	35	43	d = III-IV

SMO

a	b	c	d	classified as
70	112	1	0	a = Control
37	155	11	5	b = Benign
3	34	42	23	c = I-II
2	28	32	35	d = III-IV

RandomForest

a	b	c	d	classified as
177	60	0	6	a = Control
57	129	14	8	b = Benign
8	35	32	27	c = I-II
5	27	28	37	d = III-IV

5.2 Metrics

For predictive models

5.3 Weka: Model Exploration