NN Classiffication - Amino Acid Example

MCC

2/19/2020

Obtain uniquely sorted FP & FN from number sets

OR

Generating Outliers and Venn Diagrams

Introduction

One goal of this experiment is to determine if there are any patterns between the lists of ouutliers generated by PCA and the FP/FN generated when the machine learning models were carried out. To review quickly, the **Outliers** were produced by finding observations (in this case proteins, either myoglobin or control sets), which were greater than or equal to 3 stadnard deviations away from the of the first and second principal components. The *Outliers* were produced as can be seen in the flowchart below.

The logistic regression set included a total of x observations generated by 5 fold cross-validatioon

It is hoped that the *Outliers* generated from the PCA (unsupervised learning) will have some correlation to the 6 statisitcal learning methods investigated for this report.

```
## Load Libraries
rm(list = ls())
Libraries = c("doMC", "knitr", "readr", "ggVennDiagram")

for(p in Libraries){  # Install Library if not present
    if(!require(p, character.only = TRUE)) { install.packages(p) }
    library(p, character.only = TRUE)
}
opts_chunk$set(cache = TRUE)
```

Logit set

```
keep <- "rowIndex"

fp_fn_logit <- read_csv("../00-data/03-ml_results/fp_fn_logit.csv")

logit_fp_fn_nums <- sort(unique(unlist(fp_fn_logit[, keep], use.names = FALSE)))

length(logit_fp_fn_nums)
```

```
## [1] 119
```

```
logit_fp_fn_nums
```

```
##   [1]    1    2    8   10   46   57   58   88  100  114  130  146  150  182  183
##  [16]  239  249  252  254  302  368  400  407  449  453  501  503  516  518  526
##  [31]  531  542  547  566  573  580  592  655  910  912  913  980 1032 1033 1034
##  [46] 1035 1041 1067 1069 1092 1093 1094 1099 1100 1101 1106 1116 1117 1121 1128
##  [61] 1130 1135 1140 1141 1142 1144 1147 1150 1152 1190 1219 1222 1223 1224 1226
##  [76] 1233 1234 1264 1279 1281 1282 1471 1482 1484 1508 1510 1522 1569 1571 1574
##  [91] 1575 1576 1579 1585 1588 1589 1594 1600 1618 1622 1623 1693 1723 1780 1828
## [106] 1829 1830 1832 1833 1845 1846 1847 1848 1849 1850 1852 1853 1872 1873
```

```
write_csv(x = as.data.frame(logit_fp_fn_nums),
          path = "../00-data/04-sort_unique_outliers/logit_nums.csv")
```

- The ‘logistic regression set’ included a total of 119 unique observations containing both FP and FN.

Random Forest set

```
fp_fn_r_forest <- read_csv("../00-data/03-ml_results/fp_fn_r_forest.csv")

r_forest_fp_fn_nums <- sort(unique(unlist(fp_fn_r_forest[, keep], use.names = FALSE)))

length(r_forest_fp_fn_nums)
```

```
## [1] 46
```

```
r_forest_fp_fn_nums
```

```
##  [1]    6   57  100  130  141  150  183  453  526  534  542  570  573  580  980
## [16] 1033 1034 1035 1091 1092 1093 1100 1101 1219 1223 1226 1233 1264 1470 1471
## [31] 1510 1569 1575 1576 1579 1585 1587 1588 1594 1608 1618 1622 1623 1780 1831
## [46] 1833
```

```
write_csv(x = as.data.frame(r_forest_fp_fn_nums),
          path = "../00-data/04-sort_unique_outliers/rf_nums.csv")
```

- The ‘Random Forest set’ included a total of 46 unique observations containing both FP and FN.

SVM Linear set

```
fp_fn_svm_linear <- read_csv("../00-data/03-ml_results/fp_fn_svm_linear.csv")

svm_linear_fp_fn_nums <- sort(unique(unlist(fp_fn_svm_linear[, keep], use.names = FALSE)))

length(svm_linear_fp_fn_nums)
```

```
## [1] 120
```

```
svm_linear_fp_fn_nums
```

```
##   [1]    1    2    8   10   46   57   58   88  100  114  130  150  182  183  249
##  [16]  252  254  301  302  368  400  407  453  501  503  516  518  526  531  542
##  [31]  547  566  573  580  655  910  912  913  980 1032 1033 1034 1035 1041 1067
##  [46] 1069 1092 1093 1094 1100 1101 1106 1116 1117 1121 1130 1135 1136 1138 1139
##  [61] 1140 1141 1142 1144 1145 1152 1190 1219 1222 1223 1226 1233 1234 1245 1264
##  [76] 1279 1281 1282 1471 1482 1484 1508 1510 1569 1574 1575 1576 1579 1580 1585
##  [91] 1588 1589 1594 1600 1608 1618 1622 1623 1693 1723 1734 1780 1828 1829 1830
## [106] 1831 1832 1833 1845 1848 1849 1850 1852 1853 1858 1863 1866 1868 1872 1873
```

```
write_csv(x = as.data.frame(svm_linear_fp_fn_nums),
          path = "../00-data/04-sort_unique_outliers/svm_lin_nums.csv")
```

- The 'SVM-Linear set' included a total of 125 unique observations containing both FP and FN.

SVM Polynomial Kernel set

```
fp_fn_svm_poly <- read_csv("../00-data/03-ml_results/fp_fn_svm_poly.csv")

svm_poly_fp_fn_nums <- sort(unique(unlist(fp_fn_svm_poly[, keep], use.names = FALSE)))

length(svm_poly_fp_fn_nums)
```

```
## [1] 70
```

```
svm_poly_fp_fn_nums
```

```
##  [1]    6   15   94  115  130  136  141  150  182  183  185  445  449  452  453
## [16]  522  525  526  529  530  531  532  534  542  546  560  562  566  568  570
## [31]  579  580  582  592  912  913  980 1034 1035 1067 1091 1093 1100 1101 1109
## [46] 1121 1188 1190 1219 1226 1233 1264 1471 1510 1522 1575 1576 1579 1585 1587
## [61] 1608 1618 1621 1623 1697 1734 1773 1780 1831 1833
```

```
write_csv(x = as.data.frame(svm_poly_fp_fn_nums),
          path = "../00-data/04-sort_unique_outliers/svm_poly_nums.csv")
```

- The 'SVM-Polynomial Kernel set' included a total of 70 unique observations containing both FP and FN.

SVM Radial Bias Kernel set

```
fp_fn_svmRadialCost <- read_csv("../00-data/03-ml_results/fp_fn_svmRbf.csv")

svm_svmRadial_fp_fn_nums <- sort(unique(unlist(fp_fn_svmRadialCost[, keep], use.names = FALSE)))

length(svm_svmRadial_fp_fn_nums)
```

```
## [1] 58
```

```
svm_svmRadial_fp_fn_nums
```

```
##  [1]    6   15   94  115  130  141  150  182  183  185  192  449  453  522  525
## [16]  526  529  531  534  542  546  566  568  570  580  582  592  655  913 1034
## [31] 1035 1091 1093 1094 1100 1101 1109 1121 1190 1219 1226 1233 1264 1471 1475
## [46] 1510 1575 1576 1579 1585 1587 1608 1618 1621 1766 1780 1831 1833
```

```
write_csv(x = as.data.frame(svm_svmRadial_fp_fn_nums),
          path = "../00-data/04-sort_unique_outliers/svm_rbf_nums.csv")
```

- The 'SVM-Polynomial Kernel set' included a total of 58 unique observations containing both FP and FN.

NEED DEEP LEARNING set

```
NNModel_fp_fn_nums <- read.csv("~/Dropbox/a1_mcc_project/05-ae-nn/NN_nums.csv", sep="")
```

- The ‘DL set’ included a total of X unique observations containing both FP and FN.

Statistical Learning Method Vs Total Number of FP/FN

Statistical Method	Total Number Produced	Unique	Total/Unique
Principal Componnent Analysis	461	460	1.002
Logit	537	119	4.51
SVM Linear	496	125	3.97
SVM Polynomial	278	70	3.97
SVM Radial Basis Function	244	58	4.21
Random Forest	190	46	4.13
Deep Learning	347	133	2.61

Venn Diagrams

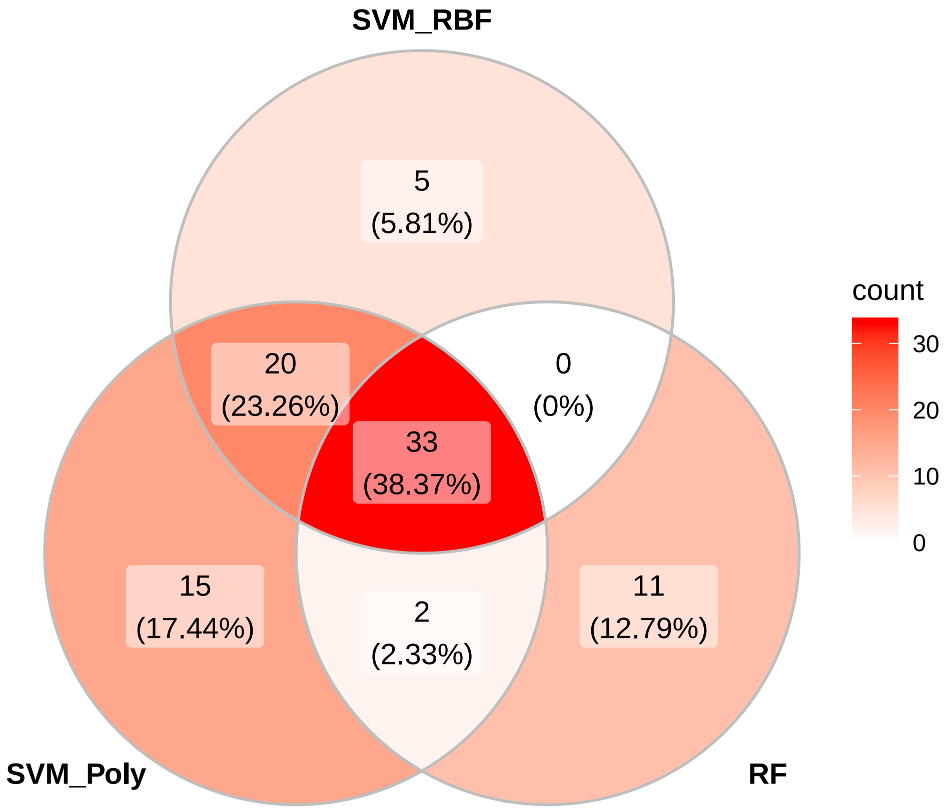
SVM_RBF ⋂ SVM_Poly ⋂ RF

This will be known as the **Round** set.

- RF ⋂ SVM_Poly ⋂ SVM_RBF = 33

```
round_set <- list(SVM_RBF = svm_svmRadial_fp_fn_nums,
                  SVM_Poly = svm_poly_fp_fn_nums,
                  RF = r_forest_fp_fn_nums)

ggVennDiagram(round_set)
```

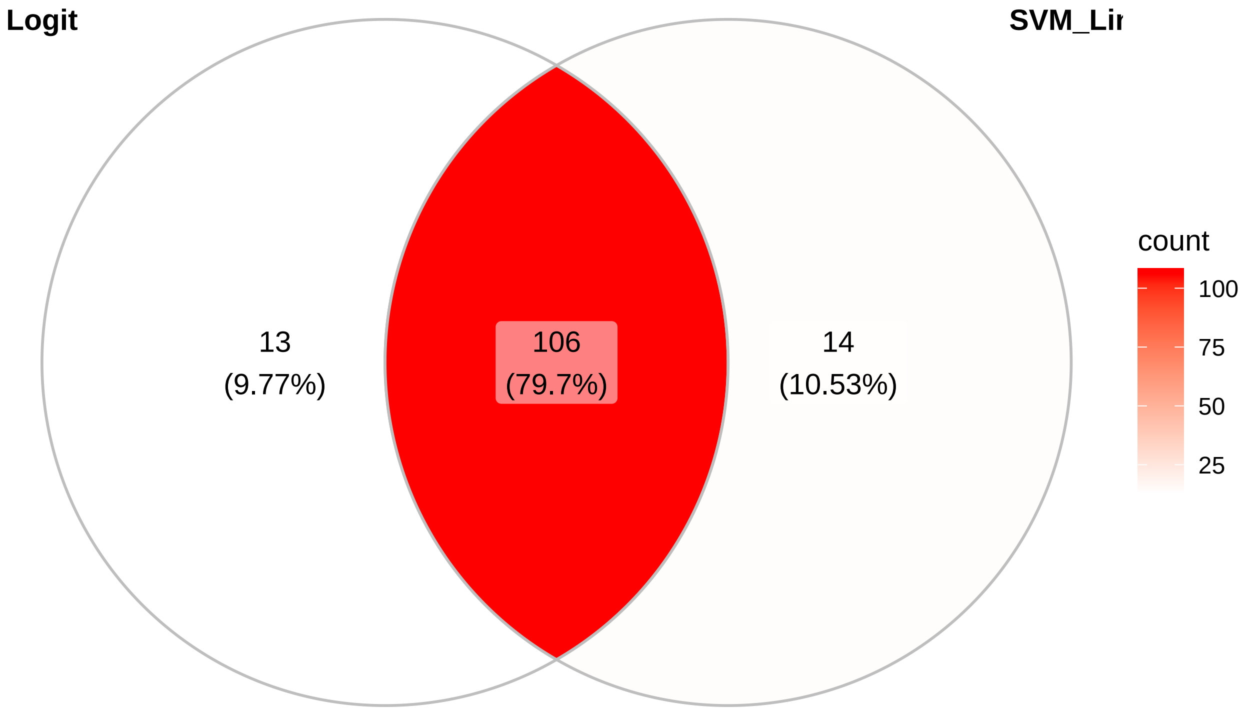


Logit ⋂ SVM_Lin

This will be known as the **Linear** set.

- Logit ⋂ SVM_Lin = 105

```
linear_set <- list(Logit = logit_fp_fn_nums,
                   SVM_Lin = svm_linear_fp_fn_nums)
ggVennDiagram(linear_set)
```



Find Intersecting numbers from *Round* Vs *Linear* sets

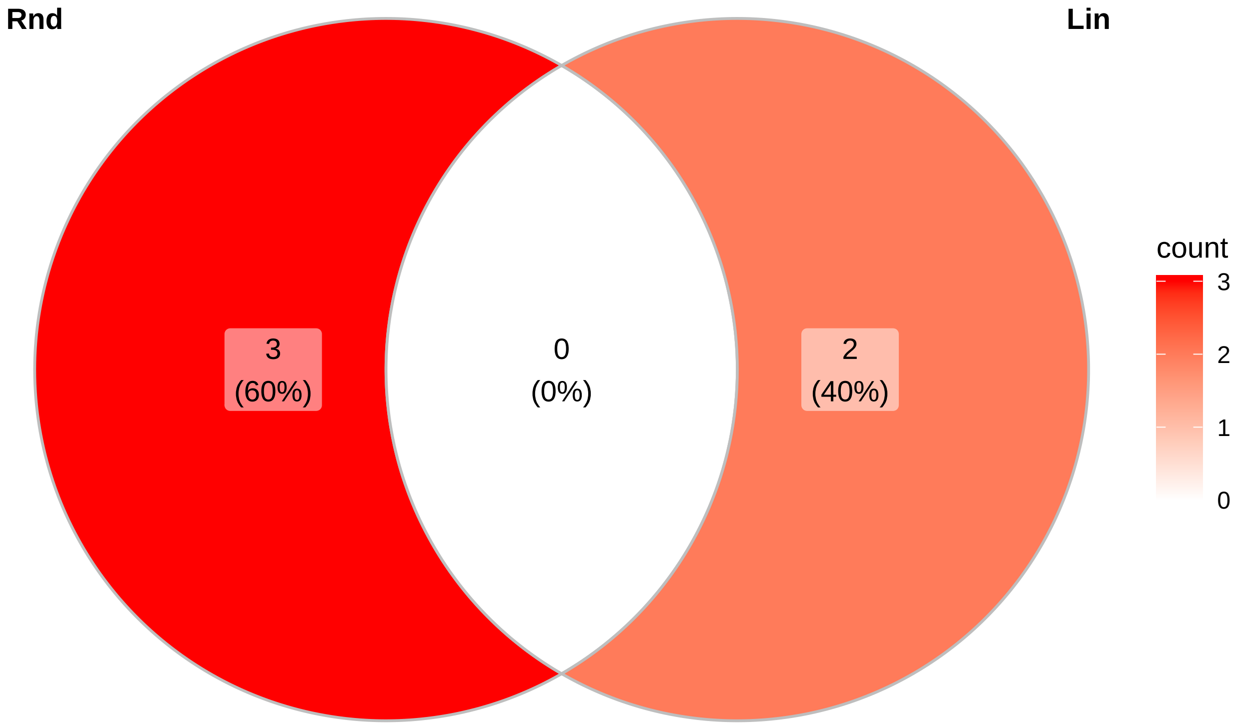
```
round_v_linear <- intersect(round_set, linear_set)
length(round_v_linear)
```

```
## [1] 0
```

```
round_v_linear
```

```
## list()
```

```
x <- list(Rnd = round_set,
          Lin = linear_set)
ggVennDiagram(x)
```



??????????????????????

Unique Round

```r
U_O_round_set <- unique(round_set)
U_O_round_set
```

```
## [[1]]
##  [1]    6   15   94  115  130  141  150  182  183  185  192  449  453  522  525
## [16]  526  529  531  534  542  546  566  568  570  580  582  592  655  913 1034
## [31] 1035 1091 1093 1094 1100 1101 1109 1121 1190 1219 1226 1233 1264 1471 1475
## [46] 1510 1575 1576 1579 1585 1587 1608 1618 1621 1766 1780 1831 1833
##
## [[2]]
##  [1]    6   15   94  115  130  136  141  150  182  183  185  445  449  452  453
## [16]  522  525  526  529  530  531  532  534  542  546  560  562  566  568  570
## [31]  579  580  582  592  912  913  980 1034 1035 1067 1091 1093 1100 1101 1109
## [46] 1121 1188 1190 1219 1226 1233 1264 1471 1510 1522 1575 1576 1579 1585 1587
## [61] 1608 1618 1621 1623 1697 1734 1773 1780 1831 1833
##
## [[3]]
##  [1]    6   57  100  130  141  150  183  453  526  534  542  570  573  580  980
## [16] 1033 1034 1035 1091 1092 1093 1100 1101 1219 1223 1226 1233 1264 1470 1471
## [31] 1510 1569 1575 1576 1579 1585 1587 1588 1594 1608 1618 1622 1623 1780 1831
## [46] 1833
```