Stat Assignment 2

STAT5003 Assignment 2

Loading and Subsetting the Labelled Data

```
#Read Data in
setwd("/Users/seanlolwut/Downloads/Final project/Datasets")
data <- read.csv('statdata.csv', header = T)

#See structure of data
str(data)
```

```
## 'data.frame':    12062 obs. of  30 variables:
##  $ Identifier         : Factor w/ 12062 levels "100043914;88",..: 1 2 3 4 5 6 7 8
 9 10 ...
##  $ GeneSymbol         : Factor w/ 3160 levels "100043914","1110018J18RIK",..: 1 1
 2 3 3 3 3 4 4 5 ...
##  $ PhosphorylationSite: int  88 89 18 133 138 1628 173 191 195 135 ...
##  $ Label              : Factor w/ 3 levels "AKT","mTOR","NULL": 3 3 3 3 3 3 3 3
 3 ...
##  $ Seq.Window         : Factor w/ 11983 levels "_____MSAGGDFG",..: 3438 8318 891 1
0609 4235 2733 1777 5129 499 8465 ...
##  $ LeftChain          : Factor w/ 11770 levels "_____M","____MA",..: 3372 8189 841
10409 4156 2670 1718 5042 450 8335 ...
##  $ Left               : Factor w/ 20 levels "A","C","D","E",..: 15 16 13 8 3 16 6
 12 16 9 ...
##  $ LeftGroup          : Factor w/ 5 levels "HydrophobicUncharged",..: 4 3 5 1 2 3
 5 3 3 4 ...
##  $ LeftCenter         : Factor w/ 57 levels "AS","AT","AY",..: 41 44 35 21 6 44 15
32 44 24 ...
##  $ Center             : Factor w/ 3 levels "S","T","Y": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CenterGroup        : Factor w/ 2 levels "HydrophobicUncharged",..: 2 2 2 2 2 2
 2 2 2 2 ...
##  $ CenterTrio         : Factor w/ 675 levels "ASA","ASC","ASD",..: 501 528 422 255
76 540 188 382 540 286 ...
##  $ CenterRight        : Factor w/ 58 levels "S_","SA","SC",..: 17 2 14 8 14 14 14
 13 14 9 ...
##  $ Right              : Factor w/ 21 levels "_","A","C","D",..: 17 2 14 8 14 14 14
13 14 9 ...
##  $ RightGroup         : Factor w/ 6 levels "BLANK","HydrophobicUncharged",..: 4 2
 6 5 6 6 6 4 6 2 ...
##  $ RightChain         : Factor w/ 11902 levels "ASAAGASP","ASAANGPL",..: 9127 9427
7144 4671 1632 10105 3655 6629 10285 5007 ...
##  $ Avg.Fold           : num  1.032 1.216 0.286 -0.41 -0.366 ...
##  $ AUC                : num  0.297 0.528 0.524 0.648 0.5 ...
##  $ X15s               : num  -0.186 0.108 0.703 -0.396 -0.396 ...
##  $ X30s               : num  0.5814 0.2802 -0.0905 -0.4971 -0.2954 ...
##  $ X1m                : num  1.8886 1.2472 0.0435 -0.3782 -0.1199 ...
##  $ X2m                : num  2.102 2.027 0.385 -0.608 -0.267 ...
##  $ X5m                : num  1.843 2.638 0.464 -0.434 -0.348 ...
##  $ X10m               : num  2.288 2.374 0.426 -0.445 -0.284 ...
##  $ X20m               : num  1.343 1.103 0.377 -0.265 -0.663 ...
##  $ X60m               : num  -1.6076 -0.0494 -0.0185 -0.2566 -0.5587 ...
##  $ Ins.1              : num  0.653 0.611 -0.199 0.174 0.229 ...
##  $ LY                 : num  0.3419 0.0462 -0.1478 -0.5707 -0.3814 ...
##  $ Ins.2              : num  2.157 3.096 -0.261 0.301 0.413 ...
##  $ MK                 : num  2.1379 3.2268 -0.0275 -0.1331 -0.1331 ...
```

```
#Subset labelled Data
AKTlabelled <- data[data$Label == 'AKT', ]
mTORlabelled <- data[data$Label == 'mTOR', ]

#Check dimensions of labelled data
dim(AKTlabelled)
```

```
## [1] 22 30
```

```
dim(mTORlabelled)
```

```
## [1] 26 30
```

Looking at Amino Acid Frequency for labelled AKT Data

```
#Frequency of center substrate for AKT
#Very Interesting that there are only two options, with 82% S
centerAKTpercent <- prop.table(table(AKTlabelled$Center))
centerAKTTotal <- table(AKTlabelled$Center)
rbind(centerAKTTotal, centerAKTpercent)
```

```
##                            S         T Y
## centerAKTTotal   18.0000000 4.0000000 0
## centerAKTpercent  0.8181818 0.1818182 0
```

```
centerGroupAktPercent <- sort(prop.table(round(table(AKTlabelled$CenterGroup), 2)), d
ecreasing = T)
centerGroupAktTotal <- sort(table(AKTlabelled$CenterGroup), decreasing = T)
rbind(centerGroupAktTotal, centerGroupAktPercent)
```

```
##                        PolarUncharged HydrophobicUncharged
## centerGroupAktTotal                22                    0
## centerGroupAktPercent               1                    0
```

```
#Frequency of left substrate for AKT
#Interestingly here that "S" is once again the most popular, it's likely that binding
 sites with an "S" close to the phosphorylation site are substrates of AKT (obviously
 in this case since it's labelled) but show a decrease in sensitivity to Insulin (For
 note -- if you put the key (enzyme) in the lock (substrate) it will fold to better f
it the key (kind of like a magnet), even if S isn't the exact phosphorylation site, i
f it's close it will still fold and lead to a reaction, probably just weaker.
leftAKTpercent <- sort( round( prop.table(table(AKTlabelled$Left)), 2), decreasing =
T)
leftAKTTotal <- sort(table(AKTlabelled$Left), decreasing = T)
rbind(leftAKTTotal, leftAKTpercent)
```

```
##                    S    N    F    H    T    Y    A    G    I    K    Q    R
## leftAKTTotal   4.00 3.00 2.00 2.00 2.00 2.00 1.00 1.00 1.00 1.00 1.00 1.00
## leftAKTpercent 0.18 0.14 0.09 0.09 0.09 0.09 0.05 0.05 0.05 0.05 0.05 0.05
##                    V C D E L M P W
## leftAKTTotal   1.00 0 0 0 0 0 0 0
## leftAKTpercent 0.05 0 0 0 0 0 0 0
```

```
leftGroupAktPercent <- sort(round(prop.table(table(AKTlabelled$LeftGroup)), 2), decre
asing = T)
leftGroupAktTotal <- sort(table(AKTlabelled$LeftGroup), decreasing = T)
rbind(leftGroupAktTotal, leftGroupAktPercent)
```

```
##                      PolarUncharged HydrophobicUncharged PositiveCharge
## leftGroupAktTotal             10.00                 7.00           4.00
## leftGroupAktPercent            0.45                 0.32           0.18
##                      Special NegativeCharge
## leftGroupAktTotal       1.00              0
## leftGroupAktPercent     0.05              0
```

```
#Frequency of right substrate for AKT
#Worth noting that "S"" isn't frequent, and "F" was top 3 in both.
rightAKTpercent <- sort( round(prop.table(table(AKTlabelled$Right)),2) , decreasing =
 T)
rightAKTTotal <- sort(table(AKTlabelled$Right), decreasing = T)
rbind(rightAKTTotal, rightAKTpercent)
```

```
##                      F    A    V    L    M    E    I    N    Q    S    T
## rightAKTTotal     5.00 3.00 3.00 2.00 2.00 1.00 1.00 1.00 1.00 1.00 1.00
## rightAKTpercent 0.23 0.14 0.14 0.09 0.09 0.05 0.05 0.05 0.05 0.05 0.05
##                      W _ C D G H K P R Y
## rightAKTTotal     1.00 0 0 0 0 0 0 0 0 0
## rightAKTpercent 0.05 0 0 0 0 0 0 0 0 0
```

```
rightGroupAKTpercent <- sort( round(prop.table(table(AKTlabelled$RightGroup)),2) , de
creasing = T)
rightGroupAKTTotal <- sort(table(AKTlabelled$RightGroup), decreasing = T)
rbind(rightGroupAKTTotal, rightGroupAKTpercent)
```

```
##                       HydrophobicUncharged PolarUncharged NegativeCharge
## rightGroupAKTTotal                   17.00           4.00           1.00
## rightGroupAKTpercent                  0.77           0.18           0.05
##                       BLANK PositiveCharge Special
## rightGroupAKTTotal        0              0       0
## rightGroupAKTpercent      0              0       0
```

Looking at Amino Acid Frequency for labelled mTOR Data

```
#Frequency of center substrate for mTOR
#"S" and "T" are once again more likely, with slightly less favour towards "S"
centermTORpercent <- prop.table(table(mTORlabelled$Center))
centermTORTotal <- table(mTORlabelled$Center)
rbind(centermTORTotal, centermTORpercent)
```

```
##                             S         T Y
## centermTORTotal   20.0000000 6.0000000 0
## centermTORpercent  0.7692308 0.2307692 0
```

```
centerGroupmTORpercent <- sort(prop.table(table(mTORlabelled$CenterGroup)), decreasin
g = T)
centerGroupmTORTotal <- sort(table(mTORlabelled$CenterGroup), decreasing = T)
rbind(centerGroupmTORTotal, centerGroupmTORpercent)
```

```
##                          PolarUncharged HydrophobicUncharged
## centerGroupmTORTotal                 26                    0
## centerGroupmTORpercent                1                    0
```

```
#Frequency of left substrate for mTOR
#"S" and "T" are once again more likely, with slightly less favour towards "S"
leftmTORpercent <- sort( round( prop.table(table(mTORlabelled$Left)), 2), decreasing
= T)
leftmTORTotal <- sort(table(mTORlabelled$Left), decreasing = T)
rbind(leftmTORTotal, leftmTORpercent)
```

```
##                     G    S    T    F    L    M    A    H    K    N    P
## leftmTORTotal    5.00 4.00 3.00 2.00 2.00 2.00 1.00 1.00 1.00 1.00 1.00
## leftmTORpercent 0.19 0.15 0.12 0.08 0.08 0.08 0.04 0.04 0.04 0.04 0.04
##                     Q    R    V C D E I W Y
## leftmTORTotal    1.00 1.00 1.00 0 0 0 0 0 0
## leftmTORpercent 0.04 0.04 0.04 0 0 0 0 0 0
```

```
leftGroupmTORpercent <- sort( round( prop.table(table(mTORlabelled$LeftGroup)), 2), d
ecreasing = T)
leftGroupmTORTotal <- sort(table(mTORlabelled$LeftGroup), decreasing = T)
rbind(leftGroupmTORTotal, leftGroupmTORpercent)
```

```
##                      PolarUncharged HydrophobicUncharged Special
## leftGroupmTORTotal             9.00                 8.00    6.00
## leftGroupmTORpercent           0.35                 0.31    0.23
##                      PositiveCharge NegativeCharge
## leftGroupmTORTotal             3.00              0
## leftGroupmTORpercent           0.12              0
```

```
#Frequency of right substrate for mTOR
#"P" is very frequent, more so than either "S" or "T" in the middle, a key find.
rightmTORpercent <- sort( round(prop.table(table(mTORlabelled$Right)),2) , decreasing
 = T)
rightmTORTotal <- sort(table(mTORlabelled$Right), decreasing = T)
rbind(rightmTORTotal, rightmTORpercent)
```

```
##                     P    F    L _ A C D E G H I K M N Q R S T V W Y
## rightmTORTotal   23.00 2.00 1.00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## rightmTORpercent  0.88 0.08 0.04 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

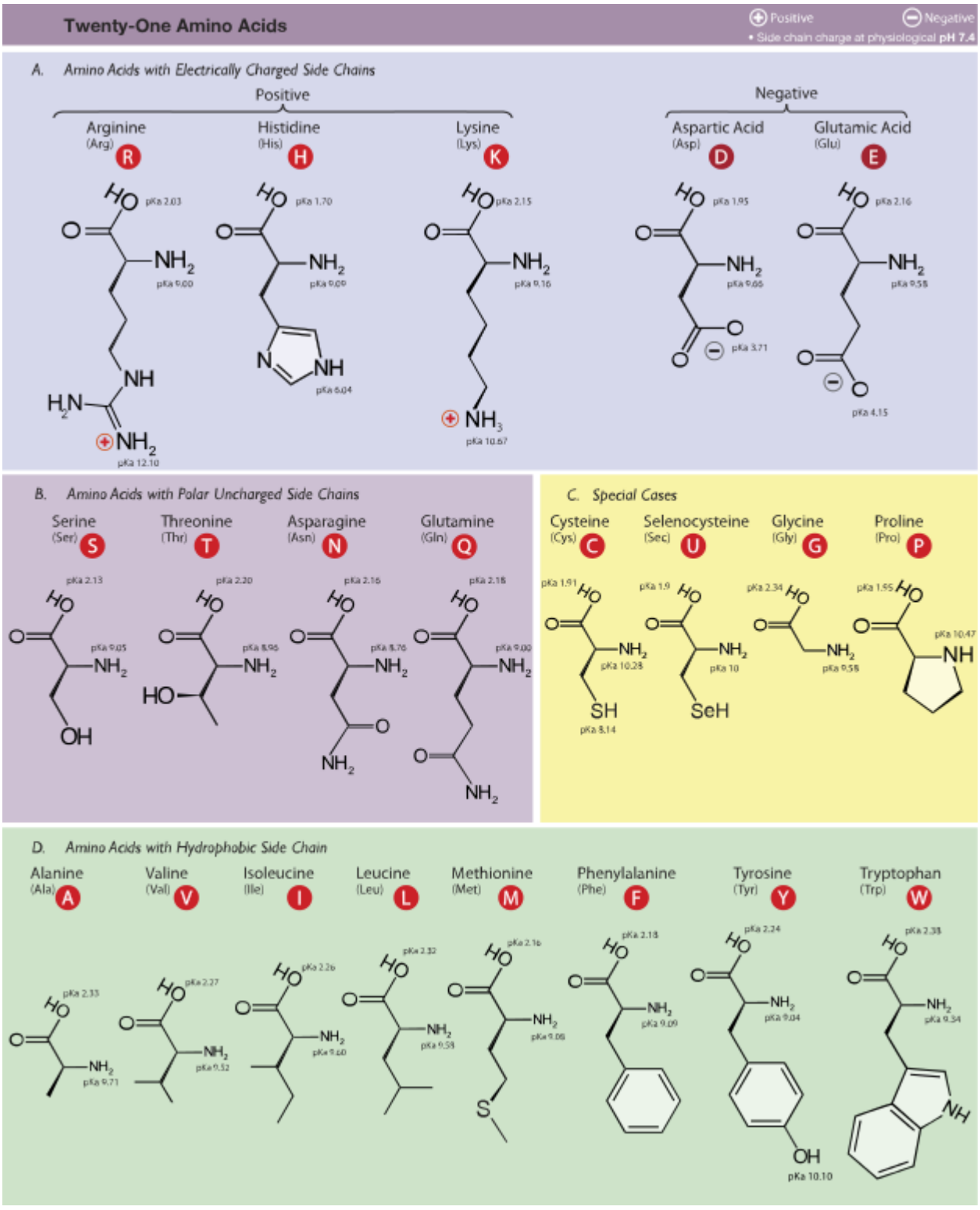
```
rightGroupmTORpercent <- sort( round(prop.table(table(mTORlabelled$RightGroup)),2) ,
decreasing = T)
rightGroupmTORTotal <- sort(table(mTORlabelled$RightGroup), decreasing = T)
rbind(rightGroupmTORTotal, rightGroupmTORpercent)
```

```
##                         Special HydrophobicUncharged BLANK NegativeCharge
## rightGroupmTORTotal       23.00                 3.00     0              0
## rightGroupmTORpercent      0.88                 0.12     0              0
##                         PolarUncharged PositiveCharge
## rightGroupmTORTotal                  0              0
## rightGroupmTORpercent                0              0
```

After realising that “S” and “T” are frequently in the middle of the amino acid sequence, it’s worth noting that amino acids can be split into three groups. These groups are important as the amino acids in these groups will exhibit the same folding behavious.

For example when you’re cleaning an oily mess, you don’t use water… Yeah pretty sure we all know this.

Point being, we can group these amino acids into groups, this should reduce the variance and improve our ability to classify the data.



Caption for the picture.

Correlation between different variables

```
library(corrgram)
#Somewhat negative correlation betwen average fold and AUC for AKT
cor(AKTlabelled$Avg.Fold, AKTlabelled$AUC)
```

```
## [1] -0.5823193
```

```
cor(AKTlabelled$LY, AKTlabelled$AUC)
```

```
## [1] -0.3989324
```

```
cor(AKTlabelled$MK, AKTlabelled$AUC)
```

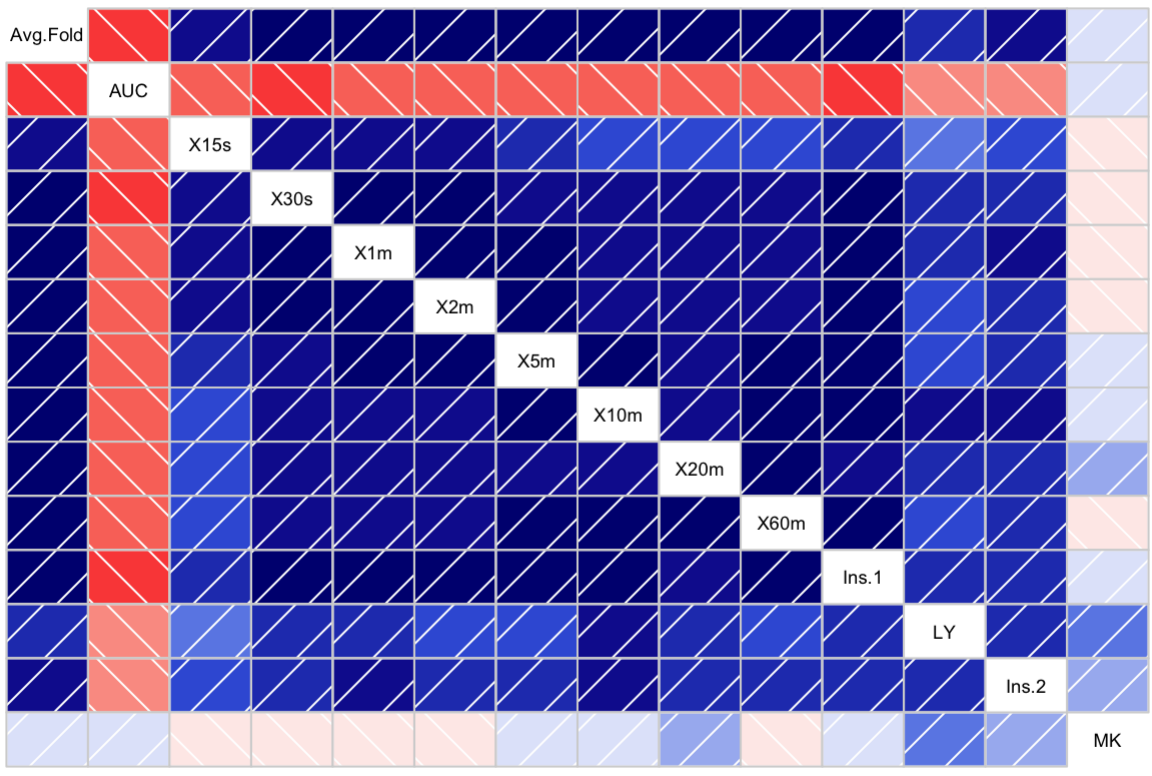
```
## [1] 0.05610034
```

```
#No correlation between fold and AUC for mTOR.
cor(mTORlabelled$Avg.Fold, mTORlabelled$AUC)
```

```
## [1] -0.003026305
```

```
aktNumeric <- AKTlabelled[,17:30]

corrgram(aktNumeric)
```



Test Model on AKT vs mTOR labelled data only, 90% Cross Validation

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(e1071)
set.seed(24)

#combine both AKT and mTOR labelled datasets
labelledData <- data[data$Label != 'NULL', ]

#removing the centergroup field as it only has one variable
#removing other variables as there are too many levels and errors are thrown with new
 variables
columnsToUse <- which(colnames(labelledData) %in% c("Label", "LeftGroup", "Center",
"RightGroup"))
columnsToUse <- c(columnsToUse, 17:30)
labelledData <- labelledData[, columnsToUse]

str(labelledData)
```

```
## 'data.frame':    48 obs. of  18 variables:
##  $ Label     : Factor w/ 3 levels "AKT","mTOR","NULL": 2 2 1 2 2 1 1 1 1 1 ...
##  $ LeftGroup : Factor w/ 5 levels "HydrophobicUncharged",..: 3 4 3 3 1 3 5 4 4 1
 ...
##  $ Center    : Factor w/ 3 levels "S","T","Y": 1 1 2 1 1 1 1 2 1 1 ...
##  $ RightGroup: Factor w/ 6 levels "BLANK","HydrophobicUncharged",..: 6 2 4 6 6 2 2
2 2 3 ...
##  $ Avg.Fold  : num  0.426 0.99 3.506 0.649 0.552 ...
##  $ AUC       : num  0.657 0.436 0.203 0.536 0.523 ...
##  $ X15s      : num  -0.17605 -0.00727 1.79325 0.00931 -0.42623 ...
##  $ X30s      : num  -0.2174 0.16 2.9496 -0.0156 -0.9644 ...
##  $ X1m       : num  -0.335 0.547 3.693 0.207 -0.194 ...
##  $ X2m       : num  -0.445 1.076 3.641 0.341 -0.24 ...
##  $ X5m       : num  0.255 1.504 4.085 1.006 1.399 ...
##  $ X10m      : num  1.05 1.5 3.9 1.26 1.45 ...
##  $ X20m      : num  1.51 1.59 3.98 1.2 2.04 ...
##  $ X60m      : num  1.77 1.55 4.01 1.19 1.35 ...
##  $ Ins.1     : num  1.76 1.23 2.44 1.03 1.84 ...
##  $ LY        : num  1.224 -1.482 0.902 0.192 1.12 ...
##  $ Ins.2     : num  2.177 1.37 2.615 0.385 1.995 ...
##  $ MK        : num  1.728 -0.344 -0.85 0.335 2.11 ...
```

```
#create 80% cross validation
indexPartition <- createDataPartition(y = labelledData$Label, p = 0.8)[[1]]
```

```
## Warning in createDataPartition(y = labelledData$Label, p = 0.8): Some
## classes have no records ( NULL ) and these will be ignored
```

```
#need to be numeric
TrainingLabelled <- as.data.frame((lapply(labelledData[indexPartition, ],
as.numeric)))
TestLabelled <- as.data.frame((lapply(labelledData[-indexPartition, ], as.numeric)))

#linear SVM model
svm.labelled.model <- svm(x = TrainingLabelled[ , -1], y = TrainingLabelled[,1], kern
el = 'linear', type = 'C-classification', scale = F)

#prediction on test data
svm.decisions <- predict(svm.labelled.model, TestLabelled[,-1])
table(svm.decisions, TestLabelled[,1])
```

```
##
## svm.decisions 1 2
##             1 4 1
##             2 0 4
```

Random Down-Sampling 100 Models

```
set.seed(24)

#create null list
sampleLists <- list()

#Create two new columns for one vs all
data$AKT = as.factor( data$Label == "AKT")
data$mTOR = as.factor( data$Label == 'mTOR')

#For loop to create 100 downsamples
for (i in 1:100) {

  down.Sample <- downSample( x = data[, c(-1, -4,-32)], data[ ,31], list = T, yname =
 'AKT')

  sampleLists[[i]] <- down.Sample$x

}

#build empty list of models
list.model <- list()

for (j in 1:100) {

#Convert the dataframe to numeric so SVM can be used
sampleLists[[j]] <- as.data.frame( lapply(sampleLists[[j]], as.numeric))

#Create model using positive label vs unlabelled.
svm.model.akt <- svm(x = sampleLists[[j]][ , -29], y = sampleLists[[j]][ , 29], kerne
l = 'linear', type = 'C-classification', scale = F)

list.model[[j]] <- svm.model.akt

}

#remove the columns for the full dataset
dataTest <- data[, c(-1, -4,-32)]
#convert variables to numeric for SVM
dataTest <- as.data.frame(lapply(dataTest, as.numeric))

#Initialise empty dataframe
total.decisions <- data.frame()

for (k in 1:100) {

  test.decisons <- predict(list.model[[k]], dataTest[, -29])

  total.decisions <- rbind(total.decisions, test.decisons)
  total.decisions <- as.data.frame(total.decisions)

}

#find the mean decision for each observation and classify
final.decisions <- colMeans(total.decisions) >= 1.5
#final results
table(final.decisions)
```

```
## final.decisions
## FALSE  TRUE
## 11238   824
```