

Localización subcelular de proteínas tipo I

Marc Bañuls Tornero

31/12/2019

Contents

Análisis de los datos y preparación general inicial	1
k-Nearest Neighbour	7
1) Transformación de los datos	7
2) Entrenar el modelo	8
3) Evaluar y mejorar el rendimiento del modelo	8
Naive Bayes	10
1) Transformación de los datos	10
2) Entrenar el modelo	11
3) Evaluar y mejorar el rendimiento del modelo	11
Artificial Neural Network	14
1) Transformación de los datos	14
2) Entrenar el modelo	15
3) Evaluar y mejorar el rendimiento del modelo	15
Support Vector Machine	18
1) Transformación de los datos	19
2) Entrenar el modelo	19
3) Evaluar y mejorar el rendimiento del modelo	19
Arbol de Decisión	21
1) Transformación de los datos	21
2) Entrenar el modelo	21
3) Evaluar y mejorar el rendimiento del modelo	26
Conclusiones	61

Importamos la tabla de datos `yeast.data` de manera que esté correctamente dividida en columnas:

```
datos<-read.table("yeast.data", header = FALSE)
```

Análisis de los datos y preparación general inicial

Aunque en el enunciado de la PEC ya se han explicado los datos a tratar, realizaremos un análisis descriptivo de los datos:

```
str(datos)
```

```
## 'data.frame':   1484 obs. of  10 variables:
## $ V1 : Factor w/ 1462 levels "6P2K_YEAST","6PGD_YEAST",...: 33 34 35 3 5 4 6 101 7 8 ...
## $ V2 : num  0.58 0.43 0.64 0.58 0.42 0.51 0.5 0.48 0.55 0.4 ...
## $ V3 : num  0.61 0.67 0.62 0.44 0.44 0.4 0.54 0.45 0.5 0.39 ...
## $ V4 : num  0.47 0.48 0.49 0.57 0.48 0.56 0.48 0.59 0.66 0.6 ...
## $ V5 : num  0.13 0.27 0.15 0.13 0.54 0.17 0.65 0.2 0.36 0.15 ...
## $ V6 : num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
```

```
## $ V7 : num 0 0 0 0 0 0.5 0 0 0 0 ...
## $ V8 : num 0.48 0.53 0.53 0.54 0.48 0.49 0.53 0.58 0.49 0.58 ...
## $ V9 : num 0.22 0.22 0.22 0.22 0.22 0.22 0.22 0.34 0.22 0.3 ...
## $ V10: Factor w/ 10 levels "CYT","ERL","EXC",...: 7 7 7 8 7 1 7 8 7 1 ...
```

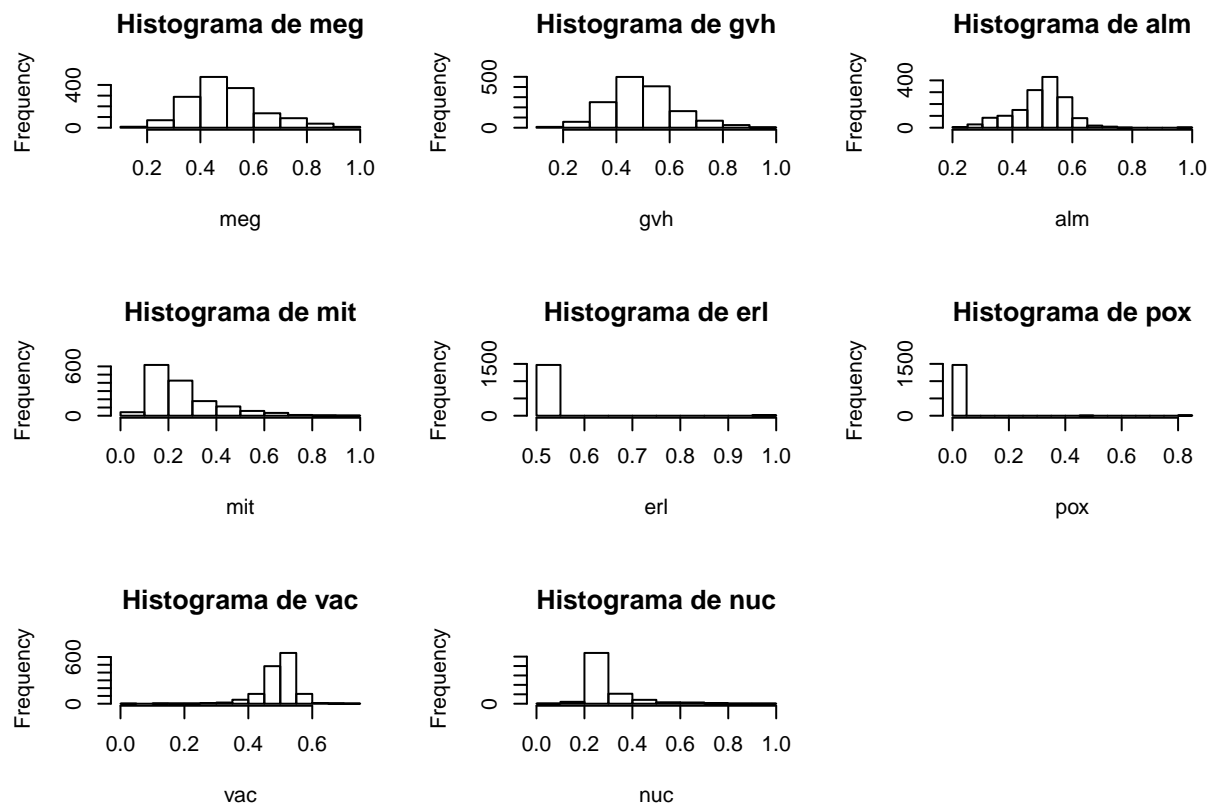
```
summary(datos)
```

```
##           V1           V2           V3           V4
## EF1A_YEAST: 2   Min.   :0.1100   Min.   :0.1300   Min.   :0.21
## H3_YEAST   : 2   1st Qu.:0.4100   1st Qu.:0.4200   1st Qu.:0.46
## H4_YEAST   : 2   Median :0.4900   Median :0.4900   Median :0.51
## IF4A_YEAST: 2   Mean    :0.5001   Mean    :0.4999   Mean    :0.50
## MAT2_YEAST: 2   3rd Qu.:0.5800   3rd Qu.:0.5700   3rd Qu.:0.55
## MTC_YEAST : 2   Max.    :1.0000   Max.    :1.0000   Max.    :1.00
## (Other)    :1472
##           V5           V6           V7           V8
## Min.   :0.0000   Min.   :0.5000   Min.   :0.0000   Min.   :0.0000
## 1st Qu.:0.1700   1st Qu.:0.5000   1st Qu.:0.0000   1st Qu.:0.4800
## Median :0.2200   Median :0.5000   Median :0.0000   Median :0.5100
## Mean    :0.2612   Mean    :0.5047   Mean    :0.0075   Mean    :0.4999
## 3rd Qu.:0.3200   3rd Qu.:0.5000   3rd Qu.:0.0000   3rd Qu.:0.5300
## Max.    :1.0000   Max.    :1.0000   Max.    :0.8300   Max.    :0.7300
##
##           V9           V10
## Min.   :0.0000   CYT    :463
## 1st Qu.:0.2200   NUC    :429
## Median :0.2200   MIT    :244
## Mean    :0.2762   ME3    :163
## 3rd Qu.:0.3000   ME2    : 51
## Max.    :1.0000   ME1    : 44
##                (Other): 90
```

Como podemos observar, nos encontramos con una tabla de datos de 10 columnas y 1484 registros. La primera columna sirve como identificador de la secuencia conteniendo el nombre definido en la base de datos SWISS-PROT, siendo por ello por lo que se trata de un factor. Los valores desde la columna 2 a la columna 9 se tratan de características de la secuencia analizada que pueden ayudar a determinar la localización subcelular de la secuencia proteica. Cada uno de estos vectores es de tipo numérico y observamos que ya están todos ellos normalizados (los valores se encuentran siempre entre 0 y 1). Finalmente en la última columna tenemos el lugar en el que se encuentra la secuencia determinada.

Para observar en que rango de valores se encuentran las distintas variables, podemos realizar un histograma de los datos:

```
par(mfrow=c(3,3))
hist(datos$V2, main = "Histograma de meg", xlab = "meg")
hist(datos$V3, main = "Histograma de gvh", xlab = "gvh")
hist(datos$V4, main = "Histograma de alm", xlab = "alm")
hist(datos$V5, main = "Histograma de mit", xlab = "mit")
hist(datos$V6, main = "Histograma de erl", xlab = "erl")
hist(datos$V7, main = "Histograma de pox", xlab = "pox")
hist(datos$V8, main = "Histograma de vac", xlab = "vac")
hist(datos$V9, main = "Histograma de nuc", xlab = "nuc")
```



Con estos histogramas observamos que los valores de meg y gvh son muy similares entre ellos, tendiendo a tener valores de 0.5 aunque se encuentran dispersos entre 0.3 y 0.7. En la variable alm observamos que los valores también se acumulan mayoritariamente en 0.5 aunque aquí no hay valores que superen el límite de 0.8. En el histograma de la variable mit observamos que la mayoría de las observaciones se encuentran en valores entre 0.2 y 0.3. En cambio, en la variable erl todos los valores se acumulan en 0.5 y en la variable pox todos los valores se acumulan en 0. En el histograma de vac observamos que esta variable tiende a estar en valores entre 0.4 y 0.6. Finalmente el histograma de nuc indica que sus valores suelen rondar por 0.2.

Además, otro análisis que podemos hacer es la observación de una posible correlación entre estas variables, pudiendo hacer un análisis rápido con la correlación de pearson. De esta manera, podemos deducir que valores pueden afectar a otros a la hora de saber la localización de la secuencia.

```
cor(datos[, -c(1,10)])
```

```
##          V2          V3          V4          V5          V6
## V2  1.000000000  0.5816314195 -0.163951300  0.158175472  0.064921537
## V3  0.581631420  1.0000000000 -0.271800035  0.140313937  0.060823302
## V4 -0.163951300 -0.2718000351  1.000000000  0.059668297 -0.008083462
## V5  0.158175472  0.1403139371  0.059668297  1.000000000 -0.005930705
## V6  0.064921537  0.0608233024 -0.008083462 -0.005930705  1.000000000
## V7  0.005596973  0.0003918342  0.009377914 -0.009039816 -0.009674234
## V8  0.075042684  0.0887593904 -0.185805390 -0.103591413  0.043626910
## V9 -0.124540384 -0.1029839928 -0.022042837 -0.054796535  0.002829256
##          V7          V8          V9
## V2  0.0055969730  0.07504268 -0.124540384
## V3  0.0003918342  0.08875939 -0.102983993
## V4  0.0093779144 -0.18580539 -0.022042837
## V5 -0.0090398164 -0.10359141 -0.054796535
```

```
## V6 -0.0096742336  0.04362691  0.002829256
## V7  1.0000000000  0.02089969 -0.035658650
## V8  0.0208996906  1.00000000  0.089690412
## V9 -0.0356586498  0.08969041  1.000000000
```

Analizando los resultados de la correlación en todas las variables solo observamos una leve correlación que puede ser significativa entre estas variables. Concretamente hablando, se trata de que hay una correlación positiva de 0.58 entre la variable meg y la variable gvh. Respecto a las demás variables, no parece haber correlaciones significativas.

Hay un par de factores a tratar antes de poder tener los datos preparados para los procesamientos iniciales de cada algoritmo. El primero de ellos es que observando el primer vector (el identificador) nos encontramos que el factor de este vector tiene menos niveles que número de registros existen. Por lo tanto podemos llegar a pensar que hay secuencias repetidas en esta base de datos, por lo que vamos a comprobar esto:

```
length(unique(datos$V1)) == nrow(datos)
```

```
## [1] FALSE
```

Esto indica que no todos los identificadores son únicos, y por lo tanto sí que hay algún identificador repetido. Si encontramos que sus características no difieren entre ellos, deberemos eliminar estos registros repetidos para evitar que los algoritmos utilicen datos repetidos, que pueden modificar el comportamiento de éstos.

```
frecuencia <- data.frame(table(datos$V1))
repetidos <- frecuencia[frecuencia$Freq > 1,]
repetidos
```

```
##      Var1 Freq
## 294 EF1A_YEAST  2
## 409   H3_YEAST  2
## 410   H4_YEAST  2
## 471 IF4A_YEAST  2
## 575 MAT2_YEAST  2
## 641  MTC_YEAST  2
## 903 RL12_YEAST  2
## 905 RL15_YEAST  2
## 908 RL18_YEAST  2
## 909 RL19_YEAST  2
## 910 RL1A_YEAST  2
## 911  RL2_YEAST  2
## 918 RL35_YEAST  2
## 923 RL41_YEAST  2
## 924 RL44_YEAST  2
## 936 RLUB_YEAST  2
## 998 RS22_YEAST  2
## 999 RS24_YEAST  2
## 1000 RS28_YEAST  2
## 1008 RS41_YEAST  2
## 1009 RS4E_YEAST  2
## 1011  RS8_YEAST  2
```

Podemos observar con un par de valores repetidos si sus características se repiten o no. Para ello recogemos los dos primeros valores repetidos obtenidos en la variable `repetidos`:

```
EF1A <- subset(datos, datos$V1 == "EF1A_YEAST")
H3 <- subset(datos, datos$V1 == "H3_YEAST")
str(EF1A)
```

```
## 'data.frame': 2 obs. of 10 variables:
## $ V1 : Factor w/ 1462 levels "6P2K_YEAST","6PGD_YEAST",...: 294 294
## $ V2 : num 0.63 0.63
## $ V3 : num 0.43 0.43
## $ V4 : num 0.48 0.48
## $ V5 : num 0.11 0.11
## $ V6 : num 0.5 0.5
## $ V7 : num 0 0
## $ V8 : num 0.51 0.51
## $ V9 : num 0.22 0.22
## $ V10: Factor w/ 10 levels "CYT","ERL","EXC",...: 1 1
```

```
str(H3)
```

```
## 'data.frame': 2 obs. of 10 variables:
## $ V1 : Factor w/ 1462 levels "6P2K_YEAST","6PGD_YEAST",...: 409 409
## $ V2 : num 0.39 0.39
## $ V3 : num 0.39 0.39
## $ V4 : num 0.55 0.55
## $ V5 : num 0.52 0.52
## $ V6 : num 0.5 0.5
## $ V7 : num 0 0
## $ V8 : num 0.37 0.37
## $ V9 : num 0.16 0.16
## $ V10: Factor w/ 10 levels "CYT","ERL","EXC",...: 8 8
```

Habiendo realizado la función `str()` de ambos identificadores observamos que son exactamente iguales entre ellos, así que podemos suponer que todos los valores repetidos no nos serán de utilidad a la hora de crear posteriores modelos. Por ello, utilizaremos a partir de ahora una tabla con valores de identificador únicos.

```
datos_u <- datos[!duplicated(datos$V1),]
```

Ahora que tenemos valores únicos, podemos modificar el vector de clasificación como se pide en el enunciado de la PEC. Para ello filtraremos las clases (y sus respectivos registros) que no se requieran.

Primero filtraremos por los clasificadores que queremos mantener:

```
datos_final <- subset(datos_u, datos_u$V10 == "CYT" | datos_u$V10 == "ME1" | datos_u$V10 == "ME2" | datos_u$V10 == "ME3" | datos_u$V10 == "MIT" | datos_u$V10 == "NUC")
str(datos_final)
```

```
## 'data.frame': 1372 obs. of 10 variables:
## $ V1 : Factor w/ 1462 levels "6P2K_YEAST","6PGD_YEAST",...: 33 34 35 3 5 4 6 101 7 8 ...
## $ V2 : num 0.58 0.43 0.64 0.58 0.42 0.51 0.5 0.48 0.55 0.4 ...
## $ V3 : num 0.61 0.67 0.62 0.44 0.44 0.4 0.54 0.45 0.5 0.39 ...
## $ V4 : num 0.47 0.48 0.49 0.57 0.48 0.56 0.48 0.59 0.66 0.6 ...
## $ V5 : num 0.13 0.27 0.15 0.13 0.54 0.17 0.65 0.2 0.36 0.15 ...
## $ V6 : num 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
## $ V7 : num 0 0 0 0 0 0.5 0 0 0 0 ...
## $ V8 : num 0.48 0.53 0.53 0.54 0.48 0.49 0.53 0.58 0.49 0.58 ...
## $ V9 : num 0.22 0.22 0.22 0.22 0.22 0.22 0.22 0.34 0.22 0.3 ...
## $ V10: Factor w/ 10 levels "CYT","ERL","EXC",...: 7 7 7 8 7 1 7 8 7 1 ...
```

Ahora combinaremos los clasificadores ME1, ME2 y ME3 en el clasificador MEM:

```
datos_final$V10 <- factor(datos_final$V10, levels = c("CYT", "ME1", "ME2", "ME3", "MIT", "NUC"), labels = c("CYT", "ME", "ME", "MEM", "MIT", "NUC"))
str(datos_final)
```

```
## 'data.frame': 1372 obs. of 10 variables:
## $ V1 : Factor w/ 1462 levels "6P2K_YEAST","6PGD_YEAST",...: 33 34 35 3 5 4 6 101 7 8 ...
```

```
## $ V2 : num 0.58 0.43 0.64 0.58 0.42 0.51 0.5 0.48 0.55 0.4 ...
## $ V3 : num 0.61 0.67 0.62 0.44 0.44 0.4 0.54 0.45 0.5 0.39 ...
## $ V4 : num 0.47 0.48 0.49 0.57 0.48 0.56 0.48 0.59 0.66 0.6 ...
## $ V5 : num 0.13 0.27 0.15 0.13 0.54 0.17 0.65 0.2 0.36 0.15 ...
## $ V6 : num 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
## $ V7 : num 0 0 0 0 0 0.5 0 0 0 0 ...
## $ V8 : num 0.48 0.53 0.53 0.54 0.48 0.49 0.53 0.58 0.49 0.58 ...
## $ V9 : num 0.22 0.22 0.22 0.22 0.22 0.22 0.22 0.34 0.22 0.3 ...
## $ V10: Factor w/ 4 levels "CYT","MEM","MIT",...: 3 3 3 4 3 1 3 4 3 1 ...
```

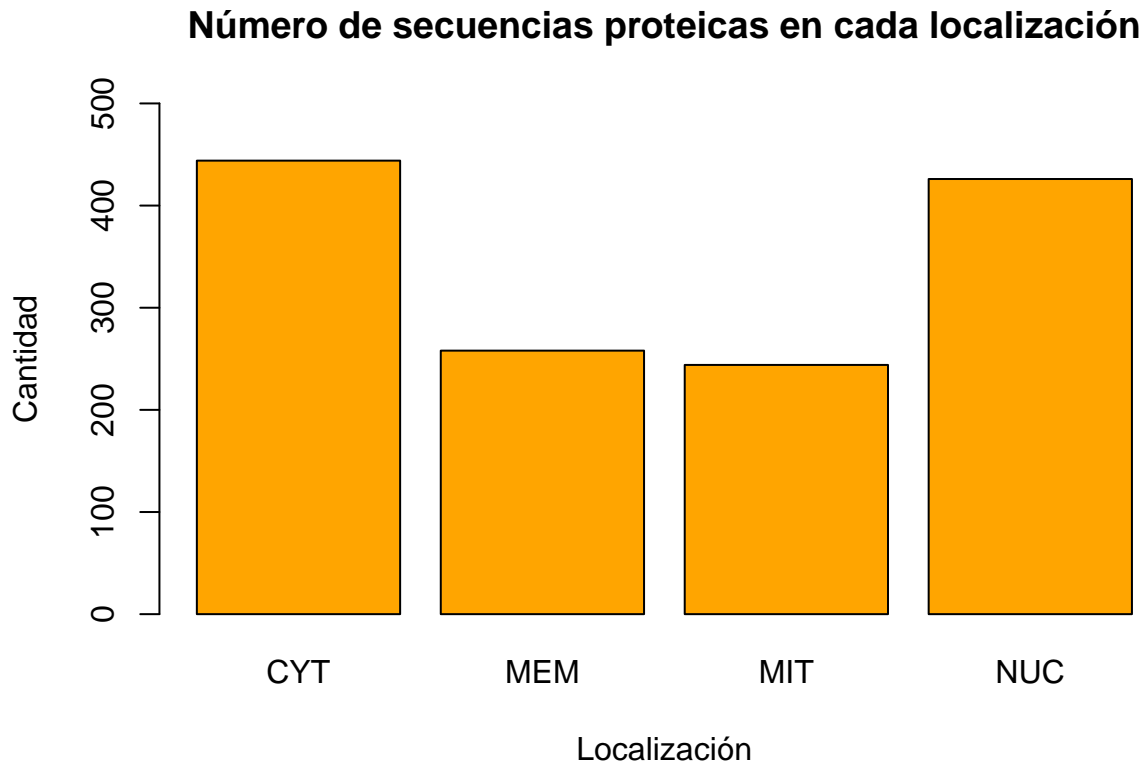
```
summary(datos_final)
```

```
##           V1           V2           V3           V4
## 6P2K_YEAST: 1   Min.   :0.1100   Min.   :0.1300   Min.   :0.2100
## 6PGD_YEAST: 1   1st Qu.:0.4000   1st Qu.:0.4200   1st Qu.:0.4500
## AAR2_YEAST: 1   Median :0.4800   Median :0.4900   Median :0.5100
## AATC_YEAST: 1   Mean    :0.4922   Mean    :0.4939   Mean    :0.4994
## AATM_YEAST: 1   3rd Qu.:0.5700   3rd Qu.:0.5600   3rd Qu.:0.5500
## ABC1_YEAST: 1   Max.    :1.0000   Max.    :1.0000   Max.    :1.0000
## (Other)    :1366
##           V5           V6           V7           V8
## Min.   :0.0000   Min.   :0.5000   Min.   :0.000000   Min.   :0.0000
## 1st Qu.:0.1600   1st Qu.:0.5000   1st Qu.:0.000000   1st Qu.:0.4800
## Median :0.2200   Median :0.5000   Median :0.000000   Median :0.5100
## Mean    :0.2611   Mean    :0.5033   Mean    :0.001939   Mean    :0.5013
## 3rd Qu.:0.3200   3rd Qu.:0.5000   3rd Qu.:0.000000   3rd Qu.:0.5300
## Max.    :1.0000   Max.    :1.0000   Max.    :0.830000   Max.    :0.7300
##
##           V9           V10
## Min.   :0.0000   CYT:444
## 1st Qu.:0.2200   MEM:258
## Median :0.2200   MIT:244
## Mean    :0.2798   NUC:426
## 3rd Qu.:0.3100
## Max.    :1.0000
##
```

Utilizando las funciones `summary()` y `str()` en los datos preparados observamos que el vector de clasificación (V10) tiene ahora las 4 etiquetas que se necesitaban, con los valores de ME1, ME2 y ME3 en la misma etiqueta MEM.

Ahora podemos realizar unos gráficos y tablas para comprender mejor la distribución de los datos. Primero realizaremos un histograma de las clasificaciones para observar como de equilibrados estan los registros:

```
plot(datos_final$V10, ylim = range(0,500), col = "orange", xlab = "Localización", ylab = "Cantidad", ma
```



Para observar esto en formato de tabla y en porcentajes:

```
prop.table(table(datos_final$V10))
```

```
##  
##      CYT      MEM      MIT      NUC  
## 0.3236152 0.1880466 0.1778426 0.3104956
```

Observamos que hay una mayor representación de las secuencias proteicas del citoplasma y núcleo (32.3% y 31% respectivamente) en comparación a la membrana y mitocondria (18.8% y 17.7% respectivamente). Podemos tener en cuenta esto a la hora de interpretar los resultados obtenidos en los posteriores modelos, ya que podemos tener un underfitting en la clasificación de proteínas de membrana y mitocondriales.

Cabe mencionar que para la utilización de los distintos modelos descartaremos la columna que sirve de identificador y modificaremos la columna clasificadora como sea necesario para poder realizar cada modelo.

k-Nearest Neighbour

1) Transformación de los datos

Tenemos los valores en formato numérico y ya normalizado, y los clasificadores se encuentran en formato de factores debidamente etiquetados, por lo que no es necesario procesar más los datos.

Realizamos la separación de los datos en train y test. Utilizaremos la semilla aleatoria "12345" y asignaremos el 67% de los datos a train y el restante 33% a test. Guardamos primero los valores sin etiquetas:

```
set.seed(12345)  
separacion_datos<-sample(1:nrow(datos_final), nrow(datos_final)*0.67,replace = FALSE)
```

```
train_knn<-datos_final[separacion_datos,2:9]
test_knn<-datos_final[-separacion_datos,2:9]
```

Ahora guardamos las etiquetas de los datos de train y test (que se encuentran en la columna 10):

```
train_label_knn<-datos_final[separacion_datos,10]
test_label_knn<-datos_final[-separacion_datos,10]
```

2) Entrenar el modelo

Para entrenar el modelo necesitamos saber que valor de K vamos a utilizar. Vamos a utilizar el método común para encontrar un valor bueno (aunque no implique el óptimo), que consiste en usar como valor la raíz cuadrada del número de registros en el modelo de train. Además se recomienda utilizar un valor impar para evitar empates en el entrenamiento. En nuestro caso tenemos 919 registros, por lo que su raíz cuadrada es 30.3150128. Por lo tanto para que sea impar utilizaremos una $k = 29$. También realizaremos modelos con una $k = 27$ y $k = 31$ para observar si existe overfitting o underfitting.

```
set.seed(12345)
knn29_datos<- knn(train= train_knn, test= test_knn, cl= train_label_knn, k= 29, prob= TRUE)
```

3) Evaluar y mejorar el rendimiento del modelo

Realizamos la matriz de confusión para evaluar el rendimiento del modelo recién creado.

Matriz de confusión de una knn con $k = 29$

```
confusionMatrix(knn29_datos, test_label_knn)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction CYT MEM MIT NUC
##           CYT  99   6  23  61
##           MEM   4  65   9   1
##           MIT   8   3  44  15
##           NUC  29   6   4  76
##
## Overall Statistics
##
##           Accuracy : 0.6269
##           95% CI : (0.5806, 0.6716)
##           No Information Rate : 0.3377
##           P-Value [Acc > NIR] : < 2.2e-16
##
##           Kappa : 0.487
##
##           Mcnemar's Test P-Value : 1.649e-05
##
## Statistics by Class:
##
##           Class: CYT Class: MEM Class: MIT Class: NUC
## Sensitivity           0.7071      0.8125      0.55000      0.4967
## Specificity           0.7125      0.9625      0.93029      0.8700
## Pos Pred Value        0.5238      0.8228      0.62857      0.6609
## Neg Pred Value        0.8447      0.9599      0.90601      0.7722
## Prevalence            0.3091      0.1766      0.17660      0.3377
```



```
## Detection Rate      0.2185    0.1435    0.09713    0.1678
## Detection Prevalence 0.4172    0.1744    0.15453    0.2539
## Balanced Accuracy   0.7098    0.8875    0.74015    0.6834
```

Observamos que el modelo knn con $k = 29$ tiene una precisión del 62%. Podemos considerar como baja esta precisión, así que podemos utilizar distintas knn para intentar mejorar el rendimiento del modelo. Probaremos el modelo con $k = 27$ y otro con $k = 31$ para saber que valor de k puede mejorar la precisión:

```
set.seed(12345)
knn27_datos<- knn(train= train_knn, test= test_knn, cl= train_label_knn, k= 27, prob= TRUE)
knn31_datos<- knn(train= train_knn, test= test_knn, cl= train_label_knn, k= 31, prob= TRUE)
```

Matriz de confusión de una knn con $k = 27$

```
confusionMatrix(knn27_datos, test_label_knn)

## Confusion Matrix and Statistics
##
##              Reference
## Prediction CYT MEM MIT NUC
##      CYT   97   6  25  60
##      MEM    5  65   9   2
##      MIT    7   3  42  15
##      NUC   31   6   4  76
##
## Overall Statistics
##
##              Accuracy : 0.6181
##              95% CI : (0.5716, 0.663)
##      No Information Rate : 0.3377
##      P-Value [Acc > NIR] : < 2.2e-16
##
##              Kappa : 0.4746
##
## Mcnemar's Test P-Value : 2.736e-05
##
## Statistics by Class:
##
##              Class: CYT Class: MEM Class: MIT Class: NUC
## Sensitivity      0.6929      0.8125      0.52500      0.4967
## Specificity      0.7093      0.9571      0.93298      0.8633
## Pos Pred Value   0.5160      0.8025      0.62687      0.6496
## Neg Pred Value   0.8377      0.9597      0.90155      0.7708
## Prevalence       0.3091      0.1766      0.17660      0.3377
## Detection Rate   0.2141      0.1435      0.09272      0.1678
## Detection Prevalence 0.4150      0.1788      0.14790      0.2583
## Balanced Accuracy 0.7011      0.8848      0.72899      0.6800
```

Matriz de confusión de una knn con $k = 31$

```
confusionMatrix(knn31_datos, test_label_knn)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction CYT MEM MIT NUC
##      CYT 100   6  20  64
```

```
##          MEM    5  67  10   1
##          MIT    7   2  44  16
##          NUC   28   5   6  72
##
## Overall Statistics
##
##              Accuracy : 0.6247
##              95% CI : (0.5783, 0.6695)
##      No Information Rate : 0.3377
##      P-Value [Acc > NIR] : < 2.2e-16
##
##              Kappa : 0.4848
##
## McNemar's Test P-Value : 1.057e-05
##
## Statistics by Class:
##
##              Class: CYT Class: MEM Class: MIT Class: NUC
## Sensitivity          0.7143      0.8375      0.55000      0.4706
## Specificity          0.7125      0.9571      0.93298      0.8700
## Pos Pred Value       0.5263      0.8072      0.63768      0.6486
## Neg Pred Value       0.8479      0.9649      0.90625      0.7632
## Prevalence           0.3091      0.1766      0.17660      0.3377
## Detection Rate       0.2208      0.1479      0.09713      0.1589
## Detection Prevalence 0.4194      0.1832      0.15232      0.2450
## Balanced Accuracy     0.7134      0.8973      0.74149      0.6703
```

Observamos en las matrices de confusión que los tres modelos son muy similares entre ellos en todos los aspectos, aunque existe un pequeño aumento en la precisión del modelo a medida que aumentamos el valor de K. Igualmente como el valor de la precisión no aumenta significativamente y la sensibilidad y especificidad de los distintos clasificadores no varía en gran medida, podemos quedarnos con cualquiera de los tres modelos. Aun así, una precisión del modelo menor al 63% resulta pobre, por lo que no es recomendable utilizar este algoritmo en métodos o ámbitos donde las predicciones del modelo sean de gran importancia.

Naive Bayes

Como el algoritmo de Naive Bayes utiliza variables categóricas en vez de numéricas, debemos cambiar las variables numéricas de nuestra base de datos por variables categóricas. Como los valores numéricos ya están normalizados, podemos subdividir en grupos los valores según en que rango numérico se encuentren.

1) Transformación de los datos

Hemos pensado en crear 5 grupos para todas las variables: “muy_baja” para valores entre 0-0.20, “baja” entre 0.21-0.40, “media” entre 0.41-0.60, “alta” entre 0.61-0.80, y “muy_alta” entre 0.81-1.00. De esta manera tendremos los niveles de reconocimiento de señal y los scores subdivididos correctamente en 5 grupos, permitiendo la aplicación del algoritmo. Para dividir en grupos la tabla según distintos puntos de corte utilizamos la función `cut()`

```
datos_naive <- datos_final
datos_naive<- as.data.frame(sapply(datos_naive[2:9],function(x) cut(x, breaks = c(0.00,0.20,0.40,0.60,0.80,1.00))
```

Observamos el resultado de la transformación:

```
str(datos_naive)
```

```
## 'data.frame':   1372 obs. of  8 variables:
```

```
## $ V2: Factor w/ 5 levels "alta","baja",...: 3 3 1 3 3 3 3 3 2 ...
## $ V3: Factor w/ 5 levels "alta","baja",...: 1 1 1 3 3 2 3 3 2 ...
## $ V4: Factor w/ 4 levels "alta","baja",...: 3 3 3 3 3 3 3 1 3 ...
## $ V5: Factor w/ 5 levels "alta","baja",...: 5 2 5 5 3 5 1 5 2 5 ...
## $ V6: Factor w/ 2 levels "media","muy_alta": 1 1 1 1 1 1 1 1 1 ...
## $ V7: Factor w/ 3 levels "media","muy_alta",...: 3 3 3 3 3 1 3 3 3 ...
## $ V8: Factor w/ 4 levels "alta","baja",...: 3 3 3 3 3 3 3 3 3 ...
## $ V9: Factor w/ 5 levels "alta","baja",...: 2 2 2 2 2 2 2 2 2 ...
```

Observamos que las variables numéricas se han convertido en variables categóricas (algunas columnas tienen menos de 5 niveles debido a que en los registros no se encuentran valores dentro de algún grupo). Cabe mencionar que en esta variable hemos eliminado el identificador debido a que no es necesario. La columna con los clasificadores la podemos utilizar igualmente en el algoritmo aunque ahora se encuentre separada. Además, como ya es una variable categórica no necesita ser procesada.

Creamos los train y test datasets de los datos recién procesados y de las etiquetas:

```
set.seed(12345)
separacion_datos_naive<-sample(1:nrow(datos_naive), nrow(datos_naive)*0.67,replace = FALSE)

train_naive<-datos_naive[separacion_datos_naive,]
test_naive<-datos_naive[-separacion_datos_naive,]

train_label_naive<-datos_final[separacion_datos,10]
test_label_naive<-datos_final[-separacion_datos,10]
```

2) Entrenar el modelo

Utilizamos Naive Bayes del paquete 'e1071' para entrenar al modelo:

```
set.seed(12345)
bayes_m<- naiveBayes(train_naive,train_label_naive, laplace = 0)
```

3) Evaluar y mejorar el rendimiento del modelo

Realizamos la función de predicción para observar el rendimiento del modelo:

```
test_pred<- predict(bayes_m,test_naive)
```

Ahora que tenemos la predicción del modelo, comparamos con los valores reales del test dataset con una matriz de confusión:

```
confusionMatrix(test_pred,test_label_naive)

## Confusion Matrix and Statistics
##
##              Reference
## Prediction  CYT  MEM  MIT  NUC
##      CYT    79   10   26   60
##      MEM     0   64   11    2
##      MIT     6    1   33   11
##      NUC    55    5   10   80
##
## Overall Statistics
##
##              Accuracy : 0.5651
##              95% CI : (0.5181, 0.6113)
```

```
##      No Information Rate : 0.3377
##      P-Value [Acc > NIR] : < 2.2e-16
##
##              Kappa : 0.3951
##
##      McNemar's Test P-Value : 1.377e-05
##
## Statistics by Class:
##
##              Class: CYT Class: MEM Class: MIT Class: NUC
## Sensitivity          0.5643      0.8000      0.41250      0.5229
## Specificity          0.6933      0.9651      0.95174      0.7667
## Pos Pred Value       0.4514      0.8312      0.64706      0.5333
## Neg Pred Value       0.7806      0.9574      0.88308      0.7591
## Prevalence           0.3091      0.1766      0.17660      0.3377
## Detection Rate       0.1744      0.1413      0.07285      0.1766
## Detection Prevalence 0.3863      0.1700      0.11258      0.3311
## Balanced Accuracy     0.6288      0.8826      0.68212      0.6448
```

Observamos que la precisión general es baja, llegando casi al 50% de error. Observando más detenidamente los distintos clasificadores vemos que donde se producen menores problemas a la hora de predecir es en la clase MEM, puede que debido a una menor cantidad de muestras o que las señales producidas son más reconocibles que en las otras clases. También observamos que el mayor problema de predicción es la baja sensibilidad de las clases. Esto puede intentar mejorarse subdividiendo en más grupos los valores o poniendo el valor de Laplace en 1. Por lo tanto, probaremos estas dos propuestas de mejora:

Laplace = 1

Aumentamos el valor de Laplace a 1:

```
set.seed(12345)
bayes_m_1<- naiveBayes(train_naive,train_label_naive, laplace = 1)
test_pred_1<- predict(bayes_m_1,test_naive)
```

Realizamos ahora la nueva matriz de confusión:

```
confusionMatrix(test_pred_1,test_label_naive)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction CYT MEM MIT NUC
##      CYT   78  10  26  61
##      MEM    0  65  11   1
##      MIT    8   1  33  11
##      NUC   54   4  10  80
##
## Overall Statistics
##
##              Accuracy : 0.5651
##              95% CI : (0.5181, 0.6113)
##      No Information Rate : 0.3377
##      P-Value [Acc > NIR] : < 2.2e-16
##
##              Kappa : 0.3957
##
```

```
## McNemar's Test P-Value : 3.703e-05
##
## Statistics by Class:
##
##           Class: CYT Class: MEM Class: MIT Class: NUC
## Sensitivity      0.5571      0.8125      0.41250      0.5229
## Specificity      0.6901      0.9678      0.94638      0.7733
## Pos Pred Value   0.4457      0.8442      0.62264      0.5405
## Neg Pred Value   0.7770      0.9601      0.88250      0.7607
## Prevalence       0.3091      0.1766      0.17660      0.3377
## Detection Rate   0.1722      0.1435      0.07285      0.1766
## Detection Prevalence 0.3863      0.1700      0.11700      0.3267
## Balanced Accuracy 0.6236      0.8902      0.67944      0.6481
```

No se observa ningún cambio significativo en el rendimiento del modelo, así que el valor de Laplace no parece ser un factor que afecte en gran medida a este modelo.

Separación en 10 grupos

Ahora vamos a subdividir las variables numéricas en una mayor cantidad de grupos, separando cada 0.1. Etiquetamos los grupos del 1 al 10 para saber en que decimal se encuentran:

```
datos_naive_2 <- datos_final
datos_naive_2 <- as.data.frame(sapply(datos_naive_2[2:9],function(x) cut(x, breaks = c(0.00,0.10,0.20,0.30,0.40,0.50,0.60,0.70,0.80,0.90,1.00))
```

Realizamos el train y test de los valores, sin necesidad de hacerlo de las etiquetas (ya tenemos la variable de la función anterior):

```
train_naive_2 <- datos_naive_2[separacion_datos_naive,]
test_naive_2 <- datos_naive_2[-separacion_datos_naive,]
```

Ahora creamos el modelo con los datos más subdivididos:

```
set.seed(12345)
bayes_m_2 <- naiveBayes(train_naive_2,train_label_naive, laplace = 0)
```

```
test_pred_2 <- predict(bayes_m_2,test_naive_2)
```

Obtenemos la nueva matriz de confusión:

```
confusionMatrix(test_pred_2,test_label_naive)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction CYT MEM MIT NUC
##      CYT   89   8  23  63
##      MEM    0  67   9   2
##      MIT   10   2  39  14
##      NUC   41   3   9  74
##
## Overall Statistics
##
##           Accuracy : 0.5938
##           95% CI : (0.547, 0.6394)
##      No Information Rate : 0.3377
##      P-Value [Acc > NIR] : < 2.2e-16
##
```

```
##                      Kappa : 0.4395
##
## McNemar's Test P-Value : 0.0006407
##
## Statistics by Class:
##
##                      Class: CYT Class: MEM Class: MIT Class: NUC
## Sensitivity           0.6357      0.8375      0.48750      0.4837
## Specificity           0.6997      0.9705      0.93029      0.8233
## Pos Pred Value        0.4863      0.8590      0.60000      0.5827
## Neg Pred Value        0.8111      0.9653      0.89433      0.7577
## Prevalence            0.3091      0.1766      0.17660      0.3377
## Detection Rate        0.1965      0.1479      0.08609      0.1634
## Detection Prevalence  0.4040      0.1722      0.14349      0.2804
## Balanced Accuracy     0.6677      0.9040      0.70890      0.6535
```

Observamos una significativa mejora general, aumentando la precisión del 56% de los modelos anteriores al 59% en este modelo. Aunque sigue siendo una precisión baja, se ha mejorado el rendimiento del modelo, indicando que si aumentamos el número de grupos facilita un correcto entrenamiento.

También podemos probar a realizar el entrenamiento del modelo a partir de los valores numéricos. La función permite el uso de variables numéricas pero no es lo recomendable, ya que se asume que los valores siguen una distribución gaussiana, siendo posible que no sea el caso. Por ello, consideramos que este último modelo realizado es el que mejor rendimiento tiene.

Artificial Neural Network

Para utilizar el algoritmo de ANN debemos tener en la tabla de datos una columna para cada clasificador, es decir, una columna para cada localización de la secuencia (CYT, MEM, MIT y NUC).

1) Transformación de los datos

Descartaremos la variable identificadora de la secuencia (V1) y crearemos 4 columnas (una para cada tipo de clasificador) donde se indicará con TRUE o FALSE si la secuencia está localizada o no en esa localización. Posteriormente eliminaremos la columna de clasificadores original.

```
datos_ann <- datos_final[-1]
datos_ann$CYT <- datos_ann$V10 == "CYT"
datos_ann$MEM <- datos_ann$V10 == "MEM"
datos_ann$MIT <- datos_ann$V10 == "MIT"
datos_ann$NUC <- datos_ann$V10 == "NUC"
datos_ann <- datos_ann[-9] #Eliminamos la columna con los 4 clasificadores juntos
```

Observamos como ha quedado el dataframe:

```
str(datos_ann)

## 'data.frame':   1372 obs. of  12 variables:
## $ V2 : num  0.58 0.43 0.64 0.58 0.42 0.51 0.5 0.48 0.55 0.4 ...
## $ V3 : num  0.61 0.67 0.62 0.44 0.44 0.4 0.54 0.45 0.5 0.39 ...
## $ V4 : num  0.47 0.48 0.49 0.57 0.48 0.56 0.48 0.59 0.66 0.6 ...
## $ V5 : num  0.13 0.27 0.15 0.13 0.54 0.17 0.65 0.2 0.36 0.15 ...
## $ V6 : num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
## $ V7 : num  0 0 0 0 0 0.5 0 0 0 0 ...
## $ V8 : num  0.48 0.53 0.53 0.54 0.48 0.49 0.53 0.58 0.49 0.58 ...
## $ V9 : num  0.22 0.22 0.22 0.22 0.22 0.22 0.22 0.34 0.22 0.3 ...
```

```
## $ CYT: logi FALSE FALSE FALSE FALSE FALSE TRUE ...
## $ MEM: logi FALSE FALSE FALSE FALSE FALSE FALSE ...
## $ MIT: logi TRUE TRUE TRUE FALSE TRUE FALSE ...
## $ NUC: logi FALSE FALSE FALSE TRUE FALSE FALSE ...
```

Tenemos las 8 variables numéricas esperadas y ahora los clasificadores se encuentran separados en cuatro columnas distintas y indicados con los valores lógicos TRUE (sí que se localiza en este lugar) y FALSE (no se localiza en este lugar).

Hacemos la selección de datos, 67% para train dataset y 33% para el test dataset:

```
set.seed(12345)
separacion_datos_ann<-sample(1:nrow(datos_ann), nrow(datos_ann)*0.67,replace = FALSE)
```

Creemos dos variables para guardar los datos seleccionados para el train dataset y test dataset:

```
train_ann<-datos_ann[separacion_datos_ann,]
test_ann<-datos_ann[-separacion_datos_ann,]
```

2) Entrenar el modelo

Utilizamos el paquete `neuralnet` para crear el modelo. Vamos a realizar el modelo con 5 nodos ocultos:

```
set.seed(12345)
ann_m_5<- neuralnet(CYT+MEM+MIT+NUC ~ ., data = train_ann,hidden = 5,linear.output = FALSE )
```

Podemos visualizar el modelo:

```
plot(ann_m_5)
```

Ahora podemos realizar la evaluación del modelo comparándolo con los resultados del test dataset (recogemos el valor de probabilidad de ser clasificado en cada caso, guardado en la variable “net.result”):

```
ann_m_5_result<- compute(ann_m_5,test_ann[,1:8])$net.result
```

Creemos una función que nos permita transformar el output binario en output categórico:

```
maxidx <- function(arr) {
  return(which(arr == max(arr)))
}
```

Aplicamos la función recién creada en los resultados del modelo:

```
transf_ann_m_5<-apply(ann_m_5_result, 1, maxidx)
```

Realizamos la predicción con los resultados anteriores:

```
predicted_ann_m_5<- c("CYT","MEM","MIT","NUC")[transf_ann_m_5]
```

3) Evaluar y mejorar el rendimiento del modelo

Realizamos la matriz de confusión para evaluar el rendimiento del modelo:

```
confusionMatrix(table(predicted_ann_m_5,datos_final$V10[-separacion_datos_ann]))
```

```
## Confusion Matrix and Statistics
##
##
## predicted_ann_m_5 CYT MEM MIT NUC
##              CYT  88   6  24  52
##              MEM   3  70   8   3
```

```
##           MIT    7    0  44  11
##           NUC   42    4    4  87
##
## Overall Statistics
##
##           Accuracy : 0.638
##           95% CI : (0.5918, 0.6823)
##           No Information Rate : 0.3377
##           P-Value [Acc > NIR] : < 2.2e-16
##
##           Kappa : 0.5006
##
## McNemar's Test P-Value : 0.0008678
##
## Statistics by Class:
##
##           Class: CYT Class: MEM Class: MIT Class: NUC
## Sensitivity           0.6286    0.8750    0.55000    0.5686
## Specificity           0.7380    0.9625    0.95174    0.8333
## Pos Pred Value        0.5176    0.8333    0.70968    0.6350
## Neg Pred Value        0.8163    0.9729    0.90793    0.7911
## Prevalence            0.3091    0.1766    0.17660    0.3377
## Detection Rate        0.1943    0.1545    0.09713    0.1921
## Detection Prevalence  0.3753    0.1854    0.13687    0.3024
## Balanced Accuracy     0.6833    0.9187    0.75087    0.7010
```

Observamos que utilizando 5 nodos ocultos obtenemos un 64% de precisión del modelo, siendo la predicción de CYT y de NUC los que menor precisión tienen y la predicción de MEM la que mejor porcentaje de precisión obtiene. El rendimiento del modelo es bastante más elevado que en algoritmos anteriores, pero sigue teniendo un rendimiento bajo.

Para intentar mejorar el rendimiento del modelo podemos aumentar el número de nodos ocultos. Por ello, vamos a crear dos modelos nuevos, uno con 3 nodos ocultos y otro con 8 nodos ocultos. De esta manera observaremos si el rendimiento del modelo tiende a aumentar o disminuir según el número de nodos ocultos.

3 nodos ocultos

Utilizamos el paquete `neuralnet` para crear el modelo. Vamos a realizar el modelo con 3 nodos ocultos:

```
set.seed(12345)
ann_m_3<- neuralnet(CYT+MEM+MIT+NUC ~ ., data = train_ann,hidden = 3,linear.output = FALSE)
```

Podemos visualizar el modelo:

```
plot(ann_m_3)
```

Ahora podemos realizar la evaluación del modelo comparándolo con los resultados del test dataset (recogemos el valor de probabilidad de ser clasificado en cada caso, guardado en la variable “net.result”):

```
ann_m_3_result<- compute(ann_m_3,test_ann[,1:8])$net.result
```

Aplicamos la función recién creada en los resultados del modelo:

```
transf_ann_m_3<-apply(ann_m_3_result, 1, maxidx)
```

Realizamos la predicción con los resultados anteriores:

```
predicted_ann_m_3<- c("CYT", "MEM", "MIT", "NUC")[transf_ann_m_3]
```


Realizamos la matriz de confusión para evaluar el rendimiento del modelo:

```
confusionMatrix(table(predicted_ann_m_3,datos_final$V10[-separacion_datos_ann]))
```

```
## Confusion Matrix and Statistics
##
##
## predicted_ann_m_3 CYT MEM MIT NUC
##           CYT  88   6  23  51
##           MEM   3  68  11   3
##           MIT   8   3  40  12
##           NUC  41   3   6  87
##
## Overall Statistics
##
##           Accuracy : 0.6247
##           95% CI : (0.5783, 0.6695)
##           No Information Rate : 0.3377
##           P-Value [Acc > NIR] : < 2e-16
##
##           Kappa : 0.4828
##
## Mcnemar's Test P-Value : 0.01421
##
## Statistics by Class:
##
##           Class: CYT Class: MEM Class: MIT Class: NUC
## Sensitivity           0.6286      0.8500      0.5000      0.5686
## Specificity           0.7444      0.9544      0.9383      0.8333
## Pos Pred Value        0.5238      0.8000      0.6349      0.6350
## Neg Pred Value        0.8175      0.9674      0.8974      0.7911
## Prevalence            0.3091      0.1766      0.1766      0.3377
## Detection Rate        0.1943      0.1501      0.0883      0.1921
## Detection Prevalence  0.3709      0.1876      0.1391      0.3024
## Balanced Accuracy      0.6865      0.9022      0.7192      0.7010
```

Observamos una leve disminución en el rendimiento del modelo, por lo que intentaremos ahora aumentar el número de nodos ocultos a 8:

10 nodos ocultos y threshold = 0.05

Ahora realizamos el modelo con 10 nodos ocultos, teniendo que aumentar el umbral de error a 0.05 para que el algoritmo no sea demasiado costoso computacionalmente (si no ponemos este threshold llega al valor máximo de stepmax, si subimos en gran medida el valor de stepmax el coste computacional será elevado):

```
set.seed(12345)
ann_m_10<- neuralnet(CYT+MEM+MIT+NUC ~ ., data = train_ann,hidden = 10,linear.output = FALSE, threshold
```

Podemos visualizar el modelo:

```
plot(ann_m_10)
```

Ahora podemos realizar la evaluación del modelo comparándolo con los resultados del test dataset (recogemos el valor de probabilidad de ser clasificado en cada caso, guardado en la variable “net.result”):

```
ann_m_10_result<- compute(ann_m_10,test_ann[,1:8])$net.result
```

Aplicamos la función recién creada en los resultados del modelo:

```
transf_ann_m_10<-apply(ann_m_10_result, 1, maxidx)
```

Realizamos la predicción con los resultados anteriores:

```
predicted_ann_m_10<- c("CYT", "MEM", "MIT", "NUC")[transf_ann_m_10]
```

Realizamos la matriz de confusión para evaluar el rendimiento del modelo:

```
confusionMatrix(table(predicted_ann_m_10,datos_final$V10[-separacion_datos_ann]))
```

```
## Confusion Matrix and Statistics
##
##
## predicted_ann_m_10 CYT MEM MIT NUC
##          CYT   90    4   22   50
##          MEM    2   70    8    2
##          MIT    5    1   45   13
##          NUC   43    5    5   88
##
## Overall Statistics
##
##              Accuracy : 0.6468
##              95% CI   : (0.6008, 0.6908)
##      No Information Rate : 0.3377
##      P-Value [Acc > NIR] : < 2.2e-16
##
##              Kappa   : 0.5126
##
##  Mcnemar's Test P-Value : 0.001122
##
## Statistics by Class:
##
##              Class: CYT Class: MEM Class: MIT Class: NUC
## Sensitivity          0.6429      0.8750      0.56250      0.5752
## Specificity          0.7572      0.9678      0.94906      0.8233
## Pos Pred Value       0.5422      0.8537      0.70312      0.6241
## Neg Pred Value       0.8258      0.9730      0.91003      0.7917
## Prevalence           0.3091      0.1766      0.17660      0.3377
## Detection Rate       0.1987      0.1545      0.09934      0.1943
## Detection Prevalence 0.3664      0.1810      0.14128      0.3113
## Balanced Accuracy    0.7000      0.9214      0.75578      0.6992
```

Observamos que la precisión ha aumentado ligeramente si comparamos la precisión global con el modelo de 5 nodos ocultos (una diferencia de un 1%). Este modelo se puede considerar mejor que el anterior debido al ligero aumento en su rendimiento sin aumentar su coste computacional. Al intentar aumentar el número de nodos a más de 10, era necesario un aumento del umbral de error y el aumento de pasos necesarios para generar el modelo, lo cual resultaba en un aumento en el coste computacional. Igualmente, entre 5 y 10 nodos no ha aumentado el rendimiento del modelo suficientemente como para recomendar la utilización de modelos con mayor número de nodos en los datos utilizados en este PEC.

Support Vector Machine

Para el algoritmo de SVM necesitamos que todas las variables excepto los clasificadores (que permanecen como factores, variables categóricas) sean numéricas, por lo que podemos utilizar directamente los datos iniciales sin procesar. Además, los valores numéricos deben estar normalizados, y los nuestros (como ya

hemos dicho anteriormente) ya se encuentran normalizados correctamente. Cabe mencionar que también eliminaremos el primer vector identificador debido a que no es necesario en el algoritmo.

1) Transformación de los datos

Eliminamos el primer vector identificador para tener la tabla de datos preparada:

```
datos_svm <- datos_final[-1]
str(datos_svm)

## 'data.frame':    1372 obs. of  9 variables:
## $ V2 : num  0.58 0.43 0.64 0.58 0.42 0.51 0.5 0.48 0.55 0.4 ...
## $ V3 : num  0.61 0.67 0.62 0.44 0.44 0.4 0.54 0.45 0.5 0.39 ...
## $ V4 : num  0.47 0.48 0.49 0.57 0.48 0.56 0.48 0.59 0.66 0.6 ...
## $ V5 : num  0.13 0.27 0.15 0.13 0.54 0.17 0.65 0.2 0.36 0.15 ...
## $ V6 : num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
## $ V7 : num  0 0 0 0 0 0.5 0 0 0 0 ...
## $ V8 : num  0.48 0.53 0.53 0.54 0.48 0.49 0.53 0.58 0.49 0.58 ...
## $ V9 : num  0.22 0.22 0.22 0.22 0.22 0.22 0.22 0.34 0.22 0.3 ...
## $ V10: Factor w/ 4 levels "CYT","MEM","MIT",...: 3 3 3 4 3 1 3 4 3 1 ...
```

Realizamos la división de los datos en train y test:

```
set.seed(12345)
separacion_datos_svm<-sample(1:nrow(datos_svm), nrow(datos_svm)*0.67,replace = FALSE)
```

Creamos dos variables para guardar los datos seleccionados para el train dataset y test dataset:

```
train_svm<-datos_svm[separacion_datos_svm,]
test_svm<-datos_svm[-separacion_datos_svm,]
```

2) Entrenar el modelo

Utilizamos la función `ksvm()` del paquete `kernlab` para entrenar el modelo SVM mediante el kernel ‘vanilladot’, que consiste en el algoritmo lineal de SVM:

```
set.seed(12345)
svm_m_linear <- ksvm(V10 ~ ., data = train_svm, kernel = "vanilladot")
```

```
## Setting default kernel parameters
```

Ahora utilizamos la función `predict()` para utilizar el modelo recién entrenado para predecir el test dataset:

```
predicted_svm_m_linear <- predict(svm_m_linear, test_svm[1:8])
```

3) Evaluar y mejorar el rendimiento del modelo

Realizamos ahora la matriz de confusión para observar el rendimiento del modelo:

```
confusionMatrix(predicted_svm_m_linear, test_svm$V10)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction CYT MEM MIT NUC
##      CYT 109   4  24  74
##      MEM   4  69  11   3
##      MIT   9   3  44  14
##      NUC  18   4   1  62
```

```
##
## Overall Statistics
##
##           Accuracy : 0.6269
##           95% CI : (0.5806, 0.6716)
##       No Information Rate : 0.3377
##       P-Value [Acc > NIR] : < 2.2e-16
##
##           Kappa : 0.49
##
## Mcnemar's Test P-Value : 1.927e-10
##
## Statistics by Class:
##
##           Class: CYT Class: MEM Class: MIT Class: NUC
## Sensitivity           0.7786      0.8625      0.55000      0.4052
## Specificity           0.6741      0.9517      0.93029      0.9233
## Pos Pred Value        0.5166      0.7931      0.62857      0.7294
## Neg Pred Value        0.8719      0.9699      0.90601      0.7527
## Prevalence            0.3091      0.1766      0.17660      0.3377
## Detection Rate        0.2406      0.1523      0.09713      0.1369
## Detection Prevalence  0.4658      0.1921      0.15453      0.1876
## Balanced Accuracy      0.7263      0.9071      0.74015      0.6643
```

La precisión general del modelo es de un 63%, y observamos que de manera general la precisión de cada clase es mayor a un 65%, por lo que la predicción aún no ser especialmente buena da buenos resultados. Los mayores problemas encontrados son la baja sensibilidad en la clase MIT y NUC (55% y 40% respectivamente).

Para intentar mejorar el modelo podemos utilizar el kernel 'rbf' nombrado en la función 'rbfdot'.

kernel rbf

Vamos a utilizar el kernel rbf y comparar los resultados:

```
set.seed(12345)
svm_m_rbf <- ksvm(V10 ~ ., data = train_svm, kernel = "rbfdot")
```

Ahora utilizamos la función `predict()` para utilizar el modelo recién entrenado para predecir el test dataset:

```
predicted_svm_m_rbf <- predict(svm_m_rbf, test_svm[1:8])
```

Realizamos ahora la matriz de confusión para observar el rendimiento del modelo:

```
confusionMatrix(predicted_svm_m_rbf, test_svm$V10)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction CYT MEM MIT NUC
##           CYT  97   5  27  60
##           MEM   2  66   9   2
##           MIT   3   1  40  13
##           NUC  38   8   4  78
##
## Overall Statistics
##
##           Accuracy : 0.6203
```

```
##                      95% CI : (0.5738, 0.6652)
##      No Information Rate : 0.3377
##      P-Value [Acc > NIR] : < 2.2e-16
##
##                      Kappa : 0.4746
##
##      McNemar's Test P-Value : 4.181e-07
##
## Statistics by Class:
##
##                      Class: CYT Class: MEM Class: MIT Class: NUC
## Sensitivity           0.6929      0.8250      0.5000      0.5098
## Specificity           0.7061      0.9651      0.9544      0.8333
## Pos Pred Value        0.5132      0.8354      0.7018      0.6094
## Neg Pred Value        0.8371      0.9626      0.8990      0.7692
## Prevalence            0.3091      0.1766      0.1766      0.3377
## Detection Rate        0.2141      0.1457      0.0883      0.1722
## Detection Prevalence  0.4172      0.1744      0.1258      0.2826
## Balanced Accuracy     0.6995      0.8951      0.7272      0.6716
```

Observamos que el rendimiento global del modelo ha disminuido en un 1% respecto al modelo en que se utiliza el kernel lineal. Comparando las estadísticas por clase entendemos que esta disminución de la precisión es debida a que hay una disminución en la sensibilidad significativa en todas las clases

Arbol de Decisión

Para utilizar el algoritmo del Árbol de decisión los datos pueden ser tanto nominales como numéricos. Al tener los datos preparados como numéricos, ya se encuentran preparados para su utilización en el modelo. También al basarnos en un árbol de decisiones, no es necesario normalizar los datos aunque éstos ya se encuentran normalizados.

1) Transformación de los datos

Los datos ya se encuentran listos para su uso, pero eliminamos la primera columna (con datos identificadores) debido a que no es necesaria:

```
datos_tree <- datos_final[-1]
```

Para este algoritmo debemos tener los clasificadores separados de los otros valores en los train y test datasets. Creamos los train y test datasets basándonos en esto:

```
set.seed(12345)
separacion_datos_tree<-sample(1:nrow(datos_tree), nrow(datos_tree)*0.67,replace = FALSE)
```

```
train_tree <- datos_svm[separacion_datos_svm,1:8]
test_tree <- datos_svm[-separacion_datos_svm,1:8]
```

```
train_tree_label <- datos_svm[separacion_datos_svm,9]
test_tree_label <- datos_svm[-separacion_datos_svm,9]
```

2) Entrenar el modelo

Para entrenar el modelo vamos a utilizar el algoritmo de arbol de decisión c5.0 utilizando la función `c5.0()` que obtenemos del paquete `C50`:

```
set.seed(12345)
tree_m<- C5.0(train_tree,train_tree_label)
```

Podemos observar las decisiones del modelo en su clasificación:

```
summary(tree_m)
```

```
##
## Call:
## C5.0.default(x = train_tree, y = train_tree_label)
##
##
## C5.0 [Release 2.07 GPL Edition]      Thu Jan 09 14:13:22 2020
## -----
##
## Class specified by attribute `outcome'
##
## Read 919 cases (9 attributes) from undefined.data
##
## Decision tree:
##
## V4 <= 0.43:
##   ...V4 <= 0.38: MEM (114/9)
##   :   V4 > 0.38:
##   :     ...V5 > 0.32:
##   :       :...V9 > 0.29: MEM (4/1)
##   :       :   V9 <= 0.29:
##   :       :     ...V2 > 0.68: MEM (4/1)
##   :       :       V2 <= 0.68:
##   :       :         ...V3 > 0.49: MIT (7)
##   :       :         V3 <= 0.49:
##   :       :           ...V2 <= 0.5: MIT (2/1)
##   :       :           V2 > 0.5: MEM (2)
##   :     V5 <= 0.32:
##   :       ...V3 > 0.64:
##   :         :...V2 <= 0.75: MEM (5)
##   :         :   V2 > 0.75:
##   :         :     ...V2 <= 0.78: MIT (2)
##   :         :       V2 > 0.78: MEM (3)
##   :       V3 <= 0.64:
##   :         ...V9 > 0.26:
##   :           :...V4 <= 0.42: MEM (9/3)
##   :           :   V4 > 0.42: NUC (4)
##   :           V9 <= 0.26:
##   :             ...V4 <= 0.41: MEM (20/4)
##   :             V4 > 0.41:
##   :               ...V4 > 0.42: MEM (6/2)
##   :               V4 <= 0.42:
##   :                 ...V2 <= 0.45: MEM (2)
##   :                 V2 > 0.45: CYT (5/1)
##   V4 > 0.43:
##   ...V5 > 0.39:
##     ...V3 > 0.68: MEM (4)
##     :   V3 <= 0.68:
```

```

##      :      :...V9 <= 0.32:
##      :      :...V3 > 0.52: MIT (48/4)
##      :      :      V3 <= 0.52:
##      :      :      :...V8 > 0.54: MIT (8/1)
##      :      :      :      V8 <= 0.54:
##      :      :      :...V8 <= 0.41: CYT (3)
##      :      :      :      V8 > 0.41:
##      :      :      :...V9 <= 0.11: CYT (2)
##      :      :      :      V9 > 0.11: MIT (40/11)
##      :      V9 > 0.32:
##      :      :...V3 > 0.5: MIT (8/1)
##      :      :      V3 <= 0.5:
##      :      :      :...V9 > 0.59: NUC (6/1)
##      :      :      :      V9 <= 0.59:
##      :      :      :...V3 <= 0.41: CYT (2)
##      :      :      :      V3 > 0.41:
##      :      :      :...V8 <= 0.5: NUC (4)
##      :      :      :      V8 > 0.5: CYT (3/1)
##      V5 <= 0.39:
##      :...V9 > 0.32:
##      :...V8 > 0.52:
##      :      :...V3 <= 0.45: NUC (17/4)
##      :      :      V3 > 0.45:
##      :      :      :...V2 <= 0.39: MEM (3/1)
##      :      :      :      V2 > 0.39:
##      :      :      :...V9 <= 0.37: NUC (5/1)
##      :      :      :      V9 > 0.37: CYT (14/2)
##      :      V8 <= 0.52:
##      :      :...V4 <= 0.45:
##      :      :...V5 <= 0.19: MEM (3/1)
##      :      :      V5 > 0.19: NUC (4/1)
##      :      :      V4 > 0.45:
##      :      :...V3 > 0.52: NUC (24/1)
##      :      :      V3 <= 0.52:
##      :      :      :...V2 <= 0.5: NUC (48/9)
##      :      :      :      V2 > 0.5:
##      :      :      :...V4 > 0.53: NUC (5)
##      :      :      :      V4 <= 0.53:
##      :      :      :...V3 <= 0.41: NUC (3/1)
##      :      :      :      V3 > 0.41: CYT (8)
##      V9 <= 0.32:
##      :...V2 > 0.73:
##      :...V8 <= 0.48:
##      :      :...V2 <= 0.85: MEM (2/1)
##      :      :      V2 > 0.85: NUC (2)
##      :      :      V8 > 0.48:
##      :      :...V3 <= 0.6: CYT (3)
##      :      :      V3 > 0.6: MEM (8/1)
##      V2 <= 0.73:
##      :...V4 <= 0.47:
##      :...V2 <= 0.4:
##      :      :...V8 > 0.48: NUC (11/2)
##      :      :      V8 <= 0.48:
##      :      :      :...V9 > 0.25: CYT (3/2)

```

```

##      :      :      V9 <= 0.25:
##      :      :      :...V4 <= 0.45: CYT (2)
##      :      :      V4 > 0.45: NUC (5/1)
##      :      V2 > 0.4:
##      :      :...V4 <= 0.44:
##      :      :...V5 <= 0.22: MEM (4/1)
##      :      :      V5 > 0.22: CYT (2/1)
##      :      V4 > 0.44:
##      :      :...V3 <= 0.53:
##      :      :...V8 <= 0.46: NUC (3/1)
##      :      :      V8 > 0.46: CYT (21/9)
##      :      V3 > 0.53:
##      :      :...V2 <= 0.5: MEM (2)
##      :      V2 > 0.5:
##      :      :...V9 > 0.26: CYT (2)
##      :      V9 <= 0.26:
##      :      :...V4 > 0.45: MIT (3)
##      :      V4 <= 0.45:
##      :      :...V2 <= 0.64: CYT (5/2)
##      :      V2 > 0.64: MIT (2)
##      V4 > 0.47:
##      :...V3 <= 0.39:
##      :      :...V3 <= 0.25: NUC (9/1)
##      :      V3 > 0.25:
##      :      :...V5 > 0.35: CYT (6)
##      :      V5 <= 0.35:
##      :      :...V3 <= 0.28:
##      :      :...V3 <= 0.27: CYT (3)
##      :      :      V3 > 0.27: MIT (3/1)
##      :      V3 > 0.28:
##      :      :...V9 > 0.25: NUC (21/4)
##      :      V9 <= 0.25:
##      :      :...V3 > 0.32:
##      :      :...V8 <= 0.53: NUC (26/10)
##      :      :      V8 > 0.53: CYT (7/1)
##      :      V3 <= 0.32:
##      :      :...V2 <= 0.41: CYT (3/1)
##      :      V2 > 0.41:
##      :      :...V8 <= 0.5: MIT (2)
##      :      V8 > 0.5: NUC (5)
##      V3 > 0.39:
##      :...V5 <= 0.2:
##      :      :...V9 <= 0.23: CYT (109/38)
##      :      V9 > 0.23:
##      :      :...V4 > 0.59: NUC (4)
##      :      V4 <= 0.59:
##      :      :...V5 > 0.12:
##      :      :...V8 <= 0.42: NUC (2)
##      :      :      V8 > 0.42: CYT (46/10)
##      :      V5 <= 0.12:
##      :      :...V9 <= 0.26: NUC (4)
##      :      V9 > 0.26:
##      :      :...V4 <= 0.56: CYT (4/1)
##      :      V4 > 0.56: NUC (2)

```



```

##          V5 > 0.2:
##          :...V4 > 0.57:
##              :...V8 <= 0.45: CYT (10/2)
##              :   V8 > 0.45:
##              :   :...V4 > 0.61: MIT (9/1)
##              :       V4 <= 0.61:
##              :       :...V3 > 0.54: CYT (2)
##              :       V3 <= 0.54:
##              :       :...V5 > 0.3: MIT (5)
##              :       V5 <= 0.3:
##              :       :...V4 > 0.58: NUC (4/1)
##              :       V4 <= 0.58:
##              :       :...V9 <= 0.24: MIT (5/2)
##              :       V9 > 0.24: NUC (3/1)
##          V4 <= 0.57:
##          :...V2 > 0.56:
##              :...V9 > 0.3: CYT (3/2)
##              :   V9 <= 0.3:
##              :   :...V9 > 0.28: MIT (2)
##              :       V9 <= 0.28:
##              :       :...V3 <= 0.6: CYT (16/1)
##              :       V3 > 0.6: MIT (3/1)
##          V2 <= 0.56:
##          :...V2 <= 0.3: CYT (3/1)
##          V2 > 0.3:
##          :...V2 > 0.5:
##              :...V4 <= 0.51: CYT (8)
##              :   V4 > 0.51:
##              :   :...V3 <= 0.47: CYT (3)
##              :       V3 > 0.47:
##              :       :...V3 <= 0.54: NUC (3)
##              :       V3 > 0.54: CYT (3/1)
##          V2 <= 0.5:
##          :...V8 > 0.53: CYT (12/3)
##          V8 <= 0.53:
##          :...V9 <= 0.19: CYT (3)
##          V9 > 0.19:
##          :...V2 <= 0.37: NUC (6/1)
##          V2 > 0.37:
##          :...V2 <= 0.42: CYT (3)
##          V2 > 0.42:
##          :...V8 <= 0.48: [S1]
##          V8 > 0.48: [S2]
##
## SubTree [S1]
##
## V3 <= 0.44: MIT (3)
## V3 > 0.44: NUC (6/1)
##
## SubTree [S2]
##
## V5 > 0.33: NUC (6/3)
## V5 <= 0.33:
## :...V3 <= 0.56: CYT (8/1)

```

```
##      V3 > 0.56: NUC (3/1)
##
##
## Evaluation on training data (919 cases):
##
##      Decision Tree
##      -----
##      Size      Errors
##
##      96  173(18.8%)  <<
##
##
##      (a)   (b)   (c)   (d)   <-classified as
##      ----  ----  ----  ----
##      247    6    15    36   (a): class CYT
##      6     170    2     (b): class MEM
##      21     7    129    7   (c): class MIT
##      53     12    8    200  (d): class NUC
##
##
## Attribute usage:
##
## 100.00% V4
## 87.60% V5
## 86.07% V9
## 82.59% V3
## 63.76% V2
## 46.79% V8
##
##
## Time: 0.0 secs
```

Observamos el distinto nivel de importancia de los datos proporcionados, siendo los valores del vector 4 (la puntuación de ALOM) el que ha tenido un 100% de importancia durante la creación del árbol de decisión. Además, el modelo ha calculado el porcentaje de error durante el entrenamiento, siendo el 18.8%.

Ahora utilizamos la función `predict()` para utilizar el modelo recién entrenado para predecir el test dataset:

```
predicted_tree_m <- predict(tree_m, test_tree)
```

3) Evaluar y mejorar el rendimiento del modelo

Para observar el rendimiento del modelo realizamos la matriz de confusión de los datos predecidos:

```
confusionMatrix(predicted_tree_m, test_tree_label)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction  CYT  MEM  MIT  NUC
##      CYT    87   3   21   55
##      MEM     4   71   12    8
##      MIT    15    1   36   12
##      NUC    34    5   11   78
##
## Overall Statistics
```

```
##
##           Accuracy : 0.6004
##           95% CI : (0.5537, 0.6459)
##      No Information Rate : 0.3377
##      P-Value [Acc > NIR] : < 2e-16
##
##           Kappa : 0.4522
##
##  McNemar's Test P-Value : 0.01301
##
## Statistics by Class:
##
##           Class: CYT Class: MEM Class: MIT Class: NUC
## Sensitivity           0.6214      0.8875      0.45000      0.5098
## Specificity           0.7476      0.9357      0.92493      0.8333
## Pos Pred Value        0.5241      0.7474      0.56250      0.6094
## Neg Pred Value        0.8153      0.9749      0.88689      0.7692
## Prevalence            0.3091      0.1766      0.17660      0.3377
## Detection Rate        0.1921      0.1567      0.07947      0.1722
## Detection Prevalence  0.3664      0.2097      0.14128      0.2826
## Balanced Accuracy     0.6845      0.9116      0.68747      0.6716
```

El modelo tiene una precisión global del 60% lo cual (como se ha mencionado en los modelos anteriores) es un poco bajo. El modelo tiene esta precisión debido principalmente a la baja especificidad de las clases MIT y NUC, las cuales rondan por el 50%. En la matriz de confusión observamos también que la clase CYT tiene un alto número de predicciones erróneas debido a que cree que 55 valores pertenecen a la clase NUC, provocando esta disminución de la precisión en la clase.

Para intentar obtener una mejora del modelo utilizando este algoritmo, se plantea utilizar un “boosting” de valor 10 para aumentar el número árboles de decisión creados. En el anterior modelo el atributo de boosting no se ha especificado y por tanto ha utilizado su valor estándar (1). Este atributo se puede modificar añadiendo el atributo `trial` en el modelo.

trials = 10

Realizamos un modelo con un boosting de valor 10 (10 trials):

```
set.seed(12345)
tree_m_10<- C5.0(train_tree,train_tree_label, trials = 10)
```

Observamos el nuevo árbol creado por este modelo:

```
summary(tree_m_10)

##
## Call:
## C5.0.default(x = train_tree, y = train_tree_label, trials = 10)
##
##
## C5.0 [Release 2.07 GPL Edition]      Thu Jan 09 14:13:22 2020
## -----
##
## Class specified by attribute `outcome'
##
## Read 919 cases (9 attributes) from undefined.data
##
## ----- Trial 0: -----
```

```

##
## Decision tree:
##
## V4 <= 0.43:
## :...V4 <= 0.38: MEM (114/9)
## :   V4 > 0.38:
## :     :...V5 > 0.32:
## :       :...V9 > 0.29: MEM (4/1)
## :       :   V9 <= 0.29:
## :       :     :...V2 > 0.68: MEM (4/1)
## :       :       V2 <= 0.68:
## :       :         :...V3 > 0.49: MIT (7)
## :       :         V3 <= 0.49:
## :       :           :...V2 <= 0.5: MIT (2/1)
## :       :           V2 > 0.5: MEM (2)
## :     V5 <= 0.32:
## :       :...V3 > 0.64:
## :         :...V2 <= 0.75: MEM (5)
## :         :   V2 > 0.75:
## :         :     :...V2 <= 0.78: MIT (2)
## :         :       V2 > 0.78: MEM (3)
## :       V3 <= 0.64:
## :         :...V9 > 0.26:
## :           :...V4 <= 0.42: MEM (9/3)
## :           :   V4 > 0.42: NUC (4)
## :         V9 <= 0.26:
## :           :...V4 <= 0.41: MEM (20/4)
## :           V4 > 0.41:
## :             :...V4 > 0.42: MEM (6/2)
## :             V4 <= 0.42:
## :               :...V2 <= 0.45: MEM (2)
## :               V2 > 0.45: CYT (5/1)
## V4 > 0.43:
## :...V5 > 0.39:
## :   :...V3 > 0.68: MEM (4)
## :   :   V3 <= 0.68:
## :   :     :...V9 <= 0.32:
## :   :       :...V3 > 0.52: MIT (48/4)
## :   :       :   V3 <= 0.52:
## :   :       :     :...V8 > 0.54: MIT (8/1)
## :   :       :       V8 <= 0.54:
## :   :       :         :...V8 <= 0.41: CYT (3)
## :   :       :         V8 > 0.41:
## :   :       :           :...V9 <= 0.11: CYT (2)
## :   :       :           V9 > 0.11: MIT (40/11)
## :     V9 > 0.32:
## :       :...V3 > 0.5: MIT (8/1)
## :       :   V3 <= 0.5:
## :       :     :...V9 > 0.59: NUC (6/1)
## :       :       V9 <= 0.59:
## :       :         :...V3 <= 0.41: CYT (2)
## :       :         V3 > 0.41:
## :       :           :...V8 <= 0.5: NUC (4)
## :       :           V8 > 0.5: CYT (3/1)

```

```

## V5 <= 0.39:
## :...V9 > 0.32:
## :...V8 > 0.52:
## : :...V3 <= 0.45: NUC (17/4)
## : : V3 > 0.45:
## : : :...V2 <= 0.39: MEM (3/1)
## : : V2 > 0.39:
## : : :...V9 <= 0.37: NUC (5/1)
## : : V9 > 0.37: CYT (14/2)
## : V8 <= 0.52:
## : :...V4 <= 0.45:
## : :...V5 <= 0.19: MEM (3/1)
## : : V5 > 0.19: NUC (4/1)
## : V4 > 0.45:
## : :...V3 > 0.52: NUC (24/1)
## : V3 <= 0.52:
## : :...V2 <= 0.5: NUC (48/9)
## : V2 > 0.5:
## : :...V4 > 0.53: NUC (5)
## : V4 <= 0.53:
## : :...V3 <= 0.41: NUC (3/1)
## : V3 > 0.41: CYT (8)
## V9 <= 0.32:
## :...V2 > 0.73:
## :...V8 <= 0.48:
## : :...V2 <= 0.85: MEM (2/1)
## : : V2 > 0.85: NUC (2)
## : V8 > 0.48:
## : :...V3 <= 0.6: CYT (3)
## : V3 > 0.6: MEM (8/1)
## V2 <= 0.73:
## :...V4 <= 0.47:
## :...V2 <= 0.4:
## : :...V8 > 0.48: NUC (11/2)
## : : V8 <= 0.48:
## : : :...V9 > 0.25: CYT (3/2)
## : : V9 <= 0.25:
## : : :...V4 <= 0.45: CYT (2)
## : : V4 > 0.45: NUC (5/1)
## : V2 > 0.4:
## : :...V4 <= 0.44:
## : :...V5 <= 0.22: MEM (4/1)
## : : V5 > 0.22: CYT (2/1)
## : V4 > 0.44:
## : :...V3 <= 0.53:
## : :...V8 <= 0.46: NUC (3/1)
## : : V8 > 0.46: CYT (21/9)
## : V3 > 0.53:
## : :...V2 <= 0.5: MEM (2)
## : V2 > 0.5:
## : :...V9 > 0.26: CYT (2)
## : V9 <= 0.26:
## : :...V4 > 0.45: MIT (3)
## : V4 <= 0.45:

```

```

##          :          : ...V2 <= 0.64: CYT (5/2)
##          :          : V2 > 0.64: MIT (2)
## V4 > 0.47:
## : ...V3 <= 0.39:
##       : ...V3 <= 0.25: NUC (9/1)
##       :   V3 > 0.25:
##       :   : ...V5 > 0.35: CYT (6)
##       :   :   V5 <= 0.35:
##       :   :   : ...V3 <= 0.28:
##       :   :   :   : ...V3 <= 0.27: CYT (3)
##       :   :   :   :   V3 > 0.27: MIT (3/1)
##       :   :   :   :   V3 > 0.28:
##       :   :   :   :   : ...V9 > 0.25: NUC (21/4)
##       :   :   :   :   :   V9 <= 0.25:
##       :   :   :   :   :   : ...V3 > 0.32:
##       :   :   :   :   :   :   : ...V8 <= 0.53: NUC (26/10)
##       :   :   :   :   :   :   :   V8 > 0.53: CYT (7/1)
##       :   :   :   :   :   :   :   V3 <= 0.32:
##       :   :   :   :   :   :   :   : ...V2 <= 0.41: CYT (3/1)
##       :   :   :   :   :   :   :   :   V2 > 0.41:
##       :   :   :   :   :   :   :   :   : ...V8 <= 0.5: MIT (2)
##       :   :   :   :   :   :   :   :   :   V8 > 0.5: NUC (5)
## V3 > 0.39:
## : ...V5 <= 0.2:
##       : ...V9 <= 0.23: CYT (109/38)
##       :   V9 > 0.23:
##       :   : ...V4 > 0.59: NUC (4)
##       :   :   V4 <= 0.59:
##       :   :   : ...V5 > 0.12:
##       :   :   :   : ...V8 <= 0.42: NUC (2)
##       :   :   :   :   V8 > 0.42: CYT (46/10)
##       :   :   :   :   V5 <= 0.12:
##       :   :   :   :   : ...V9 <= 0.26: NUC (4)
##       :   :   :   :   :   V9 > 0.26:
##       :   :   :   :   :   : ...V4 <= 0.56: CYT (4/1)
##       :   :   :   :   :   :   V4 > 0.56: NUC (2)
## V5 > 0.2:
## : ...V4 > 0.57:
##       : ...V8 <= 0.45: CYT (10/2)
##       :   V8 > 0.45:
##       :   : ...V4 > 0.61: MIT (9/1)
##       :   :   V4 <= 0.61:
##       :   :   : ...V3 > 0.54: CYT (2)
##       :   :   :   V3 <= 0.54:
##       :   :   :   : ...V5 > 0.3: MIT (5)
##       :   :   :   :   V5 <= 0.3:
##       :   :   :   :   : ...V4 > 0.58: NUC (4/1)
##       :   :   :   :   :   V4 <= 0.58:
##       :   :   :   :   :   : ...V9 <= 0.24: MIT (5/2)
##       :   :   :   :   :   :   V9 > 0.24: NUC (3/1)
## V4 <= 0.57:
## : ...V2 > 0.56:
##       : ...V9 > 0.3: CYT (3/2)
##       :   V9 <= 0.3:

```

```

##          :   :...V9 > 0.28: MIT (2)
##          :       V9 <= 0.28:
##          :       :...V3 <= 0.6: CYT (16/1)
##          :           V3 > 0.6: MIT (3/1)
##      V2 <= 0.56:
##      :...V2 <= 0.3: CYT (3/1)
##          V2 > 0.3:
##          :...V2 > 0.5:
##              :...V4 <= 0.51: CYT (8)
##              :       V4 > 0.51:
##              :       :...V3 <= 0.47: CYT (3)
##              :           V3 > 0.47:
##              :           :...V3 <= 0.54: NUC (3)
##              :               V3 > 0.54: CYT (3/1)
##      V2 <= 0.5:
##      :...V8 > 0.53: CYT (12/3)
##          V8 <= 0.53:
##          :...V9 <= 0.19: CYT (3)
##              V9 > 0.19:
##              :...V2 <= 0.37: NUC (6/1)
##                  V2 > 0.37:
##                  :...V2 <= 0.42: CYT (3)
##                      V2 > 0.42:
##                      :...V8 <= 0.48: [S1]
##                          V8 > 0.48: [S2]
##
## SubTree [S1]
##
## V3 <= 0.44: MIT (3)
## V3 > 0.44: NUC (6/1)
##
## SubTree [S2]
##
## V5 > 0.33: NUC (6/3)
## V5 <= 0.33:
## :...V3 <= 0.56: CYT (8/1)
##     V3 > 0.56: NUC (3/1)
##
## ----- Trial 1: -----
##
## Decision tree:
##
## V4 <= 0.43:
## :...V4 <= 0.36: MEM (70.9/5.5)
## :   V4 > 0.36:
## :       :...V5 <= 0.27:
## :           :...V8 <= 0.56: MEM (55.4/15)
## :               :   V8 > 0.56: NUC (6.1/1.6)
## :           V5 > 0.27:
## :               :...V2 <= 0.58:
## :                   :...V5 <= 0.51: NUC (19.5/8.5)
## :                       :   V5 > 0.51: MIT (4)
## :                   V2 > 0.58:
## :                       :...V9 > 0.29: MEM (4.3/1.8)

```

```

## :                V9 <= 0.29:
## :                :...V3 <= 0.65: MIT (6.9/1.6)
## :                V3 > 0.65: MEM (8.1/1.6)
## V4 > 0.43:
## :...V5 > 0.34:
## :...V2 <= 0.35: NUC (8.5)
## :   V2 > 0.35:
## :   :...V2 > 0.81: NUC (5.1/2.4)
## :       V2 <= 0.81:
## :       :...V9 > 0.32:
## :           :...V3 <= 0.46: NUC (10.7/2.4)
## :           :   V3 > 0.46: MIT (9.9/2.4)
## :           V9 <= 0.32:
## :           :...V3 <= 0.44:
## :           :...V4 <= 0.49: MIT (5.7/2.4)
## :           :   V4 > 0.49: CYT (27.1/8.1)
## :           V3 > 0.44:
## :           :...V5 > 0.79: CYT (3.7/1.8)
## :           V5 <= 0.79:
## :           :...V9 <= 0.21: CYT (4.3/0.8)
## :           V9 > 0.21: MIT (83.9/16.8)
## V5 <= 0.34:
## :...V2 > 0.7: MEM (17.6/9.3)
## :   V2 <= 0.7:
## :   :...V4 <= 0.47:
## :       :...V9 > 0.28:
## :       :   :...V8 <= 0.43: CYT (3.4)
## :       :       V8 > 0.43:
## :       :       :...V2 <= 0.41:
## :       :       :   :...V9 <= 0.35: MEM (6.9/2.4)
## :       :       :       V9 > 0.35: NUC (9.7/0.8)
## :       :       V2 > 0.41:
## :       :       :...V5 <= 0.19: CYT (7.5/1.6)
## :       :       V5 > 0.19: NUC (6.7/0.8)
## :       V9 <= 0.28:
## :       :...V9 > 0.23: NUC (11.1/6.1)
## :       V9 <= 0.23:
## :       :...V3 <= 0.52:
## :       :       :...V8 <= 0.43: MIT (2.6/0.8)
## :       :       :   V8 > 0.43:
## :       :       :   :...V5 <= 0.14: NUC (7.1/3.4)
## :       :       :       V5 > 0.14: MEM (19.9/10.9)
## :       V3 > 0.52:
## :       :...V3 > 0.73: NUC (2.6)
## :       V3 <= 0.73:
## :       :...V2 <= 0.45: NUC (2.4/0.8)
## :       V2 > 0.45: MIT (12.8/4)
## V4 > 0.47:
## :...V9 > 0.25:
## :   :...V8 > 0.52:
## :       :...V5 > 0.24:
## :       :   :...V4 <= 0.54: MIT (7.7/4)
## :       :       V4 > 0.54: CYT (6.7/1.6)
## :       V5 <= 0.24:

```



```

##      :      : ...V3 <= 0.3: NUC (4.8/0.8)
##      :      :      V3 > 0.3:
##      :      :      ...V8 <= 0.54: CYT (25.9/5.7)
##      :      :      V8 > 0.54:
##      :      :      ...V3 <= 0.53: CYT (7.7/1.6)
##      :      :      V3 > 0.53: NUC (5.5)
##      :      V8 <= 0.52:
##      :      ...V4 > 0.58: NUC (28.7/3.4)
##      :      V4 <= 0.58:
##      :      ...V2 > 0.63: NUC (4.3/0.8)
##      :      V2 <= 0.63:
##      :      ...V9 > 0.64: NUC (5.7)
##      :      V9 <= 0.64:
##      :      ...V8 <= 0.47:
##      :      :      ...V4 <= 0.56: NUC (24.5/5.1)
##      :      :      :      V4 > 0.56: CYT (7.7/0.8)
##      :      :      V8 > 0.47:
##      :      :      ...V4 > 0.51: NUC (36.6/17.2)
##      :      :      V4 <= 0.51:
##      :      :      ...V2 > 0.48: CYT (15.8/1.6)
##      :      :      V2 <= 0.48:
##      :      :      ...V3 <= 0.46: CYT (14/2.4)
##      :      :      V3 > 0.46: NUC (4.3)
##      V9 <= 0.25:
##      ...V5 > 0.28:
##      :      ...V2 <= 0.36: MEM (2.6/0.8)
##      :      V2 > 0.36:
##      :      ...V3 <= 0.42: MIT (3.2/0.8)
##      :      V3 > 0.42: CYT (13.1/3.4)
##      V5 <= 0.28:
##      ...V4 <= 0.53:
##      :      ...V5 > 0.21:
##      :      :      ...V8 <= 0.48: MIT (7.7/2.4)
##      :      :      :      V8 > 0.48: CYT (18.6/2.4)
##      :      :      V5 <= 0.21:
##      :      :      ...V8 > 0.56: CYT (9.3/3.7)
##      :      :      V8 <= 0.56:
##      :      :      ...V9 <= 0.23: NUC (59.4/27.1)
##      :      :      V9 > 0.23:
##      :      :      ...V5 <= 0.18: NUC (10.3/2.4)
##      :      :      V5 > 0.18: CYT (3.2)
##      V4 > 0.53:
##      ...V3 > 0.58:
##      :      ...V2 <= 0.51: CYT (6.7)
##      :      V2 > 0.51: MIT (7.9/0.8)
##      V3 <= 0.58:
##      :      ...V2 > 0.58: CYT (10.1/4.5)
##      :      V2 <= 0.58:
##      :      ...V4 > 0.61:
##      :      :      ...V5 <= 0.24: CYT (13/3.4)
##      :      :      :      V5 > 0.24: MIT (6.9/2.6)
##      :      :      V4 <= 0.61:
##      :      :      ...V4 <= 0.55:
##      :      :      :      ...V5 <= 0.11: NUC (4.5)

```

```

## : V5 > 0.11:
## : :...V3 <= 0.56: CYT (25.9/8.3)
## : V3 > 0.56: NUC (3.7)
## V4 > 0.55:
## :...V3 <= 0.43:
## :...V2 <= 0.46: NUC (17.2/2.4)
## : V2 > 0.46: CYT (12.8/3.4)
## V3 > 0.43:
## :...V2 > 0.53: NUC (10.8/2.6)
## V2 <= 0.53:
## :...V5 > 0.24: NUC (5.1/0.8)
## V5 <= 0.24:
## :...V4 > 0.58: CYT (9.3/2.6)
## V4 <= 0.58:
## :...V2 <= 0.47: MIT (9.7/1.6)
## V2 > 0.47: CYT (4)
##
## ----- Trial 2: -----
##
## Decision tree:
##
## V4 <= 0.45:
## :...V4 <= 0.38: MEM (89.2/17.7)
## : V4 > 0.38:
## : :...V2 > 0.62:
## : :...V2 > 0.78: MEM (7.3)
## : : V2 <= 0.78:
## : : :...V4 > 0.44: MIT (5.2/2.6)
## : : V4 <= 0.44:
## : : :...V2 <= 0.74: MEM (12.8/4.8)
## : : V2 > 0.74: MIT (4.8/0.7)
## : V2 <= 0.62:
## : :...V5 > 0.32:
## : :...V4 <= 0.41: MEM (9.5/5)
## : : V4 > 0.41:
## : : :...V8 <= 0.53: MIT (12.1/2.6)
## : : V8 > 0.53: NUC (2.8)
## : V5 <= 0.32:
## : :...V8 <= 0.43: CYT (2.8)
## : V8 > 0.43:
## : :...V2 <= 0.47:
## : :...V8 > 0.53: NUC (8.4/0.7)
## : : V8 <= 0.53:
## : : :...V5 <= 0.18: MEM (11.4/4.8)
## : : V5 > 0.18: NUC (16.6/6.2)
## : V2 > 0.47:
## : :...V9 > 0.3: MEM (8.5/3.3)
## : V9 <= 0.3:
## : :...V3 <= 0.48: CYT (14/4.2)
## : V3 > 0.48:
## : :...V5 <= 0.24: MEM (15.1/8.2)
## : V5 > 0.24: NUC (5.6/2.6)
## V4 > 0.45:
## :...V5 > 0.34:

```

```

## :...V2 > 0.83: MEM (4.8/2.2)
## : V2 <= 0.83:
## : :...V2 <= 0.35: NUC (7.1)
## : V2 > 0.35:
## : :...V3 <= 0.39:
## : :...V3 <= 0.31: NUC (4.6/0.7)
## : : V3 > 0.31: CYT (13.9/2.6)
## : V3 > 0.39:
## : :...V5 > 0.82: NUC (3.8/1.5)
## : V5 <= 0.82:
## : :...V9 > 0.32:
## : :...V4 <= 0.58: MIT (12.4/6.8)
## : : V4 > 0.58: CYT (2.6)
## : V9 <= 0.32:
## : :...V8 > 0.54: MIT (10.3)
## : V8 <= 0.54:
## : :...V7 > 0: MIT (2.2)
## : V7 <= 0:
## : :...V5 <= 0.39:
## : :...V4 <= 0.6: CYT (12/4.8)
## : : V4 > 0.6: MIT (5.5/0.7)
## : V5 > 0.39:
## : :...V3 > 0.52: MIT (28.8/3.8)
## : V3 <= 0.52:
## : :...V8 <= 0.41: CYT (3.3)
## : V8 > 0.41:
## : :...V9 > 0.25: MIT (2)
## : V9 <= 0.25:
## : :...V2 <= 0.51: MIT (15/3.7)
## : V2 > 0.51:
## : :...V3 <= 0.42: MIT (2.6)
## : V3 > 0.42: CYT (12.1/1.4)
## V5 <= 0.34:
## :...V2 > 0.7:
## : :...V4 <= 0.49: MEM (3.6)
## : : V4 > 0.49:
## : : :...V4 <= 0.51: CYT (3.9)
## : : V4 > 0.51: NUC (9.9/5)
## V2 <= 0.7:
## :...V9 > 0.31:
## : :...V8 <= 0.52:
## : : :...V3 > 0.48: NUC (25.1/2.9)
## : : : V3 <= 0.48:
## : : : :...V9 <= 0.49: CYT (40.3/16.7)
## : : : V9 > 0.49: NUC (10.9/1.5)
## : : V8 > 0.52:
## : : :...V5 > 0.25: CYT (7/4.1)
## : : V5 <= 0.25:
## : : :...V9 <= 0.37: NUC (9.1/0.7)
## : : V9 > 0.37: CYT (22.5/9.6)
## V9 <= 0.31:
## :...V4 <= 0.47:
## : :...V9 > 0.3: MEM (4.4/1.4)
## : : V9 <= 0.3:

```

```

##          :      :...V8 <= 0.45: MIT (7.2/3.3)
##          :      V8 > 0.45:
##          :      :...V3 <= 0.38: MIT (2.3)
##          :      V3 > 0.38: CYT (29/13.8)
## V4 > 0.47:
## :...V4 <= 0.53:
##       :...V8 <= 0.47: NUC (41.4/19)
##       :      V8 > 0.47:
##       :      :...V9 <= 0.19: NUC (2.3)
##       :      V9 > 0.19: CYT (138.1/47.1)
## V4 > 0.53:
## :...V3 > 0.51:
##       :...V8 <= 0.47: CYT (10.8/1.4)
##       :      V8 > 0.47: MIT (24.4/11.8)
## V3 <= 0.51:
## :...V3 <= 0.3: NUC (11.2/4.1)
##       V3 > 0.3:
##       :...V4 <= 0.55:
##       :...V4 <= 0.54:
##       :      :...V5 <= 0.15: NUC (5.7)
##       :      :      V5 > 0.15: CYT (12.1/5.8)
##       :      V4 > 0.54:
##       :      :...V8 <= 0.49: NUC (7.4)
##       :      V8 > 0.49: CYT (10/2.3)
## V4 > 0.55:
## :...V9 > 0.29: CYT (5.2)
##       V9 <= 0.29:
##       :...V5 > 0.3: MIT (3.3/0.7)
##       V5 <= 0.3:
##       :...V9 > 0.24:
##       :...V5 <= 0.14: NUC (4.9)
##       :      V5 > 0.14:
##       :      :...V8 <= 0.49: CYT (7/0.7)
##       :      V8 > 0.49: NUC (8.6/2)
## V9 <= 0.24:
## :...V9 <= 0.11: CYT (3.4/0.7)
##       V9 > 0.11:
##       :...V2 <= 0.47:
##       :...V2 <= 0.37: NUC (10/2.8)
##       :      V2 > 0.37:
##       :      :...V2 <= 0.44: CYT (13.2/2.6)
##       :      V2 > 0.44: [S1]
## V2 > 0.47:
## :...V2 > 0.57: MIT (10.3/4.4)
##       V2 <= 0.57:
##       :...V2 > 0.55: CYT (5)
##       V2 <= 0.55: [S2]
##
## SubTree [S1]
##
## V8 <= 0.55: MIT (11.1/5.1)
## V8 > 0.55: CYT (3.9/1.3)
##
## SubTree [S2]

```

```

##
## V5 > 0.24: NUC (7/2.6)
## V5 <= 0.24:
## :...V8 <= 0.39: NUC (3/0.7)
##     V8 > 0.39: CYT (15.2/5.4)
##
## ----- Trial 3: -----
##
## Decision tree:
##
## V4 <= 0.43:
## :...V4 <= 0.36: MEM (54.8/8.1)
## :   V4 > 0.36:
## :     :...V9 > 0.31:
## :       :...V5 <= 0.42: NUC (12.7/2.8)
## :       :   V5 > 0.42: MEM (2.3)
## :       V9 <= 0.31:
## :         :...V5 > 0.27:
## :           :...V9 > 0.29: CYT (3.3/0.6)
## :           :   V9 <= 0.29:
## :             :   :...V4 <= 0.4: MIT (15.6/3.4)
## :             :     V4 > 0.4:
## :             :       :...V4 <= 0.42: MEM (12.8/6.7)
## :             :       V4 > 0.42: MIT (4.8/1.2)
## :           V5 <= 0.27:
## :             :...V4 <= 0.39: MEM (17.6)
## :             V4 > 0.39:
## :               :...V8 > 0.57: NUC (3/1.1)
## :               V8 <= 0.57:
## :                 :...V5 > 0.21: NUC (10.8/3.9)
## :                 V5 <= 0.21:
## :                   :...V3 <= 0.39: CYT (5.1)
## :                   V3 > 0.39: MEM (16/6.4)
## V4 > 0.43:
## :...V5 > 0.39:
## :   :...V3 > 0.68: MEM (4.4)
## :   :   V3 <= 0.68:
## :     :   :...V2 <= 0.35: NUC (3.8)
## :     :     V2 > 0.35:
## :       :   :...V9 <= 0.36: MIT (90.9/35)
## :       :     V9 > 0.36: NUC (12.5/4.7)
## V5 <= 0.39:
## :...V9 > 0.32:
## :   :...V8 <= 0.54:
## :     :   :...V3 <= 0.35: CYT (17.6/7.1)
## :     :     :   V3 > 0.35: NUC (95.3/32.3)
## :     :     V8 > 0.54:
## :       :   :...V4 <= 0.53: NUC (11/1.9)
## :       :     V4 > 0.53: CYT (6/1.1)
## V9 <= 0.32:
## :...V2 > 0.75: MEM (14.1/7.6)
## :   V2 <= 0.75:
## :     :...V4 <= 0.47:
## :     :   :...V9 > 0.3: MEM (6.9/3.2)

```

```

##      :   V9 <= 0.3:
##      :   :...V9 > 0.23:
##      :       :...V3 <= 0.54: NUC (16.8/6.2)
##      :       :   V3 > 0.54: CYT (3.3/1.1)
##      :       V9 <= 0.23:
##      :       :...V3 > 0.52:
##      :           :...V2 <= 0.41: NUC (3.2)
##      :           :   V2 > 0.41:
##      :           :       :...V3 > 0.67: MEM (4/2.4)
##      :           :           V3 <= 0.67:
##      :           :           :...V3 <= 0.62: MIT (12.9/4.5)
##      :           :           V3 > 0.62: CYT (3.4)
##      :           V3 <= 0.52:
##      :           :...V8 <= 0.43: MIT (2.4/1.1)
##      :           V8 > 0.43:
##      :           :...V3 <= 0.32: NUC (3.4)
##      :           V3 > 0.32:
##      :           :...V2 <= 0.51: CYT (14.4/6.3)
##      :           V2 > 0.51:
##      :           :...V8 <= 0.47: MEM (3/1.1)
##      :           V8 > 0.47:
##      :           :...V3 <= 0.43: CYT (2.2)
##      :           V3 > 0.43: NUC (12/2.8)
##      V4 > 0.47:
##      :...V4 > 0.57:
##      :       :...V3 > 0.49:
##      :       :       :...V8 <= 0.45: CYT (2.3)
##      :       :       :   V8 > 0.45: MIT (16.9/6.2)
##      :       :       V3 <= 0.49:
##      :       :       :...V3 <= 0.31: MIT (10.4/5.1)
##      :       :       V3 > 0.31:
##      :       :       :...V8 <= 0.47: CYT (23.2/7.6)
##      :       :       V8 > 0.47:
##      :       :       :...V9 > 0.25: NUC (13.5/5)
##      :       :       V9 <= 0.25:
##      :       :       :...V2 <= 0.55: NUC (24.4/10.9)
##      :       :       V2 > 0.55: CYT (8.3/4.1)
##      V4 <= 0.57:
##      :...V2 <= 0.3: CYT (14.1/3.8)
##      V2 > 0.3:
##      :...V8 <= 0.36: NUC (9.3/4.3)
##      V8 > 0.36:
##      :...V2 <= 0.37:
##      :       :...V5 > 0.24: NUC (8.9/0.6)
##      :       V5 <= 0.24:
##      :       :...V5 > 0.19: CYT (2.3)
##      :       V5 <= 0.19:
##      :       :...V4 <= 0.49: NUC (6.9)
##      :       V4 > 0.49:
##      :       :...V2 <= 0.34: CYT (6.7/0.6)
##      :       V2 > 0.34: NUC (14.8/5.3)
##      V2 > 0.37:
##      :...V8 > 0.56:
##      :       :...V2 <= 0.43: MIT (5.7)

```

```

##          :   V2 > 0.43: CYT (10.9/1.3)
##          V8 <= 0.56:
##          :...V3 <= 0.51:
##              :...V9 <= 0.21: CYT (5.6)
##              :   V9 > 0.21:
##              :   :...V4 <= 0.51:
##              :       :...V3 <= 0.38: NUC (12/3.3)
##              :       :   V3 > 0.38: CYT (49.1/13.4)
##              :       V4 > 0.51:
##              :       :...V3 > 0.47: NUC (27.6/6.3)
##              :       V3 <= 0.47:
##              :       :...V8 > 0.55: CYT (5.4)
##              :       V8 <= 0.55:
##              :       :...V3 > 0.4: [S1]
##              :       V3 <= 0.4: [S2]
##          V3 > 0.51:
##          :...V5 <= 0.15: CYT (17.2/1.3)
##          V5 > 0.15:
##          :...V8 > 0.54: NUC (6.5/2.7)
##          V8 <= 0.54:
##          :...V9 > 0.23:
##              :...V3 <= 0.53: NUC (3.2/1.3)
##              :   V3 > 0.53:
##              :   :...V3 <= 0.65: CYT (12.1/1.1)
##              :   V3 > 0.65: MIT (3.3/0.6)
##          V9 <= 0.23:
##          :...V8 > 0.53: CYT (3.3)
##          V8 <= 0.53:
##          :...V2 > 0.56: [S3]
##          V2 <= 0.56: [S4]
##
## SubTree [S1]
##
## V2 <= 0.53: CYT (24.3/9.3)
## V2 > 0.53: NUC (8.8/3.1)
##
## SubTree [S2]
##
## V2 <= 0.43: NUC (5.8)
## V2 > 0.43:
## :...V4 <= 0.54: CYT (5.2/0.6)
##     V4 > 0.54: NUC (10.1/3.2)
##
## SubTree [S3]
##
## V2 <= 0.58: MIT (6.2/0.6)
## V2 > 0.58: CYT (7.4/1.9)
##
## SubTree [S4]
##
## V5 > 0.34: MIT (4.9/1.7)
## V5 <= 0.34:
## :...V5 <= 0.28: NUC (18/5.4)
##     V5 > 0.28: CYT (2.2)

```

```

##
## ----- Trial 4: -----
##
## Decision tree:
##
## V4 <= 0.46:
## :...V4 <= 0.41:
## :   :...V9 > 0.3:
## :   :   :...V5 > 0.49: CYT (3.3/1)
## :   :   :   V5 <= 0.49:
## :   :   :   :...V9 <= 0.52: MEM (20.6/6.4)
## :   :   :   :   V9 > 0.52: NUC (3.3)
## :   :   V9 <= 0.3:
## :   :   :...V3 <= 0.53:
## :   :   :   :...V9 > 0.26: MIT (9.1/3.7)
## :   :   :   :   V9 <= 0.26:
## :   :   :   :   :...V3 <= 0.5: MEM (27.1/3.4)
## :   :   :   :   :   V3 > 0.5: NUC (9.6/2.4)
## :   :   :   V3 > 0.53:
## :   :   :   :...V4 <= 0.36: MEM (14.7)
## :   :   :   :   V4 > 0.36:
## :   :   :   :   :...V5 <= 0.29: MEM (16.6/2.5)
## :   :   :   :   :   V5 > 0.29:
## :   :   :   :   :   :...V3 <= 0.78: MIT (11.2/2.5)
## :   :   :   :   :   :   V3 > 0.78: MEM (2.5)
## :   V4 > 0.41:
## :   :...V5 > 0.33:
## :   :   :...V3 <= 0.49: NUC (9.8/2.9)
## :   :   :   V3 > 0.49: MIT (9.4/2.4)
## :   :   V5 <= 0.33:
## :   :   :...V2 > 0.58:
## :   :   :   :...V3 <= 0.64: CYT (17.6/5)
## :   :   :   :   V3 > 0.64: MEM (5.4/1)
## :   :   :   V2 <= 0.58:
## :   :   :   :...V8 <= 0.44: CYT (6.1/2)
## :   :   :   :   V8 > 0.44:
## :   :   :   :   :...V3 > 0.57: NUC (17.5/10.6)
## :   :   :   :   :   V3 <= 0.57:
## :   :   :   :   :   :...V2 > 0.51:
## :   :   :   :   :   :   :...V4 > 0.45: CYT (3)
## :   :   :   :   :   :   :   V4 <= 0.45:
## :   :   :   :   :   :   :   :...V8 <= 0.53: NUC (8.3)
## :   :   :   :   :   :   :   :   V8 > 0.53: CYT (5.5/1.9)
## :   :   :   :   V2 <= 0.51:
## :   :   :   :   :...V2 > 0.47:
## :   :   :   :   :   :...V9 <= 0.32: CYT (9.2/3.2)
## :   :   :   :   :   :   V9 > 0.32: MEM (4.1)
## :   :   :   :   :   V2 <= 0.47:
## :   :   :   :   :   :...V8 > 0.53: NUC (4.6)
## :   :   :   :   :   :   V8 <= 0.53:
## :   :   :   :   :   :   :...V5 <= 0.22: MEM (19.1/7)
## :   :   :   :   :   :   :   V5 > 0.22: NUC (4.2)
## V4 > 0.46:
## :...V2 > 0.7:

```



```

##      :...V8 > 0.63: CYT (3.1)
##      :      V8 <= 0.63:
##      :      :...V3 > 0.61: MEM (7.7/1.6)
##      :      :      V3 <= 0.61:
##      :      :      :...V4 <= 0.51: CYT (5.6/2.2)
##      :      :      :      V4 > 0.51: NUC (9.9/1.4)
##      V2 <= 0.7:
##      :...V5 > 0.39:
##      :      :...V2 <= 0.35: NUC (3.2)
##      :      :      V2 > 0.35:
##      :      :      :...V3 > 0.57: MIT (10.7/2.4)
##      :      :      :      V3 <= 0.57:
##      :      :      :      :...V9 > 0.32:
##      :      :      :      :      :...V3 <= 0.46: NUC (10.9/4.6)
##      :      :      :      :      :      V3 > 0.46: MIT (5.3/2.1)
##      :      :      :      :      :      V9 <= 0.32:
##      :      :      :      :      :      :...V8 > 0.54: MIT (4.4)
##      :      :      :      :      :      :      V8 <= 0.54:
##      :      :      :      :      :      :      :...V2 <= 0.62: CYT (58.7/23.1)
##      :      :      :      :      :      :      :      V2 > 0.62: MIT (3.4)
##      V5 <= 0.39:
##      :...V9 > 0.31: NUC (119.4/49)
##      :      V9 <= 0.31:
##      :      :...V2 <= 0.37:
##      :      :      :...V8 <= 0.42: MIT (6.6/1.9)
##      :      :      :      V8 > 0.42:
##      :      :      :      :...V4 > 0.59: NUC (9.8/3.1)
##      :      :      :      :      V4 <= 0.59:
##      :      :      :      :      :...V2 <= 0.2: MEM (2.3)
##      :      :      :      :      :      V2 > 0.2:
##      :      :      :      :      :      :...V2 <= 0.31:
##      :      :      :      :      :      :      :...V5 <= 0.19: CYT (15.8/1.7)
##      :      :      :      :      :      :      :      V5 > 0.19: NUC (6.5/2.1)
##      :      :      :      :      :      :      :      V2 > 0.31:
##      :      :      :      :      :      :      :      :...V9 <= 0.16: CYT (2.1)
##      :      :      :      :      :      :      :      :      V9 > 0.16: NUC (30.6/8)
##      V2 > 0.37:
##      :...V4 <= 0.47: CYT (24.5/15.3)
##      :      V4 > 0.47:
##      :      :...V5 > 0.28:
##      :      :      :...V8 <= 0.45: CYT (5.6)
##      :      :      :      V8 > 0.45:
##      :      :      :      :...V3 > 0.55: CYT (11.5/3.1)
##      :      :      :      :      V3 <= 0.55:
##      :      :      :      :      :...V8 <= 0.5: MIT (16.5/5.1)
##      :      :      :      :      :      V8 > 0.5:
##      :      :      :      :      :      :...V3 > 0.51: MIT (4.2)
##      :      :      :      :      :      :      V3 <= 0.51:
##      :      :      :      :      :      :      :...V9 > 0.26: NUC (5/1)
##      :      :      :      :      :      :      :      V9 <= 0.26:
##      :      :      :      :      :      :      :      :...V5 <= 0.38: CYT (11.9/3)
##      :      :      :      :      :      :      :      :      V5 > 0.38: NUC (2.1)
##      V5 <= 0.28:
##      :...V3 <= 0.32: NUC (16.8/5)

```

```

##          V3 > 0.32:
##          :...V2 <= 0.4:
##              :...V3 <= 0.38: NUC (3.6)
##              :   V3 > 0.38: CYT (15)
##          V2 > 0.4:
##          :...V4 <= 0.53:
##              :...V8 <= 0.47:
##                  :   :...V8 <= 0.46: NUC (16.2/4.5)
##                  :   :   V8 > 0.46: MIT (11.3/6)
##                  :   V8 > 0.47:
##                  :   :...V8 > 0.56: CYT (7.7/2.3)
##                  :       V8 <= 0.56:
##                  :       :...V4 <= 0.49: CYT (15.6/3.2)
##                  :       :       V4 > 0.49:
##                  :       :       :...V2 > 0.53: CYT (20.4/4.9)
##                  :       :       :       V2 <= 0.53:
##                  :       :       :       :...V9 <= 0.26: NUC (31.2/7.6)
##                  :       :       :       :       V9 > 0.26: CYT (10.5/1.6)
##          V4 > 0.53:
##          :...V2 > 0.58:
##              :...V8 > 0.5: CYT (6.5)
##              :   V8 <= 0.5:
##              :   :...V5 <= 0.22: MIT (11.3/1.8)
##              :   :       V5 > 0.22: CYT (2.9)
##          V2 <= 0.58:
##          :...V3 > 0.51:
##              :...V8 <= 0.47: CYT (9/1.8)
##              :   V8 > 0.47: MIT (16.6/6.7)
##          V3 <= 0.51:
##          :...V2 > 0.51: NUC (27/10.1)
##              V2 <= 0.51:
##              :...V4 > 0.63: MIT (5.6/2.2)
##                  V4 <= 0.63:
##                  :...V3 <= 0.44:
##                      :...V5 <= 0.17: NUC (12.5/4.7)
##                      :   V5 > 0.17: CYT (11.1/1.4)
##                      V3 > 0.44:
##                      :...V9 > 0.24: NUC (4.1/0.5)
##                      :       V9 <= 0.24: [S1]
##
## SubTree [S1]
##
## V4 <= 0.55: NUC (6.5/1.5)
## V4 > 0.55: CYT (18.3/9.1)
##
## ----- Trial 5: -----
##
## Decision tree:
##
## V4 <= 0.45:
## :...V4 <= 0.38:
## :   :...V9 <= 0.3: MEM (50.6/10.6)
## :   :   V9 > 0.3:
## :   :       :...V4 <= 0.35: MEM (6.7)

```

```

## :      :      V4 > 0.35:
## :      :      :...V2 <= 0.64: NUC (11/2.1)
## :      :      V2 > 0.64: MEM (2.2)
## : V4 > 0.38:
## : :...V2 > 0.62:
## :      :...V2 > 0.78: MEM (7.7)
## :      :      V2 <= 0.78:
## :      :      :...V3 > 0.66: MEM (6/1.3)
## :      :      V3 <= 0.66:
## :      :      :...V4 <= 0.41: MEM (5.6/2.7)
## :      :      V4 > 0.41:
## :      :      :...V3 <= 0.61: MIT (8/1.8)
## :      :      V3 > 0.61: CYT (5.2/0.4)
## : V2 <= 0.62:
## : :...V5 > 0.32: MIT (22.6/10.2)
## :      V5 <= 0.32:
## :      :...V3 > 0.58:
## :      :...V5 <= 0.22: MEM (13.7/6.2)
## :      :      V5 > 0.22: CYT (7.9/4.2)
## :      V3 <= 0.58:
## :      :...V8 <= 0.43: CYT (2.2)
## :      V8 > 0.43:
## :      :...V5 > 0.22:
## :      :...V8 <= 0.53: NUC (12.1/4.1)
## :      :      V8 > 0.53: MEM (3.2)
## :      V5 <= 0.22:
## :      :...V3 > 0.48:
## :      :...V4 <= 0.4: MEM (3.4)
## :      :      V4 > 0.4: NUC (15.9/5.3)
## :      V3 <= 0.48:
## :      :...V9 > 0.45: NUC (2.8)
## :      V9 <= 0.45:
## :      :...V2 <= 0.36: MEM (5.2/1.6)
## :      V2 > 0.36:
## :      :...V5 <= 0.15: NUC (5/1.8)
## :      V5 > 0.15:
## :      :...V8 <= 0.49: MEM (5.2/1.4)
## :      V8 > 0.49: CYT (9.8/0.8)
## V4 > 0.45:
## :...V5 > 0.39:
## :...V2 > 0.83: NUC (3.6/1.6)
## :      V2 <= 0.83:
## :      :...V9 > 0.32:
## :      :...V3 <= 0.46: NUC (12/5.8)
## :      :      V3 > 0.46: MIT (7.4/3.4)
## :      V9 <= 0.32:
## :      :...V5 > 0.79: NUC (4.8/2)
## :      V5 <= 0.79:
## :      :...V3 > 0.52: MIT (32.1/5.6)
## :      V3 <= 0.52:
## :      :...V8 <= 0.47: CYT (11.5/2.6)
## :      V8 > 0.47:
## :      :...V5 > 0.46: MIT (20.8/3.5)
## :      V5 <= 0.46:

```

```

##      :      :...V5 <= 0.41: MIT (2.5)
##      :      V5 > 0.41: CYT (15.9/4.7)
## V5 <= 0.39:
## :...V2 > 0.7:
##      :...V9 > 0.31: MEM (3.6/1.5)
##      :      V9 <= 0.31:
##      :      :...V3 <= 0.61: NUC (10.4/5.1)
##      :      V3 > 0.61: MEM (6.2/2.7)
## V2 <= 0.7:
## :...V9 > 0.3:
##      :...V8 > 0.53:
##      :      :...V3 <= 0.45:
##      :      :      :...V9 <= 0.39: MIT (6.7/2.1)
##      :      :      V9 > 0.39: NUC (8.9/3.6)
##      :      :      V3 > 0.45:
##      :      :      :...V9 <= 0.58: CYT (15.7/2.4)
##      :      :      V9 > 0.58: MEM (3.9/1.4)
##      :      V8 <= 0.53:
##      :      :...V4 <= 0.47:
##      :      :      :...V9 <= 0.34: MEM (7.1/3.4)
##      :      :      V9 > 0.34: NUC (11.6/4.1)
##      :      :      V4 > 0.47:
##      :      :      :...V4 > 0.56:
##      :      :      :...V9 <= 0.45: NUC (13.7)
##      :      :      V9 > 0.45: CYT (10.7/3.7)
##      :      :      V4 <= 0.56:
##      :      :      :...V3 <= 0.22: NUC (3.4)
##      :      :      V3 > 0.22:
##      :      :      :...V8 <= 0.47: NUC (10.7/3.1)
##      :      :      V8 > 0.47: CYT (59.3/21)
## V9 <= 0.3:
## :...V4 <= 0.5:
##      :...V8 <= 0.44: MIT (9.7/3.4)
##      :      V8 > 0.44:
##      :      :...V3 > 0.66: MIT (4.7/0.9)
##      :      V3 <= 0.66:
##      :      :...V8 > 0.53: CYT (12/2.3)
##      :      V8 <= 0.53:
##      :      :...V8 > 0.52: NUC (14.5/5)
##      :      V8 <= 0.52:
##      :      :...V2 <= 0.39: NUC (14.3/4.6)
##      :      V2 > 0.39:
##      :      :...V3 <= 0.38: MIT (7.2/3.1)
##      :      V3 > 0.38:
##      :      :...V3 <= 0.43: CYT (10.9/2)
##      :      V3 > 0.43:
##      :      :...V6 > 0.5: NUC (2.1)
##      :      V6 <= 0.5:
##      :      :...V3 > 0.5: CYT (15.7/4.5)
##      :      V3 <= 0.5:
##      :      :...V8 <= 0.5: CYT (12.1/4.8)
##      :      V8 > 0.5: NUC (11/2.5)
## V4 > 0.5:
## :...V5 > 0.24:

```

```

##          :...V2 <= 0.49:
##          :   :...V8 <= 0.46: CYT (12.6/6.1)
##          :   :   V8 > 0.46:
##          :   :   :...V9 <= 0.23: NUC (22.7/7.7)
##          :   :   :   V9 > 0.23: MIT (12.1/4.7)
##          :   V2 > 0.49:
##          :   :...V4 > 0.6: MIT (6.1/1.3)
##          :   :   V4 <= 0.6:
##          :   :   :...V3 <= 0.49: CYT (13.7/5.1)
##          :   :   :   V3 > 0.49: MIT (14.5/6)
##          V5 <= 0.24:
##          :...V9 > 0.24: NUC (57.1/28.6)
##          :   V9 <= 0.24:
##          :   :...V4 > 0.6: CYT (17.1/4.7)
##          :   :   V4 <= 0.6:
##          :   :   :...V5 > 0.21: CYT (18.3/7.7)
##          :   :   :   V5 <= 0.21:
##          :   :   :   :...V5 <= 0.09: CYT (4.2)
##          :   :   :   :   V5 > 0.09:
##          :   :   :   :   :...V4 <= 0.53:
##          :   :   :   :   :...V5 <= 0.19: CYT (36/11.3)
##          :   :   :   :   :   V5 > 0.19: NUC (9.3/2.1)
##          :   :   :   V4 > 0.53:
##          :   :   :   :...V2 > 0.58: MIT (9.8/3.6)
##          :   :   :   :   V2 <= 0.58:
##          :   :   :   :   :...V3 > 0.59: MIT (5.5/2.1)
##          :   :   :   :   :   V3 <= 0.59:
##          :   :   :   :   :   :...V4 <= 0.55:
##          :   :   :   :   :   :   :...V8 <= 0.5: NUC (11.3/2.1)
##          :   :   :   :   :   :   :   V8 > 0.5: CYT (18.7/6.6)
##          :   :   :   :   :   V4 > 0.55:
##          :   :   :   :   :   :...V3 <= 0.43: NUC (20.1/8.6)
##          :   :   :   :   :   :   V3 > 0.43:
##          :   :   :   :   :   :   :...V2 <= 0.5: CYT (10.4/5.1)
##          :   :   :   :   :   :   :   V2 > 0.5: NUC (11/5)
##
## ----- Trial 6: -----
##
## Decision tree:
##
## V4 <= 0.45:
## :...V4 <= 0.38:
## :   :...V9 <= 0.3:
## :   :   :...V5 <= 0.45: MEM (43/9.2)
## :   :   :   V5 > 0.45: NUC (3.7/0.7)
## :   :   V9 > 0.3:
## :   :   :...V5 <= 0.22: MEM (6.6)
## :   :   :   V5 > 0.22: NUC (12.1/4.4)
## :   V4 > 0.38:
## :   :...V2 > 0.62:
## :   :   :...V2 > 0.78: MEM (6.7)
## :   :   :   V2 <= 0.78:
## :   :   :   :...V4 <= 0.41: MIT (8.3/3.3)
## :   :   :   :   V4 > 0.41:

```

```

## :      : ...V3 <= 0.64: CYT (11.6/5.4)
## :      :      V3 > 0.64: MEM (4)
## : V2 <= 0.62:
## :      : ...V5 > 0.53: MIT (5.1)
## :      V5 <= 0.53:
## :      : ...V4 <= 0.43:
## :      : ...V2 > 0.55: CYT (11.8/6.6)
## :      :      V2 <= 0.55:
## :      :      : ...V3 <= 0.36: MEM (7.4/4.3)
## :      :      :      V3 > 0.36:
## :      :      :      : ...V8 <= 0.5: MEM (14.8/3)
## :      :      :      :      V8 > 0.5:
## :      :      :      :      : ...V4 <= 0.4: MEM (2.2)
## :      :      :      :      :      V4 > 0.4: NUC (14.1/3)
## :      V4 > 0.43:
## :      : ...V5 > 0.32: NUC (8.9/4.1)
## :      V5 <= 0.32:
## :      : ...V5 > 0.25: CYT (5/0.7)
## :      V5 <= 0.25:
## :      : ...V3 > 0.58: MIT (11.5/6.8)
## :      V3 <= 0.58:
## :      : ...V2 <= 0.33: NUC (2.6)
## :      V2 > 0.33:
## :      : ...V8 <= 0.5: NUC (10.4/4.8)
## :      V8 > 0.5:
## :      : ...V2 <= 0.41: MEM (4.8/1.5)
## :      V2 > 0.41: CYT (10.4/4.2)
## V4 > 0.45:
## : ...V5 > 0.34:
## :      : ...V2 > 0.83: NUC (5/2.2)
## :      V2 <= 0.83:
## :      : ...V2 <= 0.35: NUC (5.8)
## :      V2 > 0.35:
## :      : ...V9 > 0.32: CYT (20.1/10.5)
## :      V9 <= 0.32:
## :      : ...V5 > 0.79: NUC (4.7/2.3)
## :      V5 <= 0.79:
## :      : ...V5 <= 0.39:
## :      : ...V3 <= 0.43: CYT (10.9/3.8)
## :      :      V3 > 0.43: MIT (19/7.1)
## :      V5 > 0.39:
## :      : ...V2 <= 0.44: CYT (17.9/6.6)
## :      V2 > 0.44:
## :      : ...V3 > 0.54: MIT (11.1)
## :      V3 <= 0.54:
## :      : ...V8 <= 0.41: CYT (3)
## :      V8 > 0.41:
## :      : ...V9 > 0.23: MIT (11.6)
## :      V9 <= 0.23:
## :      : ...V5 > 0.56: MIT (7.7)
## :      V5 <= 0.56:
## :      : ...V8 > 0.53: MIT (4.4)
## :      V8 <= 0.53:
## :      : ...V9 <= 0.21: CYT (2.3)

```

```

##      :                               V9 > 0.21:
##      :                               :...V2 <= 0.51: MIT (3.8)
##      :                               V2 > 0.51: CYT (14.8/4.6)
## V5 <= 0.34:
## :...V2 > 0.7:
##      :...V4 > 0.57: NUC (2.2)
##      :   V4 <= 0.57:
##      :   :...V5 <= 0.22: CYT (11/4.2)
##      :       V5 > 0.22: MEM (3.5)
## V2 <= 0.7:
## :...V9 > 0.31:
##      :...V5 > 0.25:
##      :   :...V5 <= 0.26: MEM (2.5/0.4)
##      :       :   V5 > 0.26:
##      :       :   :...V2 <= 0.51: NUC (11.6/4.4)
##      :       :       V2 > 0.51: CYT (9.9/2.7)
##      :       V5 <= 0.25:
##      :       :...V9 <= 0.33: NUC (10.4/1.1)
##      :           V9 > 0.33:
##      :       :...V3 <= 0.29: NUC (6.6)
##      :           V3 > 0.29:
##      :       :...V3 <= 0.33: CYT (6.6)
##      :           V3 > 0.33:
##      :       :...V8 <= 0.48: NUC (13.4/2.4)
##      :           V8 > 0.48:
##      :       :...V4 <= 0.61: CYT (64.2/28.3)
##      :           V4 > 0.61: NUC (4.2)
## V9 <= 0.31:
## :...V4 <= 0.5:
##      :...V8 > 0.53: CYT (12.5/1.1)
##      :   V8 <= 0.53:
##      :   :...V2 <= 0.21: MEM (3.9/1.6)
##      :       V2 > 0.21:
##      :       :...V4 <= 0.47:
##      :           :...V5 <= 0.12: CYT (4)
##      :           :   V5 > 0.12:
##      :           :   :...V3 > 0.58: MIT (5.5/0.7)
##      :           :       V3 <= 0.58:
##      :           :       :...V8 <= 0.44: MIT (5/1.3)
##      :           :           V8 > 0.44:
##      :           :           :...V3 <= 0.38: MIT (4.1/1.7)
##      :           :           V3 > 0.38:
##      :           :           :...V5 <= 0.18: MEM (10.6/5.1)
##      :           :           V5 > 0.18: NUC (13/3.9)
##      :       V4 > 0.47:
##      :       :...V9 > 0.3: NUC (4.3)
##      :           V9 <= 0.3:
##      :           :...V9 > 0.27: CYT (5.4/1.7)
##      :           V9 <= 0.27:
##      :           :...V8 <= 0.44: NUC (4.5/1.7)
##      :           V8 > 0.44:
##      :           :...V5 > 0.22: CYT (13.3/5.3)
##      :           V5 <= 0.22:
##      :           :...V8 > 0.52: NUC (4.2)

```

```

##          :          V8 <= 0.52:
##          :          :...V3 <= 0.37: NUC (3.5)
##          :          V3 > 0.37:
##          :          :...V4 > 0.49: CYT (7.4/1.7)
##          :          V4 <= 0.49:
##          :          :...V5 <= 0.17: NUC (12.1/4)
##          :          V5 > 0.17: CYT (3.8)
## V4 > 0.5:
## :...V3 <= 0.3:
##     :...V8 <= 0.48: NUC (7/1)
##     :   V8 > 0.48: MIT (8.3/2.9)
##     V3 > 0.3:
##     :...V9 > 0.22:
##         :...V2 > 0.65: MIT (3.2/1.1)
##         :   V2 <= 0.65:
##         :   :...V5 > 0.24:
##         :       :...V2 <= 0.39: NUC (4.9)
##         :       :   V2 > 0.39: CYT (12.1/5.9)
##         :       V5 <= 0.24:
##         :       :...V9 > 0.26: CYT (34.6/7.1)
##         :       V9 <= 0.26:
##         :       :...V5 > 0.18: CYT (8.4/0.7)
##         :       V5 <= 0.18:
##         :       :...V4 <= 0.65: NUC (17.7/4.5)
##         :       V4 > 0.65: CYT (3.1)
##     V9 <= 0.22:
##     :...V5 > 0.28:
##         :...V8 > 0.56: NUC (2.7/1.1)
##         :   V8 <= 0.56:
##         :   :...V8 <= 0.45: CYT (2.4)
##         :   V8 > 0.45: MIT (11.1/2.6)
##     V5 <= 0.28:
##     :...V4 > 0.61: MIT (15.7/7)
##     V4 <= 0.61:
##     :...V5 > 0.27: NUC (4.6)
##     V5 <= 0.27:
##     :...V3 <= 0.4:
##     :...V3 <= 0.37: NUC (12.8/4.8)
##     :   V3 > 0.37: CYT (12.4/2.2)
##     V3 > 0.4:
##     :...V4 <= 0.55:
##     :...V3 <= 0.41: MIT (5.3/2.3)
##     :   V3 > 0.41:
##     :   :...V2 > 0.55:
##     :       :...V8 <= 0.47: MIT (5.2/1.8)
##     :       :   V8 > 0.47: CYT (13.5/1.6)
##     :       V2 <= 0.55:
##     :       :...V3 <= 0.47: CYT (19.5/4.3)
##     :       V3 > 0.47:
##     :       :...V3 <= 0.52: NUC (18.8/4.8)
##     :       V3 > 0.52: CYT (21.6/7.5)
##     V4 > 0.55:
##     :...V5 > 0.24: NUC (9.6/2.8)
##     V5 <= 0.24:

```



```

##                                     :...V4 > 0.6: CYT (5.4)
##                                     V4 <= 0.6:
##                                     :...V3 <= 0.43: CYT (8.6/3.2)
##                                     V3 > 0.43:
##                                     :...V2 <= 0.46: MIT (13.6/2.9)
##                                     V2 > 0.46: [S1]
##
## SubTree [S1]
##
## V2 <= 0.52: CYT (10.7/1.5)
## V2 > 0.52:
## :...V8 <= 0.5: MIT (9/1.9)
##     V8 > 0.5: CYT (6.2/2.4)
##
## ----- Trial 7: -----
##
## Decision tree:
##
## V4 <= 0.45:
## :...V4 <= 0.36: MEM (36.8/5.1)
## :   V4 > 0.36:
## :     :...V5 > 0.27:
## :       :...V3 <= 0.4: NUC (8.2/2.2)
## :       :   V3 > 0.4:
## :       :     :...V9 > 0.29:
## :       :       :...V4 <= 0.43: MEM (9.4/4.4)
## :       :       :   V4 > 0.43: NUC (4.1/1.2)
## :       :       :     V9 <= 0.29:
## :       :       :       :...V8 <= 0.53: MIT (32.5/9.5)
## :       :       :       :   V8 > 0.53:
## :       :       :       :     :...V2 <= 0.59: NUC (3.7)
## :       :       :       :     :   V2 > 0.59: MIT (10/4.3)
## :       :       :     V5 <= 0.27:
## :       :       :       :...V4 <= 0.39: MEM (16.8/1.7)
## :       :       :       :   V4 > 0.39:
## :       :       :       :     :...V3 <= 0.54:
## :       :       :       :       :...V5 > 0.24: NUC (4.5)
## :       :       :       :       :   V5 <= 0.24:
## :       :       :       :       :     :...V3 <= 0.48:
## :       :       :       :       :       :...V2 <= 0.36: MEM (4.5/2.2)
## :       :       :       :       :       :   V2 > 0.36: CYT (23.2/10.1)
## :       :       :       :       :       :     V3 > 0.48:
## :       :       :       :       :       :       :...V3 <= 0.52: MEM (9.9/3.2)
## :       :       :       :       :       :       :   V3 > 0.52: NUC (6.7/1.4)
## :       :       :     V3 > 0.54:
## :       :       :       :...V3 > 0.73: NUC (4.4/0.6)
## :       :       :       :   V3 <= 0.73:
## :       :       :       :     :...V9 > 0.26: MEM (5.5/2.4)
## :       :       :       :     :   V9 <= 0.26:
## :       :       :       :     :     :...V3 > 0.66: MEM (3.3)
## :       :       :       :     :     :   V3 <= 0.66:
## :       :       :       :     :     :     :...V4 <= 0.44: MEM (13.9/6.1)
## :       :       :       :     :     :     :   V4 > 0.44: CYT (9.4/3.2)
## :       :     V4 > 0.45:

```

```

## :...V2 > 0.7:
## :...V4 > 0.58: NUC (5.9/2.6)
## : V4 <= 0.58:
## : :...V3 > 0.78: CYT (2.5)
## : V3 <= 0.78:
## : :...V3 <= 0.5: CYT (2.3)
## : V3 > 0.5: MEM (15.6/4.8)
## V2 <= 0.7:
## :...V5 > 0.39:
## :...V2 <= 0.35: NUC (2.7)
## : V2 > 0.35:
## : :...V9 > 0.37: NUC (11.7/6)
## : V9 <= 0.37:
## : :...V5 > 0.82: CYT (4.8/2)
## : V5 <= 0.82:
## : :...V8 <= 0.36: NUC (2.5/0.9)
## : V8 > 0.36:
## : :...V4 > 0.57: CYT (13.2/4.7)
## : V4 <= 0.57:
## : :...V3 <= 0.39: CYT (8.5/3)
## : V3 > 0.39: MIT (56.2/11.4)
## V5 <= 0.39:
## :...V9 > 0.31:
## :...V8 <= 0.52:
## : :...V8 <= 0.45: CYT (11.2/4)
## : : V8 > 0.45: NUC (78/23.8)
## : V8 > 0.52:
## : :...V5 > 0.25: CYT (8.4/5.5)
## : V5 <= 0.25:
## : :...V9 <= 0.37: NUC (9.7/1.5)
## : V9 > 0.37:
## : :...V8 > 0.58: NUC (2.1)
## : V8 <= 0.58:
## : :...V3 > 0.5: CYT (8.1)
## : V3 <= 0.5:
## : :...V4 <= 0.47: NUC (3.8)
## : V4 > 0.47:
## : :...V5 <= 0.16: NUC (8.8/2.1)
## : V5 > 0.16: CYT (6.4)
## V9 <= 0.31:
## :...V4 <= 0.5:
## :...V2 <= 0.37:
## : :...V5 <= 0.11: CYT (2.8)
## : : V5 > 0.11: NUC (26.3/11.3)
## : V2 > 0.37:
## : :...V4 <= 0.47:
## : :...V3 <= 0.58: CYT (30.7/17.6)
## : : V3 > 0.58: MIT (5.2/1.1)
## : V4 > 0.47:
## : :...V4 <= 0.48: CYT (14.7/5)
## : V4 > 0.48:
## : :...V9 > 0.28: NUC (4.7/0.9)
## : V9 <= 0.28:
## : :...V2 > 0.52: CYT (12.4/3.7)

```

```

##          :          V2 <= 0.52:
##          :          :...V5 <= 0.22: NUC (12.8/4.3)
##          :          V5 > 0.22: MIT (12.3/6.2)
## V4 > 0.5:
## :...V3 <= 0.3:
##       :...V8 <= 0.53: NUC (14.6/3.9)
##       :   V8 > 0.53: MIT (4.2/1.1)
##       V3 > 0.3:
##       :...V2 <= 0.4:
##       :...V2 > 0.37:
##       :   :...V3 <= 0.38: NUC (3.3)
##       :   :   V3 > 0.38: CYT (14.3)
##       :   V2 <= 0.37:
##       :   :...V2 <= 0.32: CYT (17.5/3.1)
##       :   V2 > 0.32:
##       :   :...V4 > 0.54: NUC (10.1/1.9)
##       :   V4 <= 0.54:
##       :   :...V9 > 0.26: NUC (2.6)
##       :   V9 <= 0.26:
##       :   :...V5 <= 0.17: NUC (9.1/2)
##       :   V5 > 0.17: CYT (4.7)
##       V2 > 0.4:
##       :...V5 <= 0.2:
##       :...V2 <= 0.42: CYT (17.9/9.9)
##       :   V2 > 0.42:
##       :   :...V4 <= 0.53:
##       :   :   :...V5 <= 0.16: CYT (14.6/2.1)
##       :   :   V5 > 0.16:
##       :   :   :...V5 > 0.19: NUC (5.5/1)
##       :   :   V5 <= 0.19:
##       :   :   :...V8 <= 0.49: NUC (12.3/3)
##       :   :   V8 > 0.49: CYT (9.4/2.5)
##       :   V4 > 0.53:
##       :   :...V5 > 0.18: CYT (12.2/1.7)
##       :   V5 <= 0.18:
##       :   :...V5 > 0.17:
##       :   :   :...V3 <= 0.48: CYT (8.5)
##       :   :   V3 > 0.48: NUC (6.6/0.9)
##       :   V5 <= 0.17:
##       :   :...V3 > 0.52: CYT (13.3/4)
##       :   V3 <= 0.52:
##       :   :...V2 <= 0.59: NUC (33.1/16.1)
##       :   V2 > 0.59: MIT (3.9/0.9)
##       V5 > 0.2:
##       :...V8 <= 0.42: CYT (7.5/1.4)
##       V8 > 0.42:
##       :...V3 <= 0.38: NUC (6.8/0.9)
##       V3 > 0.38:
##       :...V4 > 0.61: MIT (6.1)
##       V4 <= 0.61:
##       :...V8 <= 0.44: MIT (10.5/3.7)
##       V8 > 0.44:
##       :...V5 > 0.36: NUC (7.6/3)
##       V5 <= 0.36:

```

```

##                                     :...V5 > 0.28:
##                                     :...V5 > 0.35: CYT (4.2)
##                                     :   V5 <= 0.35:
##                                     :   :...V4 <= 0.57: CYT (15.5/6.5)
##                                     :       V4 > 0.57: MIT (5.9/1.5)
##                                     V5 <= 0.28:
##                                     :...V9 > 0.27: NUC (5.7/2.8)
##                                     V9 <= 0.27:
##                                     :...V2 <= 0.45: NUC (5.8/2.5)
##                                     V2 > 0.45: [S1]
##
## SubTree [S1]
##
## V5 <= 0.27: CYT (31.2/10)
## V5 > 0.27: NUC (2.2)
##
## ----- Trial 8: -----
##
## Decision tree:
##
## V4 <= 0.46:
## :...V4 <= 0.36: MEM (27.1)
## :   V4 > 0.36:
## :     :...V2 > 0.62:
## :       :...V3 > 0.65: MEM (17.2/1.2)
## :       :   V3 <= 0.65:
## :       :     :...V4 <= 0.41: MEM (11.1/5)
## :       :       V4 > 0.41:
## :       :         :...V3 <= 0.61: MIT (9.1/2.1)
## :       :         V3 > 0.61: CYT (5.4/1.1)
## :       V2 <= 0.62:
## :         :...V5 > 0.32:
## :           :...V4 <= 0.41:
## :           :   :...V9 <= 0.29: MIT (11.8/4.7)
## :           :       V9 > 0.29: MEM (7.9/5.1)
## :           :   V4 > 0.41:
## :           :     :...V5 <= 0.4: NUC (2.4)
## :           :       V5 > 0.4:
## :           :         :...V8 <= 0.53: MIT (11.1/0.8)
## :           :         V8 > 0.53: NUC (2.1)
## :           V5 <= 0.32:
## :             :...V4 <= 0.43:
## :             :   :...V9 <= 0.26: MEM (30.2/12.7)
## :             :       V9 > 0.26:
## :             :     :...V4 > 0.42: NUC (4.6)
## :             :       V4 <= 0.42:
## :             :         :...V2 <= 0.49: NUC (11.7/3.8)
## :             :         V2 > 0.49: MEM (3)
## :             V4 > 0.43:
## :             :...V2 <= 0.4: NUC (23.2/9.1)
## :             V2 > 0.4:
## :             :...V5 > 0.24: CYT (11.6/6)
## :             V5 <= 0.24:
## :             :...V3 > 0.58: MIT (11.4/5.4)

```

```

## :                               V3 <= 0.58:
## :                               :...V2 <= 0.48: MEM (13.2/2.8)
## :                               V2 > 0.48: CYT (13.5/6.6)
## V4 > 0.46:
## :...V5 > 0.37:
## :...V2 > 0.83: MEM (5.3/2.6)
## :   V2 <= 0.83:
## :   :...V3 > 0.52:
## :       :...V5 <= 0.82: MIT (32.3/3.5)
## :       :   V5 > 0.82: NUC (2.5)
## :       V3 <= 0.52:
## :       :...V9 > 0.32:
## :           :...V3 > 0.46: CYT (5.7/2.9)
## :           :   V3 <= 0.46:
## :           :   :...V4 <= 0.58: NUC (8.6/1.6)
## :           :       V4 > 0.58: CYT (4.7/0.7)
## :           V9 <= 0.32:
## :           :...V5 <= 0.39: CYT (6.6/2.8)
## :           V5 > 0.39:
## :           :...V5 > 0.87: CYT (2.3)
## :           V5 <= 0.87:
## :           :...V8 <= 0.41: CYT (3.4)
## :           V8 > 0.41: MIT (44/17.4)
## V5 <= 0.37:
## :...V2 > 0.7:
## :...V8 <= 0.54: MEM (12/5.7)
## :   V8 > 0.54: CYT (3.8/0.3)
## V2 <= 0.7:
## :...V9 > 0.3:
## :...V8 <= 0.53: NUC (111.5/47.5)
## :   V8 > 0.53:
## :   :...V3 > 0.55: NUC (7.2/3.2)
## :   V3 <= 0.55:
## :   :...V5 > 0.28: MIT (5.2/1.9)
## :   V5 <= 0.28:
## :   :...V3 <= 0.31: NUC (2.5/1.1)
## :   V3 > 0.31:
## :   :...V4 <= 0.49: NUC (5.6/1.1)
## :   V4 > 0.49: CYT (12.6/1.2)
## V9 <= 0.3:
## :...V2 <= 0.31:
## :...V4 > 0.63: MIT (2.5/0.5)
## :   V4 <= 0.63:
## :   :...V2 <= 0.2: MEM (2.4)
## :   V2 > 0.2: CYT (21/3.4)
## V2 > 0.31:
## :...V4 <= 0.47:
## :...V8 <= 0.44: MIT (2.5)
## :   V8 > 0.44:
## :   :...V3 <= 0.56: NUC (22.7/12.1)
## :   V3 > 0.56: MIT (3.9/1.6)
## V4 > 0.47:
## :...V3 <= 0.32:
## :...V8 <= 0.41: MIT (2.3)

```

```

##      : V8 > 0.41: NUC (24.5/7.6)
## V3 > 0.32:
##      :...V5 <= 0.2:
##          :...V4 <= 0.53:
##              : :...V3 > 0.65: MIT (5/2.4)
##              : : V3 <= 0.65:
##              : : :...V4 <= 0.52:
##              : : :...V5 <= 0: NUC (2)
##              : : : V5 > 0: CYT (60.2/22.9)
##              : : V4 > 0.52:
##              : : :...V9 <= 0.23: CYT (9.6/3.8)
##              : : V9 > 0.23: NUC (11.1/1.3)
##          : V4 > 0.53:
##          : :...V3 > 0.59: MIT (9.2/3.5)
##          : V3 <= 0.59:
##          : :...V9 > 0.22:
##          : : :...V8 > 0.53: CYT (4.1)
##          : : V8 <= 0.53:
##          : : :...V4 > 0.65: CYT (2.9)
##          : : V4 <= 0.65:
##          : : :...V2 <= 0.58: NUC (11.1/2.3)
##          : : V2 > 0.58: CYT (3.3)
##          : V9 <= 0.22:
##          : :...V4 <= 0.55:
##          : : :...V3 <= 0.55: CYT (15.6/4.5)
##          : : V3 > 0.55: NUC (3.7)
##          : V4 > 0.55:
##          : :...V8 > 0.55: CYT (6.7)
##          : V8 <= 0.55:
##          : :...V2 <= 0.37: MIT (4.7/2.3)
##          : V2 > 0.37:
##          : :...V8 <= 0.48: CYT (14.9/4.2)
##          : V8 > 0.48:
##          : :...V5 > 0.16: CYT (13.4/3.7)
##          : V5 <= 0.16: [S1]
## V5 > 0.2:
##      :...V3 <= 0.39:
##          :...V8 <= 0.51: NUC (9.8/2.2)
##          : V8 > 0.51: CYT (4.5/0.8)
##      V3 > 0.39:
##          :...V3 <= 0.44:
##              :...V2 <= 0.37: NUC (3.1/0.9)
##              : V2 > 0.37:
##              : :...V5 <= 0.24: CYT (8)
##              : V5 > 0.24:
##              : :...V8 <= 0.49: MIT (8.5/1)
##              : V8 > 0.49: CYT (6.3/2)
##          V3 > 0.44:
##          :...V8 <= 0.3: MIT (2.6)
##          V8 > 0.3:
##          :...V4 <= 0.48: NUC (3.6)
##          V4 > 0.48:
##          :...V4 <= 0.49: CYT (8.7/2.2)
##          V4 > 0.49:

```

```

##                                     :...V9 > 0.28: MIT (7.2/2.6)
##                                     V9 <= 0.28:
##                                     :...V9 <= 0.11: CYT (2.3)
##                                     V9 > 0.11:
##                                     :...V4 > 0.59: MIT (7.9/3.9)
##                                     V4 <= 0.59: [S2]
##
## SubTree [S1]
##
## V5 <= 0.11: CYT (2)
## V5 > 0.11: MIT (12.8/3.3)
##
## SubTree [S2]
##
## V8 > 0.52: CYT (17.6/7.1)
## V8 <= 0.52:
## :...V2 <= 0.55: NUC (31.4/9.8)
##     V2 > 0.55: CYT (13.7/5)
##
## ----- Trial 9: -----
##
## Decision tree:
##
## V4 <= 0.45:
## :...V4 <= 0.36: MEM (23.4)
## :   V4 > 0.36:
## :     :...V2 > 0.62:
## :       :...V3 > 0.65: MEM (15.4/1.7)
## :       :   V3 <= 0.65:
## :       :     :...V2 <= 0.78: MIT (23.1/10.6)
## :       :     :   V2 > 0.78: MEM (3.4)
## :       V2 <= 0.62:
## :         :...V5 > 0.32:
## :           :...V9 <= 0.29: MIT (21.4/8.2)
## :           :   V9 > 0.29: NUC (13/6.7)
## :           V5 <= 0.32:
## :             :...V4 <= 0.39:
## :               :...V9 <= 0.33: MEM (9.2)
## :               :   V9 > 0.33: NUC (4.8/1.3)
## :               V4 > 0.39:
## :                 :...V8 <= 0.43: CYT (3.4)
## :                 :   V8 > 0.43:
## :                 :     :...V3 > 0.58:
## :                 :       :...V5 <= 0.22: MEM (15.4/7)
## :                 :       :   V5 > 0.22: NUC (8.2/3.5)
## :                 :       V3 <= 0.58:
## :                 :         :...V5 > 0.22:
## :                 :           :...V3 <= 0.32: MEM (3.1)
## :                 :           :   V3 > 0.32: NUC (12.4/3.3)
## :                 :           V5 <= 0.22:
## :                 :             :...V8 <= 0.46: MEM (2.8)
## :                 :             :   V8 > 0.46:
## :                 :             :     :...V3 > 0.48: NUC (17.4/7)
## :                 :             :     :   V3 <= 0.48:

```

```

## :                               :...V5 <= 0.13: NUC (2.9)
## :                               V5 > 0.13:
## :                               :...V2 <= 0.36: MEM (4.5/2.2)
## :                               V2 > 0.36: CYT (19.9/7.2)
## V4 > 0.45:
## :...V5 > 0.37:
##   :...V2 > 0.83: NUC (5.8/2.3)
##   :   V2 <= 0.83:
##   :   :...V3 <= 0.52: CYT (76.8/37.7)
##   :       V3 > 0.52:
##   :       :...V2 <= 0.36: NUC (2.7)
##   :           V2 > 0.36:
##   :           :...V5 <= 0.82: MIT (25.3)
##   :               V5 > 0.82: NUC (2.2)
## V5 <= 0.37:
## :...V2 > 0.7:
##   :...V3 <= 0.5: CYT (2.9)
##   :   V3 > 0.5:
##   :   :...V8 <= 0.54: MEM (10.6/5.2)
##   :       V8 > 0.54: CYT (3.8/0.7)
## V2 <= 0.7:
## :...V4 <= 0.5:
##   :...V9 > 0.28:
##   :   :...V2 > 0.47:
##   :   :   :...V2 <= 0.63: CYT (26.3/4.3)
##   :   :       V2 > 0.63: NUC (3.3)
##   :   :   V2 <= 0.47:
##   :   :   :...V5 > 0.21: NUC (12.8/5.5)
##   :   :       V5 <= 0.21:
##   :   :       :...V4 <= 0.46: MEM (3.9/2.3)
##   :   :           V4 > 0.46:
##   :   :           :...V4 <= 0.48: NUC (5.5)
##   :   :               V4 > 0.48:
##   :   :               :...V2 <= 0.45: CYT (15.5/4.5)
##   :   :                   V2 > 0.45: NUC (2.6)
##   :   V9 <= 0.28:
##   :   :...V2 <= 0.23: MEM (3.6/1.5)
##   :       V2 > 0.23:
##   :       :...V8 <= 0.44: MIT (8.7/2.2)
##   :           V8 > 0.44:
##   :           :...V8 > 0.53: CYT (12.3/1.8)
##   :               V8 <= 0.53:
##   :               :...V3 > 0.65: MIT (3.3)
##   :                   V3 <= 0.65:
##   :                   :...V4 <= 0.47:
##   :                       :...V3 <= 0.46:
##   :                       :   :...V9 <= 0.26: MEM (10.7/5.7)
##   :                       :       V9 > 0.26: MIT (4.8/1.8)
##   :                       :   V3 > 0.46:
##   :                       :   :...V3 > 0.58: CYT (4.1/2)
##   :                       :       V3 <= 0.58:
##   :                       :       :...V9 <= 0.23: CYT (12.2/3.6)
##   :                       :           V9 > 0.23: NUC (4)
##   :                       V4 > 0.47:

```



```

##          :          :...V5 > 0.22:
##          :          :...V4 <= 0.48: NUC (2.1)
##          :          :   V4 > 0.48: CYT (14.4/5.7)
##          :          V5 <= 0.22:
##          :          :...V8 > 0.52: NUC (4.3)
##          :          V8 <= 0.52:
##          :          :...V8 <= 0.48: NUC (9.7/3.1)
##          :          V8 > 0.48:
##          :          :...V5 <= 0.18: CYT (11.4/1.3)
##          :          V5 > 0.18: NUC (5.6/1.4)
## V4 > 0.5:
## :...V9 > 0.48: NUC (19.5/2.5)
##   V9 <= 0.48:
##     :...V4 <= 0.51:
##       :...V5 <= 0.14: CYT (7.1)
##       :   V5 > 0.14:
##       :   :...V8 <= 0.49: NUC (9.2/0.5)
##       :   V8 > 0.49:
##       :   :...V9 > 0.34: NUC (2.5)
##       :   V9 <= 0.34:
##       :   :...V9 <= 0.19: NUC (2.3)
##       :   V9 > 0.19:
##       :   :...V3 <= 0.34: NUC (2.4)
##       :   V3 > 0.34: CYT (16/4.1)
## V4 > 0.51:
## :...V5 <= 0.19:
##   :...V3 > 0.59: CYT (14.5/5.1)
##   :   V3 <= 0.59:
##   :   :...V5 <= 0.14:
##   :   :   :...V5 <= 0.11:
##   :   :   :   :...V4 > 0.56: NUC (5.8)
##   :   :   :   :   V4 <= 0.56:
##   :   :   :   :   :...V4 <= 0.55: NUC (10/3)
##   :   :   :   :   V4 > 0.55: CYT (4.7)
##   :   :   :   V5 > 0.11:
##   :   :   :   :...V9 > 0.24: NUC (12.7/3.3)
##   :   :   :   V9 <= 0.24:
##   :   :   :   :...V3 > 0.46: CYT (10.1/2.8)
##   :   :   :   V3 <= 0.46:
##   :   :   :   :...V5 <= 0.12: MIT (5.5/1.4)
##   :   :   :   V5 > 0.12: NUC (11.3/4.1)
##   :   V5 > 0.14:
##   :   :...V2 > 0.58: CYT (11.8/1.1)
##   :   V2 <= 0.58:
##   :   :...V8 <= 0.42: NUC (13.1/6.4)
##   :   V8 > 0.42:
##   :   :...V8 <= 0.48: CYT (17.8/2.3)
##   :   V8 > 0.48:
##   :   :...V8 <= 0.52: NUC (35.1/13.7)
##   :   V8 > 0.52: CYT (24.7/7)
## V5 > 0.19:
## :...V8 <= 0.42: CYT (12.1/4)
##   V8 > 0.42:
##   :...V2 <= 0.37: NUC (18.8/5.3)

```

```

##          V2 > 0.37:
##          :...V3 <= 0.39:
##              :...V5 <= 0.33: NUC (12.3/3.1)
##              :   V5 > 0.33: CYT (3.1)
##          V3 > 0.39:
##          :...V3 <= 0.44:
##              :...V5 <= 0.25: CYT (5.9)
##              :   V5 > 0.25: MIT (10/2.1)
##          V3 > 0.44:
##          :...V2 <= 0.43: MIT (9.2/3.7)
##              V2 > 0.43:
##              :...V8 <= 0.45:
##                  :...V2 <= 0.54: NUC (8.7/2.9)
##                  :   V2 > 0.54: MIT (4.7)
##              V8 > 0.45:
##              :...V3 <= 0.45: MIT (10.8/4.3)
##                  V3 > 0.45:
##                  :...V4 <= 0.52: NUC (10.1/2.3)
##                      V4 > 0.52:
##                      :...V5 > 0.3: CYT (12.6/4.8)
##                          V5 <= 0.3: [S1]
##
## SubTree [S1]
##
## V9 <= 0.24: CYT (23.5/7.9)
## V9 > 0.24: NUC (11.2/2.6)
##
##
## Evaluation on training data (919 cases):
##
## Trial          Decision Tree
## -----
##      Size      Errors
##
##      0      96 173(18.8%)
##      1      69 232(25.2%)
##      2      71 245(26.7%)
##      3      71 232(25.2%)
##      4      73 247(26.9%)
##      5      75 269(29.3%)
##      6      94 248(27.0%)
##      7      79 223(24.3%)
##      8      78 231(25.1%)
##      9      82 236(25.7%)
## boost          48( 5.2%)  <<
##
##
##      (a)  (b)  (c)  (d)  <-classified as
##      ----  ----  ----  ----
##      283    1    8    12  (a): class CYT
##      2    171    1    4  (b): class MEM
##      3     4   155    2  (c): class MIT
##      8     3         262 (d): class NUC
##

```

```
##
## Attribute usage:
##
## 100.00% V4
## 100.00% V5
## 99.78% V9
## 97.39% V3
## 91.29% V2
## 88.90% V8
## 9.47% V7
## 4.03% V6
##
##
## Time: 0.1 secs
```

En este nuevo modelo observamos que se da una mayor importancia a todos los datos de manera general.

Ahora realizamos la predicción utilizando el modelo recién creado:

```
predicted_tree_m_10 <- predict(tree_m_10,test_tree)
```

Finalmente observamos el rendimiento del nuevo modelo con otra matriz de confusión

```
confusionMatrix(predicted_tree_m_10, test_tree_label)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction CYT MEM MIT NUC
##           CYT  81  2  24  57
##           MEM   1 69   8   1
##           MIT  13  3  41  12
##           NUC  45  6   7  83
##
## Overall Statistics
##
##           Accuracy : 0.6049
##           95% CI : (0.5582, 0.6502)
##           No Information Rate : 0.3377
##           P-Value [Acc > NIR] : < 2e-16
##
##           Kappa : 0.4552
##
## Mcnemar's Test P-Value : 0.05817
##
## Statistics by Class:
##
##           Class: CYT Class: MEM Class: MIT Class: NUC
## Sensitivity           0.5786      0.8625      0.51250      0.5425
## Specificity           0.7348      0.9732      0.92493      0.8067
## Pos Pred Value        0.4939      0.8734      0.59420      0.5887
## Neg Pred Value        0.7958      0.9706      0.89844      0.7756
## Prevalence            0.3091      0.1766      0.17660      0.3377
## Detection Rate        0.1788      0.1523      0.09051      0.1832
## Detection Prevalence  0.3620      0.1744      0.15232      0.3113
## Balanced Accuracy      0.6567      0.9178      0.71872      0.6746
```

Comparando este modelo con el anterior no se encuentran diferencias significativas en su rendimiento, por lo que la adición de 10 árboles de decisión no parece mejorar el rendimiento del modelo para nuestros datos.

Modelo random forest

Para intentar mejorar el modelo del Árbol de decisión, ahora utilizaremos el paquete `randomForest` para utilizar el modelo de random forest. En este modelo realizaremos 500 árboles de decisión (el valor estándar) y en cada partición del árbol de decisión se seleccionarán 2 características (esto se hace de manera automática recogiendo como estándar la raíz cuadrada de las características totales y redondeando por lo bajo, en este caso 8).

```
set.seed(12345)
rf_m <- randomForest(train_tree, train_tree_label, ntree = 500)
```

Ahora podemos observar el resumen del modelo:

```
rf_m

##
## Call:
##  randomForest(x = train_tree, y = train_tree_label, ntree = 500)
##                Type of random forest: classification
##                Number of trees: 500
## No. of variables tried at each split: 2
##
##           OOB estimate of  error rate: 36.56%
## Confusion matrix:
##      CYT MEM MIT NUC class.error
## CYT 190  11  22  81  0.3750000
## MEM   9 150   4  15  0.1573034
## MIT  50  12  89  13  0.4573171
## NUC  91  16  12 154  0.4358974
```

Observamos que el error estimado es del 36.5%, además de los porcentajes de error dividido por clase. De esta manera sabemos que las clases MIT y NUC serán las más propensas a reducir la precisión del modelo.

Realizamos ahora la predicción del test dataset:

```
predicted_rf_m <- predict(rf_m, test_tree)
```

Finalmente realizamos la matriz de confusión de este modelo:

```
confusionMatrix(predicted_rf_m, test_tree_label)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction  CYT MEM MIT NUC
##      CYT 102   2  27  52
##      MEM   2  72   9   3
##      MIT   5   0  41  10
##      NUC  31   6   3  88
##
## Overall Statistics
##
##              Accuracy : 0.6689
##              95% CI : (0.6234, 0.7121)
##      No Information Rate : 0.3377
```

```

##      P-Value [Acc > NIR] : < 2.2e-16
##
##      Kappa : 0.5429
##
##      McNemar's Test P-Value : 6.134e-06
##
## Statistics by Class:
##
##      Class: CYT Class: MEM Class: MIT Class: NUC
## Sensitivity      0.7286      0.9000      0.51250      0.5752
## Specificity      0.7412      0.9625      0.95979      0.8667
## Pos Pred Value   0.5574      0.8372      0.73214      0.6875
## Neg Pred Value   0.8593      0.9782      0.90176      0.8000
## Prevalence       0.3091      0.1766      0.17660      0.3377
## Detection Rate   0.2252      0.1589      0.09051      0.1943
## Detection Prevalence 0.4040      0.1898      0.12362      0.2826
## Balanced Accuracy 0.7349      0.9312      0.73614      0.7209

```

En el modelo de random forest observamos que la precisión global es del 67%. Este modelo es significativamente mejor que el modelo de Árbol de decisión utilizado anteriormente. Comparando de manera más detallada, observamos un aumento significativo en la sensibilidad de la clase CYT junto con un aumento leve de las clases MEM y NUC. Respecto a la especificidad, se observa un aumento significativo en las clases MIT y NUC y una leve disminución en las clases CYT y MEM. En las precisiones balanceadas de las clases observamos también un aumento significativo en todas ellas en este nuevo modelo, siendo el aumento más destacado la clase CYT.

Conclusiones

Ahora vamos a comentar los resultados de todos los modelos utilizados, de menor rendimiento a mayor.

El modelo con el peor rendimiento ha sido el modelo en el que se ha utilizado el algoritmo de Naive Bayes. En este caso el mejor rendimiento del modelo es del 59%, donde habíamos dividido los datos en 10 grupos. En este caso suponemos que este modelo en principio no está pensado para los datos a predecir, ya que no hemos tenido que procesar los datos de más al tener las características como numéricas.

El siguiente modelo que encontramos es el que utiliza el algoritmo del árbol de decisión, donde el rendimiento de este es del 60%. Los problemas que pueden haberse encontrado en este modelo son principalmente el peso que se le daba a cada característica a la hora de realizar cada árbol de decisión.

El modelo en el que se utiliza el algoritmo de K-NN ha obtenido una precisión del 62%, siendo el parámetro más preciso el que tiene una $k = 29$. Al intentar aumentar o disminuir este valor hemos observado que empezaba a disminuir el rendimiento del modelo, por lo que esta es la máxima precisión que se puede alcanzar en este algoritmo. Este algoritmo tiene como ventaja que es rápido y fácil de implementar, pero al tener un bajo nivel de características ha disminuido consecuentemente la precisión del modelo de manera significativa.

El modelo que utiliza el algoritmo de SVM y el kernel lineal ha obtenido una precisión del 63%, lo cual lo hace el tercer mejor modelo de esta PEC. En este PEC hemos probado los kernels lineal y radial, teniendo el kernel lineal un ligeramente mejor rendimiento, pero con un mayor conocimiento sobre los distintos kernels y los parámetros de este algoritmo, creo que se podría obtener un aumento significativo del rendimiento de este modelo.

El modelo en el que se utiliza la red neuronal artificial (ANN) tiene un rendimiento del 65% cuando hemos utilizado 10 nodos ocultos y modificado el umbral de error al 0.5. En este modelo de 10 nodos ocultos no tiene un coste computacional elevado pero requiere de casi 60000 pasos para realizar el modelo, por lo que estamos frente a un coste computacional significativo. Intentando aumentar el número de nodos requiere aumentar el límite máximo de pasos para crear el modelo o aumentar el umbral de error. Aumentando el

límite máximo de pasos da lugar a un coste computacional que va aumentando exponencialmente según el mayor número de pasos realizados. En cambio, si aumentamos el umbral de error, el coste computacional no se vería afectado pero sí disminuiría la precisión del modelo conforme fuéramos aumentando este umbral. Por lo tanto, este modelo tiene un margen de mejora para aumentar su precisión y rendimiento, pero a costa de un mayor tiempo de creación de este.

Finalmente, el modelo que mejor resultado nos ha dado, es decir, el modelo con mejor rendimiento, ha sido el que utiliza el algoritmo de random forests, obteniendo un 67% de precisión. Este modelo ha tenido un coste computacional muy bajo para el buen resultado obtenido de casi un 70%. Aun así, en términos generales no hemos conseguido encontrar un modelo con una precisión del 70% o mayor. Esto puede ser debido a la falta de registros para el train dataset o la necesidad de más variables de características para aumentar el número de factores que puedan relacionarse con cada localización de la secuencia. Otro factor que puede haber afectado es la posible varianza de los resultados y el propio margen de error en cada variable, ya que las variables de reconocimiento de señal y las puntuaciones o scores de cada secuencia no sabemos si han sido correctamente obtenidas o que reflejen el resultado real (no sabemos si se ha realizado el triplicado de cada señal para obtener su media, por ejemplo).

Como conclusión final, decir que el mejor modelo utilizado en la PEC ha sido el random forest con un 67% de precisión.