Analisis de Algoritmos

Jose Luis Osorio & Melisa Garcia

4 de mayo de 2017

Introducción

En este análisis se mostrara un resultado conjunto al unir los cuatro datasets obtenidos por los scripts y los datos sacados de cada URL. Se hará una limpieza general por todo los dataset y se unirán los datos a medida que se van limpiando. Luego de obtener los datasets benigno y maligno se procederá a hacer una combinación de estos para así tener un conjunto de datos general y poder aplicar los algoritmos de machine learning.

Importar los dataset

Se procede a importar cada uno de los datasets

Asignación de nombres de las columnas

Se procede a poner el nombre a cada una de las matrices según lo mencionado en cada una de las capas.

```
# Asignacion de nombres
names(app_maligno) <- c("URL","URL_LENGTH","NUMBER_SPECIAL_CHARACTERS","CHARSET","SERVER",
                        "CACHE_CONTROL","CONTENT_LENGTH","WHOIS_COUNTRY","WHOIS_STATEPROV",
                        "WHOIS_REGDATE", "UPDATE_DATE", "WHITIN_DOMAIN")

names(app_benigno) <- c("URL","URL_LENGTH","NUMBER_SPECIAL_CHARACTERS","CHARSET","SERVER",
                        "CACHE_CONTROL","CONTENT_LENGTH","WHOIS_COUNTRY","WHOIS_STATEPROV",
                        "WHOIS_REGDATE", "UPDATE_DATE", "WHITIN_DOMAIN")

names(red_maligno) <- c("URL","TCP_CONVERSATION_EXCHANGE","DIST_REMOTE_TCP_PORT",
                        "REMOTE_IPS","APP_BYTES","UDP_PACKETS","TCP_URG_PACKETS",
                        "SOURCE_APP_PACKETS","REMOTE_APP_PACKETS","SOURCE_APP_BYTES",
                        "REMOTE_APP_BYTES","DURATION","AVG_LOCAL_PKT_RATE",
                        "AVG_REMOTE_PKT_RATE","APP_PACKETS","DNS_QUERY_TIMES")

names(red_benigno) <- c("URL","TCP_CONVERSATION_EXCHANGE","DIST_REMOTE_TCP_PORT",
                        "REMOTE_IPS","APP_BYTES","UDP_PACKETS","TCP_URG_PACKETS",
                        "SOURCE_APP_PACKETS","REMOTE_APP_PACKETS","SOURCE_APP_BYTES",
                        "REMOTE_APP_BYTES","DURATION","AVG_LOCAL_PKT_RATE",
                        "AVG_REMOTE_PKT_RATE","APP_PACKETS","DNS_QUERY_TIMES")
```

Formato de las variables

Formato para matrices de red y algunos valores de las matrices de aplicación

```
red_benigno$URL <- as.factor(red_benigno$URL)
red_benigno$TCP_CONVERSATION_EXCHANGE <- as.numeric(red_benigno$TCP_CONVERSATION_EXCHANGE)
red_benigno$DIST_REMOTE_TCP_PORT <- as.numeric(red_benigno$DIST_REMOTE_TCP_PORT)
red_benigno$REMOTE_IPS <- as.numeric(red_benigno$REMOTE_IPS)
red_benigno$APP_BYTES <- as.numeric(red_benigno$APP_BYTES)
```

```
red_benigno$UDP_PACKETS <- as.numeric(red_benigno$UDP_PACKETS)
red_benigno$TCP_URG_PACKETS <- as.numeric(red_benigno$TCP_URG_PACKETS)
red_benigno$SOURCE_APP_PACKETS <- as.numeric(red_benigno$SOURCE_APP_PACKETS)
red_benigno$REMOTE_APP_PACKETS <- as.numeric(red_benigno$REMOTE_APP_PACKETS)
red_benigno$SOURCE_APP_BYTES <- as.numeric(red_benigno$SOURCE_APP_BYTES)
red_benigno$REMOTE_APP_BYTES <- as.numeric(red_benigno$REMOTE_APP_BYTES)
red_benigno$DURATION <- as.numeric(red_benigno$DURATION)
red_benigno$AVG_LOCAL_PKT_RATE <- as.numeric(red_benigno$AVG_LOCAL_PKT_RATE)
red_benigno$AVG_REMOTE_PKT_RATE <- as.numeric(red_benigno$AVG_REMOTE_PKT_RATE)
red_benigno$APP_PACKETS <- as.numeric(red_benigno$APP_PACKETS)
red_benigno$DNS_QUERY_TIMES <- as.numeric(red_benigno$DNS_QUERY_TIMES)


red_maligno$URL <- as.factor(red_maligno$URL)
red_maligno$TCP_CONVERSATION_EXCHANGE <- as.numeric(red_maligno$TCP_CONVERSATION_EXCHANGE)
red_maligno$DIST_REMOTE_TCP_PORT <- as.numeric(red_maligno$DIST_REMOTE_TCP_PORT)
red_maligno$REMOTE_IPS <- as.numeric(red_maligno$REMOTE_IPS)
red_maligno$APP_BYTES <- as.numeric(red_maligno$APP_BYTES)
red_maligno$UDP_PACKETS <- as.numeric(red_maligno$UDP_PACKETS)
red_maligno$TCP_URG_PACKETS <- as.numeric(red_maligno$TCP_URG_PACKETS)
red_maligno$SOURCE_APP_PACKETS <- as.numeric(red_maligno$SOURCE_APP_PACKETS)
red_maligno$REMOTE_APP_PACKETS <- as.numeric(red_maligno$REMOTE_APP_PACKETS)
red_maligno$SOURCE_APP_BYTES <- as.numeric(red_maligno$SOURCE_APP_BYTES)
red_maligno$REMOTE_APP_BYTES <- as.numeric(red_maligno$REMOTE_APP_BYTES)
red_maligno$DURATION <- as.numeric(red_maligno$DURATION)
red_maligno$AVG_LOCAL_PKT_RATE <- as.numeric(red_maligno$AVG_LOCAL_PKT_RATE)
red_maligno$AVG_REMOTE_PKT_RATE <- as.numeric(red_maligno$AVG_REMOTE_PKT_RATE)
red_maligno$APP_PACKETS <- as.numeric(red_maligno$APP_PACKETS)
red_maligno$DNS_QUERY_TIMES <- as.numeric(red_maligno$DNS_QUERY_TIMES)


app_benigno$CONTENT_LENGTH <- as.numeric(app_benigno$CONTENT_LENGTH)
app_maligno$CONTENT_LENGTH <- as.numeric(app_maligno$CONTENT_LENGTH)
```

Eliminacion de variables

Por temas de correlación analizados en el informe anterior se lleva a cabo la eliminación de la variable duración, promedio de paquetes enviados desde el servidor y promedio de paquetes enviados desde el honeypot.

También se eliminan los valores de registro y actualización del servidor, dominio. Se encontraron que estos valores no aportan un significado relevante a los algoritmos de entrenamiento.

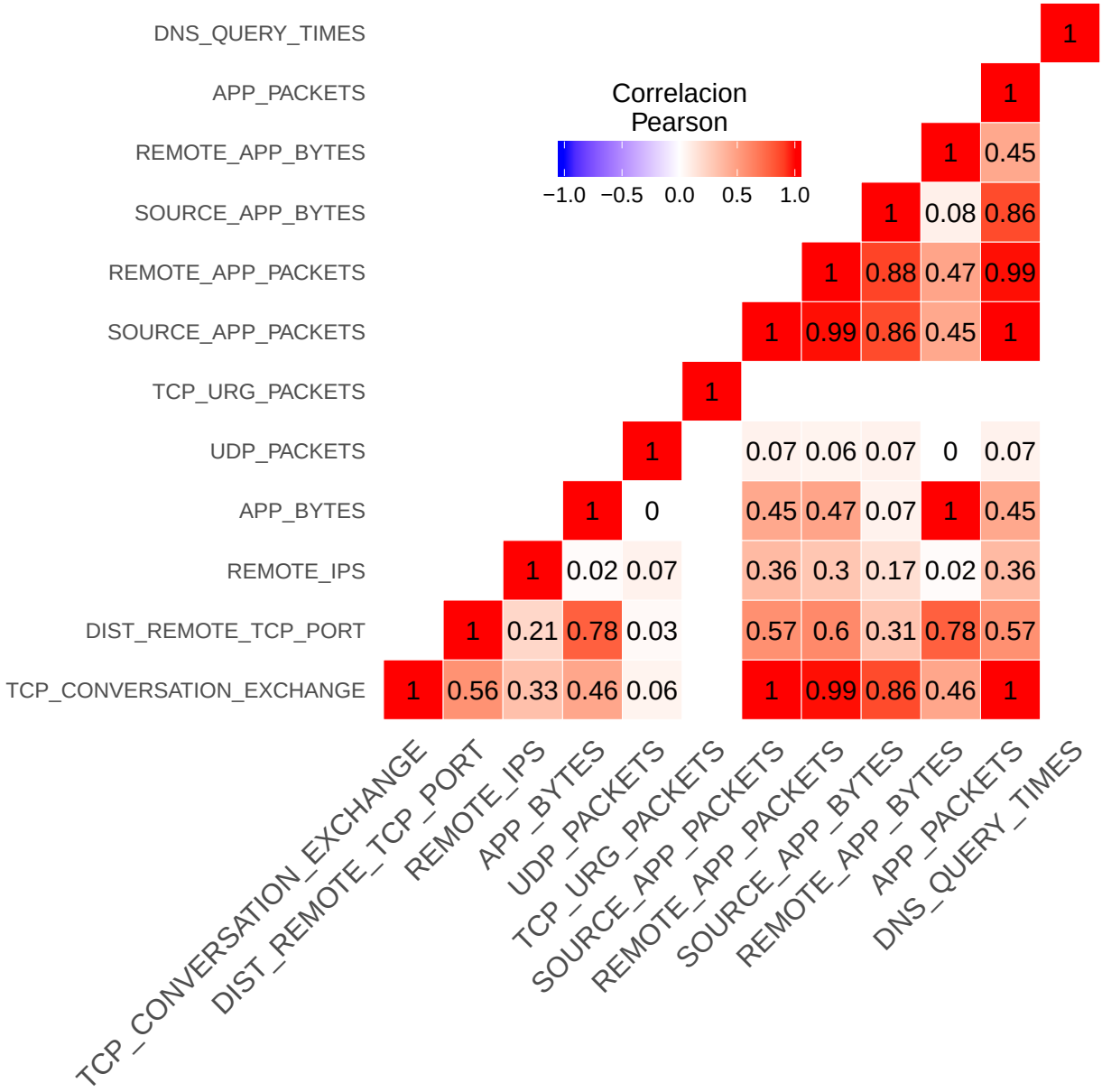
```
app_maligno <- subset(app_maligno, select=-c(WHOIS_REGDATE,UPDATE_DATE,WHITIN_DOMAIN))
app_benigno <- subset(app_benigno, select=-c(WHOIS_REGDATE,UPDATE_DATE,WHITIN_DOMAIN))

red_maligno <- subset(red_maligno,
                      select=-c(DURATION,AVG_REMOTE_PKT_RATE,
                                AVG_LOCAL_PKT_RATE))

red_benigno <- subset(red_benigno,
                      select=-c(DURATION,AVG_REMOTE_PKT_RATE,
                                AVG_LOCAL_PKT_RATE))
```

Correlacion de informacion (Red)

Red Benigno

```
library(reshape2)
library(ggplot2)

#fuente: http://www.sthda.com/english/wiki/ggplot2-quick-correlation-matrix-heatmap-r-software-and-data

get_lower_tri<-function(cormat){
    cormat[upper.tri(cormat)] <- NA
    return(cormat)
}

get_upper_tri <- function(cormat){
    cormat[lower.tri(cormat)]<- NA
    return(cormat)
}

reorder_cormat <- function(cormat){
dd <- as.dist((1-cormat)/2)
hc <- hclust(dd)
cormat <-cormat[hc$order, hc$order]
}

red_benigno.c <- round(cor(red_benigno[,2:ncol(red_benigno)]),2)

upper_tri <- get_upper_tri(red_benigno.c)

melted <- melt(upper_tri,na.rm=TRUE)


ggheatmap.b <- ggplot(melted, aes(Var2, Var1, fill = value)) +
             geom_tile(color = "white") +
             scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                                  midpoint = 0, limit = c(-1,1), space = "Lab",
                                  name="Correlacion\nPearson") +
             theme_minimal()+ # minimal theme
             theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                              size = 12, hjust = 1))+
             coord_fixed()

ggheatmap.b <-  ggheatmap.b +
            geom_text(aes(Var2, Var1, label = value), color = "black", size = 4) +
            theme(axis.title.x = element_blank(),axis.title.y = element_blank(),
                  panel.grid.major = element_blank(),
                  panel.border = element_blank(),
                  panel.background = element_blank(),
                  axis.ticks = element_blank(),
                  legend.justification = c(1, 0),
                  legend.position = c(0.6, 0.7),
                  legend.direction = "horizontal") +
            guides(fill = guide_colorbar(barwidth = 7, barheight = 1,
                                         title.position = "top", title.hjust = 0.5))
```

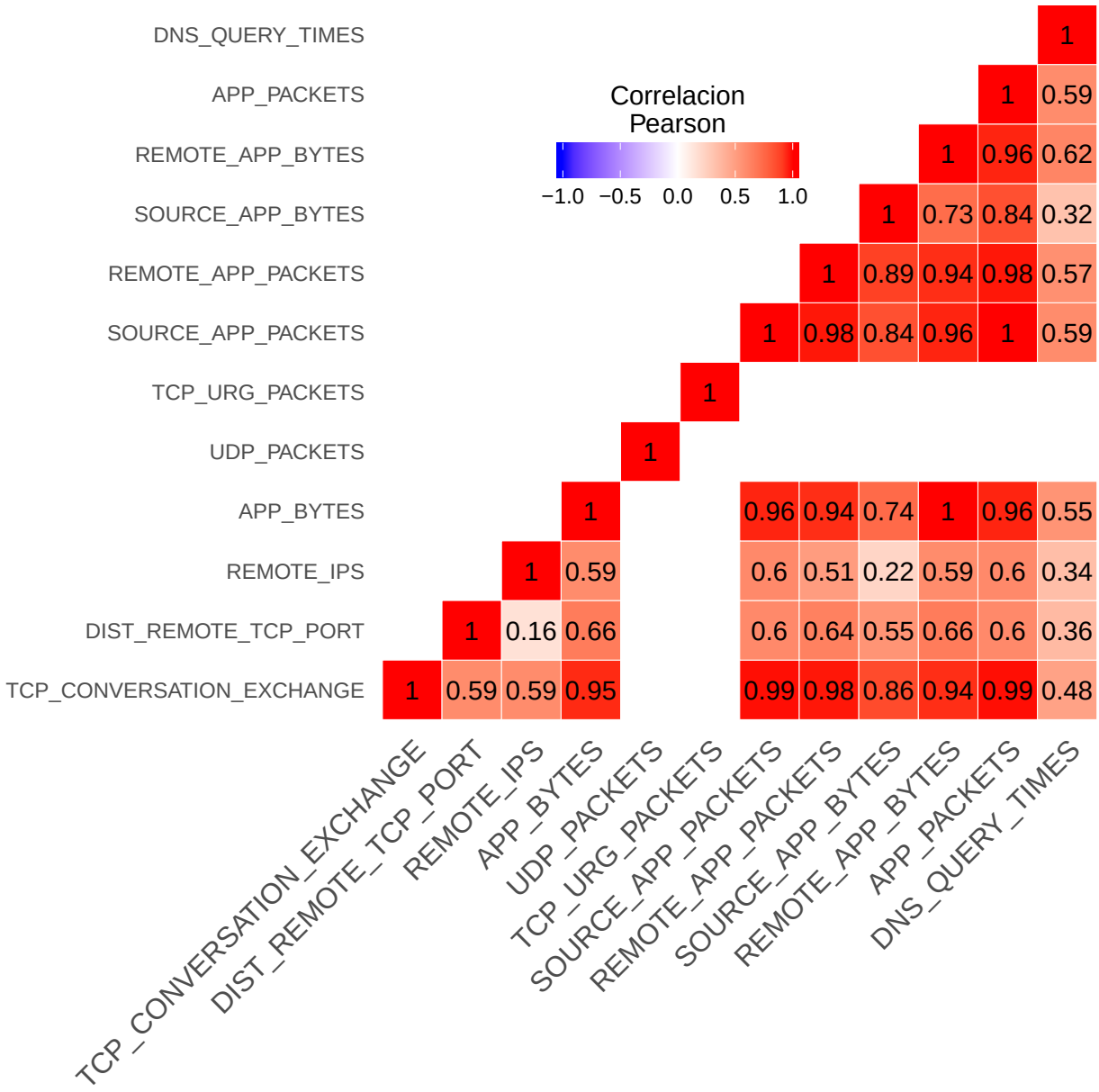
```
ggheatmap.b
```



Red Maligno

```
red_maligno.c <- round(cor(red_maligno[,2:ncol(red_maligno)]),2)

upper_tri <- get_upper_tri(red_maligno.c)

melted <- melt(upper_tri,na.rm=TRUE)
```

```r
ggheatmap.m <- ggplot(melted, aes(Var2, Var1, fill = value)) +
             geom_tile(color = "white") +
             scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                                  midpoint = 0, limit = c(-1,1), space = "Lab",
                                  name="Correlacion\nPearson") +
             theme_minimal()+ # minimal theme
             theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                              size = 12, hjust = 1))+
             coord_fixed()

ggheatmap.m <-  ggheatmap.m +
            geom_text(aes(Var2, Var1, label = value), color = "black", size = 4) +
            theme(axis.title.x = element_blank(),axis.title.y = element_blank(),
                  panel.grid.major = element_blank(),
                  panel.border = element_blank(),
                  panel.background = element_blank(),
                  axis.ticks = element_blank(),
                  legend.justification = c(1, 0),
                  legend.position = c(0.6, 0.7),
                  legend.direction = "horizontal") +
            guides(fill = guide_colorbar(barwidth = 7, barheight = 1,
                                         title.position = "top", title.hjust = 0.5))

ggheatmap.m
```

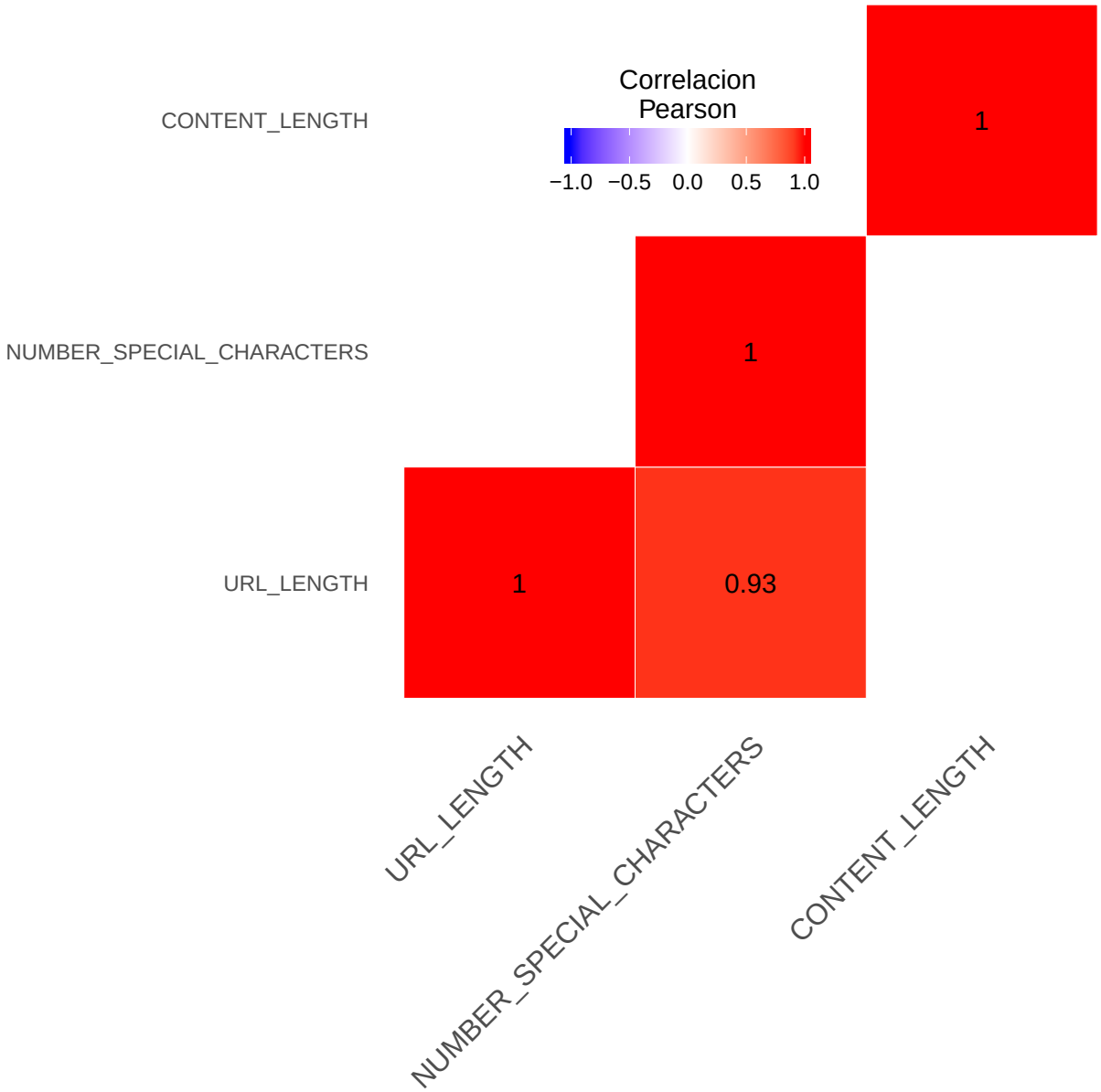


App Benigno

Se hace solo la correlación de datos numéricos

```
app_benigno.c <- round(cor(app_benigno[,c(2,3,7)]),2)

upper_tri <- get_upper_tri(app_benigno.c)

melted <- melt(upper_tri,na.rm=TRUE)
```

```r
ggheatmap.m <- ggplot(melted, aes(Var2, Var1, fill = value)) +
             geom_tile(color = "white") +
             scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                                  midpoint = 0, limit = c(-1,1), space = "Lab",
                                  name="Correlacion\nPearson") +
             theme_minimal()+ # minimal theme
             theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                              size = 12, hjust = 1))+
             coord_fixed()

ggheatmap.m <-  ggheatmap.m +
            geom_text(aes(Var2, Var1, label = value), color = "black", size = 4) +
            theme(axis.title.x = element_blank(),axis.title.y = element_blank(),
                  panel.grid.major = element_blank(),
                  panel.border = element_blank(),
                  panel.background = element_blank(),
                  axis.ticks = element_blank(),
                  legend.justification = c(1, 0),
                  legend.position = c(0.6, 0.7),
                  legend.direction = "horizontal") +
            guides(fill = guide_colorbar(barwidth = 7, barheight = 1,
                                         title.position = "top", title.hjust = 0.5))

ggheatmap.m
```

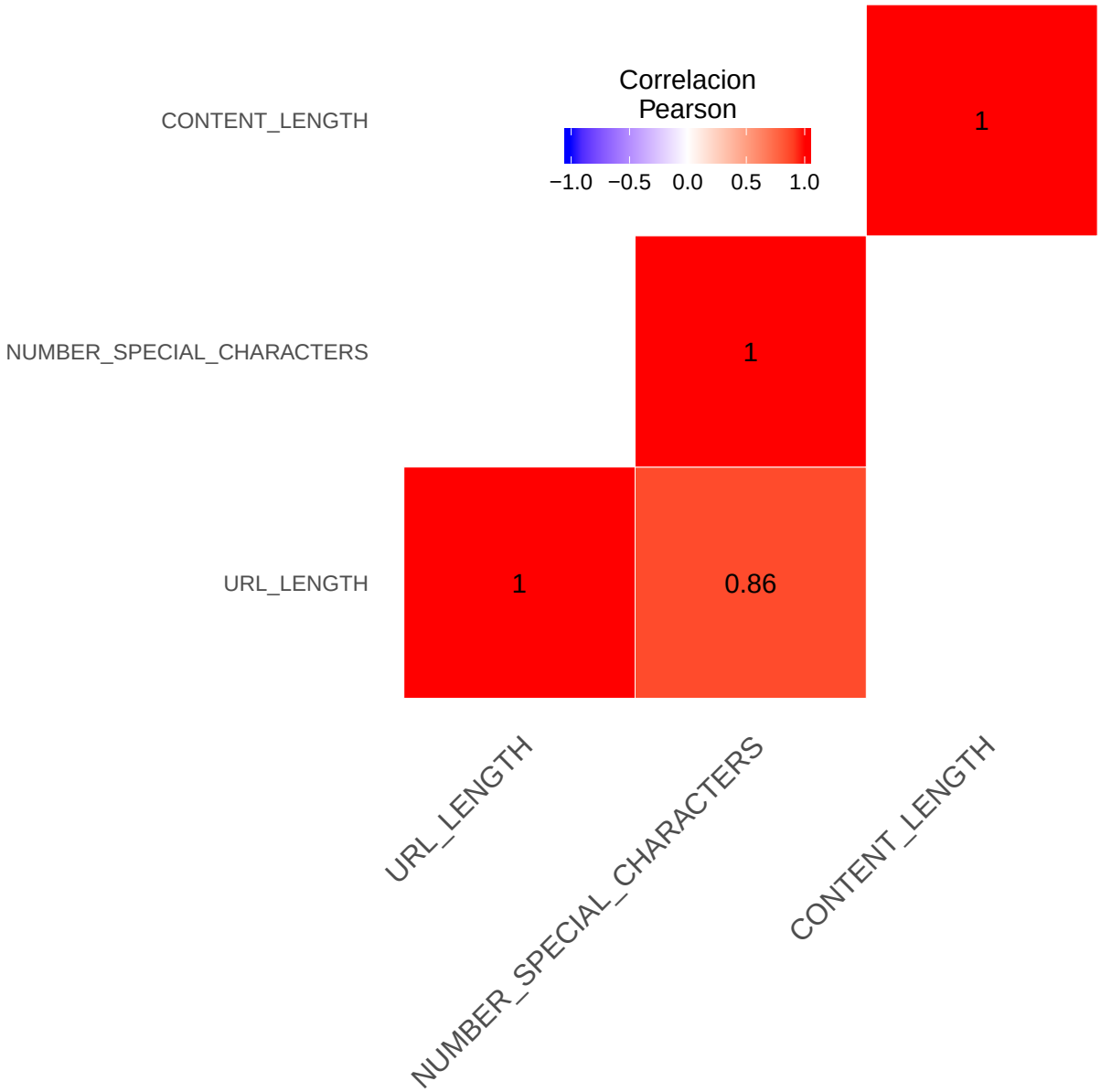


App Maligno

Se hace solo la correlación de datos numéricos

```
app_maligno.c <- round(cor(app_maligno[,c(2,3,7)]),2)

upper_tri <- get_upper_tri(app_maligno.c)

melted <- melt(upper_tri,na.rm=TRUE)
```

```r
ggheatmap.m <- ggplot(melted, aes(Var2, Var1, fill = value)) +
             geom_tile(color = "white") +
             scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                                  midpoint = 0, limit = c(-1,1), space = "Lab",
                                  name="Correlacion\nPearson") +
             theme_minimal()+ # minimal theme
             theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                              size = 12, hjust = 1))+
             coord_fixed()

ggheatmap.m <-  ggheatmap.m +
            geom_text(aes(Var2, Var1, label = value), color = "black", size = 4) +
            theme(axis.title.x = element_blank(),axis.title.y = element_blank(),
                  panel.grid.major = element_blank(),
                  panel.border = element_blank(),
                  panel.background = element_blank(),
                  axis.ticks = element_blank(),
                  legend.justification = c(1, 0),
                  legend.position = c(0.6, 0.7),
                  legend.direction = "horizontal") +
            guides(fill = guide_colorbar(barwidth = 7, barheight = 1,
                                         title.position = "top", title.hjust = 0.5))

ggheatmap.m
```



Union de matrices

Se procede a unir los resultados de la capa de red y de aplicación a su respectiva etiqueta de matrices benignas y malignas.

```
library(dplyr)
dataset_maligno <- merge(app_maligno,red_maligno, by=c("URL"))
dataset_maligno$TIPO <- 'Maligna'
```

```
dataset_benigno <- merge(app_benigno,red_benigno, by=c("URL"))
dataset_benigno$TIPO <- 'Benigna'

datos_completos <- bind_rows(dataset_maligno,dataset_benigno)

app_benigno$TIPO <- 'Benigno'
app_maligno$TIPO <- 'Maligno'

datos_app <- bind_rows(app_benigno,app_maligno)

red_benigno$TIPo <- 'Benigno'
red_maligno$TIPO <- 'Maligno'

datos_red <- bind_rows(red_benigno,red_maligno)
```

Refinamiento de variables

Se organizan algunos valores de las matrices tales como cache control y server. Estos valores se acomodan para que al binarizarlos su procedimiento sea mas adecuado.

```
library(reshape2)
library(splitstackshape)
library(dplyr)

cache_control <- cSplit(melt(datos_completos[,c(1,6)], id.vars ="URL"), "value", ",", "long")
library(data.table)
setDT(cache_control)[, c(levels(cache_control$value), "value") :=
            c(lapply(levels(value), function(x) as.integer(x == value)), .(NULL))]

cache_control <- cache_control[,-2]

cache_control <- cache_control[, lapply(.SD,sum), by=URL]

datos_completos <- merge(datos_completos,cache_control,by=c("URL"))
datos_completos <- datos_completos[,-6]

datos_completos$SERVER <- sapply(strsplit(datos_completos$SERVER,"/"), `[`, 1)

datos_completos <- data.frame(lapply(datos_completos, function(v) {
  if (is.character(v)) return(toupper(v))
  else return(v)
}))
datos_completos <- na.omit(datos_completos)


clasificacion <- datos_completos$TIPO
clasificacion <- data.frame(clasificacion)
datos_completos <- datos_completos[,-21]
```

Normalizacion de valores numericos

Una vez obtenida la matriz se normalizan todos los datos numéricos que se encuentran en el dataset.

```
datos_completos.n <- datos_completos

is.constant = function(x) all(x[1] == x)
constantes = sapply(datos_completos.n, is.constant)
datos_completos.n = datos_completos.n[,!constantes]

datos_completos.n[,c(2,3,6)] <- scale(datos_completos.n[,c(2,3,6)])
datos_completos.n[,c(9:19)] <- scale(datos_completos.n[,c(9:19)])
```

Binarizacion de variables categoricas

A continuación se binarizan las variables categóricas que se encuentran en la matriz.

```
library(caret)

dummifica2 = dummyVars( ~ ., data = datos_completos.n[2:ncol(datos_completos.n)])
datos_completos.n = predict(dummifica2, newdata = datos_completos.n)
URL <- datos_completos$URL

datos_completos.n <- data.frame(URL,datos_completos.n)
datos_completos.n <- data.frame(datos_completos.n,clasificacion)

write.csv(file='dataset_normalizado.csv', x=datos_completos.n)
```

Correlacion de datos completos

```
datos_completos.c <- round(cor(datos_completos.n[,2:(ncol(datos_completos.n)-1)]),2)
datos_completos.c <- data.frame(datos_completos.c)
write.csv(file='correlacion_datos.csv', x=datos_completos.c)
```

Aprendizaje

Aquí se muestran los algoritmos de clasificación, cual es su grado de exactitud y una matriz de confusión que muestra que tan efectivos fueron cada uno de los algoritmos.

SVM

```
ptm <- proc.time()

set.seed(1)
entrenamiento <- datos_completos.n[,2:ncol(datos_completos.n)]

train_i <- createFolds(entrenamiento$clasificacion, k=10)
svmFit <- train(clasificacion ~., method = "svmLinear", data = entrenamiento, tuneLength = 10,trControl
svmFit
```

```
## Support Vector Machines with Linear Kernel
##
```

```
## 967 samples
## 398 predictors
##   2 classes: 'BENIGNA', 'MALIGNA'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 870, 870, 871, 871, 870, 870, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9741832  0.8522505
##
## Tuning parameter 'C' was held constant at a value of 1
```

```
confusionMatrix(svmFit)
```

```
## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction BENIGNA MALIGNA
##    BENIGNA    88.7     2.3
##    MALIGNA     0.3     8.7
##
##  Accuracy (average) : 0.9741
```

```
#Tiempo ejecucion en segundos
proc.time() - ptm
```

```
##    user  system elapsed
##    4.84    0.11    5.19
```

Arbol de decision (J48)

```
ptm <- proc.time()

library(caTools)

library(RWeka)

trainJ <- createFolds(entrenamiento$clasificacion, k=10)

C45Fit <- train(clasificacion ~ ., method = "J48", data = entrenamiento,
                tuneLength = 5,
                trControl = trainControl(
                  method = "cv", indexOut = trainJ))
C45Fit
```

```
## C4.5-like Trees
##
## 967 samples
## 398 predictors
##   2 classes: 'BENIGNA', 'MALIGNA'
##
```

```
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 869, 870, 870, 871, 870, 871, ...
## Resampling results across tuning parameters:
##
##   C       M  Accuracy   Kappa
##   0.0100  1  0.9555412  0.7283572
##   0.0100  2  0.9544888  0.7175054
##   0.0100  3  0.9503651  0.6831262
##   0.0100  4  0.9451997  0.6500549
##   0.0100  5  0.9421070  0.6259616
##   0.1325  1  0.9741731  0.8524303
##   0.1325  2  0.9689863  0.8207138
##   0.1325  3  0.9658935  0.8028961
##   0.1325  4  0.9586555  0.7575108
##   0.1325  5  0.9565936  0.7523139
##   0.2550  1  0.9813896  0.8998459
##   0.2550  2  0.9679553  0.8203471
##   0.2550  3  0.9648625  0.8033050
##   0.2550  4  0.9565936  0.7489177
##   0.2550  5  0.9565936  0.7544504
##   0.3775  1  0.9855133  0.9216595
##   0.3775  2  0.9731100  0.8526901
##   0.3775  3  0.9720898  0.8468645
##   0.3775  4  0.9617590  0.7835388
##   0.3775  5  0.9607281  0.7826680
##   0.5000  1  0.9875859  0.9338375
##   0.5000  2  0.9803479  0.8956698
##   0.5000  3  0.9741516  0.8586168
##   0.5000  4  0.9628007  0.7975175
##   0.5000  5  0.9628007  0.7976901
##
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were C = 0.5 and M = 1.
```

```
confusionMatrix(C45Fit)
```

```
## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction BENIGNA MALIGNA
##    BENIGNA    88.5     0.7
##    MALIGNA     0.5    10.2
##
##  Accuracy (average) : 0.9876
```

```
#Tiempo ejecucion en segundos
proc.time() - ptm
```

```
##    user  system elapsed
##  221.42    2.20  224.47
```

Regresión Logistica

```
ptm <- proc.time()

#REGGRESION LOGISTICA
train_control<- trainControl(method = "cv", number = 10)
modl<- train(clasificacion~.,data=entrenamiento, trControl= train_control, method="glm", family= binomia
modl
```

```
## Generalized Linear Model
##
## 967 samples
## 398 predictors
##   2 classes: 'BENIGNA', 'MALIGNA'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 870, 870, 871, 870, 870, 870, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9058419  0.5704692
```

```
confusionMatrix(modl)
```

```
## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction BENIGNA MALIGNA
##    BENIGNA    82.7     3.1
##    MALIGNA     6.3     7.9
##
##  Accuracy (average) : 0.9059
```

```
#Tiempo ejecucion en segundos
proc.time() - ptm
```

```
##    user  system elapsed
##   45.31    0.50   47.00
```

```
ptm <- proc.time()

#NAIVE BAYES
library(e1071)
tune.control <- tune.control(random=F,nrepeat=5,
                             sampling=c("cross"),sampling.aggregate=mean, cross=10,
                             best.model=T, performances=T)

model <- naiveBayes(clasificacion ~., entrenamiento, tune.control)
pred <- predict(model, entrenamiento)

confusionMatrix(pred, entrenamiento$clasificacion)
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction BENIGNA MALIGNA
##    BENIGNA       0       0
##    MALIGNA     861     106
##
##                Accuracy : 0.1096
##                  95% CI : (0.0906, 0.131)
##     No Information Rate : 0.8904
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.0000
##             Specificity : 1.0000
##          Pos Pred Value :    NaN
##          Neg Pred Value : 0.1096
##              Prevalence : 0.8904
##          Detection Rate : 0.0000
##    Detection Prevalence : 0.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : BENIGNA
##
```

```r
#Tiempo ejecucion en segundos
proc.time() - ptm
```

```
##    user  system elapsed
##   10.07    0.00   11.04
```