Web page Phishing Detection

Thomas Fernandes, Yassine Ouerghi, Vanessa Kenniche, Mario Miron Ramos

2023-11-09

L'étude vise à prédire la légitimité des sites web en utilisant diverses techniques de machine learning. Le phénomène du phishing consiste en des tentatives de fraude en ligne par le biais de sites web frauduleux imitant des sites légitimes.

La variable que nous cherchons à prédire est "status", qui indique si un site web est légitime ou potentiellement frauduleux (phishing). Pour ce faire, nous disposons d'un ensemble de données équilibré de 87 variables explicatives différentes, chacune fournissant des informations sur divers aspects de 11430 sites web différents. Ces données incluent 56 variables basées sur la structure, 24 extraites du contenu des pages web correspondantes, 7 obtenues par des requêtes auprès de services externes.

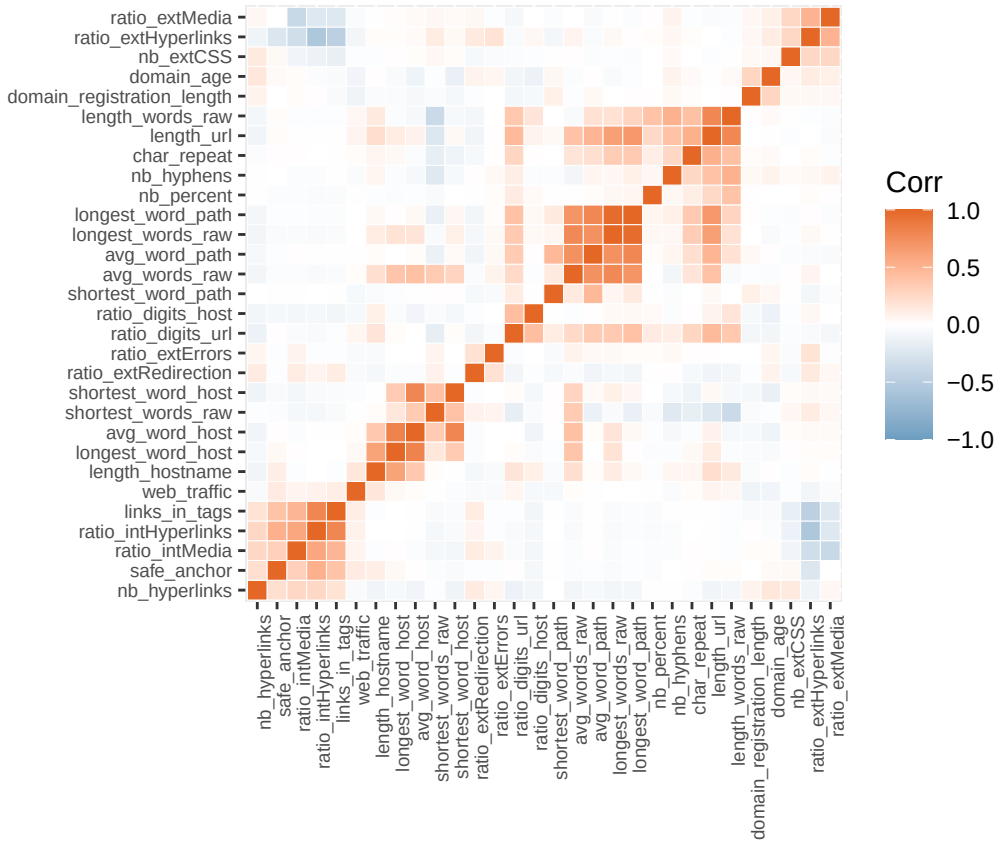
1. Présentation des données

Avant de commencer les différentes modélisations, nous allons regarder comment se structurent nos données.

1.1. Corrélation entre les variables quantitatives

```
df_present <- df

#Extraire les variables qualitatives
v_quali <- vector("logical", length = ncol(df_present) - 1)
for (i in 2:ncol(df_present)) {
  v_quali[[i]] <- (length(unique(df_present[[i]])) / sum(!is.na(df_present[[i]]))) < 0.002
}

num_cols <- character()
cat_cols <- character()

for (i in 1:length(v_quali)) {
  if (!v_quali[[i]]) {
    num_cols <- c(num_cols, names(df_present)[i])
  } else {
    cat_cols <- c(cat_cols, names(df_present)[i])
  }
}

corr <- cor(df_present[num_cols])

ggcorrplot(
  corr,
  hc.order = TRUE,
  type = "full",
  outline.color = "white",
```

```
  ggtheme = ggplot2::theme_gray,
  colors = c("#6D9EC1", "white", "#E46726"),
  show.diag = TRUE,
  tl.cex = 7,
  tl.srt = 90
)
```

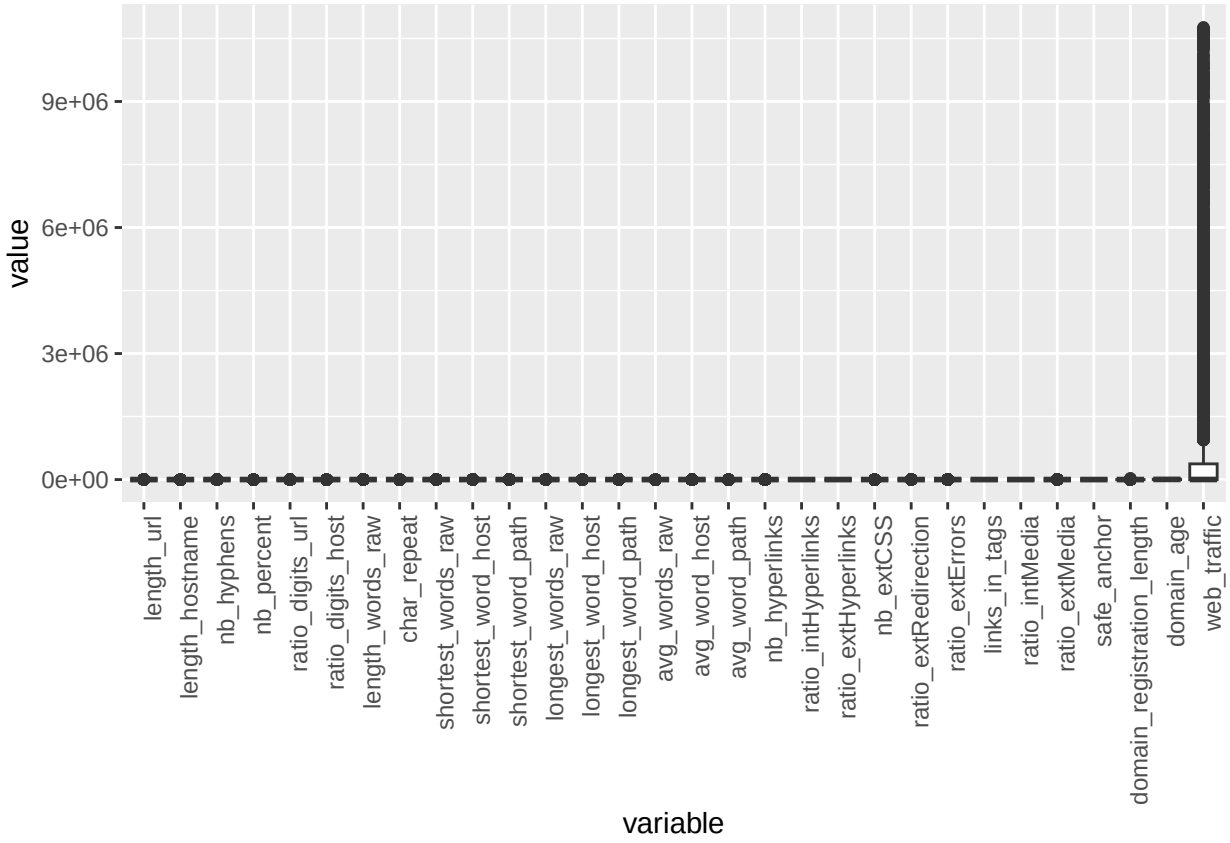


Comme on s'y attendait, on remarque que de nombreuses variables sont corrélées entre elles. C'est le cas par exemple de longest_word_path et de avg_word_path.

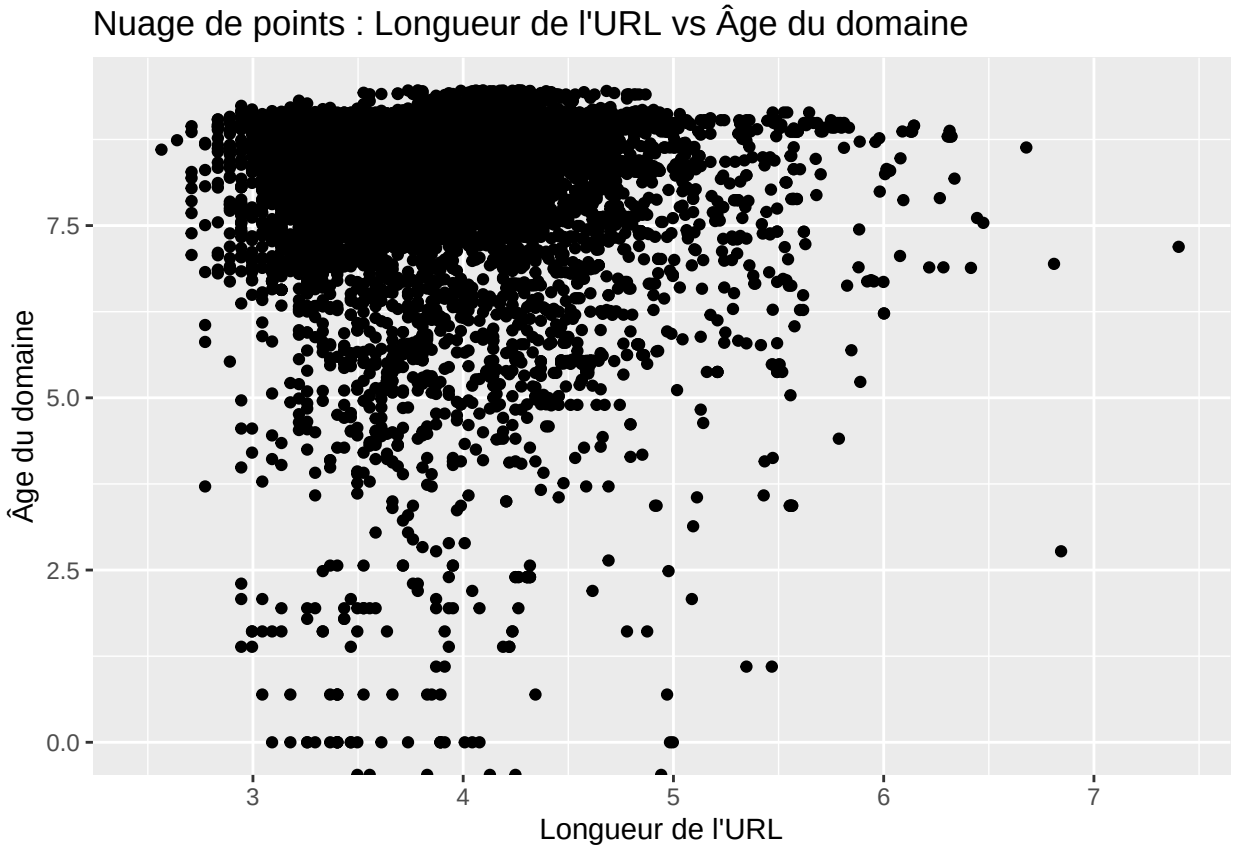
On fait un boxplot de toutes les variables

```
ggplot(data = melt(df_present[, num_cols]), aes(x = variable, y = value)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

```
## No id variables; using all as measure variables
```




```r
attach(df)

ggplot(df, aes(x = log(length_url), y = log(domain_age))) +
  geom_point() +
  labs(x = "Longueur de l'URL", y = "Âge du domaine") +
  ggtitle("Nuage de points : Longueur de l'URL vs Âge du domaine")
```



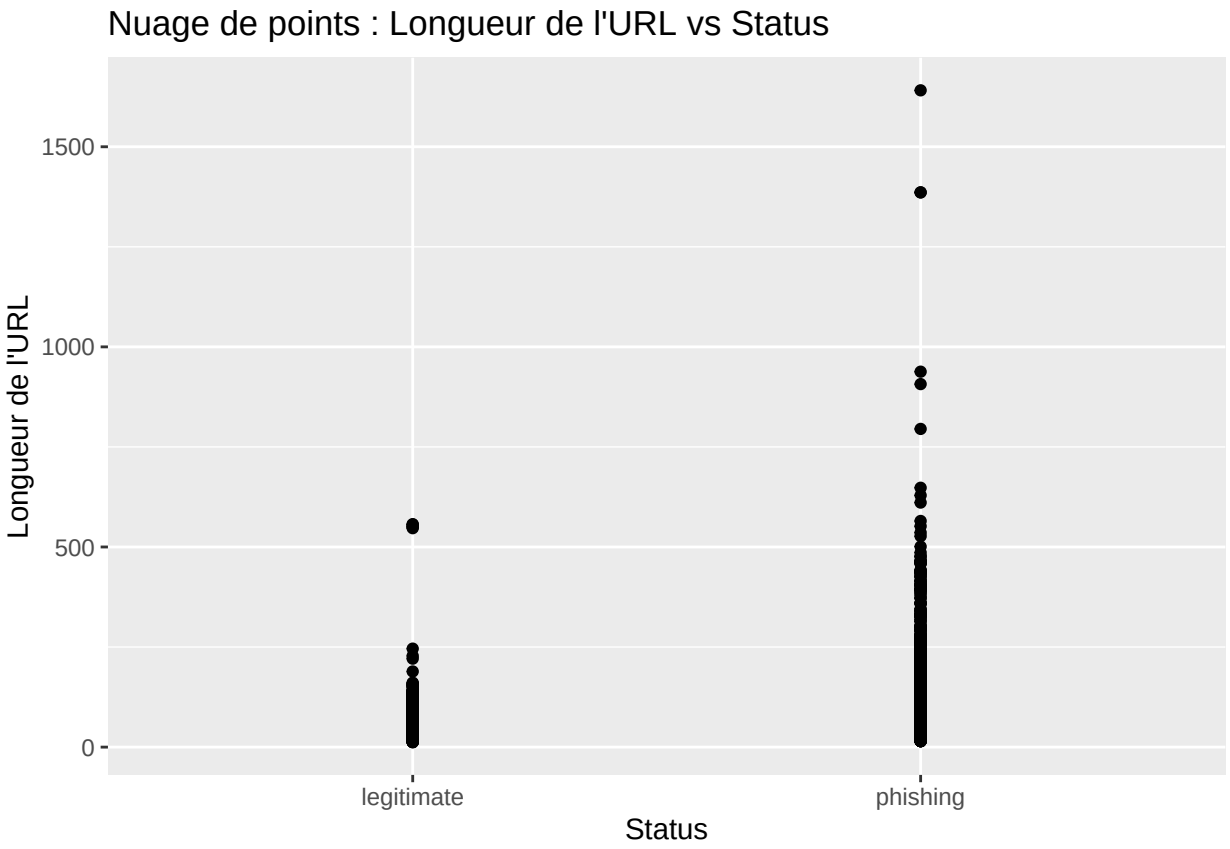

```r
max(domain_age)
```

```
## [1] 12874
```

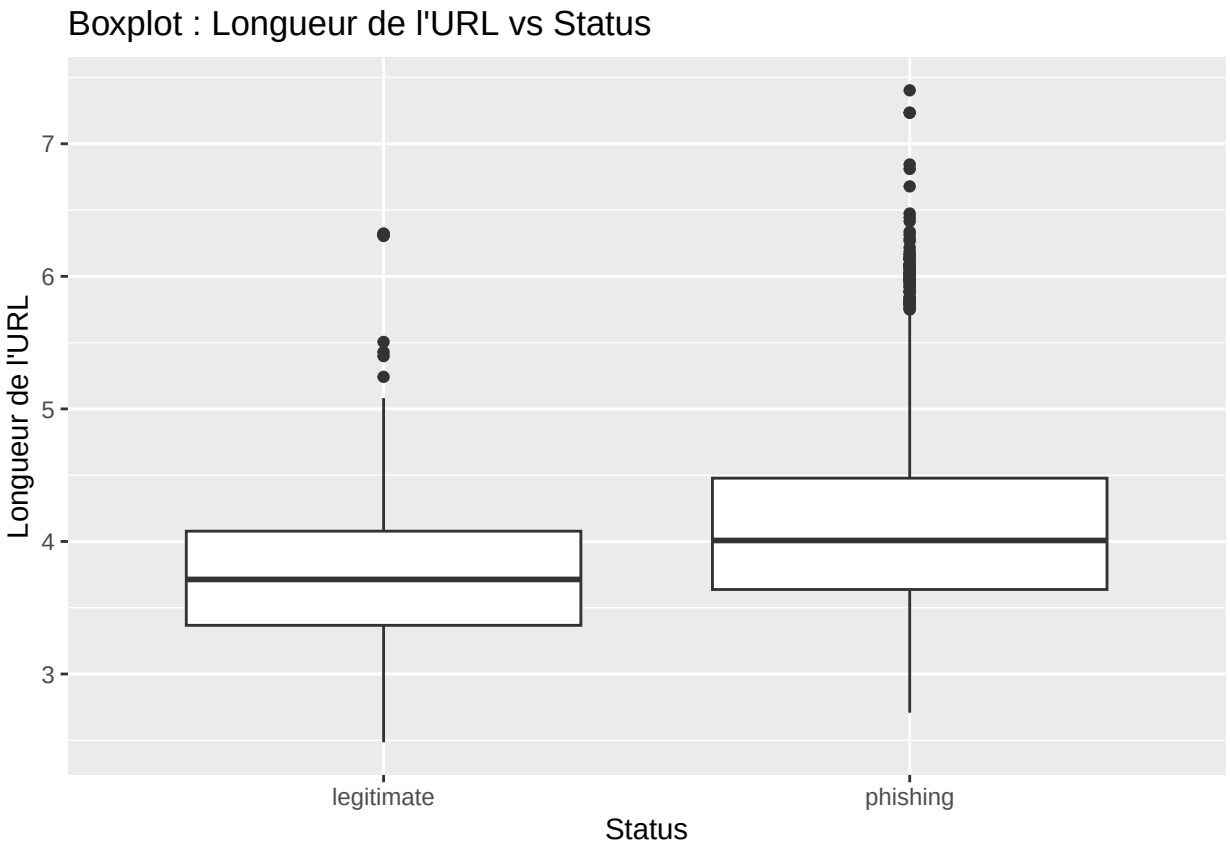
```r
min(domain_age)
```

```
## [1] -12
```

```r
# Nuage de point y = longueur url, x = status
ggplot(df, aes(x = status, y = length_url)) +
  geom_point() +
  labs(x = "Status", y = "Longueur de l'URL") +
  ggtitle("Nuage de points : Longueur de l'URL vs Status")
```

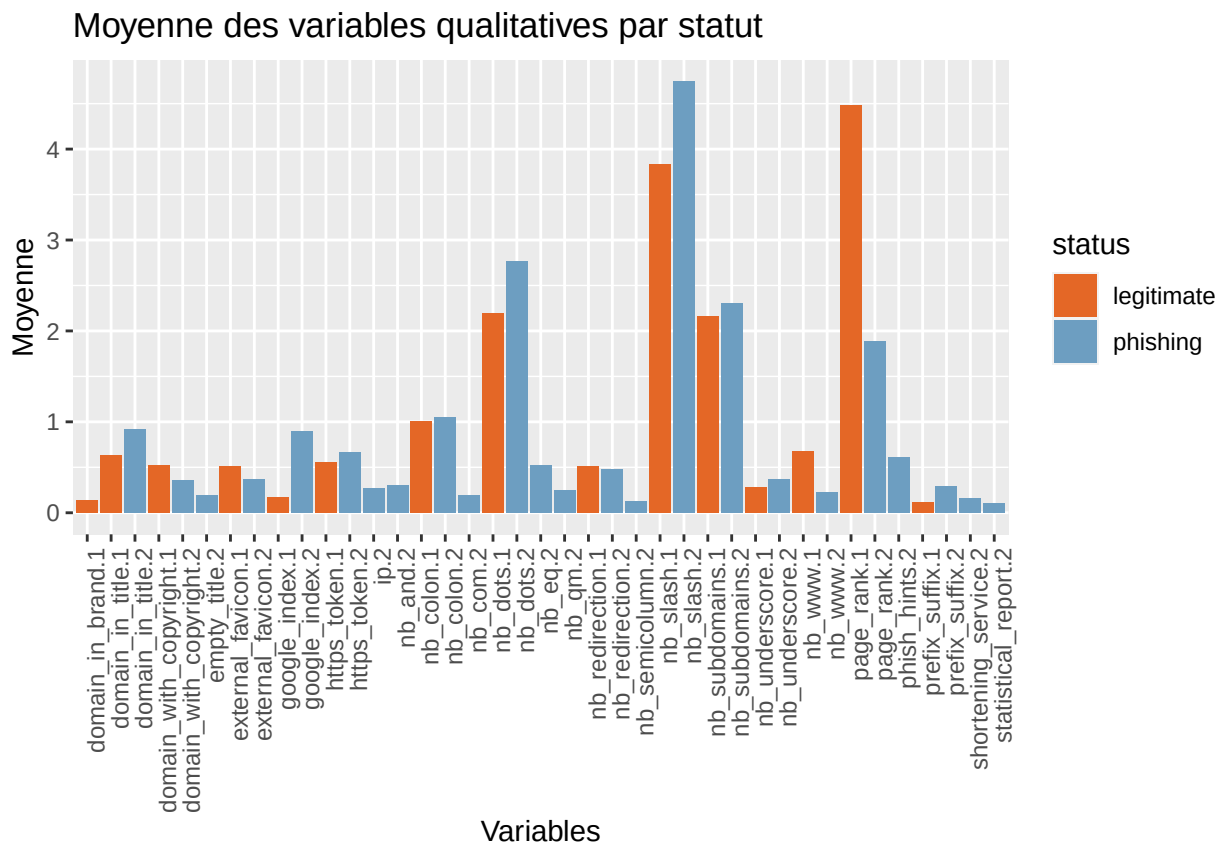



```
# Boxplot
ggplot(df, aes(x = status, y = log(length_url))) +
  geom_boxplot() +
  labs(x = "Status", y = "Longueur de l'URL") +
  ggtitle("Boxplot : Longueur de l'URL vs Status")
```



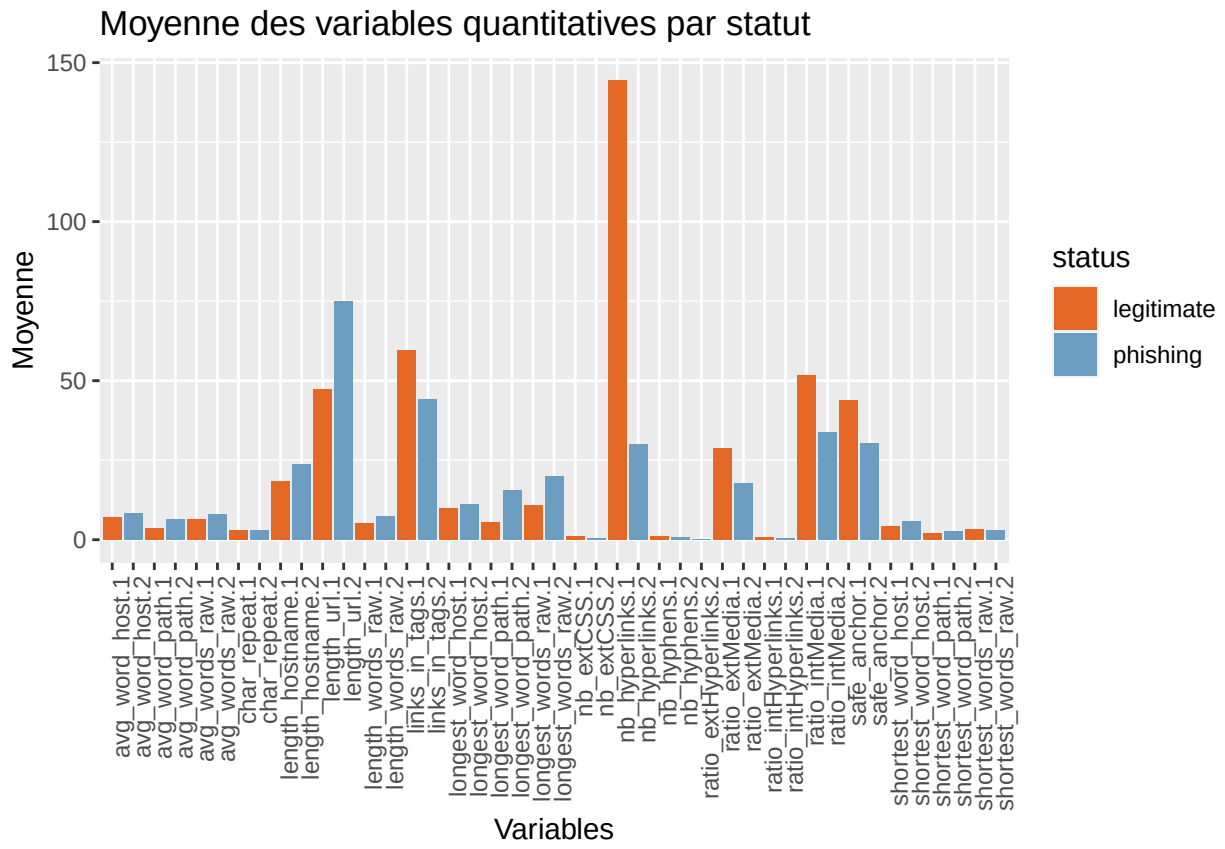
1.2. Moyenne par statut

```r
mean_by_status <- function(df, col_name) {
  df %>%
    group_by(status) %>%
    summarise(mean_value = mean(.data[[col_name]], na.rm = TRUE))
}
mean_values_list_cat <- list()

for (col in cat_cols) {
  mean_values_list_cat[[col]] <- mean_by_status(df_present, col)
}

mean_values_df_cat <- do.call(rbind, mean_values_list_cat)
mean_values_df_cat$col_names <- rownames(mean_values_df_cat)

mean_values_df_cat <- mean_values_df_cat[mean_values_df_cat$mean_value > 0.1 | mean_values_df_cat$mean_v
mean_values_df_cat <- mean_values_df_cat[!is.na(mean_values_df_cat$mean_value), ]

ggplot(mean_values_df_cat, aes(x = col_names, y = mean_value, fill = status)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Variables", y = "Moyenne", title = "Moyenne des variables qualitatives par statut") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  scale_fill_manual(values = c("#E46726", "#6D9EC1"))
```



Le rang de la page semble être la variable qualitative qui influe le plus. C'est la variable pour laquelle on voit la plus grande différence entre (en % de l'autre) la moyenne du groupe 1 et celle du 2.

```
mean_values_list_num <- list()

for (col in num_cols) {
  if (col != "web_traffic" && col != "domain_age" && col != "domain_registration_length") {
    mean_values_list_num[[col]] <- mean_by_status(df_present, col)
  }
}

mean_values_df_num <- do.call(rbind, mean_values_list_num)
mean_values_df_num$col_names <- rownames(mean_values_df_num)

mean_values_df_num <- mean_values_df_num[mean_values_df_num$mean_value > 0.3 | mean_values_df_num$mean_v
mean_values_df_num <- mean_values_df_num[!is.na(mean_values_df_num$mean_value), ]

ggplot(mean_values_df_num, aes(x = col_names, y = mean_value, fill = status)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Variables", y = "Moyenne", title = "Moyenne des variables quantitatives par statut") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  scale_fill_manual(values = c("#E46726", "#6D9EC1"))
```



Le nombre d'hyperlien semble être la variable quantitative qui influe le plus.

2. Modélisation

2.1. Préparation du jeu de données

On sépare le jeu de données en un échantillon d'entraînement et un échantillon test, qui seront les mêmes pour tous les modèles.

```
set.seed(123)
indxTrain <- createDataPartition(df$status, p = 0.75, list = FALSE)
DTrain <- df[indxTrain, ]
DTest <- df[-indxTrain, ]
```

2.2. Variable de controle

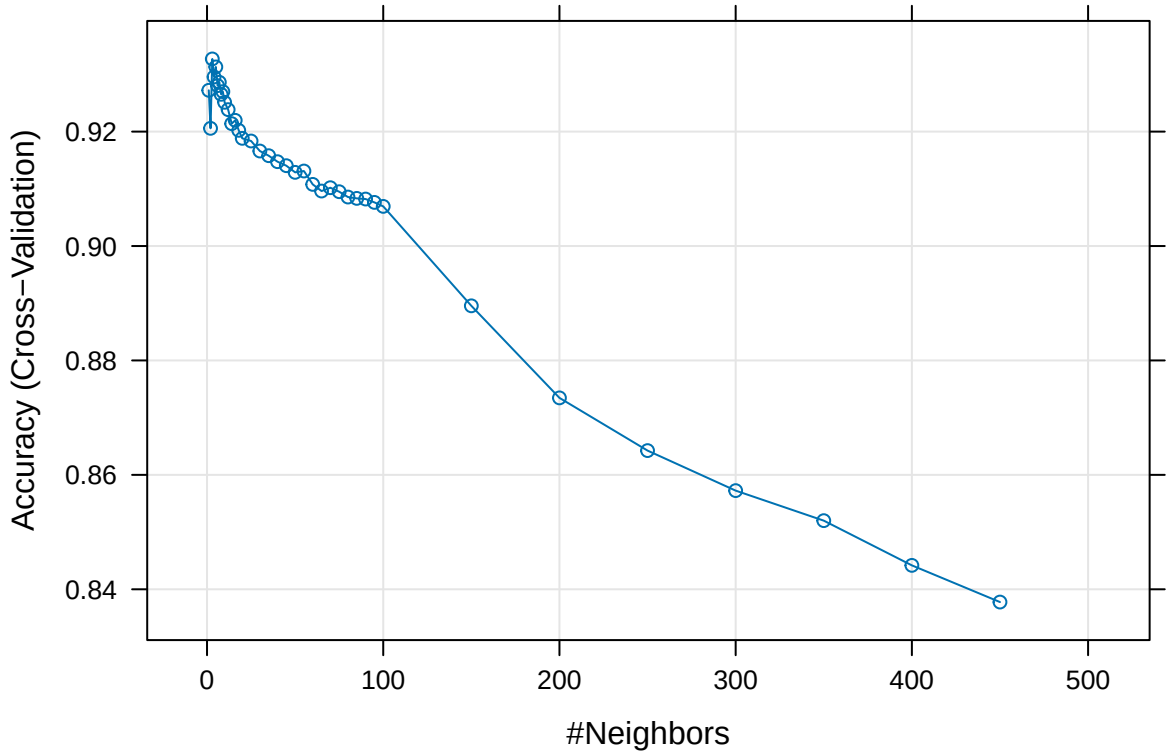
```
ctrl <- trainControl(method = "cv", number = 5)
ctrl_temp <- trainControl(method = "none")
```

2.3. Entrainement

KNN

```
set.seed(123)

k <- c(1:10, seq(12, 19, by = 2), seq(20, 99, by = 5), seq(100, 500, by = 50))
```

```r
#fit.knn.cv <- train(status ~ .,data = DTrain,method = "knn",trControl = ctrl,tuneGrid = expand.grid(k =
load("C:/Users/thoma/Desktop/Github/Web-page-Phishing-Detection/fit.knn.cv.RDATA")
plot(fit.knn.cv)
```




```r
bestK <- fit.knn.cv$bestTune$k
print(fit.knn.cv$results)
```

```
##       k  Accuracy     Kappa  AccuracySD     KappaSD
## 1     1 0.9272210 0.8544423 0.007037996 0.014075344
## 2     2 0.9205740 0.8411486 0.008217546 0.016433645
## 3     3 0.9327036 0.8654076 0.004973536 0.009946071
## 4     4 0.9295545 0.8591097 0.005333699 0.010665375
## 5     5 0.9313040 0.8626088 0.004796734 0.009591570
## 6     6 0.9280380 0.8560767 0.006182806 0.012363033
## 7     7 0.9286210 0.8572428 0.007154438 0.014305675
## 8     8 0.9265216 0.8530439 0.007300813 0.014598271
## 9     9 0.9269884 0.8539780 0.008982349 0.017960843
## 10   10 0.9251224 0.8502460 0.007676528 0.015348829
## 11   12 0.9238392 0.8476795 0.010073711 0.020143809
## 12   14 0.9213899 0.8427811 0.009162384 0.018321369
## 13   16 0.9219728 0.8439468 0.007963850 0.015925124
## 14   18 0.9202238 0.8404491 0.010435372 0.020866929
## 15   20 0.9188240 0.8376495 0.010131404 0.020258800
## 16   25 0.9183577 0.8367167 0.009644537 0.019285377
## 17   30 0.9166082 0.8332180 0.011209691 0.022414308
## 18   35 0.9157916 0.8315849 0.012512710 0.025019445
```

```
## 19  40 0.9147417 0.8294850 0.011231901 0.022457048
## 20  45 0.9140421 0.8280858 0.010916497 0.021825599
## 21  50 0.9128758 0.8257537 0.012728320 0.025447923
## 22  55 0.9131090 0.8262197 0.012234800 0.024461115
## 23  60 0.9107765 0.8215550 0.013125167 0.026241011
## 24  65 0.9096102 0.8192225 0.012240555 0.024471099
## 25  70 0.9101931 0.8203884 0.013523170 0.027037095
## 26  75 0.9094933 0.8189888 0.013058786 0.026108490
## 27  80 0.9085603 0.8171229 0.013245224 0.026481092
## 28  85 0.9083272 0.8166566 0.012243907 0.024478376
## 29  90 0.9082107 0.8164236 0.012912827 0.025816262
## 30  95 0.9076277 0.8152577 0.013075138 0.026140291
## 31 100 0.9069272 0.8138569 0.014586792 0.029164279
## 32 150 0.8895491 0.7791003 0.013204905 0.026396546
## 33 200 0.8734542 0.7469110 0.012905856 0.025795505
## 34 250 0.8642404 0.7284837 0.012999048 0.025980148
## 35 300 0.8572424 0.7144875 0.014207057 0.028396630
## 36 350 0.8519937 0.7039902 0.013480493 0.026939898
## 37 400 0.8441798 0.6883621 0.012147403 0.024274500
## 38 450 0.8377655 0.6755344 0.014413400 0.028798348
## 39 500       NaN       NaN          NA          NA
```

Regression logistique

```r
#Sans AIC
fit.lr <- train(status ~ .,
                data = DTrain,
                method = "glm",
                trControl = ctrl,
                na.action = na.omit)
```

```
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
```

```r
print(varImp(fit.lr))
```

```
## glm variable importance
##
##   only 20 most important variables shown (out of 81)
##
##                             Overall
## google_index                 100.00
## page_rank                     81.32
## nb_www                        56.29
## phish_hints                   52.56
## domain_age                    34.17
## nb_hyperlinks                 32.68
## shortening_service            32.05
## longest_words_raw             29.62
## ratio_digits_host             26.20
## length_hostname               25.31
## nb_hyphens                    24.42
## domain_in_title               23.75
## nb_underscore                 23.70
## https_token                   22.65
## domain_registration_length    22.34
## ratio_extHyperlinks           21.37
## ratio_extRedirection          20.95
## ratio_extMedia                19.75
## avg_words_raw                 19.56
## nb_space                      19.21
```

```r
#Avec AIC
#fit.lr.aic <- train(status ~ ., data = DTrain, method = "glmStepAIC", trControl = ctrl, na.action = na
load("C:/Users/thoma/Desktop/Github/Web-page-Phishing-Detection/fit.lr.aic.RDATA")
```

NB

```r
fit.nb = train(status ~ ., data = DTrain, method = "nb", trControl = ctrl)
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 29

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 74

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 211

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 280

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 304

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 331

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

```
## observation 372
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 374
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 435
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 524
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 646
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 680
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 692
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 774
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 843
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 891
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1048
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1069
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1108
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1143
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1171
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1202
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1231
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1243
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1284
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1401
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1421
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1452
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1469

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1472

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1540

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1565

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1632

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1691

## Warning: model fit failed for Fold1: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y
##   Zero variances for at least one class in variables: nb_at, nb_or, nb_star, nb_dollar, punycode, pa

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 31

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 40

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 226

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 246

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 289

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 325

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 346

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 366

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 397

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 443

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 470

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 480

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 482

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 534
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 560

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 641

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 737

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 786

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 950

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 959

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 961

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1058

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1134

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1149

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1209

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1228

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1265

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1267

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1309

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1322

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1437

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1450

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1453

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1455

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1485
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1508

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1603

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1715

## Warning: model fit failed for Fold2: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y
##   Zero variances for at least one class in variables: nb_at, nb_or, nb_star, nb_dollar, punycode, pa

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 63

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 150

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 151

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 160

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 173

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 293

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 392

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 435

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 507

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 652

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 773

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 779

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 819

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 900

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1014

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1073

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1103
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1123

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1149

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1311

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1338

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1351

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1425

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1461

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1493

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1587

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1594

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1671

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1696

## Warning: model fit failed for Fold3: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y
##   Zero variances for at least one class in variables: nb_at, nb_or, nb_star, nb_dollar, punycode, pa

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 66

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 153

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 154

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 192

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 397

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 415

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 430
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 485

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 639

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 722

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 734

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 764

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 843

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 896

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 963

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1085

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1101

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1167

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1279

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1330

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1337

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1372

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1373

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1386

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1418

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1468

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1551

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1571
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1661

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1704

## Warning: model fit failed for Fold4: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y
##   Zero variances for at least one class in variables: nb_at, nb_or, nb_star, nb_dollar, punycode, pa

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 24

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 48

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 105

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 138

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 182

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 186

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 336

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 379

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 426

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 483

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 484

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 499

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 529

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 551

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 739

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 826

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 963

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1039
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1057

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1234

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1454

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1510

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1513

## Warning: model fit failed for Fold5: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBayes.default(x, y
##   Zero variances for at least one class in variables: nb_at, nb_or, nb_star, nb_dollar, punycode, pa

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.

## Warning in train.default(x, y, weights = w, ...): missing values found in
## aggregated results
```

LDA

```
fit.lda = train(status ~ .,
                data = DTrain[, -c(9, 60, 62, 64, 69, 72)],
                method="lda",
                trControl=ctrl)
```

```
## Warning: model fit failed for Fold1: parameter=none Error in lda.default(x, grouping, ...) :
##   la variable 36 semble être constante à l'intérieur des groupes

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.
```

QDA

```
fit.qda = train(status ~ .,
                data = DTrain[, c(88, 86, 87, 21, 51, 83, 57, 36, 45, 27, 2, 5, 79, 11, 25, 82, 59, 63,
                method="qda",
                trControl=ctrl)
```

SVM - Séparateur Linéaire

```
#svmGrid_lin = seq(0.0001, 0.01 ,by = 0.001)
#fit.Lin.svm = train(status ~ ., data = DTrain, method = "svmLinear", type = "C-svc", trControl = ctrl,
#load("C:/Users/thoma/Desktop/Github/Web-page-Phishing-Detection/fit.Lin.svm.RDATA")
#plot(fit.Lin.svm)
```

SVM - Séparateur Quadratique

```
#Noyau Radial
#svmGrid_quad = seq(0.0001, 0.01 ,by = 0.001)
#fit.Quad.svm = train(status ~ ., data = DTrain, method = "svmRadial", type = "C-svc", trControl = ctrl
#load("C:/Users/thoma/Desktop/Github/Web-page-Phishing-Detection/fit.Quad.svm.RDATA")
```

```
#plot(fit.Quad.svm)
#fit.Quad.svm$bestTune

#Noyau polynomial
#svmGrid_poly = seq(0.0001, 0.01 ,by = 0.001)
#fit.Poly.svm = train(status ~ ., data = DTrain, method = "svmPoly", type = "C-svc", trControl = ctrl_t
#plot(fit.Poly.svm)
#fit.Poly.svm$bestTune
```

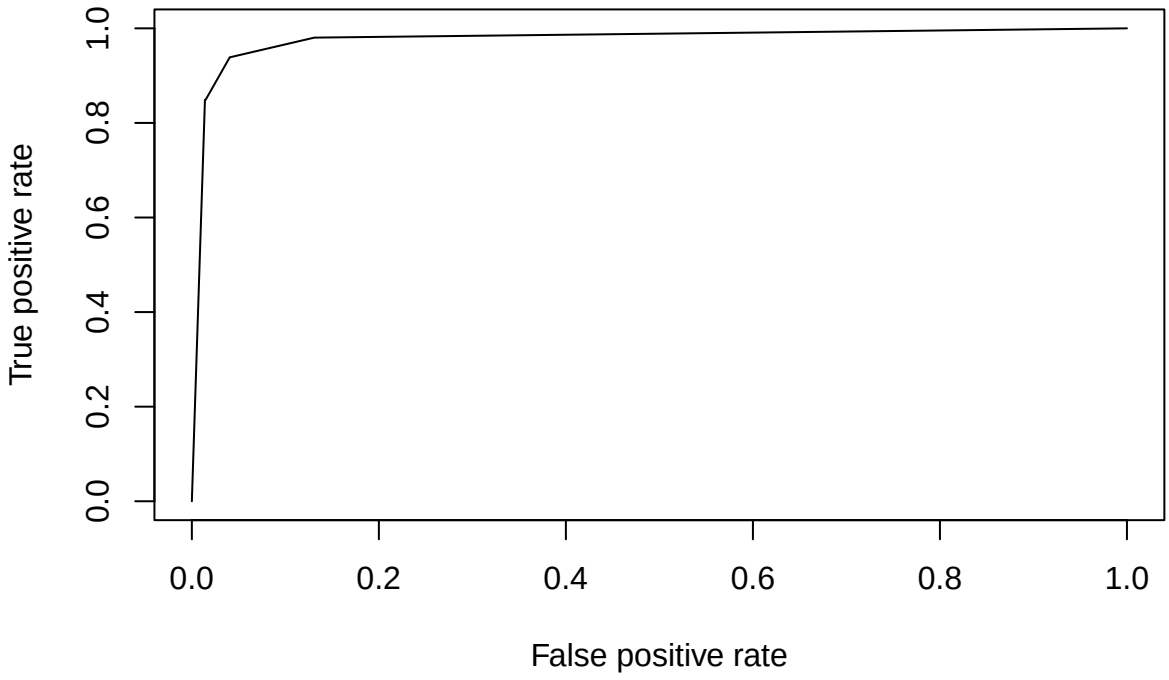
Prédictions

KNN

```
set.seed(123)
predictionsBestK <- predict(fit.knn.cv, newdata = DTest)
confusionMatrixBestK <- confusionMatrix(predictionsBestK, DTest$status)
print(confusionMatrixBestK$overall['Accuracy'])
```

```
##  Accuracy
## 0.9488796
```

```
set.seed(123)
predictionsBestK <- predict(fit.knn.cv, newdata = DTest, type = "prob")

pred.knn <- prediction(predictionsBestK[,2], DTest$status)
perf.knn <- performance(pred.knn, "tpr", "fpr")
plot(perf.knn)
```




```
auc.knn <- performance(pred.knn, "auc")@y.values[[1]]
```

Regression logistique

```
class.lr <- predict(fit.lr, newdata = DTest)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

```
print(varImp(fit.lr))
```

```
## glm variable importance
##
##   only 20 most important variables shown (out of 81)
##
##                            Overall
## google_index                100.00
## page_rank                    81.32
## nb_www                       56.29
## phish_hints                  52.56
## domain_age                   34.17
## nb_hyperlinks                32.68
## shortening_service           32.05
## longest_words_raw            29.62
## ratio_digits_host            26.20
## length_hostname              25.31
## nb_hyphens                   24.42
```

```
## domain_in_title              23.75
## nb_underscore                23.70
## https_token                  22.65
## domain_registration_length   22.34
## ratio_extHyperlinks          21.37
## ratio_extRedirection         20.95
## ratio_extMedia               19.75
## avg_words_raw                19.56
## nb_space                     19.21
```

```
class.lr.aic <- predict(fit.lr.aic, newdata = DTest)

confusionMatrixLR <- confusionMatrix(class.lr, DTest$status)
confusionMatrixLRAIC <- confusionMatrix(class.lr.aic, DTest$status)
print(confusionMatrixLR$overall['Accuracy'])
```

```
##  Accuracy
## 0.9464286
```

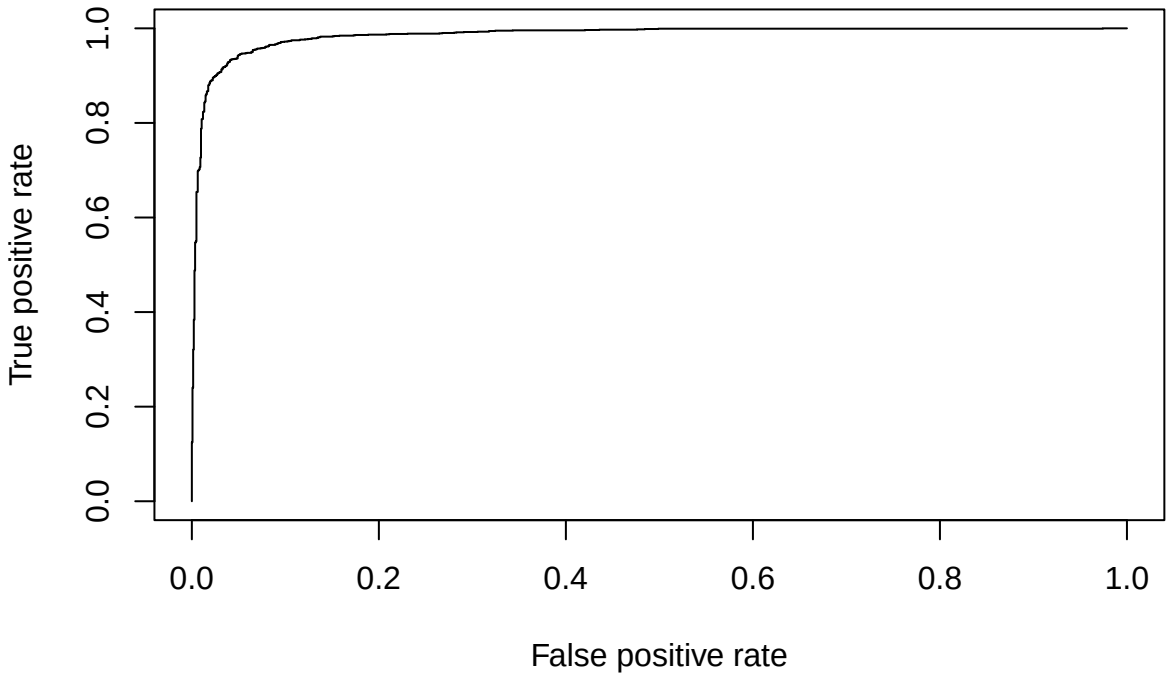
```
print(confusionMatrixLRAIC$overall['Accuracy'])
```

```
##  Accuracy
## 0.9457283
```

```
class.lr <- predict(fit.lr, newdata = DTest, type = "prob")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

```
class.lr.aic <- predict(fit.lr.aic, newdata = DTest, type = "prob")

pred.lr <- prediction(class.lr[,2], DTest$status)
perf.lr <- performance(pred.lr, "tpr", "fpr")
plot(perf.lr)
```




```
auc.lr <- performance(pred.lr, "auc")@y.values[[1]]
```

NB

```
#Accuracy
class.nb <- predict(fit.nb, newdata = DTest)
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 30
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 65
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 156
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 181
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 208
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 728
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 864
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 975
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1153

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1163

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1211

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1220

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1254

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1271

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1365

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1409

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1449

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1682

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1851

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1877

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1938

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1998

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2017

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2037

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2075

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2078

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2133

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2212

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2275
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2338

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2424

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2438

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2462

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2473

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2496

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2756

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2796
```

```
confusionMatrixNB <- confusionMatrix(class.nb, DTest$status)
print(confusionMatrixNB$overall['Accuracy'])
```

```
##  Accuracy
## 0.7405462
```

```
pred.nb = predict(fit.nb, newdata=DTest[,-88], type = "prob")
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 30

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 65

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 156

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 181

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 208

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 728

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 864

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 975

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1153

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1163

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1211
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1220

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1254

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1271

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1365

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1409

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1449

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1682

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1851

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1877

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1938

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1998

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2017

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2037

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2075

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2078

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2133

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2212

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2275

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2338

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2424

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2438
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2462
```

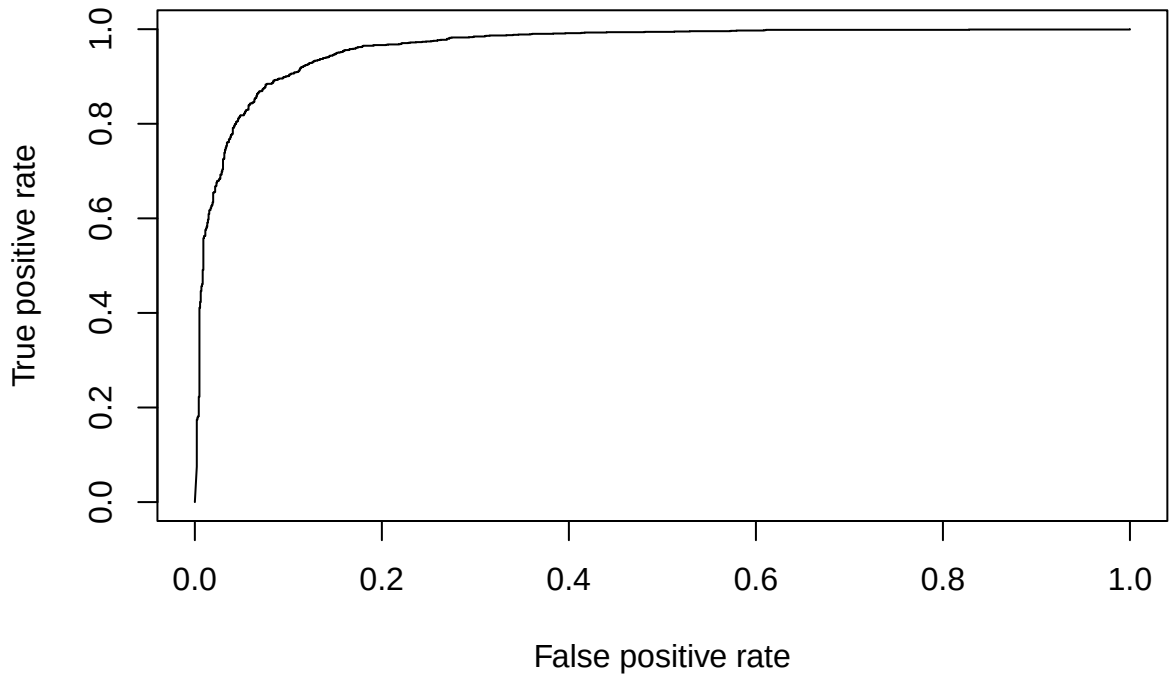
```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2473
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2496
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2756
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2796
```

```
pred.nb <- prediction(pred.nb[,2], DTest$status)
perf.nb <- performance(pred.nb, "tpr", "fpr")
plot(perf.nb)
```



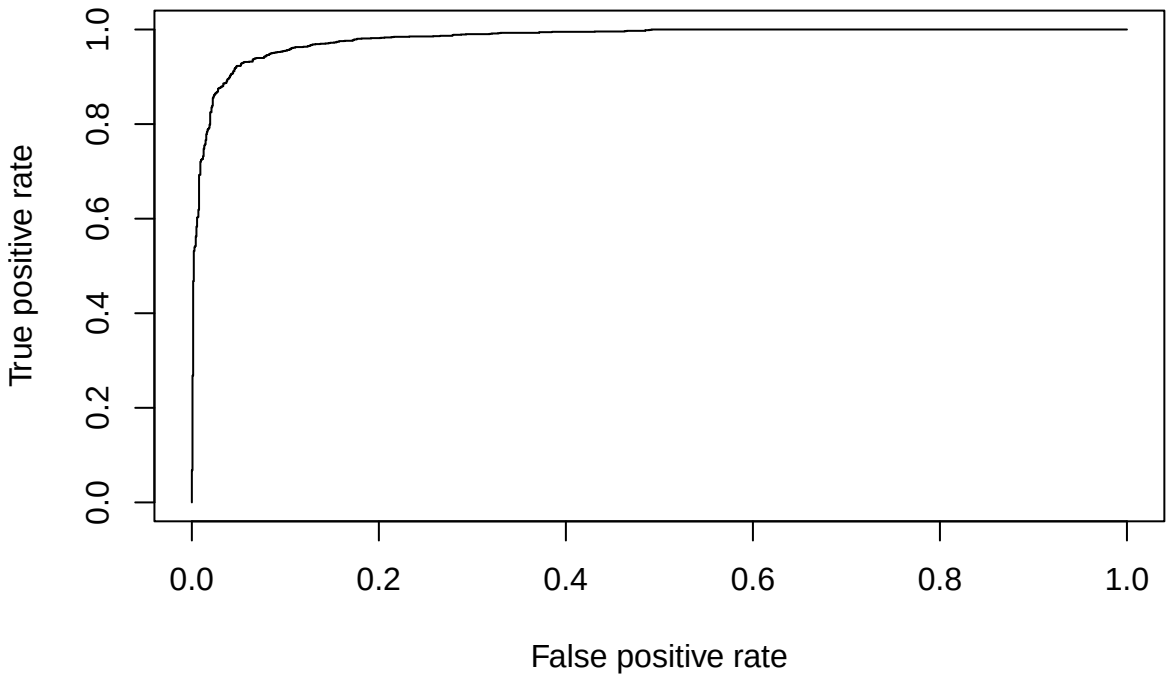

```
auc.nb <- performance(pred.nb, "auc")@y.values[[1]]
```

LDA

```
class.lda <- predict(fit.lda, newdata = DTest)
confusionMatrixLDA <- confusionMatrix(class.lda, DTest$status)
print(confusionMatrixLDA$overall['Accuracy'])
```

```
##  Accuracy
## 0.9327731
```

```
pred.lda = predict(fit.lda, newdata=DTest[,-c(9, 60, 62, 64, 69, 72, 88)], type = "prob")
pred.lda <- prediction(pred.lda[,2], DTest$status)
perf.lda <- performance(pred.lda, "tpr", "fpr")
plot(perf.lda)
```



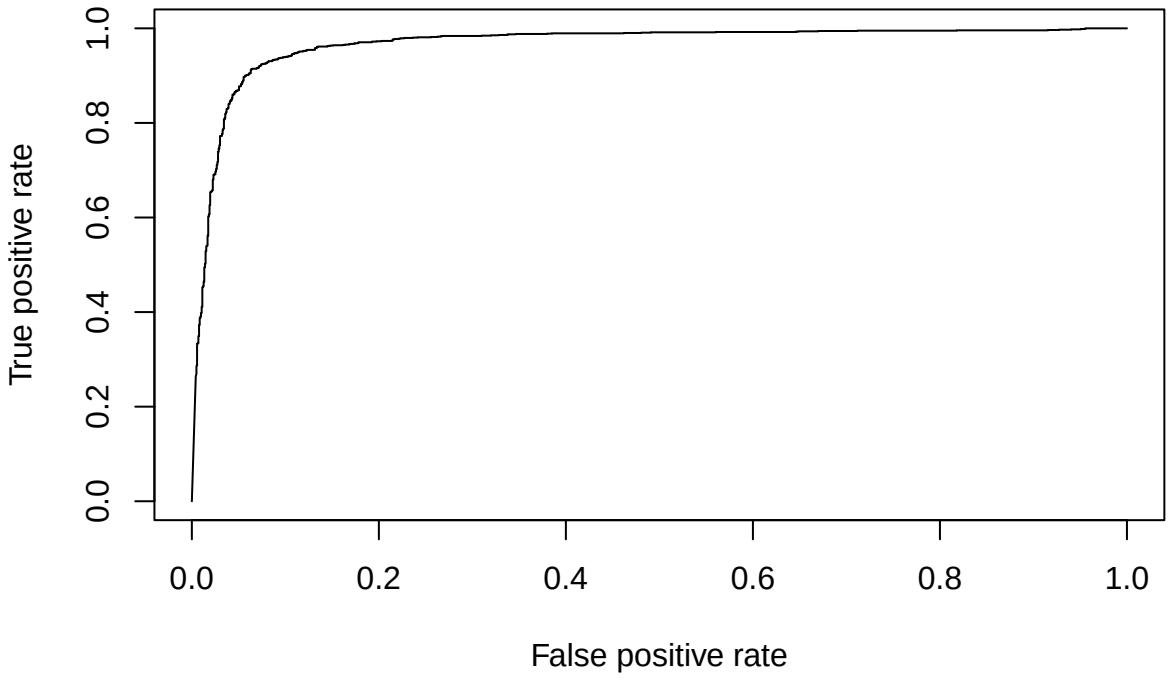

```
auc.lda <- performance(pred.lda, "auc")@y.values[[1]]
```

QDA

```
class.qda <- predict(fit.qda, newdata = DTest)
confusionMatrixQDA <- confusionMatrix(class.qda, DTest$status)
print(confusionMatrixQDA$overall['Accuracy'])
```

```
##  Accuracy
## 0.9072129
```

```
pred.qda = predict(fit.qda, newdata=DTest[,c(88, 86, 87, 21, 51, 83, 57, 36, 45, 27, 2, 5, 79, 11, 25, 8
pred.qda <- prediction(pred.qda[,2], DTest$status)
perf.qda <- performance(pred.qda, "tpr", "fpr")
plot(perf.qda)
```




```
auc.qda <- performance(pred.qda, "auc")@y.values[[1]]
```

SVM - Séparateur Linéaire

```
#class.Lin.svm <- predict(fit.Lin.svm, newdata = DTest)
#confusionMatrixLinSVM <- confusionMatrix(class.Lin.svm, DTest$status)
#print(confusionMatrixLinSVM$overall['Accuracy'])

#pred.Lin.svm = predict(fit.Lin.svm, newdata = DTest, type = "prob")
#pred.Lin.svm <- prediction(pred.Lin.svm[,2], DTest$status)
#perf.Lin.svm <- performance(pred.Lin.svm, "tpr", "fpr")
#plot(perf.Lin.svm)
#auc.Lin.svm <- performance(pred.Lin.svm, "auc")@y.values[[1]]
```

SVM - Séparateur Quadratique

```
#class.Quad.svm <- predict(fit.Quad.svm, newdata = DTest)
#confusionMatrixQuadSVM <- confusionMatrix(class.Quad.svm, DTest$status)
#print(confusionMatrixQuadSVM$overall['Accuracy'])

#pred.Quad.svm = predict(fit.Quad.svm, newdata=DTest, type = "prob")
#pred.Quad.svm <- prediction(pred.Quad.svm[,2], DTest$status)
#perf.Quad.svm <- performance(pred.Quad.svm, "tpr", "fpr")
#plot(perf.Quad.svm)
#auc.Quad.svm <- performance(pred.Quad.svm, "auc")@y.values[[1]]
```

```
#class.Poly.svm <- predict(fit.Poly.svm, newdata = DTest)
#confusionMatrixPolySVM <- confusionMatrix(class.Poly.svm, DTest$status)
#print(confusionMatrixPolySVM$overall['Accuracy'])

#pred.Poly.svm = predict(fit.Poly.svm, newdata=DTest, type = "prob")
#pred.Poly.svm <- prediction(pred.Poly.svm[,2], DTest$status)
#perf.Poly.svm <- performance(pred.Poly.svm, "tpr", "fpr")
#plot(perf.Poly.svm)
#auc.Poly.svm <- performance(pred.Poly.svm, "auc")@y.values[[1]]
```

Noyau Radial

Comparaison des modèles

On fait un dataframe récapitulatif avec en ligne les différents modèles et en colonne on aura accuracy, sensibilité, spécificité et AUC.

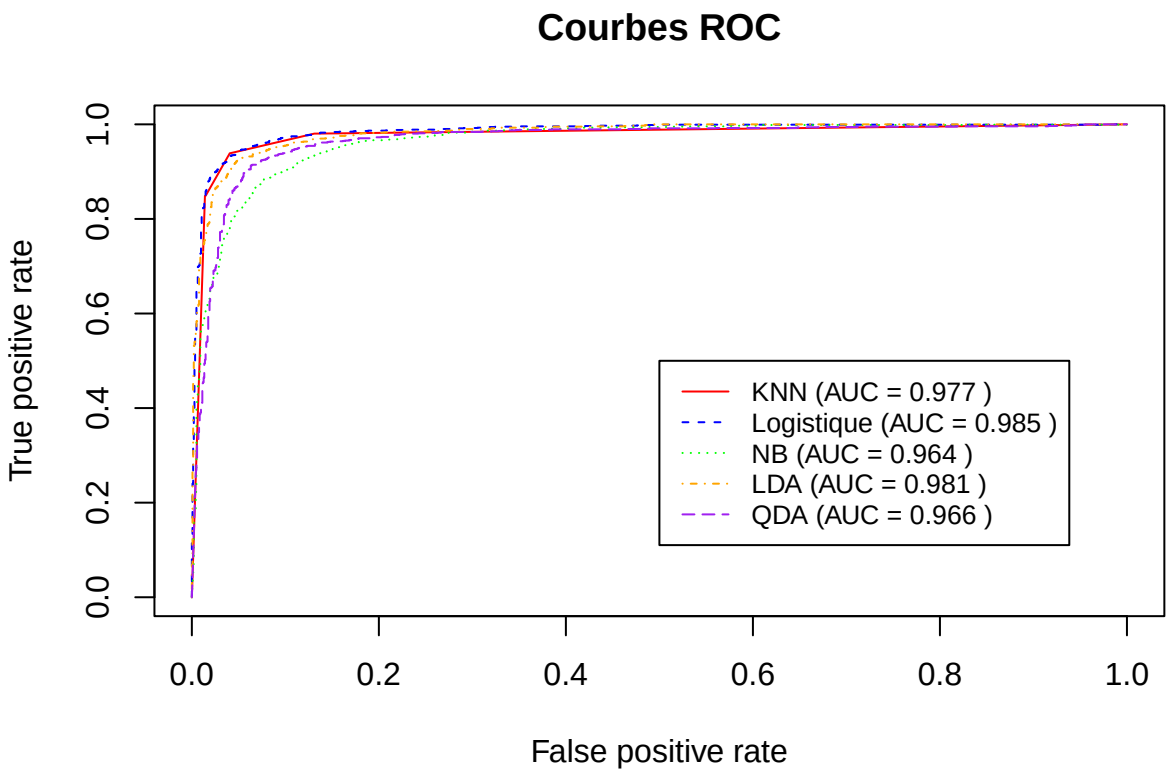
Pour l'instant, on ne fait que pour KNN, logistique, NB, LDA et QDA.

```
models <- c("KNN", "Logistique", "NB", "LDA", "QDA")
accuracy <- c(confusionMatrixBestK$overall['Accuracy'], confusionMatrixLR$overall['Accuracy'], confusio
sensibilite <- c(confusionMatrixBestK$byClass['Sensitivity'], confusionMatrixLR$byClass['Sensitivity'],
specificite <- c(confusionMatrixBestK$byClass['Specificity'], confusionMatrixLR$byClass['Specificity'],
auc <- c(auc.knn, auc.lr, auc.nb, auc.lda, auc.qda)

df_models <- data.frame(models, accuracy, sensibilite, specificite, auc)
df_models
```

```
##       models  accuracy sensibilite specificite       auc
## 1        KNN 0.9488796   0.9586835   0.9390756 0.9769111
## 2 Logistique 0.9464286   0.9460784   0.9467787 0.9852774
## 3         NB 0.7405462   0.9915966   0.4894958 0.9640560
## 4        LDA 0.9327731   0.9257703   0.9397759 0.9813622
## 5        QDA 0.9072129   0.9565826   0.8578431 0.9664962
```

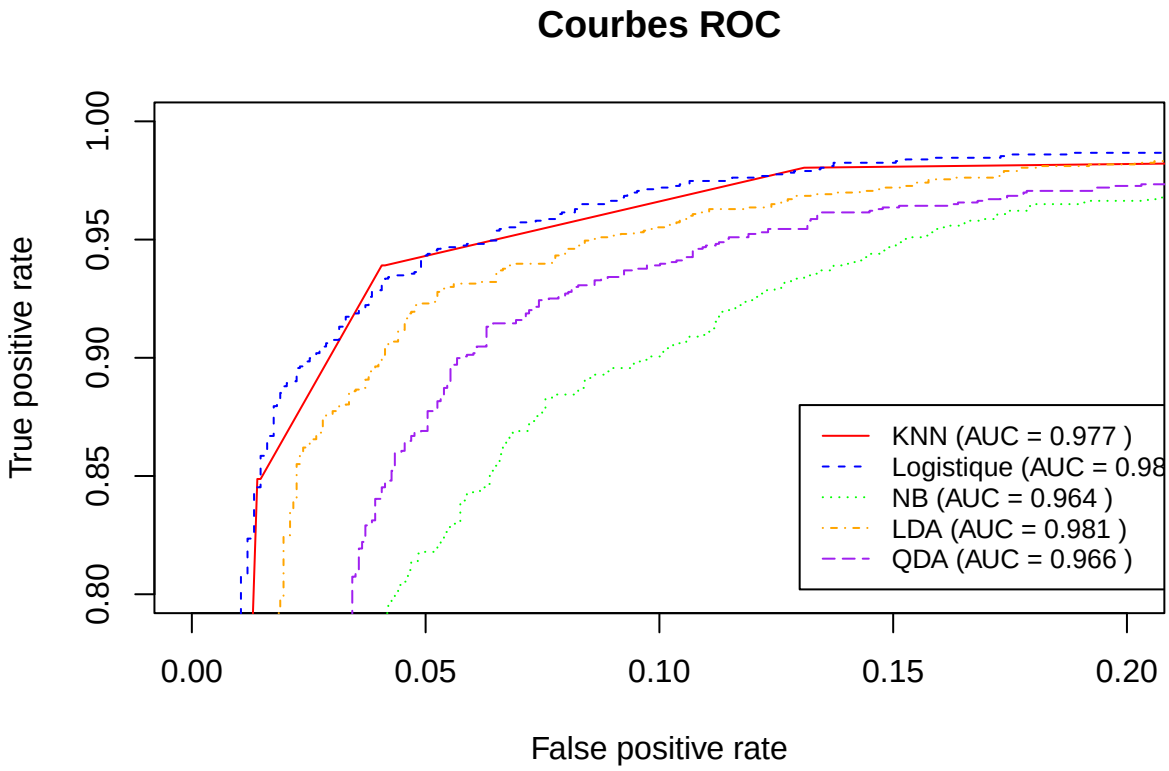
On fait 1 graphqiue avec toutes les courbes ROC et on affiche les valeurs AUC.

```
legend_labels <- c(
    paste("KNN (AUC =", round(auc.knn, 3), ")"),
    paste("Logistique (AUC =", round(auc.lr, 3), ")"),
    paste("NB (AUC =", round(auc.nb, 3), ")"),
    paste("LDA (AUC =", round(auc.lda, 3), ")"),
    paste("QDA (AUC =", round(auc.qda, 3), ")")
)

plot(perf.knn, col = "red", main = "Courbes ROC", lty = 1)
plot(perf.lr, col = "blue", add = TRUE, lty = 2)
plot(perf.nb, col = "green", add = TRUE, lty = 3)
plot(perf.lda, col = "orange", add = TRUE, lty = 4)
plot(perf.qda, col = "purple", add = TRUE, lty = 5)

legend(0.5, 0.5, legend = legend_labels, col = c("red", "blue", "green", "orange", "purple"), lty = c(1
```



On fait le même graphique mais en zoom sur la partie gauche.

```
legend_labels <- c(
    paste("KNN (AUC =", round(auc.knn, 3), ")"),
    paste("Logistique (AUC =", round(auc.lr, 3), ")"),
    paste("NB (AUC =", round(auc.nb, 3), ")"),
    paste("LDA (AUC =", round(auc.lda, 3), ")"),
    paste("QDA (AUC =", round(auc.qda, 3), ")")
)

plot(perf.knn, col = "red", main = "Courbes ROC", lty = 1, xlim = c(0, 0.2), ylim = c(0.8, 1))
plot(perf.lr, col = "blue", add = TRUE, lty = 2)
plot(perf.nb, col = "green", add = TRUE, lty = 3)
plot(perf.lda, col = "orange", add = TRUE, lty = 4)
plot(perf.qda, col = "purple", add = TRUE, lty = 5)

legend(0.13, 0.88, legend = legend_labels, col = c("red", "blue", "green", "orange", "purple"), lty = c
```


Courbes ROC
True positive rate
1.00
0.95
0.90
0.85
0.80
0.00
0.05
0.10
0.15
0.20
False positive rate
KNN (AUC = 0.977)
Logistique (AUC = 0.98
NB (AUC = 0.964)
LDA (AUC = 0.981)
QDA (AUC = 0.966)