

```

install.packages("KernelKnn")

data(ionosphere, package = 'KernelKnn')

apply(ionosphere, 2, function(x) length(unique(x)))

ionosphere = ionosphere[, -2]

X = scale(ionosphere[, -ncol(ionosphere)])

y = ionosphere[, ncol(ionosphere)]

y = c(1:length(unique(y)))[ match(ionosphere$class, sort(unique(ionosphere$class))) ]

spl_train = sample(1:length(y), round(length(y) * 0.75))

spl_test = setdiff(1:length(y), spl_train)

str(spl_train)

str(spl_test)

acc = function (y_true, preds) {
  out = table(y_true, max.col(preds, ties.method = "random"))
  acc = sum(diag(out))/sum(out)
  acc
}

library(KernelKnn)

preds_TEST = KernelKnn(X[spl_train, ], TEST_data = X[spl_test, ], y[spl_train], k = 5 ,
  method = 'euclidean', weights_function = NULL, regression = F,
  Levels = unique(y))

head(preds_TEST)

preds_TEST_tric = KernelKnn(X[spl_train, ], TEST_data = X[spl_test, ], y[spl_train], k = 10 ,
  method = 'canberra', weights_function = 'tricube', regression = F,
  Levels = unique(y))

head(preds_TEST_tric)

```

```

norm_kernel = function(W) {
  W = dnorm(W, mean = 0, sd = 1.0)
  W = W / rowSums(W)
  return(W)
}

preds_TEST_norm = KernelKnn(X[spl_train, ], TEST_data = X[spl_test, ], y[spl_train], k = 10 ,
  method = 'canberra', weights_function = norm_kernel, regression = F,
  Levels = unique(y))
head(preds_TEST_norm)

fit_cv_pair1 = KernelKnnCV(X, y, k = 10 , folds = 5, method = 'canberra',
  weights_function = 'tricube', regression = F,
  Levels = unique(y), threads = 5)
str(fit_cv_pair1)
fit_cv_pair2 = KernelKnnCV(X, y, k = 9 , folds = 5, method = 'canberra',
  weights_function = 'epanechnikov', regression = F,
  Levels = unique(y), threads = 5)
str(fit_cv_pair2)

acc_pair1 = unlist(lapply(1:length(fit_cv_pair1$preds),
  function(x) acc(y[fit_cv_pair1$folds[[x]]],
    fit_cv_pair1$preds[[x]])))
acc_pair1
cat('accuracy for params_pair1 is :', mean(acc_pair1), '\n')
acc_pair2 = unlist(lapply(1:length(fit_cv_pair2$preds),
  function(x) acc(y[fit_cv_pair2$folds[[x]]],
    fit_cv_pair2$preds[[x]])))
acc_pair2
cat('accuracy for params_pair2 is :', mean(acc_pair2), '\n')

```