```
# X[i]           : i is an independent column
# Y[i]           : i is an dependent column
# Lo[i]          : least of column i
# Hi[i]          : most of column i
# Num[i]= 0 or 1 : i is a numeric column, is "0" if we want to minimize

BEGIN{ FS=1; Big=1E32 }
     { NR==1 ? header() : data() }
END  { main(); rogues() }

#------------------------------------------------------------------------------
function header(     i) {
  srand(SEED ? SEED : 1234567891)
  for(i=1;i<=NF;i++) {
    if ($i ~ /^[A-Z]/) Num[i] = $i ~ /\+$/
    if ($i ~ /[!+-]$/) Y[i] else X[i]
  }
  for(i in Num) Hi[i] = - (Lo[i]=Big) }

function data(    i) {
  for(i=1;i<=NF;i++)
    if (i != "?") {
      if (i in Num)  {
        $i += 0
        Lo[x] = min($i, Lo[x])
        Hi[x] = max($i, Hi[x]) }
  }
  Row[NR-1][i]=$i }

#------------------------------------------------------------------------------
function ydist(row,       i,y) {
  for(i in Y)
    y += abs(norm(i,row[i]) - Num[i])^2
  return (y/length(Y)) ^ 0.5 }

function xdist(row1,row2,       i,y) {
  for(i in X)
    x  += dist(i,row1[i],row2[i])^2
  return (x/length(X)) ^ 0.5 }

function dist(i,a,b) {
  if (a=="?" and b=="?") return 1
  if (i in Num) {
    a = norm(i,a)
    b = norm(i,b)
    a = a != "?" ? a : (b<0.5 ? 1 : 0)
    b = b != "?" ? b : (a<0.5 ? 1 : 0)
    return abs(a - b) }
  return a != b }

function best(k,centers,    i,best,lo) {
  lo=Big
  centers[any(Row)] # initilize the centers
  for(i=2,i<=k,i++) centers[near(centers)] # find other centers
  for(c in centers)  { # find center with least y
    y = ydist(Row[c])
    if (y<lo) {
       lo=y; out=c}}
  return c}

function near(centers) {
  for(j=1;j<=Samples;j++) {
    c = nearest(centers,r=any(Row))
    tmp[r] = xdist(Row[c], Row[r])^2
  }
  return pick(tmp) }

function nearest(centers,r,       lo,c,d,out) {
  lo=big
  for(c in centers) {
    d = xdist(Row[r], Row[c])
    if (d<lo) {
      lo=d; out=c  }}
  return out }

#------------------------------------------------------------------------------
function rogues(    i) {
  for(i in SYMTAB) if (i~/^[a-z]) print("?",i, typeof(SYMTAB[i]) }

function pick(a,   x,r,all) {
  for (x in a) all += a[x]
  r=rand()
  for(x in a) {
    r -= a[x]/all
    if (r<=0) return x }
  return x }

function norm(i,x) {
  return x=="?" ? x : (i-Lo[x])/(Hi[x] - Lo[X] + 1/Big) }

function min(x,y) { return x<y  ? x : y }
function max(x,y) { return x>y  ? x : y }
function abs(x)   { return x>=0 ? x : -x }
function any(a)   { return int(0.5 + rand()*length(a)) }
```