k-NN Regression Implementation & Analysis

Stephen J George

2024-03-13

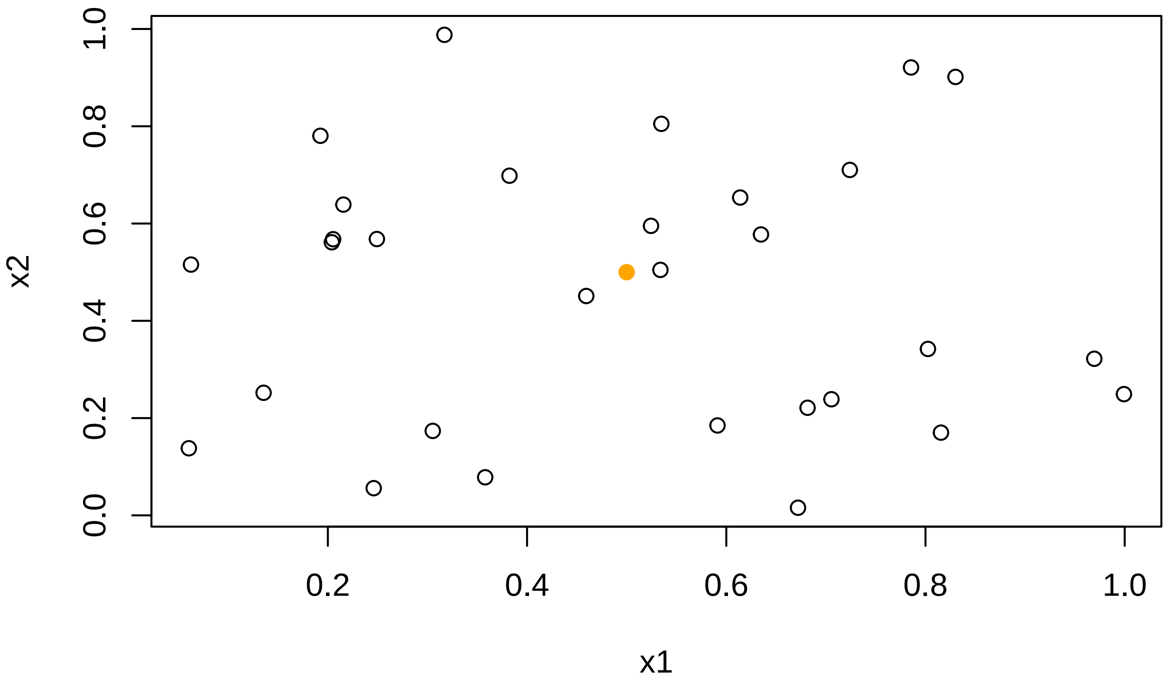
Loading some packages… for 3d plotting.

```
#library(rgl)
library(scatterplot3d)
#options(rgl.useNULL = TRUE) # Suppress the separate window.
```

We take a set of 30 random points for 2 predictor variables and defined a relationship for the response variable as well.Initially we consider 2 test points and estimate(test) value for response as well. We also plot a 3d-scatterplot for visualization among the variables as (y is dependent on x1 and x2).The test points are shown in orange in both 2d and 3d.

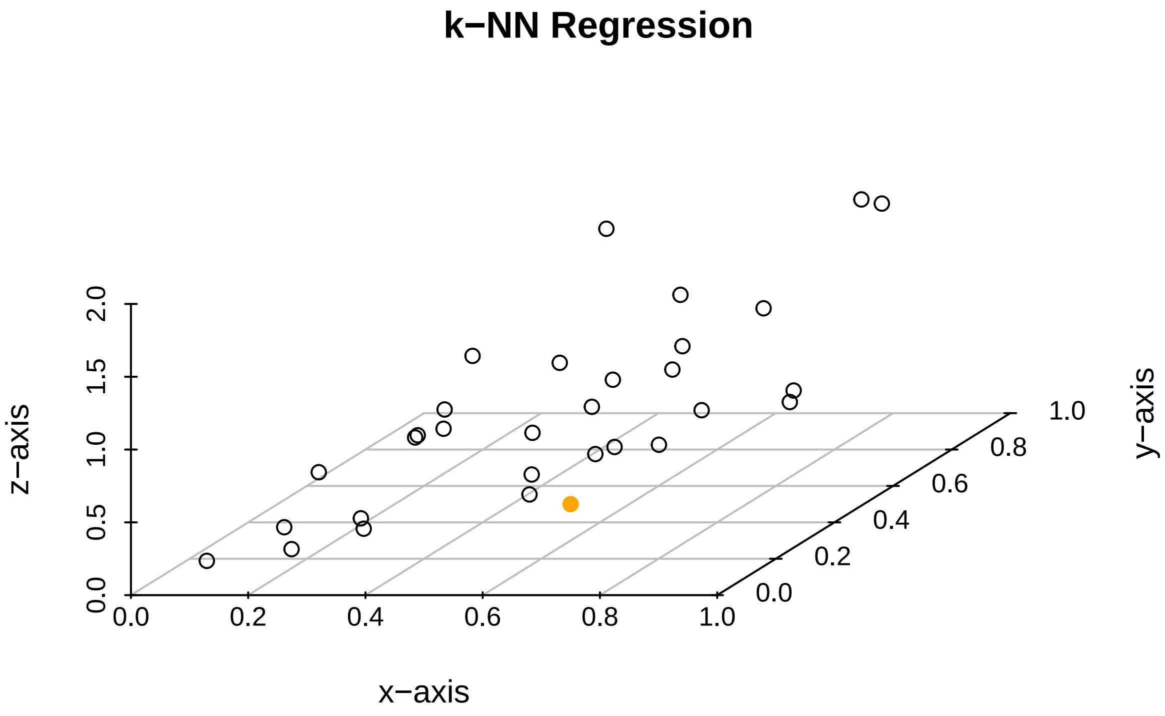
```
n = 30
x1 <- runif(n)
x2 <- runif(n)
y = x1 + x2^3
plot(x1,x2)

x1.test <- x2.test <- 0.5
y.test <- 0
points(x1.test,x2.test,col="orange",pch=19)
```



```
y.est=rep(0,n)
# plot3d(x1,x2,y,col="green",size=3)
# rglwidget()

sd <- scatterplot3d(x1,x2,y,xlab="x-axis",ylab="y-axis",zlab="z-axis",main="k-NN Regression",
                    angle = 45,box = FALSE)
sd$points3d(x1.test,x2.test,y.test,col="orange",pch=19)
```



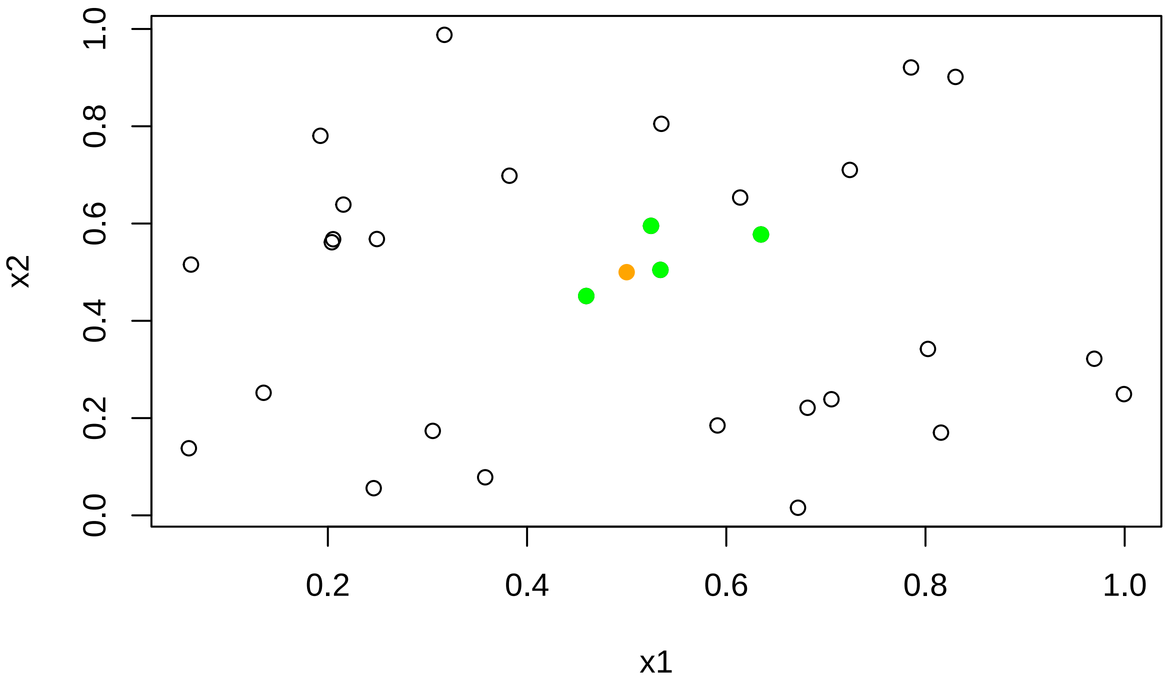
The number of neighbors is taken as 4. We compute the euclidean distance of the test points from all other points and choose the nearest k neighbors. The estimate test value of the response variable is the mean of the y values that correspond to the nearest k neighbors.

```
#number of neighbors to consider
k = 4
#distances array
d <- NULL
for (i in 1:n){
    d[i] = sqrt((x1.test-x1[i])^2 + (x2.test-x2[i])^2)
}

Sorted <- order(d)
Sorted[1:k]
```

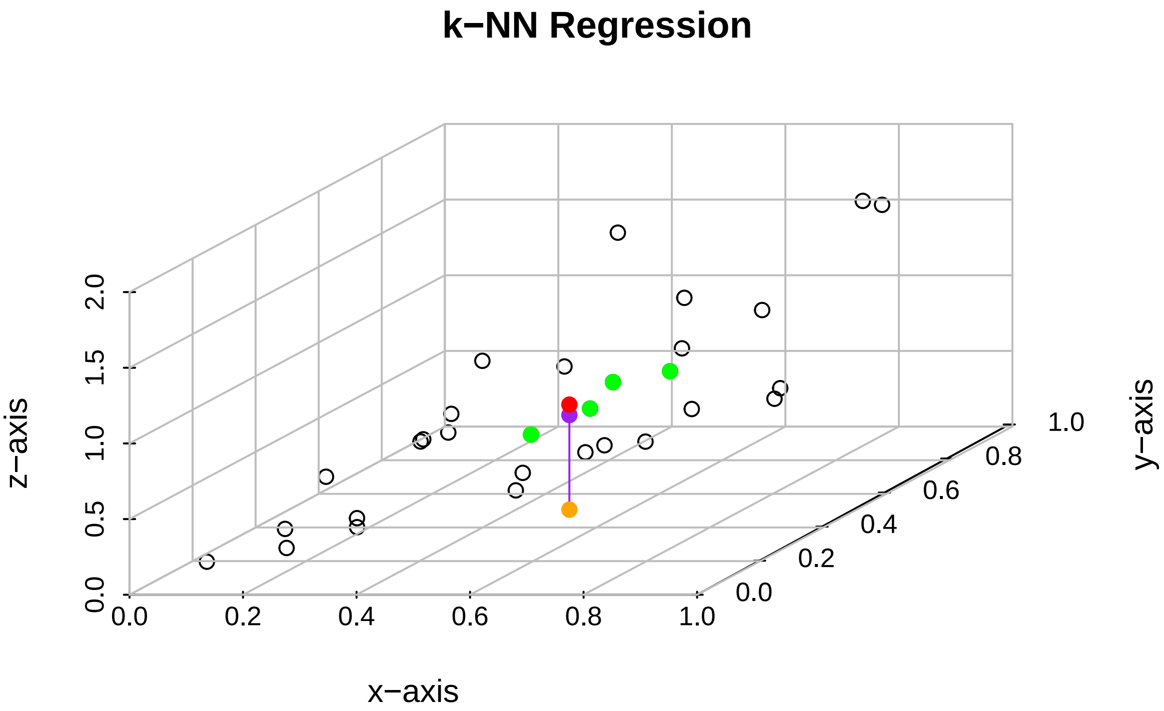
```
## [1] 26 21 10 18
```

```
plot(x1,x2)
points(x1.test,x2.test,col="orange",pch=19)
points(x1[Sorted[1:k]],x2[Sorted[1:k]],col="green",pch=19)
```



```
y.test = mean(y[Sorted[1:k]])

special = 0.5 + (0.5)^3
source('http://www.sthda.com/sthda/RDoc/functions/addgrids3d.r')
sd <- scatterplot3d(x1,x2,y,xlab="x-axis",ylab="y-axis",zlab="z-axis",main="k-NN Regression",
                    angle = 40.5,box = FALSE,grid=FALSE)
addgrids3d(x1,x2,y, grid = c("xy", "xz", "yz"))
sd$points3d(x1.test,x2.test,special,col="purple",pch=19,type="h")
sd$points3d(x1.test,x2.test,0,col="orange",pch=19)
sd$points3d(x1[Sorted[1:k]],x2[Sorted[1:k]],y[Sorted[1:k]],col="green",pch=19)
sd$points3d(x1.test,x2.test,y.test,col="red",pch=19)
```



An effort has been made to depict the test points and the estimated value of response variable in 3d (as well as some parts in 2d).The red point denotes the estimated y value while the purple depicts the y value corresponding to the test points x1.test and x2.test.

All this regression is however only done for a single case of test points in the (input) domain and the same can be done for every single point in the input domain space.