```
import numpy as np
import matplotlib.pyplot as plt
import random as rnd
unifRN= rnd.uniform


#square has corners (0.5,-0.5),(0.5,0.5),(-0.5,0.5),(-0.5,-0.5)


def dist2Center(x,y):
    if abs(x)<=0.5 and abs(y)<= 0.5 :
        d = np.sqrt(x**2 +y**2)
    if d <= np.sqrt(2):
        return d
    else:
        return None



def dist2Edge(x,y):
    if abs(x)<=0.5 and abs(y)<= 0.5 :
        d = min(abs(0.5-x), abs(-0.5-x), abs(0.5-y), abs(-0.5-y))
    else:
        print('point {},{}, not in square'.format(x,y))
        return None
    return d


n = 10000


squareArray = np.zeros(shape=(n,2))
for i in range(n):
    squareArray[i,0], squareArray[i,1]= unifRN(-0.5,0.5),unifRN(-0.5,0.5)



minRad = 0
maxRad = 0.5

eps = 0.01



while(1):
    testRad = (minRad+maxRad)/2
    print(testRad)

    distIn = []
    distOut = []
    for j in range(n):
        d2C = dist2Center(squareArray[j,0], squareArray[j,1])
        if  d2C <= testRad:
            distIn.append(d2C)

        else:
            d2E = dist2Edge(squareArray[j,0],squareArray[j,1])
            distOut.append(d2E)

    avgDistIn = np.mean(distIn)
    avgDistOut = np.mean(distOut)

    diffDist = abs(avgDistIn-avgDistOut)
    if diffDist <= eps: break

    if(avgDistIn > avgDistOut):
        maxRad = testRad
    else:
        minRad = testRad

print('Radius of a circle: {}' .format(testRad))
```