**BL.EN.U4AIE22137**

**R G ROHIT**

**Report on Functions in Python Code**

1. Introduction:

This report provides an overview and analysis of the functions utilized in the Python code provided. The code focuses on Manhattan distance, Euclidian distance, KNN Classifier, label encoding, and one-hot encoding.

2. Functions Overview:

Manhattan distance and Euclidian distance calculation of vectors

In order to calculate the Manhattan and Euclidian distance of vectors, we need to take the required dimension from the user. If the dimensions are not equal, it will thrown an error message. If the dimensions are equal, it will continue to the next function that is to calculate the Manhattan distance and Euclidian distance respectively. The loop ranges till the length of either vector 1 or vector 2 which is basically the number of dimensions it has. Then we substitute it into the formula and calculate the distance.

KNN

In this code, we are using the Manhattan distance formula to calculate the distance between two vectors, vector3 and vector4. The formula involves looping through the dimensions of the vectors and taking the absolute difference of each coordinate. We are sorting the values of x and y respectively in the for loop and then using the value of k, we calculate the nearest neightbour .The final distance is then calculated by summing up these differences.

Label Encoding

In label encoding, we need to assign a number to particular values, in order to do that we created a dictionary to map unique labels and a list to store values before appending them into it. In the for loop, if that particular value is not there in the dictionary, it adds that value to it and assigns a number to it, to assign a number greater than the current value number, we r incrementing it by 1.

One hot Encoding

In one hot encoding, we need to assign value zero and one depending on the elements. Hence, we need a for loop where if the category is equal to the number, then it will append 1 to that element and 0 to the remaining elements.

**Pseudo code**

import math

function vector\_dim(dimension):

vector = []

for i in range(0, dimension):

vector.append(input("enter element:"))

return vector

vector1 = vector\_dim(input("enter the dimension of X vector"))

vector2 = vector\_dim(input("enter the dimension of Y vector"))

function euclidian\_vector(vector1, vector2):

if length of vector1 is not equal to length of vector2:

print "The dimensions are not equal"

distance = 0

for i in range(0, length of vector1):

distance = distance + (vector2[i] - vector1[i])^2

return square root of distance

function manhattan\_vector(vector1, vector2):

if length of vector1 is not equal to length of vector2:

print "The dimensions are not equal"

distance = 0

for i in range(0, length of vector1):

distance = distance + |vector2[i] - vector1[i]|

return distance

function manhattan\_distance(vector3, vector4):

distance = 0

for i in range(0, length of vector3):

distance = distance + |vector4[i] - vector3[i]|

return distance

x = [[150], [155], [160], [161], [158]]

y = [[50], [55], [60], [59], [65]]

target = ['medium', 'medium', 'large', 'large', 'large']

value = []

distance = {}

for i in range(0, length of target):

manhattan\_dist = sum of absolute differences between elements of x[i] and y[i]

value.append(manhattan\_dist)

distance[manhattan\_dist] = target[i]

print x[i], y[i], target[i], manhattan\_dist

k = input("Enter value of K: ")

sorted\_result = sort distance by keys

function label\_encoder(data):

map = {}

encoded\_data = []

counter = 0

for i in data:

if i not in map:

map[i] = counter

counter = counter + 1

encoded\_data.append(map[i])

return encoded\_data

function one\_hot\_encoding(categories, numerical):

one\_hot\_encoded = []

for i in range(0, length of numerical):

encoded = []

for j in range(0, length of categories):

if numerical[i] == j:

encoded.append(1)

else:

encoded.append(0)

one\_hot\_encoded.append(encoded)

return one\_hot\_encoded

result\_euclidian = euclidian\_vector(vector1, vector2)

result\_manhattan = manhattan\_vector(vector1, vector2)

print "euclidian is:", result\_euclidian

print "manhattan is:", result\_manhattan

print "sorted distance:", sorted\_result

print "Nearest neighbours are:", k

for i in range(0, k):

print sorted\_result[i]

data = ['short', 'tall', 'very tall', 'short']

encoded\_data = label\_encoder(data)

print encoded\_data

size1 = ['small', 'medium', 'large', 'large', 'medium']

numerical = [0, 1, 2, 2, 1]

size = list(set(size1))

encoded\_data = one\_hot\_encoding(size, numerical)

for encoding in encoded\_data:

print encoding