Project Falling Detection: Python + kNN + Colab

Akanksha Singh(19745)

Table of contents

Introduction

Step 1: used KNN to manually to calculate the distance and predict the result.
Step 2: Calculated the value of K as 3 and got the result as positive.
Step 3: Developed the K nearest neighbors in Python from scratch.

Using KNN to manually calculate the distance and predict the result.

Accelerometer Data			Gyroscope Data			Fall (+), Not (-)
x	y	z	x	y	z	+/-
1	2	3	2	1	3	-
2	1	3	3	1	2	-
1	1	2	3	2	2	-
2	2	3	3	2	1	-
6	5	7	5	6	7	+
5	6	6	6	5	7	+
5	6	7	5	7	6	+
7	6	7	6	5	6	+
7	6	5	5	6	7	??

For accelerometer and Gyroscope:

Here, K=3. So picking 3 smallest values. According to which result is +.

A	B	C	D	E	F	G	H
7	6	5	5	6	7		+
x	y	z	x	y	z		
1	2	3	2	1	3	106	-
2	1	3	3	1	2	108	-
1	1	2	3	2	2	115	-
2	2	3	3	2	1	101	-
6	5	7	5	6	7	6	+
5	6	6	6	5	7	7	+
5	6	7	5	7	6	10	+
7	6	7	6	5	6	7	+

Develop K nearest neighbours in Python from scratch.

Modified the python code for this dataset:

```
# Test distance function
dataset = [[7,6,5,5,6,7,1],
  [1,2,3,2,1,3,0],
  [2,1,3,3,1,2,0],
  [1,1,2,3,2,2,0],
  [2,2,3,3,2,1,0],
  [6,5,7,5,6,7,1],
  [5,6,6,6,5,7,1],
  [5,6,7,5,7,6,1],
  [7,6,7,6,5,6,1]]


# row 0 (i.e., dataset[0]) is the one to be predicted
prediction = predict_classification(dataset, dataset[0], 3)

# - dataset[0][-1] is the last element of row 0 of dataset
# - Display
#    Expected 0, Got 0.
print('Expected %d, Got %d.' % (dataset[0][-1], prediction))
```

```
Expected 1, Got 1.
```

Conclusion

The expected value was 1 and the actual result also came out to be 1.

Github Link

https://github.com/codeyogg/Machine_learning/blob/main/Supervised_Learning/FallingPredictionusingKNN/FallingPredictionUsingKNN.ipynb