

# Testing k-NN using emotion features

## First approach

The first try is uses Linear NN Search with knn equal to 1, Euclidean distance and uses the 5 emotions provided by indicoio api.

[knn1NKaggle.csv](#)

0.60093

2 days ago by Carlos A

Applied Knn LinearNNSearch: 1 neighbour, Euclidean distance, 5 indicoio emotions.

The result accuracy effectiveness was 0.60093 which is low, as it's just a bit higher than a random approach.

## Second approach

I wanted to experiment with the available parameters for the knn algorithm so made tests with different configurations of distance functions (Euclidean, Chebyshev, Filtered, Manhattan and Minkowski) and nearest neighbour search algorithms (Ball Tree, KD Tree, Cover Tree and Linear Search).

<b>BallTreeTopDown1NeighEuclideanKaggle.csv</b> a day ago by <a href="#">Carlos A</a> Applied Knn BallTree TopDown: 1 neighbour, Euclidean distance, 5 indicoio emotions.	0.60093
<b>Linear1NieghMinkoswkiKaggle.csv</b> a day ago by <a href="#">Carlos A</a> Applied Knn LinearNNSearch: 1 neighbour, Minkowski distance, 5 indicoio emotions.	0.60093
<b>Linear1NeighFilteredKaggle.csv</b> a day ago by <a href="#">Carlos A</a> Applied Knn LinearNNSearch: 1 neighbour, Filtered distance, 5 indicoio emotions.	0.61014
<b>Linear1NeighChebyshevKaggle.csv</b> 2 days ago by <a href="#">Carlos A</a> Applied Knn LinearNNSearch: 1 neighbour, Chebyshev distance, 5 indicoio emotions.	0.61436
<b>Linear1NeighManhattanKaggle.csv</b> 2 days ago by <a href="#">Carlos A</a> Applied Knn LinearNNSearch: 1 neighbour, Manhattan distance, 5 indicoio emotions.	0.58965
<b>KDTree1NeighKaggle.csv</b> 2 days ago by <a href="#">Carlos A</a> Applied Knn KDTree: 1 neighbour, Euclidean distance, 5 indicoio emotions.	0.60093
<b>Linear2NeighKaggle.csv</b> 2 days ago by <a href="#">Carlos A</a> Applied Knn LinearNNSearch: 2 neighbour, Euclidean distance, 5 indicoio emotions.	0.56891
<b>knn1NKaggle.csv</b> 2 days ago by <a href="#">Carlos A</a> Applied Knn LinearNNSearch: 1 neighbour, Euclidean distance, 5 indicoio emotions.	0.60093
<b>Filtered1NeighEuclideanKaggle.csv</b> a day ago by <a href="#">Carlos A</a> Applied Knn FilteredNeighbourSearch: 1 neighbour, Euclidean distance, 5 indicoio emotions.	0.60093

Carlos Augusto Amador Manilla  
A01329447

The configuration which gave better results was Linear Search with 1 neighbour and Chebyshev distance function. I decided to make more experiments using Chebyshev distance function, with more neighbours and more features (4 personality features provided by indicoio api). But the results were not better.

[Linear1NeighChevis9FKaggle.csv](#) 0.58825

a day ago by Carlos A

Applied Knn LinearNNSearch: 1 neighbour, Chebyshev distance, 5 indicoio emotions, 4 indicoio personality.

[Linear5NeighChevyshevKaggle.csv](#) 0.59658

a day ago by Carlos A

Applied Knn LinearNNSearch: 5 neighbour, Chebyshev distance, 5 indicoio emotions.

Then, I made more unsuccessful experiments to try to increase effectiveness.

[results.csv](#) 0.59462

a day ago by Carlos A

Applied Knn LinearNNSearch: 1-7 neighbour, Chebyshev distance, 5 indicoio emotions, 4 indicoio personality, average, more weighth to more neighbours, no balance.

[results.csv](#) 0.58568

a day ago by Carlos A

Applied Knn LinearNNSearch: 1-5 neighbour, Chebyshev distance, 5 indicoio emotions, 4 indicoio personality, average, more weighth to more neighbours.

[results.csv](#) 0.58991

a day ago by Carlos A

Applied Knn LinearNNSearch: 1-5 neighbour, Chebyshev distance, 5 indicoio emotions, 4 indicoio personality, average.

[results.csv](#) 0.59358

a day ago by Carlos A

Applied Knn LinearNNSearch: 1-5 neighbour, Chebyshev distance, 5 indicoio emotions, 4 indicoio personality, 2/3 balance sum.

Try to solve the lack of balance in the training set by multiplying by  $\frac{2}{3}$  and  $\frac{1}{3}$  the prediction provided by weka, get the average of different knn values, try to give more weight to more neighbours.