1.Hamming Code

Hamming distance calculates the distance between two binary vectors, also referred to as binary strings or bitstrings for short. You are most likely going to encounter bitstrings when you one-hot encode categorical columns of data.

```
def hamming_distance(a, b):
    return sum(abs(e1 - e2) for e1, e2 in zip(a, b)) / len(a)

row1 = [0, 0, 0, 0, 0, 1]
row2 = [0, 0, 0, 0, 1, 0]

dist = hamming_distance(row1, row2)
print(dist)
```

```
0.3333333333333333
```

2.Euclidean Distance

Euclidean distance calculates the distance between two real-valued vectors. You are most likely to use Euclidean distance when calculating the distance between two rows of data that have numerical values, such a floating point or integer values.

- EuclideanDistance = sqrt(sum for i to N (v1[i] – v2[i])^2)

```
from math import sqrt

def euclidean_distance(a, b):
  return sqrt(sum((e1-e2)**2 for e1, e2 in zip(a,b)))

row1 = [10, 20, 15, 10, 5]
row2 = [12, 24, 18, 8, 7]

dist = euclidean_distance(row1, row2)
print(dist)
```

```
6.082762530298219
```

```
from scipy.spatial.distance import euclidean

row1 = [10, 20, 15, 10, 5]
row2 = [12, 24, 18, 8, 7]

dist = euclidean(row1, row2)
print(dist)
```

```
6.082762530298219
```

3) Manhattan Distance (Taxicab or City Block Distance)

The Manhattan distance, also called the Taxicab distance or the City Block distance, calculates the distance between two real-valued vectors. It is perhaps more useful to vectors that describe objects on a uniform grid, like a chessboard or city blocks.

- ManhattanDistance = sum for i to N sum |v1[i] – v2[i]|

```
from math import sqrt

def manhattan_distance(a, b):
  return sum(abs(e1-e2) for e1, e2 in zip(a,b))

row1 = [10, 20, 15, 10, 5]
row2 = [12, 24, 18, 8, 7]

dist = manhattan_distance(row1, row2)
print(dist)
```

```
13
```

```
from scipy.spatial.distance import cityblock

row1 = [10, 20, 15, 10, 5]
row2 = [12, 24, 18, 8, 7]

dist = cityblock(row1, row2)
print(dist)
```

```
13
```

4) Minkowski Distance

Minkowski distance calculates the distance between two real-valued vectors.

It is a generalization of the Euclidean and Manhattan distance measures and adds a parameter, called the “order” or “p“, that allows different distance measures to be calculated.

```
from math import sqrt

def minkowski_distance(a, b, p):
  return sum(abs(e1-e2)**p for e1, e2 in zip(a,b))**(1/p)

row1 = [10, 20, 15, 10, 5]
row2 = [12, 24, 18, 8, 7]

dist = minkowski_distance(row1, row2, 1)
```

```
print(dist)

dist = minkowski_distance(row1, row2, 2)
print(dist)
```

```
13.0
6.082762530298219
```

```
from scipy.spatial import minkowski_distance

row1 = [10, 20, 15, 10, 5]
row2 = [12, 24, 18, 8, 7]

dist = minkowski_distance(row1, row2, 1)
print(dist)

dist = minkowski_distance(row1, row2, 2)
print(dist)
```

```
13.0
6.082762530298219
```

✓ 0s completed at 8:56 AM