Assignment

In this assignment you will implement your own k-NN classifier for your labels. This classifier will be implemented as a class

```
Class Custom_knn()
    def __init__(self, number_neighbots_k, distance_parameter_p):
    def __str__(self)
    def fit(self, X, Labels):
    def predict(self, new_x):
    def draw_decision_boundary(self, new_x):
```

Your classifier will generalize the standard k-NN classifier as it will allow different distance metrics and will display the decision boundary. As before, k is the number of neighbors and $p \geq 1$ is the parameter in the Minkowski distance metric. Recall that Minkowski p-norm (distance) $d_p(A, B)$ from point $A = (x_1, y_1)$ to point $B = (x_2, y_2)$ is defined as follows:

$$d_p(A, B) = (|x_1 - x_2|^p + |y_1 - y_2|^p)^{1/p}$$

For $p = 2$ this gives Euclidean distance:

$$d_2(A, B) = (|x_1 - x_2|^2 + |y_1 - y_2|^2)^{1/2}$$

whereas for $p = 1$ this gives the "Manhattan" (street) distance:

$$d_1(A, B) = |x_1 - x_2| + |y_1 - y_2|$$

Finally, recall that in Numpy you compute this distance as follows:

```
import numpy as np
#assume A and B are your points (numpy arrays)
distance = np.linarg.norm(A-B, ord = p)
```

The method *predict()* gives you the label and the method *draw_decision_boundary(new_x)* will show the k neighbors (with their ids and colors) that were used to make a prediction for new_x. As before, your objects are weeks and your feature set is (μ, σ) for that week. Use your labels (you will have 52 labels per year for each week) from year 1 to train your classifier and predict labels for year 2.

Use the value of k that gave you highest accuracy when you used the standard kNN classifier from sklearn library (Euclidean distance) for each of the questions below:

Questions:

1. take three distance metrics: Euclidean ($p = 2$), Manhattan ($p = 1$) and generalized Minkovski for $p = 1.5$. For each value of p, compute the accuracy of your k-NN classifier on year 1 data. On x axis you plot p and on y-axis you plot accuracy. Which distance metric gives you the highest accuracy?

2. repeat this for year 2 and plot your results. Are there any differences with year 1?

3. take $p = 1.5$. In year 2, pick two weeks for which your classifier gave different labels. Use method display_decision_boundary() to show the neighbors (both colors and ids)

4. compute the confusion matrices for $p = 1$, $p = 1.5$ and $p = 2$

5. what are true positive rate (sensitivity or recall) and true negative rate (specificity) for year 2 for each p? Are there any differences for different distance methods?

6. for $p = 1$, $p = 1.5$ and $p = 2$ implement a trading strategy based on your labels for year 2 and compare the performance with the "buy-and-hold" strategy. For which value of p does your strategy result in the largest portfolio value at the end of the year?