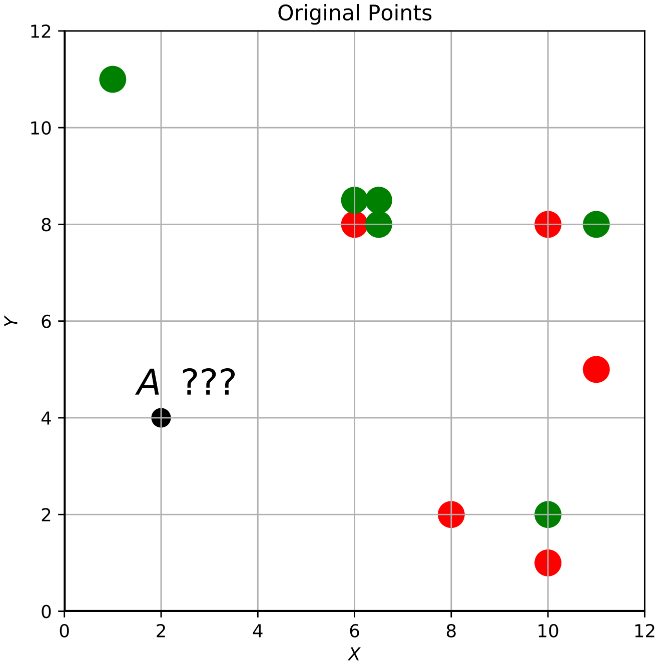
Assignment

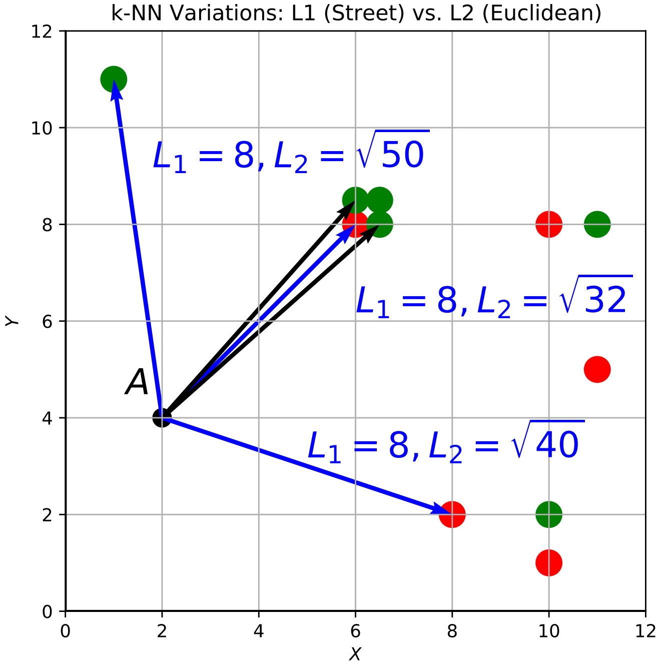
In this assignment, we will consider a number of variations of k-NN. We assume that we have N points $P_1, P_2, \ldots, P_N$ with "green" or "red" labels (classes) in training set X_{train}. For simplicity, we will confine ourselves to 2-dimensional space (just like your trading labels) and discuss the proposed variations to k-NN using geometric intuition. We will think of features as coordinates - a point P_i has coordinates (x_i, y_i). Suppose we want to classify a point A with coordinates (a_1, a_2).
We will consider the following methods (method names are "unofficial") to assign a label to this point A.



1. **k-NN with Manhattan Distance**. In sklearn, the default metric is Euclidean correponding to Minkowski distance with p=2. Recall that for any two points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ and parameter $p > 0$, the p-Minkowski distance $|P_1, P_2|_p$ (or L_p norm) is defined as

$$|P_1, P_2|_p = (|x_2 - x_1|^p + |y_2 - y_1|^p)^{1/p}$$

If $p = 2$ then we have the Euclidean (L_2-norm).

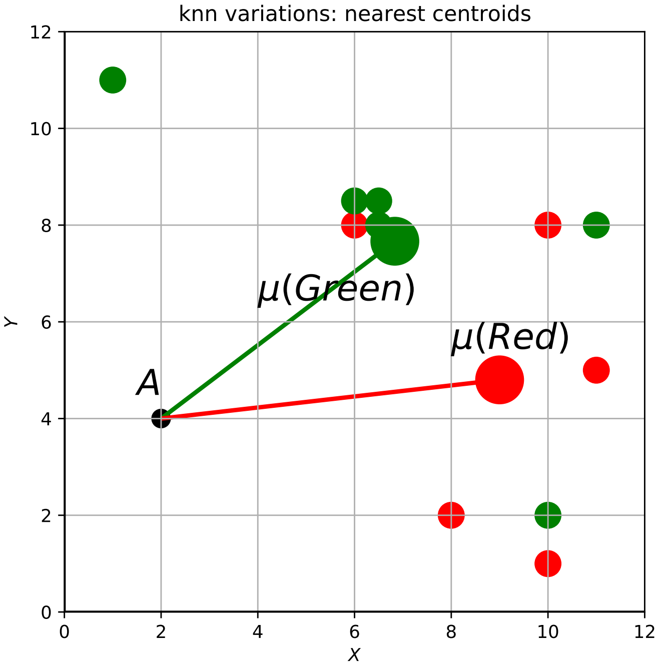


If $p = 1$ we have Manhattan (street or L_1-norm). The parameter p is one of the paremeters that can be specified (just like the number of neighbors k).

2. **k-NN with Minkowski** $p = 1.5$. Intuitively, this is between Manhattan and Euclidean.

3. **Nearest Centroid:** For each class, compute the corresponding "mean" points (i.e. "centers of gravity" or centroids) in the training set X_{train}.

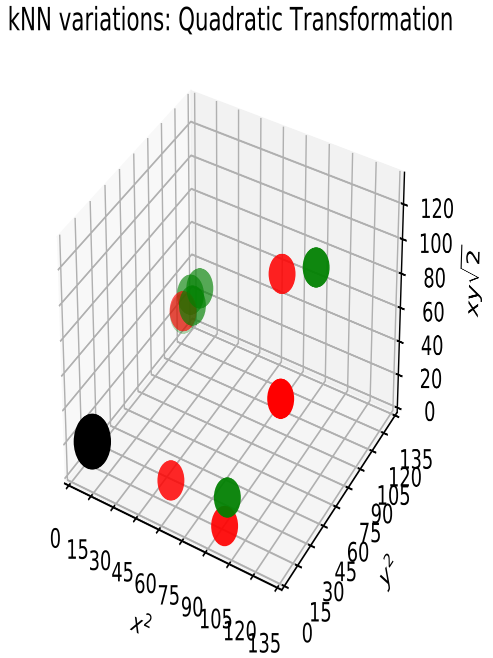
Let $\mu(X_{train}^{green})$ and $\mu(X_{train}^{red})$ be the centroids for each class. For any point A, assign the label of the nearest centroid.



4. **Domain Transformation:** We map 2-dimensional representation of our points into 3-dimensional space using the following quadratic transformation:

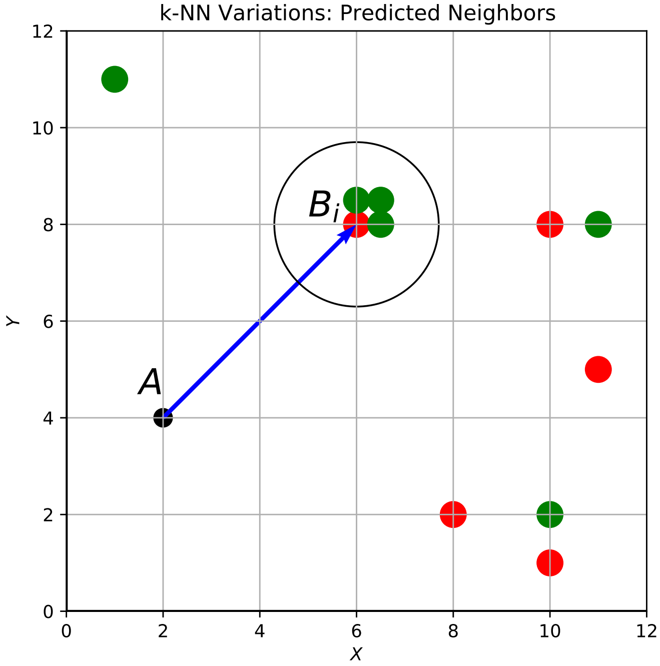
$$(x_i, y_i) \mapsto (x_i^2, x_i y_i \sqrt{2}, y_i^2)$$

and apply k-NN in the new 3-dimensional space.



kNN variations: Quadratic Transformation

5. ***k*-Predicted Neighbors:**. Find the nearest k neighbors $B_1, \ldots, B_k$ (from training set) for point A. For each such neighbor B_i, ignore its true label and compute predicted label based on its k neighbors from X_{train}. Compute the label for A using the majority of predicted labels for $B_1, \ldots, B_k$ (as opposed to the majority of true



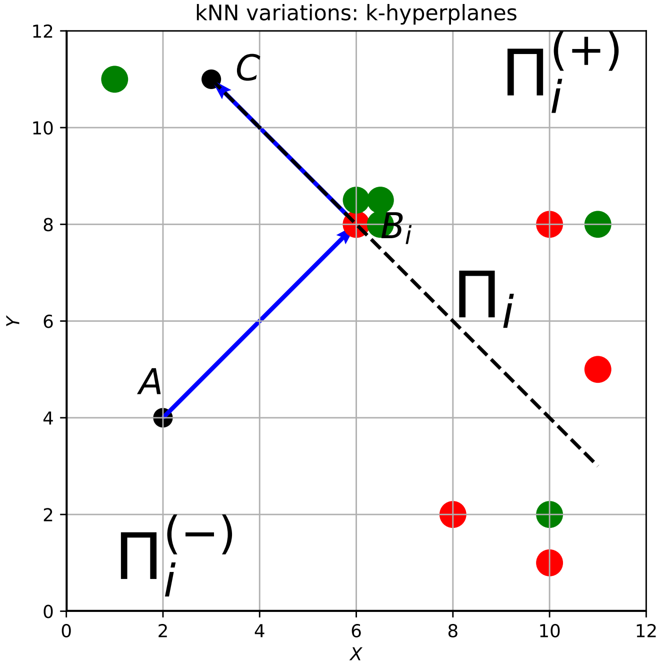
labels for $B_1, \ldots, B_k$ as in standard k-NN).

6. ***k*-Hyperplanes:** Find the nearest k neighbors $B_1, \ldots, B_k$ (from training set) for point A. For each such neighbor B_i let its coordinates be $B_i = (b_1^{(i)}, b_2^{(i)})$. Consider the vector from A to B_i given by $\overrightarrow{AB_i} = (b_1^{(i)} - a_1, b_2^{(i)} - a_2)$. We construct a hyperplane Π_i perpendicular to $\overrightarrow{AB_i}$ and passing through point B_i. To that end, for any point

$C = (x, y) \in \Pi_i$, vector $\overrightarrow{B_iC} = (x - b_1^{(i)}, y - b_2^{(i)})$ is perpendicular to $\overrightarrow{AB_i}$. Therefore, all points $(x, y) \in \Pi_i$ satisfy the equation $\overrightarrow{AB_i} \cdot \overrightarrow{B_iC} = 0$. In other words,

$$(b_1^{(i)} - a_1)(x - b_1^{(i)}) + (b_2^{(i)} - a_2)(y - b_2^{(i)}) = 0$$

The hyperplane Π_i divides the space into 2 sub-



spaces Π_i^- and Π_i^+. Let us call them "negative" and ”positive” subspaces. Each point is in either ”negative” subspace Π_i^-, ”positive” subspace Π_i^+, or on the hyperplane Π_i itself. For any point $D = (x^*, y^*)$ (not necessarily on the hyperplane Π), either $\overrightarrow{AB_i} \cdot \overrightarrow{B_iD} > 0$ and this means $D \in \Pi_i^+$, or $\overrightarrow{AB_i} \cdot \overrightarrow{B_iD} < 0$ and this means $D \in \Pi_i^+$. If $\overrightarrow{AB_i} \cdot \overrightarrow{B_iD} = 0$, we assign it to Π_i^+ for simplicity. Consider the point $D = A$ itself. For this point,

$$\begin{aligned}\overrightarrow{AB_i} \cdot \overrightarrow{B_iA} &= (b_1^{(i)} - a_1)(a_1 - b_1^{(i)}) + (b_2^{(i)} - a_2)(a_2 - b_2^{(i)}) \\ &= -\left[(b_1^{(i)} - a_1)^2 + (b_2^{(i)} - a_2)^2\right] < 0\end{aligned}$$

and therefore, $A \in \Pi_i^-$. From our training set X_{train} we compute the number of labels for all points in Π_i^- and assign to A the label of the majority of these labels. We perform this procedure for all k nearest neighbors of A, namely $B_1, \ldots, B_k$. With each neighbor, we get a label for A. The final label for A is the majority of these labels.

QUESTIONS

Question 1 (Manhattan distance $p = 1$)

1. take $k = 3, 5, 7, 9, 11$. For each value of k compute the accuracy of this classifier. On x axis you plot k and on y-axis you plot accuracy. What is the optimal value of k for year 1?
2. use the optimal value of k from year 1 to predict labels for year 2. What is your accuracy?
3. using the optimal value for k from year 1, compute the confusion matrix for year 2
4. is this value k different than the one you obtained using regular k-NN
5. what is true positive rate (sensitivity or recall) and true negative rate (specificity) for year 2?
6. implement a trading strategy based on your labels for year 2 and compare the performance with the "buy-and-hold" strategy. Which strategy results in a larger amount at the end of the year?

7. how does this method compare with regular k-NN with Euclidean? Any improvement?

Question 2 (Minkowski distance $p = 1.5$)

1. take $k = 3, 5, 7, 9, 11$. For each value of k compute the accuracy of this classifier. On x axis you plot k and on y-axis you plot accuracy. What is the optimal value of k for year 1?

2. use the optimal value of k from year 1 to predict labels for year 2. What is your accuracy?

3. using the optimal value for k from year 1, compute the confusion matrix for year 2

4. is this value k different than the one you obtained using regular k-NN

5. what is true positive rate (sensitivity or recall) and true negative rate (specificity) for year 2?

6. implement a trading strategy based on your labels for year 2 and compare the performance with the "buy-and-hold" strategy. Which strategy results in a larger amount at the end of the year?

7. how does this method compare with regular k-NN with Euclidean distance? Any improvement?

Note: For questions 3-7 below, use the Euclidean distance!

Question 3 (Nearest Centroid)

1. for each label, compute the average and median distance to the "green" and "red' centroids for the points in the training set. We can think of this distance as the average radius of the sphere centered at the centoids. Which sphere is larger (for both average and median distances)?

2. what is true positive rate (sensitivity or recall) and true negative rate (specificity) for year 2?

3. implement a trading strategy based on your labels for year 2 and compare the performance with the "buy-and-hold" strategy. Which strategy results in a larger amount at the end of the year?

4. how does this method compare with regular k-NN? Any improvement?

Question 4 (Domain Transformation)

1. take $k = 3, 5, 7, 9, 11$. For each value of k compute the accuracy of this classifier. On x axis you plot k and on y-axis you plot accuracy. What is the optimal value of k for year 1?

2. use the optimal value of k from year 1 to predict labels for year 2. What is your accuracy?

3. using the optimal value for k from year 1, compute the confusion matrix for year 2

4. is this value k different than the one you obtained using regular k-NN

5. what is true positive rate (sensitivity or recall) and true negative rate (specificity) for year 2?

6. implement a trading strategy based on your labels for year 2 and compare the performance with the "buy-and-hold" strategy. Which strategy results in a larger amount at the end of the year?

7. how does this method compare with regular k-NN? Any improvement?

Question 5 (k-Predicted Neighbors)

1. take $k = 3, 5, 7, 9, 11$. For each value of k compute the accuracy of this classifier. On x axis you plot k and on y-axis you plot accuracy. What is the optimal value of k for year 1?

2. use the optimal value of k from year 1 to predict labels for year 2. What is your accuracy?

3. using the optimal value for k from year 1, compute the confusion matrix for year 2

4. is this value k different than the one you obtained using regular k-NN

5. what is true positive rate (sensitivity or recall) and true negative rate (specificity) for year 2?

6. implement a trading strategy based on your labels for year 2 and compare the performance with the "buy-and-hold" strategy. Which strategy results in a larger amount at the end of the year?

7. how does this method compare with regular k-NN? Any improvement?

Question 6 (k-Hyperplanes)

1. take $k = 3, 5, 7, 9, 11$. For each value of k compute the accuracy of this classifier. On x axis you plot k and on y-axis you plot accuracy. What is the optimal value of k for year 1?
2. what is true positive rate (sensitivity or recall) and true negative rate (specificity) for year 2?
3. implement a trading strategy based on your labels for year 2 and compare the performance with the "buy-and-hold" strategy. Which strategy results in a larger amount at the end of the year?
4. how does this method compare with regular k-NN? Any improvement?

Question 7.

Summarize the results for regular k-NN and the its variations in the table below. Round Accuracy and Amount to integers. Color the largest values in Accuracy and Amount columns by green color and the lowest values by red. Discuss your findings.

	Method	**Best k**	**% Accuracy**	Amount
1.	Buy-and-Hold	N/A	N/A	
2.	k-NN (Euclidean, $p = 2$)			
3.	k-NN (Manhattan, $p = 1$)			
4.	k-NN (Minkowski, $p = 1.5$)			
5.	Nearest Centroid	N/A		
6.	Domain Transformation			
7.	k-Predicted Neighbors			
8.	k-Hyperplanes			