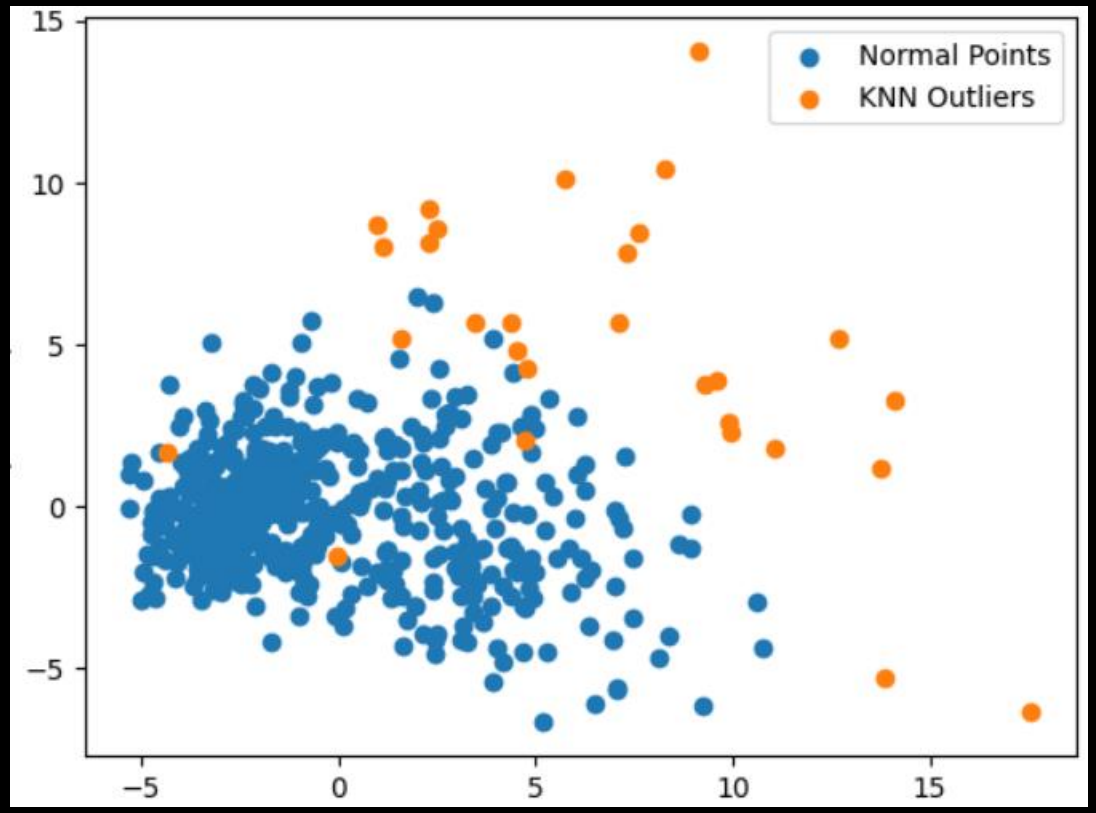
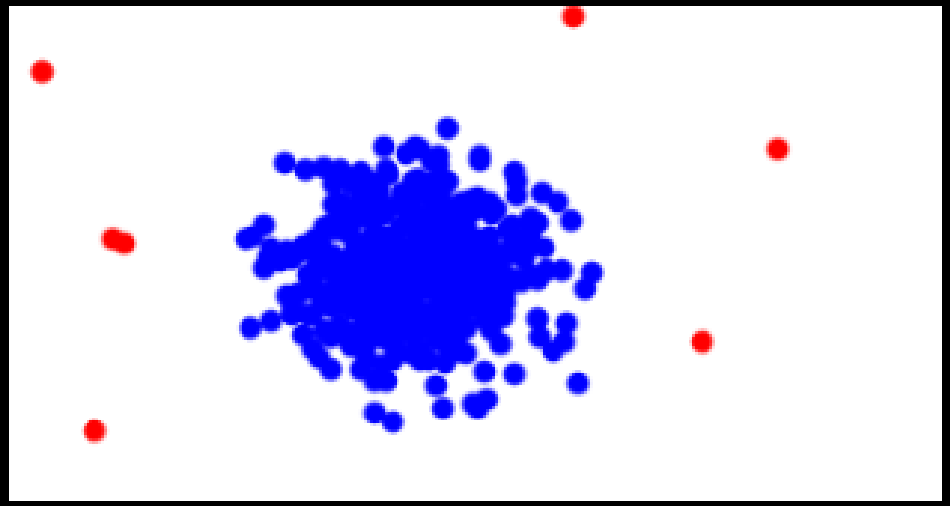
K-Nearest Neighbors (KNN)

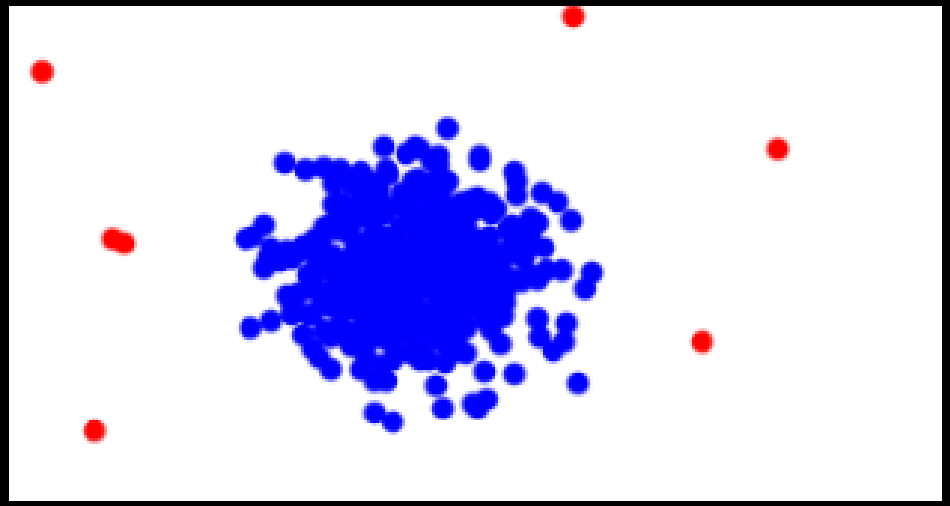
KNN identifies outliers as data points whose K nearest neighbors are far away from them.





Each row below is a data point

Age	Income	Experience	Satisfaction
23	45000	3	2
53	35000	8	4
65	48000	1	1
27	92000	8	8
54	55000	10	2
…	…	…	…



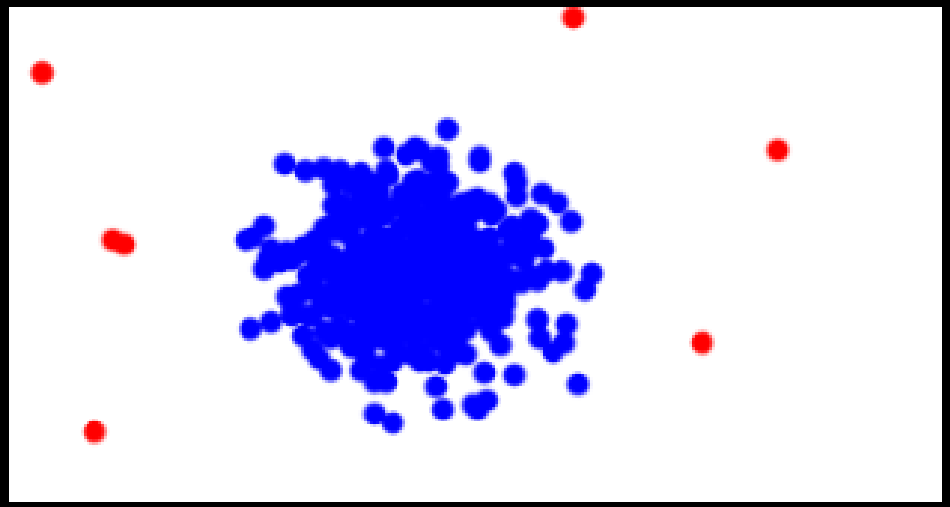
A. Distance-based outlier detection

Idea: Compute the distance to the k-th nearest neighbor for every data point.
If this distance is large compared to others → likely an outlier.

For example, if k=5 then
1) Compute distance to 5^{th} nearest neighbor for all data points
Example: P1 -> 7, P2 -> 4, P3-> 11, P4 -> 6

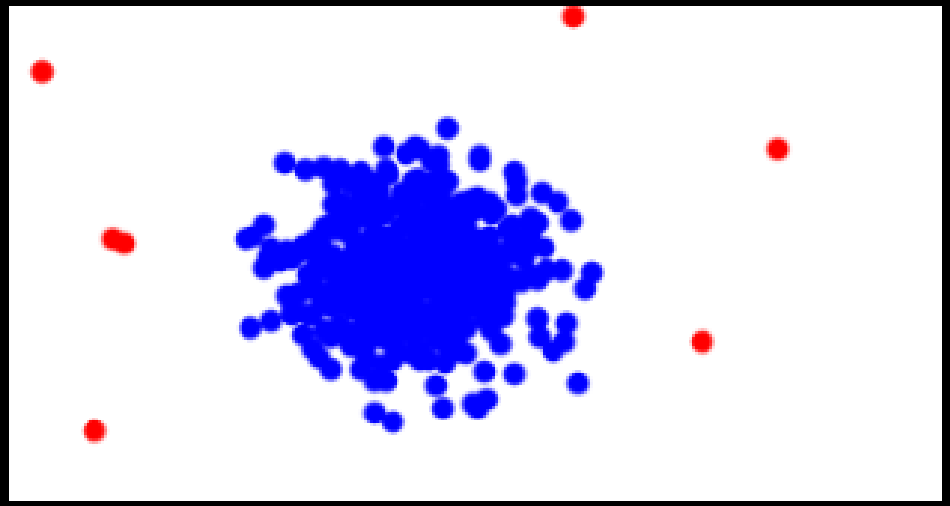
2) Sort this distance for all data points:
Example: P2, P4, P1, P3

3) Pick top 5%(or 1%). These points would be outliers
Say, P3



This distance is also called an **outlier score.**

Outlier score = Distance to k-th nearest neighbor



B. Average distance to k neighbors

Compute the **mean (or sum) of distances** to the k nearest neighbors for **every** data point.

Points with the **highest average distances** are candidates for being outliers.

This distance is also called **outlier score** $= \frac{1}{k}\sum_{neighbors} distance$

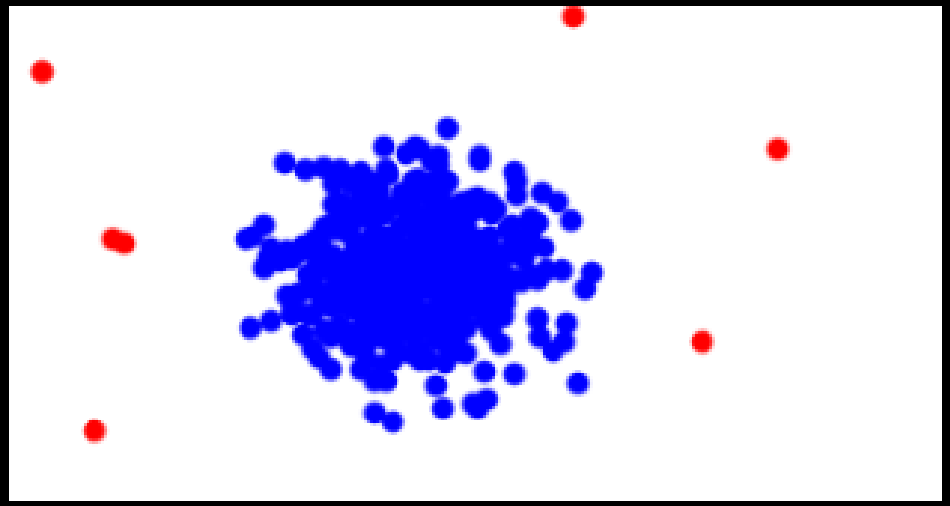
Example with k=3:

Point1 : average distance to 3 NN = 12

Point2 : average distance to 3 NN = 20 (outlier candidate)

Point3 : average distance to 3 NN = 10

Point4 : average distance to 3 NN = 11



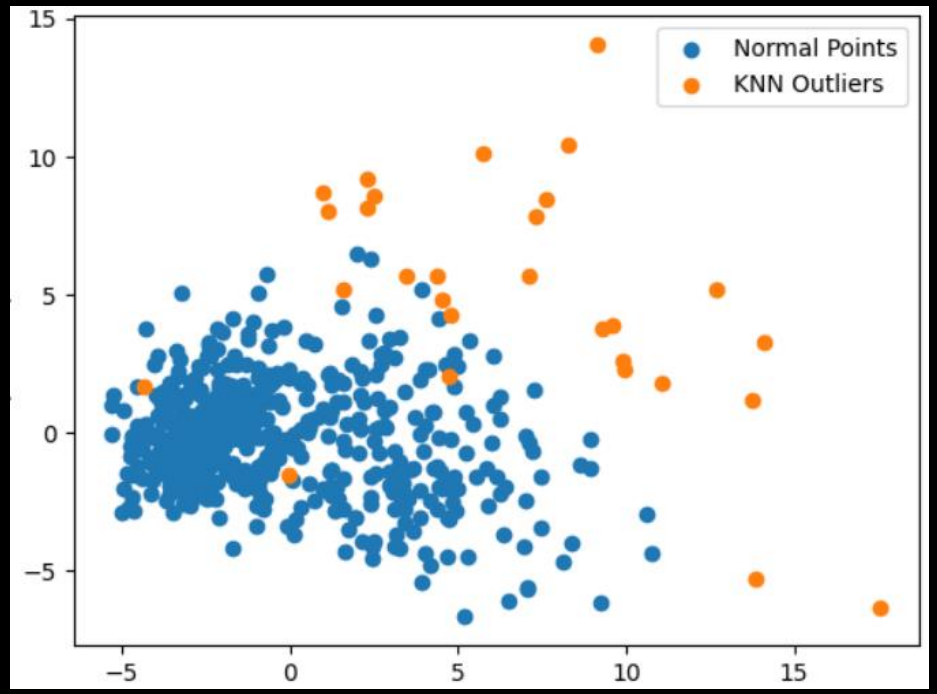
Limitations of using KNN to identify outliers

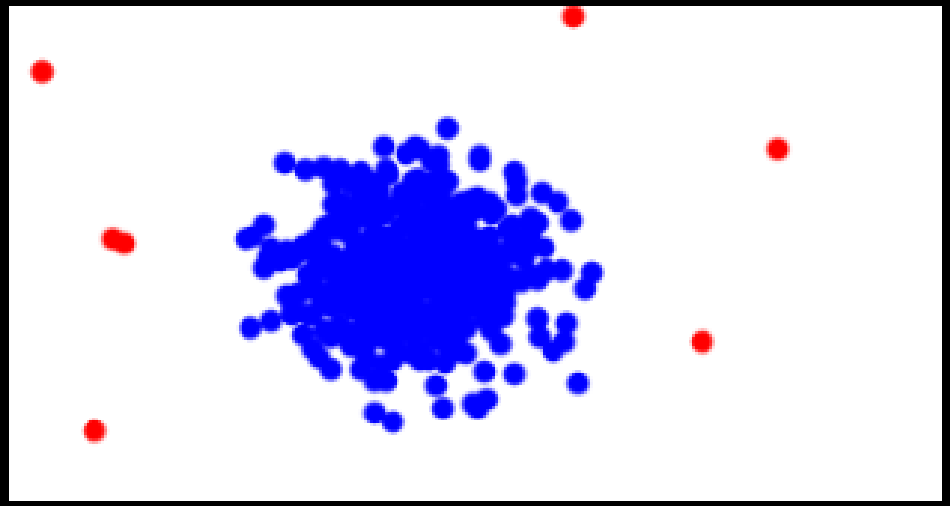
- Computationally expensive for large datasets (unless optimized):

It has to compute k-NN for all data points

- Sensitive to choice of k

Changing k = 5 to 6 would change the outlier points





STOP