K Nearest Neighbors

Data Boot Camp

Lesson 21.2

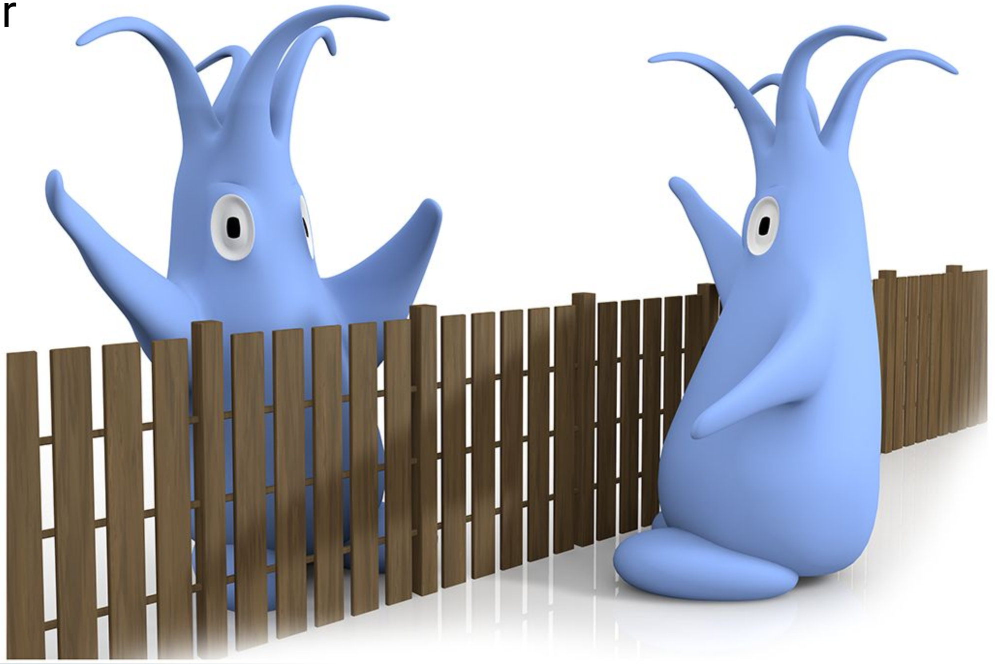
K Nearest Neighbors

K Nearest Neighbors Algorithm

K Nearest Neighbors (KNN) is a simple and robust algorithm for classification (and sometimes regression).

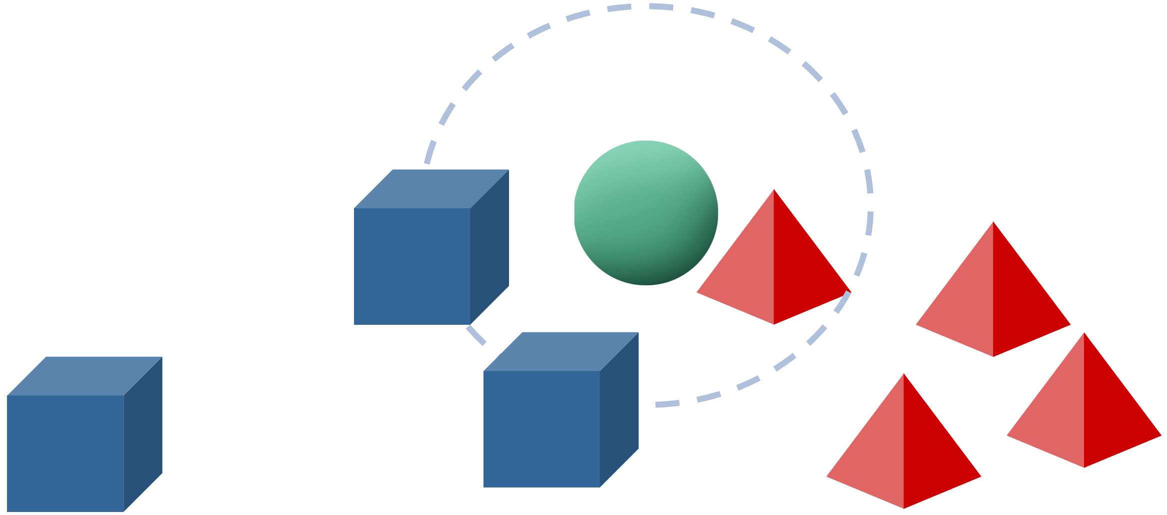
It has many benefits such as outlier insensitivity, ability to classify non-linear data, and high accuracy.

It does require a lot of memory.



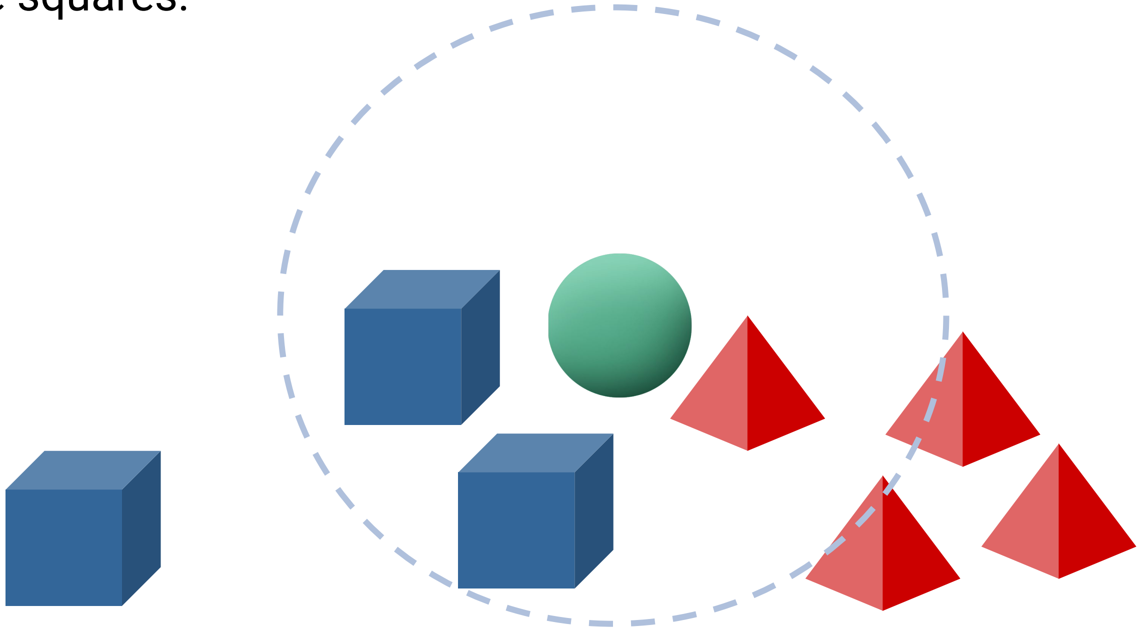
K = 1

When k = 1, this is simply the nearest neighbor. You find the point nearest to your new data point (the green circle) and that is the class that it will belong to. In this case, the closest neighbor is a red triangle, so the data point will belong to red.



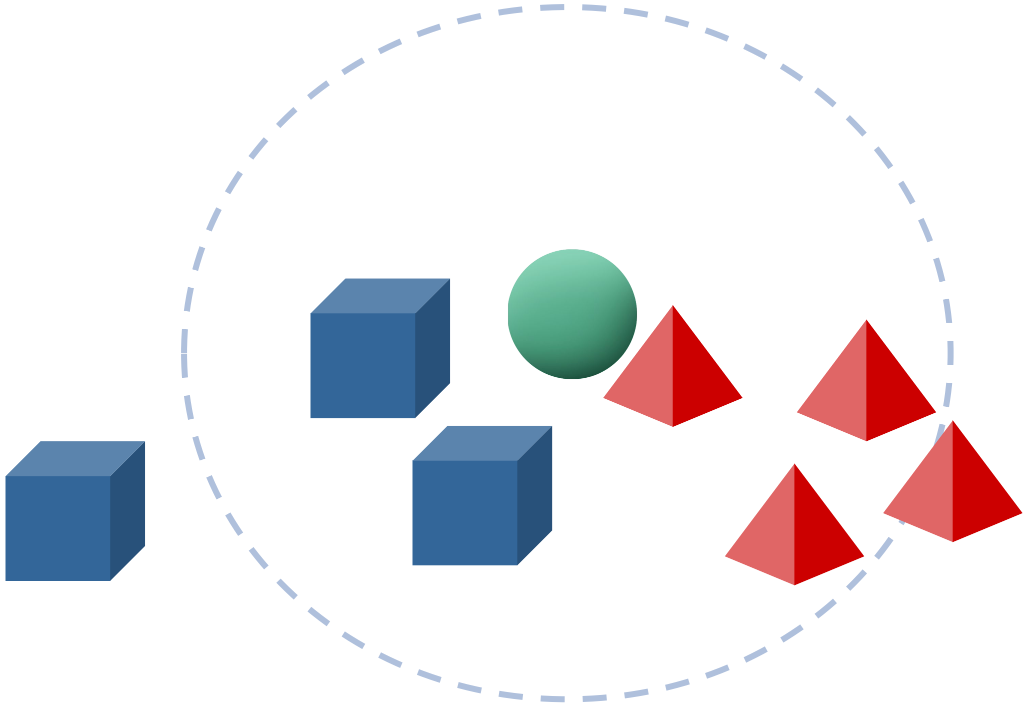
K = 3

When k = 3, we find the three closest neighbors. In this case, there are two blue squares and one red triangle, so the new data point will be grouped with the blue squares.



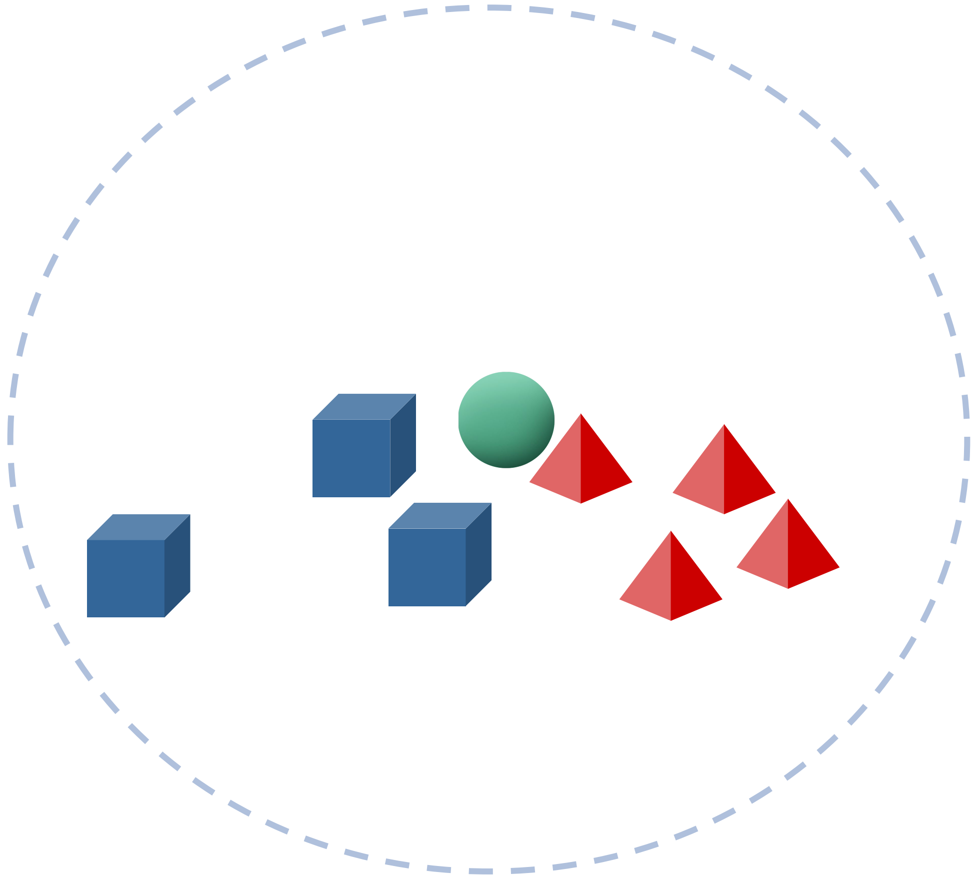
K = 5

When k = 5, there are three red triangles and two blue squares, so the new data point will belong to the red triangles.



K = 7

Finally, when k = 7, the majority are red triangles, so the new data point belongs to red.



Choosing K

Because k can vary your results, the easiest technique for choosing a k value is to loop through a range of k and calculate the score. Choose the lowest value of k where the score starts to stabilize. **Note**: We only use odd numbers so there are no ties between classes.

```
for k in range(1, 20, 2):
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(data, labels)
    score = knn.score(data, labels)
    print(f”K: {k}, Score: {score}”)
```

Questions?