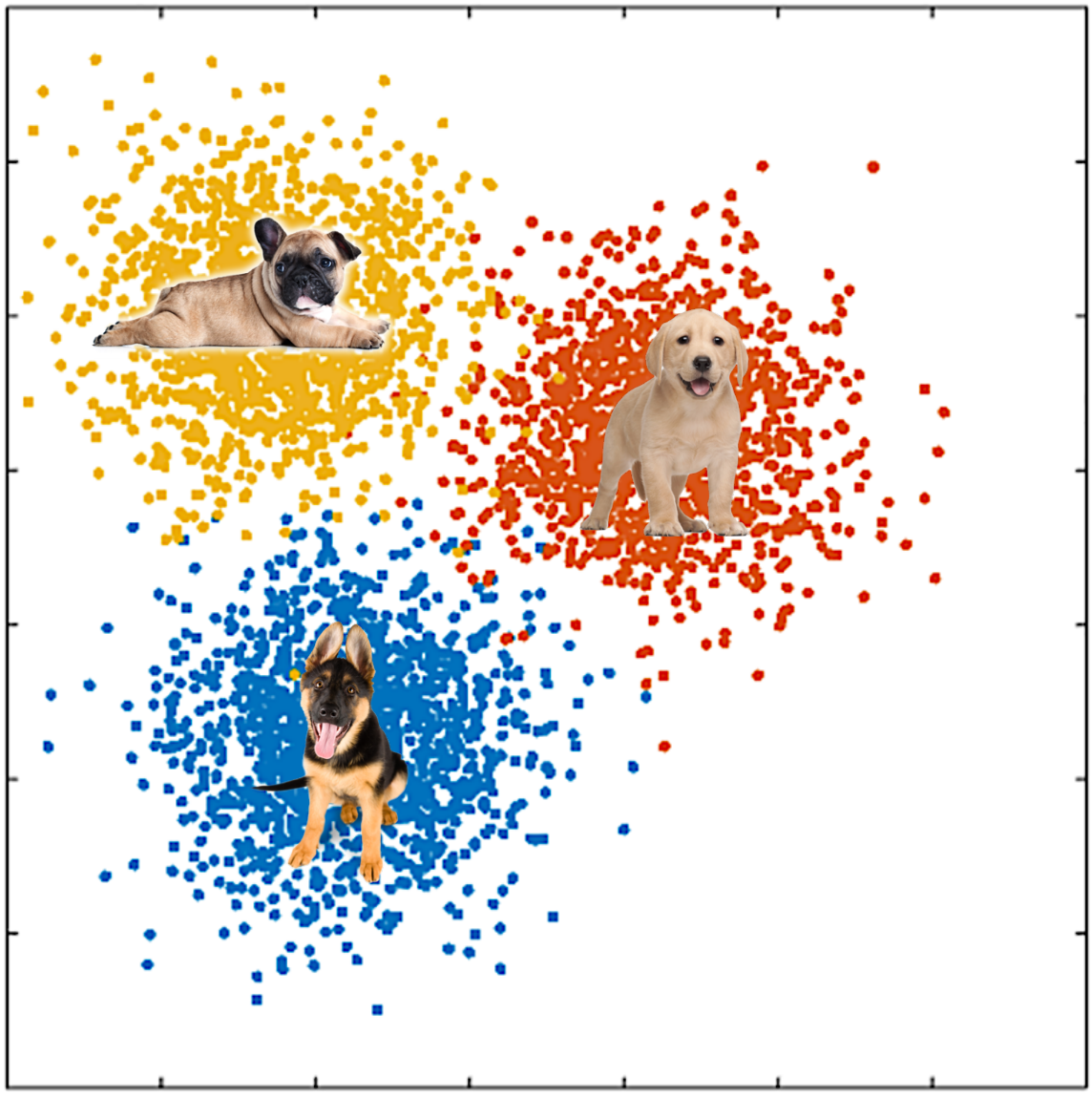
K-mean clustering

k-means clustering is an unsupervised machine learning algorithm. You'll use it to create clusters based on what your machine sees in the data. Think of it this way. Imagine we're at the animal shelter in Edinburgh. The shelter had a large social room where all the dogs get together and play. The dogs acted like people. They had their group of friends and they chatted and hung out with each other. Each time they had a social hour, they would self-organize into different groups of friends. Now imagine that the shelter was closing and all the dogs were going to be distributed into three different shelters across the city.



The organizers of the animal shelter got together and decided to make it easier on the dogs, they would cluster them based on these groups of friends. So, the shelter decided to create three clusters. That means that the k in k-means equal to three because you want to divide the groups into three clusters. Now imagine that the machine learning

algorithm got started. To start, the machine put a red, yellow, and blue colour on three random dogs. Each colour represented a potential cluster based on their social group. These will be your three centroid dogs. Now each of the centroid dogs would look at the mean distance between itself and all of the surrounding dogs. Then the machine would put the same colour collar on the dogs that were closest to these centroid dogs. As you can imagine since these centroid dogs were selected randomly, there's a pretty good chance that you won't really have any good clusters. Maybe all three centroid dogs were in the same social group. If that happens then most of the dogs would have a very large distance between these three centroids. So, the machine will try over and over again until it picks the best centroid dog. It might even do this one cluster at a time. At the end of each iteration, the machine learning algorithm checks the variance between each dog and the centroid. Once you have a good centroid dog then it's pretty straightforward to put unknown dogs into each cluster. If you put a new dog into the social area, then you can tell which social group it ends up in by just measuring the distance from the centroid dog. Also keep in mind that the dogs themselves did not cluster into three groups. There might be five or six different social groups, but there are only three shelters, so the machine learning algorithm has to do its best to create clusters that best represent the dogs' social grouping. You should also watch to make sure that you use k-means clustering if the dogs are predisposed to these social groups. If the dogs are jumping from group to group, then it'll be difficult to form real clusters. This is sometimes called a high overlap of data. Another challenge with k-means is it can be very sensitive to outliers. So, if you have a dog that's not really interested in hanging out with any of the other dogs, it will still be clustered into one of these three groups. So in a sense, the dog will be forced to find friends. Organizing dogs into three clusters for three different shelters is probably not a problem that you'll run into everyday, but k-means clustering is actually one of the most popular machine learning algorithms. One of the more interesting applications is when retailers use clustering to decide who gets promotions. They might create three clusters that they call loyal customers, somewhat loyal customers, and lowest priced shoppers. Then they'll create strategies to try and elevate somewhat loyal customers to loyal customers or they could just invite their loyal customers to participate in one of their programs. Many organizations are looking for better ways to cluster together their customers. If you can get all your loyal customers into one cluster then that can really improve your business.

k-means clustering is an instance-based or lazy learning. That means that you get all the answers in one big splash. If something changes in your data then you have to rerun the algorithm from scratch. This might make it difficult to scale these models because you're working with much more data at any one time.