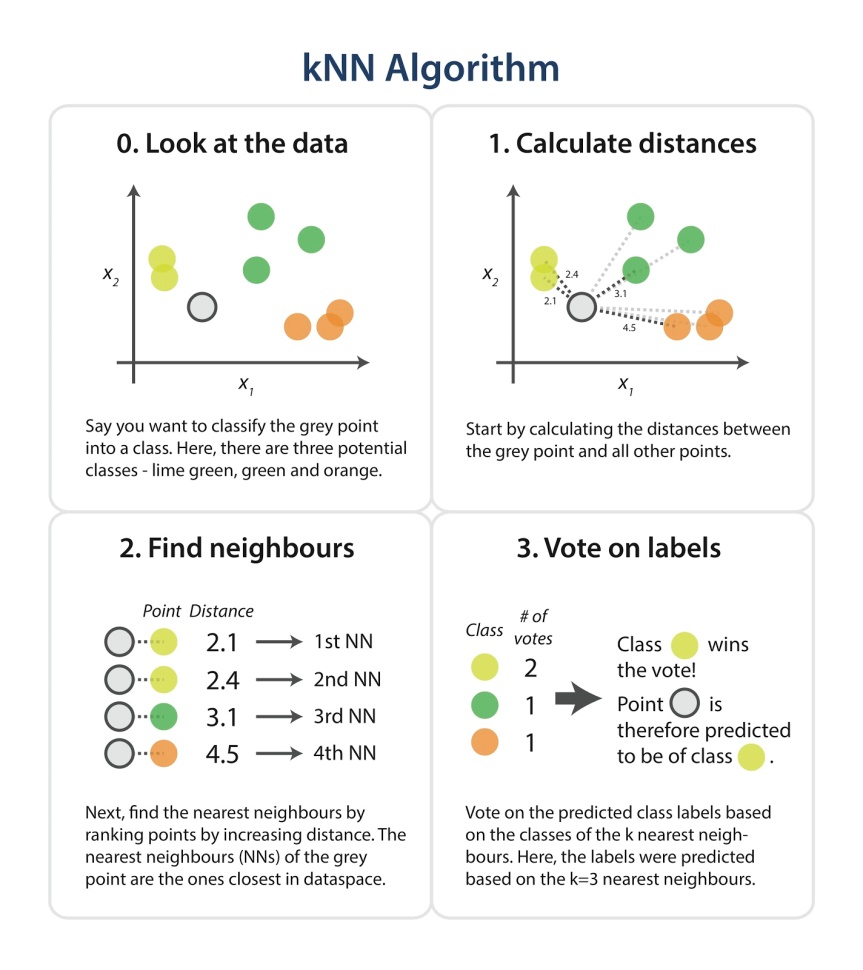
**K-Nearest Neighbour**

Supervised machine learning algorithm used to solve both classification (binary responses e.g. admitted or not) and regression problems (e.g. estimate someone’s weight given their height).

“KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).”

KNN captures the idea of similarity (sometimes called distance, proximity, or closeness). There are multiple ways of calculating distance in KNN. The straight-line distance (also called the Euclidean distance) is a popular choice.

[](https://www.google.com/url?sa=i&url=https://www.kdnuggets.com/2016/01/implementing-your-own-knn-using-python.html&psig=AOvVaw2lVcAwPO99E85z5ZSClFVM&ust=1584721702847000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCOjUn8P6pugCFQAAAAAdAAAAABAE)Choosing the right value for K

To select the K that’s right for your data, we run the KNN algorithm several times with different values of K and choose the K that reduces the number of errors we encounter while maintaining the algorithm’s ability to accurately make predictions when it’s given data it hasn’t seen before.

* As we increase K, predictions are more accurate but eventually accuracy does decline if K is increased further.
* In case of classification where mode is used, K is an odd number (to avoid ties).

Positives - The algorithm is versatile (can be used for classification, regression, and search), simple and easy to implement.

Limitation - Gets significantly slower as the number of examples and/or predictors/independent variables increase.

**Reference**:

https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

https://www.kdnuggets.com/2016/01/implementing-your-own-knn-using-python.html