GENERAL ASSEMBLY

K-NEAREST NEIGHBORS

Trevor Lindsay

LEARNING OBJECTIVES

- Build a K-Nearest Neighbors model using the scikit-learn library
- Understand the pros and cons of KNN

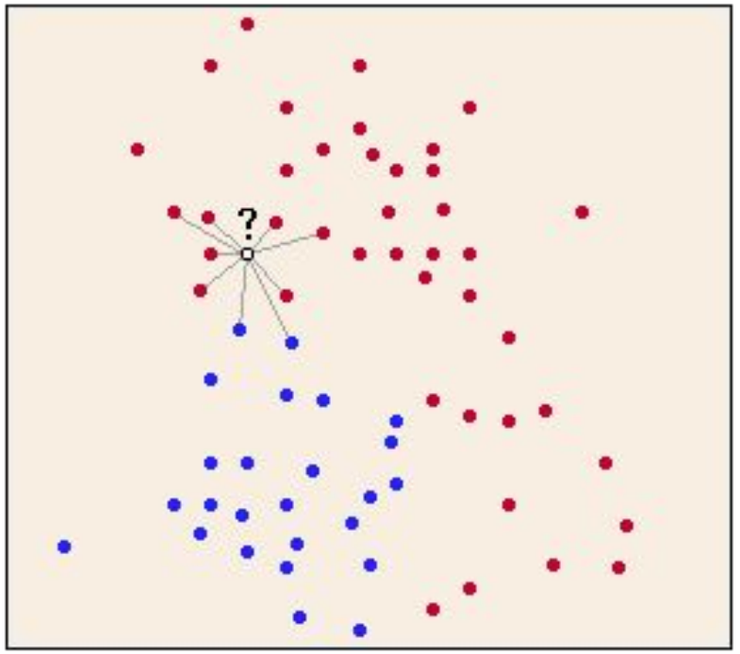
WHAT IS K-NEAREST NEIGHBORS?

WHAT IS K-NEAREST NEIGHBORS?

- K-Nearest Neighbors (KNN) is a classification algorithm that makes a prediction based upon the closest data points.
- The KNN algorithm:
  - For a given point, calculate the distance to all other points.
  - Given those distances, pick the *k* closest points.
  - Calculate the probability of each class label given those points.
  - The original point is classified as the class label with the largest probability (“votes”).

WHAT IS K-NEAREST NEIGHBORS?

- KNN uses distance to predict a class label. This application of distance is used as a measure of similarity between classifications.
- We're using shared traits to identify the most likely class label.



WHAT IS K-NEAREST NEIGHBORS?

- Suppose we want to determine your favorite type of music. How might we determine this without directly asking you?
- Generally, friends share similar traits and interests (e.g. music, sports teams, hobbies, etc). We could ask your five closest friends what their favorite type of music is and take the majority vote.
- This is the idea behind KNN: we look for things similar to (or close to) our new observation and identify shared traits. We can use this information to make an educated guess about a trait of our new observation.

DEMO

KNN IN ACTION

KNN IN ACTION

```
from sklearn import datasets, metrics
from sklearn.neighbors import KNeighborsClassifier
import pandas as pd
import numpy as np

X, y = datasets.make_classification(
    n_samples=1000,
    n_features=20,
    n_informative=5,
    weights=[0.3, 0.7],
)

# n_neighbors is our option in KNN
cls = KNeighborsClassifier(n_neighbors=5, weights='uniform')
cls.fit(X, y)

print cls.score(X, y)
```

WHAT HAPPENS IN HIGH DIMENSIONALITY?

- Since KNN works with distance, higher dimensionality of data (i.e. more features) requires significantly more samples in order to have the same predictive power.
- Consider this: with more dimensions, all points slowly start averaging out to be equally distant. This causes significant issues for KNN.
- Keep the feature space limited and KNN will do well. Exclude extraneous features when using KNN.

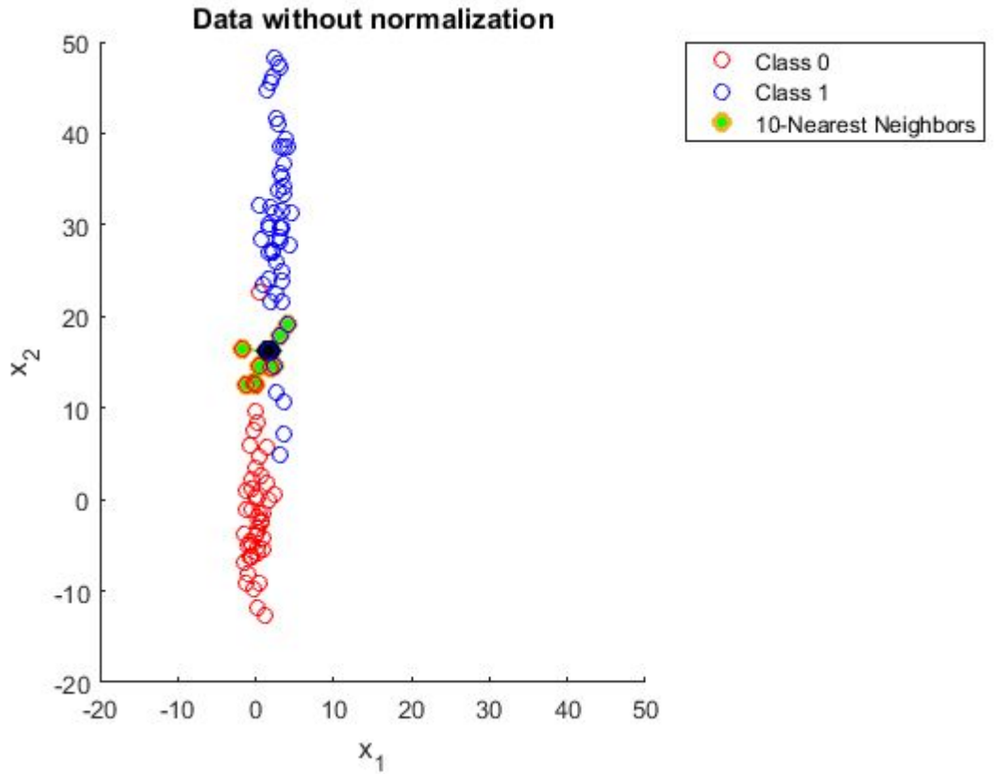
WHAT IS DISTANCE?

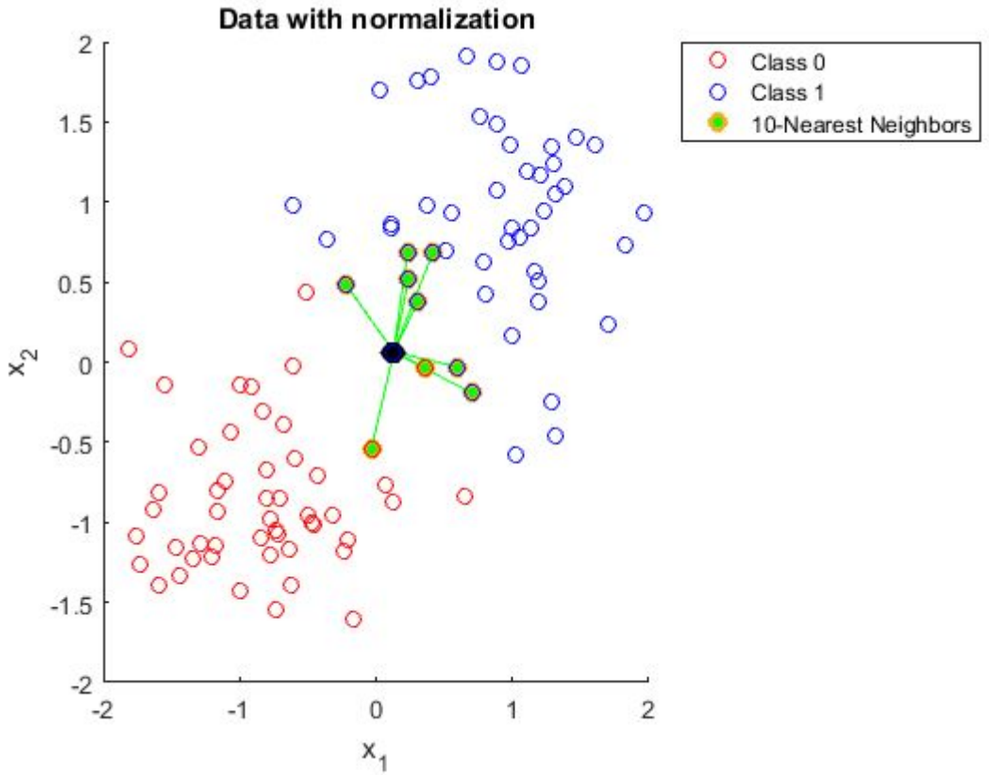
- Defines a quantitative measure of similarity
- Default sklearn KNN uses Euclidean distance
- However, depending on your data distribution and use case, there are other distance measures that maybe better suited
  - Mahanalobis
  - Cosine…
- Refer to http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.DistanceMetric.html

WHAT HAPPENS IN HIGH DIMENSIONALITY?

- Consider two different examples: classifying users of a newspaper and users of a particular toothpaste.
- The features of the newspapers are very broad and there are many: sections, topics, types of stories, writers, online vs print, etc.
- However, the features of a toothpaste are more narrow: has fluoride, controls tartar, etc.
- For which problem would KNN work better?

NORMALIZATION





PROS AND CONS

- Pros:
  - Simple to implement
  - Flexible to feature / distance choices
  - Naturally handles multi-class cases
  - Can do well in practice with enough representative data
- Cons:
  - Large search problem to find nearest neighbours
  - Must know we have a meaningful distance function
  - Requires feature scaling