Train In Data

All K Nearest Neighbours

All KNN

The idea is to remove samples from the majority class that are closest to the boundary.

All KNN

- Repeats ENN.
- Starts by exploring the 1 closest neighbour.
- Adds 1 neighbour to the KNN at each round.

All KNN

- Repeats ENN.
- Starts by exploring the 1 closest neighbour.
- Adds 1 neighbour to the KNN at each round.
- Stops after examining a user defined maximum number of neighbours.
- Or when the majority class becomes minority class.

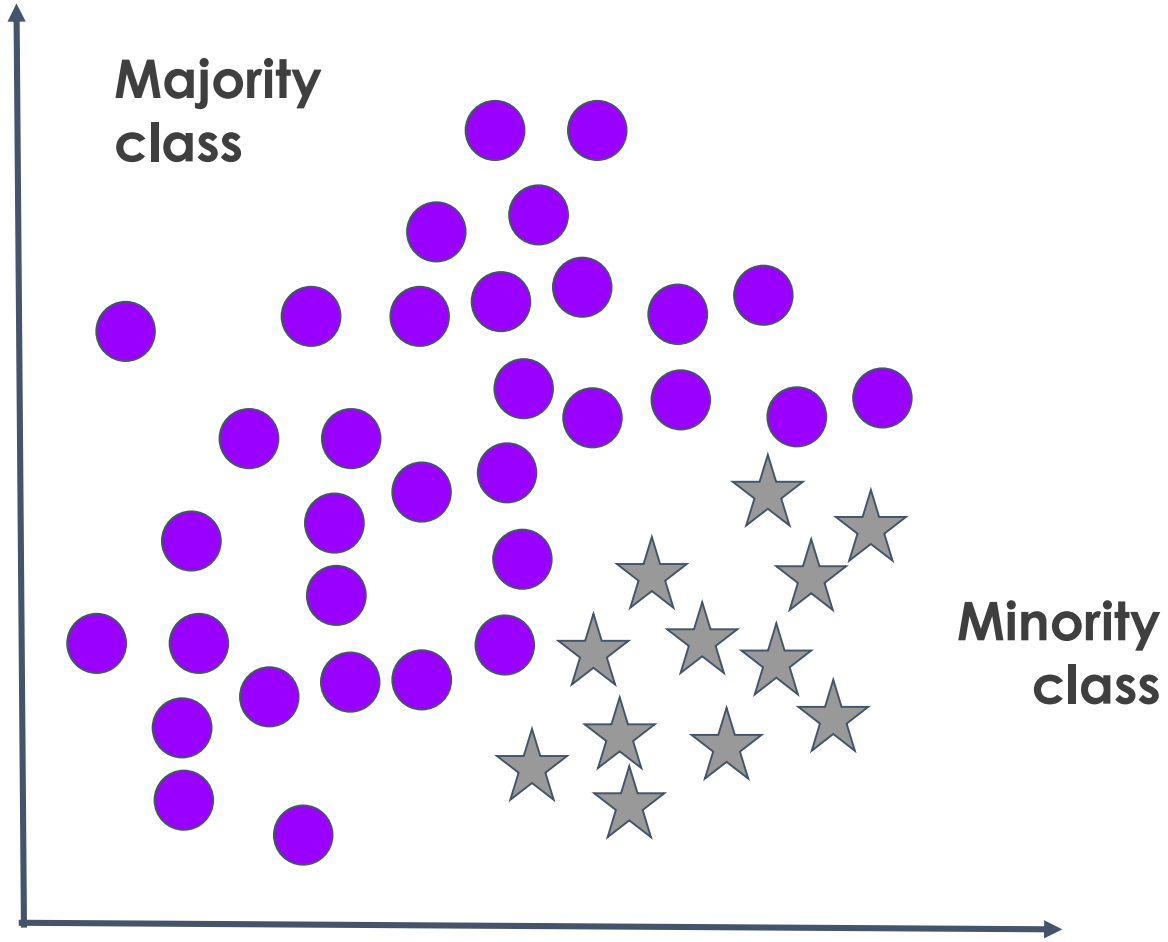
All KNN

1. Trains 1 KNN on entire dataset.
2. Finds each observation's 1 closest neighbour.
3. Decides whether to keep or remove, based on neighbours agreement with its class.
4. Repeats, but **adding 1 K** to the KNN, until:
    - A maximum number of neighbours is examined.
    - The majority class becomes minority.

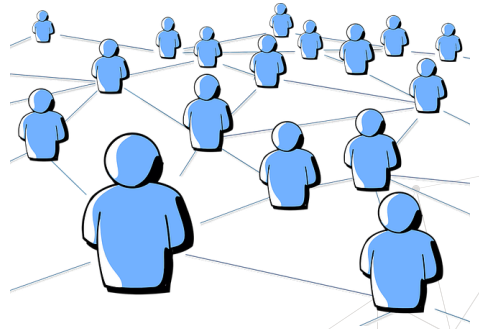
All KNN

- Final dataset shape varies
- Cleaning
- Removes hard cases

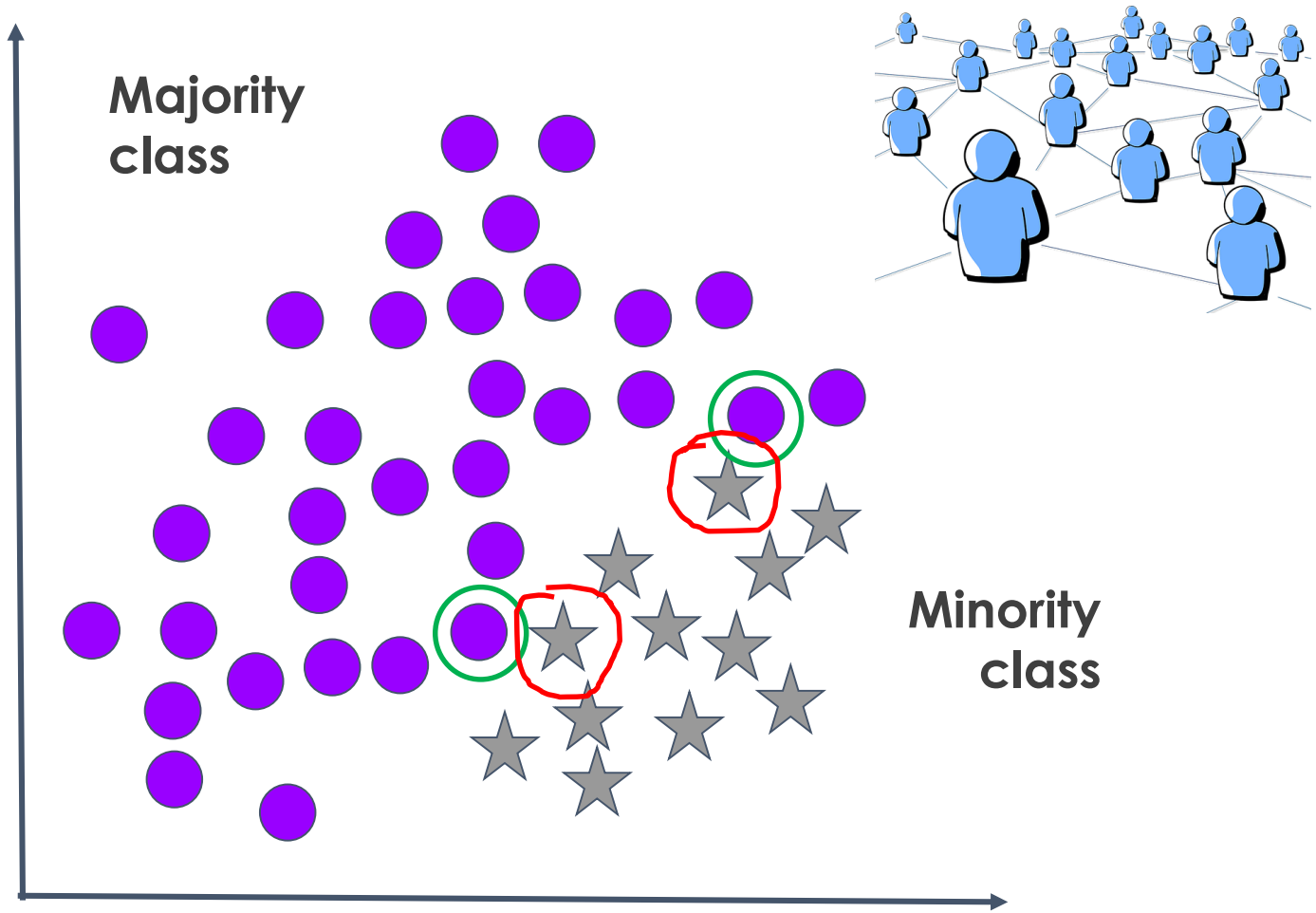
Edited Nearest Neighbours



Train a 1 KNN algorithm

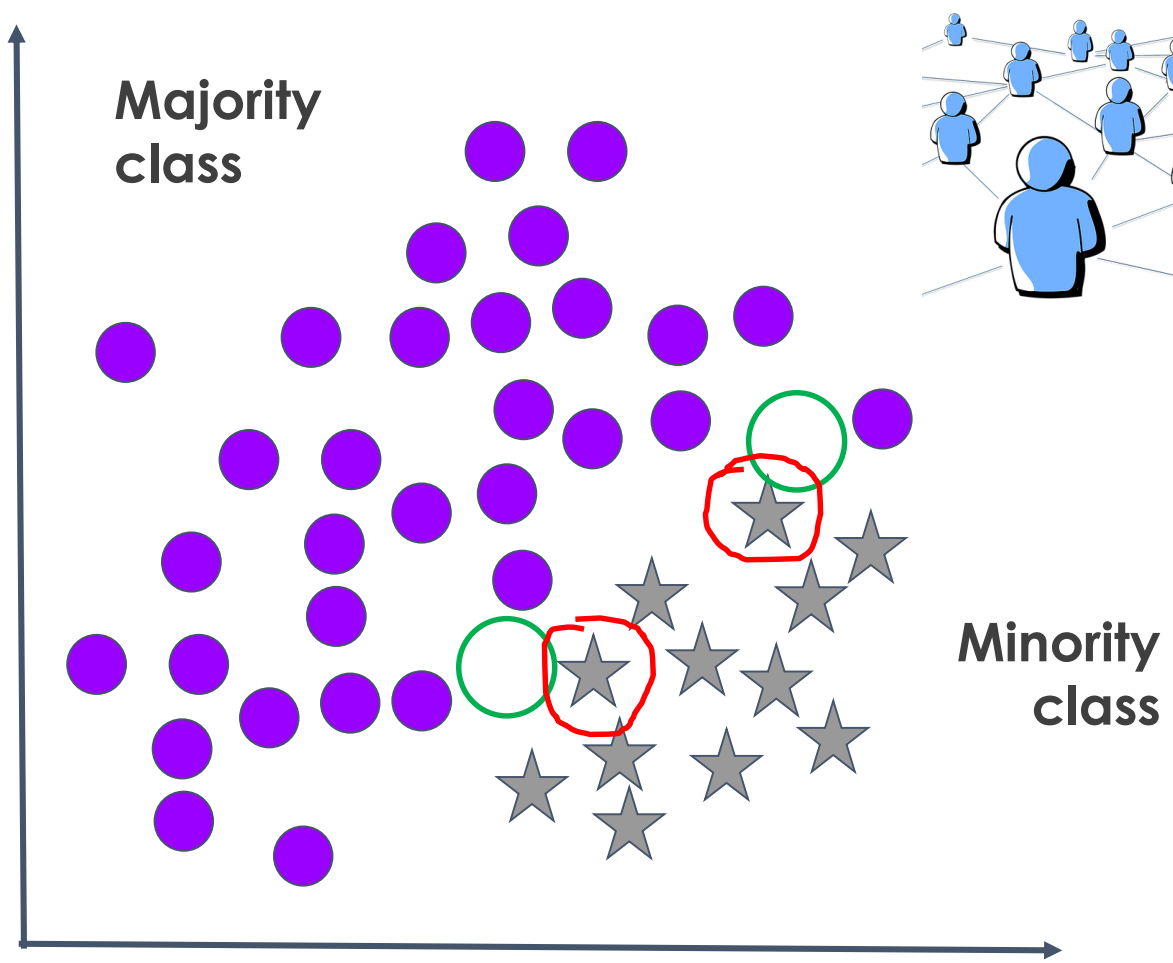


Edited Nearest Neighbours



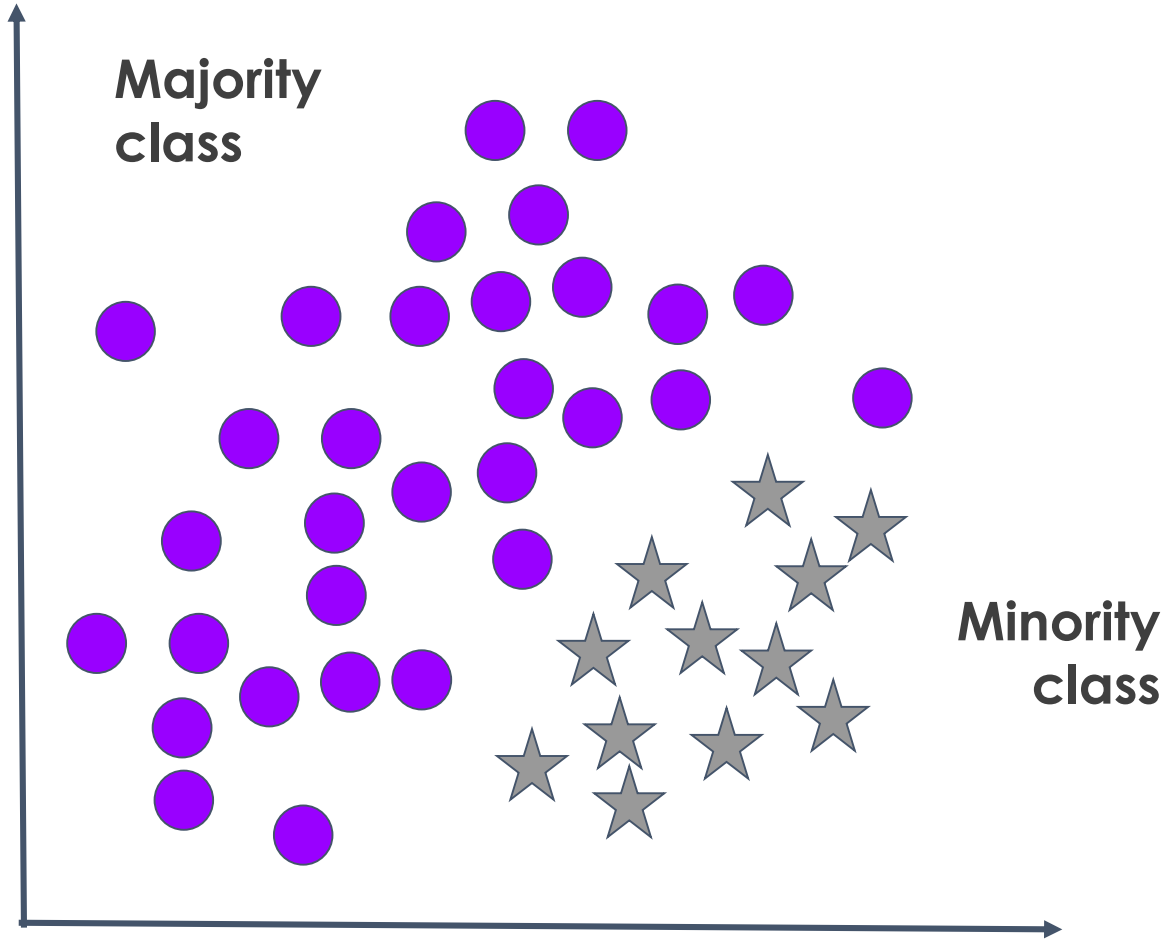
- Find each observation's 1 closest neighbours.
- For simplicity, in the diagram I only show those where the neighbours disagree with the class.

Edited Nearest Neighbours

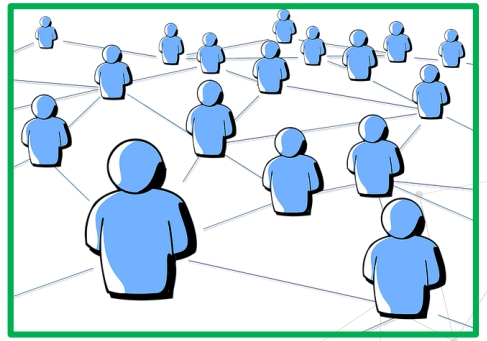


- Remove observations from the majority class, when neighbours disagree.

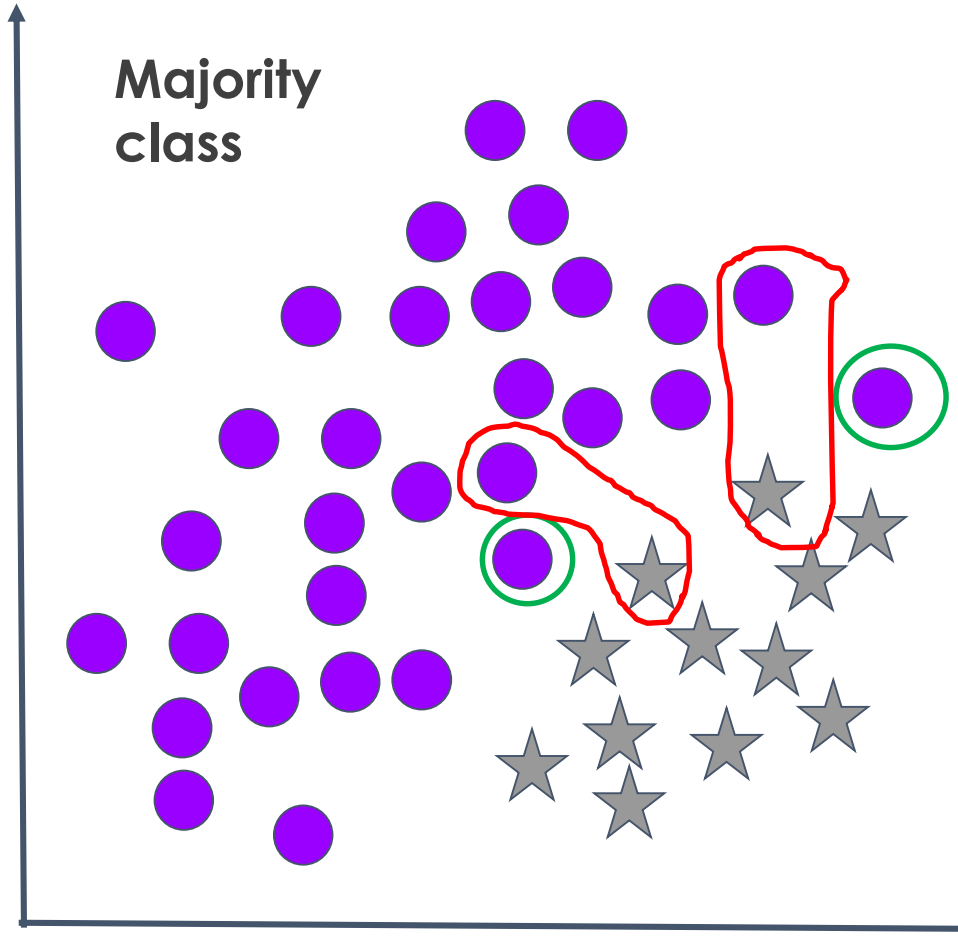
All KNN

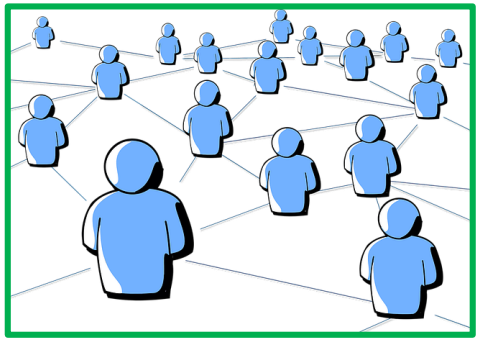


Train a **2** KNN algorithm



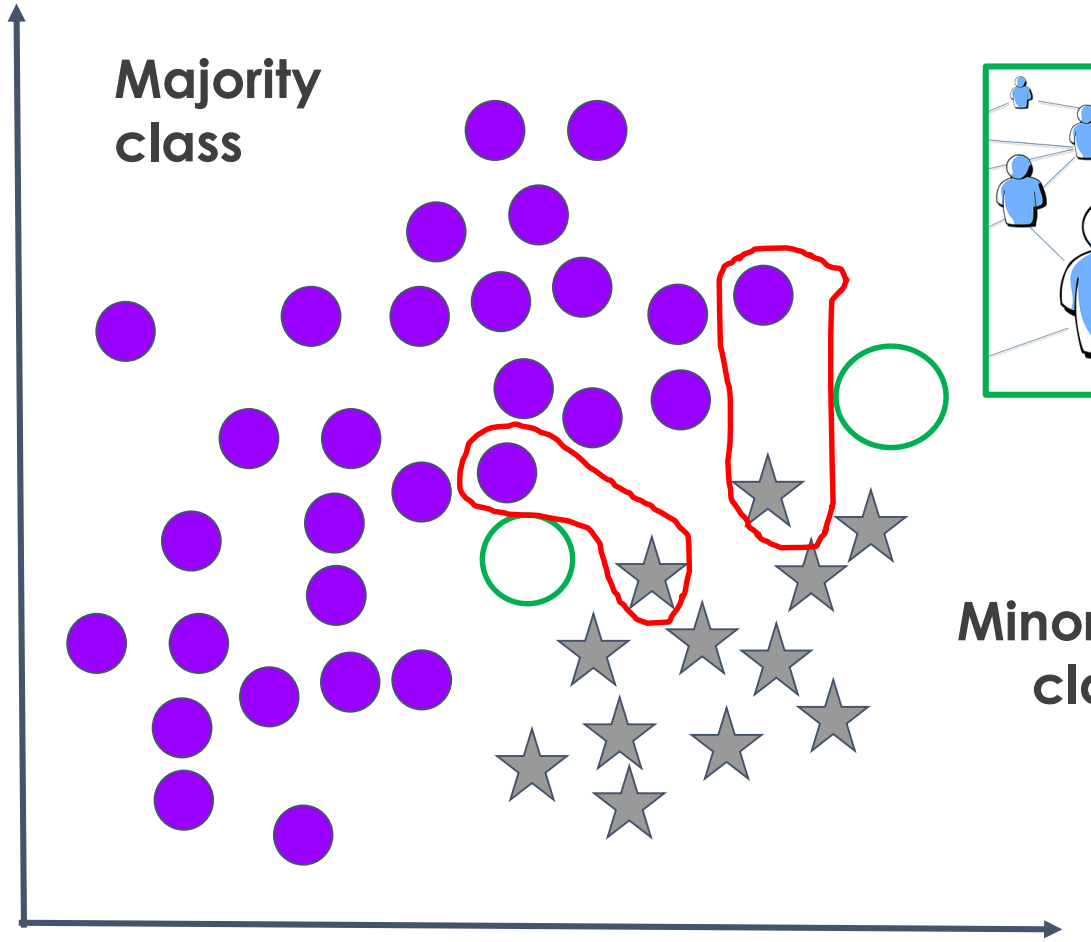
All KNN

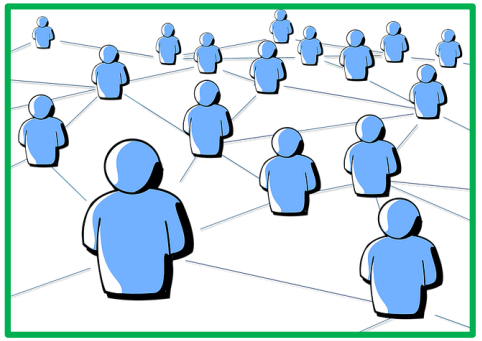




- Find each observations **2** closest neighbours.
- For simplicity, in the diagram I only show those where some of the neighbours disagree with the class.

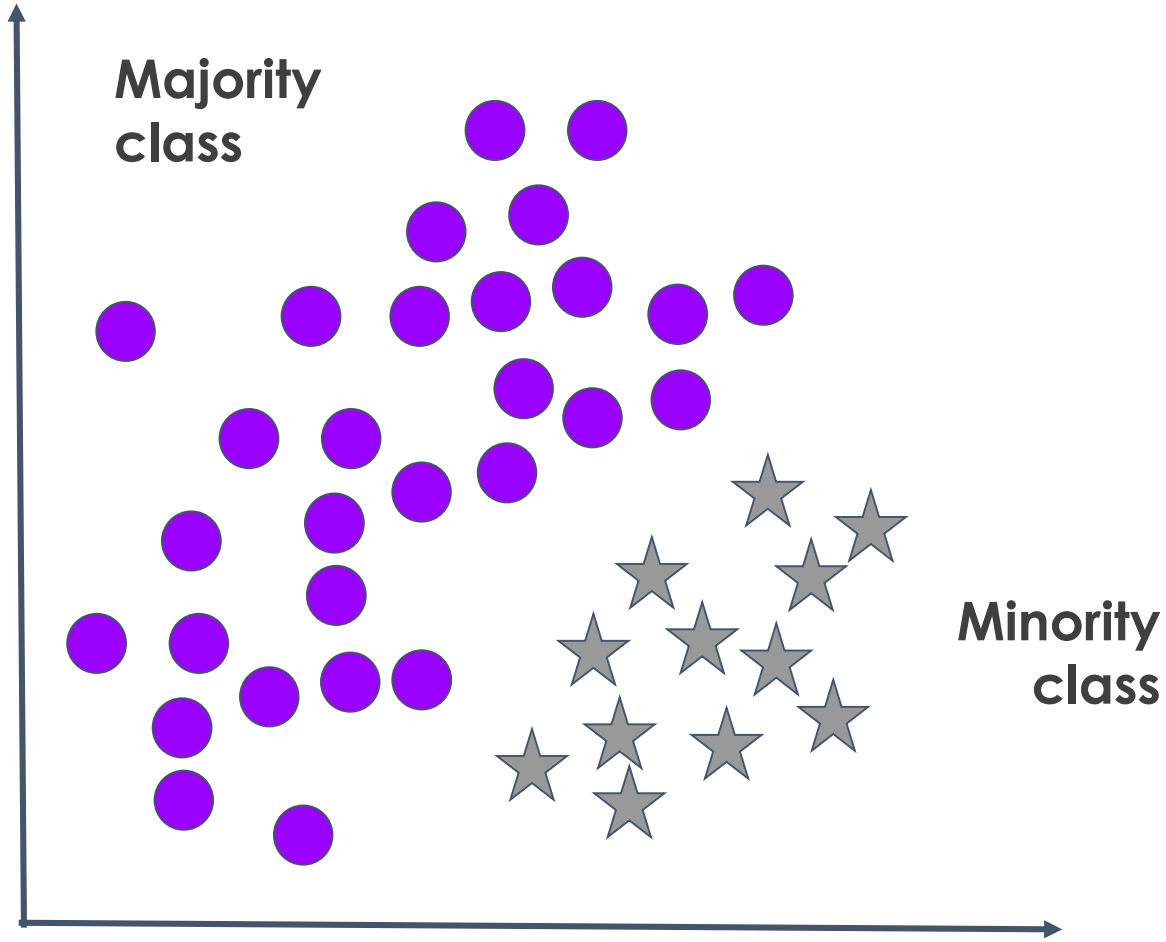
All KNN



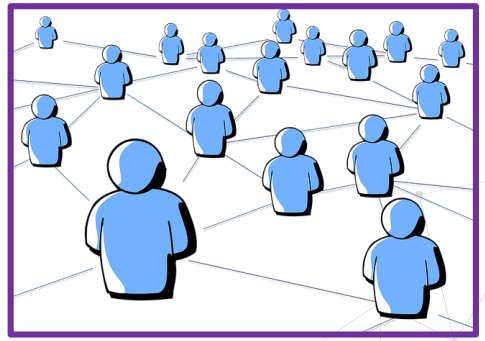


- Remove observations from the majority class, when neighbours disagree.

All KNN



Train a **3** KNN algorithm



All KNN

1. Removes more samples than ENN
2. Various passes over the dataset
3. Successive KNNs have more neighbours

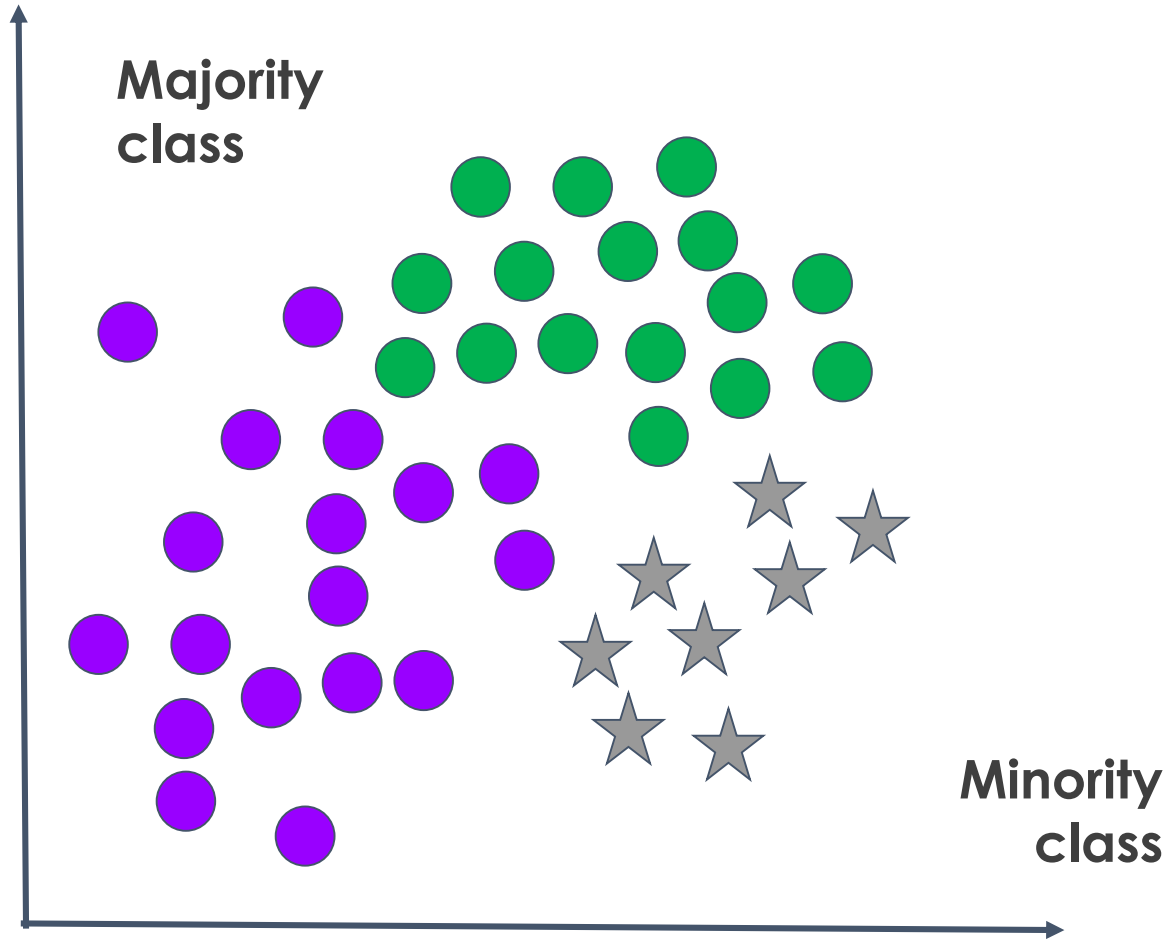
- More observations need to agree on the class
- More stringent

Imbalanced-learn: AllKNN

```
# create data

X, y = make_data(sep=2)

# set up All KNN

allknn = AllKNN(
    sampling_strategy='auto',  # undersamples only the majority class
    n_neighbors=3,
    kind_sel='all',  # all neighbours need to have the same label as the observation examined
    n_jobs=4)  # I have 4 cores in my laptop

X_resampled, y_resampled = allknn.fit_resample(X, y)
```

Maximum number of neighbours to explore

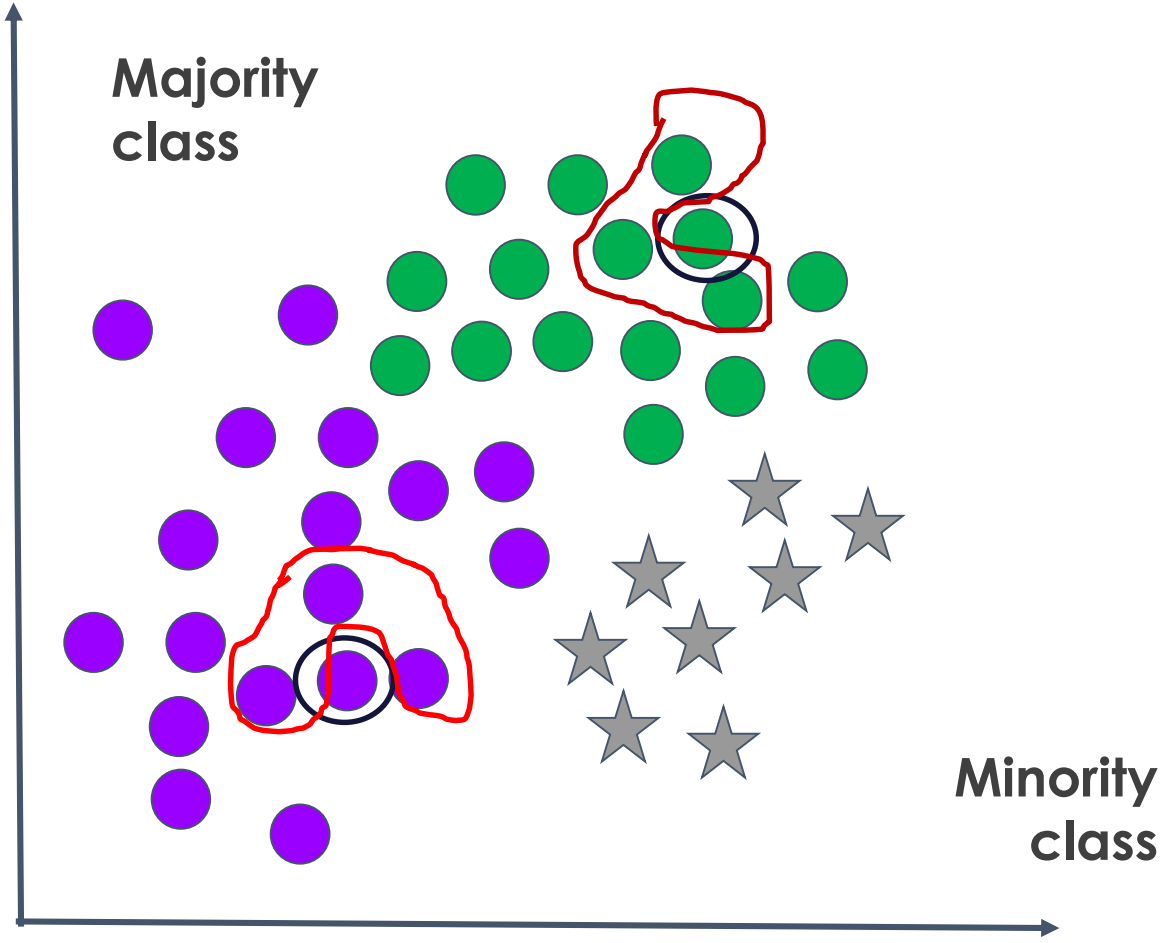
Multi-class



One vs Rest

Only majority classes are undersampled.

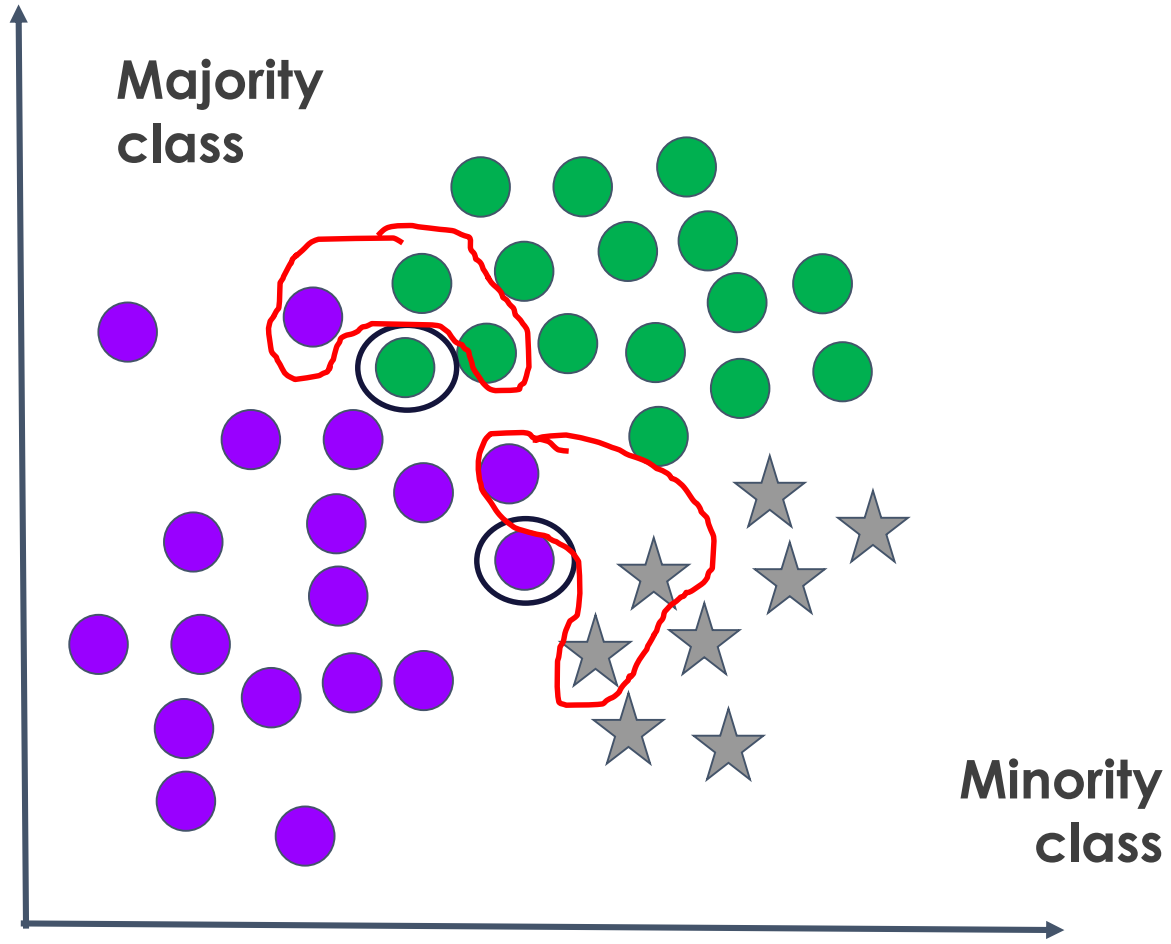
Multi-class



One vs Rest

When all or most neighbours agree, the observation is retained.

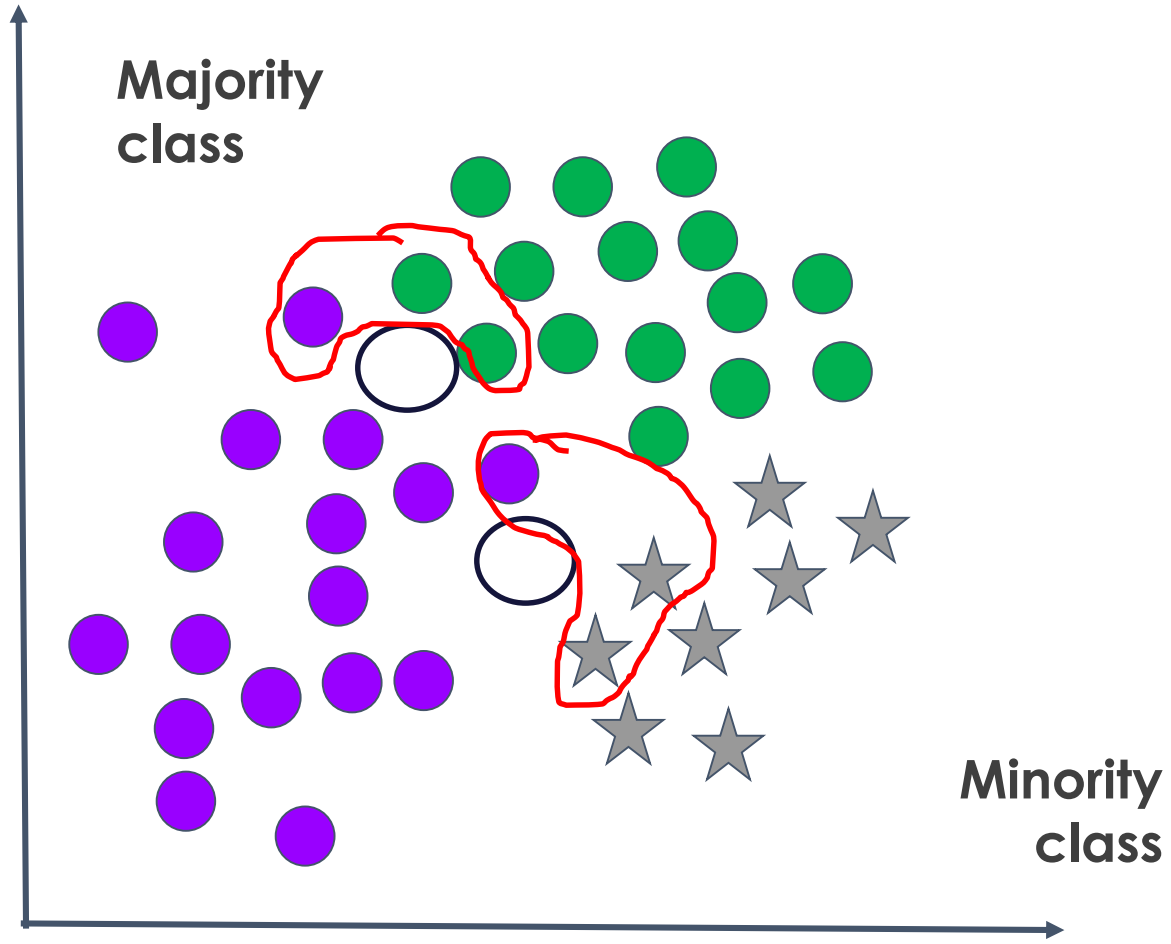
Multi-class



One vs Rest

When neighbours disagree, the observation is removed.

Multi-class



One vs Rest

When neighbours disagree, the observation is removed.

Train In Data

THANK YOU

www.trainindata.com