



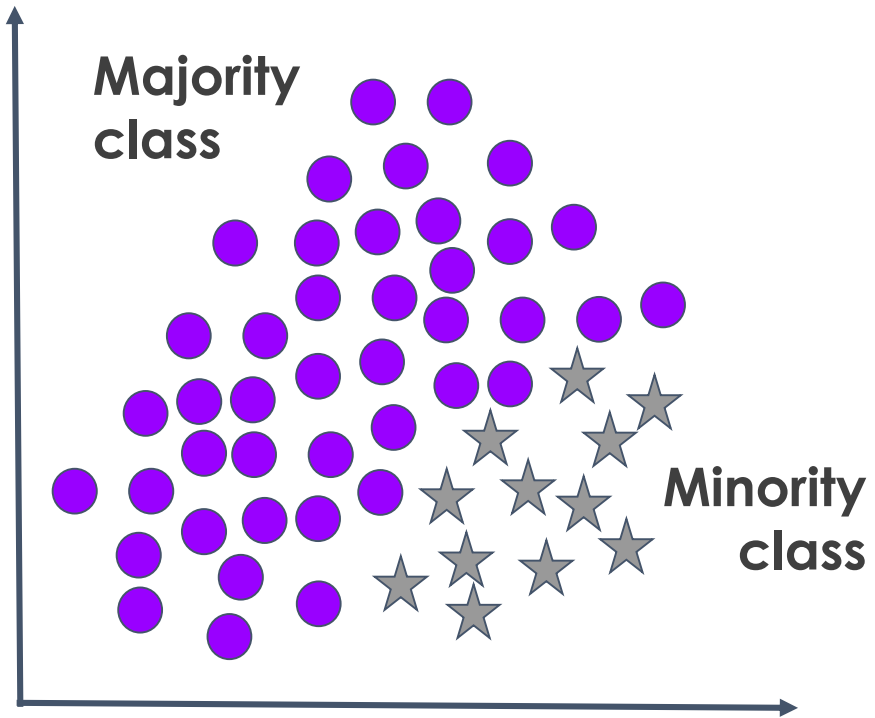
# Condensed Nearest Neighbours

# Condensed Nearest Neighbours (CNN)

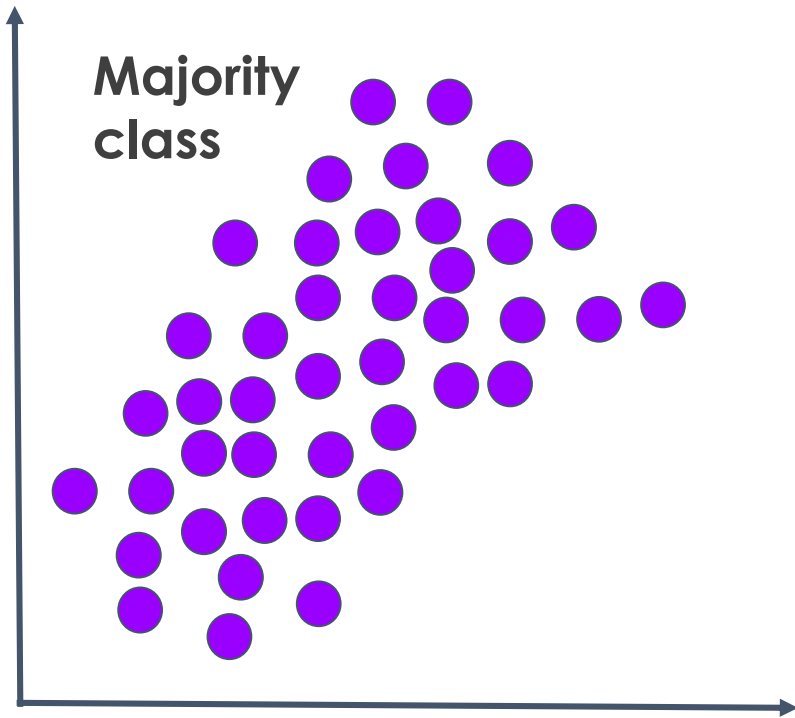
Extracts observations at the boundary between the 2 (or more) classes.

- Cleaning
- Final dataset shape varies
- Boundary matters

# Condensed Nearest Neighbours (CNN)



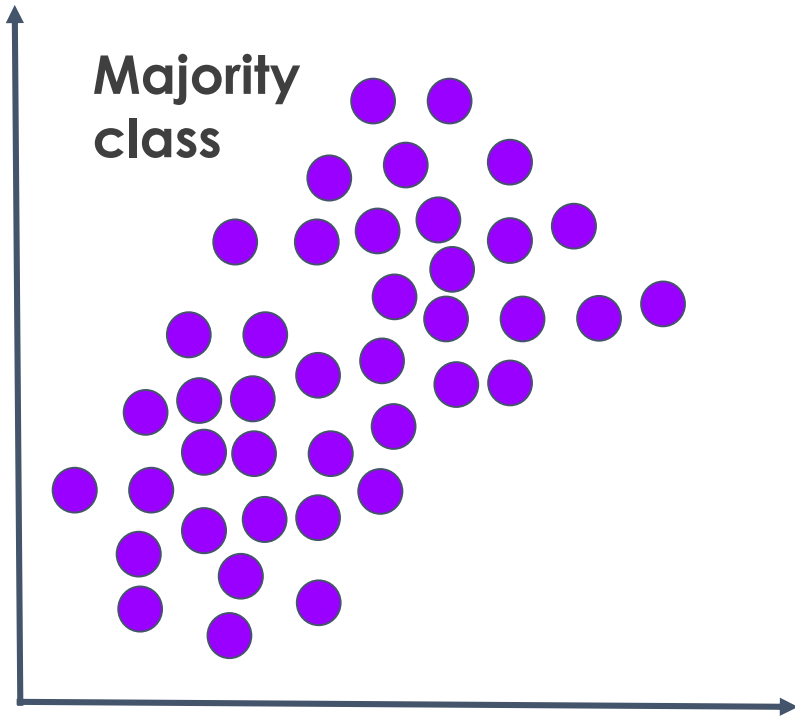
# Condensed Nearest Neighbours (CNN)



Separate minority class into a group



# Condensed Nearest Neighbours (CNN)



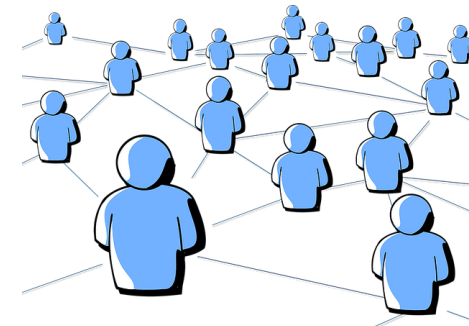
Take 1 observation  
from Majority Class  
to Minority Group



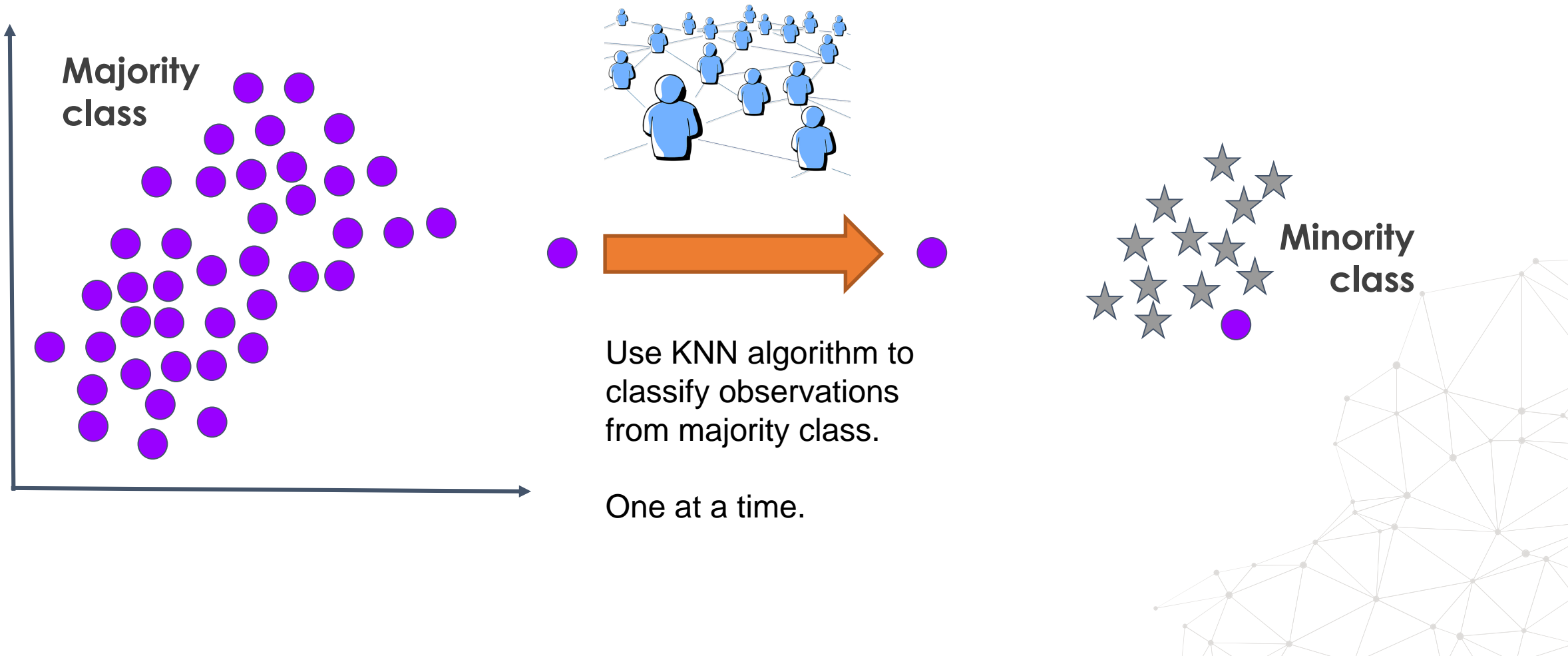
# Condensed Nearest Neighbours (CNN)



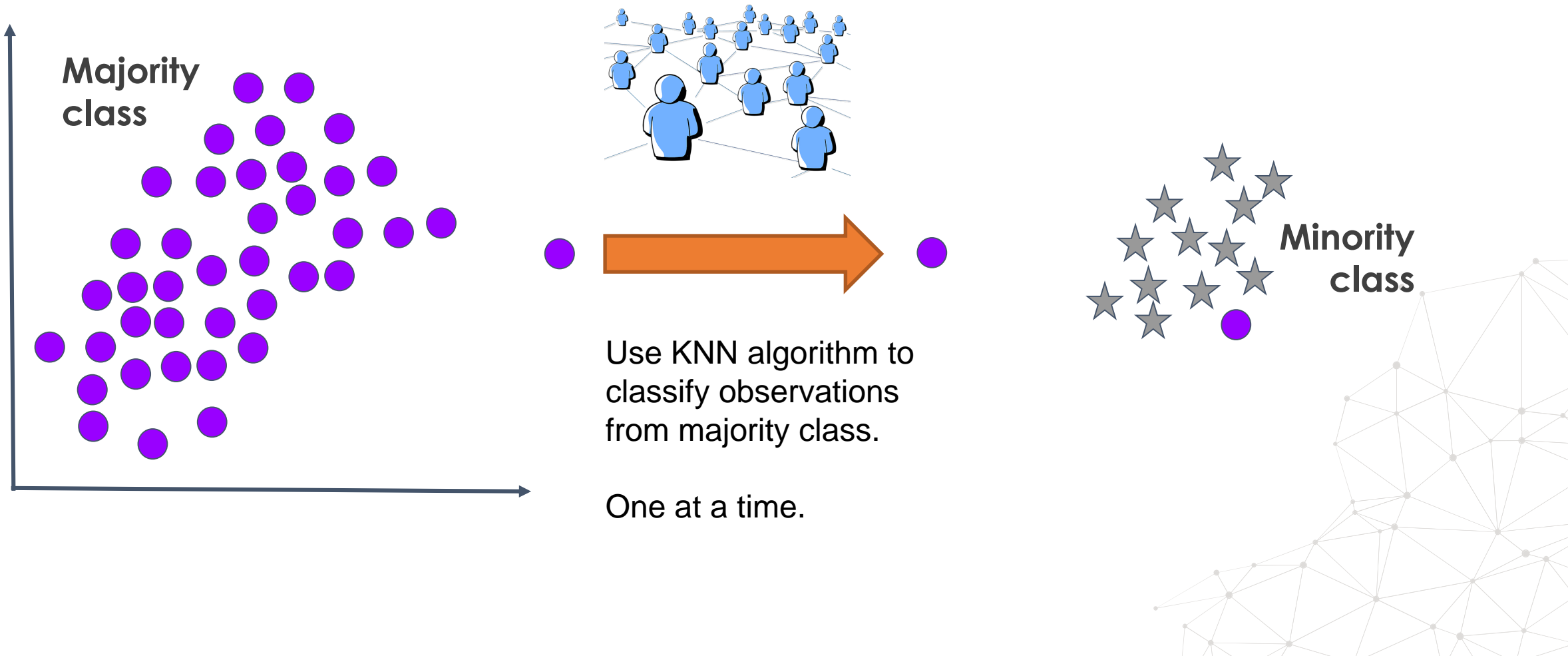
Train a 1 KNN algorithm



# Condensed Nearest Neighbours (CNN)

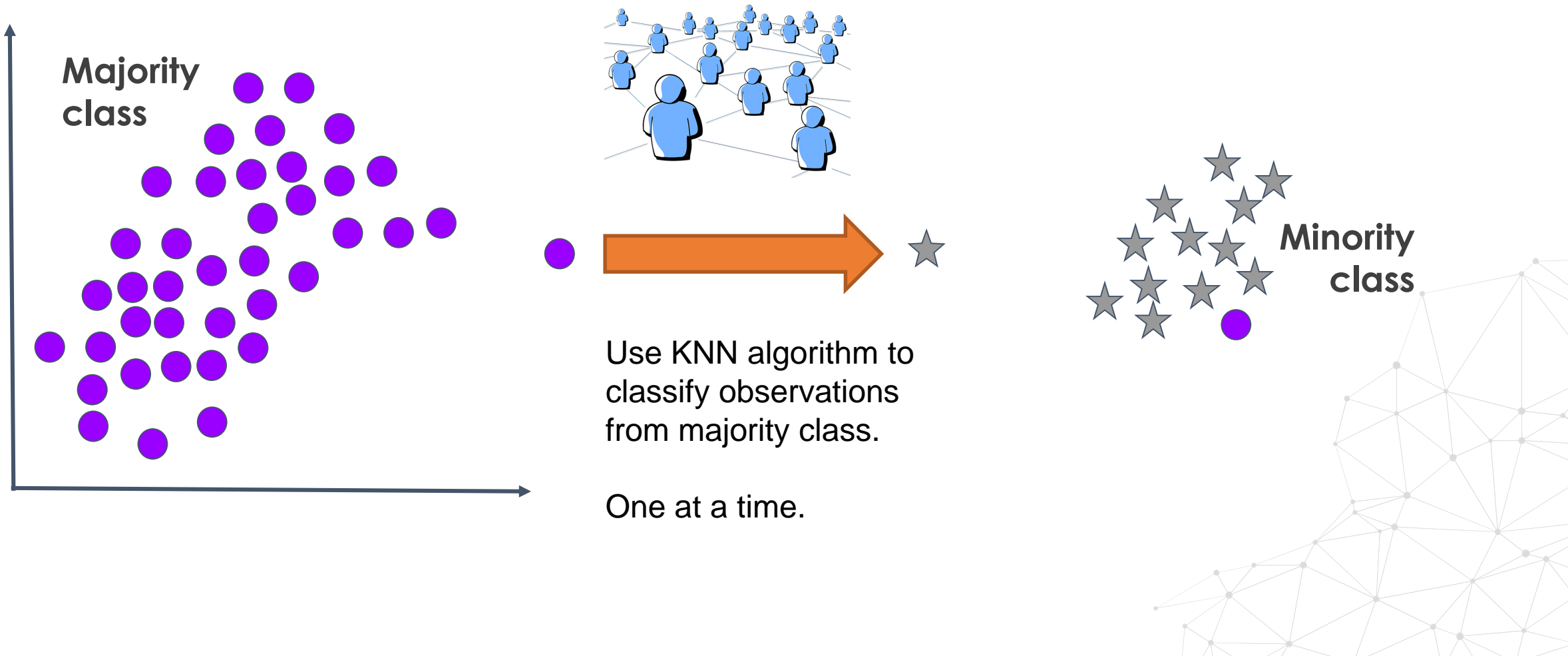


# Condensed Nearest Neighbours (CNN)

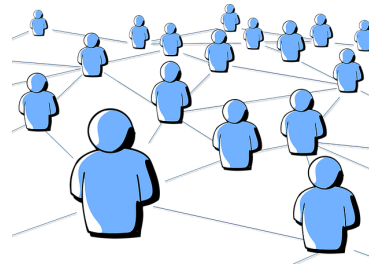
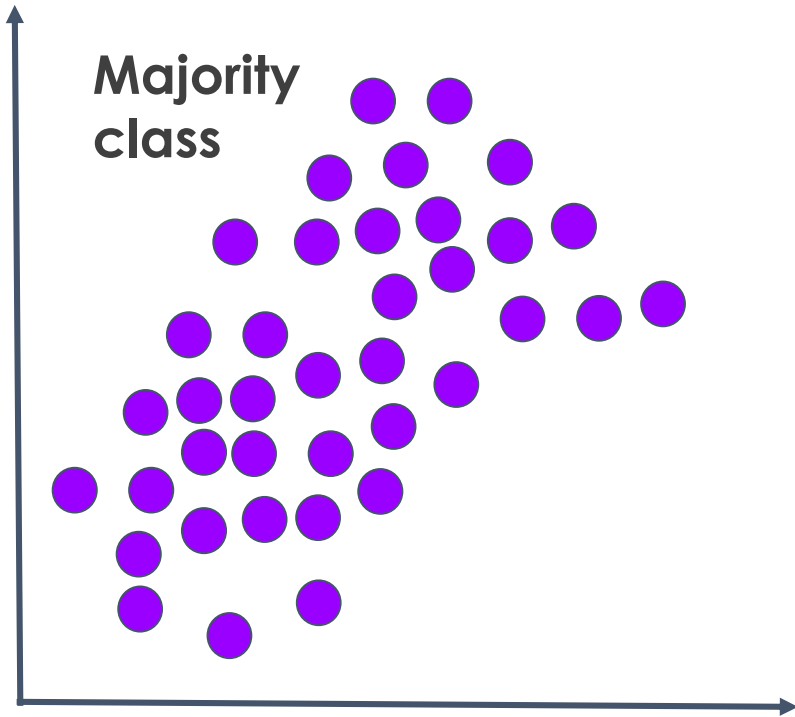




# Condensed Nearest Neighbours (CNN)



# Condensed Nearest Neighbours (CNN)



When predicted class  
does not match real  
class →

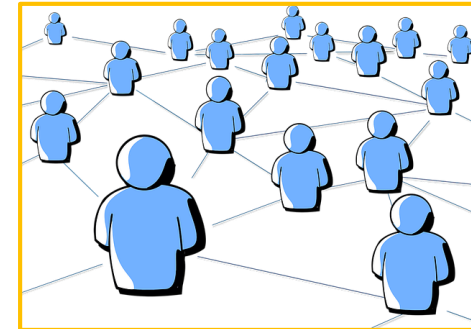
Move observation to  
minority group.



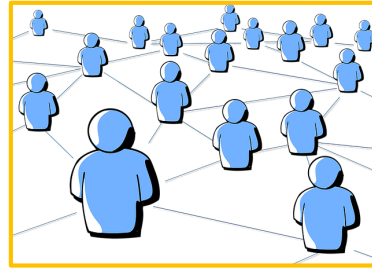
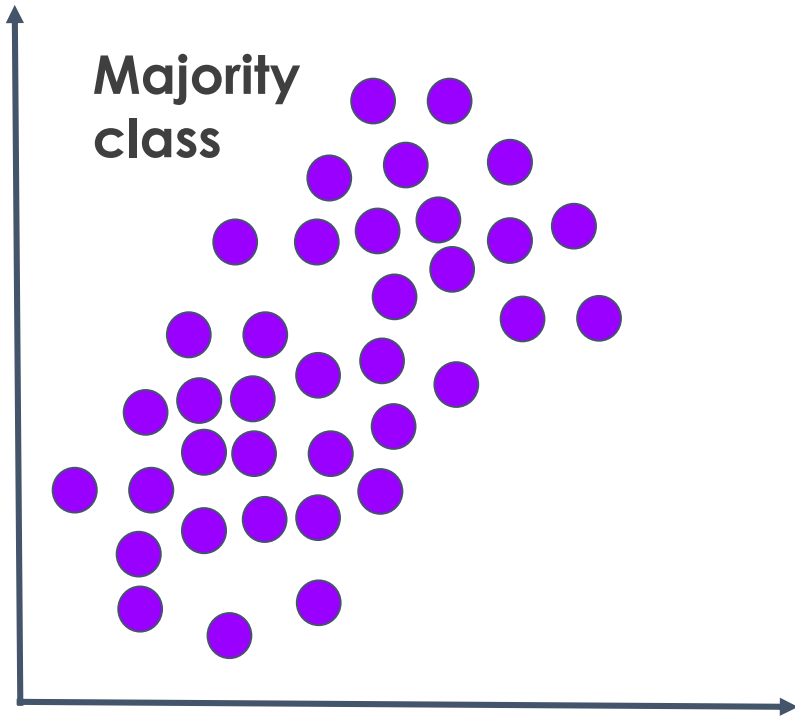
# Condensed Nearest Neighbours (CNN)



Train a new 1 KNN algorithm.



# Condensed Nearest Neighbours (CNN)

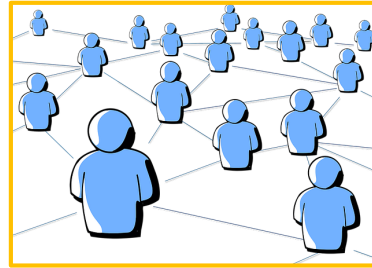
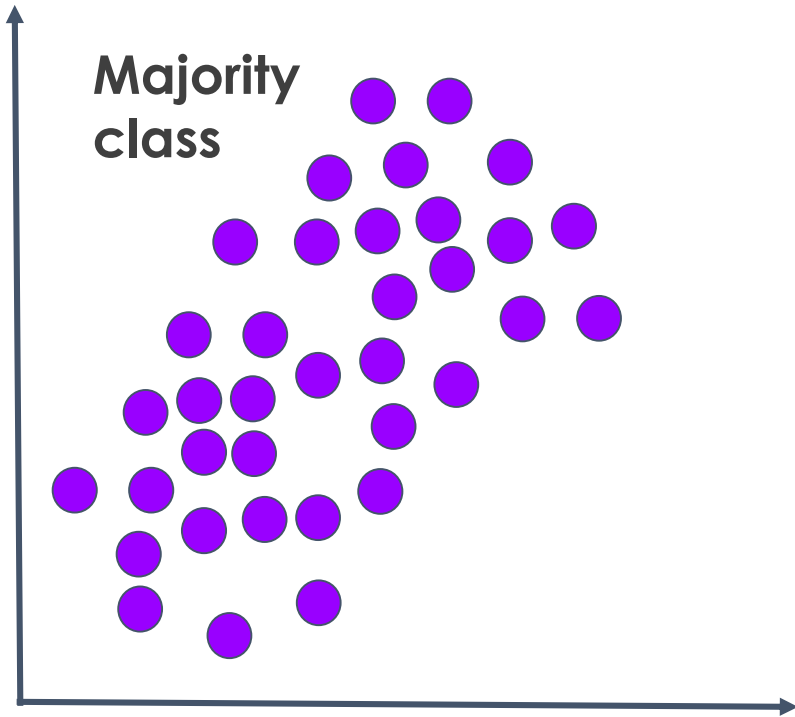


Use new KNN algorithm  
to classify observations  
from majority class.

One at a time.



# Condensed Nearest Neighbours (CNN)

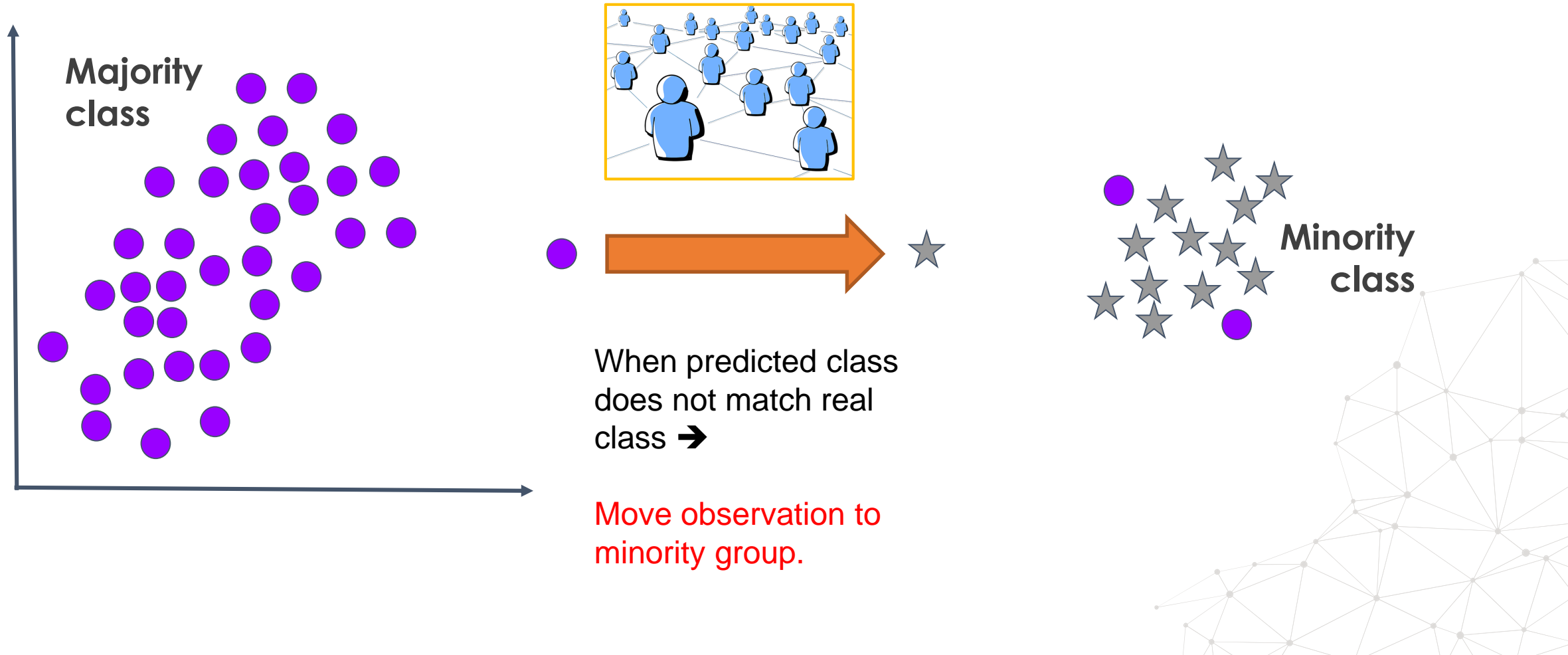


Use new KNN algorithm  
to classify observations  
from majority class.

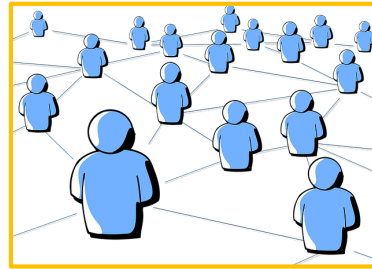
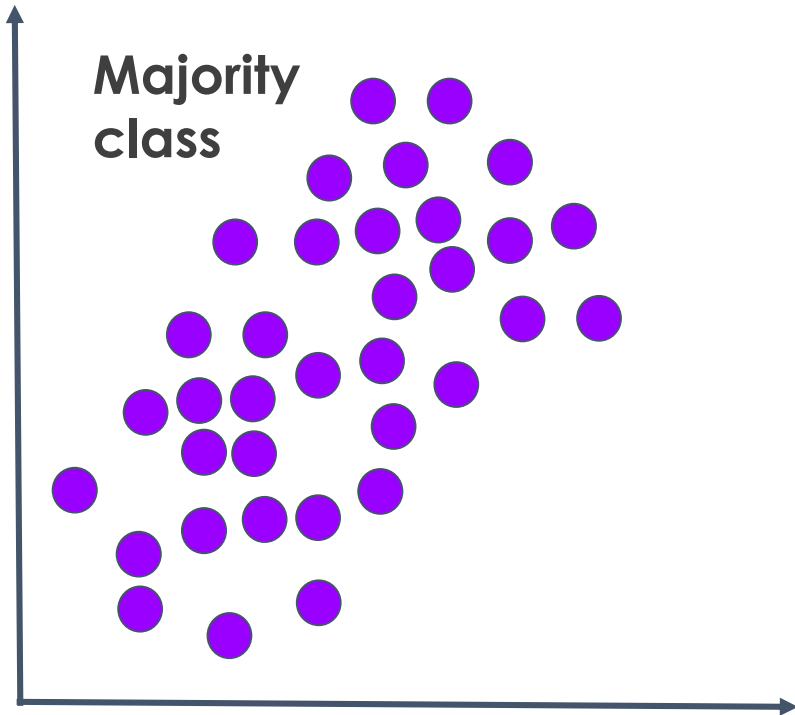
One at a time.



# Condensed Nearest Neighbours (CNN)



# Condensed Nearest Neighbours (CNN)



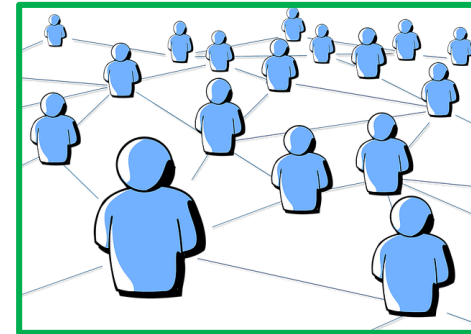
When predicted class  
does not match real  
class →

Move observation to  
minority group.

# Condensed Nearest Neighbours (CNN)

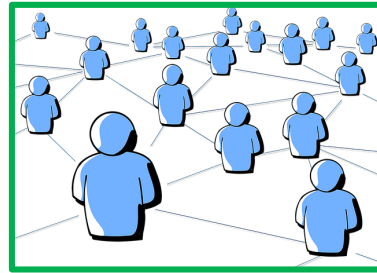
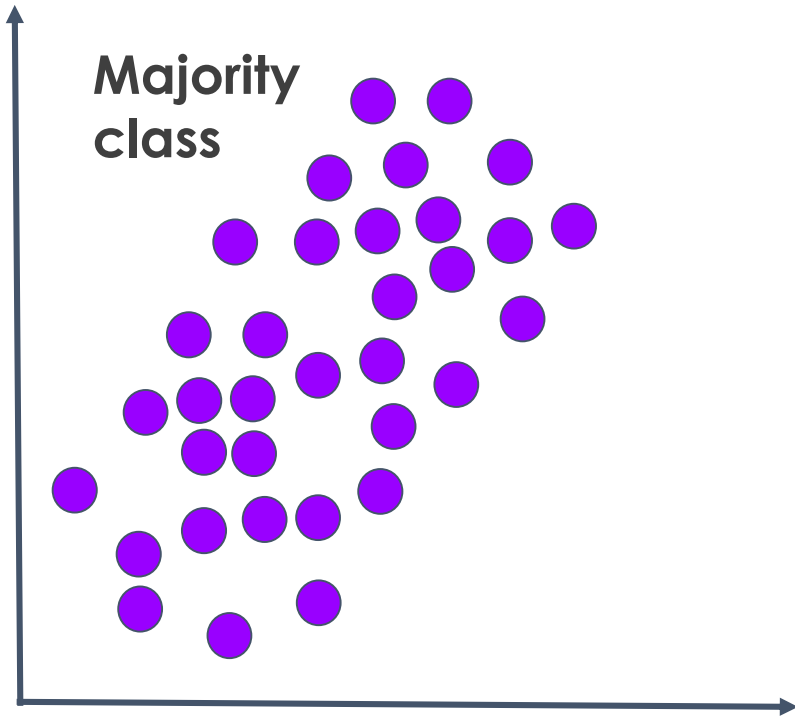


Train another 1 KNN algorithm.





# Condensed Nearest Neighbours (CNN)

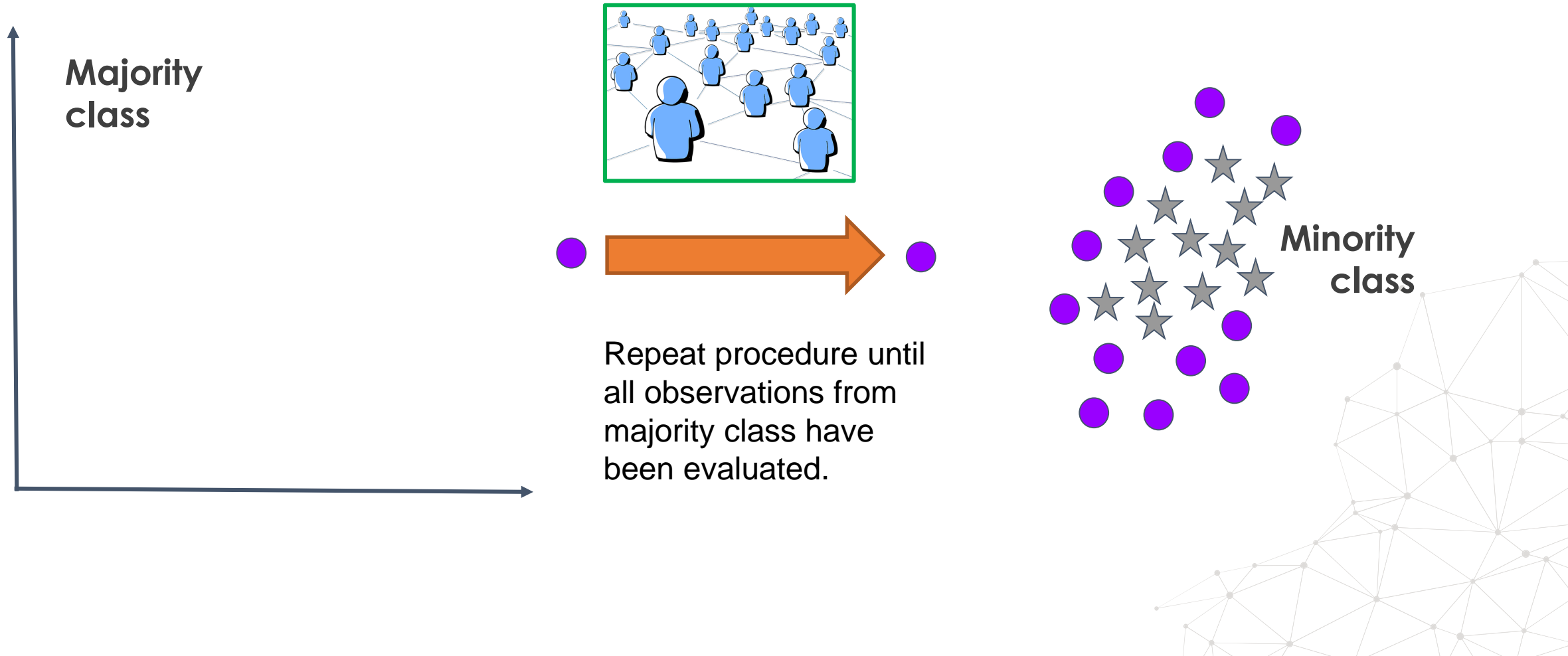


Use new KNN algorithm to classify observations from majority class.

One at a time.



# Condensed Nearest Neighbours (CNN)



# Condensed Nearest Neighbours (CNN)



The final dataset contains the minority class + all observations from the majority class that were wrongly classified by the subsequent KNN algorithms.

# Condensed Nearest Neighbours (CNN)



Focus on harder cases → improves performance



Introduces noise



# Imbalanced-learn: CNN

```
] : # create data

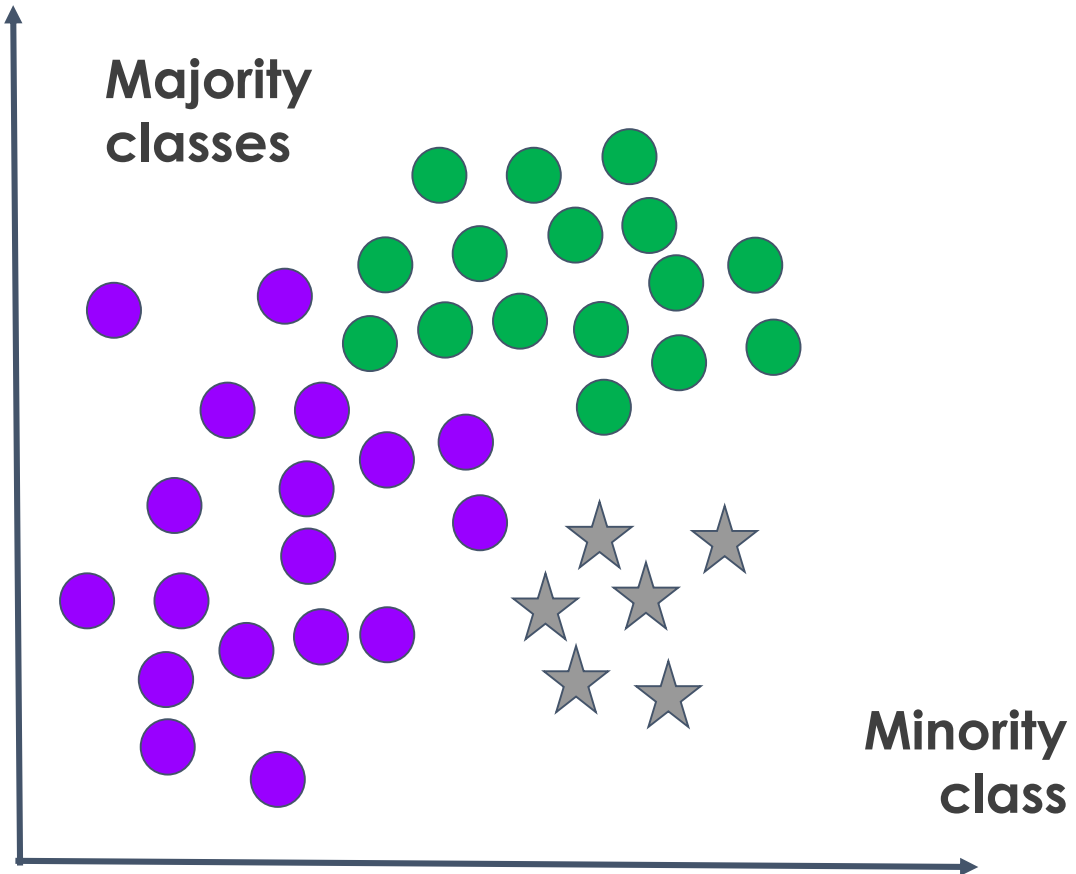
X, y = make_data(sep=2)

# set up condensed nearest neighbour transformer

cnn = CondensedNearestNeighbour(
    sampling_strategy='auto', # undersamples only the majority class
    random_state=0, # for reproducibility
    n_neighbors=1, # default
    n_jobs=4) # I have 4 cores in my laptop

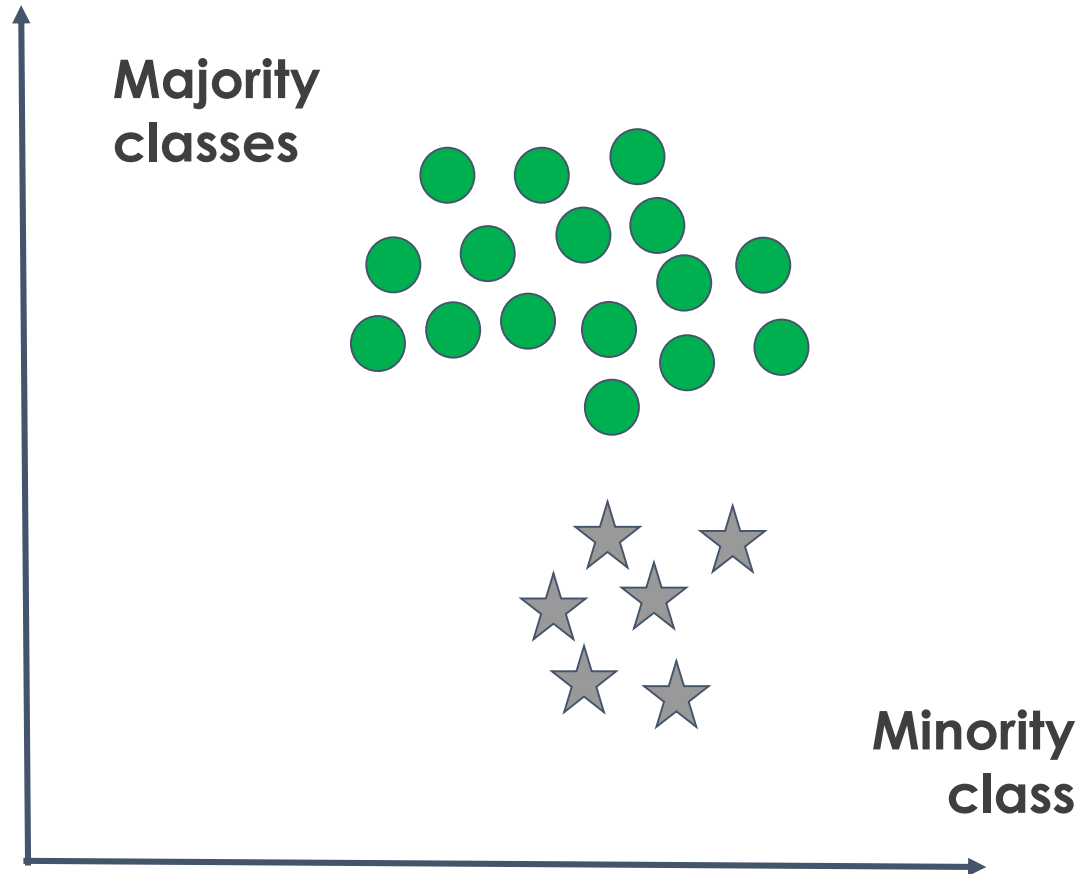
X_resampled, y_resampled = cnn.fit_resample(X, y)
```

# Multi-class



One vs One.

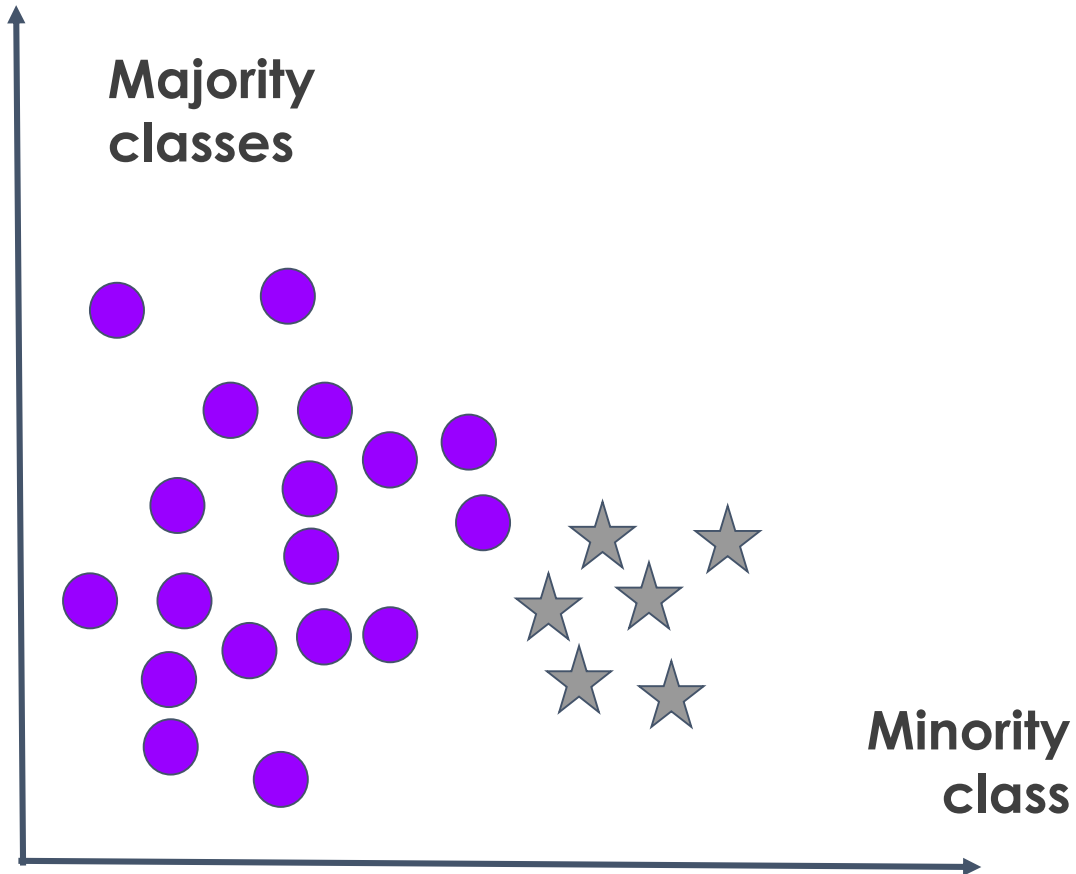
# Multi-class



One vs One.

Run entire  
procedure over 1  
majority class first

# Multi-class



One vs One.

Repeat the procedure for the other majority class.



# THANK YOU

[www.trainindata.com](http://www.trainindata.com)