



## LECTURE:

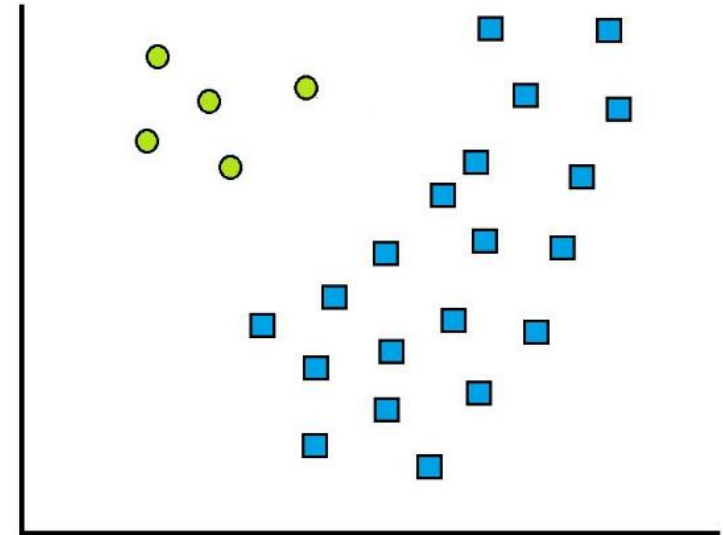
- K-Nearest Neighbor (KNN)

Spring 2024 - University of North Carolina at Charlotte

# K-Nearest Neighbors (KNN)

# K-Nearest Neighbors (KNN)

- KNN Algorithm: **supervised learning algorithm** that classifies unknown data points based on their proximity to known data points.
- So far, all machine learning models use the training data to compute a representation of the original. In contrast, KNN does not compute a new model but uses the whole data set as a model. Every single instance of your training data is one part of your model.
- It assumes that similar things are situated in close proximity.
- The number K is chosen by the user (hyperparameter of the model).
- KNN algorithm can be used for either prediction or classification: *KNeighborsRegressor* for regression and *KNeighborsClassifier* for classification.



# KNN Strategy

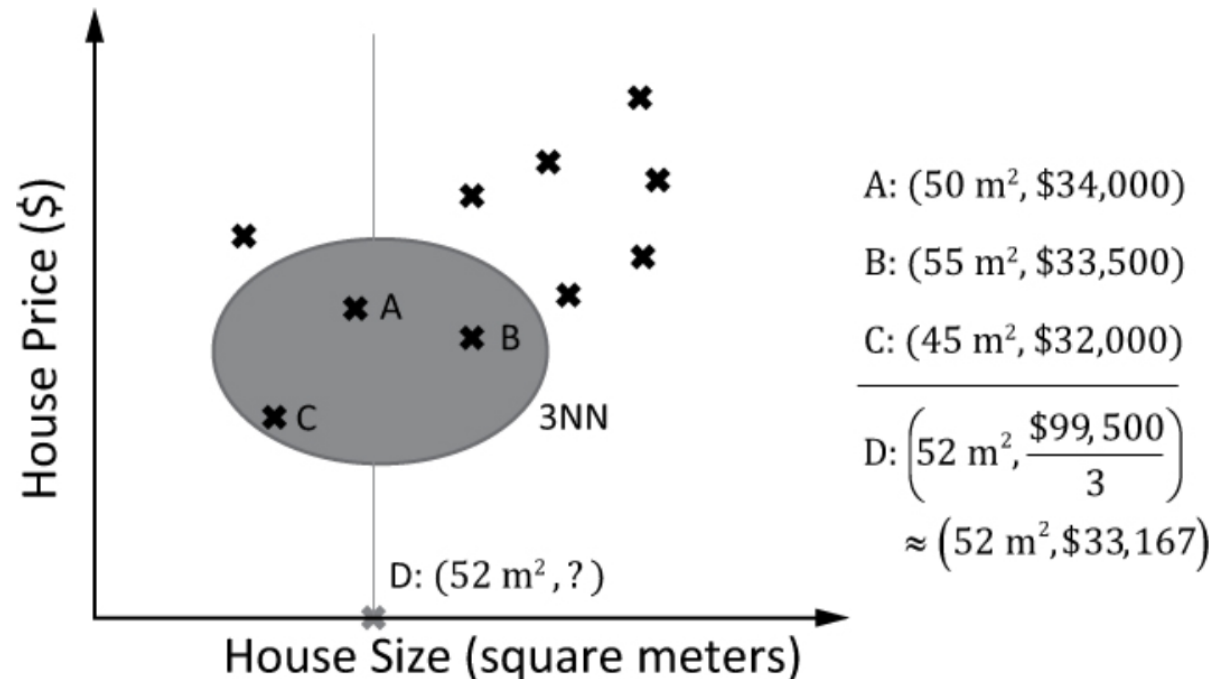
- Given an input vector  $X$ :
  1. Find the  $k$  nearest neighbors of  $x$  (according to a predefined distance metric).
  2. Aggregate the  $k$  nearest neighbors into a single prediction or classification value (use any aggregator function such as average, mean, max, or min).

## Example:

- A company sells homes for clients and has a large database of customers and house prices. One day, one client asks how much they must expect to pay for a house of 52 square meters.
- You query your KNN model. It gives you the response/prediction \$33,167. Your client finds a home for \$33,489 the same week.

How did the KNN model do?

- The KNN model calculates the  $k = 3$  nearest neighbors to the query  $D = 52$  square meters using Euclidean distance. The 3 nearest neighbors (A, B, and C) with prices \$34,000, \$33,500, and \$32,000.
- KNN model aggregates the 3 nearest neighbors by calculating the average of their values. Because  $k = 3$  in this example, you denote the model as 3NN.



# KNN Code Example


```
## Dependencies
from sklearn.neighbors import KNeighborsRegressor
import numpy as np

## Data (House Size (square meters) / House Price ($))
X = np.array([[35, 30000], [45, 45000], [40, 50000],
              [35, 35000], [25, 32500], [40, 40000]])

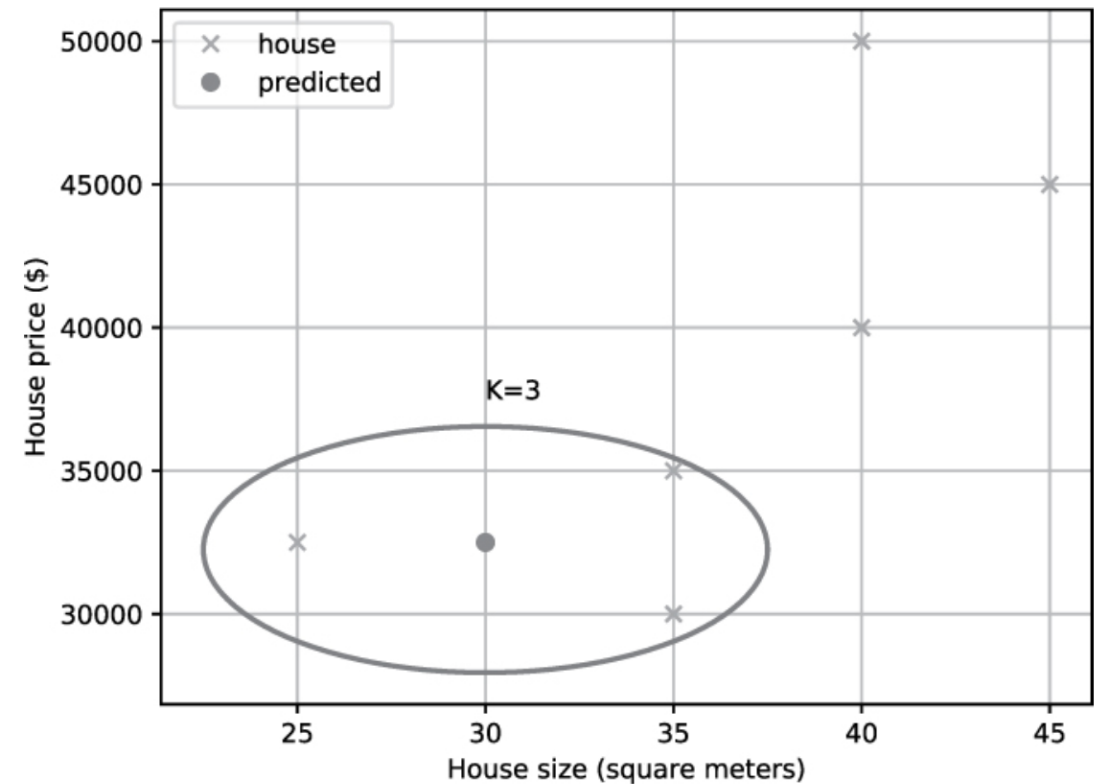
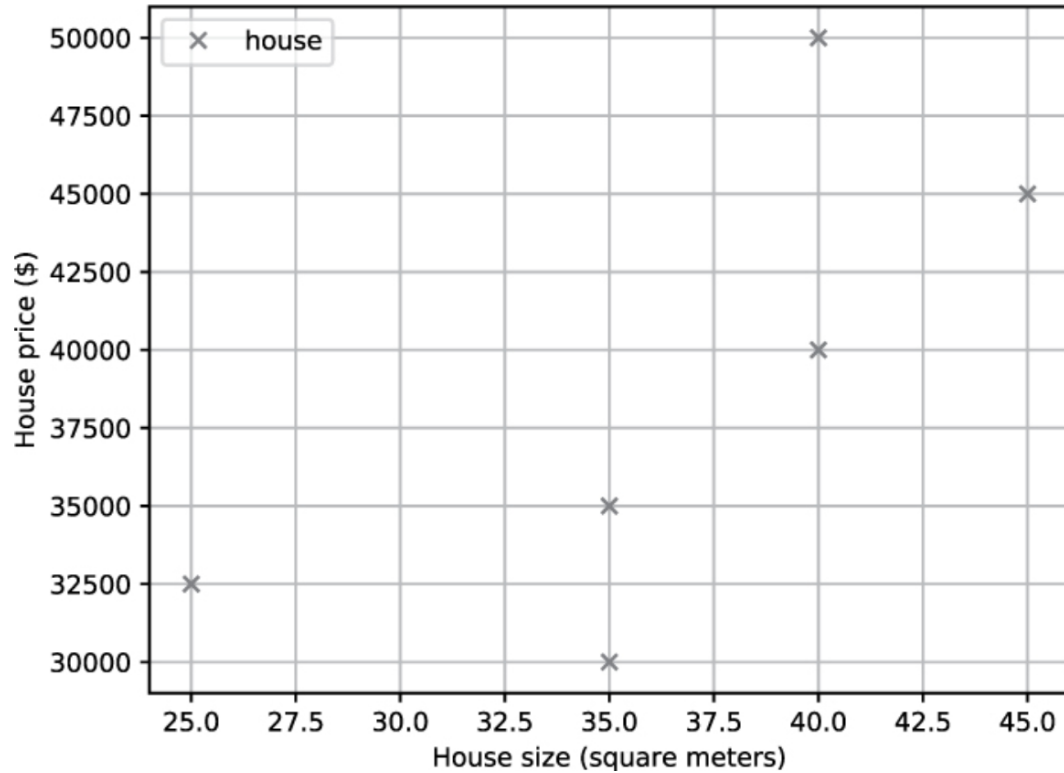
KNN = KNeighborsRegressor(n_neighbors=3).fit(X[:,0].reshape(-1,1), X[:,1])

## Result & puzzle
res = KNN.predict([[30]])
print(res)
```

Converts the  
input into an  
array like input.



- The client requests your price prediction for a house of 30 square meters.
- What does KNN with  $k = 3$  (3NN) predicts/finds the three closest houses with respect to house size and averages the predicted house price as the average of the  $k=3$  nearest neighbors. Thus, the result is \$32,500.



Questions