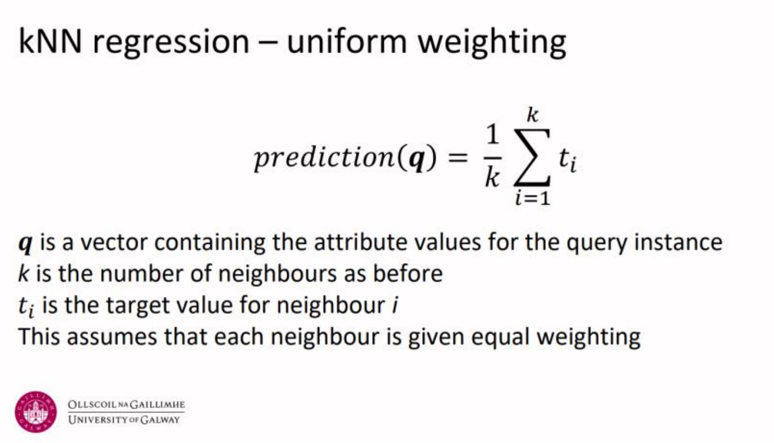
References

1. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html
2. https://bookdown.org/tpinto_home/Regression-and-Classification/k-nearest-neighbours-regression.html
3. https://www.mygreatlearning.com/blog/knn-algorithm-introduction/
4. https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html
5. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_validate.html#sklearn.model_selection.cross_validate
6. https://scikit-learn.org/stable/modules/cross_validation.html#computing-cross-validated-metrics
7. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
8. https://scikit-learn.org/stable/modules/grid_search.html#exhaustive-grid-search
9. https://www.geeksforgeeks.org/neural-networks-a-beginners-guide/
10. https://www.javatpoint.com/artificial-neural-network
11. https://www.ibm.com/topics/neural-networks#:~:text=Neural%20networks%2C%20also%20known%20as,neurons%20signal%20to%20one%20another.
12. https://www.datacamp.com/blog/what-are-neural-networks

Appendix

[1]

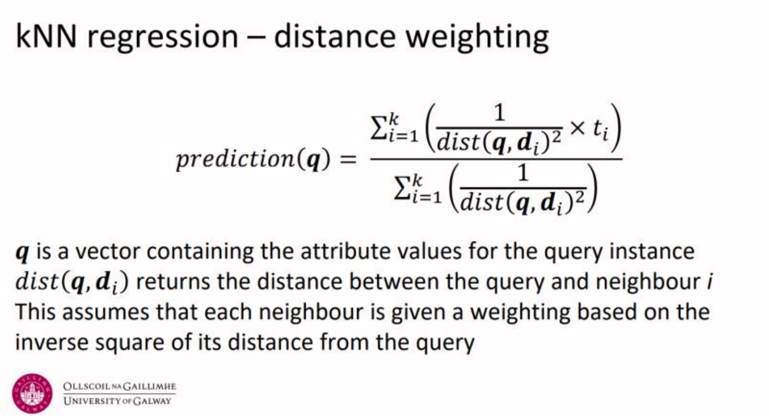


kNN regression – uniform weighting

$$prediction(\boldsymbol{q}) = \frac{1}{k}\sum_{i=1}^{k} t_i$$

$\boldsymbol{q}$ is a vector containing the attribute values for the query instance
k is the number of neighbours as before
t_i is the target value for neighbour i
This assumes that each neighbour is given equal weighting

Ollscoil na Gaillimhe
University of Galway

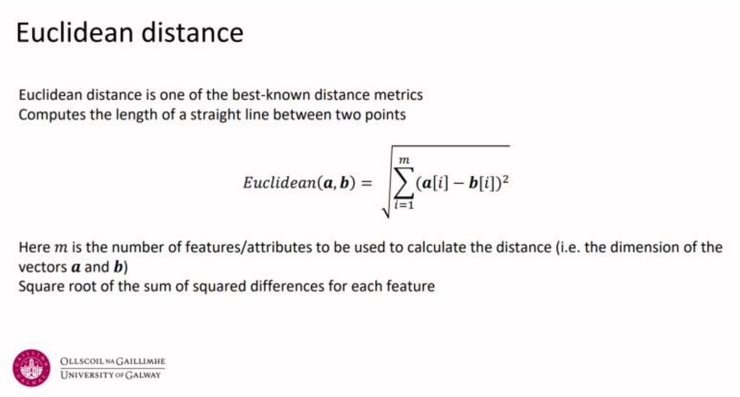


kNN regression – distance weighting

$$prediction(\boldsymbol{q}) = \frac{\sum_{i=1}^{k}\left(\frac{1}{dist(\boldsymbol{q},\boldsymbol{d}_i)^2} \times t_i\right)}{\sum_{i=1}^{k}\left(\frac{1}{dist(\boldsymbol{q},\boldsymbol{d}_i)^2}\right)}$$

$\boldsymbol{q}$ is a vector containing the attribute values for the query instance
$dist(\boldsymbol{q},\boldsymbol{d}_i)$ returns the distance between the query and neighbour *i*
This assumes that each neighbour is given a weighting based on the inverse square of its distance from the query

[2]

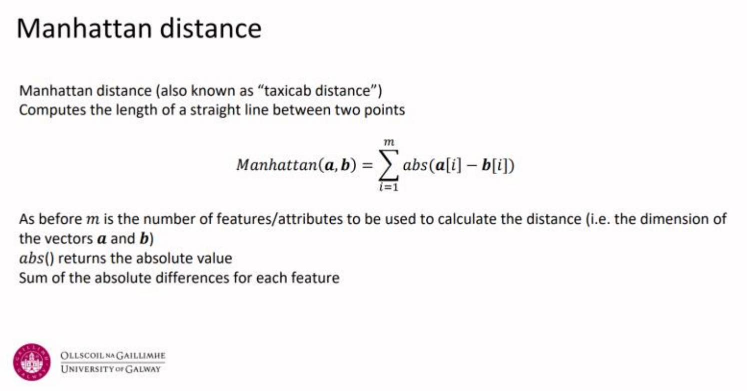


Euclidean distance

Euclidean distance is one of the best-known distance metrics
Computes the length of a straight line between two points

$$Euclidean(\boldsymbol{a},\boldsymbol{b}) = \sqrt{\sum_{i=1}^{m}(\boldsymbol{a}[i]-\boldsymbol{b}[i])^2}$$

Here m is the number of features/attributes to be used to calculate the distance (i.e. the dimension of the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$)
Square root of the sum of squared differences for each feature

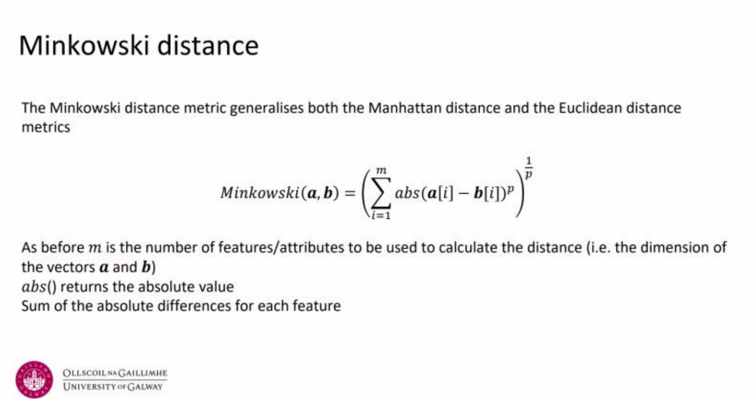


Manhattan distance

Manhattan distance (also known as “taxicab distance”)
Computes the length of a straight line between two points

$$Manhattan(\boldsymbol{a},\boldsymbol{b}) = \sum_{i=1}^{m} abs(\boldsymbol{a}[i]-\boldsymbol{b}[i])$$

As before m is the number of features/attributes to be used to calculate the distance (i.e. the dimension of the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$)
$abs()$ returns the absolute value
Sum of the absolute differences for each feature



Minkowski distance

The Minkowski distance metric generalises both the Manhattan distance and the Euclidean distance metrics

$$Minkowski(\boldsymbol{a}, \boldsymbol{b}) = \left(\sum_{i=1}^{m} abs(\boldsymbol{a}[i] - \boldsymbol{b}[i])^{p} \right)^{\frac{1}{p}}$$

As before m is the number of features/attributes to be used to calculate the distance (i.e. the dimension of the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$)
$abs()$ returns the absolute value
Sum of the absolute differences for each feature

Ollscoil na Gaillimhe
University of Galway