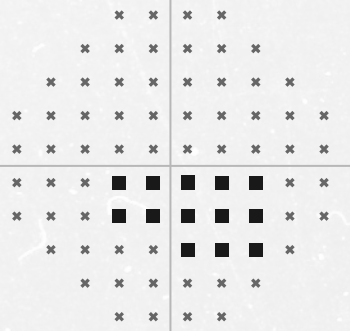
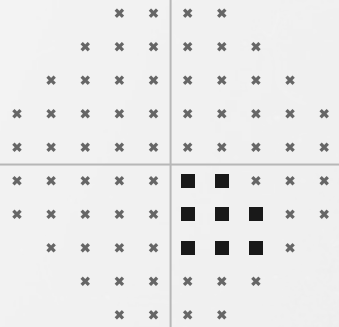
K Nearest Neighbours



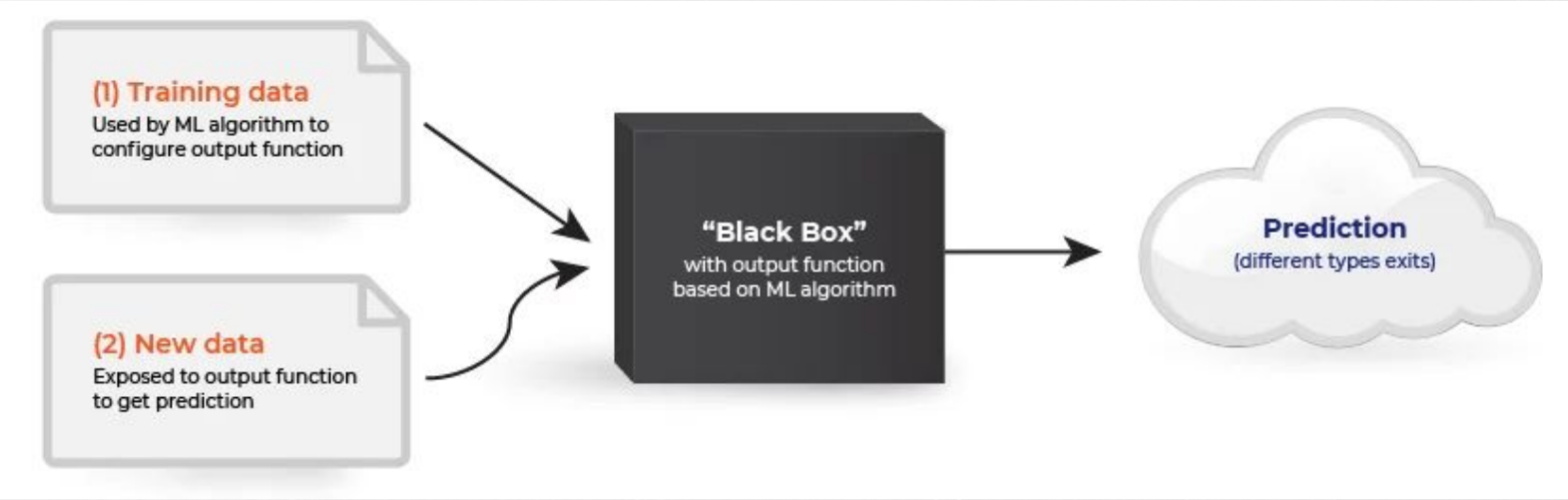
01



Goal of ML

What is the goal of a Machine Learning Algorithm?

- For every problem we encounter there is a function that can get us the output of this problem.
- The goal of a machine learning model is to find a function h also called a hypothesis that can approximate this real function
- For example we have the function of detecting objects in our brain however we don't know what it is so we try to approximate it using machine learning



What is the goal of Machine Learning Algorithms?

- For supervised learning we mentioned that it consists of training data & labels
  - x → training data
  - y → training labels
- We try to get a function h “hypothesis function” that tries to approximate the real function f(x) that outputs the training labels.

- Consistent hypothesis: Is is a hypothesis that agrees with f on all examples
  - However such a hypothesis is not always possible due to
    1) Insufficient hypothesis space
       - f(x) = sinx and we are searching in polynomials of finite degree
    2) Noisy data
- If we search in a bigger space then we can find a better hypothesis but it will be more complex to find this function

02

Nearest Neighbour Classifier

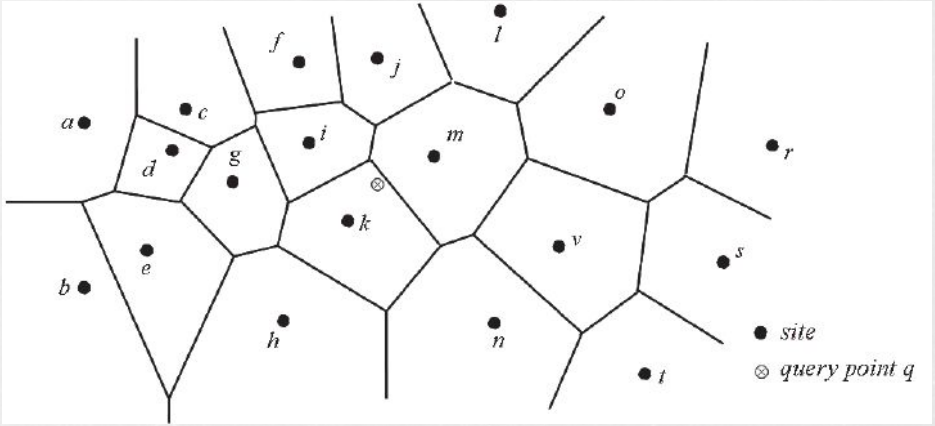
Nearest Neighbour

For nearest neighbour classifier our hypothesis is $h(x) = y^*$, where y^* is the label of our closest point. ($x^* = \text{argmin } d(x, x')$)

It is like memorization and then comparing what you memorized with a new input.

How to find the distance d(x, x'):

- By any distance metric where one of which is the euclidean distance.



Nearest Neighbour

- Example of distance Metrics
  1) Euclidean Distance

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

  2) Manhattan Distance

$$d(p, q) = \sum_{i=1}^{n} (q_i - p_i)$$

Nearest Neighbour

$$= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 2 \\ 3 & 5 & 3 \end{bmatrix}$$

Euc = 7

$$= \begin{bmatrix} 30 & 10 & 34 \\ 14 & 45 & 26 \\ 37 & 55 & 32 \end{bmatrix}$$

Euc = 102

Nearest Neighbour

NN is unstable due to noise

- In the diagram to the right, you can see that the presence of a green dot in the middle of a red region affects the output of the algorithm in this area, where it is most likely red. This illustrates how NN can be unstable due to noise.
- What is the solution?

03

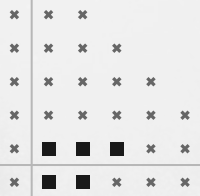
K Nearest Neighbours

KNN

- KNN is a more robust version of the normal NN algorithm
- Instead of just getting the label of one neighbour you get the label of multiple neighbours and output the most repeated label “mode”
- Choosing the value of K is a challenge. Why?
  - If K is too small the model will be tolerant to noise like mentioned in the previous slide.
  - If K is too large the model will always predict the most repeated label in the training dataset.

How to choose the K in KNN?

- This is not an easy question to answer.
- This will introduce us to 2 new topic one of which is the hyperparameters:
  - A Hyperparameter is a setting for the model that is set by the engineer.
  - They help us to guide the learning process and can affect how well a model performs.
  - K & the distance metric “like euclidean distance” are considered to be a hyperparameter for KNN

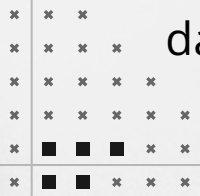


04

Data Splits

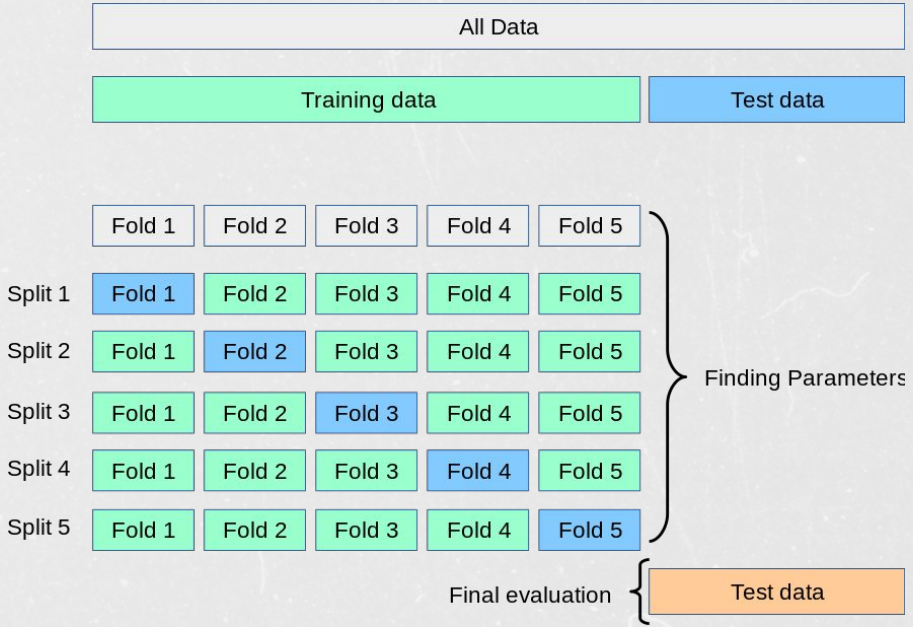
Train, Val, Test Split

- An important rule in machine learning is that you must never train and test on the same data.
- That is like solving the final exam a day before its timing, which makes us not sure that our model did generalize well or not.
- We divide our data into 3 sets:
  1) Training Set: which is used to train the model.
     a) Contains the majority of data. 70-80% for ML and upto 99% for DL
  2) Validation Set: which is used to optimize the hyperparameters and to compare different sets of hyperparameters
  3) Test Set: measure performance at the end of the model training process.
- If validation set is so small it is better to use k fold cross validation to make the data representative for future data.

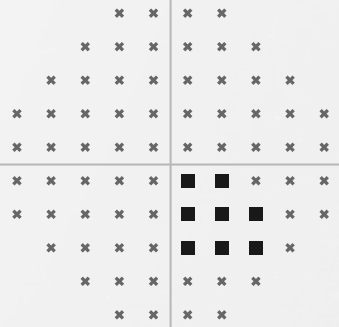


K-Fold Cross Validation

- We split the data set into 2 parts:
  - Training data
  - Test data
- For the training data we split it into k folds.
  - Run k experiment each time we train on k-1 folds and validate on 1 fold.
  - Compute the average accuracy (or any other metric more on that later)



05



Back to KNN

KNN

- KNN is called a lazy learner.
- That is because when it get the training data, it doesn't lear and make a model, it just stores the data and use this data in its predictions.
- So It is not good with large datasets, as it needs to make decision only at run time. So if the dataset is large a lot of processing is needed which will impace the performance of our algorithms
- Advantages of KNN:
  1) No Training Period
  2) New data can be added seamlessly
  3) Very easy to implement (does not have many hyper parameters.)

KNN

- KNN Disadvantages:
  1) Does not work well with large dataset
  2) Does not work well with high dimensions “imagine high quality images and how it will be hard to calculate the distance between”
  3) Sensitive to noisy data, missing values and outliers.
  4) Imbalanced data causes problems
  5) K-NN needs homogeneous features “data with different scales will not work”

Other Types of KNN

- Weighted KNN
  - Gives lower weights to further neighbours than nearer ones.
  - It adapts better than normal knn
  - The weighting function is proportional to the inverse of the distance
  - The output label is the label that have the biggest weights
- Regression with KNN
  - Instead of getting the mode of the neighbours we calculate the average of the neighbours

Sources

- https://www.youtube.com/watch?v=5AXF14_OCNE&list=PLdAoL1zKcqTW-uzoSVBNEecKHsnug_M0k&index=2
- https://www.youtube.com/playlist?list=PLblh5JKOoLUICTaGLRoHQDuF_7q2GfuJF
- https://scikit-learn.org/stable/modules/cross_validation.html

THANK YOU!