Is it Supervised/Unsupervised/Reinforcement learning?

• What does the algorithm do?

• In which situations will it be most useful?

• (Optional) Can you find any examples of where this algorithm has been used

**Bayes Theorem**

It a supervised learning algorithm.

A classifier is a machine learning model that is used to discriminate different objects based on certain features.

A Naive Bayes classifier is a probabilistic machine learning model that’s used for classification task. The crux of the classifier is based on the Bayes theorem.

Graphical user interface, text

Description automatically generated with medium confidence

Using Bayes theorem, we can find the probability of **A** happening, given that **B** has occurred. Here, **B** is the evidence and **A** is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other. Hence it is called naive.

Types of Naive Bayes Classifier:

Multinomial Naive Bayes:

This is mostly used for document classification problem, i.e whether a document belongs to the category of sports, politics, technology etc. The features/predictors used by the classifier are the frequency of the words present in the document.

Bernoulli Naive Bayes:

This is similar to the multinomial naive bayes but the predictors are boolean variables. The parameters that we use to predict the class variable take up only values yes or no, for example if a word occurs in the text or not.

**Gaussian Naive Bayes:**

When the predictors take up a continuous value and are not discrete, we assume that these values are sampled from a gaussian distribution.

**Summary for Bayes Theorem**

Naive Bayes algorithms are mostly used in sentiment analysis, spam filtering, recommendation systems etc. They are fast and easy to implement but their biggest disadvantage is that the requirement of predictors to be independent. In most of the real-life cases, the predictors are dependent, this hinders the performance of the classifier.

KNN (K – Nearest Neighbours)

K Nearest Neighbour is a simple algorithm that stores all the available cases and classifies the new data or case based on a similarity measure. It is mostly used to classifies a data point based on how its neighbours are classified.

‘k’ in KNN is a parameter that refers to the number of nearest neighbours to include in the majority of the voting process.

Using an example of analysis two types of wine. Two chemical components called Chemical A and Chemical B. Consider a measurement of Chemical A vs Chemical B level with two data points, Red and White wines. They have tested and where then fall on that graph based on how much Chemical A and how much Chemical B chemical content present in the wines.

So, we need to find out what the neighbours are in this case. Let’s say k = 5 and the new data point is classified by the majority of votes from its five neighbours and the new point would be classified as red since four out of five neighbours are red.

k’ in KNN algorithm is based on feature similarity choosing the right value of K is a process called parameter tuning and is important for better accuracy. Finding the value of k is not easy.

1. There is no structured method to find the best value for “K”. We need to find out with various values by trial and error and assuming that training data is unknown.
2. Choosing smaller values for K can be noisy and will have a higher influence on the result.

3) Larger values of K will have smoother decision boundaries which mean lower variance but increased bias. Also, computationally expensive.

Another way to choose K is though cross-validation. One way to select the cross-validation dataset from the training dataset. Take the small portion from the training dataset and call it a validation dataset, and then use the same to evaluate different possible values of K. This way we are going to predict the label for every instance in the validation set using with K equals to 1, K equals to 2, K equals to 3.. and then we look at what value of K gives us the best performance on the validation set and then we can take that value and use that as the final setting of our algorithm, so we are minimizing the validation error.

**How does KNN Algorithm works?**

In the classification setting, the K-nearest neighbour algorithm essentially boils down to forming a majority vote between the K most similar instances to a given “unseen” observation. Similarity is defined according to a distance metric between two data points. A popular one is the Euclidean distance method.

Text

Description automatically generated with medium confidence

# Pros of KNN

1. Simple to implement.
2. Flexible to feature/distance choices.
3. Naturally handles multi-class cases.
4. Can do well in practice with enough representative data.

# Cons of KNN

1. Need to determine the value of parameter K (number of nearest neighbours)
2. Computation cost is quite high because we need to compute the distance of each query instance to all training samples.
3. Storage of data
4. Must know we have a meaningful distance function.