# KNN Tutorial

## Introduction to K-Nearest Neighbors(KNN):

* Definition: KNN is a supervised machine learning algorithm used for classification and regression tasks.
* Its Place in Machine Learning: KNN belongs to the family of instance-based learning algorithms, where predictions are made based on the similarity of new data points to existing data points in the training set. It doesn’t assumptions about the data distribution.

A diagram of a group of data

Description automatically generated with medium confidence

## Theory behind KNN:

* The core idea behind KNN can be summarized by the phrase "birds of a feather flock together."
* Similar data points tend to have similar characteristics.
* KNN works on the principle of similarity, where it classifies a new data point by finding the majority class among its K nearest neighbors in the feature space.

## Scenarios and Effectiveness:

* KNN is effective in scenarios where data is nonlinear and lacks explicit patterns.
* Computer vision: Object recognition, facial recognition
* Bioinformatics: Genetic data pattern identification, disease detection
* Recommendation systems: Collaborative filtering
* It's suitable for both classification and regression tasks.
* It performs well with small to medium-sized datasets and when the classes are well-separated.

## Advantages of KNN:

* Simplicity and interpretability: KNN is easy to understand and implement.
* Non-parametric: KNN makes no assumptions about the underlying data distribution.

## Limitations and Challenges:

* Sensitivity to K: The algorithm is sensitive to the choice of k.
* Computational Intensity: It can be computationally expensive, especially with large datasets, as it requires calculating distances to all data points.
* Curse of Dimensionality: Performance may degrade with high-dimensional data due to the increased sparsity of the feature space.

## Step-by-Step Explanation of KNN Algorithm:

1. Define the number of neighbors (K).

A diagram of a diagram

Description automatically generated

we will choose the k=5.

1. Calculate the distance between the new data point and all points in the training set.

A diagram of a line graph

Description automatically generated

1. Sort the distances and identify the K nearest neighbors.

A diagram of a new data point

Description automatically generated

By computing the Euclidean distance, we identified the closest neighbors: three in category A and two in category B.

1. Use the majority class among the K nearest neighbors as the predicted class for classification tasks.

Since the three closest neighbors are from category A, it suggests that this new data point belongs to category A.

## Effect of K on Bias-Variance Trade-off:

* Small K values lead to low bias and high variance, resulting in complex decision boundaries. It is prone to overfitting.
* Large K values lead to high bias and low variance, resulting in smoother decision boundaries. It is prone to underfitting.
* Selection of K involves balancing bias and variance to achieve optimal model performance.

## Selecting K:

* Rule of thumb: k = sqrt(n), where n is the number of training examples.
* Cross-validation: Use techniques like k-fold cross-validation to find the optimal K value. Follow below steps:
* Split Data: Divide your data into smaller groups called "folds".
* Test Different K Values: Try different values of K for your model.
* Evaluate Performance: Train the model with each K value using most of the data and evaluate how well it predicts on the remaining part.
* Average Results: Calculate the average performance across all folds for each K value.
* Choose Best K: Pick the K value that gives the best average performance.
* Error Rate vs. K Plot: Plot error rate and K, then choose the K with the lowest error rate.

## Effects of Dataset Size, Noise, and Dimensionality on choosing K:

* **Dataset Size**:
* Larger Datasets: a larger value of K may be preferable to reduce variance and improve generalization. Additionally, larger datasets may require more computational resources for KNN calculations, prompting consideration of efficiency when selecting K.
* Smaller Datasets: smaller values of K may help capture finer patterns and avoid overfitting.
* **Noise**: selecting a larger value of K can help reduce the influence of noisy data and stabilize predictions, but it also carries bias and overlooking small but significant patterns in the data.
* **Dimensionality**: High-dimensional data may benefit from smaller values of K to mitigate the curse of dimensionality.

## Reference

Christopher, A. (2021, February 2). K-Nearest Neighbor. Medium. <https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4>

Band, A. (2020, May 23). How to find the optimal value of K in KNN? Towards Data Science. https://towardsdatascience.com/how-to-find-the-optimal-value-of-k-in-knn-35d936e554eb