Hackathon

where quantum computing meets information theory

Quantum K-Nearest Neighbors

Srijita Dutta

June 17, 2025

ISIT JUNE 22-27 ANN ARBOR MICHIGAN

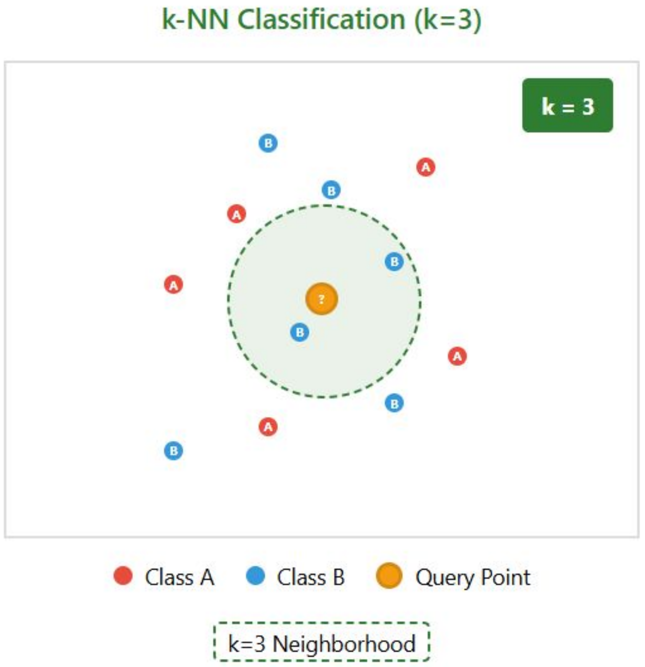
What is k-nearest Neighbors (k-NN)?

k-NN is a supervised machine learning algorithm that classifies data points based on the majority vote or average of their k nearest neighbors in the training dataset.

- k determines how many neighbors to consider
- Uses distance metrics to find nearest points
- Classification: majority vote among neighbors
- Regression: average of neighbor values
- Simple yet effective for many problems

Lazy Learner Algorithm

k-NN doesn't learn during training. It stores the entire dataset and performs calculations only when making predictions, earning it the "lazy learner" designation.



How to choose the value of k for KNN Algorithm?

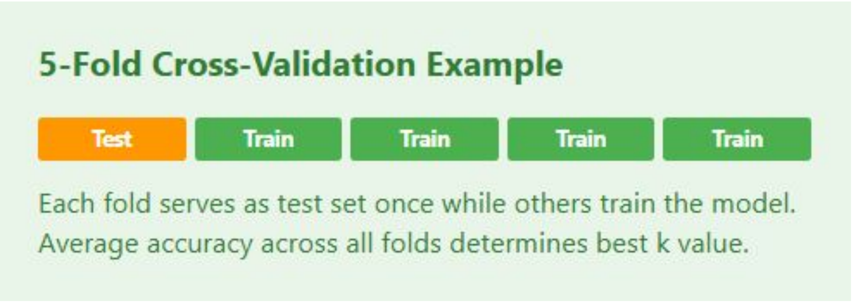
The choice of k significantly affects model performance. It balances between capturing local patterns and maintaining stability against noise and outliers.

Small k is more sensitive to noise and thus has a risk of **overfitting**, while a **Large k** is more stable but may miss patterns (**underfitting**)

Selection Methods

Elbow Method Visualization

Cross Validation Method

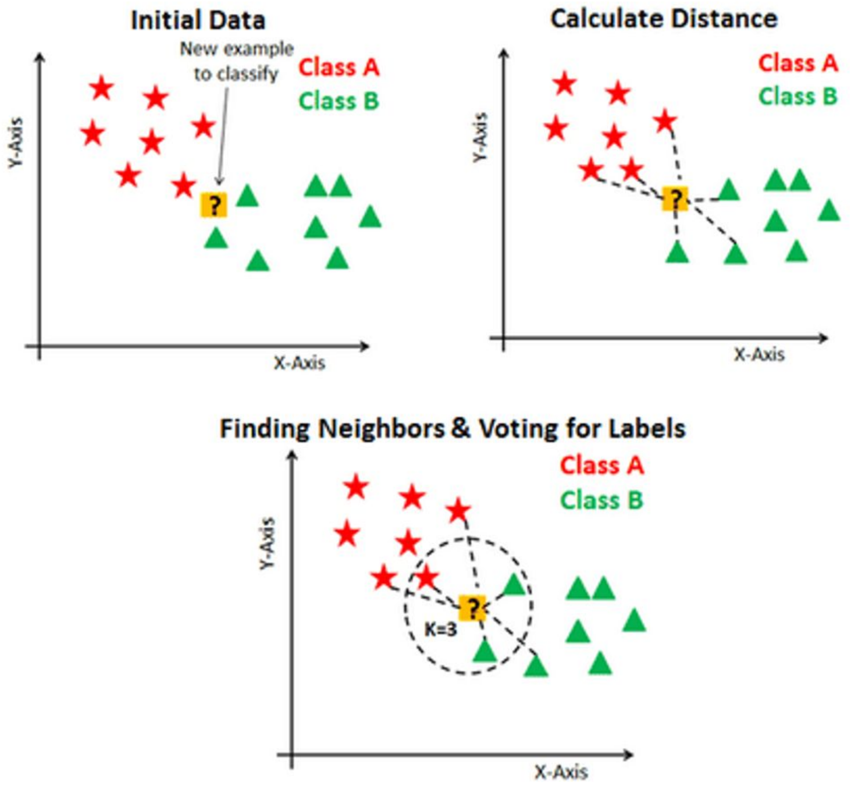


Best Practices

- Start with **k = √n** (where n is training samples) and test multiple odd values around this estimate
- Consider dataset size and noise level
- Validate on separate test set

Classical k-NN Process

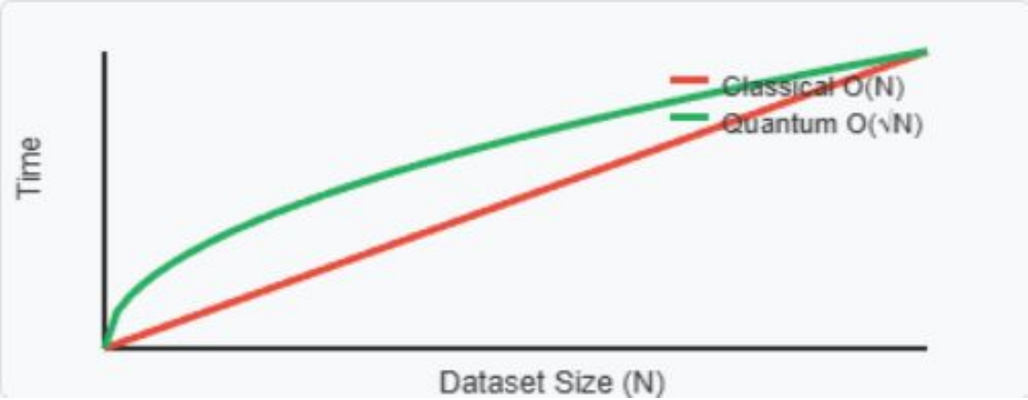
- **Step 1:** Choose an optimal value of K.
- **Step 2**: Calculate the distance between the query point and each point in the dataset using traditional distance metrics like:
  a. **Euclidean distance = $\sqrt{\sum(x_2-x_1)^2}$**
  b. **Manhattan distance = $\sum |(x_2-x_1)|$**
- **Step 3**: Sort the distances and select the k(=3 in this case) nearest neighbors.
- **Step 4**: Assign a class label (or regression value) based on the nearest neighbors.

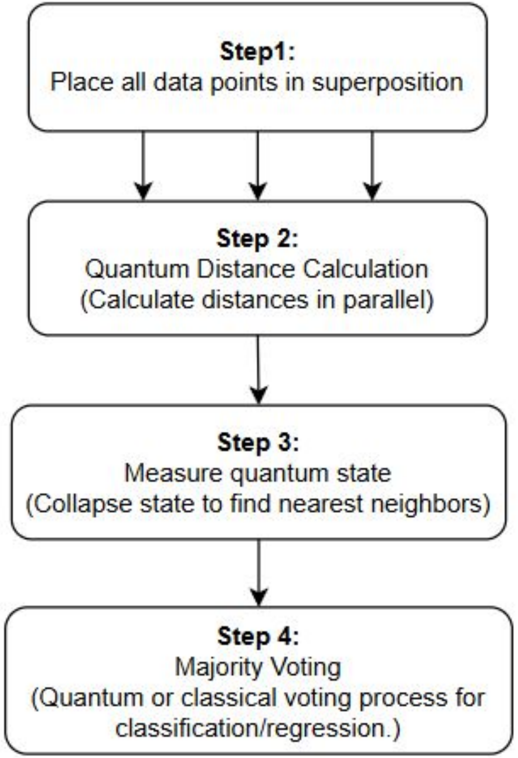


https://github.com/qosf/monthly-challenges/blob/main/challenge-2021.06-jun/challenge-2021.06-jun.ipynb

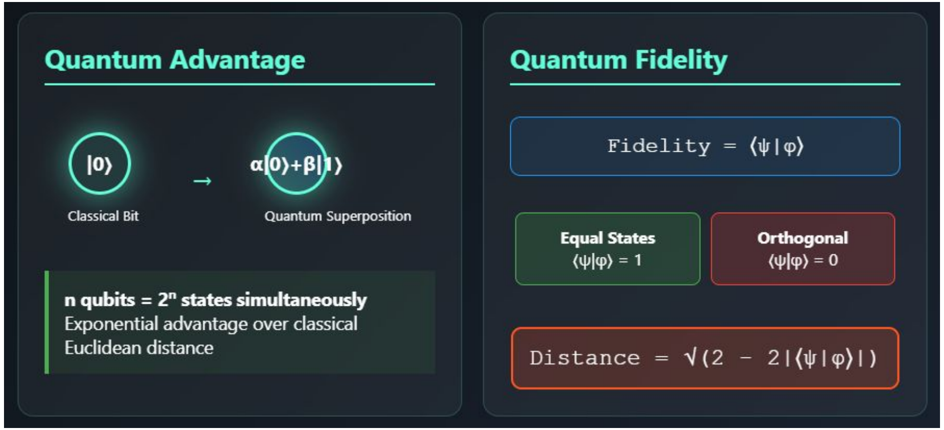
Quantum k-NN Process

- **High-Dimensional Advantage:** Quantum k-NN excels in high-dimensional spaces
- **Quantum Minimization Algorithm:** Uses Grover's search to find nearest neighbors efficiently
- **Speedup Achievement:** Reduces classical O(n) to quantum O(√n) time complexity





Distance calculation in Quantum k-nn

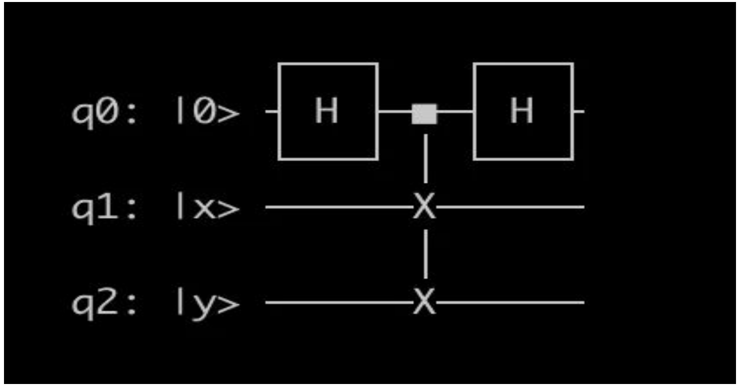


In quantum, distance is measured using **fidelity.** The fidelity of two quantum states **|ψ⟩and |φ⟩**is a measure of their similarity. Fidelity is the same as the inner product of two states and is represented by $F = |\langle\psi|\varphi\rangle|^2$

- If the two quantum states are equal then F evaluates to 1.
- If the two states are orthogonal then F evaluates to 0.

Using this method to compare two quantum states distance can be described as $\sqrt{2-2|\langle\psi|\varphi\rangle}$This algorithm uses an auxiliary qubit set to |0⟩that gets passed through a **Hadamard gate.** Then the two vectors |x⟩and |y⟩are switched using the **Controlled-SWAP gate** otherwise known as the **Fredkin gate**.

The **circuit for SWAP test** looks something like this:



Why Quantum k-NN?

Quantum k-NN can achieve a **quadratic** speedup using algorithms like Quantum Approximate Nearest Neighbor (QANN), hence, it can give faster solutions to certain problems, particularly those that involve large datasets or complex data structures.

Key Advantages:

- **Speed:** Faster processing using quantum algorithms.
- **Efficiency:** Quantum features help reduce computational complexity.
- **Better with large data:** Quantum systems can handle high-dimensional data better than classical systems.

Applications of Quantum k-NN:

Quantum k-NN can work better with high-dimensional data, which is often found in fields like:

- Image recognition
- Natural language processing
- Bioinformatics (protein folding, gene classification)

References and resources

[1] R. S. Chitambar and G. Gour, “Quantum Principal Component Analysis Using Hybrid Classical-Quantum Methods,” *arXiv preprint arXiv:2505.06441*, May 2025. [Online]. Available: https://arxiv.org/abs/2505.06441

[2] M. Schuld and N. Killoran, “Quantum Machine Learning in Feature Hilbert Spaces,” *arXiv preprint arXiv:2003.09187*, Mar. 2020. [Online]. Available: https://arxiv.org/abs/2003.09187

[3] A. Y. Guerrero-Estrada, L. F. Quezada, and G. H. Sun, “Benchmarking quantum versions of the kNN algorithm with a metric based on amplitude-encoded features,” *Scientific Reports*, vol. 14, no. 1, p. 16697, 2024. [Online]. Available: https://doi.org/10.1038/s41598-024-67392-0

[4] J. Biamonte, P. Wittek, N. Pancotti *et al.*, “Quantum machine learning,” *Nature*, vol. 549, pp. 195–202, 2017. [Online]. Available: https://doi.org/10.1038/nature23474

[5] https://github.com/nagarx/Quantum-KNN-Classifier-using-Qiskit/tree/main