TECHNISCHE UNIVERSITÄT DARMSTADT

HCML Project
Product Quantization with KDTREE

Muhammet Tarık Çepi, Dinh Khanh Thi Vo

TECHNISCHE UNIVERSITÄT DARMSTADT

1 Product Quantization

2 Product Quantization with KDTREE

3 Results

4 Limitations and Future Works

TECHNISCHE UNIVERSITÄT DARMSTADT

Product Quantization

- Product quantization is a dimensionality reduction technique that can be used to speed up nearest neighbor search.
- It works by dividing the original feature space into a number of smaller sub-spaces, and then quantizing each sub-space separately.
- This results in a much smaller representation of the original feature vector, which can be used to quickly search for similar vectors.

Product Quantization

- Let’s go step by step
- First we divide the input embedding into x subspaces.(Figure 1)
- Then we run K-Means Clustering in each of the Subspaces (Figure 2)

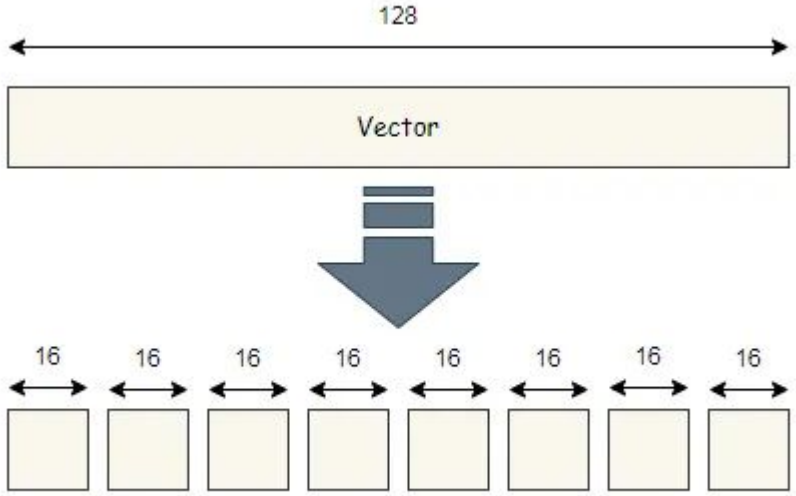


Figure 1

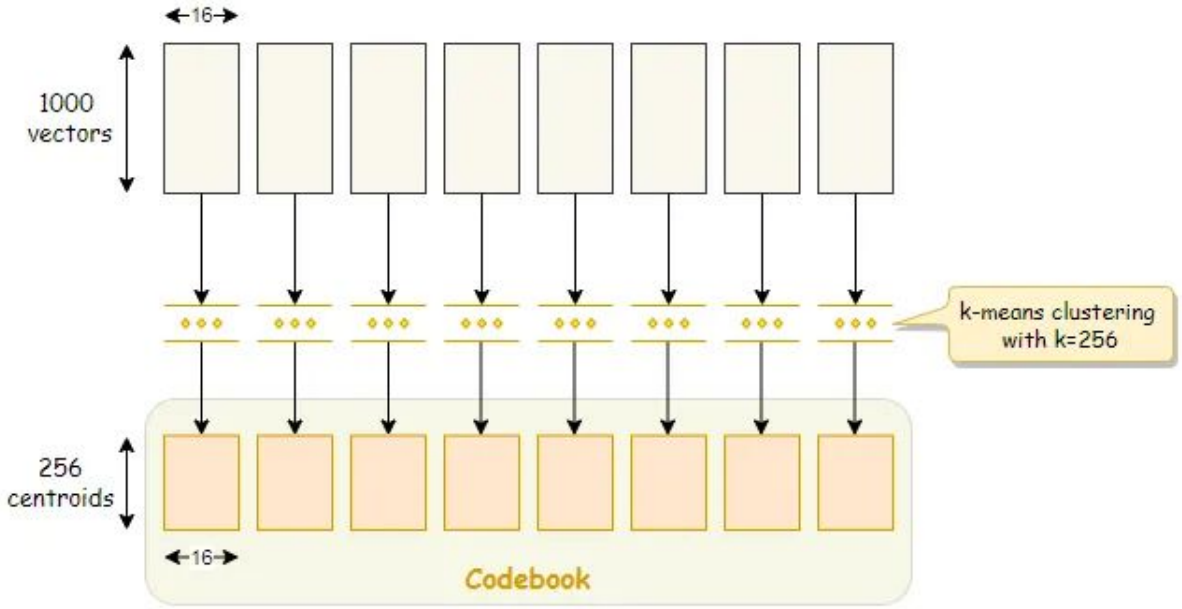
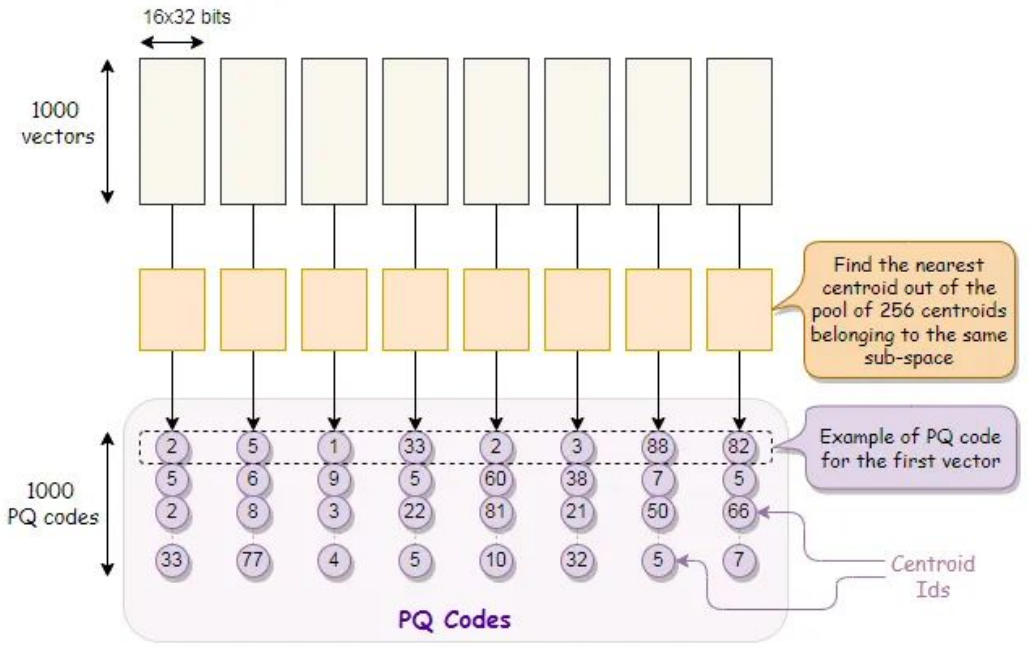


Figure 2

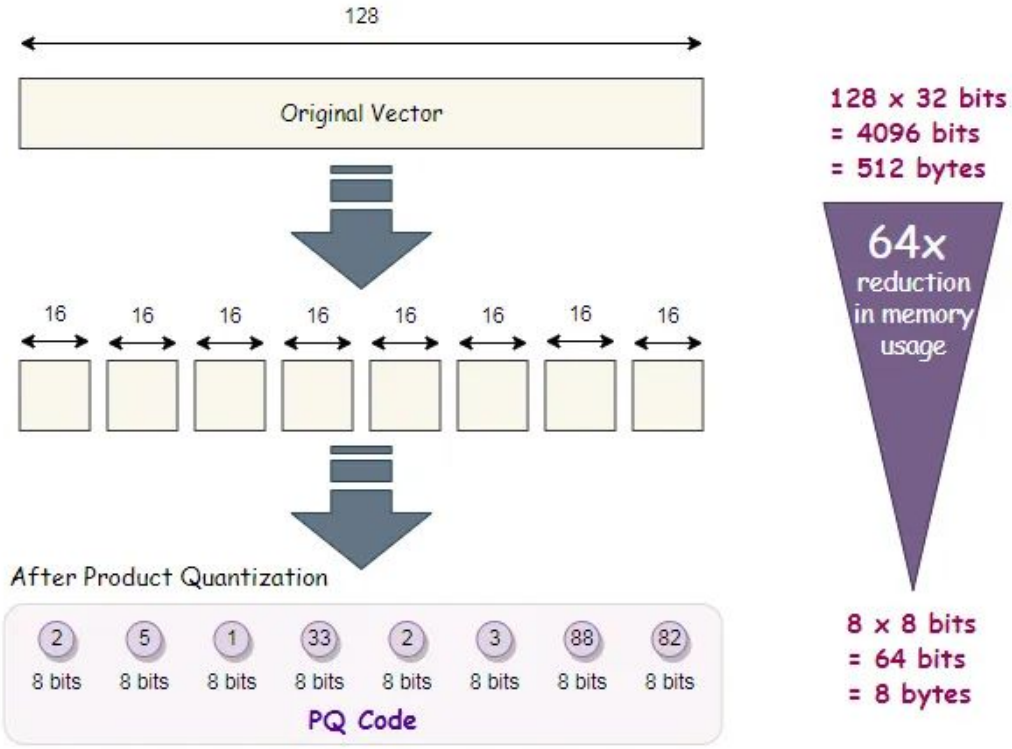
Product Quantization

- After the training each point in the subspace gets a distance to the centres of the clusters. The centers of the cluster are enumerated and called the centroids. It's an index of centers of the Clusters.
- Using the distance and centroids we create a so called PQ-Code for each embedding.PQ codes are basically centroid indexes for each subspace



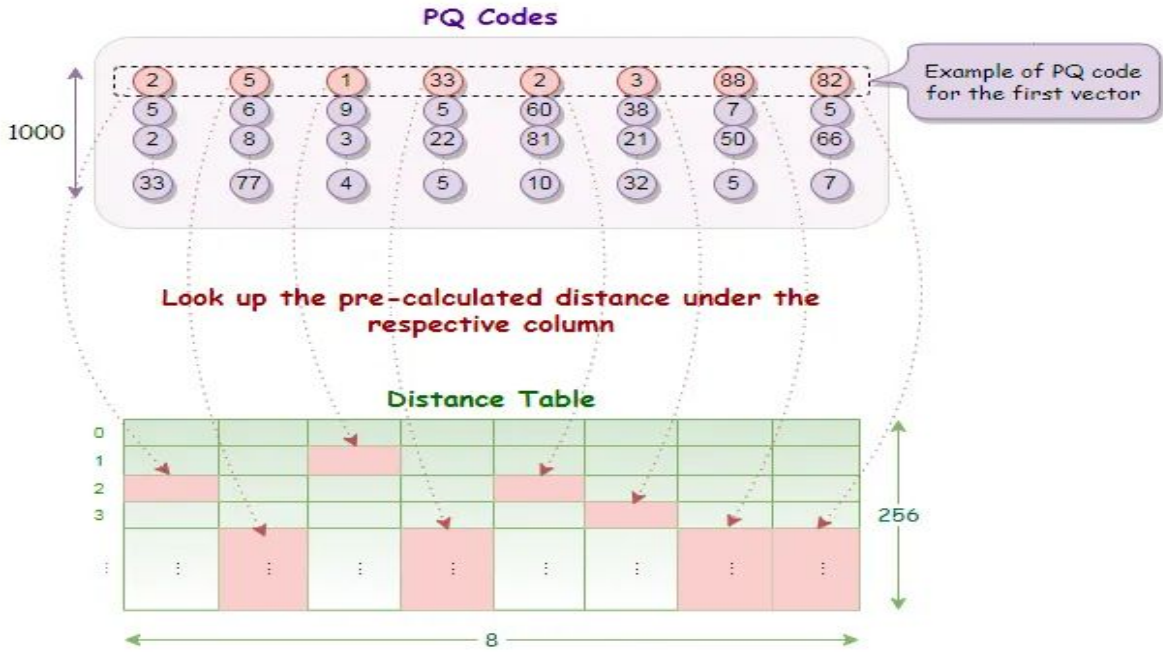
Product Quantization(Not 100%Sure)

- Memory Reduction:
  - On the figure we see an example.
  - Original data in our case is 1024*32 bits=32764 bits
  - We use 4 subspaces and 256 centroids
    - 4*16 bits=64 bits
    - So we reduce the usage by 512 times.



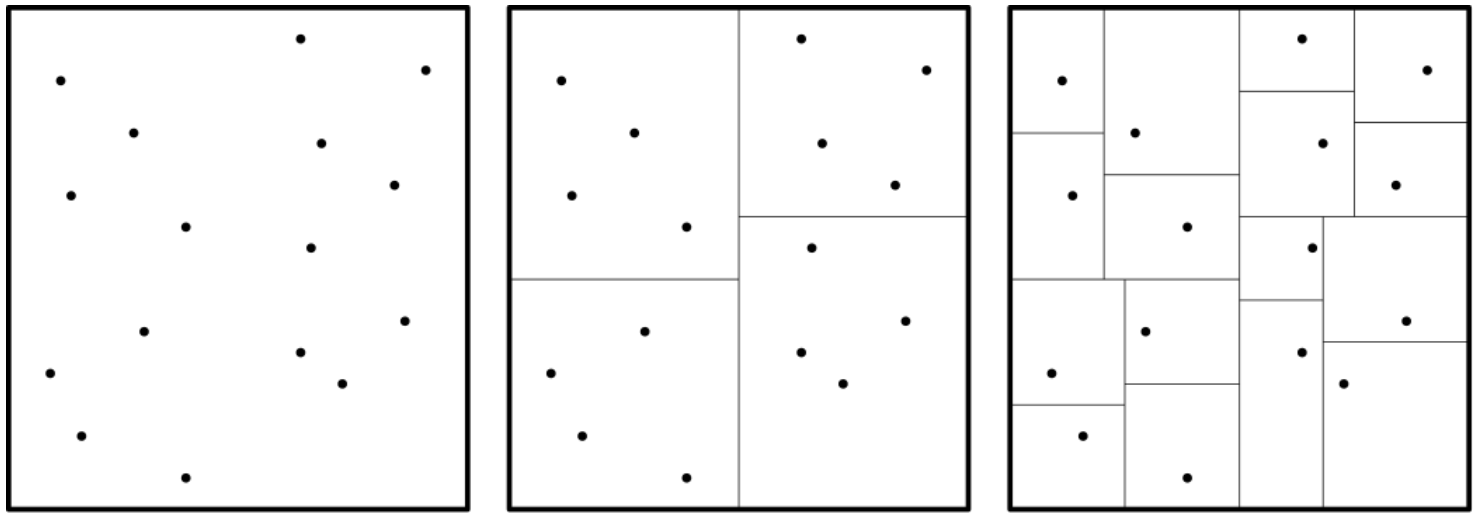
Product Quantization

- Searching
  - For searching we do same as training but with a difference. We do not calculate the PQ-Code, instead we calculate a distance table, which contains distance of the embedding to all centroids.
  - Then all Gallery PQ-Codes are used to calculate the distance. Remember, PQ-Codes are nothing but indexes. We use them as indexes in the distance table and sum up the distance.
  - We are looking for the PQ-Code with least distance



Product Quantization with KDTREE

- Product Quantization requires still a lot of computation and the penetration rate is still 100%
- To reduce the number of computation, one can use KDTree to get k numbers of PQ-Codes and compute distance.
- K-D Tree is a binary tree in which each node represents a k-dimensional point. It basically puts an hyperplane between data points on each split of tree

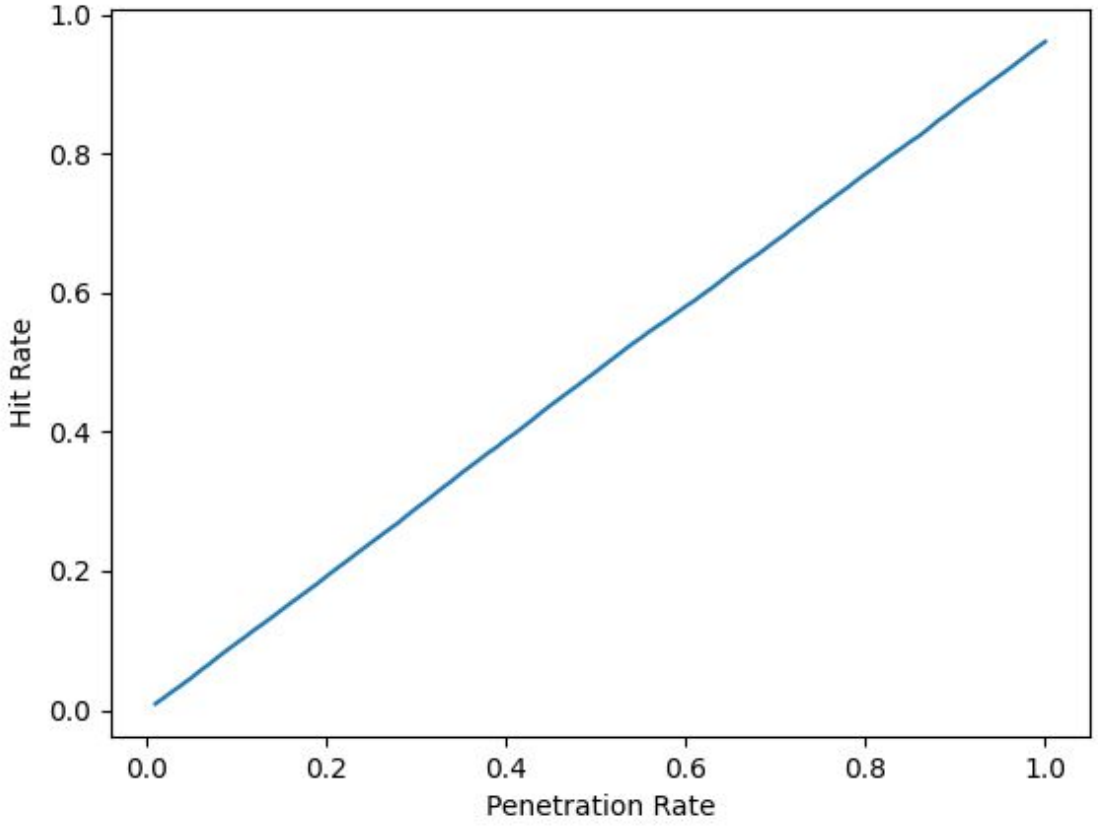


Evaluations

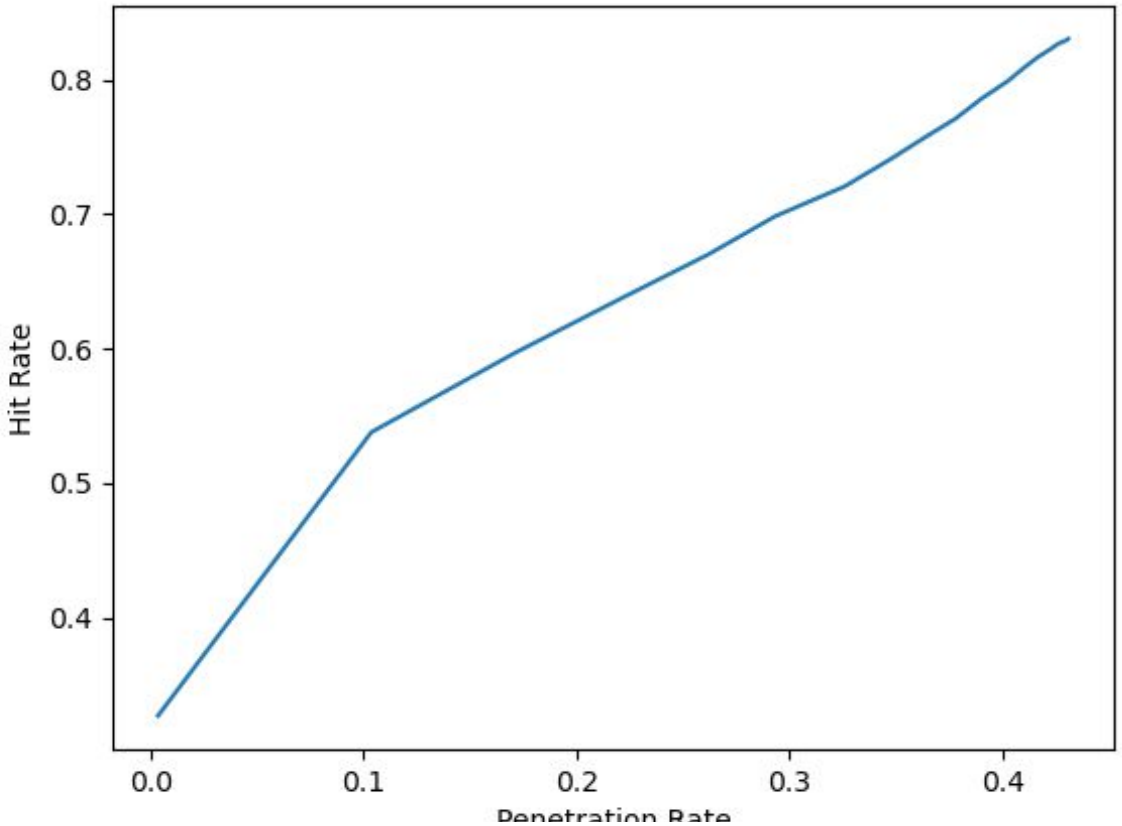
- Hit rate over penetration rate, computation time, processor
- Interpretations of penetration rate

TECHNISCHE UNIVERSITÄT DARMSTADT

Results - 1



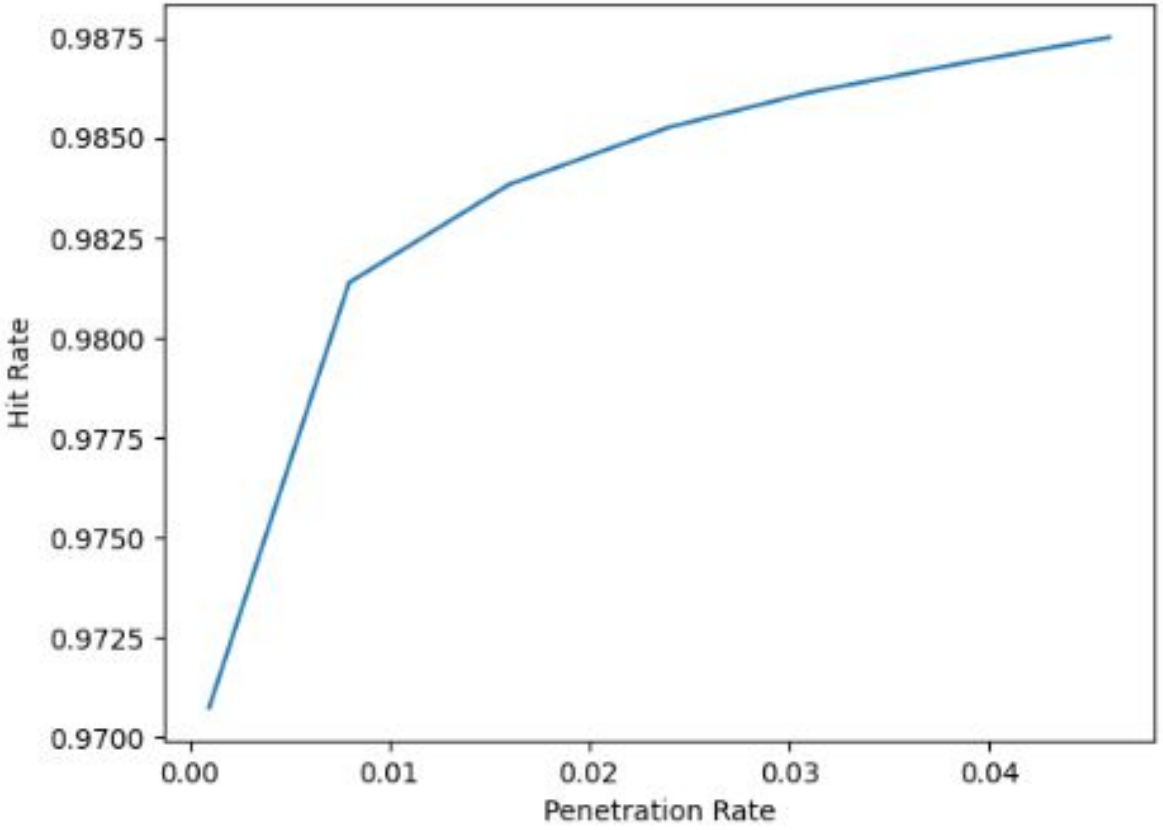
Random Indexing.



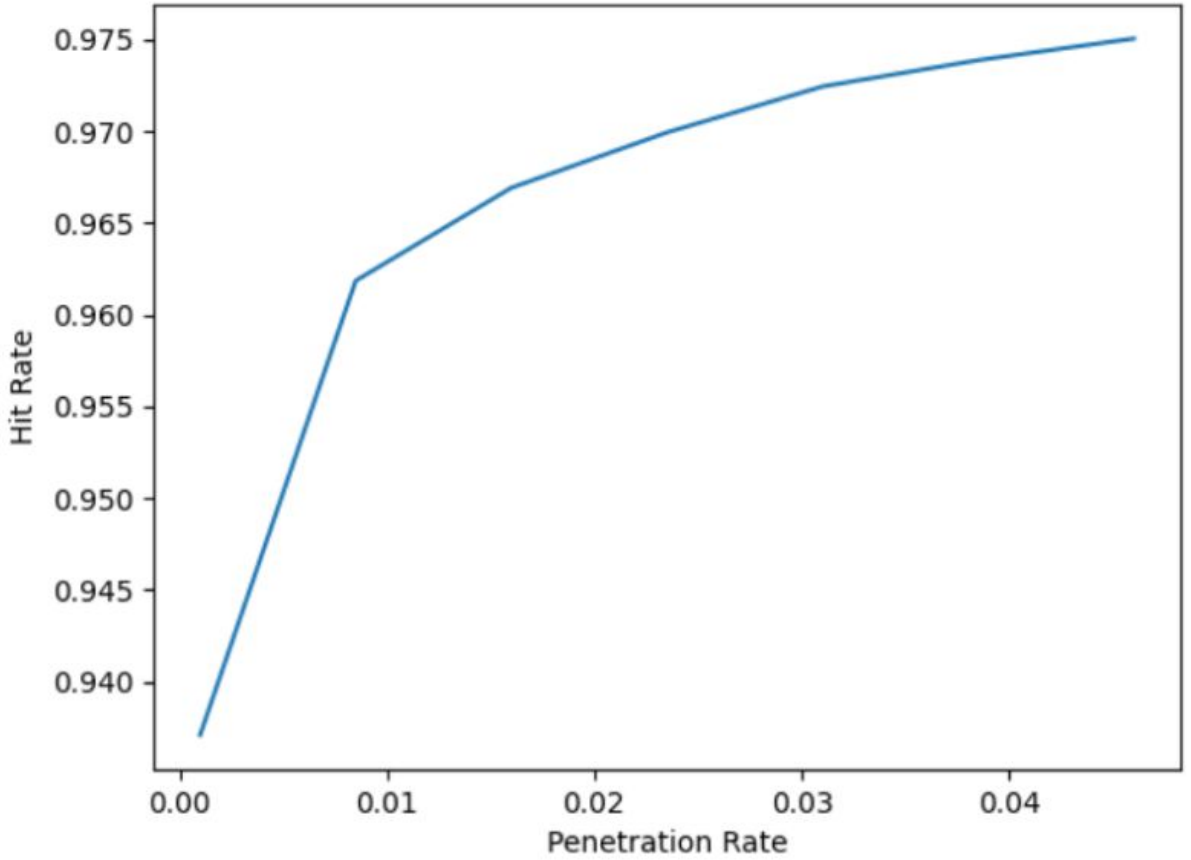
PQ with KDTREE

TECHNISCHE UNIVERSITÄT DARMSTADT

Results - 2



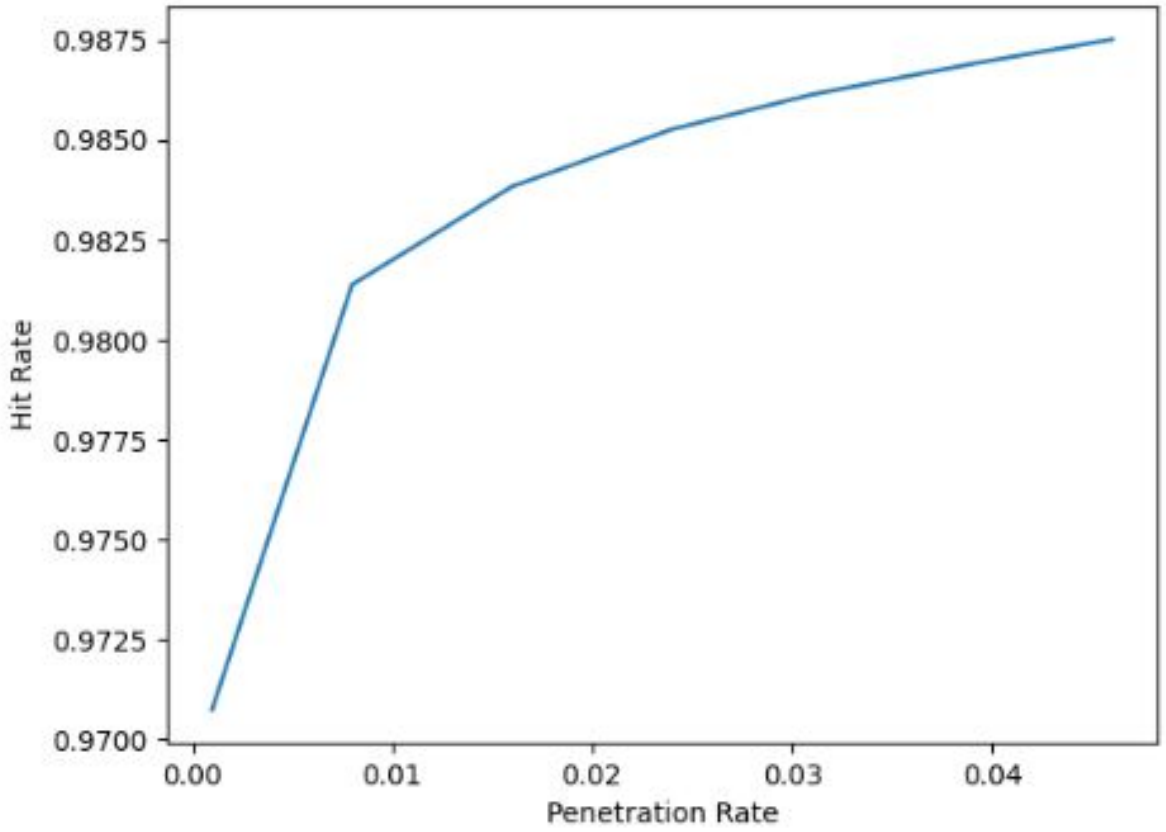
Baseline: 0.021s

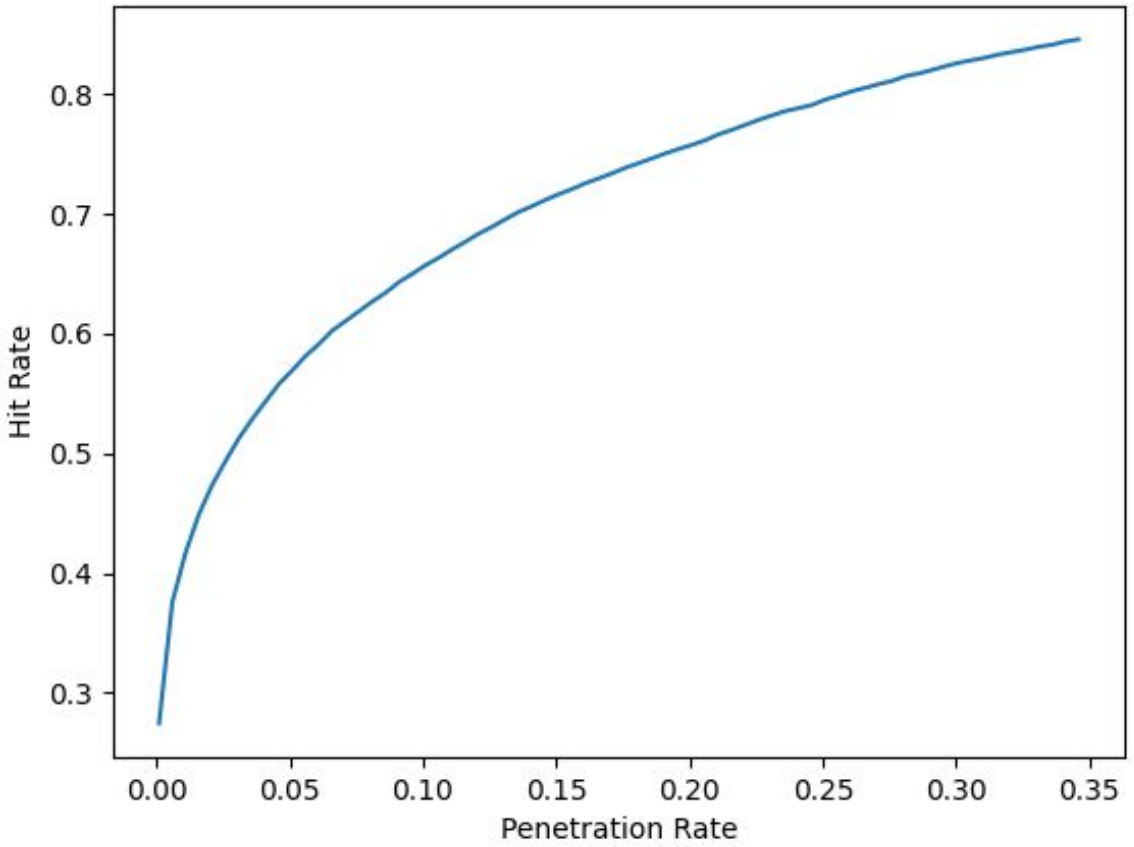


PQ (without KDTree): 0.015s

TECHNISCHE UNIVERSITÄT DARMSTADT

Results - 2

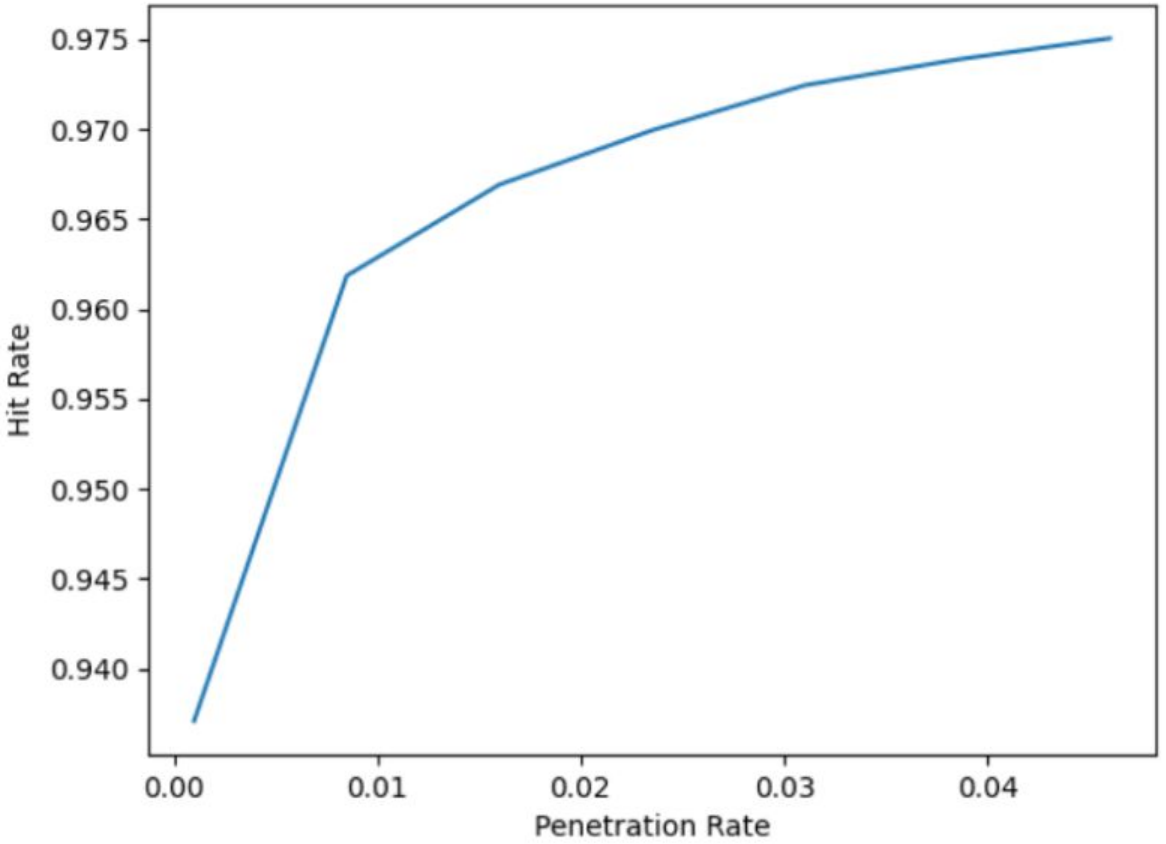




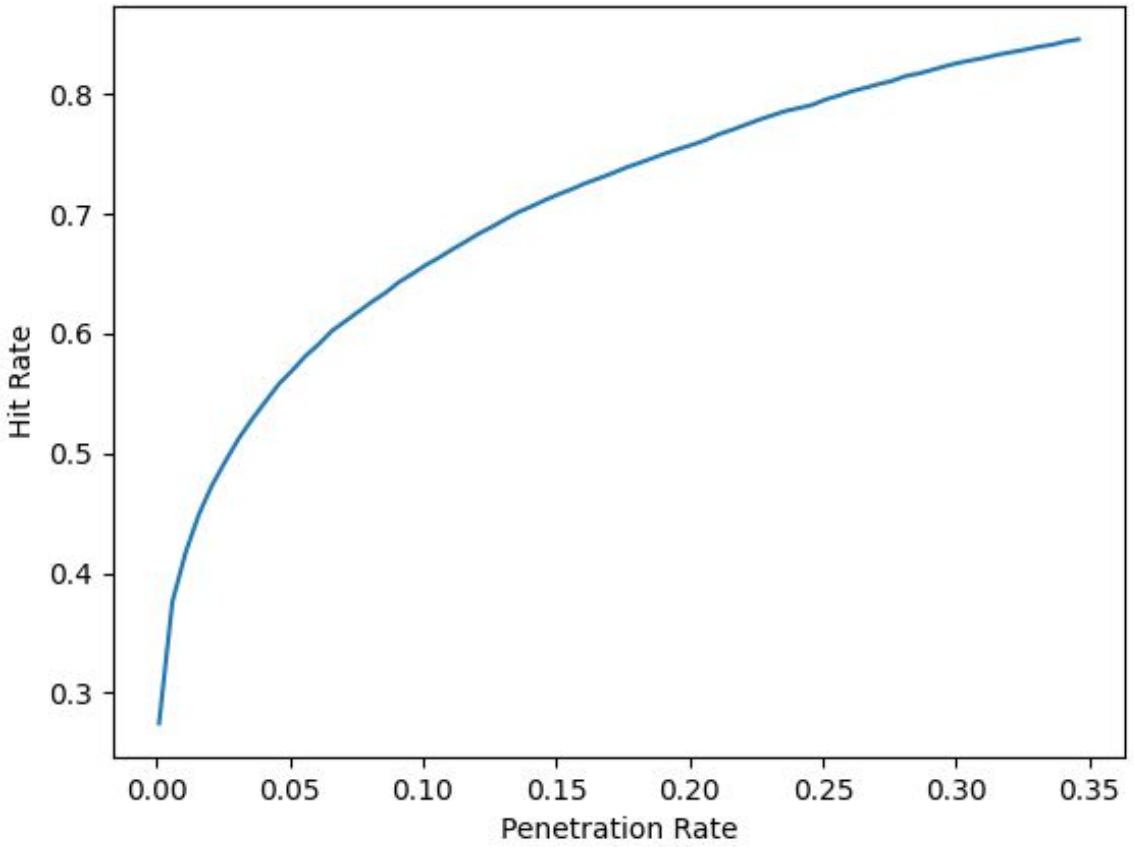
Baseline: 0.021s

PQ with KDTree: 0.0098

Results - 2



PQ (without KDTree): 0.015s



PQ with KDTree: 0.0098

Limitations and Future Works

- The Approach is:
  - Fast
  - Memory Aware
- But:
  - Drops the accuracy - Tradeoff
  - Because it’s using 2 ML algorithms
- Nevertheless
  - It can be used as a starting point.
  - Hyperparameters optimization can be a next step
  - With more intense research it can be useful.