


## Themenübersicht

- Freie Wahl einer Vektordatenbank bedeutet: Nutzung eines reinen Vektorindex (z.B. faiss) ist nicht erlaubt!
- Generative KI-Systeme dürfen eingesetzt werden. Der Einsatz **muss vollständig dokumentiert** werden z.B. im Anhang
  - Eingesetzte KI-Systeme (z.B. CoPilot, Perplexity)
  - Verwendungsweise des KI-Systems (z.B. CoPilot zur Generierung von Code, der überarbeitet wurde; Perplexity zur Generierung von Ideen für Kapitel 2)

RAG Data Engineering: pdf Umwandlung	<p>Beschreibung von Data Engineering in Zusammenhang mit RAG mit Schwerpunkt pdf Dokumente in Vektoren umwandeln und in Vektor-DB speichern. Minimum sind 50 Datensätze (Vektoren).</p> <p>Datenbank: freie Wahl einer Vektordatenbank Daten: pdf-Dokumente von <a href="https://arxiv.org/">https://arxiv.org/</a></p>
Embeddingmodelle im Vergleich	<p>Beschreibung und Vergleich von 5 verschiedenen Embeddingmodellen. Umwandlung von Daten in Vektoren und Vergleich der Ergebnisse (z.B. Recall). Minimum sind 50 Datensätze (Vektoren).</p> <p>Datenbank: freie Wahl einer Vektordatenbank Daten: Reviewtexte, z.B. <a href="https://www.kaggle.com/datasets/dongrelaxman/amazon-reviews-dataset">https://www.kaggle.com/datasets/dongrelaxman/amazon-reviews-dataset</a></p>
RAG Data Engineering: Chunking	<p>Beschreibung von Data Engineering mit Schwerpunkt Chunking (siehe z.B. <a href="https://glaforge.dev/talks/2024/10/14/advanced-rag-techniques/">https://glaforge.dev/talks/2024/10/14/advanced-rag-techniques/</a>) in Zusammenhang mit RAG. Speicherung der Daten in einer Vektordatenbank. Minimum sind 50 Datensätze (Vektoren).</p> <p>Datenbank: freie Wahl einer Vektordatenbank Daten: Reviewtexte, z.B. <a href="https://www.kaggle.com/datasets/dongrelaxman/amazon-reviews-dataset">https://www.kaggle.com/datasets/dongrelaxman/amazon-reviews-dataset</a></p>
RAG Data Engineering: Audio Transformation	<p>Beschreibung von Data Engineering im Zusammenhang mit RAG, mit Schwerpunkt auf Audio-Daten. Umwandlung von Audio-Daten in Vektoren und Speicherung in einer Vektor-</p>

	<p>Datenbank. Minimum sind 10 Datensätze (Vektoren).</p> <p>Datenbank: freie Wahl einer Vektordatenbank Daten: Transcripts von podcasts oder Videos, z.B. <a href="https://www.youtube.com/@DevoxxForever/videos">https://www.youtube.com/@DevoxxForever/videos</a></p>
GraphRAG Data Engineering	<p>Beschreibung von GraphRAG und Umsetzung eines Beispiels.</p> <p>Datenbank: freie Wahl einer Vektordatenbank, z.B. neo4j Daten: freie Auswahl</p>
RAG Data Engineering mit LlamaIndex und LangChain	<p>Beispielhafte Umsetzung einer Data Engineering Pipeline mit Texten mittels LlamaIndex und LangChain. Vergleich der beiden Pakete. Minimum sind 50 Datensätze (Vektoren).</p> <p>Datenbank: freie Wahl einer Vektordatenbank Daten: Reviewtexte, z.B. <a href="https://www.kaggle.com/datasets/dongrelaxman/amazon-reviews-dataset">https://www.kaggle.com/datasets/dongrelaxman/amazon-reviews-dataset</a></p>
HNSW in Vektordatenbanken	<p>Detaillierte Beschreibung von HNSW am Beispiel pgvector und Umsetzung eines Beispiels mit Vergleich exakter und Ähnlichkeitssuche. Minimum ist ein 6-dimensionaler Vektor sowie mindestens 50 Datensätze (Vektoren).</p> <p>Datenbank: freie Wahl einer Vektordatenbank Daten: Eigenschaften von Musik wie Tanzbarkeit, Lautstärke, z.B. <a href="https://www.kaggle.com/datasets/bricevergnou/spotify-recommendation">https://www.kaggle.com/datasets/bricevergnou/spotify-recommendation</a></p>
IVF in Vektordatenbanken	<p>Detaillierte Beschreibung von IVF am Beispiel pgvector und Umsetzung eines Beispiels mit Vergleich exakter und Ähnlichkeitssuche. Minimum ist ein 6-dimensionaler Vektor sowie mindestens 50 Datensätze (Vektoren).</p> <p>Datenbank: freie Wahl einer Vektordatenbank Daten: Eigenschaften von Musik wie Tanzbarkeit, Lautstärke, z.B. <a href="https://www.kaggle.com/datasets/bricevergnou/spotify-recommendation">https://www.kaggle.com/datasets/bricevergnou/spotify-recommendation</a></p>
RAG Data Engineering von Bilddaten	<p>Beschreibung von RAG Data Engineering von Bilddaten mit Hilfe der unten aufgeführten Daten. Minimum sind 10 Datensätze (Vektoren).</p> <p>Datenbank: freie Wahl einer Vektordatenbank Daten: Gesichter, z.B. <a href="https://www.kaggle.com/datasets/faresmostafa/smiles-datasets">https://www.kaggle.com/datasets/faresmostafa/smiles-datasets</a></p>

RAG Data Engineering von wikipedia-Artikel	<p>Beschreibung von RAG Data Engineering von Texten sowie Einlesen von wikipedia-Artikeln mittels Api/Scrapping Paket Minimum sind 10 Datensätze (Vektoren).</p> <p>Datenbank: freie Wahl einer Vektordatenbank Daten: Wikipedia-Artikel</p>
Visualisierung von Musikdaten mittels t-SNE und UMAP am Beispiel pgvector	<p>Visualisieren Sie Vektoren mittels t-SNE (t-distributed Stochastic Neighbor Embedding) und UMAP (Uniform Manifold Approximation and Projection). Erklären Sie dabei t-SNE / UMAP und zeigen Sie verschiedene Visualisierungen auf (z.B. Hervorhebung von Datensatz X in der Punktwolke). Minimum ist ein 6-dimensionaler Vektor sowie mindestens 50 Datensätze (Vektoren).</p> <p>Datenbank: freie Wahl einer Vektordatenbank Daten: Eigenschaften von Musik wie Tanzbarkeit, Lautstärke, z.B. <a href="https://www.kaggle.com/datasets/bricevergnou/spotify-recommendation">https://www.kaggle.com/datasets/bricevergnou/spotify-recommendation</a></p>
Visualisierung von Musikdaten mittels t-SNE und PCA am Beispiel pgvector	<p>Visualisieren Sie Vektoren mittels t-SNE (t-distributed Stochastic Neighbor Embedding) und PCA (Principal Component Analysis). Erklären Sie dabei t-SNE / PCA und zeigen Sie verschiedene Visualisierungen auf (z.B. Hervorhebung von Datensatz X in der Punktwolke). Minimum ist ein 6-dimensionaler Vektor sowie mindestens 50 Datensätze (Vektoren).</p> <p>Datenbank: freie Wahl einer Vektordatenbank Daten: Eigenschaften von Musik wie Tanzbarkeit, Lautstärke, z.B. <a href="https://www.kaggle.com/datasets/bricevergnou/spotify-recommendation">https://www.kaggle.com/datasets/bricevergnou/spotify-recommendation</a></p>
Evaluierung von RAG Ergebnissen mit Hilfe ausgewählter Metriken	<p>Beschreibung von Evaluierungsmetriken (mindestens 3) und Verprobung der Metriken. Minimum ist ein 6-dimensionaler Vektor sowie mindestens 50 Datensätze (Vektoren).</p> <p>Datenbank: freie Wahl einer Vektordatenbank Daten: Eigenschaften von Musik wie Tanzbarkeit, Lautstärke, z.B. <a href="https://www.kaggle.com/datasets/bricevergnou/spotify-recommendation">https://www.kaggle.com/datasets/bricevergnou/spotify-recommendation</a></p>
Evaluierung von RAG Ergebnissen mit Hilfe von Ragas	<p>Beschreibung von ragas und Verprobung von Metriken (mindestens 5). Minimum ist ein 6-dimensionaler Vektor sowie mindestens 50 Datensätze (Vektoren).</p> <p>Datenbank: freie Wahl einer Vektordatenbank Daten: Eigenschaften von Musik wie Tanzbarkeit, Lautstärke, z.B.</p>

	<a href="https://www.kaggle.com/datasets/bricevergnou/spotify-recommendation">https://www.kaggle.com/datasets/bricevergnou/spotify-recommendation</a>
Lakehouse mit DuckDB	<p>Beschreibung und Definition von Lakehouse. Umsetzung eines Beispiels mit DuckDB.</p> <p>Datenbank: DuckDB Daten: freie Auswahl – die verschiedenen Layer/Zonen sollen sich damit sinnvoll abbilden lassen</p>
Apache XTable	<p>Beschreibung von Apache XTable und beispielhafte Umsetzung als Abstrahierung von Delta Lake und Iceberg.</p> <p>Datenbank: (XTable) Daten: freie Auswahl</p>
Visualisierung von Finanzdaten	<p>Beschreibung von typischen Kennzahlen YTD, MTD sowie Beschreibung ausgewählter Gestaltungskriterien (mindestens 3) für die Visualisierung. Umsetzung dieser Kriterien in einem PowerBI Bericht anhand YTD, MTD und voriges Jahr.</p> <p>Visualisierung: PowerBI Desktop Daten: freie Wahl, auch Erzeugung von künstlichen Daten auf erlaubt</p>
Visualisierung von Daten nach IBCS	<p>Beschreibung von IBCS (International Business Communication Standards) für die Visualisierung. Umsetzung ausgewählter Kriterien in einem PowerBI Bericht.</p> <p>Visualisierung: PowerBI Desktop Daten: freie Wahl, auch Erzeugung von künstlichen Daten auf erlaubt</p>
Github Actions	<p>Beschreibung CI/CD Pipeline und github actions. Umsetzung von Codeprüfungen durch Gitleaks (<a href="https://github.com/gitleaks/gitleaks">GitHub - gitleaks/gitleaks: Protect and discover secrets using Gitleaks</a> ) für Secrets, Fossa (<a href="https://fossa.com/">https://fossa.com/</a>) für Lizenzen und Ruff (<a href="https://www.thoughtworks.com/radar/tools/ruff">https://www.thoughtworks.com/radar/tools/ruff</a>) für Linting.</p> <p>Code: MQTT-Beispielcode</p>
Open Source Lakehouse Container (mittels DuckDB)	<p>Beschreibung/Definition von Lakehouse. Erstellung eines Containers (oder mehrerer Container)), der Open Source Komponenten für ein Lakehouse enthält und Verprobung der Umsetzung anhand eines Beispiels.</p> <ul style="list-style-type: none"> <li>• Storage: Parquet, Delta und MinIO</li> <li>• Compute: DuckDB, Pandas und Ibis (<a href="https://github.com/ibis-project/ibis">https://github.com/ibis-project/ibis</a>)</li> <li>• Visualisierung: Apache Superset</li> </ul>

Open Source Lakehouse Container (mittels PySpark)	Beschreibung/Definition von Lakehouse. Erstellung eines Containers (oder mehrerer Container), der Open Source Komponenten für ein Lakehouse enthält und Verprobung der Umsetzung anhand eines Beispiels. <ul style="list-style-type: none"><li>• Storage: Parquet, Delta und MinIO</li><li>• Compute: PySpark, Pandas und Ibis (<a href="https://github.com/ibis-project/ibis">https://github.com/ibis-project/ibis</a>)</li><li>• Visualisierung: Apache Superset</li></ul>
---	--