Cheat sheet: GenAI with KNIME Analytics Platform

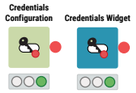
Open for Innovation KNIME

GenAI & LLMs

GenAI refers to artificial intelligence that can create content such as text, images, audio, code, and more, typically using advanced machine learning models. **LLMs** are a class of multipurpose and multimodal deep neural networks trained on vast and diverse datasets, making them capable of understanding and generating natural language and other types of content to perform a wide range of tasks (e.g., text completion, summarization, image editing, speech-to-text, etc.). Most LLMs are based on a transformer architecture and can capture complex relationships in data with multiple neural network layers and billions of fine-tunable parameters, which are further enhanced by an attention mechanism. "Large" refers precisely to the billions of parameters trained to accurately predict the next word in a sequence based on the previous ones.

Authenticate

Dedicated nodes to authenticate to an AI provider. Authentication requires credentials, which can be set at the workflow level or created within the workflow.



Output encrypted credentials (i.e., username and/or password) as a flow variable, which can be used to authenticate to an AI provider.

Open source

Authenticates to Hugging Face Hub models by selecting a valid Hugging Face API access token.

Closed source

Authenticates to all OpenAI models by selecting a valid OpenAI API key. It also allows you to specify a base URL to set the destination of the API request (e.g., to specify the URL of a local host) and connect to any server that supports the OpenAI API.

Authenticates to all Azure OpenAI models by selecting a valid Azure OpenAI API key and providing the resource endpoint.

Resources

- **KNIME Press:** Access various data science books and other cheat sheets at ***knime.com/knimepress***, including beginner and advanced topics.
- **KNIME blog:** Engaging topics, challenges, industry news, & knowledge nuggets at ***knime.com/blog***.
- **Self-paced courses:** Take our free online self-paced courses to learn about data analysis, data engineering, or data science with KNIME (with hands-on exercises) at ***knime.com/learning***.
- **KNIME Community Hub:** Store, version, automate, and collaborate on private workflows, or explore and share public workflows with the KNIME Community at ***hub.knime.com.***
- **KNIME Forum:** Join our global community & engage in conversations at ***forum.knime.com***.
- **KNIME Business Hub:** For team-based collaboration, automation, management, & deployment check out KNIME Business Hub at ***knime.com/knime-business-hub***.

KNIME AI Assistant

K-AI is an extension for KNIME Analytics Platform that enriches the software with built-in AI-powered support. Its key features include:

Q&A — K-AI understands and responds to questions about KNIME Software in natural language. Users can seek help about data operations, node configuration, KNIME resources, or features.

Build — K-AI generates workflows based on natural language descriptions. Users can describe what they want to achieve, and the AI Assistant will automatically build the corresponding workflow using the appropriate nodes, connections and configurations.

Data operations and code generation — K-AI is a built-in feature of the Expression node to help users with generic row-by-row data manipulation based on natural language descriptions. K-AI is also available in the Python Integration and the Generic ECharts View node, enabling users to seek help for the generation of Python or JavaScript code snippets.

Connect

Dedicated connector nodes to API-based or local LLMs. Supported model are suited for text generation, chatting and embeddings. Except for embedding models, these connectors also allow hyperparameter tuning (e.g., temperature, maximum response length, etc.). Capabilities and performance vary according to the AI provider.

Open source (API)

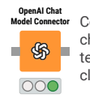
Connects to LLMs that handle text generation tasks by providing the model's Repo ID (e.g., bigscience/bloom).

Connects to chat LLMs by providing the model's Repo ID. Some models may require a System Prompt Template to describe the behaviour of the chat assistant, and a Prompt Template to define the roles in the interaction.

Connects to embedding models using the model's Repo ID.

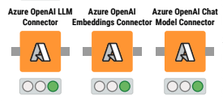
Closed source (API)

Connects to LLMs suitable for tasks such as summarization, classification, code generation, etc.



Connects to chat LLMs, suitable for building chat assistants, as well as performing all other text generation tasks (e.g., summarization, classification, code generation, etc.)

Connects to embedding models and allows to customize the size of the vector space into which documents are embedded.



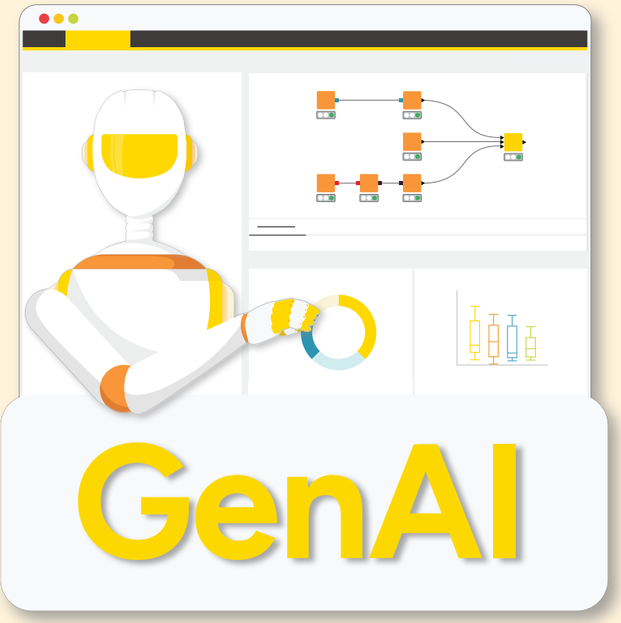
Connects to OpenAI's LLMs hosted on a Microsoft Azure instance, allowing users to leverage Azure's cloud infrastructure and services.

Open source (local)

Connects to locally-hosted LLMs. It allows the selection of a processing unit (e.g., CPU or GPU) on which the GPT4All model will run.

Connects to locally-hosted chat LLMs. Some models may require a System Prompt Template to describe the behaviour of the chat assistant, and a Prompt Template to define the roles in the interaction. It also allows the selection of a processing unit.

Connects to the embedding model by direct download from GPT4All or by specifying the local model path.

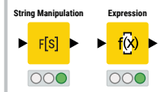


Prompt

Dedicated nodes to prompt an LLM.

Prompt engineering

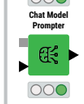
Involves designing and refining input instructions to guide the model towards generating desired responses. Common best practices include formulating clear and specific instructions, placing the request at the start, providing examples, and avoiding ambiguities, jargon, or assuming knowledge.



Perform operations on String values in columns, such as joining two or more strings, extracting substrings, formatting strings, implementing RegEx, and more. Most operations are also available in the Expressions node, which additionally supports *if* and *switch* conditions, and integrates an AI assistant to help compile functions.

Model prompters

Sends a separate prompt to the LLM for each row in the input table and outputs the corresponding response. It processes rows independently, so the model doesn't retain memory of prior rows or responses.



Allows for a conversational interaction between the human and AI. The node requires a prompt and the conversation history, so that for each prompt, it generates a response with knowledge of previous interactions.

Takes a text as input and generates a dense vector of floating-point numbers capturing the semantic meaning of the text.

Generates synthetic images based on a user prompt. Image size, quality and style can be customized.

Govern

Dedicated nodes and software features to ensure the ethical development and deployment of GenAI technologies. It involves detecting weaknesses and risks in LLMs and RAG-based systems, setting up internal controls, ensuring compliance with regulations, protecting data privacy, and addressing biases to align GenAI tools with societal values and legal standards.

Evaluation

Detects automatically critical vulnerabilities and risks associated with LLMs, such as information disclosure, hallucinations, prompt injection or the generation of harmful content, and outputs a report. It uses a combination of heuristics-based and LLM-assisted detectors.

Generates automatically a test set of multiple question types (e.g., simple, complex, distracting, etc.), relevant context for answering them, and AI-generated reference answers from the knowledge base of a RAG system.

Uses a generated test set of multiple question types to evaluate specific components of a RAG system (e.g, the generator, the retriever, or the quality of knowledge base chunks). It outputs an evaluation report.

Data anonymization

Detects sensitive Personal Identifiable Information (PII) data in English texts, using automated detection mechanisms. It outputs the detected entities, their type, the start/end index, and a certainty score.

Anonymizes sensitive PII data in English texts by replacing all occurrences of the selected PII entity types with pseudonyms.

Reverses the anonymization of the Presidio Anonymizer node by replacing the anonymized PII entities with their original information.

Model management

KNIME GenAI Gateway lets KNIME Business Hub admins manage chat and embedding models centrally, making them accessible in KNIME workflows with dedicated connector nodes. Admins can add models they trust, specifying the model's name, type, description, and authentication credentials.

Authenticates to a KNIME Hub instance. The output port allows to access resources in the configured Hub.

Lists in a table the models available in the GenAI Gateway.

Connects to an embedding model registered in the GenAI Gateway. It takes as input the Hub credentials provided by the KNIME Hub Authenticator.

Connects to a chat model registered in the GenAI Gateway. It takes as input the Hub credentials provided by the KNIME Hub Authenticator.

Continuous Deployment for Data Science

This extension for KNIME Business Hub automates the end-to-end process of deploying data science solutions safely into production. Users can validate, deploy, monitor, and update data science workflows through an intuitive UI, while admins oversee the deployment process and keep track of changes in the event log. CDDS leverages enterprise features of KNIME Software such as integrated deployment, KNIME Hub spaces with defined execution contexts, Data Apps, workflow schedulers, and triggers. It can be also customized to add validation and governance capabilities, evaluate and audit GenAI workflows, use the company-wide archival structure for auditability, or change monitoring and updating strategies.

Customize

Dedicated nodes to personalize or adjust the interactions with LLMs for a specific task or domain.

Retrieval Augmented Generation

RAG is an AI framework that enhances the generation of LLM responses by incorporating relevant information retrieved from a user-curated knowledge base (e.g., documents, guides, up-to-date information, code, terminology, etc.). RAG is often used to customize LLM responses for domain-specific applications and significantly mitigates the risk of hallucinations and unfactual statements. The implementation of RAG requires a searchable knowledge base and a user prompt. The additional use of an embedding model and a vector store is one implementation option, but a keyword-based search approach can be used as well.

Vector stores

Vector Stores are databases specialized in storing and managing objects (e.g., documents, code, dictionaries, etc.) as vector representations in a multidimensional space. The structure of these stores allows for quick and effective lookup of vectors associated with specific objects, facilitating their retrieval.

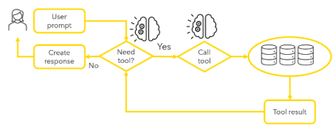
Splits lengthy documents into smaller paragraphs, while keeping semantic relations. It allows you to define a chunk size and a chunk overlap to retain context and prevent exceeding an LLM's context window.

Creates a Chroma or FAISS vector store by converting documents into numerical vectors using an embedding model and storing them. These vectors represent the semantic meaning of the documents.

Uses the vector store to find documents with similar semantic meaning for a given query. It can ouput a dissimilarity score based on L2 distance.

Agents

Conversational Retrieval Agents are AI systems designed to facilitate interactive, context-aware, and domain-specific chats with users. The agent is powered by an LLM capable of holding conversations in natural languages and configured to dynamically retrieve, if necessary, pertinent information from a user-curated and specialized knowledge base to best respond to a query.



Creates an agent, defines its function, its general behaviour, and how it interacts with the available knowledge base(s). The node requires the use of an (Azure) OpenAI's chat model.

Converts a vector store into an accessible and utilizable resource for an agent by giving it a name and a description.

Takes as inputs the output of the OpenAI Functions Agent Creator, a set of tools (e.g., vector stores), and the conversation history table. The latter is used to generate contextually relevant responses.

Fine-tuning

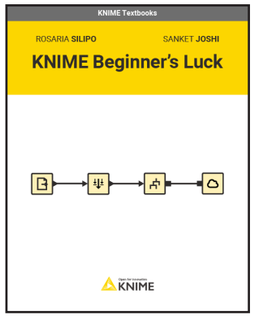
Model fine-tuning involves adapting a pre-trained model to a specific task or domain by training it further on new, task-specific data. This allows the model to leverage its existing knowledge while improving performance on the new task. For LLMs, fine-tuning is usually recommended only after attempting to get good results with prompt engineering (e.g., via few-shot learning) or RAG, for it requires a careful investment of time and effort.

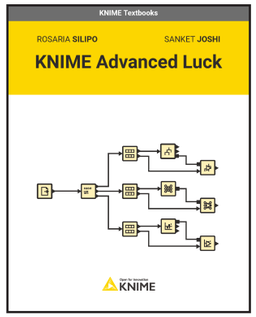
Overcomes the limitations of few-shot learning in the prompt by fine-tuning OpenAI's chat models on specialized data samples provided by the user. It requires conversation training data to be prepared following OpenAI's prescribed format. The resulting fine-tuned models lives in the user account with the AI provider.

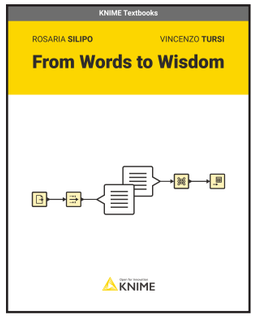
Removes irreversibly a fine-tuned model from the user's OpenAI account.

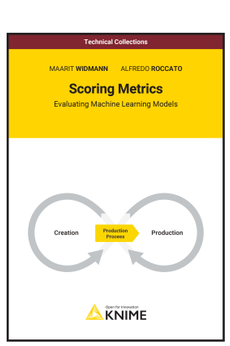
KNIME Press

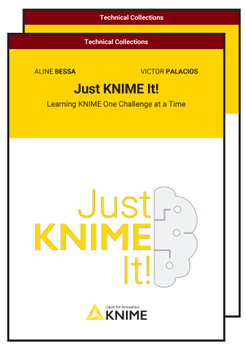
Extend your KNIME knowledge with our collection of books from KNIME Press. For beginner and advanced users, through to those interested in specialty topics such as topic detection, data blending, and classic solutions to common use cases using KNIME Analytics Platform - there's something for everyone. Available for download at www.knime.com/knimepress.

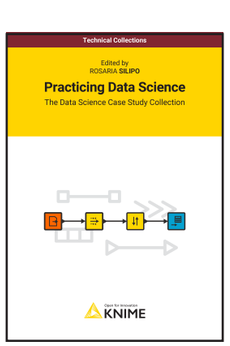


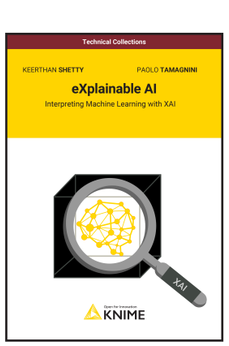


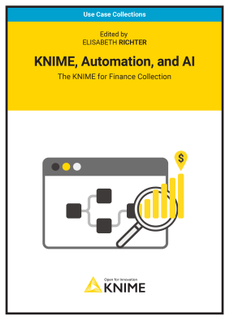


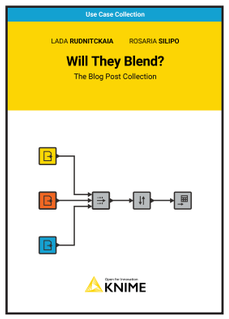


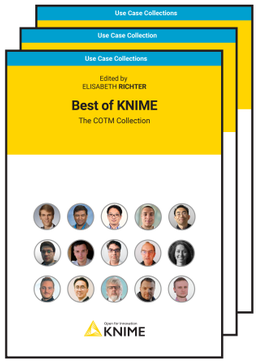


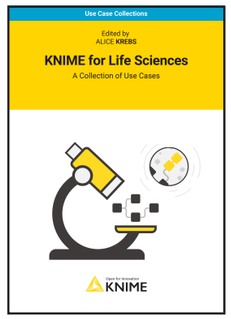


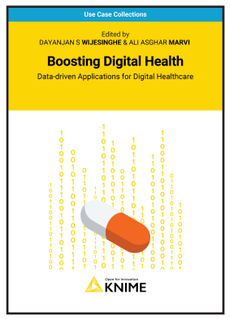


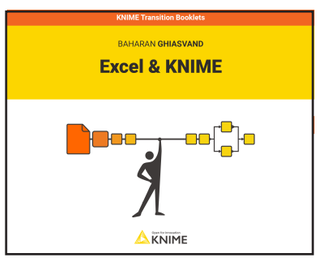


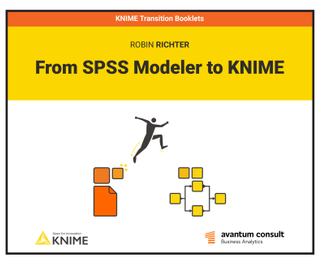


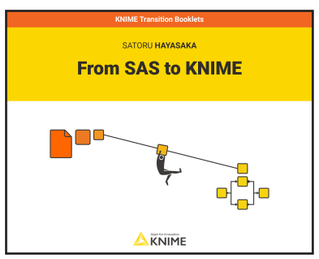


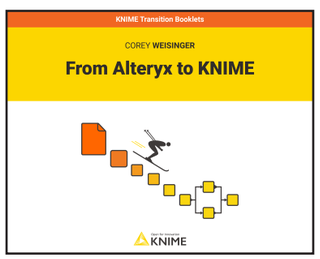


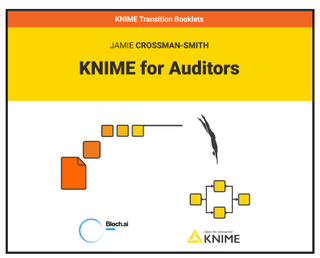












Need help?
Contact us!