**Giskard (The Evaluation & Testing framework for LLMs & ML models)**

Several notable tools have emerged to address the need for robust testing and evaluation of machine learning models, particularly Retrieval-Augmented Generation (RAG) systems. Among these tools, Giskard AI stands out as an open-source framework dedicated to testing machine learning models. It provides comprehensive solutions for identifying and addressing model vulnerabilities, ensuring reliable and high-performance models.

**Key Features:**

**Comprehensive Testing:** Giskard provides extensive testing capabilities for AI systems, ensuring they meet performance and safety standards.

**Error and Bias Identification:** The tool is proficient at spotting potential errors or biases in AI models, helping to ensure fairness and accuracy.

**Proactive Problem Detection:** Giskard detects issues before real-world deployment, mitigating risks associated with AI implementation.

**Support for AI Developers:** By verifying AI systems' performance and reliability, Giskard aids developers in delivering robust AI solutions.

**Automatic Testing:** Giskard can run numerous tests on AI models autonomously, saving time and resources for developers.

**Finding AI Risks:** The tool excels in identifying potential risks in AI models, including biases and unfair treatment of certain groups.

**Test Creation:** Giskard facilitates the creation of various tests tailored for different AI applications, ensuring thorough evaluation across multiple use cases.

**Compatibility with Popular AI Tools:** Giskard integrates seamlessly with common AI programming tools, allowing developers to incorporate it into their existing workflows easily.

**Performance Checking:** The tool evaluates the performance of AI systems, providing metrics that help developers determine if their AI models are ready for deployment.

**Safety Tests:** Giskard includes specialized tests to ensure the safety and reliability of AI systems, which is crucial for applications that impact people's lives.

**Aim of Giskard AI:**

Named after a robot in Isaac Asimov’s science fiction, Giskard AI aims to keep AI safe for everyone by:

**Running Extensive Tests:** Giskard conducts thorough testing on AI systems, providing developers with insights and verification of their work.

**Increasing Confidence:** The tool enhances confidence in AI applications, particularly those used in customer service and decision-making scenarios.

**How Giskard AI Works in Practice**

When developers use Giskard AI, they follow a few main steps:

1. **Setting Up**: First, they connect Giskard to their AI model. This is usually quite easy and doesn’t take long.
2. **Choosing Tests**: Next, they pick which tests they want to run. Giskard offers many types of tests, so they choose the ones that fit their AI’s job.
3. **Running Tests**: Giskard then runs all the chosen tests on the AI model. This can take some time, depending on how big and complex the AI is.
4. **Looking at Results**: After the tests are done, Giskard shows the results. It points out any problems it found with the AI.
5. **Fixing Issues**: If Giskard finds any problems, the developers work on fixing them. They might change parts of their AI model to make it work better.
6. **Testing Again**: Once they’ve made changes, they use Giskard to test the AI again. This helps make sure the problems are really fixed.

**RAG Evaluation Toolkit (RAGET): Automatically generate evaluation datasets & evaluate RAG application answers:**

If you're testing a RAG application, you can get an even more in-depth assessment using **RAGET**, Giskard's RAG Evaluation Toolkit.

1. **RAGET** can generate automatically a list of question, reference\_answer and reference\_context from the knowledge base of the RAG. You can then use this generated test set to evaluate your RAG agent.
2. **RAGET** computes scores *for each component of the RAG agent*. The scores are computed by aggregating the correctness of the agent’s answers on different question types.

* Here is the list of components evaluated with **RAGET**:
  + 1. **Generator**: the LLM used inside the RAG to generate the answers
    2. **Retriever**: fetch relevant documents from the knowledge base according to a user query
    3. **Rewriter**: rewrite the user query to make it more relevant to the knowledge base or to account for chat history
    4. **Router**: filter the query of the user based on his intentions
    5. **Knowledge** **Base**: the set of documents given to the RAG to generate the answers

**Giskard works with any model, in any environment and integrates seamlessly with your favorite tools:**

scikit-learn, TensorFlow, PyTorch, Colab (Google Colab), Kaggle, pandas, XGBoost, fast.ai, Kubernetes, Keras, PyTorch Ignite, Python, Jupyter, Kubeflow, AWS (Amazon Web Services), Google Cloud, Azure, Lightning AI (Creators of PyTorch Lightning), OpenAI, Hugging Face, Gemini, Weights & Biases, GitHub, MLflow, cohere.

**Licensing and Usage Information**

1. **License**: Giskard AI is distributed under the Apache 2.0 license. This permissive license allows for a wide range of uses, including modification and redistribution.
2. **Commercial Use**: Giskard AI is free to use, including for commercial purposes. There are no restrictions on its use in commercial projects.
3. **Open Source**: Yes, Giskard AI is open source. Its source code is publicly available, and contributions from the community are welcome.

**System Requirements for Giskard Library**

To effectively use the Giskard AI library, ensure that your system meets the following requirements:

1. **Operating System**:

* Linux (Ubuntu 18.04 or later)
  + MacOS (10.15 Catalina or later)
  + Windows 10

1. **Python Version**:
   * Python 3.7 or later
2. **Hardware Requirements**:
   * CPU: Multi-core processor (Intel i5 or AMD equivalent and above)
   * RAM: Minimum 8GB (16GB recommended for larger models)
   * Disk Space: At least 2GB of free space for installation and model storage
3. **Software Dependencies**:
   * pip for managing Python packages
   * Virtual environment tools like venv or conda (optional but recommended for managing dependencies)
   * Docker (optional, for containerized deployment)
4. **Additional Tools** (for integration and compatibility):
   * Jupyter Notebook or JupyterLab for interactive testing
   * Popular AI frameworks and libraries (e.g., TensorFlow, PyTorch, scikit-learn) for model compatibility

**Source:**

1. [Mastering RAG Evaluation: A Deep Dive into Giskard AI for PDF Chatbots | by Rohith Reddy Vangala | Jul, 2024 | Medium](https://medium.com/@rohithreddy66666/mastering-rag-evaluation-a-deep-dive-into-giskard-ai-for-pdf-chatbots-c6d1163330aa)
2. [Giskard-AI/giskard: 🐢 Open-Source Evaluation & Testing for LLMs and ML models (github.com)](https://github.com/giskard-AI/giskard?tab=readme-ov-file)
3. [📥 Install the Giskard Python Library - Giskard Documentation](https://docs.giskard.ai/en/stable/open_source/installation_library/index.html)