Maîtriser l'Évaluation des LLM et Workflows dans n8n

Confronter vos hypothèses à la Vérité Terrain (Ground Truth).

Anne-Marie ROY - 16/12/2025

Agenda

- Introduction: De l'Automatisation aux Agents IA
- Partie 1: Comprendre l'Évaluation des Workflows IA
- Partie 2: Méthodologie d'Évaluation
- Partie 3: Implémentation dans n8n
- Partie 4: Démonstrations Pratiques
- Partie 5: Production-Ready
- Q&A + FAQ

Introduction: De l'Automatisation aux Agents IA

🧰 De l’Automatisation aux Agents IA

Comprendre les couches avant de vouloir tout automatiser

1️⃣ Workflows classiques (la fondation)

- Systèmes déterministes et basés sur des règles
- Prévisibles, reproductibles, robustes
- « Ennuyeux, mais fiables »
- Exemples:
  - CRM/ERP: ticket support, nouveau client, automatisation de facturation, …)
- 👉 C’est ici que se crée la stabilité

2️⃣ Workflows assistés par l’IA

- Un workflow classique + une brique d’intelligence
- Exemples :
  - Scoring de priorité d’un ticket
  - Personnalisation d’e-mails
- 👉 Valeur ajoutée ciblée, sans perdre le contrôle

🧱 De l’Automatisation aux Agents IA

Comprendre les couches avant de vouloir tout automatiser

- **3️⃣ Agents IA**
  - Systèmes autonomes et adaptatifs
  - Décident, utilisent des outils, changent de trajectoire
  - Exemples :
    - Agent de support autonome
    - Agent de veille/recherche
    - Agent d’orchestration de tâche
  - 👉 Puissants mais fragiles, difficiles à déboguer

👉 NB: « *Un workflow exécute ce que vous avez décidé. Un agent décide ce qu’il va faire.* »

📊 **Pourquoi commencer par les workflows “classiques”?**

- 30 % à 200 % de ROI la 1re année (McKinsey)
- 25 % à 40 % d’économies de coûts de main-d’œuvre
- ≈ 50 % des tâches automatisables sans IA
- 🚨 Sauter directement aux agents = confusion + systèmes instables

Partie 1: Comprendre l'Évaluation des Workflows IA

Evaluation des Workflows IA

- **Problème de la boîte noire** – Les LLMs produisent des résultats différents même avec les mêmes entrées.
- **Nature probabiliste** – La sortie est façonnée par la probabilité, la température et le contexte.
- **Modèles évolutifs** – Le comportement des LLMs peut changer au fil du temps.
- **Au-delà des tests traditionnels** – Il faut tenir compte de la précision, de la cohérence, de la robustesse, des biais et du coût.
- **Vos tests unitaires classiques sont obsolètes**. Vérifier si une chaîne de caractères est exactement égale à une autre ne fonctionne pas quand la réponse peut varier sémantiquement tout en restant juste.
- **Il faut donc apprendre à mesurer de nouvelles choses** : la réponse est-elle précise ? Est-elle cohérente ?

La “mise en production” - Les Conséquences

Pourquoi devrais-je accorder de l'importance aux évaluations ?

- **Incohérences des LLMs**
  De légères modifications de votre prompt (ou même l'absence de modification) peuvent conduire à des résultats différents.
- **Dérive du contexte sur les longues conversations**
  Les LLMs peuvent progressivement perdre le fil des instructions originales à mesure que leur contexte s'agrandit.
- **Cas limites (Edge cases)**
  Des variations dans les données d'entrée, comme des variables manquantes ou des formats inattendus, peuvent entraîner des résultats imprévus.
- **Changements d'API / de modèle**
  Les fournisseurs de modèles peuvent modifier leurs API, et même leurs modèles, ce qui conduit à des résultats différents de ceux obtenus auparavant.

Le Cycle de Vie de l'Évaluation

Quand évaluer votre Workflow IA ?

- **PHASE 1 : Pendant la construction (Build) :**
  - Objectif : Itérations rapides.
  - Action : Tester différents prompts et attraper les cas limites (Edge cases) comme des entrées vides ou des formats inattendus.
- **PHASE 2 : Avant la mise en prod (Pre-Production) :**
  - Objectif : Validation globale.
  - Action : Vérifier la Non-Régression (est-ce que j'ai cassé ce qui marchait avant ?) et faire de l'**A/B Testing** (comparer Claude 3.5 Sonnet vs GPT-4o sur le ratio Coût/Qualité).
- **PHASE 3 : En Production (Monitoring) :**
  - Objectif : Surveillance continue.
  - Action : Vérifier le "Drift" (dérive) du modèle ou si une mise à jour de l'API change les résultats.

Partie 2: Méthodologie d'Évaluation

Le jeu de données "Gold Standard"

- **Objectif** : Vérité terrain pour mesurer les résultats de l'IA
- **Qualités de bonnes données** :
  - Précises, cohérentes, complètes, représentatives
  - Couvrent les cas limites
  - Suffisamment importantes pour la significativité statistique
- **Sources d'exemples** :
  - Tickets historiques de haute qualité, réponses d'experts
  - Contenu marketing très performant
  - Résultats idéaux sélectionnés par des experts (SME)

Les 3 Niveaux de l'Évaluation (Maturity Model)

On ne commence pas par des tests complexes (keep it simple!)

- **Le "Vibe Check" (Niveau 1) :**
  - Test manuel, non automatisé.
  - On discute avec le bot pour "sentir" si le ton et les réponses sont corrects.
  - Objectif : Détecter les problèmes flagrants et avoir des idées de tests.
- **Les Tests Déterministes (Niveau 2) :**
  - Basés sur du code ou des règles strictes (Regex, JSON Schema).
  - Exemple : "Est-ce que tous les outils ont été appelés ?", "Le JSON est-il valide ?".
- **LLM as a Judge (Niveau 3 - Le Graal) :**
  - Utiliser un LLM pour évaluer un autre LLM.
  - Exemple : Comparer la sémantique de la réponse générée avec une "Réponse de Référence" (Reference Answer).
  - Métriques : Correctness (Justesse), Helpfulness (Utilité), Style.

LLM as a Judge : Le Modèle "Professeur / Élève"

Faut-il utiliser le même LLM pour générer et pour noter ?

- **La réponse courte : NON. Il est souvent recommandé de changer.** Ce n'est pas comparer des "torchons et des serviettes", c'est appliquer une rigueur pédagogique.

- 1. **Éviter le biais d'auto-validation (Self-Correction Bias)**
  - Si par exemple, Claude 3.5 Sonnet se note lui-même, il risque d'être "indulgent" envers son propre style, ses tics de langage ou ses propres failles logiques.
  - Il validera une réponse car elle correspond à sa propre manière de "penser".
- 2. **Le principe hiérarchique : Le Juge doit être > l'Élève**
  - L'Élève (Production) : Pour la vitesse et le coût, vous utilisez souvent un modèle "léger" (ex: GPT-4o-mini, Claude Haiku).
  - Le Professeur (Évaluation) : Pour la notation, vous devez utiliser le modèle le plus intelligent et rigoureux possible (ex: GPT-4o, Claude 3.5 Opus), peu importe son coût/lenteur, car il ne tourne qu'en phase de test.
- 3. **Validation croisée (Cross-Validation)**
  - Faire juger un modèle d'Anthropic par un modèle d'OpenAI apporte une indépendance totale et une crédibilité accrue au score final.
- **Dans n8n : C'est trivial.**
  - Connectez simplement un nœud de modèle différent (ex: OpenAI) au nœud "Metrics", tout en laissant votre modèle de production (ex: Anthropic) connecté à votre chaîne principale.

Partie 3: Implémentation dans n8n

Démo / Exemple de Workflow d'Évaluation

- **Structure visuelle :**
  - Evaluation Trigger (Charge les données de test).
  - Votre Agent IA (Traite la demande).
  - Set Metrics - C’est le juge ! (Compare "Résultat Réel" vs "Résultat Attendu").
- **Métriques clés :**
  - Précision (Accuracy) : % de réponses correctes.
  - Coût : Tokens utilisés par exécution.
  - Latence : Temps de réponse.

Check-list avant la mise en production ☑

- Gestion d'erreur globale configurée (Error Workflow).
- Données sensibles sécurisées (Credentials, pas de clés en dur).
- Nettoyage des nœuds de debug (ex: No-Op ou Console Log).
- Test de charge (Load Testing) si gros volumes attendus.

Visualiser et Comparer (A/B Testing)
A/B Testing et Optimisation des Coûts

L'**A/B Testing** consiste à comparer deux versions d'un même élément (ici, deux modèles d'IA ou deux prompts) en les testant en parallèle sur des données identiques, pour déterminer scientifiquement laquelle offre les meilleures performances selon des critères définis (comme la qualité de la réponse ou le coût).

- **Le concept :** Utiliser un "Router" ou un "Model Selector" dans n8n pour tester deux modèles en parallèle.
- **L'analyse possible dans des outils externes (Graphana, Google Sheets, QuickChart), ou un workflow de visualisation**
  - Comparaison visuelle :
    - Axe X : Coût (Tokens) / Latence (Temps).
    - Axe Y : Score de qualité (Correctness).
- **Conclusion :** Permet de choisir le modèle le moins cher qui maintient un niveau de qualité acceptable (ex: passer de GPT-4 à GPT-4o-mini si le score reste > 4/5).

Partie 4: Démonstrations Pratiques

Démo

Démo Évaluation n8n natif

Démo Évaluation Gemini (script Python) - Passage à l’échelle

Démo Evaluation n8n + LangSmith

Jeu de données - Référentiel

- Questions sur des évènements historiques
- Question sur Lumina Corp - Base documentaire (RAG)

Focus Technique - Le Nœud "Check If Evaluating"

Astuce Technique : Isoler la logique de test

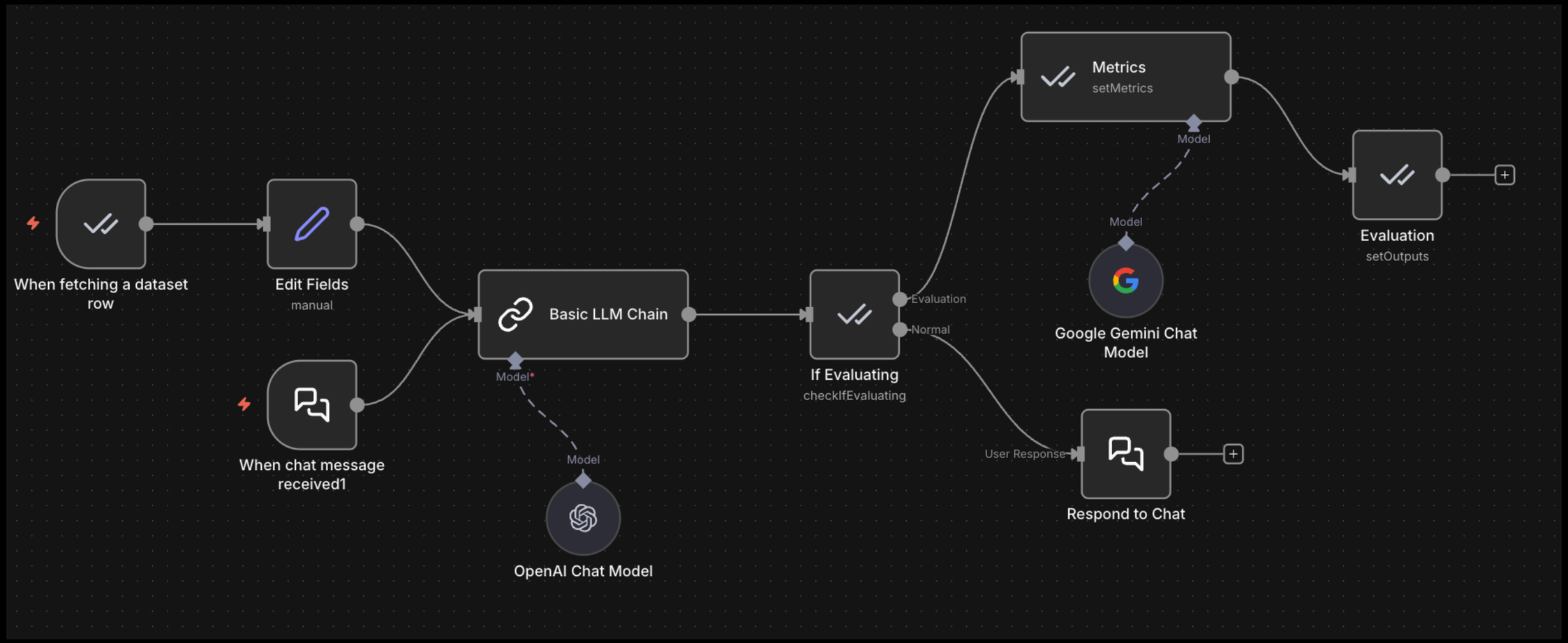
- **Le problème :** Votre workflow attend une entrée chat (Webhook/Chat Trigger), mais l'évaluation envoie des données depuis un Dataset (Tableau).
- **La solution :** Le nœud Check if Evaluating.
- **Fonctionnement :**
  - Si Mode Évaluation : Récupérer l'entrée depuis le Dataset (Golden Data).
  - Si Mode Production : Récupérer l'entrée depuis le Chat/Webhook utilisateur.
- Avantage : Permet d'avoir un seul workflow unique pour le dev, le test et la prod.

Démo

Evaluations de faits historiques

Evaluations RAG Lumina Corp

Démo n8n natif



Démo Évaluation n8n natif

Flux de travail (Workflow) :

- Lors de la récupération d'une ligne du jeu de données - Récupère la question et la réponse_de_référence depuis la table QA_Evaluations.
- Éditer les champs (Edit Fields) - Transmet tous les champs, y compris réponse_de_référence, et ajoute un champ `chatInput` contenant la question.
- Chaîne LLM basique (Basic LLM Chain) - Génère une réponse à la question (le résultat est au format texte).
- Noeud: If Evaluating - Vérifie si le mode évaluation est actif (laisse passer les données).
- Évaluation (Evaluation) - Écrit la réponse du LLM (texte) dans le jeu de données en tant que réponse_réelle.
- Métriques (Metrics) - Compare :
  - Réponse attendue : réponse_de_référence issue de votre jeu de données.
  - Réponse réelle : texte issu de la Chaîne LLM basique (la réponse générée par l'IA).

Démo (Gemini Evaluation)

Notebook (Python script)

🔄 Comment ça fonctionne :

1. Vous lui donnez un CSV avec des questions + réponses de référence + réponses réelles

2. Pour chaque ligne, il demande à Gemini : "Cette réponse est-elle bonne ?"

3. Gemini répond avec un score (0-1) et une explication

4. Les résultats sont collectés avec des métriques (temps, coût, tokens)

5. Vous récupérez un DataFrame avec tous les scores + raisonnements

Démo Langsmith

LangSmith est une plateforme d’observabilité et d’évaluation des applications LLM qui permet de tracer, analyser et mesurer le comportement réel des chaînes, agents et systèmes RAG afin d’améliorer leur qualité, leur fiabilité et leurs performances.

LangSmith fait partie de l’écosystème LangChain, mais il ne faut pas confondre les deux niveaux (librairie vs plateforme).

LangSmith n’est pas open source, il y a un pricing (SaaS) hébergée par LangChain Inc.

Elle propose :

- un free tier (usage limité, idéal pour tests / POC)
- des plans payants selon :
  - volume de traces
  - nombre d’évaluations
  - fonctionnalités avancées (datasets, comparaisons, collaboration)

Il n’existe pas de version self-hosted officielle ni de licence open source pour LangSmith.

Comparatif n8n v.s. LangSmith

Fonctionnalité	n8n	LangSmith
Exécution du workflow	✅ Natif (orchestration, automatisation)	❌ Non
Logs d'exécution	✅ Logs de workflow (étapes, erreurs)	✅ Logs de traces LLM (chains, agents)
Métriques de base (latence, tokens, coût)	⚠️ Possible (via instrumentation / nodes)	✅ Natif
Reasoning summary	❌ Non natif (possible via LLM dédié)	✅ Natif
Analyse qualitative automatique	❌ Non native (à construire manuellement)	✅ Natif (evaluators LLM)
Comparaison avec ground truth	⚠️ Manuel (scorer / Correctness node)	✅ Automatique (datasets + evaluators)
Visualisation des traces complètes LLM	❌ Non (exécution linéaire, non LLM-native)	✅ Natif (traces hiérarchiques, multi-steps)
Debug agent / chain-of-thought indirect	❌	✅ (via traces & summaries)
Positionnement principal	Orchestration & automatisation	Observabilité & évaluation LLM

Démo Lumina Corp

Les tests effectués:

1) Le test du "Needle in a Haystack" (Question sur le Vermont) :

- Le piège : L'information sur le Vermont est cachée à la page 42 du fichier texte, au milieu de centaines de lignes répétitives sur l’ISO-9001.
- Ce que ça prouve : Que l'IA (et le chunking) a bien ingéré l'intégralité du document et ne s'arrête pas aux premières pages.

Démo Lumina Corp

Les tests effectués:

3) Le test "Multi-hop" (Question sur le prix et la marge) : :

- Le piège : Le prix (199) et le coût de revient (110) sont dans le transcrit PDF. Le calcul (199 - 110) doit être fait par l’IA.
- Ce que ça prouve : La capacité de raisonnement (Reasoning) de l'agent.

Conclusion

Arrêtez de deviner. Commencez à mesurer. Validez vos hypothèses avec la vérité objective."

- L'évaluation n'est pas une étape unique, c'est un cycle continu (Build -> Measure -> Optimize).
- Ressources : Documentation n8n, Forum communautaire.

Q&A et FAQ

FAQ

- **Q1 : "Le serpent qui se mord la queue" : Comment juger le juge IA ?**

C'est la question classique sur l'utilisation d'un LLM pour en évaluer un autre

Éléments de réponse :

- **Ne pas faire une confiance aveugle** : Le “LLM-Juge n'est pas infaillible. Il est un outil pour *mettre à l’échelle* l'évaluation humaine, pas la remplacer totalement.
- **Le "Golden Dataset" (Étalon-or)** : Créez un petit jeu de données de test (ex: 50 questions/réponses) évalué par des humains experts. Testez votre LLM-Juge sur ce dataset pour mesurer son alignement avec le jugement humain.
- **Échantillonnage humain (Spot-checking)** : Dans vos workflows n8n, envoyez aléatoirement 5% des évaluations du juge à un humain pour vérification.
- **Utiliser un modèle supérieur** : Utilisez toujours le modèle le plus puissant et le moins biaisé possible comme juge (ex: GPT-4-Turbo ou Claude 3 Opus) pour évaluer des modèles plus petits ou spécialisés.

FAQ

- **Q2 : Ça coûte cher ! L'évaluation est-elle rentable en prod ?**

(Question sur le ROI des appels API supplémentaires pour l’évaluation)

Éléments de réponse :

- **Le coût de l'erreur :** Combien coûte une hallucination envoyée à un client ? Souvent bien plus cher qu'un appel API d'évaluation.
- **L'échantillonnage en production :** Dans n8n, n'évaluez pas 100% des sorties en direct. Mettez en place un nœud "Switch" aléatoire pour n'évaluer qu'une requête sur 10 ou 20 pour monitorer la tendance.
- **Le dev vs la prod :** L'évaluation est cruciale et intensive pendant la phase de développement du workflow n8n (itération). En production, elle devient du monitoring.

FAQ

- **Q3 : Au-delà du "C'est bien", quelles métriques utiliser concrètement ?**

(Question sur la définition des critères d'évaluation)

Éléments de réponse :

- **Soyez spécifiques :** Une note globale de 1 à 5 est trop vague. Décomposez le problème.
- **Les métriques RAG standards :**
  - **Fidélité (Faithfulness) :** La réponse est-elle uniquement basée sur les documents fournis (pas d'hallucination externe) ?
  - **Pertinence (Relevance) :** La réponse répond-elle vraiment à la question posée ?
  - **Critères métier :** Respect du ton (formel/amical), respect du format de sortie (JSON valide), absence de contenu toxique.

FAQ

- **Q4 : Pourquoi n8n spécifiquement pour l'évaluation ?**

Question sur la valeur ajoutée de l'outil par rapport à du code pur)

Éléments de réponse :

- **Orchestration parallèle :** n8n excelle pour lancer la génération de la réponse ET, en parallèle, lancer le workflow d'évaluation sans bloquer l'utilisateur final.
- **Historisation facile :** Après l'évaluation, il est trivial dans n8n d'envoyer le score et la réponse dans une base de données (Airtable, Postgres) pour construire des dashboards de suivi qualité dans le temps.
- **Alerting :** Si le score d'évaluation passe sous un seuil critique (ex: < 3/5), le workflow n8n peut envoyer une alerte Slack immédiate à l'équipe.

FAQ

- **Q5 : Température LLM - Quel est le bon réglage selon l'objectif?**

Ce que contrôle la température:

- Paramètre d'échantillonnage probabiliste
- Influence le choix des tokens, pas les connaissances du modèle
- Axe clé : déterminisme ↔ diversité

- La température n'augmente pas l'intelligence du modèle. Elle règle le bruit probabiliste dans la prise de décision.

Échelle pratique

Température	Effet	Usage typique
0.0	Déterministe	Classification, scoring, évaluation
0.1 – 0.2	Très stable	RAG, QA factuelle
0.3 – 0.5	Équilibré	Résumé, reformulation
0.6 – 0.8	Créatif contrôlé	Brainstorming
0.9 – 1.2+	Exploratoire	Écriture créative