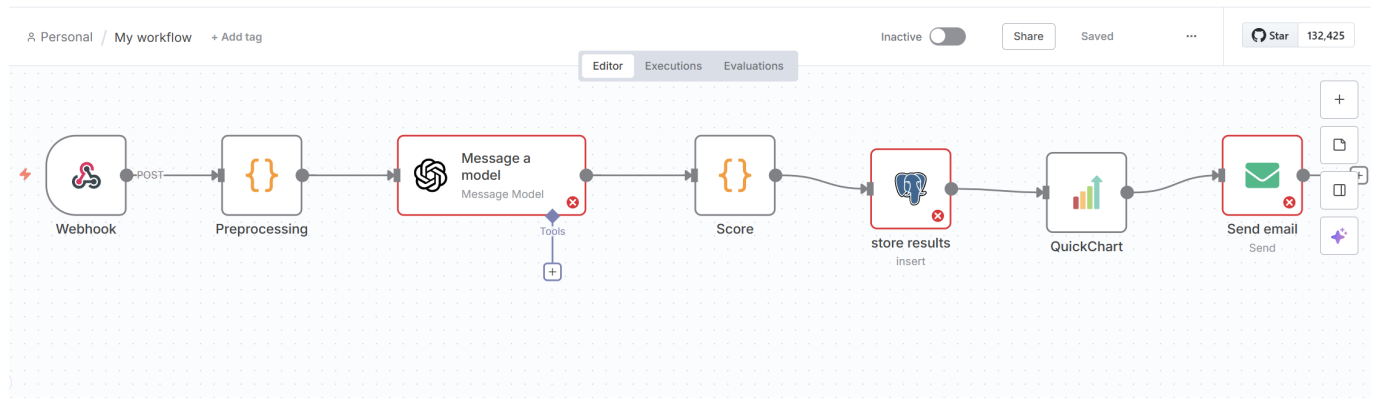


The overall workflow in n8n



Q1: How would you scale the AI pipeline to accommodate new metrics or replace AI models easily?

A: We can put each metric in its own mini-function (like `scoreConfidence()`, `scoreDepth()`). Combine them in a pipeline. Therefore, new metric = new function, it doesn't break the rest.

Q2: How do you measure and ensure confidence level, depth, and breadth in AI-generated reports?

A:

```
# Confidence: did it follow steps 1-4?
# Each one present = +1.
# then scores["confidence"] = divide by 4 -> gives a ratio between 0 and 1.

# Reasoning Depth
# We looped through all dimension names (e.g., Attention, Emotional Stability).
# Checked if the model mentioned them in its text. If it mentioned ≥2 dimensions →
we give full credit (1.0). If it only mentioned 1 dimension → 0.5. If none → 0.0.

# Breadth
# we filtered only the big gaps (where |delta| > 1).
# Then checked how many of those dimensions the model actually mentioned.
# Ratio = #covered big gaps ÷ total big gaps.

# How to ensure: We can prompt explicitly. Like write some sentences such as "Use
at least two dimension names from the list when explaining differences.", "Always
include the top 3 largest differences in your summary."
```

Q3: How would you transform raw AI analysis into practical parenting recommendations (e.g., actionable communication strategies)?

A: We can let AI agent start by identifying the gap, then briefly explain its significance, and offer a simple, practical action recommendation. For example, if a teenager think they are more responsible

than their parents, we can suggest that parents schedule a brief weekly check-in. Like the teenager lists the tasks they've completed and let parents acknowledge their progress. If the gap lies in emotional stability, parents could be encouraged to say, "I noticed your grade improved for this course. How did you do that?".

Q4: How do you prevent hallucinations and maintain trustworthiness in AI outputs?

A: we can use RAG to feed the model with real-world data as context, allowing it to provide answers based on facts. We can also optimize prompts, using methods like Few-shot and Chain-of-Thought to explicitly instruct the model not to fabricate answers.