**AI Agent Configuration Report**

This document outlines the creation and configuration of four AI agents developed for a hackathon project. These agents were built using **Amazon Bedrock**, **Anthropic Claude models**, and **OpenAI GPT models**, with each agent addressing a specific business challenge. The design of each agent focuses on solving real-world problems like workflow automation, model selection, cost estimation, and ROI calculation.

**Agent 1: Task Automation Evaluator**

**Purpose**: This agent helps businesses identify opportunities for task automation in their existing workflows. It acts as an advisor that suggests areas where repetitive tasks can be automated to save time and cost.

**Model Used**: *Amazon Bedrock – Nova Lite*

**Why this model**: Nova Lite is fast and lightweight, ideal for handling structured inputs like workflow descriptions. It provides relevant suggestions without the overhead of large-scale reasoning.

**Selected Mode**: *Instruction-following (Helpful Assistant)*

* **Memory**: OFF – the automation task is stateless and doesn't require recalling prior interactions.
* **Knowledge Base**: OFF – the focus is on analyzing the current input, not referencing external documents.
* **Fairness & Bias**: Not applicable – the suggestions are functional and task-oriented.

**How it works**: The agent first asks the user to describe their current business workflow. It then analyzes the steps and suggests which parts can be automated (e.g., repetitive data entry, scheduling, reporting), along with a reason for each suggestion. This helps businesses discover new areas of efficiency through AI.

**Agent 2: LLM Model Selector**

**Purpose**: This agent helps users choose the most appropriate Large Language Model (LLM) based on their business use case, expected output, and budget.

**Model Used**: *Amazon Bedrock – Claude 3 Sonnet*

**Why this model**: Claude 3 Sonnet provides a strong balance of cost, performance, and interpretability. It’s highly capable of nuanced decision-making, making it perfect for LLM comparison tasks.

**Selected Mode**: *Instruction-following (Helpful Assistant)*

* **Memory**: OFF – the selection task is typically a one-shot query with no prior context needed.
* **Knowledge Base**: ON – enables referencing detailed specifications and performance data of various LLMs.
* **Fairness & Bias**: ON – ensures recommendations are unbiased, especially when comparing models from different providers.

**How it works**: The agent prompts users to provide details about their task type (e.g., summarization, classification), expected output, and constraints like budget or latency. It then evaluates options and recommends one or more LLMs that best fit the criteria, backed by justifications from the knowledge base.

**Agent 3: Inference Cost Estimator**

**Purpose**: This agent estimates the operational costs involved in running LLM-based applications at a given scale. It helps businesses plan their budgets and select cost-efficient deployment strategies.

**Model Used**: *Amazon Bedrock – Claude 3 Haiku*

**Why this model**: Claude 3 Haiku is lightweight, fast, and optimized for short interactions like cost lookups and estimations. It's suitable for real-time calculations.

**Selected Mode**: *Instruction-following (Helpful Assistant)*

* **Memory**: ON – remembers previously provided usage metrics to maintain continuity during cost refinement.
* **Knowledge Base**: ON – accesses real-time pricing structures and model rate sheets.
* **Fairness & Bias**: ON – ensures accurate and non-biased cost evaluations across model providers.
* **File Uploads**: Enabled – allows uploading usage logs or CSVs containing traffic estimates.

**How it works**: The agent asks the user to input or upload usage data (e.g., number of API calls per day, duration of inference, model type). It then computes cost estimates for daily, monthly, and annual usage, and may suggest cheaper alternatives or optimizations where applicable.

**Agent 4: ROI Calculator**

**Purpose**: This agent helps business users calculate the return on investment (ROI) from integrating AI solutions into their operations. It supports strategic planning by estimating both cost and expected gain.

**Model Used**: *OpenAI – GPT-4o-mini*

**Why this model**: GPT-4o-mini offers excellent logical reasoning capabilities at a lower cost, making it ideal for detailed yet efficient multi-step financial analysis.

**Selected Mode**: *Instruction-following (Helpful Assistant)*

* **Memory**: ON – helps in carrying context during multi-turn interactions like step-by-step ROI computation.
* **Knowledge Base**: ON – provides access to industry benchmarks or past ROI case studies, if available.
* **Fairness & Bias**: ON – ensures that predictions are balanced and not overly optimistic.
* **File Uploads**: Supported – allows uploading previous project results or market reports to enhance accuracy.

**How it works**: The user inputs the cost of the AI solution and expected gains over time. The agent calculates break-even points, ROI percentage, and long-term impact. If additional data is provided, it compares the expected ROI with industry benchmarks or historical cases.

**Final Notes**

Each agent was configured carefully to meet specific judging criteria for the hackathon: **real-world impact**, **ease of use**, **goal clarity**, and **technical feasibility**. Model choices, mode settings, and feature toggles were made with purpose — ensuring each AI agent provides tangible value while remaining resource-efficient and easy to interact with.

Let me know if you’d like this report as a .docx or .pdf.

d model compatibility.