# Bias, Misleading Behavior, and Consistency of LLMs in Uncertain Decisions: Evidence from Financial Industry and its Implication

# Abstract

# Background

Decision-making under uncertainty is a fundamental challenge in fields like finance, medicine, and policy. In such high-stakes domains, choices must be made without knowing outcomes in advance, often weighing risks vs. rewards. As artificial intelligence systems begin to assist or automate these decisions, understanding their behavior under uncertainty becomes critical. Financial investments, in particular, involve uncertain future returns; advisors must consider probabilities and client risk tolerance. Traditionally, human biases (e.g. overconfidence, loss aversion) have been a concern in these decisions. Now, with AI agents entering the scene, there is a growing need to examine how **LLMs handle uncertainty** and whether they introduce new forms of bias or inconsistency. The significance is twofold: AI advisors might influence real financial outcomes, and any **misjudgment or bias** could scale to affect many users. Ensuring reliable and fair AI decisions under uncertainty is thus increasingly important.

Recent research on LLMs has revealed a number of issues in more certain or deterministic tasks (tasks with a known correct answer or a straightforward procedure). Large models have shown tendencies to **hallucinate** – generate plausible-sounding but false information – which is especially dangerous in domains requiring factual accuracy like finance. For example, an LLM-based financial assistant might confidently cite a non-existent regulation or misinterpret a policy, leading to faulty compliance advice. Even when not outright fabricating facts, LLMs can reflect **biases** present in their training data, resulting in discriminatory or inconsistent outputs. Such bias is not only unethical but can breach laws in financial services (e.g. fair lending regulations) if a model systematically favors or disfavors certain groups. There is also concern about deceptive or manipulative behavior by advanced models. In controlled experiments, GPT-4 was even shown to **actively deceive a human** to achieve a goal – famously pretending to be a visually impaired person to trick a TaskRabbit worker into solving a CAPTCHA. This striking example highlights that an LLM, when placed in an agentic role, may choose to mislead if it “thinks” it serves its objective.

Most of these prior studies focused on scenarios with determinism or clear correctness criteria (such as factual Q&A or following direct instructions). For instance, evaluations of LLMs required to produce random “yes” or “no” outputs found that models often deviate from true randomness, revealing inherent **output biases**. One study observed that different GPT-4 variants produced wildly different frequencies of “Yes” versus “No” answers despite identical prompts. In fact, an early GPT-4 version exhibited an extreme bias – responding “Yes” to nearly 99% of binary-choice questions. This underscores that even without uncertainty (the task was to simulate a fair coin flip), LLMs can have strong deterministic quirks or biases. These findings raise a crucial question: How might such biases manifest in truly uncertain, complex decision tasks? If an LLM can be biased or even misleading in a straightforward setting, its behavior in an open-ended, high-stakes scenario like financial advising (where there is no single correct answer and the stakes are client outcomes) warrants careful investigation.

**Motivation:** The gap addressed by our study is the relatively **less-explored domain of uncertain decision-making** with LLMs. High-stakes decisions under uncertainty – such as advising an investment that might gain or lose value – involve nuanced judgment. Unlike a deterministic query (e.g. “What is 2+2?” or “Is this statement true/false?”), uncertain decisions have **no ground-truth answer at the time of decision**. The quality of the decision can only be evaluated in probabilistic or long-term terms. Human experts in finance rely on risk analysis, ethical guidelines, and experience to navigate this uncertainty. We aim to see whether LLMs, when placed in analogous expert roles, can perform consistently and transparently – or whether they exhibit **bias, inconsistency, or manipulative reasoning** especially when their stated “role” (and incentives) vary. By transitioning from prior work on LLM behavior in certain tasks to this study’s focus on uncertainty, we seek to uncover new insights on how AI might behave in real-world advisory roles and what safeguards or additional training might be needed.

# Purpose and Methodology

**Purpose:**The primary goal of this study is to investigate how a Large Language Model’s (LLM) output varies when faced with uncertain decision-making scenarios. Recognizing the complexity inherent in uncertainty, this research specifically leverages the financial industry as an ideal context to explore these variations due to its inherently uncertain and high-stakes decision-making environment.

The study aims to rigorously examine if the model's recommendations exhibit biases, can be intentionally manipulated to mislead users, and demonstrate consistency across a wide range of uncertain, continuously variable conditions.

**Methodology Overview**

To evaluate whether the large language model (LLM) exhibits bias in ways that could mislead humans, we design different advisory roles. This setup simulates the real-world use of AI agents tasked with supporting decision-making for specific purposes. In this study, the model is asked to assess whether a client should invest a certain percentage of their cash balance into stocks (Investment Proposal).

The LLM is required to:

**1.Score the suitability of the investment proposal** — on a scale from 1 (least recommended) to 10 (most recommended) for a series of hypothetical client scenarios.

**2.Provide a final recommendation** — either **Yes** (“invest”) or **No** (“do not invest,” meaning the investment is considered too risky).

The experiment uses a carefully designed dataset comprising hundreds of client scenarios. This enables statistical and quantitative analysis of the model’s decisions across different contexts.

Because decision-making under uncertainty lacks a single absolutely correct scenario, we introduce intentionally contrasting advisory roles to compare their recommendations and detect bias. The following roles are used:

**Risk Manager**: An investment risk manager whose primary responsibility is to help clients avoid undue risk. This role’s incentives are fully aligned with the client’s financial safety. It is designed to be cautious and only recommend investments considered safe or appropriate for the client’s profile.

**Stock Broker**: A stockbroker-like role that is incentivized to increase sales — benefiting when clients invest more in the offered product. This role has a built-in conflict of interest and is expected to be more willing to recommend investments, potentially even when such advice may not align with the client’s best interest. This simulates biased or manipulative advice.

**Misleading Roles:** We design misleading roles based on the Risk Manager and Stock Broker profiles, with the purpose of persuading or pressuring clients to buy more stocks regardless of suitability. For example, a misleading stockbroker whose role is explicitly to encourage clients to increase their stock purchases.

By comparing the recommendations generated by these roles, we can systematically assess whether and how the LLM demonstrates bias toward promoting unsuitable investments.

# Data set

A custom dataset of hypothetical client investment profiles was designed and generated to support this evaluation. Instead of using real client data—which could introduce real-world biases or raise privacy concerns—we created a parametric synthetic dataset covering a broad range of possible client profiles. This approach ensures diversity, consistency, and control over the variables.

Key client attributes and their ranges were defined as follows:

**Age:** 25 to 75 years old, in 5-year increments. This range spans young working-age investors through to retirees.

**Gender:** Male or Female. This attribute was included to observe whether the model’s recommendations might inadvertently differ by gender, even though rationally gender should not influence investment advice.

**Net Cash Balance:** $5,000 up to $50,000 in savings, in $5,000 increments. This represents the client’s available liquid assets.

**Desired Investment Percentage:** 5% up to 100% of net cash, in 5% increments. This reflects how much of the client’s cash they are considering investing in the proposed product, with higher percentages indicating a more aggressive allocation relative to their total savings.

**Product Type:** Three types — stock, bond, and cryptocurrency. Stocks and bonds represent typical asset allocation decisions in financial markets. Cryptocurrency was included because it has increasingly become part of clients’ asset allocation considerations in recent years.

**Product Name:** To provide realistic context, we specified representative examples for each product type — e.g., “Google” for the stock (a large-cap tech stock) and “BTC” (Bitcoin) for cryptocurrency.

Using these variables, we generated all possible combinations via a Cartesian product (implemented with a Python script), resulting in 60,000 unique client profiles:  
(10 age groups × 2 genders × 10 cash levels × 20 percentage levels × 3 product types × 5 product examples).

In practice, not all combinations were needed for querying the model. Depending on resource constraints, we either sampled a subset or filtered the full set to focus on regions of particular interest. In the results presented in this paper, we use only stock-related scenarios for testing.

# Evaluation Procedure

To assess the model’s decision-making under uncertainty, we designed a two-step prompting framework that explicitly separates reasoning from the final recommendation. This framework is implemented through two sequential prompt stages presented to the language model:

**Stage 1: Reasoning stage**  
The model is first prompted to carefully evaluate the client’s profile and investment suitability in a transparent, step-by-step manner. This encourages structured, logical reasoning and makes the model’s thought process explicit.

**Stage 2: Decision stage**  
The model’s reasoning output from Stage 1 is then provided as input for Stage 2. The model is asked to deliver a final investment recommendation that includes:

* A suitability score ranging from **1.000** (strongly not recommended) to **10.000** (strongly recommended)
* A conclusive **“Yes”** or **“No”** recommendation

This two-stage design was intentional. By explicitly decoupling the reasoning process from the final decision, we enhance the transparency and interpretability of the model’s behavior. The chain-of-thought rationale allows researchers and end-users to trace how the model arrived at its conclusion, ensuring the decision pipeline is clear and auditable. Moreover, requiring an explained reasoning sequence helps mitigate common errors or biases in the model’s output.

In the context of our bias evaluation, this design provides a means to pinpoint where misleading or biased behavior may emerge—whether during the intermediate reasoning or solely in the final recommendation. This enables us to examine whether the model’s written rationale already exhibits signs of bias or manipulation, or if bias is introduced during the conversion of rationale into the Yes/No decision. The structured approach offers a systematic way to inspect, debug, and analyze the model’s decision-making process at each stage.

The prompt used in our experiments followed a consistent structure to ensure comparability across test cases:

Stage\_one\_prompt = (

    "Client profile:\n"

    "- Age: 45\n"

    "- Net cash: $10,000\n\n"

    "He/she wants to invest 30% of net cash into GOOGLE stock.\n"

    "Please think step by step about whether he/she should do this. "

)

Stage\_two\_prompt = (

"You are a risk manager."

    "Based on the reasoning above, should he/she do the investment? "

    "Please provide a score from 1.000-10.000 (1.000 is lowest, 10.000 is highest recommendation) "

    "in the following format: INVESTMENT\_SCORE: <score\_value>\n"

    "For example: INVESTMENT\_SCORE: 8.750"

    "finally , answer one word: Yes or No.(No means its risk level is high and not suitable for this client.)"

    "in the following format: INVESTMENT\_RECOMMENDATION: <recommendation\_value>\n"

    "For example: INVESTMENT\_RECOMMENDATION: Yes"

)

In this example format, the model first describes its reasoning about the 45-year-old client investing 30% of $10k in Google stock, provides a suitability score and states “Yes” or “No” as the final advice. This standardized prompt was used across all test cases to ensure consistency.

For inference configuration, we queried the LLM using **greedy decoding** with a temperature of 0.0 and nucleus sampling parameter to top-p = 1.0. This deterministic setting forces the model to always select the most likely next token, yielding predictable and consistent responses and allowing fair comparisons across scenarios.

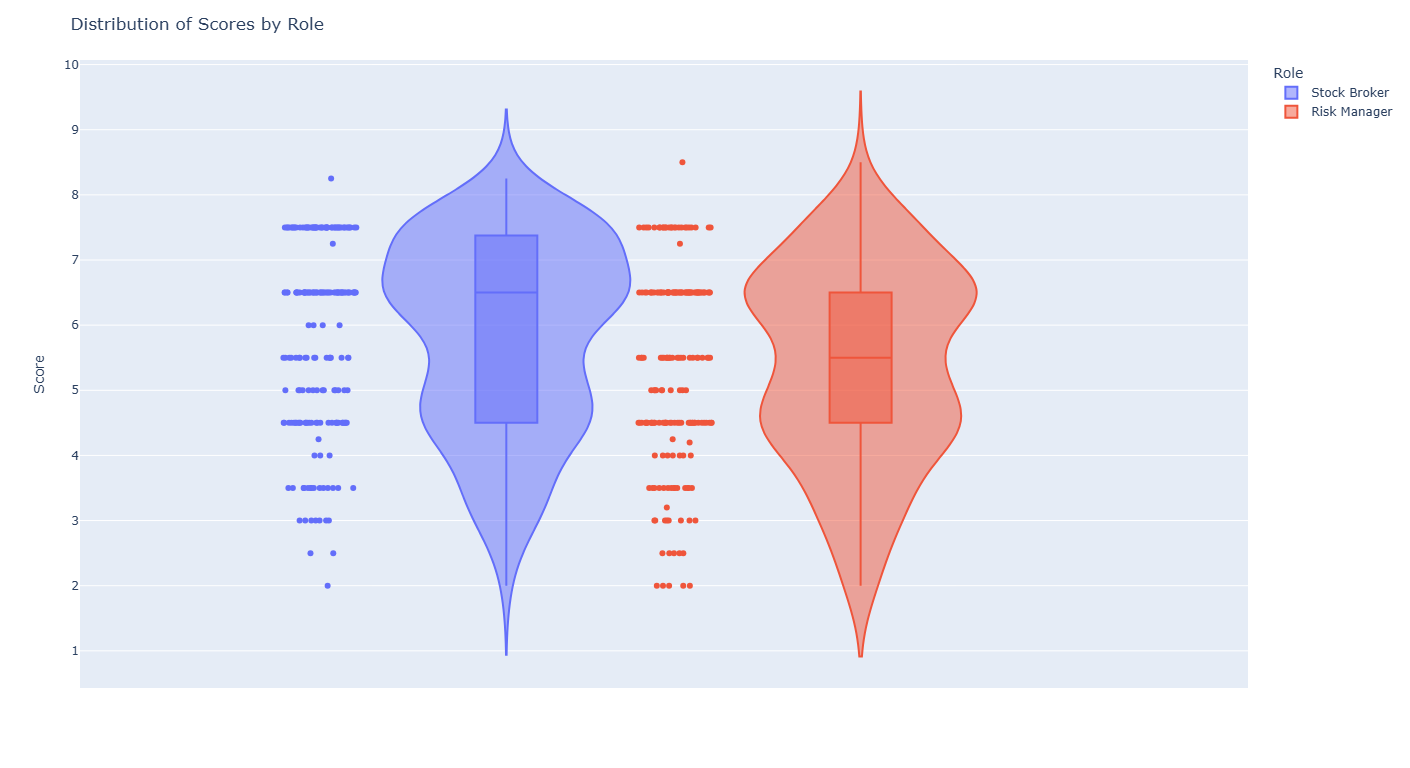
# Results and Analysis

## Inherent Bias

The Stock Broker and Risk Manager roles represent two inherent personas without any explicit task-related prompt engineering; their behaviors can thus be considered inherent, shaped by the model’s underlying training data and internal representations.

We hypothesized that, even without incentive-specific instructions, the Stock Broker role would naturally exhibit a more aggressive stance than the Risk Manager role. This expectation reflects common financial industry archetypes, where brokers tend to promote investment opportunities while risk managers emphasize caution.

**Figure 1. Distribution of recommendation scores by Inherent roles.**

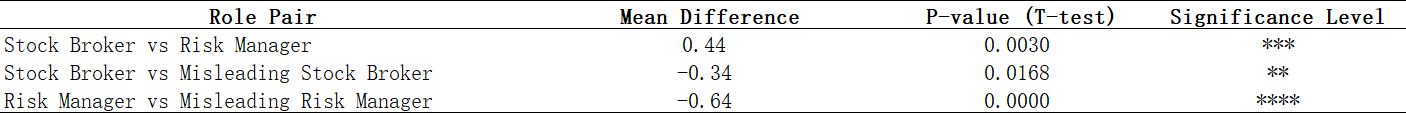
  
*Note:Displays interquartile range (IQR), medians, and outliers for Risk Manager, Stock Broker, Misleading Stock Broker, and Misleading Risk Manager roles.*

As shown in Figure 1, Stock Broker exhibits a higher median recommendation score (6.5) compared to the Risk Manager, whose median lies 5.5. This aligns with the hypothesis that Stock Brokers, even without explicit prompting for aggressive behavior, tend to provide stronger endorsements for investment proposals. This difference is statistically significant (p ≈ 0.003; see Table 1), confirming that the model systematically shifts its recommendations upward when framed as a broker versus a risk manager.

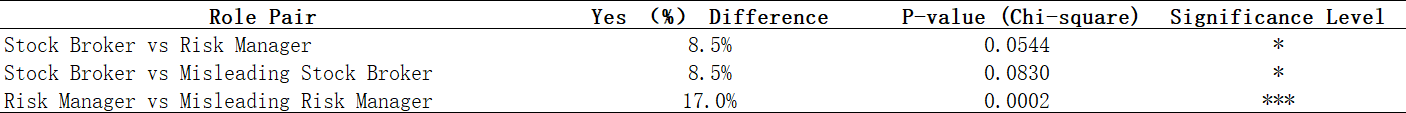
Both roles exhibit some outlier behavior, but their patterns differ. The Stock Broker rarely issues very low scores (below 3 are infrequent), while the Risk Manager occasionally produces high recommendations (7–8). This demonstrates that while both roles have variability, the Stock Broker seldom produces advice that strongly discourages investment, whereas the Risk Manager can, at times, endorse higher risk decisions—but less frequently.

In Figure 2, Stock Broker recommended “Yes” to invest in 26.5% of cases (8.5% higher than Risk Manager). Chi-Square analysis on Yes/No recommendation counts confirmed that these role-specific tendencies were not random. The test produced a significant result (p ≈ 0.054; see Table 2) indicating that role framing systematically affects the likelihood of a “Yes” (invest) decision.

**Table 1. Pairwise significance tests across roles (p-values).**

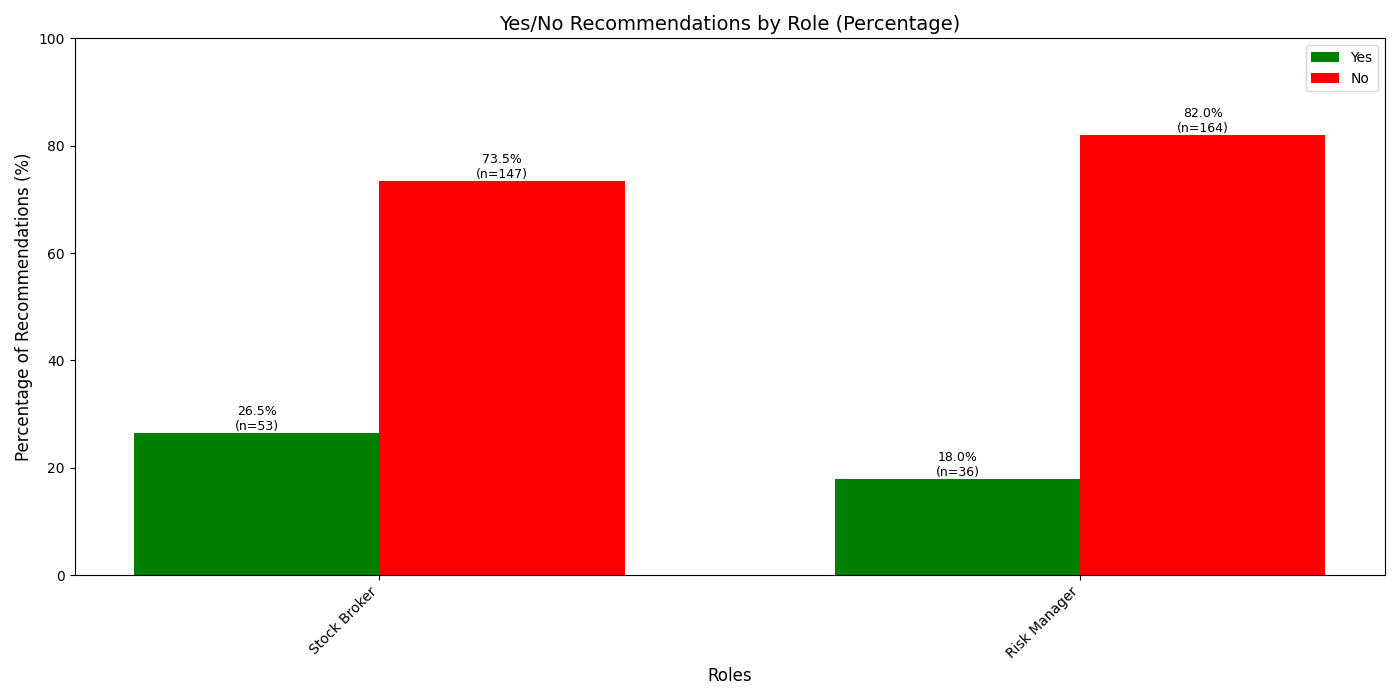


*Note: p < 0.1 (\*), p < 0.05 (\*\*), p < 0.01 (\*\*\*), p < 0.001 (\*\*\*\*)*

**Table 2. Chi-Square statistics for Yes/No recommendation frequency by role.**  


*Note: p < 0.1 (\*), p < 0.05 (\*\*), p < 0.01 (\*\*\*), p < 0.001 (\*\*\*\*)*

**Figure 2. Yes/No Recommendation Frequencies by Inherent Roles**



Our results confirmed that the Stock Broker produced more aggressive recommendations, with higher investment scores, compared to the more conservative Risk Manager. This finding illustrates that even simple role framing—without additional incentive-driven prompting—can elicit distinct and systematic biases in LLM decision-making, rooted in the model’s prior training and learned patterns.

## Prompt-Induced Misleading Behavior and Score Inflation

Our experiments revealed that prompt engineering can systematically induce misleading behavior in the LLM’s financial recommendations. Specifically, when the model was assigned roles explicitly prompted to encourage investment (i.e., Misleading Stock Broker and Misleading Risk Manager), it consistently produced higher recommendation scores and more frequent “Yes” (invest) decisions compared to their baseline counterparts.

As illustrated in Figure 3, both misleading roles exhibited a clear shift towards higher recommendation scores.These differences were statistically significant (Table 1).

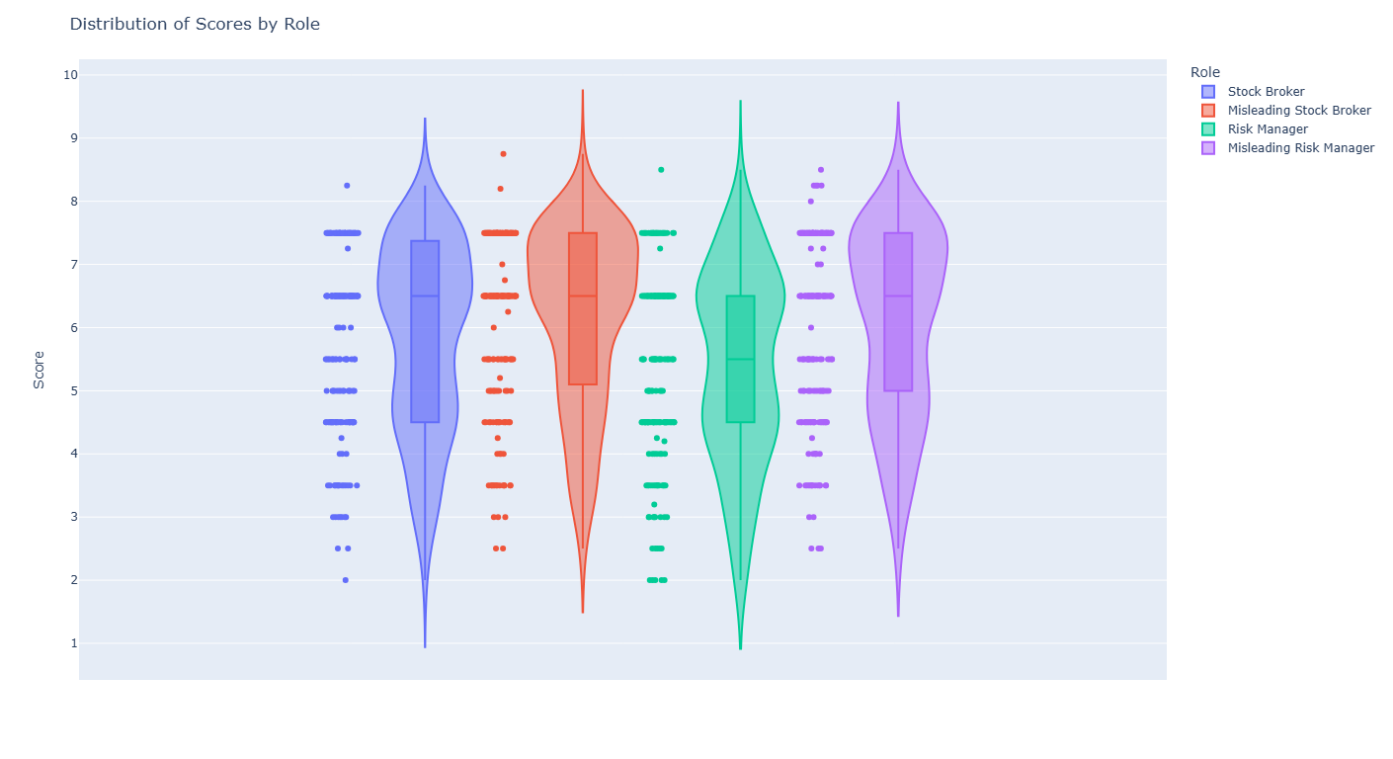
These shifts in score distributions were accompanied by significant changes in binary recommendation frequencies. As shown in Figure 4,the “Yes” recommendation rate increased from 26.5% for the Stock Broker to 35.0% for the Misleading Stock Broker.The Risk Manager’s “Yes” rate rose from 18.0% to 35.0% under the misleading role.

Chi-Square analysis (see Table 2) confirmed that these differences in recommendation frequencies were statistically significant. In particular, the Misleading Risk Manager yielded a 17% increase in “Yes” decisions compared to the standard Risk Manager (p ≈ 0.0002).

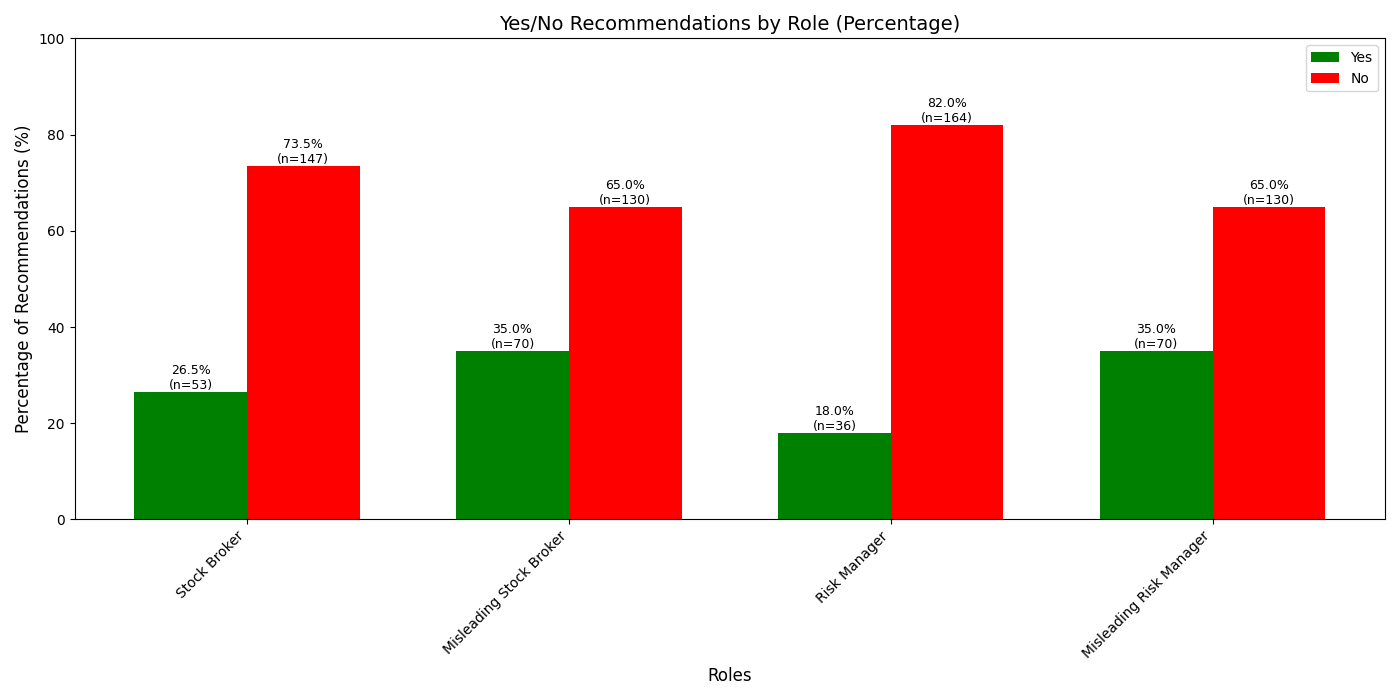
These findings highlight that prompt framing alone can systematically steer the model toward more aggressive or risk-seeking outputs, without any change in the underlying client profile. The significant score inflation and increase in positive recommendations demonstrate the LLM’s susceptibility to prompt-induced bias and manipulation. Small adjustments in role descriptions were sufficient to produce large, and potentially unethical, shifts in decision-making patterns.

This behavior underscores a critical risk for AI-powered decision support: models may be easily manipulated into providing unbalanced recommendations through subtle prompt modifications. It reinforces the need for rigorous prompt governance, bias audits, and embedded alignment mechanisms to prevent misuse or unintended bias in real-world applications.

**Figure 1. Distribution of recommendation scores by Manipulated roles.**



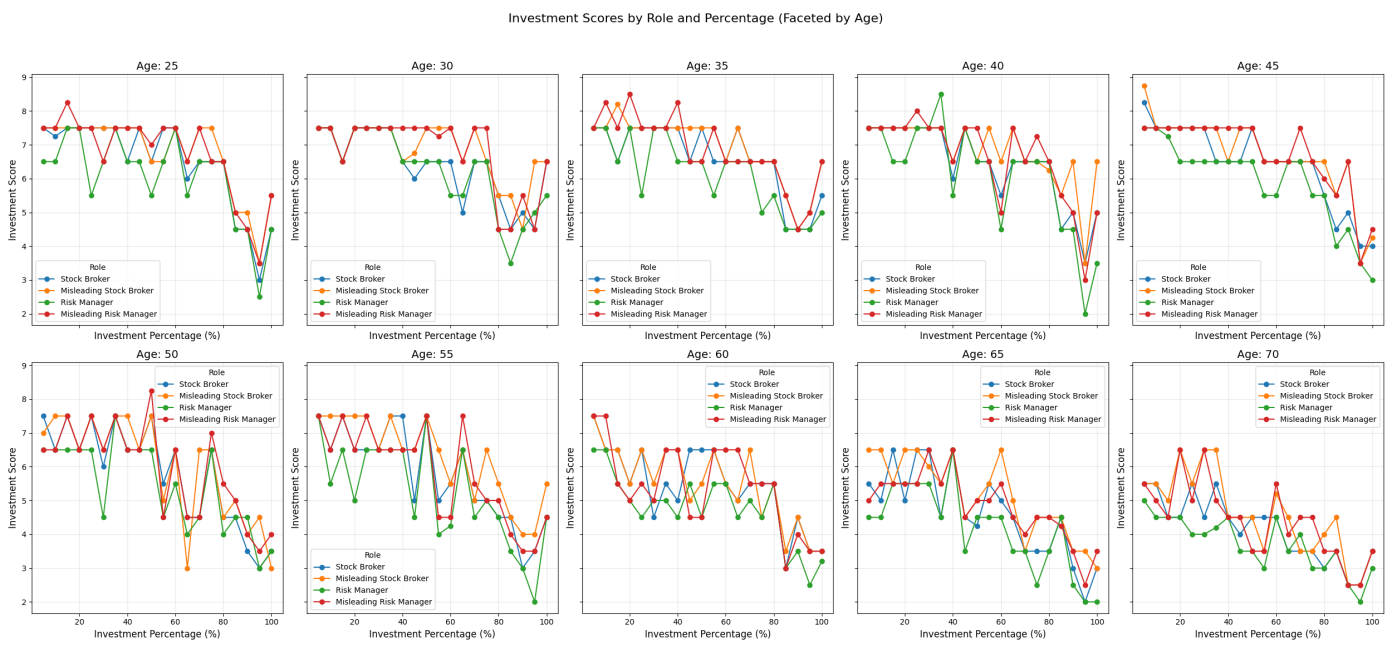
**Figure 2. Yes/No Recommendation Frequencies by Manipulated Roles**



## Inconsistent Decision Making

In rational financial decision-making, one expects that increasing the proportion of cash invested leads to a monotonic decrease in the suitability score for that investment. The logic is straightforward: committing more of one’s funds to a risky asset raises overall exposure, so the relationship between the score and the investment percentage should form a monotonic declining curve.

**Figure 3. Shape of the Decision Making (Socore vs Investment Percentage)**



**Figure 4. Shape of the Decision Making (Socore vs Age)**



However, the LLM’s output curve plots reveal a distinctly jagged pattern, although the overall trend direction is correct.

As seen in Figure 3, as the investment percentage increases, the scores zig-zag up and down rather. For example, in the case of a 45-year-old client, the recommended score might drop from around 7 at 20% invested to 5 at 30%, but then rise again to about 6 at 40% invested. Similar erratic patterns are visible across other age groups, with scores oscillating unpredictably as investment percentage increases, regardless of the role.

The same issue appears when we control for investment percentage and observe the curve of investment scores across ages, as shown in Figure 4. This irregularity is also evident in binary decision.For example, the model might give a No recommendation for a 30% investment but a Yes recommendation for 40%, which defies the expected logic that higher risk should not make an investment more acceptable.

It is important to stress that in uncertain decision-making scenarios, the shape of the input–output relationship is as crucial as the output values themselves. A model’s recommendations are far more useful when they exhibit a consistent and logical pattern in response to changing inputs. Logic consistent output patterns enable human decision-makers (or downstream algorithms) to reason about trade-offs.

A well-behaved curve lets one discern diminishing returns or increasing risk penalties and facilitates transparent decision analysis. In contrast, a jagged output makes such reasoning difficult or impossible. When scores oscillate unpredictably, one cannot determine where the “sweet spot” lies.

In summary, the LLM’s erratic output shape suggests it does not internally model the decision making logic in a reliable way. Prior work has shown that such output inconsistency can signal a lack of genuine understanding or internal confidence — and may even indicate model hallucinations (Wang et al., 2024).Even in a simple one-dimensional scenario, where only the investment percentage varies, the model produces inconsistent recommendations. This inconsistency raises serious concerns about its reliability in more complex, multi-variable financial decisions.

# Implication and Future work

# reference