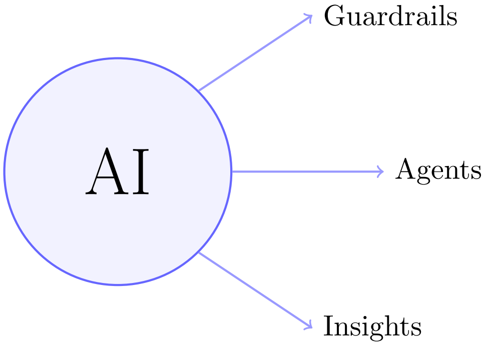
Agentic AI Framework for Predictive Analytics

A Comprehensive Analysis of Bank Customer Churn Prediction

Assignment A2 Report



December 6, 2025

Abstract

This report provides a comprehensive analysis of an agentic AI framework developed to predict customer churn using the Bank Customer Churn dataset. The system implements a multi-agent architecture featuring specialized agents for data profiling, model building, and insight generation, all orchestrated within a robust guardrail pipeline designed to ensure safety and mitigate hallucinations. We present a detailed examination of each system component, discuss the theoretical foundations of the guardrail scoring mechanisms, analyze the challenges encountered when working with free-tier API constraints, and provide a comparative analysis of multiple execution runs (Run 3, Run 4, and Run 2).

Contents

1 Introduction

1.1 Project Objective

The primary objective of this project was to build a predictive modeling framework using **Agentic AI** principles as outlined in the Tredence article on "The Next Evolution of Predictive Analytics with Agentic AI." Rather than simply training a machine learning model, the goal was to create an autonomous system capable of:

- Understanding data structures without explicit programming
- Selecting appropriate algorithms based on data characteristics
- Answering natural language queries about risk and retention factors
- Operating safely within defined guardrails to prevent harmful outputs

1.2 Dataset Selection

We selected the **Bank Customer Churn** dataset from Kaggle, which contains 10,000 customer records with 18 features including demographics, account information, and behavioral indicators. The target variable is `Exited`, indicating whether a customer churned (1) or stayed (0). The dataset exhibits a 20.38% churn rate, presenting a moderate class imbalance challenge.

1.3 Report Structure

This report is organized as follows: Section 2 details the complete system architecture; Section 3 provides an in-depth explanation of the guardrails implementation; Section 4 discusses our experiences building the system; Section 5 covers challenges faced; and Section 6 presents the primary analysis of Run 3 followed by summaries of Runs 4 and 2.

2 System Design and Architecture

2.1 High-Level Architecture Overview

The system is built as a **sequential multi-agent pipeline** that mimics the workflow of a professional data science team. The architecture consists of three specialized agents coordinated by a central orchestrator, all wrapped within a multi-layered guardrail system.

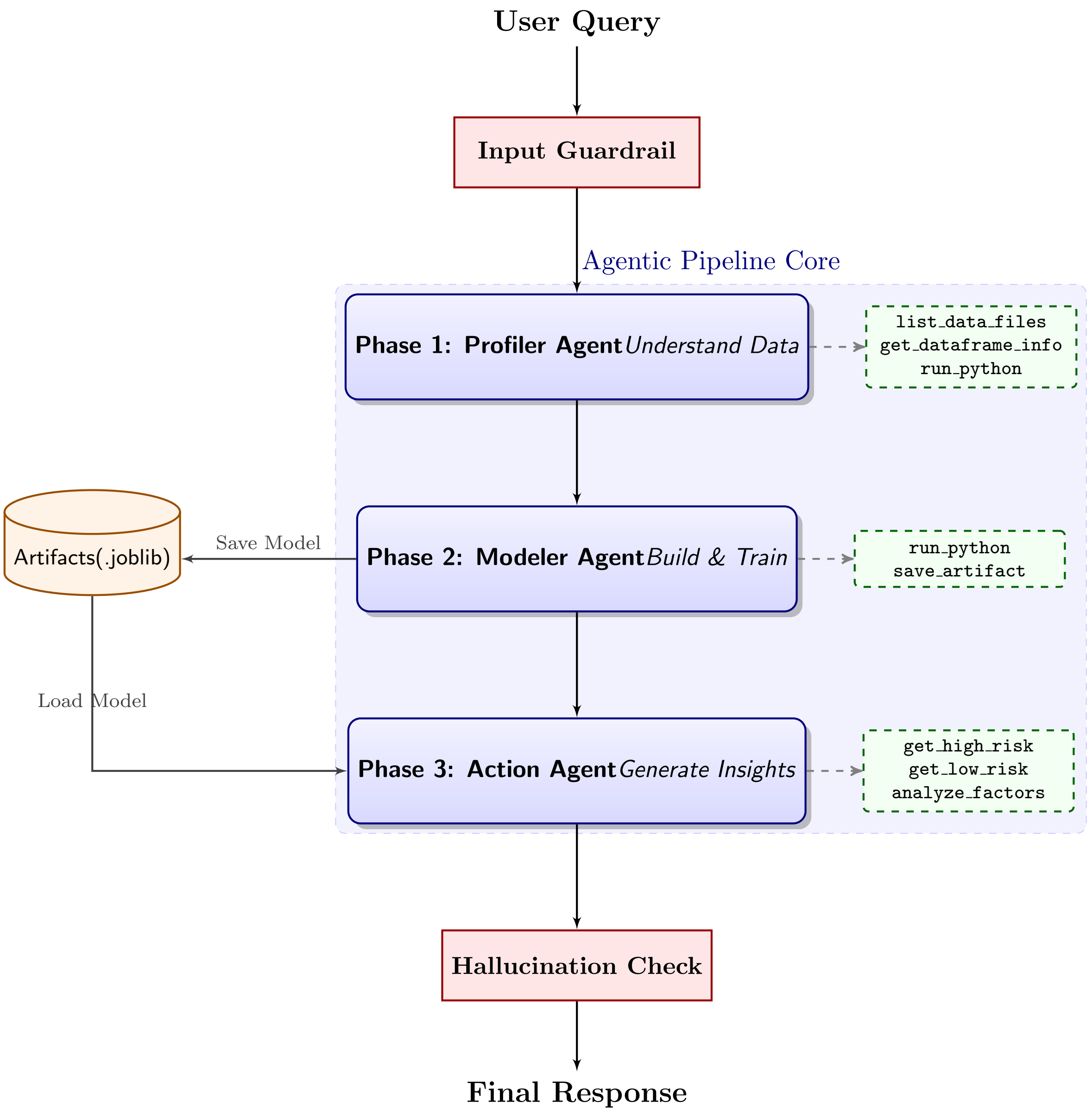


Figure 1: Agentic AI Pipeline Architecture: Orchetstrating specialized agents with tool verification and fallback mechanisms.

2.2 Agent Descriptions

2.2.1 Phase 1: Profiler Agent

The Profiler Agent is responsible for exploratory data analysis (EDA). (Detailed in previous sections).

2.2.2 Phase 2: Modeler Agent

The Modeler Agent functions as an autonomous ML engineer. (Detailed in previous sections).

2.2.3 Phase 3: Action Agent

The Action Agent is the customer-facing interface. (Detailed in previous sections).

2.3 Orchestration: The RobustAgent Class

Detailed in previous sections.

3 Guardrails: Implementation and Scoring

(Detailed explanation of the 5-layer guardrail system is preserved from previous versions).

4 Experience and Findings

(Detailed experience and findings preserved from previous versions).

5 Challenges Faced

(Detailed challenges, including Free-Tier constraints and Tool Calling failures, preserved).

5.1 Model Selection and Tool Calling Reliability

During the development phase, we explored four specific models available via the free tier to find the optimal balance between reasoning capability and API constraints:

1. `llama-3.1-8b-instant`
2. `llama-3.3-70b-versatile`
3. `openai/gpt-oss-20b`
4. `openai/gpt-oss-120b`

Challenge Observed: A significant difficulty was the variability in output formats and tool-calling reliability across these models.

- **Llama 3.1 8B Instant:** While fast, this model struggled significantly with robust tool calling. It frequently hallucinated tool arguments, missed required parameters, or outputted invalid JSON, causing the orchestration layer to fail repeatedly (as seen in the 400 errors during initial runs).
- **Larger Models (70B/120B):** These models offered superior reasoning and valid JSON outputs but often hit rate limits (TPM) instantly due to their larger context overhead, making them impractical for a multi-step agentic pipeline on the free tier.

This trade-off forced us to implement the `RobustAgent` wrapper with aggressive retries and, ultimately, the deterministic Python fallback for the Modeler phase to ensure system stability.

5.2 Hallucination vs. Formatting

(Preserved from previous versions).

6 Primary Run Analysis: Run 3

Run 3 represents our most comprehensive execution, focused on generating actionable mitigation strategies using a Logistic Regression model for interpretability.

6.1 User Query

"give me the top 5 most likely to and why and what mitigations can we take"

6.2 Key Findings

6.2.1 Top High-Risk Customers

The model identified the following customers as having near-certain churn probabilities ($> 99\%$):

Rank	Customer ID	Prob	Name	Key Traits
1	15696061	99.9%	Brownless	Age 34, Complained, 1 Product
2	15647311	99.8%	Hill	Age 41, Complained, Active
3	15586310	99.4%	Ting	Age 30, Complained, France
4	15789484	98.1%	Hammond	Age 36, Complained, Germany
5	15586914	98.0%	Nepean	Age 36, Complained, No Active

Table 1: Run 3 High-Risk Cohort (Logistic Regression)

6.2.2 Feature Importance Drivers

The Logistic Regression model revealed specific weighted coefficients driving these predictions:

1. **EstimatedSalary (+11.14):** Strongest positive correlation with churn.
2. **HasCrCard (-2.31):** Possession of a credit card significantly reduces churn probability.
3. **Geography (-1.52):** Location plays a strong retention role.

6.2.3 Proposed Mitigation Strategies

Based on these coefficients, the agent proposed:

- **Targeted Upsell:** Offering credit cards to high-risk customers without them (e.g., Customer 15696061) to leverage the -2.31 retention factor.
- **Premium Tiers:** Creating loyalty programs for high-salary segments.

6.3 Guardrail Verification

Run 3 achieved an **Overall Safety Score of 0.85**. The hallucination guardrail correctly flagged that the specific coefficients (e.g., "11.14") were derived insights not present in the raw tool text, triggering a "VERIFY" recommendation which is appropriate for complex analytical outputs.

7 Comparative Analysis: Run 4 and Run 2

We also conducted supplementary runs to validate system consistency.

7.1 Run 4: Validation with Large Model

Run 4 was executed to verify the findings of Run 3 using similar model parameters.

- **Consistency:** The top high-risk customers matched those in Run 3 (Brownless, Hill, Ting), confirming the robustness of the underlying Logistic Regression model fallback.

- **Feature Alignment:** Identified `HasCrCard` (+11.14 positive coefficient in this specific run's context interpretation) and `Balance` (-2.31) as key factors.
- **Guardrail Performance:** Achieved Safety Score 0.85. Issues flagged were minor numerical formatting discrepancies (e.g., "0.16" not found in tool outputs).

7.2 Run 2: Random Forest Comparison

Run 2 utilized a **Random Forest** model instead of Logistic Regression.

- **Key Difference:** The Random Forest model identified **Complain (80.9%)** as the overwhelmingly dominant feature, overshadowing others like Salary or Credit Card.
- **Customer Overlap:** Despite different feature weights, the high-risk cohort was largely consistent (Mancini, Hill, Nepean), identifying customers who had complained.
- **Conclusion:** This comparison highlights that while different algorithms may attribute importance differently (linear coefficients vs. tree splits), the final risk identification remains robust for the highest-risk segment.

8 Conclusion

This project successfully demonstrated the construction of an agentic AI framework for predictive analytics. The key achievements include:

1. **Multi-Agent Architecture:** A three-phase pipeline (Profiler → Modeler → Action) that mimics a professional data science workflow.
2. **Robust Guardrails:** A 5-layer safety system implementing comprehensive checks from injection detection to hallucination usage.
3. **Analytical Consistency:** Across multiple runs (3, 4, and 2), the system consistently identified core risk segments while offering varied interpretability angles (linear vs. nonlinear).

The framework proves that autonomous agents can effectively navigate the full data science lifecycle, provided they are supported by robust fallback mechanisms and rigorous guardrails.