International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Evaluation of Vertex AI Agent for Investment Product Support: Enhancing Customer Service in Asset Management

Ananth Majumdar

Email: *thisisananth[at]gmail.com*

Abstract: ***This study explores the application of a Vertex AI agent designed to support asset management customers by providing detailed information about investment products through a natural language interface. We discuss the agent's architecture, which integrates an advanced language model and API, enabling user-friendly access to data on asset classes, performance metrics, fees, risks, and liquidity. The paper evaluates the agent's accuracy, error-handling capabilities, and iterative improvements, illustrating how targeted training enhances the agent's performance in responding to customer inquiries. Future improvements for addressing calculation-based queries and data fabrication issues are also proposed.***

Keywords: Vertex AI, Investment Product Support, Asset Management, Customer Service AI, Natural Language Processing, API Integration, Machine Learning in Finance, AI Performance Evaluation, Chatbot for Financial Services, Automated Customer Support

1. Introduction

In the rapidly evolving landscape of financial services, artificial intelligence (AI) is playing an increasingly crucial role in enhancing customer experiences and operational efficiency. This study focuses on the design, development and implementation of a Vertex AI agent designed to assist customers of an asset management company by providing a natural language interface using LLM for interpreting and responding to user's queries

The primary objective of this AI agent is to streamline the process of product inquiries, allowing customers to quickly access relevant information about various investment options. By leveraging LLMs, the asset management company aims to improve customer satisfaction, reduce the workload on human customer service representatives, and potentially increase customer engagement with their products.

1.1 Related Work

In the financial sector, AI is increasingly used to enhance customer service through applications such as rapid customer onboarding using biometric technologies and personalized customer relationship management. These AI-driven systems provide instant credit application responses, conduct efficient mortgage affordability checks, and deliver tailored services based on comprehensive data analyses, significantly improving the customer experience (Zetzsche et al., 2020). [2]

Artificial Intelligence (AI) has significantly transformed various aspects of asset management. As discussed by M. Bartram et al. (2020) [3], AI enhances portfolio management through improved return and risk estimates, utilizing techniques such as artificial neural networks (ANNs) and natural language processing (NLP) for predictive analytics. In trading, AI drives algorithmic strategies, optimizing execution and minimizing costs. Risk management benefits from AI's ability to process vast amounts of data, including qualitative information from news and social media, to forecast risk factors and validate traditional models. The emergence of robo-advisors has democratized investment advice, leveraging AI to provide personalized recommendations to retail investors.

In this paper, we apply AI for customer service in asset management, specifically to answer questions about products supported by the asset management firm

2. Methodology

The Vertex AI agent for investment product support is designed as an intelligent system to handle user queries about investment products efficiently and accurately. The architecture consists of several key components that work together to process user inputs and provide informative responses. Here's a detailed description of the system:

2.1 System Architecture

1) **User Interface:**
- It is a web client that serves as the entry point for user interactions.
- Allows users to input their queries about investment products.

2) **Vertex AI Agent: This is the core component of the system, encompassing three critical subcomponents:**

a) **Intent Detection:**
   - Analyzes the user's input to understand the intent behind the query.
   - Utilizes the Gemini-1.0-pro-001 Language Model for natural language understanding.

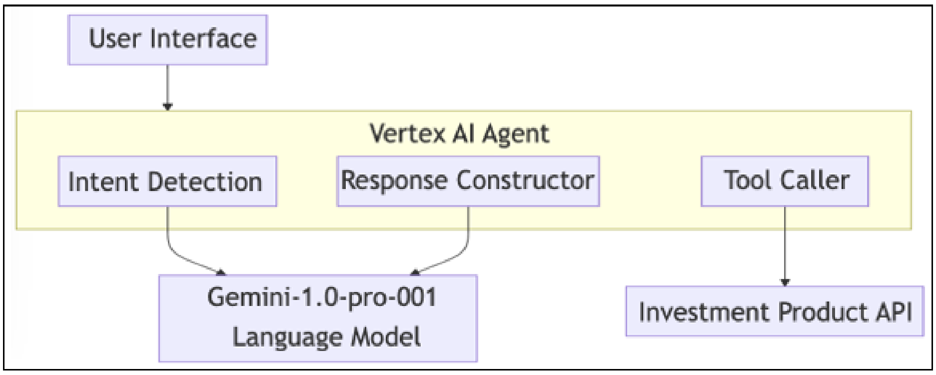
b) **Response Constructor:**
   - Formulates the final response to be sent back to the user.
   - Leverages the Gemini-1.0-pro-001 Language Model to generate coherent and contextually appropriate responses.

c) Tool Caller:
- Interfaces with the Investment Product API when specific product information is required.
- Determines when to call the API based on the detected intent and query requirements.

3) Gemini-1.0-pro-001 Language Model:
- A sophisticated AI model that powers both the Intent Detection and Response Constructor.
- Configured with specific parameters:
  - Input Token Limit: Up to 32k tokens
  - Output Token Limit: Up to 2048 tokens
  - Temperature: 0.9 (balancing creativity and coherence in responses)

4) Investment Product API:
- Serves as the primary data source for detailed investment product information.
- Accessed by the Tool Caller component when specific product data is needed to answer user queries.

By leveraging the advanced language understanding and generation capabilities of the Gemini-1.0-pro-001 model, combined with access to specific product data through the API, the system can provide accurate, context-aware responses to users' investment product inquiries.



2.2 Investment Product API

The agent accesses product information through a RESTful API that provides comprehensive data about the company's investment products. The API is defined using OpenAPI 3.0.0 specification and hosted on Google Cloud Functions.

2.2.1 API Endpoints

The API provides the following main endpoints:
1) /products: Retrieves all products or filters by criteria (asset class, minimum return, maximum risk).
2) /products/{productId}: Gets detailed information about a specific product.
3) /performance: Retrieves performance data for products.
4) /fees: Gets fee structure for products.
5) /risks: Retrieves risk assessment for products.
6) /liquidity: Gets liquidity information for products.

2.2.2 Data Schema

The API returns data in JSON format. The main data structures include:

1) Product:
- id: string
- name: string
- assetClass: string
- investmentType: string
- minimumInvestment: number

2) Performance:
- productId: string
- timeframe: string (1year, 3year, 5year)
- return: number

3) FeeStructure:
- productId: string
- managementFee: number
- performanceFee: object (optional)
  - rate: number
  - benchmark: string

4) Risk:
- productId: string
- riskLevel: string
- riskFactors: array of strings

5) Liquidity:
- productId: string
- redemptionFrequency: string
- settlementPeriod: string
- lockupPeriod: string
- redemptionFees: string

2.2.3 Sample Product Data

Here's an example of the data structure for a typical investment product:

```
{
  "name": "Global Equity Growth Fund",
  "assetClass": "Equity",
  "investmentType": "Actively managed global stock portfolio",
  "returns": {
    "oneYear": 12.5,
    "threeYear": 10.8,
    "fiveYear": 9.7
  },
  "feeStructure": {
    "managementFee": 0.75,
    "performanceFee": {
      "rate": 15,
```

```
    "benchmark": "MSCI World Index"
   },
   "minimumInvestment": 50000
  },
  "risk": {
   "riskLevel": "Moderate",
   "riskFactors": ["Market risk", "Currency risk", "Economic risk"]
  },
  "liquidity": {
   "redemptionFrequency": "Daily",
   "settlementPeriod": "T+3",
   "lockupPeriod": "None",
   "redemptionFees": "None"
  }
}
```


This comprehensive data structure allows the AI agent to access detailed information about each investment product, including its performance, fee structure, risk assessment, and liquidity options. The API's flexible query parameters enable the agent to filter and retrieve specific information based on user queries, enhancing its ability to provide accurate and relevant responses.

2.3 Agent Configuration

The agent was configured with a specific goal and set of instructions to guide its interactions with users. The goal was defined as:

"Help customers to query and find more about available investment products, their performance, fees, risks and liquidity options. If they ask anything other than this, tell them that you can only help with investment products. Also only answer based on the data from the tool. If the tool doesn't have information to answer a question, respond accordingly."
These will be included in the prompt for the LLM

The instructions provided to the agent included:
- Greeting users and asking how to help them
- Basing responses solely on information retrieved from the API tool
- Using a specific tool call format to access the API
- Recognizing and responding to queries about specific asset classes
- Limiting responses to products supported by the tool
- Avoiding assumptions about tool parameters
- Thanking users for their business

2.4 Evaluation Methodology

To assess the performance of the AI agent, we developed a comprehensive evaluation process:
1) Test Set Creation: We collected a set of 50 questions from actual users, representing a diverse range of queries about investment products. These questions were designed to cover various aspects such as product information, performance metrics, fees, risks, and liquidity options.
2) Automated Testing: We created a test script to automatically run these 50 questions through the AI agent and capture its responses.
3) Response Validation: Each response from the agent was validated for accuracy against the known correct information from the API.
4) Error Handling Assessment: We also monitored and recorded any instances where the agent returned temporary errors or required multiple attempts to provide a response.
5) Accuracy Calculation: The overall accuracy was calculated as the percentage of correct responses out of the total number of questions, excluding temporary errors that were resolved on retry.

2.5 Training and Optimization

To enhance the agent's performance and address initial limitations, we implemented a targeted training process using specific examples. This approach helped to refine the agent's understanding of industry-specific terminology and improve its ability to interpret user queries accurately.

2.5.1 Example-Based Training

We identified key areas where the agent was initially underperforming and created targeted examples to address these issues. These examples are used by the LLM as few-shot examples to improve the output.
Some notable examples include:
1) Alternative Investments Terminology: We observed that the agent struggled to identify queries related to alternative investments because the API used the shorthand "Alts" for this asset class. By providing examples that paired user queries about "alternative investments" with the corresponding "Alts" API input, we significantly improved the agent's ability to correctly identify and retrieve information about this asset class.
2) Performance Metrics Interpretation: To help the agent better understand and interpret performance-related queries, we provided examples that demonstrated how to translate qualitative terms into quantitative API parameters. For instance, we trained the agent to associate "high growth" with a minimum return greater than 7%. This allowed the agent to more accurately filter and retrieve products based on performance criteria.

2.5.2 Iterative Refinement

The training process was iterative, with regular evaluations of the agent's performance leading to the identification of new areas for improvement. As new edge cases or misinterpretations were discovered, additional examples were created and incorporated into the training set.

This approach allowed us to:
1) Improve the agent's natural language understanding, particularly for industry-specific jargon and colloquial terms.
2) Enhance the agent's ability to map user queries to appropriate API parameters.
3) Reduce the occurrence of misinterpretations and incorrect data retrievals.

2.5.3 Impact on Performance

The example-based training significantly improved the agent's accuracy and relevance in responding to user queries. It was particularly effective in addressing the following areas:

1) Asset Class Identification: The agent became more adept at correctly identifying and querying for various asset classes, including those with industry-specific abbreviations or alternative names.
2) Performance Metric Queries: The agent's ability to interpret and respond to queries about fund performance improved, with a better understanding of how to translate qualitative descriptions into quantitative parameters for the API.
3) Risk Assessment: By providing examples of how different risk levels correspond to API parameters, we enhanced the agent's ability to accurately discuss and compare the risk profiles of various investment products.

This targeted training approach proved to be a crucial step in optimizing the agent's performance and ensuring its responses aligned more closely with user expectations and industry standards.

3. Results

The evaluation of the AI agent was conducted using a comprehensive test set of 50 questions derived from simulated user queries. This approach allowed us to assess the agent's performance across a range of scenarios, including both in-scope and out-of-scope questions.

3.1 Overall Performance

Out of the 50 total questions:
- Correct responses: 48 (96%)
- Incorrect responses: 2 (4%)

3.2 Detailed Performance Breakdown

1) In-scope questions:
- Total in-scope questions: 20
- Correct responses: 18
- Incorrect responses: 2

2) Out-of-scope questions:
- Total out-of-scope questions: 30
- Correctly identified as out-of-scope: 28
- Incorrectly attempted to answer (false positives): 2

3) Error handling:
- Temporary errors encountered: 6 out of 50 queries (12%)
- Resolution: All 6 temporary errors were resolved upon retry, resulting in correct responses

3.3 Standard Machine Learning Metrics

Based on these results, we calculated the following metrics:
1) Accuracy: (48 correct responses) / (50 total questions) = 96%
2) Precision: (18 true positives) / (18 true positives + 2 false positives) = 90%
3) Recall (Sensitivity or True Positive Rate): (18 true positives) / (18 true positives + 2 false negatives) = 90%
4) Specificity (True Negative Rate): (28 true negatives) / (28 true negatives + 2 false positives) = 93.3%
5) F1 Score: 2 * (Precision * Recall) / (Precision + Recall) = 90%
6) Error Rate: (2 incorrect responses) / (50 total questions) = 4%

3.4 Analysis of Results

1) Overall Accuracy: The agent demonstrated high overall accuracy (96%), correctly handling both in-scope questions and identifying out-of-scope queries.
2) In-scope Performance: For questions within its scope, the agent showed strong performance, correctly answering 18 out of 20 questions (90% accuracy).
3) Out-of-scope Handling: The agent excelled in identifying out-of-scope questions, correctly recognizing 28 out of 30 such queries (93.3% accuracy). This ability is crucial for maintaining user trust and preventing misinformation.
4) Balanced Performance: The equal precision and recall (both 90%) indicate a well-balanced system that's equally capable of providing correct answers and avoiding false positives.
5) Error Handling: The agent encountered temporary errors in 12% of queries but demonstrated robust recovery by resolving all these errors upon retry. This suggests effective error handling mechanisms but also highlights an area for potential improvement in initial response stability.
6) Areas for Improvement: The two in-scope questions that were answered incorrectly and the two out-of-scope questions that received attempted answers represent clear targets for future refinement.

3.5 Specific Performance Improvements

The results also reflect improvements made during the development process:
1) Alternative Investments Recognition: Through targeted training, the agent learned to correctly identify and retrieve information about alternative investments, despite the API's use of the shorthand "Alts" for this asset class.
2) Performance Metric Interpretation: The agent was successfully trained to translate qualitative terms into quantitative API parameters, such as associating "high growth" with a minimum return greater than 7%.

4. Discussion

The Vertex AI agent for investment product support demonstrates the potential of AI in enhancing customer service within the financial sector. Its high accuracy rate for direct queries showcases the effectiveness of integrating AI with existing data systems to provide quick and accurate information to customers.

4.1 Strengths

- Fast Development: LLM's context helped in building the agent with a quick turnaround as the intents and parameters need not be manually specified.
- Rapid Response: The agent can quickly retrieve and present information about investment products, potentially faster than human representatives.
- Consistency: By relying on a standardized API, the agent ensures consistency in the information provided to customers.

- Availability: The AI system can handle multiple queries simultaneously, offering 24/7 support.

4.2 Areas for Improvement

1) Asset Class Misinterpretation: When asked about an international bond fund, the agent queried for 'bonds' generically, returning information about a U.S. Treasury bond fund instead.
2) Calculation Limitations: The agent struggled with queries requiring calculations beyond simple data retrieval. For example, when asked to identify the fund with the best returns, it retrieved the necessary data but failed to perform the required comparison, instead returning the return of the first fund in the list.
3) Data Fabrication: Initially, the agent would fabricate information when the API didn't have the requested data. This issue was resolved by adding explicit instructions to base responses solely on available data.

5. Conclusion

The Vertex AI agent demonstrates significant promise in providing accurate and reliable investment product information to customers. Its high overall accuracy and strong performance in identifying out-of-scope queries make it a valuable tool for enhancing customer service in the asset management industry.

The challenges and limitations identified, particularly in handling nuanced queries and performing data analysis, provide clear directions for future development. By addressing these areas for improvement, the agent's capabilities can be expanded to handle a wider range of complex queries. The success of the targeted training approach in resolving initial issues (such as data fabrication) suggests that continued iterative refinement, coupled with an expanded knowledge base and enhanced analytical capabilities, could lead to a highly sophisticated AI support system. Such a system would not only provide accurate information but also enhance the customer experience in navigating complex investment decisions.

References

[1] *Agents*. (n.d.). Google Cloud. https://cloud.google.com/dialogflow/vertex/docs/concept/agents
[2] Buckley, R., Zetzsche, D., Arner, D., & Tang, B. (2021). Regulating Artificial intelligence in Finance: Putting the human in the loop. *SSRN Electronic Journal*. https://papers.ssrn.com/sol3/Delivery.cfm/SSRN_ID3831758_code524849.pdf?abstractid=3831758&mirid=1
[3] M. Bartram, S., Branke, J., & Motahari, M. (2020). *Artification intelligence in asset managment*. CFA Institute Research Foundation. https://www.cfainstitute.org/-/media/documents/book/rf-lit-review/2020/rflr-artificial-intelligence-in-asset-management.ashx