Ignise A., International Journal of Computer Science and Mobile Applications, Vol. 12 Issue 2, February-2024, pg. 41-48.

ISSN: 2321-8363

Impact Factor: 6.308

(An Open Accessible, Fully Refereed and Peer Reviewed Journal)

Tracking Intelligence and Effectiveness of Agents

Ashrey Ignise[1]*; Yashika Vahi[2]

[1]Chief Executive Officer ArtusAI Workspaces Pvt Ltd Boston, USA

[2] Research Scientist ArtusAI Workspaces Pvt Ltd Vancouver, British Columbia, Canada

E-mail: ashrey.ignise@artusai.co

Received date: 16 Feb 2024**,** Manuscript No. IJCSMA-24-141976, **Editor assigned:** 17 Feb 2024**,** Pre QC No. IJCSMA-24-141976 (PQ); **Reviewed:** 20 Feb 2024, QC No. IJCSMA-24-141976 (Q)**; Revised:** 24 Feb 2024**,** Manuscript No. IJCSMA-24-141976 (R); **Published date:** 29 Feb 2024

Abstract

This paper explores the importance of tracking the intelligence and effectiveness of intelligent agents. By examining key metrics such as accuracy, response time, and user satisfaction, and discussing practical methods for evaluation, we provide a comprehensive guide to assessing agent performance.

Keywords: Intelligent agents; Distributed artificial intelligence; Agent applications; Co-operation

1. Introduction

Monitoring and assessing intelligent agents' performance has become vital in an environment where they are integral to many different industries. Maximizing these agents' functionality and effectiveness requires an in-depth knowledge of how well they operate, adapt, and meet the requirements of users. The objective of this paper is to clarify the value of tracking agent performance, walk through important metrics and approaches, and provide real-world examples and case studies of successful methods for assessment.

2. Performance Metrics

2.1. Accuracy

The degree to which an agent's choices or outputs match the intended or precise results is commonly referred to as accuracy [1].

©2024, IJCSMA All Rights Reserved, www.ijcsma.com

This work is licensed under a Creative Commons Attribution 4.0 International License.

2.1.1. Measuring Accuracy:

- **Complex Grid:** Typically used in tasks concerning classification, a complex grid, also referred to as the "confusion matrix" compares true positives, true negatives, false positives, and false negatives in order to evaluate performance.
- **Precision and Recall:** Recall measures the percentage of accurately detected real positives, whereas precision calculates the percentage of accurate positive predictions.
- **Accuracy Rate:** The proportion of precise projections to all of the agent's recommendations.

Example: By comparing the agent's prediction with real-world patient outcomes, one can determine the accuracy of a healthcare diagnostic system.

2.2. Response Time

The amount of time an agent takes to react to an assignment or event is known as its response time. It is an importantindicator, particularly for real-time applications where fast reactions are crucial.

2.2.1. Tracking Response Time Measurement:

- **Latency Measurement:** Determining the precise duration of time that passes between receiving a command and producing an output.
- **Benchmarking:** Analyzing the agent's reaction time in comparison with similar systems or standard practices inthe industry.
- **User Perception:** Getting feedback from users regarding the agent's level of response.

Example: The time taken between a user's query and the agent's response can be utilized to assess response timein customer support chabots.

2.3. Efficiency

Efficiency is the capability of an agent to carry out its assigned duties with the least amount of time, processing power, and data. High efficiency implies the agent can achieve its objectives without consuming unnecessary energy [2].

2.3.1. Assessing and Assessing Effectiveness:

- **Resource Utilization:** Monitoring the amount of memory and CPU that the agent uses to carry out activities.
- **Task Completion Time:** The entire amount of time needed to finish a group of tasks or procedures.
- **Throughput:** The quantity of work the agent can do in a specific amount of time.
- **For Instance:** Consider the engine of an automobile. An efficient engine generates the most power

This work is licensed under a Creative Commons Attribution 4.0 International License.

with the least amount of fuel utilized. In the same manner, an effective agent uses as few assets as possible while achieving maximum performance.

2.4. Benchmark Testing

Agent performance can be evaluated using benchmark testing, which offers an objective measure. It is useful for comparing different agents and making sure they fulfill their performance targets or the standards of the industry.

2.4.1. Benchmark Test Examples:

- **Turing Test:** It ranks an agent's potential to display intelligent behavior that is identical to human behavior.
- **Standardized Datasets:** To evaluate and contrast agent performance, employ datasets like glue for natural language processing or ImageNet for image recognition applications.
- **Performance Benchmarks:** These are specific tests designed for judging an agent's resilience, accuracy, and efficiency in regulated circumstances.

As a demonstration, let's use the example of an athlete competing in a timed race. Just as the athlete's success is measured by the race time, benchmark tests offer a standard metric to examine agent performance.

2.5. Learning Rate

The rate at which an agent acquires expertise or information that boosts its performance over time is referred to as its learning rate. For agents that are meant to respond to fresh data and environments, this is a vital component that ensures their effectiveness in the face of unpredictable situations.

2.5.1. Determining Learning Rate:

- **Performance Enhancement Over Time (PEOT**): Monitoring the agent's performance metrics (accuracy, productivity, etc.) during several learning cycles or repetitions.
- **Convergence Speed:** Represents the speed at which the agent reaches a uniform accuracy or performance level.
- **Error Reduction Rate:** Observing how quickly errors disintegrate when the agent gathers up new information.

Example: Consider a student who is soaking up fresh content of a particular subject. A quick learner picks information up swiftly and eventually starts committing less errors in that area, much as an agent with a high learning rate that improves its performance and productivity each time it gets new information.

2.6. Scalability

Scalability is the power of an agent to continue operating at its maximum potential even when the volume of data or

©2024, IJCSMA All Rights Reserved, www.ijcsma.com

This work is licensed under a Creative Commons Attribution 4.0 International License.

task complexity grows. It is critical to make certain that the agent can manage growing requirements without experiencing an evident reduction in performance.

2.6.1. Evaluating Scalability Measures:

- **Load Testing:** Analyzing how well the agent handles growing volumes of data or jobs through assessing its performance under different workloads.
- **Performance Metrics at Scale:** Monitoring key performance indicators (accuracy, reaction time, etc.) as the agent tackles more challenging tasks or larger databases.
- **Elastic Resource Utilization**: Tracking the agent's optimal use of extra computational capacity when it appears accessible.

Example: Scalability in a financial trading system may be evaluated by looking at how the system operates at peak trading times when there are a lot of transactions. A scalable system has high accuracy and low latency even when the workload of each agent is increased.

2.7. Robustness

The ability of an agent to function consistently in a range of unpredictable circumstances, such as the emergence of flaws, noise, or system breakdowns, is known as robustness [3].

2.7.1. Calculating Robustness Assessment:

- **Stress Testing:** It is a way of putting an agent under harsh circumstances or unusual inputs to observe its stability and performance.
- **Error Handling Capability:** Examining the agent's capacity to bounce back from mistakes or unanticipated events.
- **Adversarial Testing:** This involves deliberately introducing clashing inputs to assess how resilient the agent is to manipulations or cyberattacks.

Example: To make sure the car can travel safely and successfully, autonomous vehicles' robustness is assessed by subjecting the system to many different kinds of driving instances, such as severe weather, unpredictable obstacles, and system malfunctions.

3. User Satisfaction Metrics

3.1. User Feedback

Determining how well an agent serves the necessities and expectations of its consumers depends heavily on their feedback, which brings straightforward insights into customer experiences, highlighting both positive and negative

©2024, IJCSMA All Rights Reserved, www.ijcsma.com

This work is licensed under a Creative Commons Attribution 4.0 International License.

aspects [4].

3.1.1. Steps for Gathering and Examining Feedback:

- **Questions and Surveys:** Systematic tools that request feedback on experiences and ratings of satisfaction from users.
- **User interviews:** Extensive talks that dive into user experiences and obtain in-depth knowledge.
- **Feedback Forms:** Easy-to-use forms included in applications that let users quickly submit ratings and comments.
- **Sentiment Analysis:** Using Natural Language Processing (NLP) to analyze text feedback, discover the generalsentiment (positive, negative, or neutral).

Example: Post-interaction surveys can be used by a customer care Chabot to get feedback on user satisfaction. This input can then be examined to further improve the Chabot's responses and functioning.

3.2. Engagement Levels

Activity among users and frequency variation are measured by engagement levels with an agent.

3.2.1. Evaluating Involvement:

- **Usage Metrics:** Analyzing the quantity of contacts, duration of sessions as well as the amount of usage.
- **Click-through Rates (CTR):** Determining how frequently users interact with particular agent alerts or recommendations.
- **Active User Metrics:** To assess consistent involvement, count the number of daily, weekly, or monthly active users.

Example: High levels of engagement in a learning management system are declared by the number of lessons finished, time spent on the website, and frequency of logins, all of which suggest that students find the program interesting and helpful.

3.3. Task Completion Rates

The proportion of tasks that users successfully complete with the assistance of an agent is measured by task completion rates.

3.3.1. Evaluating Task Fulfillment:

- **Completion Tracking:** Keeping track of how many assignments have been initiated against how many is finished successfully.
- **Success Rates:** Calculating the ratio of tasks completed properly to the total number of attempts.

©2024, IJCSMA All Rights Reserved, www.ijcsma.com

This work is licensed under a Creative Commons Attribution 4.0 International License.

- **Drop-off Analysis:** Identifying the moments at which users give up on a job in order to identify and resolve barriers to finishing it.

Example: Task completion rates for matters like paying bills, transferring money, and applying for loans can be observed in a banking app. This shows how well an agent supports users in completing these activities.

4. Metrics in Practical Applications

4.1. Healthcare Diagnostic Systems

Intelligent agent-powered diagnostic systems are essential for helping medical professionals diagnose patients in the healthcare industry. Large volumes of scientific data are leveraged by these systems, along with advanced algorithms, to deliver quick and precise diagnoses [5].

4.1.1. Metrics in Python Programming:

- **Accuracy Rate:** Correct diagnoses the system made as opposed to those made by human experts.

  Accuracy Rate = (Number of correct diagnoses / Total number of diagnoses) × 100

- **Average Response Time:** The average time that the system takes to make a diagnosis of a patient by examining their data.

  Average Response Time = Sum of response times / Number of diagnoses

- **User Satisfaction Score:** Reviews from medical experts about the system's quality and stability.

  User Satisfaction Score = Sum of satisfaction ratings / Number of responses

4.2. Finance Automated Trading Systems

In the financial sector, automated trading systems are crucial because they perform trades at volumes and speeds that human traders have no way to match. Based on established algorithms and current conditions in the market, these computers evaluate market data and make investment judgments [6].

4.2.1. Metrics in Python Programming:

- **Trade Success Rate:** Profitable transactions of every single trade the system has made.

  Trade Success Rate = (Number of profitable trades / Total number of trades) × 100

- **Trade Execution Latency:** Time between the execution of trades and changes in market statistics. For high- frequency traders to take advantage of temporary market opportunities, low latency is a must.

  Trade Execution Latency = Sum of execution times / Number of trades

This work is licensed under a Creative Commons Attribution 4.0 International License.

- **Resource Utilization Efficiency:** Ability of the system to manage enormous amounts of trading while controlling the amount of resources being used. During instances of high market activity, metrics like CPU and memory utilization are observed.

  Resource Utilization Efficiency = Total resources used / Number of trades

Example: A banking organization might monitor the success rate of deals that transpire and compare it to industry standards to figure out how accurate their trading system is. During trading sessions, they can make sure the system's reaction time remains within acceptable boundaries by using cutting-edge monitoring tools.

4.3. Manufacturing Predictive Maintenance Systems

Automated agents are used by forecasting systems in production facilities in order to track the condition of machines and take care of malfunctions before they happen. By evaluating sensor data from equipment, these systems identify wear and tear indications and allow proactive planning of maintenance [7].

4.3.1. Metrics in Python Programming:

- **Prediction Accuracy:** Failure predictions in proportion to actual maintenance necessities. Increased precision by agents reduces unnecessary servicing and prevents unexpected malfunctions.

  Prediction Accuracy = (No. of correct predictions / Total number of predictions) × 100

- **Unplanned Downtime Reduction:** Metrics include Overall Equipment Effectiveness (OEE) in the decrease of spontaneous interruptions in processing and repair expenses.

  Unplanned Downtime Reduction = ((Previous downtime - Current downtime) / Previous downtime) × 100

- **Failure Notification Downtime:** The rate at which the system can detect errors and notify service technicians to them.

  Failure Notification Response Time = Sum of notification times / Number of notifications

Example: By comparing expected maintenance needs with observed outcomes, a manufacturing facility can determine how precise its maintenance forecasting system is at avoiding accidental malfunctions.

5. Importance of Continuous Tracking and Evaluation

Upholding and improving intelligent agent's effectiveness requires constant tracking and evaluation. Companies may make sure that their agents remain to be productive, efficient, and in accordance with user needs through the upholding and improving intelligent agents' effectiveness requires constant tracking and evaluation. Companies may make sure that their agents remain to be productive, efficient, and in accordance with user needs through the use of live tracking technologies, qualitative and quantitative approaches, and establishing trends. Research and development must proceed in order to improve inspection metrics, handle new problems, and explore fresh

©2024, IJCSMA All Rights Reserved, www.ijcsma.com

This work is licensed under a Creative Commons Attribution 4.0 International License.

applications for intelligent agents. Cooperation amongst government agencies, businesses, and academics will be essential to the development of the field and to ensuring intelligent agents' positive impacts across a range of industries [8].

References

[1] Kulkarni, Umakanth P., & Anil R. Yardi. "Agent tracking: A reliable agent forwarding mechanism." *Int Conf Comput Intell Model Control Autom Int Conf Intell Agents Web Technol Internet Commer*.1(2005).

[2] Barr, M., & J. Baker. "Archer: An intelligent agent for business monitoring." (1995): 1-5.

[3] Lyu, Michael R., et al. "Design and evaluation of a fault-tolerant mobile-agent system." *IEEE Intell Syst.* 19.5 (2004): 32-38.

[4] Urlings, Pierre, & Lakhmi C. Jain. "Teaming human and machine: A conceptual framework." *Hybrid Inf Syst.* (2002) 711-721.

[5] Wooldridge, Michael, & Nicholas R. Jennings. "Intelligent agents: Theory and practice." *Knowl Eng Rev.* 10.2 (1995): 115-152.

[6] Camacho, David, et al. "Performance Evaluation of Zeus, Jade, And Skeletonagent Frameworks." *IEEE Int Conf Syst Man Cybern*. 4. (2002).

[7] Mawlood Yunis, A., et al. "Comparing performance of two mobile agent platforms in distributed search." *Proc, IEEE/WIC/ACM Int Conf Intell Agent Technol. 2004.* (2004).

[8] Samaras, George, et al. "Mobile agent platforms for Web databases: a qualitative and quantitative assessment." *Proc, First Third Int Symp Agent Syst Appl Mob Agents.* (1999).

©2024, IJCSMA All Rights Reserved, www.ijcsma.com

This work is licensed under a Creative Commons Attribution 4.0 International License.