Journal of the Brazilian Computer Society, 2025, XX:1, doi: 10.5753/jbcs.2025.XXXXXX

Desenvolvimento de Agente Autônomo para Análise Exploratória de Dados

Lucas Zegrine Duarte [***lucasgnuzinho@gmail.com***]

Received: DD Month YYYY • **Accepted:** DD Month YYYY • **Published:** DD Month YYYY

Abstract. A Análise Exploratória de Dados (EDA) é uma etapa fundamental no ciclo de vida de projetos de ciência de dados, porém, muitas vezes é um processo manual e iterativo. Este trabalho apresenta o desenvolvimento de um agente autônomo de Inteligência Artificial projetado para automatizar e otimizar a EDA em conjuntos de dados tabulares. A solução emprega uma arquitetura robusta baseada em LangGraph para orquestrar um fluxo de trabalho determinístico, que consiste em planejamento, geração, execução e avaliação de código Python. O agente é capaz de interagir com o usuário em linguagem natural, interpretar solicitações, gerar visualizações gráficas e extrair conclusões a partir dos dados. Adicionalmente, o sistema incorpora um mecanismo de Geração Aumentada por Recuperação (RAG) para enriquecer as análises com conhecimento contextual proveniente de documentos externos. A arquitetura modular, que inclui uma fábrica de LLMs e um executor de código seguro, garante flexibilidade e confiabilidade. O agente foi validado utilizando um dataset público de fraudes de cartão de crédito, demonstrando sua eficácia em responder a uma variedade de questões analíticas e em gerar insights acionáveis.

Keywords: Agentes de IA, LangGraph, Análise Exploratória de Dados, Geração Aumentada por Recuperação, Processamento de Linguagem Natural, Automação de Análise de Dados.

1 Introdução

A Análise Exploratória de Dados (EDA) é um pilar essencial em qualquer projeto de ciência de dados, alinhada a metodologias consolidadas como o CRISP-DM (CRoss Industry Standard Process for Data Mining) IBM [2025]. Esta fase inicial de investigação permite que os analistas compreendam a estrutura, identifiquem anomalias, testem hipóteses e descubram padrões nos dados. Tradicionalmente, a EDA é um processo intensivo em trabalho, que exige a escrita manual de scripts para gerar estatísticas descritivas e visualizações.

Com o avanço dos Grandes Modelos de Linguagem (LLMs), surge a oportunidade de transformar este processo. A capacidade dos LLMs de compreender a linguagem natural e gerar código abre caminho para a criação de ferramentas interativas que automatizam tarefas de análise. O objetivo deste trabalho, conforme proposto na atividade acadêmica, é desenvolver um agente de IA genérico, capaz de realizar uma EDA completa em qualquer arquivo CSV fornecido pelo usuário. O agente deve ser capaz de responder a perguntas, gerar representações gráficas e, crucialmente, formular conclusões a partir das análises realizadas.

Para validar a solução, foi utilizado o conjunto de dados sobre fraudes de cartão de crédito disponibilizado no KaggleMachine Learning Group - ULB [2018], o dataset contém anonimização de suas features via PCA e alto desbalanceamento de classes. Este artigo detalha a arquitetura da solução, os frameworks escolhidos e demonstra a sua funcionalidade através de exemplos práticos de interação.

2 Estrutura da Solução e Frameworks

Para atender aos requisitos de robustez, flexibilidade e segurança, a solução foi arquitetada de forma modular, utilizando frameworks de ponta no ecossistema de IA.

2.1 Frameworks e Tecnologias Escolhidas

A base do projeto foi construída sobre as seguintes tecnologias:

- **LangChain e LangGraph:** O LangChain Chase [2022] foi utilizado para a integração com LLMs e a criação de componentes, enquanto o LangGraph foi escolhido como o orquestrador central do agente. Diferente de uma abordagem ReAct (Reasoning and Acting) tradicional, o LangGraph permite a definição de um fluxo de trabalho como um grafo de estados, garantindo maior controle, previsibilidade e depuração do processo de análise.
- **Streamlit:** Para a interface com o usuário, o Streamlit Mostafa *et al*. [2019] foi escolhido por sua simplicidade e rapidez no desenvolvimento de aplicações web interativas, permitindo o upload de arquivos e a criação de uma interface de chat intuitiva.
- **Pandas, Matplotlib e Seaborn:** constituem o núcleo das ferramentas de análise e visualização de dados que o agente utiliza para executar as solicitações do usuário The pandas development team [2025]; Hunter [2007]; Waskom [2024].
- **LLM Providers:** A solução foi projetada para ser agnóstica em relação ao provedor de LLM, suportando modelos como a família Gemini Gemini Team, Google [2023] do Google, GPT da OpenAI OpenAI [2023] e modelos locais via Ollama Bussing and et al. [2023].

2.2 Arquitetura Baseada em Grafo

O núcleo do agente é um grafo de estados determinístico definido com LangGraph, que segue um ciclo de vida claro

para cada consulta do usuário:

1. **Nó de Planejamento (Planner):** Recebe a pergunta do usuário e o histórico da conversa, e gera um plano passo a passo para responder à solicitação.
2. **Nó de Geração de Código (Code Generator):** Converte o plano textual em um script Python executável, utilizando as bibliotecas de análise de dados. Este nó é instruído a seguir regras estritas, como salvar gráficos em formato base64 em vez de exibi-los.
3. **Nó de Execução de Código (Code Executor):** Executa o script Python gerado em um ambiente controlado e seguro. Antes da execução, o código passa por uma etapa de sanitização para bloquear a importação de bibliotecas perigosas (e.g., `os`, `subprocess`), mitigando riscos de segurança.
4. **Nó de Conclusão (Concluder):** Recebe o resultado bruto da execução do código, a pergunta original e o histórico da conversa para formular uma resposta final em linguagem natural para o usuário.

Este fluxo estruturado aumenta a confiabilidade do agente, minimizando alucinações e garantindo que as respostas sejam fundamentadas nos dados.

2.3 Componentes Adicionais

Dois outros componentes são vitais para a funcionalidade do agente:

- **Análise Contextual com RAG:** O agente pode ter seu conhecimento aumentado através da Geração Aumentada por Recuperação (RAG) Lewis *et al*. [2020]. O usuário pode fazer upload de documentos PDF (como dicionários de dados ou relatórios) que são vetorizados e armazenados em uma base de conhecimento. O agente pode consultar esta base para obter contexto adicional e fornecer respostas mais informadas.
- **LLM Factory:** Para garantir a flexibilidade na escolha do modelo de linguagem, foi implementado o padrão de projeto *Factory*. Uma classe `LLMFactory` centraliza a criação de instâncias de LLMs, permitindo a fácil troca entre provedores sem alterar a lógica principal da aplicação.

3 Resultados e Demonstração

Para validar a funcionalidade do agente, foram realizadas uma série de perguntas sobre o dataset de fraudes de cartão de crédito. As interações a seguir demonstram a capacidade do agente em realizar análises descritivas, gráficas, de correlação, detecção de anomalias e identificação de padrões, além de sua habilidade em manter o contexto da conversa para responder a perguntas de acompanhamento e formular conclusões.

3.1 Análise Descritiva e Gráfica

A primeira etapa da análise consistiu em obter uma visão geral dos dados.

- **Usuário:** *Quais são os tipos de dados (numéricos, categóricos)?*
- **Resposta do Agente:** Com base na análise executada, os dados do seu conjunto foram classificados da seguinte forma:
  - **Colunas Numéricas:** `Time`, `Amount` e as 28 variáveis anonimizadas (`V1` a `V28`). Estas colunas representam quantidades mensuráveis.
  - **Coluna Categórica:** `Class`. Esta é uma variável binária que representa as duas categorias distintas do alvo de análise (transação normal vs. fraude).

Figure 1. Resposta do agente classificando os tipos de dados das colunas.

- **Usuário:** *A partir dessas colunas o que pode ser afirmado sobre a base?*
- **Resposta do Agente:** Com base na análise realizada, as seguintes afirmações podem ser feitas sobre o conjunto de dados:
  - **Natureza dos Dados:** Trata-se de um conjunto de dados de detecção de fraude em transações financeiras.
  - **Principal Característica:** O ponto mais crítico é o extremo desbalanceamento de classes. As transações fraudulentas (Classe 1) representam apenas 0.172% do total, tornando este um problema de detecção de eventos raros.
  - **Conclusão Principal:** O principal desafio ao trabalhar com esta base será desenvolver um modelo preditivo que consiga identificar com precisão as poucas transações fraudulentas em meio a um volume massivo de transações normais.

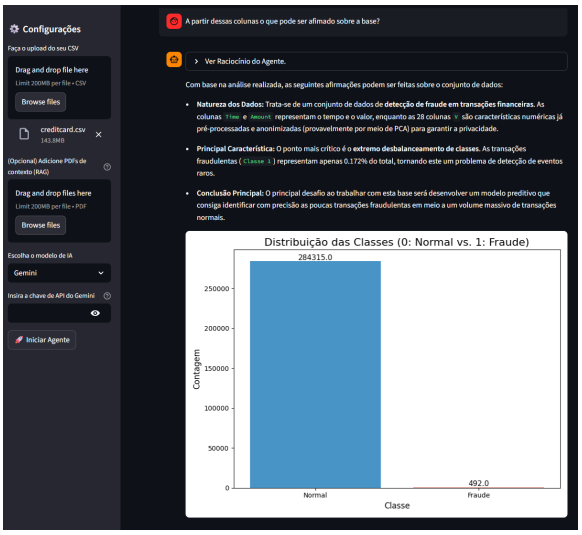

Figure 2. Gráfico e conclusões do agente ilustrando o desbalanceamento da variável 'Class'.

3.2 Análise de Relação entre Variáveis

Investigou-se a correlação entre as variáveis não anonimizadas.

- **Usuário:** *Existe alguma correlação entre 'Amount' e 'Time'?*
- **Resposta do Agente:** Não, não existe uma correlação significativa entre o valor da transação (Amount) e o tempo em que ela ocorreu (Time). A análise numérica mostra um coeficiente de correlação de Pearson de -0.01, um valor extremamente próximo de zero. A análise visual do gráfico de dispersão confirma essa conclusão.

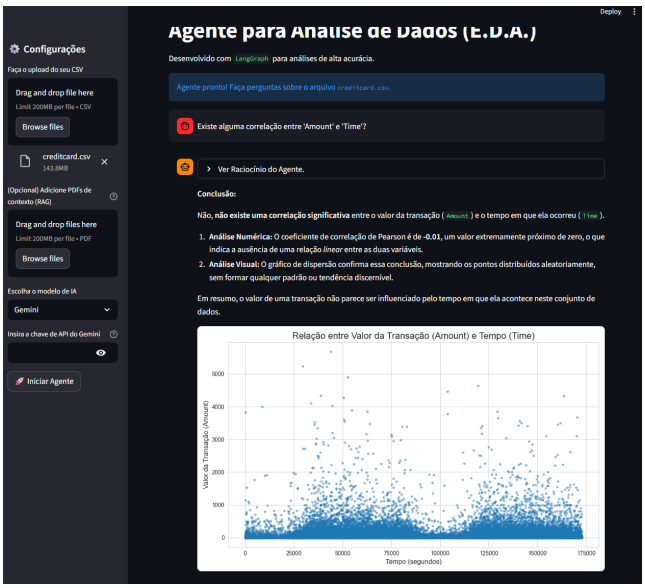

Figure 3. Gráfico de dispersão gerado para visualizar a relação entre 'Amount' e 'Time'.

3.3 Detecção de Outliers e Demonstração de Memória

Esta análise foi realizada em duas etapas para demonstrar a capacidade do agente de manter o contexto da conversa.

- **Usuário (1ª Interação):** *Existem valores atípicos nos dados?*
- **Resposta do Agente (1ª Interação):** Sim, a análise confirma a presença significativa de valores atípicos (outliers) nos dados. Utilizando o método do Intervalo Interquartil (IQR), foi identificado que quase todas as colunas numéricas, especialmente as variáveis `V27` (13.75%), `Amount` (11.20%) e `V28` (10.65%) são as que apresentam as maiores porcentagens de valores atípicos.

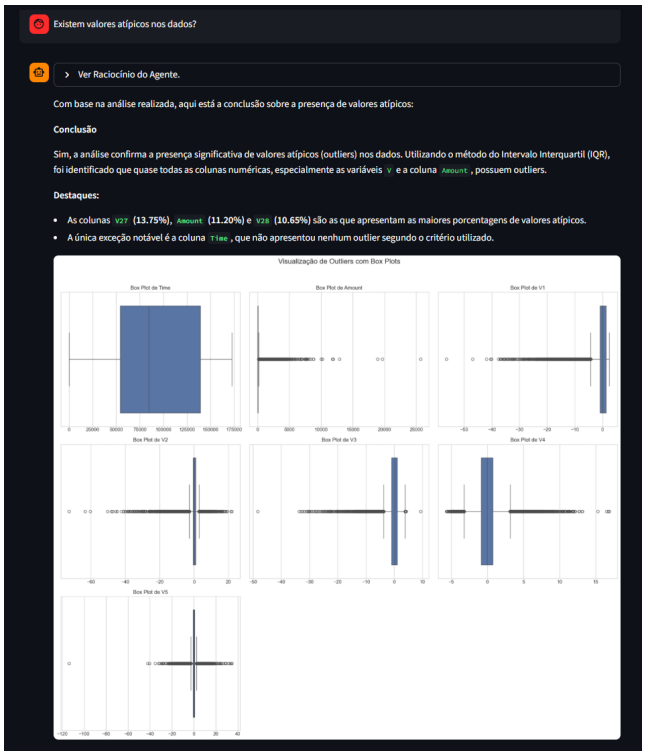

Figure 4. Box plots gerados para a detecção visual de outliers em diversas colunas.

- **Usuário (2ª Interação):** *Como esses outliers afetam a análise?*
- **Resposta do Agente (2ª Interação):** Os outliers impactam significativamente a análise. Seus destaques são:
  - **Distorcem as Medidas Estatísticas:** A média do valor das transações (`Amount`) é de 88.35, enquanto a mediana é de apenas 22.00, o que demonstra que os outliers inflam artificialmente a média.
  - **Ocultam a Distribuição Real dos Dados:** A escala do histograma da coluna `Amount` é "esticada" para abranger os outliers, dificultando a visualização da grande maioria das transações de baixo valor.
  - **Indicam Padrões Relevantes para Fraude:** O valor médio das transações fraudulentas (Classe 1) é consideravelmente maior que o das transações normais, sugerindo que valores extremos podem ser um forte indicativo de fraude.

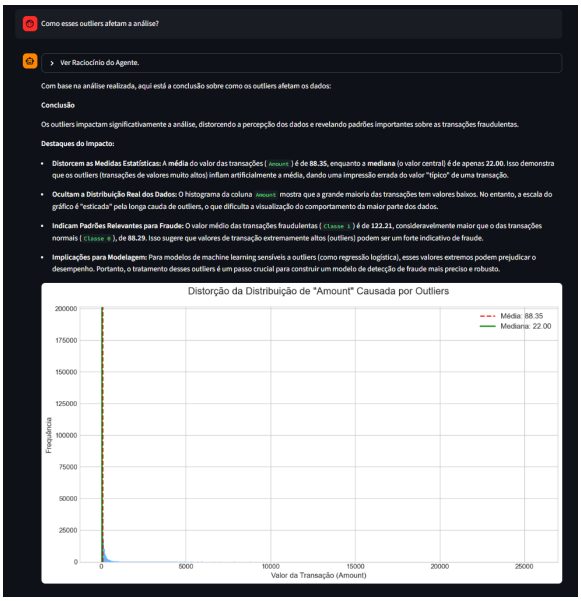

Figure 5. Histograma da coluna 'Amount' mostrando a distorção na distribuição causada pelos outliers.

Figure 6. Comparação visual entre os clusters encontrados pelo K-Means (esquerda) e as classes originais do dataset (direita) após redução de dimensionalidade com t-SNE.

3.4 Análise de Padrões e Agrupamentos (Clustering)

Por fim, o agente foi testado quanto à sua capacidade de identificar agrupamentos intrínsecos nos dados e descrever as tendências observadas nesses grupos.

- **Usuário:** *Existem agrupamentos (clusters) nos dados?*
- **Resposta do Agente:** Sim, a análise confirma a existência de agrupamentos (clusters) claros e bem definidos nos dados. Para investigar isso, os dados foram visualizados em 2D com a técnica t-SNE e agrupados com o algoritmo K-Means Scikit-learn developers [2025], buscando por 2 clusters. O algoritmo K-Means identificou com sucesso dois grupos distintos: um cluster grande e denso, e outro cluster muito menor e mais disperso. Ao comparar esses agrupamentos com as classes originais das transações (normais vs. fraudulentas), a correspondência visual é quase perfeita, indicando que os agrupamentos encontrados correspondem de forma eficaz às transações normais e fraudulentas.
- **Usuário:** *Destes agrupamentos quais tendências podem ser observadas?*
- **Resposta do Agente:** Com base na análise, as principais tendências observadas nos agrupamentos (transações normais vs. fraudulentas) são:

1. **Valor da Transação (Amount):** Transações fraudulentas (Classe 1) tendem a ter um valor médio superior (122,21) ao das transações normais (88,29).
2. **Padrão de Tempo (Time):** As transações fraudulentas não seguem o mesmo padrão cíclico das transações normais. Enquanto as transações normais apresentam picos e vales de atividade, as fraudes ocorrem de forma mais distribuída ao longo do tempo.
3. **Características Ocultas (V1-V28):** As médias das colunas V1 a V28 são notavelmente diferentes entre os dois grupos, reforçando que os dois grupos são estruturalmente distintos.

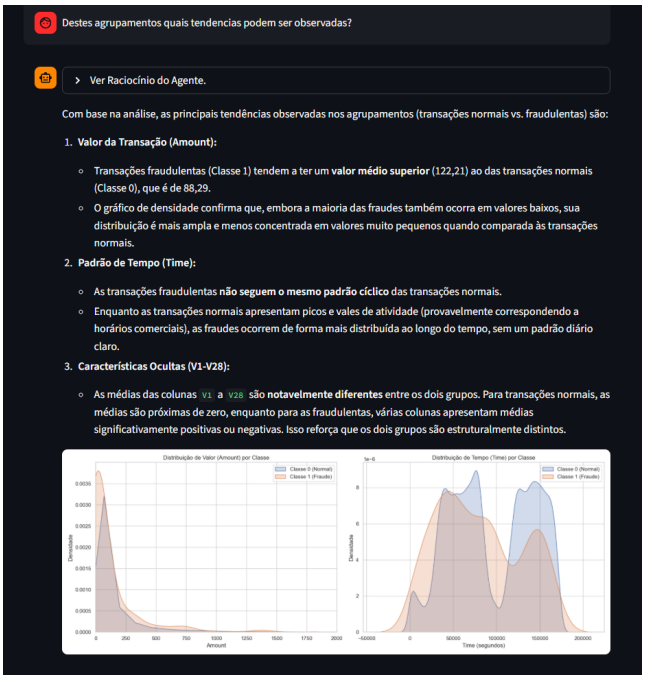

Figure 7. Gráficos de densidade (KDE) comparando as distribuições de 'Amount' e 'Time' entre as classes de transações normais e fraudulentas.

4 Conclusão

Este trabalho demonstrou com sucesso a construção de um agente autônomo para Análise Exploratória de Dados, cumprindo todos os objetivos propostos. A arquitetura baseada em LangGraph provou ser uma escolha acertada, conferindo robustez e controle ao fluxo de análise, superando as limitações de agentes ReAct mais simples. A capacidade de gerar gráficos, interagir em linguagem natural e manter o contexto conversacional para formular conclusões complexas foi validada.

A integração de funcionalidades como o RAG e a flexibilidade para múltiplos provedores de LLM tornam a ferramenta poderosa e adaptável a diferentes cenários de análise. A ênfase na segurança, através da sanitização do código gerado, é um diferencial importante para a confiabilidade da aplicação.

Como trabalhos futuros, o roteiro de IA Responsável delineado no projeto original (descrito no `readme.md`, dísponível no github: https://github.com/LucasZeD/autonomous-agent-for-exploratory-data-analysis) oferece um caminho para evolução natural do projeto.

Declarações

Agradecimentos

Agradeço à instituição I2A2 (Institut d'Intelligence Artificielle Appliquée) pela proposição do desafio e pelos materiais de estudo fornecidos, que foram fundamentais para o desenvolvimento deste projeto.

Financiamento

Este trabalho não recebeu financiamento externo.

Contribuições

A contribuição do autor, Lucas Zegrine Duarte, para este trabalho abrange as seguintes áreas, conforme a Taxonomia CRediT Brand, Allen and Allen, Liz and Altman, Micah and Hlava, Marjorie [2015]:

- **Conceituação:** Formulação da ideia e dos objetivos do agente de EDA.
- **Metodologia:** Desenho da arquitetura baseada em LangGraph e definição dos componentes.
- **Software:** Desenvolvimento e implementação de todo o código da aplicação.
- **Validação:** Teste e validação das funcionalidades do agente.
- **Investigação:** Execução da pesquisa e análise sobre o dataset de exemplo.
- **Redação – Original:** Escrita deste relatório.
- **Administração do projeto:** Gerenciamento de todas as etapas do desenvolvimento.

Interesses

O autor declara não haver conflitos de interesse.

Materiais

Os códigos-fonte gerados durante este estudo estão anexados na submissão do trabalho e disponibilizados publicamente via github `https://github.com/LucasZeD/autonomous-agent-for-exploratory-data-analysis.git`, conforme solicitado. O conjunto de dados utilizado é de acesso público e pode ser encontrado em: `https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud`.

References

Brand, Allen and Allen, Liz and Altman, Micah and Hlava, Marjorie (2015). CRediT (Contributor Roles Taxonomy). `https://credit.niso.org/`. Standard.

Bussing, J. and et al. (2023). Ollama. `https://github.com/ollama/ollama`. Software Framework.

Chase, H. (2022). LangChain. `https://github.com/langchain-ai/langchain`. Software Framework.

Gemini Team, Google (2023). Gemini: A Family of Highly Capable Multimodal Models. *arXiv preprint arXiv:2312.11805*. DOI: 10.48550/arXiv.2312.11805.

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3):90–95. DOI: 10.1109/MCSE.2007.55.

IBM (2025). CRISP-DM Help Overview. `https://www.ibm.com/docs/en/spss-modeler/saas?topic=dm-crisp-help-overview`. Acessado em 09/10/2025.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems 33*, pages 9459–9474.

Machine Learning Group - ULB (2018). Credit Card Fraud Detection. `https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud`. Acessado em 09/10/2025.

Mostafa, A., Teixeira, T., and Treuille, A. (2019). Streamlit: The fastest way to build and share data apps. `https://streamlit.io`.

OpenAI (2023). GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*. DOI: 10.48550/arXiv.2303.08774.

Scikit-learn developers (2025). 2.3. Clustering - K-means. `https://scikit-learn.org/stable/modules/clustering.html#k-means`. Acessado em 09/10/2025.

The pandas development team (2025). pandas: powerful python data analysis toolkit. `https://pandas.pydata.org`. Acessado em 09/10/2025.

Waskom, M. L. (2024). seaborn: statistical data visualization. `https://github.com/mwaskom/seaborn`. Acessado em 09/10/2025.