

Relatório Técnico de Desenvolvimento do Projeto IA-TRADE

Responsáveis técnicos:

- Anna Carolina Figuccio Alves
- Marcus Pablo Rezende Teixeira Manso

1. Introdução

O projeto **IA-TRADE** nasceu da necessidade de desenvolver uma plataforma de trading inteligente, capaz de unir análise técnica, aprendizado de máquina, backtesting robusto, simulação em tempo real e integração com fontes externas de sentimento de mercado. Desde o início, estabelecemos como objetivo criar uma solução que fosse **modular, interpretável, escalável e amigável para o usuário**.

Este relatório descreve as etapas de desenvolvimento, os principais desafios enfrentados, as decisões de arquitetura do sistema e os resultados obtidos.

2. Etapas de Desenvolvimento

Fase 1 – Estruturação Inicial e Organização

Objetivo: Criar a base sólida para desenvolvimento contínuo.

- Um dos primeiros desafios foi organizar a grande quantidade de scripts, dados e notebooks experimentais gerados nas etapas preliminares.
- Estruturei o projeto em uma arquitetura clara e modular:
 - `processed_data/`: dados históricos, de treino e validação.
 - `models/`: repositório centralizado dos modelos treinados.
 - `IA-TRADE/`: scripts principais e utilitários.
- Isso não apenas facilitou a manutenção e navegação, como também preparou o terreno para integração contínua.

Fase 2 – Padronização de Caminhos e Leitura de Arquivos

Problema identificado: Vários erros de `FileNotFoundError` surgiram por inconsistência nos caminhos relativos e absolutos, especialmente ao migrar o projeto entre ambientes.

Solução aplicada:

- Refatorei todos os scripts para utilizar caminhos relativos com base em `os.path` e `Pathlib`.
 - Testei o funcionamento em diferentes máquinas e ambientes virtuais (Windows, Linux, Google Colab) garantindo portabilidade.
-

3. Desenvolvimento de Funcionalidades

3.1 Interface Web com Streamlit

Decisão de design:

Escolhi o Streamlit pela facilidade de integração com Python e sua capacidade de criar dashboards interativos rapidamente. A arquitetura com abas permite uma divisão clara das funcionalidades.

Funcionalidades implementadas:

- Gráficos de preço com destaque visual da área de predição.
- Exibição de indicadores técnicos (candles, RSI, volume).
- Abas temáticas: Treinamento, Backtest, Simulação em Tempo Real, Performance, Settings e mais.

3.2 Módulo de Treinamento de Modelos

Desafios:

- Evitar *lookahead bias* e vazamento de dados.
- Garantir consistência nos splits temporais.

Soluções:

- Estabeleci como data limite para dados de treino 31/12/2024.
- Automatizei o salvamento de modelos (`.pkl`) e *scalers*.

Modelos utilizados:

- Random Forest
- XGBoost
- Ensemble Customizado

Cada modelo foi ajustado para prever sinais discretos: **compra, venda ou hold**.

4. Backtesting Robusto

Desafio:

Simular operações realistas sem comprometer a velocidade de execução e mantendo o sistema interpretável.

Funcionalidades implementadas:

- Modos agressivo e conservador.
 - Simulação com aplicação de **slippage**, **taxas**, **stop loss** e **take profit**.
 - Cálculo de métricas completas: **win rate**, **profit factor**, **Sharpe**, **drawdown**, **lucro total**, entre outros.
 - Exportação de relatórios em CSV para análise externa.
-

5. Bot de Trading Automatizado

Lógica desenvolvida:

- Tomada de decisão baseada em modelo treinado.
- Parâmetros ajustáveis via interface: stop, take, confiança, modo de operação.
- Explicações automáticas de cada decisão com base na integração com OpenAI (modelo GPT).

Desafio técnico:

Evitar importação circular nas funções de explicação.

Solução:

- Centralizei a função de explicação em um único módulo, importado dinamicamente quando necessário.
-

6. Análise de Previsões Externas

Implementei suporte para que o usuário possa:

- Fazer upload de CSV com sinais externos (outros modelos ou especialistas).
 - Visualizar acurácia, lucro, win rate por tipo de ação.
 - Comparar graficamente com estratégia **buy & hold**.
-

7. Sentimento de Mercado

Objetivo: Incluir perspectiva qualitativa e de contexto no sistema.

Implementações:

- Integração com **CryptoPanic API**.
 - Classificação do sentimento de manchetes com uso de modelo OpenAI.
 - Resultados exibidos em tempo real no painel.
-

8. Correções e Aprimoramentos Contínuos

Durante a consolidação do projeto, os seguintes problemas e melhorias foram resolvidos:

- Corrigidos erros de *unpacking* em funções com múltiplos retornos.
 - Tratamento para `DataFrames` vazios no Streamlit, evitando falhas.
 - Garantia de colunas mínimas mesmo sem dados (zero operações).
 - Criação de mensagens amigáveis para casos sem resultados.
 - Ajustes automáticos de parâmetros com base no modo selecionado.
 - Documentação com **README** claro, explicando desde o setup local até o uso das principais funções.
-

9. Automação e Suporte ao Usuário

Automação via Terminal:

- Scripts automatizados para movimentação e renomeação de arquivos.
- Criação de estruturas com base em comandos padrão, facilitando a instalação em novos ambientes.

Boas Práticas Recomendadas:

- Modularização de funções.
 - Uso de logs e mensagens de erro descritivas.
 - Versão de desenvolvimento com logs detalhados e versão de produção com interface limpa.
-

10. Conclusão e Próximos Passos

O projeto IA-TRADE está pronto para uso de trading quantitativo, analistas técnicos e desenvolvedores interessados em IA aplicada a finanças.

Destaques alcançados:

- Pipeline completo: dados → modelo → simulação → execução.
- Transparência via explicações IA e painel de performance.
- Modularidade que permite escalar ou integrar novas fontes, modelos e estratégias.