

04 - Avaliação do Modelo

Relatório Explicativo do Código de Avaliação da Rede Neural

Este relatório detalha o funcionamento do código responsável pela **avaliação do desempenho** da Rede Neural Artificial (RNA) treinada previamente para prever movimentos do preço do Bitcoin (subida ou queda) com base em indicadores técnicos.

1. Visão Geral do Código

Este script realiza a análise de desempenho do modelo treinado, através das seguintes etapas:

1. **Carregamento dos Dados:** Importa os dados processados.
 2. **Preparação dos Dados:** Separa os conjuntos de treino e teste.
 3. **Carregamento do Modelo Treinado:** Lê o modelo que já passou por treinamento.
 4. **Avaliação do Modelo:** Realiza previsões, calcula métricas e exibe resultados gráficos.
-

2. Explicação por Etapas

2.1 Carregamento dos Dados

A função `carregar_dados()` realiza:

- Leitura do arquivo `bitcoin_data.csv`.
- Conversão de todas as colunas para o tipo numérico usando `pd.to_numeric()`, evitando erros com dados mal formatados.

- Remoção de qualquer linha com valores ausentes (NaN) com `dropna()`.

Assim, o conjunto de dados utilizado estará limpo e adequado para ser utilizado na avaliação.

2.2 Preparação dos Dados

A função `preparar_dados()` executa:

- Separação das colunas de entrada (features) e saída (target):
 - `X` contém: `Close`, `SMA_14`, `EMA_14`, `RSI_14`, `Volatility`, `Volume_MA_14`.
 - `y` é o `Target`, convertido para `int`.
- Divisão dos dados em treino e teste com `train_test_split()`:
 - Proporção: 80% para treino e 20% para teste.
 - `random_state=42` garante a reprodutibilidade dos resultados.

Embora a avaliação seja feita apenas no conjunto de teste, os dados de treino ainda são necessários aqui por causa da função `train_test_split`.

2.3 Carregamento do Modelo Treinado

A função `carregar_modelo()` carrega o modelo previamente treinado no arquivo `modelo_rna_treinado.keras` usando `load_model()` da Keras.

Com isso, não é necessário reexecutar o processo de treinamento, apenas reutilizar o modelo salvo.

2.4 Avaliação do Modelo

A função `avaliar_modelo()` é a principal e mais analítica do código. Ela realiza os seguintes passos:

1. Previsão:

- O modelo realiza a predição sobre o conjunto de teste.
- Como a saída do modelo é uma **probabilidade** (entre 0 e 1), é aplicada uma **limiarização** com `> 0.5`, convertendo os valores em rótulos binários (0 ou 1).

2. Métricas Avaliadas:

- **Acurácia** (`accuracy_score()`): Percentual de acertos nas previsões.
- **Matriz de Confusão** (`confusion_matrix()`): Mostra os verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos.
- **Relatório de Classificação** (`classification_report()`): Apresenta precisão (precision), revocação (recall), F1-score e suporte para cada classe (subiu ou desceu).

3. Visualização da Matriz de Confusão:

- A matriz é exibida como um **heatmap** (mapa de calor) usando `seaborn.heatmap()`, o que facilita a interpretação visual da performance do modelo.
- Os rótulos personalizados (`'Subiu'` e `'Desceu'`) ajudam a compreender as categorias previstas versus reais.

3. Conclusão

O código aqui descrito representa a etapa final do pipeline de previsão de preços do Bitcoin com base em aprendizado supervisionado. Com ele, é possível verificar o quão eficaz foi o modelo treinado na tarefa de **classificação binária** (subida ou queda).

As métricas fornecidas, como a **acurácia** e o **relatório de classificação**, junto com a **visualização gráfica da matriz de confusão**, oferecem uma análise completa da performance do modelo e ajudam a identificar possíveis melhorias para ajustes futuros.