Nome: Bruno Fernandez Tonso, NºUSP: 5248339

Nome: Eduardo Parri Cuperman, NºUSP: 14567291

Nome: Maria Clara Kaseker de Lara Pinheiro Goes, NºUSP: 10831396

Nome: Matheus Lopes Kanaan, NºUSP: 14598341

Nome: Roberta Boaventura Andrade, NºUSP: 11260832

EAD0830 - IA e ML Aplicados a Finanças

Atividade Computacional 2 – Competição

Inicialmente, como a base de dados contém séries temporais financeiras univariadas (as características da série de interesse são explicadas exclusivamente a partir do comportamento da própria série), os dados precisam ser tratados em questão de sazonalidade e estacionariedade. A primeira para exclusão de ruídos e tendências e a segunda para a simplificação das propriedades matemáticas que facilitam a análise, modelagem e previsão adequadas dos dados, levando a resultados mais confiáveis e estáveis. Adicionalmente precisamos testar as séries para saber se apresentam o comportamento de uma distribuição normal, que nos dirá qual melhor metodologia aplicar para a previsão nos *t* períodos necessários. Para facilitar no desenvolvimento das análises, cada série foi renomeada para A, B, C etc.

Realizando a decomposição das séries, observou-se a ausência de sazonalidade, fator que auxilia no tratamento da estacionariedade de uma série temporal caso ela seja não estacionária. O teste *Augmented Dickey Fuller* (ADF) mostrou que nenhuma das oito séries é estacionária, ou seja, nenhuma delas possui propriedades estatísticas (média, variância e autocorrelação) constantes ao longo do tempo. Tratar a não estacionariedade ou usar um modelo que trabalhe as séries não estacionárias?

Os Teste de Jarque-Bera e de Omnibus indicam evidências significativas para rejeitar a hipótese nula de normalidade para as séries A, B, C e E, sendo as duas primeiras as séries trimestrais. Já as séries D, F, G e H apresentam comportamento de uma normal. Os valores de cada teste foram listados abaixo, lembrando que o p-valor baixo sugere que os dados não são normalmente distribuídos.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Variável** | **Estatística JB** | **p-valor JB** | **Skewness JB** | **Kurtosis JB** | **Estatística Omnibus** | **p-valor Omnibus** |
| A | 4,909 | 0,086 | -0,401 | 1,635 | 22,202 | 0,000 |
| B | 2,721 | 0,256 | 0,049 | 1,825 | 9,723 | 0,008 |
| C | 2,636 | 0,268 | 0,114 | 1,862 | 8,512 | 0,014 |
| D | 1,068 | 0,586 | 0,369 | 3,020 | 1,500 | 0,472 |
| E | 25,268 | 0,000 | 1,187 | 5,696 | 17,599 | 0,000 |
| F | 4,881 | 0,087 | 0,785 | 2,829 | 5,142 | 0,077 |
| G | 1,055 | 0,590 | 0,363 | 3,113 | 1,609 | 0,447 |
| H | 2,995 | 0,224 | 0,560 | 2,477 | 3,191 | 0,203 |

1. Séries Trimestrais: Sendo para as séries trimestrais necessário previsão de 6 trimestres a frente e, para as séries mensais, 12 períodos a frente.
2. Modelo de Redes Neurais Recorrentes (RNNs) e suas variantes (LSTM, GRU):

Prós:

Capazes de capturar dependências temporais de longo prazo

Flexíveis e podem lidar com sequências de comprimentos variados

Adequados para dados não lineares e complexos.

Excelentes para previsões de curto e médio prazo.

Contras:

Propensos a problemas de desaparecimento/explodir do gradiente, especialmente em sequências longas.

Podem ser computacionalmente intensivos para treinamento, especialmente em redes profundas.

Requerem ajuste cuidadoso dos hiperparâmetros para evitar overfitting.

Séries Temporais Adequadas: Séries com dependências temporais complexas e não lineares, como séries financeiras com padrões irregulares e não estacionários.

Dentre as variantes apresentadas no curso, optamos pela LSTM em relação à Rede Neural Feedforward ou a com retroalimentação. Isso, pois esses modelos conseguem captar dependências temporais de médio/longo prazo (múltiplas previsões e não só o próximo valor, como é o caso) melhor, já que lembram informações de eventos passados. Além disso, conseguem aprender padrões complexos e não lineares, também úteis para séries temporais. O único possível problema seria a nossa base de dados não ser grande o suficiente para o treinamento. Similarmente, os modelos ARIMA ou ARIMA Sazonal (não rede neural) erram muito em previsões temporais, porque acabam fazendo previsões baseadas em previsões, o que amplifica o erro. Analisando todos os aspectos, escolhemos a abordagem com memória para a previsão.

Para cada série, inicialmente normalizamos os dados. Em sequência, definimos o modelo com base nas seguintes especificações (as mesmas para todas as séries):

* Modelo sequencial. Ou seja, a informação no modelo flui e não é retroalimentada.
* 100 neurônios na camada LSTM.
* A função de ativação é a ‘relu’ (unidade linear retificada), uma função de ativação que introduz não linearidade na saída da unidade, permitindo que a rede aprenda relações mais complexas nos dados (como não sabíamos a origem dos dados, não queríamos limitar a análise).
* Otimizador: “adam”, adaptador da taxa de aprendizado para cada parâmetro ao longo do tempo. Tornando a convergência do modelo mais rápida e estável.
* Função perda: MSE (Erro Quadrático Médio)
* Para o treino do modelo, realizamos 200 épocas.
* Outras questões específicas da atividade que não detalharemos: número de features em cada etapa temporal ser 1 (séries univariadas); variável de saída única; sem saídas durante o treinamento.

Pontos de melhoria:

Poderíamos ter personalizado as especificações dos modelos para cada série e aumentado a complexidade deles (tentando achar um ponto ótimo para evitar overfitting), o que não fizemos.