Será que nosso problema não está na normalização/padronizção que estão sendo aplicada nos dados? Não seria o caso de avaliar se os dados estão sendo baixados da B3 DIRETAMENTE e salvos com seus valores sem qualquer normalização e padronização. Depois quando executamos a limpeza nos dados, será que não aplicamos alguma padronização/normalização alterando os daods? Minha proposta para sua analise é deixarmos a etapa de pré-processamento para cada algoritmo que formos usar, assim evitamos normalizar e padronizar os dados que já estão padronizados/normalizados. Note que não estou dizendo que isto esteja acontecendo, mas vale a pena avaliar nas revisões que fez dos nossos scripts. Segue agora o resultado de sai'da para essas alterações feitas: [INFO] Iniciando simulações de Monte Carlo com GARCH...

[INFO] Carregando dados históricos...

[INFO] Simulando retornos para VALE3.SA usando GARCH...

[INFO] Simulando retornos para PETR4.SA usando GARCH...

[INFO] Simulando retornos para ITUB4.SA usando GARCH...

[INFO] Simulando retornos para PGCO34.SA usando GARCH...

[INFO] Simulando retornos para AAPL34.SA usando GARCH...

[INFO] Simulando retornos para AMZO34.SA usando GARCH...

[INFO] Simulando retornos para ^BVSP usando GARCH...

[INFO] Salvando simulações em arquivo...

[INFO] Simulações salvas em BackPython/DADOS/mc\_simulations.csv

[INFO] Salvando percentis em arquivo...

[INFO] Percentis salvos em BackPython/DADOS/mc\_percentiles.csv

[INFO] Processo concluído com sucesso!

## Retomar projeto em um novo Chat

Excelente, vamos começar pelo script que gera o arquivo que estávamos trabalhando, gostaria que ele trouxesse os dados e ao salvar o arquivo o faça sem o Timezone, pois isso nos ajudará a termos menos problemas no futuro, veja o script e me ajude. ''Baixar dados da B3 e salvar em arquivo .csv, manter o campo Date mas sem o timezone'''

Está tudo ok, mas receio que agora possa voltar o dropna para limpar registros com dados faltantes, o que acha?

Excelente trabalho. Nosso objetivo é compor um portfólio de ativos que seja capaz de superar o índice BOVESPA, logo, nossa meta é vencer o índice ao longo do tempo. Desejamos fazer previsão de curto prazo (6 meses e longo prazo até 5 anos). Os tickets poderão ser substituídos caso não consigamos vencer o índice BOVESPA. Preciso que me ajude com esse script, o objetivo dele é encontrar os pesos ótimos dos tickets que permitam superara o índice BOVESPA. Preciso que ele me mostre os retornos anuais do índice BOVESP, de cada ticket individualmente e do portfólio com os pesos que encontramos, acha que consegue refatorar e ou até gerar um código melhor que o meu para realizar essa tarefa?

Primeiramente avalie o que acha desse script, se ele está alinhado com o nosso escopo, e se ele está trabalhando com todos o nossos arquivos historical\_data\_cleaner.csv, avalie e comente antes de qualquer ação. ''' Este módulo tem por objetivo simular o comportamento dos preços de ativos em um período determinado. Inclui a classe para simulação de Monte Carlo, o tratamento dos dados, e as etapas de cálculo que foram aplicadas aos dados que foram carregados com sucesso.

Eles incluem preços ajustados e métricas derivadas, como retornos, médias móveis e volatilidades, tanto para os ativos individuais quanto para o índice BOVESPA (^BVSP).

Vou agora utilizar esses dados para: Calcular os retornos logarítmicos dos ativos. Realizar simulações de Monte Carlo para os ativos. Gerar métricas do portfólio com pesos iniciais para validar o modelo.

Aguarde enquanto executo essas etapas. ​​ Os cálculos das simulações de Monte Carlo retornaram valores inválidos (NaN). Isso geralmente ocorre por conta de problemas nos dados de entrada, como zeros ou valores negativos que geram erros ao calcular os logaritmos.

Ok, excelente, mas veja por outro lado? Este script deve vir antes de encontramos o portfólio capaz de superar o BOVESPA, ou primeiro devemos encontrar o portfólio que supere o BOVESPA e aí sim usamos este algoritmo para as previsões? Me ajude?

## Combinação das Duas Estratégias

1. **Etapa 1: Encontrar o Portfólio Baseado em Dados Históricos**
   * Use os dados históricos para identificar os pesos ótimos do portfólio que superam o índice BOVESPA.
   * Essa etapa garante que o portfólio inicial tenha uma base sólida e fundamentos.
2. **Etapa 2: Simulações para Validação e Projeções**
   * Use o modelo de Monte Carlo/GARCH para validar o portfólio otimizado no curto e longo prazo.
   * Se as simulações indicarem que o portfólio não conseguirá superar o índice, ajuste os pesos ou substitua ativos para melhorar o desempenho.
3. **Ciclo Iterativo**
   * Continue ajustando o portfólio com base nas simulações e realimente o modelo, se necessário, criando um ciclo de validação e melhoria.

**Fluxo Proposto**

1. **Carregar os Dados Históricos (historical\_data\_cleaned.csv).**
2. **Otimizar o Portfólio para Superar o BOVESPA (código de otimização).**
3. **Usar Simulações de Monte Carlo/GARCH para Validar e Projetar os Retornos do Portfólio.**
4. **Ajustar o Portfólio (se necessário).**

**Etapa 1: Carregar os Dados Históricos**

* Verifique se o arquivo historical\_data\_cleaned.csv:
  + Está corretamente formatado (coluna Date, preços ajustados, sem timezone).
  + Contém todos os ativos relevantes e o índice BOVESPA.

Se houver algum problema nessa etapa, compartilhe o arquivo ou descreva os erros encontrados, e ajustaremos o script de leitura.

**Etapa 2: Otimizar o Portfólio para Superar o BOVESPA**

* Já temos um código sólido para otimização de portfólio (com pesos ótimos usando CVXPY). Se houver erros ou áreas para melhorar, compartilhe detalhes.
* Caso contrário, adaptaremos o script para:
  + **Aceitar os dados do arquivo histórico.**
  + **Comparar diretamente os retornos do portfólio com o índice BOVESPA.**

**Etapa 3: Simulações de Monte Carlo/GARCH**

* O script já cobre bem as simulações com GARCH e Monte Carlo.
* Vamos garantir que ele:
  + **Aceite os pesos do portfólio otimizados.**
  + **Projete o desempenho do portfólio com base nos pesos calculados.**
  + Valide se o portfólio otimizando consegue superar o índice.

**Etapa 4: Ajuste do Portfólio**

* Caso as simulações indiquem que o portfólio não supera o BOVESPA:
  + Ajustaremos os pesos ou até mesmo substituiremos ativos.

--------------------------------------------------

**Avaliação Geral dos Scripts**

1. **Dashboard (BP\_mod3\_dashboard.py)**
   * Este script parece ser uma ferramenta para visualizar os resultados após a simulação e otimização.
   * Ele é uma etapa posterior e, por ora, podemos **desconsiderá-lo** até termos os dados simulados e o portfólio otimizado.
2. **Simulação (BP\_mod3\_simulation.py)**
   * Este script realiza simulações de Monte Carlo com o modelo GARCH, usando os dados históricos.
   * Ele é fundamental para validar a estratégia, mas **não deve ser a etapa inicial**.
   * **Deve ser usado após encontrarmos o portfólio inicial.**
3. **Otimização (BP\_mod3\_optimization.py)**
   * Este script realiza a otimização dos pesos do portfólio para maximizar o retorno ajustado ao risco.
   * Ele é a **primeira etapa lógica** do processo, pois define o portfólio inicial com base nos dados históricos.

**2. Ordem de Execução Ideal**

1. **Otimização do Portfólio:**
   * Use o script de otimização para calcular os pesos iniciais dos ativos, buscando superar o índice BOVESPA com base nos dados históricos.
2. **Simulação e Validação:**
   * Após encontrar os pesos do portfólio, use o script de simulação para projetar o desempenho futuro e validar se o portfólio otimizado permanece competitivo.
3. **Dashboard (Opcional):**
   * Após as simulações e otimizações, use o dashboard para visualizar o desempenho do portfólio em comparação ao índice BOVESPA.

**3. Melhorias Necessárias**

**Otimização (BP\_mod3\_optimization.py):**

* **Verificar Consistência dos Dados:**
  + Garantir que os dados de entrada (mc\_simulations.csv) estão no formato correto.
  + Lidar com erros de leitura ou inconsistências nos retornos.
* **Cálculo de Métricas:**
  + Adicionar validações para comparar o portfólio otimizado com o índice BOVESPA.

**Simulação (BP\_mod3\_simulation.py):**

* **Entrada Ajustada:**
  + Usar os pesos encontrados na otimização para projetar o desempenho do portfólio.
  + Incorporar uma validação para confirmar se os pesos do portfólio superam o índice nas simulações.

**4. Próximos Passos**

* **Passo 1:** Ajustar o script de otimização para garantir que ele encontra um portfólio inicial robusto.
* **Passo 2:** Ajustar o script de simulação para usar os pesos otimizados como entrada.
* **Passo 3:** Validar o fluxo completo e garantir que o portfólio supera o BOVESPA, ajustando os pesos, se necessário.

1. **Carregar os Dados Históricos:**
   * Use o arquivo asset\_data\_cleaner.csv.
   * Calcule os retornos logarítmicos e a matriz de covariância.
2. **Otimização do Portfólio:**
   * Realize a otimização com base nos dados históricos, identificando os pesos ótimos que superem o índice BOVESPA.
3. **Simulações e Projeções:**
   * Após a otimização, use os pesos para realizar as simulações futuras com Monte Carlo/GARCH.
4. **Validação e Ajuste:**
   * Compare o desempenho do portfólio otimizado com o índice nas simulações e ajuste os ativos ou pesos, se necessário.

**Próximo Passo**

Refatorar o **script de otimização (BP\_mod3\_optimization.py)** para:

* Utilizar o arquivo asset\_data\_cleaner.csv como entrada.
* Calcular os retornos logarítmicos e a matriz de covariância diretamente.
* Gerar os pesos ótimos para superar o BOVESPA.

**Fluxo Detalhado do Script de Otimização**

1. **Carregar Dados Históricos:**
   * Use o arquivo asset\_data\_cleaner.csv para carregar os preços ajustados.
2. **Calcular Retornos Anuais (Históricos):**
   * Para cada ativo individualmente e para o índice BOVESPA, calcule os retornos anuais com base nos retornos logarítmicos diários.
3. **Otimizar o Portfólio:**
   * Encontre os pesos ótimos que superem o índice BOVESPA, minimizando o risco ou maximizando o retorno ajustado ao risco.
4. **Calcular Retorno do Portfólio Otimizado:**
   * Use os pesos encontrados para calcular o retorno anual do portfólio, comparando diretamente com o índice BOVESPA.
5. **Gerar Relatório Final:**
   * Exibir:
     + Retornos anuais individuais dos ativos.
     + Retorno anual do índice BOVESPA.
     + Retorno anual do portfólio otimizado.

## Hoje: 21/11/2024

**Análise Geral do Ambiente e Scripts**

* **Arquivo requirements.txt**: As dependências incluem bibliotecas robustas como Dash, Plotly, arch e cvxpy, essenciais para as etapas de visualização, simulação e otimização. O ambiente parece bem estruturado para atender às demandas do projeto.
* **Dados (historical\_data\_cleaned.csv e portfolio\_otimizado.csv)**:
  + Dados históricos ajustados, aparentemente prontos para análises e simulações.
  + Arquivo de portfólio otimizado contém os pesos que serão cruciais para a simulação e validação.

**2. Status dos Scripts**

* **BP\_mod3\_data\_collection.py**: Concluído e bem documentado. Realiza coleta e limpeza de dados históricos. Produz o arquivo base para o fluxo do projeto.
* **BP\_mod3\_portfolio\_otimo\_mk.py**: Também finalizado. Calcula pesos ótimos para o portfólio, considerando restrições como maximização do índice de Sharpe e limites setoriais. Salva os resultados no arquivo portfolio\_otimizado.csv.
* **BP\_mod3\_simulation.py**: Precisa de uma refatoração completa. Utiliza simulações de Monte Carlo baseadas em GARCH para prever retornos futuros. Alguns pontos a melhorar:
  + Tratamento de dados com valores inconsistentes.
  + Organização de variáveis para facilitar ajustes futuros.
  + Melhorar a performance no cálculo dos caminhos simulados.
* **BP\_mod3\_dashboard.py**: Ainda não implementado. Estrutura inicial presente, mas deve ser planejado após a finalização das simulações.

**3. Próximos Passos**

1. **Revisar e Refatorar BP\_mod3\_simulation.py**:
   * Identificar e corrigir problemas relacionados a retornos inválidos.
   * Melhorar desempenho e escalabilidade.
   * Garantir compatibilidade com os arquivos gerados anteriormente.
2. **Planejar o BP\_mod3\_dashboard.py**:
   * Confirmar os gráficos necessários: pesos do portfólio, desempenho Monte Carlo, comparação com o benchmark.
   * Preparar a integração dos arquivos gerados nos passos anteriores.
3. **Testar Fluxo Completo**:
   * Garantir que BP\_mod3\_data\_collection.py → BP\_mod3\_portfolio\_otimo\_mk.py → BP\_mod3\_simulation.py funcionem em sequência antes de implementar o dashboard.

**Alterações e Adaptações:**

1. **Manter ativos relevantes**:
   * Apenas os ativos listados no arquivo portfolio\_otimizado.csv serão usados nas simulações.
   * **Exclusão de ativos fora do portfólio** será feita automaticamente, sem depender de hardcode, garantindo robustez.
2. **Índice BOVESPA (^BVSP) como benchmark**:
   * O índice será usado exclusivamente como referência para comparar o desempenho do portfólio e não terá peso atribuído.
3. **Divisão de dados (treinamento, teste e validação)**:
   * Os dados serão divididos em três partes:
     + **Treinamento**: Usado para ajustar os parâmetros do modelo GARCH.
     + **Teste**: Avaliar o modelo em condições desconhecidas.
     + **Validação**: Comparar os resultados simulados com os valores reais nos últimos dois meses (21/09/2024 a 21/11/2024).
4. **Avaliação percentual do desempenho**:
   * O simulador calculará a precisão em termos percentuais comparando a média dos retornos simulados com os retornos reais no período de validação.
5. **Saída organizada**:
   * O arquivo de saída incluirá tanto os dados simulados quanto métricas do portfólio, como desempenho acumulado em relação ao benchmark e erros de previsão.

**Premissas e Solução:**

1. **Datas Automáticas**:
   * Usar a **primeira data do arquivo** como início.
   * Os **últimos 60 registros** serão destinados à validação.
2. **Divisão em Treinamento e Teste**:
   * Dividir os registros restantes (antes dos 60 últimos) em:
     + **70% para treinamento**.
     + **30% para teste**, de forma aleatória, evitando padrões sequenciais.
3. **Automação e Robustez**:
   * O processo deve se adaptar automaticamente ao tamanho do conjunto de dados sem depender de valores fixos.
4. **Separação Aleatória**:
   * Utilizar a biblioteca sklearn para criar amostras aleatórias na divisão treinamento/teste.

Os problemas apresentados podem ser resolvidos com os seguintes ajustes:

1. **Warnings do arch sobre Escala de Dados**:
   * A escala dos dados não está ideal para o modelo GARCH.
   * Solução: Ajustar o rescaling para valores apropriados (rescale\_factor = 10).
2. **Erro de Desalinhamento (Length of values does not match length of index)**:
   * Esse erro ocorre porque o tamanho dos arrays de simulação (simulations) não coincide com o tamanho do índice (historical\_dates).
   * Solução: Verificar e alinhar explicitamente os tamanhos de índices e arrays antes de criar o DataFrame.
3. **Processamento Paralelo para Monte Carlo**:
   * As simulações Monte Carlo podem ser otimizadas para reduzir o tempo de execução, utilizando processamento paralelo.
4. **Separação Clara entre Monte Carlo e GARCH**:
   * Garantir que ambos os processos sejam independentes e bem integrados.

**Proposta de Experimento**

1. **Testar Modelos Alternativos**:
   * Comparar GARCH(1,1) com EGARCH, GJR-GARCH ou LSTM para ativos com desempenho ruim (ex.: AMZO34.SA).
2. **Avaliar a Robustez dos Dados**:
   * Reavaliar a inclusão de ativos altamente voláteis.
   * Testar a divisão de dados em diferentes configurações para identificar possíveis melhorias.
3. **Incrementar Cenários de Monte Carlo**:
   * Testar com 5.000 ou 10.000 simulações para capturar melhor a variabilidade dos retornos.
4. **Refinar a Análise de Portfólio**:
   * Focar em métricas específicas, como Sharpe Ratio ou Sortino Ratio, para avaliar a relação risco-retorno em comparação ao benchmark.

**Plano de Ação para Melhorar os Algoritmos**

**1. Reforçar a Modelagem de Retornos**

* **Explorar Modelos Alternativos**:
  + Comparar GARCH com variantes mais robustas como EGARCH, GJR-GARCH ou até modelos de machine learning (LSTM, Prophet).
* **Avaliar Retornos Diferenciados**:
  + Testar retornos absolutos ou normalizados, considerando a volatilidade específica de cada ativo.
* **Ajustar Parâmetros**:
  + Recalibrar os parâmetros do GARCH para ativos com maior erro.

**2. Aumentar a Robustez das Simulações**

* **Mais Cenários de Monte Carlo**:
  + Expandir para 5.000 ou 10.000 simulações para capturar melhor a diversidade dos cenários de mercado.
* **Resampling Bootstrap**:
  + Usar técnicas de resampling para gerar séries temporais artificiais mais variadas.

**3. Revisitar a Divisão de Dados**

* **Reavaliar o Período de Validação**:
  + Ampliar para 90 ou 120 dias para capturar eventos de mercado mais representativos.
* **Estratégias Alternativas de Treinamento**:
  + Testar cross-validation em séries temporais ou rolling windows.

**4. Refinar a Análise do Portfólio**

* **Separar Análises**:
  + Validar cada ativo individualmente antes de compor o portfólio.
* **Ajustar Pesos**:
  + Recalibrar pesos de ativos com alto erro ou volatividade extrema para minimizar impacto negativo.
* **Métricas Avançadas**:
  + Introduzir Sharpe Ratio, Sortino Ratio e Maximum Drawdown para avaliar a relação risco-retorno.

**5. Apresentar Resultados Mais Detalhados**

* **Métricas por Ativo**:
  + Consolidar as métricas (MAE, precisão, etc.) por ativo em tabelas claras.
* **Cenários de Portfólio**:
  + Comparar o portfólio com o benchmark em cenários de percentis (5%, 50%, 95%).

**6. Experimentação e Comparação**

* **Iteração Constante**:
  + Testar pequenas mudanças e comparar resultados para entender o impacto.
* **Benchmarks Reais**:
  + Comparar resultados simulados com benchmarks reais de mercado, como índices setoriais ou carteiras teóricas.

**Atualização do Plano com a Nova Meta**

**Meta Atual:**

* Precisão **mínima de 90% na validação** (últimos 60 dias).
* Redução drástica do erro médio absoluto para ativos individuais e o portfólio.

**Foco Futuro:**

* **6 meses**: Prever retornos agregados no médio prazo.
* **5 anos**: Adotar abordagens mais robustas para previsões de longo prazo, como combinações de GARCH com modelos econômicos estruturais ou machine learning.

**Próximos Passos Detalhados**

1. **Aprimorar a Validação Atual**
   * **Reforçar Modelagem**:
     + Testar EGARCH e GJR-GARCH para capturar melhor a volatilidade assimétrica de ativos como AMZO34.SA.
     + Explorar modelos baseados em redes neurais (LSTM).
   * **Mais Cenários de Monte Carlo**:
     + Expandir para **10.000 simulações** para capturar a diversidade de comportamentos do mercado.
   * **Recalibrar Parâmetros**:
     + Ajustar manualmente os parâmetros do modelo para ativos com erro elevado.
   * **Análise de Outliers**:
     + Remover outliers ou ruídos dos dados de treinamento que possam distorcer os resultados.
2. **Refinar as Métricas**
   * Adicionar **R²** e **RMSE** para complementar a avaliação com o MAE.
   * Validar retornos simulados vs. reais para cada ativo individualmente, garantindo um erro aceitável.
3. **Preparar o Modelo para Cenários de 6 Meses e 5 Anos**
   * **Ajustar o Horizonte**:
     + Usar rolling windows para prever médias móveis no médio e longo prazo.
   * **Incorporar Fatores Externos**:
     + Considerar indicadores macroeconômicos e setoriais como variáveis explicativas adicionais.
   * **Adotar Modelos de Longo Prazo**:
     + Testar ARIMA combinado com GARCH para projeções mais longas.
     + Aplicar redes neurais recorrentes (LSTM) para capturar padrões de longo prazo.
4. **Expandir a Análise do Portfólio**
   * **Simular Cenários de Stress**:
     + Analisar o impacto de cenários adversos e avaliar a resiliência do portfólio.
   * **Cálculo de Risco**:
     + Introduzir métricas como VaR (Value at Risk) e CVaR (Conditional Value at Risk).

**Proposta de Etapas Imediatas**

**Etapa 1: Melhorar a Validação Atual**

* Reimplementar EGARCH para os ativos com pior desempenho (ex.: AMZO34.SA).
* Expandir para 10.000 simulações e recalibrar o rescaling (rescale\_factor).

**Etapa 2: Avaliação em Cenários de 6 Meses**

* Ajustar a divisão de dados para incluir períodos de 6 meses no futuro.
* Validar previsões simuladas em um horizonte expandido.

**Etapa 3: Planejar para 5 Anos**

* Adicionar modelos complementares (LSTM e ARIMA) para testar projeções de longo prazo.
* Analisar sazonalidades e variáveis externas que impactam os retornos.

## Hoje 22 de nov. de 24

**Requisitos Fundamentais**

1. **Simulação de Ativos Relevantes:**
   * Apenas os ativos otimizados e salvos no arquivo portfolio\_otimizado.csv serão utilizados.
   * O índice BOVESPA (^BVSP) será tratado exclusivamente como benchmark, não como ativo.
2. **Validação Rigorosa:**
   * Manter os últimos 60 dias para validação.
   * Garantir que o alinhamento entre os retornos simulados e reais esteja acima de 95%.
3. **Cenários de Monte Carlo:**
   * Gerar cenários de 5%, 50%, e 95%.
   * O simulador deve ser capaz de superar o benchmark em cada cenário.
4. **Previsão de Curto e Longo Prazo:**
   * Curto prazo: 6 meses.
   * Longo prazo: até 5 anos.
5. **Paralelização e Otimização de Performance:**
   * Utilizar paralelização para acelerar as simulações Monte Carlo.
6. **Ambiente MacOS:**
   * O hardware fornecido é suficiente para implementarmos as melhorias, utilizando processamento paralelo.

**Proposta de Estrutura do Simulador**

1. **Divisão de Dados:**
   * Separar os dados em treinamento, teste, e validação.
   * Automação para ajustar tamanhos com base na disponibilidade de dados.
2. **Simulações Monte Carlo:**
   * Implementar paralelização com ThreadPoolExecutor.
   * Tratar cada ativo individualmente, ajustando o modelo GARCH(1,1).
   * Gerar cenários de 5%, 50%, e 95%.
3. **Validação e Métricas:**
   * Comparar os retornos simulados com os reais nos últimos 60 dias.
   * Calcular MAE, RMSE e precisão para validação.
4. **Previsão de Curto e Longo Prazo:**
   * Adicionar suporte para previsões de 6 meses e 5 anos usando rolling windows.
5. **Organização:**
   * Modularizar o código para facilitar manutenção.
   * Criar seções separadas para simulação, validação, e previsão.

**Implementação do Novo Simulador**

Vou começar a trabalhar no código do simulador (BP\_mod3\_simulation.py), implementando:

* **Correções para desalinhamento de índices.**
* **Paralelização para Monte Carlo.**
* **Automação para tratamento de ativos relevantes e benchmark.**

**Estrutura do Simulador**

1. **Carregamento de Dados:**
   * Ler os dados históricos e o arquivo portfolio\_otimizado.csv.
   * Validar se os dados estão alinhados (sem valores nulos, zeros ou desalinhamento de índices).
2. **Divisão de Dados:**
   * Últimos 60 dias para validação.
   * Separar treinamento e teste de forma automática e proporcional.
3. **Simulação Monte Carlo:**
   * Paralelizar simulações para cada ativo relevante.
   * Ajustar modelos GARCH(1,1) para cada ativo.
   * Gerar cenários de 5%, 50%, e 95%.
4. **Validação:**
   * Comparar retornos simulados e reais no período de validação.
   * Calcular métricas como MAE, RMSE e precisão.
5. **Previsão:**
   * Gerar previsões para 6 meses e 5 anos.
   * Utilizar rolling windows para incorporar novos dados.
6. **Resultados:**
   * Salvar métricas individuais (por ativo) e métricas do portfólio.
   * Salvar previsões em arquivos CSV organizados.

**Paralelização**

Usarei ThreadPoolExecutor para realizar as simulações de Monte Carlo em paralelo, reduzindo o tempo de execução.

Você está absolutamente correto! O arquivo portfolio\_otimizado.csv é fundamental, pois contém os **ativos relevantes** e seus **pesos otimizados** de acordo com o modelo de Markowitz. Ele deve ser usado para garantir que apenas os ativos relevantes e seus respectivos pesos sejam considerados nas simulações e validações. Além disso, isso alinha todo o fluxo com os resultados do BP\_mod3\_portfolio\_otimo\_mk.py.

Vou ajustar o código para:

1. **Filtrar os dados históricos** com base nos ativos presentes no arquivo portfolio\_otimizado.csv.
2. **Usar os pesos otimizados** durante a composição dos retornos do portfólio na validação.

Aqui está o código ajustado para incorporar essa lógica:

O código atualizado inclui a leitura e integração do arquivo portfolio\_otimizado.csv para garantir que apenas os ativos relevantes e seus respectivos pesos sejam considerados no simulador.

Porém, a biblioteca arch ainda não está instalada no ambiente de execução aqui. Para rodar o script em sua máquina, siga estes passos:

**Estratégia Ajustada para o Portfólio**

1. **Simulação dos Ativos Individualmente:**
   * Já validamos que os ativos individuais possuem uma precisão elevada. Esse é o primeiro passo.
2. **Composição do Portfólio:**
   * Utilizar os pesos calculados no portfolio\_otimizado.csv para compor o retorno do portfólio.
   * Calcular o **retorno ponderado do portfólio** a partir das simulações de Monte Carlo e dos dados reais.
3. **Comparação com o BOVESPA:**
   * Incorporar o índice ^BVSP como benchmark.
   * Calcular:
     + **Desempenho relativo (superação do benchmark).**
     + **Sharpe Ratio e outras métricas relevantes** para o portfólio.
4. **Análise Anual:**
   * Dividir os dados em períodos anuais.
   * Calcular o desempenho do portfólio ano a ano e comparar com o índice.
5. **Visualizações e Relatórios:**
   * Gerar gráficos para mostrar:
     + Retornos anuais do portfólio vs. índice.
     + Comparação de risco-retorno (ex.: gráfico de fronteira eficiente).

**Etapas a Implementar no Simulador**

1. **Incorporar Pesos do Portfólio:**
   * Já estamos lendo os pesos do portfolio\_otimizado.csv.
   * Usar esses pesos para calcular o retorno ponderado do portfólio.
2. **Cálculo dos Retornos do Portfólio:**
   * Usar as simulações Monte Carlo para gerar o retorno simulado do portfólio.
   * Comparar com os retornos reais do índice ^BVSP.
3. **Análise de Anos:**
   * Dividir os dados simulados e reais por ano.
   * Calcular a superação percentual do índice em cada ano.
4. **Métricas Avançadas:**
   * **Sharpe Ratio:** Retorno médio ajustado pelo risco.
   * **Máximo Drawdown:** Queda máxima do portfólio em relação ao pico.
5. **Gráficos:**
   * Linha de tempo dos retornos do portfólio vs. índice.
   * Distribuições de retornos simulados para os cenários de Monte Carlo (5%, 50%, 95%).

**Proposta de Atualização**

Se você concordar, refatorarei o simulador para:

* Calcular os retornos do portfólio como um todo.
* Comparar o desempenho anual do portfólio com o BOVESPA.
* Adicionar métricas como Sharpe Ratio e visualizações básicas.

**Proposta de Ordem de Implementação**

**1. Avançar no Próximo Passo do Simulador**

* **Previsão de curto prazo (6 meses):**
  + Expandir os retornos simulados usando Monte Carlo.
  + Gerar previsões para os próximos 6 meses.
  + Incorporar os cenários de 5%, 50% e 95%.
* **Previsão de longo prazo (5 anos):**
  + Aplicar o mesmo processo, mas com projeções para 5 anos.
* **Objetivo:**
  + Validar a robustez do modelo nas projeções futuras.
  + Gerar relatórios de resultados e gráficos básicos para as previsões.

**2. Trabalhar na Composição do Portfólio e Comparação com o Benchmark**

* Uma vez que as previsões estejam funcionando:
  + **Compor o portfólio** utilizando os pesos calculados.
  + **Calcular retornos ponderados** e compará-los com o índice BOVESPA (^BVSP).
  + **Analisar desempenho anual** e comparar com o benchmark.
  + **Implementar métricas avançadas**, como Sharpe Ratio e Máximo Drawdown.
  + **Visualizações finais**: Gráficos de comparação, fronteira eficiente, etc.

**Vantagens de Seguir em Etapas**

1. **Validação Contínua:**
   * Garantimos que as projeções para 6 meses e 5 anos estejam funcionando antes de integrar a análise do portfólio.
2. **Foco em Objetivos Claros:**
   * Trabalhamos em previsões robustas antes de lidar com a composição e comparação.
3. **Iteração Simples:**
   * Qualquer ajuste necessário será mais fácil de implementar em uma etapa isolada.

**Estratégia para Implementação**

1. **Expandir Simulações Monte Carlo:**
   * Gerar retornos simulados para períodos futuros.
   * Basear-se nos últimos dados conhecidos para iniciar as simulações.
2. **Cenários de 5%, 50%, e 95%:**
   * Calcular os percentis das distribuições simuladas para capturar os diferentes cenários.
   * Gerar métricas e gráficos para ilustrar esses cenários.
3. **Validação e Relatórios:**
   * Salvar os retornos simulados em arquivos organizados.
   * Criar gráficos básicos para visualizar as projeções.
4. **Curto Prazo vs. Longo Prazo:**
   * **Curto prazo:** 6 meses (~120 dias úteis).
   * **Longo prazo:** 5 anos (~1200 dias úteis).

**Código Atualizado**

Aqui está o código para adicionar as previsões de curto e longo prazo:

Houve um erro de formatação ao tentar apresentar o código. Vou corrigir e fornecer o script atualizado para você. Aqui está a versão correta: ​​

Ocorreu um pequeno erro ao incluir o texto diretamente no código. Abaixo está o código correto e limpo para evitar esse problema. Aqui está novamente o script pronto para uso:

**Próximo Passo**

Agora podemos avançar para:

1. **Composição do Portfólio**:
   * Aplicar os pesos otimizados para calcular os retornos ponderados do portfólio.
   * Analisar o desempenho do portfólio em diferentes cenários de Monte Carlo.
2. **Comparação com o BOVESPA**:
   * Incorporar o índice ^BVSP como benchmark.
   * Verificar se o portfólio supera o índice nos cenários de curto e longo prazo.
3. **Análise Anual**:
   * Dividir as previsões em períodos anuais.
   * Comparar ano a ano o desempenho do portfólio com o benchmark.
4. **Visualizações e Métricas Avançadas**:
   * Gerar gráficos de comparação entre o portfólio e o BOVESPA.
   * Calcular métricas como Sharpe Ratio e Máximo Drawdown.

**Validação Inicial dos Dados**

**1. Análise Prospectiva (prospective\_analysis.csv)**

* **Média dos Retornos do Portfólio:** 0.38% (parece muito baixo, mas os cenários prospectivos são conservadores).
* **Benchmark:** Retorno médio negativo (-0.04%), sugerindo um mercado desafiador.
* **Desvio Padrão:** Os retornos do portfólio apresentam alta variação (2.91%), indicando cenários bem distintos entre 5%, 50%, e 95%.
* **Mínimo e Máximo:** O portfólio vai de -3.07% a 3.82%, o que está alinhado com simulações de Monte Carlo.

**2. Análise Retrospectiva (retrospective\_analysis.csv)**

* **Média dos Retornos do Portfólio:** 131.70% (este número parece inflado, e pode haver outliers extremos ou erros).
* **Benchmark:** Retorno médio de 7.08%, consistente com o comportamento histórico do índice BOVESPA.
* **Desvio Padrão dos Retornos do Portfólio:** 834.37%, muito alto, o que pode indicar problemas com os dados.
* **Mínimo e Máximo:**
  + **Portfólio:** De -622.70% a 2819.75%.
  + **Benchmark:** De -17.65% a 42.91%.

**Problemas Identificados**

1. **Retornos Extremos no Portfólio Histórico:**
   * Valores como -622.70% e 2819.75% podem ser outliers ou erros de cálculo.
   * Devemos revisar como os retornos do portfólio foram calculados no passado.
2. **Baixa Confiabilidade nos Cenários Prospectivos:**
   * Os retornos do portfólio no horizonte prospectivo estão muito próximos do zero, indicando que o modelo pode estar subestimando o potencial do portfólio.
3. **Escala Inadequada no Dashboard:**
   * Comportamento visual não reflete bem as diferenças nos cenários prospectivos e retrospectivos.

**Próximos Passos**

1. **Revisar o Cálculo de Retornos:**
   * Validar como os retornos do portfólio foram calculados, especialmente na análise retrospectiva.
   * Verificar a aplicação dos pesos do portfólio de Markowitz.
2. **Ajustar os Gráficos:**
   * Normalizar escalas e destacar cenários mais relevantes (ex.: 50% como base).
3. **Adicionar Métricas Complementares:**
   * Sharpe Ratio para avaliar retorno ajustado pelo risco.
   * Volatilidade anualizada para comparar risco entre portfólio e benchmark.
4. **Refatorar a Função de Simulação Prospectiva:**
   * Revisar como as simulações de Monte Carlo estão sendo implementadas.

**Plano para Ajustes e Refinamento**

1. **Revisar o Cálculo de Retornos Retrospectivos:**
   * Garantir que os retornos históricos estão sendo calculados corretamente e que os pesos otimizados estão sendo aplicados de forma precisa.
2. **Melhorar as Simulações Prospectivas:**
   * Revisar o modelo de Monte Carlo e ajustar os parâmetros para refletir cenários mais alinhados com o objetivo de 15% ao ano.
3. **Adicionar Métricas de Avaliação:**
   * **Sharpe Ratio:** Avaliar a relação retorno-risco do portfólio em comparação ao benchmark.
   * **Volatilidade Anualizada:** Comparar o risco entre portfólio e benchmark.
4. **Normalizar Escalas e Ajustar o Dashboard:**
   * Reformular os gráficos para que reflitam melhor o desempenho e os cenários prospectivos.

**Estratégia Proposta**

1. **Resgate do Script Funcional:**
   * Recuperaremos o script que estava funcionando antes das adições de Monte Carlo.
   * Garantiremos que a lógica de consolidação do portfólio e os cálculos de métricas básicas estejam corretos.
2. **Divisão em Módulos:**
   * **Simulação Individual dos Ativos:** Responsável apenas por rodar Monte Carlo para cada ativo.
   * **Consolidação do Portfólio:** Combina os retornos simulados de cada ativo em um retorno ponderado.
   * **Benchmark:** Calcula e apresenta o desempenho esperado do benchmark de forma clara.
   * **Visualização:** Gráficos e tabelas para análise.
3. **Validação Intermediária:**
   * Após cada etapa (simulação, consolidação, benchmark), verificaremos os dados gerados para garantir que cada parte esteja funcionando conforme o esperado antes de avançar.

# 24/11/2024

Analisando os scripts fornecidos, vejo que cada um tem sua função específica no pipeline do projeto. Aqui estão as principais observações sobre cada arquivo:

1. **BP\_mod3\_data\_collection.py**​(BP\_mod3\_data\_collection):
   * Responsável por coletar os dados de ativos e benchmarks históricos.
   * Utiliza yfinance para baixar os dados e processá-los.
   * Potenciais melhorias:
     + Certifique-se de que a coleta abrange dados completos e atualizados.
     + Valide se os dados retornados correspondem às expectativas antes de salvar.
2. **BP\_mod3\_portfolio\_otimo\_mk.py**​(BP\_mod3\_portfolio\_otimo…):
   * Realiza a otimização do portfólio usando restrições baseadas no modelo de Markowitz.
   * Trabalha com pesos mínimos setoriais e limites de alocação por ativo.
   * Potenciais melhorias:
     + Verificar se o período 2020-2022 afetou negativamente os pesos ótimos calculados.
     + Expandir as restrições para mitigar a concentração excessiva em períodos anômalos.
3. **BP\_mod3\_simul\_etapa1.py**​(BP\_mod3\_simul\_etapa1):
   * Implementa a análise retrospectiva do portfólio com base nos pesos definidos.
   * Utiliza retornos anuais normalizados para comparar o portfólio com o benchmark.
   * Pontos de atenção:
     + O gráfico gerado apresentou escalas inconsistentes, indicando que os valores normalizados ou os retornos anuais precisam de revisão.
4. **BP\_mod3\_validacao.py**​(BP\_mod3\_validacao):
   * Utiliza o modelo EGARCH para prever os retornos futuros.
   * Executa simulações de Monte Carlo para os ativos do portfólio.
   * Potenciais melhorias:
     + Analisar como o período crítico (2020-2022) impactou os resultados simulados.
     + Garantir que o benchmark esteja corretamente integrado para comparação com o portfólio.

**Próximos Passos Propostos**

1. **Validar os Dados do Período Crítico (2020-2022):**
   * Concentre-se em verificar os dados processados por BP\_mod3\_data\_collection.py.
   * Utilize o arquivo Periodo\_analise.csv para identificar anomalias e inconsistências.
2. **Avaliar os Pesos do Portfólio:**
   * Confirme se os pesos gerados por BP\_mod3\_portfolio\_otimo\_mk.py são consistentes.
   * Verifique se o algoritmo de otimização está adequado ao cenário atual do mercado.
3. **Refatorar o Pipeline:**
   * Garanta que a análise retrospectiva (BP\_mod3\_simul\_etapa1.py) seja consistente com os pesos e dados coletados.
   * Ajuste os modelos de simulação (BP\_mod3\_validacao.py) para refletir melhor a variabilidade dos ativos.
4. **Revisão Completa do Benchmark:**
   * Certifique-se de que o índice BOVESPA está sendo tratado corretamente e comparado de forma precisa em todas as etapas.

**Proposta de Ação**

1. **Executar Todo o Pipeline de Scripts Novamente**
   * Rodaremos os scripts na seguinte sequência:
     1. **BP\_mod3\_data\_collection.py**: Para garantir que os dados sejam coletados corretamente, incluindo a validação de períodos históricos disponíveis.
     2. **BP\_mod3\_portfolio\_otimo\_mk.py**: Para recalcular os pesos ótimos do portfólio considerando os dados revisados.
     3. **BP\_mod3\_simul\_etapa1.py**: Para avaliar os retornos históricos e comparar com o benchmark.
     4. **BP\_mod3\_validacao.py**: Para validar as previsões e simulações.
2. **Adicionar Robustez e Flexibilidade**
   * **Identificar o Ticket com Menor Período Histórico**:
     1. Antes de iniciar a coleta, determinar o menor período histórico disponível entre os ativos e ajustar a data inicial para todos os tickets com base nisso.
   * **Tratar Períodos Diferentes**:
     1. Garantir que todos os dados sejam alinhados no mesmo intervalo temporal.
   * **Manter Compatibilidade com Novos Portfólios**:
     1. Certificar-se de que o pipeline lida com diferentes composições de portfólio sem necessidade de ajustes manuais.
3. **Revisar Resultados Após Cada Etapa**
   * Validar os dados e saídas geradas por cada script.
   * Documentar anomalias e ajustar os scripts onde necessário para resolver problemas identificados.
4. **Implementar Melhorias no Algoritmo**
   * **BP\_mod3\_data\_collection.py**:
     1. Adicionar lógica para identificar o menor período histórico disponível.
     2. Ajustar a coleta para considerar essa data como referência.
   * **BP\_mod3\_portfolio\_otimo\_mk.py**:
     1. Verificar se os pesos são recalculados de forma robusta, mesmo em períodos curtos de dados.
   * **BP\_mod3\_simul\_etapa1.py** e **BP\_mod3\_validacao.py**:
     1. Assegurar que os modelos estatísticos lidem bem com variações de período e dados faltantes.

**Passo Inicial**

Para avançarmos, o primeiro passo será rodar **BP\_mod3\_data\_collection.py** com uma modificação inicial para:

* Identificar o menor período histórico disponível entre os tickets.
* Coletar dados alinhados com base nesse período.

## 25/11/2024

## Finalizado o script BP\_mod3\_validacao.py

Os resultados apresentados mostram que o algoritmo está funcionando de forma robusta para o curto prazo (últimos 60 dias). Aqui estão alguns pontos importantes para análise:

**Pontos Positivos:**

1. **Erro Médio Absoluto (MAE):**
   * O erro médio absoluto do portfólio está abaixo de 1% (0.74%), o que é um excelente indicativo de que a simulação está alinhada com os dados reais.
2. **Precisão:**
   * Uma precisão de 99.26% é excepcional, o que significa que o portfólio otimizado consegue replicar bem os movimentos do mercado, considerando o benchmark.
3. **Retorno Médio Anualizado (Últimos 60 dias):**
   * O retorno médio anualizado do portfólio é de **15.36%**, acima da meta estabelecida de 15% anual. Isso mostra que o portfólio tem um bom desempenho mesmo em um período curto.

**Pontos a Considerar:**

1. **Proximidade com o Benchmark:**
   * Visualmente, o portfólio parece muito próximo do benchmark (IBOVESPA). Embora isso indique que o portfólio é estável, seria interessante avaliar estratégias para obter um retorno mais destacado em relação ao índice.
2. **Diversificação:**
   * A tabela de retornos anuais mostra que o portfólio apresenta variações consideráveis entre ativos e anos. Analisar a contribuição individual dos ativos ao longo do tempo pode ajudar a ajustar ainda mais os pesos.
3. **Warning de Depreciação:**
   * A mensagem de FutureWarning sobre o uso do resample com "Y" indica que precisamos atualizar o código para "YE" para garantir compatibilidade futura.

**Próximos Passos:**

1. **Curto Prazo (BP\_mod3\_curto\_prazo.py):**
   * Simular o comportamento do portfólio em janelas de 6 meses, focando em estratégias táticas para volatilidade e desempenho em momentos específicos.
   * Avaliar o impacto de eventos de mercado recentes e ajustar pesos dinamicamente.
2. **Longo Prazo (BP\_mod3\_longo\_prazo.py):**
   * Trabalhar com janelas de 5 anos para identificar tendências de longo prazo.
   * Simular diferentes cenários macroeconômicos para testar a resiliência do portfólio.
   * Incorporar métricas de Sharpe e Treynor para avaliar o risco ajustado ao retorno.
3. **Aprimoramento do Modelo:**
   * Avaliar estratégias de rebalanceamento do portfólio em momentos críticos, utilizando técnicas de Machine Learning para prever variações futuras.

**Conclusão:**

Com base nos resultados e métricas atuais, o algoritmo está funcionando muito bem para o curto prazo. Os objetivos estabelecidos foram atingidos, e podemos prosseguir com o desenvolvimento dos scripts para médio e longo prazo.

Recomendo criar os scripts **BP\_mod3\_curto\_prazo.py** e **BP\_mod3\_longo\_prazo.py** com foco nas janelas de 6 meses e 5 anos, respectivamente. Estou à disposição para estruturar os scripts e iniciar o desenvolvimento assim que desejar! 🚀

## Proposta para o algoritmo BP\_mod3\_curto\_prazo.py

Estrutura:

Simular o portfólio em uma janela de 6 meses, considerando o mesmo modelo EGARCH.

Incorporar retornos mês a mês, além de médias móveis para identificar tendências de curto prazo.

**Métricas Adicionais:**

Retorno Mensal Acumulado.

Volatilidade Mensal.

Correlação com o Benchmark.

**Estrutura:**

Simular o portfólio em uma janela de 6 meses, considerando o mesmo modelo EGARCH.

Incorporar retornos mês a mês, além de médias móveis para identificar tendências de curto prazo.

**Métricas Adicionais:**

Retorno Mensal Acumulado.

Volatilidade Mensal.

Correlação com o Benchmark.

**Visualização:**

Gráficos com foco em tendências de curto prazo:

Retornos mensais acumulados do portfólio vs benchmark.

Análise de dispersão dos retornos diários no período.

**Análise de Ajuste Dinâmico:**

Possibilidade de sugerir ajustes nos pesos do portfólio caso algum ativo apresente desempenho muito descolado da média.

**Flexibilidade para Avaliar Outros Algoritmos:**

Deixar aberta a possibilidade de utilizar outros modelos além do EGARCH é muito importante.

Isso nos permite testar métodos como GARCH simplificado, SARIMA, ou até mesmo redes neurais LSTM, caso necessário.

Podemos introduzir uma camada de validação cruzada para comparar diferentes algoritmos e decidir qual se adapta melhor ao curto prazo.

**Portfólio Sempre Superando o Índice:**

Adicionar uma métrica que garanta, na média mensal, que o portfólio supere o benchmark é crucial para validar a eficácia da estratégia.

Essa abordagem nos ajuda a garantir que o portfólio está gerando "alpha" (retorno acima do mercado) consistentemente.

Podemos implementar uma lógica para ajustar dinamicamente os pesos caso o portfólio esteja ficando muito próximo ou abaixo do benchmark.

Gráficos com foco em tendências de curto prazo:

Retornos mensais acumulados do portfólio vs benchmark.

Análise de dispersão dos retornos diários no período.

**Análise de Ajuste Dinâmico:**

Possibilidade de sugerir ajustes nos pesos do portfólio caso algum ativo apresente desempenho muito descolado da média.

**Flexibilidade para Avaliar Outros Algoritmos:**

Deixar aberta a possibilidade de utilizar outros modelos além do EGARCH é muito importante.

Isso nos permite testar métodos como GARCH simplificado, SARIMA, ou até mesmo redes neurais LSTM, caso necessário.

Podemos introduzir uma camada de validação cruzada para comparar diferentes algoritmos e decidir qual se adapta melhor ao curto prazo.

**Portfólio Sempre Superando o Índice:**

Adicionar uma métrica que garanta, na média mensal, que o portfólio supere o benchmark é crucial para validar a eficácia da estratégia.

Essa abordagem nos ajuda a garantir que o portfólio está gerando "alpha" (retorno acima do mercado) consistentemente.

Podemos implementar uma lógica para ajustar dinamicamente os pesos caso o portfólio esteja ficando muito próximo ou abaixo do benchmark.

Compreendi agora. Estamos buscando projeções de retorno futuro com base em simulações de Monte Carlo e uma validação da capacidade do portfólio otimizado de superar uma meta média mensal de 15%. A estrutura do algoritmo será ajustada para gerar os seguintes cenários:

1. **Simulação com Monte Carlo**: Gerar cenários futuros (5%, 50%, 95%) para o portfólio nos próximos 6 meses.
2. **Validação com base no modelo de Machine Learning**: Usar os dados históricos para projetar retornos futuros e verificar se os retornos simulados atingem a meta.
3. **Visualização clara**: Apresentar as projeções em gráficos, destacando os diferentes percentis e a meta de retorno.

Vamos reformular o script para atender a esse novo objetivo:

**Estrutura do Script Reformulado**

* **Entrada**: Arquivo historical\_data\_cleaned.csv com dados históricos e portfolio\_otimizado.csv com pesos do portfólio ótimo.
* **Simulação**: Usar Monte Carlo para projetar cenários para os próximos 6 meses.
* **Machine Learning**: Validar se as projeções médias superam a meta mensal de 15%.
* **Saída**: Gráficos com os percentis e a meta, além de uma tabela com os retornos projetados.

Vou ajustar o script e garantir que todos esses pontos sejam abordados. Vou organizar o código agora.

Ótima observação! Para atender completamente à sua solicitação, vou ajustar o script para:

1. **Adicionar o portfólio real aos cenários futuros simulados**:
   * Projete o desempenho do portfólio ótimo com base no histórico recente e compare-o com as projeções de Monte Carlo.
2. **Fornecer métricas de precisão**:
   * Calcule o **Erro Médio Absoluto (MAE)** e o **Erro Quadrático Médio (RMSE)** entre as projeções do portfólio ótimo e os cenários futuros.
   * Verifique se o EGARCH é adequado como previsor para este horizonte de tempo.
3. **Exploração de possíveis melhorias**:
   * Caso o desempenho não seja satisfatório, identifique o quão necessário seria explorar outros algoritmos, como LSTM.

**Concordo plenamente com a sua visão. Os dados financeiros apresentam uma complexidade intrínseca, com padrões não lineares, eventos extremos e alta sensibilidade a fatores exógenos. Por isso, não devemos nos apegar a um único modelo ou abordagem que não esteja performando bem. Ao invés disso, devemos focar em um pipeline flexível e robusto, que permita a experimentação e adaptação.**

**Estratégia sugerida:**

1. **Ampliar o conjunto de algoritmos:**
   * **Explorar modelos alternativos:**
     + **Redes Neurais Recorrentes (RNN), como LSTM ou GRU, para capturar dependências temporais mais complexas.**
     + **Modelos baseados em gradiente, como XGBoost ou LightGBM, para detectar padrões em features derivadas de séries temporais.**
     + **SARIMA/Prophet, para avaliar padrões sazonais em conjunto com tendências.**
   * **Comparar a performance de cada modelo em relação ao Monte Carlo e benchmarks históricos.**
2. **Validação cruzada robusta:**
   * **Implementar validação temporal em blocos (time-series split), onde o conjunto de validação sempre segue cronologicamente o treinamento.**
   * **Garantir que o modelo seja testado em múltiplos períodos futuros e não apenas em um intervalo fixo.**
3. **Criar um ensemble de modelos:**
   * **Combinar previsões de múltiplos modelos para mitigar as limitações de cada abordagem individual.**
   * **Exemplo: utilizar Monte Carlo para cenários extremos, LSTM para capturar tendências não lineares e EGARCH para prever volatilidade.**
4. **Ajuste contínuo:**
   * **Incorporar feature engineering avançado, como indicadores financeiros (RSI, MACD, etc.), clusters de volatilidade e eventos externos.**
   * **Automatizar a seleção de hiperparâmetros com grid search ou Bayesian optimization.**
5. **Definir um critério de sucesso claro:**
   * **Avaliar os modelos com métricas relevantes ao contexto financeiro:**
     + **Erro absoluto médio (MAE) ajustado ao retorno percentual.**
     + **Quantas vezes o modelo supera a meta (15% anual ajustado).**
     + **Probabilidade de superar o benchmark em diferentes cenários.**

**Por que essa abordagem?**

**Os dados financeiros exigem resiliência do modelo e a capacidade de se adaptar a novos padrões ou choques de mercado. Um pipeline dinâmico e com múltiplos algoritmos permitirá:**

* **Avaliar o melhor modelo para cada contexto (curto ou longo prazo).**
* **Entender onde cada abordagem falha ou se sobressai.**
* **Criar previsões mais robustas e realistas.**

**Se você concordar, podemos:**

1. **Implementar o próximo script com LSTM para previsões de curto prazo.**
2. **Desenvolver o pipeline com validação cruzada para garantir robustez.**

**Que tal darmos esse passo?**

**Estrutura inicial para BP\_mod3\_curto\_prazo\_lstm.py**

1. **Preparação dos dados**:
   * Normalizar os retornos históricos para adequar à entrada do LSTM.
   * Criar uma janela deslizante (sliding window) para prever os próximos retornos baseados nos retornos passados.
2. **Modelo LSTM**:
   * Arquitetura inicial simples (com possibilidade de ajuste):
     + Camadas de LSTM com dropout para evitar overfitting.
     + Camada densa para saída.
   * Hiperparâmetros ajustáveis:
     + Número de épocas, tamanho da janela, taxa de aprendizado.
3. **Validação cruzada temporal**:
   * Divisão dos dados com o método **time-series split**.
   * Treinar e validar o modelo em diferentes períodos para avaliar estabilidade e generalização.
4. **Previsão e métricas**:
   * Prever os próximos 6 meses com cenários Monte Carlo (baseado nos resíduos).
   * Comparar previsões do LSTM com benchmarks e meta de retorno.
5. **Visualização**:
   * Adicionar gráficos claros:
     + Curvas dos cenários (5%, 50%, 95%).
     + Comparação com o benchmark e a meta de retorno.

**. LSTM como Foco Principal**

O **LSTM** pode ser ajustado para capturar dependências temporais de longo prazo, o que é crucial em séries temporais financeiras. Precisamos refinar os seguintes aspectos:

* **Hiperparâmetros**: Aumentar ou ajustar os timesteps, número de neurônios nas camadas, e otimizar a taxa de aprendizado.
* **Validação cruzada**: Garantir que o modelo não está apenas memorando os dados históricos, mas generalizando bem.
* **Dados adicionais**: Incorporar variáveis macroeconômicas ou indicadores de mercado, como taxas de juros, para enriquecer o conjunto de entrada.

**2. Monte Carlo para Cenários**

O Monte Carlo continuará sendo uma ferramenta importante, mas precisamos melhorar:

* **Distribuição**: Considerar distribuições mais realistas que capturem eventos extremos (como retornos em cauda longa).
* **Volatilidade Dinâmica**: Incorporar mudanças ao longo do tempo, em vez de usar valores estáticos de média e desvio padrão.

**3. Modelos Alternativos**

Se o LSTM ainda não alcançar resultados satisfatórios, podemos explorar:

* **Transformers**: Utilizar atenção para capturar padrões de longo prazo nas séries temporais.
* **Prophet**: Excelente para capturar sazonalidade e tendências em dados temporais.
* **XGBoost/LightGBM**: Testar modelos de machine learning baseados em árvores para prever retornos.

**4. Próximos Passos no Script**

1. **Foco no LSTM**: Refatorar o script para focar exclusivamente no LSTM, mantendo Monte Carlo como ferramenta de apoio para explorar cenários.
2. **Desenvolvimento Incremental**:
   * Implementar validação cruzada e otimização automática de hiperparâmetros (com ferramentas como Optuna ou GridSearchCV).
   * Adicionar métricas no dashboard que ajudem a monitorar a qualidade do modelo.
3. **Iteração Rápida**:
   * Testar com subconjuntos dos dados antes de executar em toda a série histórica.

**5. Como Seguir em Frente**

Se você concordar, posso:

1. **Remover o EGARCH** do script e implementar ajustes no LSTM com validação cruzada.
2. Melhorar o Monte Carlo, adicionando distribuições mais avançadas.
3. Refatorar o dashboard para trazer métricas e insights mais claros sobre o desempenho futuro.

**Passos a seguir no código:**

1. **Remover EGARCH completamente.**
2. **Otimizar o LSTM**:
   * Implementar validação cruzada.
   * Adicionar otimização de hiperparâmetros.
3. **Refinar o Monte Carlo**:
   * Incorporar distribuições mais realistas.
   * Adaptar os cenários às características do mercado.
4. **Aprimorar o Dashboard**:
   * Visualizar resultados do LSTM e Monte Carlo lado a lado.
   * Adicionar métricas como precisão, erro médio absoluto e erro médio quadrático.

**Correções e Melhorias**

**1. O Problema da Linha Reta no Previsto**

* Isso indica **problemas com o modelo de regressão ou engenharia de features**. A linha reta geralmente aparece quando o modelo não está conseguindo capturar a variabilidade dos dados.
* Precisamos revisar:
  + **Features**: As entradas do modelo estão suficientemente explicativas? Por exemplo, indicadores como volatilidade, médias móveis e correlações históricas devem ser incluídos.
  + **Modelo**: Talvez a Regressão Linear seja inadequada ou mal ajustada. Podemos experimentar XGBoost ou Regressão Ridge para maior flexibilidade e captura de padrões.

**2. Retorno do Monte Carlo**

* Como discutido, MC é uma **ferramenta de balizamento** para avaliar as previsões do modelo preditivo.
* **MC não deve ser removido.** Ele servirá para:
  + Comparar cenários (5%, 50%, 95%) com as previsões do modelo.
  + Avaliar se o portfólio se comporta dentro de expectativas realistas.

**Novo Pipeline Reestruturado**

Aqui está uma abordagem revisada:

**1. Pré-Processamento**

* **Selecionar Features**:
  + Retornos ajustados (diferenciais diários).
  + Volatilidade histórica (desvio padrão móvel).
  + Indicadores técnicos (média móvel de 20 e 50 períodos, RSI, etc.).
  + Correlação histórica com o benchmark (BOVESPA).
* **Criar Conjuntos de Treino e Teste**:
  + Dividir os dados históricos (12 anos) em janelas temporais.
  + Validar em **janelas consecutivas (validação temporal)** para evitar vieses de data.

**2. Modelagem**

* **Monte Carlo como Cenário Base**:
  + Gerar cenários de MC para servir como balisadores.
  + Avaliar as previsões do modelo para verificar alinhamento com os cenários (ex.: 50% mediana, ou fora do intervalo 5%-95%).
* **Modelo de Regressão Melhorado**:
  + Substituir a Regressão Linear por:
    - **XGBoost**: Para capturar padrões não-lineares.
    - **Ridge Regression**: Para evitar overfitting e melhorar generalização.
* **Paralelização**:
  + Paralelizar o ajuste de múltiplos modelos com diferentes parâmetros (usar joblib).

**3. Validação**

* **Validação Temporal**:
  + Avaliar o modelo em janelas consecutivas (ex.: validar 1 ano de dados a partir dos 12 anos).
  + Comparar previsões em janelas temporais passadas antes de extrapolar para os próximos 6 meses.
* **Comparação com Monte Carlo**:
  + Identificar em qual cenário (5%, 50%, 95%) o portfólio mais se alinha.

**4. Visualização**

* **Gráficos Dinâmicos**:
  + Previsões do modelo (no conjunto de teste) vs. Observado.
  + Previsões para os próximos 6 meses vs. MC (cenários 5%, 50%, 95%).

## 26/11/24 - Agora sai ou não ...

**Estratégia do Novo Script**

1. **Treinamento com Todo o Histórico**:
   * O LSTM será configurado para aproveitar toda a janela histórica disponível (12 anos).
   * Isso permitirá capturar ciclos de longo prazo e sazonalidades.
2. **Monte Carlo Reestruturado**:
   * Gerar múltiplos cenários futuros com base nos padrões históricos reais.
   * Alimentar esses cenários no pipeline de previsão para enriquecer as projeções.
3. **Previsão em Etapas**:
   * Previsões de 30 dias serão acumuladas para alcançar 180 dias.
   * Isso evita que o modelo extrapole diretamente padrões não claros.
4. **Feature Engineering**:
   * Adicionar variáveis auxiliares como médias móveis, volatilidade histórica e tendências para melhorar o aprendizado.
5. **Integração Final**:
   * Um dashboard claro para visualização de cenários futuros e métricas reais.