

1. Visão Geral do Projeto

Este projeto tem como objetivo a previsão do IBC-Br (Índice de Atividade Econômica do Banco Central), amplamente utilizado pelo mercado financeiro e acadêmico como uma proxy mensal para o Produto Interno Bruto (PIB) brasileiro. Dada a divulgação defasada do PIB oficial (trimestral), o IBC-Br serve como um termômetro ágil da atividade econômica.

A abordagem adotada utiliza técnicas de Machine Learning supervisionado, focando na construção de um Stacking Ensemble. Esta estratégia combina as previsões de múltiplos modelos heterogêneos para reduzir a variância do erro e aumentar a generalização das previsões em dados fora da amostra (out-of-sample). A aplicação desta metodologia foi inspirada no estudo de Micheletti (2024), que investigou a viabilidade do Stacking para a previsão do PIB brasileiro frente a modelos base como Random Forest e Gradient Boosting.

2. Engenharia e Processamento de Dados

2.1. Fonte e Tratamento Inicial

Os dados brutos foram carregados do arquivo `dados_macro_bruto.xlsx`. O pipeline de pré-processamento, implementado no módulo `pipeline.py`, realiza a limpeza inicial e o preenchimento de dados faltantes para garantir a continuidade das séries temporais.

2.2. Feature Engineering

Para capturar a complexidade da economia brasileira, foram criadas variáveis baseadas em teoria econômica:

- **Juro Real:** Diferença entre a taxa Selic e a inflação (IPCA), medindo o custo real do capital.
- **Termos de Troca:** Relação entre preços de exportação e importação (ou taxa de câmbio), indicando a riqueza relativa do país.
- **Hiatos do Produto:** Diferença entre o IBC-Br atual e suas tendências de longo prazo (médias móveis de 12 e 24 meses), indicando se a economia está aquecida ou ociosa.
- **Volatilidade Financeira:** Desvio padrão móvel do Dólar e do Ibovespa (janelas de 3 e 6 meses) para mensurar incerteza.
- **Defasagens (Lags):** Variáveis defasadas em 3 e 12 meses para capturar a inércia

e ciclos sazonais.

2.3. Transformação e Estacionariedade

Modelos de regressão requerem séries estacionárias (média e variância constantes no tempo). O sistema aplicou:

- **Diferenciação (Diff):** Para taxas (ex: Selic, Juro Real).
- **Log-retorno (Pct Change):** Para índices de preços e atividade.
- **Preservação de Nível:** Variáveis como "Hiatos" e "Balança de Empregos" foram mantidas em nível conforme a teoria econômica sugere estacionariedade ou relevância estrutural.

2.4. Seleção de Features e Teste de Estacionariedade (ADF)

A seleção de variáveis foi conduzida de forma automatizada pelo módulo `feature_selection.py`, utilizando o Teste Augmented Dickey-Fuller como critério rigoroso de filtragem.

Fundamentação Teórica:

O ADF é um teste de hipótese estatística fundamental em econometria para verificar a presença de uma raiz unitária em uma série temporal. A presença de raiz unitária indica que a série é não-estacionária, ou seja, suas propriedades estatísticas (como média e variância) mudam com o tempo, o que pode levar a modelos instáveis.

- **Hipótese Nula:** A série possui raiz unitária.
- **Critério de Decisão:** Se o p-valor calculado for menor que o nível de significância ($\alpha = 0.05$), rejeita-se a hipótese nula. Isso confirma estatisticamente que a série é estacionária e segura para uso em modelos de regressão.

Aplicação no Pipeline:

O objetivo desta etapa foi prevenir o fenômeno de regressão espúria, onde o modelo detecta falsas correlações fortes entre variáveis apenas porque ambas possuem tendências de crescimento (ex: PIB e população), sem que haja causalidade real.

- **Entrada:** O algoritmo avaliou as 100 features geradas na etapa de engenharia.
- **Filtragem:** 15 variáveis falharam no teste ($p\text{-valor} > 0.05$) e foram descartadas. Entre as removidas estavam agregados monetários em nível (M1, M1_Real) e tendências de longo prazo do IBC, que, por definição, não orbitam uma média

fixa.

- **Resultado Final:** Foram mantidas 85 variáveis explicativas comprovadamente estacionárias para o treinamento dos modelos.

3. Metodologia de Modelagem

3.1. Modelos Base

A implementação dos modelos foi realizada utilizando as bibliotecas Scikit-Learn e XGBoost, com reproduzibilidade garantida pela fixação da semente aleatória (`random_state=42`). A construção de cada modelo envolveu uma estratégia de otimização de hiperparâmetros via `GridSearchCV` acoplada a uma validação cruzada temporal (`TimeSeriesSplit`), garantindo que o ajuste respeitasse a ordem cronológica dos dados.

O pipeline incluiu uma lógica adaptativa: para janelas de treino com amostras suficientes, executou-se a busca em grade (grid search). Para janelas iniciais menores, o sistema adotou automaticamente um conjunto de parâmetros conservadores pré-definidos para evitar o overfitting em séries curtas.

As configurações específicas exploradas foram:

- **Random Forest Regressor:** Otimização do número de árvores (`n_estimators`: 150, 200) e controle rigoroso de profundidade (`max_depth`: 5, 7) e folhas (`min_samples_leaf`: 2, 4). Utilizou-se a raiz quadrada das features (`max_features='sqrt'`) em cada divisão para decorrelacionar as árvores.
- **Gradient Boosting Regressor:** Foco em taxas de aprendizado lentas (`learning_rate`: 0.01, 0.05) combinadas com subamostragem estocástica (`subsample`: 0.7) e árvores rasas (`max_depth`: 2, 3) para maximizar a generalização.
- **ElasticNet:** A regularização foi ajustada explorando diferentes penalidades (`alpha`: 0.01 a 1.0) e taxas de mistura (`l1_ratio`: 0.2, 0.5, 0.8), permitindo ao modelo transitar entre o comportamento Lasso (seleção de variáveis) e Ridge (redução de coeficientes).
- **XGBoost (XGB):** Implementado com paralelismo (`n_jobs=-1`) e regularização adicional via amostragem de colunas por árvore (`colsample_bytree`: 0.7), além de restrições de profundidade (`max_depth`: 3, 5) similares ao Gradient Boosting padrão.

3.2. Estratégia de Validação e Stacking

A estrutura de treinamento foi desenhada para replicar estritamente o cenário real de previsão econômica, onde apenas dados do passado estão disponíveis no momento da inferência, eliminando qualquer risco de vazamento de dados (data leakage).

Protocolo de Janela Expansiva (Expanding Window):

Em vez de uma divisão estática de treino/teste, o pipeline implementou um loop iterativo de validação temporal:

1. **Inicialização:** O treinamento começa com uma janela mínima de 48 meses (4 anos) de dados históricos.
2. **Ciclo de Previsão:** Para cada mês subsequente, o modelo é treinado utilizando todos os dados disponíveis desde o início da série até o mês imediatamente anterior, sendo então solicitado a prever apenas o valor do mês seguinte.
3. **Padronização Isolada:** Em cada iteração, o StandardScaler é ajustado apenas nos dados de treino daquela janela específica, garantindo que a média e o desvio padrão do futuro não contaminem o passado.
4. **Expansão:** Após a previsão, o dado real do mês é incorporado ao conjunto de treino, e o ciclo se repete até o fim da série.

Construção do Meta-Learner via NNLS:

Para combinar as previsões dos quatro modelos base (RF, GB, EN, XGB), adotou-se uma abordagem de Blending otimizada via Non-Negative Least Squares (NNLS).

- **Otimização Convexa:** Diferente de uma regressão linear comum, o NNLS restringe os pesos para serem estritamente não-negativos.
- **Interpretação Econômica:** Um modelo só recebe peso se contribuir positivamente para a redução do erro. Modelos redundantes ou com baixa performance têm seus pesos zerados matematicamente, atuando como uma seleção automática de modelos.
- **Normalização:** Os pesos resultantes foram normalizados para que sua soma seja igual a 1, garantindo que a previsão final permaneça na mesma escala das previsões individuais, preservando a coerência do índice IBC-Br projetado.

4. Análise dos Resultados

Os modelos foram avaliados tanto na previsão das variações mensais quanto na reconstrução do nível do índice.

4.1. Métricas de Performance

A avaliação dos modelos foi conduzida em duas etapas distintas: a capacidade de prever a variação mensal (log-retorno) e a precisão na reconstrução do nível do índice IBC-Br, que é a variável de interesse econômico final. As métricas utilizadas foram o RMSE (Raiz do Erro Quadrático Médio), que penaliza grandes desvios, o MAE (Erro Absoluto Médio) e o MAPE (Erro Percentual Absoluto Médio).

Performance em Variações: Nesta etapa, avalia-se a capacidade do modelo de capturar a volatilidade de curto prazo.

- **Dominância Não-Linear:** O XGBoost (RMSE 0.0250) e o Ensemble (RMSE 0.0256) apresentaram os menores erros, demonstrando superioridade técnica sobre modelos mais simples.
- **Limitação Linear:** O modelo ElasticNet teve o pior desempenho (RMSE 0.0308), indicando que relações lineares são insuficientes para explicar a dinâmica complexa da economia brasileira. A diferença significativa entre o ElasticNet e os modelos de boosting confirma a existência de não-linearidades cruciais nos dados macroeconômicos.

Performance em Níveis: Ao reintegrar as variações para obter o valor nominal do IBC-Br, observamos o impacto real do erro acumulado ao longo de 191 meses de teste.

- **Precisão Absoluta (MAE):** O Ensemble apresentou um Erro Absoluto Médio de 1.84 pontos. Considerando que o IBC-Br oscila frequentemente acima de 140 pontos, um erro médio inferior a 2 pontos indica alta aderência à série real.
- **Erro Relativo (MAPE):** Em termos percentuais, o Ensemble (1.86%) e o XGBoost (1.84%) mantiveram o erro abaixo da barreira de 2%. Isso significa que, na grande maioria dos meses, as previsões capturaram a tendência correta da atividade econômica, com desvios marginais que não alteram a interpretação do cenário macroeconômico.

4.2. Teste de Superioridade Estatística

Para validar se a redução do erro observada era estatisticamente relevante ou apenas fruto do acaso, utilizou-se o Teste de Diebold-Marian, implementado no módulo

`statistical_tests.py`.

Metodologia:

O teste DM compara duas séries de erros de previsão para determinar se elas possuem a mesma acurácia preditiva.

- **Hipótese Nula:** A diferença de precisão entre os dois modelos é zero (empate estatístico).
- **Interpretação:** Um p-valor baixo (tipicamente < 0.05) indica evidência suficiente para rejeitar a hipótese nula, confirmando que um modelo é estatisticamente superior ao outro.

Análise dos Resultados:

1. Ensemble vs Random Forest (P-valor: 0.0004):

O valor extremamente baixo permite rejeitar a hipótese nula com 99,9% de confiança. Isso prova que o ganho de performance do Ensemble sobre o Random Forest não é aleatório; o Stacking corrige sistematicamente os erros do RF.

2. Ensemble vs Gradient Boosting (P-valor: 0.4552):

O p-valor alto indica que não se pode rejeitar a hipótese nula. Estatisticamente, o Ensemble e o Gradient Boosting (GB) estão empatados.

Impacto na Conclusão:

Este resultado influencia a decisão final de modelagem ao demonstrar que o Gradient Boosting sozinho já captura a quase totalidade do sinal preditivo disponível nos dados. O Ensemble atua aqui como um mecanismo de redução de risco: ele garante que a performance nunca seja inferior ao melhor modelo individual, mas, neste cenário específico, o custo computacional extra do Stacking traz ganhos marginais em relação ao GB puro.

5. Conclusão

O projeto demonstrou que a utilização de variáveis macroeconômicas defasadas, combinada com algoritmos de Boosting e técnicas de Ensemble, produz previsões altamente precisas para o IBC-Br. O modelo final atingiu um MAPE de 1.86%, situando-se na faixa de alta precisão (erros inferiores a 2%) para previsões econômicas.

Comparativo com a Literatura (Micheletti, 2024): Ao contrastar estes resultados com o estudo de Micheletti (2024), que aplicou Stacking para previsão do PIB brasileiro, observam-se divergências importantes nas conclusões sobre a utilidade do Ensemble:

1. **Eficácia do Stacking:** Enquanto Micheletti (2024) concluiu que o Stacking não justificava sua complexidade por não superar significativamente os modelos base, este projeto encontrou evidência estatística (teste Diebold-Mariano, p-valor 0.0004) de que o Ensemble é superior ao Random Forest. Embora haja um empate técnico com o Gradient Boosting (similar ao achado do paper), o Ensemble aqui atuou como um redutor de variância eficaz.
2. **Diferencial Metodológico:** A divergência de sucesso pode ser atribuída à escolha do meta-learner. Micheletti utilizou uma Regressão Linear simples para combinar os modelos, o que pode permitir pesos negativos e overfitting. Este projeto utilizou NNLS (Non-Negative Least Squares), que força pesos positivos e zera a contribuição de modelos ineficientes (como o ElasticNet, que teve baixa performance), garantindo que o Ensemble nunca seja pior que seus componentes individuais.
3. **Consistência das Métricas:** O MAPE de 1.86% obtido neste trabalho é altamente competitivo, alinhando-se com os melhores resultados de curto prazo (1 a 3 meses) apresentados por Micheletti (1.43% a 3.25%).

Em suma, a etapa rigorosa de seleção de features via teste ADF (para garantir estacionariedade) e o uso de um meta-learner restritivo (NNLS) permitiram que o Stacking Ensemble deste projeto superasse as limitações observadas na literatura recente, consolidando-se como uma ferramenta robusta para mitigar o risco de modelo na previsão econômica.