Relatório TP2 ML

Diogo Oliveira Neiss - 2021421915

Introdução

O Trabalho Prático 2 de Aprendizado de Máquina teve como objetivo desenvolver um algoritmo de boosting, especificamente o Adaboost, para classificação binária. Utilizou-se a base de dados Tic-Tac-Toe para os testes, e os modelos criados foram analisados por meio de validação cruzada com 5 partições.

Implementação

A linguagem escolhida para o desenvolvimento do trabalho foi Python, devido à sua vasta gama de bibliotecas úteis para ciência de dados e aprendizado de máquina. O algoritmo AdaBoost foi feito com as bibliotecas NumPy, Pandas e Scikit-Learn (ou SkLearn).

Para organizar o ambiente de desenvolvimento, que envolvia vários pacotes diferentes, utilizou-se um ambiente virtual python. Os pacotes utilizados estão disponíveis no arquivo `requirements.txt` e instruções de uso estão descritas no arquivo `README.md`

Arquivos e pastas

O arquivo principal está na pasta `/notebooks`, com o nome `main.ipynb`. O `sklearn.ipynb` é um _notebook_ de testes e experimentos com a biblioteca _scikit-learn_ para criação dos benchmarks.A pasta `images` será criada após execução do notebook principal

Classificador

A implementação do classificador AdaBoost seguiu o padrão da biblioteca Scikit-Learn, permitindo armazenar o classificador e suas funções necessárias em um objeto, de forma que a classe criada possua um construtor e métodos de treinamento e predição.

A implementação foi projetada especificamente para classificação binária, sendo incompatível com outros formatos de dados ou tarefas.

Benchmark

Foi utilizada uma implementação existente do algoritmo para comparação de resultados, desejando que o algoritmo estivesse sempre em uma margem de tolerância das métricas encontradas no benchmark. Para isso, foi escolhida a implementação da biblioteca Scikit-Learn para validação posterior.

Validação cruzada

A implementação da validação cruzada seguiu o conteúdo teórico ensinado nas aulas e utilizou a implementação da biblioteca Scikit-Learn.

Utilizou-se o módulo `KFold` para realizar as separações para a validação cruzada em 5 folds. Além disso, as funções de cálculo de métricas, como acurácia e F1-score, também foram retiradas dessa biblioteca.

Análise dos resultados

A análise do modelo de AdaBoost foi feita com validação cruzada com 5 partições e variação do número de estimadores, calculando os resultados das métricas de análise para cada partição e calculando a média final para gerar o valor final do modelo.

Foram utilizadas as seguintes métricas base

1. Erro
2. Acurácia
3. F1 Score

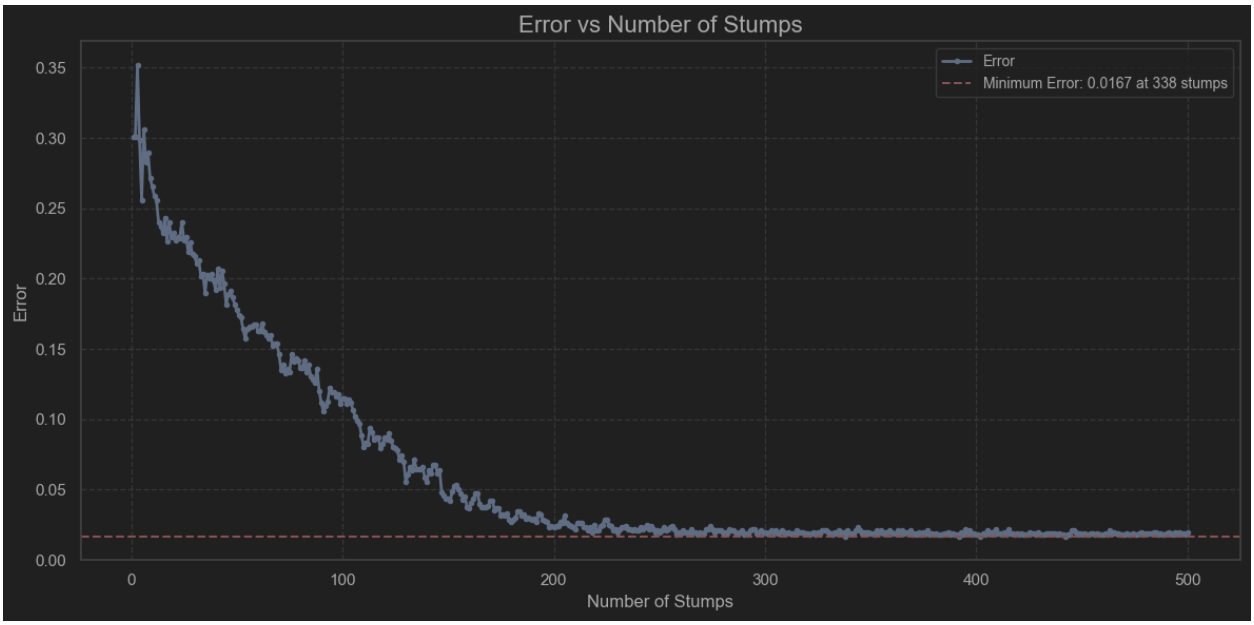
Para compreender como a variação do número de estimadores impacta o algoritmo foi feita a variação do números de estimadores máximos, em 50, 100, 250 e 500 para todas as métricas. O código foi configurado para salvar os plots na pasta `/images`.

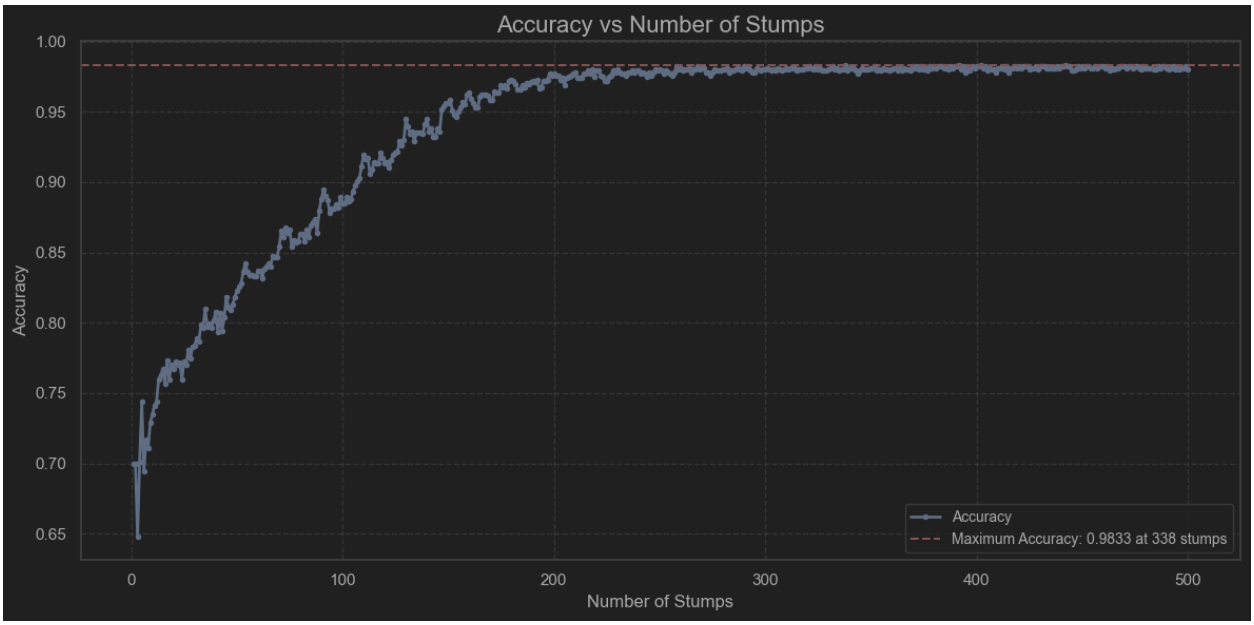
Foram feitos **gráficos dos estimadores**, mostrando o erro e peso por stump, **gráficos expositivos**, mostrando o valor das 3 métricas com o número de stumps e também um plot conjunto, **gráficos de agregação**, com janelas móveis de tamanho variável para média (tamanho 1 usado como base), **gráficos de variação**, comparando a mudança entre cada valor de acordo com as janelas móveis, **gráficos de benchmark,** comparando com a implementação pronta de uma biblioteca.

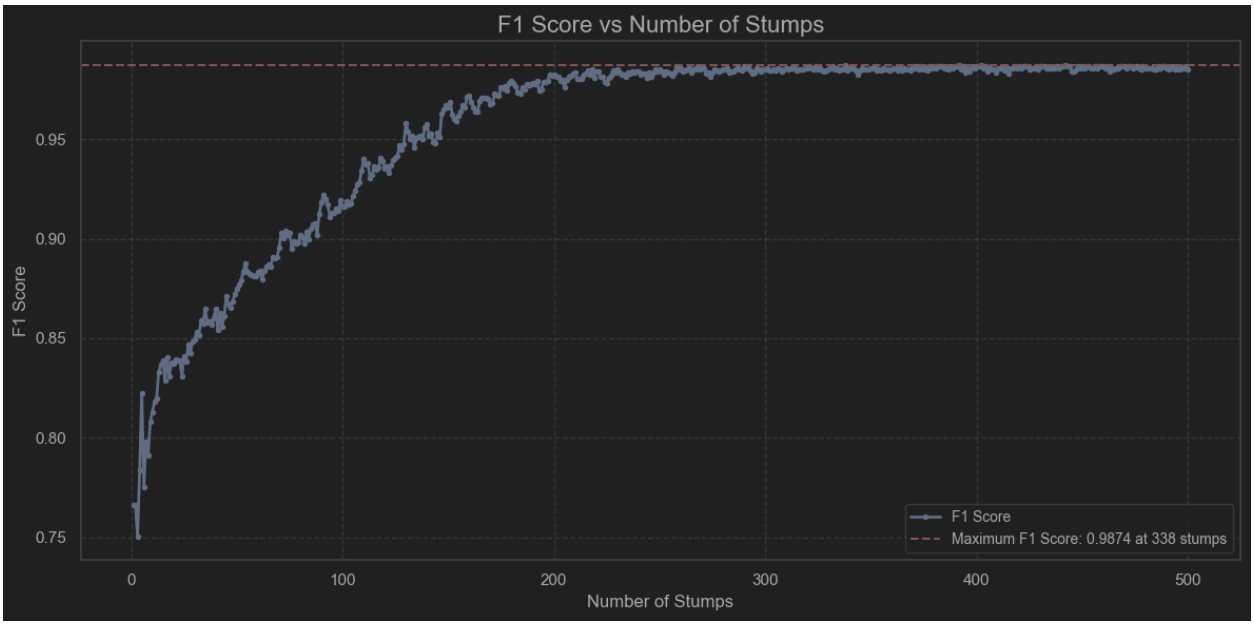
Colocaremos todos os gráficos e análises apenas para 500, os demais estão na pasta de imagens e servem apenas para comparação em um intervalo menor, porém a conclusão é a mesma.

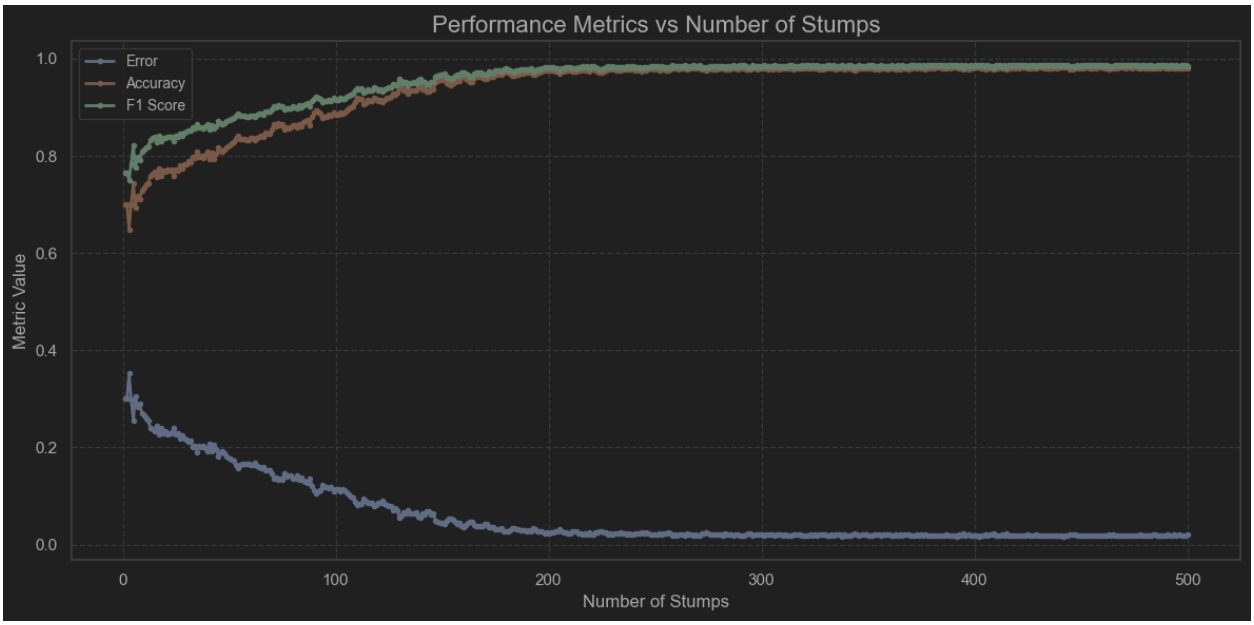
500 estimadores máximos

Nos quatro gráficos abaixo, pode-se ver a evolução do erro, f1 e acurácia do modelo conforme o aumento do número de estimadores utilizados. O erro é definido como um menos o valor da acurácia



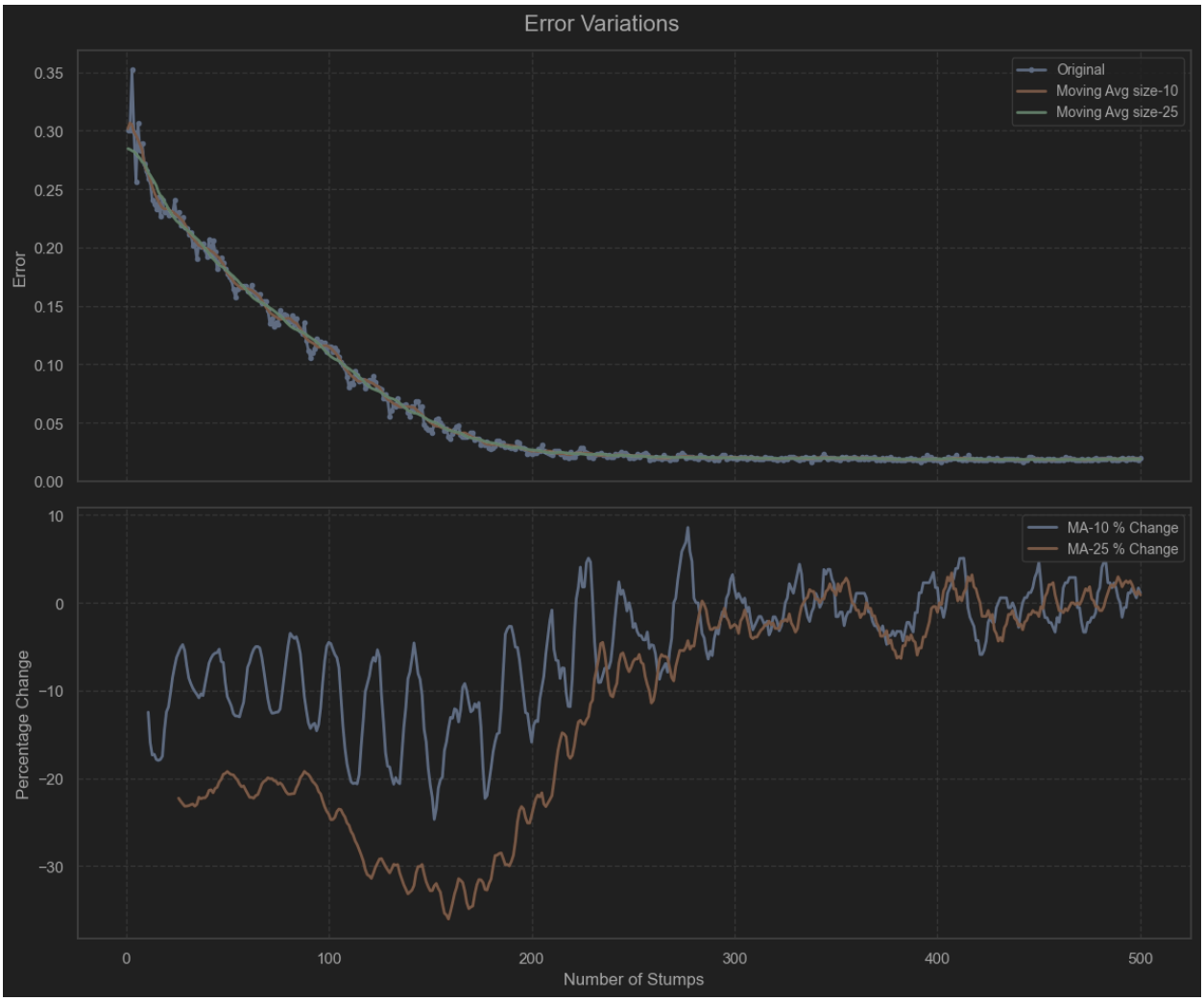

Accuracy vs Number of Stumps
Accuracy
Number of Stumps
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0
100
200
300
400
500
Accuracy
Maximum Accuracy: 0.9833 at 338 stumps


F1 Score vs Number of Stumps
F1 Score
Number of Stumps
0.95
0.90
0.85
0.80
0.75
0
100
200
300
400
500
F1 Score
Maximum F1 Score: 0.9874 at 338 stumps

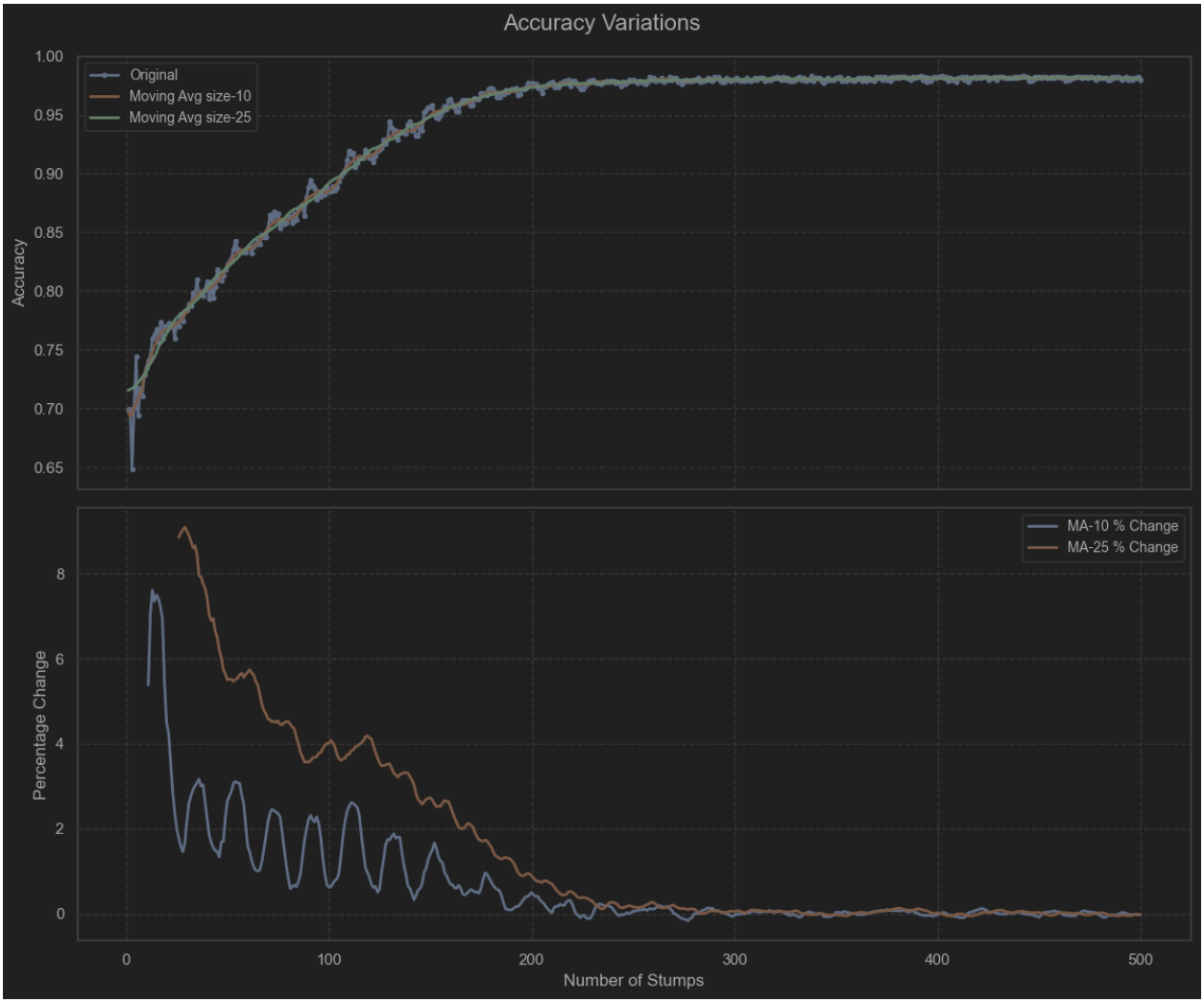


Observa-se que com o aumento do número de estimadores o erro diminui e acurácia/f1 aumentam como esperado, porém o ganho de aumento no número de stumps diminui depois de 200 iterações, com o modelo praticamente convergido depois de 300.

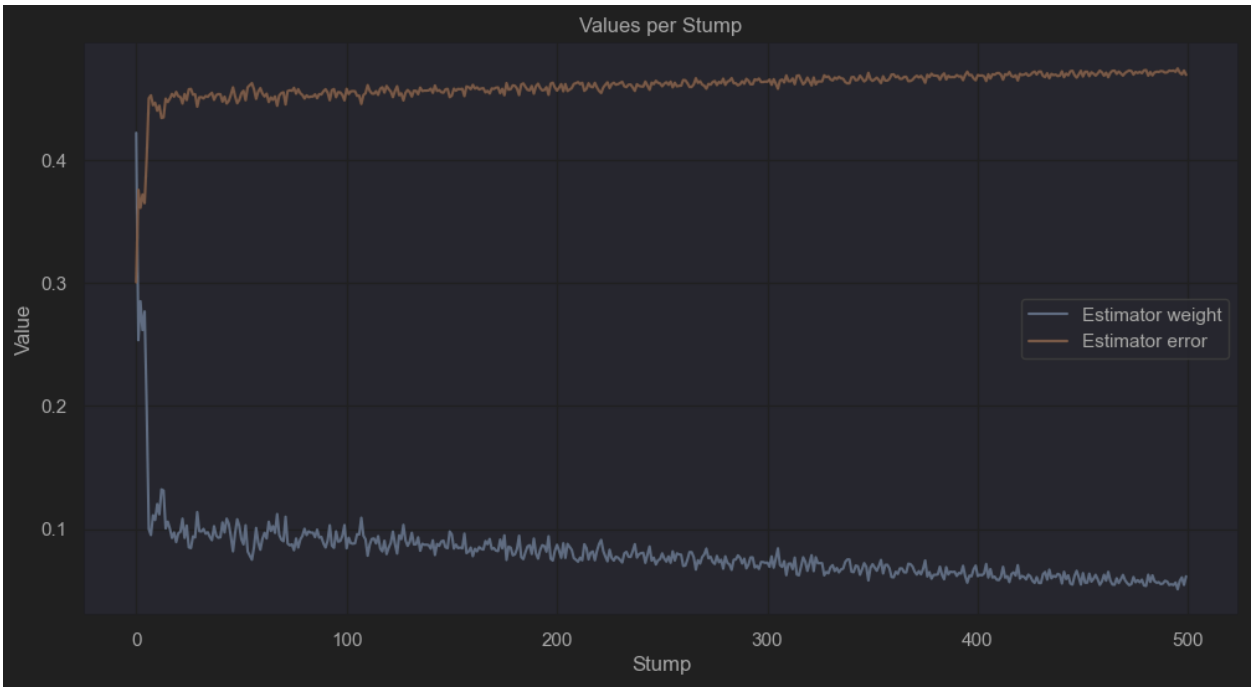
Isso é explicado pela redução no viés no modelo, permitindo aumento na generalização e que o erro em dados ainda não vistos seja menor. O melhor valor encontrado para a semente aleatória utilizada por 338 stumps.



Como observado anteriormente, o erro para de cair (possui variação positiva) após ~350 stumps e começa a oscilar, indicando um limite prático para o dataset. A visualização com média móvel de 25 permite caputar bem a tendência do modelo.

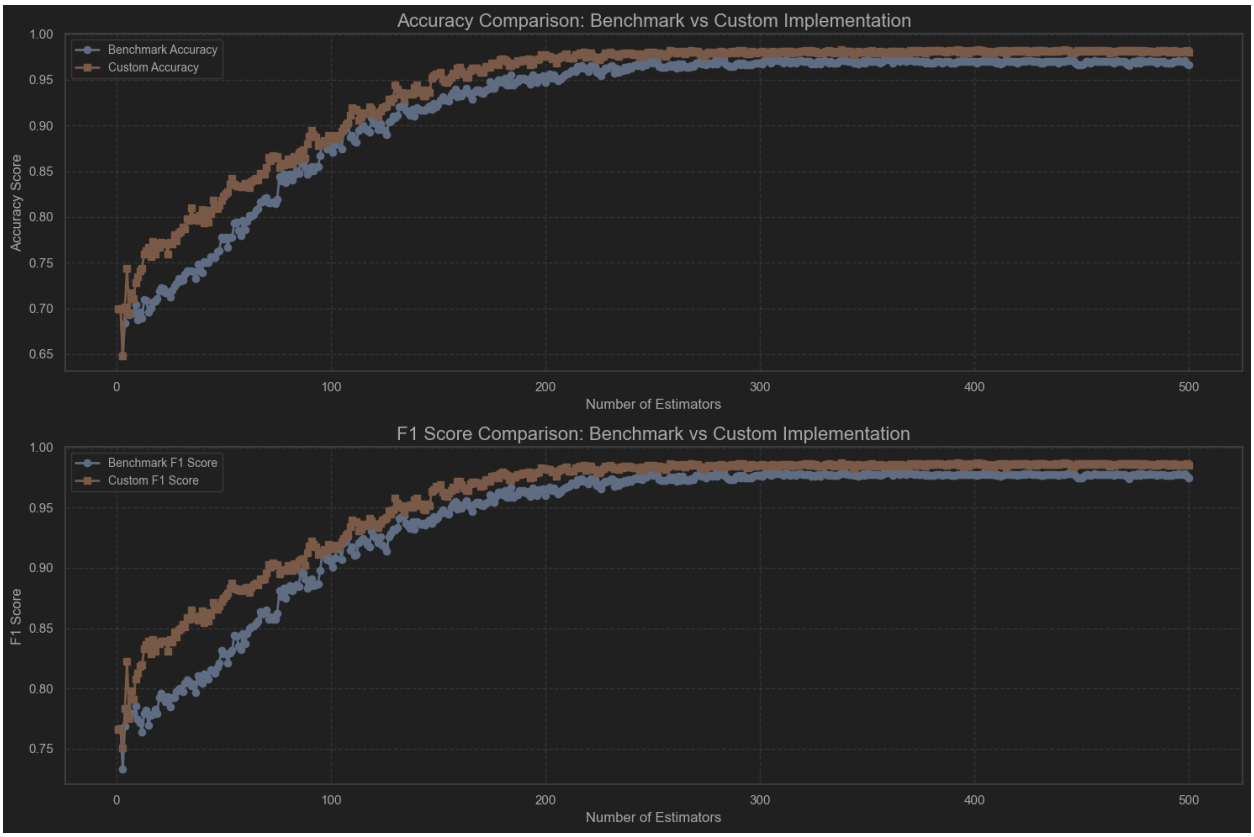

Accuracy Variations
Original
Moving Avg size-10
Moving Avg size-25
Accuracy
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
MA-10 % Change
MA-25 % Change
Percentage Change
8
6
4
2
0
0
100
200
300
400
500
Number of Stumps

É clara a convergência do modelo depois de 300, com a variação percentual da média móvel oscilando em 0.



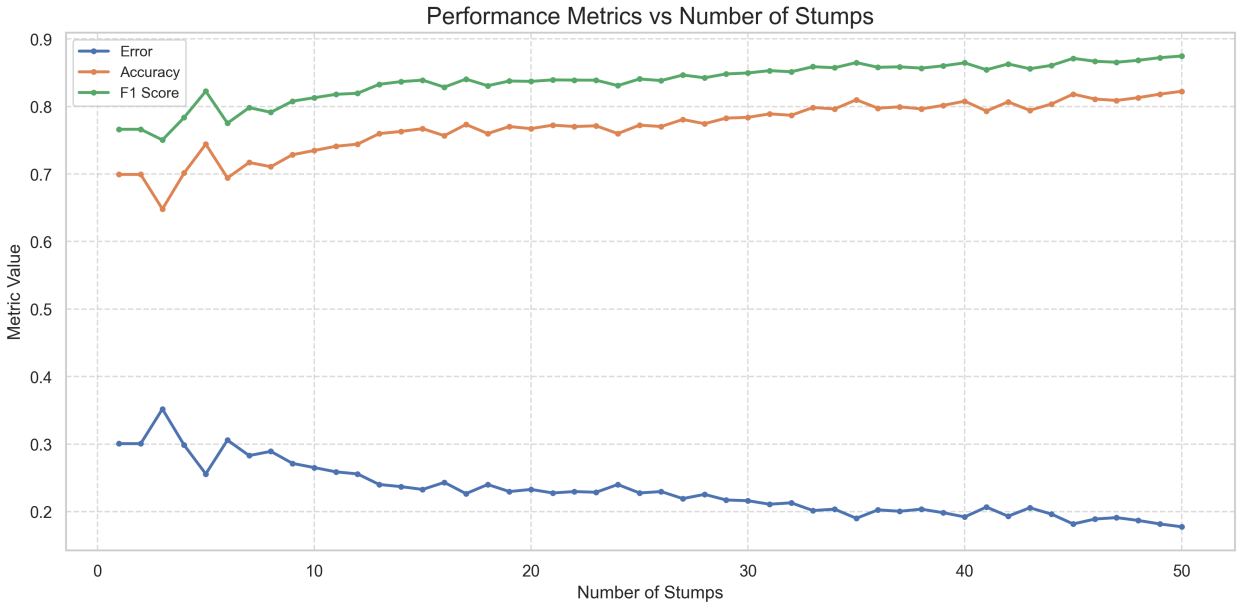
No gráfico acima podemos observar o que acontece a cada adição de um stump: os pesos dos estimadores diminuem conforme aumenta o número de estimadores, isto é, novos estimadores causam pouco impacto no erro de classificação, já que seus pesos são muito pequenos.

Portanto, após certo ponto, o aumento do número de estimadores não necessariamente torna o modelo mais eficaz e como foi comentado nos gráficos anteriores, pode até piorar o modelo.



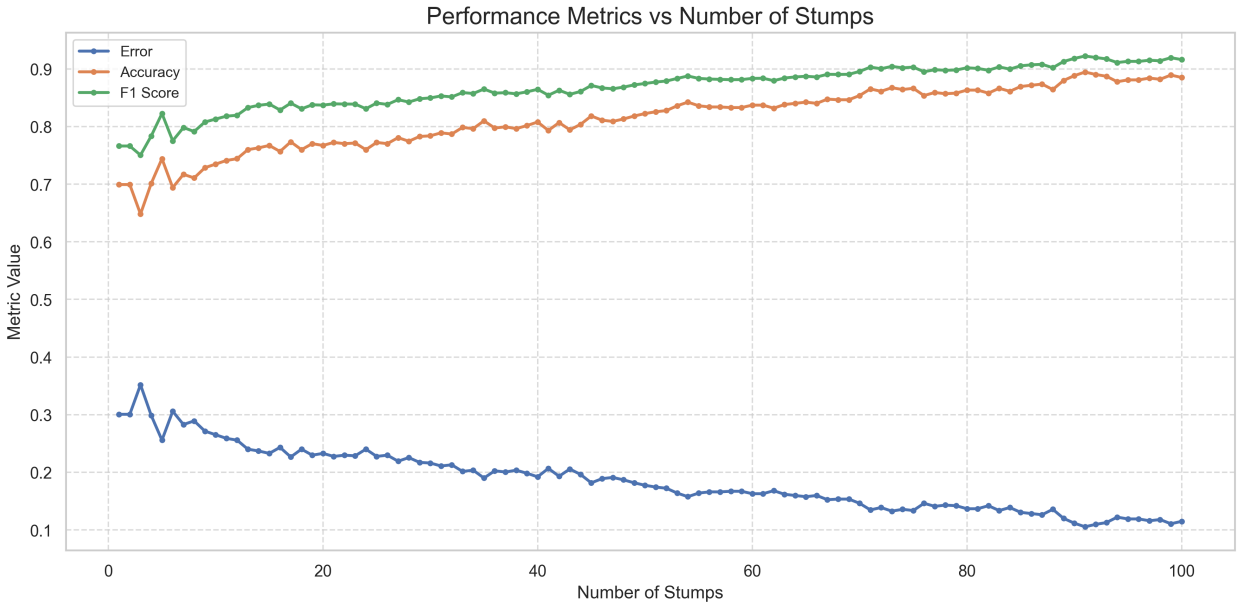
Por fim, vamos analisar a implementação customizada com a do sk-learn, denominado como benchmark. Os modelos estiveram muito próximos depois de 100 iterações e convergiram para números parecidos. A hipótese é de que a diferença se dê ao azar de uma semente aleatória infeliz, o jeito que a validação cruzada foi computada e alguma forma de regularização implementada na biblioteca.

50


Performance Metrics vs Number of Stumps
Error
Accuracy
F1 Score
Metric Value
Number of Stumps
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
10
20
30
40
50

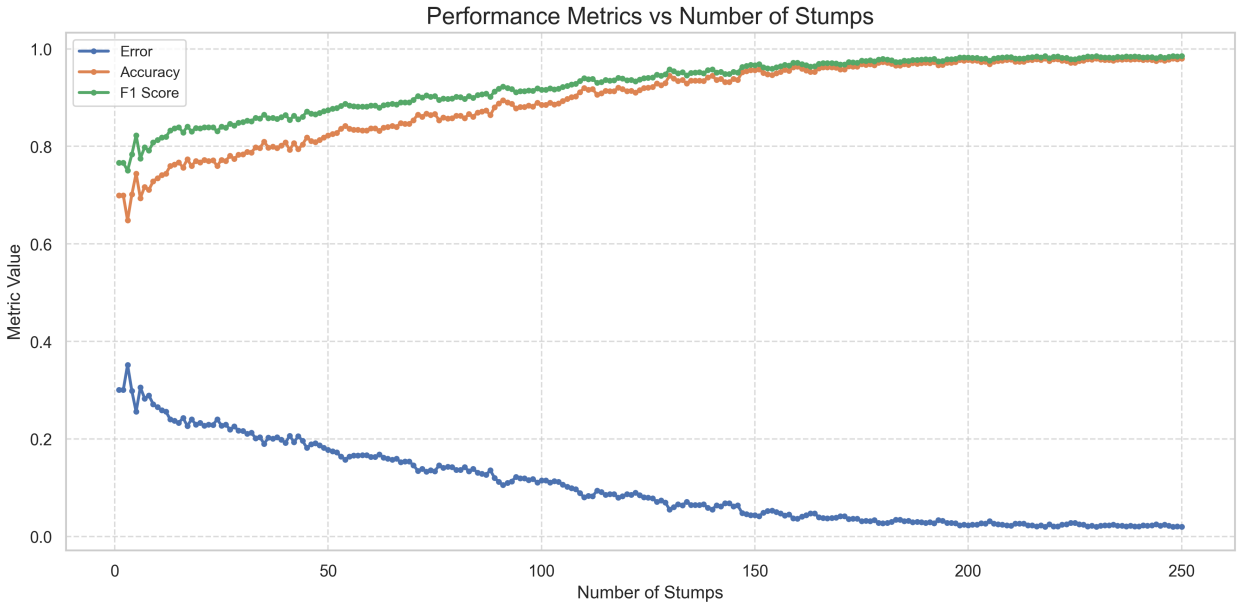

Values per Stump
Estimator weight
Estimator error
Value
Stump
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0
10
20
30
40
50

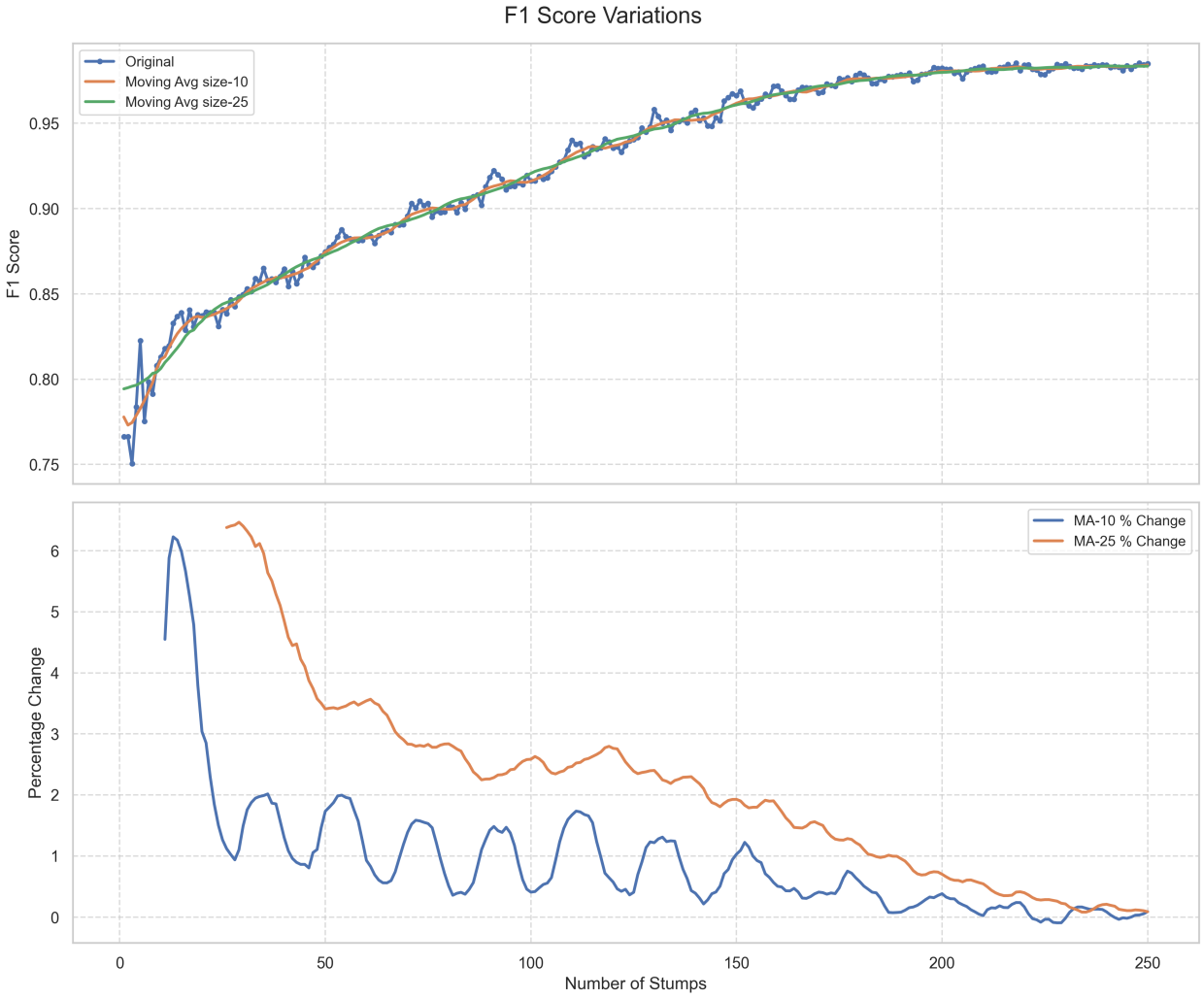
100

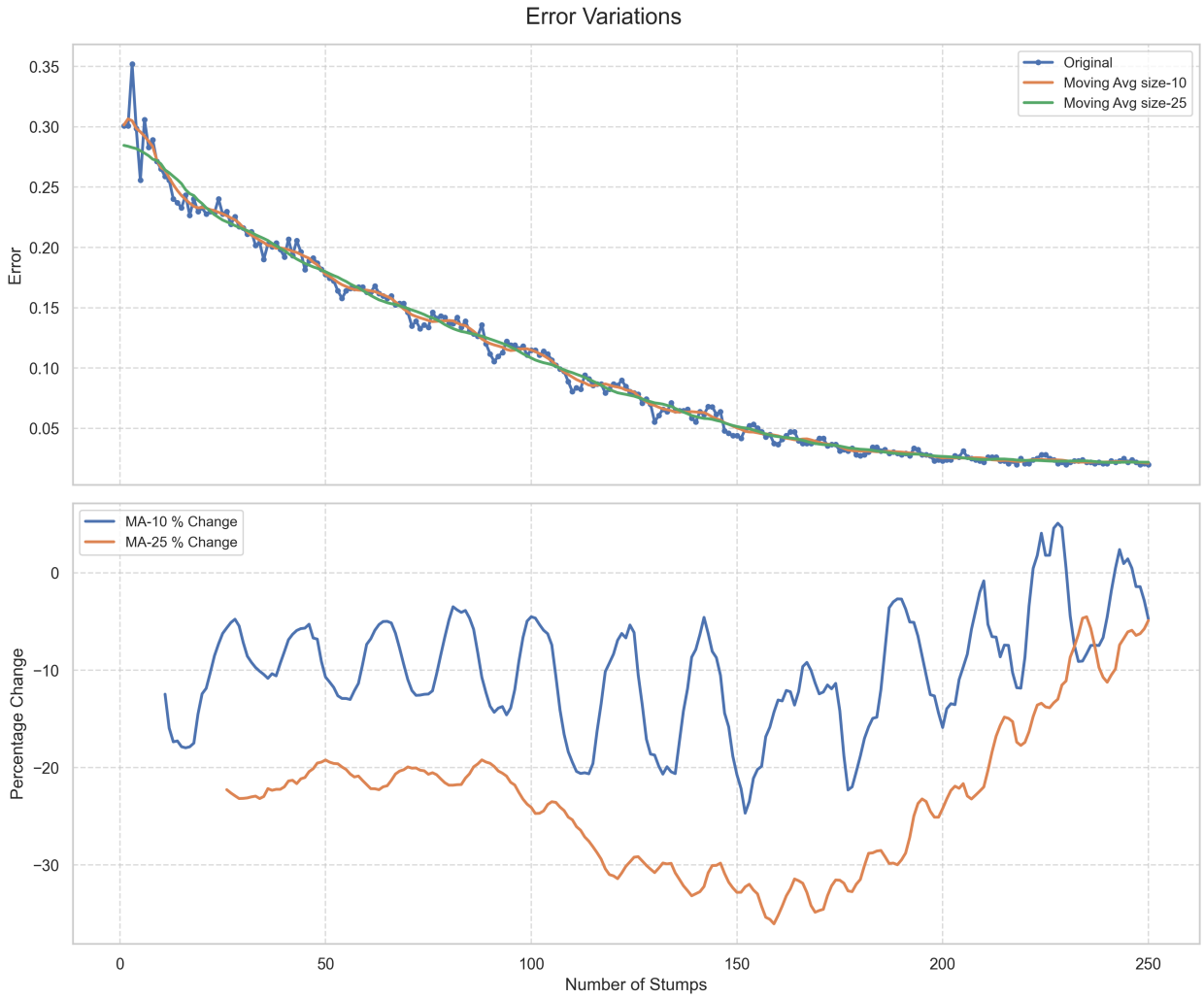

Performance Metrics vs Number of Stumps
Error
Accuracy
F1 Score
Metric Value
Number of Stumps
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
20
40
60
80
100

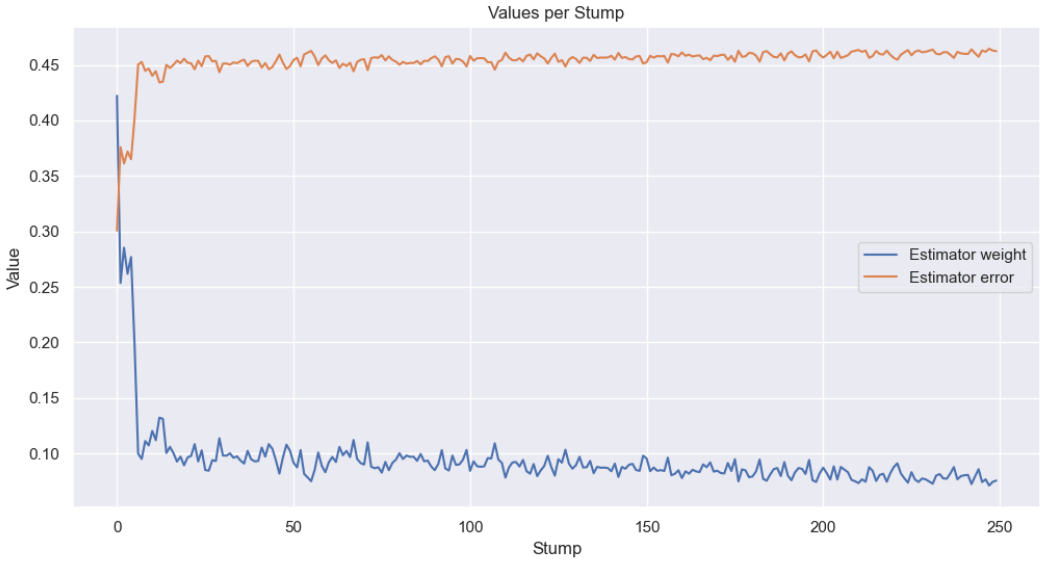

Values per Stump
Estimator weight
Estimator error
Value
Stump
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0
20
40
60
80
100

250


Performance Metrics vs Number of Stumps
Error
Accuracy
F1 Score
Metric Value
Number of Stumps


F1 Score Variations
Original
Moving Avg size-10
Moving Avg size-25
F1 Score
MA-10 % Change
MA-25 % Change
Percentage Change
Number of Stumps


Error Variations
Original
Moving Avg size-10
Moving Avg size-25
Error
0.35
0.30
0.25
0.20
0.15
0.10
0.05
MA-10 % Change
MA-25 % Change
Percentage Change
0
−10
−20
−30
0
50
100
150
200
250
Number of Stumps


Values per Stump
Estimator weight
Estimator error
Value
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0
50
100
150
200
250
Stump

Conclusão

O processo de boosting foi demonstrado como uma metodologia extremamente eficaz para reduzir o viés em modelos simples através das análises e discussões apresentadas neste relatório. Verificou-se que, à medida que o número de estimadores aumenta, o erro de classificação diminui, aprimorando a capacidade de generalização do modelo. No entanto, os benefícios do boosting têm seus limites. Especificamente neste estudo, o modelo convergiu por volta de 338 estimadores, de um total de 500 testados, com a adição de novos estimadores não proporcionando ganhos significativos de desempenho para justificar o tempo superior de execução

Essa convergência sugere que, embora o boosting seja uma ferramenta poderosa, seu uso deve ser planejado de forma inteligente para maximizar os benefícios sem desperdiçar recursos computacionais com variação de parâmetros e validação cruzada.

Os resultados mostram que o algoritmo desenvolvido está muito próximo de implementações consolidadas, demonstrando sua corretude e eficiência.