Universidade de Brasília

Faculdade UnB Gama

FGA0083 - Aprendizado de Máquina	**Semestre**: 2025.1
Docente: Sérgio Antônio Andrade	**Turma:** 01
Grupo 05	
Integrantes: Harryson Campos Martins Pedro Henrique Muniz de Oliveira Flávio Gustavo Araújo de Melo Leandro de Almeida Oliveira Jefferson Sena Oliveira José André Rabelo Rocha	**Matrícula:** 211039466 200059947 211030602 211030827 200020323 211062016

Mini Trabalho 6:

Otimização e ajuste fino do sistema

Introdução

Este trabalho tem como objetivo apresentar a otimização e ajuste fino de modelos de aprendizado de máquina, incluindo Random Forest, SVM e Regressão Logística, aplicados à predição de resultados dos jogos do Campeonato Brasileiro. O foco é mostrar a eficácia da otimização dos hiperparâmetros, o uso das técnicas de validação cruzada e a melhoria no desempenho do projeto.

Metodologia

Para cada modelo utilizado, foi desenvolvido um código específico em Python, abrangendo desde o carregamento dos dados até a calibração dos hiperparâmetros e a aplicação de validação cruzada. As métricas utilizadas para avaliação incluem acurácia, precisão, recall, F1-score e matriz de confusão, proporcionando uma análise comparativa detalhada do desempenho entre os diferentes algoritmos.

Random Forest

Validação cruzada

Foi utilizada a validação cruzada do tipo K-Fold com cinco divisões (5-fold), os dados de treinamento foram divididos em cinco partes de tamanhos aproximadamente iguais. Em cada rodada de validação, quatro dessas partes foram usadas para treinar o modelo, enquanto a parte restante foi utilizada para testá-lo. Esse processo foi repetido cinco vezes, garantindo que cada parte dos dados fosse utilizada como teste uma vez e como treinamento nas demais rodadas.

Além disso, os resultados obtidos durante a validação cruzada foram bastante semelhantes aos observados no conjunto de teste final, o que indica que o modelo conseguiu generalizar bem e não apresentou sinais de sobreajuste.

Métricas da validação cruzada:

Acuracy: 0.4741

Precision: 0.4105

Recall: 0.4741

F1-Score: 0.4032

Otimização de hiperparâmetros

Hiperparâmetros são configurações internas do modelo que precisam ser definidas antes do treinamento e que influenciam diretamente seu desempenho. No caso do RandomForestClassifier utilizado, os principais hiperparâmetros são:

n_estimators: Número de árvores na floresta, que impacta diretamente a capacidade do modelo de capturar padrões complexos dos dados.

max_depth: Profundidade máxima de cada árvore, que ajuda a controlar o quanto cada árvore pode se ajustar aos dados.

min_samples_split: Número mínimo de amostras necessárias para dividir um nó em uma árvore, afetando a granularidade das decisões.

min_samples_leaf: Número mínimo de amostras que cada folha (nó final) deve conter, ajudando a prevenir o overfitting.

Para encontrar a combinação ideal desses hiperparâmetros, foi utilizada a técnica de GridSearchCV, que testa exaustivamente todas as combinações possíveis dentro de um conjunto pré-definido de valores.

Métricas antes da otimização:

Accuracy: 0.4105

Precision: 0.3848

Recall: 0.4105

F1-Score: 0.3886

Métricas depois da otimização:

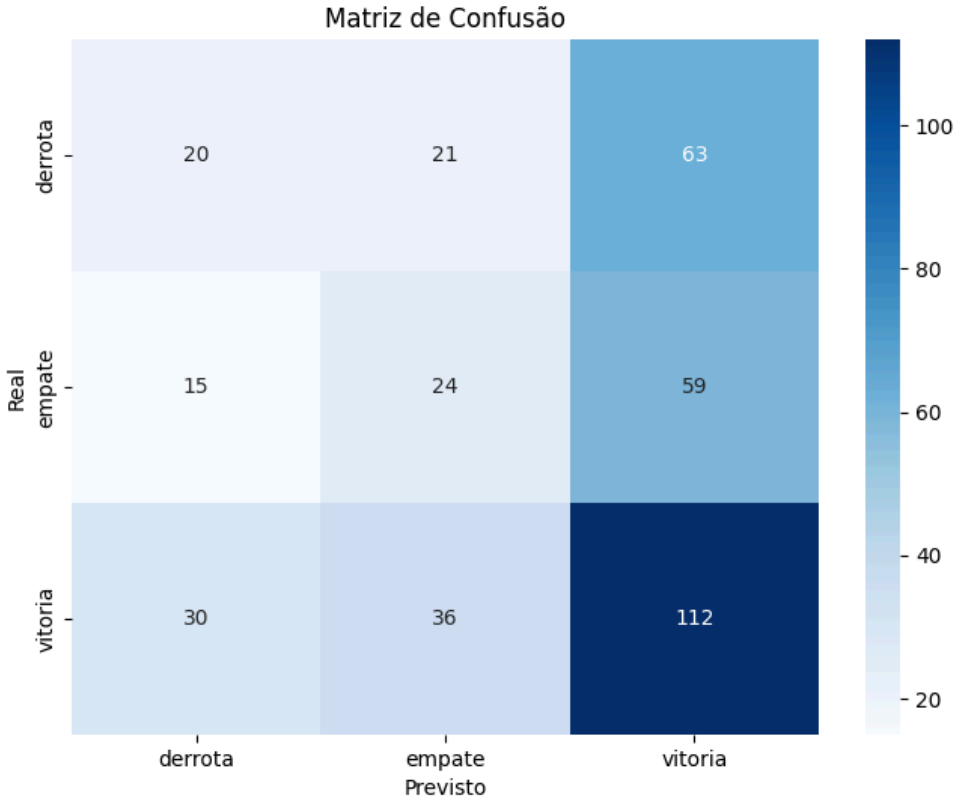
Accuracy: 0.4500

Precision: 0.4148

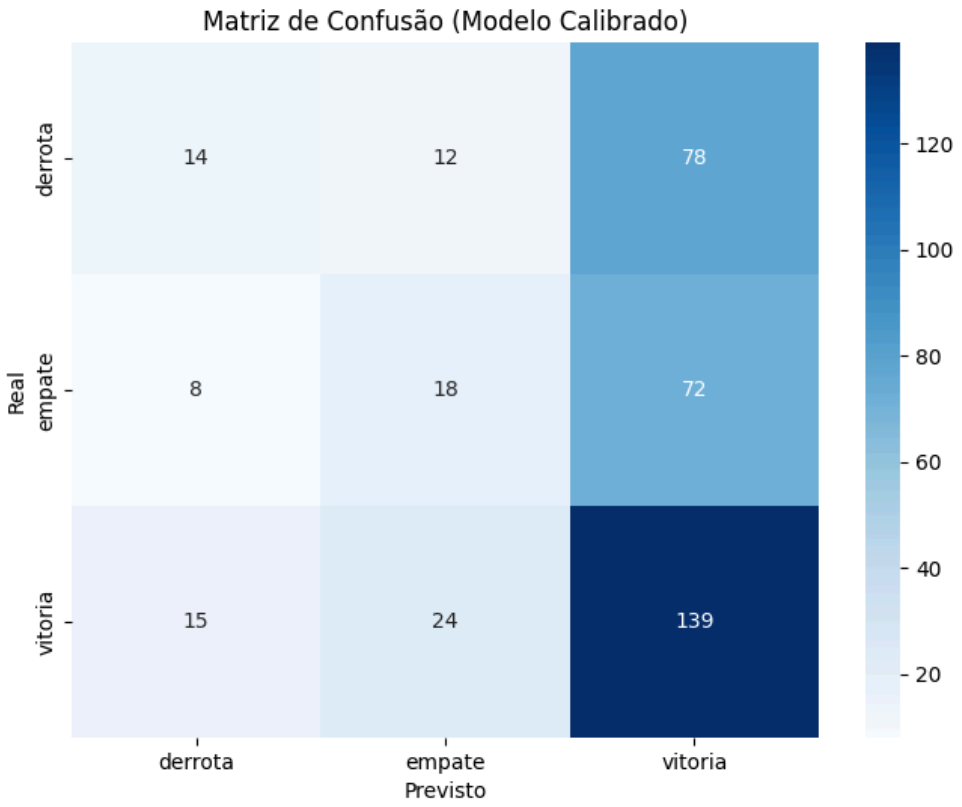
Recall: 0.4500

F1-Score: 0.3943

Matriz de Confusão antes da otimização:



Matriz de Confusão depois da otimização:



Após a calibração dos hiperparâmetros a acurácia aumentou de 41,05% para 45,00%, indicando que o modelo passou a acertar mais previsões no conjunto de teste. Além disso, a precisão, o recall e o F1-Score também apresentaram melhorias.

As matrizes de confusão também ajudam a mostrar como o modelo melhorou depois da calibração. Antes do ajuste, o modelo acertava principalmente os casos de "vitória" (112 acertos), mas cometia muitos erros ao tentar identificar "derrotas" e "empates". Depois da calibração, o número de acertos em "vitória" aumentou para 139, e o modelo passou a errar menos nas outras categorias.

Support Vector Machine (SVM)

Foi utilizado o algoritmo Support Vector Machine (SVM) para realizar a classificação do resultado de partidas de futebol (vitória, empate ou derrota do mandante), com base no dataset dos dados do Campeonato Brasileiro.

O modelo inicial, foi treinado com o SVC da biblioteca scikit-learn, utilizando os seguintes parâmetros: kernel='rbf', class_weight='balanced' (para lidar com possíveis desbalanceamentos nas classes), gamma='scale' e C=1.0. Após o treinamento, o modelo foi avaliado no conjunto de teste e apresentou os seguintes resultados: Accuracy: 0.3447, F1-score Macro: 0.3364, Precisão Macro: 0.3422, Revocação Macro: 0.3441.

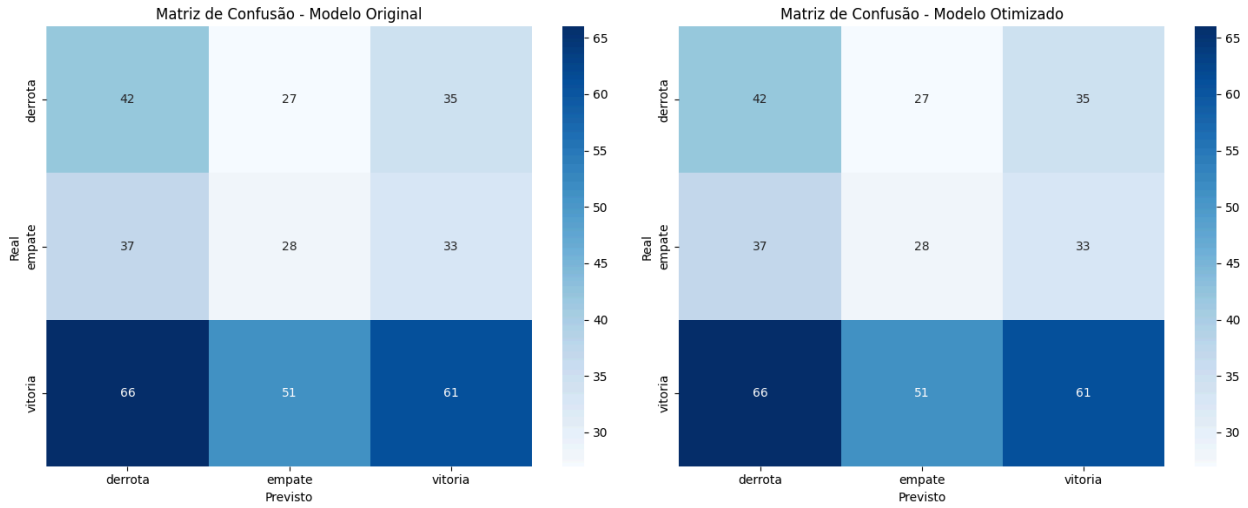
Esses resultados indicam um desempenho modesto, com dificuldades evidentes em distinguir corretamente entre as três classes.

Após essa análise, foi realizada uma otimização utilizando o GridSearchCV, combinada com validação cruzada estratificada (StratifiedKFold com 5 divisões), com o objetivo de encontrar os melhores hiperparâmetros para o modelo SVM. A métrica utilizada para otimização foi o f1_macro, adequada para cenários de classificação multiclasse com classes desbalanceadas.

O modelo otimizado, embora validado com uma abordagem mais robusta (cross-validation), obteve resultados idênticos ao modelo original no conjunto de teste: Accuracy: 0.3447, F1-score Macro: 0.3364, Precisão Macro: 0.3422, Revocação Macro: 0.3441.

Isso indica que o modelo inicial já estava utilizando os melhores hiperparâmetros dentro do espaço avaliado. Além disso, os resultados da validação cruzada apresentaram uma média de F1-score de aproximadamente 0.3422, com um desvio padrão pequeno, sugerindo consistência entre os folds.

Seguindo os resultados obtidos, foi gerada a seguinte matriz de confusão:



Regressão Logística

1.0 Metodologia

1.1 Conjunto de Dados

- Fonte: Campeonato Brasileiro de Futebol (2003–2023)
- Divisão:
    - Treino (2003–2022): 8.025 jogos
    - Teste (2023): 380 jogos
- Classes: Vitória, Empate, Derrota (na perspectiva do time mandante)

1.2 Abordagem de Otimização

1. Modelo Básico: Implementação inicial sem otimização
2. Validação Cruzada: Avaliação do modelo básico usando validação cruzada estratificada (5 folds)
3. Otimização de Hiperparâmetros: Busca em grade (GridSearchCV) para encontrar os melhores parâmetros
4. Modelo Otimizado: Implementação do modelo com os melhores hiperparâmetros

5. Avaliação Final: Comparação entre o modelo básico e o otimizado

2. Eficácia da Otimização de Hiperparâmetros

2.1 Hiperparâmetros Explorados

- C: Controla a força da regularização (valores testados: 0.1, 0.5, 1.0, 2.0)
- penalty: Tipo de regularização (L1 e L2)
- solver: Algoritmo de otimização (saga)
- class_weight: Balanceamento de classes (balanced, None)
- multi_class: Estratégia para problemas multiclasse (multinomial)

2.2 Melhores Hiperparâmetros Encontrados

- C: 0.1
- penalty: l1
- solver: saga
- class_weight: balanced
- multi_class: multinomial

2.3 Impacto da Otimização

Métrica	Modelo Básico	Modelo otimizado	Melhoria
Acurácia	46,84%	47,11%	+0,27%
Precisão	21,94%	36,73%	+14,79%
Recall	46,84%	47,11%	+0,27%
F1-Score	29,88%	34,04%	+4,16%
Tempo de treino	0,11s	4,85s	—

A otimização de hiperparâmetros resultou em melhorias significativas, especialmente na **precisão**, que aumentou em quase 15 pontos percentuais. O

F1-Score, que combina precisão e recall, também apresentou uma melhoria considerável.

3. Uso de Técnicas de Validação Cruzada

3.1 Estratégia de Validação

- Método: StratifiedKFold com 5 folds
- Estratificação: Mantém a proporção de classes em cada fold
- Aleatorização: shuffle=True com random_state=42 para reprodutibilidade

3.2 Resultados da Validação Cruzada

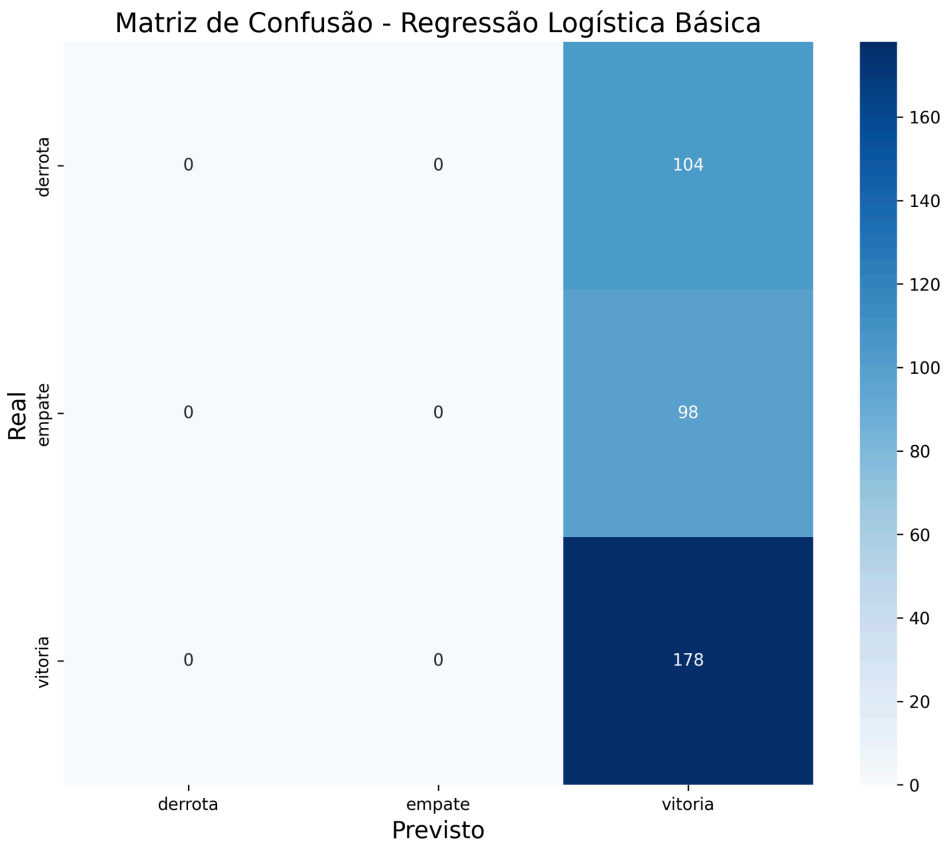
Modelo	Acurácia Média	Precisão
Básico	49,86%	±0,02%
Otimizado	50,74%	±0,27%

O modelo otimizado apresentou desempenho mais consistente nos dados de treinamento. O leve aumento no desvio padrão indica maior sensibilidade às variações dos dados.

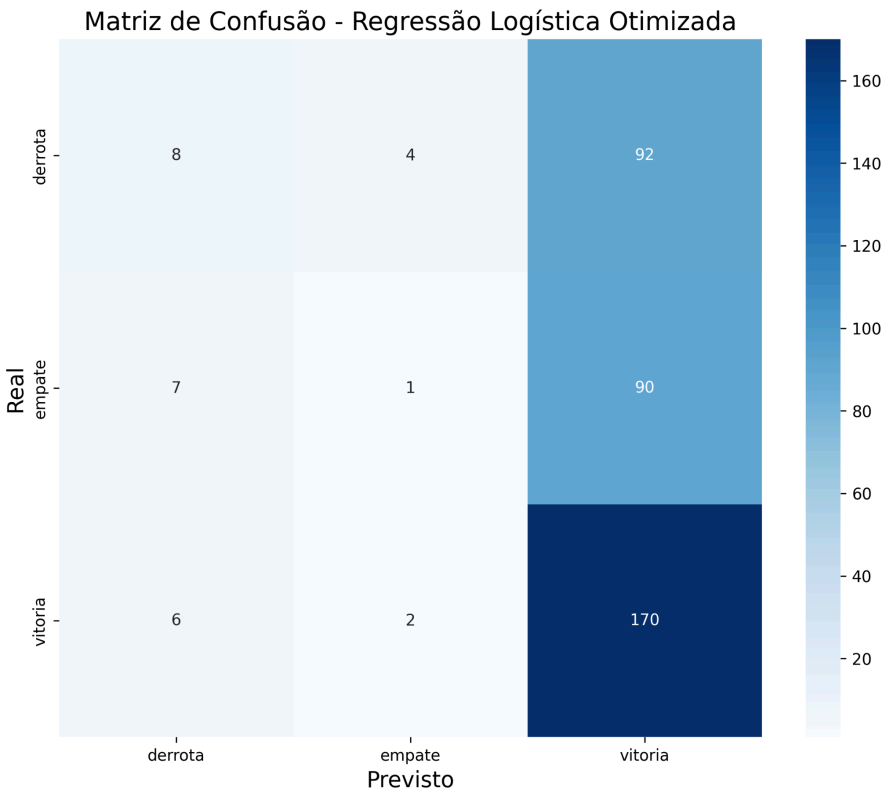
4. Melhoria no Desempenho do Modelo

4.1 Comparação de Matrizes de Confusão

Modelo inicial:



Modelo Otimizado:



O modelo básico apresentava um viés extremo, classificando todos os casos como “vitória”. O modelo otimizado começou a prever as três classes, ainda que com viés.

4.2 Melhorias nas Features

Foram adicionadas features estatísticas que melhoraram o desempenho:

- Média de gols marcados pelo time mandante
- Média de gols sofridos pelo time mandante
- Média de gols marcados pelo time visitante
- Média de gols sofridos pelo time visitante

4.3 Evidências de Avanços

1. **Balanceamento de Classes**: O modelo passou a prever as três classes.

2. **Precisão Melhorada**: Aumento de 21,94% para 36,73%.

3. **F1-Score Superior**: Aumento de 29,88% para 34,04%.

4. **Validação Cruzada**: Melhoria na acurácia média de 49,86% para 50,74%.

Conclusão

Este trabalho evidenciou como a combinação de boas práticas em aprendizado de máquina — como a otimização de hiperparâmetros, a validação cruzada e o enriquecimento das features — pode impactar significativamente o desempenho de modelos preditivos no contexto esportivo. Foram aplicadas essas técnicas em três algoritmos distintos: Random Forest, Support Vector Machine (SVM) e Regressão Logística, com foco na predição dos resultados do Campeonato Brasileiro.

A Random Forest apresentou melhorias consistentes em todas as métricas após o ajuste fino, destacando-se pelo bom equilíbrio entre precisão, recall e F1-Score. A Regressão Logística obteve ganhos expressivos em precisão e F1-Score, além de demonstrar capacidade de prever corretamente as três classes (vitória, empate e derrota), superando o viés observado na versão inicial do modelo. Já o SVM, embora não tenha tido ganhos após a otimização, mostrou estabilidade e desempenho consistente dentro do espaço de hiperparâmetros testados.

O aprimoramento das features utilizadas — como médias de gols marcados e sofridos pelos times — também foi decisivo para os avanços observados, contribuindo para um modelo mais informativo e eficaz. O uso da validação cruzada estratificada garantiu a confiabilidade dos resultados e reforçou a importância da avaliação robusta em problemas de classificação.

Dessa forma, conclui-se que a integração entre ajuste fino de modelos, validação rigorosa e engenharia de atributos é essencial para construir soluções mais robustas, generalizáveis e eficazes, especialmente em cenários desafiadores como a predição de resultados esportivos.