Influência do desbalanceamento de classes no algoritmo SVM

TRABALHO PRÁTICO - APRENDIZAGEM COMPUTACIONAL I

Diogo Padilha, 202305188
Leonor Rodrigues, 202304869
Rita Nunes, 202305232
PL5 - Grupo 4
L.IACD || FCUP
2024/2025

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

VISÃO GERAL

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

- O **objetivo** deste trabalho é investigar como o desbalanceamento de classes afeta o desempenho do algoritmo **Support Vector Machine (SVM)**, em especial para o kernel Linear. Posteriormente a ser avaliado, iremos propor algumas abordagens para mitigar os seus efeitos, aplicando alterações ao algoritmo, de modo a torná-lo mais robusto e aplicável a 49 datasets, comparando os resultados iniciais e os resultados após alterações.

- A **abordagem** começa por implementar o algoritmo SVM do zero. De seguida, é avaliado o seu desempenho em datasets gerados com diferentes níveis de desbalanceamento, analisando o comportamento através de métricas de performance relevantes. De seguida, procuramos uma nova proposta do SVM e sua respetiva implementação, de modo a aumentar a performance do algoritmo na característica em causa, retirando conclusões finais.

VISÃO GERAL

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

- Os **resultados** demonstram que o desbalanceamento de classes afeta significativamente o desempenho do SVM original, com uma redução na capacidade de classificar corretamente a classe minoritária, pois a classe maioritária assume o controlo do algoritmo.

- A **análise** sugere a necessidade de ajustes no SVM com o objetivo de melhorar o desempenho na globalidade dos datasets. Assim, implementamos medidas como a aplicação de custos para os erros e o ajuste do bias. A **combinação das propostas de alteração** resultou no novo SVM final, que é teoricamente mais robusto para os dados em causa. Para forma de avaliação e comparação inicial e final, calculamos a mediana e a média do G-Mean e o desvio padrão, bem como gráficos de relação entre o G-mean e o Desbalanceamento. Além disso, realizamos um teste estatístico e outros gráficos de relevância.

SVM (SUPPORT VECTOR MACHINES)

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

- O algoritmo selecionado, Support Vector Machine (SVM), é utilizado principalmente para classificação, embora possa também ser adaptado para regressão.
- O principal objetivo é encontrar um hiperplano ótimo que maximize a margem entre duas classes de dados. A margem corresponde à distância entre o hiperplano e os pontos de dados mais próximos de cada classe, conhecidos como vetores de suporte.
- Quanto maior a margem, maior tende a ser a capacidade do modelo de generalizar e classificar corretamente novos dados.
- Explorámos três tipos de kernel: Linear, Polinomial e RBF (Radial Basis Function), de modo a verificar qual o mais sensível a dados desbalanceados. Após esta análise, focamo-nos em aplicar as alterações a um só tipo de kernel, para testes mais aprofundados.
- Os kernels permitem que o SVM mapeie os dados para um espaço de maior dimensão, onde uma separação linear se torna possível, mesmo que os dados sejam não linearmente separáveis no espaço original.
  - **Kernel Linear:** Separa os dados com uma linha reta (em 2D) ou hiperplano (em dimensões superiores). Simples e eficiente, mas apenas é eficaz para dados linearmente separáveis.
    - *np.dot(x, y.T)* - produto escalar entre os vetores;
  - **Kernel Polinomial:** Permite criar fronteiras de decisão curvas. A complexidade da separação aumenta com o grau escolhido.
    - *np.dot(x, y.T) ** self.degree* - eleva o produto escalar para gerar curvas mais flexíveis.
  - **Kernel RBF (Radial Basis Function):** Define fronteiras altamente não lineares com base na distância euclideana entre pontos. Usa uma função gaussiana para controlar a largura da influência dos pontos.
    - *np.exp(-self.gamma * dist.cdist(x, y) ** 2).flatten()* - calcula a similaridade entre vetores com base na distância.

SVM (SUPPORT VECTOR MACHINES)

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

- A classe *BaseEstimator* tem como função organizar e validar os dados X e y antes de qualquer cálculo.
- O treino do SVM consiste em encontrar valores ideais para os coeficientes alphas, α, e do bias, b, que definem o hiperplano ideal. É utilizada a técnica de **SMO - Sequential Minimal Optimization** de forma simplificada.
- Antes de treinar, definimos *C*, *kernel*, *tol* e *max_iter*. Os dados são verificados e a matriz de kernel é calculada. Esta matriz guarda a similaridade entre todos os pares de pontos de treino.
- De seguida, começa a otimização iterativa do SMO:
  - Inicia com todos os alphas iguais a zero;
  - Em cada passo, seleciona dois índices i e j e ajusta os seus alphas;
  - Calcula os limites *permitidos L* e *H* para alpha[j];
  - Mede o erro (*e_i*, *e_j*) atual do modelo atual para os pontos i e j;
  - Atualiza os valores de alpha[i] e alpha[j], respeitando os limites;
  - Ajusta o bias (*b*) com base na nova posição do vetores de suporte.
- Este ciclo repete-se até a diferença entre os alphas antigos e os novos ser inferior à tolerância (tol), ou até atingir o número máximo de iterações.
- No final do treino, os vetores de suporte são os pontos cujo alpha é maior do que zero, e o bias também é definido.
- Para classificar um novo exemplo, o modelo:
  - Calcula a similaridade entre o novo ponto e os vetores de suporte (com o kernel escolhido);
  - Aplica uma soma ponderada dos alphas, y e similaridade para obter o valor da função de decisão.
  - Usa np.sign() para atribuir +1 ou -1 com base nesse valor.

SVM - MÉTRICAS DE AVALIAÇÃO

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

- True Positive (TP) → Modelo previu "minoritário" e era mesmo "minoritário".
- False Positive (FP) → Modelo previu "minoritário", mas era "maioritário" (erro).
- False Negative (FN) → Modelo previu "maioritário", mas era "minoritário" (erro).
- True Negative (TN) → Modelo previu "maioritário" e era mesmo "maioritário".

Accuracy

Proporção total de acertos. Em cenários desbalanceados pode ser enganosa: prever sempre a classe maioritária pode resultar numa accuracy elevada, mas sem utilidade prática para o problema.

Precision

Proporção de verdadeiros positivos entre as previsões positivas. TP / (TP + FP). Útil quando é importante evitar falsos positivos. Menos crítica se o objetivo for identificar todos os exemplos da classe minoritária.

Recall

Proporção de reais positivos corretamente identificados: TP / (TP + FN). Fundamental quando queremos evitar que exemplos da classe minoritária passem despercebidos.

F1-score

Média equilibrada entre Precision e Recall. Equilibra bem a preocupação entre não perder positivos e não gerar muitos falsos positivos. Relevante quando existe um trade-off entre ambas.

ROC-AUC

Mede a capacidade do modelo distinguir entre classes. Divide a taxa de verdadeiros positivos (TPR = Recall) no eixo Y e a taxa de falsos positivos (FPR) no eixo X. Um valor próximo de 1 indica um bom separador.

G-Mean

Raiz quadrada do produto entre o Recall e a Specificity. Avalia o desempenho equilibrado entre classes, penalizando modelos que negligenciam uma das classes.

Deste modo, com este conjunto de métricas conseguimos uma avaliação mais robusta do desempenho do modelo, sobretudo no que diz respeito à deteção da classe minoritária. A métrica principal adotada é o G-Mean.

SVM - DESBALANCEAMENTO DE CLASSES

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

- O desbalanceamento de classes ocorre quando uma classe tem significativamente mais amostras do que a outra. Esta situação é muito comum em dados reais, como por exemplo, em casos de doenças raras, deteção de vírus, spam ou fraudes.

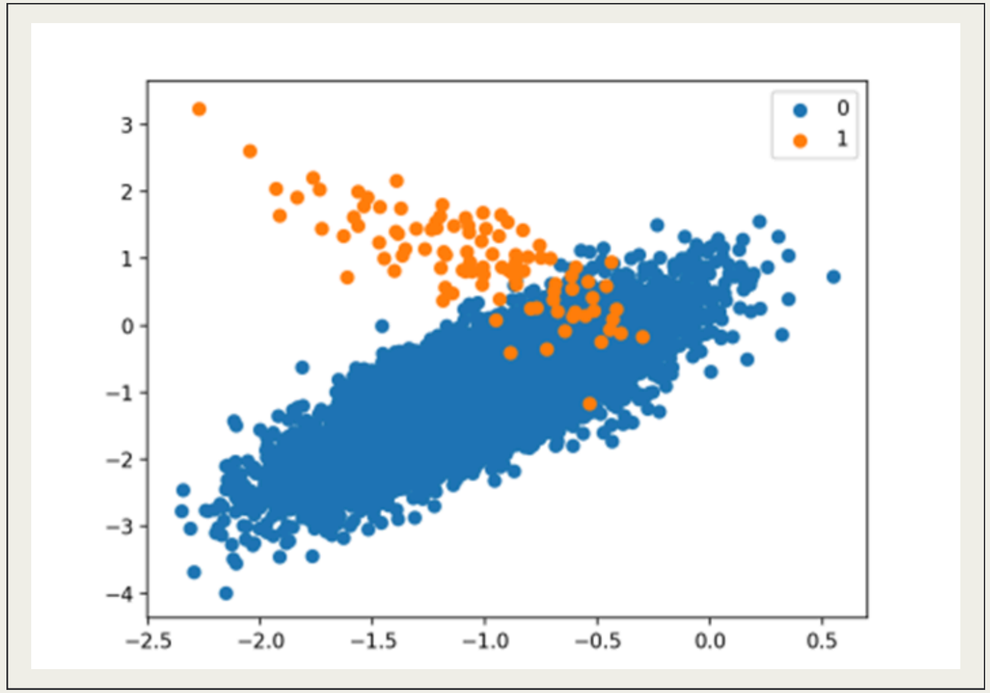


IMAGEM 1

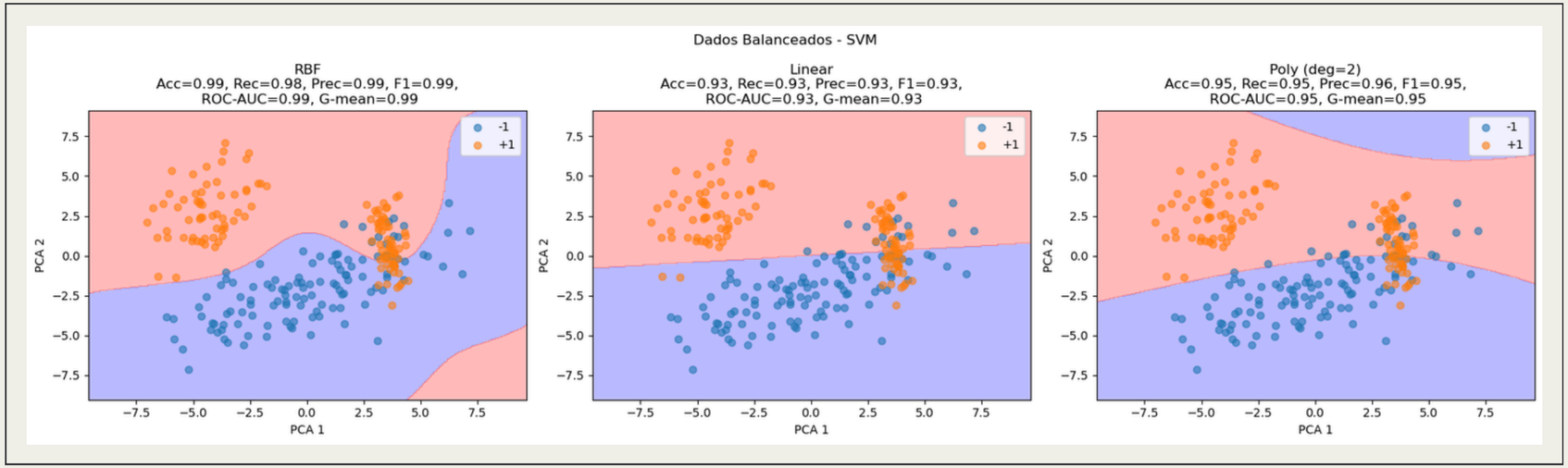
Comportamento do SVM perante o desbalanceamento de classes:

- O SVM tenta encontrar um hiperplano que maximiza a margem total, sem considerar o desbalanceamento/desequilíbrio entre as classes.
- Como resultado, o modelo tem tendência a favorecer a classe maioritária, pois esta assume o domínio na função que elabora o hiperplano.
- A classe minoritária é frequentemente mal classificada, afetando negativamente métricas como o recall e o F1-score.
- O modelo pode aparentar ter accuracy elevada, mas falha em capturar a classe mais importante — a minoritária.
- Logo, os modelos SVM não são na sua natureza robustos ao desbalanceamento de classes, sendo necessário introduzir técnicas específicas para corrigir essa limitação.

PCA (PRINCIPAL COMPONENT ANALYSIS)

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

- PCA é uma técnica de redução de dimensionalidade que transforma os dados num novo sistema de coordenadas, mantendo a maior parte da variabilidade usando menos dimensões.
- Assim, facilita a visualização de dados complexos e ajuda a eliminar redundâncias e ruído, focando nas componentes mais relevantes. Além disso, pode melhorar a separação entre classes quando existe correlação entre variáveis.



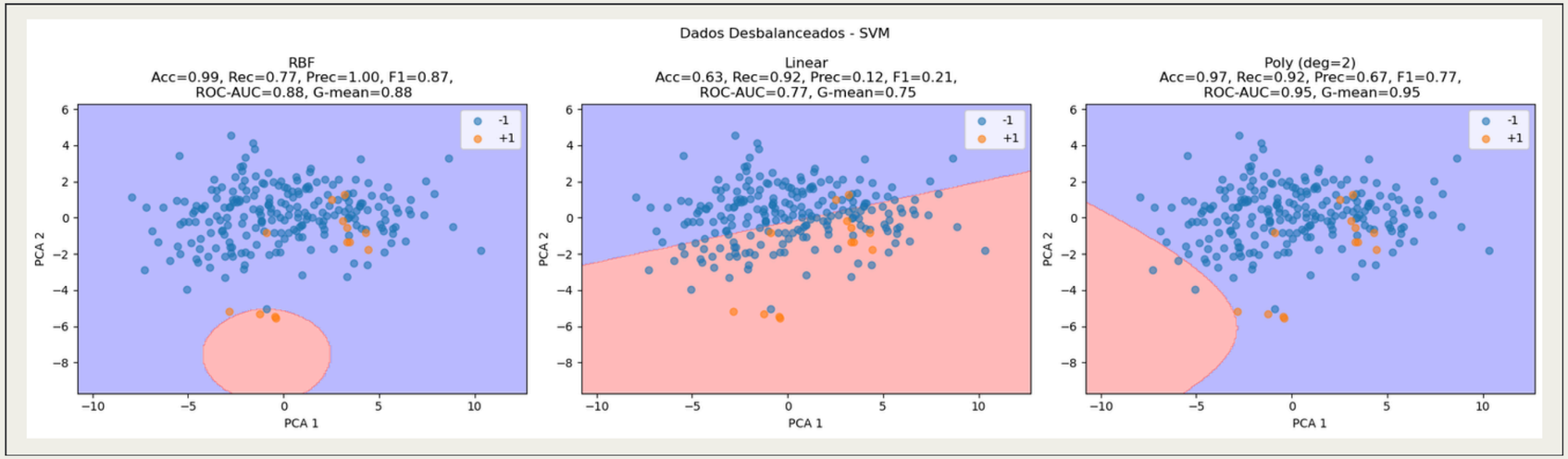


IMAGEM 2

ESCOLHA DO SVM LINEAR

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

A escolha de implementar melhorias no modelo **SVM Linear**, em vez dos modelos com Kernel Polinomial e RBF, baseou-se nas seguintes observações:

- Na avaliação com dados balanceados, o Kernel Linear apresentou desempenho competitivo, com G-mean = 0.93 e Recall = 0.93, definindo uma fronteira clara entre classes após projeção com PCA. Por sua vez, o Kernel Polinomial (grau 2) obteve também um bom desempenho e Kernel RBF demonstrou ser ainda mais robusto.

- Com dados desbalanceados, o impacto negativo foi notório no Kernel Linear: a Precision reduziu para 0.12 e a G-mean para 0.75, apesar de um recall ainda razoável, o que revela que o modelo linear é sensível ao desbalanceamento. O Kernel Polinomial, por sua vez reduziu a Precision e o F1-Score. O RBF revelou-se de certa forma adaptável ao cenário com dados desbalanceados.

- Do ponto de vista teórico, o SVM Linear tem a vantagem de ser mais interpretável e controlável, permitindo aplicar alterações algorítmicas e verificar os seus resultados. Além disso, por ter uma estrutura mais controlada e previsível, permite avaliar com maior clareza o efeito isolado de alterações específicas para mitigar o desbalanceamento, sem que a complexidade do modelo interfira nos resultados.

- Assim, optamos por focar as alterações no SVM Linear, com o objetivo de melhorar a classificação da classe minoritária para dados desbalanceados, pois foi o que sofreu mais alterações nas métricas, sem comprometer a interpretabilidade nem introduzir complexidade desnecessária.

CARACTERÍSTICAS DOS DATASETS

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Para análise dos 49 datasets para estudo do desbalanceamento de classes, optamos por realizar:

- Contagem dos valores da classe minoritária e da classe maioritária
- Gráfico da percentagem da classe minoritária de cada dataset (nrº valores da classe minoritária / total)

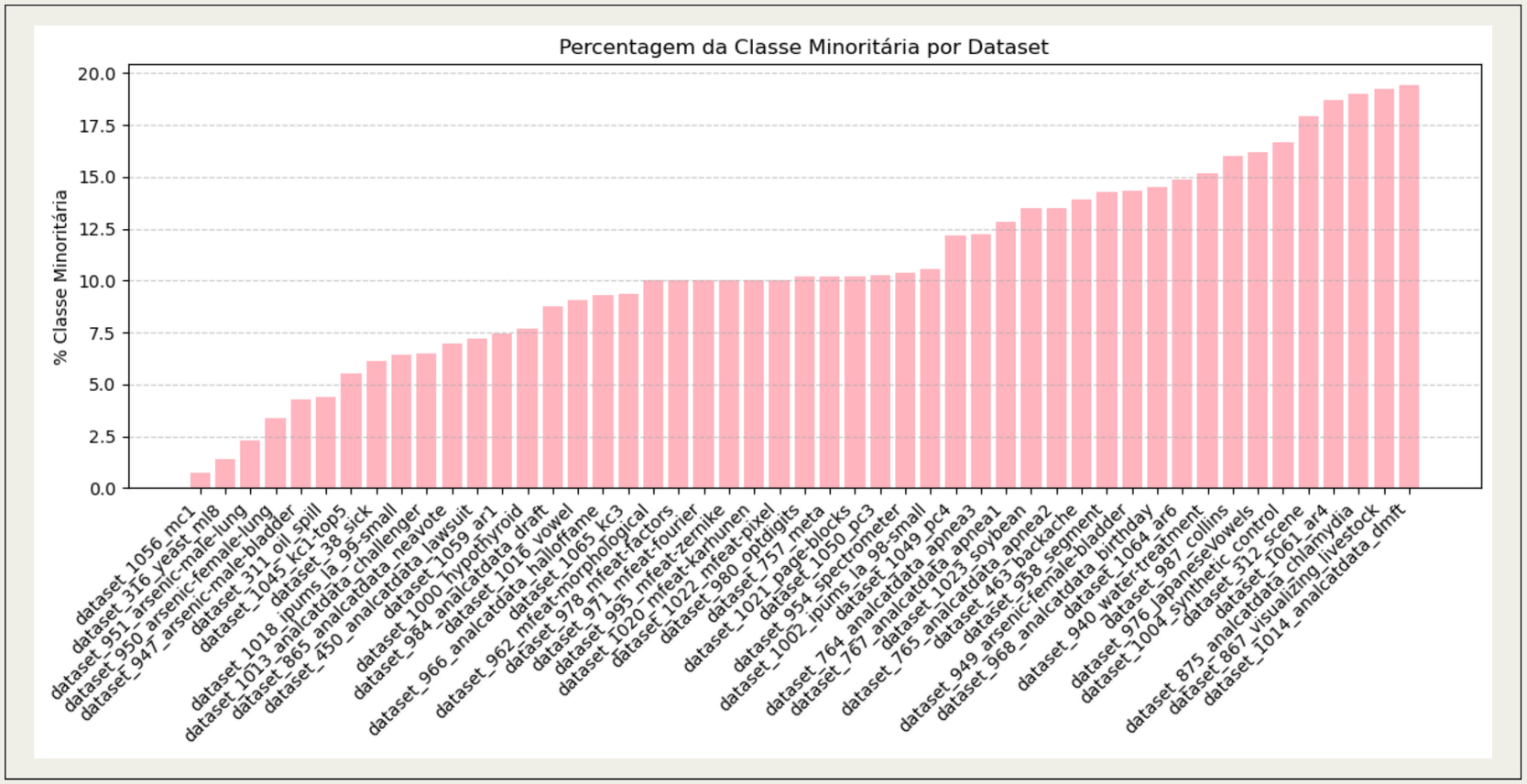


IMAGEM 3

CARACTERÍSTICAS DOS DATASETS

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

- Gráfico do desbalanceamento de cada dataset (nrº valores da classe minoritária / nrº valores da classe maioritária).

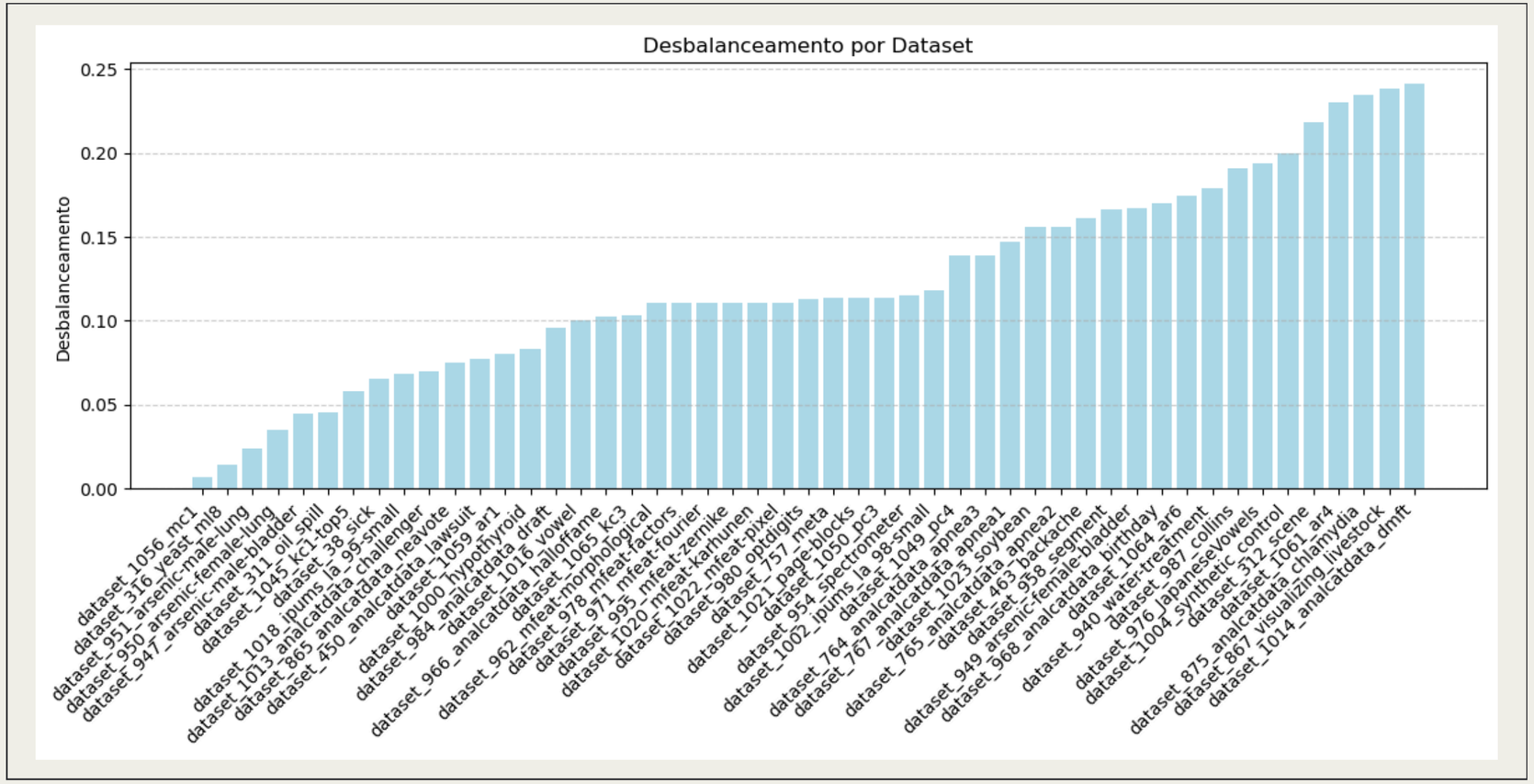


IMAGEM 4

- Assim, sabendo o enviesamento dos datasets em causa, podemos avançar para as propostas de combate ao desbalanceamento de classes.

PROPOSTAS DE MODIFICAÇÃO - MOTIVAÇÃO

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

- O SVM original **não distingue** entre classes ao penalizar erros: todos os exemplos incorretamente classificados contribuem da mesma forma para a função de custo. Num cenário com classes desbalanceadas, esta abordagem **favorece** inevitavelmente a **classe maioritária**, levando o modelo a **comprometer a deteção da classe minoritária** - que é precisamente a mais difícil e relevante de acertar em cenários desbalanceados.

- De modo a mitigar este e outros problemas, inicialmente considerámos usar pesos nas classes, *oversampling* e *undersampling* para lidar com o desbalanceamento. No entanto, estas abordagens são soluções superficiais que não resolvem a origem do problema: **o SVM continua a ignorar o impacto desigual da classe minoritária na otimização.**

- Desta forma, é necessário **intervir diretamente no funcionamento interno do algoritmo SVM,** para garantir que este considere a importância desigual das classes na definição da margem e na atualização dos parâmetros.

PROPOSTA 1

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Priorização periódica da classe minoritária na escolha dos pares *(i, j)*

A proposta 1 altera a estratégia de seleção dos pares *(i, j)* no algoritmo SMO com o objetivo de dar maior atenção à classe minoritária, que tende a ser subrepresentada no processo de otimização. A alteração está implementada no método *_train*.

Modificações principais:

- **Identificação da classe minoritária:** No início do treino, são identificados e guardados os índices dos exemplos pertencentes à classe minoritária - assumida como *y == 1.*
- **Nova política de escolha do índice *i*:**
  - Introduz-se o parâmetro minority_every_n_iters, que define a frequência de iterações principais de treino em que o modelo força a escolha de um exemplo da classe minoritária como índice *i*.
  - Nestas iterações específicas, para cada amostra *j,* o índice *i* é selecionado aleatoriamente de entre os exemplos da classe minoritária, garantindo *i ≠ j.*
  - Nas restantes iterações, a seleção de *i* segue a estratégia original (aleatória e diferente de *j*).

Efeito esperado da modificação:

Ao garantir que, de forma periódica, uma parte da otimização do SVM se foque explicitamente em exemplos da classe minoritária, pretende-se:

- Corrigir o comportamento padrão do SVM para a classe maioritária;
- Forçar ajustes no hiperplano de separação que levem em consideração os exemplos mais escassos;
- Melhorar a capacidade do modelo identificar corretamente exemplos minoritários, com ganhos esperados em Recall, F1-score e G-Mean.

PROPOSTA 2

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Introdução de penalização assimétrica com custos *Ck*

A proposta 2 tem como principal objetivo melhorar a sensibilidade do SVM à classe minoritária ao aplicar penalizações diferentes - custos *Ck* - para cada amostra, reforçando assim o impacto dos erros de classificação desta classe na otimização.

Modificações principais:

- Introdução de penalização diferenciada por amostra (*Ck*):
  - Foi criado um vetor *Ck* com o mesmo tamanho do número de amostras de treino.
  - Para cada amostra k, o custo *Ck* é definido da seguinte forma: se k pertence à classe minoritária, entao $Ck = C \times minority\_cost\_weight$. Caso contrário (classe maioritária), $Ck = C$.
- Integração no algoritmo SMO:
  - A função _find_bounds(i, j) foi adaptada para utilizar *Ck[i]* e *Ck[j]* nos cálculos dos limites inferior (L) e superior (H) dos multiplicadores de Lagrange.
  - A condição usada para decidir se a amostra *i* ou *j* é um vetor de suporte na margem também foi ajustada. Em vez de verificar $0 < \alpha k < C$, o código agora verifica $0 < \alpha k < Ck[k]$.

Efeito esperado da modificação:

- O SVM é incentivado a minimizar mais agressivamente os erros cometidos na classe minoritária.
- A função objetivo do SVM penaliza mais fortemente violações da margem por amostras minoritárias (maiores ξk), o que influencia a posição da fronteira de decisão.

PROPOSTA 3

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Margens assimétricas no cálculo do erro

A proposta número 3 introduz uma modificação no cálculo do erro de previsão usado no algoritmo SMO, com o objetivo de incorporar margens assimétricas entre classes, reforçando assim a penalização dos erros cometidos sobre a classe minoritária.

Modificações principais:

- Foi alterado o método _error(i) que substitui o cálculo tradicional do erro, normalmente definido como $f(x_i) - y_i$, por uma versão ajustada com fatores multiplicativos diferentes consoante a classe da amostra.
- São introduzidos os hiperparâmetros *rho_minority*, para aumentar a margem de decisão exigida para prever corretamente a classe minoritária; e *rho_majority*, valor padrão ou reduzido para a margem da classe maioritária.

Este mecanismo atua diretamente no cálculo do erro para cada ponto de treino, influenciando a seleção e atualização dos pares de multiplicadores de Lagrange (α_i, α_j) no processo de otimização.

Efeito esperado da modificação:

- Ao aumentar o impacto dos erros cometidos sobre as amostras da classe minoritária, esta proposta força o modelo a dar mais atenção a estes casos, ajustando a fronteira de decisão de forma a priorizar uma melhor sensibilidade (recall) nesta classe.
- Este ajuste atua como uma margem de tolerância assimétrica, exigindo maior confiança nas decisões relativas à classe minoritária.

PROPOSTA 4

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Ajuste do Bias pós-treino para maximizar o G-Mean

A proposta 4 introduz uma etapa de ajuste final no SVM, após o treino, com o objetivo de melhorar o equilibrio entre Recall e Specificity - especialmente em cenários de classes desbalanceadas.

Modificações principais:

- Após o treino, é realizada uma busca exaustiva pelo melhor valor de bias (*b*), dentro de um intervalo definido em torno do valor original, aprendido pelo modelo.
- A busca é feita iterando a *bias*, avaliando, para cada um dos valores, o desempenho do modelo num conjunto de validação (*X_val, y_val*), com base na métrica G-Mean (*geometric_mean_score*).
- O valor de *b* que maximiza o G-Mean no conjunto de validação é então guardado como *bias* final do modelo.

Efeito esperado da modificação:

- Ao ajustar o *bias* de decisão do modelo para maximizar o G-Mean, esta proposta permite deslocar a fronteira de decisão de forma controlada de modo a melhorar simultaneamente a recall da classe minoritária e a specificity da classe maioritária.
- Este tipo de ajuste é simples, eficiente e não interfere com o processo de otimização SMO, funcionando como um mecanismo complementar que pode ser particularmente eficaz quando o desbalanceamento de classes causa enviesamento na fronteira de decisão.

PROPOSTA FINAL

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Assim, o SVM Modificado incorpora quatro estratégias para lidar de forma eficaz com o desbalanceamento de classes, focado para problemas de classificação binária:

Seleção informada dos pares (i, j)

A cada 5 iterações, força-se a inclusão de um exemplo da classe minoritária como *i,* assegurando que o SMO atualize os multiplicadores com impacto direto na classe minoritária.

Penalização diferenciada com Ck

Aplicação de um custo *Ck* mais elevado para erros sobre a classe minoritária, ajustando os limites de otimização e priorizando uma classificação mais correta.

Margens assimétricas no cálculo do erro

Maior impacto dos erros da classe minoritária antes da atualização dos multiplicadores de Lagrange, promovendo uma margem mais rigorosa e protetora para esta classe.

Ajuste pós-treino do bias (b)

Otimização de *b* após o treino, com o objetivo de maximizar o G-Mean, melhorando o equilíbrio entre Recall e Specificity.

ANÁLISE DOS DATASETS

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Os testes realizados para obter os resultados apresentados foram realizados da seguinte forma, de modo a garantir consistência:

- Em comum para as duas implementações:
  - Implementação própria do SVM (sem utilização de scikit-learn)
  - Leitura e pré-processamento dos datasets
  - Hiperparâmetros: *kernel linear*, custo *C = 0.6*, número máximo de iterações *max_iter = 200*, tolerância *tol = 1e-3*
  - Validação cruzada (5 folds)
  - Recolha das métricas: Accuracy, Precision, Recall, F1-Score, ROC-AUC e G-Mean
  - Avaliação final feita sobre os mesmos dados de teste de cada fold.

- Para a implementação do SVM original:
  - Utilização do algoritmo SMO simplificado padrão, sem qualquer modificação específica para lidar com desbalanceamento de classes.
  - Seleção aleatória dos pares *(i, j)*, custo C fixo, margem simétrica e bias tradicional.

- Para a implementação do SVM modificado:
  - Aplicação das 4 propostas referidas, que implicam os novos hiperparâmetros: *minority_every_n_iters = 5,* minority_cost_weight = 20, *rho_minority = 1.2, rho_majority = 1.0.*
  - Nos 80% reservados ao treino, inclusão de uma separação adicional treino/validação (75%/25%) dentro de cada fold, usada apenas para otimizar o bias após o treino.

ANÁLISE DOS DATASETS

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

A avaliação do desempenho dos modelos SVM original e SVM modificado foi realizada com base em validação cruzada estratificada de 5 folds, garantindo que a proporção entre classes fosse preservada em cada partição. Para cada fold, foram recolhidas as previsões do modelo nos dados de teste e calculadas as métricas já referidas. Estas métricas foram depois agregadas por média aritmética ao longo dos 5 folds, originando os resultados finais por dataset.

Para garantir consistência entre as diferentes métricas binárias, os rótulos da variável alvo foram convertidos de {-1, 1} para {0, 1}, sendo a classe minoritária representada como 1. Esta normalização foi essencial para a correta utilização das métricas.

Os resultados foram armazenados em ficheiros distintos:

- resultados_svm_linear.csv para o modelo SVM original
- resultados_svm_linear_modificado.csv para o modelo SVM modificado

Este processo assegura uma comparação justa entre os modelos, sendo ambos avaliados com os mesmos dados de teste em cada fold. A única diferença entre as implementações encontra-se nas propostas realizadas no algoritmo em si.

COMPARAÇÃO DOS MODELOS

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Para comparação direta de ambas as implementações do SVM, a original e a modificada, utilizamos o seguinte:

- **Confusion Matrix**

Foram geradas matrizes de confusão por dataset, que representam o desempenho final dos modelos ao longo dos 5 folds. Para cada ficheiro de previsões (y_test, y_pred), as classes foram mapeadas para {0, 1} com a classe minoritária sempre representada por 1, assegurando coerência na leitura dos resultados.

Através desta análise é possível:

- Observar diretamente desequilíbrios de classificação, como a existência de muitos falsos negativos (FN), comuns em datasets desbalanceados.
- Avaliar se o modelo favorece a classe maioritária (0), ignorando a deteção da classe minoritária (1).
- Verificar melhorias no modelo modificado, como a redução de FN ou o aumento de TP, que indicam uma classificação mais correta da classe minoritária.

As imagens foram guardadas nas pastas:

- *confusion_matrix* - para o SVM original.
- *confusion_matrix_modificado* - para o SVM modificado

COMPARAÇÃO DOS MODELOS

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

- **Média, Mediana e Desvio Padrão com G-Mean**

	Mediana	Média	Desvio Padrão
SVM Original	0.5430	0.5357	0.3020
SVM Modificado	0.6560	0.6419	0.2415

IMAGEM 5

- A mediana do G-Mean no modelo modificado é superior, sugerindo uma melhoria sistemática no equilíbrio entre recall e specificity.
- A média também tende a ser superior no modelo modificado, indicando ganho geral.
- O desvio padrão pode fornecer uma ideia sobre a consistência do modelo entre datasets, onde valores mais baixos indicam um desempenho mais estável.

Estas métricas suportam quantitativamente a conclusão de que a versão modificada do SVM consegue melhorar a classificação da classe minoritária, sem sacrificar em excesso o desempenho na classe maioritária.

- **Gráfico de Desbalanceamento Vs. G-Mean para cada dataset**

Este gráfico compara o impacto do desbalanceamento no G-Mean. Em ambos os casos, os pontos estão dispersos, sem evidência clara de correlação. No entanto, destaca-se que no modelo modificado desaparecem os casos com G-Mean igual a 0, indicando uma melhoria relevante na capacidade de detetar ambas as classes, mesmo em datasets mais desbalanceados.

COMPARAÇÃO DOS MODELOS

- **Gráfico de melhoria do parâmetro G-Mean por dataset, em relação à percentagem da classe minoritári**

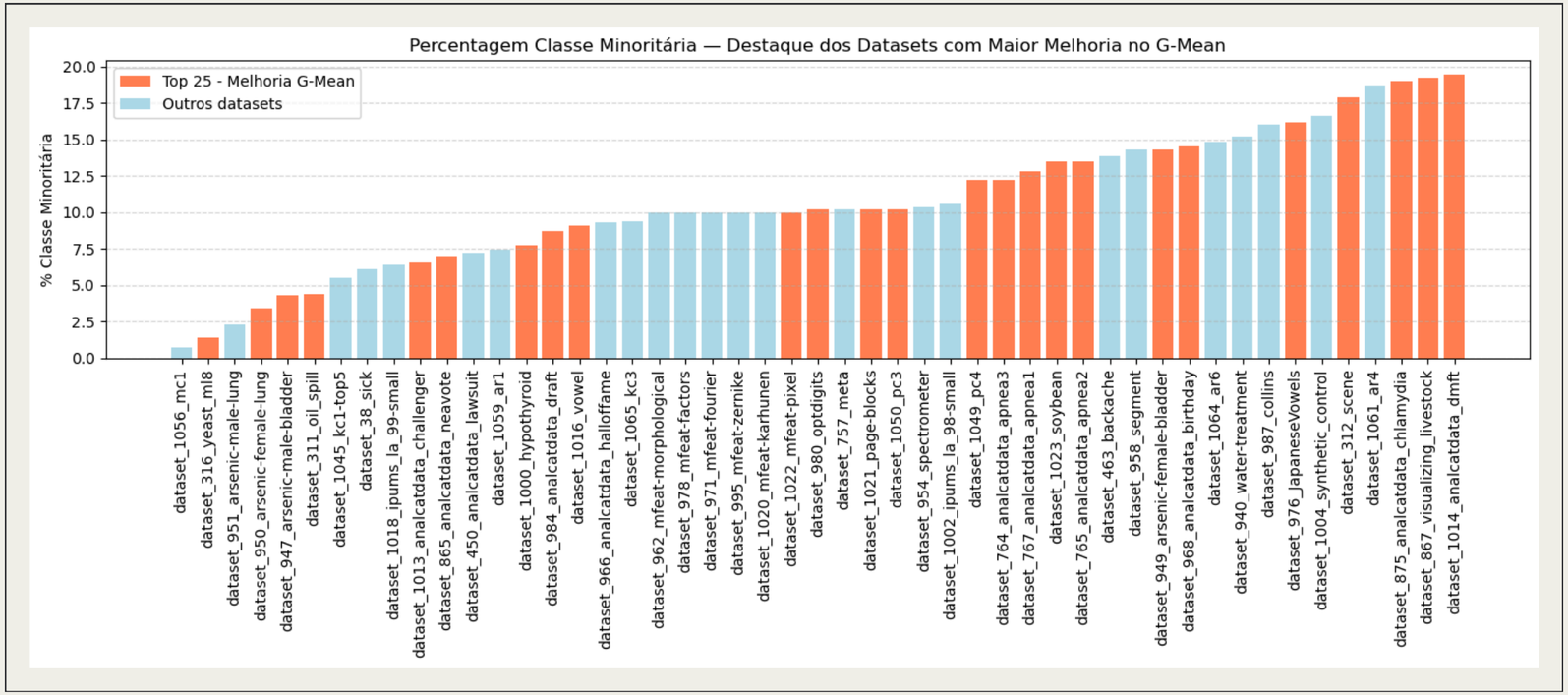


IMAGEM 6

Este gráfico apresenta a percentagem da classe minoritária em cada dataset, destacando a cor de laranja os 25 datasets que registaram as maiores melhorias no G-Mean após aplicadas as modificações ao SVM. A visualização evidencia que essas melhorias ocorreram tanto em datasets com forte desbalanceamento, ou seja, com uma baixa percentagem da classe minoritária, como nos datasets menos desbalanceados dentro dos existentes. Isto sugere que as modificações realizadas tornam o SVM eficaz tanto para grandes desbalanceamentos como para desbalanceamentos mais moderados, sendo, por isso, um modelo de certa forma robusto dentro de dados desbalanceados.

COMPARAÇÃO DOS MODELOS

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

- **Justificação dos resultados do G-Mean**

Através da análise do gráfico anterior, verifica-se que existiram tanto aumentos, como diminuições na métrica do G-Mean, para comparação dos dois modelos.
Em vários datasets, como:

- dataset_1002_ipums_la_98-small
- dataset_1018_ipums_la_99-small
- dataset_1049_pc4
- dataset_1059_ar1
- dataset_1061_ar4
- dataset_463_backache
- dataset_954_spectrometer
- dataset_958_segment
- dataset_995_mfeat-zernike

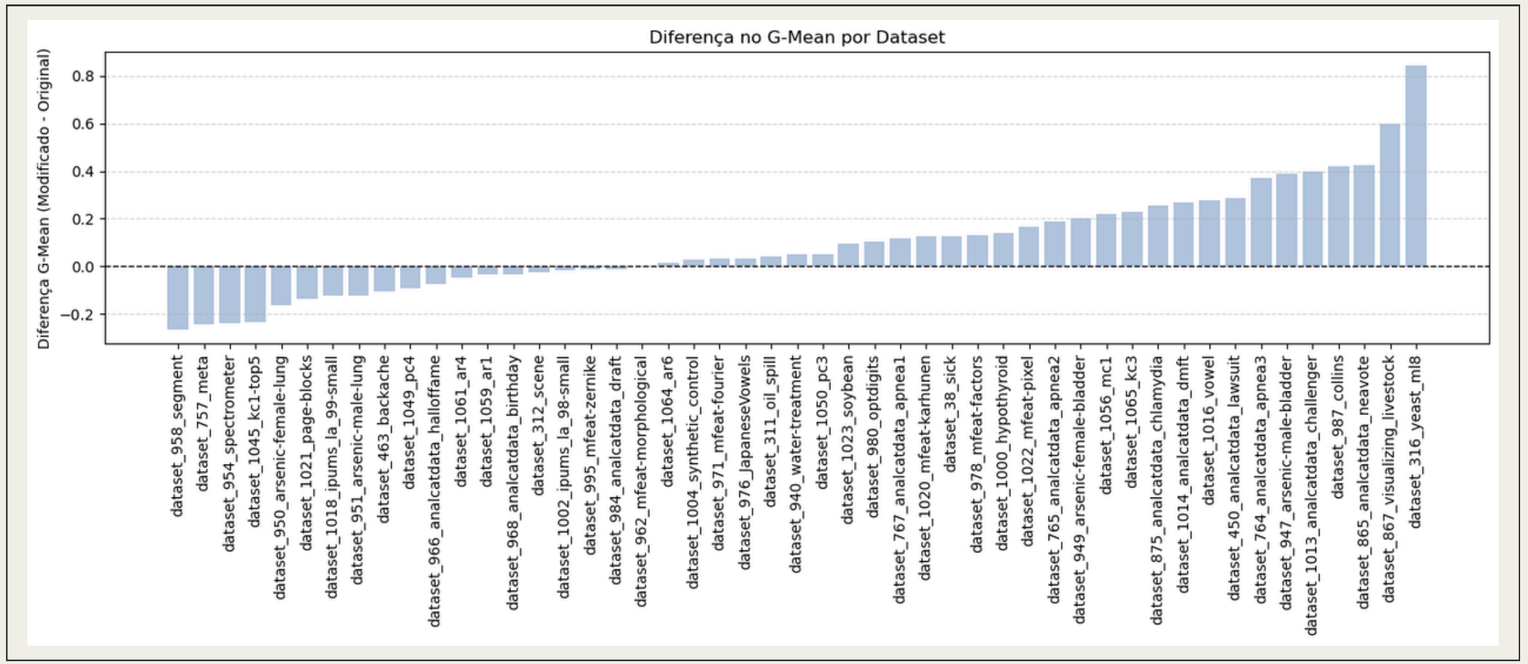


IMAGEM 7

o G-Mean reduziu, mas o Recall melhorou. Isto pode significar que o modelo está a detetar mais verdadeiros positivos da classe minoritária, ainda que isso tenha ocorrido à custa de um aumento nos falsos positivos.

Este tipo de comportamento pode ser desejável em contextos mais sensíveis, onde as consequências de não identificar corretamente um caso relevante são mais graves do que as de gerar um falso alarme.

TESTE ESTATÍSTICO DE WILCOXON

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Com o objetivo de comparar estatisticamente o desempenho (medido pelo G-Mean) entre a versão Inicial do SVM e a versão Final (com as melhorias propostas), a partir de 49 datasets, realizamos o **Teste Estatístico de Wilcoxon** para amostras emparelhadas.

Este teste não paramétrico foi escolhido por ser robusto e adequado para comparar duas amostras relacionadas (emparelhadas por dataset) sem assumir uma distribuição normal dos dados. Investigou-se se o SVM Final demonstra um desempenho (G-Mean) significativamente superior ao SVM Inicial - teste unicaudal. Um p-valor inferior ao nível de significância (α=0.05) foi considerado como evidência de uma diferença estatisticamente significativa.

- O resultado obtido foi:
    - ***Estatística: 331.0***
    - ***p-valor: 0.0083906520477131 03***
    - ***Há diferença estatística significativa entre os modelos***.

Este resultado permite concluir, com um nível de confiança de 95%, que o SVM modificado apresenta um desempenho (medido pelo G-Mean) significativamente superior ao do SVM original. Este ganho pode ser atribuído às modificações introduzidas no algoritmo, que o tornaram mais sensível à classe minoritária em contextos de desbalanceamento.

Assim, as propostas de modificação ao algoritmo SVM tiveram impacto positivo real e estatisticamente validado, reforçando a sua utilidade em cenários onde o equilíbrio entre classes é crítico.

CONCLUSÕES

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

- Este trabalho demonstrou que o algoritmo SVM, na sua versão original, não está preparado para lidar eficazmente com situações de desbalanceamento de classes, uma limitação crítica em muitos cenários do mundo real. Ao tratar todos os erros de forma igual, o modelo tende a favorecer a classe maioritária, comprometendo seriamente a capacidade de identificar corretamente a classe minoritária — muitas vezes a mais relevante.

- Foram propostas e implementadas modificações diretamente no núcleo do algoritmo (SMO), com o objetivo de aumentar a influência da classe minoritária durante o processo de otimização. Através de estratégias como a penalização diferenciada, margens assimétricas e ajustes internos de bias, foi possível reequilibrar o foco do SVM.

- A eficácia destas alterações foi validada empiricamente com recurso à média e mediana do G-Mean e ao desvio padrão, bem como a diferentes gráficos de comparação e, por fim, confirmada estatisticamente com o Teste de Wilcoxon, que revelou uma melhoria significativa no desempenho do SVM Modificado.

- Estes resultados reforçam que intervenções diretas no algoritmo são essenciais para obter classificações mais justas e eficazes em cenários de desbalanceamento de classes.

CONCLUSÕES

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Apesar das melhorias introduzidas no SVM original mostrarem ganhos claros em cenários de desbalanceamento, há ainda espaço para aprofundar e generalizar estas propostas, de forma a torná-las mais robustas e aplicáveis a outros contextos.

- Avaliar impacto em variantes do SVM: Adaptar as modificações em contextos multiclasse, para avaliar se os benefícios se mantêm em problemas mais complexos;

- Aplicação a diferentes algoritmos: Generalizar os princípios testados (margens assimétricas, bias ajustado, penalização diferenciada) para outros modelos, como árvores de decisão ou redes neuronais;

- Otimização Automática de Hiperparâmetros: A seleção dos coeficientes de penalização diferenciada e outros parâmetros foi feita manualmente. Poderá automatizar-se este processo com algoritmos como Grid Search ou Random Search, de forma a maximizar o G-Mean de forma sistemática.

- Testes com Outros Tipos de Kernel: O foco deste trabalho foi essencialmente no kernel Linear, futuras experiências com kernels não lineares (por exemplo, RBF, Polinomial ou sigmoid) permitirão avaliar a robustez das alterações propostas em espaços de maior complexidade.

- Aplicação em Domínios Reais: A validação em datasets públicos é fundamental, mas a aplicação em domínios reais — como deteção de fraude, diagnóstico médico ou spam — permitirá avaliar o impacto prático e ético de melhorar a sensibilidade à classe minoritária.

REFERÊNCIAS

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Para nos ajudar na realização deste projeto, recorremos a:

- Github disponibilizado
  - https://github.com/rushter/MLAlgorithms/tree/master/mla/svm

- Explicação do SMO
  - https://cs229.stanford.edu/materials/smo.pdf
  - https://www.microsoft.com/en-us/research/wp-content/uploads/1998/04/sequential-minimal-optimization.pdf