Análise de Potabilidade de Água Usando Modelos de Aprendizado de Máquina

Guilherme G. S. Silva[1], **Guilherme S. Lopes**[1],
Lucas R. Aragão[1], **Roberto Sérgio R. de Meneses**[1]

Departamento de Computação
Universidade Federal do Ceará, Brazil

{guigalvao, guilhermesousalopes, lucasrodaragao, robertomeneses}@alu.ufc.br

Abstract. *Como trabalho final da Disciplina de Aprendizado de máquina, nesse artigo analisaremos o desempenho de diferetes modelos de ML para a tarefa de classificação no conjunto de dados water potability, retirado do site Kaggle. Mediremos o desempenho baseado em diferentes métricas e ao final discutiremos os resultados.*

1. Introdução

Nos últimos anos observou-se um grande crescimento no uso de inteligências artificiais no cotidiano humano, podendo auxiliar em diferentes aspectos da vida. As aplicações que iniciaram sendo simples detectores de emails spam, hoje escalaram para aplicações de carros autônomos e os modelos de linguagem em larga escala (LLM), como o ChatGPT, da Open AI. Entretanto, ainda existem diversas áreas da sociedade que ainda não usufruem das melhorias trazidas pelo uso de inteligência artificial.

Entre os grandes problemas presentes na pauta mundial, o acesso a água potável é um assunto que sempre está sendo discutido. Segundo dados da Unicef, mais de 2.2 bilhões de pessoas ainda não possuem acesso a fontes de água confiáveis. O objetivo deste trabalho é com base em dados recolhidos e analisados, decidir se uma amostra de água é potável ou não usando diferentes modelos de *Machine Learning* (ML). Isso será feito baseado em um processo de treinamento criterioso e robusto, que será melhor explicado à frente.

Esse trabalho se divide da seguinte forma: A seção 2 apresenta o conjunto de dados utilizado e discute alguns trabalhos anteriores. Seção 3 explica a metodologia do trabalho e da criação dos modelos. Seção 4 discute os resultados dos experimentos. Por fim, a seção 5 conclui o trabalho.

2. Fundamentação teórica

Essa seção descreve brevemente o problema e mostra com mais profundidade os dados usados.

2.1. Apresentação do Conjunto de dados escolhido

O dataset escolhido foi o Water Quality, facilmente encontrado no site Kaggle[1]. Nosso dataset contém 3276 tuplas, em que cada uma representa uma amostra de água de um

[1]https://www.kaggle.com/datasets/adityakadiwal/water-potability

Atributo	Descrição
Ph	Nível de acidez da água
Hardness	Indica a quantidade de minerais da água
Solids	Quantidade de sólidos dissolvidos na água
Chloramines	Quantidade de derivados de amônia na água
Sulfate	Quantidade de sulfatos dissolvidos na água
Conductivity	Condutividade da água
Organic carbon	Quantidade de carbono orgânico presente na água
Trihalomethanes	Quantidade de trilometanos
Turbidity	Mede a quantidade de luz emitida pela água

Tabela 1. Descrição dos atributos

Atributo	Tipo	Média	Mediana	#valores nulos
Ph	Numérico	$7,08$	$7,03$	491
Hardness	Numérico	$196,36$	$196,96$	0
Solids	Numérico	$22014,09$	$20927,83$	0
Chloramines	Numérico	$7,12$	$7,13$	0
Sulfate	Numérico	$333,77$	$333,07$	781
Conductivity	Numérico	$426,20$	$421,88$	0
Organic carbon	Numérico	$14,28$	$14,21$	0
Trihalomethanes	Numérico	$66,39$	$66,62$	162
Turbidity	Numérico	$3,96$	$3,95$	0

Tabela 2. Informações sobre os atributos do conjunto de dados Water potability

local diferente. Ao todo são 10 atributos, sendo 9 desses características da água retirada e o último atributo sendo o rótulo daquela amostra, em que 0 representa que a amostra não é apropriada para consumo e 1 indica que aquela água é potável.

Os demais atributos são *Ph*,*Hardness*, *solids*, *Chloramines*, *Sulfate*, *Conductivity*, *Organic Carbon*, *Trihalomethanes*, *Turbidity*. Todos tratam de aspectos físico ou químicos da água, além disso todos são representados por valores reais positivos. A tabela 1 traz uma breve descrição de cada atributo. Nosso objetivo aqui não é se aprofundar no assunto, por isso não iremos além em relação a importância de cada atributo no problema em si, por exemplo saber que uma quantidade alta de trilometanos pode indicar a presença de substâncias cancerígenas ou que um PH muito baixo pode causar diversos problemas ao organismo humano.

Por outro lado, a tabela 2 traz aspectos que podem ser mais importantes para o âmbito de aprendizado de máquina, que serão melhores aprofundados na seção 3. Por fim, é importante salientar que os valores faltantes de cada coluna foram substituídos pelas medianas destas colunas.

2.2. Trabalhos relacionados

São inúmeros os trabalhos que tratam do problema de água potável e usam de algoritmos de aprendizado de máquina para solucionar estes. Dentre alguns dos exemplos, podemos citar [Kaddoura, 2022], [Patel et al., 2023] e [Poudel et al., 2022], todos os três usam o

Métrica	Descrição
Acurácia	Percentual de acertos gerais do modelo, considerando todos os verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos.
Precisão	Proporção de exemplos corretamente classificados como positivos em relação ao total de exemplos classificados como positivos.
Revocação	Proporção de exemplos positivos corretamente identificados em relação ao total de exemplos realmente positivos.
F1-score	Média harmônica entre precisão e revocação, usada quando há um balanço entre as duas métricas.
Curva ROC	Gráfico que mostra a taxa de verdadeiros positivos (sensibilidade) versus a taxa de falsos positivos, permitindo avaliar o desempenho do modelo.
Curva PR	Gráfico que mostra a relação entre precisão e revocação em diferentes limiares, especialmente útil quando há desbalanceamento de classes.

Tabela 3. Descrição das métricas usadas

mesmo conjunto de dados usados neste trabalho. Um ponto negativo que é as etapas de treinamento e avaliação dos modelos não são bem explicadas.

3. Metodologia

3.1. Algoritmos usados

O problema escolhido de predição de potabilidade de água é um clássico problema de classificação binária. Tendo isso em vista, escolhemos os modelos mais robustos entre aqueles vistos durante a disciplina de aprendizado de máquina. Para a criação destes usamos as implementações da biblioteca Scikit-Learn do Python.

O primeiro escolhido foi o ***MultiLayer Perceptron - MLP***. O motivo da escolha é simples, redes neurais artificiais são de longe a família de modelos mais comentada nos últimos anos, e deixá-las de foras de um trabalho de classificação como este seria um grande erro. Além deles usamos um modelo de ***Random Forest - RF*** e outro de ***Support Vector Machine - SVM***, por entendermos que seriam os dois mais robustos dos modelos restantes. Todos os modelos foram vistos durante a disciplina de aprendizado de máquina.

Além disso, utilizamos um modelo de ***Principal Component Analysis - PCA***. A ideia aqui não seria prever diretamente a categoria que a amostra faz parte ou tentar reduzir a dimensionalidade do problema visando o treinamento dos modelos. O PCA foi usado aqui para fins de visualização, nós entendemos que é interessante tentar reduzir o problema para 2 dimensões e tentar observar se os dados são visivelmente separáveis ou não, quando se aplica essa técnica.

3.2. Métricas

Decidimos usar uma gama de diferentes métricas para avaliar os diferentes aspectos dos modelos. Por se tratar de um problema de classificação usamos as métricas de acurácia, precisão, revocação, f1-score, curva roc e curva *precision-recall*. A tabela 3 detalha as métricas usadas.

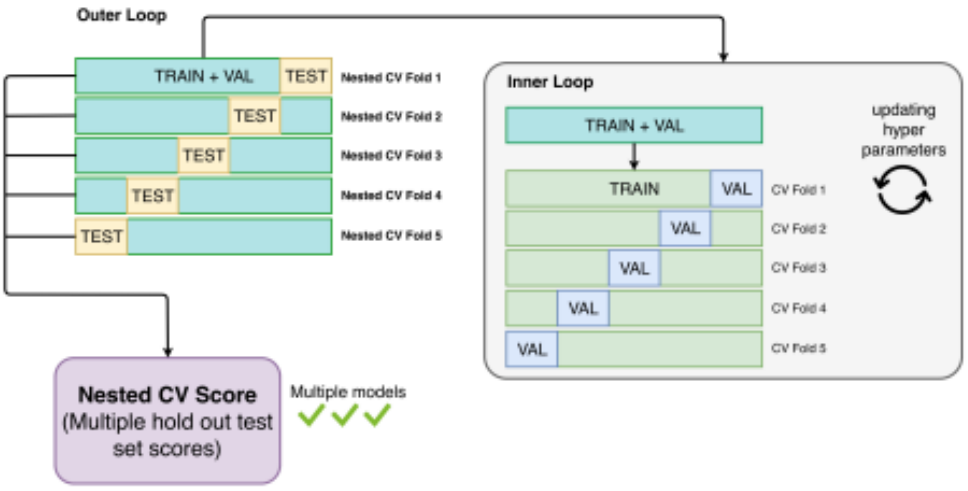


Figura 1. Exemplo da validação cruzada aninhada

3.3. Treinamento dos modelos

O treinamento dos modelos é, obviamente, a parte mais importante deste trabalho. Para isso usamos o processo de validação cruzada aninhada (***nested cross validation***). O ponto aqui é testar diferentes conjuntos de hiper parâmetros para cada modelo e assim decidir qual obteve o melhor desempenho. A decisão é baseada no desempenho médio de cada conjunto ao longo das diferentes partições, como é visto na figura 1. Para o nosso problema entendemos que o pior caso possível é quando uma amostra que não pode ser destinada ao consumo humano é rotulada como potável, ou seja os falsos positivos. Decidimos ter tuas abordagens diferentes, a primeira usa a precisão como métrica de escolha e a segunda usa o f1-score.

Quando utilizamos a precisão como métrica de decisão, os modelos gerados tendem a ser mais cautolosos em relação a previsão de positivos, com isso temos um alto número de falsos negativos. Por outro lado, ao usar o F1-score como métrica de escolha, os modelos tendem a prever mais valores positivos, pois também é levado em conta o número de falso negativos. Ao longo das tabelas de resultados que virão a seguir, usaremos a sigla **ES-P** para nos referir aos *scores* relativos ao modelos criados usando a precisão como métrica de escolha no grid search e, **ES-F** para os que utilizarem *F1-score*.

Para cada partição do loop interno, todos os dados são normalizados baseados no conjunto de treino. Após isso, para cada conjunto de hiper parâmetros, um modelo é criado, treinado e faz a predição do conjunto de validação; sua predição é comparada com os resultados reais e o valor da métrica de escolha é armazenado para que a média do seu desempenho seja calculada ao final do loop interno. Ao final do loop interno será retornado o modelo com o melhor desempenho, este, por sua vez, será retreinado, agora com o conjunto de treino completo daquela partição e fará a predição do conjunto de teste. É importante ressaltar que os valores também são normalizados em cada loop externo, sempre baseado nos dados de treino.

Ao final do loop externo, teremos a média do desempenho dos modelos campeões de cada partição. O objetivo desse tipo de validação é saber o desempenho de cada família de modelo.

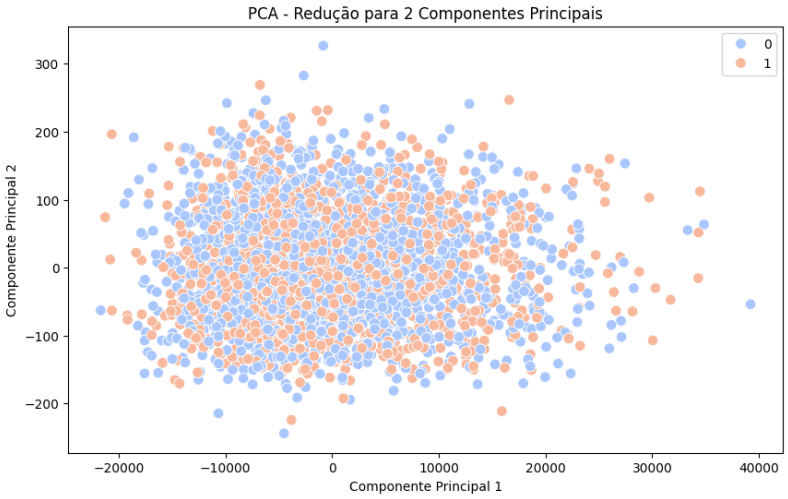


Figura 2. PCA resultante

Esse processo será feito para as 3 famílias de modelos escolhidas, MLP, Random Forest e SVM. E ao final desse processo meticuloso, teremos a família "campeã", ou seja, aquela que teve o melhor desempenho médio. Após isso, é feito um processo de grid search similar ao *inner loop* da validação cruzada, para que o melhor conjunto de hiperparâmetros seja escolhido.

O último passo dos nossos experimentos é a avaliação do modelo com o melhor conjunto de hiperparâmetros daquela família. O modelo final será testado baseado em uma divisão simples de treino e teste, com os dados de teste sendo normalizados baseados nos dados de treino. Ao final de tudo isso teremos uma avaliação completa do modelo, usando todas as métricas citadas anteriormente.

4. Experimentos

Nessa seção serão discutidos os resultados dos experimentos envolvendo a análise de componentes principais e os modelos usados na validação cruzada aninhada.

4.1. Análise de PCA

A análise do PCA nesse trabalho teve como fim concluir se seria visivelmente claro a distinção entre as duas classes num plano bidimensional. O processo foi simples, utilizamos a implementaçao do sklearn e colocamos o número de componentes como 2 e fizemos o *fit* dos dados normalizados. O *plot* resultante pode ser visto na figura 2.

Os resultados mostram que tal separação não é possível. As amostras de água potável e imprópria se misturam, impossibilitando qualquer tentativa de delimitar se um dado é potável ou não, de maneira visual.

4.2. Resultados Validação cruzada aninhada

Tivemos duas abordagens diferentes para encontrar os melhores conjuntos de hiperparâmetro, na primeira usamos a precisão como métrica de avaliação e, na segunda utilizamos o f1-score. Entretanto para ambos os casos, ao final da validação cruzada aninhada retornamos o valor de precisão média daquela família de modelos.

Modelo	Precisão ES-P	Precisão ES-F
MLP	$0,61$	$0,56$
Random forest	$0,67$	$0,62$
SVM	$1,00$	$0,57$

Tabela 4. Resultados obtidos na validação cruzada aninhada

Hiperparâmetro	Valores usados
hidden_layer_sizes	[(5,), (10,), (15,), (5,5), (10,10), (15,15)]
activation	[relu]
solver	[adam, sgd]
alpha	[0.001, 0.01, 0.1]
learning_rate	[constant, adaptive]
learning_rate_init	[0.01, 0.1]
momentum	[0.5, 0.6, 0.7, 0.8, 0.9]

Tabela 5. Hiperparâmetros MLP

Os resultados das validações cruzadas podem ser encontradas na tabela 4. Na primeira estratégia, o SVM foi o modelo com a melhor precisão, entretanto o número de positivos previstos é extremamente baixo, e portanto possui um altíssimo número de falsos negativos. Na segunda estratégia, o modelo que obteve o melhor desempenho foi o de random forest, aqui o número de previsões é mais balanceado, entretanto o número de falsos positivos cresce bastante.

4.3. Grid Search com Validação Cruzada

Após os resultados gerais, partimos para o grid search com validação cruzada. Aqui, o processo é igual ao loop interno da validação aninhada, e assim como na etapa anterior, testamos as duas abordagens, tanto com precision sendo a métrica de escolha quanto usando o f1-score para tal função. O desempenho do modelo campeão de cada classe de algoritmos será discutido a seguir para as duas estratégias.

O conjunto de hiperparâmetros usados no grid search do MLP estão presentes na tabela 5. Para a ES-P, obtivemos como melhor conjunto de hiperparâmetros, { *'mlp_activation': 'relu', 'mlp_alpha': 0.01, 'mlp_hidden_layer_sizes': (10,), 'mlp_learning_rate': 'adaptive', 'mlp_learning_rate_init': 0.1, 'mlp_momentum': 0.8, 'mlp_solver': 'adam'* }. Já para a ES-F, o conjunto encontrado foi, {*'mlp_activation': 'relu', 'mlp_alpha': 0.01, 'mlp_hidden_layer_sizes': (15, 15), 'mlp_learning_rate': 'constant', 'mlp_learning_rate_init': 0.01, 'mlp_momentum': 0.7, 'mlp_solver': 'adam'*}.

No caso do random forest os hiperparâmetros usados no grid search estão na tabela 6. Para a estratégia com usando a precisão para a escolha, os valores escolhidos pelo processo de grid search foram, {*classifier_criterion: entropy, classifier_max_depth: 8, classifier_min_samples_leaf: 1, classifier_min_samples_split: 10, classifier_n_estimators: 100*}. Na segunda estratégia os valores são, {*classifier_criterion: gini, classifier_max_depth: None, classifier_min_samples_leaf: 1, classifier_min_samples_split: 5, classifier_n_estimators: 100*}.

Hiperparâmetro	Valores usados
n_estimators	[100, 200, 300]
max_depth	[4, 6, 8, 10, None]
min_samples_split	[2, 5, 10]
min_samples_leaf	[1, 2, 4]
criterion	[gini, entropy]

Tabela 6. Hiperparâmetros Random Forest

Hiperparâmetro	Conjunto usado
C	$[2^{-5}, 2^{-4}, 2^{-3}, ..., 2^{15}]$
gamma	$[2^{-15}, 2^{-14}, 2^{-13}, ..., 2^{3}]$

Tabela 7. Hiperparâmetros SVM

Métrica	ES-P	ES-F
Acurácia	$0,67$	0.64
F1-score	$0,41$	$0,47$
Revocação	$0,30$	$0,41$
Precisão	$0,66$	$0,56$
Roc AUC	$0,66$	$0,61$
PR AUC	$0,60$	$0,54$

Tabela 8. Resultados finais MLP

Os hiperparâmetros usados na criação do SVM estão registrados na tabela 7. O svm usando ES-P resultou nos seguintes hiperparâmetros, { *'svm_C': 0.5, 'svm_gamma': 0.015625* }. Por fim, o conjunto escolhido na estratégia que usa o f1-score foi, { *'svm_C': 4096, 'svm_gamma': 0.03125* }.

4.4. Resultados obtidos usando os melhores modelos de cada classe

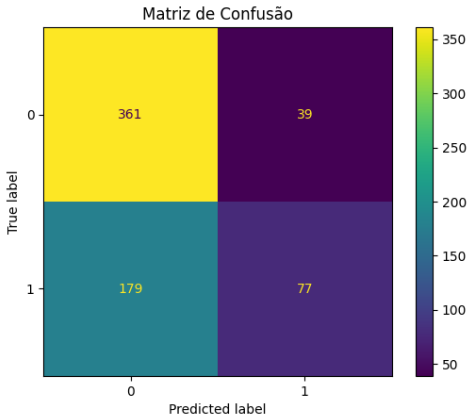
Após recolher os melhores hiperparâmetros, o algoritmo usado roda uma última vez cada modelo, com o melhor conjunto de hiperparâmetros e obtém as métricas de avaliação citadas na tabela 3.

Os resultados do MLP podem ser vistos na tabela 8. Como esperado, cada modelo obteve melhor desempenho nas suas métricas de escolha. Ambos modelos tiveram comportamentos muito similares para acurácia, f1-score, área da curva roc e área da curva precision recall. As maiores diferenças foram na revocação, o que pode ser explicado pelo alto número de falsos negativos, como pode ser visto na figura 3, em que o modelo que se baseia na precisão (MLP ES-P) prevê um número menor de positivos.

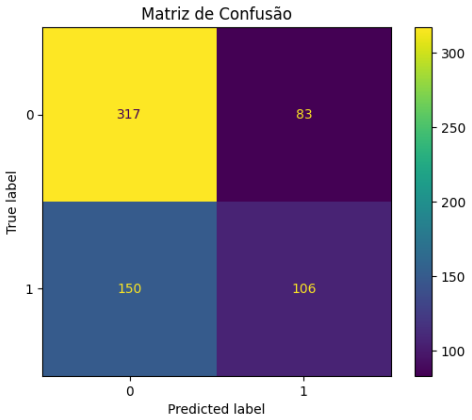
Os resultados dos modelos de random forest estão na tabela 9. Diferentemente do

Métrica	ES-P	ES-F
Acurácia	$0,65$	0.66
F1-score	$0,26$	$0,43$
Revocação	$0,16$	$0,32$
Precisão	$0,80$	$0,64$
Roc AUC	$0,65$	$0,66$
PR AUC	$0,58$	$0,59$

Tabela 9. Resultados finais Random Forest

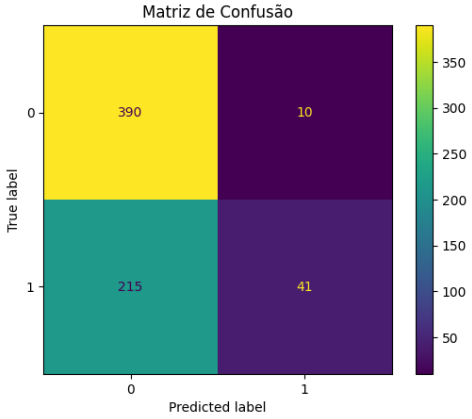


(a) Matriz de confusão para o MLP ES-P

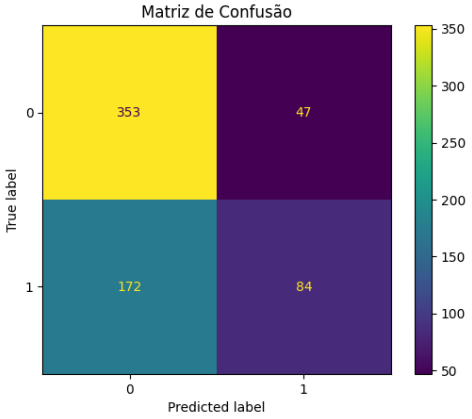


(b) Matriz de confusão para o MLP ES-F

Figura 3. Matriz de confusão para os modelos de MLP para as duas abordagens



(a) Matriz de confusão para Random Forest ES-P



(b) Matriz de confusão para o Random Forest ES-F

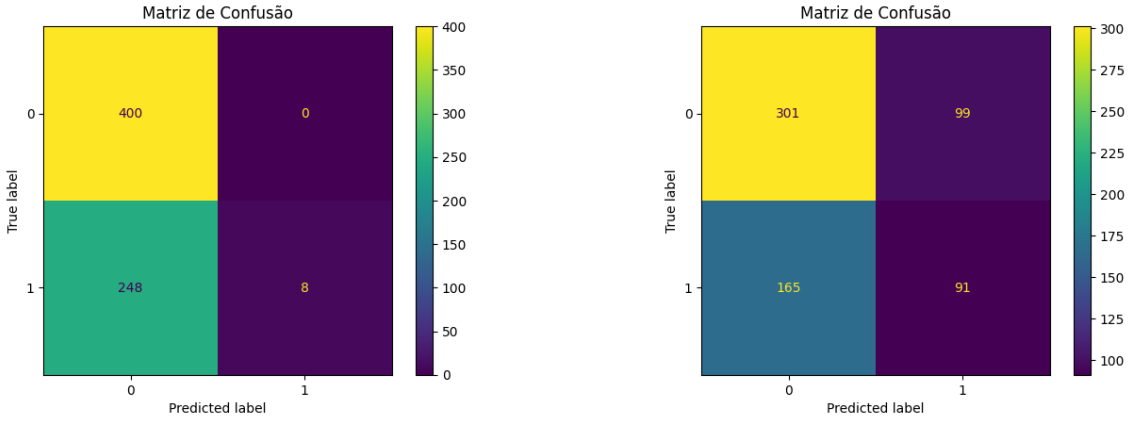
Figura 4. Matriz de confusão para os modelos de Random Forest para as duas abordagens

MLP existe uma diferença considerável entre algumas métricas, a revocação e a precisão possuem valores bastantes distintos, no caso da primeira o valor do ES-P é apenas metade da ES-F, indicando um comportamento conservador, ou seja, poucas amostras são previstas como positivas, entretanto quando se prevê uma instância como positiva, existe uma chance alta de ela realmente ser positiva, tendo em vista o alto valor para a precisão. A figura 4 explicita bem isso. Nela é possível observar o baixo número de valores positivos previstos, que resultam em uma alta precisão, que pode mascarar o desempenho ruim do modelo para os demais casos.

Por fim, os resultados dos modelos de SVM estão sintetizados na tabela 10. Aqui temos um exemplo ainda pior do que foi visto nos modelos de random forest. O modelo escolhido pela estratégia que usa precisão como métrica de escolha tem um comportamento extremamente peculiar, o modelo não errou nenhuma vez quando previu que o valor era positivo, entretanto ele dificilmente previu que uma instância fosse positiva, algo que pode ser visto nos valores de f1-score e revocação, que beiram o zero. Mais

Métrica	ES-P	ES-F
Acurácia	$0,62$	0.59
F1-score	$0,06$	$0,40$
Revocação	$0,03$	$0,35$
Precisão	$1,00$	$0,47$
Roc AUC	$0,67$	$0,59$
PR AUC	$0,60$	$0,50$

Tabela 10. Resultados finais SVM



(a) Matriz de confusão para SVM ES-P

(b) Matriz de confusão para SVM ES-F

Figura 5. Matriz de confusão para os modelos de Random Forest para as duas abordagens

uma vez as matrizes de confusão podem mostrar bem esse comportamento. Na figura 5, é possível observar bem a situação, apenas 8 amostras foram preditas como potáveis. Resultados como este mostram o perigo em se basear somente em uma métrica e como pode ser danoso para um trabalho de aprendizado de máquina.

5. Conclusão

Ao final deste trabalho, concluímos que nenhum dos modelos obteve o desempenho que esperávamos no início dele. Os resultados em geral foram baixos, apesar de, no nosso modo de ver, a metodologia aplicada ter sido rigorosa e robusta. Para trabalhos futuros, seria interessante testar com uma variedade ainda maior de modelos de aprendizado de máquina, algo que não foi possível devido às limitações de hardware que tivemo. Aplicar todo o processo de treinamento para mais uma família de modelos poderia custar muito tempo.

Entretanto, levamos alguns ensinamentos deste trabalho. O principal deles é o perigo de tentar se basear somente na métrica do problema específico e como isso pode mitigar o desempenho geral de um modelo. No svm, por exemplo, obtivemos um salto de 0,36 no f1-score apenas por usá-lo como métrica de decisão dentro do grid search.

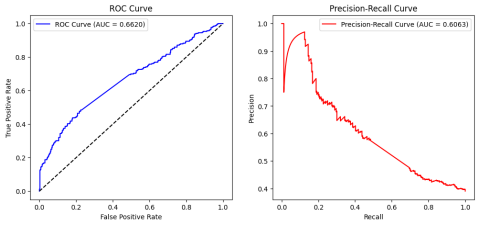
Referências

S. Kaddoura. Evaluation of machine learning algorithm on drinking water quality for better sustainability. *Sustainability*, 14(18):11478, 2022.

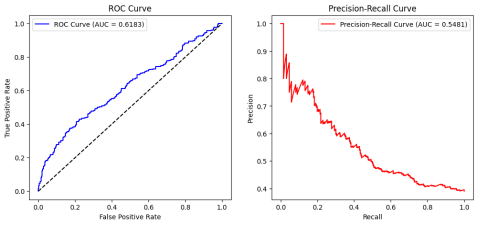
S. Patel, K. Shah, S. Vaghela, M. Aglodiya, and R. Bhattad. Water potability prediction using machine learning. 2023.

D. Poudel, D. Shrestha, S. Bhattarai, and A. Ghimire. Comparison of machine learning algorithms in statistically imputed water potability dataset. *Journal of Innovations in Engineering Education*, 5(1):38–46, 2022.

A. Curvas Roc e precision recall

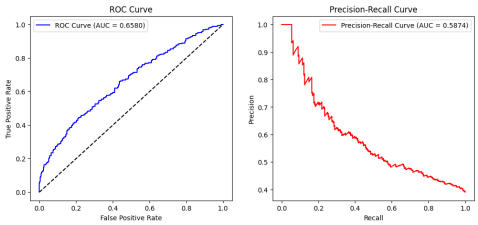


(a) Curvas para o modelo MLP ES-P

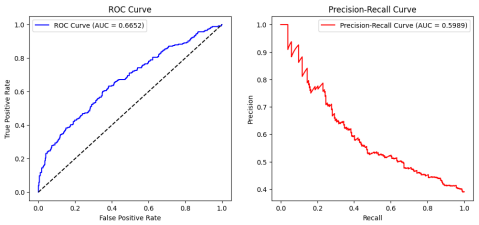


(b) Curvas para o modelo MLP ES-F

Figura 6. Curvas Roc e Precision recall para o MLP

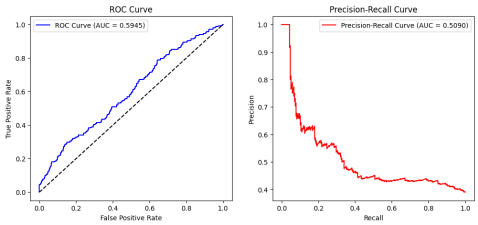


(a) Curvas para o modelo Random Forest ES-P

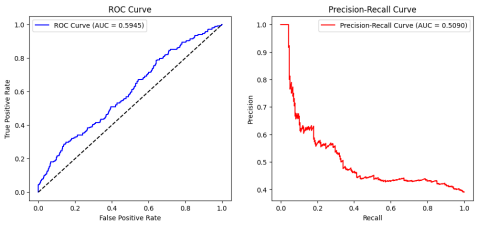


(b) Curvas para o modelo Random Forest ES-F

Figura 7. Curvas Roc e Precision recall para Random Forest



(a) Curvas para o modelo SVM ES-P

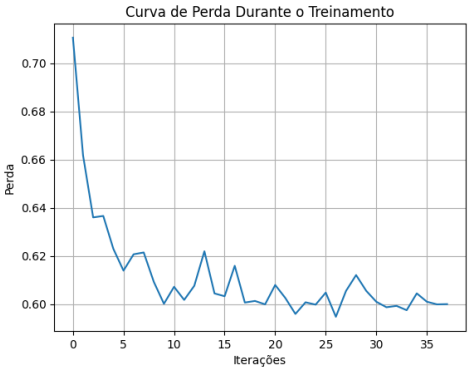


(b) Curvas para o modelo SVM ES-F

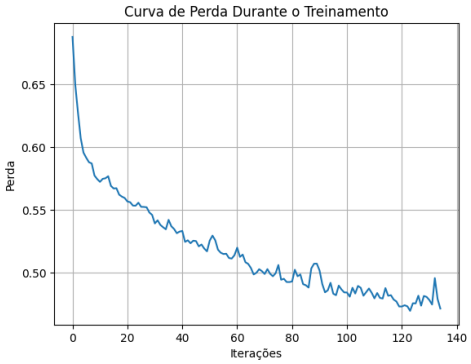
Figura 8. Curvas Roc e Precision recall para SVM

B. Curvas de aprendizado

Não foi explicitado durante o texto, mas todos os modelos convergiram. E aqui seguem as curvas de aprendizado dos dois modelos MLP finais, com o melhor conjunto de hiperparâmetros para cada estratégia.



(a) Curvas para o modelo MLP ES-P



(b) Curvas para o modelo MLP ES-F

Figura 9. Curvas de perda dos modelos finais