

# Otimização evolutiva multiobjetivo aplicadas a parametrização de um problema de classificação por aprendizagem de máquina

**Igor Morgado**

4 de Novembro de 2020

## **Resumo**

Métodos de aprendizagem de máquina normalmente necessitam de uma grande quantidade de dados para atingir resultados de boa qualidade. Estes imensos conjuntos de dados também apresentam uma desvantagem, a necessidade de uma vasta quantidade de recursos computacionais para analisar e calcular os pesos para serem utilizados na fase de generalização. Por outro lado, conjuntos de dados maiores, com mais características e amostras nem sempre levam a modelos melhores, além disso há um limite de saturação onde mais dados atrapalham mais que ajudam normalmente chamado de modelo super ajustado. Neste trabalho propomos o uso de uma otimização multi-objetivo evolutiva para estudar dois objetivos conflitantes: precisão e quantidade de características no dado de treinamento. Usando o *Non Dominated Sorting Genetic Algorithm* (NSGA-II), fomos capazes de fazer a busca no conjunto de decisão, não somente extraíndo a cardinalidade ótima para o conjunto de características como também escolhendo quais características melhor representam o dado relativo com a acurácia. Usando este método fomos capazes de reduzir o espaço de busca de  $10^{60} - 1$  possibilidades e encontramos uma solução aceitável. Esta técnica então foi aplicada ao conjunto de dados de teste de Sonares para classificar duas classes diferentes, resultando uma melhora de 25% comparado com a busca local ou métodos de força bruta. O resultado final é um conjunto de soluções dada

pela primeira frente de um conjunto de Pareto que pode ser utilizada por um especialista ou uma algoritmo especialista.

Palavras chave: Métodos evolutivos, algoritmos genéticos, aprendizagem de máquina, redes neurais, Perceptron multi camadas, meta heurísticas, NSGA-II.

## 1 Introdução

### 1.1 Problemas de classificação

Um problema de classificação em aprendizado de máquina consiste resumidamente em fazer com que um modelo consiga identificar um padrão sem que seja necessário um especialista introduzir qualquer tipo de conhecimento sobre o problema sendo tratado. Um dos exemplos clássicos é apresentar uma grande quantidade de dígitos escritos a mão (conforme a Figura 1), corretamente catalogados e uma vez apresentado uma quantidade suficiente de amostras o método é capaz de dada uma amostra inédita o modelo é capaz de corretamente classificar a qual classe esta nova amostra pertence.



Figura 1: Exemplo de entradas do conjunto de dados MNIST

A primeira etapa do processo é chamada de **aprendizagem**, o momento no qual são apresentados diversos exemplos para o modelo. O aprendizado de uma criança pode ser utilizado como uma metáfora desta etapa, onde diuturnamente

reforçamos para uma criança o nome dos objetos, congratulando quando ela corretamente identifica e corrigindo quando há uma identificação equivocada.

A segunda etapa do processo é chamada de **generalização**. Nesta etapa, o modelo já treinado é capaz de identificar com um certo grau de acurácia que uma dada amostra pertence a uma classe ou outra. No caso dos dados MNIST, um novo dígito feito a mão é inserido no modelo e este retorna a classe que este dígito pertence, neste modelo as classes são os dígitos de 0 à 9.

É importante perceber que se um modelo é treinado com o conjunto de dados do MNIST, ele nada sabe sobre outras entidades, como gatos e cachorros. Se apresentarmos uma foto de um cachorro ele vai retornar uma das 10 classes treinadas, ou seja, ele pode retornar como resposta a classe '3' (o número 3 aqui foi escolhido arbitrariamente).

Existem diversas técnicas e modelos que podem ser aplicados aos problemas de classificação em aprendizado de máquina. Cada um destes modelos tem suas características e usos saindo-se melhor quando aplicados em um conjunto de dados ou outro. Nenhum modelo é superior em todas as situações (ou pior em todas as situações), desta forma, é necessário entender os tipos de problemas a serem resolvidos, os modelos e suas respectivas características.

## 1.2 Hiper Dimensionalidade

Na maior parte dos casos a entrada do método de aprendizagem de máquina é um vetor  $n$  dimensional, onde  $n$  é normalmente grande, de dezenas à milhões.

Um grande número de dimensões leva invariavelmente a necessidade de um maior poder computacional e um maior tempo de execução, além disso em muitos casos nem todos os atributos contêm dados relevantes para o modelo de classificação. Apesar de contra intuitivo, em muitos casos, uma maior quantidade de atributos pode levar a uma menor acurácia de predição do modelo tanto na fase de treinamento quanto na fase de generalização.

Buscamos então reduzir o número de atributos de entrada para elaborar um modelo mais simples, onde os atributos de entrada possuam maior relevância e melhor razão sinal ruído levando a geração de um método com maior capacidade de generalização com menor tempo computacional.

O problema que decidir quais são os melhores atributos é um problema complexo, o espaço de busca é da ordem de  $2^n - 1$  onde  $n$  é o número de atributos existentes no dado. No nosso caso específico  $n = 60$ . Mesmo que consideremos que cada teste tome somente 1 segundo, fazer uma busca de todas as possíveis combinações dos atributos necessitaria uma quantidade de poder computacional não disponível mesmo em um supercomputador.

É possível então utilizar métodos heurísticos para realizar a busca do melhor conjunto de atributos. Este método nos permitirá então fazer uma busca através de tentativas *espertas* encontrar um conjunto de soluções admissíveis ao nosso problema.

## 2 Metodologia

### 2.1 Conjunto de Dados de Sonar

Neste trabalho escolhemos utilizar o conjunto de dados **Sonar, Minas vs. Rochas**.(DUA e GRAFF 2017). Este conjunto de dados é obtido de sensores reais que foram aplicado sobre uma mina e uma rocha em diferentes ângulos de incidência e variáveis frequências sob diversas condições.

Cada registro possui 60 atributos, relacionado com um dado ângulo de incidência ou condição, não há registro no conjunto de dados qual o ângulo ou a condição este atributo esta relacionado. Em cada um destes atributos há um valor numérico entre 0.0 e 1.0 com precisão de 4 casas decimais relacionado a energia refletida integrada sobre uma dada quantidade de tempo fixa. Desta forma um registro é composto de 60 (sessenta) valores numéricos e 1 (uma) classe que pode ser **Rocha** ou **Mina**.

Este conjunto de dado possui um total de 208 registros, sendo destes 111 minas e 97 rochas.

Podemos ver na figura 2 que resposta ao estímulo das rochas e das minas são na média similares, o desvio padrão é menor nos extremos das bandas e maior nas faixas centrais. A média e a variância dos dados se sobrepõem em todo espectro tornando o problema de classificação bastante complexo.

Este conjunto de dados foi organizado por GORMAN e SEJNOWSKI 1988 para

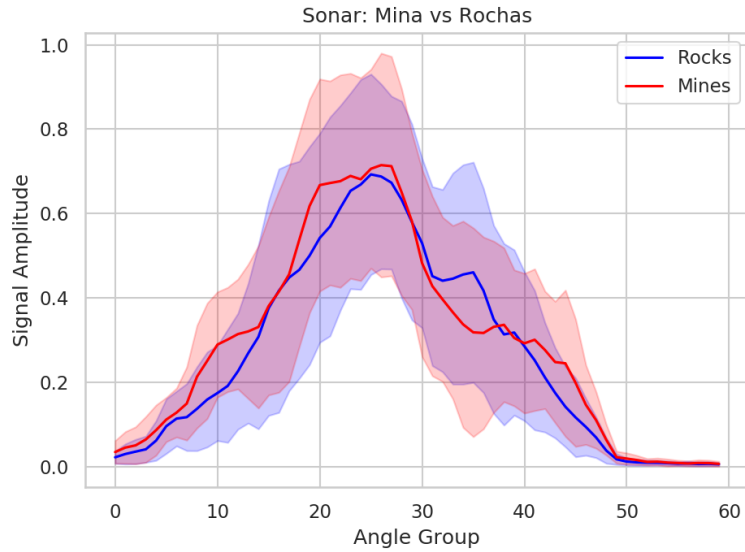


Figura 2: Classes: Média e desvio padrão por atributo

o estudo de redes neurais aplicadas a um problema de classificação. Optamos utiliza-lo para este trabalho pois possui uma quantidade razoável de atributos que é necessário para nossa proposta e um conjunto de registros pequeno, reduzindo o tempo de execução dos testes.

## 2.2 Algoritmo Genético

Dentre as abordagens heurísticas, existem os algoritmos genéticos, estes são uma metáfora, baseando-se na teoria da seleção natural de Charles Darwin. Neste modelo cada solução é um indivíduo que possui um conjunto de genes, estes genes representam um parâmetro da solução. Um conjunto de diferentes indivíduos que são ao mesmo tempo diferentes soluções é chamada de população conforme mostrada na tabela 1.

Uma população pode conter uma quantidade arbitrária e normalmente larga de indivíduos. Os genes estão relacionados com o problema que procuramos solucionar, por exemplo, genes binários podem estar relacionados com a ativação ou não de uma determinada propriedade.

No método genético há uma função custo, normalmente chamada de *fitness*

Indivíduo 1	0	1	1	0	1	0
Indivíduo 2	1	1	1	1	1	0
Indivíduo 3	0	1	1	1	1	0
Indivíduo 4	0	0	1	0	1	0

Tabela 1: Uma população com 4 indivíduos. Cada indivíduo possui um cromossomo com 6 genes binários.

(aptidão), que associa um indivíduo a um valor real que procuramos otimizar.

De uma população os indivíduos com melhores índices de aptidão são escolhidos e então realizam um cruzamento genético (normalmente dois a dois) onde parte do cromossomo de um indivíduo é trocado com o cromossomo do outro, conforme mostrado na tabela 2

Indivíduo 1	0	1	1	0	1	0
Indivíduo 2	1	1	1	1	1	0
Filho 1	0	1	1	1	1	0
Filho 2	1	1	1	0	1	0

Tabela 2: Seleção de dois indivíduos e o respectivo cruzamento

Estes novos indivíduos possuem a probabilidade de sofrer uma mutação em algum gene, no caso binário isso significa trocar o valor do bit. Este procedimento é repetido para cada um dos pares escolhidos para produzir a nova geração, estas novos indivíduos formarão a nova população e a etapa se repete por uma quantidade de ciclos (ou até uma condição de parada ser encontrada).

Este método permite realizar a busca em um espaço sem prender-se a pontos críticos locais de métodos analíticos clássicos como o gradiente descendente.

## 2.3 Problemas multi objetivo

Os problemas de aprendizado de maquina são normalmente multi objetivo, isto é melhorar um dos objetivos pode levar a piorar um outro objetivo esperado, devemos então encontrar uma maneira de balancear estes objetivos e encontrar um conjunto com as melhores soluções possíveis.

No nosso caso particular queremos encontrar um modelo o mais simples possível, isto é, com o menor número de atributos, por outro lado queremos que este modelo tenha a melhor acurácia possível.

Quanto mais atributos nosso modelo tem mais informação ele possui para realizar o treinamento, que leva em teoria a modelos mais precisos, infelizmente se muitos dados forem utilizados, corre-se o risco do modelos ficar super ajustado a um caso específico perdendo sua capacidade de generalização que reflete diretamente na acurácia.

Devemos então fazer uma escolha entre simplicidade e robustez. Uma menor quantidade de atributos leva a um modelo mais simples e mais fácil de ser compreendido. Por outro lado queremos uma grande acurácia, neste caso mais informação dada o modelo pode ser uma vantagem.

Desta forma não é a simples tarefa de otimizar o erro do treinamento, mas também reduzir a quantidade de atributos a serem utilizados.

A figura 3 apresenta um grupo de soluções possíveis, no eixo vertical temos a quantidade de atributos (neste gráfico não é possível saber quais atributos foram selecionados) e no eixo horizontal a acurácia do conjunto proposto. Por exemplo temos 4 (quatro) soluções que utilizam 24 atributos e com diferentes acurácias, equivalentemente, temos 3 (três) soluções com 0.79 de acurácia dados diferentes quantidades de atributos. Estes casos individualmente são simples de serem decididos, pois temos fixado um objetivo e podemos escolher o outros. Por exemplo é evidente que para o problema com 24 atributos iremos preferir aquele que possui maior acurácia, da mesma forma, fixando a acurácia em 0.79, tomamos como melhor solução aquele que possui a menor quantidade de atributos. Mas como comparar por exemplo o resultado com 20 atributos e acurácia 0.73 e o resultado com 24 atributos e acurácia 0.80? A escolha nem sempre é simples.

Esta tomada de decisão é importante pois se pretendemos percorrer o espaço

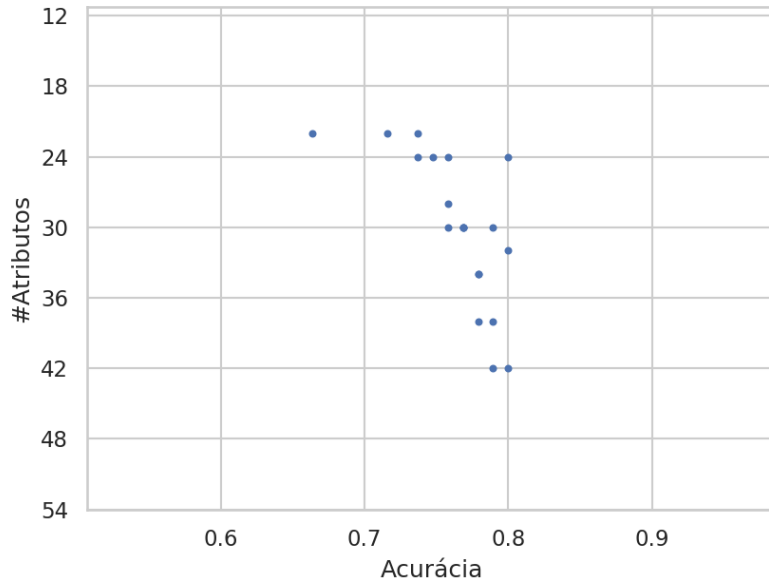


Figura 3: Possíveis soluções multiobjetivo

de soluções admissíveis, devemos encontrar uma forma de ordenar as soluções de para que possamos encontrar os pontos críticos deste espaço.

## 2.4 NSGA-II

Para solucionar o problema multi objetivo, neste trabalho utilizaremos o método NSGA-II desenvolvido por DEB et al. 2002, este método soluciona o problema de complexidade computacional para realizar a busca dentro do espaço de soluções de um problema multi objetivo.

Para solucionar o problema de ordenação de soluções o método utiliza o conceito de *dominância de Pareto* para construir as *frentes de Pareto* conforme mostrado na figura 4.

As frentes de Pareto tornam-se então uma forma natural de ordenar indivíduos em um problema multi objetivo, onde não somente a função de aptidão deve ser levada em conta mas outros fatores. Este critério de decisão então é utilizado para escolher os indivíduos mais aptos a gerarem a nova geração para o algoritmo genético.



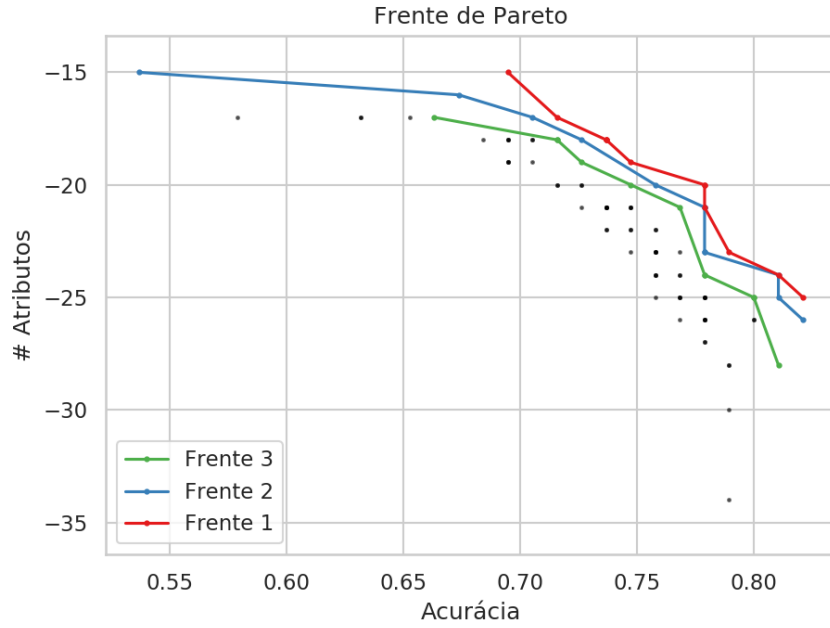


Figura 4: Frentes de Pareto

## 2.5 Modelagem

Nosso conjunto de dados contém informação relativa ao sinal refletido em diversos ângulos sobre uma mina (artefato bélico) e uma rocha, sobre diversas condições. O arquivo que armazena os dados está no formato CSV (*Comma Separated Values*) com 61 colunas e 208 linhas onde cada uma das primeiras 60 colunas é o sinal refletido integrado em um tempo finito e o último a classe a qual o dado pertence que pode ser R (rocha) ou M (mina).

```
0.0200,0.0371,0.0428,0.0207,0.0954,0.0986,0.1539,0.1601,
0.3109,0.2111,0.1609,0.1582,0.2238,0.0645,0.0660,0.2273,
0.3100,0.2999,0.5078,0.4797,0.5783,0.5071,0.4328,0.5550,
0.6711,0.6415,0.7104,0.8080,0.6791,0.3857,0.1307,0.2604,
0.5121,0.7547,0.8537,0.8507,0.6692,0.6097,0.4943,0.2744,
0.0510,0.2834,0.2825,0.4256,0.2641,0.1386,0.1051,0.1343,
0.0383,0.0324,0.0232,0.0027,0.0065,0.0159,0.0072,0.0167,
0.0180,0.0084,0.0090,0.0032,R
```

Listing 1: Exemplo de uma linha do arquivo de conjunto de dados

Nossa proposta é verificar como obter o modelo mais fiel (maior acurácia) e mais simples o possível (menor número de dimensões), desta forma deveremos verificar: a. a menor quantidade possível de dimensões que mantém elevada acurácia; e b. quais são as colunas que resultam no melhor resultado.

Para responder estes critérios, iremos tomar subconjuntos do conjunto de dado completo, estes sub conjuntos serão dados pela redução de um número entre 0 e 59 de colunas do original. Este subconjunto será então submetido a uma função de aptidão para verificar a qualidade deste.

Esta escolha de quais colunas serão parte do subconjunto sera modelado como o indivíduo (do modelo genético), que terá por isso um cromossomo de 60 genes binários. Cada um destes genes está relacionado diretamente com uma coluna do conjunto de dados de sonar. Por exemplo o indivíduo dado por  $[1, 1, 1, 1, 1, \dots, 1, 1, 1]$  representa um modelo onde todas as colunas serão utilizadas. Por outro lado o indivíduo com  $[1, 0, 0, 0, \dots, 0, 0, 1]$  representa um modelo onde somente a primeira e a última coluna serão consideradas.

Geraremos então uma população inicial completamente aleatória (o valor pode ser visto na tabela 3), cada indivíduo será então submetido a função de aptidão que retorna dois resultados conflitantes entre si: a. o escore de acurácia (baseado no modelo de aprendizagem de máquina); b. o número de atributos utilizados.

## 2.6 Função de aptidão

A função de aptidão vai utilizar o subconjunto referente as colunas ativas do indivíduo sendo analisado este subconjunto é dividido em 5 grupos divididos aleatoriamente e estratificados, de forma que as proporções de classes presentes no dado original sejam preservadas em cada um destes subconjuntos (conforme figura 5)

Cada um destes subconjuntos então é aplicado a um modelo de aprendizado de máquina, obtendo 5 resultados de acurácia independentes. Uma média destes é então tomada como o valor da função aptidão que pretendemos maximizar.

Nosso trabalho realizou dois experimentos independentes, um utilizando o método Naive Bayes e outro utilizando o Perceptron Multi Camadas (sigla MLP em inglês), os parâmetros de cada um dos experimentos podem ser vistos na tabela 3

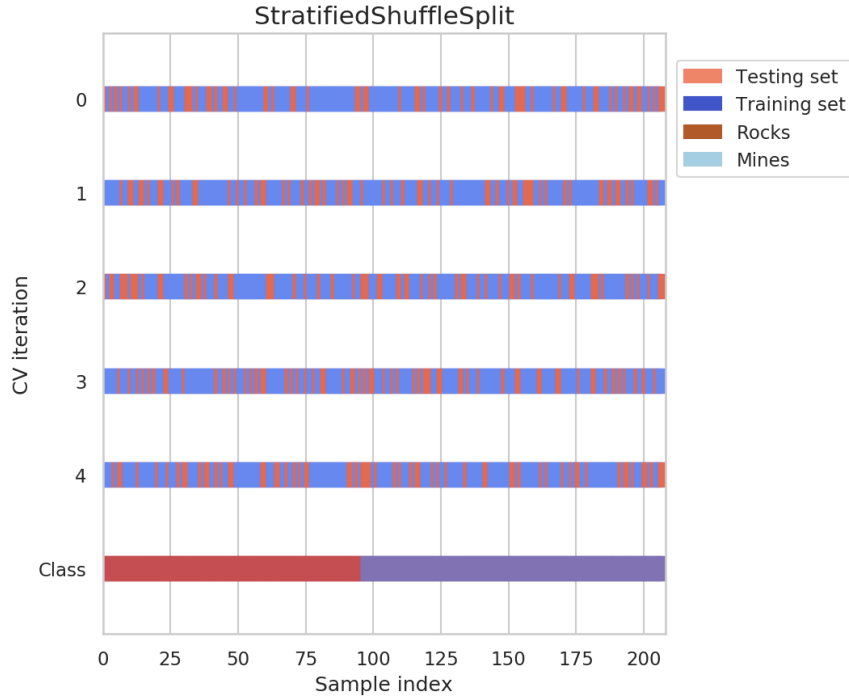


Figura 5: Técnica de validação cruzada: 5 grupos aleatórios estratificados

e na seção 3 (Resultados).

Uma vez calculada a aptidão de cada um dos indivíduos da população, os elementos são colocados no espaço objetivo com duas dimensões (acurácia e quantidade de atributos usados), neste espaço, encontramos as frentes de Pareto para realizar a seleção da nova geração utilizando o algoritmo NSGA-II.

Para elaboração deste método utilizamos a linguagem Python 3.7 e a biblioteca DEAP 1.3 (FORTIN et al. 2012). Todo código pode ser encontrado online (MORGADO 2019).

### 3 Resultados

Dois experimentos independentes foram executados um utilizando o Naive Bayes como classificador para a função de aptidão e o outro utilizando o Multi Layer Perceptron

	NaiveBayes	MLP
#Gerações	1000	500
#População	300	120
Random Seed	42	42
P(Crossover)	0.8	0.8
P(Mutação)	0.1	0.1
Seg. Teste	0.3	0.3

Tabela 3: Dados da modelagem

### 3.1 Naive Bayes

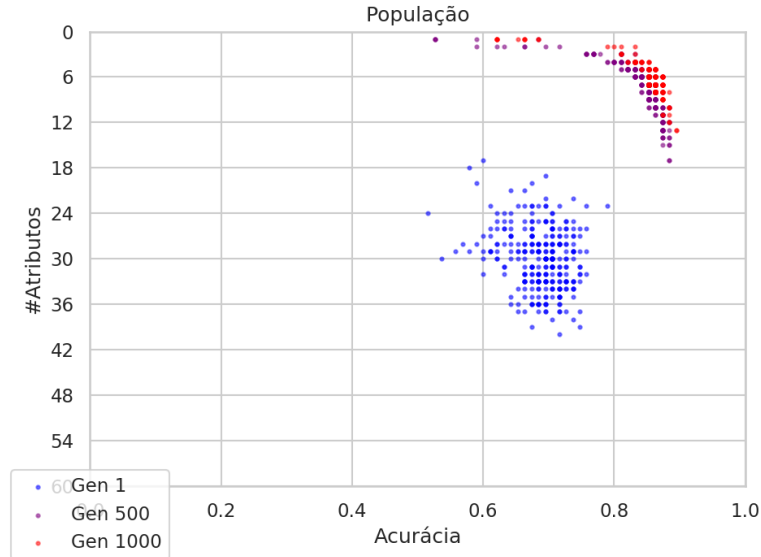


Figura 6: Espaço objetivo do modelo Naive Bayes em 3 gerações distintas.

A população inicial possui uma distribuição aleatória e uniforme com relação aos atributos escolhidos, é esperado então que aproximadamente metade dos atributos estejam ativos em média, como podemos ver no histograma apresentado figura 7a, esta figura apresenta o número de ocorrências de cada atributo na primeira geração do método Naive Bayes. Estes valores são obtidos somando cada coluna da população por atributo. Para ilustrar, vamos supor que temos a popula-

ção inicial  $P_0 = \{[1, 1, 0, 1], [1, 0, 0, 0], [0, 1, 1, 0], [1, 0, 0, 1]\}$ , somamos cada coluna e obtemos o resultado  $dist(P_0) = \{3, 2, 1, 2\}$ , este então é o resultado apresentado no histograma.

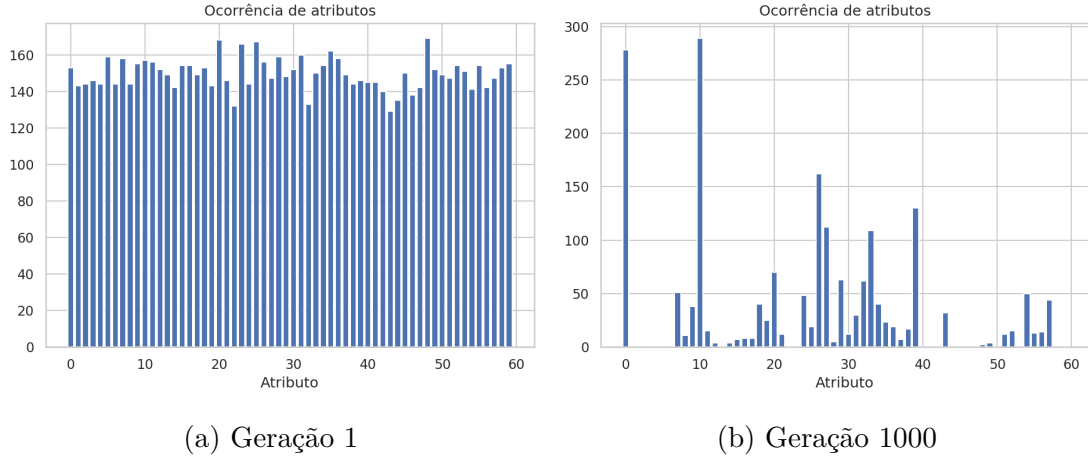


Figura 7: Histograma do total de ocorrências de cada atributo na população no método Naive Bayes

Além desta característica, a população possui na primeira geração uma distribuição uniforme em torno do ponto  $(0.67, 30)$  no espaço objetivo como apresentado na figura 6. Com o passar das gerações a população se concentra próximo a frente de Pareto (pela característica da seleção) com exceção de um pequeno grupo de mutações.

Com a passagem das gerações podemos perceber que o método começa a privilegiar alguns atributos. Percebe-se que no histograma apresentado na figura 7b o dado apresenta uma característica multi modal, com 4 modas bem evidentes em torno dos atributos 0, 10, 26 e 39 e uma outra em torno de 34

A figura 8 mostra o comportamento da seleção dos atributos da população em cada geração. Na base temos a primeira geração, onde os atributos estão uniformemente distribuídos. Após poucas gerações o método converge para priorizar o atributo 10 então o 0 e depois o 26 e o 39. Com o avançar das gerações a maior parte da população esta dominada por este atributo, mostrando uma alta probabilidade da existência deste atributo na população. As gerações seguintes servem para limpar os atributos que podem ser considerados ruído e não contribuem muito para o modelo.

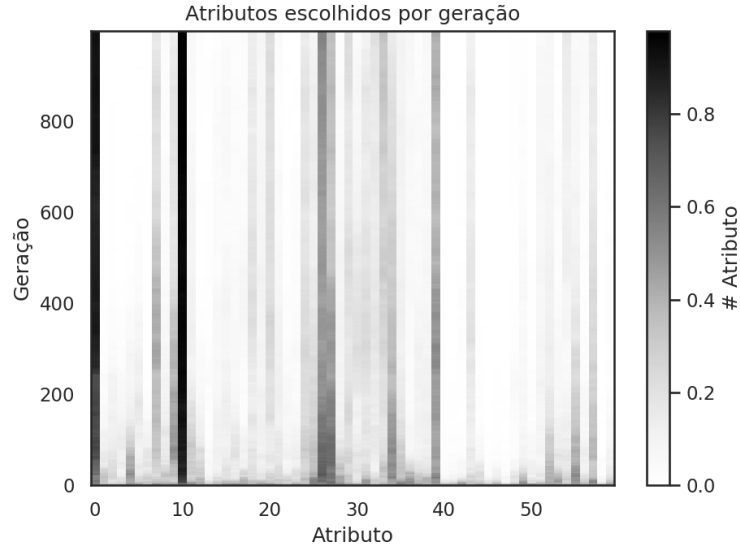


Figura 8: Evolução das porcentagem de atributos na população no método Naive Bayes.

A evolução do hiper volume é bem rápida nas primeiras 200 gerações, com pequenos saltos devido a mutações e praticamente estagnando após 500 gerações, há pouca ou nenhuma evolução entre a geração 500 e 1000 como podemos ver na figura 9.

Desta forma a figura 6 mostra que uma quantidade entre 3 e 12 atributos são suficientemente bons para caracterizar o problema, sem a necessidade do todo conjunto de dados.

### 3.2 Perceptron Multi Camadas

A mesma abordagem foi realizada para o experimento do MLP, uma população de 120 indivíduos com distribuição uniforme inicial foi gerada como podemos ver na figura 11a, podemos perceber que pelo tamanho da população ser menor, a distribuição fica menos uniforme, mas ainda assim bem distribuída.

No espaço objetivo do experimento o número de atributos ativos médio é 30 com um espalhamento uniforme da acurácia entre 0.63 e 0.8 na geração inicial. Após 250 gerações grande parte da população já está concentrada nas 3 primeiras frentes de Pareto, como pode ser visto na figura 11a.

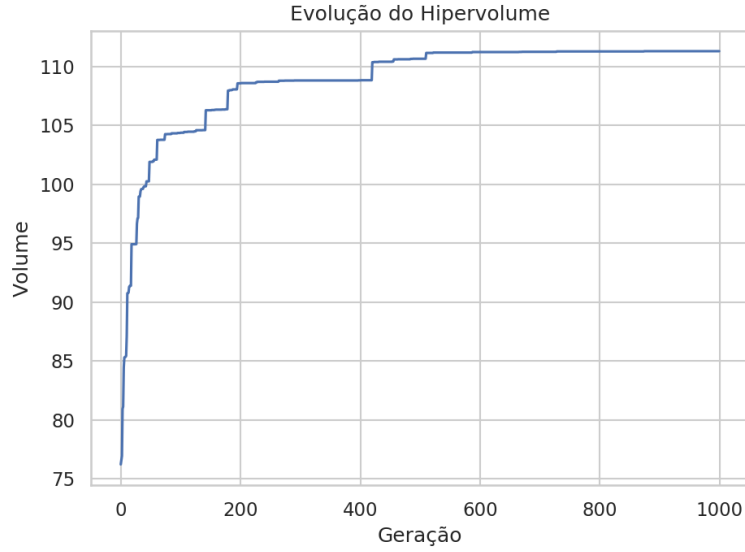


Figura 9: Hiper volume do método Naive Bayes

No final do processo, o NSGA-II + MLP apresenta um histograma multi modal com ênfase clara em 4 atributos 10, 26, 39, 44 como mostrado na figura 11b. Percebe-se pela figura 10 que os resultados sobre a frende de Pareto possuem resultados com alta acurácia, mesmo utilizando um único atributo, no ponto (0.8, 1) e uma acurácia de 0.91 utilizando somente 6 atributos.

No experimento, a população final ainda apresentava indivíduos com genes de todas as características, mesmo que a maior deles possuíssem o gene 10, 39 e 44 , como podemos ver na figura 12.

O hiper volume cresce rapidamente nas primeiras 15 iterações e encontra seu primeiro grande platô após 125 gerações. Há um outro salto próximo a geração 260 e mantém-se estagnado até o final da simulação, como mostrado na figura 13.

### 3.3 Considerações finais e trabalhos futuros

Os resultados foram interessantes, mas ainda inconclusivos. Devido a limitações computacionais não foi possível gerar populações maiores, esperava uma população inicial que apresentasse o histograma mais próximo da distribuição com desvio padrão menor do que 1%, possuímos a intuição que isso pode levar a um

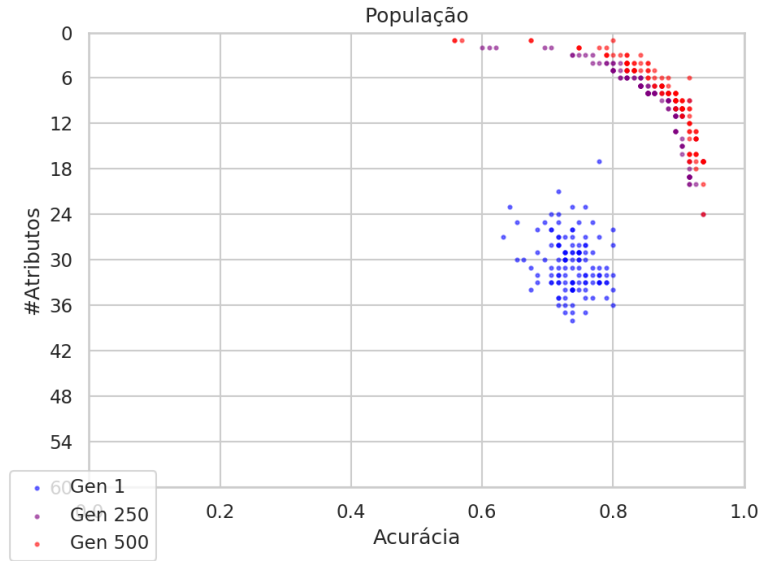


Figura 10: População em diferentes gerações do MLP

processo mais robusto. Infelizmente não encontramos material para refutar ou confirmar esta conjectura. Assim há espaço na área para entender e criar alguma medida de validação de consistência das populações iniciais.

Pela mesma razão, não foi possível realizar múltiplas baterias de simulações, com sementes aleatórias diferentes para verificar se o comportamento do método é consistente no mesmo conjunto de dados, isso nos permitira analisar se encontramos de fato os pontos críticos globais ou somente um ponto local. Ainda, interrompemos nossa simulação MLP muito cedo, com somente 500 gerações devido a existência de muitos genes diferentes em toda população é muito provável que houvesse a possibilidade de atingir resultados superiores na medida de hiper volume.

Conseguimos mostrar que uma quantidade maior de atributos não significa um melhor resultado e resultados muito próximos foram obtidos com uma grande redução do número de atributos, simplificando o modelo, confirmando a nossa hipótese inicial. Desta forma o podemos treinar o modelo utilizando somente uma quantidade reduzida dos atributos.

Nossa medida da função aptidão é obtida através da média de 5 modelos de classificação executados. Neste trabalho não levamos em consideração o desvio



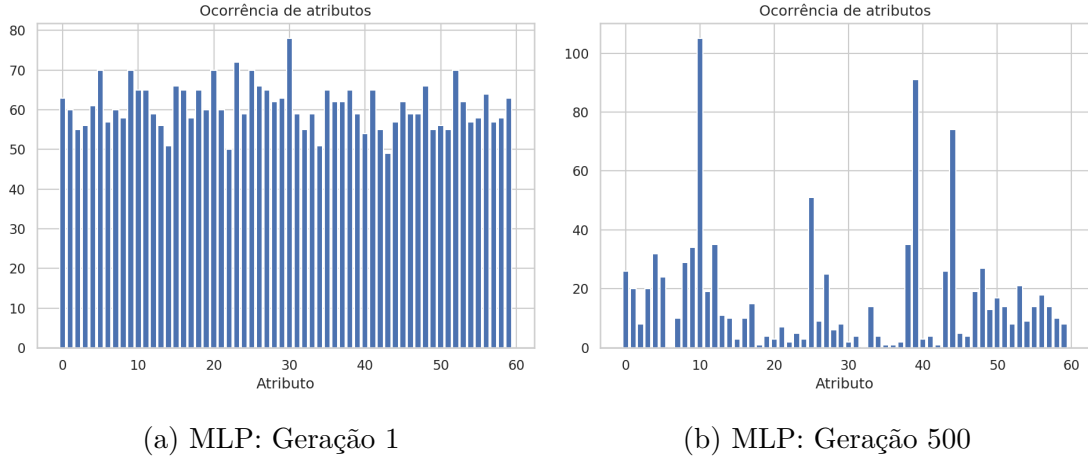


Figura 11: Histograma da distribuição de genes na população no experimento MLP

padrão, acreditamos que este dado possa ser uma medida para apresentar a consistência do método, mas será realizado em uma outra etapa. Além disso, foi realizada a estratificação em 5 grupos somente, verificar se a consistência é mantida com outros parâmetros pode também auxiliar na comprovação da robustez do método, isso também não foi realizado por falta de poder computacional.

Não esperávamos que o Naive Bayes e o MLP escolhessem atributos diferentes. O Naive Bayes ressaltou os genes 0, 10, 26, 33 e 39. Por outro lado o método MLP enfatizou os genes 10, 25, 38, 39 e 44. Nossa conjectura seria de que ambos métodos trouxessem resultados similares, por isso acreditamos que este pode ser um sintoma da baixa aleatoriedade do experimento e um número pequeno de gerações, população e baterias. Assim que possível iremos reproduzir o experimento utilizando mais tempo e poder computacional para realizar esta análise.

Não foi experimentada alterações nos hiper parâmetros do algoritmo genético como, probabilidade de cruzamento e probabilidade de mutação. Seria interessante verificar o comportamento e encontrar as regiões admissíveis para estes parâmetros.

Uma vez mitigados os pontos acima será possível fazer uma análise mais profunda com diferentes modelos de aprendizado de máquina aplicados com esta metodologia, este método talvez possa ser utilizado de alguma forma para estimar os hiper parâmetros do próprio modelo de aprendizagem de máquina.

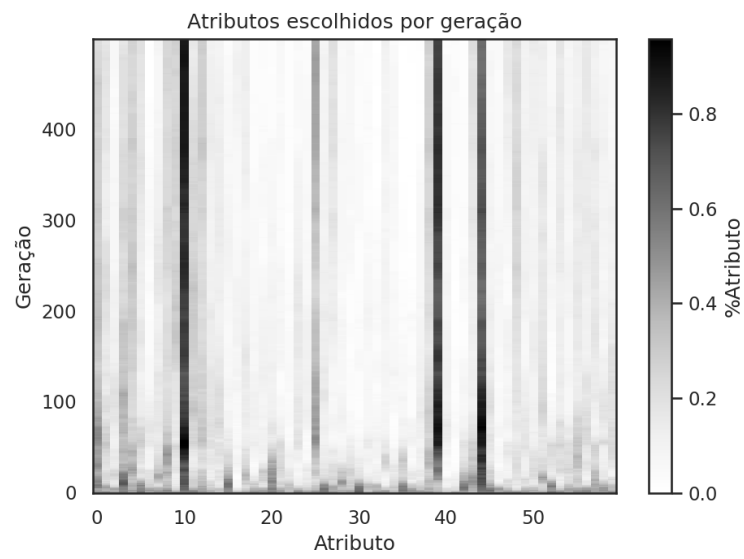


Figura 12: Evolução da seleção dos genes no MLP

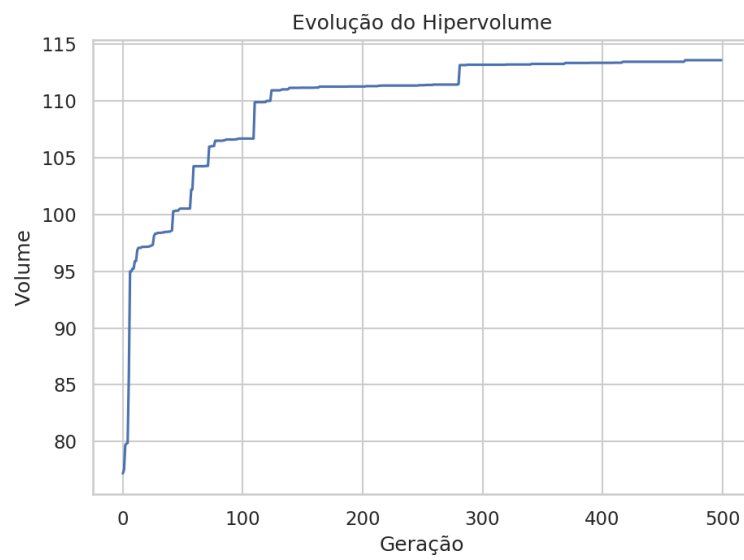


Figura 13: Hiper volume do MLP

## Referências

- DEB, K. et al. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. Em: *IEEE Transactions on Evolutionary Computation* 6.2 (2002), pp. 182–197. ISSN: 1941-0026. DOI: 10.1109/4235.996017.
- DUA, Dheeru e Casey GRAFF. *UCI Machine Learning Repository*. 2017. URL: <http://archive.ics.uci.edu/ml>.
- FORTIN, Félix-Antoine et al. “DEAP: Evolutionary Algorithms Made Easy”. Em: *Journal of Machine Learning Research* 13 (2012), pp. 2171–2175.
- GORMAN, R Paul e Terrence J SEJNOWSKI. “Analysis of hidden units in a layered network trained to classify sonar targets”. Em: *Neural Networks* 1 (1988), p. 75.
- MORGADO, Igor. *Mogann: Multi objetivo genético aplicado em redes neuronais*. 2019. URL: <https://github.com/igormorgado/mogann>.