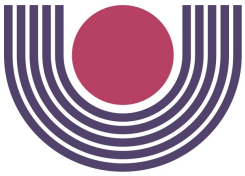

Unioeste - Universidade Estadual do Oeste do Paraná
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
Colegiado de Ciência da Computação
Curso de Bacharelado em Ciência da Computação

Análise da acurácia de classificadores treinados com bases de complexidade semelhantes

Vitor Lisboa Nogueira

CASCAVEL
2020

Vitor Lisboa Nogueira

Análise da acurácia de classificadores treinados com bases de complexidade semelhantes

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, do Centro de Ciências Exatas e Tecnológicas da Universidade Estadual do Oeste do Paraná - Campus de Cascavel

Orientador: Prof. André Luiz Brun

CASCAVEL
2020

Vitor Lisboa Nogueira

Análise da acurácia de classificadores treinados com bases de complexidade semelhantes

Monografia apresentada como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel, aprovada pela Comissão formada pelos professores:

Prof. André Luiz Brun (Orientador)
Colegiado de Ciência da Computação,
UNIOESTE

Prof. Adair Santa Catarina
Colegiado de Ciência da Computação,
UNIOESTE

Prof. Adriana Postal
Colegiado de Ciência da Computação,
UNIOESTE

Cascavel, 27 de julho de 2021

DEDICATÓRIA

Dedico este trabalho a todas as pessoas que, assim como eu, se aventuraram no caminho do TI, todos aqueles que já se formaram em algum curso da areá, ou que ainda vão se formar. A todos que decidiram por trilhar o mesmo caminho que o meu.

"Ando devagar porque já tive pressa e levo esse sorriso porque já chorei demais. Hoje me sinto mais forte, mais feliz, quem sabe, e só levo a certeza de que muito pouco sei, ou nada sei"
— Almir Sater.

AGRADECIMENTOS

Gostaria de agradecer a todos que me apoiaram e me propiciaram a chegar aqui. Em primeira lugar, agradeço a minha namorada, que me apoia em tudo que faço. Agradeço também aos meus pais, por não terem desistido de mim depois de todo esse tempo. Agradeço aos meus professores por todo o conhecimento que me transmitiram. Agradeço ao pessoal do trabalho, que me liberaram alguns dias para que eu pudesse terminar o TCC, e agradeço a todos os meus amigos, pela paciência e compreensão nesse momento tão importante para mim.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

ARFF	*Attribute-Relation File Format*
ECoL	*Extended Complexity Library*
KEEL	*Knowledge Extraction based on Evolutionary Learning*
KNN	*K-nearest neighbors*
LKC	*Ludmila Kuncheva Collection of Real Medical Data*
ML	*Machine Learning*
MLP	*Multilayer perceptron*
MST	*Minimum SpanningTree*
RP	Reconhecimento de Padrões
RSS	*Randon Subspace*
SMC	Sistema de Múltiplos Classificadores
UCI	Universidade da Califórnia em Irvine

Lista de Símbolos

φ Função de ativação
μ Média
σ Desvio-padrão
δ Distância euclidiana
ρ Coeficiente de correlação de Pearson

Sumário

Resumo

É de conhecimento geral que o comportamento dos classificadores é dependente do conjunto de dados em que foi treinado. Sendo assim vê-se necessário estudar melhor tais conjuntos de treino. Visando suplantar esta necessidade, foram propostas as medidas de complexidade que buscam permitir uma melhor compreensão dos conjuntos. Dado esse contexto, o presente trabalho teve como objetivo analisar a relação entre as métricas de complexidade e a acurácia do classificador treinado, tendo como hipótese que conjuntos de dados de complexidade semelhante geram classificadores de acurácias semelhantes. Para tal avaliação, foi desenvolvido um protocolo experimental robusto baseado na geração de 100 subconjuntos a partir dos quais foram treinados 100 *perceptrons* e extraídas 12 medidas de complexidade. Para observar se a acurácia dos classificadores se altera de acordo com a variação das métricas de complexidade, foi realizada uma análise par-a-par entre todos os classificadores do *pool*. O protocolo foi testado em 26 bases de dados com 10 repetições cada visando alcançar dados estatisticamente válidos. Observou-se que a métrica que apresentou uma similaridade mais forte com a acurácia foi L1 apresentando uma diferença não significante para com T1, F2, L2, N2, L3 e N4, e uma diferença significante para com F1, N3, F4, N1 e F3, as quais mostraram a menor similaridade com o variação de acurácia dos classificadores.

Palavras-chave: Aprendizagem de Máquina, Reconhecimento de Padrões, Sistemas de Múltiplos Classificadores, Métricas de Complexidade, Acurácia, Correlação.

Capítulo 1

Introdução

Atualmente, uma das principais áreas da computação é a inteligência artificial, que pode ser dividida em diversas subáreas, dentre as quais temos o reconhecimento de padrões (RP). A tarefa, *a priori*, parece algo simples dada a facilidade humana em fazê-lo. Conseguimos, por exemplo, reconhecer uma pessoa com base em suas características, como: rosto, voz, fisionomia, cor (da pele, dos olhos, do cabelo), etc. (**??**). Por outro lado, computadores não têm a mesma facilidade em reconhecer padrões, visto sua dificuldade em coletar e representar tais características e ao fato de ainda ser incerto o processo que os seres humanos usam para reconhecer tais padrões.

Dentre as atribuições do reconhecimento de padrões tem-se a classificação, que consiste em distinguir e delegar uma classe a objetos com base em suas características. Por exemplo, podemos distinguir um lobo-guará de um urso polar por sua cor e habitat, uma vez que o lobo-guará é vermelho e vive na América do Sul enquanto um urso polar é branco e vive no Ártico. As entidades que desempenham esta função são denominadas classificadores.

Para rotular corretamente objetos, os classificadores passam por um processo de análise de um conjunto de dados já existente. Tal processo é conhecido como aprendizagem ou treinamento. O aprendizado de máquina (ML, no original em inglês *Machine Learning*) é uma abordagem que consiste em modificar o comportamento de um sistema com base em experiências acumuladas em sua execução (**??**).

O mais tradicional tipo de classificação é a monolítica, que consiste em um único classificador responsável por atribuir a classe a todos os objetos do conjunto. Contudo, tais classificadores podem ter problemas para classificar conjuntos que apresentam maior variabilidade (**??**). Visando suplantar tal dificuldade surgem, como alternativa, os Sistemas de Múltiplos Classificadores (SMCs) – sistema com diversos classificadores distintos, que partem da premissa que

classificadores diferentes geram erros diferentes – que visam responder melhor à variabilidade de certos problemas (**??**).

Os SMCs podem ser divididos em três fases: a fase de geração, a fase de seleção, e a fase de integração. Na primeira, um *pool* de classificadores é gerado, podendo variar entre a estratégia homogênea – classificadores com a mesma técnica de aprendizagem, porém treinados em conjuntos distintos (ou com parâmetros distintos) – ou heterogênea – classificadores com técnicas de aprendizagem distintas, porém treinados no mesmo conjunto. Na segunda fase, um subconjunto destes classificadores é selecionado segundo um critério pré estabelecido, enquanto na última fase, uma decisão final é tomada com base nas predições dos classificadores selecionados (**??**).

Para as abordagens homogêneas onde cada classificador treina em um conjunto diferente de instâncias, as principais técnicas da fase de geração de *pools* são *Bagging* (**??**), *Boosting* (**??**) e *Randon Subspace* (RSS) (**??**).

O *Bagging* é um método que consiste em sortear, com reposição, instâncias do conjunto de treino para formar n subconjuntos, podendo repetir uma instância dentro de um mesmo subconjunto assim como em subconjuntos distintos.

O método *Boosting*, assim como *Bagging*, consiste em sortear, com reposição, instâncias do conjunto de treino para n subconjuntos, porém, ao contrário do método *Bagging*, o *Boosting* aplica pesos às instâncias. A cada novo subconjunto criado, um classificador testa com o conjunto de treino, aumentando os pesos, e consequentemente a probabilidade de serem sorteadas as instâncias que forem classificadas erroneamente, repetindo o processo até formar os n subconjuntos. A ideia é que os conjuntos criados foquem nos exemplos mais difíceis de se acertar.

Ao contrário das abordagens anteriores, o RSS não cria subconjuntos sorteando as instâncias, mas sim, variando o conjunto de atributos da base de dados. Este método consiste em criar subconjuntos sorteando k atributos dentre as x características das instâncias (sendo $k < x$), dessa forma, cada classificador aprenderá em um conjunto de atributos distinto.

É de conhecimento geral que o comportamento dos classificadores é dependente do conjunto de dados em que foi treinado, por isso tem-se a preocupação em como gerar subconjuntos mais adequados. Para tentar descrever tais conjuntos foram propostas as medidas de complexidade, que tentam, através de índices, descrever o grau de complexidade do conjunto usado no treino. Tais medidas são separadas em três categorias (**??**): sobreposição; separabilidade das classes e

medidas de geometria, topologia e densidade.

Em problemas mais simples, as instâncias têm classes com atributos mais distintos, sendo mais fáceis diferenciá-las. Em conjuntos mais complexos, as classes têm atributos sobrepostos, tornando difícil a distinção das instâncias de classes diferentes. Considere, por exemplo, o conjunto apresentado na Figura 1.1-a. Na ilustração foram escolhidas cinco instâncias de cada classe (5 círculos e 5 x's) para se efetuar o treinamento. No primeiro caso, o *Bagging* escolheu as instâncias destacadas em azul (Figura 1.1-b). Percebe-se que as instâncias selecionadas são bastante distintas entre os dois grupos, tendo suas características de comprimento e peso com valores bem diferentes. Neste caso, o problema apresentará uma complexidade pequena.

No segundo exemplo (Figura 1.1-c), nota-se que as instâncias selecionadas estão bem mais próximas, tendo valores de comprimento e peso sobrepostos. Os índices de complexidade deste grupo mostrarão valores mais elevados, caracterizando o problema como mais difícil.

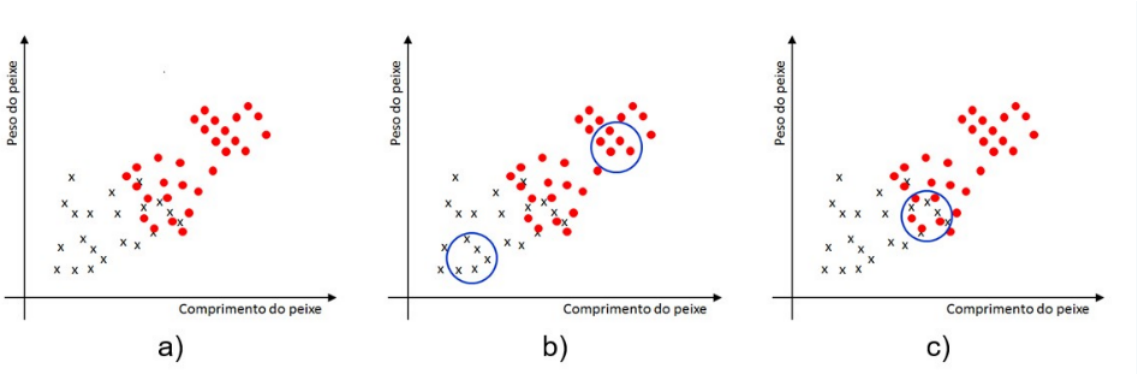

Figura 1.1: Exemplo de estimação de complexidade (**??**)

Estudos indicam que a análise da complexidade dos conjuntos de treino pode contribuir com as etapas de geração e seleção de SMCs (**??**) (**??**) (**??**). Visto isso, neste trabalho busca-se compreender melhor o funcionamento das medidas de complexidade, e qual sua relação com a acurácia dos classificadores gerados, tendo como hipótese que classificadores treinados com conjuntos de complexidade semelhantes têm acurácia semelhante.

Para que a combinação dos classificadores presentes em um pool é necessário que haja certa diversidade entre os elementos. Esta diversidade depende que os classificadores comentam erros complementares. Além disso, é importante que os membros do conjunto não sejam muito robustos. Isso faz com que a diversidade diminua. Sabendo que há trabalhos na literatura que adotam metidas de complexidade como critério de aptidão para selecionar ou gerar subconjuntos do pool de classificadores (**??**) (**??**) (**??**), é importante entender se os classificadores que

treinam em conjuntos de dificuldade similares apresentam comportamentos similares em termos de taxa de acertos. Essa informação pode ajudar na compreensão e definição de medidas de complexidade na etapa da seleção dos classificadores.

1.1 Objetivos

Este trabalho tem como objetivo geral analisar e correlacionar medidas de complexidade aplicadas aos conjuntos de treino, podendo assim apontar se classificadores treinados em conjuntos de dados com comportamentos semelhantes tem desempenhos semelhantes em termos de acurácia.

Os objetivos específicos do trabalho são:

- Implementação da técnica de *Bagging* para a geração dos subconjuntos;
- Implementação de um protocolo para obtenção das medidas de complexidade referente aos conjuntos de treino gerados pelo *Bagging*;
- Analisar as medidas de complexidade dos subconjuntos de treino, comparando o desempenho dos classificadores gerados em termos de acurácia e seu comportamento em termos de complexidade de conjunto de treino;
- Identificar quais medidas de complexidade têm comportamento mais parecido com a acurácia dos classificadores.

1.2 Organização do trabalho

Este trabalho é dividido da seguinte forma: Na **Introdução**, foram apresentados conceitos introdutórios de assuntos abordados no trabalho e os objetivos do mesmo. No segundo Capítulo, **Classificação**, os conceitos de classificadores e SMC são aprofundados. No Capítulo 3 (**Complexidade**) o conceito de complexidade é aprofundado e são apresentadas as medidas de complexidade. Na **Metodologia** (quarto Capítulo), é apresentado o o processo utilizado no desenvolvimento do trabalho, assim como a descrição de testes e análises. No Capítulo de **Resultados e discussões** os resultados do trabalho são apresentados e analisados. Já nas **Considerações finais**, esclarecimentos e percepções a respeito do desenvolvimento do trabalho são apresentados, assim como possíveis trabalhos futuros.

Capítulo 2

Classificação

A classificação, juntamente com regressão[1] e agrupamento [2], é uma das principais atribuições do reconhecimento de padrões (**??**), consistindo em distinguir e delegar uma classe a objetos com base em suas características (**??**). Por exemplo, podemos distinguir uma ervilha de um milho verde por sua cor e formato, uma vez que a ervilha é verde e tem forma arredondada enquanto um milho verde é amarelo e de formato achatado.

Nesse exemplo, escolher tamanho como uma característica para ser avaliada não é interessante, dado que tanto o milho verde quanto a ervilha têm faixas de tamanho semelhantes. Dito isso, nota-se a importância de selecionar atributos adequados para a tarefa da classificação.

As entidades que desempenham a função de classificar objetos são denominadas classificadores que, quando sozinhos recebem o nome de classificadores monolíticos ou individuais e, quando em grupo, têm o nome de Sistema de Múltiplos Classificadores (SMC).

2.1 Classificadores monolíticos

A abordagem mais tradicional de classificação é a monolítica, que consiste na construção de um único modelo de classificação cujo papel é atribuir uma classe aos objetos. Todos os exemplares do conjunto de teste são classificados pelo mesmo modelo construído.

De maneira geral, a construção de um classificador pode ser dividida em três etapas: treino, validação e teste. A etapa de treino, também conhecida como aprendizagem, é aquela de maior importância para um classificador. Nela, o classificador recebe objetos já classificados e, com

[1] Problemas de regressão consistem em interpolar um conjunto de dados, a fim de estimar valores intermediários ou fora da faixa previamente analisada.

[2] Problemas de agrupamento consistem em agrupar objetos semelhantes não classificados de acordo com suas características. Ao contrário dos problemas de classificação, o agrupamento não costuma contar com um conjunto prévio de classes, sendo estas criadas durante o processamento dos dados.

base em seus atributos, ajusta-se para melhor reconhecer cada classe.

A etapa de validação, por sua vez, serve para calibrar os parâmetros do algoritmo utilizado, visando o melhor ajuste para o problema. Já a etapa de teste é utilizada para estimar o desempenho do classificador diante do problema em questão.

Diferentes tipos de classificadores têm diferentes tipos de treinamento, por exemplo: o K-ésimo Vizinho mais Próximo (KNN, no original no inglês *k-nearest neighbors*) (**??**) apenas salva as instâncias de treino em um espaço cartesiano e, para classificar uma nova instância, compara sua posição neste mesmo espaço com todos os elementos do conjunto de treino e seleciona os k elementos mais próximos. A classe mais frequente nesta vizinhança é então atribuída à instância de teste.

Por outro lado, árvores de decisão buscam o atributo com maior grau de discriminação entre as classes e o divide em dois ou mais "nós" que representam os possíveis valores que o atributo pode assumir. Para cada novo nó, a árvore busca o novo atributo com maior nível de discriminação e divide novamente, repetindo este processo até que cada folha da árvore contenha apenas instâncias de uma única classe (**??**).

Já *perceptrons* tem seu funcionamento baseado em neurônios artificiais e suas conexões sinápticas que, por sua vez, têm como inspiração o funcionamento do sistema nervoso dos seres vivos (**??**). No caso, cada *perceptron* representa um único neurônio do sistema nervoso.

Seu treinamento consiste em ajustar um vetor de pesos, com cada um associado a um atributo do objeto. Como ilustrado na figura 2.1, cada valor de atributo (x_i) é multiplicado por seu respectivo peso (w_i), e então somados juntamente com o limiar de ativação (b) gerando o valor z. O resultado z então passa pela função de ativação (φ) retornando a classe sugerida y. A cada nova instância treinada, o algoritmo verifica se a classe sugerida é a mesma da classe esperada e, de acordo com o resultado, ajusta os pesos dos atributos, esperando errar menos nas próximas instâncias treinadas (**??**).

O *perceptron* multicamada (MLP - no original no inglês *Multilayer perceptron*), como o nome sugere, consiste em diversas estruturas *perceptron* organizadas em camadas, onde a saída de cada camada alimenta a entrada da seguinte. Seu treinamento, assim como de *perceptrons* individuais, se dá pelo ajuste de vetores de pesos, porém, no caso de MLPs, esse ajuste deve ser feito para cada neurônio de cada camada (**??**).

Contudo, classificadores monolíticos podem ter problemas para classificar conjuntos que apresentam maior variabilidade. Por exemplo, em um caso onde há muito ruído e poucos dados

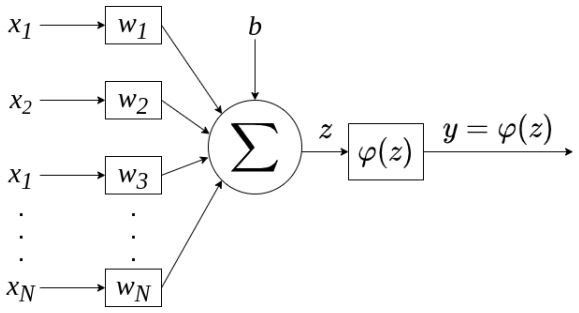

Figura 2.1: Modelo computacional de um perceptron

no conjunto de treino, um único classificador acaba não sendo o suficiente para catalogar todas as instâncias corretamente (**??**).

Visando suplantar tal dificuldade, surge como alternativa Sistemas de Múltiplos Classificadores, que se baseiam na premissa de classificadores diferentes cometem erros diferentes. Tais sistemas visam responder melhor à variabilidade de certos problemas (**??**).

2.2 Sistemas de Múltiplos Classificadores

Os sistemas de múltiplos classificadores são considerados uma das abordagens de aprendizado mais robustas e precisas (**??**). Neles diferentes classificadores são reunidos para tomar uma decisão.

Os SMCs podem ser divididos em três fases: geração, seleção e integração (também conhecida como: fusão, ou combinação) (**??**). Na primeira etapa, um *pool* de classificadores é gerado, na segunda, um ou mais classificadores são selecionados, enquanto na última fase, analisa-se as predições dos classificadores selecionados para enfim tomar uma decisão final (**??**).

Como ilustrado na Figura 2.2, a sequência padrão consiste na fase de geração seguida pela seleção dos classificadores e, por fim, a integração dos mesmos. Contudo, caso todos os classificadores gerados sejam consultados para obter um resultado final, torna-se dispensável a fase de seleção, passando diretamente para a fase de integração. Por outro lado, caso a fase de seleção eleja apenas um classificador, torna-se dispensável a fase de integração (**??**).

2.2.1 Geração

A fase de geração de *pools* pode variar entre estratégia homogênea ou heterogênea. Na primeira, todos os classificadores são construídos com a mesma técnica de aprendizagem, sendo

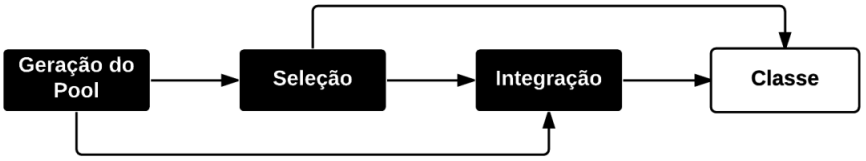

Figura 2.2: Fases de um sistema de múltiplos classificadores (**??**).

gerados subconjuntos de treino distintos para cada classificador ou variando-se os parâmetros do modelo de aprendizagem. Já na estratégia heterogênea, cada classificador possui uma técnica de aprendizagem distinta, porém são treinados no mesmo conjunto (**??**).

Visando atingir um melhor desempenho nas abordagens homogêneas, é comum a aplicação de técnicas para gerar múltiplos conjuntos distintos para treinar os classificadores. As técnicas mais comuns para criação de subconjuntos de treino para a fase de geração de *pools* são *Bagging* (**??**), *Boosting* (**??**) e *Random Subspace* (RSS) (**??**). Cada uma é melhor detalhada nas seções seguintes.

Bagging

Bagging é um método que consiste em sortear, com reposição, instâncias do conjunto de treino para formar *n* subconjuntos, podendo repetir uma instância dentro de um mesmo subconjunto assim como em subconjuntos distintos, sempre visando manter a proporção das classes do conjunto original em cada um do novos subconjuntos (**??**).

Por exemplo, como ilustrado na Figura 2.3, as instâncias ($C_1, C_2...C_n$) do conjunto de treino (Figura 2.3-A) são sorteados pelo algoritmo de *Bagging* (Figura 2.3-B) e geram os *n* subconjuntos (Figura 2.3-C), cada qual com uma fração dos elementos contidos no conjunto original. Cada subconjunto será utilizado no processo de treino de um novo classificador.

Boosting

O método *Boosting* (**??**), assim como *Bagging*, consiste em sortear, com reposição, instâncias do conjunto de treino para compor *n* subconjuntos. Porém, ao contrário do primeiro, esta estratégia aplica pesos às instâncias, o qual representa a chance daquela instância ser sorteada.

A cada novo subconjunto criado, um classificador faz a rotulação do conjunto de treino original. Para cada instância classificada erroneamente, o algoritmo aumenta o peso dessa instância e, consequentemente, a probabilidade dela ser sorteada para os próximos conjuntos. Esse

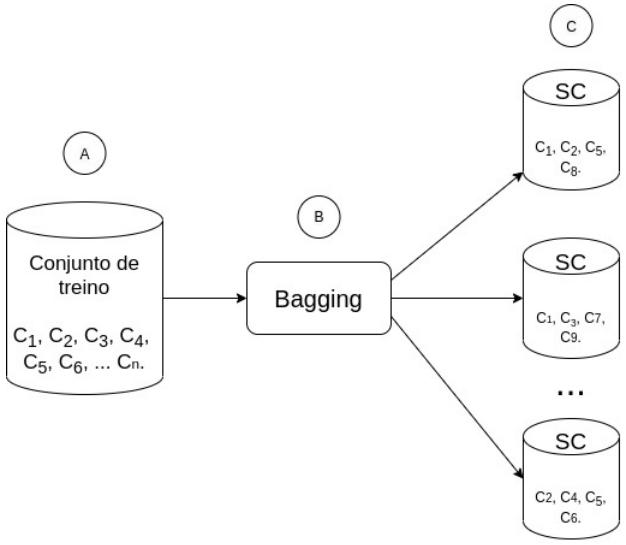

Figura 2.3: Exemplo do funcionamento do *Bagging*.

processo é repetido até formar os n subconjuntos. A ideia é que os conjuntos criados foquem nos exemplos mais difíceis de classificar (**??**).

A Figura 2.4 apresenta o funcionamento do método de *Boosting*. Na ilustração 2.4-A, temos o conjunto de treino original, que passa pelo algoritmo de *Boosting* (Figura 2.4-B), gerando os subconjuntos da Figura 2.4-C considerando os pesos de cada instância. A cada subconjunto novo, um classificador avalia as instâncias do conjunto de treino original (Figura 2.4-D), modificando tais pesos, visando beneficiar as instâncias com maior dificuldade de classificação. O processo de geração de subconjunto e avaliação de instâncias se repete até terem sido gerados todos os n subconjuntos do *pool*.

Random Subspace

Ao contrário das abordagens anteriores, o RSS não cria subconjuntos sorteando as instâncias, mas sim, variando o conjunto de atributos da base de dados (**??**). Este método consiste em formar subconjuntos sorteando k atributos distintos dentre as x características das instâncias (sendo $k < x$), com cada atributo podendo pertencer a mais de um conjunto. Dessa forma, cada classificador aprenderá em um conjunto de atributos distinto (**??**). Ao selecionar aleatoriamente os atributos, espera-se criar classificadores complementares, o que faz com que cometam erros diferentes, sendo essa uma característica positiva em cenários de combinação de classificadores (**??**).

Uma representação deste algoritmo é apresentada na Figura 2.5. Na ilustração vemos que

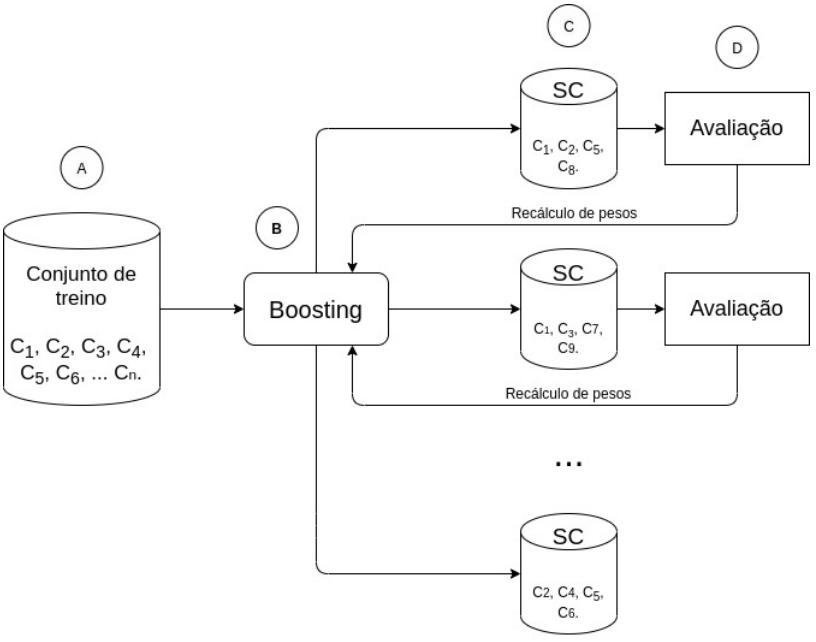

Figura 2.4: Exemplo do funcionamento do *Boosting*.

o conjunto original de treino composto por *x* atributos ($A_1, A_1, A_1, ..., A_x$) (Figura 2.5-A) é submetido ao algoritmo RSS (Figura 2.5-B) que cria *n* subconjuntos, com todas as instâncias do conjunto original, mas com apenas quatro atributos (Figura 2.5-C).

2.2.2 Seleção

A etapa de seleção de classificadores em um SMC consiste em determinar qual ou quais classificadores serão empregados na tarefa de rotular cada novo objeto a ser analisado. Tal seleção pode ser feita de forma estática ou dinâmica (**??**).

Uma seleção estática consiste em, ainda na etapa de treino, selecionar os classificadores considerados mais adequados segundo algum critério pré definido, normalmente aqueles que demonstraram melhor acurácia. Estes classificadores irão compor o grupo com função de rotular todas as novas instâncias (**??**).

Por outro lado, o método de seleção dinâmica consiste em escolher um ou mais classificadores para cada nova instância avaliada. Tais classificadores são selecionados de acordo com as características da instância de teste. Dessa forma para cada nova instância realiza-se uma nova seleção (**??**).

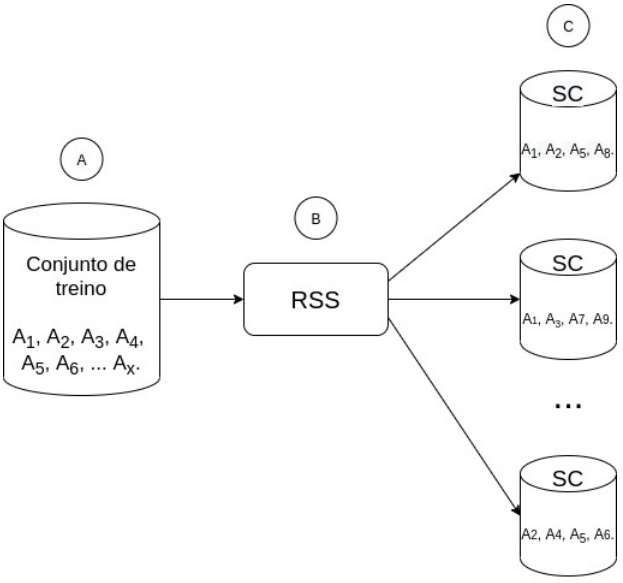

Figura 2.5: Exemplo do funcionamento do *Random Subspace*.

2.2.3 Integração

Quando é formado um grupo de classificadores, torna-se necessário definir uma forma de combinar a opinião individual de cada um deles em uma única opinião para rotular a instância.

Tal combinação pode ser feita de três formas diferentes: paralela, serial ou híbrida (**??**). No método paralelo, todos os classificadores realizam a tarefa de rotulação sobre uma mesma instância, e suas opiniões são combinadas no resultado final. As regras mais utilizadas para a combinação das opiniões são a regra da soma, do produto, do mínimo, do máximo, da média e da mediana (**??**).

O método serial, também conhecido como método em cascata, consiste em ordenar de forma crescente os classificadores conforme sua robustez (**??**). Cada instância é submetida ao classificador mais simples e, com base em um patamar pré-estabelecido, pode ser classificada ou rejeitada. Caso seja rejeitada, ela segue para a análise do classificador seguinte. Este processo se repete até que todas as instâncias sejam rotuladas, ou até que todos os classificadores sejam consultados.

Esta técnica possibilita a economia de processamento, visto que os classificadores mais complexos apenas serão utilizados em casos onde os demais não são capazes de rotular a instância. Contudo, tais classificadores se tornam incapazes de corrigir erros dos classificadores mais simples, visto que apenas as instâncias rejeitadas são passadas para frente (**??**).

O método híbrido, como o nome sugere, combina aspectos das estratégias paralela e serial,

podendo assim, conter diferentes múltiplas cascatas de classificadores, funcionando de forma paralela. Ele visa, ao mesmo tempo, possibilitar o processamento independente das instâncias, como também a checagem e correção de possíveis erros (**??**).

Capítulo 3

Complexidade

É de conhecimento geral que o comportamento dos classificadores está diretamente ligado à qualidade do conjunto de dados utilizado em seu treinamento, por isso a preocupação em gerar subconjuntos mais adequados. Uma estratégia para descrever tais conjuntos é a utilização das medidas de complexidade que tentam, através de índices, descrever o grau de complexidade do conjunto (**??**).

Em problemas mais simples, as instâncias têm classes com atributos mais distintos, sendo mais fáceis diferenciá-las. Em conjuntos mais complexos, as classes têm atributos que se sobrepõe, tornando difícil a distinção das instâncias de classes diferentes.

As medidas de complexidade comumente são separadas em três categorias (**??**) (**??**) (**??**): sobreposição (F1, F2, F3, F4); separabilidade das classes (L1, L2, N1, N2, N3) e medidas de geometria, topologia e densidade (L3, N4, T1, T2, D1, D2, D3). Outra forma de categorizar uma medida de complexidade é pelo foco da análise que está sendo efetuada no conjunto de dados. A Tabela 3.1 ilustra esse método de taxonomia (**??**).

Tabela 3.1: Taxonomia das medidas de complexidade segundo **??**).

Taxonomia	**medidas**
Medidas baseadas em atributos	F1, F2, F3, F4
Informações da vizinhança	N1, N2, N3, N4, T1
Linearidade	L1, L2, L3
Dimensionalidade	T2
Outros	D1, D2, D3

3.1 Medidas de sobreposição das classes

O objetivo das medidas de sobreposição é estimar o quão sobrepostas duas classes estão no espaço de características. Para tal, examina-se o intervalo e a dispersão dos atributos de instâncias de classes diferentes.

3.1.1 Relação Máxima do Discriminante de Fischer (F1)

Esta medida de sobreposição estima o quão separadas são duas classes de acordo com uma única característica, analisando a eficácia de um atributo em separar as classes (**??**). É possível considerar F1 como sendo a distância entre os centros de duas classes. Para tal, calcula-se F1 através de um índice cujo valor aumenta à medida que mais afastados os centros das classes sejam um do outro (**??**).

A Figura 3.1 apresenta de forma mais clara a ideia do funcionamento do Discriminante de Fischer. É possível identificar o centro das classes vermelha e azul e a medida (δ_1) da distância (geralmente a distância euclidiana) entre o centro de ambas as classes.

Neste exemplo, as classes vermelha e azul são linearmente separáveis, não apresentando sobreposição. Todavia, as classes ainda podem apresentar sobreposição dependendo de suas distribuições, sendo necessário empregar mais de uma medida de complexidade para avaliá-las corretamente (**??**)

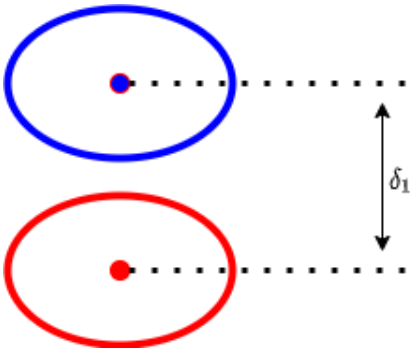

Figura 3.1: Funcionamento do Discriminante de Fischer (δ_1) em classes linearmente separáveis.

Para calcular F1 em problemas onde existem apenas duas classes, utiliza-se a Equação 3.1, que compara as médias e desvio-padrões das classes para cada um de seus atributos, medindo assim a distância entre elas. Onde μ_1 , μ_2 representam as médias (centroides) das duas classes, já σ_1 , σ_2 representam seus desvio-padrões (**??**).

$$F1_i = \frac{(\mu_{1_i} - \mu_{2_i})^2}{\sigma_{1_i}^2 - \sigma_{2_i}^2} \tag{3.1}$$

Porém, para casos mais gerais, onde existem N (N > 2) classes utiliza-se a Equação 3.2, em que n_i denota o número de instâncias da classe i, δ é uma medida de distância, μ_i é a média da classe i e x_j^i representa o elemento j que pertence a classe i.

$$F1 = \frac{\sum_{i=1}^{C} n_i.\delta(\mu, \mu_i)}{\sum_{i=1}^{C}\sum_{j=1}^{n_i} \delta(x_j^i, \mu_i)} \tag{3.2}$$

3.1.2 Sobreposição de Atributos por Classe (F2)

A medida F2 estima a sobreposição dos valores de um atributo de instâncias de duas classes diferentes (**??**). Como ilustra a Figura 3.2, é possível calcular esta sobreposição de um atributo encontrando seus valores máximos e mínimos e, em seguida, calculando-se a razão entre a intersecção da região das duas classes e amplitude total do atributo.

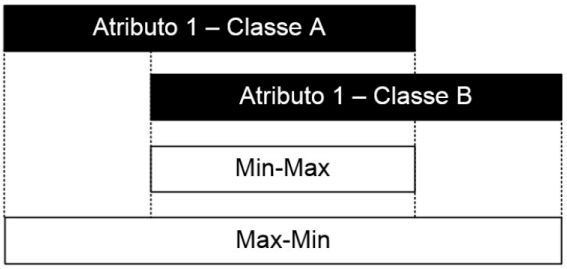

Figura 3.2: Representação gráfica do cálculo de F2 (**??**)

O valor total de F2 é dado pelo produto entre os valores F2 individuais de cada um dos atributos do conjunto. A Equação 3.3 representa o cálculo de F2, onde i é o índice do atributo sendo verificado, d indica a número de características, f_i corresponde ao atributo sendo verificado e c_1 e c_2 correspondem às classes em análise.

$$F2 = \prod_{i=1}^{d} \frac{MIN(max(f_i, c_1), max(f_i, c_2)) - MAX(min(f_i, c_1), min(f_i, c_2))}{MAX(max(f_i, c_1), max(f_i, c_2)) - MIN(min(f_i, c_1), min(f_i, c_2))} \tag{3.3}$$

Dado as características de uma multiplicação, percebe-se que para o valor de F2 ser zero, basta que apenas um atributo não tenha sobreposição entre as classes (**??**).

3.1.3 Eficiência Máxima por Atributo Individual (F3)

A medida de F3 calcula a capacidade individual que cada atributo tem de classificar as instâncias. A eficiência de cada característica é calculada pela razão entre o número de instâncias que podem ser separados por esta característica pelo número total de instâncias do conjunto (**??**). Sendo assim, espera-se que quanto maior o valor de F3, mais discriminante seja o atributo. A eficiência máxima individual de um atributo será definida pelo maior valor obtido entre todos os atributos considerados (**??**).

3.1.4 Eficiência Coletiva dos Atributos (F4)

A medida F4, assim como F3, calcula a propriedade discriminante dos atributos, porém, ao contrário de F3, nesta medida todos os atributos são considerados para o valor do índice.

Para calcular a propriedade discriminante coletiva, primeiramente, seleciona-se o atributo que consegue separar o maior número de instâncias (maior poder discriminativo). Em seguida, todas as instâncias que puderam ser discriminadas são removidas do conjunto de dados. O próximo atributo mais discriminativo é selecionado, repetindo o processo de eliminar as instâncias discriminadas por tal atributo. Este procedimento é repetido até que todas as instâncias sejam discriminadas, ou até que todos os atributos tenham sidos analisados (**??**). O valor de F4 é dado pela razão de instâncias que foram discriminadas, pelo número total de elementos do conjunto.

3.2 Medidas de separabilidade das classes

As medidas de separabilidade avaliam o quão complexo é o comportamento dos conjuntos nas regiões de fronteira entre as classes, geralmente adotando estratégias de vizinhança.

3.2.1 Soma Minimizada da Distância de Erro de um Classificador Linear (L1)

A medida L1 é calculada através da soma das distâncias euclidianas (δ) entre fronteira linear formada por um classificador linear ótimo e cada uma das amostras classificadas erroneamente (**??**). Tal processo pode ser melhor visualizado na Figura 3.3, onde temos a linha vermelha representando a fronteira linear criada pelo classificador, e duas instâncias classificadas erroneamente com suas respectivas distâncias δ_1 e δ_2 até a fronteira.

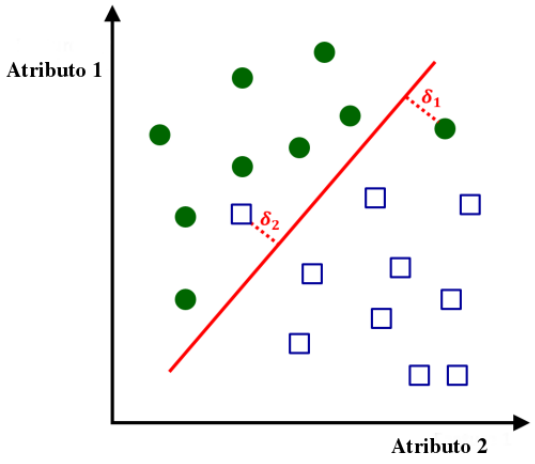

Figura 3.3: Classificador linear com duas instâncias classificadas erroneamente. Adaptado de **??**)

A medida L1 evidencia o quanto os dados do conjunto de treino são linearmente separáveis (**??**). Um valor de L1 igual a zero, expressa que nenhuma instância foi classificada erroneamente, indicando que os dados do conjunto são linearmente separáveis. Entretanto, quanto maior for o valor de L1, maior o grau de complexidade do conjunto de amostras. Sendo assim, o conjunto de dado apresenta menor grau de separabilidade (**??**).

3.2.2 Taxa de Erro de um Classificador Linear sobre o Treino (L2)

L2 representa a medida da taxa de erro obtida através da utilização de um classificador linear ótimo sobre as instâncias de treino (**??**), (**??**) (**??**). A ideia é utilizar o mesmo classificador criado em L1 para verificar quantas amostras estão posicionadas no lado oposto da fronteira linear que o lado correspondente à sua classe. O valor de L2 é calculado pela razão entre o número de elementos classificados erroneamente e número total de instâncias analisadas.

Um valor de L2 igual a zero, indica que as duas classes são linearmente separáveis. Entretanto, quanto mais próximo de 1 o valor estiver, menos linearmente separáveis elas são (**??**).

3.2.3 Fração de Pontos na Região de Fronteiras (N1)

N1 se baseia na construção de uma árvore de cobertura mínima (MST - *Minimum Spanning Tree*), na qual todos os pontos do conjunto de dados são conectados de forma a minimizar a soma das distâncias. A Figura 3.4 ilustra um conjunto de dados com duas classes, onde as linhas contínuas ligam elementos membros de mesma classe, enquanto as linhas serrilhadas

conectam elementos de classes distintas. Os pontos conectados com elementos de outra classe são considerados como elementos de fronteira (**??**).

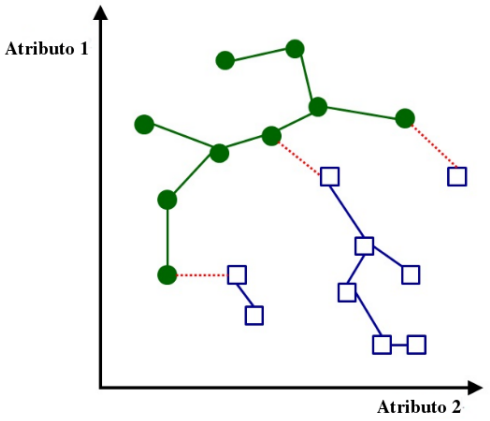

Figura 3.4: Árvore de cobertura mínima construída com base em duas classes. Adaptado de **??**)

Calcula-se N1 através da razão entre a contagem de elementos de fronteira pelo número total de instâncias do conjunto.

3.2.4 Proporção das Distâncias Intra/Inter classes até o vizinho mais próximo (N2)

A medida N2 consiste em comparar a distância de cada elemento e seu vizinho mais próximo dentro da classe, com a distância até o vizinho mais próximo pertencente a outra classe (**??**).

Assim como apresentado na Figura 3.5, o método calcula a distância euclidiana entre cada elemento do conjunto e seus vizinhos internos e externos mais próximos, representados respectivamente pela lina azul e linha pontilhada vermelha. N2 se dá pela razão entre a somatória de todas as distâncias euclidianas dos elementos para seu vizinho mais próximo em sua classe, e a somatória de todas as distâncias entre os mesmos elementos e os vizinhos mais próximos de outra classe (**??**), (**??**).

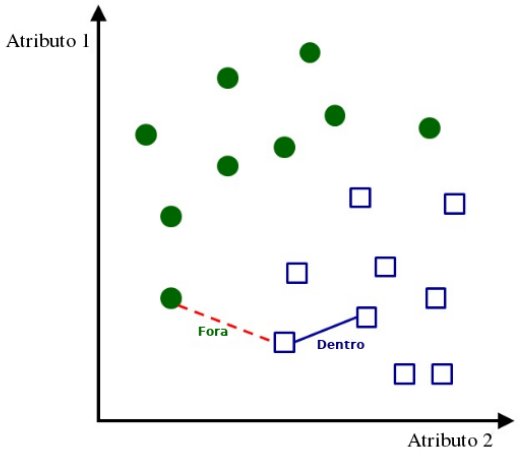

Figura 3.5: Representação da distância entre os vizinhos mais próximos dentro e fora da classes. Adaptado de **??**)

3.2.5 Taxa de erro do classificador KNN pela abordagem *Leave-One-Out* (N3)

A medida N3 é estimada pela taxa de erro de um classificador KNN com o valor de k sendo configurado como $k = 1$ (**??**). A taxa de erros é estimada pelo método *Leave-One-Out*, que consiste em testar cada instância em um classificador treinado com os demais elementos do conjunto (**??**).

Valores próximos de 0 para N3 indicam espaçamento entre os elementos da região de fronteira das classes, enquanto valores próximos de 1 indicam sobreposição entre tais classes (**??**).

3.3 Medidas de geometria, topologia e densidade

As medidas de Geometria, Topologia e Densidade buscam apresentar uma compreensão mais espacial do relacionamento das classes, por meio da descrição da geometria das mesmas.

3.3.1 Fração de Esferas de Cobertura Máxima (T1)

A medida T1 representa o número de círculos necessários para cobrir cada classe (**??**). Inicialmente, é criado um círculo centralizado em cada instância do conjunto de dados, o qual cresce até atingir uma instância de outra classe. As circunferências que estão completamente situadas dentro de outras circunferências são ditas redundantes, sendo assim descartadas (**??**). Por fim, T1 é calculado através da razão entre a contagem de circunferências e o número total de instâncias do conjunto, conforme ilustra a Figura 3.6.

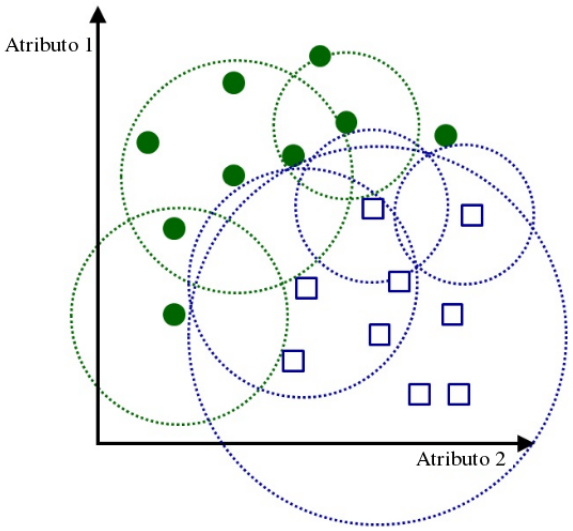

Figura 3.6: Representação das esferas necessárias para cobrir uma classe. Adaptado de **??**)

Se o conjunto apresentar instâncias muito próximas de uma classe vizinha, a quantidade de circunferências tende a ser maior, enquanto seu tamanho tende a diminuir, aumentando assim o valor de T1 (**??**).

3.3.2 Número médio de pontos por dimensão (T2)

A medida T2 descreve a densidade da distribuição espacial de amostras através da razão entre número de instâncias no conjunto perante o número de atributos (**??**). Esta medida é apontada pelos autores como uma medida que não traz informações pertinentes a respeito da separabilidade das classes quando utilizado um classificador linear. Contudo, T2 fornece informações relevantes em casos onde os classificadores são não lineares, como por exemplo o KNN (**??**).

3.3.3 Não-Linearidade de um Classificador Linear (L3)

O método L3 analisa a não linearidade de um classificador em relação ao conjunto de dados. Este método consiste em criar um conjunto de teste através da interpolação linear, com coeficientes gerados aleatoriamente, entre duas instâncias da mesma classe randomicamente selecionadas. Então L3 corresponderá ao valor da taxa de erro quando avalia-se o novo conjunto de testes utilizando um classificador linear similar ao de L1 (**??**).

O processo de criação do conjunto de teste pode ser visualizado na Figura 3.7. Primeiramente, o conjunto de treino (Figura 3.7-a) passa pelo sorteio de elementos e coeficiente para

a interpolação linear gerando os novos atributos do conjunto de teste (Figura 3.7-b). A Figura 3.7-c ilustra o conjunto de teste já formado.

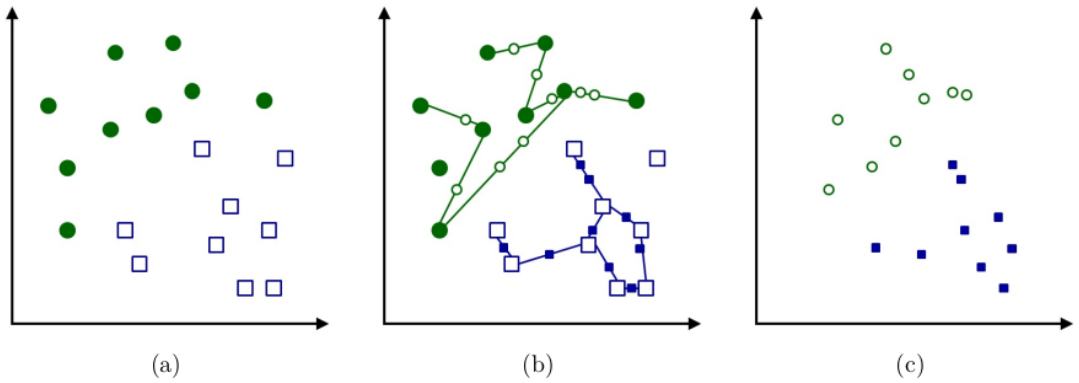

Figura 3.7: Processo de geração do conjunto de teste adotado em L3 (**??**)

3.3.4 Não-Linearidade de um Classificador KNN (N4)

A medida N4 se utiliza do mesmo princípio de criação do conjunto de testes adotado por L3. Todavia, ao invés de um classificador linear, é empregado o classificador KNN para calcular a taxa de erro sobre o conjunto de testes (**??**) (**??**) (**??**).

3.3.5 Densidade (D1)

A medida D1 pode ser descrita como o número médio de elementos por unidade de volume onde as amostras estão distribuídas. O valor do volume é obtido pelo produto dos comprimentos de todos os intervalos de características de todas as classes (**??**).

3.3.6 Volume de Vizinhança Local (D2)

O método D2 representa a média entre os volumes (V_i) ocupados pelos k vizinhos mais próximos de cada instância do conjunto de treino (**??**).

Tendo em mente que $N_k(x_i)$ representa o conjunto de k vizinhos mais próximos de uma instância (x_i), então o volume (V_i) pode ser calculado conforme Equação 3.4, sendo que $max(f_h, N_k(x_i))$ e $min(f_h, N_k(x_i))$ correspondem, respectivamente, aos valores máximo e mínimo do atributo f_h dentro do conjunto formado pelos k vizinhos mais próximos da cada elemento (x_i).

$$V_i = \prod_{h-1}^{d} (max(f_h, N_k(x_i)) - min(f_h, N_k(x_i))) \tag{3.4}$$

3.3.7 Densidade da Classe na Região de Sobreposição (D3)

A medida D3 busca evidenciar a densidade relativa de cada classe na região de sobreposição das classes. Tais regiões são as que costumam incorrer na maioria dos erros, sendo assim consideradas regiões críticas para a classificação (**??**).

O processo se inicia com a busca dos k vizinhos mais próximos de cada instância x_i, seguindo da verificação das classes dos elementos dessa vizinhança. Caso a maioria destes elementos seja de uma classe diferente da instância analisada, considera-se que ela faz parte de uma região de sobreposição. O valor de D3 é obtido pela razão entre o número de elementos, de uma determinada classe, dentro de uma região de sobreposição pelo total de instâncias pertencentes à tal classe.

Capítulo 4

Metodologia

Este trabalho tem como objetivo analisar a relação entre a assinatura de complexidade do conjunto de treino usado para treinamento de um classificador com a sua respectiva acurácia. O objetivo é avaliar se classificadores que foram treinados em conjuntos de complexidade similar têm acurácias parecidas.

Para tanto, é necessária a construção de uma aplicação cuja entrada seja uma base de dados genérica e a saída sejam as métricas de complexidade de cada subconjunto de treino, bem como a acurácia de cada classificador. A Figura 4.1 ilustra o passo a passo de tal aplicação. Cada uma das etapas são descritas na sequência do documento.

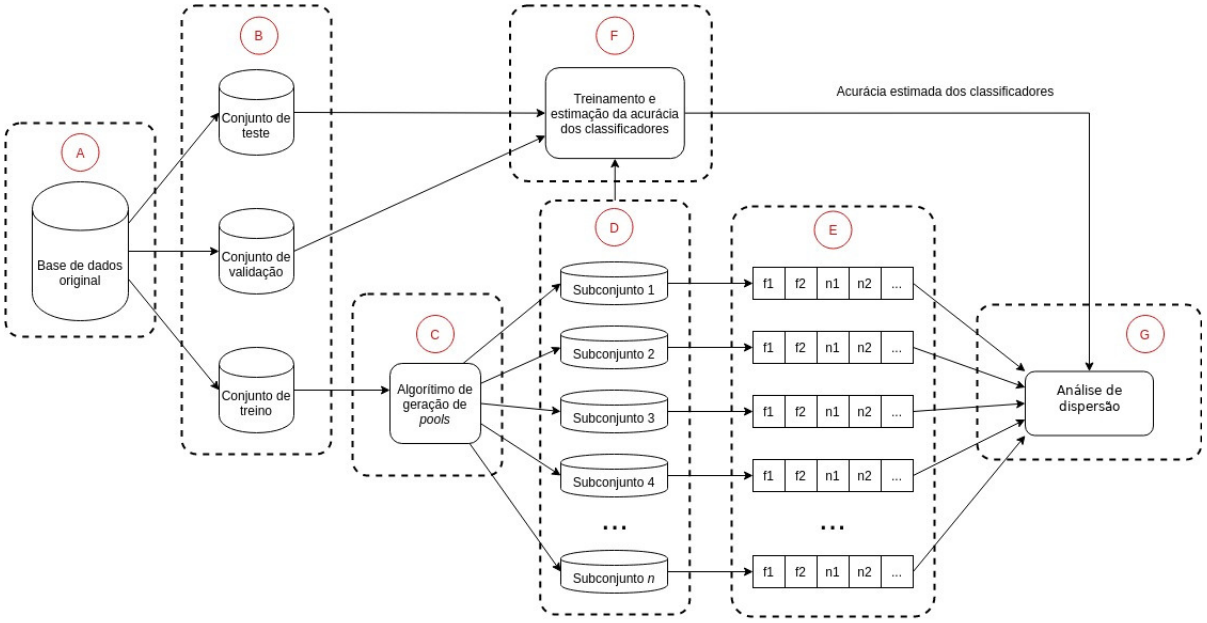

Figura 4.1: Fluxograma das etapas realizadas do trabalho

Inicialmente, cada base é dividida em conjuntos de treino, validação e teste (Figura 4.1-A, B). Em seguida, a partir do conjunto de treino, com a implementação do algoritmo de *Bagging*

(4.1-C), são gerados 100 subconjuntos para treino (Figura 4.1-D). Tais conjuntos são gravados em arquivos do tipo ARFF [1] para análise posterior.

Na etapa seguinte (Figura 4.1-E), estes arquivos são utilizados para estimação dos índices de complexidade empregando-se a biblioteca ECoL (*Extended Complexity Library*) (**??**). Para cada conjunto de treino fornecido, a biblioteca retorna um vetor contendo os índices de complexidade requisitados.

Os subconjuntos gerados são então usados para a construção dos modelos de classificação (Figura 4.1-F). Os classificadores construídos são então submetidos à estimação de sua acurácia. Esse processo será feito sobre o conjunto de teste. Tais valores são enviados à etapa seguinte do processo (Figura 4.1-G).

Após a conclusão da estimação dos índices de complexidade e da acurácia dos classificadores gerados, é realizada a análise da relação entre tais índices (Figura 4.1-G). Nesta fase, é estimado o comportamento das variações entre pares de classificadores para a acurácia e complexidade.

Os métodos implementados neste trabalho foram desenvolvidos em linguagem Python, com a utilização da biblioteca *scikit-learn* (**??**), biblioteca essa que visa facilitar o uso de algoritmos de classificação, regressão e agrupamento.

4.1 Base de dados

O processo de desenvolvimento adotou o mesmo conjunto de bases de dados utilizado no trabalho de **??**), composto por 26 bases, as quais são formadas apenas de atributos numéricos e não apresentam valores faltantes. Além disso, tais bases são frequentemente usadas na literatura como base para avaliação de métodos de classificação.

Destas bases, quatorze são originárias do repositório da UCI – Universidade da Califórnia em Irvine – (**??**), duas são procedentes do repositório KEEL (*Knowledge Extraction based on Evolutionary Learning*) (**??**), outras quatro pertencentes à LKC (*Ludmila Kuncheva Collection of Real Medical Data*) (**??**), quatro provenientes do projeto STATLOG (**??**) e duas bases geradas artificialmente com o *toolbox* PRTools do Matlab.

A Tabela 4.1 apresenta cada uma das bases de dados analisadas nesse trabalho juntamente

[1] *Attribute-Relation File Format* é um formato de arquivo que visa representar, de forma padronizada, conjuntos de dados com instâncias independentes e não ordenadas, e que não envolvem relacionamentos entre as mesmas (**??**).

com sua respectiva origem. Além disso, são apresentadas as principais informações de cada base, as quais: quantidade total de instâncias do conjunto, o número de atributos de cada instância de base, e o número de classes em que um objeto dessa base pode ser classificado.

Tabela 4.1: Principais características das bases usadas nos experimentos

Base de dados	Nº Instâncias	Nº Atributos	Nº Classes	Fonte
Adult	690	14	2	UCI
Banana	2000	2	2	PRTools
Blood	748	4	2	UCI
CTG	2126	21	3	UCI
Diabetes	766	8	2	UCI
Faults	1941	27	7	UCI
German	1000	24	2	STATLOG
Haberman	306	3	2	UCI
Heart	270	13	2	STATLOG
ILPD	583	10	2	UCI
Ionosphere	350	34	2	UCI
Laryngeal1	213	16	2	LKC
Laryngeal3	353	16	3	LKC
Lithuanian	2000	2	2	PRTools
Liver	345	6	2	UCI
Mammo	830	5	2	KEEL
Monk	432	6	2	KEEL
Phoneme	5404	5	2	STATLOG
Segmentation	2310	19	7	UCI
Sonar	208	60	2	UCI
Thyroid	692	16	2	LKC
Vehicle	847	18	4	STATLOG
Vertebral	300	6	2	UCI
WBC	569	30	2	UCI
WDVG	5000	21	3	UCI
Weaning	302	17	2	LKC

O processo se inicia com a divisão das bases em três novos conjuntos, sendo eles treino, validação, teste. Esta divisão é feita de forma aleatória mantendo as proporções (estratificação) das classes da base de dados original em cada novo conjunto.

O primeiro conjunto, treino, tem como função servir de base para o aprendizado do classificador, sendo este composto de 50% da base de dados original. O segundo conjunto, Validação, é usado no processo de calibragem de parâmetros do classificador, tendo 25% das instâncias do conjunto. Já o conjunto de teste é utilizado na estimação da acurácia do classificador, contendo os 25% dos dados restantes.

A partir do conjunto de treino criado, são gerados 100 subconjuntos de treino por meio do

algoritmo de *Bagging*. Cada subconjunto contém 20% das instâncias da base original, podendo repetir instâncias entre os conjuntos ou mesmo dentro de um mesmo conjunto. O *Bagging* foi escolhido como algoritmo de geração de *pools* por ser simples e amplamente utilizado na literatura.

Por exemplo, ao dividir a base Banana com suas 2000 (duas mil) instâncias teremos: 1000 (mil) instâncias no conjunto de treino, 500 (quinhentas) no conjunto de validação, e as 500 (quinhentas) instâncias restantes no conjunto de teste. Além disso, cada subconjunto de treino contaria com 200 (duzentas) instâncias. Em todas as divisões é respeitada a proporção das classes presentes no conjunto original.

4.2 Construção dos classificadores

Buscando maior variabilidade entre as acurácias dos classificadores treinados por cada subconjunto, foi implementado um classificador fraco do tipo *perceptron*, de modo que pequenas alterações nos conjuntos de treino resultem em uma maior variação entre os classificadores em termos de opinião e taxas de acerto. Tal configuração é adequada para um SMC de *pool* homogênea pois permite que os classificadores tenham erros complementares.

Visando atingir uma configuração de parâmetros que menos interfira na acurácia dos classificadores, optou-se em adotar as configurações padrões do *perceptron* da biblioteca *scikit-learn* da linguagem Python3, buscando assim, garantir que a acurácia de cada classificador melhor represente seus respectivos subconjuntos de treino.

Vale ressaltar que os resultados deste trabalho se valem apenas para a complexidade de subconjuntos gerados pelo algoritmo de *Bagging* e com classificadores do tipo *perceptron*, necessitando de um trabalhos futuros para validar os resultados em outras configurações.

4.3 Estimação das Acurácias dos Classificadores

Nesta etapa, após cada um dos subconjuntos gerados pelo *Bagging* terem sido usados para criar modelos de classificação, estes são submetido à estimação de sua acurácia.

O processo consiste em empregar cada um dos 100 (cem) classificadores gerados na classificação do conjunto de teste e estimar o percentual de instâncias rotuladas corretamente. O valor resultante é um índice pertencente ao intervalo $[0, 1]$ e corresponde à relação do número de acertos perante o total de instâncias do conjunto.

4.4 Estimação das Métricas de Complexidade

Assim que os subconjuntos são construídos é realizada a fase de estimação da assinatura de complexidade. Para tal, foi empregada a biblioteca ECoL (*Extended Complexity Library*) (**??**) que é uma biblioteca de aprendizado de máquina, implementada na linguagem de programação R, que dispõe de um total de vinte e nove descritores de complexidade para problemas de classificação e regressão, dos quais, por serem os descritores para problemas de classificação mais comuns na literatura, apenas doze serão empregados neste trabalho. Tais descritores são apresentados na Tabela 4.2.

Tabela 4.2: Métricas presentes na biblioteca ECoL adotadas no trabalho

Métricas de Complexidade	**Sigla**
Relação Máxima do Discriminante de Fischer	F1
Sobreposição de Atributos por Classe	F2
Eficiência Máxima por Atributo Individual	F3
Eficiência Coletiva dos Atributos	F4
Soma Minimizada da Distância de Erro de um Classificador Linear	L1
Taxa de Erro de um Classificador Linear sobre o Treino	L2
Não-Linearidade de um Classificador	L3
Fração de Pontos na Região de Fronteiras	N1
Proporção das Distâncias Intra/Inter Classes até o vizinho mais próximo	N2
Taxa de erro do classificador KNN pela abordagem *Leave-One-Out*	N3
Não-Linearidade de um Classificador KNN	N4
Fração de Esfera de Cobertura Máxima	T1

A biblioteca ECoL por padrão retorna todos os índices de complexidades normalizados no intervalo $[0, 1]$ evitando assim que alguma das métricas tenha mais peso do que outra, dada sua faixa de variação.

4.5 Análise da Relação Acurácia vs Complexidade

Após estimar as métricas de complexidade e a acurácia dos classificadores, o passo seguinte consiste em analisar a relação entre cada métrica de complexidade e a acurácia dos classificadores treinados com cada subconjunto.

Visando ilustrar como o processo é realizado, considere um conjunto composto de cinco classificadores fictícios (C_1, C_2, C_3, C_4, C_5). A Tabela 4.3 apresenta as acurácias do conjunto de classificadores e a Tabela 4.4, as medidas da complexidade F2 estimada sobre os conjuntos de

treino dos cinco classificadores. Pode-se perceber que todos os índices levantados encontram-se no intervalo entre 0 e 1.

Tabela 4.3: Exemplo de acurácia de cinco classificadores fictícios

Acurácia dos Classificadores	
Classificador	Acurácia
C_1	0,8000
C_2	0,8500
C_3	0,9000
C_4	0,8700
C_5	0,8300

Tabela 4.4: Valor da medida F2 referente aos conjuntos usados para treinar os cinco classificadores fictícios

Medida F2 dos conjuntos	
Conjunto de treino	F2
CT_1	0,7300
CT_2	0,7000
CT_3	0,6500
CT_4	0,5700
CT_5	0,7000

A fim de estimar a relação entre as acurácias dos classificadores e as métricas de complexidade usadas para descrever a complexidade dos conjuntos sobre os quais tais classificadores foram treinados realizou-se a combinação de todos para todos os elementos do conjunto, formando assim uma combinação de n classificadores combinados dois a dois. O número total de combinações NTC pode ser estimado pela Equação 4.1. Tal comportamento foi obtido através do desenvolvimento da fórmula da combinação de n elementos tomados p a p, conforme apresentado na Equação 4.2. Dessa forma, para um conjunto composto por 100 classificadores, como o adotado neste trabalho, obtém-se um total de 4950 combinações possíveis.

$$NTC = \frac{n^2 - n}{2} \tag{4.1}$$

$$C_{n,p} = \frac{n!}{p!(n-p)!} = \frac{n * (n-1) * (n-2)!}{2!(n-2)!} = \frac{n * (n-1)}{2} = \frac{n^2 - n}{2} \tag{4.2}$$

A partir dos valores de acurácia e da medida F2 apresentados nas tabelas 4.3 e 4.4, respectivamente, é possível calcular a discrepância dos comportamentos de dois classificadores. Na

Tabela 4.5 são apresentados os valores das diferenças em termos de acurácia para cada combinação de classificadores. Já na Tabela 4.6 são apresentadas as diferenças entre os pares de conjuntos de treino usados nos mesmos classificadores, mas agora no espaço de complexidade da medida F2.

Tabela 4.5: Exemplo de tabela de dissimilaridade da acurácia

	C_1	C_2	C_3	C_4	C_5
C_1	0,0000	-0,0500	-0,1000	-0,0700	-0,0300
C_2		0,0000	-0,0500	-0,0200	0,0200
C_3			0,0000	0,0300	0,0700
C_4				0,0000	0,0400
C_5					0,0000

Tabela 4.6: Exemplo de tabela de dissimilaridade da medida de complexidade

	CT_1	CT_2	CT_3	CT_4	CT_5
CT_1	0,0000	0,0300	0,0800	0,1600	0,0300
CT_2		0,0000	0,0500	0,1300	0,0000
CT_3			0,0000	0,0800	-0,0500
CT_4				0,0000	-0,0400
CT_5					0,0000

Por exemplo, supondo que um classificador C_1 tenha o valor de acurácia igual a 0,8 e seu conjunto de treino tenha a medida F2 igual a 0,73, e que um segundo classificador C_2 tenha 0,85 de acurácia e medida F2 igual a 0,7. Nesse caso, no campo das tabelas de dissimilaridade de acurácia e de F2 que representa os valores de $C_1 - C_2$, irá receber -0,05 (em destaque na Tabela 4.5) para a acurácia e 0,03 para F2 (em destaque na Tabela 4.6). É possível notar que quanto mais próximo de zero um campo da tabela é, mais semelhante é tal característica entre os classificadores. Tal fato é observado nas diagonais principais de ambas as tabelas onde há a combinação do classificador com ele próprio.

Após gerar as tabelas de dissimilaridade, o passo seguinte é a construção de uma tabela onde cada linha representa uma combinação de classificadores diferente em que as colunas representam os dados de uma tabela de dissimilaridade, sendo uma delas a de acurácia, e a outra de uma das métricas de complexidade (Tabela 4.7). Considerando o anterior, a linha que representa a combinação entre os classificadores C_1 e C_2 (em destaque na tabela) recebe o valor -0,05 na coluna da acurácia e 0,03 na coluna da medida de complexidade.

Dado o cenário ilustrado em que o *pool* era composto por cinco classificadores, a Tabela 4.7 é composta por dez linhas, descrevendo a relação entre todos os pares de classificadores.

Tabela 4.7: Exemplo de tabela de pontos do gráfico

	Dissimilaridade da acurácia	Dissimilaridade de F2
C_1 vs C_2	-0,0500	0,0300
C_1 vs C_3	-0,1000	0,0800
C_1 vs C_4	-0,0700	0,1600
C_1 vs C_5	-0,0300	0,0300
C_2 vs C_3	-0,0500	0,0500
C_2 vs C_4	-0,0200	0,1300
C_2 vs C_5	0,0200	0,0000
C_3 vs C_4	0,0300	0,0800
C_3 vs C_5	0,0700	-0,0500
C_4 vs C_5	0,0400	0,1300

De forma a melhor representar a diferença em termos de acurácia de complexidade dos classificadores comparados os valores apresentados na Tabela 4.7 podem ser plotados em um gráfico bidimensional em que o eixo das abscissas corresponde às diferenças em termos de acurácia (primeira coluna da tabela) e o eixo das ordenadas refere-se às diferenças em termos da medida de complexidade (segunda coluna da tabela).

Com base no exemplo construído, a Figura 4.2 apresenta a distribuição das diferenças encontradas entre os cinco classificadores nos dois espaços de interesse. Uma vez que a diferença máxima em termos de acurácia e complexidade estará dentro do intervalo $[-1, 1]$, o gráfico explora apenas esse espectro do plano cartesiano. Cada ponto representado no gráfico corresponde à dissimilaridade de dois classificadores. Assim sendo, são representados ao todo, dez pontos na ilustração, fruto de todas as combinações do cinco classificadores.

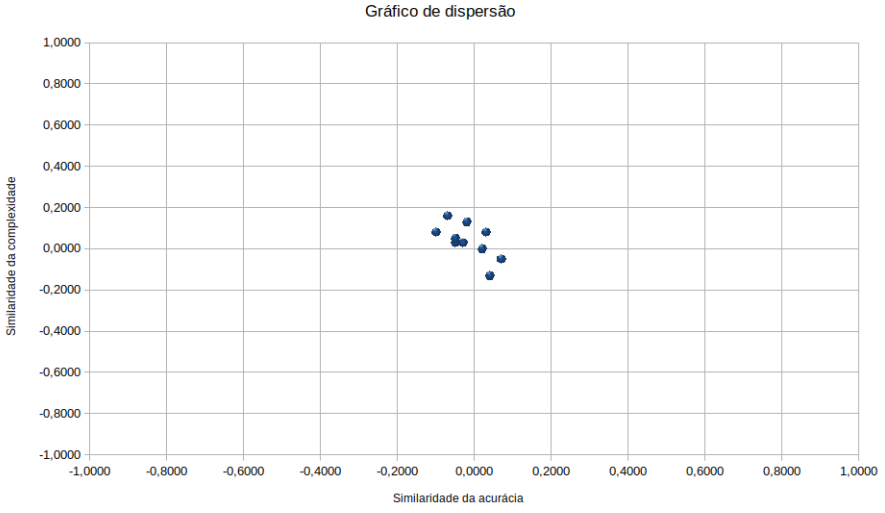

Figura 4.2: Exemplo de acurácia e métrica de complexidade.

A dispersão dos pontos no espaço de acurácia e complexidade reflete o grau de similaridade entre os classificadores. Caso o comportamento destes seja parecido, a tendência é que o ponto que representa a sua relação esteja próximo da origem. Por outro lado, se a diferença for grande, em qualquer dos eixos, a tendência é que o ponto esteja posicionado mais longe da origem.

Para quantificar a relação entre a acurácia e a métrica de complexidade optou-se em adotar a Distância Euclidiana Média (DEM) entre todos os pontos. A Equação 4.3 detalha como é calculada esta medida. Na equação, NTC representa o número total de pontos (proveniente da Equação 4.1), p corresponde a um ponto do conjunto e $\delta(p_i, p_j)$ refere-se à distância euclidiana entre os pontos i e j. Espera-se que quanto menor o DEM, maior a similaridade entre a acurácia e a complexidade dos classificadores do conjunto.

$$DEM = \frac{\sum_{i=1}^{NTC-1} \sum_{j=i+1}^{NTC} \delta(p_i, p_j)}{NTC} \tag{4.3}$$

Capítulo 5

Resultados e discussões

5.1 Análise da dispersão

Após a realização dos testes descritos na Seção 4.5, é possível analisar o comportamento de cada gráfico de dispersão em diferentes bases de dados. Tal análise torna evidente grandes diferenças na relação Acurácia-Complexidade de diferentes bases de dados.

Na análise da relação Acurácia-F3 (Figura 5.1) podemos ver uma variação mediana nos eixos da acurácia e no eixo da métrica F3 para a base Blood. Interessante observar que os classificadores apresentam "faixas" de acurácia que consistem em classificadores que apresentam comportamento muito parecido perante o conjunto de teste.

Na Figura 5.2 pode-se perceber uma alta variação no eixo da métrica F3 para a base Ionosphere e ao mesmo tempo, pouca variação no eixo das acurácias, indicando que classificadores tendem a ter desempenhos próximos apesar das variações na medida F3.

Já a Figura 5.3 apresenta uma baixa variação em ambos os eixos (variação na medida F3 e acurácia) para a base WDVG. Percebe-se que praticamente toda a concentração dos pontos ocorre no espaço [-0,15,0,15] para as duas variáveis estimadas.

Tais comportamentos, por sua vez, ficam evidenciados ao se comparar Distância Euclidiana Média (DEM) de cada uma das bases. O valor de dispersão para a base Blood (no exemplo ilustrado) foi de 0,4394. A base Ionosphere, na repetição apresentada, alcancçou dispersão de 0,2322. Já a base WDVG, que mostrou-se muito mais concentrada, indicando grande similaridade entre a acurácia e F3, apresentou um valor de 0,0661. Tais valores serão melhor apresentados na Seção 5.2.

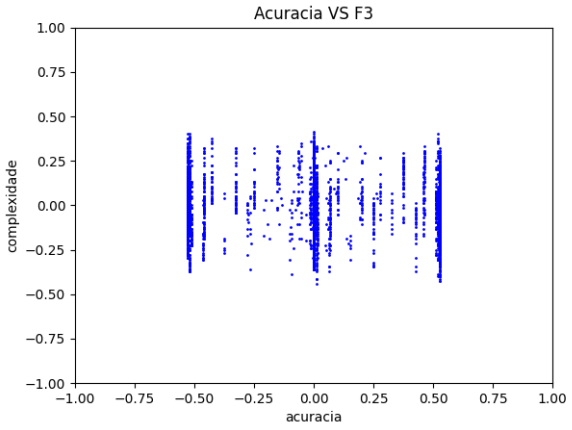

Figura 5.1: Comportamento da medida F3 para a base Blood para a segunda repetição

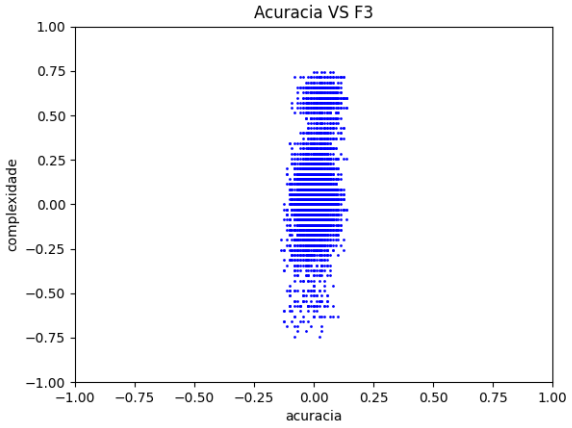

Figura 5.2: Comportamento da medida F3 para a base Ionosphere para a primeira repetição

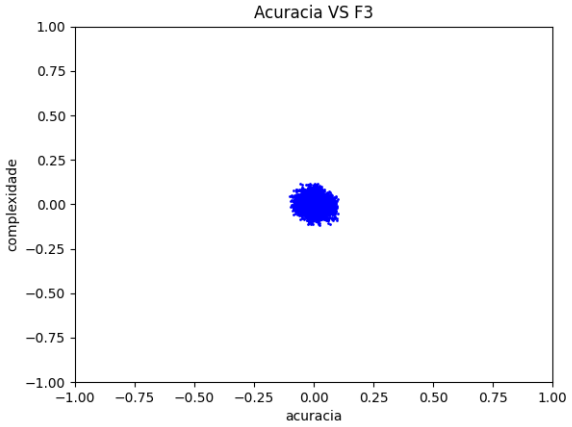

Figura 5.3: Comportamento da medida F3 para a base WDVG para a segunda repetição

Outro apontamento observado diz respeito às métricas de complexidade que apresentaram valor semelhante para todos os conjuntos, gerando assim um gráfico que variou apenas na acu-

rácia. A Figura 5.4 ilustra um exemplo de métrica que não varia para todos os conjuntos. No cenário, a medida F2 não apresentou variações entre os 100 conjuntos usados no treinamento dos classificadores, apenas na acurácia dos mesmos, variando no intervalo [-0,2, 02]. O valor observado para DEM neste caso foi de 0,0492, um valor baixo em comparação aos demais.

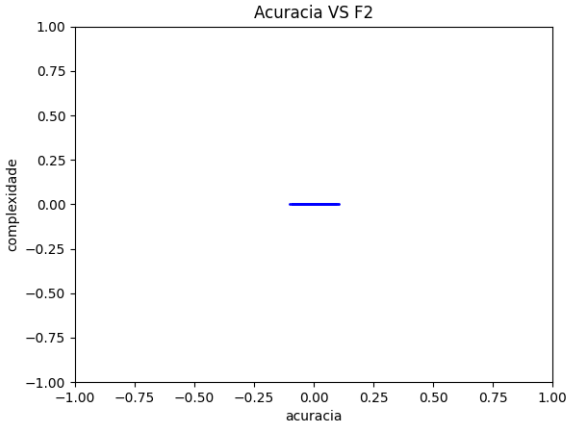

Figura 5.4: Comportamento da medida F2 para a base Adult durante a segunda repetição

Ao contrário do observado na Figura 5.4, há casos onde o gráfico apresenta boa dispersão, como pode ser observado na Figura 5.5. No entanto, apesar da dispersão cobrir uma região mais abrangente do espaço, o valor para DEM foi de 0,3384, o que indica que ainda há concentração dos elementos próximos da origem do sistema.

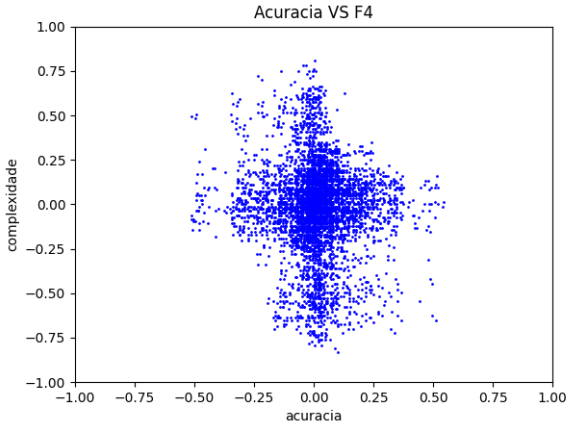

Figura 5.5: Comportamento da medida F4 para a base Mammo durante a quarta repetição

Outro comportamento interessante observado ao longo das repetições é apresentado na Figura 5.6. Na ilustração é possível perceber que os classificadores apresentaram faixas de acurácia e, ao longo dessas, pequenas variações na medida T1. Neste caso o valor obtido para DEM

foi de 0,3890, confirmando certa dissimilaridade entre o comportamento em termos de acurácia frente aos valores da medida T1 estimada sobre os conjuntos em que foram treinados.

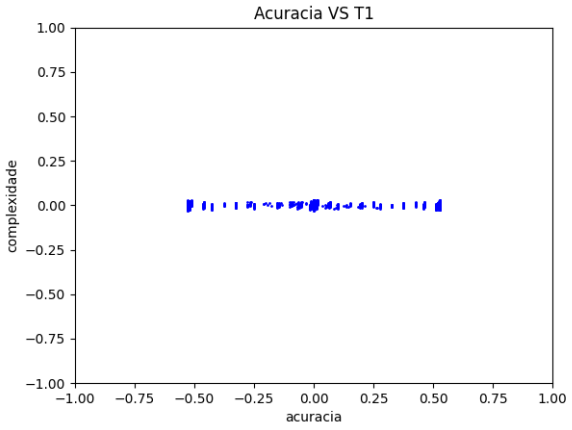

Figura 5.6: Comportamento da medida T1 para a base Blood durante a segunda repetição

5.2 Análise da Distância Euclidiana Média (DEM)

A fim de validar o experimento, repetiu-se 10 vezes o processo, desde a geração dos subconjuntos até a obtenção dos DEMs e, então, calculou-se a média dos resultados. A Tabela 5.1 contém as médias e desvios padrão das 10 repetições para cada base (linhas) e métrica de complexidade (colunas). Destaca-se em vermelho (negritados) a métrica que obteve o menor valor de dispersão para cada base, sendo assim a métrica com maior relação complexidade-acurácia. Por outro lado, em azul (sublinhados) são representadas as métricas com maior valor de DEM. Tais valores são as combinações de acurácia e medida de complexidade que apresentam as maiores variações.

A partir do valores médios das DEMs pode-se analisar quais métricas obtiveram os menores e maiores valores para cada base de dados, sendo esta informação inversamente proporcional à relação da métrica com acurácia. Percebe-se que a métrica que conseguiu mais vezes o menor valor de DEM é F2, sendo este o menor em 14 das 26 bases de dados. A medida L1 apresentou a maior similaridade (menor DEM) em 6 bases. Já as medidas L2, L3 e N3 apresentaram a menor dispersão para duas bases presentes no conjunto. Por fim, a F4 teve a menor dispersão para a base Segmentation.

Tabela 5.1: Valor médio das dispersões para cada medida de complexidade considerando 10 repetições executadas

	F1	F2	F3	F4	L1	L2	L3	N1	N2	N3	N4	T1
Adult	0,085 (0,021)	**0,072 (0,024)**	0,177 (0,057)	0,151 (0,094)	0,086 (0,028)	0,089 (0,027)	0,088 (0,029)	0,145 (0,029)	0,089 (0,021)	0,094 (0,023)	0,077 (0,022)	0,073 (0,021)
Banana	0,142 (0,034)	0,154 (0,039)	0,164 (0,036)	0,201 (0,044)	0,133 (0,034)	0,144 (0,033)	0,147 (0,035)	0,137 (0,034)	0,131 (0,035)	**0,130 (0,035)**	0,146 (0,034)	0,133 (0,034)
Blood	0,400 (0,019)	0,456 (0,045)	0,439 (0,019)	0,470 (0,021)	0,396 (0,016)	0,396 (0,017)	0,420 (0,016)	0,426 (0,014)	0,400 (0,014)	0,405 (0,014)	0,415 (0,016)	**0,384 (0,016)**
CTG	0,051 (0,005)	**0,029 (0,003)**	0,127 (0,026)	0,048 (0,015)	0,032 (0,004)	0,035 (0,004)	0,034 (0,004)	0,063 (0,004)	0,040 (0,003)	0,042 (0,003)	0,041 (0,004)	0,030 (0,003)
Diabetes	0,159 (0,018)	0,164 (0,028)	0,215 (0,041)	0,275 (0,050)	0,162 (0,020)	0,179 (0,020)	0,188 (0,023)	0,204 (0,017)	0,162 (0,019)	0,172 (0,020)	0,166 (0,021)	**0,148 (0,020)**
Faults	0,071 (0,009)	**0,051 (0,006)**	0,070 (0,006)	0,056 (0,006)	0,052 (0,006)	0,052 (0,006)	0,052 (0,006)	0,090 (0,007)	0,063 (0,006)	0,071 (0,006)	0,064 (0,007)	**0,051 (0,006)**
German	0,057 (0,007)	0,111 (0,063)	0,144 (0,027)	0,400 (0,054)	0,079 (0,007)	0,078 (0,008)	0,081 (0,010)	0,120 (0,008)	0,071 (0,006)	0,080 (0,005)	0,058 (0,007)	**0,053 (0,007)**
Haberman	0,238 (0,073)	0,287 (0,069)	0,318 (0,066)	0,393 (0,071)	0,232 (0,073)	0,242 (0,075)	0,287 (0,068)	0,298 (0,059)	0,236 (0,074)	0,256 (0,067)	0,263 (0,066)	**0,214 (0,080)**
Heart	0,119 (0,019)	**0,092 (0,016)**	0,286 (0,049)	0,125 (0,026)	**0,092 (0,022)**	0,093 (0,022)	**0,092 (0,021)**	0,229 (0,020)	0,124 (0,015)	0,137 (0,014)	0,096 (0,021)	0,101 (0,020)
ILPD	0,074 (0,012)	**0,064 (0,011)**	0,147 (0,024)	0,100 (0,017)	0,088 (0,012)	0,116 (0,010)	0,133 (0,018)	0,152 (0,012)	0,089 (0,009)	0,107 (0,010)	0,102 (0,008)	0,065 (0,011)
Ionosphere	0,084 (0,016)	**0,066 (0,013)**	0,232 (0,042)	0,099 (0,027)	**0,066 (0,013)**	0,069 (0,012)	0,068 (0,013)	0,162 (0,018)	0,093 (0,009)	0,099 (0,007)	0,096 (0,013)	0,071 (0,013)
Laryngeal1	0,155 (0,019)	**0,103 (0,017)**	0,255 (0,048)	0,156 (0,070)	0,112 (0,017)	0,124 (0,015)	0,121 (0,017)	0,258 (0,028)	0,147 (0,017)	0,169 (0,022)	0,138 (0,018)	0,131 (0,017)
Laryngeal3	0,136 (0,018)	**0,078 (0,015)**	0,125 (0,015)	0,096 (0,016)	0,081 (0,015)	0,084 (0,015)	0,085 (0,016)	0,191 (0,019)	0,112 (0,012)	0,132 (0,016)	0,113 (0,017)	0,083 (0,014)
Lithuanian	0,144 (0,012)	0,155 (0,020)	0,171 (0,023)	0,175 (0,039)	0,137 (0,012)	0,146 (0,012)	0,148 (0,012)	0,141 (0,012)	0,134 (0,013)	**0,133 (0,013)**	0,141 (0,013)	0,134 (0,013)
Liver	0,133 (0,009)	0,135 (0,015)	0,199 (0,021)	0,181 (0,038)	0,150 (0,011)	0,186 (0,014)	0,206 (0,020)	0,228 (0,013)	0,150 (0,009)	0,175 (0,012)	0,166 (0,010)	**0,120 (0,009)**
Mammo	0,150 (0,034)	0,187 (0,064)	0,197 (0,033)	0,332 (0,075)	0,126 (0,036)	0,138 (0,036)	0,149 (0,037)	0,177 (0,030)	0,125 (0,035)	0,129 (0,035)	0,146 (0,034)	**0,111 (0,036)**
Monk	0,094 (0,016)	0,212 (0,094)	0,152 (0,032)	0,121 (0,065)	0,101 (0,019)	0,121 (0,027)	0,126 (0,032)	0,181 (0,019)	0,102 (0,016)	0,122 (0,018)	0,096 (0,020)	**0,077 (0,018)**
Phoneme	0,101 (0,012)	0,126 (0,013)	0,113 (0,013)	0,121 (0,011)	0,101 (0,014)	0,110 (0,012)	0,115 (0,012)	0,111 (0,011)	0,100 (0,013)	0,102 (0,012)	0,107 (0,012)	**0,097 (0,013)**
Segmentation	0,136 (0,017)	**0,133 (0,017)**	0,138 (0,017)	**0,133 (0,017)**	**0,133 (0,017)**	**0,133 (0,017)**	0,137 (0,017)	0,148 (0,016)	0,136 (0,017)	0,137 (0,017)	0,137 (0,017)	**0,133 (0,017)**
Sonar	0,136 (0,016)	**0,115 (0,017)**	0,294 (0,032)	0,156 (0,037)	**0,115 (0,017)**	**0,115 (0,017)**	**0,115 (0,017)**	0,257 (0,016)	0,155 (0,015)	0,177 (0,015)	0,120 (0,016)	0,118 (0,016)
Thyroid	0,056 (0,010)	**0,044 (0,009)**	0,094 (0,026)	0,052 (0,012)	0,046 (0,010)	0,050 (0,010)	0,051 (0,012	0,082 (0,013)	0,064 (0,009)	0,058 (0,010)	0,049 (0,012)	0,144 (0,023)
Vehicle	0,124 (0,013)	0,094 (0,012)	0,133 (0,014)	0,106 (0,014)	0,095 (0,011)	0,097 (0,011)	0,098 (0,011)	0,151 (0,011)	0,109 (0,011)	0,122 (0,011)	0,124 (0,011)	**0,093 (0,012)**
Vertebral	0,149 (0,015)	**0,090 (0,012)**	0,218 (0,028)	0,138 (0,060)	0,106 (0,017)	0,123 (0,020)	0,123 (0,025)	0,198 (0,018)	0,121 (0,009)	0,128 (0,015)	0,125 (0,014)	0,106 (0,015)
WBC	0,258 (0,059)	**0,238 (0,065)**	0,281 (0,056)	0,239 (0,066)	**0,238 (0,065)**	0,239 (0,065)	0,239 (0,065)	0,267 (0,060)	0,250 (0,061)	0,246 (0,062)	0,240 (0,065)	0,251 (0,062)
WDVG	0,042 (0,004)	**0,038 (0,005)**	0,066 (0,006)	0,040 (0,005)	0,041 (0,005)	0,042 (0,005)	0,041 (0,005)	0,064 (0,004)	0,045 (0,004)	0,047 (0,005)	0,040 (0,005)	**0,038 (0,005)**
Weaning	0,109 (0,019)	**0,093 (0,019)**	0,236 (0,035)	0,151 (0,050)	**0,093 (0,019)**	0,095 (0,020)	0,094 (0,020)	0,202 (0,019)	0,129 (0,016)	0,145 (0,018)	0,109 (0,018)	0,095 (0,019)

Com relação aos comportamentos de maior dispersão podemos destacar a medida F3 que obteve o maior valor para DEM em 12 das 26 bases testadas. Além de F3, outra medida que apresentou valores altos para DEM foi F4, tendo alcançado o maior valor em 7 bases. A medida N1 apresentou a maior dispersão em quatro bases de dados, enquanto F2 obteve os maiores valores de DEM em duas bases. Já a medida T1 apresentou a maior dispersão apenas para a base Thyroid.

Além da análise de similaridade das medidas de complexidade perante o comportamento da acurácia dos classificadores, analisou-se também as variações das métricas de uma mesma base de dados. Neste sentido destacou-se a base Blood ao apresentar os maiores valores de DEM em todas as métricas, variando de 0,470 em F4 e 0,384 em T1 (Figura 5.7).

A base WDVG, por sua vez, apresentou os menores valores de dispersão média, tendo como maior valor F3 = 0,066 e menor valor F2 = 0,038 (Figura 5.8).

Em outros cenários, como para a base German, observou-se um comportamento mais discrepante entre as métricas, como representado na Figura 5.9. Pode-se perceber que o valor de F4 mostrou-se consideravelmente maior aos demais índices de complexidade. Ela apresentou um DEM de 0,4000 enquanto a média das outras onze métricas foi de 0,0847.

Além dos gráficos dos valores de DEM para cada base de dados aqui apresentados, os gráficos restantes são apresentados no Apêndice A deste trabalho.

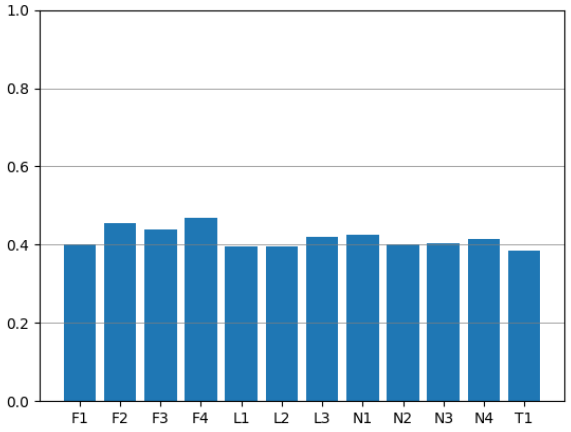

Figura 5.7: Comportamento médio das 12 medidas de complexidade para a base Blood

Visando analisar se as medidas apresentaram comportamento estatisticamente significativo ao longo das 26 bases de dados aplicou-se o teste de Kruskal-Wallis com 5% de significância. Os resultados indicaram rejeição da hipótese H_0, indicando que há pelo menos uma métrica de

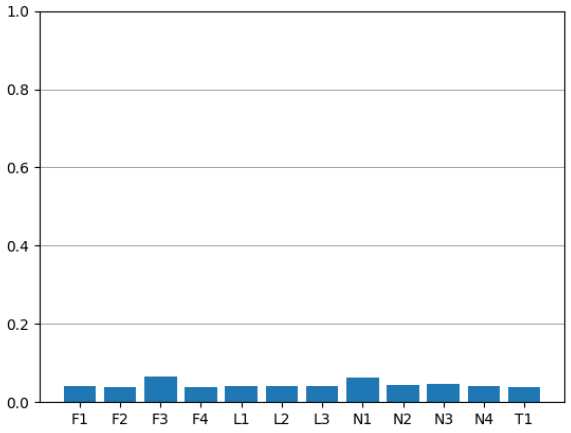

Figura 5.8: Comportamento médio das 12 medidas de complexidade para a base WDVG

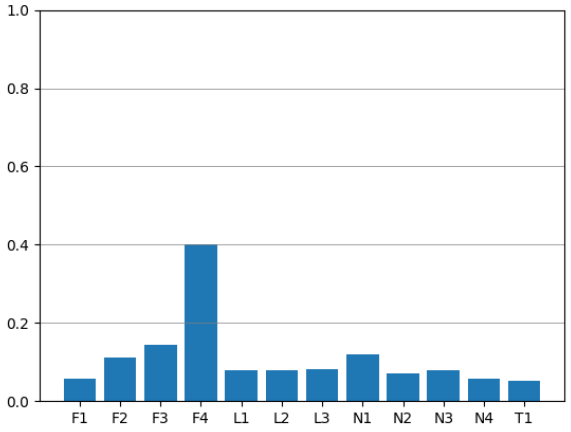

Figura 5.9: Comportamento médio das 12 medidas de complexidade para a base German

complexidade com comportamento distinto das demais.

Para criar um *ranking* das métricas de complexidade (Tabela 5.2) e identificar aquelas que mais influenciam a acurácia de um classificador utilizou-se o teste *post-hoc* de *Nemenyi*, tendo como entrada as médias disponíveis na Tabela 5.1.

Com base nos resultados do teste de *Nemenyi*, obteve-se o valor para determinar se existe, ou não, uma diferença estatisticamente significativa entre cada par de medidas de complexidade. A Figura 5.10 apresenta uma matriz de diferença todos-com-todos em que quanto mais escuro é a intersecção da métricas, mais distintos são os conjuntos. As métricas da matriz estão ordenadas de acordo com a Tabela 4.2. Pode-se perceber que em grande parte das combinações não há diferença significativa. No entanto, em diversos casos observou-se que as métricas apresentaram

Tabela 5.2: Ranking das métricas de complexidade

Medida	Ranking Médio
L1	3,1346
T1	3,1923
F2	4,2885
L2	5,3269
N2	5,6923
L3	6,0777
N4	6,1538
F1	6,3846
N3	7,5769
F4	8,8077
N1	10,3846
F3	11,0000

variações significativas ao longo das 26 bases de dados.

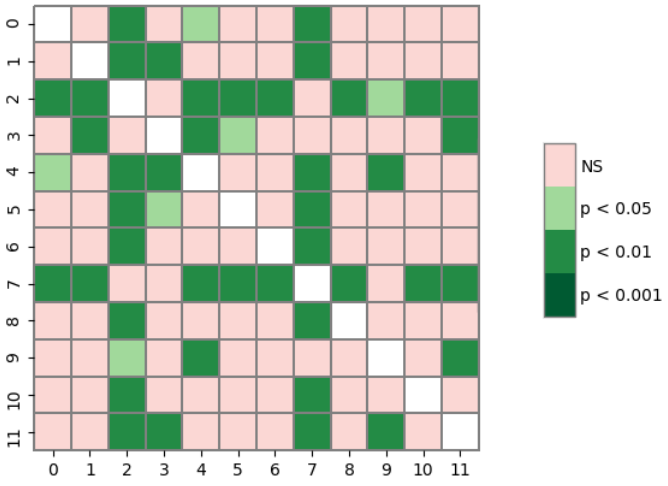

Figura 5.10: Relação de diferença entre métricas

Dado os resultados do teste de *Nemenyi*, também é possível construir um gráfico de diferença crítica (Figura 5.11), onde gráficos são ordenados das melhores métricas de complexidade para as piores, e as barras horizontais correspondem à distância crítica (DC). Caso a diferença de rankings médios entre as medidas seja maior que DC considera-se que a diferença entre elas é significativa.

Os valores dos rankings médios de cada medida de complexidade, para as 26 bases de dados são apresentados na Tabela 5.2. Com base nos valores tabelados e na Figura 5.11 podemos ver que a métrica com maior relação com a acurácia é a L1, tendo, porém, diferença crítica

apenas com as medidas F3, N1, F4, N3 e F1, sendo não significativa a diferença entre L1 e as demais métricas. Já F3 mostrou-se a pior métrica em relação a acurácia, tendo uma diferença não significativa apenas com N1 e F4.

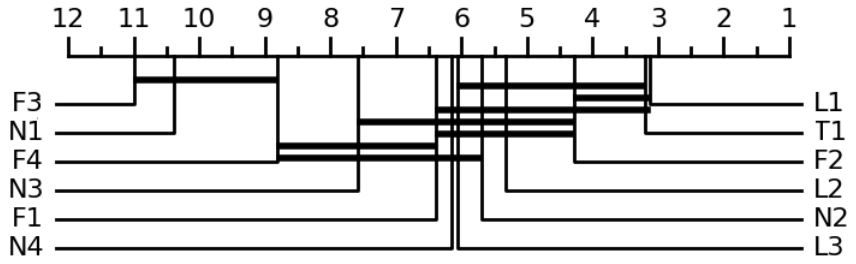

Figura 5.11: Gráfico de diferença crítica das métricas de complexidade

Notou-se que, apesar de ter o maior número de menores resultados, F2 apresentou apenas o terceiro melhor desempenho entre as métricas de maior relação com a acurácia. Isso pode indicar maior instabilidade da métrica perante diferentes conjunto. Já a métrica F3, que havia sido a métrica com os maiores resultados, teve este resultado confirmado pelo teste de *Nemenyi*, sendo a métrica com menor relação com a acurácia dos classificadores.

Capítulo 6

Considerações finais

Com base no fato de que há relação entre o comportamento dos classificadores e os conjuntos sobre os quais eles foram treinados, este trabalho propôs: elencar quais métricas de complexidade retiradas do conjunto de treino de um classificador melhor se relacionam com a acurácia do mesmo, no sentido de que ao treinar classificadores com conjuntos de complexidade semelhante, obtém-se resultados em termos de acurácia semelhante.

Para realizar tal análise, desenvolveu-se um protocolo para a geração de subconjuntos através do algoritmo *Bagging* que eram, então, usados no treinamento de perceptrons. Além disso, a partir de tais conjuntos foram obtidos seus descritores de complexidade.

Com o intuito de maior robustez na análise, foram estudadas 26 bases de dados, cada uma gerando 100 subconjuntos por repetição, em um total de 10 repetições.

Ao fim deste trabalho, percebe-se que existe relação entre algumas métricas de complexidade provenientes dos conjuntos de treino e a acurácia de classificadores. Dentre as métricas que apresentaram comportamento mais similar entre a acurácia e a complexidade dos conjuntos de treino podemos destacar L1, T1, F2 e L2. Por outro lado, as medidas que mostraram menor similaridade em termos de complexidade foram F3, N1, F4 e N3.

Porém, como todos os testes foram feitos com classificadores lineares do tipo *perceptron*, não é possível afirmar que outras configurações de classificadores apresentem os mesmos resultados, sendo assim necessário uma análise mais completa contando com diferentes estratégias de classificação.

6.1 Trabalhos futuros

Dadas as dificuldades enfrentadas ao decorrer deste trabalho, e as limitações do mesmo, propõem-se como trabalhos futuros os seguintes itens:

- Desenvolver análise de relação complexidade-acurácia utilizando classificadores baseados em conceitos diferentes do *perceptron*, como KNN, SVM, Naive Bayes ou mesmo um Multlayer Perceptron.

  Este trabalho utilizou apenas classificadores do tipo *perceptron* para análise. Percebeu-se então que aqueles que se utilizam de técnicas de classificação linear, como L1 e L2, obtiveram bons resultados, enquanto técnicas que utilizam estratégias de vizinhança, como N1 e N3, e técnicas inspiradas no funcionamento de árvores de decisão, como F3 e F4, obtiveram resultados ruins.

  Tem-se, então, a necessidade de validar os resultados comparando com os obtidos com outras estratégias de classificação.

- Estudo da relação das métricas de complexidade com a acurácia em algoritmos de regressão:

  Tendo em vista que a biblioteca ECoL apresenta métricas de complexidade para problemas de regressão, acredita-se ser útil uma análise da relação dessas métricas com a acurácia dos algoritmos de regressão.

Apêndice A

Gráficos

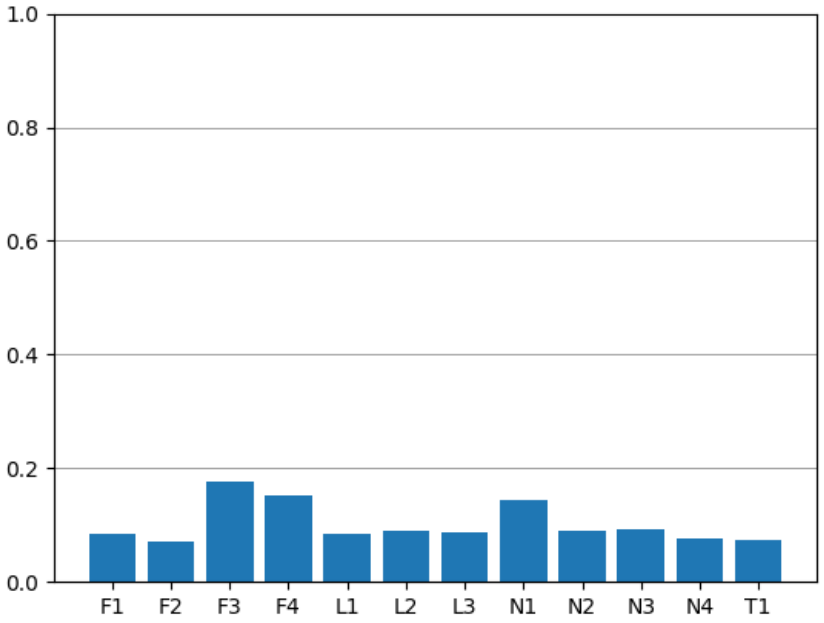

Figura A.1: Comparação entra métricas da base Adult

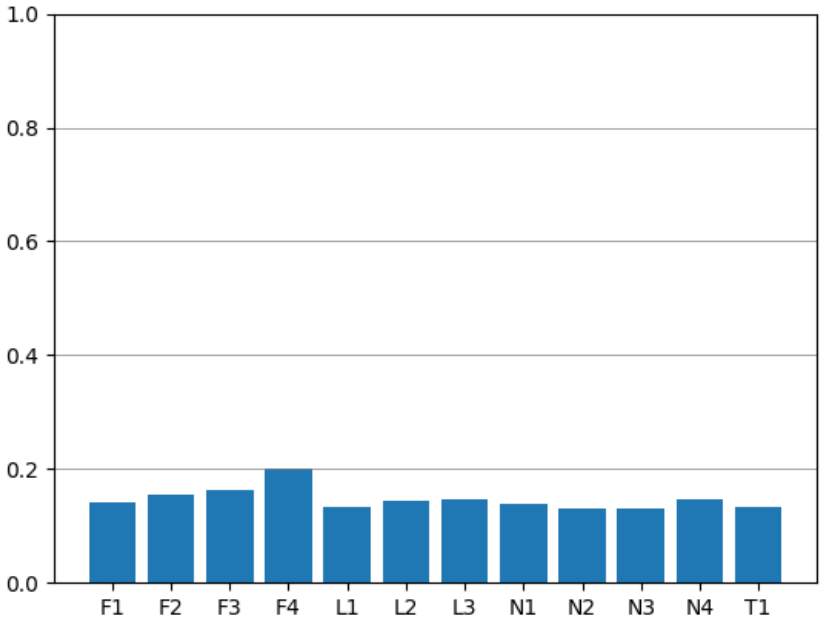

Figura A.2: Comparação entra métricas da base Banana

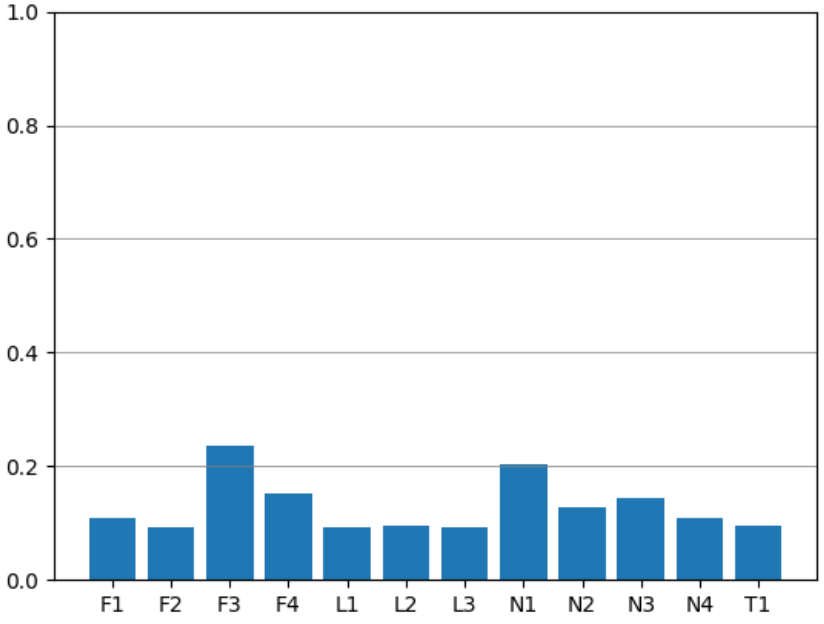

Figura A.26: Comparação entra métricas da base Weaning

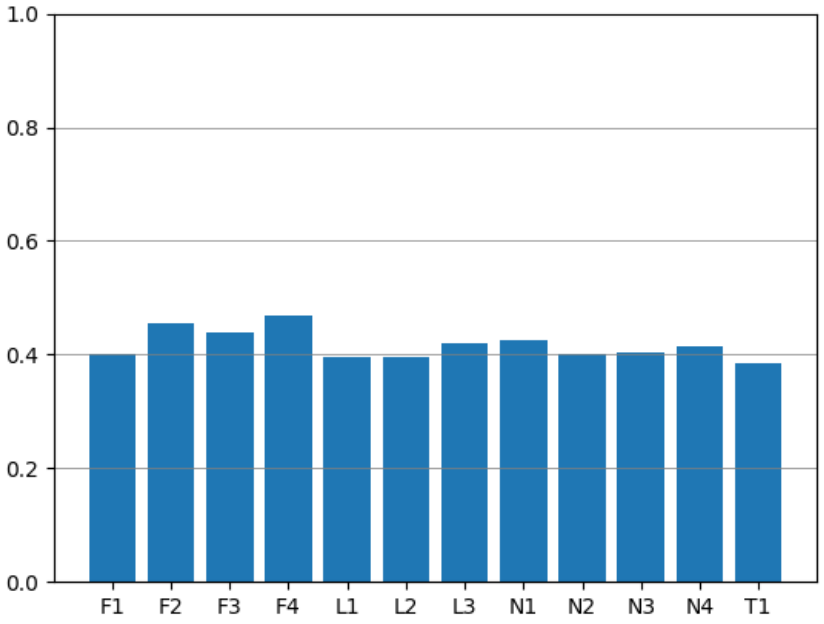

Figura A.3: Comparação entra métricas da base Blood

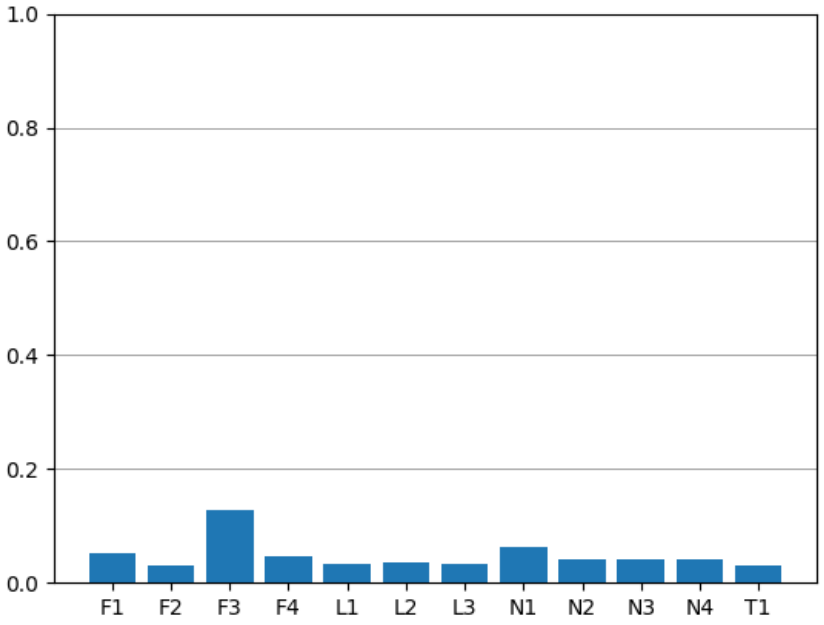

Figura A.4: Comparação entra métricas da base CTG

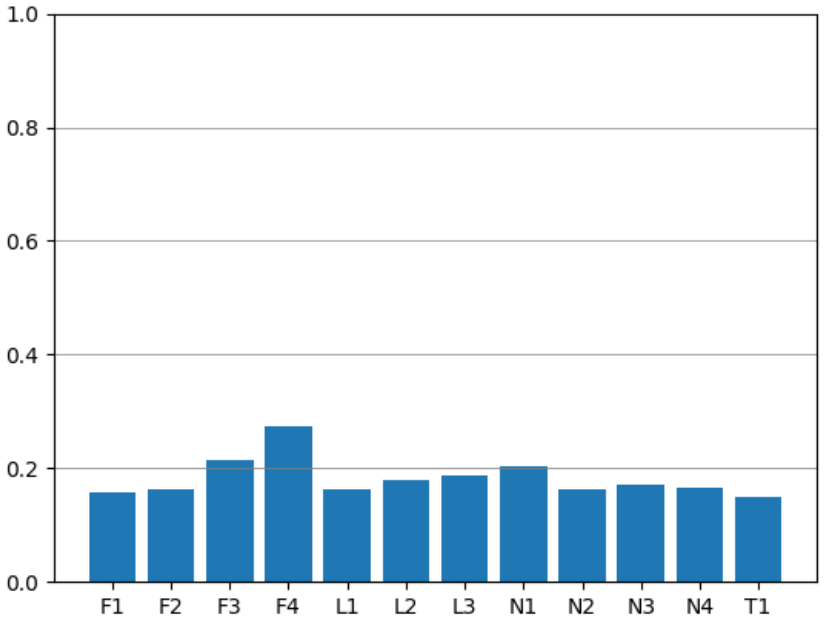

Figura A.5: Comparação entra métricas da base Diabetes

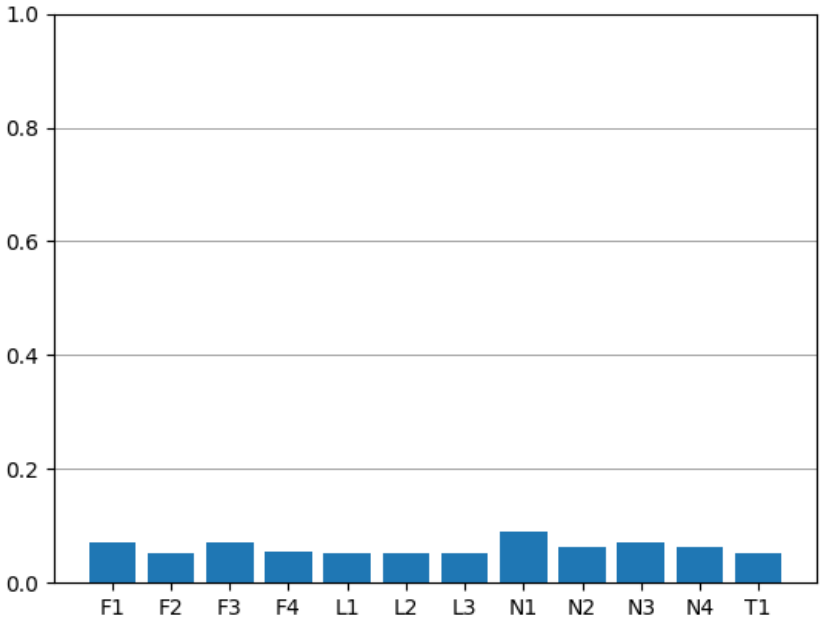

Figura A.6: Comparação entra métricas da base Faults

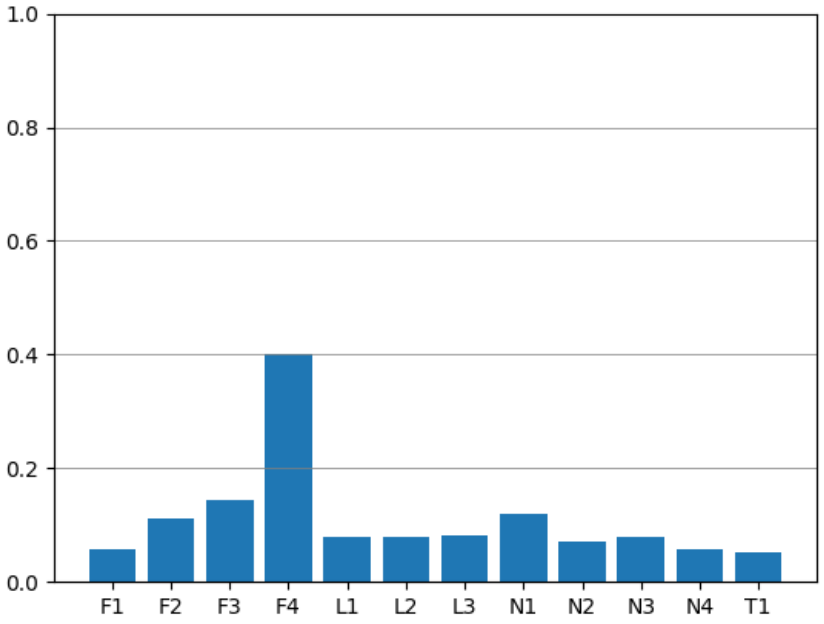

Figura A.7: Comparação entra métricas da base German

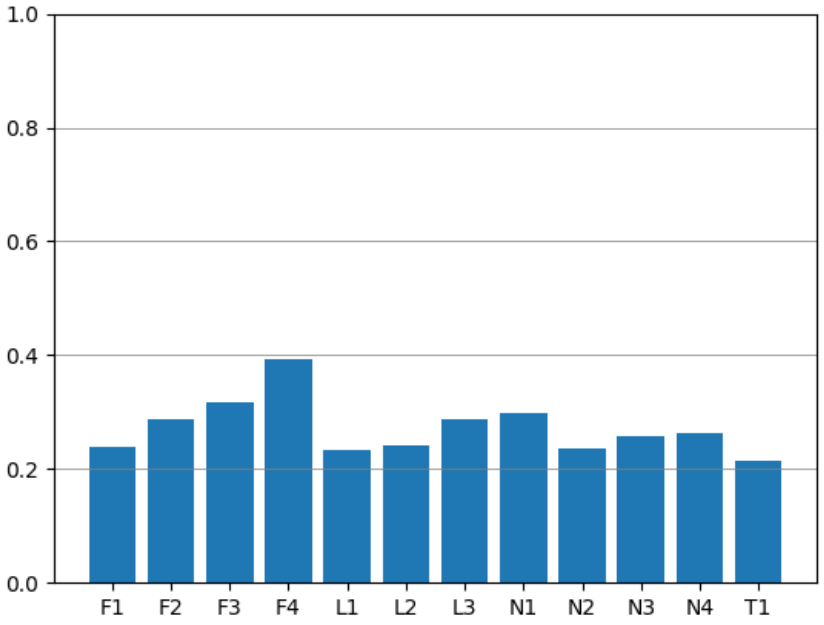

Figura A.8: Comparação entra métricas da base Haberman

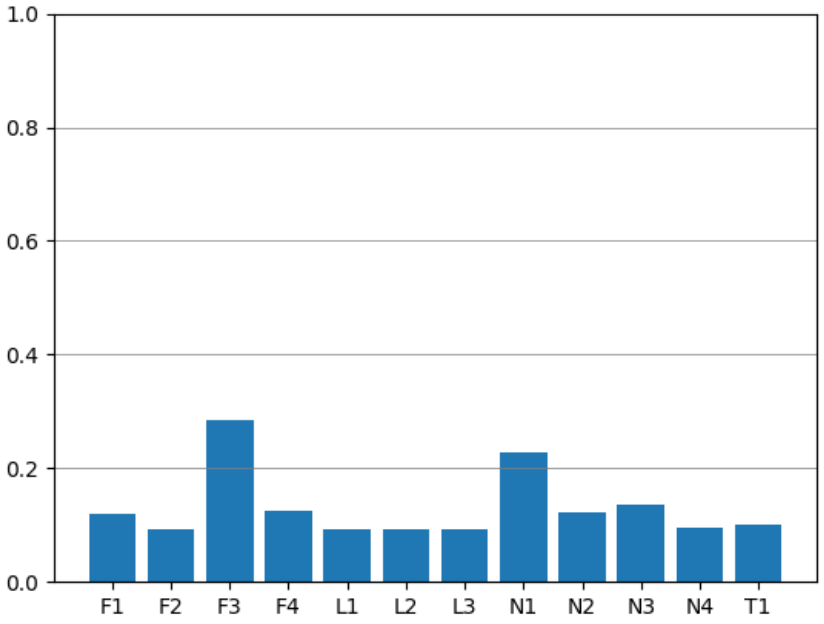

Figura A.9: Comparação entra métricas da base Heart

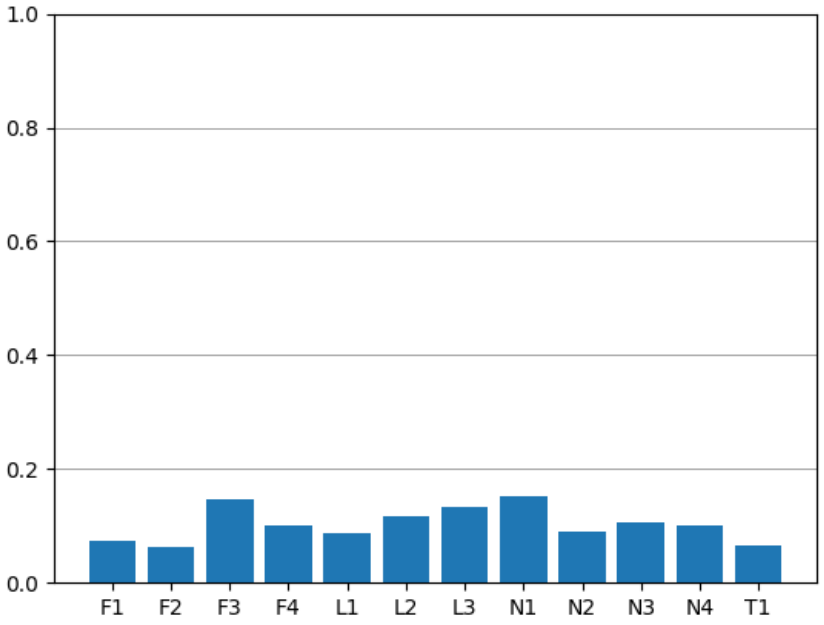

Figura A.10: Comparação entra métricas da base ILPD

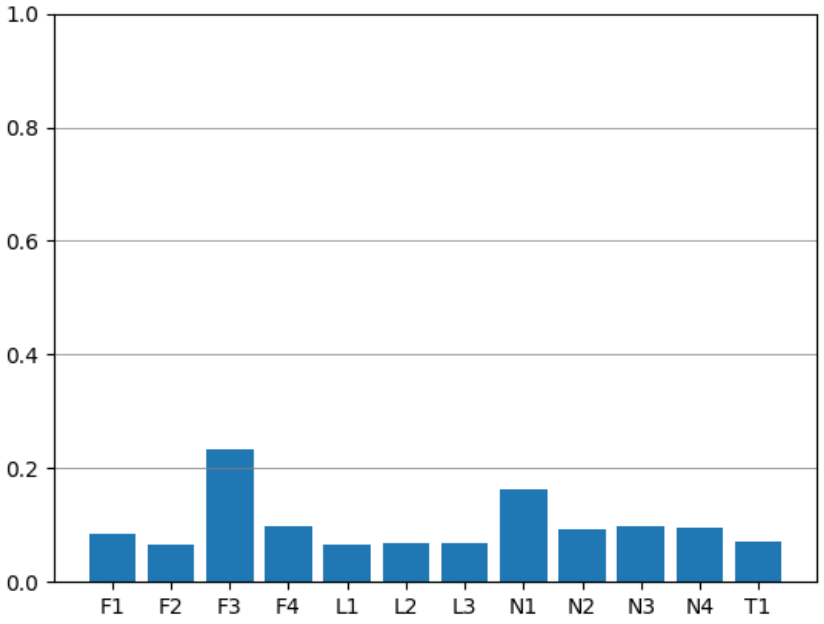

Figura A.11: Comparação entra métricas da base Ionosphere

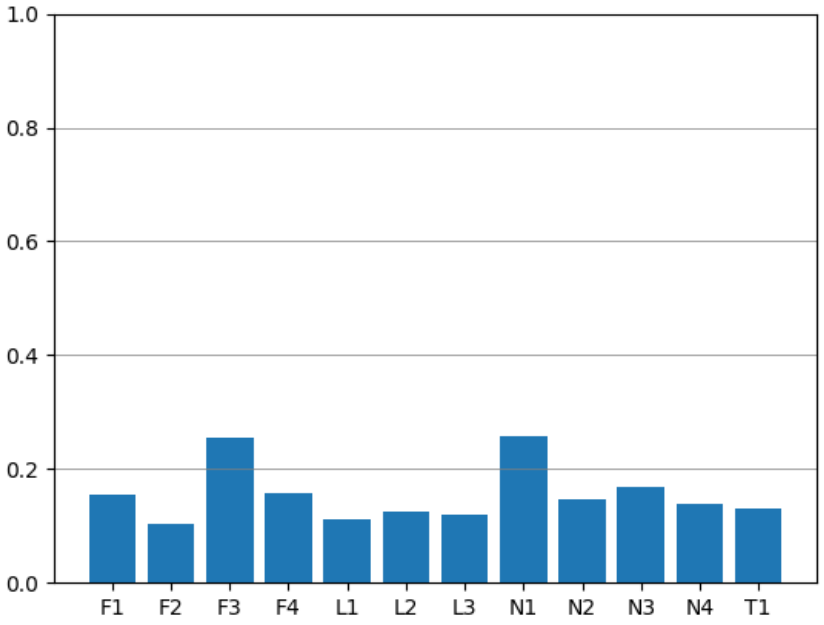

Figura A.12: Comparação entra métricas da base Laryngeal1

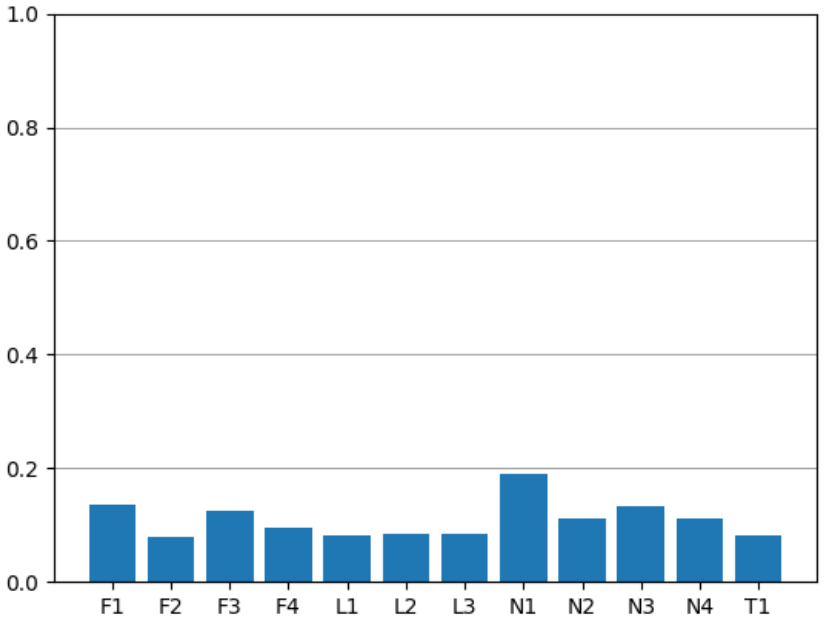

Figura A.13: Comparação entra métricas da base Laryngeal3

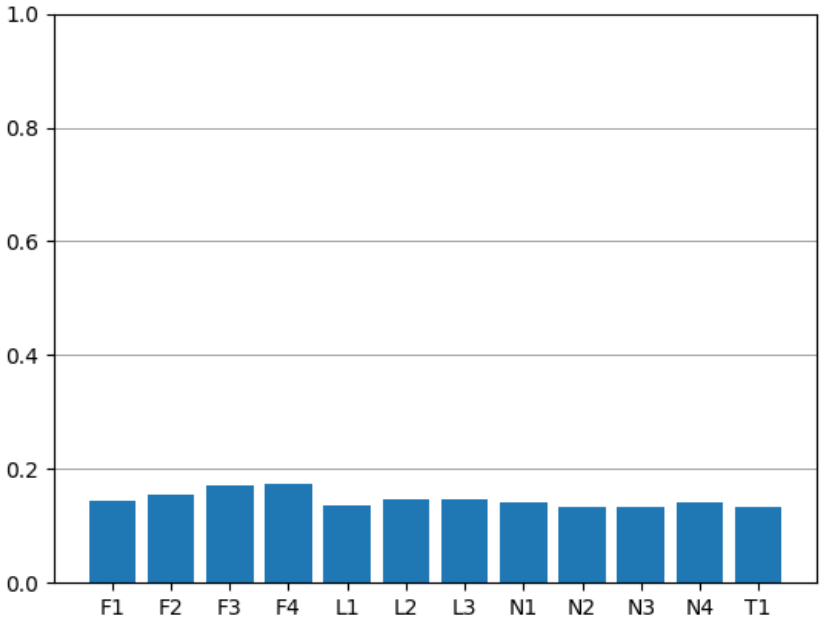

Figura A.14: Comparação entra métricas da base Lithuanian

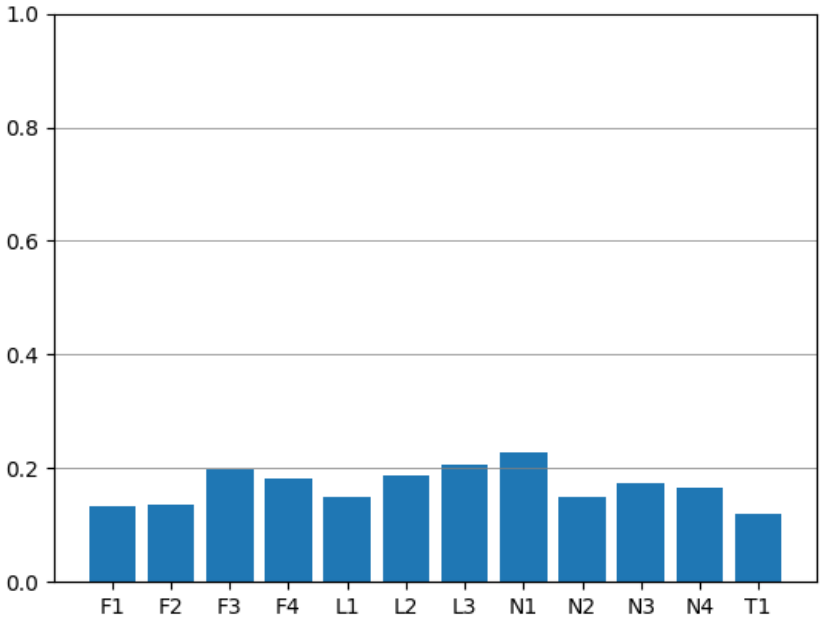

Figura A.15: Comparação entra métricas da base Liver

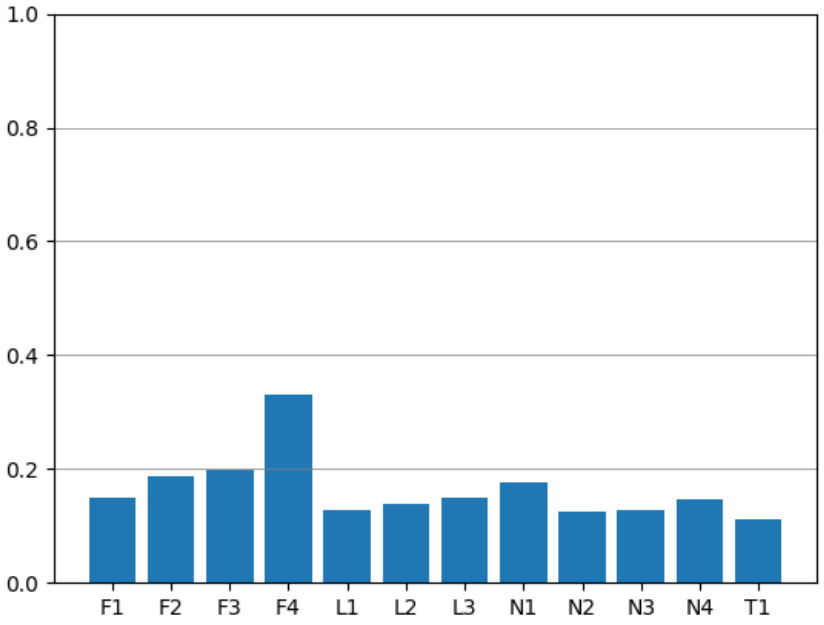

Figura A.16: Comparação entra métricas da base Mammo

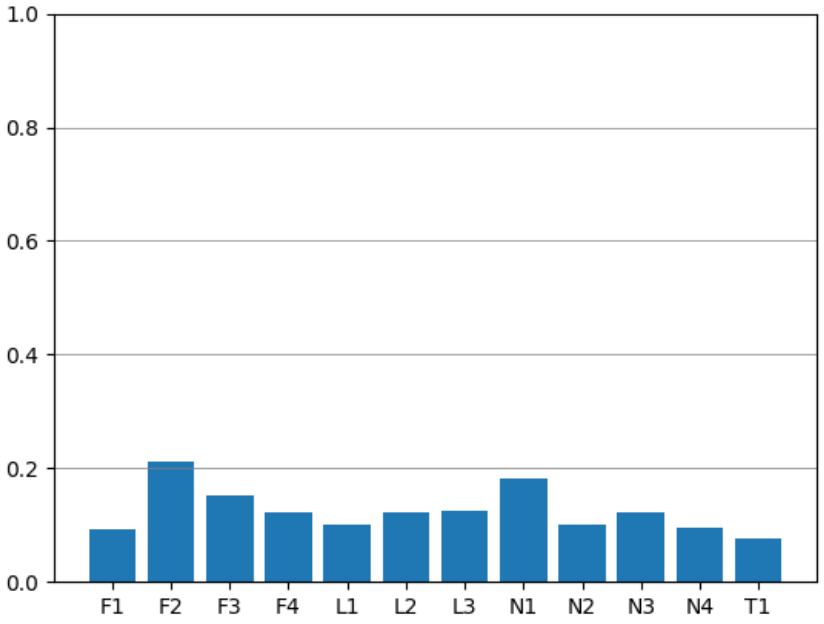

Figura A.17: Comparação entra métricas da base Monk

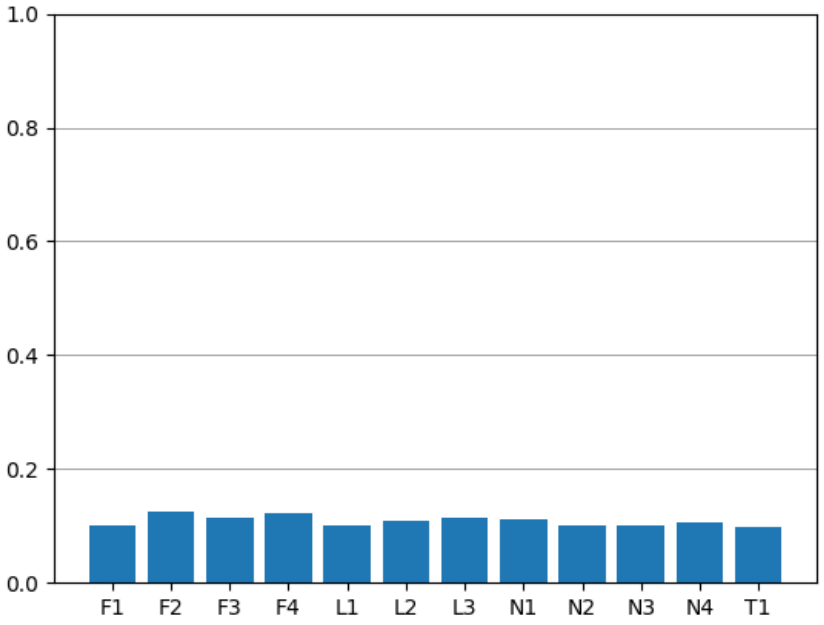

Figura A.18: Comparação entra métricas da base Phoneme

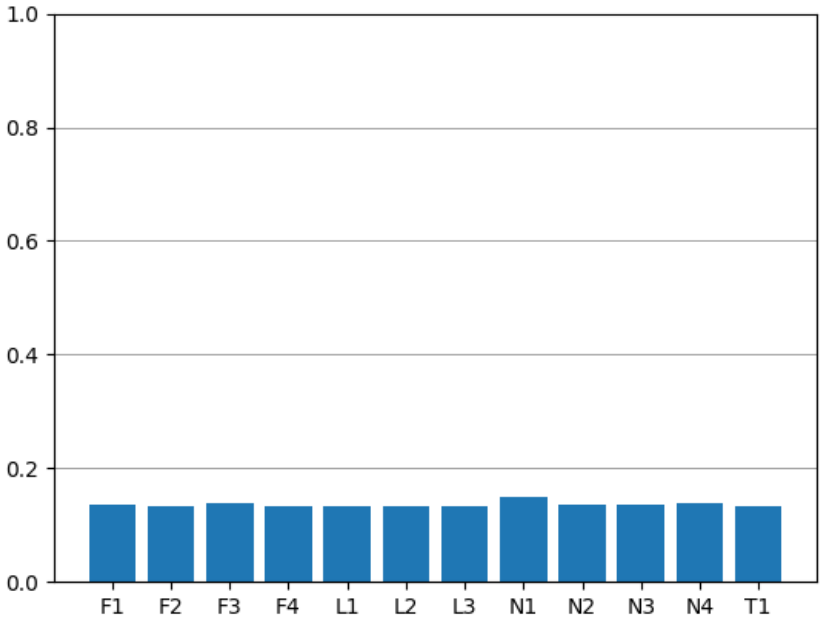

Figura A.19: Comparação entra métricas da base Segmentation

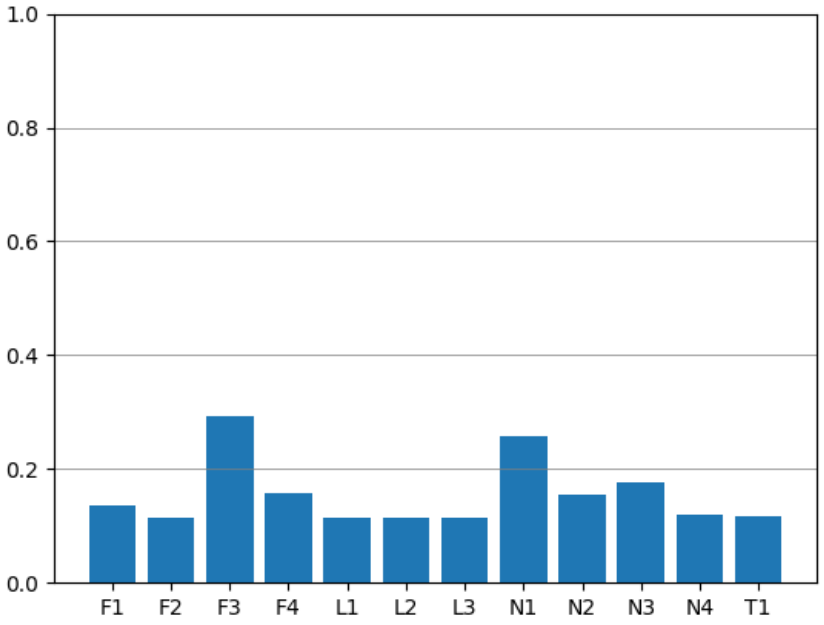

Figura A.20: Comparação entra métricas da base Sonar

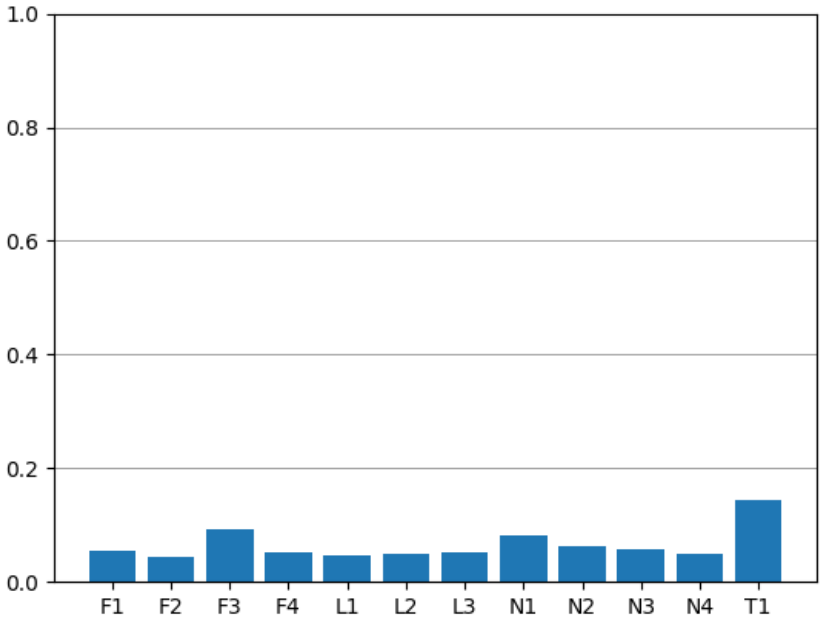

Figura A.21: Comparação entra métricas da base Thyroid

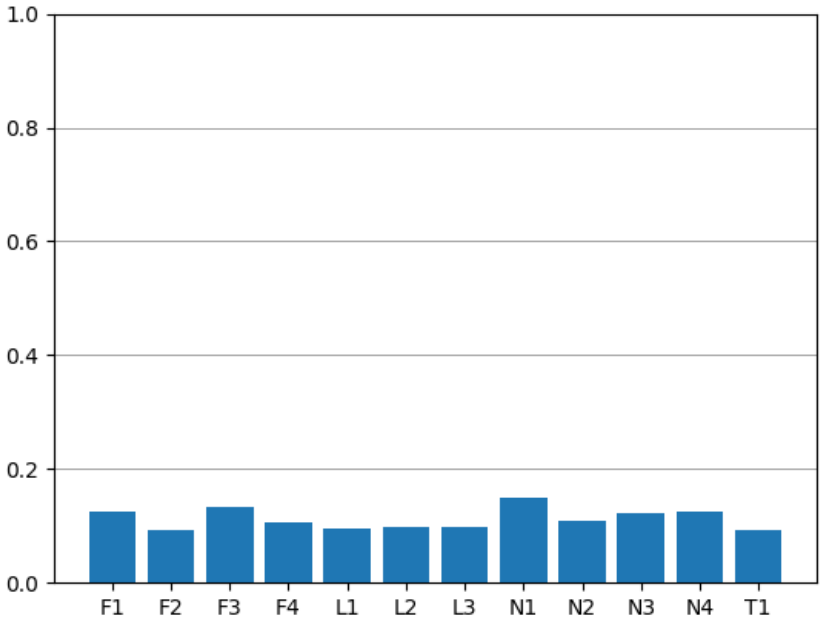

Figura A.22: Comparação entra métricas da base Vehicle

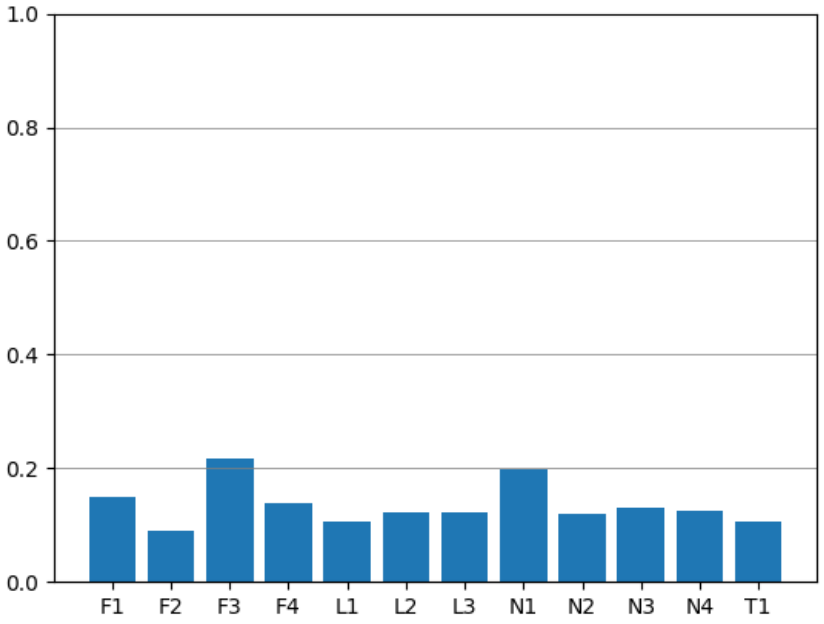

Figura A.23: Comparação entra métricas da base Vertebral

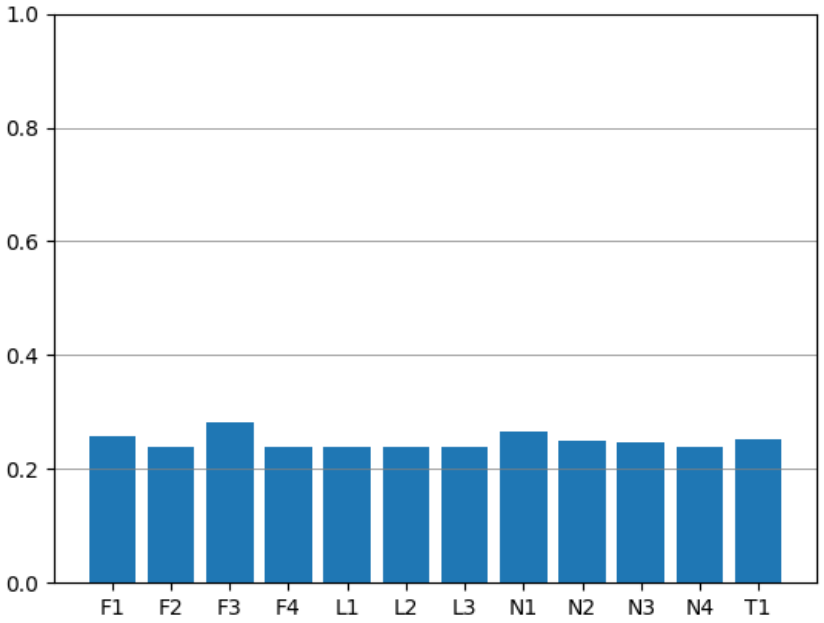

Figura A.24: Comparação entra métricas da base WBC

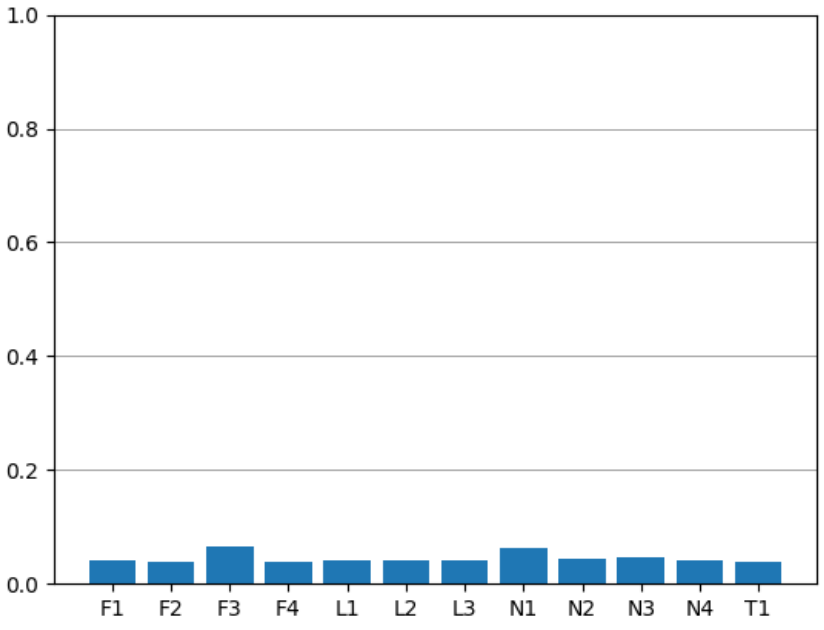

Figura A.25: Comparação entra métricas da base WDVG

Referências Bibliográficas

ALCALÁ-FDEZ, J. et al. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, v. 17, n. 2-3, p. 255–287, 2011.

BREIMAN, L. Bagging predictors. *Machine Learning*, v. 24, p. 123–140, 1996.

BRITTO JR., A. S.; SABOURIN, R.; OLIVEIRA, L. E. S. Dynamic selection of classifiers — a comprehensive review. *Pattern Recognition*, v. 47, p. 3665–3680, 2014.

BRUN, A. L. *Geração e Seleção de Classificadores com base na Complexidade do Problema*. Tese (Tese de Doutorado) — Pontifícia Universidade Católica do Paraná, Curitiba, 2017.

BRUN, A. L. et al. Contribution of data complexity features on dynamic classifier selection. In: *2016 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2016. p. 4396–4403.

BRUN, A. L. et al. A framework for dynamic classifier selection oriented by the classification problem difficulty. *Pattern Recognition*, v. 76, p. 175–190, 2018.

CASTRO, A. A. M. de; PRADO, P. P. L. do. Pattern recognition algorithms. *Rev. Ciênc. Exatas*, Taubaté, v. 5-8, p. 129–145, 2002.

COVER, T. M.; HART, P. E. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, v. 13, n. 1, p. 21–27, 1967.

DUA, D.; GRAFF, C. *UCI Machine Learning Repository*. 2017. Disponível em: <http://archive.ics.uci.edu/ml>. Acesso em: 20 jul 2021.

FREUND, Y.; SCHAPIRE, R. E. Experiments with a new boosting algorithm. *Proceedings of the 13th International Conference on Machine Learning*, Jan 1996.

HO, T. K. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 20, n. 8, p. 832–844, Aug 1998.

HO, T. K.; BASU, M. Measuring the complexity of classification problems. *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, v. 2, p. 43–47, 2000.

HO, T. K.; BASU, M. Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, n. 3, p. 289–300, Mar 2002.

KING, R. D.; FENG, C.; SUTHERLAND, A. Statlog: Comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence*, Taylor & Francis, v. 9, n. 3, p. 289–333, 1995. Disponível em: <https://doi.org/10.1080/08839519508945477>. Acesso em: 20 jul 2021.

KO, A. H.; SABOURIN, R.; BRITTO JR., A. S. From dynamic classifier selection to dynamic ensemble selection. *Pattern Recognition*, v. 41, p. 1718 – 1731, 2008.

KUNCHEVA, L. I. *Combining Pattern Classifiers*. 1. ed. New Jersey: JOHN WILEY & SONS, INC., 2004.

LANDEROS, A. I. *Data Complexity and Classifier Eselection*. Tese (Tese de Doutorado) — University of Alabama, Curitiba, 2008.

LEBOURGEOIS, F.; EMPTOZ, H. Pretopolgical approach for supervised learning. *IEEE Proceedings of ICPR '96*, p. 256–260, 1996.

LORENA, A. C. et al. How complex is your classification problem? a survey on measuring classification complexity. *ACM Computing Surveys (CSUR)*, v. 52, p. 1–34, 2019.

LU, Y. Knowledge integration in a multiple classifier system. *Applied Intelligenc*, v. 6, p. 75–86, 1996.

MACIÀ, N. et al. Clearner excellence biased by data set selection: A case for data characterisation and artificial data sets. *Pattern Recognition*, v. 46, n. 3, p. 1054–1066, 2013.

MONTEIRO, M. et al. Classifier pool generation based on a two-level diversity approach. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. [S.l.: s.n.], 2021. p. 2414–2421.

ORRIOLS-PUIG, A.; MACIÀ, N.; HO, T. K. *Documentation for the Data Complexity Library in C++*. Barcelona, 2010.

PALMIERE, S. E. *Arquiteturas e Topologias de Redes Neurais Artificiais*. 2016. Disponível em: <https://www.embarcados.com.br/redes-neurais-artificiais/>. Acesso em: 20 jul 2021.

PALMIERE, S. E. *Rede Perceptron de uma única camada*. 2016. Disponível em: <https://www.embarcados.com.br/rede-perceptron-de-uma-unica-camada/>. Acesso em: 20 jul 2021.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

PONTI JR., M. P. Combining classifiers: From the creation of ensembles to the decision fusion. *Conference on Graphics, Patterns, and Images Tutorials*, p. 1–10, 2011.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, v. 65, n. 6, p. 386–408, 1958.

SANTOS, M. J. dos. *Análise da relação entre tamanho e complexidade de um conjunto de dados*. Dissertação (Monografia de Graduação) — UNIOESTE – Universidade Estadual do Oeste do Paraná, Cascavel, Dec 2018.

SILVA JR., E. J. D. *Método de Classificação em Cascata de Dois Níveis: uma alternativa para a redução do custo de sistemas baseados em múltiplos classificadores*. Dissertação (Dissertação de Mestrado) — Pontifícia Universidade Católica do Paraná, Curitiba, Oct 2015.

SOTOCA, J.; SÁNCHEZ, J.; MOLLINEDA, R. A review of data complexity measures and their applicability to pattern classification problems. *Actas del III Taller Nacional de Mineria de Datos y Aprendizaje*, Jan 2005.

STANGE, R. L. *Adaptatividade em Aprendizagem de Máquina: Conceitos e Estudo de Caso*. Dissertação (Dissertação de Mestrado) — ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO, São Paulo, Dec 2011.

SÁNCHEZ, J. S.; MOLLINEDA, R. A.; SOTOCA, J. M. An analysis of how training data complexity affects the nearest neighbor classifiers. *Springer-Verlag London Limited*, London, v. 10, n. 3, p. 189–201, Sep 2007.

WITTEN, I. H.; FRANK, E. Data mining practical machine learning tools and techniques. In: ____. 2. ed. San Francisco: Elsevier, 2005. capítulo Input: Concepts, instances, and attributes, p. 41–60.