

Exercício n.1: Sistemas de Múltiplos Classificadores

Filipe Coelho de Lima Duarte

December 18, 2020

Repositório com o código: https://github.com/filipeclduarte/ensemble_learning

Conjuntos de dados:

1. **Breast Cancer Wisconsin (WDBC):** <https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29>

Quantidade de instâncias: 569

Quantidade de atributos: 32

Quantidade de classes: 2 (M=malignant, B=benign)

2. **banknote authentication (BANK):** <http://archive.ics.uci.edu/ml/datasets/banknote+authentication>

Quantidade de instâncias: 1372

Quantidade de atributos: 5

Quantidade de classes: 2

3. **Connectionist Bench (ION):** <https://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+%28Sonar%2C+Mines+vs.+Rocks%29>

Quantidade de instâncias: 208

Quantidade de atributos: 60

Quantidade de classes: 2 (R=rock, B=mine)

4. **Climate Model Simulation Crashes (CLIMA):** <https://archive.ics.uci.edu/ml/datasets/climate+model+simulation+crashes>

Quantidade de instâncias: 540

Quantidade de atributos: 18

Quantidade de classes: 2 (0=failure, 1=success)

5. **Fertility (FERT):** <https://archive.ics.uci.edu/ml/datasets/Fertility>

Quantidade de instâncias: 100

Quantidade de atributos: 10

Quantidade de classes: 2 (N=normal, O=altered)

Questão 1

Selecione cinco bases de dados públicas que contenham características diferentes e, para cada base, calcule o Oracle no conjunto de teste para:

- Bagging
- Adaboost
- Random Subspace (50%)
- Random Oracles

variando o número de classificadores-base 10, 20, ..., 100. Use o Perceptron como classificador-base e divida os fold usando o 5-fold cross-validation. Analise os resultados.

Resposta:

De uma forma geral, para todos os conjuntos de dados selecionados, o **Adaboost** demonstrou-se como o modelo cujo Oracle possuiu, em média, a maior Acurácia. Uma possível explicação seja o fato do Adaboost produzir modelos sequencialmente, o que garantiria uma melhoria de desempenho a cada classificador produzido por ele. O modelo **Bagging** ficou em segundo lugar, com resultados médios em torno de 98%, sendo que com uma presença de maior variação entre os Oracles quando comparado com o Adaboost. Ainda, observei que para o modelo Bagging um aumento da quantidade de classificadores (L) melhorou, em média, o desempenho (acurácia do Oracle) para os dados observados neste trabalho.

Em contrapartida, os modelos **Random Linear Oracles (RLO)** e **Random Subspace** apresentaram os menores valores de acurácia e as maiores variações (mensurada pelo desvio padrão). Dos resultados analisados, notou-se que o nível de desbalanceamento dos dados demonstrou correlação positiva com a variância do desempenho dos Oracles do modelo Random Linear Oracles, de tal maneira que quando maior o nível do desbalanceamento, maior foi o valor da variância da acurácia dos Oracles. A seguir apresento os resultados por meio de tabelas e gráficos de boxplot para cada conjunto de dados.

1.1 WDBC

Podemos visualizar o boxplot referente ao desempenho dos Oracles dos modelos analisados (Adaboost, Bagging, Random Subspace e Random Linear Oracle - RLO) para 10, 20, ..., 100 classificadores. Para o conjunto de dados WDBC notamos que há um destaque para o modelo Adaboost o qual apresenta-se como o melhor, auferindo acurácia de 100% para todos os casos exceto para $L = 10$ ou 20.

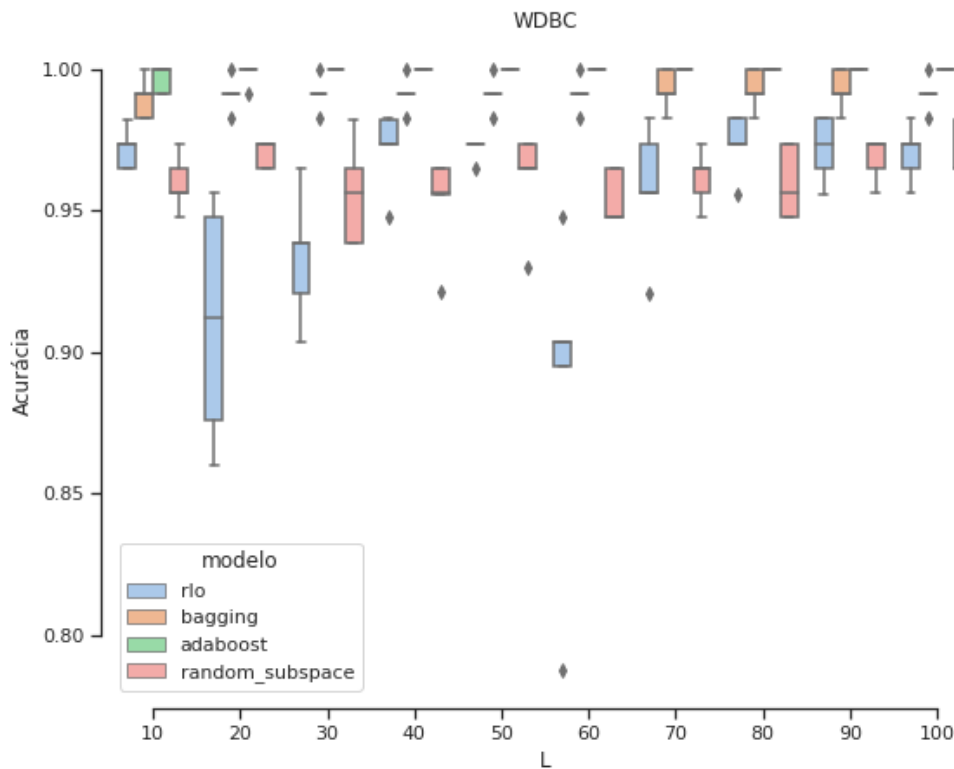
WDBC: Acurácia do Oracle

L	Modelo	Média	Desvio padrão
10	adaboost	0.996491	0.004805
	bagging	0.989474	0.007339
	random_subspace	0.959603	0.009920
	rlo	0.971899	0.007284
20	adaboost	0.998246	0.003923
	bagging	0.991228	0.006203
	random_subspace	0.970113	0.004891
	rlo	0.910309	0.042468
30	adaboost	1.000000	0.000000
	bagging	0.991228	0.006203
	random_subspace	0.956109	0.018553
	rlo	0.933194	0.022970
40	adaboost	1.000000	0.000000
	bagging	0.991228	0.006203
	random_subspace	0.952554	0.018172
	rlo	0.971899	0.014386
50	adaboost	1.000000	0.000000
	bagging	0.991228	0.006203
	random_subspace	0.961341	0.018175
	rlo	0.971868	0.004062
60	adaboost	1.000000	0.000000
	bagging	0.991228	0.006203
	random_subspace	0.954324	0.009525
	rlo	0.887347	0.059428
70	adaboost	1.000000	0.000000
	bagging	0.992982	0.007339
	random_subspace	0.961357	0.009930
	rlo	0.957755	0.023811
80	adaboost	1.000000	0.000000
	bagging	0.992982	0.007339
	random_subspace	0.959603	0.013242
	rlo	0.973607	0.010902
90	adaboost	1.000000	0.000000
	bagging	0.992982	0.007339
	random_subspace	0.968374	0.007807
	rlo	0.971852	0.011572
100	adaboost	1.000000	0.000000
	bagging	0.991228	0.006203
	random_subspace	0.968390	0.014641
	rlo	0.970113	0.010043

O Bagging ficou em segundo lugar com acurácia mnédia próxima de 99%, além disso houve uma maior variação dos resultados para L = 10, 70, 80 e 90, o que pode mostrar haver uma relação não linear cônica entre quantidade de

classificadores e performance do Oracle no conjunto de teste.

No que diz respeito aos modelos Random Linear Oracles e Random Subspace, notei que o primeiro superou em seis (6) casos dos dez (10) cenários avaliados. Além disso, o RLO demonstrou maior variação para poucos (20 e 30) classificadores, enquanto o Random Subspace apresentou maior variação para 30, 40 e 50. Dessa forma, a performance desses dois modelos foi semelhante, entretanto o Random Subspace demonstrou maior estabilidade ao longo dos cenários. Diante disso, para o conjunto de dados WDBC, percebi que um aumento na quantidade de Classificadores melhorou a performance do Oracle do RLO e do Random Subspace.



1.2 BANK

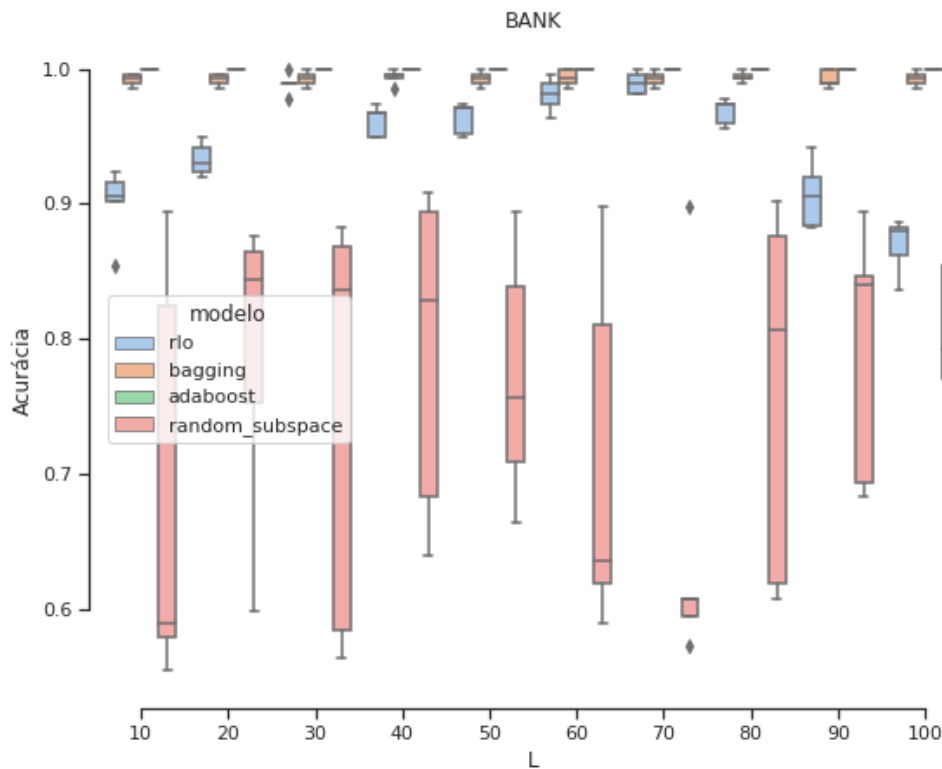
Para o conjunto de dados BANK, os Oracles dos modelos Adaboost alcançaram desempenho de 100% para todos os folds. Por outro lado, o modelo **Random Subspace** foi notadamente o pior, em média, cujos valores atingiram no máximo o valor de 90%. Ainda, a média da acurácia dos Oracles desse modelo ficou bem abaixo da acurácia média dos Oracles dos demais modelos. Ele demonstrou elevado valor de desvio padrão o qual pode ser visualizado na tabela quanto no gráfico boxplot (caixa vermelha).

BANK: Acurácia do Oracle			
L	Modelo	Média	Desvio padrão
10	adaboost	1.000000	0.000000
	bagging	0.991981	0.004756
	random_subspace	0.688199	0.158762
	rlo	0.900141	0.027162
20	adaboost	1.000000	0.000000
	bagging	0.991981	0.004756
	random_subspace	0.787156	0.116064
	rlo	0.932945	0.012253
30	adaboost	1.000000	0.000000
	bagging	0.992711	0.005766
	random_subspace	0.747153	0.159326
	rlo	0.989067	0.007742
40	adaboost	1.000000	0.000000
	bagging	0.994166	0.005538
	random_subspace	0.790819	0.123206
	rlo	0.961375	0.011679
50	adaboost	1.000000	0.000000
	bagging	0.992711	0.005766
	random_subspace	0.772653	0.093917
	rlo	0.963559	0.011791
60	adaboost	1.000000	0.000000
	bagging	0.993439	0.006522
	random_subspace	0.710068	0.136001
	rlo	0.981048	0.012742
70	adaboost	1.000000	0.000000
	bagging	0.992711	0.005766
	random_subspace	0.656048	0.135873
	rlo	0.989797	0.008309
80	adaboost	1.000000	0.000000
	bagging	0.994171	0.004151
	random_subspace	0.761879	0.140566
	rlo	0.968664	0.009825
90	adaboost	1.000000	0.000000
	bagging	0.992709	0.006823
	random_subspace	0.791435	0.096853
	rlo	0.906713	0.024907
100	adaboost	1.000000	0.000000
	bagging	0.992711	0.005766
	random_subspace	0.803092	0.083571
	rlo	0.869532	0.021193

Uma possível explicação para os Oracles do Random Subspace auferirem os piores resultados para BANK se deve ao fato dessa base de dados possuir uma pequena quantidade de atributos, apenas 5 variáveis. Isso faz com que,

para cada classificador gerado, sejam escolhidas aproximadamente 2,5 atributos (features), o que pode não ser suficiente para gerar condições de separação linear. Nesse sentido, a baixa dimensionalidade talvez não gere condições para que o modelo Perceptron consiga separar linearmente as classes.

No que diz respeito ao modelo RLO, notei que houve uma relação não linear côncava entre quantidade de classificadores de tal maneira que o melhor desempenho foi obtido entre 30, 60 e 70 classificadores e, em seguida, o aumento de L fez com que a performance começasse a piorar.



1.3 ION

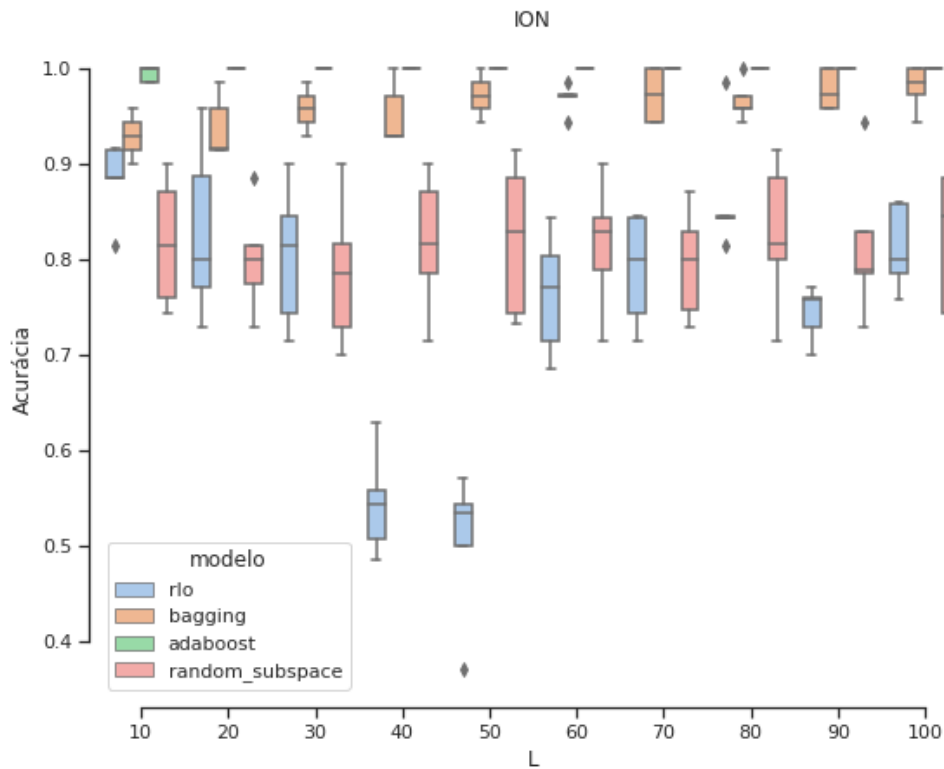
Assim como nos conjuntos de dados anteriores, verifiquei que, para a base de dados ION, o Oracle do Adaboost demonstrou superioridade comparado aos demais modelos, pois a sua acurácia média no conjunto de teste foi de 100%, exceto para $L = 10$. Acredita-se portanto que dez (10) classificadores não sejam suficientes para que o Adaboost gere diversidade e precisão suficientes de tal maneira que possibilite o Oracle alcançar o teto (100%) de desempenho.

Em relação ao Oracle do modelo Bagging, verifiquei que houve melhoria na acurácia quando do aumento da quantidade dos classificadores, ao passo que

o adaboost apresentou 100% de acurácia e 0,000 de desvio, exceto para $L = 10$, quando os valores foram de 0,994326 e 0,00777. Corroborar com os achados para os demais conjuntos de dados.

ION: Acurácia do Oracle			
L	Modelo	Média	Desvio padrão
10	adaboost	0.994326	0.007770
	bagging	0.928732	0.022717
	random_subspace	0.817827	0.068077
	rlo	0.883099	0.041143
20	adaboost	1.000000	0.000000
	bagging	0.937384	0.032682
	random_subspace	0.800644	0.057649
	rlo	0.828893	0.092282
30	adaboost	1.000000	0.000000
	bagging	0.957264	0.022589
	random_subspace	0.786237	0.078493
	rlo	0.803300	0.075487
40	adaboost	1.000000	0.000000
	bagging	0.951630	0.032716
	random_subspace	0.817666	0.073107
	rlo	0.544266	0.054975
50	adaboost	1.000000	0.000000
	bagging	0.971549	0.022494
	random_subspace	0.820765	0.082013
	rlo	0.504185	0.078456
60	adaboost	1.000000	0.000000
	bagging	0.968652	0.015669
	random_subspace	0.814889	0.068954
	rlo	0.763421	0.063990
70	adaboost	1.000000	0.000000
	bagging	0.971509	0.028572
	random_subspace	0.795010	0.058652
	rlo	0.789014	0.058900
80	adaboost	1.000000	0.000000
	bagging	0.965835	0.021607
	random_subspace	0.826237	0.078420
	rlo	0.866157	0.068035
90	adaboost	1.000000	0.000000
	bagging	0.977223	0.021640
	random_subspace	0.814889	0.079920
	rlo	0.743541	0.029036
100	adaboost	1.000000	0.000000
	bagging	0.980080	0.023869
	random_subspace	0.829014	0.084010
	rlo	0.811831	0.045015

Ainda em relação ao conjunto de dados ION, podemos ver resultados inferiores para os Oracles dos modelos RLO e Random Subspace, e um fato interessante do ponto de vista comparativo entre eles, uma vez que ambos demonstraram resultados próximos, exceto quando $L = 40$ e 50 . Para estes casos, obteve-se os Oracles do RLO obtiveram acurácia média em torno de 50-54%, denotando desempenho significativamente inferior os demais casos. Isso pode ter ocorrido em decorrência da alta dimensionalidade em relação à quantidade de instâncias deste conjunto de dados, tendo como consequência poucos padrões por atributos para serem utilizados como instâncias para o modelo base (i.e., Perceptron).



1.4 CLIMA

Verifiquei também que, para a base de dados CLIMA, os Oracles do Adaboost continuaram demonstrando desempenho superior comparado aos demais modelos, pois a sua acurácia média no conjunto de teste foi de 100%, exceto para $L = 10$.

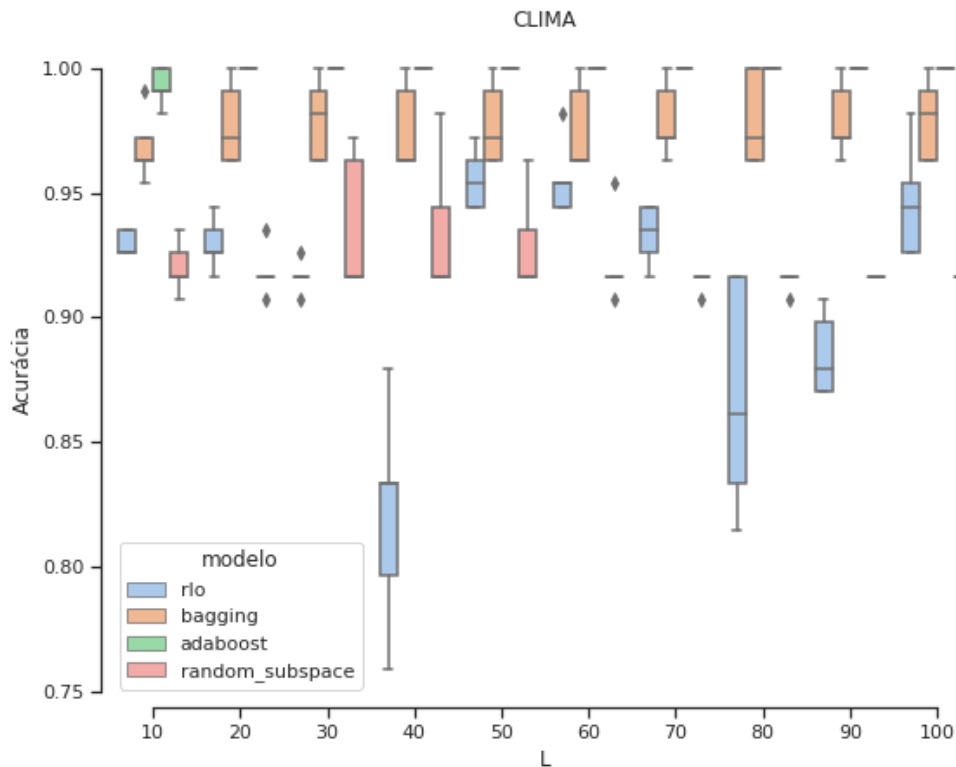
No que diz respeito ao Bagging, para os dados CLIMA, ele demonstrou o maior grau de variação de desempenho dos Oracles.

CLIMA: Acurácia do Oracle

L	Modelo	Média	Desvio padrão
10	adaboost	0.992593	0.007747
	bagging	0.968519	0.014042
	random_subspace	0.920370	0.010557
	rlo	0.929630	0.005072
20	adaboost	1.000000	0.000000
	bagging	0.977778	0.016820
	random_subspace	0.918519	0.010143
	rlo	0.929630	0.010557
30	adaboost	1.000000	0.000000
	bagging	0.979630	0.016563
	random_subspace	0.937037	0.028085
	rlo	0.916667	0.006547
40	adaboost	1.000000	0.000000
	bagging	0.975926	0.018050
	random_subspace	0.935185	0.028539
	rlo	0.820370	0.045172
50	adaboost	1.000000	0.000000
	bagging	0.977778	0.016820
	random_subspace	0.929630	0.020286
	rlo	0.955556	0.012073
60	adaboost	1.000000	0.000000
	bagging	0.975926	0.018050
	random_subspace	0.922222	0.018050
	rlo	0.955556	0.015215
70	adaboost	1.000000	0.000000
	bagging	0.979630	0.015215
	random_subspace	0.914815	0.004141
	rlo	0.933333	0.012073
80	adaboost	1.000000	0.000000
	bagging	0.979630	0.018976
	random_subspace	0.914815	0.004141
	rlo	0.868519	0.046940
90	adaboost	1.000000	0.000000
	bagging	0.979630	0.015215
	random_subspace	0.916667	0.000000
	rlo	0.885185	0.016820
100	adaboost	1.000000	0.000000
	bagging	0.979630	0.016563
	random_subspace	0.914815	0.004141
	rlo	0.946296	0.023055

Quanto aos modelos RLO e Random Subspace, notei que o primeiro demonstrou maior variação e menor média da acurácia dos Oracles. Vale destacar que para $L = 40$, os Oracles do RLO apresentaram o pior desempenho com média

(desvio) de 0,82037 (0,045172). Destaco que a base de dados CLIMA apresenta o maior grau de desbalanceamento dentre os conjuntos de dados selecionados. Dentre as 540 instâncias 494 são de classe 0, ao passo que 46 são da classe 1.



1.5 FERT

A base FERT demonstrou resultado diferente em relação ao Adaboost no sentido em que não houve variação para $L = 10$, contudo apresentou variação para $L = 20$ em que a acurácia média do Oracle foi de 0.99, ao passo que o desvio padrão foi de 0.022361.

Faço um ressalva quanto à característica desse conjunto de dados, o qual apresenta poucas instâncias ($N = 100$) e um elevado nível de desbalanceamento, pois a classe 0 contou com 88 observações, restando apenas 12 instâncias para a classe 1.

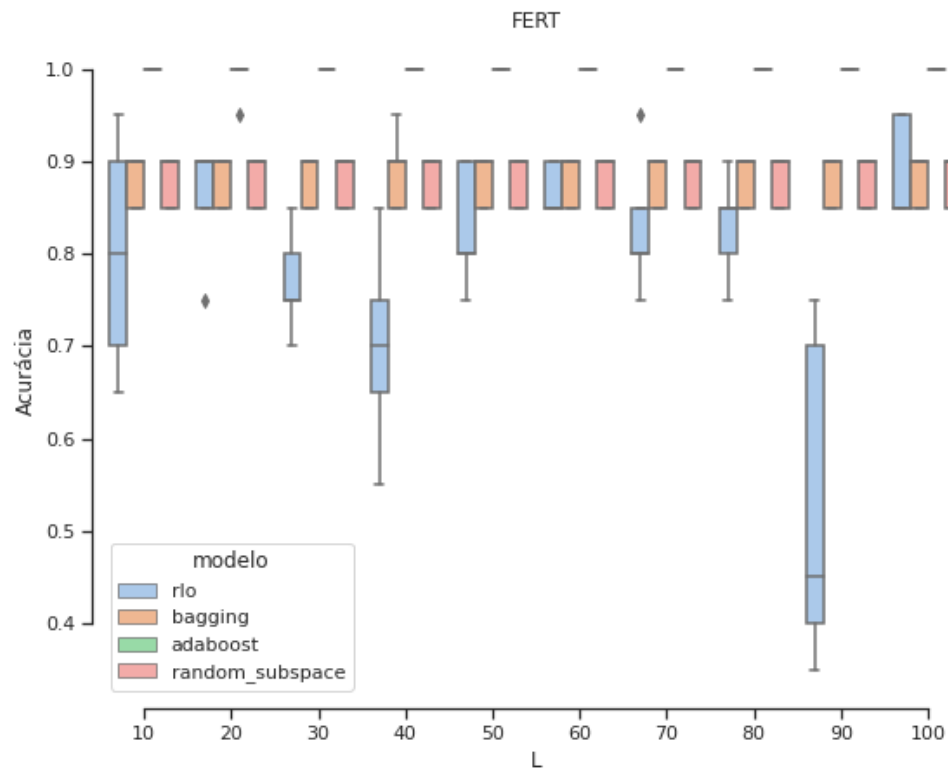
Sendo assim, essa característica pode ser importante para que houvesse pouco espaço para que o modelo Bagging produzisse resultados tão bons quanto o Adaboost, ou seja, quando da presença de poucos padrões, tendo em vista que o Bagging realiza bootstrap com as instâncias de treinamento.

FERT: Acurácia do Oracle

L	Modelo	Média	Desvio padrão
10	adaboost	1.00	0.000000
	bagging	0.88	0.027386
	random_subspace	0.88	0.027386
	rlo	0.80	0.127475
20	adaboost	0.99	0.022361
	bagging	0.88	0.027386
	random_subspace	0.88	0.027386
	rlo	0.86	0.065192
30	adaboost	1.00	0.000000
	bagging	0.88	0.027386
	random_subspace	0.88	0.027386
	rlo	0.77	0.057009
40	adaboost	1.00	0.000000
	bagging	0.89	0.041833
	random_subspace	0.88	0.027386
	rlo	0.70	0.111803
50	adaboost	1.00	0.000000
	bagging	0.88	0.027386
	random_subspace	0.88	0.027386
	rlo	0.83	0.067082
60	adaboost	1.00	0.000000
	bagging	0.88	0.027386
	random_subspace	0.88	0.027386
	rlo	0.87	0.027386
70	adaboost	1.00	0.000000
	bagging	0.88	0.027386
	random_subspace	0.88	0.027386
	rlo	0.83	0.075829
80	adaboost	1.00	0.000000
	bagging	0.88	0.027386
	random_subspace	0.88	0.027386
	rlo	0.83	0.057009
90	adaboost	1.00	0.000000
	bagging	0.88	0.027386
	random_subspace	0.88	0.027386
	rlo	0.53	0.182346
100	adaboost	1.00	0.000000
	bagging	0.88	0.027386
	random_subspace	0.88	0.027386
	rlo	0.89	0.054772

Além disso, notei uma semelhança de performance entre aquele e o Random Subspace, cujos resultados demonstraram média e desvio semelhante em todos os casos exceto quando $L = 40$.

No que diz respeito ao Random Linear Oracles, verifiquei um desempenho inferior e maior variação quanto às mudanças da quantidade de classificadores bases. Ainda, notei que houve uma maior variação na acurácia do Oracle do RLO quando a massa de dados possuía elevado nível de desbalanceamento.



Questão 2

Use as mesma bases de dados e os mesmos folds da questão anterior e, para cada base:

use o SGH para gerar o pool de classificadores no conjunto de treinamento;

calcule o Oracle do pool no conjunto de teste;

verifique quantas instâncias por classe foram incorretamente classificadas;

verifique quantos hiperplanos por classe foram gerados.

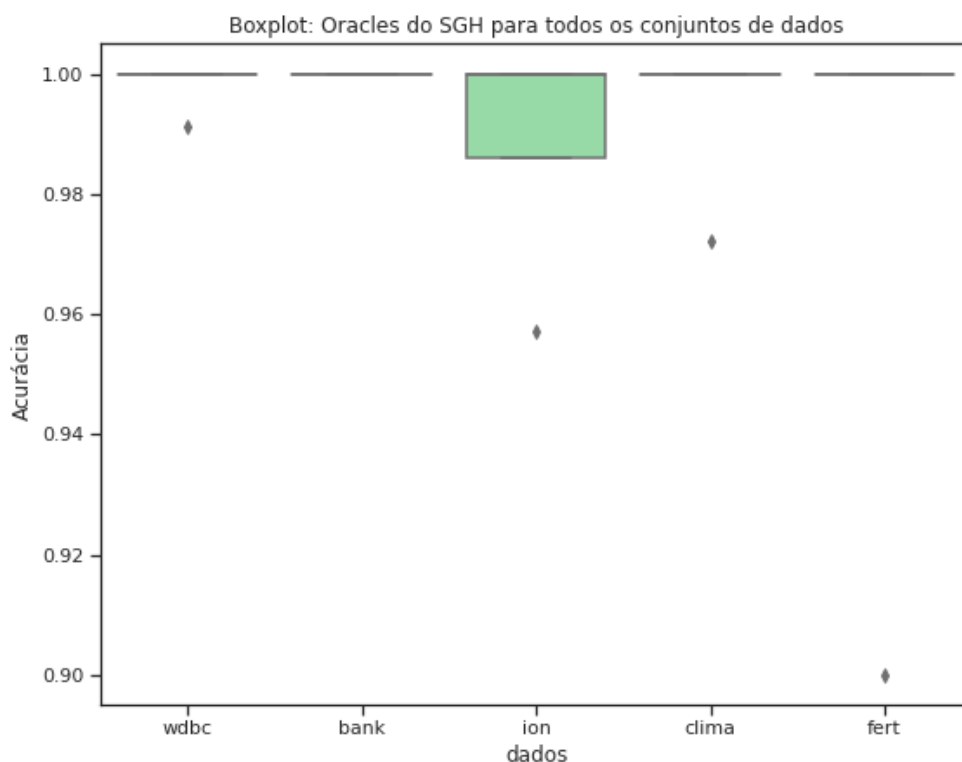
Analise os resultados coletados.

Resposta:

O Oracle do modelo SGH foi avaliado para todos os conjuntos de dados descritos no quesito n.1. Dessa maneira, utilizei os mesmos folds e apliquei um modelo SGH para cada um deles, cujos resultados estão detalhados a seguir em tabelas e gráficos:

Estatística	Oracles SGH				
	WDBC	BANK	ION	CLIMA	FERT
Média	0.998230	1.0	0.988612	0.994444	0.980000
Desvio padrão	0.003958	0.0	0.018619	0.012423	0.044721

Verificou-se um elevado nível de acurácia para todos os dados. Como destaque temos o BANK, cujos Oracles obtiveram acruácia de 100% para os dados de teste. Os dados com os piores resultados foram ION e FERT com médias (desvios) 0.988612 (0.018610) e 0.98 (0.044721), respectivamente.

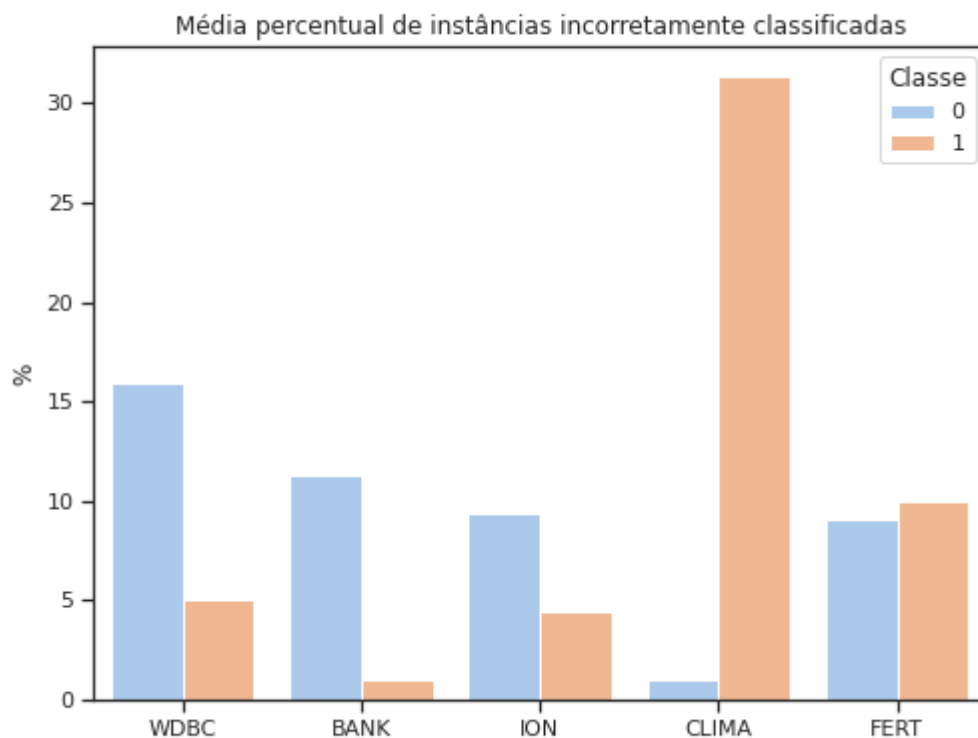


CLIMA é o conjunto de dados que apresenta a maior quantidade de dados desbalanceados, como se pode ver a partir da tabela 'Quantidade de Instâncias por fold'. Em segundo lugar temos o dataset **FERT**. Isso pode explicar o fato do modelo SGH ter apresentado um maior percentual (%) de instâncias de classe 1 incorretamente classificadas. Contudo, a severidade do erro foi consideravelmente maior para CLIMA, o qual demonstrou valor aproximadamente de 31%.

Quantidade de Instâncias por fold		
Dados	Classe	Qtd.
WDBC	0	357
	1	212
BANK	0	762
	1	610
ION	0	225
	1	126
CLIMA	0	494
	1	46
FERT	0	88
	1	12

Por outro lado, para os conjuntos de dados WDBC, BANK e ION, houve maior percentual de erros para as classes com maior quantidade de instâncias.

Qtd. de instâncias incorretamente classificadas pelo SGH								
Dados	Classe	Folds					Estatísticas	
		1	2	3	4	5	Média	Desvio pad.
WDBC	0	59	59	53	60	54	57.0	3.240
	1	15	14	9	6	9	10.6	3.782
BANK	0	77	74	99	97	84	86.2	11.389
	1	5	6	0	3	17	6.2	6.457
ION	0	24	16	21	22	22	21.0	3.000
	1	0	27	0	1	0	5.6	11.971
CLIMA	0	10	6	0	3	7	5.2	3.834
	1	12	13	17	13	17	14.4	2.408
FERT	0	4	13	18	5	0	8.0	7.314
	1	2	0	1	0	3	1.2	1.304



Diante da tabela "Qtd. de Hiperplanos por fold/dados" verifiquei que para todos os casos, exceto CLIMA-fold3 e FERT-fold5, foram gerados apenas 3 hiperplanos pelo modelo SGH. Como todos os conjuntos de dados diziam respeito à problemas de classificação binária, e como foram necessários pelo menos 2 hiperplanos, podemos concluir que nem todos os problemas (padrões existentes) eram linearmente separáveis, pois caso o fossem, a rigor bastaria um hiperplano para tal tarefa.

Qtd. de Hiperplanos por fold/dados					
Fold	wdbc	bank	ion	clima	fert
1	3	3	3	3	3
2	3	3	3	3	3
3	3	3	3	2	3
4	3	3	3	3	3
5	3	3	3	3	2

Questão 3

Quais são as diferenças e semelhanças entre o SGH e o Random Oracles?

Resposta:

O modelo Random Linear Oracle (RLO) é um sistema de múltiplos classificadores que segue a ideia de gerar modelos (classificadores base) simples e especializados, a partir da geração de um hiperplano que separa as instâncias de treinamento, de tal maneira que seja possível gerar diversidade entre os classificadores base. O particionamento é realizado por um hiperplano o qual fará o processo de separar as instâncias em duas partes, onde para cada uma delas treina-se um perceptron (classificador base).

O modelo Self Generating Hyperplanes (SGH) é um sistema de múltiplos classificadores determinístico cuja intenção está no desenvolvimento de um modelo com 100% de acurácia no conjunto de treinamento.

No que diz respeito às semelhanças entre esses modelos, é possível elencar o fato de ambos buscarem a geração de classificadores base simples e especializados e por gerarem hiperplanos.

Por outro lado, quanto às diferenças entre os modelos, podemos destacar o fato do modelo SGH ser determinístico, enquanto o RLO têm a característica de aleatoriedade no processo de geração dos Oracles. Em segundo lugar, o SGH se propõe a construir um ensemble com taxa de acerto de 100%, ao passo que isso não é o objetivo do RLO. Além disso, o classificador base do RLO é treinado, diferentemente do SGH o qual evita o processo de treinamento do classificador base. No SGH, o classificador base é um hiperplano calculado a partir dos dados de treinamento.