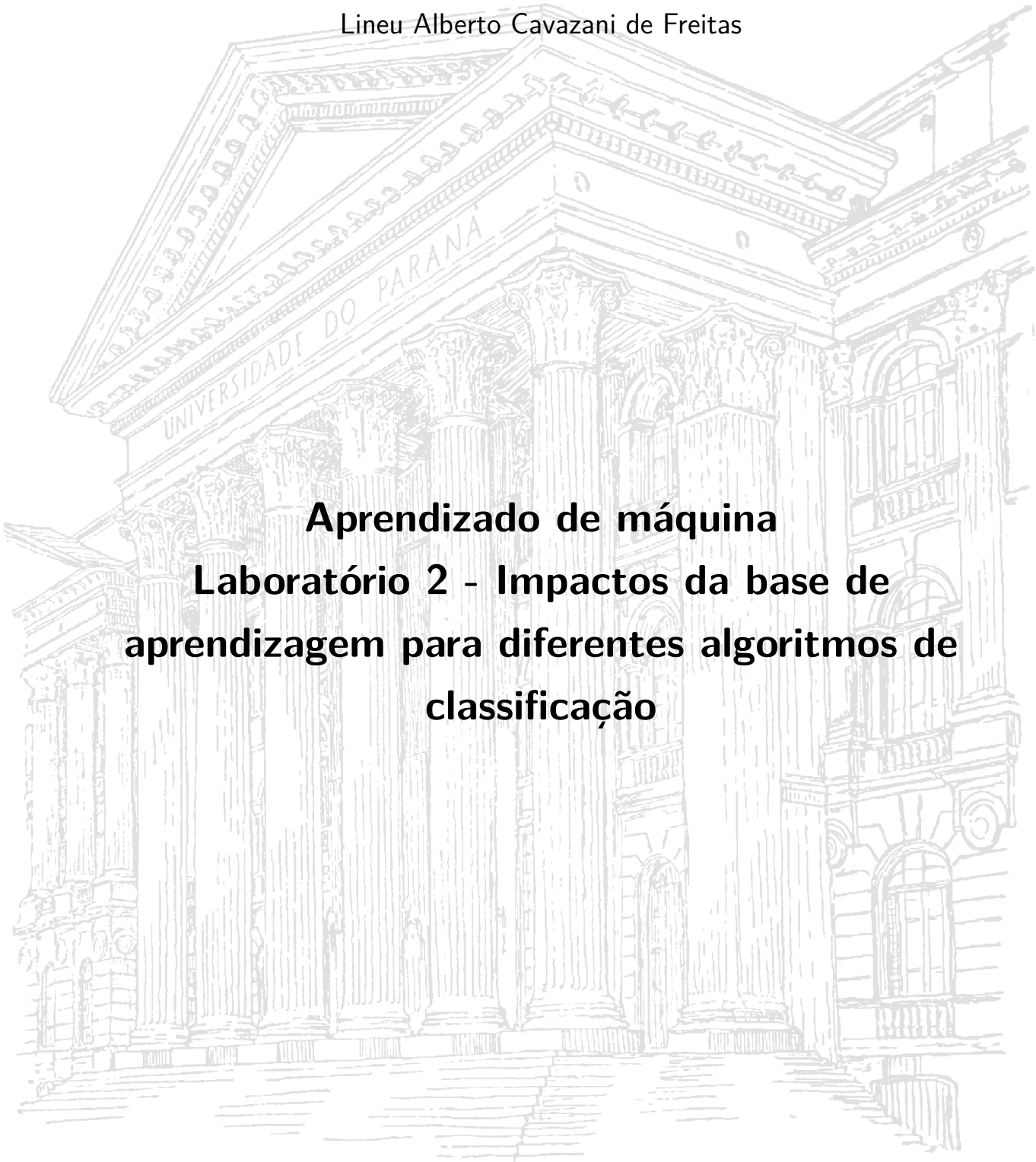
Universidade Federal do Paraná

Lineu Alberto Cavazani de Freitas



Aprendizado de máquina
Laboratório 2 - Impactos da base de aprendizagem para diferentes algoritmos de classificação

Curitiba

2021

Lineu Alberto Cavazani de Freitas

Aprendizado de máquina
Laboratório 2 - Impactos da base de aprendizagem para diferentes algoritmos de classificação

Relatório apresentado à disciplina Aprendizado de Máquina, ministrada pelo professor Luiz Eduardo Soares de Oliveira, no Programa de Pós Graduação em Informática da Universidade Federal do Paraná.

Universidade Federal do Paraná

Curitiba
2021

Sumário

1 Introdução

Aprendizado de máquina consiste em programar computadores de forma que eles aprendam a partir de dados. No cenário em que temos dados rotulados e uma variável alvo definida por categorias, recomenda-se o uso de técnicas de aprendizado supervisionado para fins de classificação. Dentre os diversos classificadores bem definidos e implementados, podemos citar o kNN, o Naive Bayes, a LDA, a regressão logística e o Perceptron.

A ideia geral do kNN consiste em, para uma unidade, encontrar os k mais próximos (similares) a ele e atribuir a classe mais frequente. Como existe a necessidade de obter a distância de um ponto para os outros, há um alto custo computacional. Outra característica deste classificador é que não requer treinamento, apenas teste, pois necessita apenas de distâncias. Contudo, a fase de teste demanda tempo considerável.

O Naive Bayes é um classificador baseado em probabilidade, mais especificamente, no pensamento bayesiano. Tem como base o teorema de Bayes, que consiste em alterar as probabilidades a priori (vindas dos vetores de característica) em estimativas de probabilidade a posteriori. A ideia do teorema é a revisão de crenças conforme surgem novas evidências e, na prática, atualiza-se a probabilidade a posteriori através da priori e da verossimilhança. O algoritmo, por sua vez, é chamado de ingênuo pois faz uma forte suposição: de que os atributos são independentes, contudo costuma apresentar bons resultados. O treinamento do Naive Bayes consiste na obtenção das probabilidades condicionais. Isto é, probabilidade da característica dado o desfecho. No caso de variáveis discretas, as probabilidades são beaseadas em frequências. Para contínuas as probabilidades comumente utilizadas são discretizar a variável em categorias ou usar uma função densidade de probabilidade (em geral, usa-se distribuição normal).

A LDA é um classificador baseado em transformações que tentam maximizar a distância entre classes e minimizar a distância intra classe. Na prática, o algoritmo busca encontrar a melhor projeção no espaço que discrimine as categorias. Trata-se de um classificador bastante simples e rápido, muito conhecido em contextos de redução de dimensionalidade. Além disso, é um modelo que apresenta certos pressupostos que, quando atendidos, geram resultados similares ao Naive Bayes, contudo, sem os pressupostos, ainda assim os resultados costumam ser satisfatórios.

A regressão logística é um classificador originalmente binário que apresenta uma característica bastante interessante: para fins de predição, fornece, além da classe, a probabilidade associada. Ou seja, a fronteira de decisão é baseada em um limiar de probabilidade. Além disso, apesar de se tratar de um classificador binário, diversas im-

plementações adaptam este algoritmo para lidar com múltiplas classes. Trata-se de um método bastante atrativo principalmente porque costuma apresentar bons resultados em problemas desbalanceados.

O Perceptron é um classificador linear baseado numa rede neural de um único neurônio que funciona bem para problemas linearmente separáveis. Seu funcionamento se dá a partir de um vetor de entrada, que é associado a pesos. Esta informação é sumarizada, passada por uma função de ativação e o resultado final é a resposta. Os pesos são obtidos na fase de aprendizagem e trata-se de um algoritmo "online", isto é, a cada exemplo visto, os pesos são atualizados, o que gera um ajuste mais fino. Contudo, como a cada exemplo o modelo é revisto, os dados devem estar bem embaralhados para evitar o chamado *Cathastrophic Forgetting* (convergir para uma única classe). Outra característica interessante do Perceptron é que, se o problema for linearmente separável, o algoritmo vai convergir e encontrar uma fronteira.

O objetivo deste relatório é apresentar os resultados da comparação de desempenho dos classificadores mencionados em função da disponibilidade da base de treinamento.

2 Descrição da atividade

O objetivo da tarefa é, considerando os classifcadores kNN, Naive Bayes, LDA, regressão logística e Perceptron, avaliar qual deles necessita de menos exemplos para aprender, qual tamanho de base de treinamento é suficiente para cada classificador, qual deles apresenta um melhor desempenho com poucos ou muitos dados, qual deles se mostra o mais rápido e ainda avaliar quais classificadores são complementares no problema sugerido.

Para o trabalho foram disponibilizados dois conjuntos de dados: um de treino e um de teste. Ambos os conjuntos referentes a um problema de classificação com 10 classes balanceadas e 132 características. A base de treino continha 20 mil exemplos e a de teste continha 58646.

Foi realizado um experimento no qual, para cada algoritmo, foram treinados 20 modelos aumentando o tamanho da base de treino de mil em mil, isto é, a primeira versão utilizava apenas 1000 exemplos para treinar, a segunda utilizava 2000 e assim por diante, até o último teste que utilizava todas as 20 mil observações no treinamento. Cada um destes modelos foi devidamente avaliado na base de teste, mantida fixa com todas as 58646 observações. Com isso, os resultados contém a análise de 20 modelos e 5 algoritmos, totalizando 100 cenários. Para cada cenário a métrica utilizada para avaliação foi a acurácia, considerando que trata-se de um problema balanceado.

Através da análise gráfica do comportamento da acurácia para cada classificador associado a cada tamanho de base de treinamento, buscou-se responder as perguntas inicialmente propostas. Após esta análise gráfica, explorou-se a matriz de confusão dos modelos que fizeram uso de toda a base de treinamento.

Os modelos foram ajustados utilizando a biblioteca *scikit-learn*, disponível para linguagem Python. Para execução do trabalho utilizou-se a plataforma Google Colaboratory (ou "Colab") que permite escrever código Python diretamente no navegador. Já a análise dos resultados foram realizadas no software R.

Quanto as parametrizações utilizadas para cada classificador no *scikit-learn*: para os classificadores Naive Bayes, LDA e Perceptron foram utilizados os parâmetros default das funções *GaussianNB()*, *LinearDiscriminantAnalysis()* e *Perceptron()*, respectivamente. Para o kNN, através da função *KNeighborsClassifier()* foram utilizados 5 vizinhos, função peso uniforme, algoritmo kd_tree, leaf_size igual a 30, e distância de Minkowski com p igual a 2 (distância Euclideana). Para regressão logística, foi utilizada a função *LogisticRegression()* utilizando algoritmo sag para otimização, indicado para problemas com conjuntos de dados grandes e número de iterações máximo igual a 500.

3 Resultados obtidos

As representações gráficas dos resultados dos experimentos estão representadas nas Figuras 1 e 2. Em ambas, no eixo horizontal está representado o número de exemplos utilizados da base de treinamento. No eixo vertical é mostrada a acurácia. As cores representam cada classificador. Na Figura 1 é possível comparar melhor o desempenho de cada classificador de acordo com o tamanho da base de treino. A Figura 2, por sua vez, fornece uma visão mais clara de quais algoritmos apresentam comportamento mais estável e também daqueles mais instáveis.

Os resultados mostram que, considerando o cenário com o menor número de exemplos tomados para treinamento, a acurácia mais alta foi observada para LDA e kNN, fornecendo indicativo de que, em cenários com poucos dados disponíveis para treinamento, estas são boas escolhas. A pior acurácia no cenário com poucos dados foi observada para o Naive Bayes. Contudo, apesar de um resultado abaixo dos demais, notou-se que o Naive Bayes apresenta um considerável salto de acurácia nos três primeiros pontos (mil, 2 mil e 3 mil exemplos).

A análise mostra ainda uma estabilidade na acurácia em todos os algoritmos a partir de 9 mil exemplos fornecidos para treino, isto é, o tamanho da base de treinamento deixa de ser relevante a partir de 9 mil. Avaliando a Figura 1, nota-se que, de 9 mil em diante, há um desempenho superior do kNN, seguido pela LDA, regressão logística e Naive Bayes. Estes quatro classificadores apresentam um padrão consideravelmente estável conforme aumenta-se o tamanho do treino. O Perceptron apresentou resultados satisfatórios em termos de acurácia, porém, comparado aos demais, apresentou grande oscilação.

No cenário considerando toda a base de aprendizagem, Perceptron e kNN aparecem praticamente empatados. O Perceptron, apesar da já mencionada oscilação, parece apresentar uma tendência crescente de acurácia conforme aumenta-se o treino. Já o kNN apresenta aparente estabilidade desde muito cedo. Tais resultados são indicativos de que talvez o Perceptron poderia se mostrar superior aos demais com mais exemplos para treinamento.

A Figura 2 fornece uma visão mais geral da velocidade com que os classificadores se estabilizam. É possível notar que o Naive Bayes não chega em nenhum cenário a uma acurácia superior a 0.9, contudo existe um salto considerável na acurácia para poucos exemplos. O kNN apresentou boa escalabilidade de acurácia e uma grande estabilidade após determinado número de exemplos no treino. O LDA apresentou comportamento bastante similar ao kNN. A regressão logística apresentou uma curva mais lenta até

a estabilização da acurácia. Por fim, na Figura 2 fica ainda mais evidente as oscilações presentes no Perceptron.

Considerando os classificadores que usaram toda a base de treinamento, buscou-se avaliar quais eram complementares com o objetivo de conjecturar quais deles poderiam ser combinados para se obter um melhor resultado. Tal análise foi feita baseada na avaliação das matrizes de confusão para cada modelo, resultados apresentados nas Tabelas 1 e 2.

Os resultados mostram que as classificações mais precisas na base de teste para as classes 0, 1, 3, 5 e 7 foram feitas pelo Perceptron. Já as classes 2, 6 e 8 foram melhores classificadas pelo kNN. A categoria 4 teve maior número de acertos quando utilizou-se regressão logística. Já para a categoria 9, LDA apresentou melhores resultados. Com isso, há indício de que os classificadores complementares são o kNN e o Perceptron.

kNN	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	**5472**	3	1	15	6	2	26	2	32	1
1	0	**6105**	175	119	56	6	35	66	34	59
2	12	11	**5607**	165	3	1	16	51	20	2
3	4	1	25	**5646**	2	51	1	53	20	16
4	12	11	13	3	**5305**	9	132	24	11	202
5	9	3	9	489	4	**4842**	41	16	83	43
6	31	10	4	2	3	44	**5724**	0	40	0
7	1	25	41	119	54	1	0	**5773**	7	76
8	36	24	42	114	32	38	50	27	**5165**	167
9	16	9	17	107	78	9	9	131	34	**5403**

Naive Bayes	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	**5220**	1	11	32	2	1	41	0	251	1
1	1	**5184**	583	238	86	22	85	340	80	36
2	9	24	**5289**	447	4	1	8	52	53	1
3	2	1	212	**5390**	1	33	0	127	31	22
4	14	2	44	12	**5273**	0	32	44	90	211
5	9	6	29	103	31	**4958**	46	2	169	186
6	78	7	89	8	15	90	**5286**	0	285	0
7	1	47	175	426	21	1	1	**5323**	60	42
8	175	5	53	182	23	7	38	13	**5112**	87
9	25	5	62	151	221	4	0	55	184	**5106**

Tabela 1 – Matrizes de confusão kNN e Naive Bayes.

LDA	0	1	2	3	4	5	6	7	8	9
0	**5358**	10	11	15	19	0	47	17	80	3
1	0	**6027**	222	85	9	22	38	199	31	22
2	22	41	**5605**	12	1	0	4	175	27	1
3	1	12	29	**5470**	1	19	1	247	23	16
4	20	71	42	0	**5208**	0	86	5	29	261
5	9	11	6	314	4	**5015**	50	24	67	39
6	77	49	37	15	56	36	**5460**	0	125	3
7	0	58	47	6	58	1	0	**5882**	22	23
8	80	59	38	5	51	29	54	57	**4961**	361
9	34	31	9	91	69	7	16	98	29	**5429**

Reg. Log.	0	1	2	3	4	5	6	7	8	9
0	**5381**	5	16	12	15	4	69	6	51	1
1	1	**5595**	116	269	200	74	179	74	78	69
2	22	18	**5585**	89	12	1	33	82	45	1
3	4	3	37	**5597**	16	39	1	74	20	28
4	35	8	30	1	**5315**	2	104	41	9	177
5	6	12	23	497	78	**4728**	50	22	73	50
6	87	26	0	1	20	96	**5517**	0	111	0
7	0	41	40	121	165	2	0	**5600**	17	111
8	83	43	47	59	85	46	53	58	**5000**	221
9	55	22	8	143	251	0	4	150	19	**5161**

Perceptron	0	1	2	3	4	5	6	7	8	9
0	**5532**	1	0	6	0	1	18	1	1	0
1	14	**6114**	46	217	14	176	27	43	2	2
2	88	32	**5548**	137	2	0	16	62	3	0
3	5	3	12	**5698**	0	60	1	28	2	10
4	116	13	46	17	**5172**	7	108	39	5	199
5	21	5	4	129	3	**5318**	40	1	6	12
6	129	8	5	4	5	57	**5648**	0	2	0
7	2	42	51	157	31	4	0	**5796**	1	13
8	329	39	45	457	35	225	185	20	**4211**	149
9	89	36	26	115	106	25	3	83	3	**5327**

Tabela 2 – Matrizes de confusão LDA, regressão logística e Perceptron.

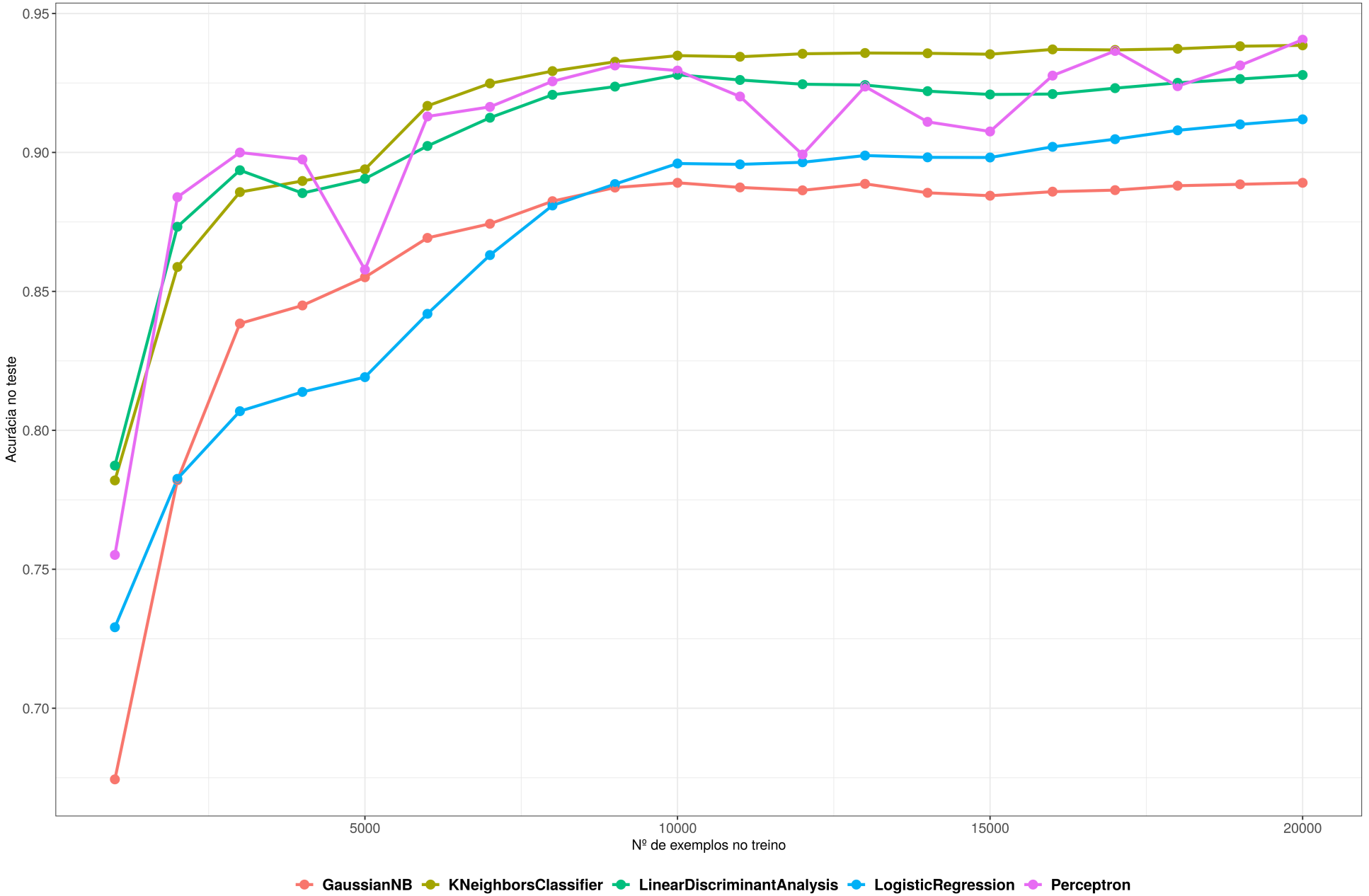


Figura 1 – Acurácia na base de teste em função do número de exemplos usado no treino.

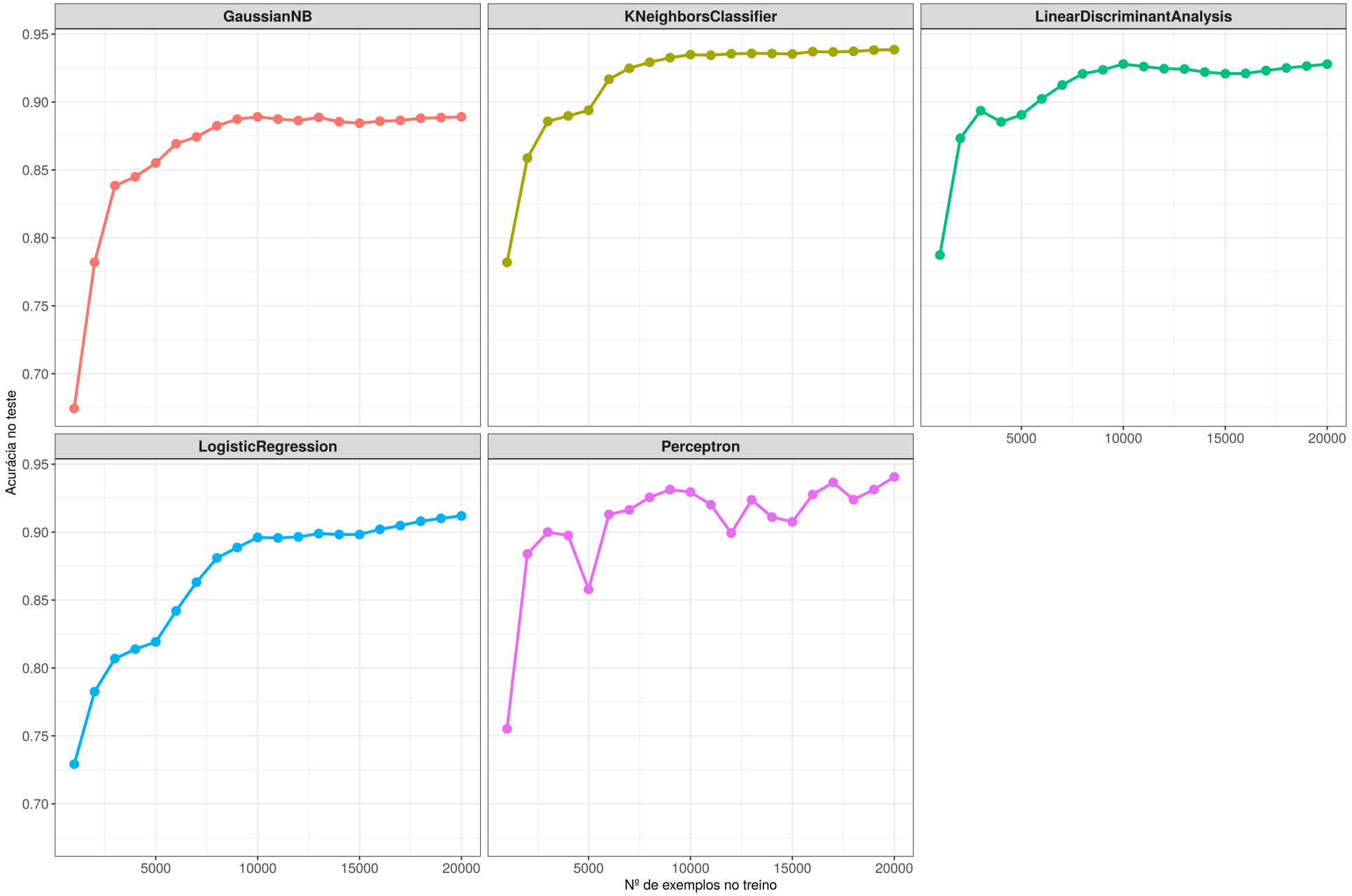


Figura 2 – Acurácia na base de teste em função do número de exemplos usado no treino (painel).

4 Considerações finais

Os resultados mostraram que, neste problema, considerando poucos exemplos, a acurácia mais alta foi observada para LDA e kNN. Ou seja, este estudo sugere que com poucos dados disponíveis para treinamento estas escolhas são atrativas. O estudo sugere ainda que, para este problema, um lote de exemplos superior a 9 mil não traz grandes benefícios no treino dos algoritmos. Quanto a velocidade de classificação das 58646 unidades para teste, a única que se mostrou demorada foi o kNN, tomando aproximadamente 3 minutos na plataforma Google Colab. Os demais classificadores se mostraram bastante rápidos.

Quanto à acurácia dos classificadores testados verificou-se um desempenho superior do kNN, seguido pela LDA, regressão logística e Naive Bayes. O kNN apresentou boa escalabilidade de acurácia e uma grande estabilidade. Como esperado, o LDA apresentou comportamento bastante similar ao kNN, porém levemente inferior. Comparado aos demais, a regressão logística se mostrou sensível ao tamanho da base de treino, necessitando de uma base maior para atingir acurácia à altura dos demais. Notou-se também que, para bases pequenas de treinamento, regressão logística foi inferior ao Naive Bayes; contudo, para bases maiores, este padrão se inverte. O Perceptron apresentou grandes oscilações conforme alterava-se o tamanho da base, mas no cenário com toda a base de treinamento, se mostrou como o melhor classificador para o problema.

Considerando toda a base de aprendizagem, Perceptron e kNN apresentam acurácia bastante similar. Notou-se que o Perceptron oscilou consideravelmente, mas parece apresentar uma acurácia crescente e estabilizando, sugerindo que é um bom candidato para bases de treinamento mais volumosas. O kNN, por sua vez, se mostrou mais estável. Deste modo, uma escolha segura seria o kNN, uma escolha mais arriscada e que necessitaria de mais dados para treino seria o Perceptron. Além disso, estes classificadores se mostraram complementares, dando indídio de que uma combinação entre eles pode gerar um classificador mais poderoso para o problema em questão.

Por fim, vale ressaltar que esta análise foi meramente exploratória e para um estudo mais consistente o ideal seria trabalhar com replicação, isto é, em vez de selecionar lotes de mil da base fornecida, selecionar mil linhas ao acaso, obter os modelos, acurácia e repetir este procedimento algumas vezes para cada modelo. Deste modo seria possível obter estimativas pontuais e intervalares da acurácia para cada cenário, fornecendo uma ideia mais realista de quais deles são mais estáveis.