

Relatório Laboratório 2 – Aprendizado De Máquina (CI1171)

Objetivo:

A partir de conjuntos de dados fornecidos pelo professor da disciplina, um para treino contendo 20000 amostras balanceadas e outro com 58646 amostras para teste. O experimento prático consistiu em analisar o impacto da base de aprendizagem na classificação de 5 algoritmos diferentes. Abordando acurácia, tempo de classificação, número de amostras necessárias para um bom resultado e confusões.

Determinação dos algoritmos:

Os algoritmos de classificação utilizados no experimento foram:

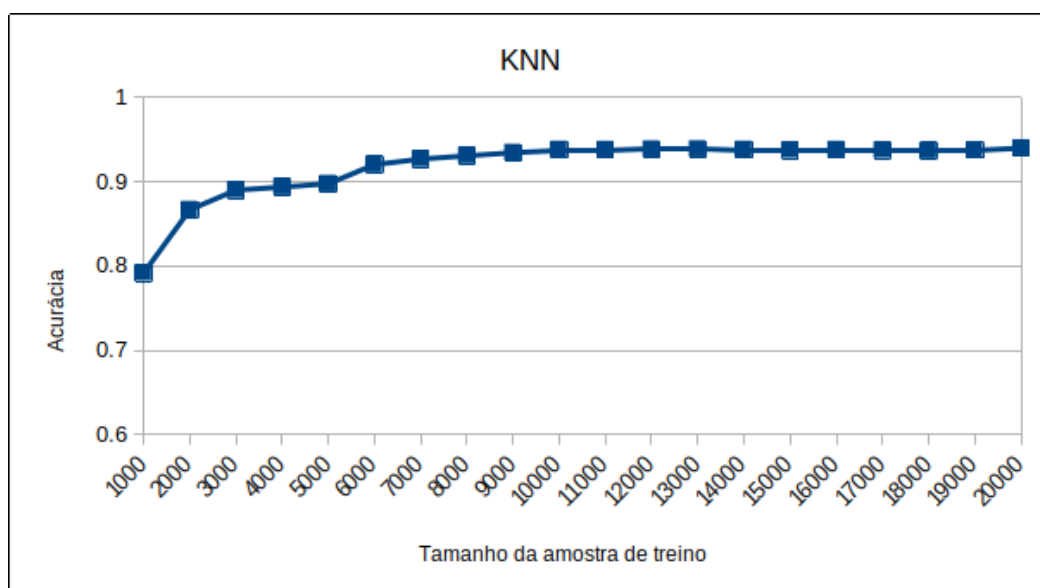
- *K-nearest Neighbors* (KNN);
- *Naive Bayes*;
- *Linear Discriminant Analysis* (LDA);
- *Logistic Regression* (LR);
- *Perceptron*.

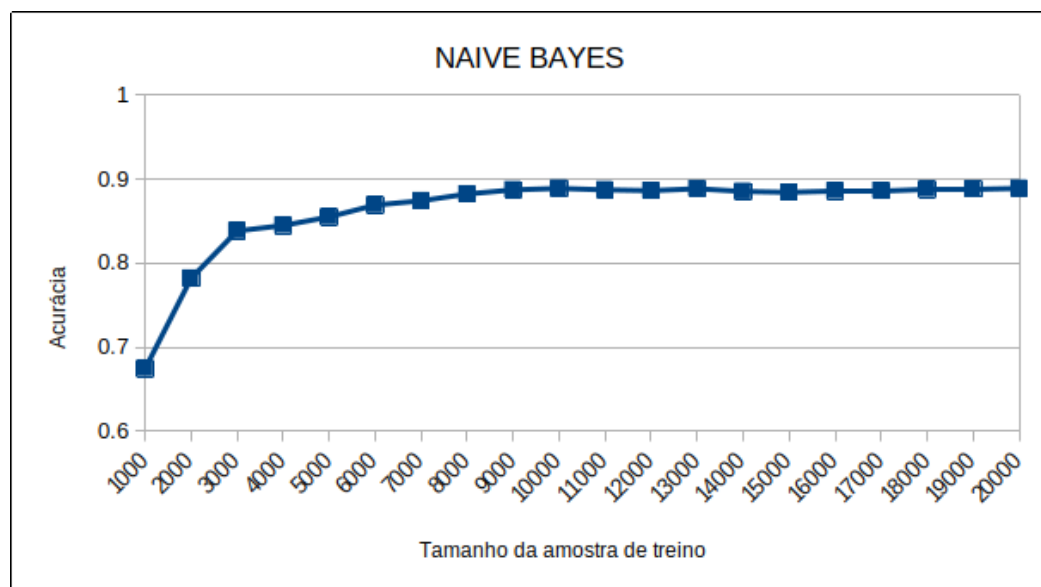
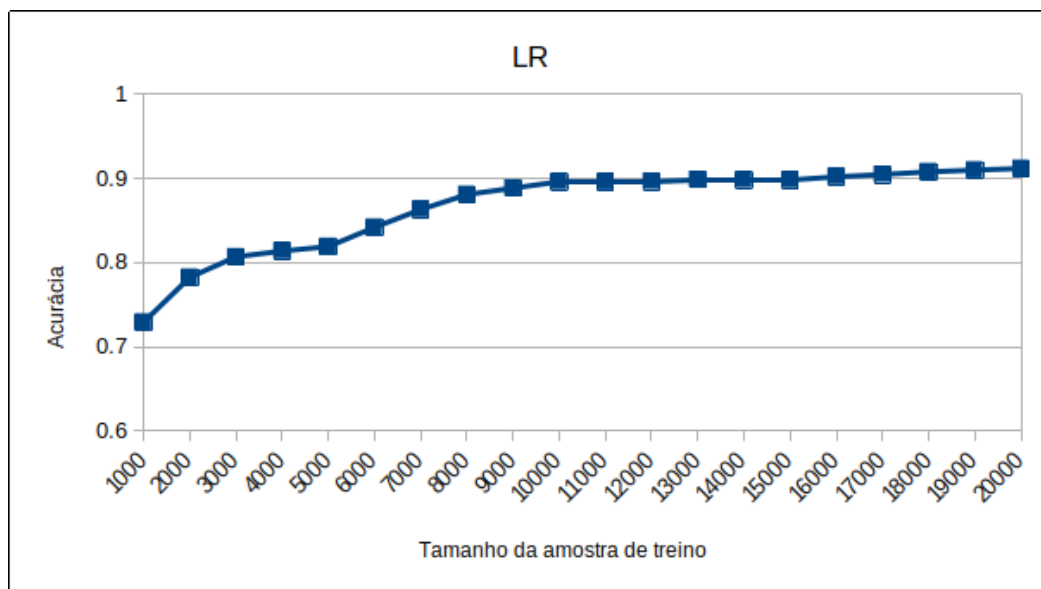
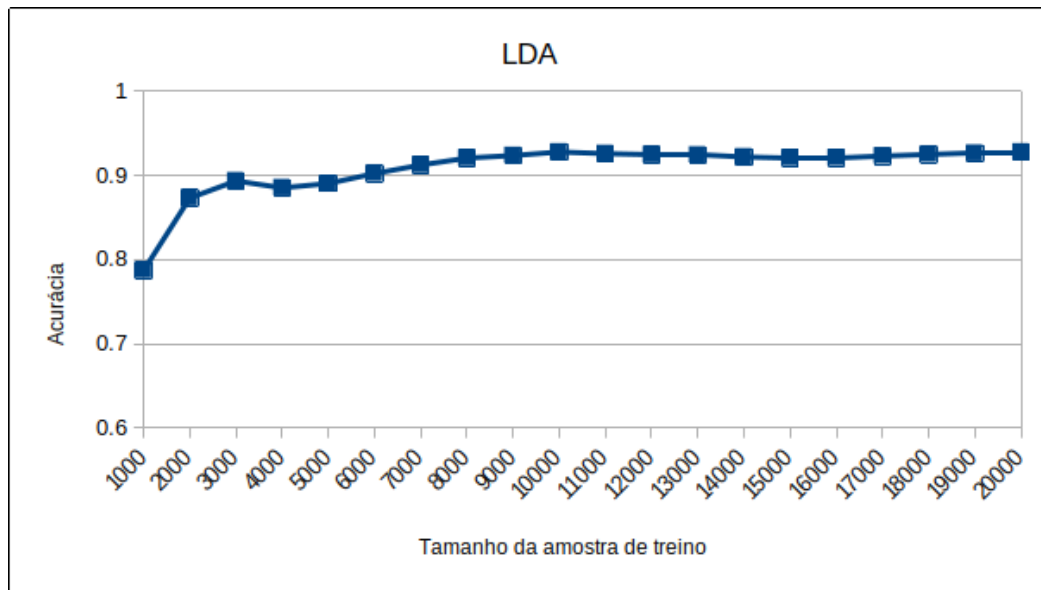
Tais algoritmos foram importados da biblioteca scikit-learn para Python. Utilizando-se os parâmetros padrões para todos os classificadores, uma vez que não houve uma especificação de quais deveriam ser usados. Com exceção para o LR, que foi usado um número máximo de iterações de 150, teto do número mínimo observado para que todos os testes convergissem sem emitir *warning*.

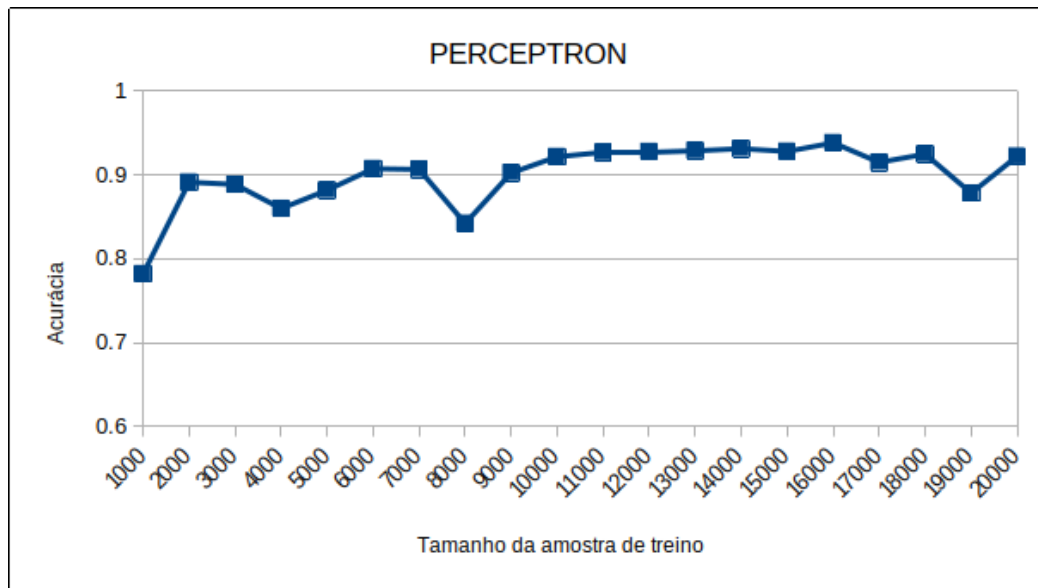
Análise:

Observação: primeiro serão exibidos os resultados e depois serão discutidos.

- **Comparação de desempenho em blocos de amostras**







Primeiramente, percebe-se diferentes comportamentos em cada classificador de acordo com o tamanho da amostra:

KNN: A partir de 10000 amostras começa a perder e ganhar desempenho alternadamente. Entretanto, chega a ter um pico em 20000 amostras, mesmo que tenha ganhado apenas 0.002 de acurácia. Resultando em um valor final de aproximadamente 0.94.

LDA: Apesar de se apresentar de forma mais inconsistente inicialmente, estabiliza-se no seu pico de desempenho com 10000 amostras. Acurácia final igual a 0.927.

LR: Com o início mais lento de todos os classificadores, o LR se posiciona da mesma forma que o KNN, o auge de seu desempenho se dá nas 20000 amostras – 0.911.

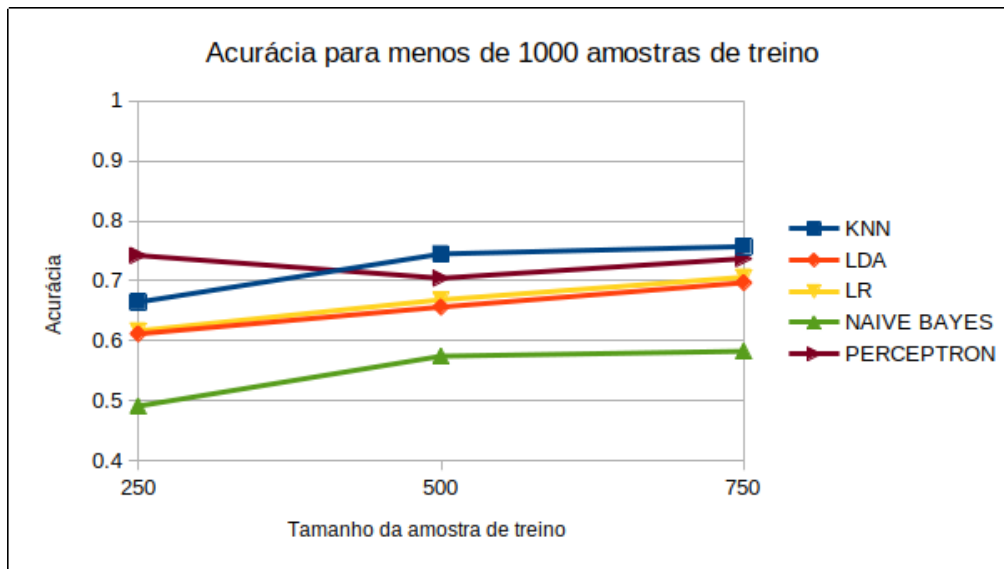
Naive Bayes: Curiosamente, o seu pico se dá em dois lugares (9 dígitos de precisão), em 10000 e 20000 amostras. Logo, consideraremos o primeiro, com uma acurácia de 0.889.

Perceptron: Apesar dos ganhos rápidos de desempenho, se manteve inconsistente por toda a execução. Tendo o pico em 16000 amostras com um resultado final de 0.938.

De forma geral, pode-se dizer que o LDA obteve os melhores resultados em proporção a base de dados. Exigindo apenas 62.5% do que o *Perceptron* necessitou para obter 0.011 a mais de acurácia.

- **Melhor desempenho com uma base menor que 1000 amostras**

Ao se testar o desempenho em cenários com poucos exemplos, considerou-se as seguintes quantidades: 250, 500 e 750.

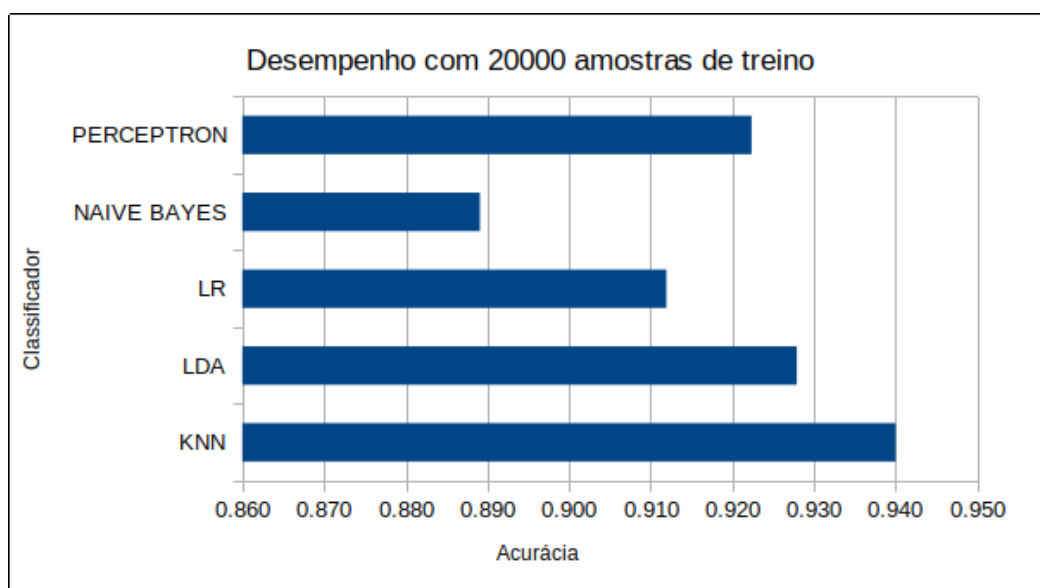


Pode-se perceber que com muitas poucas possibilidades de treino, o Perceptron foi o que teve o melhor resultado.

Entretanto, com uma base um pouco maior já foi suficiente para que o KNN passasse a liderar em diante. Além de que manteve um ganho consistente de desempenho, ao invés do seu concorrente. Possuindo de forma geral uma melhor classificação.

- **Melhor desempenho utilizando a base inteira de treino (20000 amostras)**

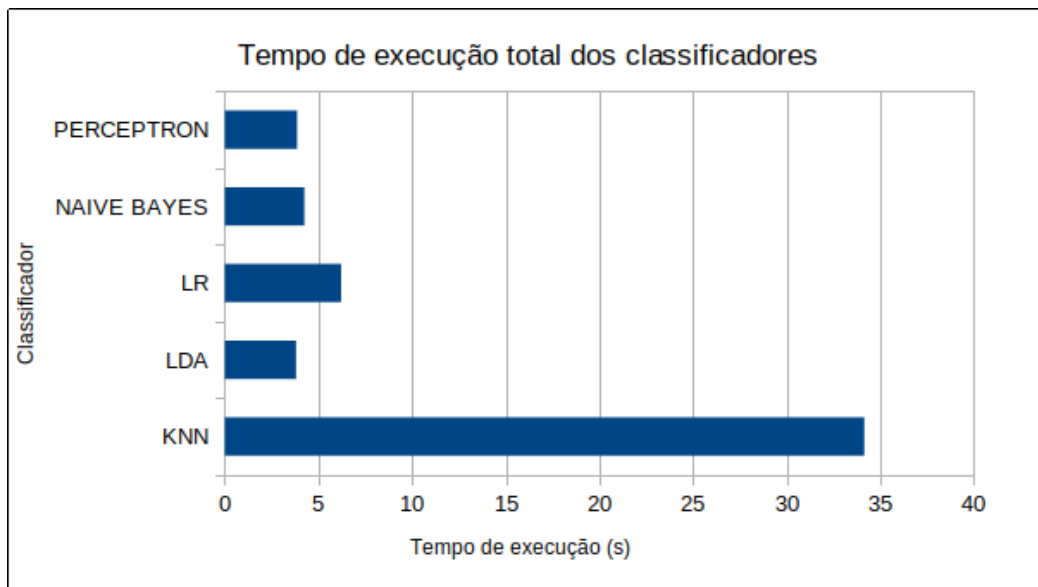
Utilizando-se da base inteira de treinamento, notou-se novamente um domínio do algoritmo KNN, com 94% de acertos.



- **Menor tempo de classificação**

Realizando a medição do tempo de execução total dos classificadores, incluindo treinamento e classificação. E considerando a execução "exclusiva" em um processador Ryzen 5 3600 em *stock* (variando o *clock* entre 3.6 e 4.2 Ghz).

Pode-se dizer que houve um empate entre o LDA e o *Perceptron* na decisão pelo classificador mais rápido, com um tempo final de 3.801 e 3.856 respectivamente e uma variação menor que 5%.



- **Análise das matrizes de confusão**

Analizando as matrizes de confusão resultantes dos classificadores (utilizando toda a base de treinamento), percebeu-se que os erros e acertos variam muito entre os classificadores. Para uma abordagem compreensível, foi estabelecido dois intervalos e filtrados os classificadores que não pertencem a eles.

Primeiramente, classificadores com números de acertos maiores ou menores que: mediana de acertos +5% ou -5%, respectivamente (intervalo total = 10%)

Classificadores com menos 5% ou mais em comparação com a mediana	
Confusão	Classificador
0	Perceptron
1	Naive Bayes
2	
3	
4	
5	
6	
7	
8	
9	

Classificadores com mais 5% ou mais em comparação com a mediana	
Confusão	Classificador
0	
1	knn
2	
3	
4	
5	Perceptron
6	
7	
8	Perceptron
9	

Analisando os resultados, podemos dizer que seria uma boa ideia utilizar o KNN para melhorar os resultados de outros classificadores, talvez o *Perceptron* também dependendo do número de confusões no *label* 0. Entretanto, não seria aconselhável utilizar o *Naive Bayes* para isso.

O segundo intervalo foi de: mediana de acertos +2.5% ou -2.5% (intervalo total = 5%).

Classificadores com menos 2.5% ou mais em comparação com a mediana	
Confusão	Classificador
0	p
1	lr, nb
2	nb
3	nb
4	
5	lr
6	nb
7	nb
8	
9	nb

Classificadores com mais 2.5% ou mais em comparação com a mediana	
Confusão	Classificador
0	knn
1	knn, lda, p
2	
3	
4	p
5	p
6	knn
7	knn, lda
8	p
9	lda, p

Observamos que o algoritmo de Naive Bayes foi de longe o que apresentou os piores resultados entre os classificadores, seguido pelo LR. Logo, não é aconselhável utilizá-los. Por outro lado, seria interessante utilizar o Perceptron, seguido pelo KNN e o LDA. Talvez, ainda mais interessante seria se estivessem em conjunto, ajudando na confusão do Perceptron no dígito 0.