

## Laboratório Aprendizagem de Máquina

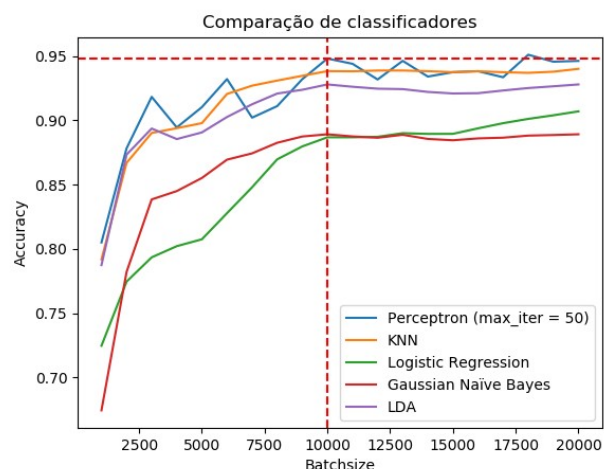
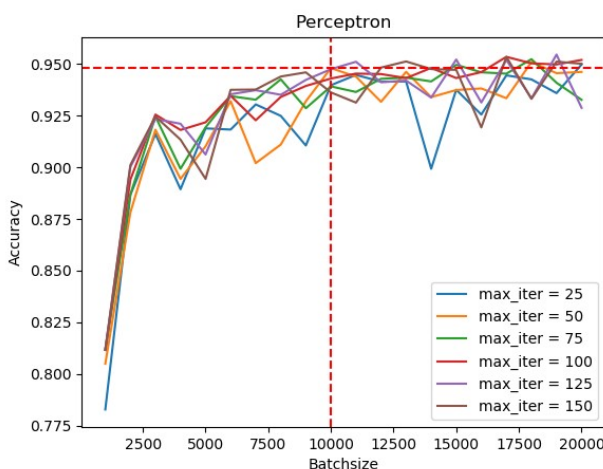
### Lab3: Avaliar desempenho do perceptron.

1) Execute o programa **perceptronDemo.py** para visualizar a fronteira de decisão construída pelo perceptron. Você pode verificar que se os dados forem linearmente separáveis o perceptron encontra uma fronteira.

2) Execute o script **perceptron.py** e compare os resultados com os classificadores utilizados no laboratório passado.

a) Quantas iterações são necessárias para que o perceptron encontre a melhor acurácia?

b) Qual é o tamanho da base de treinamento para que perceptron alcance a melhor acurácia?



O desempenho do algoritmo Perceptron foi avaliado variando-se o argumento **max\_iter** = (25, 50, 75, 100, 125 e 150) para diferentes tamanhos de conjunto de treinamento (batchsize), até o uso total da base de treino (20000 instâncias). Em termos gerais, as variações testadas de **max\_iter** não produziram aumento substancial na métrica de desempenho do algoritmo, mostrando curvas similares, com picos e declínios de acurácia em função do tamanho da base de treino, isto é, ora o aumento da base de treino foi benéfico proporcionando aumento da acurácia (mesmo que mínimo), ora não o foi. O aumento do desempenho no Perceptron foi bastante significativo quando aumentou-se o tamanho da base de treino de 1000 para 2000 exemplos. No entanto, a partir deste ponto (batchsize = 2000) o aumento da acurácia foi pouco significativo. O aumento de ambos (batchsize e **max\_iter**) implica na elevação no custo computacional. Dessa forma, para o conjunto de treino em específico, a decisão do tamanho da base e do argumento **max\_iter** da função **Perceptron()** devem ser considerados conjuntamente para escolha do modelo com melhor capacidade de generalização. Acredito que uma base de treino com **batchsize = 10000** e **max\_iter = 50** seja suficiente para que o algoritmo alcance uma acurácia satisfatória. Por outro lado, poder-se-ia usar uma base menor ponderando, porém, o uso de outro valor em **max\_iter**. Finalmente, comparou-se o desempenho do Perceptron (batchsize = 10000 e **max\_iter = 50**) com os classificadores (KNN, Logistic Regression, Naive Bayes e LDA). O desempenho do Perceptron apesar da sua simplicidade foi claramente superior do que Logistic Regression, Naive Bayes e LDA. Por outro lado, pode-se dizer que seu desempenho foi bastante similar ao KNN, porém a observância do custo computacional coloca o Perceptron em um patamar de superioridade em relação ao KNN.