**Inteligência Artificial -  UFRJ - 2019.1**

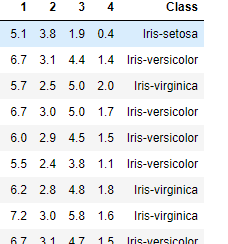
Bruno Hryniewicz dos Santos Cruz  - DRE 115150099

**Relatório de trabalho**

**Iris Plants Dataset.**

**Visão geral**

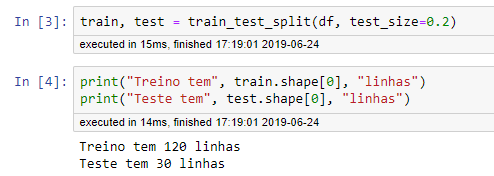
Este dataset se trata objetos com 4 atributos e estes atributos definem 1 entre 3 classes para esse objeto.



Para classificar estes objetos será utilizado o algoritmo PLA (Perceptron Learning Algorithm).

**Conjuntos** **Treino e Teste**

Para utilizar o PLA dividimos de maneira aleatória dataset em dois conjuntos, um de treino e um de teste, com 80% em treino e 20% em teste em relação ao conjunto geral.

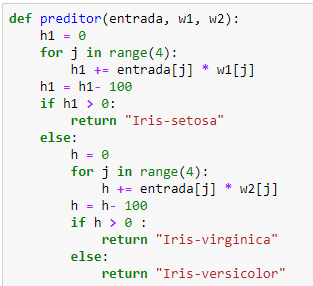


**Treinamento**O PLA tenta dividir dois conjuntos se eles são linearmente separáveis. Sendo assim, para dividirmos em 3 foi preciso utilizar por duas vezes o PLA. A primeira foi feita para separar a classe “Iris-setosa” das demeais classes. Esses conjuntos eram linearmente separável o algoritmo chegou ao fim retornando o vetor que prediz se a classe é “Iris-setosa” ou não.

Na segunda fase do PLA, para todos aqueles que meu preditor diz não ser “Iris-setosa” verifico qual das duas classes restantes eles são. Sendo que essas classes não são linearmente separáveis. Então, meu programa precisa de um caso de parada. Assim, a cada iteração testo o novo modelo e se ele for melhor que o anterior, o armazeno, então limito o número máximo de iterações.

**Predição**

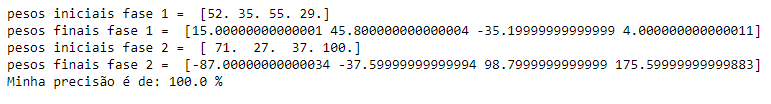
Dado as duas equações do treinamento (w1, w2) e o vetor x de atributos, segue o algoritmo da predição.



**Teste**

Após implementar o treinamento e a predição, posso tentar prever itens no meu conjunto teste e descobrir a eficácia do meu modelo ao comparar com o valor esperado.

Para isso, testo de duas maneiras, a primeira é fazendo um modelo com os pesos de entrada aleatórios e os conjuntos de teste e treino também aleatórios. Assim minha precisão fica oscilando pois sempre tenho modelos diferentes onde, na segunda fase podem ter parado antes de chegar na melhor equação para aquele conjunto de treino e teste. Assim tenho um modelo novo e com precisão diferente cada vez que rodo. A outra maneira é pegar os conjuntos de teste e treino e os pesos de entrada fixos assim fixando um modelo.



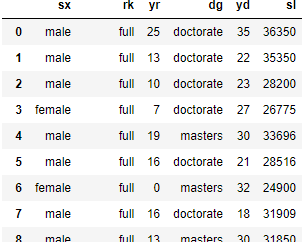
**Conclusão**

O dataset pode ser estimado com uma precisão acima de 90% na maioria dos modelos que treinei.

**Discrimination in Salaries**

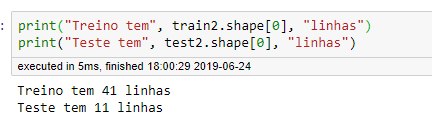
**Visão geral**

Este dataset se trata atributos de professores e seus salários. Usaremos Uma Regressão linear para tentar prever os salários dos professores. Os atributos que são strings foram traduzidos para atributos numéricos.

****

**Conjuntos** **Treino e Teste**

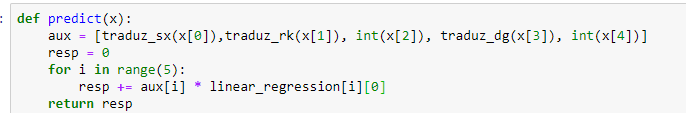
Para utilizar o PLA dividimos de maneira aleatória dataset em dois conjuntos, um de treino e um de teste, com 80% em treino e 20% em teste em relação ao conjunto geral.

****

**Treinamento**O treinamento da regressão linear é muito simples, bastando apenas calcular a matriz pseduo inversa e multiplicar pelos vetor saída esperada, obtendo assim o vetor de pesos que irá ser usado no preditor. Serão feitos 2 treinamentos, um considerando sexo e outro desconsiderando sexo, em busca de um viés nos dados.

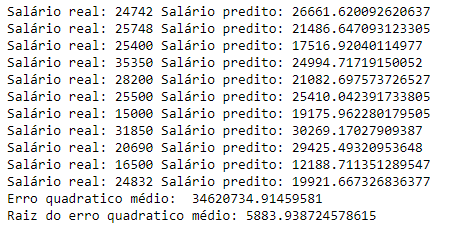
**Predição**

Dado o vetor de saída da regressão linear e o vetor x de atributos, segue o algoritmo da predição.

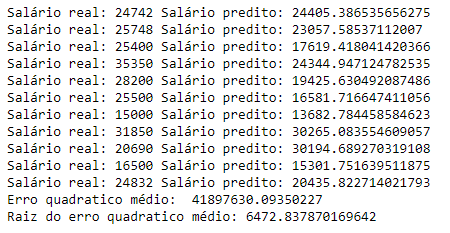


**Teste**

Após realizar o teste com os dois modelos ( incluindo ou não sexo como parâmetro), descobri que os resultados são diferentes entre eles e dependendo do conjunto de treino e teste, ocorre com uma grande frequência que o modelo com sexo incluso erra menos do que o modelo sem esse atributo.

Considerando sexo: 

Desconsiderando sexo:



**Conclusão**

Utilizando regressão linear posso mensurar salário dos professores com uma precisão razoável. Porem o mais interessante a se observar que a precisão da predição ser melhor considerando o sexo do indivíduo do que sendo imparcial com o sexo, o que mostra que neste dataset existe um viés de diferença salarial de acordo com o sexo.