Universidade Federal de Mato Grosso do Sul
Campus Ponta Porã
Inteligência Artificial

Trabalho Prático I

Regressão Linear

Aluno: Daniel de Leon Bailo da Silva
Professor: Daniel Matte Freitas

Setembro
2019

Sumário

Resumo

Este trabalho consiste em implementar a *Regressão Linear com uma variável* utilizando sua forma matricial. O objetivo do mesmo consiste em plotar os valores de $J(\theta)$ para diferentes valores de α, que seria a Taxa de Aprendizagem(Learning Rate) do algoritmo.

Este trabalho está disponibiliado num repositório do *GitHub* para melhor controle do versionamento do programa, e o mesmo se encontra em um *Jupyter Notebook (Python)* e em um script escrito em *Python* também.

https://github.com/danbailo/T1-Inteligencia_Artificial

1 Introdução

Em estatística ou econometria, regressão linear é uma equação para se estimar a condicional (valor esperado) de uma variável y, dados os valores de algumas outras variáveis x.

A regressão, em geral, tem como objetivo tratar de um valor que não se consegue estimar inicialmente.

A regressão linear é chamada "linear"porque se considera que a relação da resposta às variáveis é uma função linear de alguns parâmetros. Os modelos de regressão que não são uma função linear dos parâmetros se chamam modelos de regressão não-linear. Sendo uma das primeiras formas de análise regressiva a ser estudada rigorosamente, e usada extensamente em aplicações práticas. Isso acontece porque modelos que dependem de forma linear dos seus parâmetros desconhecidos, são mais fáceis de ajustar que os modelos não-lineares aos seus parâmetros, e porque as propriedades estatísticas dos estimadores resultantes são fáceis de determinar.

Modelos de regressão linear são frequentemente ajustados usando a abordagem dos mínimos quadrados, mas que também pode ser montada de outras maneiras, tal como minimizando a "falta de ajuste"em alguma outra norma (com menos desvios absolutos de regressão), ou através da minimização de uma penalização da versão dos mínimos quadrados. Por outro lado, a abordagem de mínimos quadrados pode ser utilizado para ajustar a modelos que não são modelos lineares. Assim, embora os termos "mínimos quadrados"e "modelo linear"estejam intimamente ligados, eles não são sinônimos.

2 Métodos

2.1 Notação Algébrica

Definições:

- $h(x) = \theta_1 x + \theta_0$
- $J(h) = \frac{1}{2m} \sum_{i=0}^{m} ((h(x_i) - y_i)^2)$

Algoritmo; sendo i a iteração:

- $\theta_1^i = \theta_1^{i-1} - \alpha \frac{\partial}{\partial \theta_1^{i-1}} J \rightarrow \ \theta_1^{i-1} - \alpha \frac{1}{m} \sum_{i=0}^{m} ((h(x_i) - y_i)x)$
- $\theta_0^i = \theta_0^{i-1} - \alpha \frac{\partial}{\partial \theta_0^{i-1}} J \rightarrow \ \ \theta_0^{i-1} - \alpha \frac{1}{m} \sum_{i=0}^{m} ((h(x_i) - y_i)1)$

2.2 Notação Matricial

Definições:

Considere x' a entrada, então:

- $X = \begin{bmatrix} 1 & x'_1 \\ 1 & x'_2 \\ 1 & .. \\ 1 & x'_m \end{bmatrix}$, $\Theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$
- $H = X \cdot \Theta$
- $E = H - Y$
- $J = \frac{1}{2m}(E^T \cdot E)$

Algoritmo; sendo i a iteração:

- $\Theta_i = \Theta_{i-1} - \frac{\alpha}{m}(X^T \cdot E)$

Sendo que a cada iteração, o valor de H, E e J devem ser atualizados.

2.3 Cáculo da Regressão Linear em Python

A função abaixo realiza o cálculo da Regressão Linear na sua forma matricial, foi escolhido essa forma para ser programado, pois assim é realizado menos operações e também operações matriciais são paralelizáveis, portanto, para enormes *Datasets*, é inviável realizar o cálculo de forma sequencial(iterativa).

Esta função, recebe como parâmetro o número de iterações que irá treinar o algoritmo, o α, que é a taxa de aprendizagem, e os valores de θ inicial, que são os pesos; é importante observar que, nesse caso, a variável "theta" é um vetor coluna $2x1$.

```
    def linear_regression(num_it, alpha, theta):
        err = np.zeros(num_it)
        for i in range(num_it):
            h = np.dot(X,theta)
            e = h-y
            J = (np.dot(e.T,e)/(2*m))
            err[i] = J
            theta = theta - ((alpha/m)*np.dot(X.T,e))
        return err, theta
```

3 Análise dos Resultados

Considerando os dados de teste que foram disponibilizados, com a entrada representando o eixo X, e a saída o eixo Y, o algoritmo tem o objetivo de predizer a saída, visto que existe uma relação linear entre os dados, e para isso, foi utilizado algumas técnicas dispondo de gráficos para selecionar o melhor valor de α local dentre o intervalo conhecido para os testes.

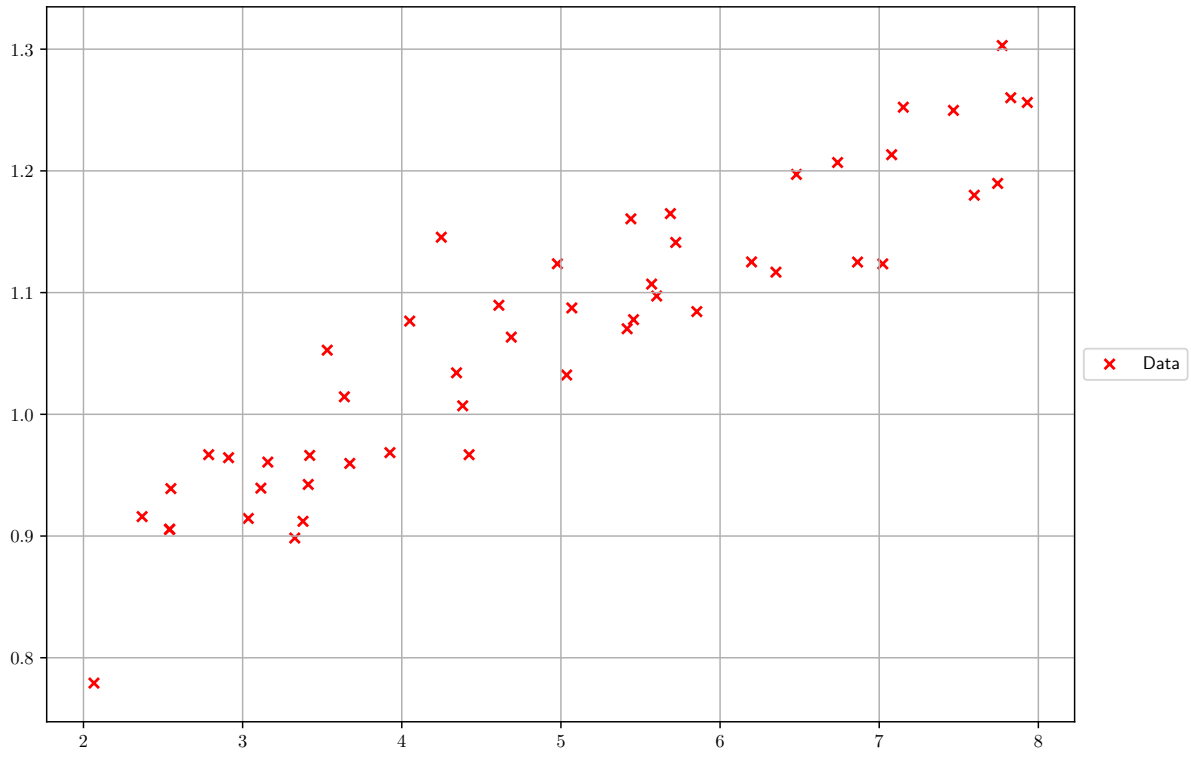

Figura 1: Dados disponibilizados.

Os valores deste intervalo mencionado acima, foram escolhidos partindo de uma hipótese, e a partir desta hipótese, foi possível entender como o *erro* iria se comportar para aumentar ou diminuir o valor deste α. É importante ressaltar, que quanto menor o erro, ou seja, quanto mais próximo de 0, melhor será a predição do algoritmo. No melhor caso, onde o erro é 0, a predição é perfeita.

Para todos os testes, os pesos (θ) foram inicializados com θ_0, θ_1=$(0, 0)$ e inicialmente foi-se utilizada 100 iterações para observar o comportamento do algoritmo.

Com isso, o primeiro insight obtido após selecionar um valor para o α, foi de que era necessário diminuir ainda mais seu valor, pois o mesmo estava estourando exponecialmente, ou seja, estava se divergindo da solução, e o que estamos buscando, é um α que se convirja a solução ideal.

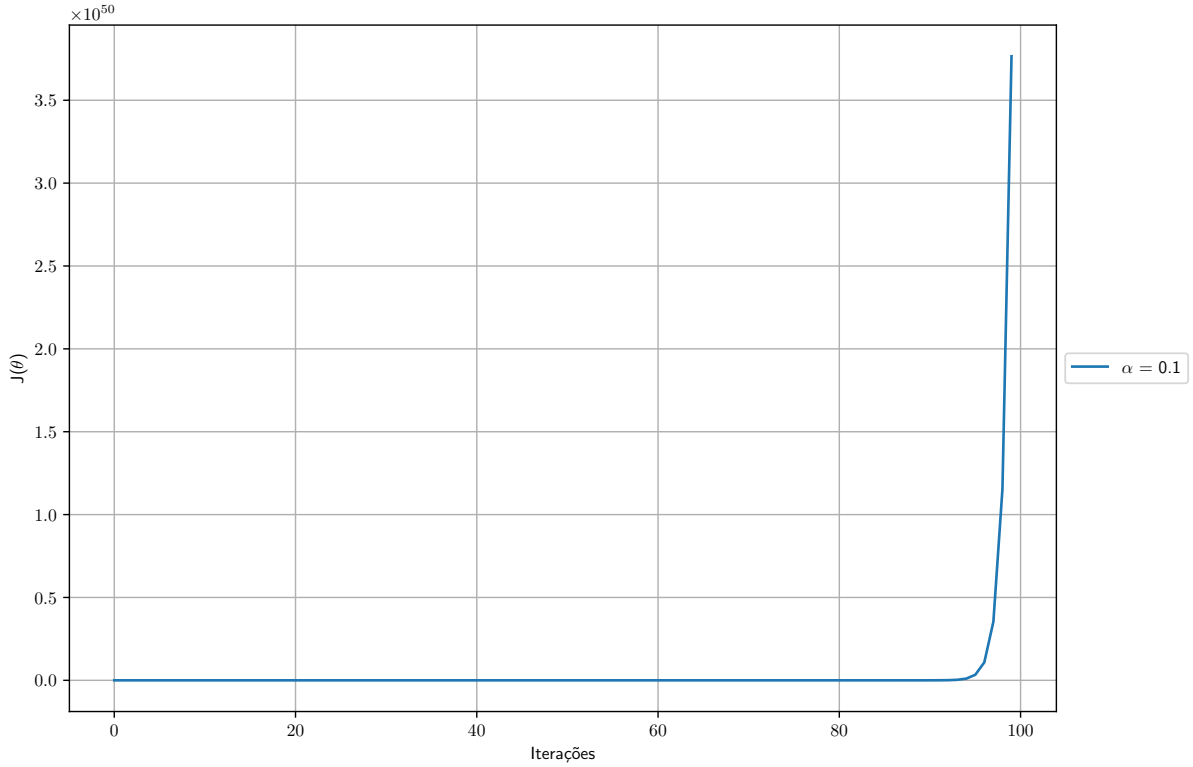

Figura 2: Primeiro insight obtido a partir de um valor de α inicial.

Com os dados dispostos, se realizarmos a aplicação do algoritmo, utilizando o valor de α acima, que foi de 0.1, iremos obter a seguinte predição.

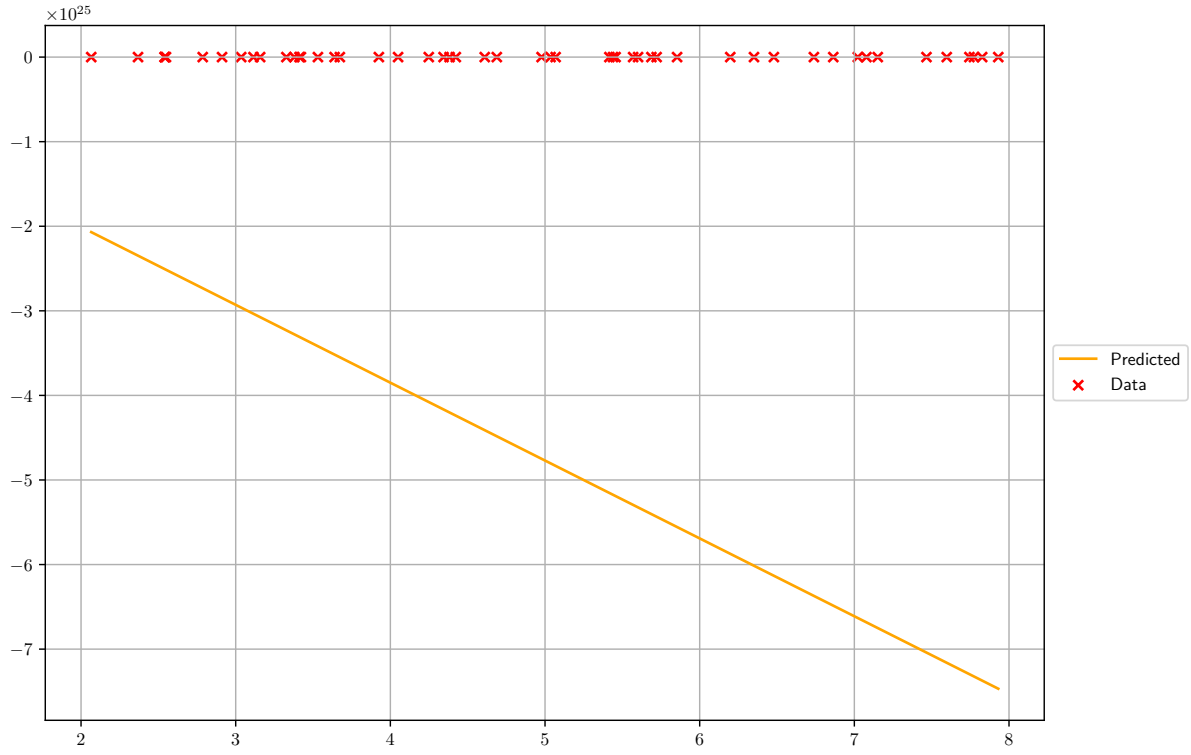

Figura 3: Predição com mau treinamento.

Como podemos ver, numa escala logarítmica, a nossa predição está absurdamente longe dos dados que foram dispostos, aproximadamente 10^{25} se comparado no eixo y.

Então, foi realizado alguns testes para tentar selecionar o melhor valor para α, logo, é importante notar que, quanto mais próximo de 0, melhor será a solução que o algoritmo irá encontrar.

Neste gráfico, é possivel observar que, para alguns valores de α, como o próprio 0.1, que já foi testado anteriormente, e também, 0.08 e 0.09, convergem da solução e "estouram" exponencialmente, portanto esses valores serão descartados para as próximas análises.

É possivel notar também que, para alguns valores temos uma convergência mais rápida, ou seja, essses valores se aproximam mais rápido de 0; isso significa que, na *Descida de Gradiente*, o algoritmo irá realizar um salto bem próximo de uma boa solução(valor próximo de 0), e em contrapartida, alguns valores também convergem, mas de forma mais lenta.

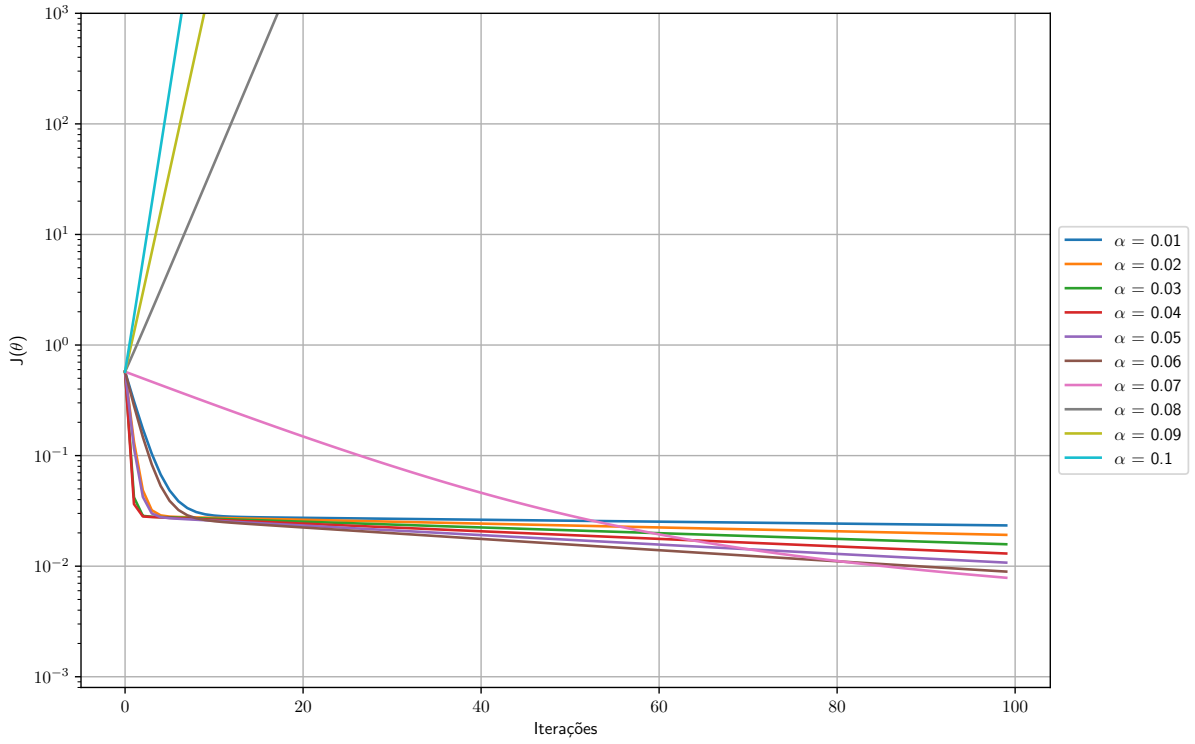

Figura 4: Verificando o erro a partir de diferentes valores de α.

Logo, a seguinte questão é levantada: "Para este modelo, o valor de α que converge mais rápido, será de fato o melhor valor para utilizar no treinamento do algoritmo?".

Para responder essa questão, foi realizado alguns testes para a quantidade atual de iterações e posteriormente será aumentado este número, para verificar como que o modelo se comporta quando submetido a uma quantidade maior de iterações(treinamento).

No gráfico abaixo, pode-se observar melhor como está atualmente o comportamento do algoritmo para cada valor de α. De primeira impressão, é possivel ver que o valor que mais demora a convergir é 0.07, porém, o mesmo é o que está mais próximo do eixo, pode-se afirmar que para 100 iterações, este seria um bom valor, mas ainda não sabemos se este será de fato o melhor, visto a sua demora a convergir.

Entretanto, não é possivel saber de fato, qual é o valor que converge mais rápido, pois nessa escala, os valores estão muito próximos um do outro.

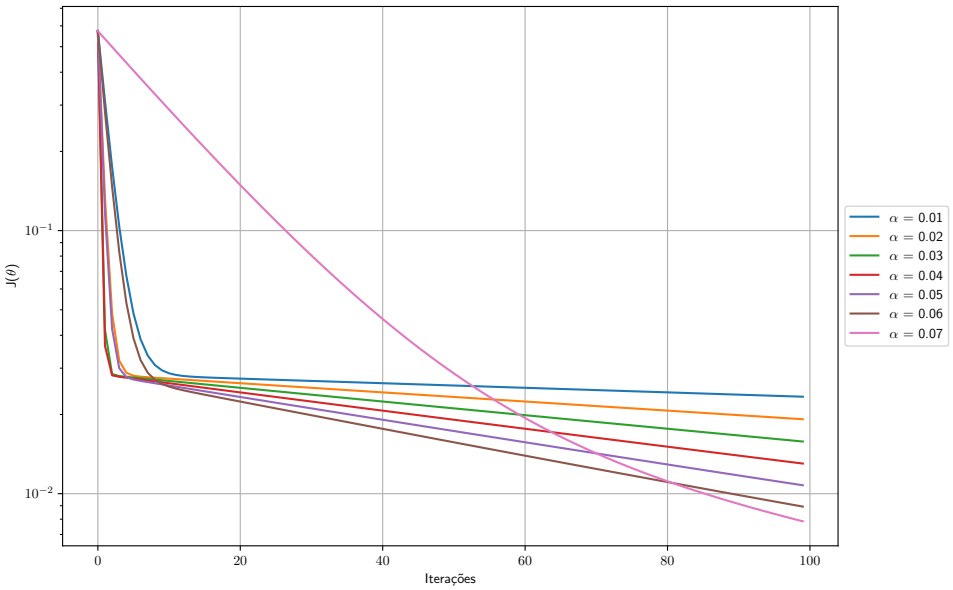

Figura 5: Intervalo de 100 iterações.

Com isso, o gráfico abaixo foi estreito num intervalo menor, relacionado a figura 8. Agora é possível ver com mais clareza que o valor 0.04, referente a linha vermelha, é o valor que converge mais rápido. Portanto, este pode também pode ser considerado um bom valor para realizar o treinamento.

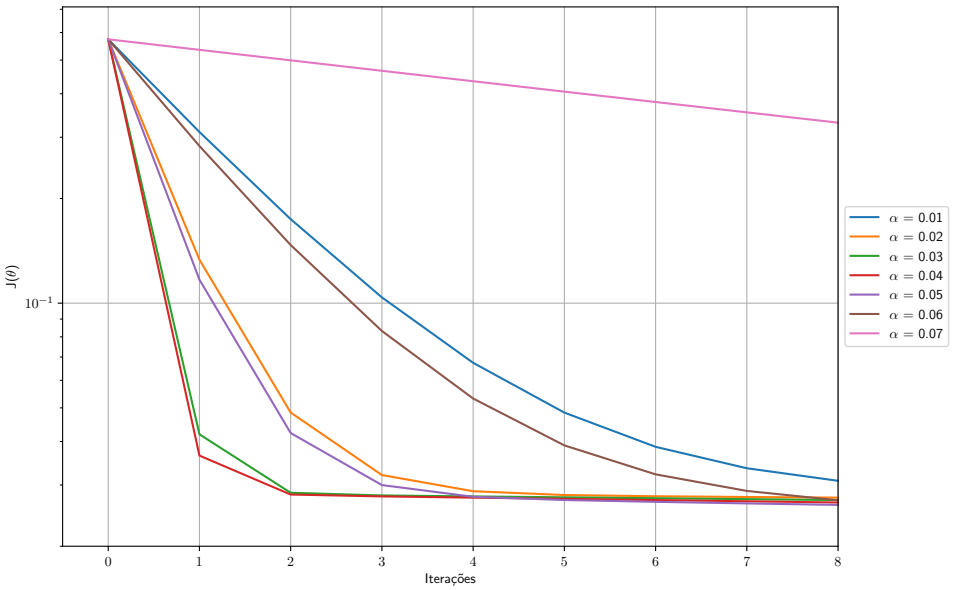

Figura 6: Intervalo estreito mais a esquerda do eixo.

Continuando a análise para selecionar o valor de α, agora veremos como está se comportando os valores num intervalo mais restrito ao final do eixo x. Assim como na figura 8, como os valores estão bem diferentes entre si, o comportamento é o mesmo que já foi notado.

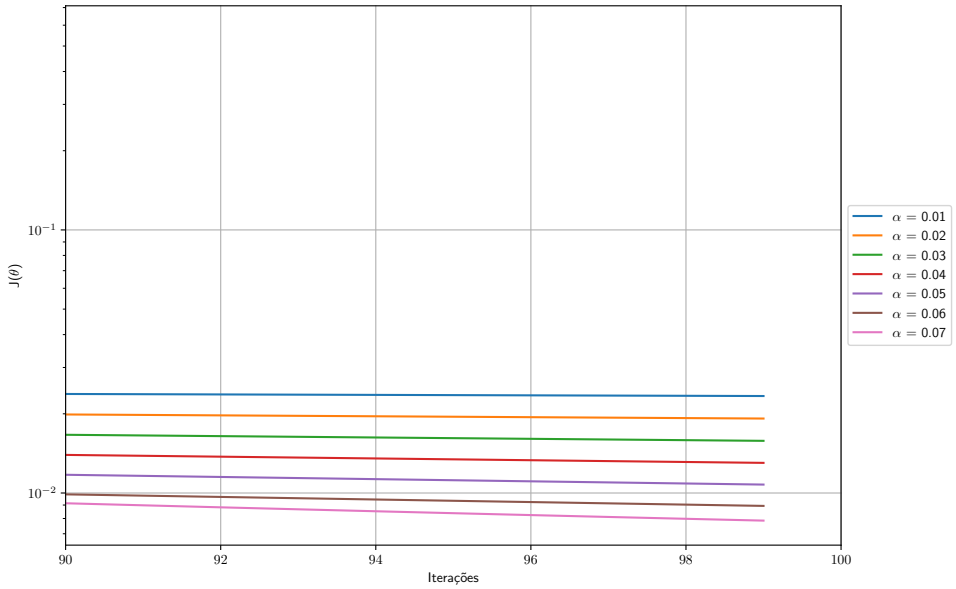

Figura 7: Intervalo estreito mais a direita do eixo.

Porém, é importante lembrar que, para este treinamento foi utilizando somente **100 iterações**, este é um número pequeno de iterações, mesmo sabendo que os dados dispostos não são tão grandes; porém, para obter uma análise mais concreta para selecionar um bom α para predizer os valores de y, será realizado outra análise, mas agora utilizando **1000 iterações**, e será observado se o comportamento do modelo continua o mesmo, ou se sofre alguma alteração durante o treinamento.

3.1 Treinamento utilizando 1000 iterações

Ao realizar o treinamento utilizando 1000 iterações, pode-se notar que alguns valores de α, temos alguns valores que convergiram a partir das 500 iterações e outros não, mesmo que com 100 iterações, pareciam ser bons valores, pelo fato de convergir mais rápido que outros valores.

Logo, podemos afirmar que, não é pelo fato de uma solução convergir mais rápido, que esta será melhor que as outras, pois ela pode melhorar depois em outro intervalo de iterações.

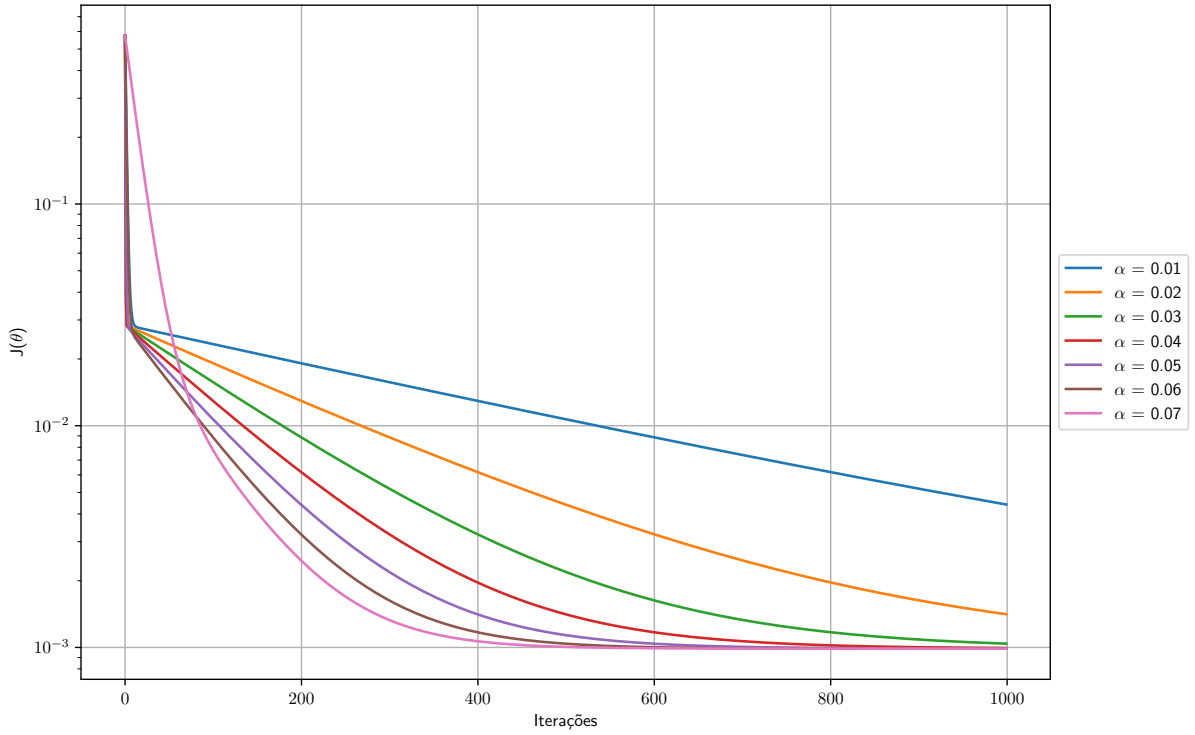

Figura 8: Intervalo de 1000 iterações.

Porém, novamente, como agora os valores estão muito próximos um do outro, é necessário ampliar o intervalo ao final do eixo x para saber de fato qual valor está com a menor amplitude em relação ao eixo y. Para isso, foi gerado dois gráficos, a figura 9 e 10, as duas figuras estão estreitas no intervalo $[990, 1000]$ para X, e $[10^{-3}, 10^{-1}]$, [9.8 x 10^{-4}, 1.06 x 10^{-3}], respectivamente.

No gráfico da figura 9, é possivel notar a presença de mais de uma linha entre as cores *verde* e *rosa*, que representam os valores 0.03 e 0.07, respectivamente, mas não dá para distinguir qual valor de fato é este.

No gráfico da figura 10, é pode-se observar quais são valores presentes neste intervalo, considerando que este intervalo de y é muito pequeno.

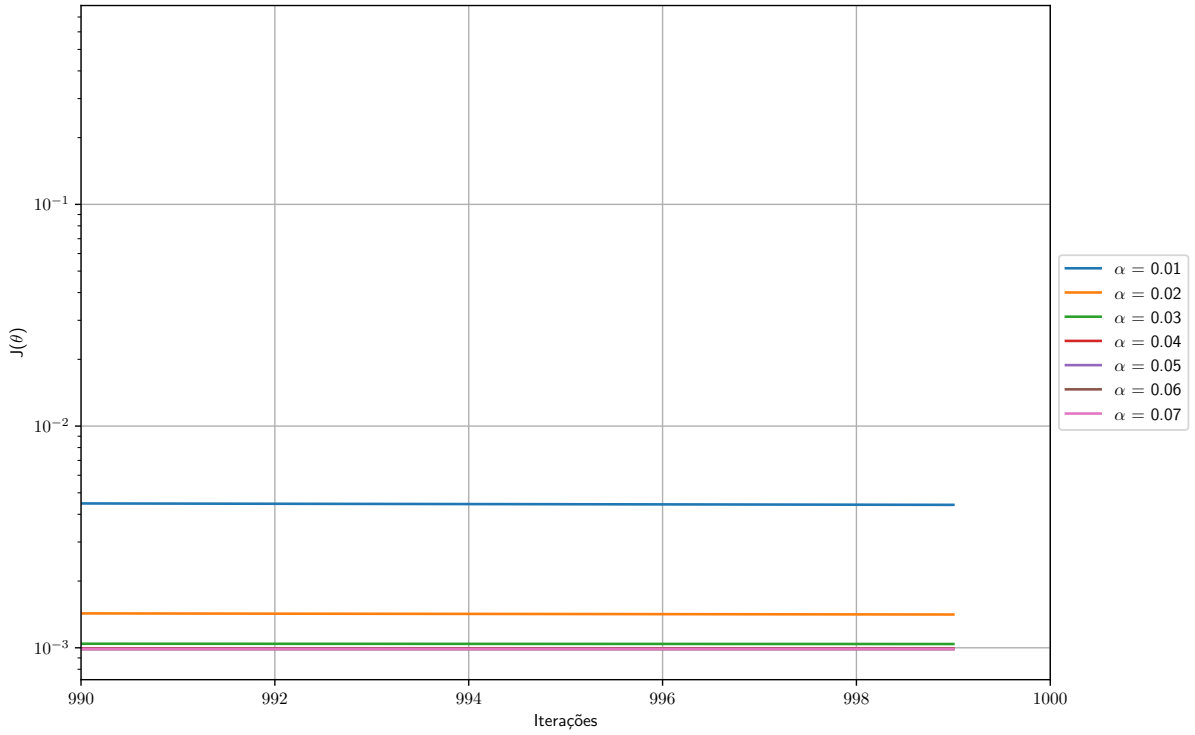

Figura 9: Intervalo estrito mais a direita para 1000 iterações.

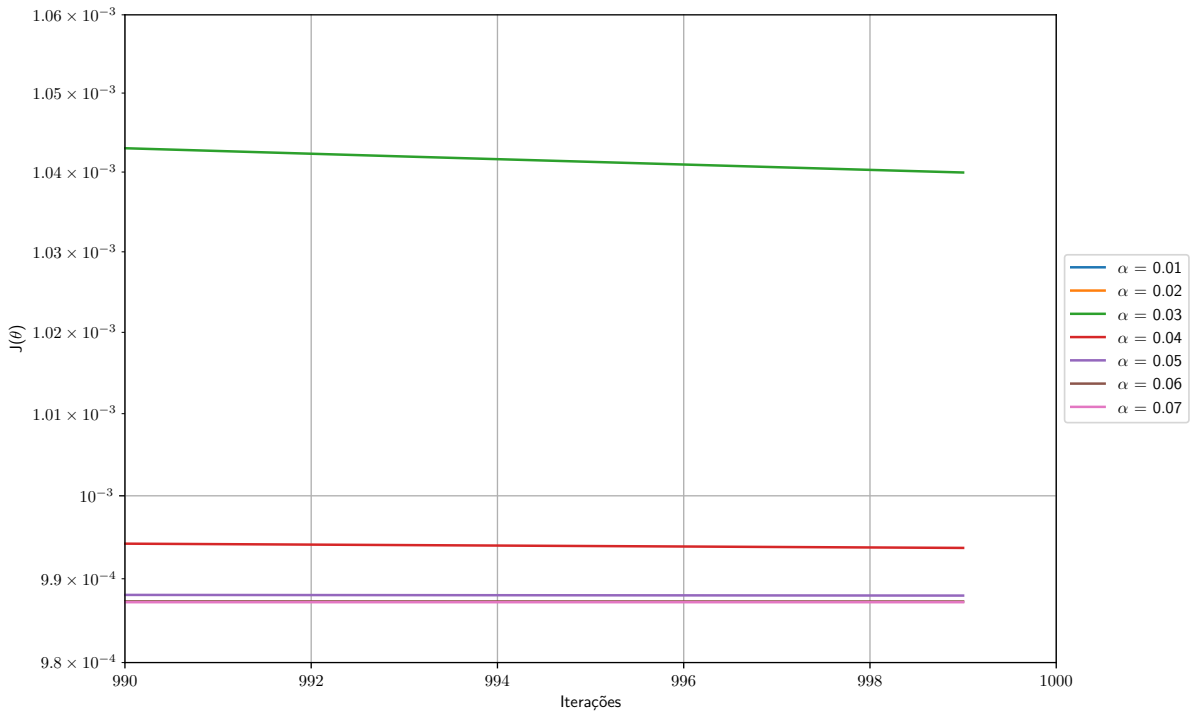

Figura 10: Zoom no intervalo.

4 Interpretação dos Resultados

Após estudarmos o comportamento do erro($J(\theta)$) deste modelo para os α selecionados, podemos de fato selecionar um valor de α global dentre os locais utilizados para estudo. Com o estudo do erro, é possivel saber quando teremos um bom e um mau modelo, ou seja, quanto mais divergente do eixo for este erro, pior será o modelo.

O gráfico abaixo, mostra as predições para todos os valores de α estudados quando estes convergiam.

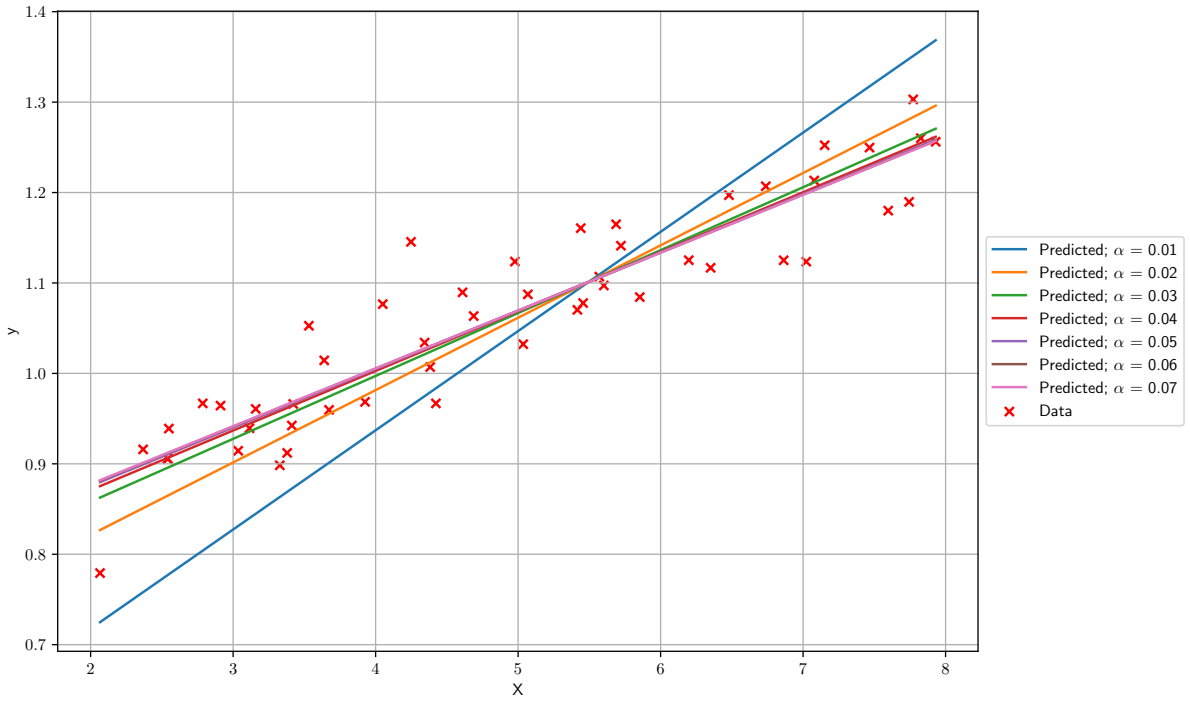

Figura 11: Todas as predições para os $\alpha's$ convergentes.

5 Conclusão

Após as análises, podemos ver que o α que mais se mostrou próximo de uma solução exata foi o de valor 0.07.

Podemos ver que, dispersos no plano temos alguns *outliers*, mas de forma geral, existe uma relação linear entre os dados.

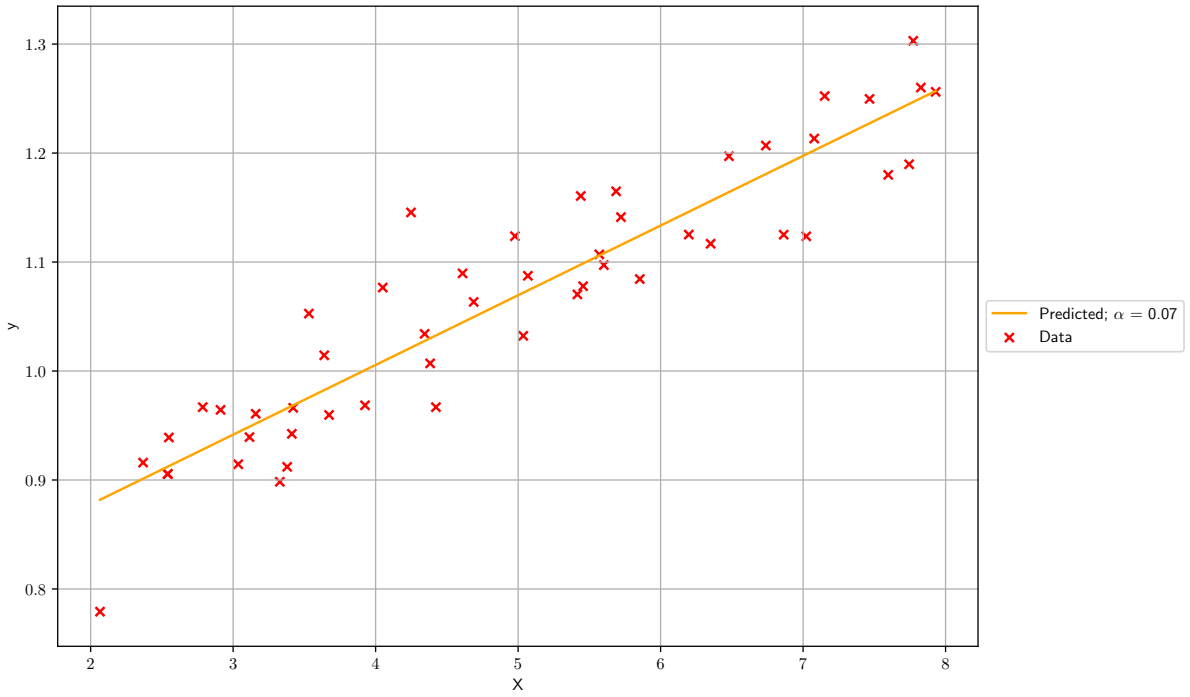

Figura 12: Modelo final.