Primeiro Trabalho Prático - MO444

Luis Gustavo Lorgus Decker
209819
luisgustavo.decker@gmail.com

Luiz Antonio Falaguasta Barbosa
075882
lafbarbosa@gmail.com

I. Introdução

Uma regressão linear consiste em uma equação na forma

$$h\theta(x) = \theta_0 + \theta_1 * x_1 + \theta_2 * x_2 + \cdots + \theta_n * x_n, \quad (1)$$

onde θ_i é um coeficiente e x_i é uma característica oriunda de um dado sob qual queremos obter uma estimativa. Por exemplo, sendo x a área de uma residência, a equação $\theta_0 + \theta_1 * x$ é uma reta no domínio $\mathbb{R}^2$
que pretende estimar o custo da residência a partir da área x.

Para encontrar quais coeficientes θ_i melhor se adequam ao modelo, utilizamos a medida de erros quadrados, definida por

$$\frac{1}{2n}\sum_{i=0}^{n}(h\theta(x) - y(x))^2. \quad (2)$$

Nossa intenção é minimizar esta função ao máximo, ajustando da melhor maneira possível a equação linear definida por $h\theta$ aos dados de treinamento. Definimos como $J(\theta_0, \theta_1, \ldots, \theta_n)$ a função que correlaciona o valor de cada θ_i ao erro quadrado obtido. Para encontrar os melhores valores de θ_i, precisamos encontrar o mínimo global da função $J(\theta_i)$. Podemos fazer isso utilizando a noção de derivada da função $J(\theta_i)$ da seguinte maneira:

$$\theta_0 = \theta_0 - \alpha\frac{1}{n}\sum_{i=0}^{n}(h\theta(x_i) - y(x_i)) \quad (3)$$

$$\theta_i = \theta_0 - \alpha\frac{1}{n}\sum_{i=0}^{n}(h\theta(x_i) - y(x_i)) * x_i \quad (4)$$

onde a variável αé chamada de taxa de aprendizado, e serve para acelerar o processo de procura pelo fundo da função $J(\theta_i)$. Este processo é repetido até que a função seja minimizada.

Neste trabalho, utilizamos modelos baseados em regressões lineares para predizer o ano de lançamento de uma música, utilizando como dados de entrada x um conjunto com 90 características:

- 12 médias de timbre, calculadas cada uma sobre um segmento composto de um vetor de timbres 12-dimensional,
- 78 covariâncias relacionadas a estas 12 classes de timbres, nos 12 segmentos.

II. Materiais e Métodos

Para realizar o desenvolvimento desta aplicação, foram utilizadas as seguintes tecnologias:

- Python 3:
  Escolhemos a linguagem python devido a sua simplicidade e facilidade de lidar com grandes quantidades de dados, e também devido ao fato que a biblioteca que escolhemos como base para nosso projeto ter sido escrita nesta linguagem. Utilizamos também o Jupyter Notebook, que nos permite adicionar anotações dinâmicas junto ao código e visualizar em tempo de execução os resultados obtidos, além de gráficos e reports.
- scikit-learn
  Esta biblioteca é um conjunto de ferramentas simples e eficientes para mineração de dados e exploração de dados. Ela conta com a implementação de um submódulo completo relacionado a regressão linear, permitindo que nos focássemos na exploração dos dados e na semântica do modelo ao invés de em detalhes de implementação.
- NumPy
  Esta biblioteca conta com uma miríade de subrotinas relacionadas a tratamento numérico de dados, facilitando o carregamento dos dados e a manipulação dos mesmos no escopo do programa.

III. Metodologia

Para executar os primeiros testes e montar uma estrutura básica de nossa aplicação, separamos um subconjunto composto de 500 dados retirados do conjunto de treinamento que nos foi oferecido, com 463.715 dados. Sobre este conjunto foram escolhidos aleatoriamente a cada iteração 100 dados como conjunto de validação e os 400 restantes foram utilizados como conjunto de treinamento.

Após implementados os modelos de regressão linear, executamos testes sobre os dados do subconjunto escolhido, visando encontrar um modelo que nos fosse mais satisfatório na tarefa de criar uma regressão linear para predizer o ano de lançamento de uma música a partir de suas características.

Com os testes executados, escolhemos um modelo que apresentou resultados mais satisfatórios e aplicamos o mesmo sobre o conjunto de 463.715 dados de treinamento oferecido, e após o treino testamos o regressor linear sobre o conjunto de testes oferecido, contendo 36.285 dados.

.

IV. Soluções Propostas

A primeira solução testada foi a de uma regressão linear simples. Neste modelo, a função $h\theta(x)$ é utilizada assim como descrita, e os coeficientes θ_i são ajustados de acordo com o algoritmo de gradiente descendente.

Após isso, implementamos uma solução baseada em uma regressão linear, onde inserimos um coeficiente de normalização ao processo de treinamento. Este coeficiente de normalização é incluído na equação da seguinte maneira:

$$\theta_i = \theta_0 - \alpha \frac{1}{n} \sum_{i=0}^{n} (h\theta(x_i) - y(x_i)) * x_i + \alpha * w, \quad (5)$$

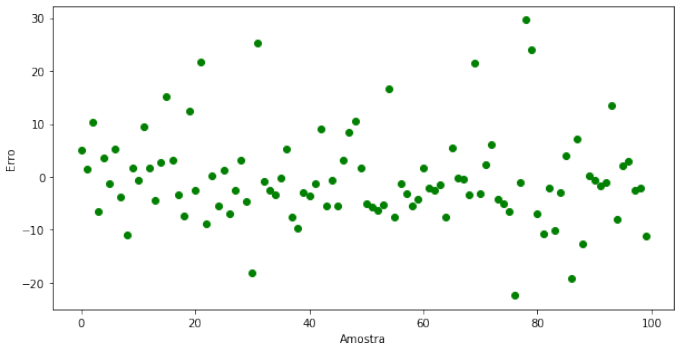
sendo w o coeficiente de normalização.

No último modelo proposto, semelhante ao anterior, setamos um número fixo de iterações no processo de descida do gradiente para a minimização de $J(\theta_i)$.

V. Experiments and Discussion

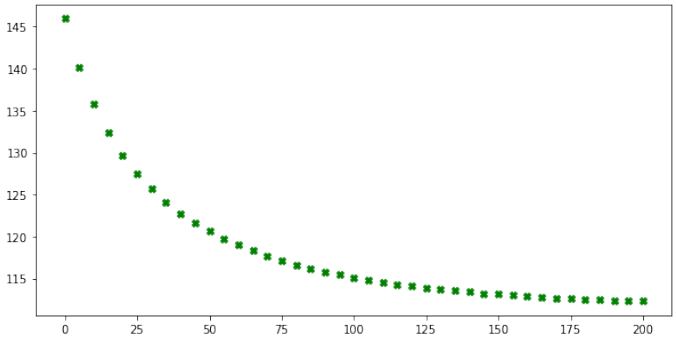
No primeiro modelo proposto, obtivemos um erro quadrático médio de 145.95. Ao analisar o gráfico dos erros de cada dado do conjunto, verificamos que a maioria dos dados se encontra com erro entre -10 e 10, com alguns com erro maior.

Figure 1. Scatterplot do erro de cada amostra



Após isso, implementamos o segundo modelo, utilizando um coeficiente de de normalização. Primeiro, aplicamos um coeficiente de 100, e verificamos que o erro médio quadrático diminuiu para $115,13$ o que representa uma redução em 30.83. Como verificamos que o coeficiente apresenta uma redução no erro, decidimos retreinar com varios coeficientes, variando de 0 a 200, de 5 a 5. Analisando o resultado, percebemos que ao aumentar o alfa, diminui o erro.

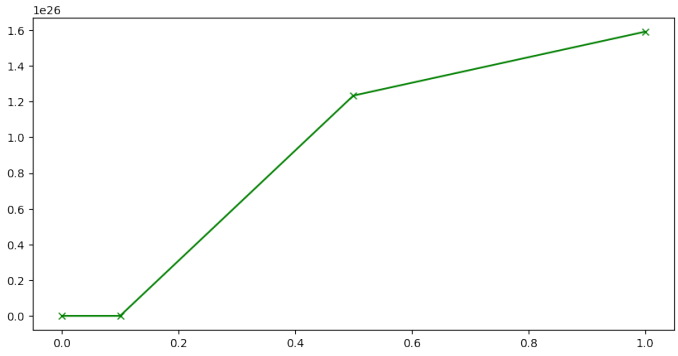
Figure 2. Scatterplot do erro para cada coeficiente utilizado



Decidimos então iterar sobre este coeficiente de 5 em 5, até que o erro se mantivesse ou aumentasse. Com isso, chegamos a um erro de 112,3 com um coeficiente igual a 250, o que se mostrou como uma redução de 2.83 em relação ao alfa igual a 100, e de 33.66 em relação ao primeiro modelo.

No terceiro modelo, para analisar os efeitos da escolha da taxa de aprendizado (learning rate), utilizaremos o método de descida de gradiente histocástica, com um número limitado de 100 iterações, e learning rates iguais a 0,00001 , 0,1 , 0,5 e 1,0. Para analisar o resultado, plotamos o erro para cada learning rate.

Figure 3. Plot do erro para cada learning rate

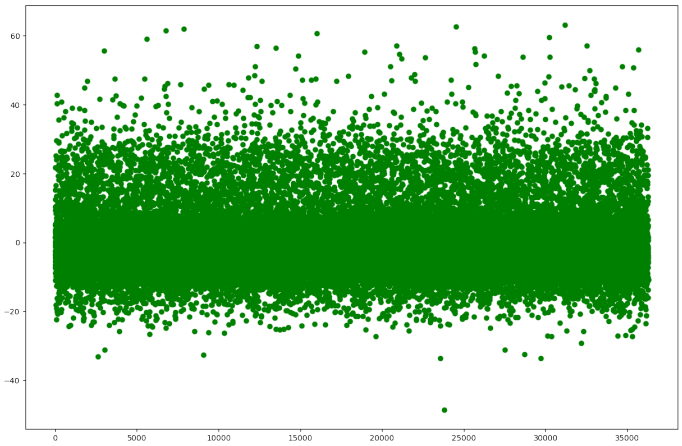


Plotando o erro para diferentes learning rates, observamos o resultado esperado: Para learning rates muito altos, a descida do gradiente diverge, gerando um erro quadrado maior. Para muito baixos, apesar de não tão gritante quanto para os valores mais altos, também observamos um pequeno aumento, pois devido ao número limitado de iterações, "Não dá tempo" de otimizar ainda mais o valor dos coeficientes.

Dos modelos que testamos, o que apresentou menor erro quadrado foi o Ridge, que é o método de regressão linear com pesos de penalidade nos coeficientes, com um alfa de 250. Este modelo que escolhemos como regressor e aplicaremos no conjunto de testes.

Utilizando o conjunto completo de dados, com um coeficiente de normalização igual ao ótimo igual a 250 obtido durante a validação, obtivemos o erro quadrático médio de 91.3162.

Figure 4. Scatterplot do erro de cada amostra do teste



VI. Conclusão

Devido a alta complexidade dos dados, que se relacionavam a musicas, e pelo fato que os dados oferecidos já são analises feitas acima dos dados originais, obtivemos um resultado não muito satisfatório.

Para melhores resultados, uma maior exploração dos dados deve ser feita, realizando uma análise na semântica das características de cada amostra, e adaptando um modelo mais complexo que uma simples regressão linear para realizar esta tarefa.

Verificamos que, apesar de apresentar erros um pouco grandes dependendo do dado analisado, a regressão linear que utilizamos poderia ser usada como estimador de década de lançamento de uma música, pois grande parte dos erros se manteve entre 20 e -20 no conjunto de testes.