UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR)
PROJETO REA

JOE MICHAEL HIDEYUKI FURUYA TAKAHASSI

RECURSOS DE INTERACAO EDUCACIONAL DIGITAL APLICADOS A FISICA, QUIMICA, A MATEMATICA E AS ENGENHARIAS USANDO O SCILAB E O PYTHON

PROGRAMA DE APOIO AO DESENVOLVIMENTO DE RECURSOS EDUCACIONAIS ABERTOS NA GRADUAÇÃO DA UTFPR (ÁREAS TRANSVERSAIS)

CURITIBA

2021

JOE MICHAEL HIDEYUKI FURUYA TAKAHASSI

RECURSOS DE INTERACAO EDUCACIONAL DIGITAL APLICADOS A FISICA, QUIMICA, A MATEMATICA E AS ENGENHARIAS USANDO O SCILAB E O PYTHON

Apostila elaborada para o projeto de Recursos Educacionais Abertos descritos no edital 26/2021 e ofertada pela UTFPR, apresentado aos avaliadores da banca como requisito para a conclusão do projeto.

Orientador: Prof. Dr. Antônio Carlos Amaro de Faria Júnior

CURITIBA

2021

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Esse material foi elaborado para o projeto ”Recursos de Interação Educacional Digital Aplicados à Física, Química, à Matemática e às Engenharias usando o Scilab e o Python”, ofertado pela UTFPR e pertencente ao Edital 26/2021. Para cada exemplo, foi exibido os respectivos códigos fontes assim como foi escrito uma explicação dos principais conceitos relacionados àquele tema.

1.2 OBJETIVOS

O objetivo do projeto é auxiliar o estudante no aprendizado de conceitos importantes tanto no campo da Física, Química, Matemática e Engenharias, através da elaboração de exemplos práticos no ambiente Jupyter Lab. Para isso, foi consultado referências bibliográficas para a elaboração da fundamentação teórica assim como foi utilizado algumas bibliotecas específicas do Python pertinentes para cada exemplo em questão.

2 EXEMPLOS

Nesta seção serão apresentados os exemplos práticos escolhidos para o projeto e feitos com o auxílio do Jupyter Lab, ambiente de desenvolvimento do Python.

2.1 REGRESSÃO LINEAR

2.1.1 O QUE É REGRESSÃO LINEAR?

Uma regressão linear nada mais é do que uma equação matemática utilizada para se estimar o valor de uma variável y, dados os valores de algumas outras variáveis x. Os modelos de regressão **simples** envolvem somente duas variáveis: uma independente x e uma dependente y. Ela é chamada **linear** poque a relação entre os parâmetros se dá por uma **função linear** do tipo:

$$F(x) = y = m \cdot x + c$$

Onde x é a variável independente, y é a variável dependente, m é o **coeficiente angular/declive** e c é o **coeficiente linear/intercepto**. O parâmetro c é o valor de $F(x)$ quando $x = 0$. O parâmetro m é a variação em $F(x)$ quando variamos x em 1 unidade.

O que pretendemos aqui é, a partir de um conjunto de dados (x, y), obter um modelo linear de função $F(x) = y$ que relacione de modo mais exato possível a relação entre as variáveis x e y. Para que possamos entender a regressão linear em sua integridade, duas concepções são bastante importantes: a de **Função de Custo** e **Gradiente Descendente**.

2.1.2 O QUE É FUNÇÃO DE CUSTO?

No contexto de otimização algorítmica, a função que queremos minimizar ou maximizar é denominada **função objetivo**. Quando queremos minimizá-la, ela é chamada **função de custo**. A função de custo computa a diferença entre o valor obtido em nosso modelo (y_{modelo}) e o valor real (y_{real}). **Entropia cruzada** e **Erro Quadrático Médio** são os tipos de funções de custo mais comuns. A fórmula para o Erro Quadrático Médio para n amostras de dados é:

$$E = \frac{1}{n}\sum_{i=0}^{n}(y_{i,real} - y_{i,modelo})^2 = \frac{1}{n}\sum_{i=0}^{n}(y_{i,real} - (m \cdot x_i + c))^2$$

Em sua essência, o conceito de função de custo é bastante simples: é um método para avaliar o quão bem o seu algoritmo modela o conjunto de dados em estudo. Se as predições estiverem totalmente erradas, a função de custo terá um valor maior. Se as predições estiverem corretas, a função de custo terá um valor menor. Na medida que mudados o nosso algoritmo, a função de custo irá nos dizer se estamos indo para o caminho correto.

2.1.3 GRADIENTE DESCENDENTE

Gradiente Descendente é um algoritmo de optimização usado para obter o mínimo de uma função diferenciável. Aqui, ela será utilizada para realizar o ajuste dos parâmetros m e c de forma iterativa com o objetivo de encontrar os valores para esses parâmetros que minimizem a função de custo o máximo possível. O Método do Gradiente se inicia atribuindo valores aleatórios para os parâmetros m e c, valores estes que irão melhorar gradualmente a cada iteração, dando um pequeno passo de cada vez até que a função de custo convirja para um mínimo. O tamanho dos passos é definido por um parâmetro denominado **taxa de aprendizado**.

Na aplicação do Método do Gradiente, utilizamos as derivadas parciais da função de custo em relação a m e em relação a c. O objetivo do método é, então, achar o mínimo global da função de custo a partir dessas derivadas parciais:

$$D_m = \frac{2}{n}\sum_{i=0}^{n}(y_{i,real} - (m \cdot x_i + c)) \cdot (-x_i) = \frac{-2}{n}\sum_{i=0}^{n}(y_{i,real} - y_{i,modelo})$$

$$D_c = \frac{-2}{n}\sum_{i=0}^{n}(y_{i,real} - y_{i,modelo})$$

2.1.4 REGRESSÃO LINEAR POR GRADIENTE DESCENDENTE NO PYTHON

Vamos modelar uma função $F(x)$ para um conjuntos de dados armazenados em um arquivo csv, utilizando o Erro Quadrático Médio e o Gradiente Descendente.

Primeiramente, vamos plotar o gráfico de dispersão das amostras para se ter uma ideia inicial da distribuição dos dados. Usaremos o conjunto de dados armazenados neste link: `https://github.com/Joe-Taka/REA/blob/main/C%C3%B3digos/Exemplo%201%20-%20Regress%C3%A3o%20Linear/data.csv`

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
# Definindo um tamanho para a figura
plt.figure(figsize=(7,6))
# Pegando os valores do arquivo 'csv'
data = pd.read_csv('data.csv')
# Pegando a 1° coluna
X = data.iloc[:, 0]
# Pegando a 2° coluna
Y = data.iloc[:, 1]
# Gráfico de dispersão para as amostras
plt.scatter(X, Y)
plt.show()
```

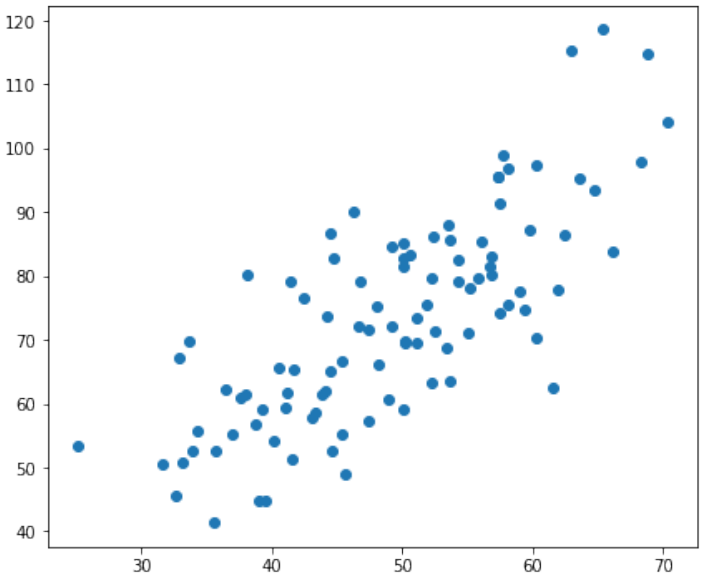

Figura 1: Gráfico de dispersão

Aplicando o **Gradiente Descendente**: começamos atribuindo valores nulos para os parâmetros m e c e iteramos as derivadas parciais da função de custo mil vezes, até encontrar os valores ótimos para os parâmetros:

```
# Suposições iniciais para os parâmetros
m = 0
c = 0
plt.figure(figsize=(7,6))
# Taxa de aprendizado
L = 0.0001
# Número de iterações para o Gradiente Descendente
epochs = 1000
n = float(len(X)) # Número de elementos de X
# Aplicando a iteração do Método Gradiente
for i in range(epochs):
        Y_pred = m*X + c  # Valor atual suposto para Y
        D_m = (-2/n) * sum(X * (Y - Y_pred))
        D_c = (-2/n) * sum(Y - Y_pred)
        m = m - L * D_m  # Atualizando m
        c = c - L * D_c  # Atualizando c
print (f"Valor para m: {m:.2f}; Valor para c: {c:.2f}")
# Valor para m: 1.48; Valor para c: 0.10
Y_pred = m*X + c
plt.scatter(X, Y)
plt.plot([min(X), max(X)], [min(Y_pred),
max(Y_pred)], color='red')
plt.show()
```

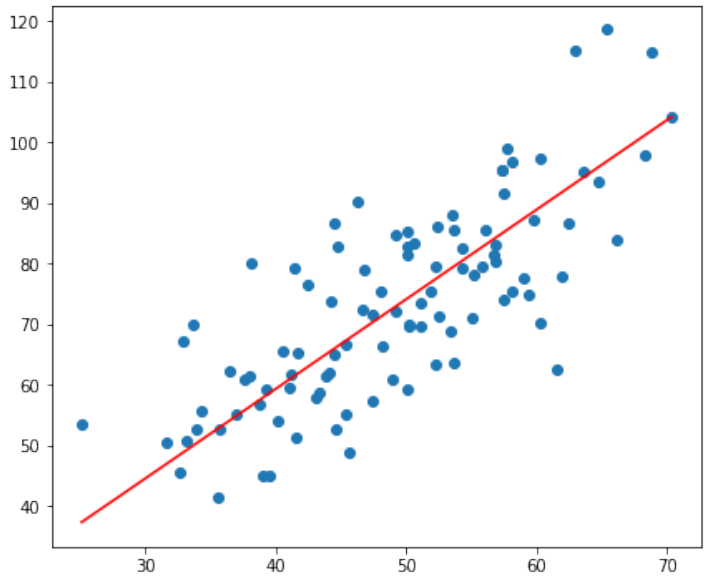

Figura 2: Regressão Linear

Para o conjunto de dados fornecidos em nosso exemplo, a função do nosso modelo fica, então:

$$F(x) = 1,48 \cdot x + 0,10$$

2.1.5 APLICAÇÕES DA REGRESSÃO LINEAR

- Aprendizado de máquina, como uma ferramenta de predição.
- Em modelo de negócios, pode ser utilizado para gerar insights à respeito do comportamento do consumidor, de modo a entender os fatores que possam influenciar na lucratividade do negócio. Por exemplo, se as vendas de uma empresa aumentaram constantemente todos os meses nos últimos anos, ao se realizar uma análise linear dos dados de vendas com as vendas mensais, a empresa é capaz de prever as vendas nos meses futuros.
- Química analítica, principalmente na calibração de dados para gerar as chamadas **curvas de calibração**.

- Nas ciências dos materiais, como uma forma de prever as propriedades de certos materiais.
- Em finanças, como uma ferramenta para traçar as curvas de tendências de ativos.

2.2 TEOREMA DE BAYES

2.2.1 O QUE É O TEOREMA DE BAYES?

O teorema de Bayes fornece um modo de calcular a probabilidade de uma hipótese baseado na probabilidade de outras hipóteses **a priori**.

2.2.2 FÓRMULA DO TEOREMA DE BAYES

$$P(A|B) = \frac{P(A) \cdot P(B|A)}{P(B)}$$

- $P(A|B)$: probabilidade do evento 'A' ocorrer dado que 'B' ocorreu
- $P(A)$: probabilidade de 'A' ocorrer
- $P(B|A)$: probabilidade do evento 'B' ocorrer dado que 'A' ocorreu
- $P(B)$: probabilidade de 'B' ocorrer

Portanto, o Teorema de Bayes afirma que a probabilidade posterior do evento A (ou seja, a probabilidade do evento A dado que o evento B ocorreu) é igual à probabilidade contrária P(B | A) multiplicada pela probabilidade de A e dividido pela probabilidade de B.

2.2.3 TEOREMA DE BAYES UTILIZANDO SCIKIT-LEARN NO PYTHON

Vamos utilizar um conjunto de dados que incluem 4 observações sobre um determinado momento do dia (clima - claro, chuvoso ou geada; feriado ou dia útil; período - manhã, horário de almoço ou noite; houve ou não congestionamento) e, a partir desses dados base, tentaremos prever a probabilidade de acontecer um congestionamento utilizando valores

fornecidos apenas das 3 condições iniciais. Para isso, usaremos a biblioteca **Scikit-Learn** do Python.

```
def getTempo():
return ['Claro', 'Claro', 'Claro', 'Claro', 'Claro',
'Claro', 'Chuvoso', 'Chuvoso', 'Chuvoso', 'Chuvoso',
'Chuvoso', 'Chuvoso', 'Geada', 'Geada', 'Geada', 'Geada',
'Geada', 'Geada']

def getDiaSem():
return ['Útil', 'Útil', 'Útil',
'Feriado', 'Feriado', 'Feriado',
'Útil', 'Útil', 'Útil',
'Feriado', 'Feriado', 'Feriado',
'Útil', 'Útil', 'Útil',
'Feriado', 'Feriado', 'Feriado']

def getHorario():
return ['Manhã', 'Almoço', 'Noite',
'Manhã', 'Almoço', 'Noite',
'Manhã', 'Almoço', 'Noite',
'Manhã', 'Almoço', 'Noite',
'Manhã', 'Almoço', 'Noite',
'Manhã', 'Almoço', 'Noite']

def getCongest():
return ['Sim', 'Não', 'Sim',
'Não', 'Não', 'Não',
'Sim', 'Sim', 'Sim',
'Não', 'Não', 'Não',
'Sim', 'Sim', 'Sim',
'Sim', 'Não', 'Sim']

Tempo = getTempo()
diaDaSem = getDiaSem()
Horario = getHorario()
Congest = getCongest()

pd.DataFrame(zip(Tempo,diaDaSem,Horario,Congest),
columns=['Tempo','Dia da semana',
'Horário','Congestionamento'])
```

	Tempo	Dia da semana	Horário	Congestionamento
0	Claro	Útil	Manhã	Sim
1	Claro	Útil	Almoço	Não
2	Claro	Útil	Noite	Sim
3	Claro	Feriado	Manhã	Não
4	Claro	Feriado	Almoço	Não
5	Claro	Feriado	Noite	Não
6	Chuvoso	Útil	Manhã	Sim
7	Chuvoso	Útil	Almoço	Sim
8	Chuvoso	Útil	Noite	Sim
9	Chuvoso	Feriado	Manhã	Não
10	Chuvoso	Feriado	Almoço	Não
11	Chuvoso	Feriado	Noite	Não
12	Geada	Útil	Manhã	Sim
13	Geada	Útil	Almoço	Sim
14	Geada	Útil	Noite	Sim
15	Geada	Feriado	Manhã	Sim
16	Geada	Feriado	Almoço	Não
17	Geada	Feriado	Noite	Sim

Tabela 1: Condições do dia

2.2.3.1 PRÉ-PROCESSAMENTO DOS DADOS

Vamos utilizar o comando 'sklearn.preprocessing.LabelEncoder' para **normalizar** os dados, ou seja, alterar os valores das colunas numéricas no conjunto de dados para uma escala comum, sem distorcer os intervalos de valores ou provocar perda de informações. No exemplo analisado, os valores do tipo texto serão transformados em valores numéricos. Ainda, usamos o método **fit_transform()** para evitar possíveis erros que poderiam ser provocados por valores nulos, inexistentes ou indefinidos no nosso conjunto de dados. Por fim, organizamos os dados relevantes (**features**) para o nosso exemplo em uma única lista.

```
labelEncoder = preprocessing.LabelEncoder()
# Normalizando e aplicando o 'fit_transform'
normTempo = labelEncoder.fit_transform(Tempo)
normDiaDaSem = labelEncoder.fit_transform(diaDaSem)
normHorario = labelEncoder.fit_transform(Horario)
normCongest = labelEncoder.fit_transform(Congest)
print("-----------------normalização-----------------")
print(normTempo)
print(normDiaDaSem)
print(normHorario)
print(normCongest)
# Organizando os dados relevantes (features)
features = []
for i in range(len(normTempo)):
        features.append([normTempo[i], normDiaDaSem[i],
        normHorario[i]])
print("-----------------features-----------------")
print(features)
# Aplicando o Método de Bayes
modelo = GaussianNB()
# Treinando o modelo
modelo.fit(features, normCongest)
```

2.2.3.2 FAZENDO AS PREDIÇÕES

```
# ["Geada", "Útil", "Manhã"]
print(modelo.predict([[2, 1, 1]]))
# resultado = [1]
# ["Claro", "Feriado", "Almoço"]
print (modelo.predict([[1, 0, 0]]))
# resultado = [0]
```

Portanto, para as condiçoes [”Geada”, ”Útil”, ”Manhã”] por exemplo, há uma grande probabilidade de ocorrer congestionamento. Para as condiçoes [”Claro”, ”Feriado”, ”Almoço”], a probabilidade de ocorrer um congestionamento é baixa.

2.2.4 APLICAÇÕES DO TEOREMA DE BAYES

- Aprendizado de máquinas, principalmente nas classificações de dados.
- Pode ser utilizado na química para avaliar a densidade de probabilidade da composição química de um sistema em termos das densidades de probabilidade das abundâncias das diferentes espécies químicas.
- Pode ser utilizado para prever a qualidade de água com base na concentração de toxinas e outros poluentes que possam ultrapassamr os limites numéricos do padrão de qualidade da água.
- Em engenharia estrutural, auxiliando o engenheiro a determinar se a estrutura é segura ou não, com base nos valores de reabilidade.

2.3 TRANSFORMADA DE FOURIER

2.3.1 O QUE É A TRANSFORMADA DE FOURIER?

A transformada de Fourier é uma transformação matemática que ”quebra”uma forma de onda (função ou sinal) em uma soma de funções periódicas. A análise de Fourier converte um sinal do seu domínio original para uma representação no domínio da frequência.

Jean-Baptiste Joseph Fourier (Auxerre, 21 de março de 1768 — Paris, 16 de maio de 1830) foi o matemático e físico francês que descobriu que todo sinal pode ser descrito como uma superposição de senóides complexas.

2.3.2 SINAIS

Os sinais traduzem a evolução de uma grandeza ao longo do tempo ou espaço e podem ser classificados segundo os critérios:

- Continuidade: sinais contínuos ou discretos
- Periodicidade: sinais periódicos ou aperiódicos

2.3.2.1 SINAIS CONTÍNUOS

Um sinal diz-se contínuo se o seu domínio for IR ou um intervalo contínuo de IR. Assim:

$$x : IR \rightarrow IR$$

$$x : [a,b] \rightarrow IR$$

- Sinais contínuos são sinais que não possuem espaços distinguíveis entre os seus valores. - Advém de grandezas que podem ser medidas.

2.3.2.2 SINAIS DISCRETOS

- Sinais discretos são sinais que possuem espaços entre os seus valores.
- Advém de grandezas que são contadas.
- Só os sinais discretos podem ser armazenados e processados em computadores digitais.
- Pode-se converter um sinal contínuo num sinal discreto através da coleção de amostras do sinal contínuo.

2.3.2.3 SINAIS PERIÓDICOS

Um sinal discreto x(n) é periódico com período $N \in IN$ se:

$$x(n+N) = x(n), \forall\ n\ \in Z$$

2.3.3 TIPOS DE ABORDAGENS

De acordo com o tipo de sinal que estamos lidando, existem 4 diferentes tipos de abordagem:

- Sinais contínuos e aperiódicos: Transformada de Fourier
- Sinais contínuos e periódicos: Série de Fourier
- Sinais discretos e aperiódicos: Transformada de Fourier de Tempo Discreto (TFTD)
- Sinais discretos e periódicos: Transformada Discreta de Fourier (TDF)

3 CRONOGRAMA E CUSTOS DO PROJETO

3.1 CRONOGRAMA

* Apresentar o cronograma proposto e o final (lista e Diagrama de Gantt).

3.2 CUSTOS

* Apresentar o custo do projeto (tabela)

4 CONCLUSÕES

4.1 CONCLUSÕES

4.2 TRABALHOS FUTUROS