

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA DE CIÊNCIAS EXATAS E DA COMPUTAÇÃO
CIÊNCIAS DA COMPUTAÇÃO
INTELIGÊNCIA ARTIFICIAL
Professor CLARIMAR JOSE COELHO

Aluno **MARCOS RODOLFO CRUVINEL GOULART QUERINO**

ATIVIDADE EXTRA DISCIPLINAR II

Escrever um programa em Python para implementar a regressão linear múltipla usando os dados de sêmen. Devem ser calculados: R^2 , R^2 ajustado, SQT, SQR e SQE.

GOIÂNIA

2020

1. Desenvolvimento

Primeiramente, os dados de teste repassados pelo professor foram transcritos em um arquivo 'semen.csv' (em anexo). E em seguida, foram lidos pelo código (em anexo):

```
# Importação das bibliotecas
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
import statsmodels.formula.api as sm

# Carregar dados do arquivo .csv para variavel 'base'
base= pd.read_csv('semen.csv') # Usamos a biblioteca 'pandas' para ler
                                # esse arquivo

base.shape # 'shape' da forma a esse arquivo, uma vez que seus dados estão
            # numa variavel (array)

# Regressao linear multipla
# 4 variaveis independentes: x1, x2, x3 e x4
x= base.iloc[:,0:4].values #[:,1:4]= pega todas as linhas e as colunas de
                            # 0 a 3 (nesse 'range' o ultimo valor é ignorado,
                            # no caso, o '4')
y= base.iloc[:,4].values # Passa a coluna 'y' para a y

# Criação do modelo, treinamento, visualização dos coeficientes e do score
modelo= LinearRegression()
modelo.fit(x,y)

# Interceptação (aonde os dados encontram o eixo 'y')
print('\nInterceptação (aonde os dados encontram o eixo Y)\n')
```

```

print(modelo.intercept_)

# Inclinação (ângulo da reta)
print("\nInclinação (ângulo da reta)\n")
print(modelo.coef_)

# Score R2 (coeficiente de determinação, mostra o quanto o modelo consegue
# explicar os valores obtidos)
modelo.score(x,y)

# Geração de previsões
previsoes= modelo.predict(x)
print('\nGerando previsões:\n',previsoes)

# Criação do modelo usando a biblioteca statsmodel, para obter R2 ajustado
modelo_ajustado= sm.ols(formula= 'y ~ x1 + x2 + x3 + x4', data= base)
modelo_experiente= modelo_ajustado.fit() # O metodo 'fit' treina o modelo
print('\nRegressao linear multipla\n')
print(modelo_experiente.summary())

# Visualização dos resultados com grafico de dispersão
# As linhas abaixo estão marcadas como comentário porque não fui capaz de
# desenvolver o gráfico de dispersão para regressão linear multipla
# Erro: "ValueError: x and y must be the same size"
# plt.scatter(x,y)
# plt.plot(x, previsoes, color= 'red')

```

2. Resultados

Dados impressos no console:

Interceptação (aonde os dados encontram o eixo Y)

39.15734995497774

Inclinação (ângulo da reta)

[0. 1.01610044 -1.8616492 -0.34326049]

Gerando previsões:

[27.35141063 32.26231743 27.34954936 38.3095766 15.5447301 26.10806578
28.2531601 26.22185321 32.08818399 26.06763631 37.25235515 32.48791746
28.20324389]

Imagem 1.

Regressao linear multipla					
OLS Regression Results					
=====					
Dep. Variable:	y	R-squared:	0.912		
Model:	OLS	Adj. R-squared:	0.882		
Method:	Least Squares	F-statistic:	30.98		
Date:	Thu, 01 Oct 2020	Prob (F-statistic):	4.50e-05		
Time:	16:32:51	Log-Likelihood:	-25.533		
No. Observations:	13	AIC:	59.07		
Df Residuals:	9	BIC:	61.33		
Df Model:	3				
Covariance Type:	nonrobust				
=====					
	coef	std err	t	P> t	[0.025 0.975]

Intercept	19.5787	2.944	6.651	0.000	12.920 26.237
x1	19.5787	2.944	6.651	0.000	12.920 26.237
x2	1.0161	0.191	5.323	0.000	0.584 1.448
x3	-1.8616	0.267	-6.964	0.000	-2.466 -1.257
x4	-0.3433	0.617	-0.556	0.592	-1.739 1.053
=====					
Omnibus:	2.087	Durbin-Watson:	1.568		
Prob(Omnibus):	0.352	Jarque-Bera (JB):	1.548		
Skew:	0.730	Prob(JB):	0.461		
Kurtosis:	2.148	Cond. No.	6.99e+16		
=====					

3. Dificuldades

Não foi possível calcular SQT, SQR e SQE. Mesmo se esses dados estivessem na imagem 1 (acima), não saberia localizá-los. Estudei e fiz inúmeras pesquisas mas não consegui encontrar nenhuma implementação em Python desses dados, e não fui capaz de entender sua explicação matemática, para então implementar por conta própria.

Também não foi possível criar o gráfico de dispersão. Fiz um exemplo funcional mas com regressão linear simples. Mas para a múltipla, ocorre erro, e não fui capaz de resolvê-lo.

4. Referências bibliográficas

Curso de formação de cientista de dados com Python. Disponível em <https://www.udemy.com>. Acesso em 30 de setembro de 2020.

Nascimento, João Paulo Cândido. OS DESAFIOS EM LIDAR COM DADOS PROBLEMÁTICOS: UM ESTUDO EM CIÊNCIA DE DADOS SOBRE A DENGUEEM BRASÍLIA / DF. 2019. Trabalho de conclusão de curso. Faculdade de Engenharia elétrica – FEELT. Uberlândia, Minas Gerais. 2019. Disponível em: <https://repositorio.ufu.br/bitstream/123456789/26678/2/DesafiosLidarDados.pdf>. Acesso em 30 de setembro de 2020.

Regressão Linear Múltipla em Python. Disponível em <http://artedossdados.blogspot.com/2013/09/regressao-linear-multipla-em-python.html>. Acesso em 30 de setembro de 2020.

Entenda o que é o formato CSV e saiba como importar e exportar esses arquivos. Disponível em <https://rockcontent.com/br/blog/csv/>. Acesso em 01 de outubro de 2020.

[#29] Salvar arquivo CSV no Python. Disponível em: <https://www.youtube.com/watch?v=Ry8nhTMH7uo>. Acesso em 01 de outubro de 2020.

Implementando Regressão Linear Simples em Python. Disponível em <https://medium.com/data-hackers/implementando-regress%C3%A3o-linear-simples-em-python-91df53b920a8>. Acesso em 01 de outubro de 2020.

Modelo de Regressão Linear — Mínimos Quadrados Ordinários. Disponível em <https://medium.com/@caderno/regress%C3%A3o-linear-parte-1-330faf2f0010>. Acesso em 01 de outubro de 2020.