regressao-copy

December 11, 2023

Previsão de Salário (Regressão Linear, SVM e MLP)

10/12/2023

Brenno de Oliveira da Rosa - 2021029935 Eduardo Alves Carvalho - 2021017550 Lucas Luan Belarmino Barbosa - 2021017872

[44]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import cross_val_score, train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVR
from sklearn.neural_network import MLPRegressor
```

No quadro acima são realizados os imports da maioria das bibliotecas que serão utilizadas ao decorrer do arquivo. Já o quadro abaixo realiza a leitura dos dados do arquivo, onde buscamos demonstrar o crescimento do salario de funcionarios de uma empresa ao passar dos anos.

[45]:
```
# Exemplo de carregamento e divisão de dados
data = pd.read_csv('salario/Salary Data.csv')
#Colocar x como bebida em dia util e tipo de moradia
# data.info()
data.head()
```

[45]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0

Aqui checa-se se há alguma celula vazia entre os dados, para que seja retirada. Como não há nenhuma, não é necessário realizar o processso de retirada da mesma.

[46]:
```
data.isnull().sum()
```

[46]:
```
YearsExperience    0
Salary             0
dtype: int64
```

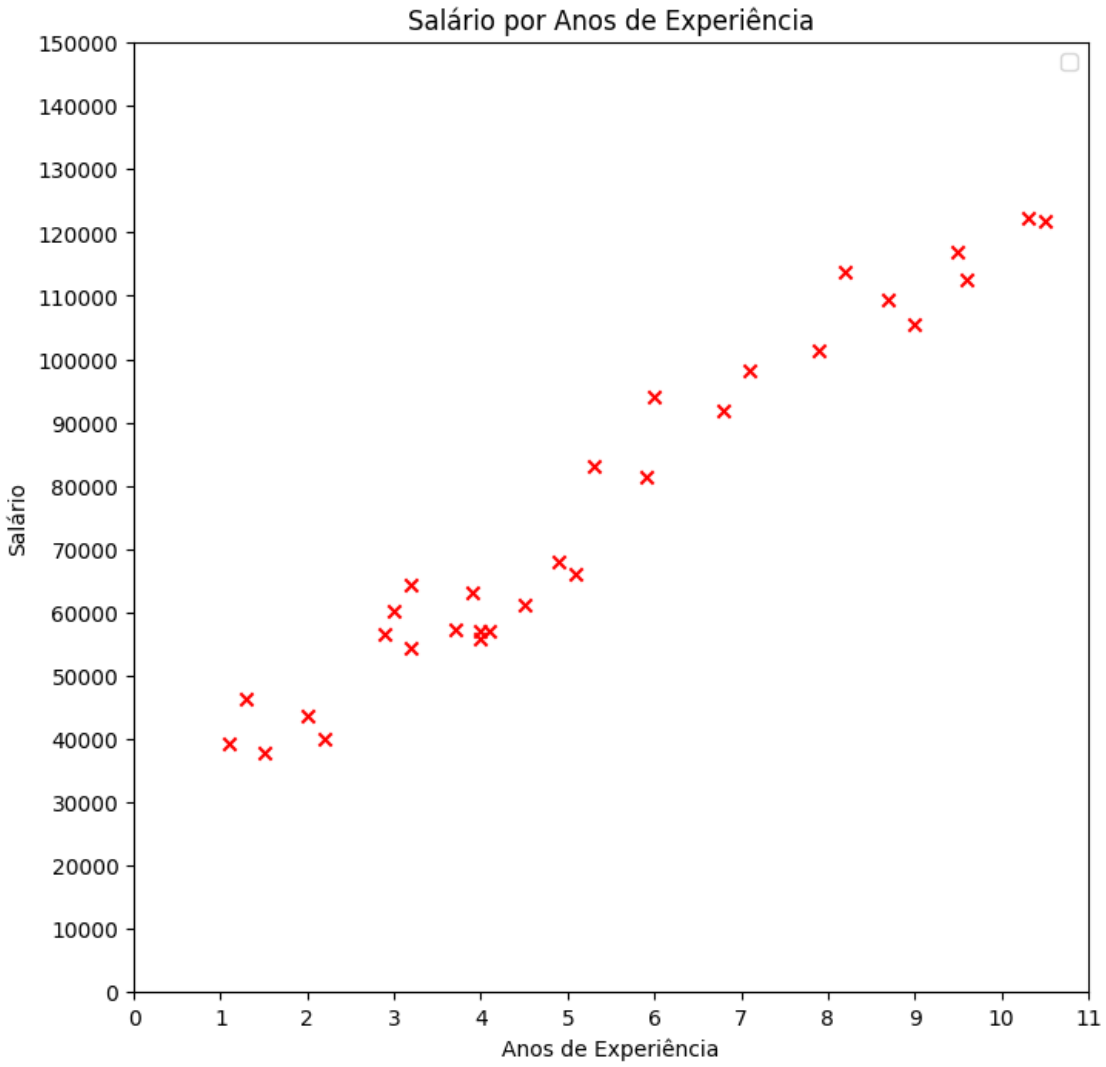
Por meio do código a seguir, plotamos um gráfico para a melhor vizualização dos dados contidos no arquivo.

[47]:
```
# Criando um gráfico de linhas
plt.figure(figsize=(8, 8))
plt.scatter(data['YearsExperience'], data['Salary'], marker='x', color='red')
# Adicionando . para os valores do eixo y
plt.yticks(np.arange(0, 160000, 10000))
plt.xticks(np.arange(0, 12, 1))

# Adicionando rótulos e título
plt.xlabel('Anos de Experiência')
plt.ylabel('Salário')
plt.title('Salário por Anos de Experiência')
plt.legend()

# Exibindo o gráfico
plt.show()
```

```
No artists with labels found to put in legend.  Note that artists whose label
start with an underscore are ignored when legend() is called with no argument.
```



Aqui, criamos e testamos o modelo da Regressão Logistica.

[48]:
```
from sklearn.model_selection import train_test_split
x_val = data['YearsExperience'].values.reshape(-1,1)
y_val = data['Salary']
x_train, x_test, y_train, y_test = train_test_split(x_val, y_val, test_size=0.
  ↪3, random_state=42)
from sklearn.linear_model import LinearRegression
# regressão (treinamento)
my_model = LinearRegression()
my_model.fit(x_train, y_train)
print("R_squared =", my_model.score(x_test, y_test)) # r2 squared
#regressão (teste)
test_pred = my_model.predict(x_test)
```

```
R_squared = 0.9414466227178214
```

[49]:
```
plt.plot(x_test, y_test, '+', label='Valores reais')
plt.plot(x_test, test_pred, '-', label='Valores previstos')

plt.xlabel('Anos de Experiência')
plt.ylabel('Salário')
plt.title('Salário por Anos de Experiência')
plt.legend()
plt.show()
```

[50]:
```
print(my_model.intercept_, my_model.coef_, my_model.score(x_test, y_test))
```

```
25918.438334893202 [9339.08172382] 0.9414466227178214
```

Com isso a equação obtida é y = 9339.08172382x + 25918.438334893202

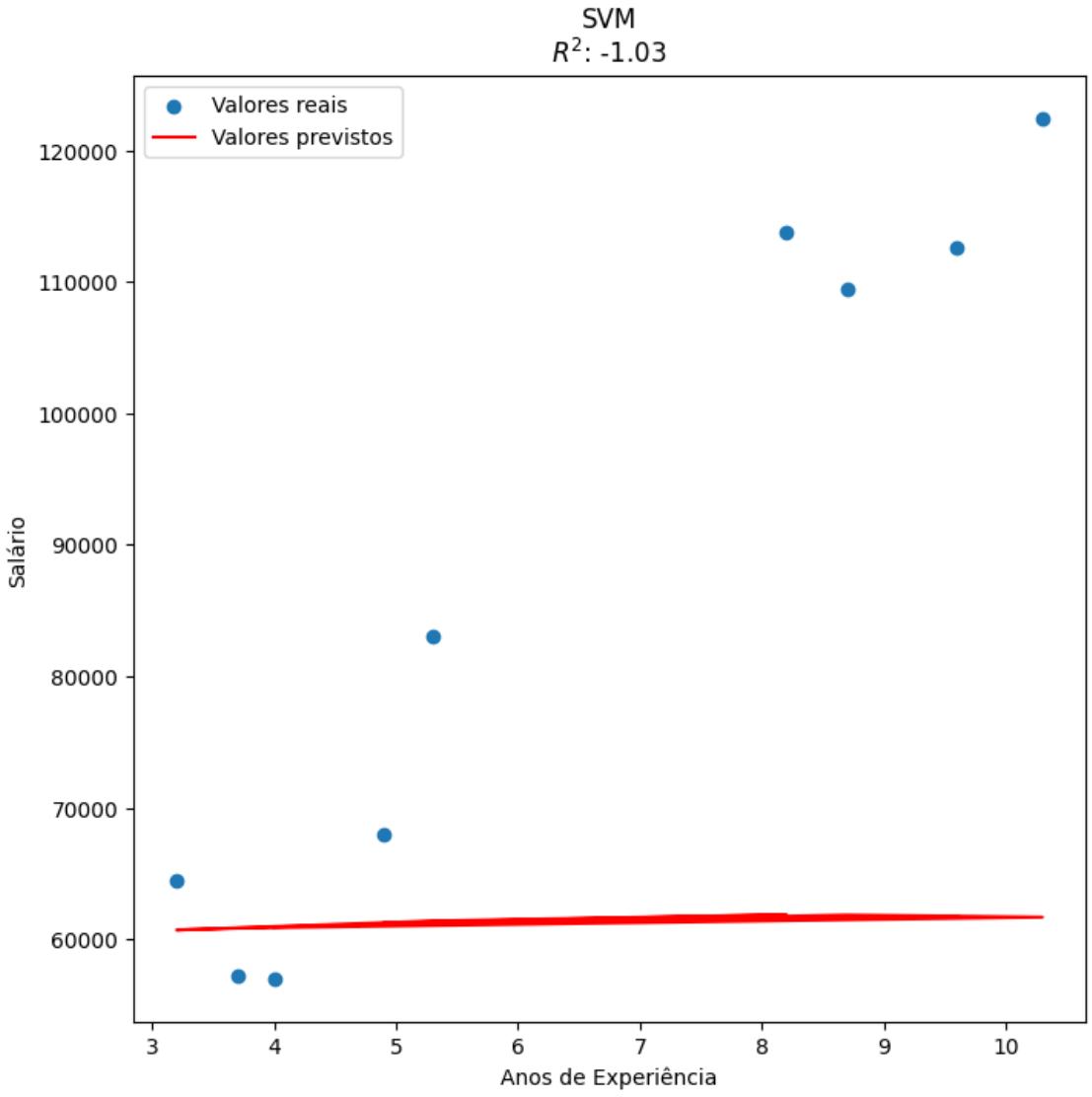
Com o código abaixo, criamos e testamos o modelo do SVM.

[52]:
```
# SVM
svr = SVR(kernel='rbf', C=100, gamma=0.1, epsilon=0.1)
svr.fit(x_train, y_train)
```

```
svr_r2 = svr.score(x_test, y_test)

# Plotando gráfico para SVM
plt.figure(figsize=(8, 8))
plt.scatter(x_test, y_test, label='Valores reais')
plt.plot(x_test, svr.predict(x_test), label='Valores previstos', color='red')
plt.title(f'SVM\n$R^2$: {svr_r2:.2f}')
plt.xlabel('Anos de Experiência')
plt.ylabel('Salário')
plt.legend()
```

[52]: <matplotlib.legend.Legend at 0x1a75e922110>



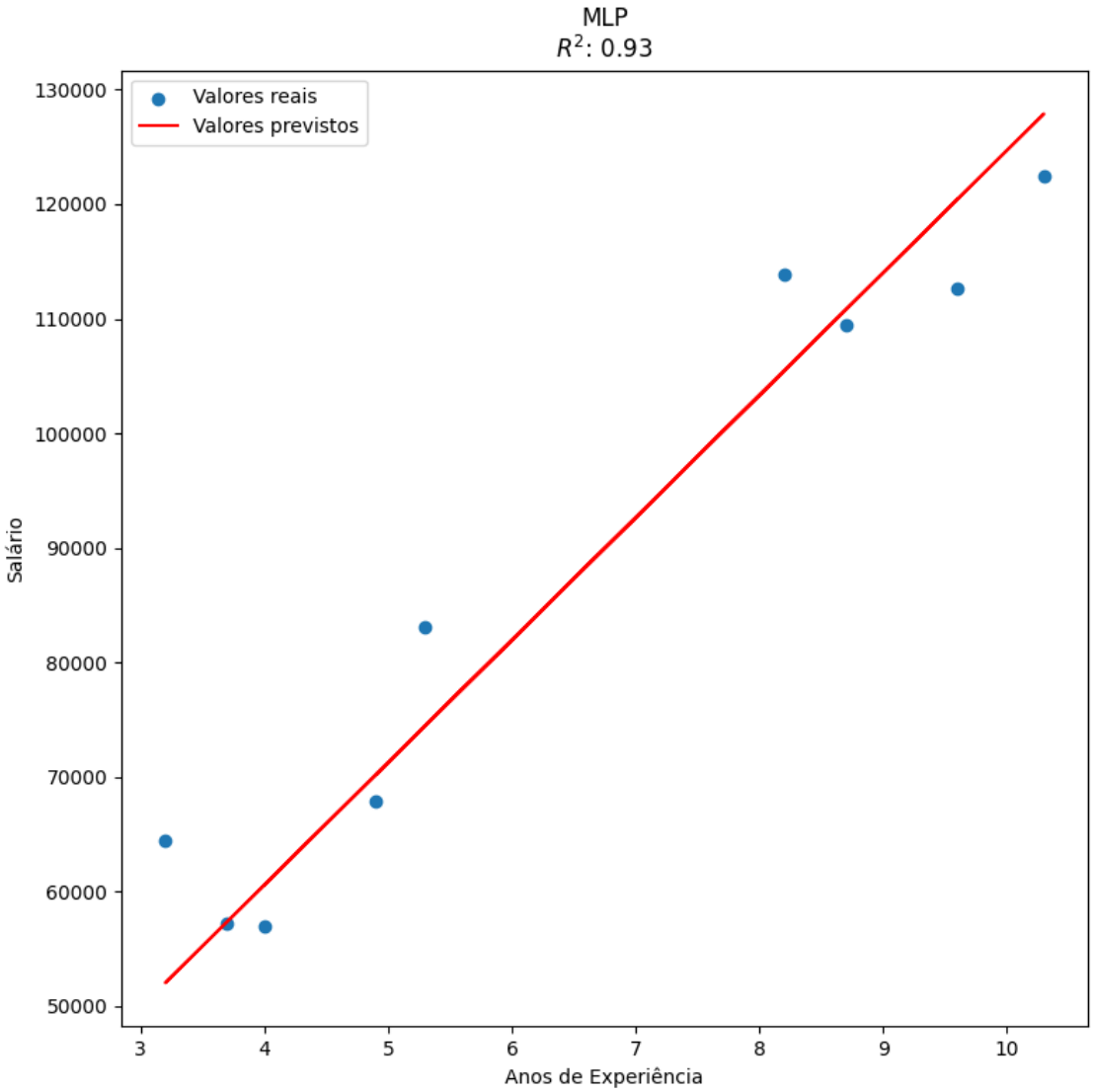
Já aqui, criamos e testamos o modelo da Regressão Logistica.

[54]:
```
# MLP
mlp = MLPRegressor(hidden_layer_sizes=(100, 100, 100), max_iter=1000,␣
 ↪activation='relu', solver='adam', random_state=42)
mlp.fit(x_train, y_train)
mlp_r2 = mlp.score(x_test, y_test)

# Plotando gráfico para MLP
plt.figure(figsize=(8, 8))
plt.scatter(x_test, y_test, label='Valores reais')
plt.plot(x_test, mlp.predict(x_test), label='Valores previstos', color='red')
plt.title(f'MLP\n$R^2$: {mlp_r2:.2f}')
plt.xlabel('Anos de Experiência')
plt.ylabel('Salário')
plt.legend()

plt.tight_layout()
plt.show()
```

```
C:\Users\Usuario\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz
5n2kfra8p0\LocalCache\local-packages\Python311\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (1000) reached and
the optimization hasn't converged yet.
  warnings.warn(
```


MLP
R^2: 0.93
Valores reais
Valores previstos
Salário
Anos de Experiência
130000
120000
110000
100000
90000
80000
70000
60000
50000
3
4
5
6
7
8
9
10