Classificação - Segundo Trabalho IA - 2023.2

Para a classificação com algoritmo de regressão logística, foi utilizada a seguinte base de dados: Heart Attack Analysis

Segue o código com os comentários nas células anteriores:

Importando as bibliotecas e funções.

In [ ]:
```
import piplite
await piplite.install('seaborn')
```

In [ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report
from sklearn.neural_network import MLPClassifier
from sklearn import metrics
```

Base de dados de análise de ataques cardíacos

Esse dataset tem informações sobre pacientes, com o objetivo de prever se terão ataques cardíacos ou não.

In [ ]:
```
heart = pd.read_csv('data/heart.csv')
heart.head()
```

Out[ ]:

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2

O conjunto de dados é baseado em aspectos como idade, sexo, pressão arterial, índice de colesterol no sangue, entre outros.

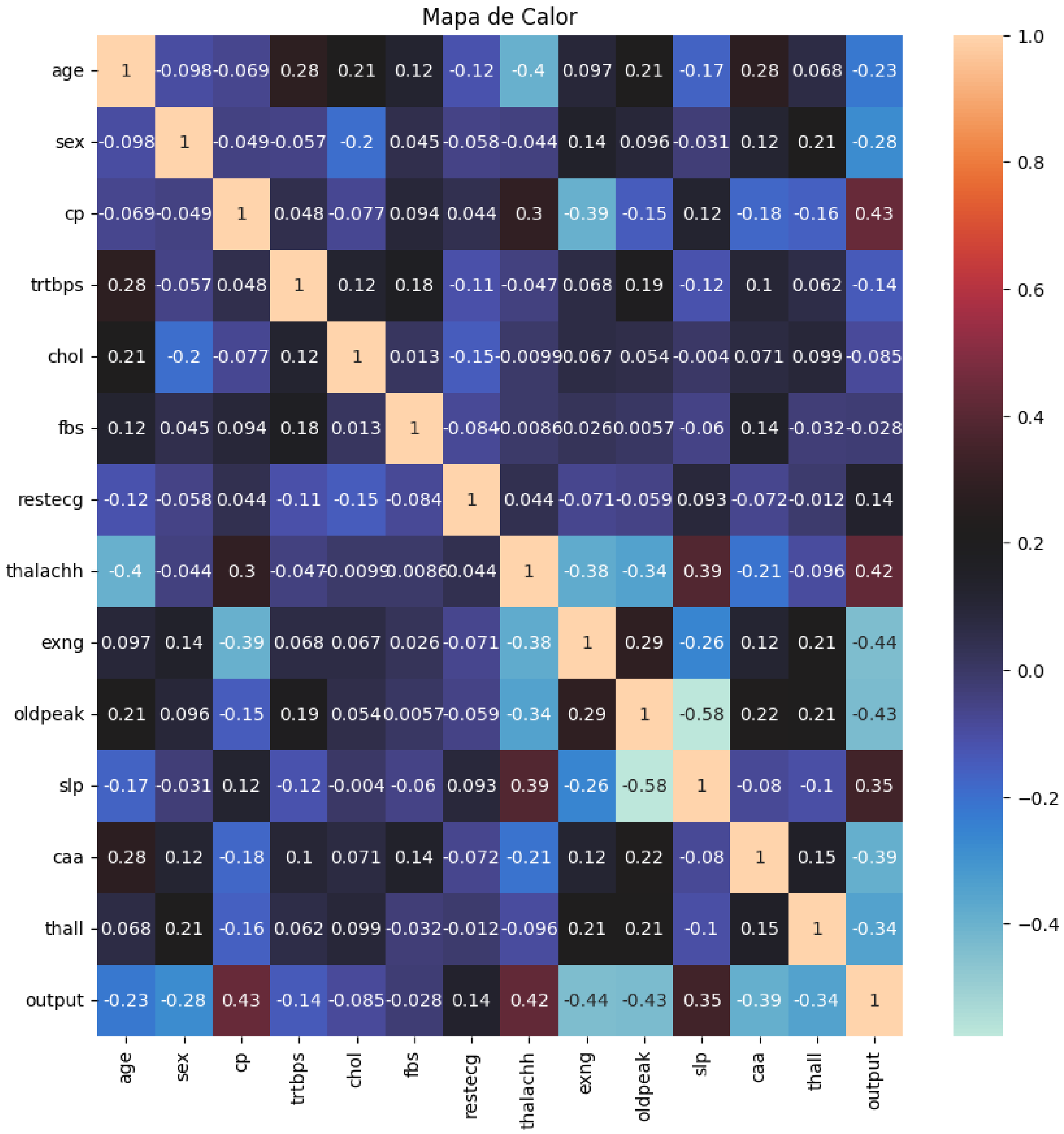
In [ ]:
```
heart.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       303 non-null    int64  
 1   sex       303 non-null    int64  
 2   cp        303 non-null    int64  
 3   trtbps    303 non-null    int64  
 4   chol      303 non-null    int64  
 5   fbs       303 non-null    int64  
 6   restecg   303 non-null    int64  
 7   thalachh  303 non-null    int64  
 8   exng      303 non-null    int64  
 9   oldpeak   303 non-null    float64
 10  slp       303 non-null    int64  
 11  caa       303 non-null    int64  
 12  thall     303 non-null    int64  
 13  output    303 non-null    int64  
dtypes: float64(1), int64(13)
memory usage: 33.2 KB
```

Heatmap

O mapa térmico (heatmap) de correlação é utilizado para verificar a força das relações entre as variáveis no conjunto de dados.

In [ ]:
```
plt.figure(figsize=(10, 10))
sns.heatmap(heart.corr(), annot=True, cmap='icefire').set_title('Mapa de Calor')
plt.show()
```



Para obter-se uma melhor observação dos valores em relação ao resultado (coluna 'output'), os resultados da correlação são elencados abaixo.

```
In [ ]: correlacoes = heart.corr().loc[:, 'output'].drop('output')
        maiores_correlacoes = correlacoes.nlargest(10)
        print(f"Os maiores valores de correlação com o resultado:")
        print(maiores_correlacoes)
```

```
Os maiores valores de correlação com o resultado:
cp          0.433798
thalachh    0.421741
slp         0.345877
restecg     0.137230
fbs        -0.028046
chol       -0.085239
trtbps     -0.144931
age        -0.225439
sex        -0.280937
thall      -0.344029
Name: output, dtype: float64
```

Os dados são dividos entre treino e teste.

```
In [ ]: X = heart.drop('output', axis = 1)
        y = heart['output']

        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, randc
```

Em seguida, os dados são normalizados.

```
In [ ]: scaler = StandardScaler()
        X_train = scaler.fit_transform(X_train)
        X_test = scaler.transform(X_test)
```

Regressão Logística

É realizada a chamada da função de Regressão Logística. O modelo é treinado e testado.

```
In [ ]: lr = LogisticRegression(max_iter = 1000)
        lr.fit(X_train, y_train)
        y_pred = lr.predict(X_test)
        accuracy = accuracy_score(y_test, y_pred)
        print(f"Acurácia da Regressão Logística: {accuracy:.4f}")
        print("Classification Report:")
        print(classification_report(y_test, y_pred))
```

```
Acurácia da Regressão Logística: 0.8132
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.78      0.79        41
           1       0.82      0.84      0.83        50

    accuracy                           0.81        91
   macro avg       0.81      0.81      0.81        91
weighted avg       0.81      0.81      0.81        91
```

SVM

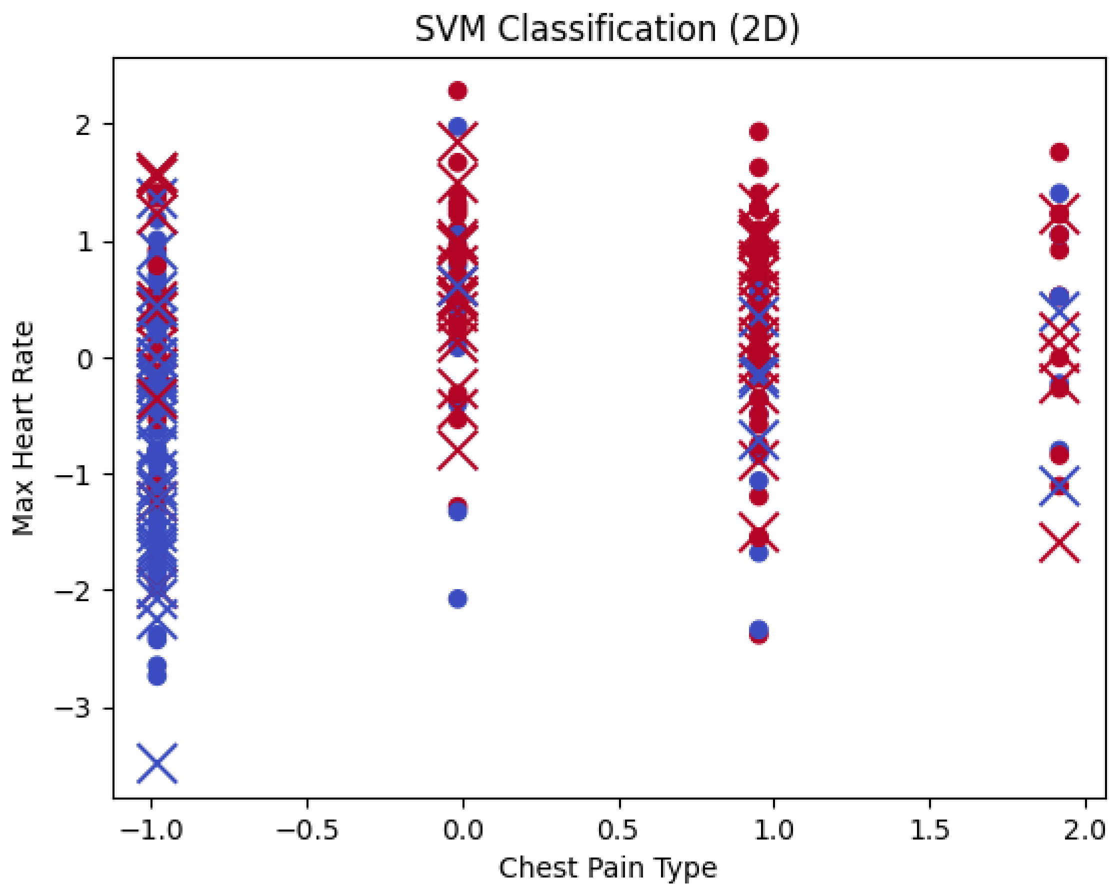
É realizada a chamada da função de Máquina de Vetor de Suporte. O modelo é treinado e testado.

```
In [ ]: svm_classifier = SVC(kernel = 'rbf', C = 1, gamma = 'scale', random_state = 42)
        svm_classifier.fit(X_train, y_train)
        y_pred = svm_classifier.predict(X_test)
        accuracy = accuracy_score(y_test, y_pred)
        print(f"Acurácia do método SVM: {accuracy:.4f}")
        print("Classification Report:")
        print(classification_report(y_test, y_pred))
```

```
Acurácia do método SVM: 0.8242
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.80      0.80        41
           1       0.84      0.84      0.84        50

    accuracy                           0.82        91
   macro avg       0.82      0.82      0.82        91
weighted avg       0.82      0.82      0.82        91
```

Para observação no gráfico, foram escolhidas as colunas referentes ao tipo de dor no peito ('cp') e à frequência cardíaca máxima alcançada.

In [ ]:
```
plt.scatter(X_train[:, 2], X_train[:, 7], c = y_train, cmap = plt.cm.coolwarm)
plt.scatter(X_test[:, 2], X_test[:, 7], c=y_test, cmap=plt.cm.coolwarm, marker='
plt.title('SVM Classification (2D)')
plt.xlabel('Chest Pain Type')
plt.ylabel('Max Heart Rate')
plt.show()
```



MLP

É realizada a chamada da função de Percetron de Múltiplas Camadas. O modelo é treinado e testado, suas iterações são apresentadas.

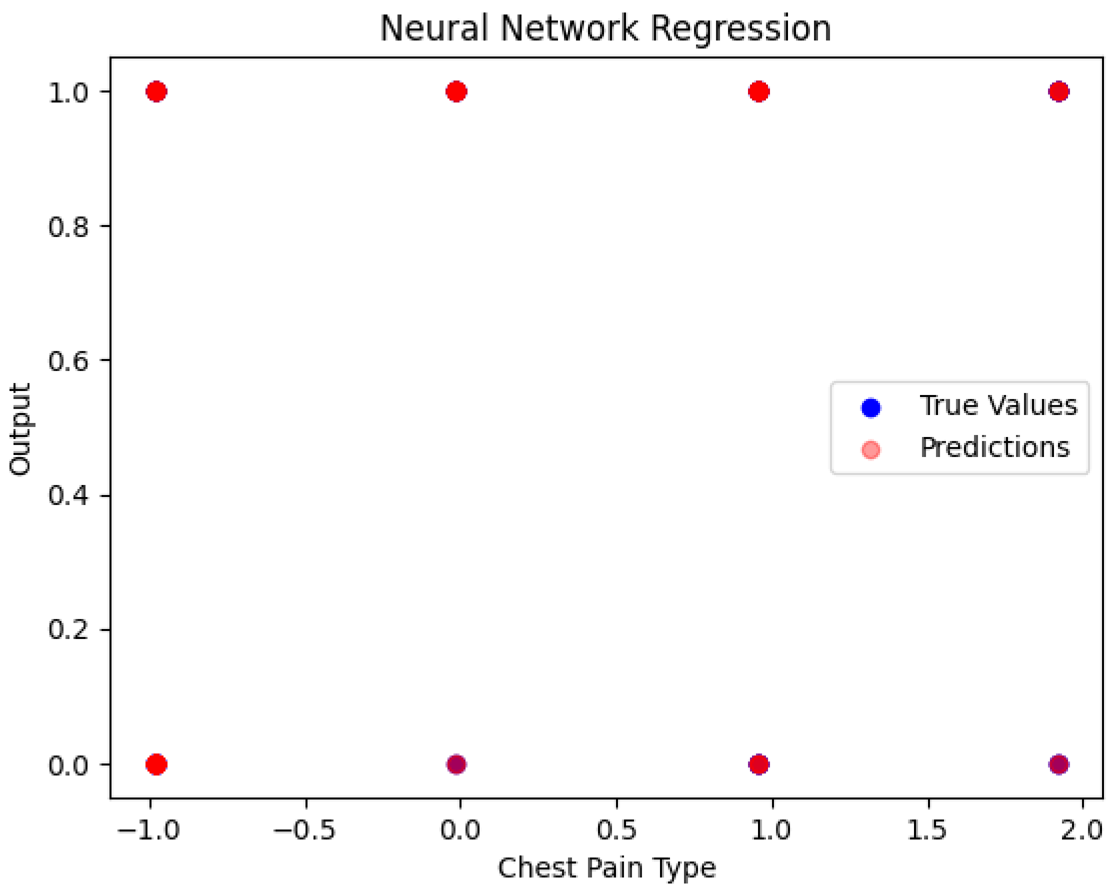
In [ ]:
```
mlp_classifier = MLPClassifier(hidden_layer_sizes=(50, 50), activation = 'relu',
mlp_classifier.fit(X_train, y_train)
y_pred = mlp_classifier.predict(X_test)
```

```python
accuracy = accuracy_score(y_test, y_pred)
print(f"Acurácia do método MLP: {accuracy:.4f}")
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

```
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. S
topping.
Acurácia do método MLP: 0.8242
Classification Report:
              precision    recall  f1-score   support

           0       0.79      0.83      0.81        41
           1       0.85      0.82      0.84        50

    accuracy                           0.82        91
   macro avg       0.82      0.82      0.82        91
weighted avg       0.83      0.82      0.82        91
```

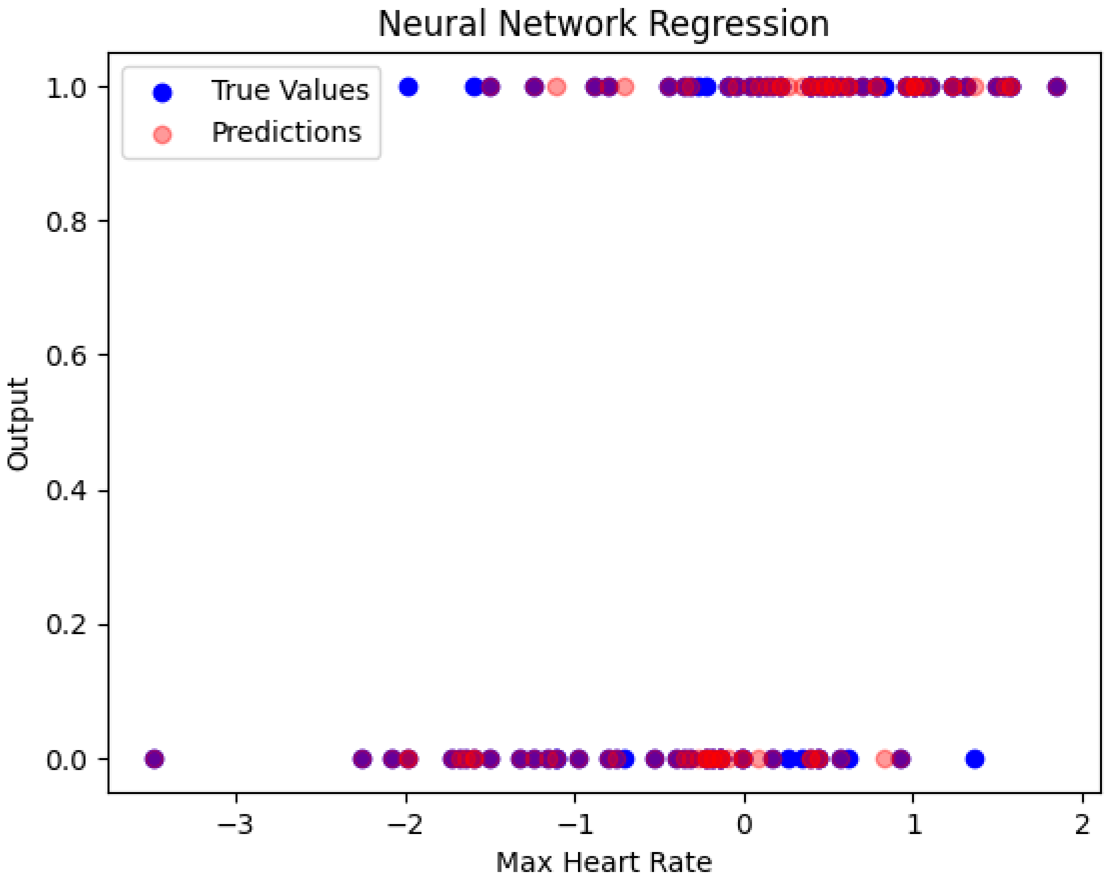
Novamente, observa-se o resultado de acordo com as varíaveis mais influentes no resultado.

In [ ]:
```
plt.scatter(X_test[:, 2], y_test, color='blue')
plt.scatter(X_test[:, 2], y_pred, color = 'red', alpha =.4)

plt.xlabel('Chest Pain Type')
plt.ylabel('Output')
plt.title('Neural Network Regression')

plt.legend(['True Values', 'Predictions'])
plt.show()
```



In [ ]:
```
plt.scatter(X_test[:, 7], y_test, color='blue')
plt.scatter(X_test[:, 7], y_pred, color = 'red', alpha =.4)

plt.xlabel('Max Heart Rate')
plt.ylabel('Output')
```

```python
plt.title('Neural Network Regression')

plt.legend(['True Values', 'Predictions'])
plt.show()
```



Resultados

Observando a acurácia, destaca-se o desempenho muito semelhante entre os três métodos. Um possível motivo para isso é a base de dados não possuir uma complexidade alta a ponto de precisar de abordagens diferentes para interpretar correta e satisfatoriamente os dados, de acordo com o requisito do trabalho (80% de acurácia). Assim, os métodos SVM e MLP, mais complexos, obtiveram um resultado ligeiramente superior à Regressão Logística.

Regressão - Trabaho IA - 2023.2

Para a classificação com algoritmo de regressão linear, SVM e MLP foi utilizada a seguinte base de dados: neural net regression data. Segue o código com os comentários nas células anteriores: Importando as bibliotecas e funções.

```
import pandas as pd
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
```

Base de dados

Lendo dados do arquivo CSV

```
profit = pd.read_csv('fake_reg.csv')
profit.head()
```

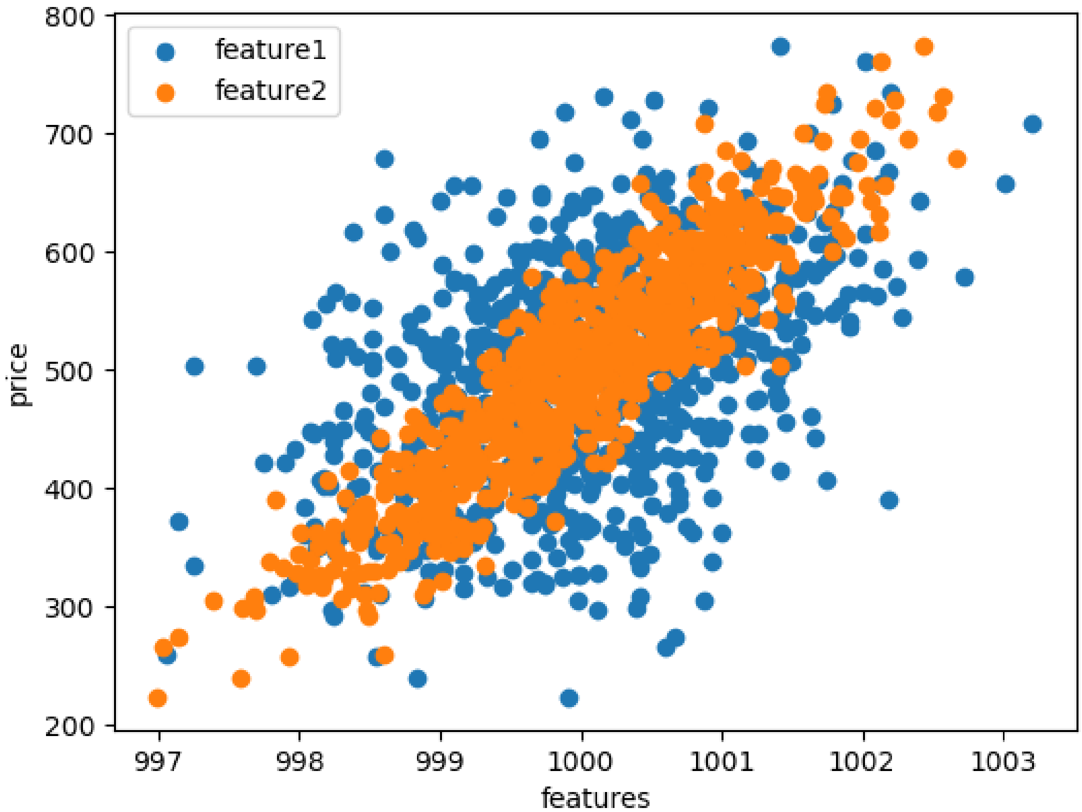
	price	feature1	feature2
0	461.527929	999.787558	999.766096
1	548.130011	998.861615	1001.042403
2	410.297162	1000.070267	998.844015
3	540.382220	999.952251	1000.440940
4	546.024553	1000.446011	1000.338531

dividindo em treinamento(80%) e teste(20%):

```
x_val = profit.drop(['price'], axis=1)
y_val = profit['price']
X_train, X_test, y_train, y_test = train_test_split(x_val, y_val,
test_size=0.2, random_state=42)
```

Plotando os dados

```
plt.scatter(X_train['feature1'], y_train, label='feature1')
plt.scatter(X_train['feature2'], y_train, label='feature2')
#plt.scatter(X_train['Marketing Spend'], y_train, label='Marketing
Spend Budget')
plt.xlabel('features')
plt.ylabel('price')
plt.legend()
plt.show()
```



Normalizando os dados

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train) # Don't cheat - fit only on training data
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test) # apply same transformation to test
data
```

Criando e utilizando MLP para regreção

```
from sklearn.neural_network import MLPRegressor
from sklearn.datasets import make_regression
from sklearn.model_selection import train_test_split
model = MLPRegressor(
    hidden_layer_sizes=[62,62],
    activation='relu',
    solver='adam',
    verbose=True,
    random_state=1,
    max_iter=1400)
model = model.fit(X_train, y_train)
```

```
Iteration 1, loss = 127663.64691961
Iteration 2, loss = 127554.74469284
Iteration 3, loss = 127444.92467452
Iteration 4, loss = 127332.32677298
Iteration 5, loss = 127217.03250398
Iteration 6, loss = 127095.17022512
Iteration 7, loss = 126966.26891460
Iteration 8, loss = 126827.70731930
Iteration 9, loss = 126675.78869056
Iteration 10, loss = 126509.50896640
Iteration 11, loss = 126327.40313574
Iteration 12, loss = 126127.01491168
Iteration 13, loss = 125902.34616198
Iteration 14, loss = 125659.52946040
Iteration 15, loss = 125386.01183819
Iteration 16, loss = 125082.56318260
Iteration 17, loss = 124754.38114170
Iteration 18, loss = 124387.71711036
Iteration 19, loss = 123988.48180558
Iteration 20, loss = 123547.98989864
Iteration 21, loss = 123065.03557686
Iteration 22, loss = 122539.39076850
Iteration 23, loss = 121964.16799312
Iteration 24, loss = 121343.16477053
Iteration 25, loss = 120662.51391963
Iteration 26, loss = 119944.68825320
Iteration 27, loss = 119147.83614609
Iteration 28, loss = 118303.33343480
Iteration 29, loss = 117386.81436537
Iteration 30, loss = 116413.51552792
Iteration 31, loss = 115367.59573187
Iteration 32, loss = 114235.39319650
Iteration 33, loss = 113037.22379949
Iteration 34, loss = 111761.97066179
Iteration 35, loss = 110405.32063594
Iteration 36, loss = 108967.23207268
Iteration 37, loss = 107443.22153157
Iteration 38, loss = 105860.70513740
Iteration 39, loss = 104172.99766098
Iteration 40, loss = 102408.32216260
Iteration 41, loss = 100572.88876531
Iteration 42, loss = 98632.73550495
Iteration 43, loss = 96661.54484015
Iteration 44, loss = 94555.92355812
Iteration 45, loss = 92407.56821236
Iteration 46, loss = 90169.57229675
Iteration 47, loss = 87893.95129082
Iteration 48, loss = 85509.18482856
Iteration 49, loss = 83105.59818814
Iteration 50, loss = 80610.28064933
```

```
Iteration 51, loss = 78073.27136360
Iteration 52, loss = 75490.97925757
Iteration 53, loss = 72865.45255015
Iteration 54, loss = 70194.11850501
Iteration 55, loss = 67519.88644987
Iteration 56, loss = 64835.49462285
Iteration 57, loss = 62120.44078938
Iteration 58, loss = 59370.01268115
Iteration 59, loss = 56641.61836670
Iteration 60, loss = 53937.20567435
Iteration 61, loss = 51247.08642080
Iteration 62, loss = 48583.52888027
Iteration 63, loss = 45950.17160806
Iteration 64, loss = 43354.50343651
Iteration 65, loss = 40836.22305860
Iteration 66, loss = 38345.06448705
Iteration 67, loss = 35942.08279910
Iteration 68, loss = 33599.40539349
Iteration 69, loss = 31361.42548150
Iteration 70, loss = 29185.48392180
Iteration 71, loss = 27112.63489903
Iteration 72, loss = 25137.61853971
Iteration 73, loss = 23288.76353799
Iteration 74, loss = 21510.10598070
Iteration 75, loss = 19851.52637978
Iteration 76, loss = 18297.93737848
Iteration 77, loss = 16847.93153257
Iteration 78, loss = 15526.00393182
Iteration 79, loss = 14313.38077571
Iteration 80, loss = 13184.22480798
Iteration 81, loss = 12172.96462457
Iteration 82, loss = 11251.05955193
Iteration 83, loss = 10420.02935285
Iteration 84, loss = 9685.60320504
Iteration 85, loss = 9003.92873212
Iteration 86, loss = 8444.22402966
Iteration 87, loss = 7926.85703235
Iteration 88, loss = 7473.90797617
Iteration 89, loss = 7087.57367107
Iteration 90, loss = 6742.06750706
Iteration 91, loss = 6442.88115625
Iteration 92, loss = 6176.53348477
Iteration 93, loss = 5956.37798362
Iteration 94, loss = 5755.89831947
Iteration 95, loss = 5589.97420835
Iteration 96, loss = 5435.56876658
Iteration 97, loss = 5298.53711057
Iteration 98, loss = 5182.19466812
Iteration 99, loss = 5076.83212509
Iteration 100, loss = 4983.16712118
```

```
Iteration 101, loss = 4892.49480302
Iteration 102, loss = 4812.44782891
Iteration 103, loss = 4732.88593559
Iteration 104, loss = 4659.57640874
Iteration 105, loss = 4590.55027968
Iteration 106, loss = 4523.20506433
Iteration 107, loss = 4460.34629229
Iteration 108, loss = 4393.77679044
Iteration 109, loss = 4331.59895662
Iteration 110, loss = 4271.44450967
Iteration 111, loss = 4211.35980571
Iteration 112, loss = 4152.13077762
Iteration 113, loss = 4094.27210203
Iteration 114, loss = 4035.83282913
Iteration 115, loss = 3979.08485024
Iteration 116, loss = 3924.15720994
Iteration 117, loss = 3867.65232543
Iteration 118, loss = 3811.39627582
Iteration 119, loss = 3756.79483508
Iteration 120, loss = 3702.03388946
Iteration 121, loss = 3648.80678280
Iteration 122, loss = 3594.70078230
Iteration 123, loss = 3542.19999375
Iteration 124, loss = 3489.54430948
Iteration 125, loss = 3437.13166311
Iteration 126, loss = 3385.37684836
Iteration 127, loss = 3334.15985739
Iteration 128, loss = 3283.02659085
Iteration 129, loss = 3232.55302920
Iteration 130, loss = 3181.36603405
Iteration 131, loss = 3132.00861331
Iteration 132, loss = 3080.31565055
Iteration 133, loss = 3029.49632603
Iteration 134, loss = 2980.09430726
Iteration 135, loss = 2933.29450235
Iteration 136, loss = 2884.32406953
Iteration 137, loss = 2836.11938117
Iteration 138, loss = 2788.60356460
Iteration 139, loss = 2742.63207013
Iteration 140, loss = 2694.17346421
Iteration 141, loss = 2649.81015191
Iteration 142, loss = 2604.82109605
Iteration 143, loss = 2559.98848871
Iteration 144, loss = 2516.26657290
Iteration 145, loss = 2471.34184722
Iteration 146, loss = 2429.83339692
Iteration 147, loss = 2387.36454305
Iteration 148, loss = 2345.40025884
Iteration 149, loss = 2303.95367044
Iteration 150, loss = 2263.13794944
```

```
Iteration 151, loss = 2223.47711894
Iteration 152, loss = 2185.06940314
Iteration 153, loss = 2145.25832236
Iteration 154, loss = 2107.64291900
Iteration 155, loss = 2067.88326966
Iteration 156, loss = 2031.88049662
Iteration 157, loss = 1995.49589249
Iteration 158, loss = 1958.92117581
Iteration 159, loss = 1922.90169040
Iteration 160, loss = 1887.91916134
Iteration 161, loss = 1854.21424517
Iteration 162, loss = 1819.06378722
Iteration 163, loss = 1786.65761024
Iteration 164, loss = 1753.56308313
Iteration 165, loss = 1722.05404158
Iteration 166, loss = 1689.56674188
Iteration 167, loss = 1659.58367230
Iteration 168, loss = 1628.02629439
Iteration 169, loss = 1597.99882118
Iteration 170, loss = 1568.32138430
Iteration 171, loss = 1539.78959085
Iteration 172, loss = 1510.13782283
Iteration 173, loss = 1482.14961753
Iteration 174, loss = 1454.51745057
Iteration 175, loss = 1427.22664809
Iteration 176, loss = 1400.22925796
Iteration 177, loss = 1374.02449431
Iteration 178, loss = 1347.81495275
Iteration 179, loss = 1322.18414405
Iteration 180, loss = 1296.73658159
Iteration 181, loss = 1272.14703481
Iteration 182, loss = 1248.37442157
Iteration 183, loss = 1224.28039354
Iteration 184, loss = 1201.21260794
Iteration 185, loss = 1178.06959954
Iteration 186, loss = 1155.46550875
Iteration 187, loss = 1133.41979096
Iteration 188, loss = 1112.25178157
Iteration 189, loss = 1090.14565069
Iteration 190, loss = 1069.66698692
Iteration 191, loss = 1049.70430929
Iteration 192, loss = 1028.31499794
Iteration 193, loss = 1009.12008224
Iteration 194, loss = 989.30017988
Iteration 195, loss = 971.09335936
Iteration 196, loss = 951.47540420
Iteration 197, loss = 933.05174947
Iteration 198, loss = 915.16210875
Iteration 199, loss = 898.00284780
Iteration 200, loss = 880.16188163
```

```
Iteration 201, loss = 862.59464675
Iteration 202, loss = 846.54078974
Iteration 203, loss = 829.72354718
Iteration 204, loss = 814.07330294
Iteration 205, loss = 797.52210110
Iteration 206, loss = 781.67070515
Iteration 207, loss = 767.50263440
Iteration 208, loss = 751.87402317
Iteration 209, loss = 737.11686319
Iteration 210, loss = 723.04664269
Iteration 211, loss = 709.31826895
Iteration 212, loss = 694.67660239
Iteration 213, loss = 681.27440032
Iteration 214, loss = 668.48861642
Iteration 215, loss = 655.38272255
Iteration 216, loss = 642.25857721
Iteration 217, loss = 629.84662690
Iteration 218, loss = 617.76999760
Iteration 219, loss = 606.24835121
Iteration 220, loss = 594.17503485
Iteration 221, loss = 582.33729462
Iteration 222, loss = 571.49064809
Iteration 223, loss = 560.09236950
Iteration 224, loss = 549.52754396
Iteration 225, loss = 538.96962250
Iteration 226, loss = 528.63831017
Iteration 227, loss = 517.96119680
Iteration 228, loss = 508.45444836
Iteration 229, loss = 498.59772498
Iteration 230, loss = 488.76443339
Iteration 231, loss = 479.78106545
Iteration 232, loss = 470.66512891
Iteration 233, loss = 461.20389545
Iteration 234, loss = 452.63622276
Iteration 235, loss = 443.74873398
Iteration 236, loss = 435.48724982
Iteration 237, loss = 426.75313803
Iteration 238, loss = 418.75030898
Iteration 239, loss = 410.46779070
Iteration 240, loss = 402.53975226
Iteration 241, loss = 394.73773001
Iteration 242, loss = 387.45932964
Iteration 243, loss = 379.60431482
Iteration 244, loss = 372.40634609
Iteration 245, loss = 365.14057439
Iteration 246, loss = 358.02281052
Iteration 247, loss = 351.15029458
Iteration 248, loss = 344.49682908
Iteration 249, loss = 337.69509515
Iteration 250, loss = 331.03296424
```

```
Iteration 251, loss = 324.88614864
Iteration 252, loss = 318.74498912
Iteration 253, loss = 312.38736349
Iteration 254, loss = 306.44557463
Iteration 255, loss = 300.41531293
Iteration 256, loss = 294.35163681
Iteration 257, loss = 288.79469953
Iteration 258, loss = 283.17081549
Iteration 259, loss = 277.81681716
Iteration 260, loss = 272.27605779
Iteration 261, loss = 267.04219486
Iteration 262, loss = 262.08381451
Iteration 263, loss = 256.91140223
Iteration 264, loss = 251.94705217
Iteration 265, loss = 247.15309464
Iteration 266, loss = 242.52092082
Iteration 267, loss = 237.76783725
Iteration 268, loss = 233.24160431
Iteration 269, loss = 228.68790523
Iteration 270, loss = 224.35231502
Iteration 271, loss = 220.37274080
Iteration 272, loss = 215.91368731
Iteration 273, loss = 211.90044656
Iteration 274, loss = 207.93000668
Iteration 275, loss = 204.00499960
Iteration 276, loss = 200.18462144
Iteration 277, loss = 196.49622517
Iteration 278, loss = 192.71676676
Iteration 279, loss = 189.28630890
Iteration 280, loss = 185.91302694
Iteration 281, loss = 182.56945729
Iteration 282, loss = 179.02907592
Iteration 283, loss = 175.62115337
Iteration 284, loss = 172.41118204
Iteration 285, loss = 169.23446911
Iteration 286, loss = 166.16014294
Iteration 287, loss = 163.19925813
Iteration 288, loss = 160.09365735
Iteration 289, loss = 157.14234836
Iteration 290, loss = 154.42693347
Iteration 291, loss = 151.68743291
Iteration 292, loss = 148.83795778
Iteration 293, loss = 146.31781621
Iteration 294, loss = 143.57134775
Iteration 295, loss = 141.00320577
Iteration 296, loss = 138.62468582
Iteration 297, loss = 136.13664815
Iteration 298, loss = 133.74684182
Iteration 299, loss = 131.38505665
Iteration 300, loss = 129.07060113
```

```
Iteration 301, loss = 126.87324018
Iteration 302, loss = 124.61075621
Iteration 303, loss = 122.58200841
Iteration 304, loss = 120.43869651
Iteration 305, loss = 118.32081649
Iteration 306, loss = 116.40585655
Iteration 307, loss = 114.41424066
Iteration 308, loss = 112.51470088
Iteration 309, loss = 110.66391999
Iteration 310, loss = 108.81263743
Iteration 311, loss = 107.00736805
Iteration 312, loss = 105.26190818
Iteration 313, loss = 103.57054117
Iteration 314, loss = 101.79299586
Iteration 315, loss = 100.12122747
Iteration 316, loss = 98.54862593
Iteration 317, loss = 96.94539832
Iteration 318, loss = 95.45893477
Iteration 319, loss = 93.95709652
Iteration 320, loss = 92.52884927
Iteration 321, loss = 90.87198626
Iteration 322, loss = 89.46729622
Iteration 323, loss = 88.10720863
Iteration 324, loss = 86.68853713
Iteration 325, loss = 85.27126772
Iteration 326, loss = 84.08308937
Iteration 327, loss = 82.68856959
Iteration 328, loss = 81.42628322
Iteration 329, loss = 80.20696044
Iteration 330, loss = 79.03342714
Iteration 331, loss = 77.75943017
Iteration 332, loss = 76.55388114
Iteration 333, loss = 75.43322306
Iteration 334, loss = 74.31373106
Iteration 335, loss = 73.29123712
Iteration 336, loss = 72.18677117
Iteration 337, loss = 71.11863846
Iteration 338, loss = 70.08114645
Iteration 339, loss = 69.10407005
Iteration 340, loss = 68.15279639
Iteration 341, loss = 67.12020714
Iteration 342, loss = 66.29028324
Iteration 343, loss = 65.24927873
Iteration 344, loss = 64.37991198
Iteration 345, loss = 63.47851861
Iteration 346, loss = 62.65567517
Iteration 347, loss = 61.75785179
Iteration 348, loss = 60.93488353
Iteration 349, loss = 60.13218776
Iteration 350, loss = 59.30676645
```

```
Iteration 351, loss = 58.58075530
Iteration 352, loss = 57.81924215
Iteration 353, loss = 57.03276842
Iteration 354, loss = 56.31203687
Iteration 355, loss = 55.53201757
Iteration 356, loss = 54.86632547
Iteration 357, loss = 54.20300893
Iteration 358, loss = 53.50621803
Iteration 359, loss = 52.84404174
Iteration 360, loss = 52.20212636
Iteration 361, loss = 51.57942833
Iteration 362, loss = 50.94250315
Iteration 363, loss = 50.34139289
Iteration 364, loss = 49.77052095
Iteration 365, loss = 49.14175419
Iteration 366, loss = 48.56023455
Iteration 367, loss = 48.00329250
Iteration 368, loss = 47.46299402
Iteration 369, loss = 46.91321428
Iteration 370, loss = 46.40102173
Iteration 371, loss = 45.90476900
Iteration 372, loss = 45.39429013
Iteration 373, loss = 44.96160669
Iteration 374, loss = 44.38395107
Iteration 375, loss = 43.95549108
Iteration 376, loss = 43.47727538
Iteration 377, loss = 43.04319160
Iteration 378, loss = 42.58859505
Iteration 379, loss = 42.14490923
Iteration 380, loss = 41.68941723
Iteration 381, loss = 41.26548603
Iteration 382, loss = 40.87206396
Iteration 383, loss = 40.41038252
Iteration 384, loss = 40.01062308
Iteration 385, loss = 39.61254829
Iteration 386, loss = 39.17877224
Iteration 387, loss = 38.80181434
Iteration 388, loss = 38.39781982
Iteration 389, loss = 38.04137687
Iteration 390, loss = 37.62225136
Iteration 391, loss = 37.28568862
Iteration 392, loss = 36.90370646
Iteration 393, loss = 36.51895450
Iteration 394, loss = 36.15826407
Iteration 395, loss = 35.84000289
Iteration 396, loss = 35.47930867
Iteration 397, loss = 35.14706461
Iteration 398, loss = 34.81143307
Iteration 399, loss = 34.45173952
Iteration 400, loss = 34.09032881
```

```
Iteration 401, loss = 33.76393265
Iteration 402, loss = 33.46106306
Iteration 403, loss = 33.10156185
Iteration 404, loss = 32.83657557
Iteration 405, loss = 32.50897046
Iteration 406, loss = 32.24131409
Iteration 407, loss = 31.86553665
Iteration 408, loss = 31.60302785
Iteration 409, loss = 31.34407512
Iteration 410, loss = 31.03715396
Iteration 411, loss = 30.79681468
Iteration 412, loss = 30.48325176
Iteration 413, loss = 30.26129620
Iteration 414, loss = 29.97993547
Iteration 415, loss = 29.73196704
Iteration 416, loss = 29.47772046
Iteration 417, loss = 29.23012249
Iteration 418, loss = 29.00772275
Iteration 419, loss = 28.76698303
Iteration 420, loss = 28.53576803
Iteration 421, loss = 28.30524507
Iteration 422, loss = 28.06242557
Iteration 423, loss = 27.82306233
Iteration 424, loss = 27.62005588
Iteration 425, loss = 27.39525655
Iteration 426, loss = 27.20361661
Iteration 427, loss = 27.00922665
Iteration 428, loss = 26.78034527
Iteration 429, loss = 26.58283911
Iteration 430, loss = 26.39615833
Iteration 431, loss = 26.21804838
Iteration 432, loss = 26.00233832
Iteration 433, loss = 25.85828993
Iteration 434, loss = 25.64732395
Iteration 435, loss = 25.47480878
Iteration 436, loss = 25.29639215
Iteration 437, loss = 25.13746977
Iteration 438, loss = 24.96834939
Iteration 439, loss = 24.79514411
Iteration 440, loss = 24.64095353
Iteration 441, loss = 24.47904285
Iteration 442, loss = 24.34860769
Iteration 443, loss = 24.18173690
Iteration 444, loss = 24.02740821
Iteration 445, loss = 23.89244175
Iteration 446, loss = 23.73433859
Iteration 447, loss = 23.60477484
Iteration 448, loss = 23.47736236
Iteration 449, loss = 23.34095625
Iteration 450, loss = 23.19743173
```

```
Iteration 451, loss = 23.08209477
Iteration 452, loss = 22.95130895
Iteration 453, loss = 22.82547421
Iteration 454, loss = 22.70668501
Iteration 455, loss = 22.59855726
Iteration 456, loss = 22.47840412
Iteration 457, loss = 22.36318219
Iteration 458, loss = 22.24251574
Iteration 459, loss = 22.11569082
Iteration 460, loss = 22.03755248
Iteration 461, loss = 21.88602322
Iteration 462, loss = 21.78412536
Iteration 463, loss = 21.66275708
Iteration 464, loss = 21.57576263
Iteration 465, loss = 21.45779616
Iteration 466, loss = 21.34627597
Iteration 467, loss = 21.26437292
Iteration 468, loss = 21.16380697
Iteration 469, loss = 21.05748962
Iteration 470, loss = 20.95917057
Iteration 471, loss = 20.86049218
Iteration 472, loss = 20.74099991
Iteration 473, loss = 20.65584089
Iteration 474, loss = 20.56814103
Iteration 475, loss = 20.48430931
Iteration 476, loss = 20.39023838
Iteration 477, loss = 20.29152884
Iteration 478, loss = 20.20767596
Iteration 479, loss = 20.11632809
Iteration 480, loss = 20.03180433
Iteration 481, loss = 19.94897098
Iteration 482, loss = 19.86183127
Iteration 483, loss = 19.78967697
Iteration 484, loss = 19.71416928
Iteration 485, loss = 19.62578514
Iteration 486, loss = 19.54147525
Iteration 487, loss = 19.45090006
Iteration 488, loss = 19.36781627
Iteration 489, loss = 19.29888893
Iteration 490, loss = 19.21735356
Iteration 491, loss = 19.13275641
Iteration 492, loss = 19.05927590
Iteration 493, loss = 18.98484261
Iteration 494, loss = 18.91556578
Iteration 495, loss = 18.83307433
Iteration 496, loss = 18.75872968
Iteration 497, loss = 18.69265936
Iteration 498, loss = 18.62826125
Iteration 499, loss = 18.58296636
Iteration 500, loss = 18.49711010
```

```
Iteration 501, loss = 18.42659305
Iteration 502, loss = 18.36819030
Iteration 503, loss = 18.32262278
Iteration 504, loss = 18.22592210
Iteration 505, loss = 18.16670342
Iteration 506, loss = 18.11311361
Iteration 507, loss = 18.04795382
Iteration 508, loss = 17.98210412
Iteration 509, loss = 17.94473855
Iteration 510, loss = 17.87138938
Iteration 511, loss = 17.80371613
Iteration 512, loss = 17.75506189
Iteration 513, loss = 17.70226957
Iteration 514, loss = 17.64048650
Iteration 515, loss = 17.58644985
Iteration 516, loss = 17.52268216
Iteration 517, loss = 17.49595835
Iteration 518, loss = 17.42485130
Iteration 519, loss = 17.37240870
Iteration 520, loss = 17.31908988
Iteration 521, loss = 17.27637568
Iteration 522, loss = 17.21682072
Iteration 523, loss = 17.16496445
Iteration 524, loss = 17.11496976
Iteration 525, loss = 17.06226829
Iteration 526, loss = 16.99864684
Iteration 527, loss = 16.94655190
Iteration 528, loss = 16.91451094
Iteration 529, loss = 16.84584256
Iteration 530, loss = 16.80993452
Iteration 531, loss = 16.74332475
Iteration 532, loss = 16.70826000
Iteration 533, loss = 16.64369222
Iteration 534, loss = 16.58390772
Iteration 535, loss = 16.54955212
Iteration 536, loss = 16.49872377
Iteration 537, loss = 16.44998640
Iteration 538, loss = 16.41288139
Iteration 539, loss = 16.35866114
Iteration 540, loss = 16.32844797
Iteration 541, loss = 16.29873221
Iteration 542, loss = 16.26450884
Iteration 543, loss = 16.20705118
Iteration 544, loss = 16.16230335
Iteration 545, loss = 16.12764129
Iteration 546, loss = 16.08872297
Iteration 547, loss = 16.07025430
Iteration 548, loss = 16.02007373
Iteration 549, loss = 15.97504589
Iteration 550, loss = 15.94549046
```

```
Iteration 551, loss = 15.90726724
Iteration 552, loss = 15.86935060
Iteration 553, loss = 15.82739068
Iteration 554, loss = 15.80964883
Iteration 555, loss = 15.76097289
Iteration 556, loss = 15.72435571
Iteration 557, loss = 15.70103259
Iteration 558, loss = 15.65954838
Iteration 559, loss = 15.63641141
Iteration 560, loss = 15.59372710
Iteration 561, loss = 15.57353621
Iteration 562, loss = 15.54954835
Iteration 563, loss = 15.50531977
Iteration 564, loss = 15.47812133
Iteration 565, loss = 15.45145082
Iteration 566, loss = 15.42756626
Iteration 567, loss = 15.40314795
Iteration 568, loss = 15.36051177
Iteration 569, loss = 15.33857595
Iteration 570, loss = 15.30561885
Iteration 571, loss = 15.28786287
Iteration 572, loss = 15.25175601
Iteration 573, loss = 15.23228852
Iteration 574, loss = 15.20567319
Iteration 575, loss = 15.18290428
Iteration 576, loss = 15.15613481
Iteration 577, loss = 15.14956386
Iteration 578, loss = 15.12450260
Iteration 579, loss = 15.10108822
Iteration 580, loss = 15.07489634
Iteration 581, loss = 15.04945818
Iteration 582, loss = 15.01948523
Iteration 583, loss = 14.99658255
Iteration 584, loss = 14.97730641
Iteration 585, loss = 14.95289868
Iteration 586, loss = 14.92703539
Iteration 587, loss = 14.91072566
Iteration 588, loss = 14.89929074
Iteration 589, loss = 14.88464355
Iteration 590, loss = 14.84670449
Iteration 591, loss = 14.84765616
Iteration 592, loss = 14.80287021
Iteration 593, loss = 14.78129048
Iteration 594, loss = 14.76912814
Iteration 595, loss = 14.76256869
Iteration 596, loss = 14.72377989
Iteration 597, loss = 14.71560019
Iteration 598, loss = 14.69312745
Iteration 599, loss = 14.67283368
```

```
Iteration 600, loss = 14.65406163
Iteration 601, loss = 14.63031138
Iteration 602, loss = 14.63689338
Iteration 603, loss = 14.59830646
Iteration 604, loss = 14.57570318
Iteration 605, loss = 14.55185914
Iteration 606, loss = 14.53027359
Iteration 607, loss = 14.54121489
Iteration 608, loss = 14.49936422
Iteration 609, loss = 14.48112050
Iteration 610, loss = 14.46220270
Iteration 611, loss = 14.44129523
Iteration 612, loss = 14.42131701
Iteration 613, loss = 14.40881496
Iteration 614, loss = 14.38482057
Iteration 615, loss = 14.38024310
Iteration 616, loss = 14.34759036
Iteration 617, loss = 14.32468827
Iteration 618, loss = 14.31211778
Iteration 619, loss = 14.29523295
Iteration 620, loss = 14.27366952
Iteration 621, loss = 14.25545515
Iteration 622, loss = 14.24129338
Iteration 623, loss = 14.21967394
Iteration 624, loss = 14.20830398
Iteration 625, loss = 14.19884800
Iteration 626, loss = 14.17064370
Iteration 627, loss = 14.14743121
Iteration 628, loss = 14.13765109
Iteration 629, loss = 14.11704886
Iteration 630, loss = 14.10557715
Iteration 631, loss = 14.08734191
Iteration 632, loss = 14.06521599
Iteration 633, loss = 14.04732192
Iteration 634, loss = 14.03557169
Iteration 635, loss = 14.03256953
Iteration 636, loss = 14.00506605
Iteration 637, loss = 13.98182432
Iteration 638, loss = 13.97213502
Iteration 639, loss = 13.95862213
Iteration 640, loss = 13.93401322
Iteration 641, loss = 13.92363094
Iteration 642, loss = 13.90421608
Iteration 643, loss = 13.89511638
Iteration 644, loss = 13.87171028
Iteration 645, loss = 13.86204179
Iteration 646, loss = 13.84399477
Iteration 647, loss = 13.83476600
Iteration 648, loss = 13.80852551
```

```
Iteration 649, loss = 13.79056271
Iteration 650, loss = 13.79523086
Iteration 651, loss = 13.76965074
Iteration 652, loss = 13.75613539
Iteration 653, loss = 13.74161881
Iteration 654, loss = 13.72560519
Iteration 655, loss = 13.71050919
Iteration 656, loss = 13.69490372
Iteration 657, loss = 13.68544291
Iteration 658, loss = 13.66645020
Iteration 659, loss = 13.65813563
Iteration 660, loss = 13.64636367
Iteration 661, loss = 13.62887904
Iteration 662, loss = 13.63969461
Iteration 663, loss = 13.61177889
Iteration 664, loss = 13.58839106
Iteration 665, loss = 13.58516400
Iteration 666, loss = 13.57492422
Iteration 667, loss = 13.55350599
Iteration 668, loss = 13.55031094
Iteration 669, loss = 13.53566086
Iteration 670, loss = 13.50836868
Iteration 671, loss = 13.50764344
Iteration 672, loss = 13.48948882
Iteration 673, loss = 13.48053485
Iteration 674, loss = 13.46533939
Iteration 675, loss = 13.46218988
Iteration 676, loss = 13.44688855
Iteration 677, loss = 13.42742268
Iteration 678, loss = 13.42835969
Iteration 679, loss = 13.40312243
Iteration 680, loss = 13.39077406
Iteration 681, loss = 13.38268775
Iteration 682, loss = 13.40229972
Iteration 683, loss = 13.35134954
Iteration 684, loss = 13.34961491
Iteration 685, loss = 13.34283980
Iteration 686, loss = 13.32324175
Iteration 687, loss = 13.31183775
Iteration 688, loss = 13.30804639
Iteration 689, loss = 13.29580628
Iteration 690, loss = 13.27635015
Iteration 691, loss = 13.26548507
Iteration 692, loss = 13.25242876
Iteration 693, loss = 13.24011110
Iteration 694, loss = 13.23671238
Iteration 695, loss = 13.21949916
Iteration 696, loss = 13.21268022
Iteration 697, loss = 13.21441112
```

```
Iteration 698, loss = 13.19035742
Iteration 699, loss = 13.19087505
Iteration 700, loss = 13.18849322
Iteration 701, loss = 13.19516420
Iteration 702, loss = 13.19027561
Iteration 703, loss = 13.15007057
Iteration 704, loss = 13.16048587
Iteration 705, loss = 13.12778166
Iteration 706, loss = 13.11711616
Iteration 707, loss = 13.10715491
Iteration 708, loss = 13.10099740
Iteration 709, loss = 13.09405587
Iteration 710, loss = 13.08086199
Iteration 711, loss = 13.07217395
Iteration 712, loss = 13.07244506
Iteration 713, loss = 13.05350739
Iteration 714, loss = 13.05027120
Iteration 715, loss = 13.04587996
Iteration 716, loss = 13.03957405
Iteration 717, loss = 13.03247849
Iteration 718, loss = 13.03131371
Iteration 719, loss = 13.00782705
Iteration 720, loss = 13.00632033
Iteration 721, loss = 12.98869752
Iteration 722, loss = 12.98122919
Iteration 723, loss = 12.97295133
Iteration 724, loss = 12.97369720
Iteration 725, loss = 12.97000565
Iteration 726, loss = 12.96603320
Iteration 727, loss = 12.94351852
Iteration 728, loss = 12.92421924
Iteration 729, loss = 12.91782811
Iteration 730, loss = 12.91734509
Iteration 731, loss = 12.90385612
Iteration 732, loss = 12.89086415
Iteration 733, loss = 12.89001367
Iteration 734, loss = 12.88078195
Iteration 735, loss = 12.87420651
Iteration 736, loss = 12.86413402
Iteration 737, loss = 12.87445143
Iteration 738, loss = 12.87703510
Iteration 739, loss = 12.85969035
Iteration 740, loss = 12.84459965
Iteration 741, loss = 12.83151154
Iteration 742, loss = 12.83020324
Iteration 743, loss = 12.83621331
Iteration 744, loss = 12.81357655
Iteration 745, loss = 12.80360150
Iteration 746, loss = 12.80252570
```

```
Iteration 747, loss = 12.78783271
Iteration 748, loss = 12.78332330
Iteration 749, loss = 12.78215277
Iteration 750, loss = 12.77069482
Iteration 751, loss = 12.76400194
Iteration 752, loss = 12.75410915
Iteration 753, loss = 12.75456573
Iteration 754, loss = 12.77046224
Iteration 755, loss = 12.73189678
Iteration 756, loss = 12.72260041
Iteration 757, loss = 12.71958464
Iteration 758, loss = 12.71236979
Iteration 759, loss = 12.70231878
Iteration 760, loss = 12.70106326
Iteration 761, loss = 12.69062822
Iteration 762, loss = 12.68211749
Iteration 763, loss = 12.67192615
Iteration 764, loss = 12.67140416
Iteration 765, loss = 12.66482259
Iteration 766, loss = 12.66372536
Iteration 767, loss = 12.66238203
Iteration 768, loss = 12.64717043
Iteration 769, loss = 12.64596022
Iteration 770, loss = 12.63054890
Iteration 771, loss = 12.62932851
Iteration 772, loss = 12.61735728
Iteration 773, loss = 12.62373142
Iteration 774, loss = 12.61985961
Iteration 775, loss = 12.61801866
Iteration 776, loss = 12.61945997
Iteration 777, loss = 12.60283279
Iteration 778, loss = 12.59123155
Iteration 779, loss = 12.58288841
Iteration 780, loss = 12.58161252
Iteration 781, loss = 12.58386620
Iteration 782, loss = 12.56541576
Iteration 783, loss = 12.58144845
Iteration 784, loss = 12.56316877
Iteration 785, loss = 12.55460736
Iteration 786, loss = 12.55117151
Iteration 787, loss = 12.54898377
Iteration 788, loss = 12.55772903
Iteration 789, loss = 12.54260487
Iteration 790, loss = 12.54176769
Iteration 791, loss = 12.53884610
Iteration 792, loss = 12.52925188
Iteration 793, loss = 12.51995122
Iteration 794, loss = 12.51799668
Iteration 795, loss = 12.51261571
```

```
Iteration 796, loss = 12.51417624
Iteration 797, loss = 12.50957686
Iteration 798, loss = 12.50224285
Iteration 799, loss = 12.50478482
Iteration 800, loss = 12.49588053
Iteration 801, loss = 12.48837570
Iteration 802, loss = 12.48294856
Iteration 803, loss = 12.48390136
Iteration 804, loss = 12.48105598
Iteration 805, loss = 12.48526625
Iteration 806, loss = 12.46802678
Iteration 807, loss = 12.47236645
Iteration 808, loss = 12.46194148
Iteration 809, loss = 12.45437424
Iteration 810, loss = 12.45319943
Iteration 811, loss = 12.44776907
Iteration 812, loss = 12.45375803
Iteration 813, loss = 12.44897994
Iteration 814, loss = 12.42864633
Iteration 815, loss = 12.42716244
Iteration 816, loss = 12.42560371
Iteration 817, loss = 12.42056913
Iteration 818, loss = 12.41646895
Iteration 819, loss = 12.41561089
Iteration 820, loss = 12.41293982
Iteration 821, loss = 12.41168193
Iteration 822, loss = 12.41281813
Iteration 823, loss = 12.39825464
Iteration 824, loss = 12.39572656
Iteration 825, loss = 12.39122487
Iteration 826, loss = 12.39609062
Iteration 827, loss = 12.38406947
Iteration 828, loss = 12.39281261
Iteration 829, loss = 12.38640025
Iteration 830, loss = 12.37821580
Iteration 831, loss = 12.36471631
Iteration 832, loss = 12.37606733
Iteration 833, loss = 12.36426487
Iteration 834, loss = 12.36746370
Iteration 835, loss = 12.36976767
Iteration 836, loss = 12.35070139
Iteration 837, loss = 12.36266984
Iteration 838, loss = 12.34648173
Iteration 839, loss = 12.36378453
Iteration 840, loss = 12.36750660
Iteration 841, loss = 12.34765628
Iteration 842, loss = 12.34632060
Iteration 843, loss = 12.33857829
Iteration 844, loss = 12.33594379
```

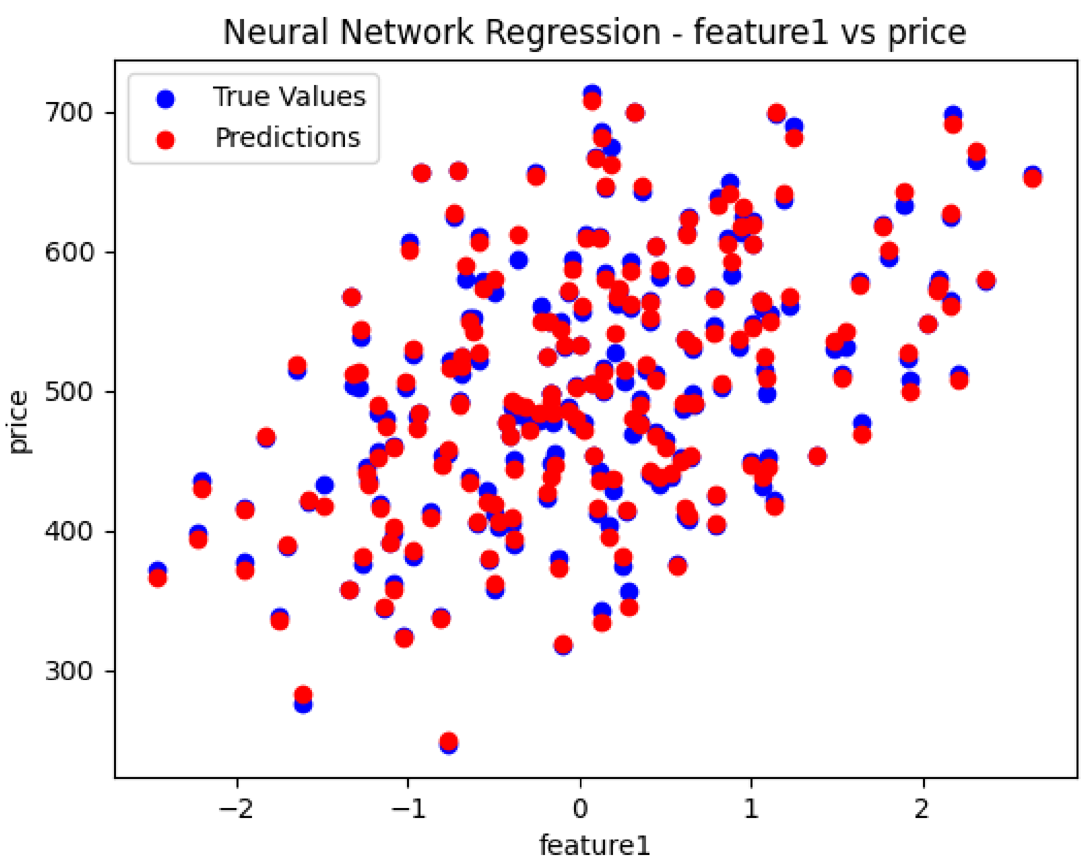
```
Iteration 845, loss = 12.32420433
Iteration 846, loss = 12.33439119
Iteration 847, loss = 12.32092021
Iteration 848, loss = 12.30673765
Iteration 849, loss = 12.32545726
Iteration 850, loss = 12.31174443
Iteration 851, loss = 12.30114975
Iteration 852, loss = 12.29781681
Iteration 853, loss = 12.29131591
Iteration 854, loss = 12.29933235
Iteration 855, loss = 12.29986930
Iteration 856, loss = 12.29144319
Iteration 857, loss = 12.28409654
Iteration 858, loss = 12.30540983
Iteration 859, loss = 12.28094372
Iteration 860, loss = 12.27468124
Iteration 861, loss = 12.27203684
Iteration 862, loss = 12.26995911
Iteration 863, loss = 12.26397821
Iteration 864, loss = 12.27115901
Iteration 865, loss = 12.27358604
Iteration 866, loss = 12.25440806
Iteration 867, loss = 12.26924328
Iteration 868, loss = 12.26633705
Iteration 869, loss = 12.25222271
Iteration 870, loss = 12.25788281
Iteration 871, loss = 12.24605415
Iteration 872, loss = 12.24697651
Iteration 873, loss = 12.25387667
Iteration 874, loss = 12.23491770
Iteration 875, loss = 12.23617717
Iteration 876, loss = 12.24661398
Iteration 877, loss = 12.22618634
Iteration 878, loss = 12.24327635
Iteration 879, loss = 12.22824757
Iteration 880, loss = 12.21378895
Iteration 881, loss = 12.21004824
Iteration 882, loss = 12.22394622
Iteration 883, loss = 12.22252781
Iteration 884, loss = 12.22003893
Iteration 885, loss = 12.22053286
Iteration 886, loss = 12.21404003
Iteration 887, loss = 12.21579023
Iteration 888, loss = 12.20261724
Iteration 889, loss = 12.21477867
Iteration 890, loss = 12.21196331
Iteration 891, loss = 12.18965935
Iteration 892, loss = 12.22040145
Iteration 893, loss = 12.18730327
```

```
Iteration 894, loss = 12.18733008
Iteration 895, loss = 12.21534242
Iteration 896, loss = 12.17703651
Iteration 897, loss = 12.18539147
Iteration 898, loss = 12.17182883
Iteration 899, loss = 12.17165868
Iteration 900, loss = 12.16177216
Iteration 901, loss = 12.17394806
Iteration 902, loss = 12.17221693
Iteration 903, loss = 12.16772313
Iteration 904, loss = 12.16825965
Iteration 905, loss = 12.14527510
Iteration 906, loss = 12.15566161
Iteration 907, loss = 12.15366184
Iteration 908, loss = 12.14853896
Iteration 909, loss = 12.14337072
Iteration 910, loss = 12.14511402
Iteration 911, loss = 12.15512499
Iteration 912, loss = 12.13612177
Iteration 913, loss = 12.13434514
Iteration 914, loss = 12.13376529
Iteration 915, loss = 12.13093740
Iteration 916, loss = 12.12445169
Iteration 917, loss = 12.12136886
Iteration 918, loss = 12.13690841
Iteration 919, loss = 12.10605462
Iteration 920, loss = 12.11614664
Iteration 921, loss = 12.12214794
Iteration 922, loss = 12.10983164
Iteration 923, loss = 12.10709778
Iteration 924, loss = 12.11566436
Iteration 925, loss = 12.10060265
Iteration 926, loss = 12.11107637
Iteration 927, loss = 12.08433447
Iteration 928, loss = 12.08297917
Iteration 929, loss = 12.09753573
Iteration 930, loss = 12.08577139
Iteration 931, loss = 12.08531963
Iteration 932, loss = 12.08921580
Iteration 933, loss = 12.07857273
Iteration 934, loss = 12.07827668
Iteration 935, loss = 12.07448953
Iteration 936, loss = 12.07284879
Iteration 937, loss = 12.07350690
Iteration 938, loss = 12.07547404
Iteration 939, loss = 12.06436984
Iteration 940, loss = 12.06706033
Iteration 941, loss = 12.06293206
Iteration 942, loss = 12.07743856
```

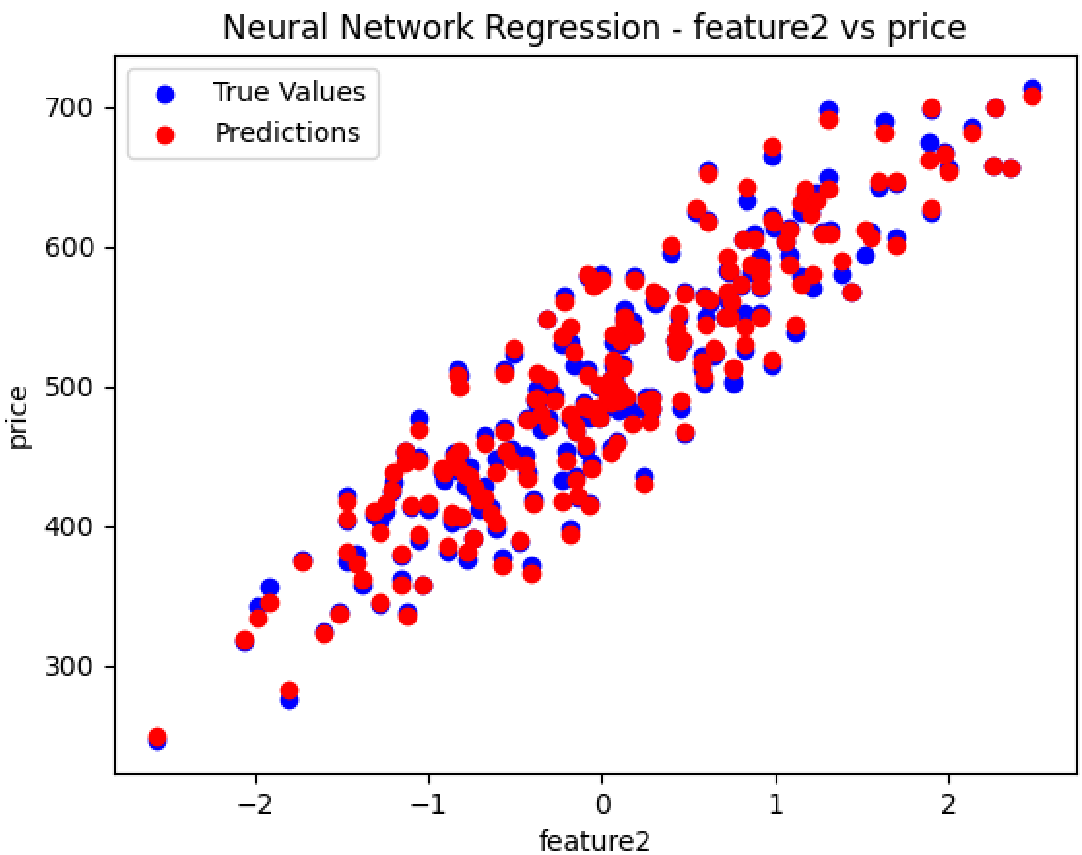
```
Iteration 943, loss = 12.04845523
Iteration 944, loss = 12.06593075
Iteration 945, loss = 12.06509896
Iteration 946, loss = 12.04223003
Iteration 947, loss = 12.04685290
Iteration 948, loss = 12.04076626
Iteration 949, loss = 12.04093753
Iteration 950, loss = 12.02961785
Iteration 951, loss = 12.03546856
Iteration 952, loss = 12.03359577
Iteration 953, loss = 12.03557247
Iteration 954, loss = 12.03681044
Iteration 955, loss = 12.02672107
Iteration 956, loss = 12.02500488
Iteration 957, loss = 12.02744439
Iteration 958, loss = 12.02856905
Iteration 959, loss = 12.03617253
Iteration 960, loss = 12.02087239
Iteration 961, loss = 12.02652999
Iteration 962, loss = 12.00909034
Iteration 963, loss = 12.00082703
Iteration 964, loss = 12.01238000
Iteration 965, loss = 12.00045847
Iteration 966, loss = 12.01610998
Iteration 967, loss = 12.00313883
Iteration 968, loss = 11.99265574
Iteration 969, loss = 12.01532735
Iteration 970, loss = 12.00805097
Iteration 971, loss = 12.01914730
Iteration 972, loss = 11.99467873
Iteration 973, loss = 11.98712103
Iteration 974, loss = 11.99077557
Iteration 975, loss = 11.99620781
Iteration 976, loss = 11.99023597
Iteration 977, loss = 12.02290562
Iteration 978, loss = 11.99750169
Iteration 979, loss = 11.99207492
Iteration 980, loss = 12.00676676
Iteration 981, loss = 12.02228869
Iteration 982, loss = 12.00075333
Iteration 983, loss = 11.98754963
Iteration 984, loss = 11.97595760
Iteration 985, loss = 11.97757728
Iteration 986, loss = 11.97924427
Iteration 987, loss = 11.98274545
Iteration 988, loss = 11.96365330
Iteration 989, loss = 11.96276812
Iteration 990, loss = 11.96157196
Iteration 991, loss = 11.97714887
```

```
Iteration 992, loss = 11.95161107
Iteration 993, loss = 11.97820865
Iteration 994, loss = 11.96349974
Iteration 995, loss = 11.96695710
Iteration 996, loss = 11.96198223
Iteration 997, loss = 11.95493759
Iteration 998, loss = 11.97699391
Iteration 999, loss = 11.95045140
Iteration 1000, loss = 11.95603471
Iteration 1001, loss = 11.95767189
Iteration 1002, loss = 11.93837739
Iteration 1003, loss = 11.94490632
Iteration 1004, loss = 11.94855388
Iteration 1005, loss = 11.93889354
Iteration 1006, loss = 11.94654739
Iteration 1007, loss = 11.93192139
Iteration 1008, loss = 11.93499755
Iteration 1009, loss = 11.93905208
Iteration 1010, loss = 11.93142842
Iteration 1011, loss = 11.93864682
Iteration 1012, loss = 11.93373677
Iteration 1013, loss = 11.92623433
Iteration 1014, loss = 11.91887489
Iteration 1015, loss = 11.92331539
Iteration 1016, loss = 11.92312636
Iteration 1017, loss = 11.91940804
Iteration 1018, loss = 11.92761101
Iteration 1019, loss = 11.92297517
Iteration 1020, loss = 11.91695818
Iteration 1021, loss = 11.91914838
Iteration 1022, loss = 11.94465575
Iteration 1023, loss = 11.95719062
Iteration 1024, loss = 11.91415362
Iteration 1025, loss = 11.93670609
Iteration 1026, loss = 11.91569604
Iteration 1027, loss = 11.90033059
Iteration 1028, loss = 11.90558868
Iteration 1029, loss = 11.91319874
Iteration 1030, loss = 11.91955327
Iteration 1031, loss = 11.90389053
Iteration 1032, loss = 11.90427080
Iteration 1033, loss = 11.89657151
Iteration 1034, loss = 11.90163773
Iteration 1035, loss = 11.89160908
Iteration 1036, loss = 11.92294742
Iteration 1037, loss = 11.89333508
Iteration 1038, loss = 11.90627367
Iteration 1039, loss = 11.90932942
Iteration 1040, loss = 11.89284074
Iteration 1041, loss = 11.89408846
```

```
Iteration 1042, loss = 11.91264320
Iteration 1043, loss = 11.90913432
Iteration 1044, loss = 11.91191373
Iteration 1045, loss = 11.90074428
Iteration 1046, loss = 11.90242843
Training loss did not improve more than tol=0.000100 for 10
consecutive epochs. Stopping.

y_pred = model.predict(X_test)
model.score(X_test, y_test) # R Squared

0.9966286345849324

plt.scatter(X_test[:,0], y_test, color='blue')
plt.scatter(X_test[:,0], y_pred, color='red')
# Set the axis labels and title
plt.xlabel('feature1')
plt.ylabel('price')
plt.title('Neural Network Regression - feature1 vs price')
# Add a legend to the plot
plt.legend(['True Values', 'Predictions'])
plt.show()
```



```
plt.scatter(X_test[:,1], y_test, color='blue')
plt.scatter(X_test[:,1], y_pred, color='red')
# Set the axis labels and title
plt.xlabel('feature2')
plt.ylabel('price')
plt.title('Neural Network Regression - feature2 vs price')
# Add a legend to the plot
plt.legend(['True Values', 'Predictions'])
plt.show()
```



Imprimindo métricas do modelo

```
from sklearn import metrics
import numpy as np
def print_evaluate(real, predicted):
    mae = metrics.mean_absolute_error(real, predicted)
    mape = metrics.mean_absolute_percentage_error(real, predicted)
    mse = metrics.mean_squared_error(real, predicted)
    rmse = np.sqrt(mse)
    r2_square = metrics.r2_score(real, predicted)
    print("MAE:", mae)
    print("MAPE:", mape)
```

```
    print("MSE:", mse)
    print("RMSE:", rmse)
    print("R2 Square:", r2_square)

print("Métricas de Desempenho (RNA):")
print("\nDados de teste:")
print_evaluate(y_test, y_pred)
```

```
Métricas de Desempenho (RNA):

Dados de teste:
MAE: 4.27513938553281
MAPE: 0.008729802073419278
MSE: 28.427624825648152
RMSE: 5.331756260900169
R2 Square: 0.9966286345849324
```

Comparando com Regressor Linear

```
from sklearn.linear_model import LinearRegression
# regressão (treinamento)
my_model = LinearRegression()
my_model.fit(X_train, y_train)
print("R2 (test) =", my_model.score(X_test, y_test)) # r2 squared
```

```
R2 (test) = 0.9969271004939649
```

```
#regressão (teste)
y_pred = my_model.predict(X_test)
print("Métricas de Desempenho (Linear):")
print("\nDados de teste:")
print_evaluate(y_test, y_pred)
```

```
Métricas de Desempenho (Linear):

Dados de teste:
MAE: 4.059737026111618
MAPE: 0.008309205902040373
MSE: 25.910936231970744
RMSE: 5.0902786006240115
R2 Square: 0.9969271004939649
```

Comparando com SVM

```
from sklearn.svm import SVR
svr = SVR(C=30.0, epsilon=0.5)
svr.fit(X_train, y_train)
svr.score(X_test, y_test)
#regressão (teste)
y_pred = svr.predict(X_test)
```

```python
print("Métricas de Desempenho (Linear):")
print("\nDados de teste:")
print_evaluate(y_test, y_pred)
```

```
Métricas de Desempenho (Linear):

Dados de teste:
MAE: 4.802595762419471
MAPE: 0.010398254884641956
MSE: 56.17818954367445
RMSE: 7.4952111073454395
R2 Square: 0.9933375649049029
```

Ressultados

Observando os resultados, vê-se que eles são muito parecidos nos três métodos. Uma possiblidade pode ser que a base de dados não é tão complexa de uma maneiera de necessitar de diferentes abordagens.