Proj1_resolvido

March 19, 2025

1 Projeto de Regressão Linear

Parabéns! Foi contratado para trabalhar numa empresa de e-commerce sediada em Nova York, que vende roupa online, mas também tem consultoria em estilo e vestuário na loja. Os clientes entram na loja e são aconselhados por um estilista pessoal. Voltam para casa e encomendam a roupa pretendida através do site ou da app.

A empresa quer decidir se deve investir na app ou no site.

Siga as etapas para analisar os dados do cliente.

1.1 Imports

** Importe pandas, numpy, matplotlib e seaborn. **

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

1.2 Obter dados

Trabalharemos com o ficheiro EcommerceCustomers. Possui informações do cliente, como Email, Endereço e sua cor Avatar. Também possui colunas de valores numéricos:.

- Avg. Session Length: Tempo médio das sessões de consultoria de estilo na loja.
- Time on App: tempo médio gasto no app em minutos.
- Time on Website: tempo médio gasto no site em minutos.
- Lenght of Membership: Há quantos anos o cliente é membro.

** Ler o ficheiro EcommerceCustomers para um DataFrame chamado clientes. **

[2]:
```
clientes = pd.read_csv('EcommerceCustomers.csv')
```

** Verifique o cabeçalho dos clientes e confira os seus métodos info () e describe(). **

[3]:
```
clientes.head()
```

[3]:
```
                          Email  \
0      mstephenson@fernandez.com
1              hduke@hotmail.com
```

```
2              pallen@yahoo.com
3       riverarebecca@gmail.com
4  mstephens@davidson-herman.com

                                              Address            Avatar  \
0       835 Frank Tunnel\nWrightmouth, MI 82180-9605             Violet
1     4547 Archer Common\nDiazchester, CA 06566-8576          DarkGreen
2  24645 Valerie Unions Suite 582\nCobbborough, D…             Bisque
3   1414 David Throughway\nPort Jason, OH 22070-1220        SaddleBrown
4  14023 Rodriguez Passage\nPort Jacobville, PR 3…  MediumAquaMarine

   Avg. Session Length  Time on App  Time on Website  Length of Membership  \
0            34.497268    12.655651        39.577668              4.082621
1            31.926272    11.109461        37.268959              2.664034
2            33.000915    11.330278        37.110597              4.104543
3            34.305557    13.717514        36.721283              3.120179
4            33.330673    12.795189        37.536653              4.446308

   Yearly Amount Spent
0           587.951054
1           392.204933
2           487.547505
3           581.852344
4           599.406092
```

[4]:
```
clientes.describe()
```

[4]:
```
       Avg. Session Length  Time on App  Time on Website  \
count           500.000000   500.000000       500.000000
mean             33.053194    12.052488        37.060445
std               0.992563     0.994216         1.010489
min              29.532429     8.508152        33.913847
25%              32.341822    11.388153        36.349257
50%              33.082008    11.983231        37.069367
75%              33.711985    12.753850        37.716432
max              36.139662    15.126994        40.005182

       Length of Membership  Yearly Amount Spent
count            500.000000           500.000000
mean               3.533462           499.314038
std                0.999278            79.314782
min                0.269901           256.670582
25%                2.930450           445.038277
50%                3.533975           498.887875
75%                4.126502           549.313828
max                6.922689           765.518462
```

[5]:
```
clientes.info()
```

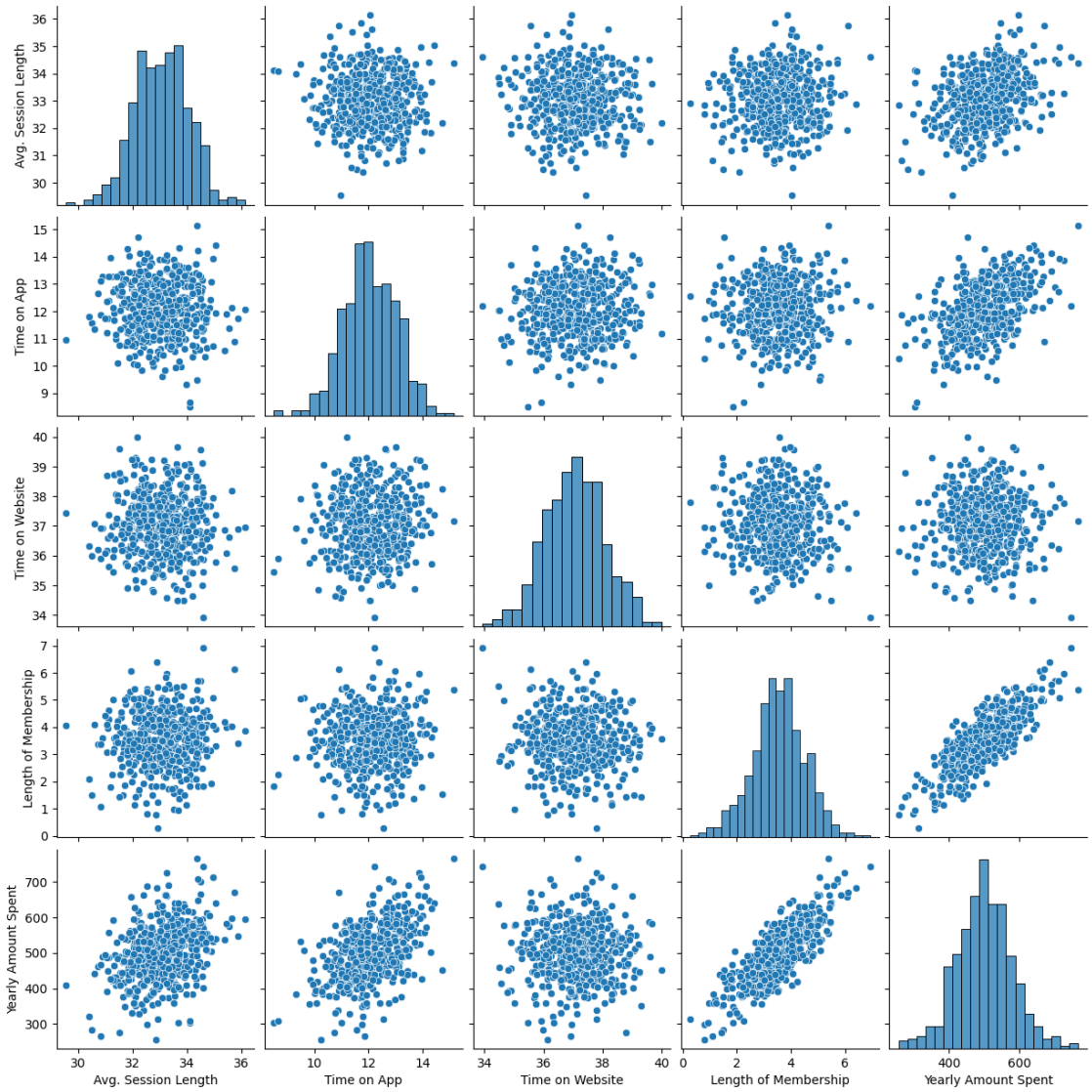
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Email                 500 non-null    object 
 1   Address               500 non-null    object 
 2   Avatar                500 non-null    object 
 3   Avg. Session Length   500 non-null    float64
 4   Time on App           500 non-null    float64
 5   Time on Website       500 non-null    float64
 6   Length of Membership  500 non-null    float64
 7   Yearly Amount Spent   500 non-null    float64
dtypes: float64(5), object(3)
memory usage: 31.4+ KB
```

1.3 Análise de dados exploratória

** Use seaborn para criar um pairplot e comparar as relações no conjunto de dados **

[6]:
```
sns.pairplot(clientes)
```

[6]:
```
<seaborn.axisgrid.PairGrid at 0x135c948e0>
```



** Neste gráfico, o que parece ser a caraterística mais correlacionada com o valor anual gasto (Yearly Amount Spent)? **

[7]:
```
# Length of Membership
```

1.4 Treinar e testar os dados

Agora que explorámos um pouco os dados, vamos avançar e dividir os dados em conjuntos de treino e teste. ** Defina uma variável X igual a todas as caraterísticas numéricas dos clientes e uma variável Y igual à coluna Valor Anual Gasto (Yearly Amount Spent). **

[8]:
```
X = clientes[['Avg. Session Length', 'Time on App', 'Time on Website',
              'Length of Membership']]
```

[9]:
```
Y = clientes['Yearly Amount Spent']
```

** Use train_test_split da sklearn para dividir os dados em conjuntos de treino e teste. Defina test_size = 0.3 e random_state = 101 **

[10]:
```
from sklearn.model_selection import train_test_split
```

[11]:
```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3,␣
↪random_state=101)
```

1.5 Treinar o modelo

** Importe LinearRegression do sklearn.linear_model **

[12]:
```
from sklearn.linear_model import LinearRegression
```

** Crie uma instância de um modelo LinearRegression () chamado lm. **

[13]:
```
lm = LinearRegression()
```

** Treine lm nos dados de treino. **

[14]:
```
lm.fit(X_train,y_train)
```

[14]:
```
LinearRegression()
```

Faça um Print dos coeficientes do modelo

[15]:
```
# Print da interceção
print(lm.intercept_)
```

```
-1047.932782250239
```

[16]:
```
#Print dos coeficientes
print(lm.coef_)
```

```
[25.98154972 38.59015875  0.19040528 61.27909654]
```

1.6 Previsão de dados de teste

Agora que nos ajustámos ao modelo, vamos avaliar o seu desempenho ao prever os valores de teste!

** Use lm.predict() para prever o conjunto X_test dos dados. **

[17]:
```
predictions = lm.predict(X_test)
predictions
```

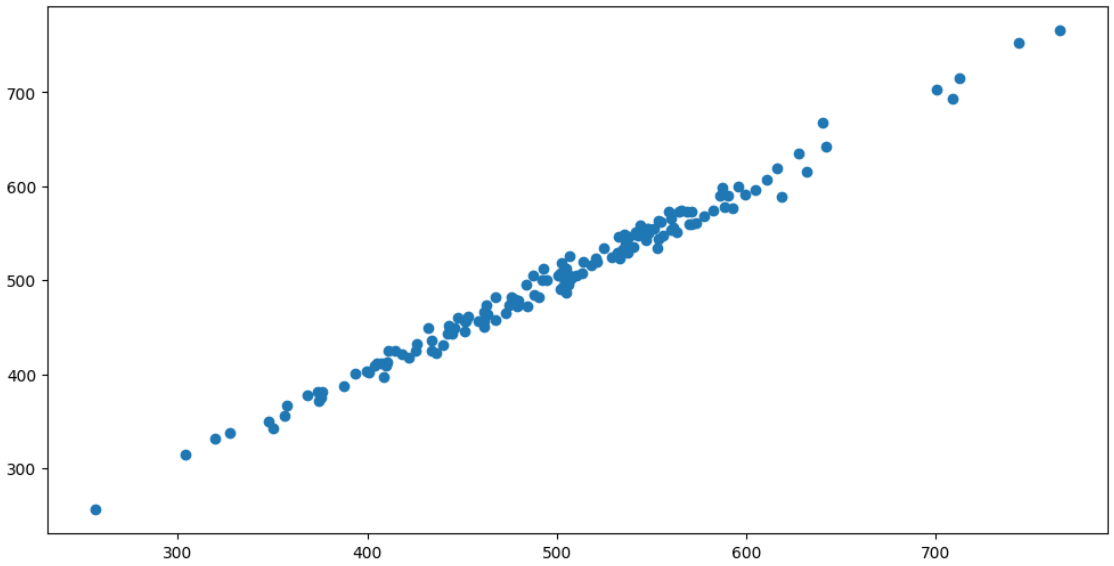
[17]:
```
array([456.44186104, 402.72005312, 409.2531539 , 591.4310343 ,
       590.01437275, 548.82396607, 577.59737969, 715.44428115,
       473.7893446 , 545.9211364 , 337.8580314 , 500.38506697,
       552.93478041, 409.6038964 , 765.52590754, 545.83973731,
       693.25969124, 507.32416226, 573.10533175, 573.2076631 ,
```

```
397.44989709, 555.0985107 , 458.19868141, 482.66899911,
559.2655959 , 413.00946082, 532.25727408, 377.65464817,
535.0209653 , 447.80070905, 595.54339577, 667.14347072,
511.96042791, 573.30433971, 505.02260887, 565.30254655,
460.38785393, 449.74727868, 422.87193429, 456.55615271,
598.10493696, 449.64517443, 615.34948995, 511.88078685,
504.37568058, 515.95249276, 568.64597718, 551.61444684,
356.5552241 , 464.9759817 , 481.66007708, 534.2220025 ,
256.28674001, 505.30810714, 520.01844434, 315.0298707 ,
501.98080155, 387.03842642, 472.97419543, 432.8704675 ,
539.79082198, 590.03070739, 752.86997652, 558.27858232,
523.71988382, 431.77690078, 425.38411902, 518.75571466,
641.9667215 , 481.84855126, 549.69830187, 380.93738919,
555.18178277, 403.43054276, 472.52458887, 501.82927633,
473.5561656 , 456.76720365, 554.74980563, 702.96835044,
534.68884588, 619.18843136, 500.11974127, 559.43899225,
574.8730604 , 505.09183544, 529.9537559 , 479.20749452,
424.78407899, 452.20986599, 525.74178343, 556.60674724,
425.7142882 , 588.8473985 , 490.77053065, 562.56866231,
495.75782933, 445.17937217, 456.64011682, 537.98437395,
367.06451757, 421.12767301, 551.59651363, 528.26019754,
493.47639211, 495.28105313, 519.81827269, 461.15666582,
528.8711677 , 442.89818166, 543.20201646, 350.07871481,
401.49148567, 606.87291134, 577.04816561, 524.50431281,
554.11225704, 507.93347015, 505.35674292, 371.65146821,
342.37232987, 634.43998975, 523.46931378, 532.7831345 ,
574.59948331, 435.57455636, 599.92586678, 487.24017405,
457.66383406, 425.25959495, 331.81731213, 443.70458331,
563.47279005, 466.14764208, 463.51837671, 381.29445432,
411.88795623, 473.48087683, 573.31745784, 417.55430913,
543.50149858, 547.81091537, 547.62977348, 450.99057409,
561.50896321, 478.30076589, 484.41029555, 457.59099941,
411.52657592, 375.47900638])
```

** Crie um diagrama de dispersão (scatterplot) dos valores reais de teste em relação aos valores preditos. **

[18]:
```
plt.figure(figsize=(12, 6))
plt.scatter(y_test,predictions)
```

[18]:
```
<matplotlib.collections.PathCollection at 0x145389940>
```



1.7 Avaliar o Modelo

Vamos avaliar o desempenho do nosso modelo.

** Calcule o erro absoluto médio, o erro quadrado médio e o erro quadrado médio da raiz. **

[19]:
```
from sklearn import metrics

print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

```
MAE: 7.2281486534308215
MSE: 79.81305165097433
RMSE: 8.933815066978626
```

1.8 Conclusão

Ainda desejamos descobrir a resposta à pergunta original. Concentramos-nos no desenvolvimento de apps móveis ou de sites? Ou talvez isto realmente não importe e o tempo como membro é o que é realmente importante? Vamos ver se podemos interpretar os coeficientes para ter uma idéia.

** Recrie o quadro de dados abaixo. **

[20]:
```
coeff_df = pd.DataFrame(lm.coef_,X.columns,columns=['Coefficient'])
coeff_df
```

[20]:

	Coefficient
Avg. Session Length	25.981550
Time on App	38.590159

```
Time on Website          0.190405
Length of Membership    61.279097
```

** Como é que pode interpretar estes coeficientes? **

[ ]:
```
"""
Interpretacao dos coeficientes:

- Mantendo todas as outras variáveis constantes, um aumento de 1 unidade em **Avg. Session Length ** está associado a um **aumento de $ 25,98**.
- Mantendo todas as outras variáveis constantes, um aumento de 1 unidade em **Time on App ** está associado a um **aumento de $ 38,59**.
- Mantendo todas as outras variáveis constantes, um aumento de 1 unidade em **Time onWebsite ** está associado a um **aumento de $ 0,19**.
- Mantendo todas as outras variáveis constantes, um aumento de 1 unidade em **Length of Membership ** está associado a um **aumento de $ 61,28**.
"""
```

** Acha que a empresa se deve concentrar mais na sua App ou no site? **

[ ]:
```
# Deve concentrar-se na App.
```