In [1]:
```
import pyforest
```

In [2]:
```
customers = pd.read_csv('Ecommerce Customers.csv')
```

In [3]:
```
customers
```

Out[3]:

	Email	Address	Avatar	Avg. Session Length	Time on App	Time on Website	L Mer
0	mstephenson@fernandez.com	835 Frank Tunnel\nWrightmouth, MI 82180-9605	Violet	34.497268	12.655651	39.577668	
1	hduke@hotmail.com	4547 Archer Common\nDiazchester, CA 06566-8576	DarkGreen	31.926272	11.109461	37.268959	
2	pallen@yahoo.com	24645 Valerie Unions Suite 582\nCobbborough, D...	Bisque	33.000915	11.330278	37.110597	
3	riverarebecca@gmail.com	1414 David Throughway\nPort Jason, OH 22070-1220	SaddleBrown	34.305557	13.717514	36.721283	
4	mstephens@davidson-herman.com	14023 Rodriguez Passage\nPort Jacobville, PR 3...	MediumAquaMarine	33.330673	12.795189	37.536653	
...	...	...	...	...	...	...	
495	lewisjessica@craig-evans.com	4483 Jones Motorway Suite 872\nLake Jamiefurt,...	Tan	33.237660	13.566160	36.417985	
496	katrina56@gmail.com	172 Owen Divide Suite 497\nWest Richard, CA 19320	PaleVioletRed	34.702529	11.695736	37.190268	
497	dale88@hotmail.com	0787 Andrews Ranch Apt. 633\nSouth Chadburgh, ...	Cornsilk	32.646777	11.499409	38.332576	
498	cwilson@hotmail.com	680 Jennifer Lodge Apt. 808\nBrendachester, TX...	Teal	33.322501	12.391423	36.840086	
499	hannahwilson@davidson.com	49791 Rachel Heights Apt. 898\nEast Drewboroug...	DarkMagenta	33.715981	12.418808	35.771016	

500 rows × 8 columns

In [4]:
```
cust_time_money = customers[['Avg. Session Length', 'Time on App', 'Time on Website', 'L
```

In [5]:
```
cust_time_money
```

Out[5]:

	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
0	34.497268	12.655651	39.577668	4.082621	587.951054

1	31.926272	11.109461	37.268959	2.664034	392.204933
2	33.000915	11.330278	37.110597	4.104543	487.547505
3	34.305557	13.717514	36.721283	3.120179	581.852344
4	33.330673	12.795189	37.536653	4.446308	599.406092
...	...	...	...	...	...
495	33.237660	13.566160	36.417985	3.746573	573.847438
496	34.702529	11.695736	37.190268	3.576526	529.049004
497	32.646777	11.499409	38.332576	4.958264	551.620145
498	33.322501	12.391423	36.840086	2.336485	456.469510
499	33.715981	12.418808	35.771016	2.735160	497.778642

500 rows × 5 columns

In [6]:
```
customers.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
```

In [7]:
```
from sklearn.preprocessing import MinMaxScaler
mm = MinMaxScaler()
```

In [8]:
```
cust_time_money.iloc[:,:] = mm.fit_transform(cust_time_money[['Avg. Session Length', 'Ti
```

```
C:\Users\Aarav\AppData\Local\Temp\ipykernel_19452\1892120098.py:1: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  cust_time_money.iloc[:,:] = mm.fit_transform(cust_time_money[['Avg. Session Length',
'Time on App', 'Time on Website', 'Length of Membership', 'Yearly Amount Spent']])
```

In [9]:
```
cust_time_money
```

Out[9]:

	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
0	0.751425	0.626620	0.929816	0.573101	0.651040
1	0.362306	0.393016	0.550801	0.359869	0.266355
2	0.524953	0.426378	0.524803	0.576396	0.453725
3	0.722409	0.787050	0.460890	0.428434	0.639055
4	0.574861	0.647702	0.594748	0.627768	0.673552

...	...	...	...	...	...
495	0.560784	0.764183	0.411098	0.522589	0.623324
496	0.782491	0.481592	0.537882	0.497028	0.535285
497	0.471354	0.451931	0.725412	0.704722	0.579642
498	0.573625	0.586699	0.480394	0.310634	0.392650
499	0.633178	0.590837	0.304887	0.370560	0.473831

500 rows × 5 columns

In [10]:
```
X = cust_time_money[['Avg. Session Length', 'Time on App', 'Time on Website', 'Length of
Y = cust_time_money.iloc[:,4]
```

In [11]:
```
X
```

Out[11]:

	Avg. Session Length	Time on App	Time on Website	Length of Membership
0	0.751425	0.626620	0.929816	0.573101
1	0.362306	0.393016	0.550801	0.359869
2	0.524953	0.426378	0.524803	0.576396
3	0.722409	0.787050	0.460890	0.428434
4	0.574861	0.647702	0.594748	0.627768
...	...	...	...	...
495	0.560784	0.764183	0.411098	0.522589
496	0.782491	0.481592	0.537882	0.497028
497	0.471354	0.451931	0.725412	0.704722
498	0.573625	0.586699	0.480394	0.310634
499	0.633178	0.590837	0.304887	0.370560

500 rows × 4 columns

In [12]:
```
Y
```

Out[12]:
```
0      0.651040
1      0.266355
2      0.453725
3      0.639055
4      0.673552
         ...
495    0.623324
496    0.535285
497    0.579642
498    0.392650
499    0.473831
Name: Yearly Amount Spent, Length: 500, dtype: float64
```

In [13]:
```
from sklearn.model_selection import train_test_split
```

In [14]:
```
x_train, x_test, y_train, y_test = train_test_split(X, Y, random_state=101)
```

In [15]:
```
from sklearn.linear_model import LinearRegression
```

```
model = LinearRegression()
```

In [16]:
```
model.fit(x_train, y_train)
```

Out[16]:
```
LinearRegression()
```

In [17]:
```
model.coef_
```

Out[17]:
```
array([0.33850379, 0.50307265, 0.0030146 , 0.80158779])
```

In [18]:
```
model.intercept_
```

Out[18]:
```
-0.36731263767622496
```

In [19]:
```
y_predicted = model.predict(x_test)
```

In [20]:
```
from sklearn.metrics import r2_score
```

In [21]:
```
r2_score(y_test,y_predicted)
```

Out[21]:
```
0.9889411290595322
```

In [ ]:
```

```