df= pd.read_csv('Ecommerce Customers') In [73]: df.head() Out[73]: Length of **Yearly Amount** Avg. Session Time on Time on **Email Address Avatar** Membership Website Length App Spent 835 Frank Tunnel\nWrightmouth, MI mstephenson@fernandez.com 34.497268 4.082621 587.951054 Violet 12.655651 39.577668 0 82180-9605 4547 Archer Common\nDiazchester, CA hduke@hotmail.com 1 DarkGreen 31.926272 11.109461 37.268959 2.664034 392.204933 06566-8576 24645 Valerie Unions Suite 2 pallen@yahoo.com Bisque 33.000915 11.330278 37.110597 4.104543 487.547505 582\nCobbborough, D... 1414 David Throughway\nPort Jason, OH riverarebecca@gmail.com 3 SaddleBrown 34.305557 13.717514 581.852344 36.721283 3.120179 22070-1220 14023 Rodriguez Passage\nPort Jacobville, PR 3... mstephens@davidson-MediumAquaMarine 33.330673 4 12.795189 37.536653 4.446308 599.406092 herman.com In [74]: df.info() <class 'pandas.core.frame.DataFrame'> RangeIndex: 500 entries, 0 to 499 Data columns (total 8 columns): Column Non-Null Count Dtype **Email** 500 non-null object 500 non-null Address object 500 non-null Avatar object Avg. Session Length 500 non-null float64 Time on App 500 non-null float64 Time on Website 500 non-null float64 Length of Membership float64 500 non-null Yearly Amount Spent 500 non-null float64 dtypes: float64(5), object(3) memory usage: 31.4+ KB df.describe() In [75]: Out[75]: Avg. Session Length Time on App Time on Website Length of Membership Yearly Amount Spent 500.000000 500.000000 500.000000 500.000000 500.000000 count 33.053194 12.052488 37.060445 3.533462 499.314038 mean std 0.992563 0.994216 1.010489 0.999278 79.314782 min 29.532429 33.913847 256.670582 8.508152 0.269901 25% 32.341822 11.388153 36.349257 2.930450 445.038277 11.983231 50% 33.082008 37.069367 3.533975 498.887875 **75**% 33.711985 12.753850 37.716432 4.126502 549.313828 15.126994 765.518462 36.139662 40.005182 6.922689 max sns.pairplot(df) Out[76]: <seaborn.axisgrid.PairGrid at 0x1b1a4f1ab60>

In [72]:

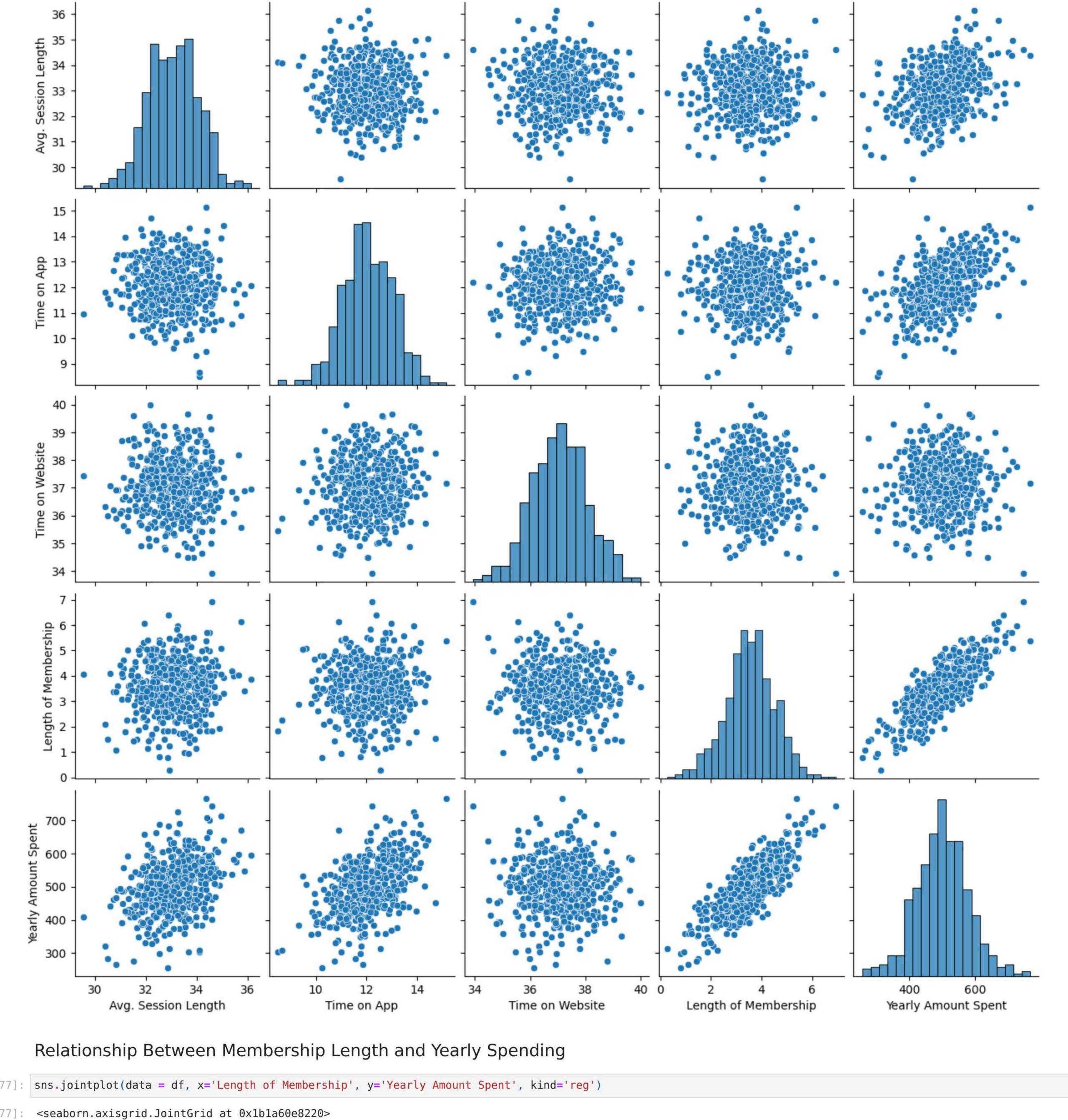
import numpy as np

import pandas as pd

%matplotlib inline

import seaborn as sns

import matplotlib.pyplot as plt



700 -

In [80]: lm.intercept_

Out[81]:

-1047.9327822502391

Interpreting Model Coefficients

Time on App 38.590159

Avg. Session Length 25.981550

Time on Website

Predicting Test Data

Length of Membership 61.279097

predictions = lm.predict(x_test)

pd.DataFrame(lm.coef_ , x.columns, columns=['coef'])

coef

0.190405

Yearly Amount Spent 400 300 Length of Membership Splitting the Dataset into Training and Testing Sets • Test Size = 0.3 Random State = 101 In [78]: from sklearn.model_selection import train_test_split x = df[['Avg. Session Length', 'Time on App', 'Time on Website', 'Length of Membership']]y = df['Yearly Amount Spent'] x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=101) from sklearn.linear_model import LinearRegression lm = LinearRegression() lm.fit(x_train, y_train) Out[79]: LinearRegression 🕒 🥛 LinearRegression()

plt.scatter(y_test, predictions) <matplotlib.collections.PathCollection at 0x1b1a6c3fa60> 700 600 500 400

300 600 700 400 500 300 The scatter plot shows a strong positive linear relationship between actual and predicted values, indicating that the model provides a good fit and accurately captures the patterns in the data. sns.histplot(y_test-predictions, kde=True, bins=50) <Axes: xlabel='Yearly Amount Spent', ylabel='Count'> 6 Count -10

Yearly Amount Spent The histogram of the residuals shows a roughly normal distribution, indicating that the residuals are symmetrically spread around zero.

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

The model's Mean Absolute Error is 7.23, Mean Squared Error is 79.81, and Root Mean Squared Error is 8.93, meaning the predictions are off by about 7.23 units on average, with a

MSE : 79.81305165097385 RMSE : 8.9338150669786	MAE	:	7.228148653430811	
RMSE: 8.9338150669786	MSE	:	79.81305165097385	
	RMSE	:	8.9338150669786	

MAE = metrics.mean_absolute_error(y_test, predictions)

Model Evaluation

'\nMSE : ', MSE,

'\nRMSE : ', RMSE)

typical error of 8.93 units.

In [85]: **from** sklearn **import** metrics