

Linear Regression Training project: Ecommerce Client

sources

GitHub: <https://github.com/alejandro-ao/py-ecommerce-spending-predictions/blob/main/02%20Linear%20Regression%20-%20E-commerce%20clients.ipynb>

Youtube: <https://www.youtube.com/watch?v=O2Cw82YR5Bo>

```
In [30]: import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
import statsmodels.api as sm
```

Getting the Data

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

```
In [3]: customers.head()
```

Out[3]:

	Email	Address	Avatar	Avg. Session Length	Tim
0	mstephenson@fernandez.com	835 Frank Tunnel\nWrightmouth, MI 82180-9605	Violet	34.497268	12.65!
1	hduke@hotmail.com	4547 Archer Common\nDiazchester, CA 06566-8576	DarkGreen	31.926272	11.10!
2	pallen@yahoo.com	24645 Valerie Unions Suite 582\nCobbborough, D...	Bisque	33.000915	11.33!
3	riverarebecca@gmail.com	1414 David Throughway\nPort Jason, OH 22070-1220	SaddleBrown	34.305557	13.71!
4	mstephens@davidson-herman.com	14023 Rodriguez Passage\nPort Jacobville, PR 3...	MediumAquaMarine	33.330673	12.79!



In [4]: `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 [5]: `customers.describe()`

Out[5]:	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
count	500.000000	500.000000	500.000000	500.000000	500.000000
mean	33.053194	12.052488	37.060445	3.533462	499.314038
std	0.992563	0.994216	1.010489	0.999278	79.314782
min	29.532429	8.508152	33.913847	0.269901	256.670582
25%	32.341822	11.388153	36.349257	2.930450	445.038277
50%	33.082008	11.983231	37.069367	3.533975	498.887875
75%	33.711985	12.753850	37.716432	4.126502	549.313828
max	36.139662	15.126994	40.005182	6.922689	765.518462

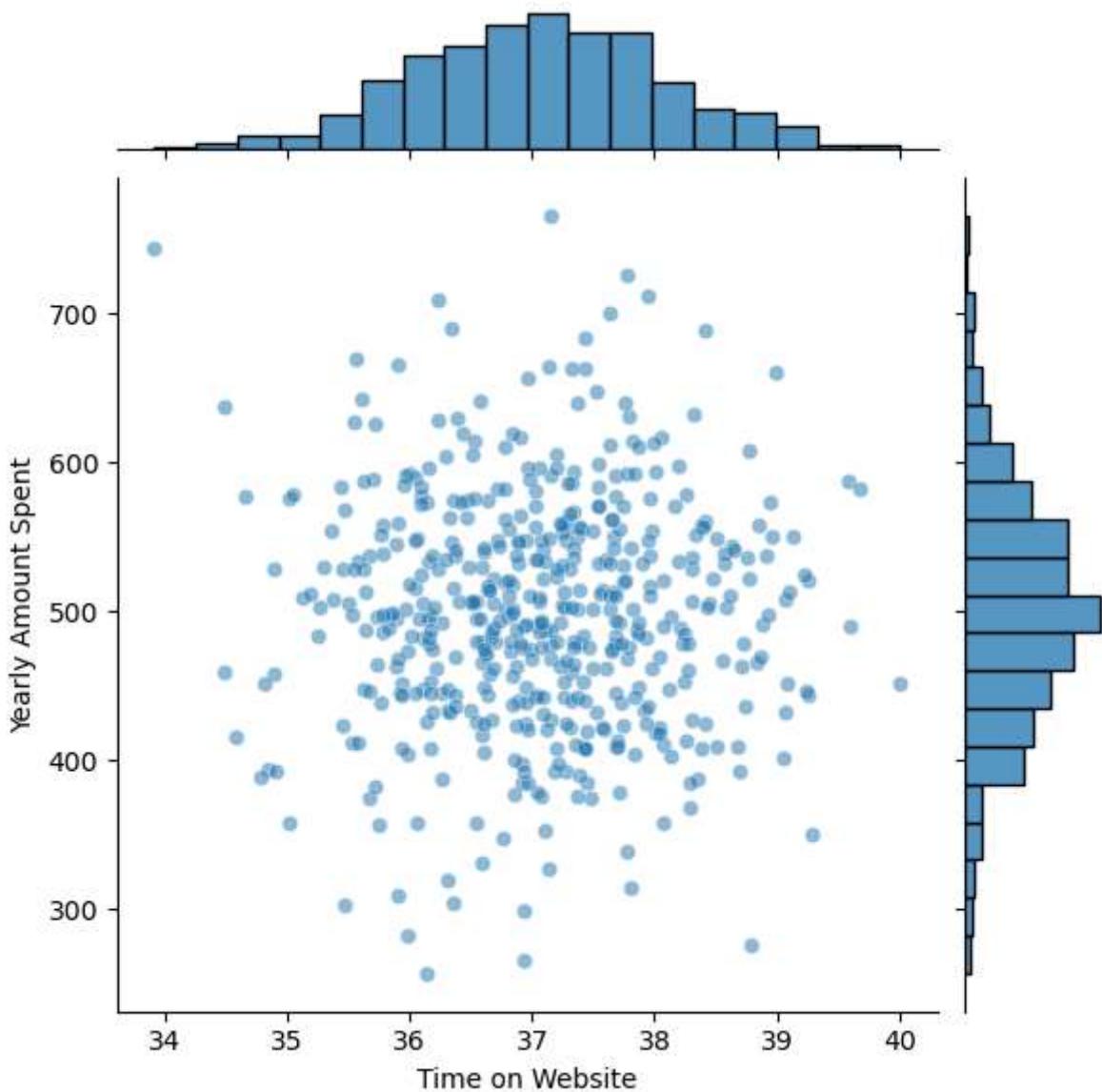
Exploratiivinen data-analyysi

Ensimmäiseksi kysymme seuraavan kysymyksen: Miten asiakkaan aika kullakin alustalla suhteutuu siihen, kuinka paljon he kuluttavat vuodessa? Näyttää siltä, että työpötäverkkosivustolla vietetyn ajan ja vuosittaisen kulutuksen välillä ei ole merkittävää korrelaatiota. Toisessa kuvassa puolestaan näyttää olevan pieni korrelaatio sovelluksessa vietetyn ajan ja vuosittaisen kulutuksen välillä. Tämä johtuu luultavasti siitä, että nämä asiakkaat viettävät vähemmän aikaa selailun puhelimella. Ehkä maksuprosessi on sovelluksessa nopeampi tai toimintakehotukset ovat siellä tehokkaampia.

Pariplotin analysoinnin jälkeen havaitsemme, että kahden muuttujan välillä on yksi merkittävä positiivinen korrelaatio: jäsenyyden pituus ja vuosittaiset kulutukset. Lopuksi luomme uudelleen tämän kuvan visualisoidaksemme regressiosuoran.

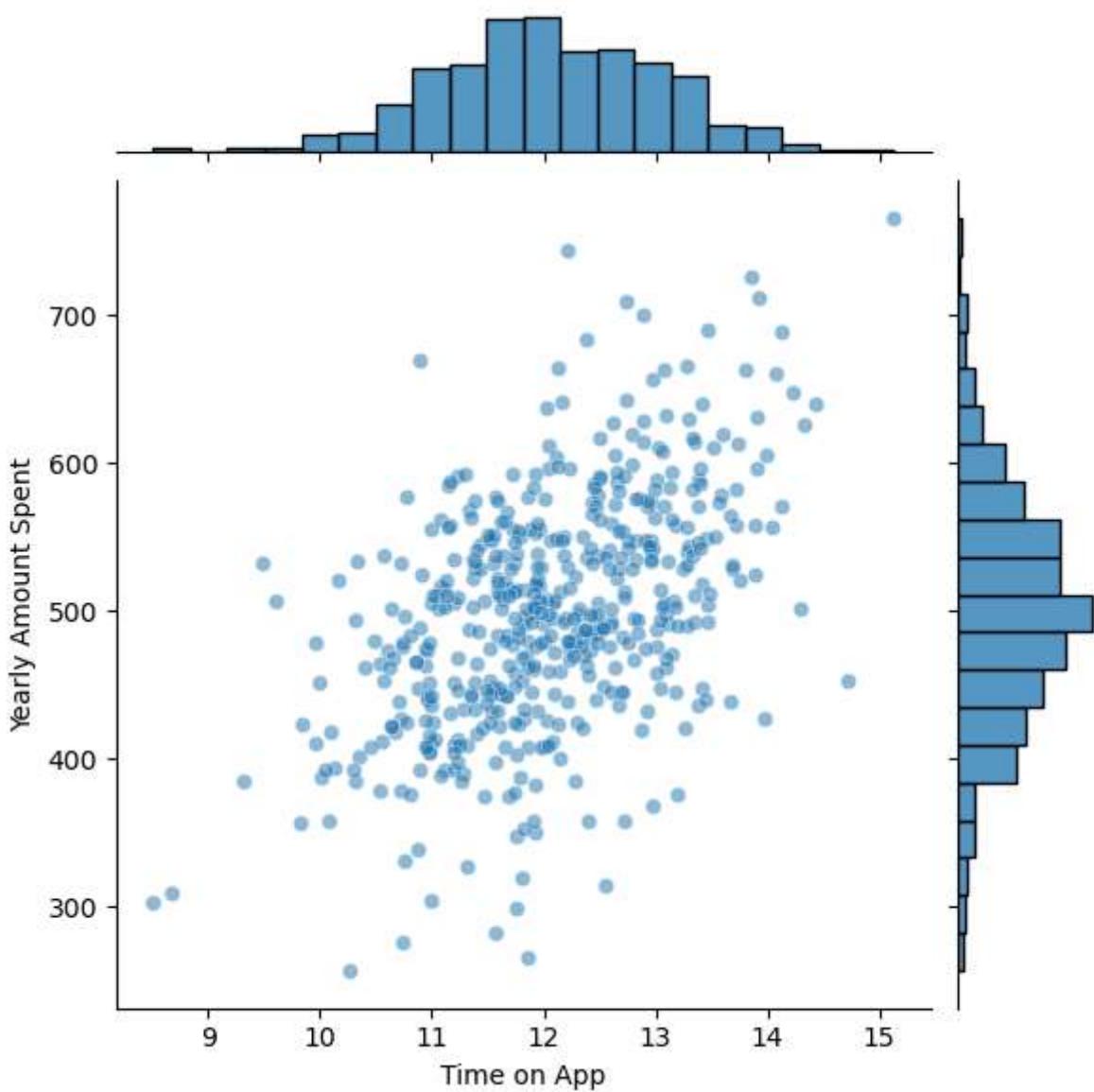
```
In [6]: # time on website vs yearly amount spent  
sns.jointplot(x='Time on Website', y='Yearly Amount Spent', data=customers, alpha=0)
```

```
Out[6]: <seaborn.axisgrid.JointGrid at 0x7f47dcb39f90>
```



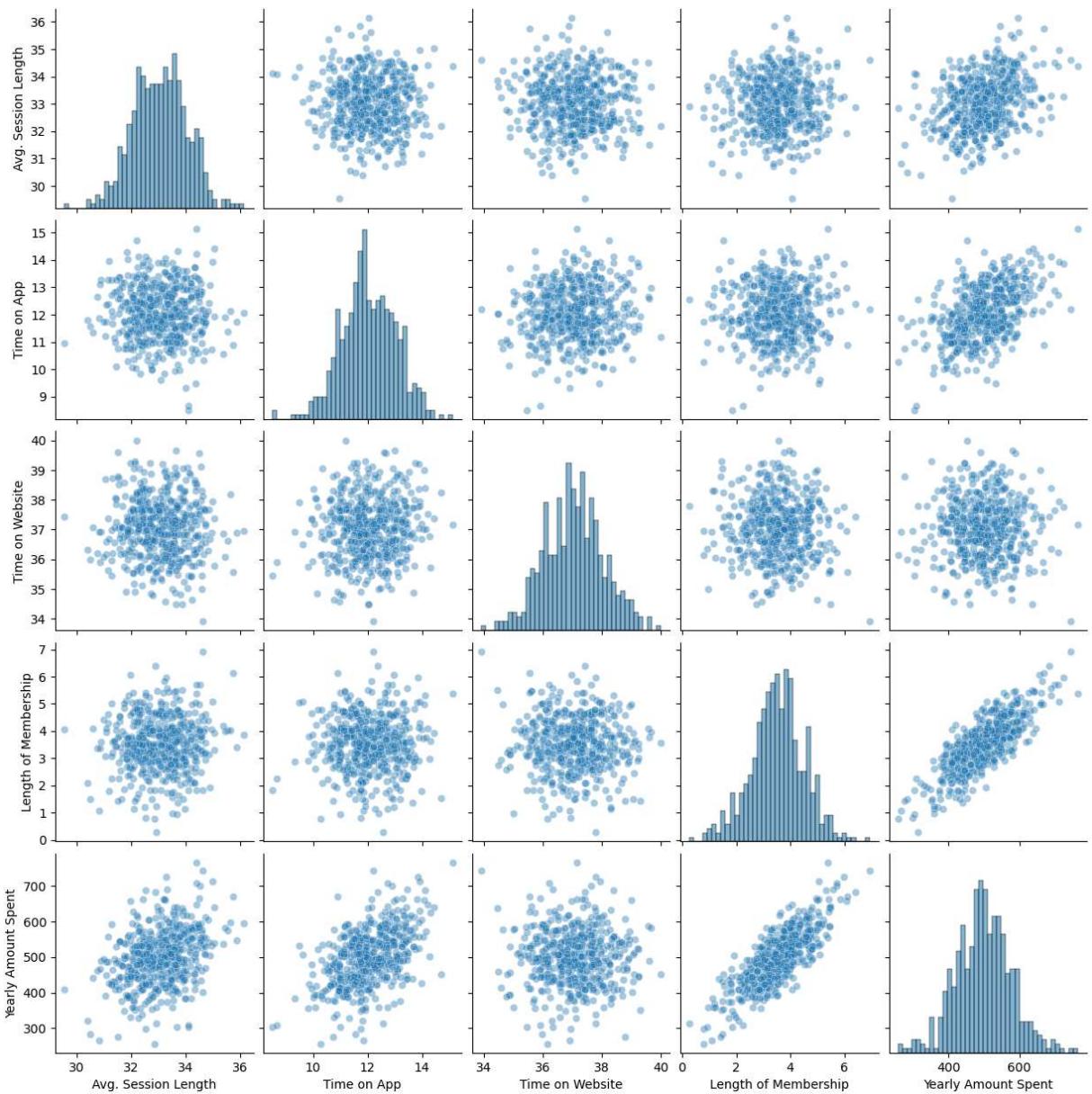
```
In [7]: # time on app vs yearly amount spent  
sns.jointplot(x='Time on App', y='Yearly Amount Spent', data=customers, alpha=0.5)
```

```
Out[7]: <seaborn.axisgrid.JointGrid at 0x7f48ce769a90>
```



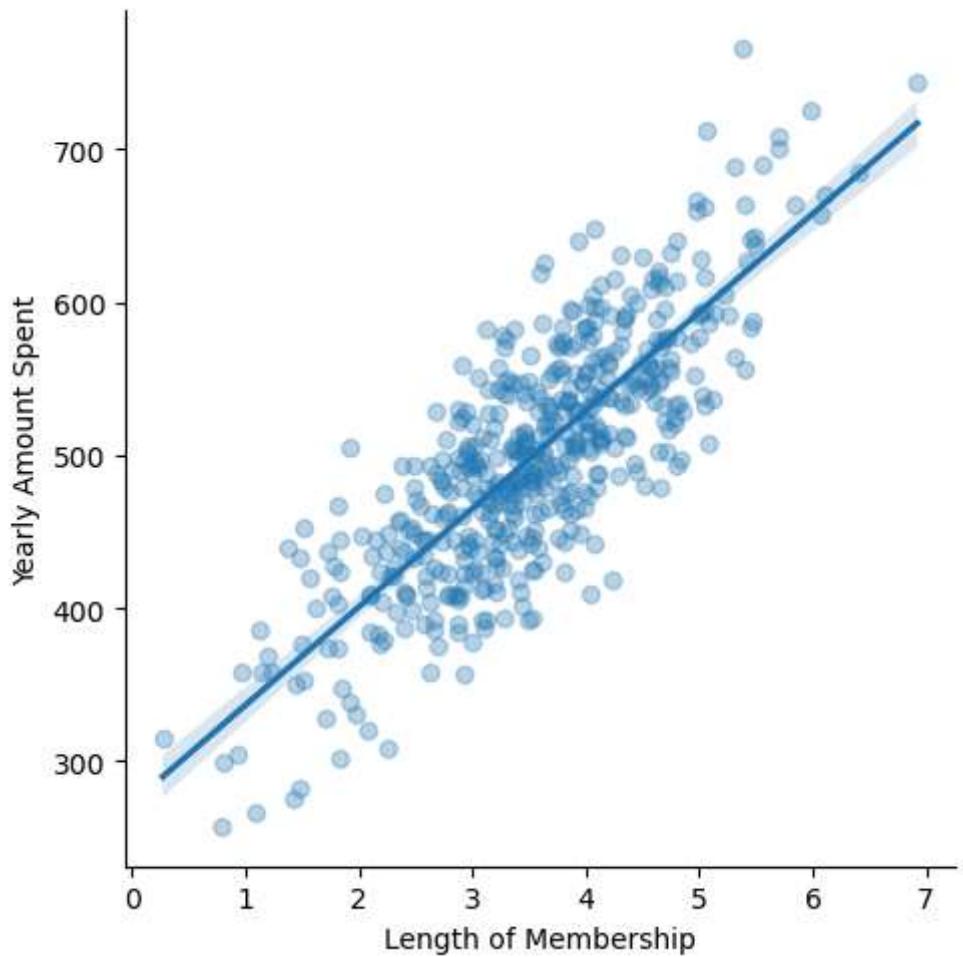
```
In [8]: sns.pairplot(customers,
                    kind='scatter',
                    plot_kws={'alpha':0.4},
                    diag_kws={'alpha':0.55, 'bins':40})
```

```
Out[8]: <seaborn.axisgrid.PairGrid at 0x7f47dca70050>
```



```
In [9]: # Length of membership vs yearly amount spent
sns.lmplot(x='Length of Membership',
            y='Yearly Amount Spent',
            data=customers,
            scatter_kws={'alpha':0.3})
```

Out[9]: <seaborn.axisgrid.FacetGrid at 0x7f47d1c31690>



Datan jakaminen

X ovat ennustajat, ja y on tulosmuuttuja. Tavoitteemamme on luoda malli, joka ottaa X-muuttujien arvot ja ennustaa y:n lineaarisen regressioalgoritmin avulla. Käytämme SciKit Learn -kirjastoa mallin luomiseen.

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

```
In [11]: 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 [12]: X = customers[['Avg. Session Length', 'Time on App', 'Time on Website', 'Length of Membership']]
y = customers['Yearly Amount Spent']
```

```
In [13]: X.head()
y.head()
```

```
Out[13]: 0    587.951054
1    392.204933
2    487.547505
3    581.852344
4    599.406092
Name: Yearly Amount Spent, dtype: float64
```

```
In [14]: X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3, random_state=42)
```

Mallin kouluttaminen monimuuttujaregressiolla käyttäen Scikit Learnia

Tässä osiossa luomme mallin ja syötämme sille harjoitusdatan. Tämä malli kertoo meille, mikä syöte vaikuttaa eniten tulokseen (vuosittaisiin kulutuksiin). Kuten kuviot ehdottivat, havaitsimme, että tärkein kerroin on "Jäsenyyden kesto" -ennustajalla, jota seuraavat "Aika sovelluksessa" ja "Keskimääriinen istunnon pituuus". Aika verkkosivustolla ei näytä olevan merkittävä tekijä siihen, kuinka paljon asiakas kuluttaa vuodessa.

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

```
In [16]: lm = LinearRegression()
```

```
In [18]: lm.fit(X_train, y_train)
```

```
Out[18]: ▾ LinearRegression  
          LinearRegression()
```

```
In [19]: # the coefficients  
lm.coef_
```

```
Out[19]: array([25.72425621, 38.59713548, 0.45914788, 61.67473243])
```

```
In [20]: # r squared  
lm.score(X, y)
```

```
Out[20]: 0.9842821675307221
```

```
In [21]: # The coefficients in a dataframe  
cdf = pd.DataFrame(lm.coef_, X.columns, columns=['Coef'])  
print(cdf)
```

	Coef
Avg. Session Length	25.724256
Time on App	38.597135
Time on Website	0.459148
Length of Membership	61.674732

Mallin kouluttaminen monimuuttujaregressiolla käyttäen OLS-menetelmää

Mahdollistaa saamaan tarkempia tietoja mallista.

```
In [31]: X = sm.add_constant(X_train)  
model = sm.OLS(y_train, X)  
model_fit = model.fit()  
print(model_fit.summary())
```

```

          OLS Regression Results
=====
Dep. Variable: Yearly Amount Spent    R-squared:      0.985
Model:                          OLS    Adj. R-squared:  0.985
Method: Least Squares    F-statistic:   5825.
Date:       Wed, 16 Oct 2024    Prob (F-statistic): 3.46e-315
Time:           09:53:13    Log-Likelihood: -1296.2
No. Observations:      350    AIC:             2602.
Df Residuals:         345    BIC:             2622.
Df Model:                   4
Covariance Type:    nonrobust
=====

=====
```

	coef	std err	t	P> t	[0.025	0.
975]						

const	-1050.6537	26.458	-39.710	0.000	-1102.694	-99
8.614						
Avg. Session Length	25.7243	0.534	48.137	0.000	24.673	2
6.775						
Time on App	38.5971	0.528	73.045	0.000	37.558	3
9.636						
Time on Website	0.4591	0.520	0.884	0.377	-0.563	
1.481						
Length of Membership	61.6747	0.516	119.540	0.000	60.660	6
2.690						
=====						
Omnibus:	1.523	Durbin-Watson:	2.024			
Prob(Omnibus):	0.467	Jarque-Bera (JB):	1.262			
Skew:	-0.108	Prob(JB):	0.532			
Kurtosis:	3.199	Cond. No.	2.56e+03			
=====						

Notes:

- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [2] The condition number is large, 2.56e+03. This might indicate that there are strong multicollinearity or other numerical problems.

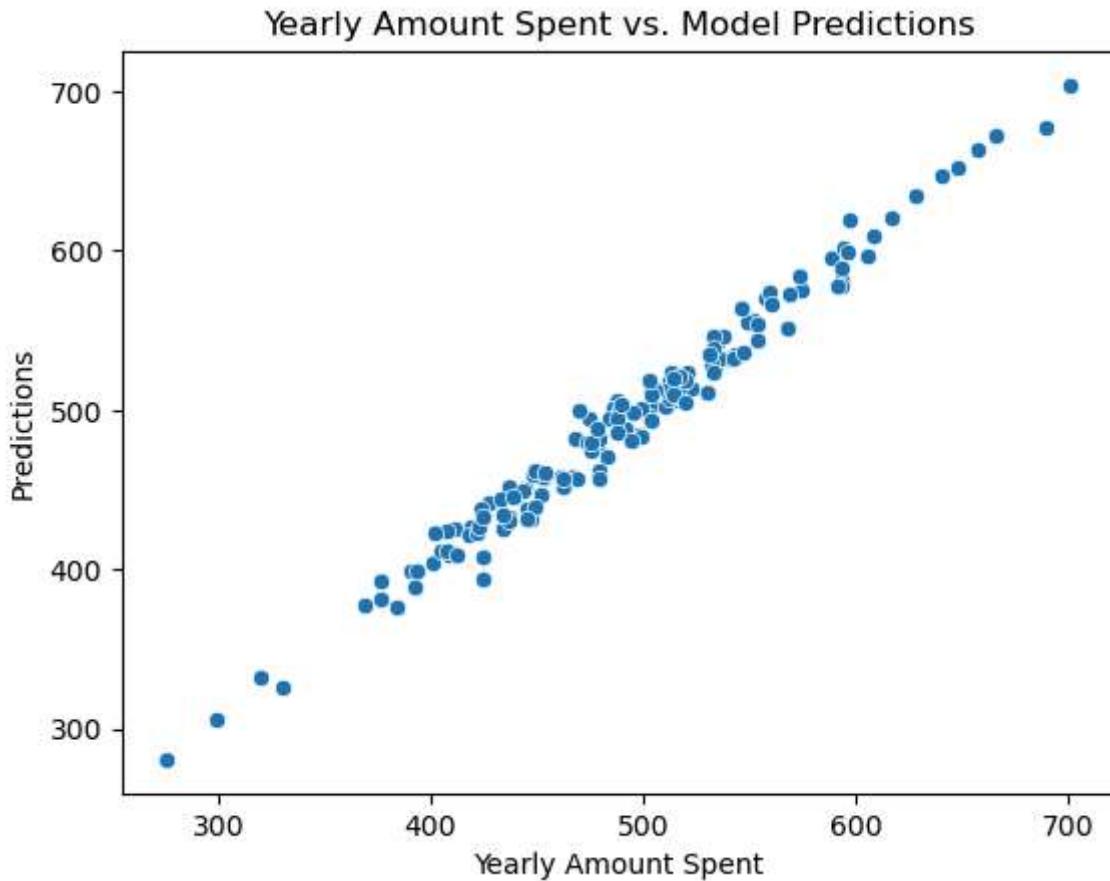
Testidatan ennustaminen

Nyt kun malli on koulutettu, voimme käyttää sitä ennusteiden tekemiseen ja mallin arvioimiseen. Alla oleva hajontakaavio kuvaaa todelliset y-avrot verrattuna mallin ennusteisiin. Malli näyttää toimivan tarkasti.

```
In [32]: predictions = lm.predict(X_test)
```

```
In [34]: # Scatter plot of actual values of y vs predicted values.
sns.scatterplot(x=y_test, y=predictions)
plt.ylabel('Predictions')
```

```
plt.title('Yearly Amount Spent vs. Model Predictions')
plt.show()
```



Mallin arvointi

```
In [35]: from sklearn.metrics import mean_squared_error, mean_absolute_error
import math
```

```
In [36]: print('Mean Absolute Error:',mean_absolute_error(y_test, predictions))
print('Mean Squared Error:',mean_squared_error(y_test, predictions))
print('Root Mean Squared Error:',math.sqrt(mean_squared_error(y_test, predictions)))
```

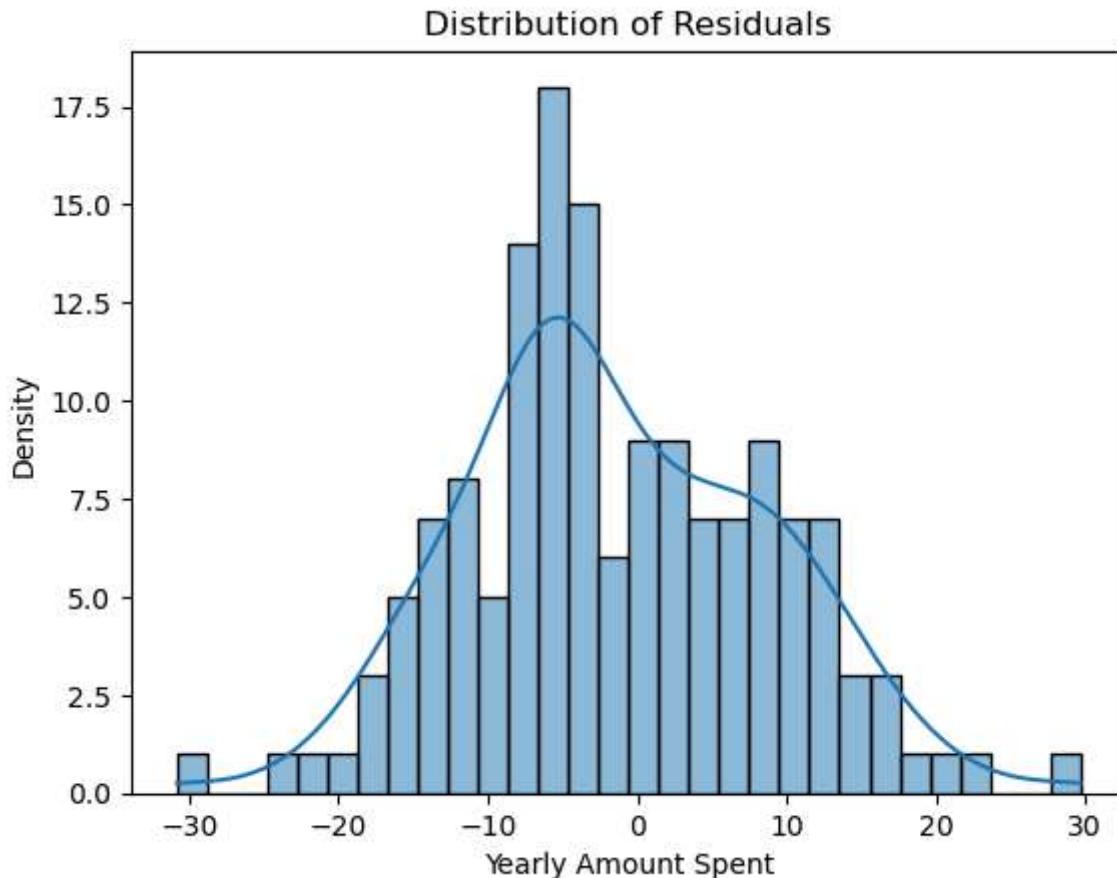
Mean Absolute Error: 8.426091641432116
 Mean Squared Error: 103.91554136503333
 Root Mean Squared Error: 10.193897260863155

Residuals (Jäännökset)

Mallin ennusteiden jäännösten jakaumakuvaaja. Niiden tulisi olla normaalijakautuneita.

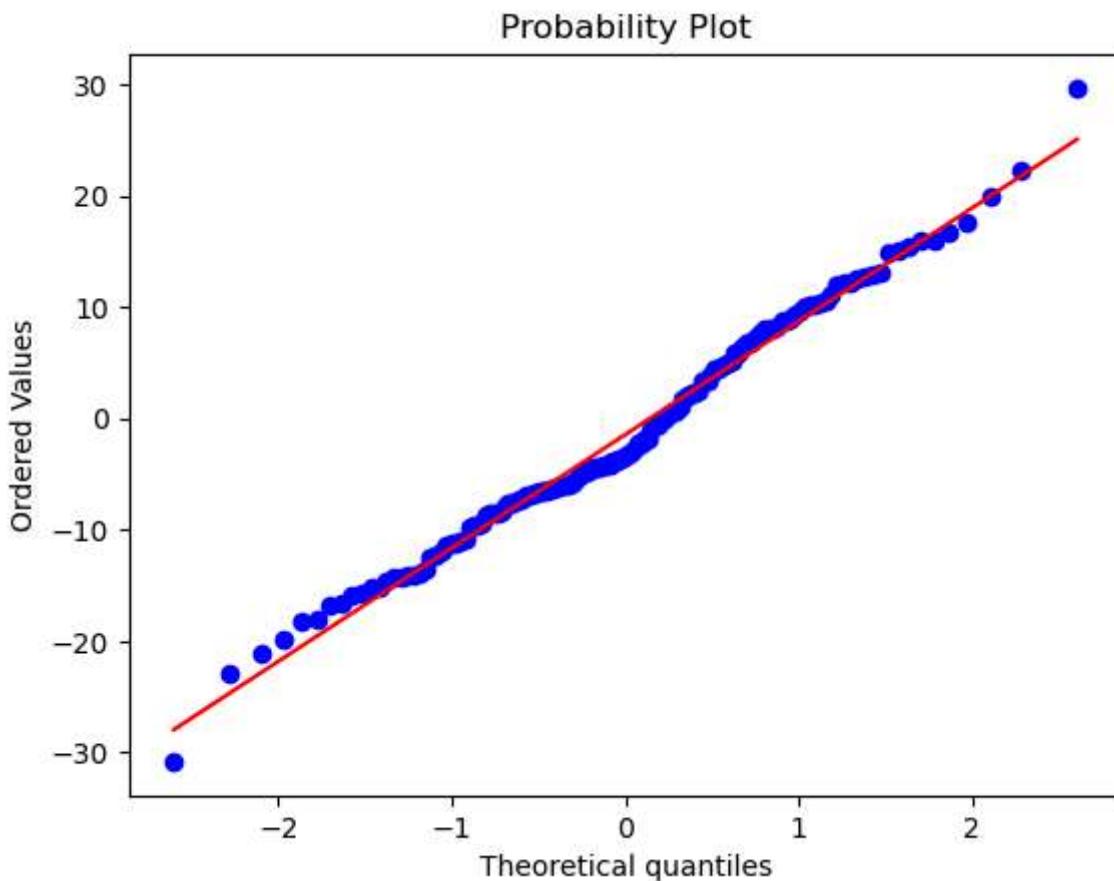
```
In [38]: residuals = y_test - predictions
sns.histplot(residuals, bins=30, kde=True)
plt.xlabel('Yearly Amount Spent')
plt.ylabel('Density')
```

```
plt.title('Distribution of Residuals')
plt.show()
```



```
In [39]: import pylab
import scipy.stats as stats

stats.probplot(residuals, dist="norm", plot=pylab)
pylab.show()
```



Johtopäätös

Tämän analyysin tulkinta voi olla hankala. Mallin mukaan asiakkaiden kannalta merkittävin tekijä ei ole sovelluksessa tai verkkosivustolla vietetty aika, vaan heidän jäsenyytensä kesto. Kuitenkin kahdesta ennustetekijästä (työpöytä vs. sovellus) sovellus vaikuttaa olevan huomattavasti vahvempi tekijä. Itse asiassa työpöytäsivustolla vietetty aika ei näytä korreloivan lainkaan! Toisin sanoen, datan perusteella asiakkaan työpöytäsivustolla viettämällä ajalla ei näytä olevan juuri mitään tekemistä sen kanssa, kuinka paljon rahaa he kuluttavat.

Tätä voidaan tulkita kahdella eri tavalla. Ensinnäkin tämä voisi tarkoittaa, että työpöytäsivusto tarvitsee enemmän kehitystä, jotta se saisi kävijät ostamaan enemmän. Toiseksi, tämä voisi tarkoittaa, että ihmiset ovat yleensä enemmän mobiilisovellusten vaikutuksen alaisena kuin työpöytäsivustojen. Joten ehkä kannattaisi keskittää ponnistelut tämän tosiasian hyödyntämiseen. Tämän tiedon tulkinta vaatii kuitenkin asianuntiemusta verkkokaupan markkinointialalta. Analyysimme ja mallimme kuitenkin tekevät erittäin hyvää työtä ennustetekijöiden painotuksen suhteen.

In []: