Linear Regression on Ecommerce

In [ ]:
```
#Libraries
```

In [ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
#Read CSV file
```

In [ ]:
```
from google.colab import files

uploaded = files.upload()
```

Browse...

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving Ecommerce_Customers.csv.csv to Ecommerce_Customers.csv (1).csv
```

In [ ]:
```
df = pd.read_csv('Ecommerce_Customers.csv.csv')
display(df.head())
```

	Email	Address	Avatar	Avg. Session Length	Time A
0	mstephenson@fernandez.com	835 Frank Tunnel\nWrightmouth, MI 82180-9605	Violet	34.497268	12.655
1	hduke@hotmail.com	4547 Archer Common\nDiazchester, CA 06566-8576	DarkGreen	31.926272	11.109
2	pallen@yahoo.com	24645 Valerie Unions Suite 582\nCobbborough, D...	Bisque	33.000915	11.330
3	riverarebecca@gmail.com	1414 David Throughway\nPort Jason, OH 22070-1220	SaddleBrown	34.305557	13.717
4	mstephens@davidson-herman.com	14023 Rodriguez Passage\nPort Jacobville, PR 3...	MediumAquaMarine	33.330673	12.795

In [ ]:
```
df.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 [ ]:

```
df.describe()
```

Out[ ]:

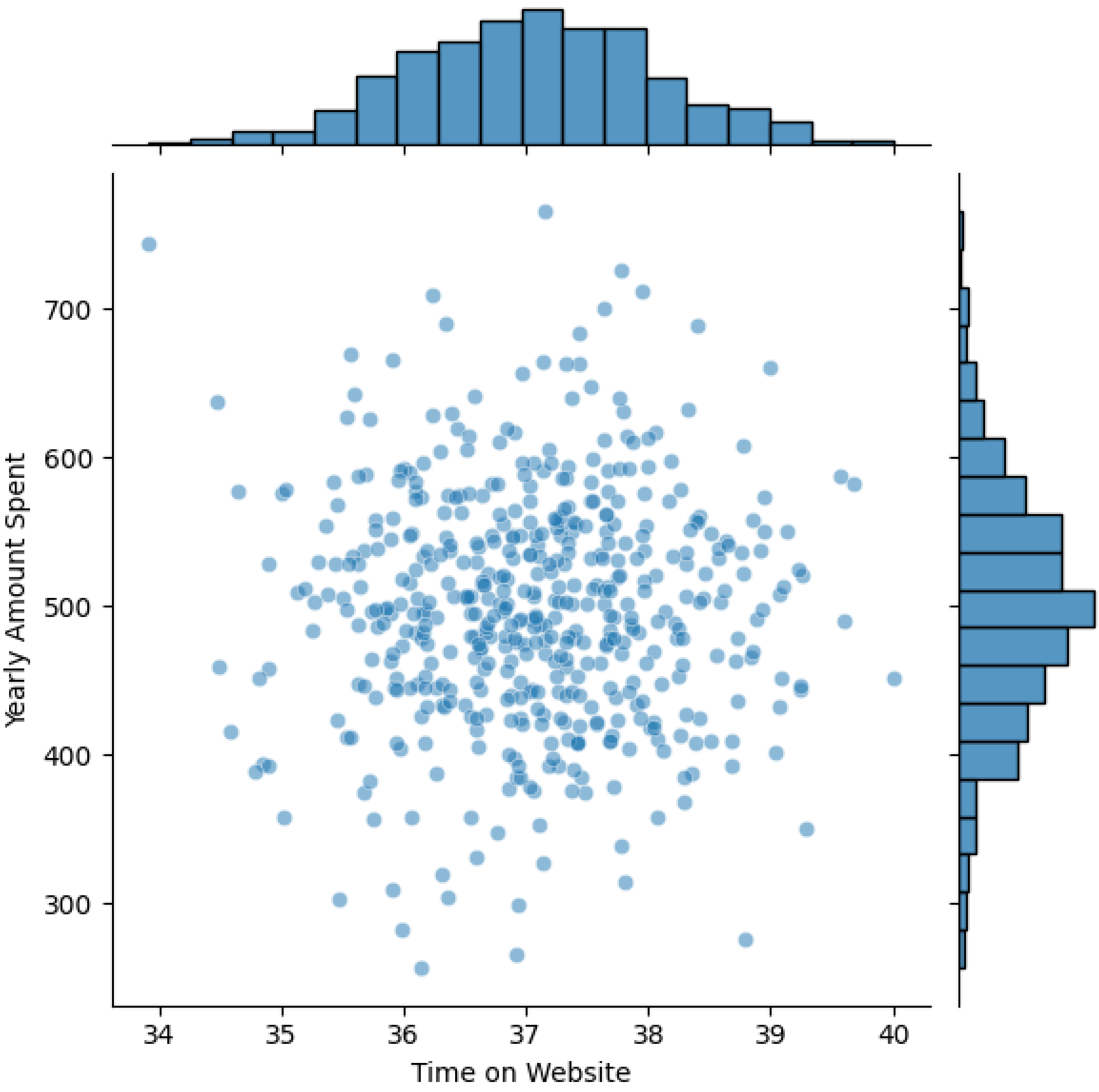
	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
count	500.000000	500.000000	500.000000	500.000000	500.000000
mean	33.053194	12.052488	37.060445	3.533462	499.314038
std	0.992563	0.994216	1.010489	0.999278	79.314782
min	29.532429	8.508152	33.913847	0.269901	256.670582
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

In [ ]:

```
#EDA
sns.jointplot(x='Time on Website',y='Yearly Amount Spent',data=df,alpha=0.5)
```

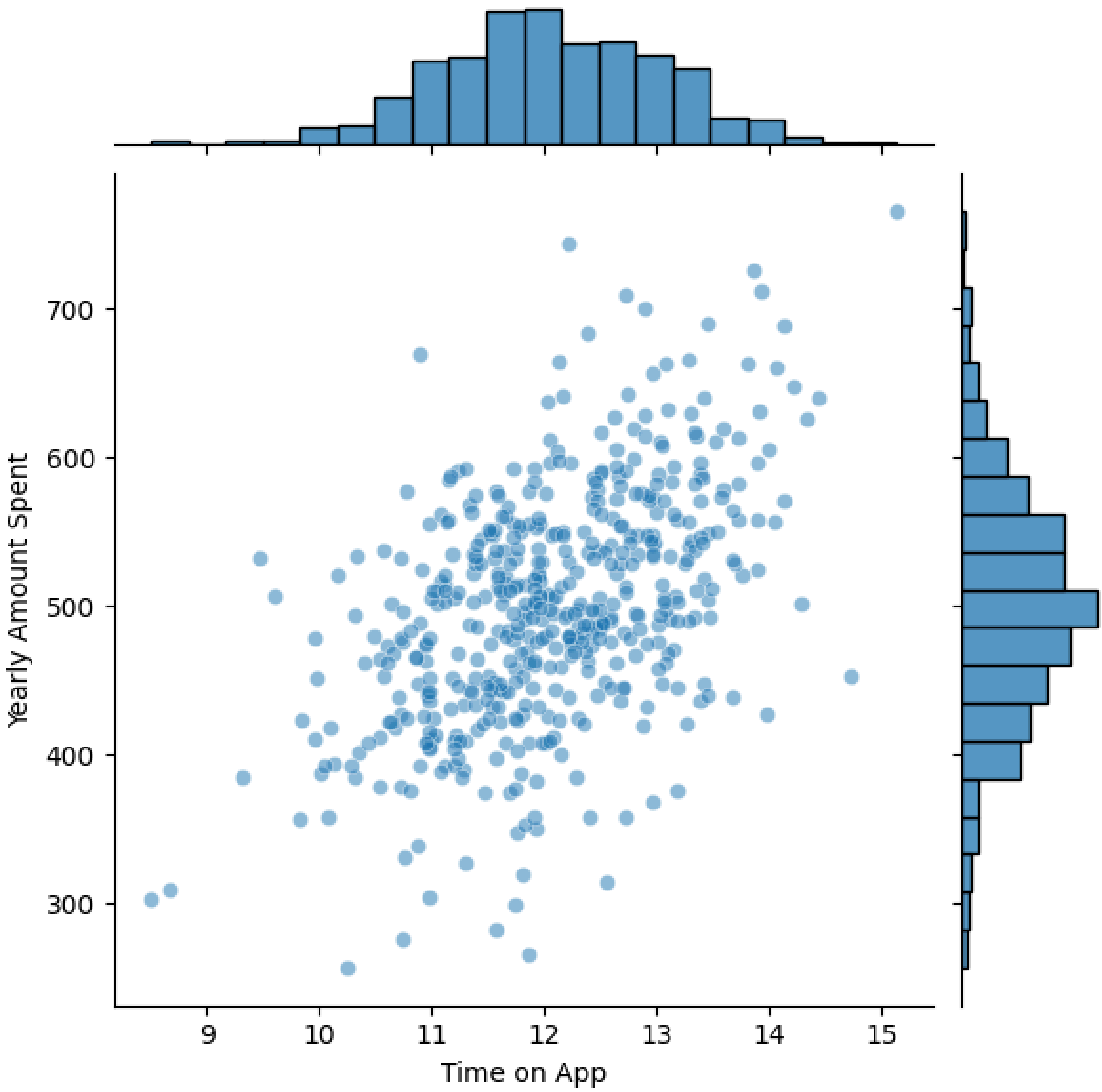
Out[ ]:

```
<seaborn.axisgrid.JointGrid at 0x7edaef249d00>
```



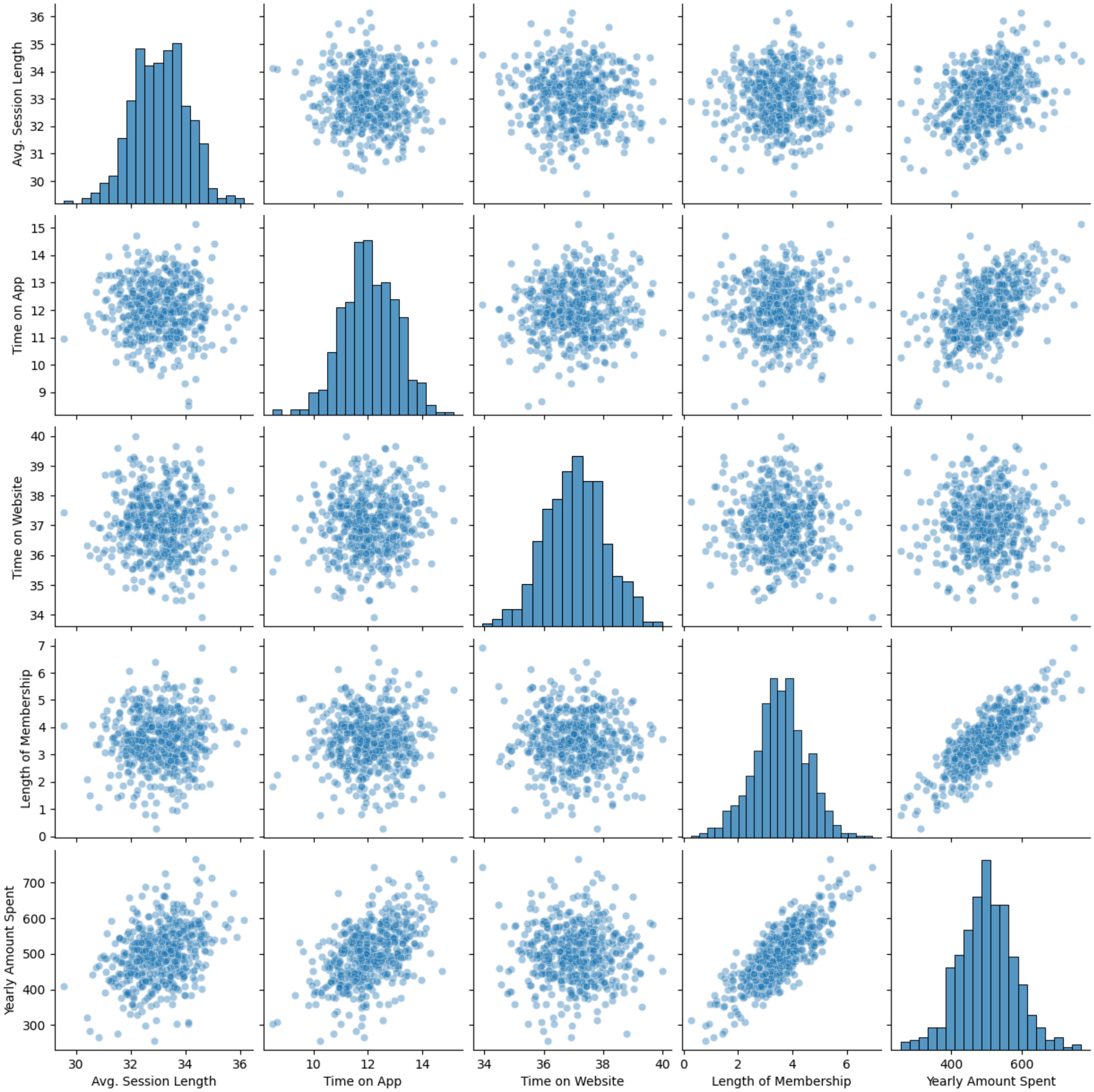
```
In [ ]: sns.jointplot(x='Time on App',y='Yearly Amount Spent',data=df,alpha=0.5)

Out[ ]: <seaborn.axisgrid.JointGrid at 0x7edaeefce8a0>
```



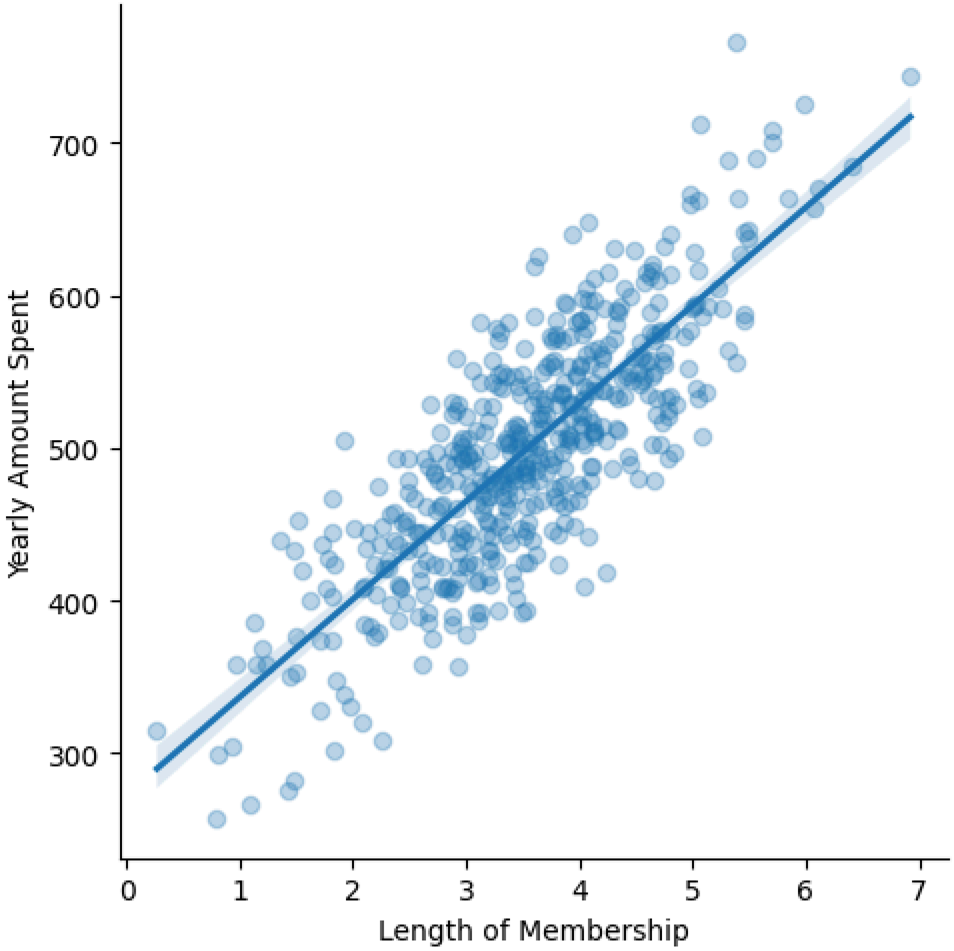
```
In [ ]: sns.pairplot(df, kind='scatter', plot_kws={'alpha': 0.4})
```

```
Out[ ]: <seaborn.axisgrid.PairGrid at 0x7edaeabd92e0>
```



```
In [ ]: sns.lmplot(x = 'Length of Membership', y = 'Yearly Amount Spent', data = df, s
```

```
Out[ ]: <seaborn.axisgrid.FacetGrid at 0x7edae97a9280>
```



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

```
In [ ]: x = df[['Avg. Session Length', 'Time on App','Time on Website', 'Length of Mem
        y = df['Yearly Amount Spent']
```

```
In [ ]: X_train, X_test, y_train, y_test = train_test_split(x,y, test_size=0.3, random
```

```
In [ ]: #Training the model
```

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

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

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

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

```
In [ ]: lm.coef_
```

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

In [ ]:
```
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
```

In [ ]:
```
#predictions
```

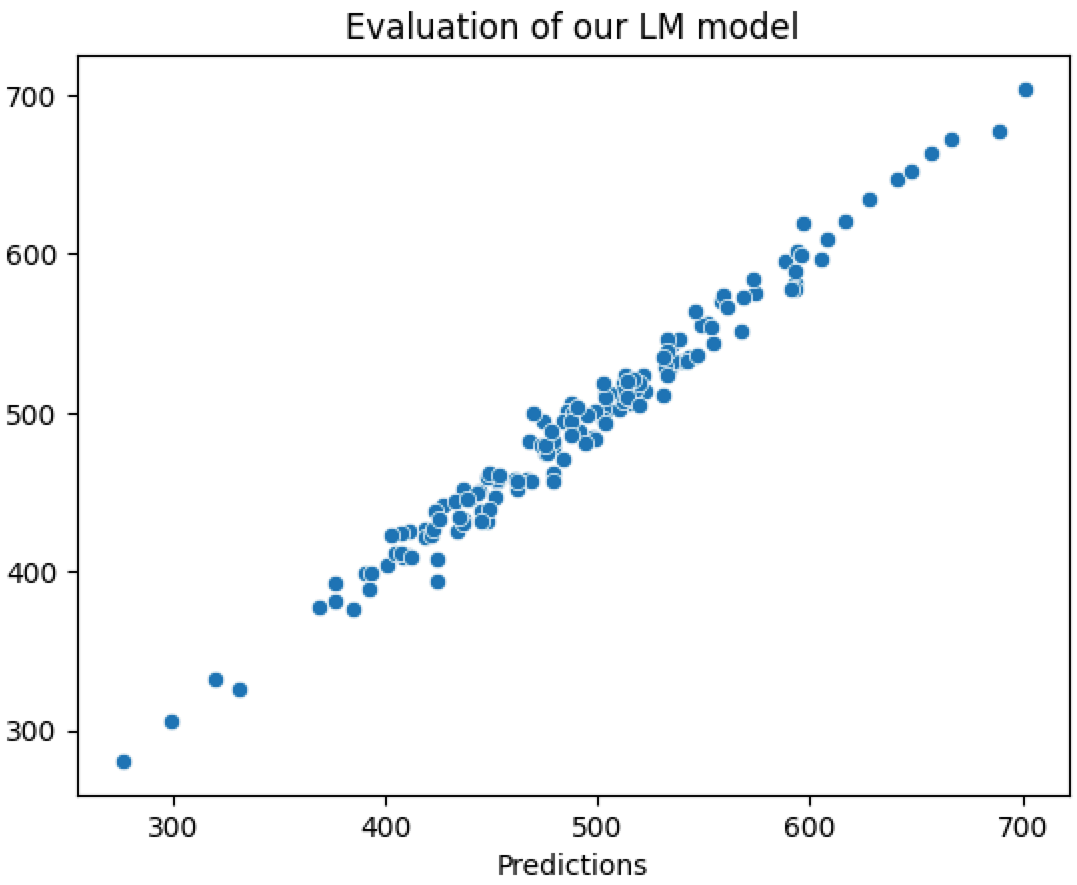
In [ ]:
```
predictions = lm.predict(X_test)
```

In [ ]:
```
predictions
```

```
Out[ ]: array([403.66993069, 542.57756289, 427.06591658, 502.02460425,
               410.12143559, 569.93442508, 531.93431341, 506.29650969,
               408.71870658, 473.97737105, 441.46912726, 425.33703059,
               425.1297229 , 527.61676714, 431.45684016, 424.0769184 ,
               575.76543296, 484.89856554, 458.35936863, 481.96502182,
               502.32441491, 513.63783554, 507.58877002, 646.57464283,
               450.24372141, 496.27043415, 556.40457807, 554.95630839,
               399.64237199, 325.84623136, 532.89783259, 478.12238702,
               501.05701845, 305.97335848, 505.77244448, 483.79591969,
               518.8331528 , 438.18241857, 456.71094234, 471.04609461,
               494.44008972, 445.31155755, 508.78802753, 501.04594193,
               488.83499673, 535.38079541, 595.20129802, 514.04714872,
               280.76758312, 433.10112367, 421.70823427, 481.23640152,
               584.71372272, 608.7748096 , 563.98513427, 494.72804869,
               394.52133407, 456.4197529 , 573.08767515, 499.6984241 ,
               512.83277025, 392.12434043, 480.05057697, 481.54520299,
               475.1117359 , 546.2717533 , 430.85039085, 602.16082001,
               422.3695128 , 493.57280186, 528.74970313, 581.49002635,
               620.19139276, 512.56880298, 411.76623862, 498.47637494,
               461.51337557, 446.41371051, 448.07229961, 535.44710412,
               599.45225302, 619.33717662, 494.15919062, 671.99976398,
               532.46469814, 438.90606319, 515.04975242, 546.7821954 ,
               331.94282076, 510.51987447, 536.57891032, 500.19533618,
               376.92345776, 573.73961388, 479.68031607, 588.61435483,
               485.69922203, 456.40200844, 399.25197845, 451.5098931 ,
               519.40693826, 434.71194217, 596.13049586, 487.91791966,
               407.46691799, 524.16812757, 504.12982787, 452.11540623,
               524.21791295, 457.59311643, 444.19371592, 457.80432916,
               448.76590761, 438.31789012, 677.04967982, 566.09639245,
               651.93616661, 381.08127926, 577.5577254 , 578.35797052,
               518.61431291, 538.94532336, 377.4301223 , 663.30814872,
               523.83158824, 456.86065622, 446.07594402, 388.55038282,
               521.03242183, 431.94999241, 460.08016327, 426.31959507,
               433.30417088, 634.89577554, 462.41086078, 460.71673829,
               512.49535288, 703.83033889, 411.84238624, 551.54681408,
               553.33669558, 409.68202123, 423.34491341, 509.66438623,
               509.88865178, 543.67591782, 504.31300469, 519.18802223,
               520.03155195, 535.13855037])
```

```
In [ ]: sns.scatterplot(x=y_test,y=predictions)
        plt.xlabel('Predictions')
        plt.title("Evaluation of our LM model")
```

```
Out[ ]: Text(0.5, 1.0, 'Evaluation of our LM model')
```



In [ ]:

```
from sklearn.metrics import mean_squared_error, mean_absolute_error
import math
```

In [ ]:

```
print("MAE: ",mean_absolute_error(y_test,predictions))
print("MSE",mean_squared_error(y_test,predictions))
print("RMSE",math.sqrt(mean_squared_error(y_test,predictions)))
```

```
MAE:  8.426091641432116
MSE 103.91554136503333
RMSE 10.193897260863155
```