Start a Spark Session

In [1]:
```
import findspark
```

In [2]:
```
findspark.init('/home/ubuntu/spark-2.4.5-bin-hadoop2.7')
```

In [3]:
```
import pyspark
```

In [4]:
```
from pyspark.sql import SparkSession
```

In [5]:
```
spark = SparkSession.builder.appName('Basics').getOrCreate()
```

Read & Explore Data

In [6]:
```
df = spark.read.csv('ecommerce.csv', inferSchema=True, header=True)
```

In [7]:
```
df.printSchema()
```

```
root
 |-- Email: string (nullable = true)
 |-- Address: string (nullable = true)
 |-- Avatar: string (nullable = true)
 |-- Avg Session Length: double (nullable = true)
 |-- Time on App: double (nullable = true)
 |-- Time on Website: double (nullable = true)
 |-- Length of Membership: double (nullable = true)
 |-- Yearly Amount Spent: double (nullable = true)
```

In [8]:
```
df.head()
```

Out[8]:
```
Row(Email='mstephenson@fernandez.com', Address='835 Frank TunnelWrightmout
h, MI 82180-9605', Avatar='Violet', Avg Session Length=34.49726772511229, T
ime on App=12.65565114916675, Time on Website=39.57766801952616, Length of
Membership=4.0826206329529615, Yearly Amount Spent=587.9510539684005)
```

In [9]:
```
for i in df.head():
    print(i)
```

```
mstephenson@fernandez.com
835 Frank TunnelWrightmouth, MI 82180-9605
Violet
34.49726772511229
12.65565114916675
39.57766801952616
4.0826206329529615
587.9510539684005
```

In [10]:
```
df.columns
```

Out[10]:
```
['Email',
 'Address',
 'Avatar',
 'Avg Session Length',
 'Time on App',
 'Time on Website',
 'Length of Membership',
 'Yearly Amount Spent']
```

In [11]:
```
df.describe(['Avg Session Length', 'Time on App', 'Time on Website', 'Length
```

```
+-------+------------------+------------------+------------------+---------
-----------+------------------+
|summary|Avg Session Length|       Time on App|   Time on Website|Length of
Membership|Yearly Amount Spent|
+-------+------------------+------------------+------------------+---------
-----------+------------------+
|  count|               500|               500|               500|
500|               500|
|   mean| 33.05319351819619|12.052487937166134| 37.06044542094859|   3.5334
61555915055|  499.3140382585909|
| stddev|0.9925631110845354|0.9942156084725424|1.0104889067564033|  0.99927
75024112585|   79.3147815497068|
|    min|29.532428967057943| 8.508152176032603| 33.91384724758464|  0.26990
10899842742| 256.67058229005585|
|    max| 36.13966248879052|15.126994288792467|40.005181638101895|   6.9226
89335035808|  765.5184619388373|
+-------+------------------+------------------+------------------+---------
-----------+------------------+
```

Spark Feature Engineering

In [12]:
```
from pyspark.ml.feature import VectorAssembler
```

In [13]:
```
orAssembler(inputCols=['Avg Session Length', 'Time on App', 'Time on Website'
```

In [14]:
```
vectordata = vector.transform(df)
```

In [15]:
```
vectordata.head()
```

Out[15]:
```
Row(Email='mstephenson@fernandez.com', Address='835 Frank TunnelWrightmout
h, MI 82180-9605', Avatar='Violet', Avg Session Length=34.49726772511229, T
ime on App=12.65565114916675, Time on Website=39.57766801952616, Length of
Membership=4.0826206329529615, Yearly Amount Spent=587.9510539684005, featu
res=DenseVector([34.4973, 12.6557, 39.5777, 4.0826]))
```

In [16]:
```
vectordata.select('features').show()
```

```
+--------------------+
|            features|
+--------------------+
|[34.4972677251122...|
|[31.9262720263601...|
|[33.0009147556426...|
|[34.3055566297555...|
|[33.3306725236463...|
|[33.8710378793419...|
|[32.0215955013870...|
|[32.7391429383803...|
|[33.9877728956856...|
|[31.9365486184489...|
|[33.9925727749537...|
|[33.8793608248049...|
|[29.5324289670579...|
|[33.1903340437226...|
|[32.3879758531538...|
|[30.7377203726281...|
|[32.1253868972878...|
|[32.3388993230671...|
|[32.1878120459321...|
|[32.6178560628234...|
+--------------------+
only showing top 20 rows
```

In [17]:
```
spark_data = vectordata.select('features', 'Yearly Amount Spent')
```

In [18]:
```
spark_data.show()
```

```
+--------------------+------------------+
|            features|Yearly Amount Spent|
+--------------------+------------------+
|[34.4972677251122...|  587.9510539684005|
|[31.9262720263601...|  392.2049334443264|
|[33.0009147556426...| 487.54750486747207|
|[34.3055566297555...|  581.8523440352177|
|[33.3306725236463...|  599.4060920457634|
|[33.8710378793419...|   637.102447915074|
|[32.0215955013870...|  521.5721747578274|
|[32.7391429383803...|  549.9041461052942|
|[33.9877728956856...|  570.2004089636196|
|[31.9365486184489...|  427.1993848953282|
|[33.9925727749537...|  492.6060127179966|
|[33.8793608248049...|  522.3374046069357|
|[29.5324289670579...|  408.6403510726275|
|[33.1903340437226...|  573.4158673313865|
|[32.3879758531538...|  470.4527333009554|
|[30.7377203726281...|  461.7807421962299|
|[32.1253868972878...| 457.84769594494855|
|[32.3388993230671...| 407.70454754954415|
|[32.1878120459321...|  452.3156754800354|
|[32.6178560628234...|   605.061038804892|
+--------------------+------------------+
only showing top 20 rows
```

In [19]:
```
train_data, test_data = spark_data.randomSplit([.7,.3])
```

In [20]:
```
train_data.head(), train_data.describe().show()
```

```
+-------+------------------+
|summary|Yearly Amount Spent|
+-------+------------------+
|  count|               343|
|   mean|  494.7344323801276|
| stddev|  78.04967500996317|
|    min|   266.086340948469|
|    max|  744.2218671047146|
+-------+------------------+
```

Out[20]:
```
(Row(features=DenseVector([29.5324, 10.9613, 37.4202, 4.0464]), Yearly Amou
nt Spent=408.6403510726275),
 None)
```

In [21]:
```
test_data.head(), test_data.describe().show()
```

```
+-------+------------------+
|summary|Yearly Amount Spent|
+-------+------------------+
|  count|               157|
|   mean|  509.3191644771436|
| stddev|  81.36801406032563|
|    min| 256.67058229005585|
|    max|  765.5184619388373|
+-------+------------------+
```

Out[21]:
```
(Row(features=DenseVector([30.5744, 11.351, 37.0888, 4.0783]), Yearly Amoun
t Spent=442.06441375806565),
 None)
```

In [22]:
```
train_data.count(), test_data.count()
```

Out[22]:
```
(343, 157)
```

Building a Spark ML Model & Model Evaluation

In [23]:
```
from pyspark.ml.regression import LinearRegression
```

In [24]:
```
lr = LinearRegression(featuresCol='features', labelCol='Yearly Amount Spent')
```

In [25]:
```
lr_model = lr.fit(train_data)
```

In [26]:
```
result = lr_model.evaluate(test_data)
```

In [27]:
```
print("RMSE:", result.rootMeanSquaredError)
```

```
RMSE: 9.51265624205725
```

In [28]:
```
print("R2:", result.r2)
```

```
R2: 0.9862446632403771
```

In [29]:
```
unlabeled_test = test_data.select('features')
```

In [30]:
```
unlabeled_result = lr_model.transform(unlabeled_test)
```

In [31]:
```
unlabeled_result.show()
```

```
+--------------------+------------------+
|            features|        prediction|
+--------------------+------------------+
|[30.5743636841713...|442.74321046431896|
|[30.9716756438877...|  489.133391682353|
|[31.0613251567161...| 494.6341281920111|
|[31.1280900496166...| 564.8556367524479|
|[31.3091926408918...|430.52027914566474|
|[31.3895854806643...|409.62344893684076|
|[31.4252268808548...| 534.8134706254341|
|[31.4459724827577...|  481.722038813831|
|[31.5761319713222...| 543.6536471329969|
|[31.6098395733896...| 427.7896369321079|
|[31.6548096756927...| 468.1622352002139|
|[31.6739155032749...|502.52296488070647|
|[31.7216523605090...|349.61439757979747|
|[31.8164283341993...| 518.4884982914152|
|[31.8745516945853...| 398.9197960271008|
|[31.9048571310136...| 491.4826965194027|
|[31.9120759292006...| 389.8739088932232|
|[31.9453957483445...| 663.4999487169118|
|[31.9480174211613...| 456.0284248402504|
|[31.9764800614612...| 325.8186923516014|
+--------------------+------------------+
only showing top 20 rows
```

Thank You!!