BDSN ASSIGNMENT

Submitted by :- ISHAAN NIRMAL (A21015)

BACKGROUND

- In this emerging tech world every small to large scale businesses are going online in order to increase their sales but there is one common problem which all of them face – How to target more and more customers so that they spend more on their products and how to predict what amount is the new customer is gonna spend in the ecommerce platform .
- Here comes the use of predictive analysis but why is predictive analysis important for ecommerce platforms , the answer is simple – “Predictive analytics enables e-commerce businesses to understand what products customers are looking for, helps in identifying popular and not-so-popular products and product categories”.

OBJECTIVE

- The objective of this project is that we have the dataset of an ecommerce business we will examine that data and then we will build a linear regression model which will predict the customers yearly spend on the company's product .

DATASET

	Email	Address	Avatar	Avg Session Length	Time on App	Time on Website	Length
2	mstephenson@fernandez.com	835 Frank TunnelWrightmouth, MI	Violet	34.49726773	12.65565115	39.57766802	
3	hduke@hotmail.com	4547 Archer CommonDiazchester,	DarkGreen	31.92627203	11.10946073	37.26895887	
4	pallen@yahoo.com	24645 Valerie Unions Suite 582Cob	Bisque	33.00091476	11.33027806	37.11059744	
5	riverarebecca@gmail.com	1414 David ThroughwayPort Jason	SaddleBrown	34.30555663	13.71751367	36.72128268	
6	mstephens@davidson-herman.co	14023 Rodriguez PassagePort Jacol	MediumAquaMarine	33.33067252	12.79518855	37.5366533	
7	alvareznancy@lucas.biz	645 Martha Park Apt. 611Jeffreych	FloralWhite	33.87103788	12.02692534	34.47687763	
8	katherine20@yahoo.com	68388 Reyes Lights Suite 692Josep	DarkSlateBlue	32.0215955	11.36634831	36.68377615	
9	awatkins@yahoo.com	Unit 6538 Box 8980DPO AP 09026-4	Aqua	32.73914294	12.35195897	37.37335886	
10	vchurch@walter-martinez.com	860 Lee KeyWest Debra, SD 97450-	Salmon	33.9877729	13.38623528	37.53449734	
11	bonnie69@lin.biz	PSC 2734, Box 5255APO AA 98456-7	Brown	31.93654862	11.81412829	37.14516822	
12	andrew06@peterson.com	26104 Alexander GrovesAlexandri	Tomato	33.99257277	13.33897545	37.22580613	
13	ryanwerner@freeman.biz	Unit 2413 Box 0347DPO AA 07580-2	Tomato	33.87936082	11.584783	37.08792607	
14	knelson@gmail.com	6705 Miller Orchard Suite 186Lake	RoyalBlue	29.53242897	10.9612984	37.42021558	
15	wrightpeter@yahoo.com	05302 Dunlap FerryNew Stephanie	Bisque	33.19033404	12.95922609	36.1446667	
16	taylormason@gmail.com	7773 Powell Springs Suite 190Sam	DarkBlue	32.38797585	13.14872569	36.61995708	
17	jstark@anderson.com	49558 Ramirez Road Suite 399Phill	Peru	30.73772037	12.63660605	36.21376309	
18	wjennings@gmail.com	6362 Wilson MountainJohnsonfurt	PowderBlue	32.1253869	11.73386169	34.89409275	
19	rebecca45@hale-bauer.biz	8982 Burton RowWilsonton, PW 88	OliveDrab	32.33889932	12.01319469	38.38513659	
20	alejandro75@hotmail.com	64475 Andre Club Apt. 795Port Dar	Cyan	32.18781205	14.71538754	38.24411459	
21	samuel46@love-west.net	544 Alexander Heights Suite 768N	LightSeaGreen	32.61785606	13.98959256	37.1905038	
22	megan33@gmail.com	84426 Julia VistaNorth Teresa, KY 5	PeachPuff	32.91278511	11.36549203	37.60779252	
23	agolden@yahoo.com	PSC 2490, Box 2120APO AE 15445-2	Black	33.50308726	12.8779837	37.44102134	
24	vstafford@hotmail.com	PSC 5723, Box 8159APO AA 74738	Olive	31.53160448	13.37856278	38.73400629	
25	denise22@hernandez-townsend.(	USNS CardenasFPO AA 85439-9449	Silver	32.90325097	11.65757592	36.77260376	

DATASET COLUMN DESCRIPTION

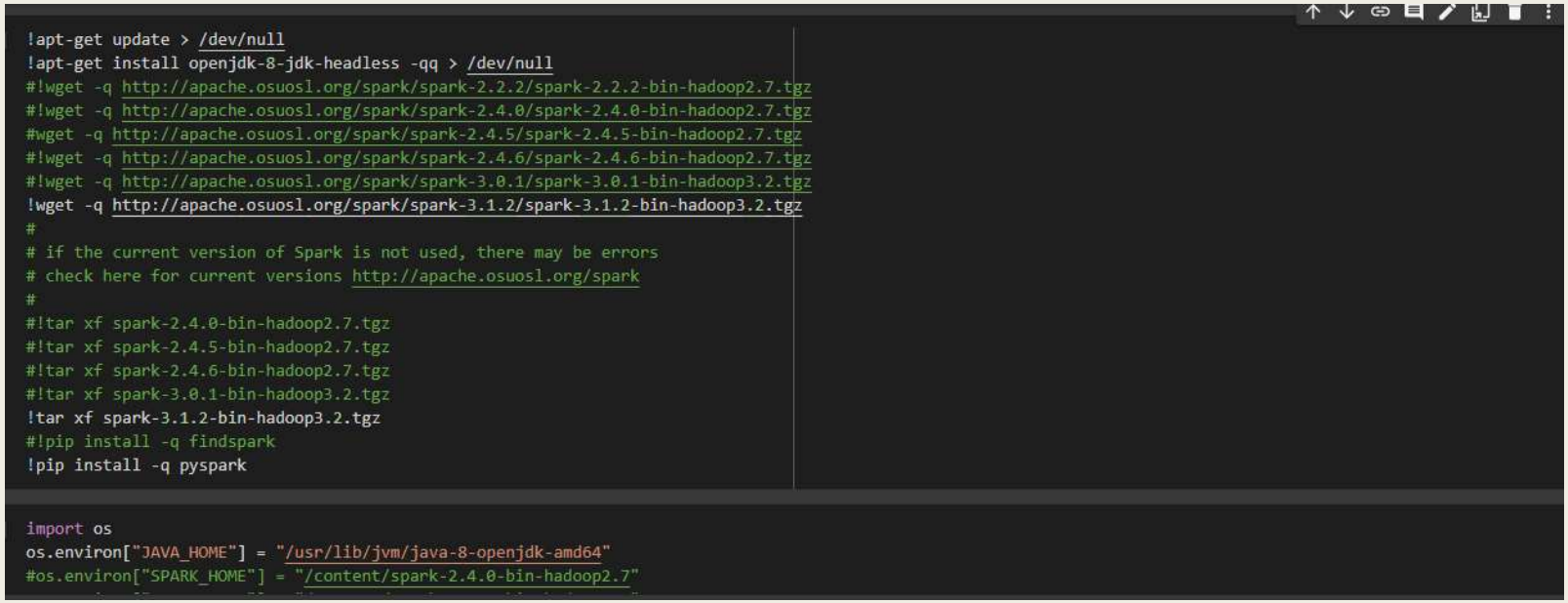
- EMAIL- consist up of the email id's of the customers
- AVATAR
- ADDRESS – consist up of the delivery address / main address of the customers
- AVG SESSION LENGTH – average amount of time spent by customer
- TIME ON APP – time spent by customers on the app
- TIME ON WEBSITE – time spent by customers on the website
- LENGTH OF MEMBERSHIP – time for which customers have been the members
- YEARLY SPENT AMOUNT – amount spent by the customer on yearly basis

FLOW OF PROJECT & TOOLS USED

- The project has been done on google collab and the tools used are
- SPARK
- PYTHON

Creating the environment

- First of all we will create the working environment for spark in google collab as displayed in the picture below .

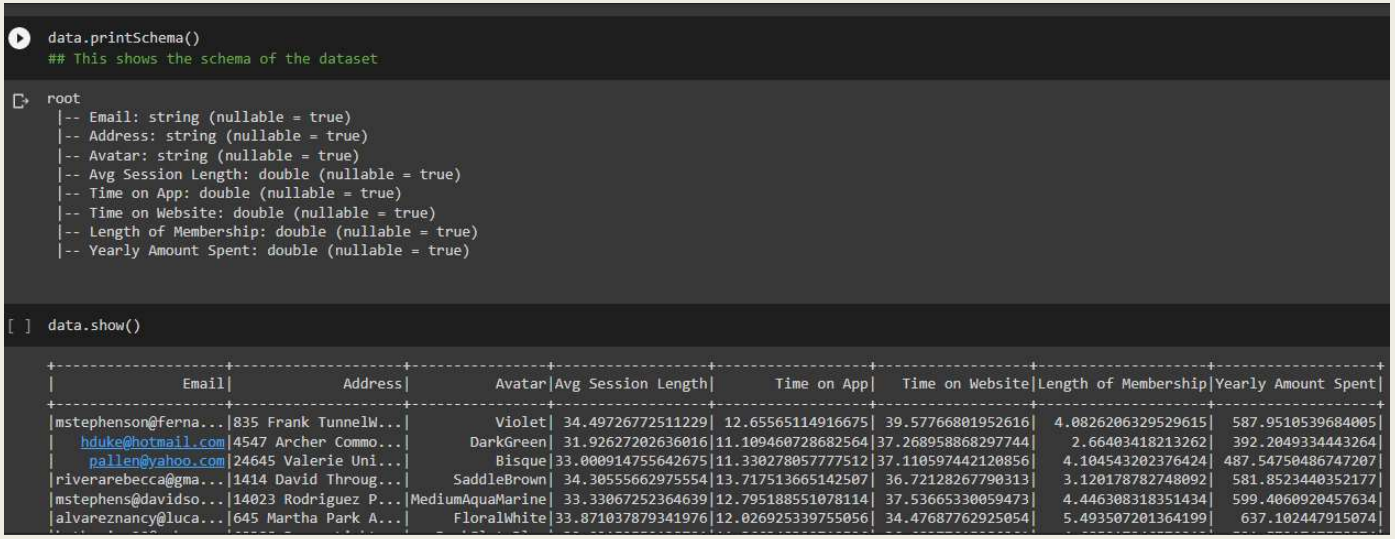


```
!apt-get update > /dev/null
!apt-get install openjdk-8-jdk-headless -qq > /dev/null
#!wget -q http://apache.osuosl.org/spark/spark-2.2.2/spark-2.2.2-bin-hadoop2.7.tgz
#!wget -q http://apache.osuosl.org/spark/spark-2.4.0/spark-2.4.0-bin-hadoop2.7.tgz
#wget -q http://apache.osuosl.org/spark/spark-2.4.5/spark-2.4.5-bin-hadoop2.7.tgz
#!wget -q http://apache.osuosl.org/spark/spark-2.4.6/spark-2.4.6-bin-hadoop2.7.tgz
#!wget -q http://apache.osuosl.org/spark/spark-3.0.1/spark-3.0.1-bin-hadoop3.2.tgz
!wget -q http://apache.osuosl.org/spark/spark-3.1.2/spark-3.1.2-bin-hadoop3.2.tgz
#
# if the current version of Spark is not used, there may be errors
# check here for current versions http://apache.osuosl.org/spark
#
#!tar xf spark-2.4.0-bin-hadoop2.7.tgz
#!tar xf spark-2.4.5-bin-hadoop2.7.tgz
#!tar xf spark-2.4.6-bin-hadoop2.7.tgz
#!tar xf spark-3.0.1-bin-hadoop3.2.tgz
!tar xf spark-3.1.2-bin-hadoop3.2.tgz
#!pip install -q findspark
!pip install -q pyspark
```

```
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
#os.environ["SPARK_HOME"] = "/content/spark-2.4.0-bin-hadoop2.7"
```

CHECKING THE SCHEMA OF DATA & NULL VALUES

- We checked the schema of the data & found that the schema of the data provided was right nad there was no need of changing or transforming the schema
- Then we checked for the null values and found that there were no null values present in the dataset .



```
data.printSchema()
## This shows the schema of the dataset
```

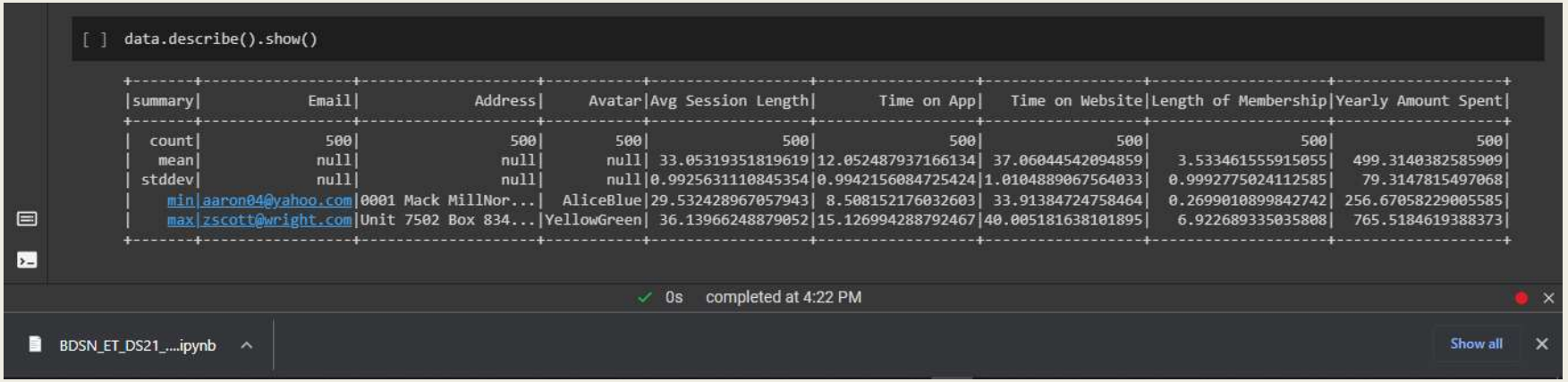
```
root
 |-- Email: string (nullable = true)
 |-- Address: string (nullable = true)
 |-- Avatar: string (nullable = true)
 |-- Avg Session Length: double (nullable = true)
 |-- Time on App: double (nullable = true)
 |-- Time on Website: double (nullable = true)
 |-- Length of Membership: double (nullable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

```
data.show()
```

Email	Address	Avatar	Avg Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
mstephenson@ferna...	835 Frank TunnelW...	Violet	34.49726772511229	12.65565114916675	39.57766801952616	4.0826206329529615	587.9510539684005
hduke@hotmail.com	4547 Archer Commo...	DarkGreen	31.92627202636016	11.109460728682564	37.268958868297744	2.66403418213262	392.2049334443264
pallen@yahoo.com	24645 Valerie Uni...	Bisque	33.000914755642675	11.330278057777512	37.110597442120856	4.104543202376424	487.54750486747207
riverarebecca@gma...	1414 David Throug...	SaddleBrown	34.30555662975554	13.717513665142507	36.72128267790313	3.120178782748092	581.8523440352177
mstephens@davidso...	14023 Rodriguez P...	MediumAquaMarine	33.33067252364639	12.795188551078114	37.53665330059473	4.446308318351434	599.4060920457634
alvareznancy@luca...	645 Martha Park A...	FloralWhite	33.871037879341976	12.026925339755056	34.47687762925054	5.493507201364199	637.102447915074

GENERAL STATS

- Before going for the modelling part of the data we just saw an overview of the description of the data .



```
[ ]  data.describe().show()
```

summary	Email	Address	Avatar	Avg Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
count	500	500	500	500	500	500	500	500
mean	null	null	null	33.05319351819619	12.052487937166134	37.06044542094859	3.533461555915055	499.3140382585909
stddev	null	null	null	0.9925631110845354	0.9942156084725424	1.0104889067564033	0.9992775024112585	79.3147815497068
min	aaron04@yahoo.com	0001 Mack MillNor...	AliceBlue	29.532428967057943	8.508152176032603	33.91384724758464	0.2699010899842742	256.67058229005585
max	zscott@wright.com	Unit 7502 Box 834...	YellowGreen	36.13966248879052	15.126994288792467	40.005181638101895	6.922689335035808	765.5184619388373

0s completed at 4:22 PM

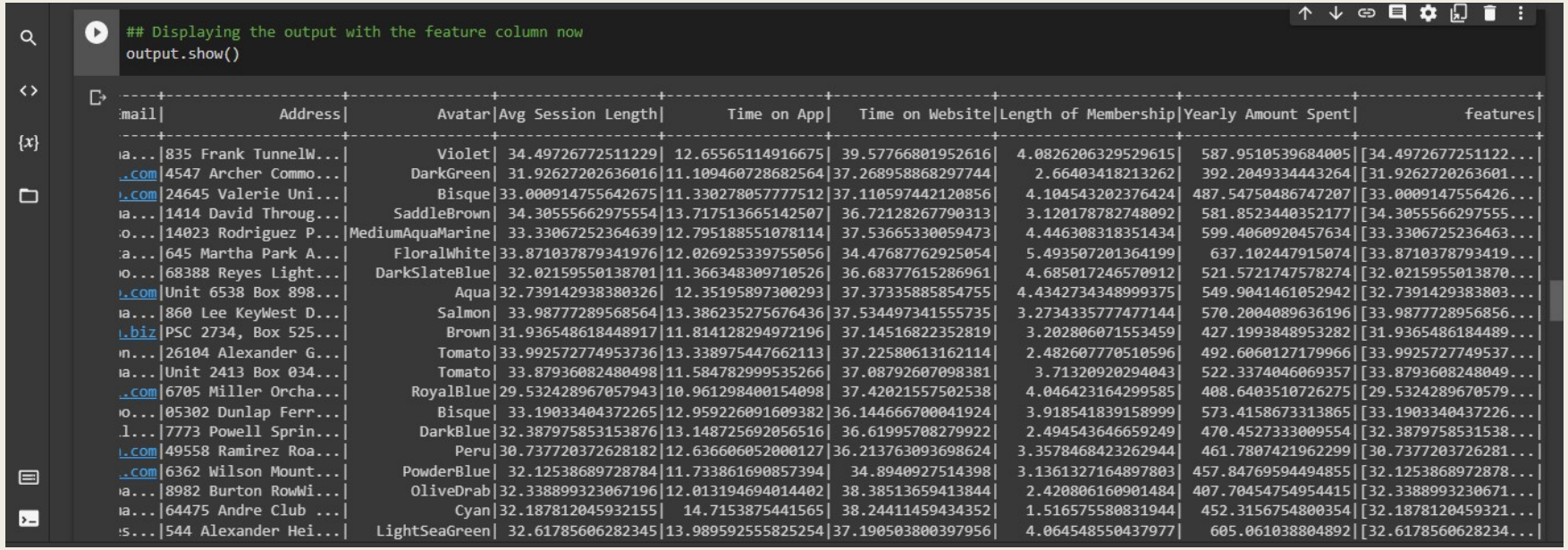
BDSN_ET_DS21_....ipynb

Show all

MODELLING PART

- For the modelling process since we are doing this project using spark we had to change the data frame .
- So we changed our dataframe , we did that because spark accepts the dataframe in the form of “labels” & features .
- we did this using the vector assembler library .

- After using the vector assembler library we got a new column named features which is a form of vector and the dataframe now looked like this .



```
## Displaying the output with the feature column now
output.show()
```

```
+----------+--------------------+---------------+------------------+------------------+------------------+--------------------+------------------+--------------------+
|     email|             Address|         Avatar|Avg Session Length|       Time on App|   Time on Website|Length of Membership|Yearly Amount Spent|            features|
+----------+--------------------+---------------+------------------+------------------+------------------+--------------------+------------------+--------------------+
|     ia...|835 Frank TunnelW...|         Violet| 34.49726772511229| 12.65565114916675| 39.57766801952616|  4.0826206329529615|  587.9510539684005|[34.4972677251122...|
|  .com|4547 Archer Commo...|      DarkGreen| 31.92627202636016|11.109460728682564|37.268958868297744|    2.66403418213262|  392.2049344443264|[31.9262720263601...|
|  .com|24645 Valerie Uni...|         Bisque|33.000914755642675|11.330278057777512|37.110597442120856|   4.104543202376424| 487.54750486747207|[33.0009147556426...|
|     ia...|1414 David Throug...|    SaddleBrown| 34.30555662975554|13.717513665142507| 36.72128267790313|   3.120178782748092|  581.8523440352177|[34.3055566297555...|
|     :o...|14023 Rodriguez P...|MediumAquaMarine| 33.33067252364639|12.795188551078114| 37.53665330059473|   4.446308318351434|  599.4060920457634|[33.3306725236463...|
|     :a...|645 Martha Park A...|    FloralWhite|33.871037879341976|12.026925339755056| 34.47687762925054|   5.493507201364199|   637.102447915074|[33.8710378793419...|
|     io...|68388 Reyes Light...|  DarkSlateBlue| 32.02159550138701|11.366348309710526| 36.68377615286961|   4.685017246570912|  521.5721747578274|[32.0215955013870...|
|  .com|Unit 6538 Box 898...|           Aqua|32.739142938380326| 12.35195897300293| 37.37335885854755|  4.4342734348999375|  549.9041461052942|[32.7391429383803...|
|     ia...|860 Lee KeyWest D...|         Salmon| 33.98777289568564|13.386235275676436|37.534497341555735|   3.2734335777477144|  570.2004089636196|[33.9877728956856...|
|  .biz|PSC 2734, Box 525...|          Brown|31.936548618448917|11.814128294972196| 37.14516822352819|   3.202806071553459|  427.1993848953282|[31.9365486184489...|
|     in...|26104 Alexander G...|         Tomato|33.992572774953736|13.338975447662113| 37.22580613162114|   2.482607770510596|  492.6060127179966|[33.9925727749537...|
|     ia...|Unit 2413 Box 034...|         Tomato| 33.87936082480498|11.584782999535266| 37.08792607098381|    3.71320920294043|  522.3374046069357|[33.8793608248049...|
|  .com|6705 Miller Orcha...|     RoyalBlue|29.532428967057943|10.961298400154098| 37.42021557502538|   4.046423164299585|  408.6403510726275|[29.5324289670579...|
|     io...|05302 Dunlap Ferr...|         Bisque| 33.19033404372265|12.959226091609382|36.144666700041924|   3.918541839158999|  573.4158673313865|[33.1903340437226...|
|     :l...|7773 Powell Sprin...|       DarkBlue|32.387975853153876|13.148725692056516| 36.61995708279922|   2.494543646659249|  470.4527333009554|[32.3879758531538...|
|  .com|49558 Ramirez Roa...|           Peru|30.737720372628182|12.636606052000127|36.213763093698624|   3.3578468423262944|  461.7807421962299|[30.7377203726281...|
|  .com|6362 Wilson Mount...|     PowderBlue| 32.12538689728784|11.733861690857394|  34.8940927514398|   3.1361327164897803|  457.8476959494855|[32.1253868972878...|
|     ia...|8982 Burton RowWi...|      OliveDrab|32.338899323067196|12.013194694014402| 38.38513659413844|   2.420806160901484|  407.7045475495441 5|[32.3388993230671...|
|     ia...|64475 Andre Club ...|           Cyan|32.187812045932155|  14.7153875441565| 38.24411459434352|   1.516575580831944|  452.3156754800354|[32.1878120459321...|
|     :s...|544 Alexander Hei...|  LightSeaGreen| 32.61785606282345|13.989592555825254|37.190503800397956|   4.064548550437977|   605.061038804892|[32.6178560628234...|
+----------+--------------------+---------------+------------------+------------------+------------------+--------------------+------------------+--------------------+
```

MODEL USED

- We used a linear regression model here because the we were doing the analysis on a continuous set of data
- So to start the modelling process we first of all split the data set into training and test data .
- Training data – is used to train the model
- Test data – is used to validate the model
- We split the data set by 70:30 ratio

- 70 being the training data
- 30 being the test data

```
[ ] ## Splitting the data into train & test
    X_train , Y_test = new_data.randomSplit([0.7,0.3])

[ ] X_train.cache()
    Y_test.cache()

    DataFrame[features: vector, Yearly Amount Spent: double]

    X_train.describe().show()

    +-------+------------------+
    |summary|Yearly Amount Spent|
    +-------+------------------+
    |  count|               358|
    |   mean| 502.01951451152667|
    | stddev|  82.17590366737566|
    |    min| 256.67058229005585|
    |    max|  765.5184619388373|
    +-------+------------------+
```

- After splitting up the data set we called the linear regression library of spark
- We trained the model .

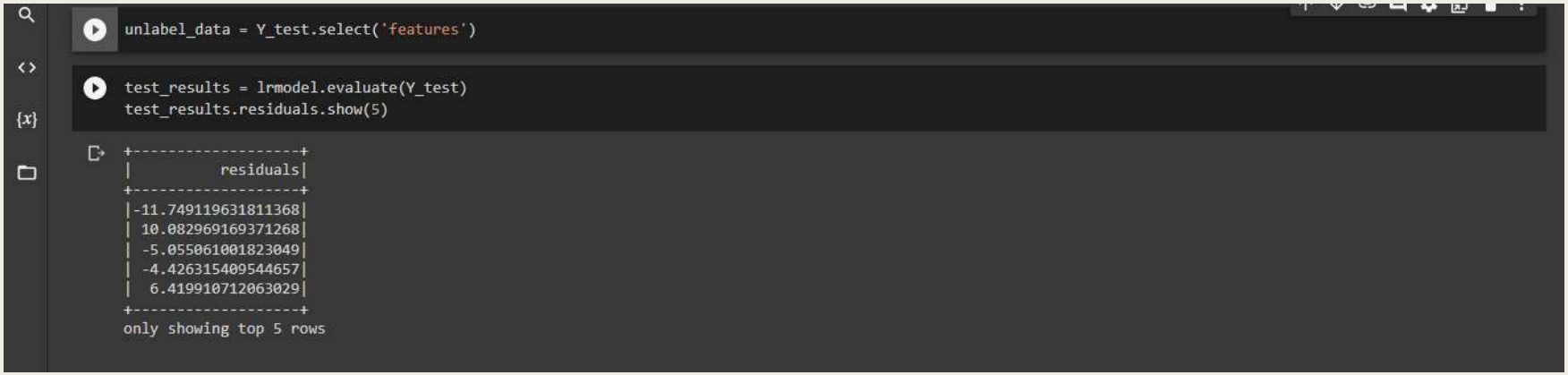
```
[ ] Y_test.describe().show()

+-------+------------------+
|summary|Yearly Amount Spent|
+-------+------------------+
|  count|               142|
|   mean| 492.4931896772456|
| stddev| 71.41536554117073|
|    min| 319.9288698031936|
|    max| 744.2218671047146|
+-------+------------------+

[ ] from pyspark.ml.regression import LinearRegression
    ## Creating the object for the model
    lr = LinearRegression(labelCol="Yearly Amount Spent")

[ ] ## Now we are fitting the model to the data
    lrmodel = lr.fit(X_train,)
```

- Now after training the mode l our next job was to fit the model
- So we did that with our test data .

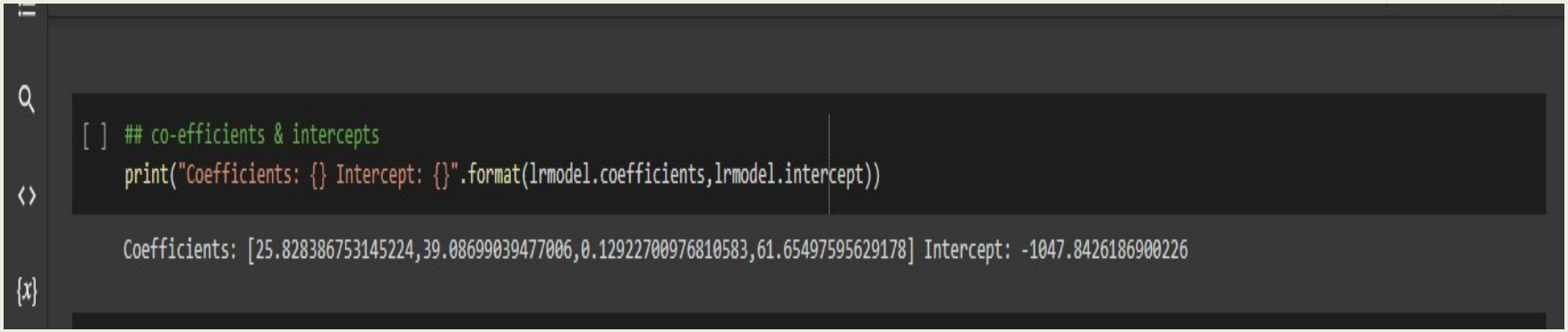


- Now after fitting the data the next step was to see the prediction of the model so we got the following output as the prediction output .

```
[ ] pred = lrmodel.transform(unlabel_data)
    pred.show()
    ## Displaying the predictions done
    ## These predictions are the people's yearly amount spent on the company's products
```

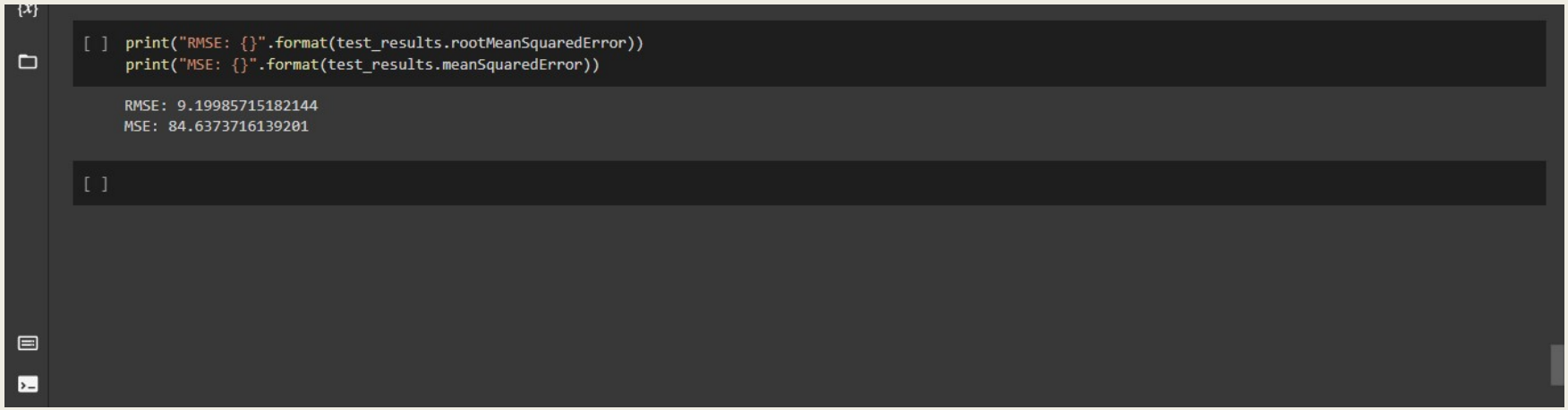
```
+--------------------+------------------+
|            features|        prediction|
+--------------------+------------------+
|[30.3931845423455...|  331.677989435005|
|[30.7377203726281...| 451.6977730268586|
|[30.8364326747734...|472.55696142881266|
|[30.8794843441274...|494.63291539439933|
|[30.9716756438877...| 488.2186990448297|
|[31.0472221394875...| 387.2553765225696|
|[31.1280900496166...| 565.2321589983799|
|[31.3584771924370...| 491.8016397350764|
|[31.3662121671876...|426.63603770919144|
|[31.5261978982398...| 417.7954205931435|
|[31.6098395733896...| 428.0958187366334|
|[31.7207699002873...| 545.5527130800422|
|[31.7216523605090...|349.59568889321895|
|[31.7242025238451...| 510.4423322498967|
|[31.7366356860502...|494.74412382541254|
|[31.8124825597242...| 396.7653634151186|
|[31.8186165667690...| 448.9703544127092|
|[31.8648325480987...| 450.9753141611334|
|[31.8745516945853...| 397.6600699859639|
```

- So these were the predictions obtained after fitting the model .
- Now we got the coefficient and the intercepts for the model .
- What is a co-efficient – The values which multiply the predictor values .
- What is an intercept – The expected value of Y when X =0 .



These were the values of co-efficients & intercept that were obtained .

- Next we calculated the MSE & RMSE values which is displayed below
- RMSE – standard deviation of the residuals .
- MSE - how close a regression line is to a set of points



- So basically our objective for this project was to predict the yearly spent and we have achieved that objective using the linear regression model .
- This brings us to the end of this project .