Coventry University
FACULTY OF ENGINEERING, ENVIRONMENT & COMPUTING

(7082CEM)

Coursework

Big Data Analytics and Visualization Using PySpark

MODULE LEADER: Dr. Marwan Fuad

Student Name: SHERY SHAJAN

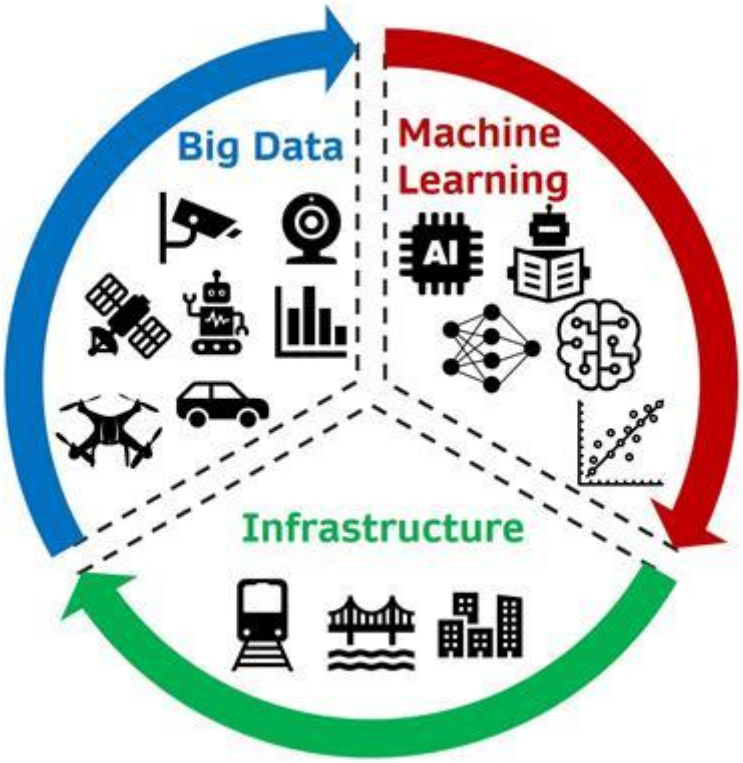
SID: 11438085

ECOMMERCE CUSTOMERS DATA ANALYSIS AND EXPENSE PREDICTION

I can confirm that all work submitted is my own: Yes

Contents

INTRODUCTION

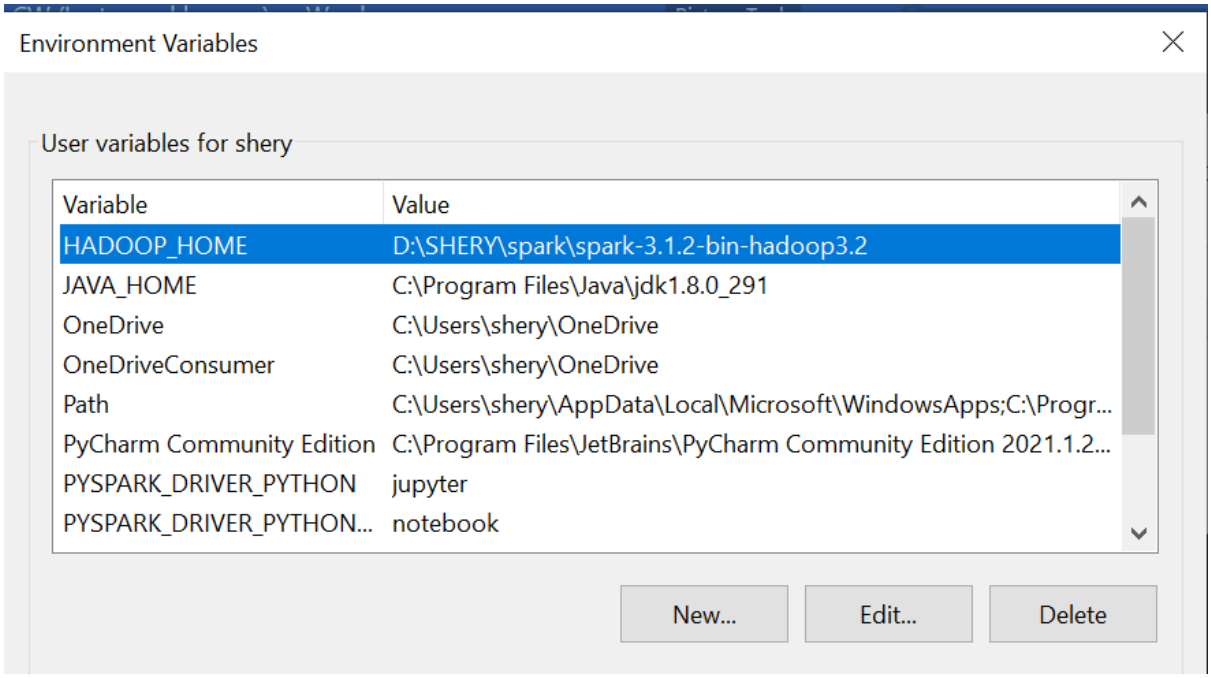


This course work aims on the implementation of Big Data in Machine Learning. As we know Big Data refers to huge datasets of both structured and unstructured data which can't be handled using traditional methods. So here the role of machine learning comes, this accelerates the process with the help of decision-making algorithms. Machine Learning algorithms are valuable for gathering and integrating the data for large organizations. On using Machine learning for big data analysis, the human intervention can be reduced as there are perfect Machine Learning Algorithms in place. Here I am using Pyspark, which is part of the Spark platform, which can provide a better analysis and visualisations. Ongoing further we will go through the implementation part of these software. As the Pyspark is new to me, I will use documentation from their website as reference. At the beginning we will setup the dataset with some pre-processing steps. Once the data is processed and ready for analysis, we will use various functions and machine learning techniques will be applied in order to gain a better understanding of the dataset. The configuration and deployment have a major role in a CW study. Spark and Pyspark can be parallelly deployed with other programs, here I am using Jupyter which is an interactive web-based environment used for live coding and also, for visualisation. The aim of this course work is to process the data and apply machine learning techniques and functions. Here I have taken a dataset from Kaggle. The dataset is Ecommerce Customers.csv, we will be performing a data analysis and then will see how we can apply an algorithm using Pyspark. Now let us look into each step.

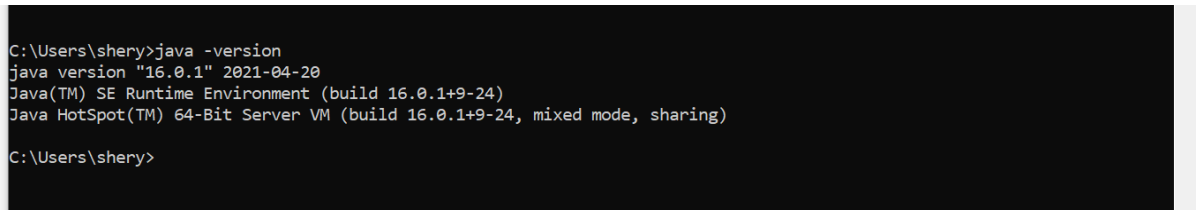
IMPLEMENTATION

Here we are discussing on the deployment of Pyspark. The configuration and installation of the application within Pyspark and Jupyter will be explained further along with the processes and implementation. I have gone through many videos for installation and finally implemented using the below steps. I gone through changhsinlee.com site I have mentioned the link below for installation process. At first, I downloaded the Spark distribution from spark.apache.org. As mentioned earlier I am using Jupyter note book, so for both python and jupyter notebook by installing the Python 3.x version of Anaconda distribution and then Java8 version from Oracle. After downloading all the files then start the installation. After installation we need to set the environment variables as below and then check the installed java version. Then install the findspark and Pyspark in anaconda PowerShell using pip.

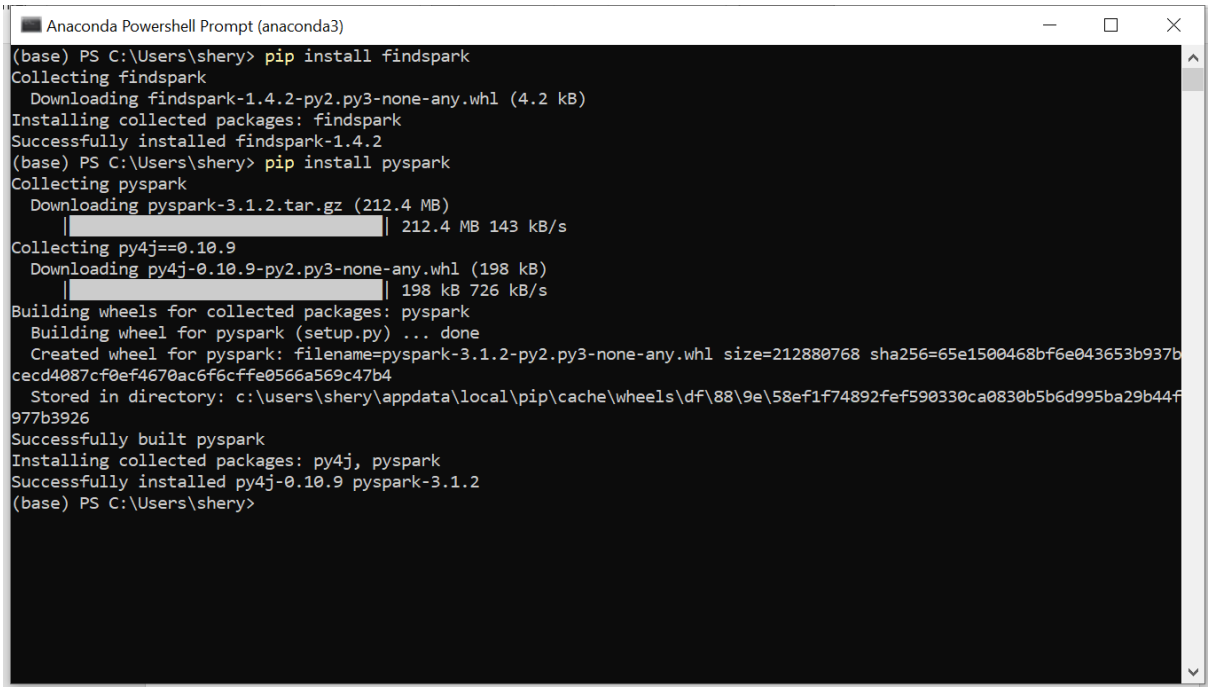
- Setting Environment Variable



- Checking the Installed Java version



- Installation of findspark and Pyspark in anaconda PowerShell using pip.



Now we are ready to start the exploration of our dataset analysis. Now let me take you to the dataset.

Data Set and Data Pre-Processing

I have taken the dataset from Kaggle 'Ecommerce Customers.csv'. As we know that Ecommerce customer service is the means by which online organizations furnish help to clients including settling on online buy choices to settling issues — all while making a consistent client experience across channels and stages. We have 8 columns with 500 entries.

Attribute	Datatype
Email	String
Address	String
Avg Session length	Double
Time on App	Double
Time on Website	Double
Length of membership	Double
Yearly Amount Spent	Double

Now we got an idea about the dataset, let's explore the data. At first, we need to load the dataset for pre-processing in jupyter notebook.

```
#Load Dataset
cust_data=spark.read.csv("Ecommerce Customers.csv",inferSchema=True,header=True)
```

I have saved the dataset in my local folder and opened the jupyter notebook (anaconda3) and opened a new notebook form same directory. Once the dataset is loaded, I used printSchema() function for getting the columns and their corresponding data types.

```
cust_data.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)
```

On exploring the data using show() function, I found some null values. So, I used the dropna() function to remove the null valued rows as a part of pre-processing.

- Before Dropping Null Values

```
cust_data.show()
```

```
+--------------------+--------------------+---------------+------------------+-----------+---------------+--------------------
+-------------------+
|               Email|             Address|         Avatar|Avg Session Length|Time on App|Time on Website|Length of Membership
|Yearly Amount Spent|
+--------------------+--------------------+---------------+------------------+-----------+---------------+--------------------
+-------------------+
|mstephenson@ferna...|    835 Frank Tunnel|           null|              null|       null|           null|                null
|               null|
|         Wrightmouth|      MI 82180-9605"|         Violet|       34.49726773|12.65565115|    39.57766802|         4.082620633
|         587.951054|
|   hduke@hotmail.com|  4547 Archer Common|           null|              null|       null|           null|                null
|               null|
|         Diazchester|      CA 06566-8576"|      DarkGreen|       31.92627203|11.10946073|    37.26895887|         2.664034182
|        392.2049334|
|    pallen@yahoo.com|24645 Valerie Uni...|           null|              null|       null|           null|                null
|               null|
|         Cobbborough|      DC 99414-7564"|         Bisque|       33.00091476|11.33027806|    37.11059744|         4.104543202
|        487.5475049|
|riverarebecca@gma...|1414 David Throug...|           null|              null|       null|           null|                null
|               null|
|          Port Jason|      OH 22070-1220"|    SaddleBrown|       34.30555663|13.71751367|    36.72128268|         3.120178783
|         581.852344|
|mstephens@davidso...|14023 Rodriguez P...|           null|              null|       null|           null|                null
```

- After Dropping Null Values

```
#Exploring Columns
#delete null columns
cust_data.dropna().show(truncate=False)
```

```
+------------------+---------------+-----------------+------------------+-----------+---------------+--------------------+------
-------------+
|Email             |Address        |Avatar           |Avg Session Length|Time on App|Time on Website|Length of Membership|Yearly
Amount Spent|
+------------------+---------------+-----------------+------------------+-----------+---------------+--------------------+------
-------------+
|Wrightmouth       | MI 82180-9605"|Violet           |34.49726773       |12.65565115|39.57766802    |4.082620633         |587.95
1054        |
|Diazchester       | CA 06566-8576"|DarkGreen        |31.92627203       |11.10946073|37.26895887    |2.664034182         |392.20
49334       |
|Cobbborough       | DC 99414-7564"|Bisque           |33.00091476       |11.33027806|37.11059744    |4.104543202         |487.54
75049       |
|Port Jason        | OH 22070-1220"|SaddleBrown      |34.30555663       |13.71751367|36.72128268    |3.120178783         |581.85
2344        |
|Port Jacobville   | PR 37242-1057"|MediumAquaMarine|33.33067252       |12.79518855|37.5366533     |4.446308318         |599.40
6092        |
|Jeffreychester    | MN 67218-7250"|FloralWhite      |33.87103788       |12.02692534|34.47687763    |5.493507201         |637.10
24479       |
|Josephbury        | WV 92213-0247"|DarkSlateBlue    |32.0215955        |11.36634831|36.68377615    |4.685017247         |521.57
21748       |
|West Debra        | SD 97450-0495"|Salmon           |33.9877729        |13.38623528|37.53449734    |3.273433578         |570.20
9409        |
|Alexandriaport    | WY 28244-9149"|Tomato           |33.99257277       |13.33897545|37.22580613    |2.482607771         |492.60
60127       |
|Lake Shanestad    | MO 75696-5051"|RoyalBlue        |29.53242897       |10.9612984 |37.42021558    |4.046423164         |408.64
```

- Then we can check for the existence of null value for every column and count its frequency and then display it in a tabulate.

```
cust_data.select([count(when(col(c).isNull(), c)).alias(c) for c in cust_data.columns]).show()
```

```
+-----+-------+------+-----------------+-----------+--------------+--------------------+------------------+
|Email|Address|Avatar|Avg Session Length|Time on App|Time on Website|Length of Membership|Yearly Amount Spent|
+-----+-------+------+-----------------+-----------+--------------+--------------------+------------------+
|    0|      0|     0|                0|          0|             0|                   0|                 0|
+-----+-------+------+-----------------+-----------+--------------+--------------------+------------------+
```

- Renaming the Columns for better understanding.

```
#To rename a column, we can use withColumnRenamed method
cust_data = cust_data.withColumnRenamed('Avg Session Length', 'Avg_Session_Length')
cust_data = cust_data.withColumnRenamed('Time on App', 'Time_on_App')
cust_data = cust_data.withColumnRenamed('Time on Website', 'Time_on_Website')
cust_data = cust_data.withColumnRenamed('Length of Membership', 'Length_of_Membership')
cust_data = cust_data.withColumnRenamed('Yearly Amount Spent', 'Yearly_Amount_Spent')
cust_data.show(5)
```

```
+---------------+---------------+----------------+------------------+-----------+---------------+--------------------+---------
----------+
|          Email|        Address|          Avatar|Avg_Session_Length|Time_on_App|Time_on_Website|Length_of_Membership|Yearly_Am
ount_Spent|
+---------------+---------------+----------------+------------------+-----------+---------------+--------------------+---------
----------+
|    Wrightmouth| MI 82180-9605"|          Violet|       34.49726773|12.65565115|    39.57766802|         4.082620633|
587.951054|
|    Diazchester| CA 06566-8576"|       DarkGreen|       31.92627203|11.10946073|    37.26895887|         2.664034182|        3
92.2049334|
|    Cobbborough| DC 99414-7564"|          Bisque|       33.00091476|11.33027806|    37.11059744|         4.104543202|        4
87.5475049|
|     Port Jason| OH 22070-1220"|     SaddleBrown|       34.30555663|13.71751367|    36.72128268|         3.120178783|
581.852344|
|Port Jacobville| PR 37242-1057"|MediumAquaMarine|       33.33067252|12.79518855|     37.5366533|         4.446308318|
599.406092|
+---------------+---------------+----------------+------------------+-----------+---------------+--------------------+---------
----------+
only showing top 5 rows
```

DATA ANALYSIS AND VISUALISATION

- Now we have got a clean data for deeper exploration. Here we are using filter function for querying the data.

```
##Query Data
#filter method and this will return all the records that match the condition
cust_data.filter(cust_data["Avatar"] == "Violet").show(10)
```

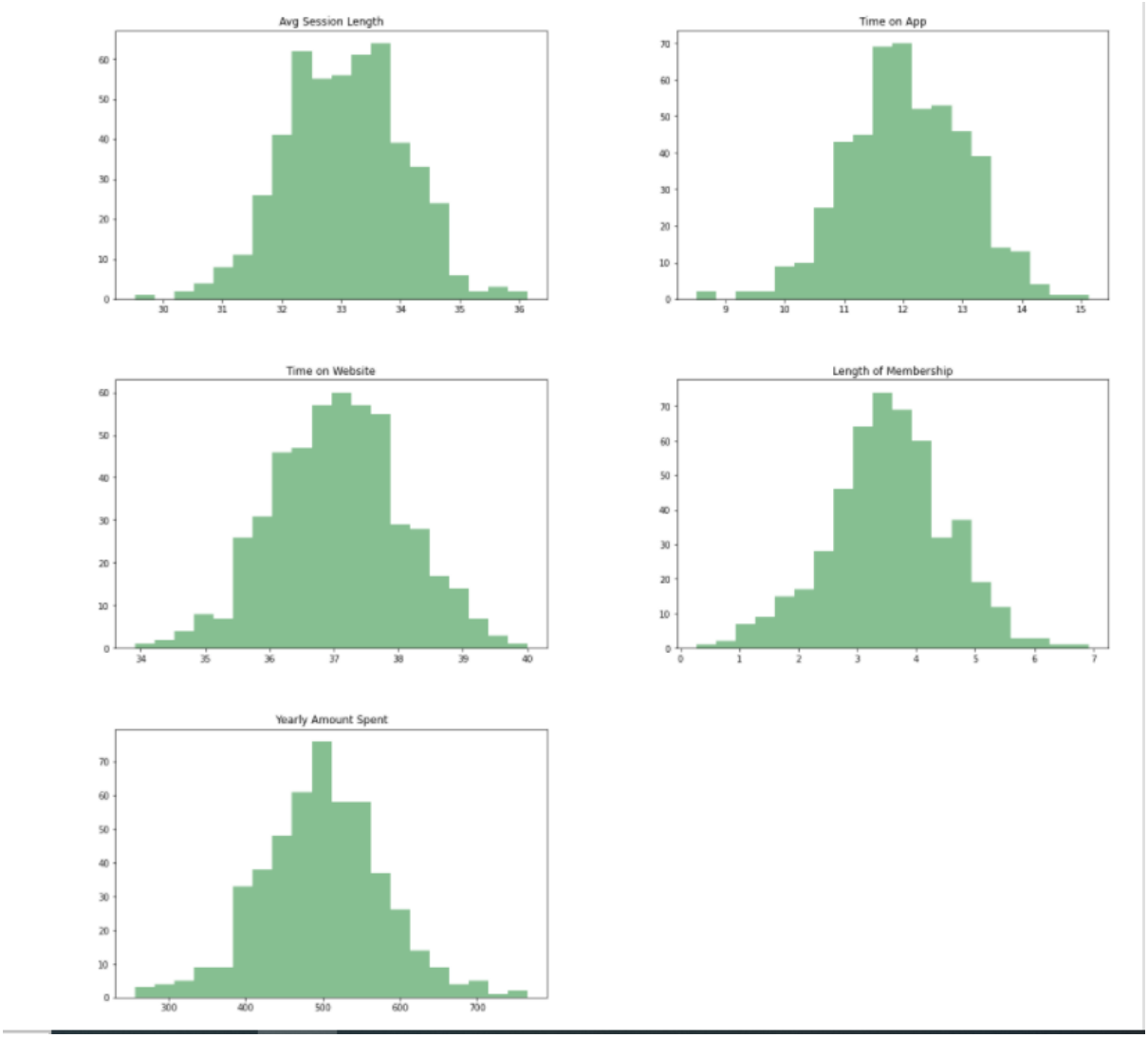
```
+-----------------+---------------+------+-----------------+-----------+---------------+------------------+-----------------
--+
|            Email|        Address|Avatar|Avg_Session_Length|Time_on_App|Time_on_Website|Length_of_Membership|Yearly_Amount_Spe
nt|
+-----------------+---------------+------+-----------------+-----------+---------------+------------------+-----------------
--+
|      Wrightmouth| MI 82180-9605"|Violet|      34.49726773|12.65565115|    39.57766802|         4.082620633|         587.9510
54|
|North Christopher|      RI 60962"|Violet|      33.73264839|12.13879388|    36.85388246|         1.623419609|        399.98387
16|
|       Jonesshire|      GU 33532"|Violet|      33.66661568|10.98576379|    36.35250277|         0.936497597|        304.13559
16|
+-----------------+---------------+------+-----------------+-----------+---------------+------------------+-----------------
--+
```

```
#Query data with > 300 as Yearly Amount Spent
cust_data.filter(cust_data["Yearly_Amount_Spent"] > 300).show(10)
```

```
+---------------+---------------+---------------+-----------------+-----------+---------------+------------------+---------
----------+
|          Email|        Address|          Avatar|Avg_Session_Length|Time_on_App|Time_on_Website|Length_of_Membership|Yearly_Am
ount_Spent|
+---------------+---------------+---------------+-----------------+-----------+---------------+------------------+---------
----------+
|    Wrightmouth| MI 82180-9605"|         Violet|      34.49726773|12.65565115|    39.57766802|         4.082620633|
587.951054|
|    Diazchester| CA 06566-8576"|      DarkGreen|      31.92627203|11.10946073|    37.26895887|         2.664034182|        3
92.2049334|
|    Cobbborough| DC 99414-7564"|         Bisque|      33.00091476|11.33027806|    37.11059744|         4.104543202|        4
87.5475049|
|     Port Jason| OH 22070-1220"|    SaddleBrown|      34.30555663|13.71751367|    36.72128268|         3.120178783|
581.852344|
```

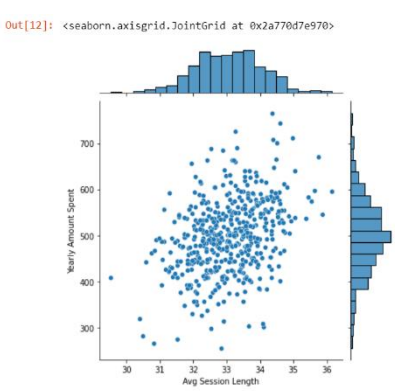
Exploratory Data Analysis

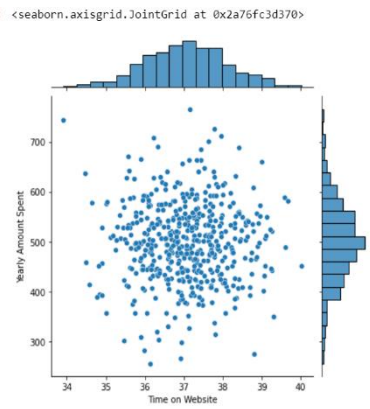
- Univariate information perception plots assist us with appreciating the enumerative properties just as an illustrative synopsis of the specific information variable. These plots help in understanding the area/position of perceptions in the information variable, its circulation, and scattering. Here we can see each of the categories Avg Session Length, Time on App, Time on Website, Length of Membership, Yearly Amount Spent.

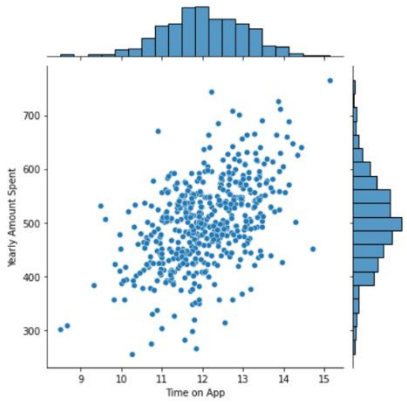


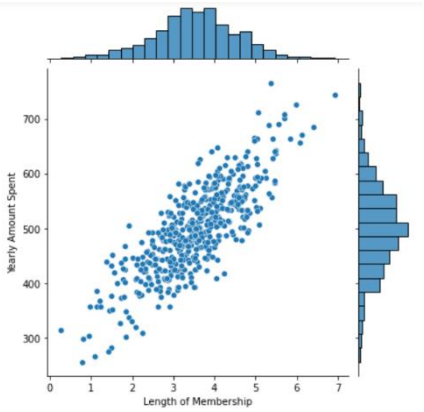
Univariate information perception plots

- Multivariate Analysis is performed to comprehend connections between various fields in the dataset (or) discovering cooperation between factors more than two. Here we can check how this Yearly Amount Spent is in relation with other variables.

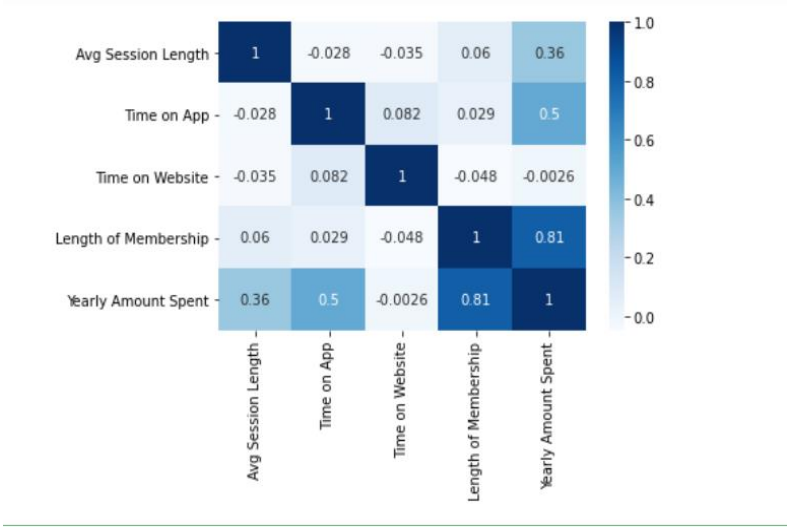








- In order to check the how closely the variables are correlated to each other. That is, it explains how one or more variables are related to each other. Thus, correlation analysis helped to quantify the degree to which the attributes are related. On plotting the data, I was able to find a Linear relationship between attributes. The same is defined in summary of correlation between variables by heatmap representation below.



APPLYING MACHINE LEARNING

Now we have analysed our data and can see that how Customers Yearly Amount Spent is most important and how it depends on the other variables like Avg Session Length, Time on App, Time on Website, Length of Membership. Our aim is to predict the yearly amount spent based on the independent variables. So, I am using the Machine Learning Algorithm for the same here. As we know Machine Learning is one of the numerous applications of Artificial Intelligence (AI) where the essential point is to empower PCs to adapt naturally with no human help. With the assistance of Machine Learning, PCs can handle the assignments that were, up to this point, just took care of and did by individuals. It's anything but an interaction of encouraging a framework on the most proficient method to make precise forecasts when taken care of with the right information. It can take in and improve from past experience without being explicitly modified for an undertaking. AI basically centres around creating PC projects and calculations that make expectations and gain from the given information.
We have a set of different types of ML algorithms available which predict with very good rate of accuracy. I am choosing here a classification algorithm which is one of the most widely used Linear Regression Algorithm. Since it is a predictive analysis algorithm based on the Probability concept. This is a powerful machine learning algorithm that works best on binary classification problems and multi-class classification problems. As the first step we need to get one vector as an independent variable and one dependant variable using Pyspark Libraries as below.

- A vector as an independent variable and one dependant variable using Pyspark Libraries

```
from pyspark.ml.linalg import Vectors
from pyspark.ml.feature import VectorAssembler
```

```
featureassembler=VectorAssembler(inputCols=["Avg_Session_Length","Time_on_App","Time_on_Website","Length_of_Membership"],
                                 outputCol="Independent_Features")
```

```
featureassembler.setHandleInvalid("skip").transform(cust_data).show
```

```
<bound method DataFrame.show of DataFrame[Email: string, Address: string, Avatar: string, Avg_Session_Length: double, Time_on_A
pp: double, Time_on_Website: double, Length_of_Membership: double, Yearly_Amount_Spent: double, Independent_Features: vector]>
```

```
output=featureassembler.transform(cust_data)
```

```
output.select("Independent_Features")
```

```
DataFrame[Independent_Features: vector]
```

- Splitting Dataset before applying the algorithm for training and training purpose.

```
#Splitting Dataset
train_data,test_data=finalized_data.randomSplit([0.75,0.25])
```

- Applying the Linear Regression

On Applying the Linear Regression, the supervised machine learning algorithm where the predicted output is continuous and has a constant slope. It's used to predict values within a continuous range, then trying to classify them into categories. Here we are predicting the Yearly amount spent for Independent Features. The Independent Features are Avg_Session_Length, Time_on_App, Time_on_Website, Length_of_Membership. We are able to predict them as below. Thus, this is a good prediction Model.

```
regressor=LinearRegression(featuresCol='Independent_Features', labelCol='Yearly_Amount_Spent')
regressor=regressor.fit(train_data)
```

```
regressor.coefficients
```

```
DenseVector([25.4803, 38.0527, 0.3847, 61.8461])
```

```
regressor.intercept
```

```
-1034.4315279744703
```

```
pred_results=regressor.evaluate(test_data)
```

```
pred_results.predictions.show(40)
```

```
+------------------+------------------+-----------------+
|Independent_Features|Yearly_Amount_Spent|       prediction|
+------------------+------------------+-----------------+
|[30.39318454,11.8...|        319.9288698|331.97710083094853|
|[30.57436368,11.3...|        442.0644138|443.04535659140174|
|[30.97167564,11.7...|        494.6386098| 488.7767235429217|
|[31.06621816,11.7...|        448.9332932|462.61820685111456|
|[31.12397435,12.3...|        486.9470538|508.89023708518835|
```

DISCUSSION AND CONCLUSION

From the above set of data processing and visualization we are able to analyse the Customer details Email, Address, and Avatar which are string and the Avg Session Length, Time on App, Time on Website, Length of Membership etc are the double, that how these data are in relationships between Yearly Amount Spent and check their corelation. We were able to gather that information that all these are in corelation with the Yearly Amount Spent variable. We know that most of the customers choose different ways in using the services many on App some in website and based on these dependencies we are able to predict the Yearly amount spend by the customer as discussed above. On further research we can also try applying other algorithms and check how well the predictions can be done and make the system more accurate. Thus, I can conclude that the system can very well predict the yearly usage and the amount spend.

REFERENCES

**** Dataset Link : https://www.kaggle.com/srolka/ecommerce-customers***

****Code Uploaded link:***
https://drive.google.com/drive/folders/1YIjTCJBZTBCdNAjS91PxyrIdKkCipvjJ?usp=sharing

- ✓ https://spark.apache.org/docs/latest/index.html
- ✓ Chang Hsin Lee (December 30, 2017), How to Install and Run PySpark in Jupyter Notebook on Windows[Online]
  https://changhsinlee.com/install-pyspark-windows-jupyter/
- ✓ Cory Maklin (June30 2019),Spark MLlib Python Example — Machine Learning At Scale [Online]
  https://towardsdatascience.com/machine-learning-at-scale-with-apache-spark-mllib-python-example-b32a9c74c610
- ✓ rajeevsinghAD (Mar 14, 2018),MatchError: null while trying to read a null value. [Online]
  https://github.com/Azure/azure-cosmosdb-spark/issues/167
- ✓ Karlijn Willems (July 28th, 2017),Apache Spark Tutorial: ML with PySpark [Online]
  https://www.datacamp.com/community/tutorials/apache-spark-tutorial-machine-learning#load
- ✓ Wei Song (15 October 2021), Machine Learning Methods and Big Data Analytics in Structural Health Monitoring [Online]
  https://www.frontiersin.org/research-topics/16803/machine-learning-methods-and-big-data-analytics-in-structural-health-monitoring
- ✓ Exercise 6 - Linear Regression (Python)
  [Online] https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/1779476228152266/1437143569842658/5673666086694627/latest.html
- ✓ GUEST BLOG (JULY 29, 2020), Univariate Data Visualizations With Illustrations in Python [Online] https://www.analyticsvidhya.com/blog/2020/07/univariate-analysis-visualization-with-illustrations-in-python/