**7082CEM Big Data Management and Data Visualization**

**Coursework**

**Dataset Analysis and Visualization Using Big Data Programs**

**Module Leader:** Dr. Marwan Fuad

**Student Name:** Bharathi Cherukuri

**Student ID:** 12088229

**Project Title:** Sales Analysis

**Dataset Name:** Supermarket Sales

**Dataset Link:** <https://www.kaggle.com/datasets/aungpyaeap/supermarket-sales>

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

**Abstract**

In this project, I have selected the supermarket sales dataset for analysis. Here I am going to explain and demonstrate how to set up and use PySpark for one of the popular Machine Learning algorithm i.e., Regression. Three methods that are used and compared in this paper are: Linear Regression, Decision Tree Regression and Gradient-boosted tree Regression.

Big Data Analysis and Machine Learning algorithms were applied using PySpark and using Tableau on this dataset for analysing the columns and will also attach the screenshots of visualizations. The analysis mainly focus on the columns like branch, city, product line, gender, customer type, payment, Gross income etc. for the betterment of future sales.

**Contents**

1. **Introduction**
2. **Details about Dataset**
3. **Data Analysis using PySpark**
4. **Data Visualization Using Tableau**
5. **Conclusions from the Analysis**
6. **References**
7. **Introduction**

This report was written as part of the 7082CEM Big Data Management and Data Visualisation Course work: Dataset Analysis and visualization using Big Data Programs.

This project focuses of demonstration of different analytical techniques for different structured and unstructured big data sets to support decision-making, identifying and selecting appropriate analytical techniques for big data analysis and applying appropriate methods that are suitable for visualizing big data.

The project involves application of PySpark, basically Spark in addition to Python Programming Language and Tableau which is a visual analytics platform transforming the way we use the data to solve problems.

The main aim of this project is to analyse Supermarket Sales Dataset and to achieve the below objectives:

1. Use PySpark to load and process the dataset
2. Explore and analyse data using Regression
3. Visualize data using Tableau
4. discuss the findings from PySpark and T`ableau
5. **Details about the Dataset:**

The dataset is one of the sales of Supermarket Company which has recorded in 3 different branches 1)Branch A (Yangon) 2) Branch B (Mandalay) 3)Branch C (Naypyitaw) for 3 months from Jan-March 2019.The dataset contains 17 columns and 1000 rows. The datatype of the columns contains character, strings, integer, float etc.

The below table provides a summary of all the variables and the variable description:

|  |  |  |
| --- | --- | --- |
| Variable/Column name | Variable/Column Data type | Variable / Column Description |
| Invoice id | String | Computer generated sales slip invoice identification number |
| Branch | Character | 3 branches are available identified by A, B and C |
| City | String | Location of  supermarket |
| Customer type | String | Type of customers, recorded by Members for customers using member card and Normal for without member card. |
| Gender | String | Gender type of customer |
| Product line | String | General item categorization groups - Electronic accessories, Fashion accessories, Food and beverages, Health and beauty, Home and lifestyle, Sports and travel |
| Unit price | Float | Price of each product in $ |
| Quantity | Integer | Number of products purchased by customer |
| Tax | Float | 5% tax fee for customer buying |
| Total | Float | Total price including tax |
| Date | String | Date of purchase (Record available from January 2019 to March 2019) |
| Time | String | Purchase time (10am to 9pm) |
| Payment | String | Payment used by customer for purchase (3 methods are available – Cash, Credit card and Ewallet) |
| COGS | Float | Cost of goods sold |
| Gross margin percentage | Float | Gross margin percentage |
| Gross income | Float | Gross income |
| Rating | Float | Customer stratification rating on their overall shopping experience (On a scale of 1 to 10) |

1. **Details about PySpark Analysis:**

PySpark is an API of Apache Spark which is an open-source, distributed processing system used for big data processing which is developed in Scala programming language. The Spark has development APIs in Scala, Java, Python and R and supports code reuse across multiple workloads.

For this Project, I Downloaded VMware workstation 16.2.3 and Ubuntu-22.04-64 bit and installed VMware with Ubuntu. As a part of lab sessions I downloaded Hadoop-2.7.3 and spark-2.3.0-bin-hadoop2.7 and installed Java, Hadoop and Spark in the terminal of my virtual machine. I installed Python 3.10 and installed PySpark in the terminal of my Virtual machine. After the installation I can able to enter to the PySpark session in the terminal itself and can able to code.

**Step -1: Java Version:**

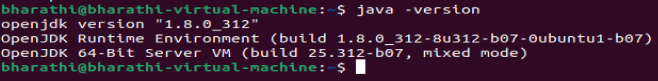
Command:

sudo apt –get remove openjdk\*

sudo apt install openjdk-8-jdk -y

java -version

Screenshot:



**Step -2: Hadoop Installation:**

Command:

export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64

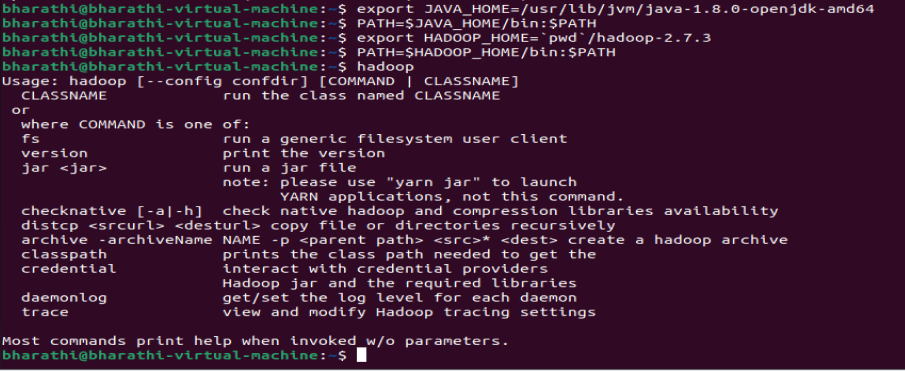
PATH=$JAVA\_HOME/bin:$PATH

export HADOOP\_HOME=`pwd`/hadoop-2.7.3

PATH=$HADOOP\_HOME/bin:$PATH

hadoop

Screenshot:



**Step -3: Spark Installation:**

Command:

export SPARK\_HOME=`pwd`/spark-2.3.0-bin-hadoop2.7

PATH=$SPARK\_HOME/bin:$PATH

Spark-shell

Screenshot:



**Step -4: Python Version:**

Command:

sudo apt update

sudo apt install python3

python3 –version

Screenshot:

****

**Step -5: PySpark and jupyter notebook Installation:**

Command: open ~/.bashrc file and add below path and save it

alias python=python3

export PYTHONPATH=$SPARK\_HOME/python:$SPARK\_HOME/python/lib/py4j-0.10.7-src.zip:$PYTHONPATH

export PYSPARK\_PYTHON=python3

After this logout and login again.

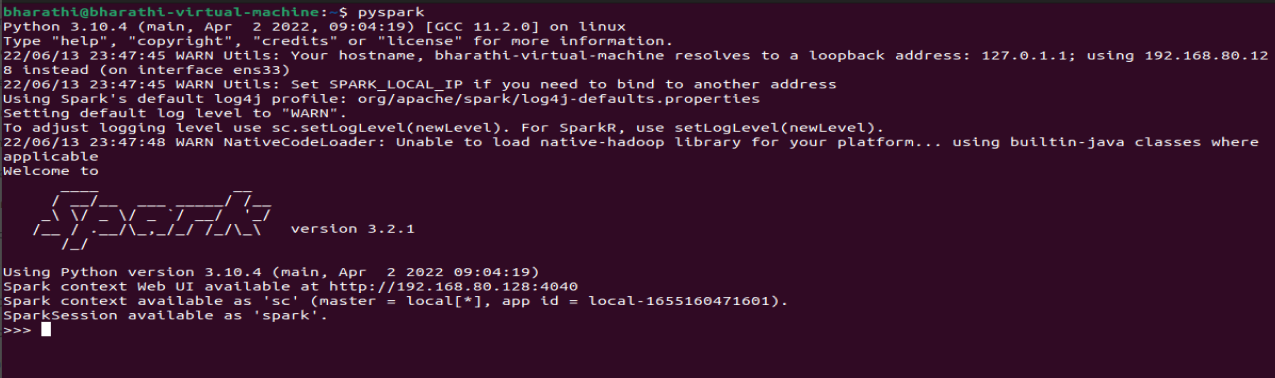
Pyspark

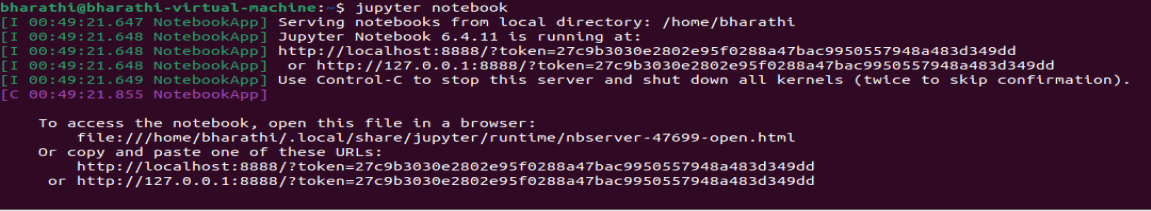
Ctrl + z

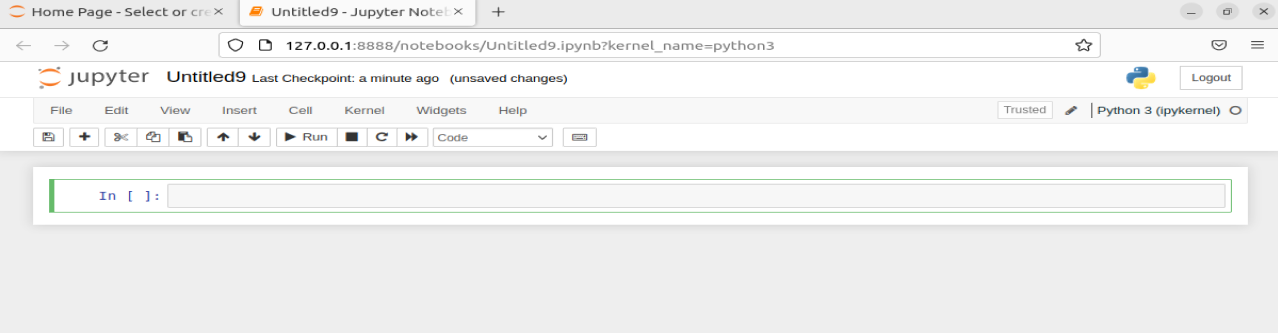
pip install jupyter

jupyter notebook

Screenshot:







After successful installation of PySpark session, I downloaded the dataset from kaggle which was selected and submitted in the Project Proposal and copied the csv file to the same folder which I saved my spark files.

**Step6: Importing the libraries and Read csv file:**

Command:

import pyspark

from pyspark.sql import SparkSession

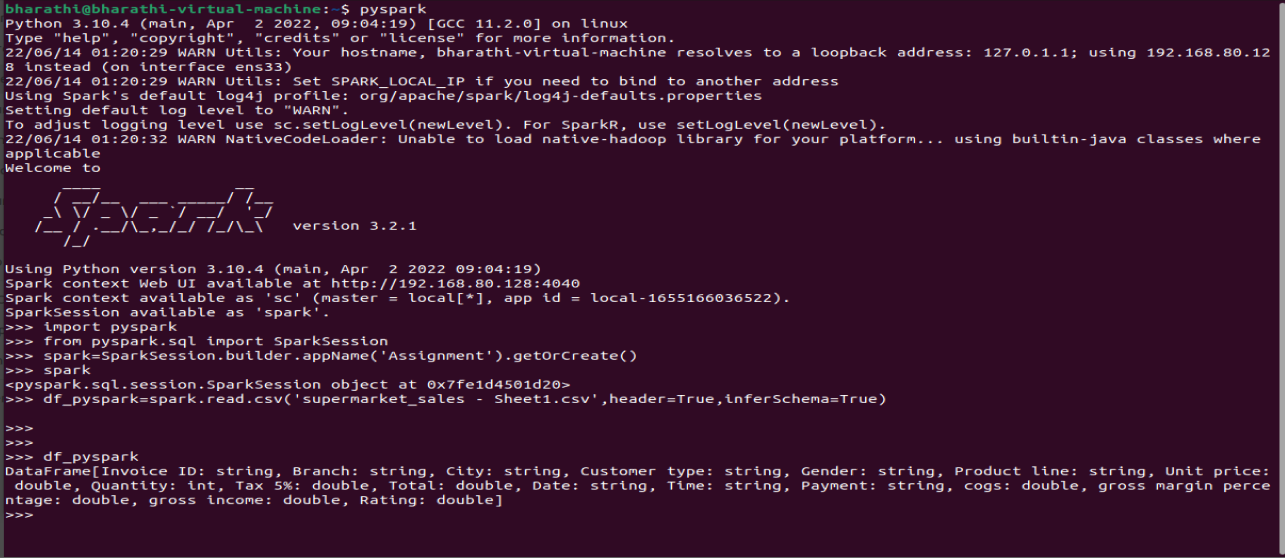
spark = SparkSession.builder.appName(‘Assignment’).getorCreate()

spark

df\_pyspark = spark.read.csv(‘supermarket\_sales - sheet1.csv’,header=True,inferSchema=True)

df\_pyspark

Screenshot:



Command:

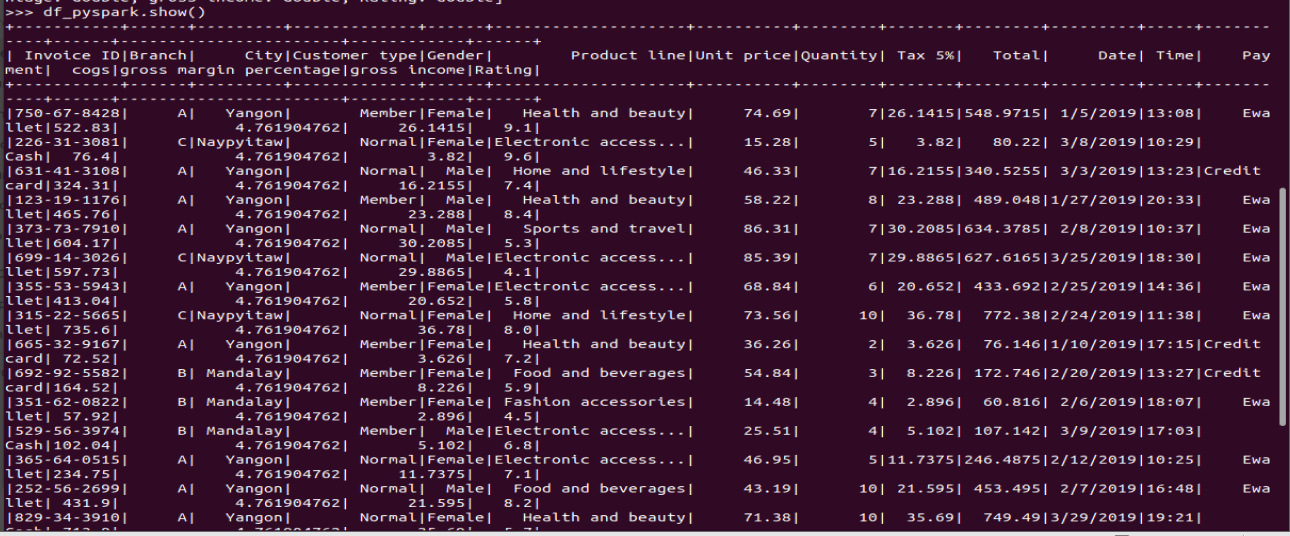
df\_pyspark.show()

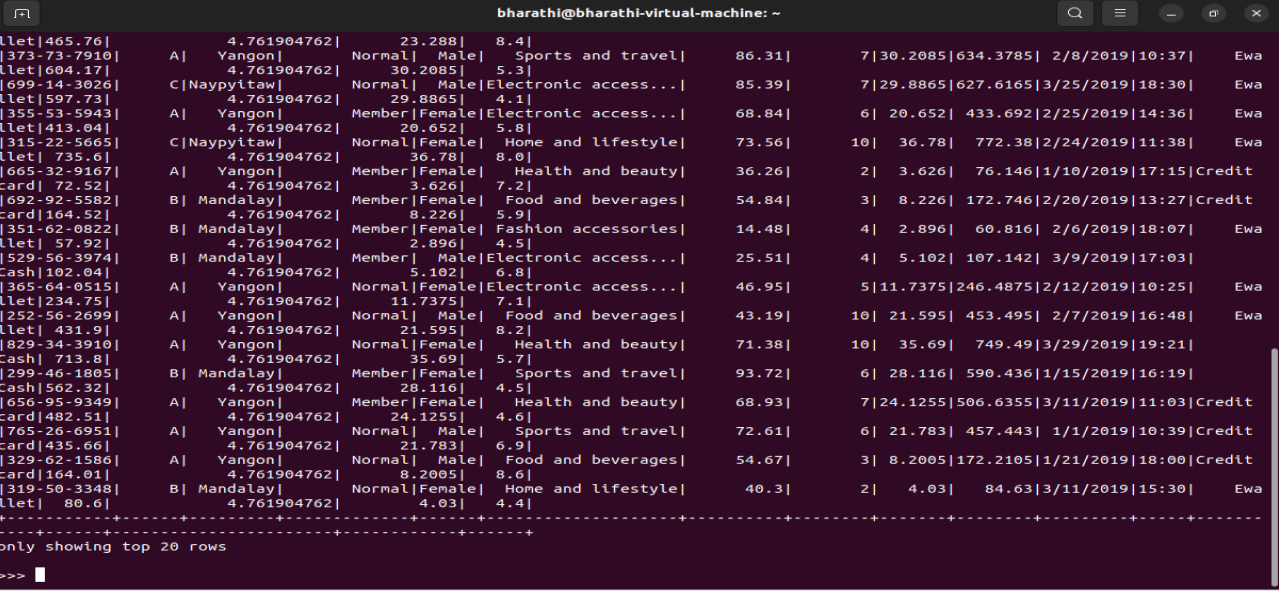
type(df\_pyspark)

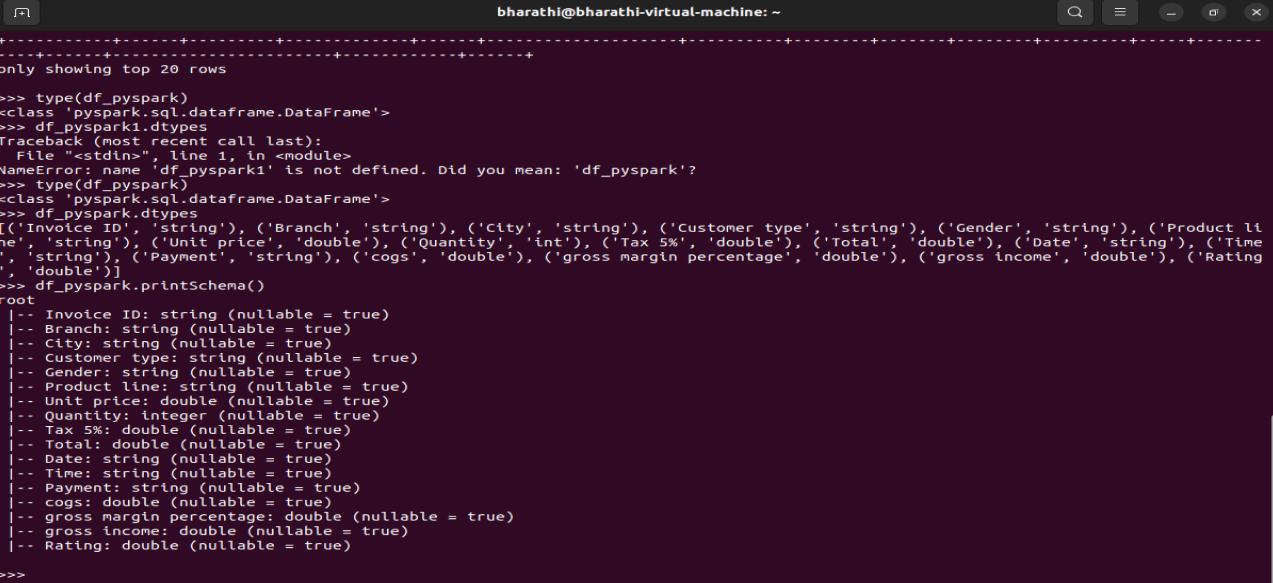
df\_pyspark.dtypes

df\_pyspark.printSchema()

Screenshot:







**Step7: Dropping columns which are not use full for our analysis**

For our analysis we don’t need some of the columns.so I am dropping those columns Invoice Id, Time, Date, Branch and gross margin percentage.

Branch and city represents same. So I am dropping branch column also. Group margin percentage is also not useful for our analysis because we have same value for all 1000 rows.so dropping that column also.

Command:

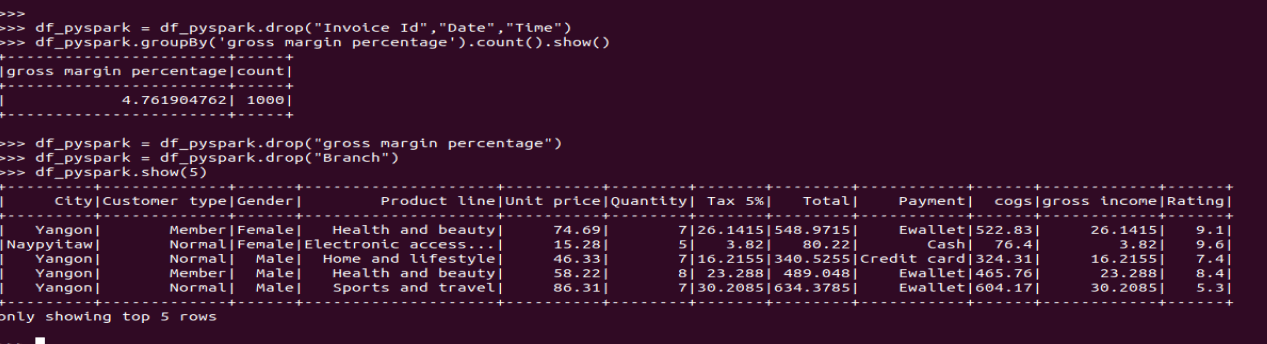
df\_pyspark=df\_pyspark.drop(“Invoice Id”,”Date”,”Time”)

df\_pyspark.groupBy(‘group margin percentage’).count().show()

df\_pyspark=df\_pyspark.drop(“Branch”)

df\_pyspark.show(5)

Screenshot:



**Step8: Total count of values grouped by columns**

Command:

df\_pyspark.groupBy(‘Gender’).count().show()

df\_pyspark.groupBy(‘Product line’).count().show()

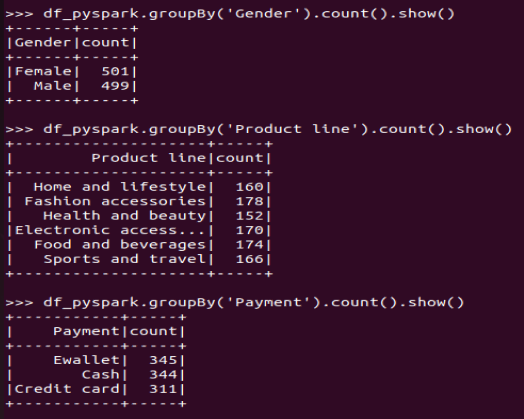
df\_pyspark.groupBy(‘Payment’).count().show()

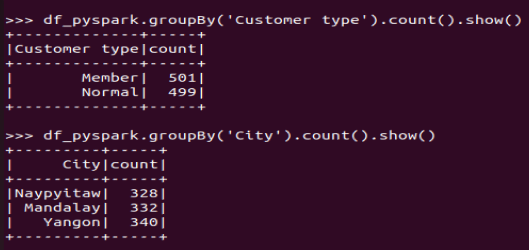
df\_pyspark.groupBy(‘Customer type’).count().show()

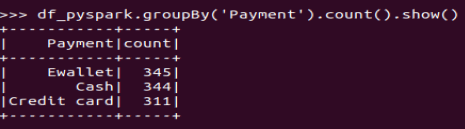
df\_pyspark.groupBy(‘City’).count().show()

df\_pyspark.groupBy(‘Payment).count().show()

Screenshot:







**Step9: adding new columns by converting string to float using String Indexer**

As we have columns of all datatypes like string, character, integer, float etc. we have to convert all other columns to float datatype for our analysis.

Command:

from pyspark.ml.feature import StringIndexer

indexer = StringIndexer(inputCol=”Product line”,outputCol=”Product\_cat”)

indexed = indexer.fir(df\_pyspark).transform(df\_pyspark)

indexer = StringIndexer(inputCol=”City”,outputCol=”City\_cat”)

indexed1 = indexer.fir(indexed).transform(indexed)

indexer = StringIndexer(inputCol=”Gender”,outputCol=”Gender\_cat”)

indexed2 = indexer.fir(indexed1).transform(indexed1)

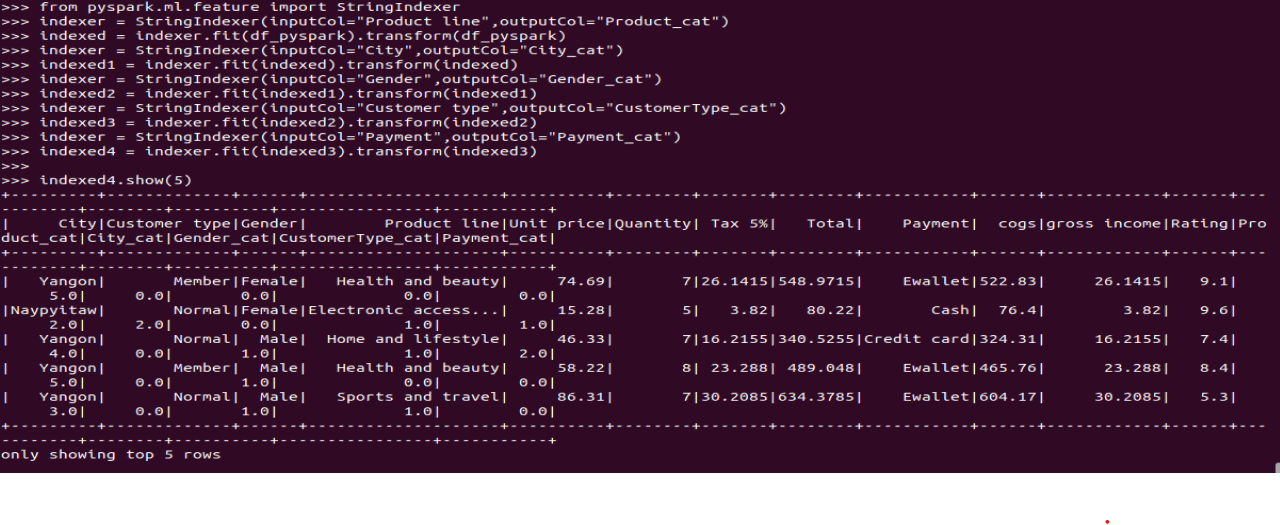
indexer = StringIndexer(inputCol=”Customer type”,outputCol=”customerType\_cat”)

indexed3 = indexer.fir(indexed2).transform(indexed2)

indexer = StringIndexer(inputCol=”Payment”,outputCol=”Payment\_cat”)

indexed4 = indexer.fir(indexed3).transform(indexed3)

indexed4.show(5)



**Step10: Importing Vector Assembler and creating a new dataframe**

Vector Assembler is a transformer that combines a given list of columns into a single vector column. Its useful for combining raw features and features generated by different feature transformers into a single feature vector, in order to trail ML models.

Command:

from pyspark.ml.linalg import Vector

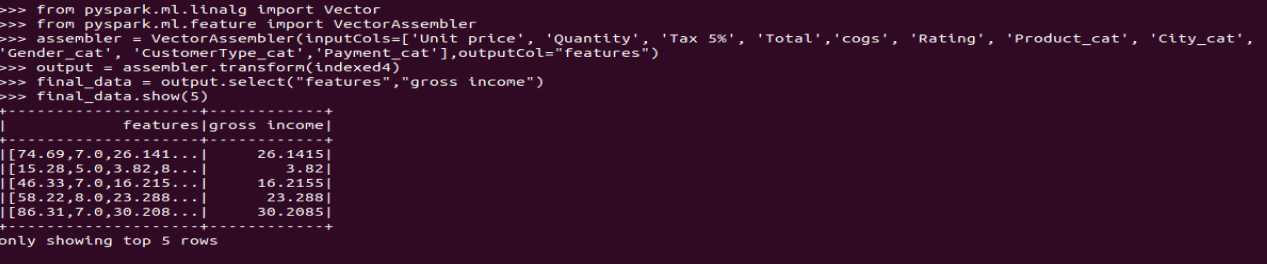
from pyspark.ml.feature import VectorAssembler

assembler = VectorAssembler(inputCols=[‘Unit price’,’Quantity’,’Tax 5%’,’Total’,’cogs’,’Rating’,’Product\_cat’,’City\_cat’,’Gender\_cat’,’CustomerType\_cat,’Payment\_cat’],outputCol=”features”)

output = assembler.transform(indexed4)

final\_data = output.select(“features”,”gross income”)

final\_data.show()



**Step11: Linear Regression**

A method to help us understand the relation between two variables. The predictor (independent) variable X and the target (dependent) variable y.

For Linear Relationship between the variables y = b+ax, the parameter b is the intercept and a is the slope. To fit the model in PySpark, first we import LinearRegression from pyspark.ml.regression; then create linear regression object using the constructor. We define the predictor variable and target variable. Then use the method fit to fit the model and find the parameters b and a. the input are the features and the targets.we can obtain a prediction using the method predict. The intercept b and slope a are attributes of the object model.

Multi Linear Regression is used to explain the relation between one continuous target variable (y) and two or more predictor (x) variables. If we have 4 predictor variables then our model looks like Y = b0+b1 X1 + b2 X2 + b3 X3 + b4 X4 .We can fit the multiple regression as follows: we can extract the 4 predictor variables and store them in the variable Z. Then train the model as before using the method train, with the features or depended variables and the target variable. We can also obtain a prediction using the method predict .In this case , the input is an array or data frame with 4 columns, the number of rows correspond to the number of samples.

Two important measures to determine the fit of a model:

* Mean Squared Error (MSE)
* R-squared (R^2)

These measures are a way to numerically determine how good the model fits on our data.

To measure the MSE, we find the difference between the actual value y and the predicted value yhat then square it. R-squared is also called the coefficient of determination. It’s a measure to determine how close the data is to the fitted regression line. The mean square error is perhaps the most intuitive numerical measure for determining if a model is good or not. MSE value zero or close to it and R2 value should be 1 , this means the line is a good fit for the data.

Command:

train\_data,test\_data = final\_data.randomSplit([0.7,0.3])

from pyspark.ml.regression import LinearRegression

lr = LinearRegression(featuresCol=”features”,labelCol=”gross income”)

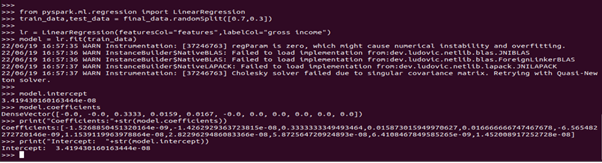
model = lr.fit(train\_data)

model.intercept

model.coefficients

print(“Coefficients:”+str(model.coefficients))

print(“Intercept:”+str(model.intercept))

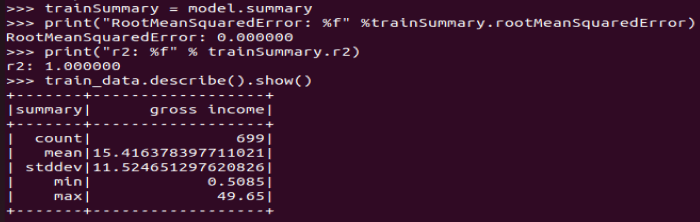


trainSummary = model.summary

print("RootMeanSquaredError: %f" %trainSummary.rootMeanSquaredError)

print("r2: %f" % trainSummary.r2)

train\_data.describe().show()



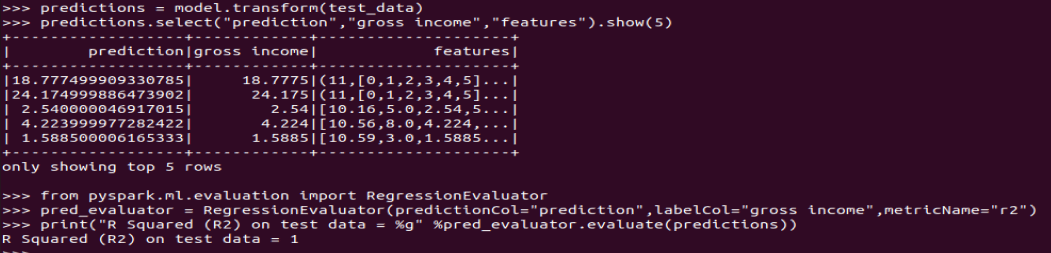
predictions = model.transform(test\_data)

predictions.select("prediction","gross income","features").show(5)

from pyspark.ml.evaluation import RegressionEvaluator

pred\_evaluator = RegressionEvaluator(predictionCol="prediction",labelCol="gross income",metricName="r2")

print("R Squared (R2) on test data = %g" %pred\_evaluator.evaluate(predictions))



res = model.evaluate(test\_data)

print(f"""

Linear Regression results Report

=====================================================

RootMeanSquaresError:\t {res.rootMeanSquaredError}

MeanSquaredError:\t {res.meanSquaredError}

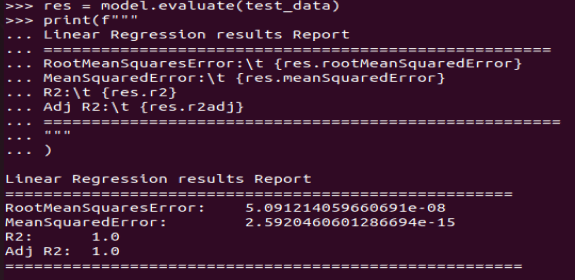
R2:\t {res.r2}

Adj R2:\t {res.r2adj}

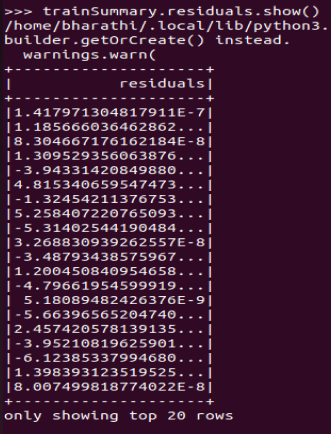
======================================================

"""

)



trainSummary.residuals.show()



Linear Regression Root Mean Squared Error (RMSE) on test data = 5.09

**Step12: Decision tree regression**

from pyspark.ml.regression import DecisionTreeRegressor

dt = DecisionTreeRegressor(featuresCol = 'features',labelCol = 'gross income')

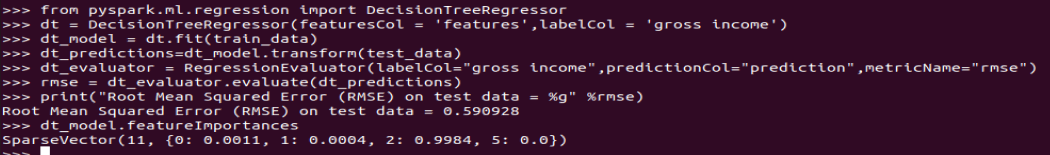
dt\_model = dt.fit(train\_data)

dt\_predictions=dt\_model.transform(test\_data)

dt\_evaluator = RegressionEvaluator(labelCol="gross income",predictionCol="prediction",metricName="rmse")

rmse = dt\_evaluator.evaluate(dt\_predictions)

print("Root Mean Squared Error (RMSE) on test data = %g" %rmse)



Decision Regression Root Mean Squared Error (RMSE) on test data = 0.59

**Step12:Gradient-boosted tree regression**

from pyspark.ml.regression import GBTRegressor

gbt = GBTRegressor(featuresCol = 'features',labelCol = 'gross income',maxIter=10)

gbt\_model = gbt.fit(train\_data)

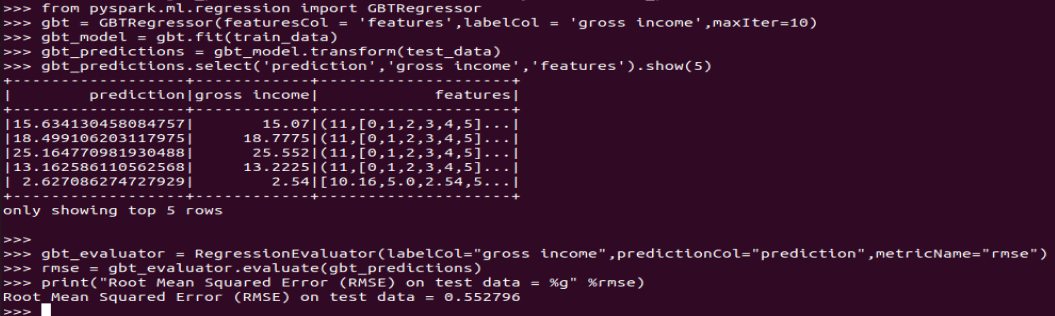
gbt\_predictions = gbt\_model.transform(test\_data)

gbt\_predictions.select('prediction','gross income','features').show(5)

gbt\_evaluator = RegressionEvaluator(labelCol="gross income",predictionCol="prediction",metricName="rmse")

rmse = gbt\_evaluator.evaluate(gbt\_predictions)

print("Root Mean Squared Error (RMSE) on test data = %g" %rmse)



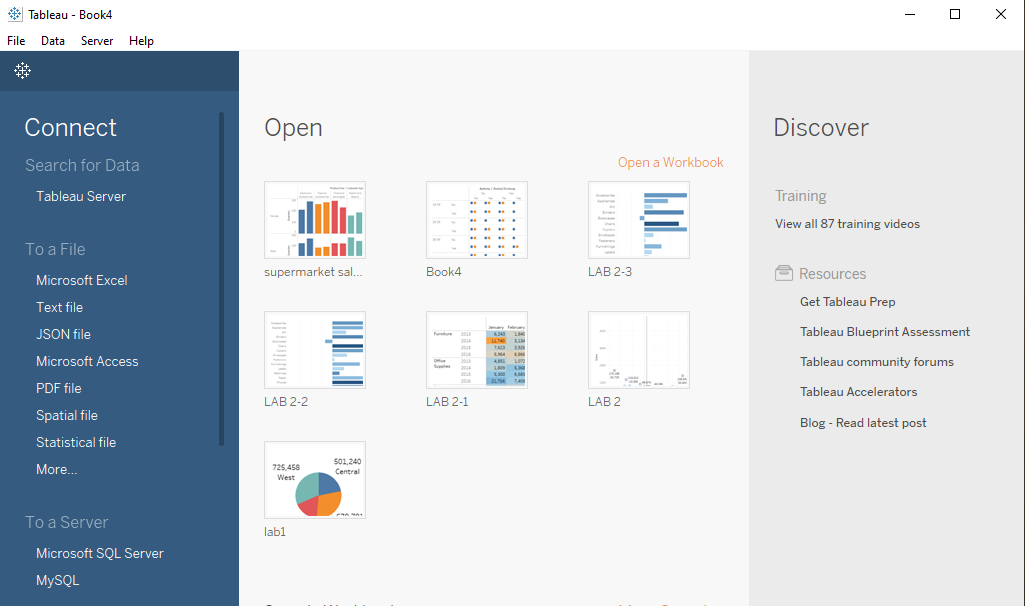
Gradient-boosted Regression Root Mean Squared Error (RMSE) on test data = 0.55

1. **Data Visualization Using Tableau**

For this Project, I downloaded Tableau desktop software 2022-1 version and registered for 1 year free student license key and got license key mailed to the registered mail ID. Then after entering the license key in the tableau desktop, I can able to use Tableau Desktop software for 1 year.

**Step 1: Opening Tableau**

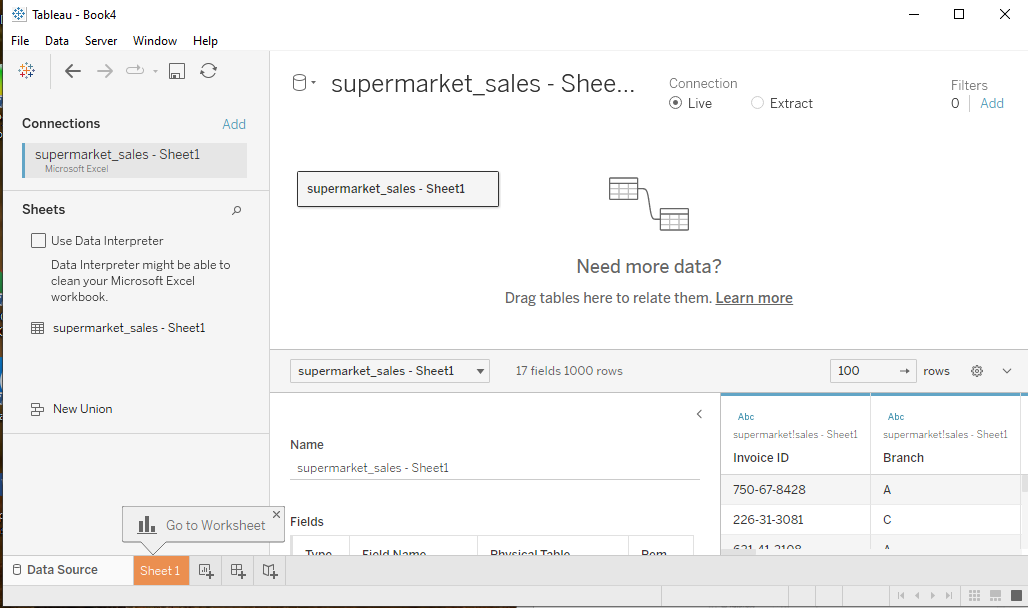
The below screenshot shows the main view of Tableau Desktop. After opening Tableau, we have to open our Dataset from the left hand side Connect bar.



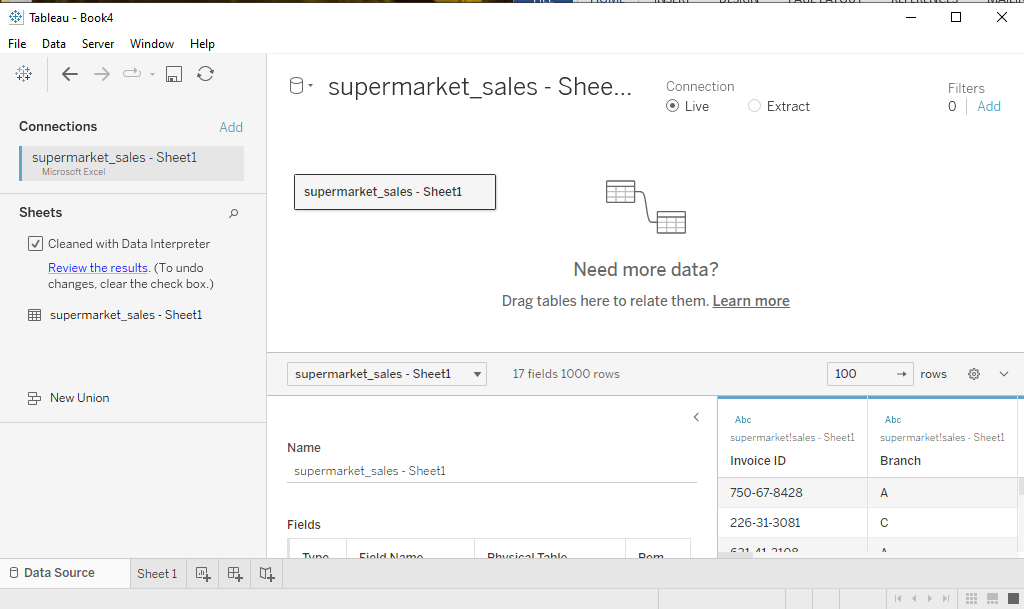
**Step 2: Loading CSV file**

As I have csv file I will be selecting Microsoft excel and open the supermarket\_sales-sheet1 file from the tableau repository which I have saved it before.

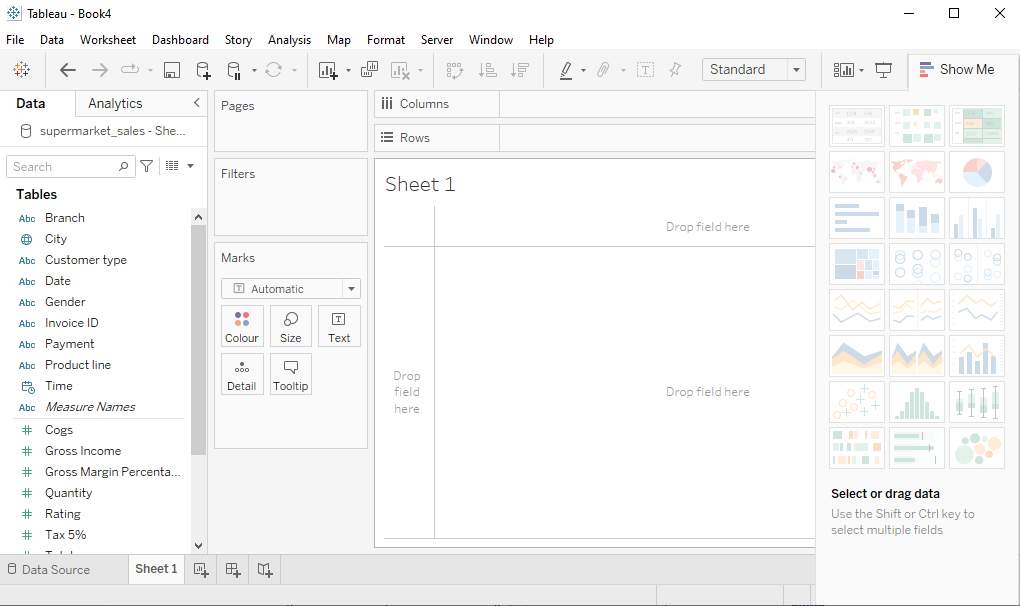
Below screenshot shows the Tableau loaded with all the data from the csv file.



**Step 3: Cleaning the Dataset**

After loading the data set into connect block, we can find our dataset in the connections block and below that we have to select Use Data Interpreter so that the dataset get cleaned and it will be ready to use. 

Next we can go to sheet 1 and start our visualizations. Below screen shows our dataset loaded with all the data with column names.



In the above screenshot, we can create visualizations by dragging the columns (of our dataset which are listed under Tables block) to the columns and rows block, middle part sheet 1 shows the plots of our selection.

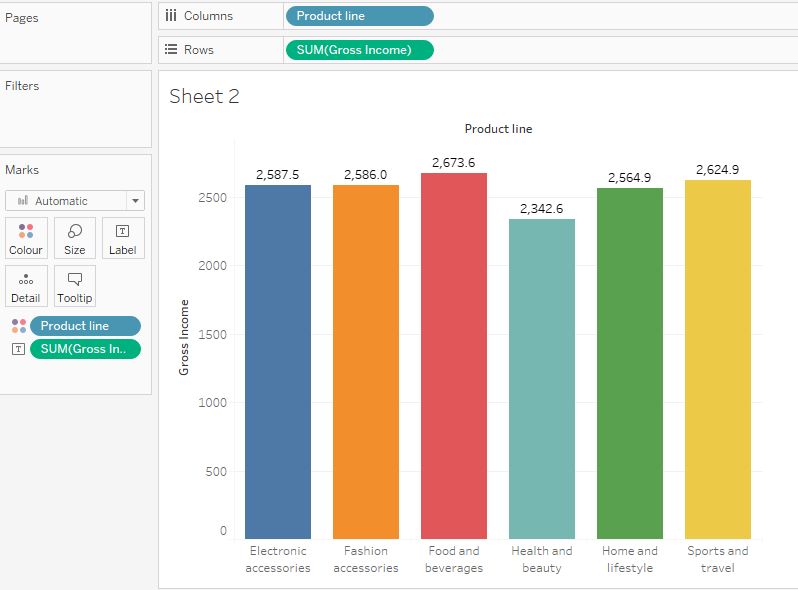
**Step4: Visualizations**

In this project, I visualize a dataset of supermarket sales in Myanmar in three cities: Yangon, Mandalay and Naypyidaw. The dataset contains 3 months sales report. The dataset shows sales of six products: Food and beverages, Sports and travel, Electronic accessories, Fashion accessories, Home and lifestyle and Health and beauty.

**Visualization 1:** Plot between Gross Income and Product Line

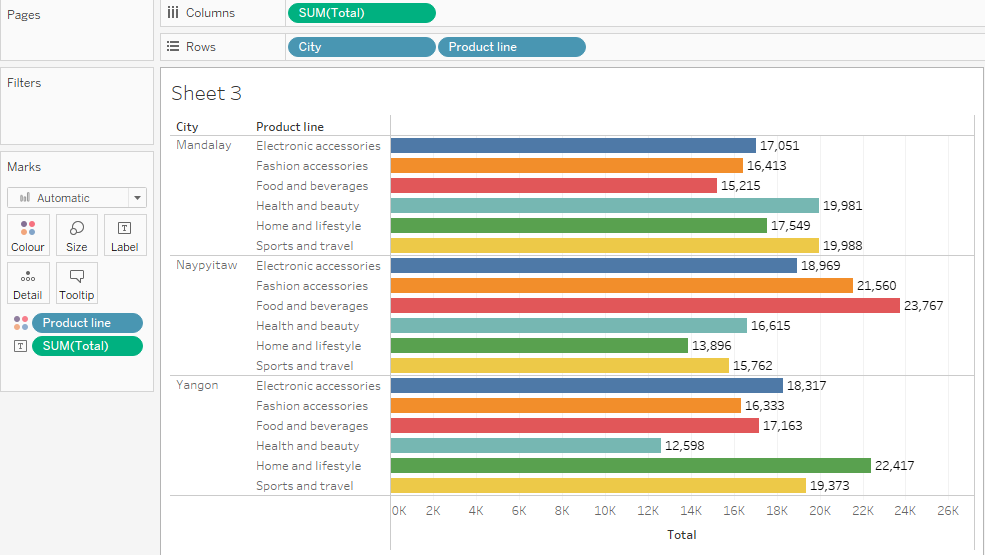
Gross income is the total income earned by the super market after deducting original cost price and taxes. Below plot shows the visualization against the data of product line and Gross income. The below plot states that Food and beverages are seen to have a higher gross income and the least being health and beauty.

With this type of visual we can see the products that are doing well and those that are dragging behind. This helps in coming up in ways to improve the products that are dragging and make them improve the gross income of the supermarket.



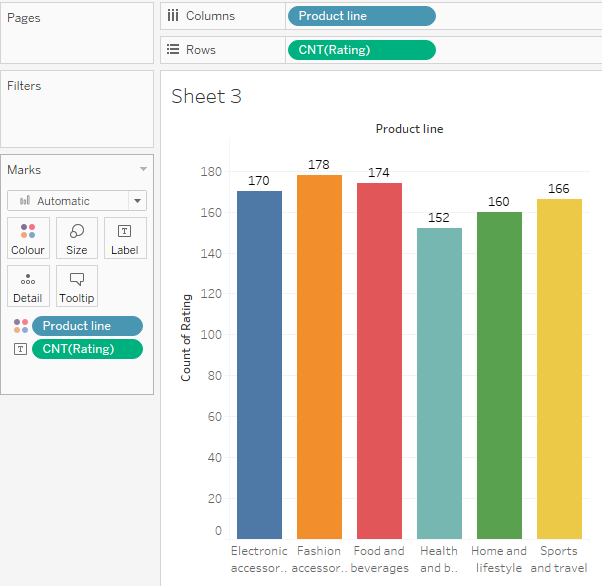
**Visualization 2:** Plot between Product sales in different cities with respect to total price

This is a visual on the performance of products as per the different cities. In Mandalay, the leading product is Sports and travel. In Naypyitaw, the leading product is Food and beverages and in Yangon the leading product is Home and lifestyle. A least product in one city is the leading product in the other city.



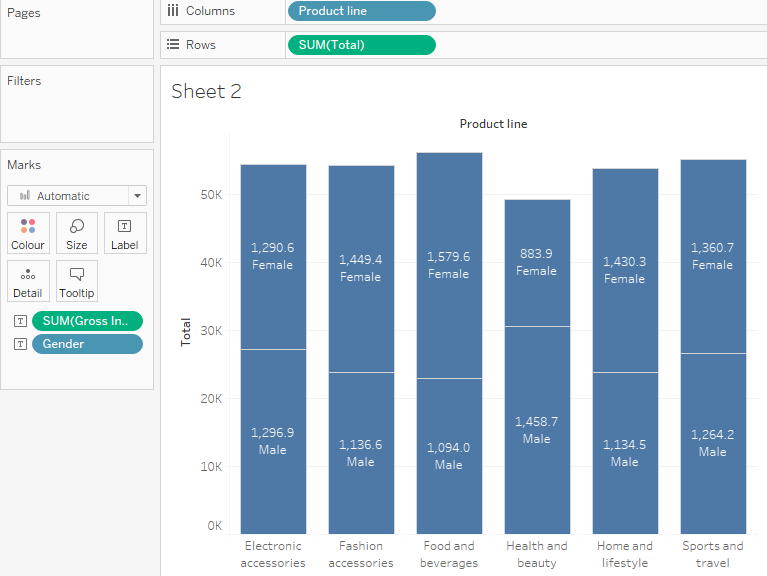
**Visualization 3:** Plot between Count of Ratings and Product Line

Rating is given by customers for the products they purchased in the supermarket. Below plot shows the visualization against the data of product line and Rating. The below plot states that Fashion accessories has high rating compared to other



**Visualization 4:** Plot between Product Line and total price with respect to gender

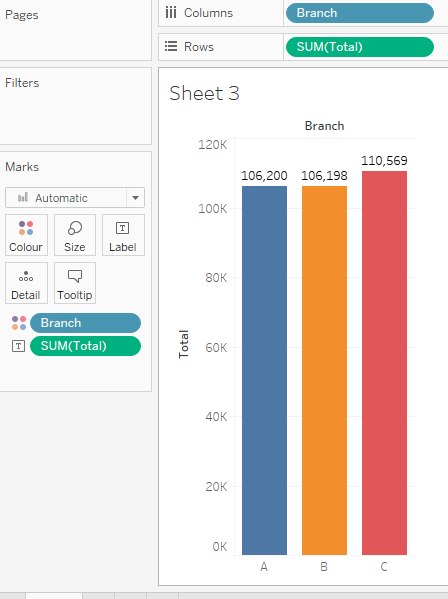
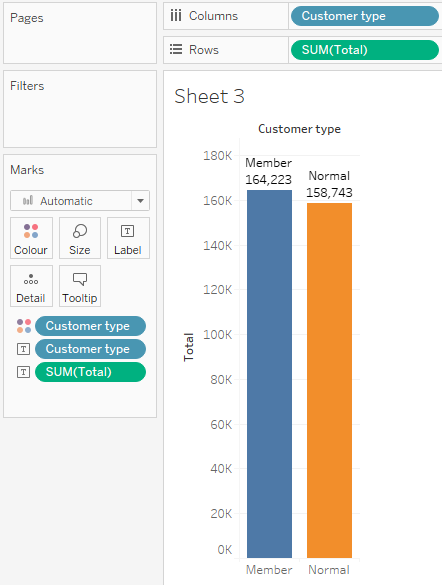
This total sale gender wise with respect to product line. It states that total amount of sale particular gender purchased with respect to product line. By seeing below visual, we can come to know that there is high sale by female for product line Food and beverages and least sale by female in Health and beauty.

****

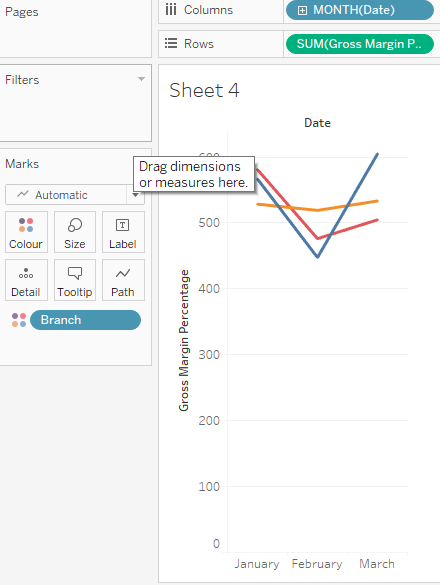
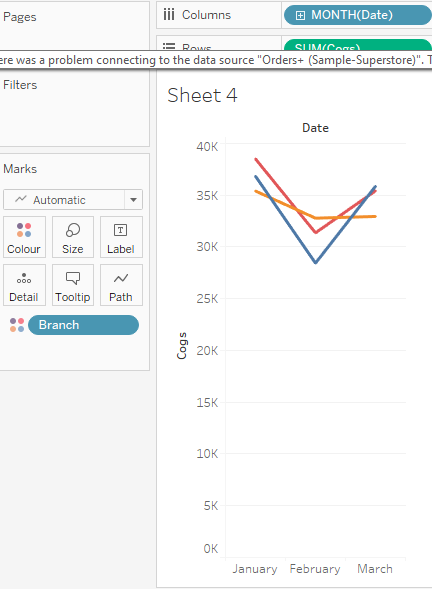
**Visualization 5:** Plot between supermarket branches and total sale and also between customer type and total sale.

This is the branch wise sales report. The supermarket has three branches: A,B and C. Comparing the three branches, C appears to have the highest total sale then A has second highest sale and B is the least.

Here In the plot between customer type and total sale, customer with membership card will tend to purchase more compared to normal customer.

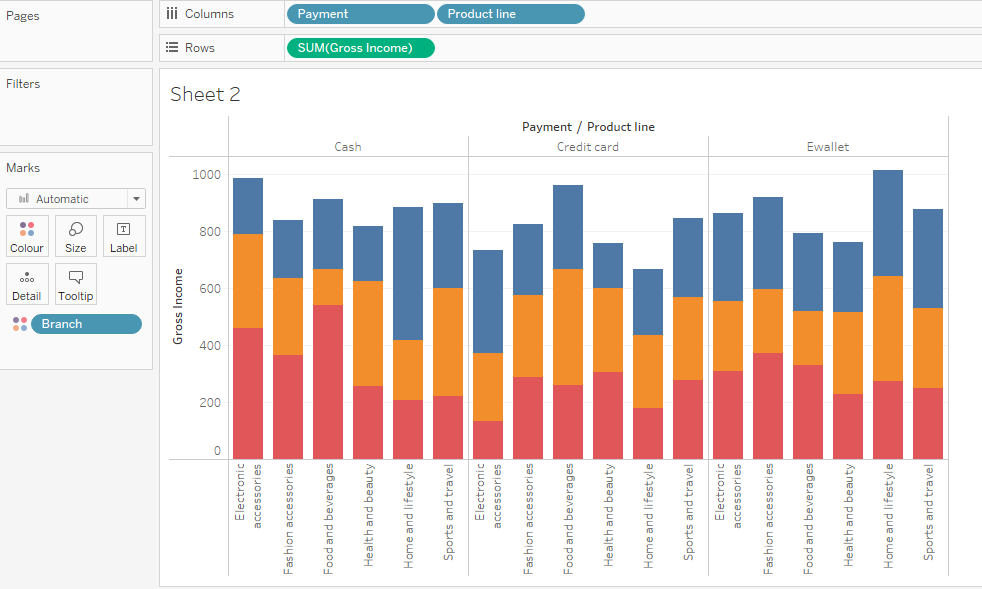
**** 

**Visualization 6:** Plot of Gross Margin Percentage and cost of goods in each branch month wise

**** ****

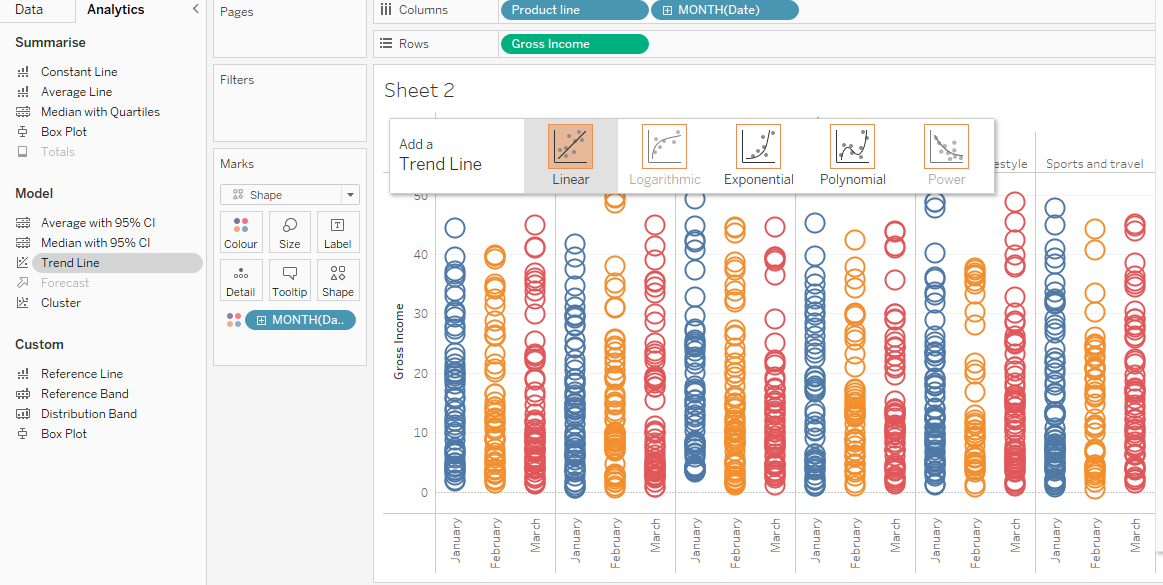
Gross Margin Percentage is the measure of the performance of the business.the above visuals shows the Gross Margin Percentage trend of the different branches. Here we can see that there is a decrease in Gross margin percentage

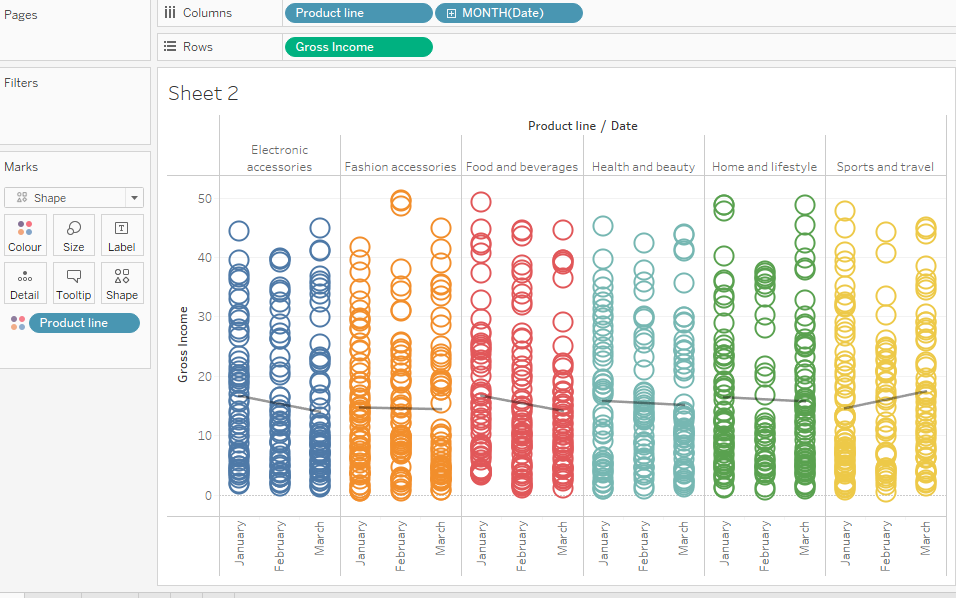
**Visualization 6:** Plot of Gross Income and Product line with respect to payment type

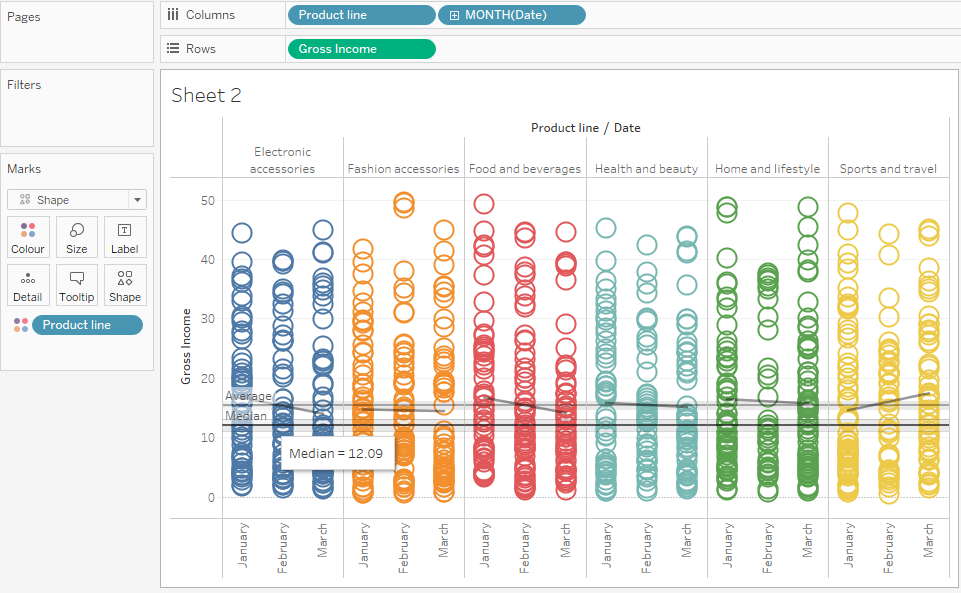


**Visualization 6:** Linear Regression between product line and gross income using Trend line model in Analytics.

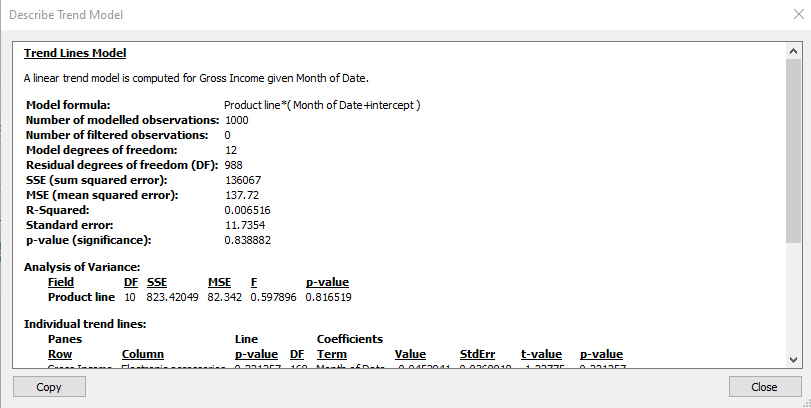
To add a trend line to a visualization first select Analytics pane, from there drag Trend Line into the view, and then drop it on the linear model. Below are the plots of

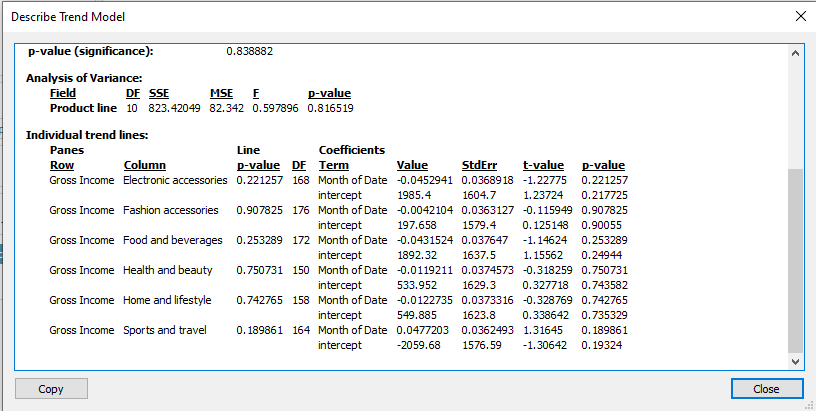


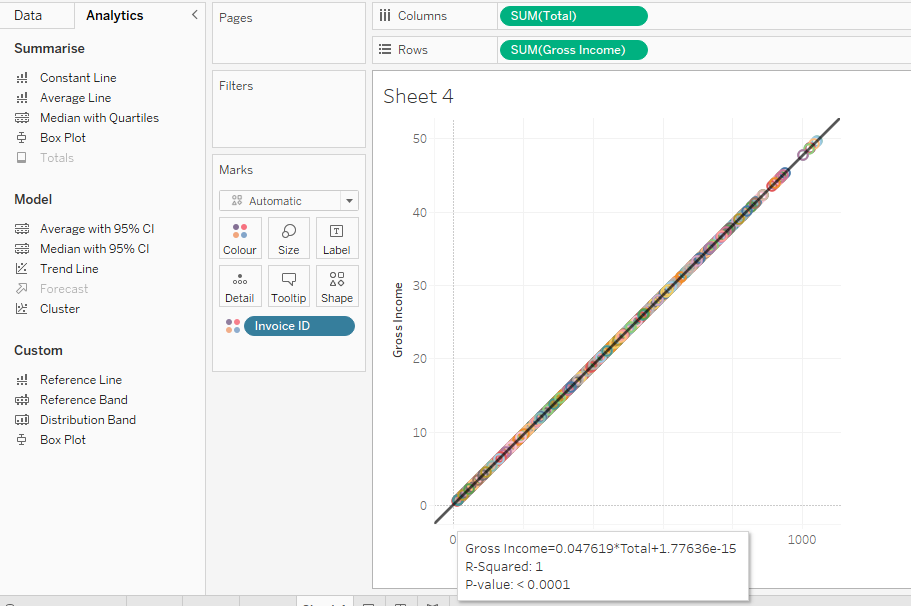


****

****







1. **Conclusions**

**Results of Regression Analysis:**

Three methods that are used and compared in this paper. They are: Linear Regression, Decision Tree Regression and Gradient-boosted tree Regression.

* Linear Regression Root Mean Squared Error (RMSE) on train data = 0
* Linear Regression Root Mean Squared Error (RMSE) on test data = 5.09
* Decision Regression Root Mean Squared Error (RMSE) on test data = 0.59
* Gradient-boosted Regression Root Mean Squared Error (RMSE) on test data = 0.55

Linear regression (train dataset- 70%) performed the best on our Supermarket Sales dataset.

RMSE = 0, R2 = 1, Mean = 15.41, Standard Deviation = 11.52

**Results of Tableau Visualization:**

Linear Regression results: Median = 12.09, Average = 15.38, R2 = 1

Below are the observations from the visualization done on the dataset.

1. Total Customers = 1000
2. Total Females = 501
3. Total males = 499
4. Min Rating = 4
5. Max Rating = 10
6. Average Rating = 6.97
7. Best Average Rating is for Food & Beverages
8. Max Average Gross Income in Home & Lifestyle
9. Min Average Gross Income in Fashion Accessories
10. Max Average total bill in Home & Lifestyle
11. Min Average total bill in Fashion Accessories
12. Maximum people pay through e-wallet
13. Maximum people comes from Yangon city
14. Max Average sales of Fashion Accessories is from Females
15. Max Average sales of Health & Beauty is from Males
16. **References:**

[https://www.codeitbro.com/install-pyspark-on-ubuntu-with-jupyter-notebook/#:~:text=Steps%20to%20install%20PySpark%20on%20Ubuntu,-PySpark%20is%20an&text=Step%201%3A%20Install%20Apache%20spark.&text=Step%202%3A%20Move%20the%20package,directory%20using%20these%20terminal%20commands.&text=Step%203%3A%20Now%20add%20JAVA\_HOME,path%20to%20the%20bashrc%20file](https://www.codeitbro.com/install-pyspark-on-ubuntu-with-jupyter-notebook/#:~:text=Steps%20to%20install%20PySpark%20on%20Ubuntu,-PySpark%20is%20an&text=Step%201%3A%20Install%20Apache%20spark.&text=Step%202%3A%20Move%20the%20package,directory%20using%20these%20terminal%20commands.&text=Step%203%3A%20Now%20add%20JAVA_HOME,path%20t).

<https://towardsdatascience.com/exploratory-data-analysis-using-spermarket-sales-data-in-python-e99d329a07fc>

<https://medium.com/@hellenwain_54279/supermarket-sales-361530767afa>

https://towardsdatascience.com/building-a-linear-regression-with-pyspark-and-mllib-d065c3ba246a

<https://sparkbyexamples.com/pyspark/pyspark-cast-column-type/>

<https://www.codegrepper.com/code-examples/python/cast+pyspark+column+to+float>

<https://www.geeksforgeeks.org/how-to-change-column-type-in-pyspark-dataframe/>

<https://sparkbyexamples.com/spark/spark-cast-string-type-to-integer-type-int/>

<https://towardsdatascience.com/beginners-guide-to-pyspark-bbe3b553b79f#:~:text=PySpark%20is%20a%20great%20language,a%20great%20way%20to%20start><https://spark.apache.org/docs/3.1.1/api/python/reference/api/pyspark.ml.feature.OneHotEncoder.html>

<https://towardsdatascience.com/machine-learning-linear-regression-using-pyspark-9d5d5c772b42>

<https://runawayhorse001.github.io/LearningApacheSpark/regression.html>

<https://medium.com/@hellenwain_54279/supermarket-sales-361530767afa>

<https://spark.apache.org/docs/latest/ml-classification-regression.html>

<https://www.datasciencemadesimple.com/pyspark-tutorial/>

<https://www.datasciencemadesimple.com/typecast-integer-to-decimal-and-integer-to-float-in-pyspark/>

<https://www.geeksforgeeks.org/how-to-change-column-type-in-pyspark-dataframe/>

<https://runawayhorse001.github.io/LearningApacheSpark/regression.html#decision-tree-regression>