# MSc in Data Science

## *Big Data Management*

***Project 1***

Zoi Papakonstantinou
*AM: dit2118dsc*
*Email: dit2118dsc@go.uop.gr*

Nikolaos-Marios Tsarouchas
*AM: dit2120dsc*
*Email: dit2120dsc@go.uop.gr*

# Contents

# Abstract

The purpose of this project is to study a dataset regarding customers who have visited a supermarket chain. The chain has physical and digital stores. The data queries had to be implemented either in Java or Pig. As a team we decided to process the data in Pig. In the first chapter, we analyze the data using python and search for various deficiencies, such as missing values, duplicate values, outliers, and other kind of noise. Additionally, we tried to process the data through the Pig but we used a lot of manual work to get the results, which is not consistent with the project. In the following chapters, we used Pig to answer the questions of the assignment. For each question, we wrote a Map / Reduce pseudocode and explained it through a schematic representation.


# 1. Exploratory Data Analysis

Data processing is a key factor applied to data collection, so we approached the data in two ways; Pig and Python. In Pig, we created the "Age" attribute to exclude customers over the age of 120 and dimensions for the IQR calculation to exclude outliers from income. On the other hand, we used ready-made packages to process the extreme values in Python, so we only excluded the extreme elements from Age.

### Pig

As mentioned above, we tried to create all the steps in Pig to clear the data from the outliers, regarding the age and the income, if necessary. Initially, we created features that calculate the age of the customer and excluded all customers who are over 120 years old. Consequently, we couldn't use the Pig functions to calculate the quartiles, so we created the attributes for each quartile. For the IQR calculation, we sorted the data (2.237 instances) by income in ascending order and we defined the Q1 (25% is the 559$^{th}$ instance) & Q3 (75% is the 1.678$^{th}$ instance). As outliers, we considered all the values that belong below Q1 – 1.5*IQR = -15.839 and above the Q3 + 1.5*IQR = 118.753. Observing the following results, we found that the income (6th column) is above the upper quarter, so the following results are considered outliers while for the lower quarter we did not have anything similar.

```
(2230,5555,1975,Graduation,Divorced,153924.0,0,0,7/2/21,81,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,47,Yes)
(2231,11181,1949,PhD,Married,156924.0,0,0,29/8/20,85,2,1,2,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,73,Yes)
(2232,4931,1977,Graduation,Together,157146.0,0,0,29/4/20,13,1,0,1725,2,1,1,0,0,28,0,1,0,0,0,0,0,0,0,45,Yes)
(2233,8475,1973,PhD,Married,157243.0,0,1,1/3/21,98,20,2,1582,1,2,1,15,0,22,0,0,0,0,0,0,0,0,0,49,Yes)
(2234,5336,1971,Master,Together,157733.0,1,0,4/6/20,37,39,1,9,2,0,8,0,1,0,1,1,0,0,0,0,0,0,0,51,Yes)
(2235,1501,1982,PhD,Married,160803.0,0,0,4/8/19,21,55,16,1622,17,3,4,15,0,28,1,0,0,0,0,0,0,0,0,40,Yes)
(2236,1503,1976,PhD,Together,162397.0,1,1,3/6/20,31,85,1,16,2,1,2,0,0,0,1,1,0,0,0,0,0,0,0,46,Yes)
(2237,9432,1977,Graduation,Together,666666.0,1,0,2/6/20,23,9,14,18,8,1,12,4,3,1,3,6,0,0,0,0,0,0,0,45,Yes)
```

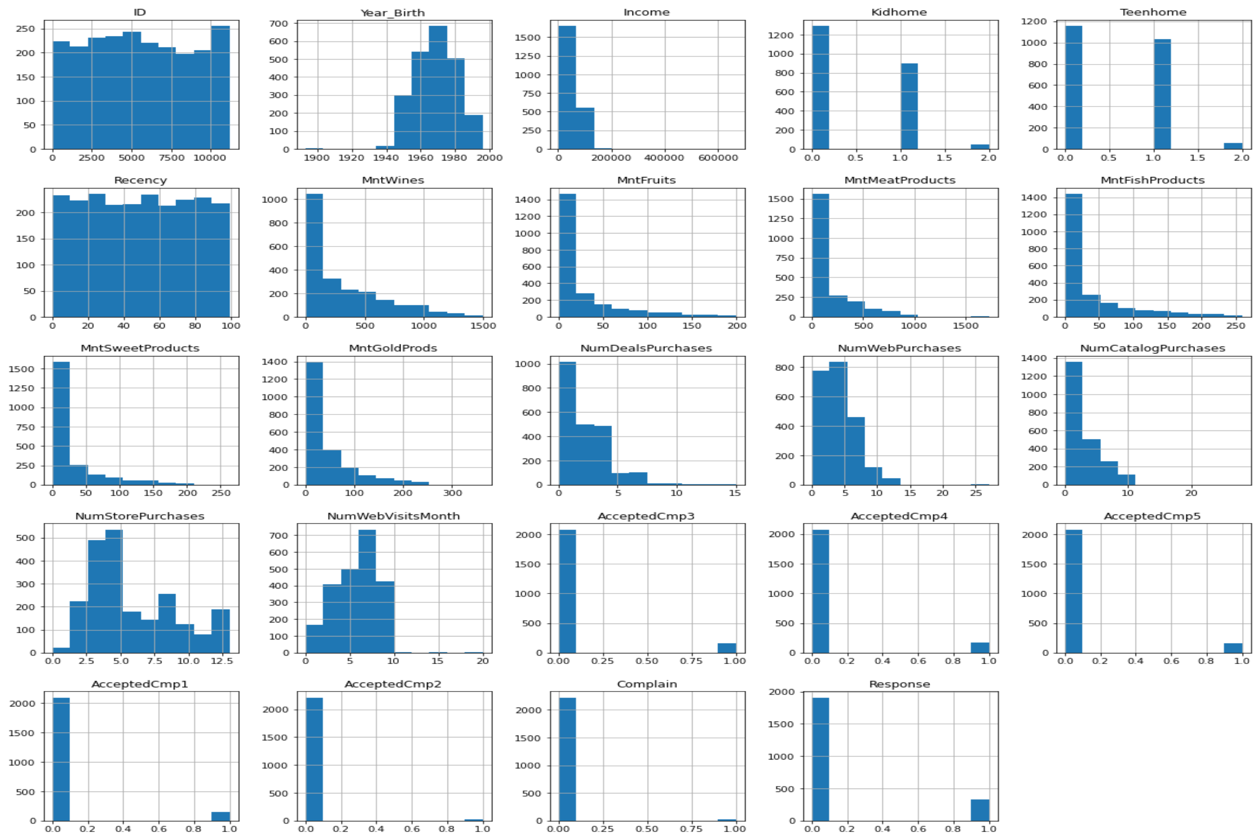
The commands can be found in the script "1strequest.pig" at GitHub .

## Python

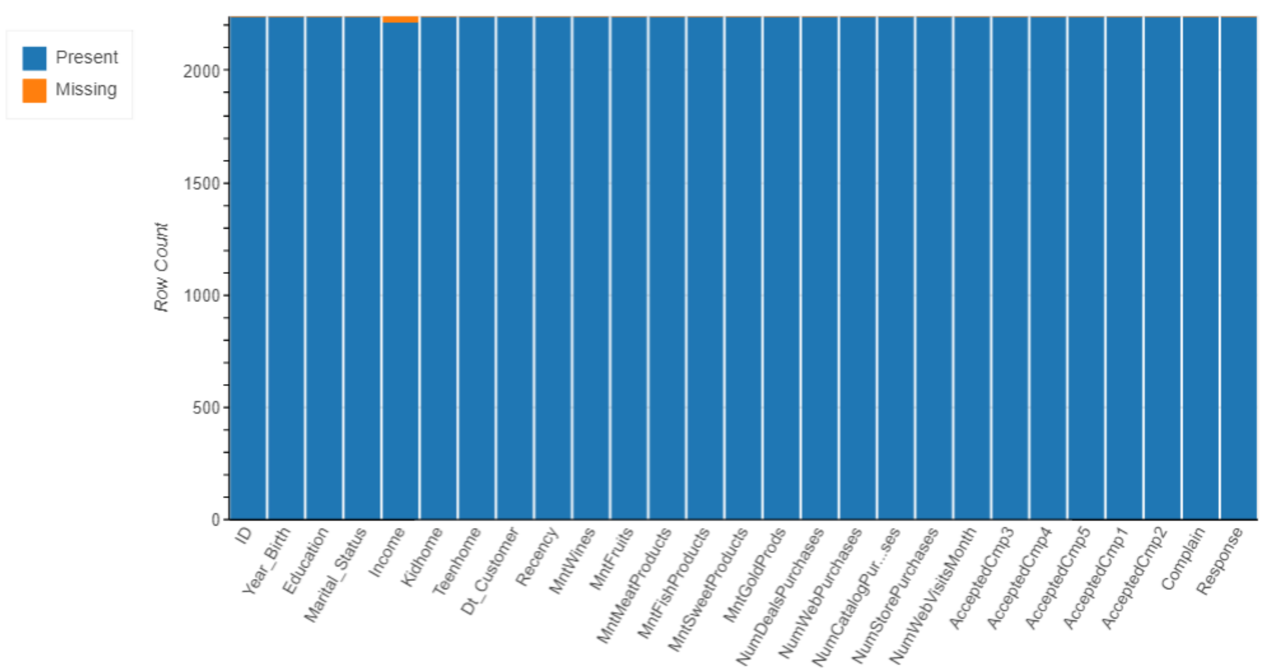
The dataset personality analysis consists of 2.240 instances of customer data. Each instance has 27 attributes out of which 15 are numerical and 12 categorical. The following matrix portrays the data type of each feature:

| *Feature* | Data type |
|---|---|
| *ID* | Numerical |
| *Year_Birth* | Numerical |
| *Education* | Categorical |
| *Marital_Status* | Categorical |
| *Income* | Numerical |
| *Kidhome* | Categorical |
| *Teenhome* | Categorical |
| *Dt_Customer* | Categorical |
| *Recency* | Numerical |
| *MntWines* | Numerical |
| *MntFruits* | Numerical |
| *MntMeatProducts* | Numerical |
| *MntFishProducts* | Numerical |
| *MntSweetProducts* | Numerical |
| *MntGoldProds* | Numerical |
| *NumDealsPurchases* | Numerical |
| *NumWebPurchases* | Numerical |
| *NumCatalogPurchases* | Numerical |
| *NumStorePurchases* | Numerical |
| *NumWebVisitsMonth* | Numerical |
| *AcceptedCmp3* | Categorical |
| *AcceptedCmp4* | Categorical |
| *AcceptedCmp5* | Categorical |
| *AcceptedCmp1* | Categorical |
| *AcceptedCmp2* | Categorical |
| *Complain* | Categorical |
| *Response* | Categorical |

The histograms of the feature are depicted in the figure below:



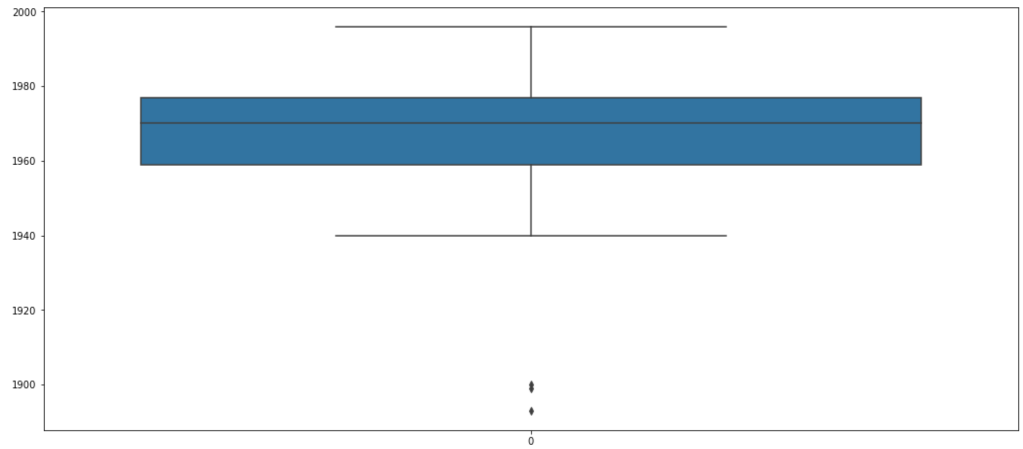

Based on the analysis performed there are 24 missing values in the dataset regarding the income feature.



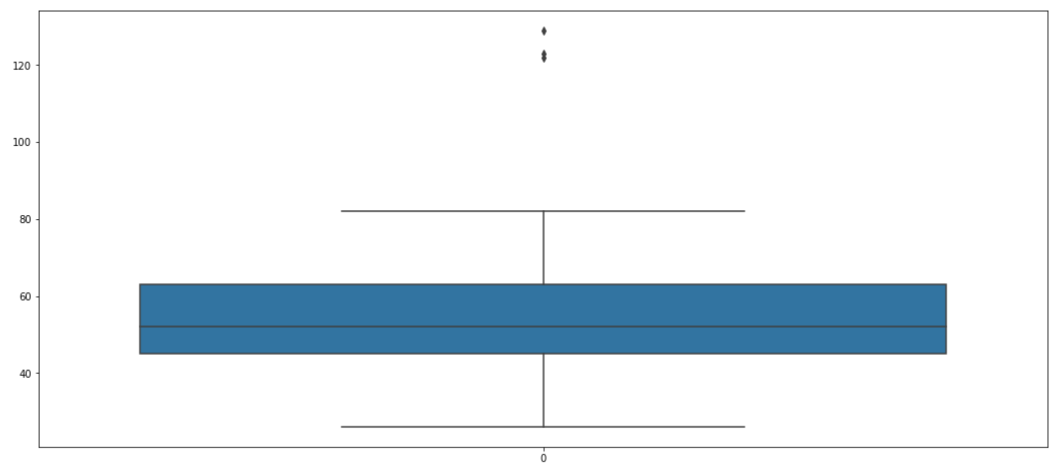

There were not any duplicated instances in the datasets.

Regarding the outliers of age and income we performed analysis via boxplots and the results can be seen below:

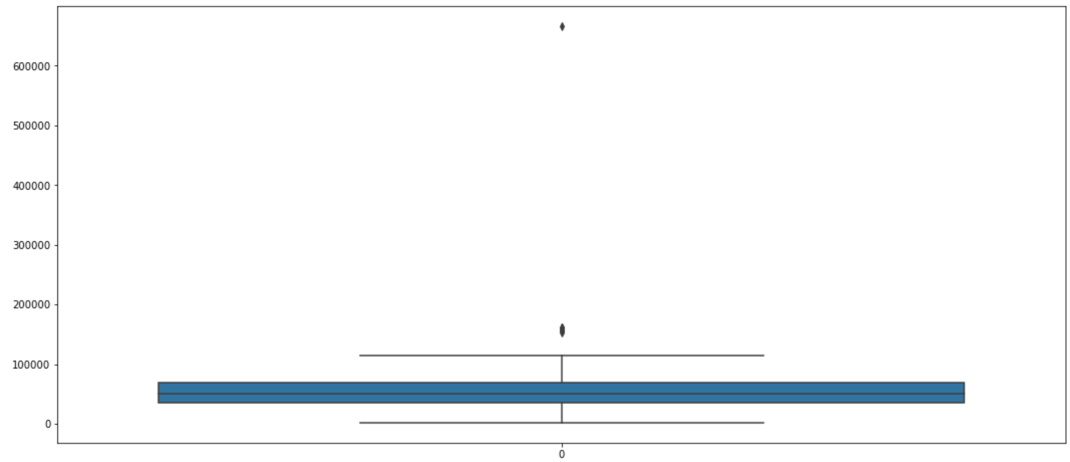
- For year of birth three instances are classified as outliers. The year of birth for these instances is 1893, 1899 and 1900. These customers are above 120 years old and more specifically 129, 123 and 122 respectively.




The above boxplot can be represented directly to age feature, which was created for the purposes of the assignment and can be seen below:




- As far as the Income is concerned 8 outliers were found. The boxplot below indicates that those customers annual income ranges from 153k to 667k.




Following the exploratory analysis and taking everything into consideration we decided to proceed to the following actions in the dataset:

- Omit the outliers regarding the age of the customers since it seems impossible to have three customers with age above 120.
- Remove the 24 instances with missing values in Income.
- Keep all the instances regarding the outliers detected for income, since it seems plausible for a customer to have annual income above 150k.
- Convert the feature Dt_Customer to datetime.

The jupyter notebook with the abovementioned analysis is available at GitHub. The dataset after the data cleansing was used for Pig queries and is available here.

In order to install and run the provided scripts a user manual is provided here.

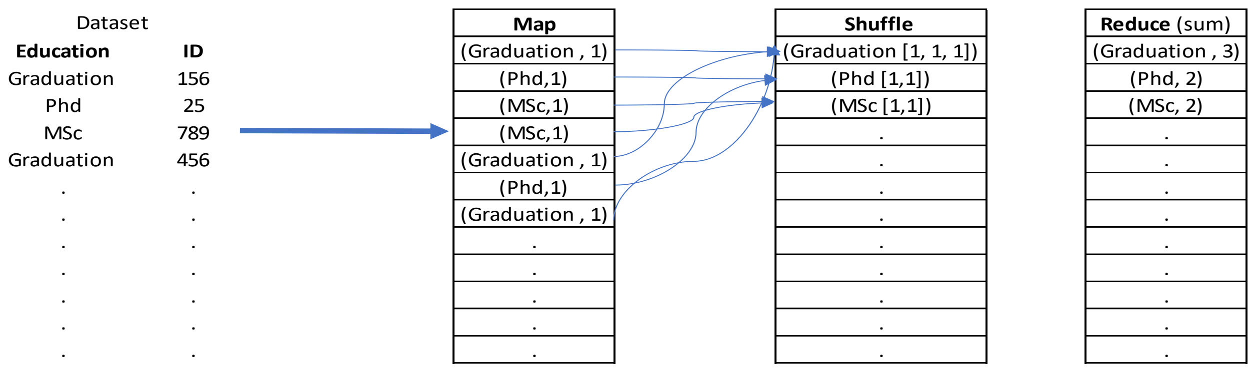
## 2. Customer Education Status ~ Question 1

After processing the data, we calculated the number of customers per education status. At first, we created a Map loop corresponding to "1" in each education status that appeared in our dataset. Based on the intermediate step of the Map/Reduce pseudocode we grouped the data (shuffle) and finally, the Reducer collected all the "1" depending on how many times they have appeared per category. To better understand the pseudocode, we constructed a schematic representation that shows us the intermediate step.

### ❖ Map Reduce

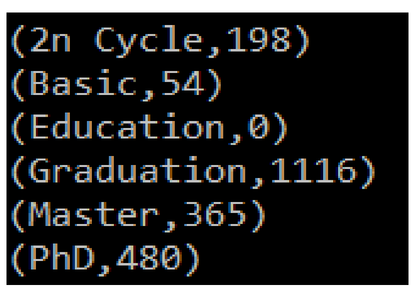
```
Class Mapper {
          method Map (String Education, int ID)
          for all strings s in Education:
                    emit (string s, count 1)
}

Class Reducer {
          method Reduce (string s, counts [c1, c2, ...])
          sum = 0
          for all counts c in [c1, c2, …]
                    sum = sum + c
                    emit (string s, count sum)
}
```

### ❖ Schematic representation




### ❖ The results of the Hadoop are shown in the figure below:

In addition to the pseudocode and schematic representation, we executed some commands in Pig to show the results we analysed above. It was observed that the education status "Graduation" has the most customers compare to the other categories. The commands can be found in the script "2ndrequest.pig" at GitHub.



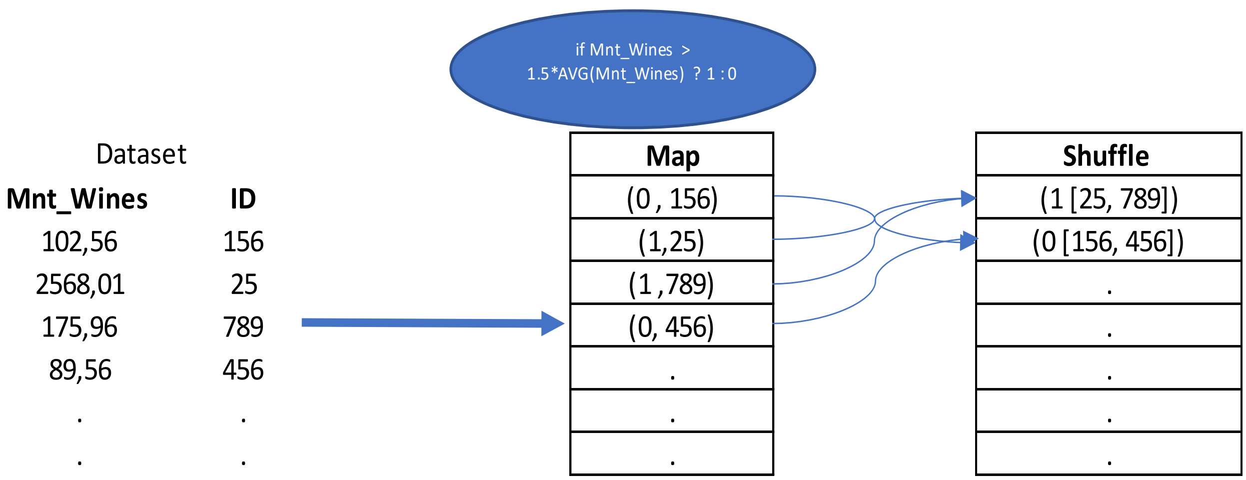

## 3. Wine buyers' prediction ~ Question 2

Unlike the previous question, we did not have to write code for the Reducer because the result was given to us directly from the Map. In the pseudocode Map, we created a variable that calculates average wine sales and added a loop that checks if the wine sales for each customer are 50% higher than the average.

## ❖ Map Reduce

```
Class Mapper {
        method Map (int Mnt_Wines, int ID)
        Mnt_Wines_Avg = AVG(Mnt_Wines)
        for all integer i in Mnt_Wines:

                if Mnt_Wines > 1.5*Mnt_Wines_Avg:
                        int variable = 1
                else:
                        int variable = 0

                        emit (variable, ID)
}
```

## ❖ Schematic representation




## ❖ The results of the Hadoop are shown in the figure below:

The purpose of this task was to manage the data so that we could export the results in order. For example, some customers spent the same amount on wine but had a different income and often a different educational background. The commands can be found in the script "3rdrequest.pig" at GitHub.

```
(1,737,73,PhD,Married,80360.0,1493)
(2,3174,63,Graduation,Together,87771.0,1492)
(3,5536,63,Graduation,Together,87771.0,1492)
(4,1103,46,Master,Married,81929.0,1486)
(5,5547,40,PhD,Married,84169.0,1478)
(6,8362,40,PhD,Married,84169.0,1478)
(7,3009,60,PhD,Widow,71670.0,1462)
(8,1665,58,PhD,Divorced,64140.0,1459)
(9,9743,67,Graduation,Married,76998.0,1449)
(10,11088,51,PhD,Together,78642.0,1396)
(11,4580,53,Graduation,Married,75759.0,1394)
(12,4943,69,Graduation,Married,70503.0,1379)
(13,9260,77,PhD,Married,70356.0,1349)
(14,7431,31,PhD,Single,68126.0,1332)
(15,3138,66,Graduation,Single,91249.0,1324)
(16,4475,73,PhD,Married,69098.0,1315)
(17,6292,36,PhD,Married,82333.0,1311)
(18,10140,39,PhD,Together,70123.0,1308)
(19,10133,52,Graduation,Single,93790.0,1302)
(20,8732,53,Master,Widow,67369.0,1298)
(21,8545,68,Graduation,Divorced,85683.0,1296)
(22,203,47,Master,Single,81169.0,1288)
(23,6932,81,PhD,Married,93027.0,1285)
(24,7919,46,PhD,Together,72335.0,1285)
(25,7962,35,PhD,Single,95169.0,1285)
```

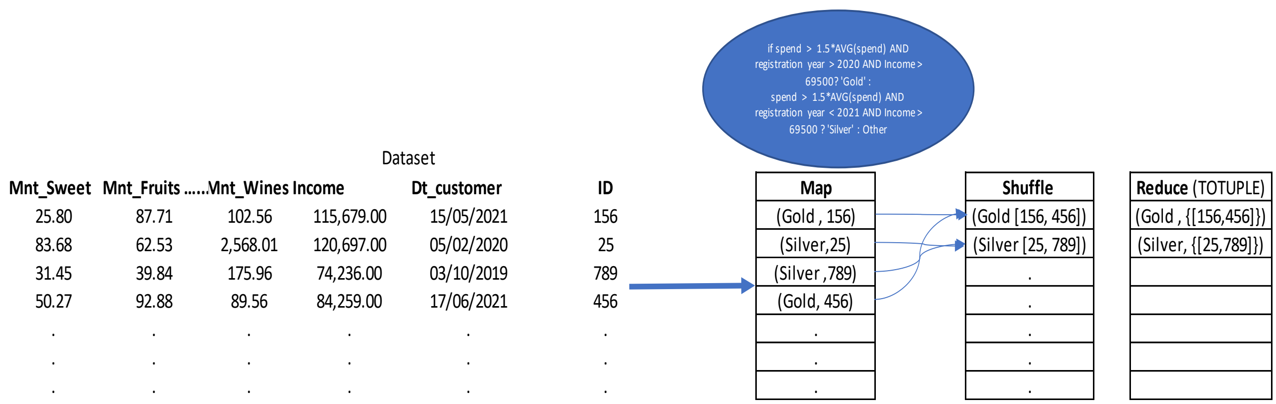
```
(611,4390,68,Graduation,Together,75315.0,459)
(612,5675,62,PhD,Divorced,50611.0,459)
(613,7230,62,PhD,Divorced,50611.0,459)
2022-05-22 02:18:43,294 [main] INFO  org.apache.
```

## 4. Customer Categorization ~ Question 3

Customer categorization in a company is a very important part as you can create targeted actions or increase its sales. In this case, we had to categorize the customers in Cold & Silver. In the Map/Reduce pseudocode, we defined three variables "spend", "average spend" and "Year", each calculating the total amount in all mnt columns, the average of the above, and the year the customers received ID in the company, respectively. Next, we created a loop that identifies customers according to the criteria. Customers who are registered in the last year, have an income of over 69,500 and spend more money than the average of the categories, while silver has the same criteria, the only change is in the year of registration, which must be more than one year in company. To display the results for each status, we made a table that classifies the customer's identity according to the result that will come out in the loop above. For a better understanding of the result, we used the schematic execution that shows the steps performed by the pseudocode.

## ❖ Map Reduce

```
Class Mapper {
          method Map (int ID, Columns[int Income, int MntSweets…int Mnt_Wines, datetime Dt_Customer
])
          spend = MntWines + MntFruits + MntMeat + MntFish+ MntSweet+ MntGold
          avg_spend = AVG(MntWines + MntFruits + MntMeat + MntFish+ MntSweet+ MntGold)
          registrationyear = GetYear(Dt_Customer)

          for all id in ids[id1,id2,..]:
                    for all id in ids:
                              if spend > avg_spend AND GetYear(Dt_Customer)>2020 AND Income>69500:

                                        Class = 'Gold'

                              if spend > avg_spend AND GetYear(Dt_Customer)<2021 AND Income>69500:
                                        Class = 'Silver:
                              emit (Class, ids)
}

Class Reducer {
          method Reduce (Class, ids [id1, id2, ...])
          gold_ids = []
          silver_ids =[]
          for all id in [id1, id2, …]:
                    if Class = 'Gold':
                              gold_ids = ids.append(id)
                    if Class = 'Silver:
                              silver_ids = ids.append(id)
                    emit (Class, ids)
}
```




## ❖ The results of the Hadoop are shown in the figure below:

The following results show the classification of customers by status as generated by the code in Pig. The commands can be found in the script "4rthrequest.pig" at [GitHub](GitHub).

```
(Gold,({(202),(437),(477),(486),(830),(988),(1020),(1127),(1139),(1225),(1277),(1340),(1371),(1399),(1446),(1577),(1592)
,(1619),(1826),(2008),(2186),(2225),(2410),(2532),(2591),(2669),(2781),(2831),(2909),(2939),(2963),(3005),(3010),(3194),
(3334),(3434),(3595),(3711),(4149),(4248),(4310),(4331),(4394),(4676),(4679),(4686),(4767),(4964),(4998),(5046),(5067),(
5303),(5350),(5545),(5558),(5721),(5735),(5832),(5848),(6036),(6343),(6384),(6428),(6521),(6652),(6931),(6963),(7010),(7
059),(7106),(7321),(7348),(7451),(7723),(7861),(7875),(7959),(8029),(8093),(8318),(8475),(8504),(8545),(8659),(8746),(89
08),(8931),(9206),(9264),(9298),(9369),(9400),(9589),(10037),(10102),(10129),(10133),(10164),(10286),(10413),(10469),(10
548),(10767),(10909),(10925),(10955),(10972),(10981),(11074)}))
(Silver,({(0),(146),(158),(175),(203),(241),(257),(291),(295),(313),(375),(454),(460),(500),(569),(590),(697),(701),(737
),(821),(1000),(1031),(1065),(1071),(1079),(1081),(1103),(1137),(1168),(1172),(1173),(1215),(1232),(1357),(1361),(1411),
(1501),(1524),(1544),(1553),(1570),(1628),(1650),(1745),(1763),(1772),(1859),(1891),(1915),(1945),(1964),(1966),(2002),(
2004),(2075),(2079),(2088),(2109),(2114),(2147),(2150),(2176),(2181),(2254),(2324),(2345),(2429),(2565),(2611),(2612),(2
631),(2674),(2730),(2793),(2797),(2798),(2802),(2886),(2894),(2931),(2975),(3009),(3074),(3104),(3138),(3139),(3174),(31
79),(3254),(3403),(3422),(3426),(3433),(3503),(3507),(3520),(3523),(3560),(3619),(3661),(3667),(3690),(3698),(3725),(374
9),(3762),(3828),(3852),(3859),(3867),(3910),(3919),(3967),(3979),(4002),(4070),(4114),(4127),(4137),(4207),(4216),(4261
),(4278),(4299),(4322),(4356),(4418),(4427),(4483),(4501),(4507),(4518),(4530),(4543),(4580),(4597),(4611),(4637),(4640)
,(4646),(4654),(4673),(4687),(4786),(4789),(4837),(4856),(4931),(4943),(4947),(4974),(4994),(5031),(5077),(5080),(5113),
(5114),(5125),(5147),(5153),(5204),(5236),(5302),(5331),(5332),(5341),(5386),(5453),(5536),(5538),(5547),(5687),(5718),(
5796),(5830),(5831),(5955),(5989),(6024),(6049),(6055),(6071),(6072),(6086),(6173),(6203),(6246),(6248),(6260),(6262),(6
274),(6292),(6303),(6320),(6406),(6417),(6421),(6422),(6565),(6566),(6583),(6606),(6694),(6721),(6749),(6810),(6853),(69
05),(6906),(6932),(6935),(6945),(6974),(6977),(6982),(6988),(7022),(7053),(7118),(7215),(7274),(7290),(7313),(7366),(738
1),(7396),(7428),(7441),(7503),(7521),(7627),(7734),(7789),(7832),(7849),(7851),(7899),(7919),(7960),(7962),(7966),(7972
),(7999),(8015),(8164),(8204),(8278),(8314),(8362),(8370),(8395),(8492),(8537),(8584),(8601),(8602),(8624),(8643),(8690)
,(8707),(8717),(8722),(8895),(8923),(8925),(8969),(9010),(9029),(9058),(9121),(9145),(9185),(9209),(9212),(9220),(9260),
(9265),(9274),(9292),(9305),(9349),(9451),(9463),(9493),(9495),(9499),(9504),(9529),(9560),(9606),(9743),(9757),(9790),(
9799),(9826),(9904),(9930),(9938),(9952),(9999),(10057),(10089),(10092),(10095),(10140),(10150),(10156),(10163),(10177),
(10314),(10383),(10394),(10424),(10430),(10479),(10486),(10489),(10505),(10590),(10619),(10678),(10736),(10814),(10868),
(10936),(10949),(10965),(10983),(10992),(11056),(11071),(11088),(11112)}))
2022-05-22 15:32:44,575 [main] INFO  org.apache.pig.Main - Pig script completed in 5 seconds and 750 milliseconds (5750
```

## Conclusion

Nowadays, the customer approach has been changed in relation with the past. Customer personality analysis helps a company modify its products based on customers' preferences instead of spending money on bulk product promotions. A quite important part of the company is the profile analysis of each customer. The purpose of the analysis is the promotion of products the customer prefers and suggestion of others, according to his/her buying habits.

For the afore mentioned process, the usage of Pig language is suitable. Additionally, we consider it to be a very practical language that helps to quickly extract the results. The main part of the project was to figure out which key we would use in each case to get the desired results. In order to approach the results, we cleaned our data from the outliers in age and income and then with different commands depending on the case we extracted the corresponding result.